004
Б79

Б. БӨРІБАЕВ

С/С++ ТІЛДЕРІНДЕ ПРОГРАММАЛАУ

практикалық курс

ЭВЕРО

004
Б79

Б. БӨРІБАЕВ

С/С++ ТІЛДЕРІНДЕ ПРОГРАММАЛАУ

(практикалық курс)

Жоғары оқу орындарына арналған оқу құралы

ЭВЕРО
Алматы – 2015
Эверо

004.438(075.8)

УДК 004.4 (075.8)
ББК 32.973-018.1 я 73
Б 79

Пікір берушілер: техника ғылымдарының докторы,
профессор **Сейкетов У. Ә.**
техника ғылымдарының докторы,
доцент **Жомартова Ш.Ә.**

Б 79 Бөрібаев Б. C/C++ тілдерінде программалау (практикалық курс):оқу құралы. - Алматы: Эверо, 2015. -320 б.

ISBN 978-601-240-388-6

Бұл оқу құралы жоғары оқу орындарындағы жаратылыстану саласында немесе техникалық бағытта білім алып жатқан студенттерге «Алгоритмдеу және программалау тілдері», «Алгоритмдер және мәліметтер құрылымы» және «Программалау технологиялары» тәрізді пәндерді оқу кезінде теориялық білімдерді игеру мен зертханалық жұмыстар орындауды жеңілдетуге арналған. Мұнда C/C++ тілдерін негізге ала отырып алгоритм құру және программалау тәсілдерін үйретуден практикалық мағлұматтар беріліп, әрбір тақырып бойынша жинақталған есептердің шығарылу жолдары толық көрсетілген. Зертханалық жұмыстарда студенттердің өз беттерімен орындауларына арналған тапсырмалар келтірілген. Ұсынылып отырған оқу құралы программалауды өз бетінше оқып үйренгісі келетін оқырмандардың да қажетіне жарайды деген сенімдеміз.

688041

С.Торайғыров
атындағы ПМУ-дің
академик С.Бейсембаев
атындағы ғылыми
КІТАПХАНАСЫ

УДК 004.4 (075.8)
ББК 32.973-018.1 я 73

ISBN 978-601-240-388-6

© Б. Бөрібаев, 2015
© Эверо, 2015

КІРІСПЕ

Біз күнделікті өмірде белгілі бір әрекеттер орындау барысында көптеген алгоритмдерді қолданып жүрміз. Міне осы тұрғыдан алғанда, өмір деген алда тұрған мәселелерді шеше алатын алгоритмдер жиыны деп түсінуге болады. Сол себепті соларды жүйелі түрде оқып, келешекте пайдалану тәсілдерін үйренуіміз қажет. Мысалы, кез келген формуланы есептеу, квадрат теңдеуді шешу, жиымдарды өңдеу алгоримдері, т.с.с.

Бұл оқу құралы өмір талабына сай жазылып, жаратылыстану ғылымдары мен техникалық салада білім алу барысында жеке оқу пәндері болып қалыптасқан *Алгоритмдер және мәліметтер құрылымы*, *Алгоритмдеу және программалау тілдері*, *Программалау технологиялары*, *Жүйелік программалау* тәрізді сабақтарда қолдану үшін шығарылып отыр. Мұнда алгоритмді құрастыру, жазу жайлы түсініктер беріліп, қазіргі кездегі кең тараған программалау тілдері болып саналатын C/C++ ерекшеліктері қарастырылып, алгоритмдерді компьютерде программалап, соларды орындау шаралары жүзеге асырылады.

Өткен ғасырдың 80-ші жылдары басында UNIX операциялық жүйесінің басқаруымен жұмыс істейтін компьютерлер әлемінде пайда болған тілдің бірі С тілі болды. Содан бері ол дербес компьютерлер мен мэйнфреймдерде (үлкен ЭЕМдер) жұмыс істейтін негізгі тілдердің біріне айналды. Программалық жабдықтамалармен айналысатын көптеген фирмалар бұл тілді мәтіндік процессорларды, электрондық кестелерді, компиляторларды құру үшін кең қолданып келеді. Соңғы кездерде C/C++ тілі ең көп таралған тілдердің біріне айналды.

С тілінде программалауды үйрену компьютерлік технологияларды игергісі келетін әрбір маманның алғашқы қадамдарының бірі болуы тиіс. Олай дейтініміз C/ C++ және осыларға ұқсас тілдер қазіргі кезде үздіксіз даму үстінде. Сол себепті назарларыңызға ұсынылып отырған оқу құралының алдына қойған мақсаты студенттерді тиімді де қысқа мәтінді программалар жазуға және олардың нәтижелерін алып, түрлендіру істеріне машықтандыру болып табылады. Ол тұтынушының программалау дәрежесінің кез келген деңгейіне арналған. Программалауды енді үйрене бастаған студенттер үшін ол оқу құралы болса, тәжірибелі программалаушылар үшін анықтамалық құрал ретінде пайдаланылуы мүмкін.

Кітапта теориялық мағлұматтардың көптеген нақты мысалдар мен жаттығулар арқылы түсіндіріліуі кітаптың негізгі бір ерекшелігі деуге болады.

Әрбір студент программа құрастыру кезінде кеткен қателерді жөндеп, компьютер арқылы дұрыс нәтиже алуы үшін алдын ала мәліметтерді сипаттап, олардың типін ажыратып, қажетті операторлар тізбегін жаза білуі керек.

Сонымен қатар жасалған программалар мәтіні пернетақтада C/C++ біріктірілген программалық жүйесінің редакторы арқылы теріліп, компьютер жадында, кейіннен дискілерде сақталып, әр адам жасаған ақпараттар архивіне айналуы тиіс.

Программалар тегін таратылатын Turbo C, Borland C компиляторларында орындалып, нәтижелері келтірілген, бірақ оларды өзгеріссіз (немесе аздап өзгерту арқылы) MS Visual C++ 6.0, Microsoft Bisual Studio 2010 сияқты программалау орталарында да орындай беруге болады. Осы компиляторларды пайдаланып,

әрбір студенттің өзі жасаған программаларының керектілерін файлдарда сақтап, принтер арқылы қағазға шығарып, әр межелік бақылау сайын атқарылған жұмыстарын жинақтап отырғаны жөн болады.

Программалық орта ретінде Turbo C жүйесінің таңдалып алынуына негізгі себеп – C/C++ тілдерін оқып үйрену жоғары оқу орнының бірінші курсында басталады, студенттер мұның алдында, мектеп қабырғасында кейде университетте де Паскаль тілін Turbo Pascal жүйесі негізінде оқып үйренеді. Сондықтан программаны теру және орындау ортасы Turbo Pascal және Turbo C ортасында бірдей болғандықтан, студенттер компилятор ерекшеліктерін игеруге уақыт кетірмей, бірден жаңа тілді үйрене бастайды.

Бұл оқу құралында компьютер көмегімен программа құру, график тұрғызу, оларды экранға (қағазға) басып шығару жолдары, C/C++ тілдерінде жазылған программаны машина тіліне аударып объектілі модуль жасау тәсілдері де көрсетіледі.

Қазіргі кезде барлық елдерде ең кең тараған программалау тілдеріне осы C/C++ тілдері жатады деп айтуға болады. Олай болса, егемендік алған еліміздің жас талапкерлері де программалау тілінің, алгоритмдердің терминдерін ана тілінде оқып үйренуі қажет.

Информатика терминдері ағылшын тілінде қалыптасып, одан кейін басқа тілдерге аударылып келеді. Мемлекеттік тілімізге әлі информатика терминдері толық аударылып, олардың стандарты бекітілмегендіктен, кітап соңында қолданылған терминдердің бірсыпыра аудармасы келтірілген. Кейбір атаулар түсініксіз болған жағдайда қосымшалардағы глоссарийден оның негізгі мағынасын біліп алуға болады.

Қазақ тілінде жарық көріп отырған бұл оқу құралы төл тілімізде дәріс алып жатқан студенттеріміз үшін программалау негіздерінің қыры мен сырын меңгеру жолында көмекші құрал бола алады деген ойдамыз. Бұған дейін де қазақ тілінде C/C++ тіліне арналған бірнеше оқулықтар мен оқу құралдары шыққан болатын. Ізденгіш талапкерлерге олармен танысқанның да артықтығы болмайды.

Сөз соңында бұл оқу құралы программалау тілін өздігінен үйренушілерге де пайдалы болатынын айтқымыз келеді. Кітап материалын игерген студент өздігінен күрделілігі орташа программаларды құрастыра алады деген сенімдеміз.

1 АЛГОРИТМДЕР ЖӘНЕ ПРОГРАММАЛАР. СЫЗЫҚТЫҚ ЖӘНЕ ТАРМАҚТЫ АЛГОРИТМДЕРДІ ПРОГРАММАЛАУ

1.1 Алгоритмдерді жазу жолдары

Әрбір компьютер берілген алгоритммен, яғни жоспармен жұмыс істейді. Алгоритмді заңдылық, реттелген амалдар жиыны, кезекпен орындалатын операциялар тізімі деп ұғынған жөн. Бұл ұғым қазіргі кезде кеңінен қолданылып жүр.

Алгоритм – берілген есептің шығару жолын реттелген амалдар тізбегі түріне келтіру. Кез келген есепті қарапайым амалдарды тізбектей орындау арқылы шығаруға болады. Алгоритмді компьютерде орындау үшін оны программа түрінде жазып шығу керек.

Программа – алгоритмді машинаға түсінікті нұсқаулар тізімі ретінде жазу. Программа машинаға түсінікті командалардан тұрады. Осы командалар тізбегі орындалу барысында есептің нәтижесі шығады. Әрбір компьютер алдын ала жазылған программамен істейді. Алгоритмдерді компьютерде орындау үшін оларды алдын ала жазып алу керек, яғни ол белгілі бір заңдылықпен өрнектелуі тиіс. Жалпы алгоритмді жазып өрнектеу түрлеріне:

1) табиғи тіл арқылы жазу;

2) графика жолымен жазу;

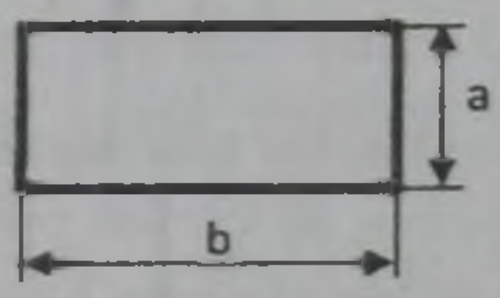
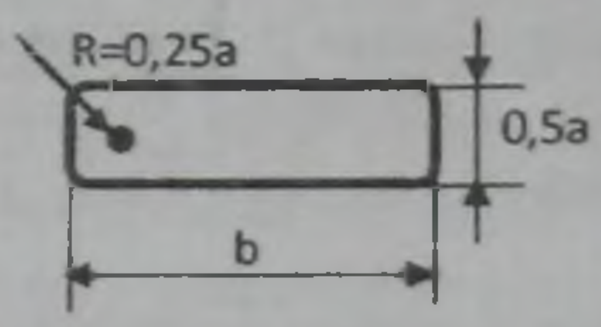
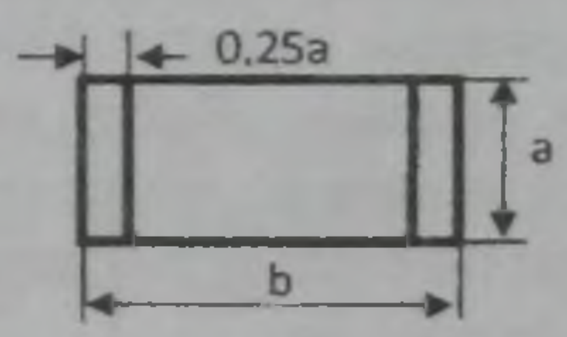
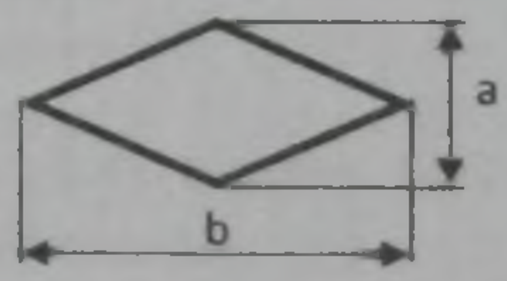
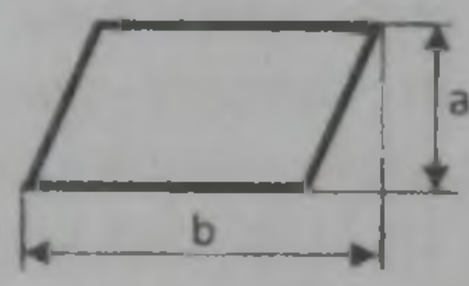
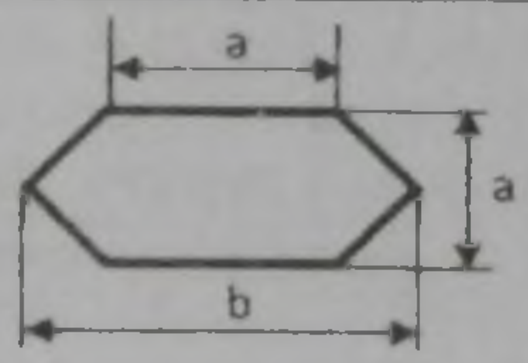
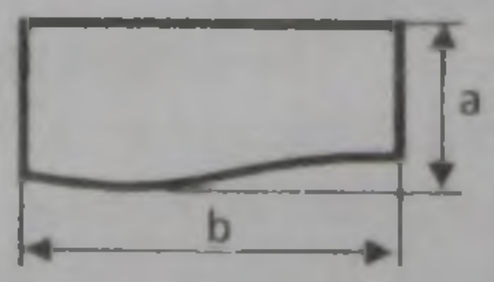
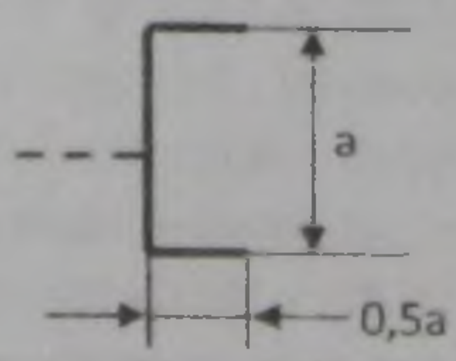
3) алгоритмдік тілдермен жазу жолдарын жатқызуға болады.

Бірақ алгоритмді табиғи тілде өрнектеу компьютерлерде сирек колданылады, өйткені онда дәлдік, нақтылық болмайды.

Алгоритмдерді графика жолымен жазу, онан кейін оны программаға айналдыру істері мемлекеттік стандартпен бекітіліп ақпарат өңдеу жұмысында кеңінен қолданылып келеді. Алгоритмдік, яғни программалау тілдері есептерді шығару жолын баяндау-өрнектеу үлгісі, белгілі бір проблеманы шешу негізінде орындалатын әрекеттерге басшылық, ой еңбегін үнемдеуге мүмкіндік беретін әдіс, есеп шешімін табуды автоматтандыруға қажетті іс-әрекет, жаңа проблеманы шешу кезінде қолданылатын тәсілдер, күрделі процестерді өрнектеу және математикалық дәлдікпен анық етіп жазу құралы бола алады. Бұл оқу құралындағы программалар С және С++ тілдерінде жазылған.

1.2 Алгоритмдерді графикалық түрде жазу

Алгоритмдерді өрнектеудің көп тараған түрі – оны график аркылы бейнелеу. Ал бұл – түсінікті, анық, көрнекті түр болып есептеледі. Тек оларды сызу көбірек еңбекті талап етеді. Графикалық жолмен алгоритмдерді жазу үшін мемлекеттік стандарт белгіленген, онда кез келген амал белгілі бір геометриялық фигурамен өрнектеледі. Ол фигуралар немесе блоктар амалдар символы деп те аталады. Блоктар бағытталған сызықтармен байланысып, бірінен соң бірі орналасады. Жиі колданылатын амалдар, яғни мәліметтерді программаға енгізу, формуламен есептеу, шарттардың орындалуын тексеру, нәтижені қағазға басу символдары 1.1-кестеде көрсетілген. Осы суреттегі көрсетілген блоктардан (символдардан) алгоритм схемалары құрастырылады. Алгоритмдер схемасымен ақпаратты өңдеудің әрбір сатысы немесе орындалатын операциялар реті анықталады. Кейде алгоритмдер схемасын оның блок-схемасы деп те атайды.

Іс-әрекеттің аты	Блоктың пішімі	Атқаратын жұмысы
Процесс		Математикалық өрнектерді есептеу
Басы – соңы		Алгоритмдерді бастау, аяқтау
Алдын ала анықталған процесс (ішкі программа)		Қосалқы программаларға кіру және шығу
Шешім қабылдау		Есеп шығару жолын таңдау
Енгізу-шығару		Мәліметтерді енгізу және шығару
Модификация		Цикл басы
Құжат		Нәтижені баспаға (қағазға) шығару
Түсініктеме		Схеманы, формулаларды түсіндіру

1.1 сурет. Алгоритмдерді бейнелеу блоктары

Сонымен алгоритм блоктармен немесе геометриялық көпбұрыштар түріндегі фигуралармен өрнектеледі. Әр блоктың ішіне орындалатын іс-әрекеттің (амалдың) мазмұны жазылады. Символдардың (блоктардың) бір кіру және бір шығу сызықтары болуға тиіс.

Алгоритмдер блоктардың өзара байланысуына қарай үш түрлі бірыңғай құрылымға – сызықтық, тармақтық және циклдік болып үш топқа бөлінеді. Енді соларды карастырайық.

1.3 Алгоритмдердің бірыңғай құрылымы

Күрделі алгоритмдерді құру үшін қарапайым бірыңғайланған алгоритмдік құрылымдар қолданылады. Олар сызықтық, тармақталу және цикл құрылымдарынан тұрады.

Программалау теориясында кез келген күрделі программаны үш түрлі құрылымнан құрастыруға болатыны дәлелденген, олар: *сызықтық*, *тармақты* және *циклдік* құрылымдар. Осы үшеуі құрылымдық программалаудың негізгі конструкциялары, яғни құраушылары болып саналады (1.2 сурет).

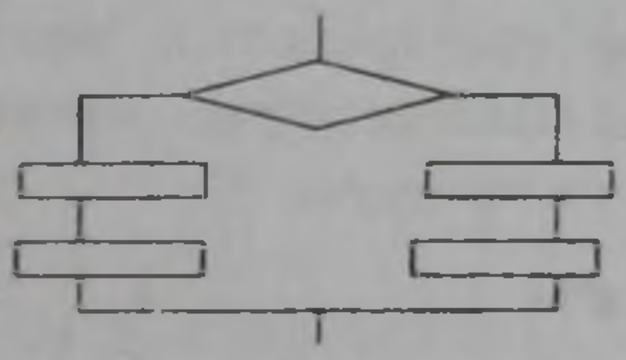
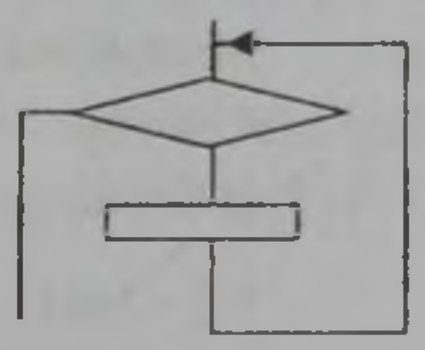
Сызықтық құрылым бірінен кейін бірі орындалып тізбектеле орналасқан бірнеше операторлардан тұрады.

Тармақты – шартқа байланысты екі оператордың бірінің орындалуы.

Цикл – операторлар бөлігінің бірнеше рет қайталана орындалуы.

Оператор – тілдің қарапайым сөйлемі, ол белгілі бір әрекет немесе амал орындап, ; таңбасымен аяқталады.

Негізгі конструкцияларды пайдалану максаты – қарапайым құрылымды программа алу болып саналады. Мұндай программалар оңай оқылады, түзетіледі және керек болса, оңай өзгертіледі. Құрылымдық программалауда **goto** операторын колдануға болмайды, өйткені ол программа логикасын түсінуді киындатады. Бірақ кейде **goto** операторын колдану қажет болатын кездер болады.

Сызықтық құрылым	Тармақты құрылым	Циклдік құрылым
		

1.2 сурет. Алгоритмдердің бірыңғай (типтік) құрылымдары

Программа жұмысын басқару операторларын программаның басқарушы конструкциясы деп атайды. Олар:

- құрама операторлар;
- таңдау операторлары;
- цикл операторлары;
- көшу операторы.

1.4 Мәліметтерді енгізу және шығару

C++ тілінде арнаулы енгізу және шығару құралдары жоқ – ол стандартты кітапханада орналасқан функциялар, типтер және объектілер көмегімен іске асады. Осы бағытта қолданылатын негізгі екі тәсіл бар, олар: С тілінің функциялары және C++ объектілері.

С тілі стилінде мәліметтерді енгізу/шығару үшін `stdio.h` кітапханалық файлында сипатталған төмендегі функциялар пайдаланылады.

1) `printf(формат тіркесі, аргументтер тізімі);`

формат тіркесі – қостырнақшамен қоршалып, аргументтердің қалай (экранда) бейнеленетінін көрсететін символдар тіркесі. Мысалы:

```
printf("Пи саны мәні = %f\n", M_PI);
```

Формат тіркесінде мыналар болуы мүмкін:

а) мәтін ретінде шығарылатын символдар;

ә) түрлендіру спекцификациялары;

б) басқару символдары.

Әрбір аргуменке төмендегі спецификациялардың біреуі сәйкес келуі тиіс:

`%d, %i` – бүтін оңдық сан;

`%f` – жылжымалы нүктелі оңдық сан;

`%e, %E` – экспоненциалдық түрдегі жылжымалы нүктелі сан;

`%u` – таңбасыз оңдық сан;

`%c` – символ;

`%s` – сөз тіркесі.

Формат тіркестері ішінде басқару символдары да бола береді, олар:

`\n` – келесі жана жолға көшуді атқаратын басқару символы;

`\t` – көлденен табуляция;

`\a` – дыбыстық сигнал, т.б.

Бұларға қоса формат тіркестері құрамында форматтау модификаторлары қолданылуы мүмкін, олар шығарылатын мәннің ұзындығына сәйкес өріс енін тағайындайды. Модификаторлар – бұлар шығарылатын мәнге және саннын бөлшегіне берілетін орындардың ең кіші (минималды) мөлшерін көрсететін сандар, олар былай беріледі:

`%[-]m[.p]C`, мұндағы

– (минус) таңбасы – шығарылатын мәнді сол жақ шеткс туралау керек екендігін, m – өрістің минималды енін (бүтін сандар ені осы m санынан аз болса, онда ол бос орындармен толтырылып жазылады),

p – жылжымалы нүктелі сандар үшін үтірден кейінгі цифрлар санын (бөлшек ені p-дан аз болса, оның оң жағы нөлдермен толықтырылады), бүтін сандар үшін шығарылатын цифрлардың ең төменгі санын (сандағы цифрлар саны p-дан аз болса, оның сол жағы нөлдермен толықтырылады),

C – шығарылу форматы спецификациясына сәйкес символды көрсетеді.

Мысалы:

```
printf("Пи саны мәні = %f\n", M_PI);
printf("\nФормат спецификациялары:\n%10.5d – бүтін,\
\n%10.5f – жылжымалы нүктелі,\n%10.5e – экспоненциалды түрде,\
\n%10s – сөз тіркесі",10,10.0,10.0,"10");
```

Осының нәтижесінде экранға мыналар шығарылады:

Пи саны мәні = 3.141593
Формат спецификациялары:
00010 - бүтін, // сол жақта 5 босорын
10.00000 - жылжымалы нүктелі, // сол жақта 2 босорын
1.00000e+01 - экспоненциалды түрде, // сол жақта босорын жок
10 – сөз тіркесі // сол жақта 8 босорын

2) `scanf(формат тіркесі,аргументтер тізімі);`

мұндағы аргументтер ретінде айнымалылар адрестері пайдаланылады. Мысалы:

```
scanf("%d%f",&x,&y);
```

1.5 C тілінде мәліметтерді форматталған түрде шығару

Мәліметтерді форматталған түрде шығару үшін `printf()` функциясы пайдаланылады. Ол бүтін және нақты сандарды, символдарды, сөз тіркестерін немесе осы типтердің араласқан түрлерін де шығарады. Тек сөз тіркестерін ғана шығару кезінде `puts()` функциясын қолдануға болады.

Келесі программа осы функциясының әртүрлі мүмкіндіктерін көрсетуге арналған [5].

```
/* Бұл программа printf() функциясының форматтау специфика-
торларын пайдалану тәсілдерін көрсетеді. Программа кодын
талдай отырып, оның нәтижесіне назар аударыңдар
 (1.3-сурет, 1.4-сурет) */
#include <stdio.h>
#include <conio.h>
void main()
{
char c = 'A', psz[] = "A string for experiments";
int iln = 0,    // жолдар санауышы
ivalue = 1234;
double dPi = 3.14159265;

//  1-жол - c символын шығару
printf  ("\n[%2d] %c", ++iln, c);  /* 2d - оқуға ыңғайлы
болуы үшін бір цифрдан тұратын жол нөміріне екі орын
(позиция) берілген; ++iln - санауышты бірге арттыру */

//  2-жол - c символының ASCII-кодын шығару
printf ("\n[%2d] %d", ++iln, c);

//  3-жол - ASCII-коды 90-ға тең символды шығару
printf ("\n[%2d] %c", ++iln, 90);
```

```
//   4-жол - ivalue мәнін сегіздік жүйеде шығару
printf ("\n[%2d] %o", ++iln, ivalue);
//   5-жол - ivalue мәнін оналтылық жүйеде шығару
//   әріптер төменгі регистрде
printf("\n[%2d] %x", ++iln, ivalue);
//   6-жол - ivalue мәнін оналтылық жүйеде шығару
//   әріптер жоғарғы регистрде
printf("\n[%2d] %X", ++iln, ivalue);
//   7-жол - өрістің минималды енін 5 орын етіп,бір символды
//   шығару, босорындарды қоса отырып, оң жақ шетке
//   (келісім бойынша) туралау
printf("\n[%2d] %5c", ++iln, c);
//   8-жол - өрістің минималды енін 5 орын етіп,бір символды
//   шығару,босорындарды қоса отырып,сол жақ шетке  туралау
printf("\n[%2d] %-5c", ++iln, c);
//   9-жол - сөз тіркесін шығару, 24 символ шығады
// (тіркестегі символдар саны 24)
printf("\n[%2d] %s", ++iln, psz);
//   10-жол - тіркестің,кем дегенде,5 символын шығару
//   24 символ бейнеленеді
printf("\n[%d] %5s", ++iln, psz);
//   11-жол - тіркестің,кем дегенде,38 символын шығару
//   оң жақ шетке босорындармен толтыру арқылы туралау
printf("\n[%d] %38s", ++iln, psz) ;
//   12-жол - тіркестің,кем дегенде,38 символын шығару
//   сол жақ шетке босорындармен толтыру арқылы туралау
printf("\n[%d] %-38s", ++iln, psz);
// 13-жол - ivalue мәнін шығару,келісім бойынша 4 цифр шығады
printf("\n[%d] %d", ++iln, ivalue);
// 14-жол - ivalue  мәнін таңбасымен шығару
printf("\n[%d] %+d", ++iln, ivalue);
/* 15-жол - ivalue  мәнін кем дегенде, 3 цифрмен шығару
бірақ сіздің сұранысыңыз орындалмай, 4 цифр шығады */
printf("\n[%d] %3d", ++iln, ivalue);
// 16-жол - ivalue  мәнін кем дегенде, 10 цифрмен шығару,
// мәндерді оң жақ шетке босорындар қоса отырып туралау
printf("\n[%d] %10d", ++iln, ivalue);
// 17-жол - ivalue  мәнін кем дегенде, 10 цифрмен шығару,
// мәндерді сол жақ шетке босорындар қоса отырып туралау
printf("\n[%d] %-10d", ++iln, ivalue);
// 18-жол - ivalue  мәнін кем дегенде, 10 цифрмен шығару,
// мәндерді оң жақ шетке нөлдер қоса отырып туралау
```

```
printf("\n[%d] %010d", ++iln, ivalue);
// 19-жол - dPi мәнін келісім бойынша форматпен шығару
printf("\n[%d] %f", ++iln, dPi);
// 20-жол - dPi мәнін шығару, ең кіші өріс ені 20-ға тең
// мәндерді оң жақ шетке, босорындар қоса отырып туралау
printf("\n[%d] %20f", ++iln, dPi);
// 21-жол - dPi мәнін шығару, ең кіші өріс ені 20-ға тең
// мәндерді оң жақ шетке нөлдер қоса отырып туралау
printf("\n[%d] %020f", ++iln, dPi);
// 22-жол - dPi мәнін шығару, ең кіші өріс ені 20-ға тең
// мәндерді сол жақ шетке босорындар қоса отырып туралау
printf("\n[%d] %-20f", ++iln, dPi);
// 23-жол - сөз тіркесінің 19 символын  шығару,
// ең кіші өріс ені 19-ға тең
printf("\n[%d] %19.19s", ++iln, psz);
// 24-жол - сөз тіркесінің алғашқы екі символын  шығару
printf("\n[%d] %.2s", ++iln, psz);
/* тіркестің 32 жолын шығару керек, ал экранда тек 25 жол
көрінеді,сондықтан алғашқы жолдарды(7 жолды) көру үшін
экранды аялдата тұру керек, әйтпесе ол жылдам "өте шығады".
Экранды аялдатуды келесі түрде орындауға болады: */
printf("\nPress any key to continue, please!");
while (!kbhit());
/* kbhit()функциясы бір перне басуды күттіріп тұрады.Ол
егер перне басылса,"ақиқат" мәнін, әйтпесе - "жалған" мәнін
қайтарады. Сонымен,бос цикл жұмысы кез келген бір перне
басылғанша, жалғасады. Бұл функция conio.h тақырып файлында
сипатталған. Мұнда экранды аялдата тұруды getch(); функциясы
арқылы орындауға болады. */
// 25-жол - тіркестің алғашқы екі символын  шығару,
// ең кіші өріс ені 19-ға тең, мәндерді оң жақ шетке
// босорындар қоса отырып туралау
printf("\n[%d] %19.2s", ++iln, psz);
// 26-жол - тіркестің алғашқы екі символын  шығару,
// ең кіші өріс ені 19-ға тең, мәндерді сол жақ шетке
// босорындар қоса отырып туралау
printf("\n[%d] %-19.2s", ++iln, psz);
// 27-жол - тіркестің алғашқы алты символын  шығару,
// ең кіші өріс ені 19-ға тең, мәндерді оң жақ шетке
// босорындар қоса отырып туралау
printf("\n[%d] %*.*s", ++iln, 19, 6, psz);
// 28-жол - dPi мәнін шығару, ең кіші өріс ені 10-ға тең,
// ондық нүктеден соң сегіз цифр
```

```
printf("\n[%d] %10.8f", ++iln, dPi);
// 29-жол - dPi мәнін шығару, ең кіші өріс ені 20-ға тең,
// ондық нүктеден соң екі цифр, мәндерді оң жақ шетке
// босорындар қоса отырып туралау
printf("\n[%d] %20.2f", ++iln, dPi);
// 30-жол - dPi мәнін шығару, ең кіші өріс ені 20-ға тең,
// ондық нүктеден соң төрт цифр, мәндерді сол жақ шетке
// босорындар қоса отырып туралау
printf ("\n[%d] %-20.4f", ++iln, dPi);
// 31-жол - dPi мәнін шығару, ең кіші өріс ені 20-ға тең,
// ондық нүктеден соң төрт цифр, мәндерді оң жақ шетке
// босорындар қоса отырып туралау
printf("\n[%d] %20.4f", ++iln, dPi);
// 32-жол - dPi мәнін шығару, ең кіші өріс ені 20-ға тең,
// ондық нүктеден соң екі цифр, мәндерді оң жақ шетке
// босорындар қоса отырып туралау,ғылыми формат (экспонента)
printf("\n[%d] %20.2e\n", ++iln, dPi);
// Көру терезесін аялдату
getch();
}
```

Осы программа жұмысының нәтижесі:

```
[ 1] A
[ 2] 65
[ 3] Z
[ 4] 2322
[ 5] 4d2
[ 6] 4D2
[ 7]                A
[ 8] A
[ 9] A string for experiments
[10] A string for experiments
[11]                    A string for experiments
[12] A string for experiments
[13] 1234
[14] +1234
[15] 1234
[16]              1234
[17] 1234
[18] 0000001234
[19] 3.141593
[20]                  3.141593
[21] 0000000000003.141593
[22] 3.141593
[23] A string for experi
[24] A
Press any key to continue, please!
```

1.3 сурет. Форматталып шыққан нәтижелердің алғашқы беті

```
[10] A string for experiments
[11]                  A string for experiments
[12] A string for experiments
[13] 1234
[14] +1234
[15] 1234
[16]       1234
[17] 1234
[18] 0000001234
[19] 3.141593
[20]              3.141593
[21] 0000000000003.141593
[22] 3.141593
[23] A string for experi
[24] A
Press  any  key  to  continue, please!
[25]                    A
[26] A
[27]                A stri
[28] 3.14159265
[29]                   3.14
[30] 3.1416
[31]                  3.1416
[32]             3.14e+000
Press any key to continue
```

1.4 сурет. Форматталып шыққан нәтижелердің екінші беті

1.6 C++ тілінде мәліметтерді ағындар арқылы шығару

Си++ тілінің кітапханасын пайдаланғанда **iostream.h** атты кітапханалық файл колданылады, онда пернетақтадан енгізу кезінде **cin**, экранға шығаруда **cout** стандартты мәлімет ағындары және соларға сәйкес төмендегі операциялар да анықталған.

а) << – мәліметтерді ағынға жазу операциясы;

ә) >> – мәліметтерді ағыннан оку операциясы;

```
#include <iostream.h>;
   . . . . . .
cout << "\ nЭлементтер санын енгізіңіз: ";
cin >> n;
```

Мұндағы C++ тілінде келтірілген тақырыптық <iostream.h> файлында енгізу/шығаруды баскару кластары жиынын сипаттау мәліметтері бар. **Iostream.h** - енгізу/шығару кітапханасының стандартты тақырып файлы. Бұл кітапхана программадағы cout ағымы жайлы акпарат береді. Бұл файлда мәліметті пернетақтадан енгізуге арналған стандартты cin объект-ағымдары және экранға

мәлімет шығаратын cout объект-ағымдары анықталып, ағымға мәлімет беру << және ағымнан мәлімет оқу >> операциялары қарастырылған.

C++ тілі стиліндегі негізгі енгізу/шығару функциялары оның кластары кітапханасын пайдаланады. Оны түсіндіру үшін тағы бір программа мысалын қарастырайық.

```
#include <iostream.h>    // C++ тіліндегі программа
int main()
{  int i;
   cout << "Бүтін сан енгізіңіз:\n";
   cin >> i;
   cout << "Сіз " << i << "санын енгіздіңіз, рахмет!";
   return 0;
}
```

Мұндағы <iostream.h> – енгізу/шығару кітапханасының стандартты тақырып файлы. Ол программадағы мәліметтер ағымы жайлы ақпарат береді, iostream сөзі input/output stream сөздерінің қысқаша жазылуынан шыққан сөз. Бұл файлда мәліметті пернетақтадан енгізуге арналған стандартты cin ағымы және экранға мәлімет шығаратын cout ағымы анықталып, ағымға мәлімет беру << және ағымнан мәлімет оқу >> операциялары арқылы орындалады.

cin – ағылшынша «C» және «input» сөздерінен, ал **cout** – «C» және «output» сөздерінен құралған.

cout << "Бүтін сан енгізіңіз:\n"; жолы экранға қос тырнақшадағы сөз тіркесін шығарып, курсорды келесі жолға көшіреді.

cin >> i; жолы пернелерден енгізілген бүтін санды i айнымалысына меншіктейді.

```
cout << "Сіз " << i << "санын енгіздіңіз, рахмет!";
```

жолы қос тырнақшадағы тіркестерді өзгеріссіз, ал i айнымалысы орнына оның енгізілген сандық мәнін экранға шығарып береді.

C тілінің енгізу/шығару функцияларына қарағанда, C++ тілінің ағымдары әртүрлі типтермен жеңіл, әрі жылдам жұмыс істеуді қамтамасыз ете алады деп саналады.

Ағымнан мәлімет оқу оны *ағымнан алу* деп, ал ағымға мәлімет шығару оларды *ағымға қосу* деп айтылады.

C++ тілінде мәліметтерді форматталған түрде шығаруды толығырақ қарастырайық.

C++ тілінде мәліметтерді экранға шығару мысалдары

Келесі программа осы функциясының әртүрлі мүмкіндіктерін көрсетуге арналған [5].

```
/* Бұл программа C++ тілінде мәліметтерді шығару үшін
қолданылатын жалаушалар мен функциялардың жұмысын көрсетеді.
```

```
Мұндағы барлық материалдар түсінікті болмауы мүмкін, бірақ
практика жүзінде қолдануға болатын бірсыпыра мысалдар оларды
түсінуге және пайдалануға жол ашады.  */

#include  <strstrea.h> // осы тақырыптық файл іске қосылады,
                       // өйткені ostrstream осына сипатталған

#define  MAX_LENGTH 20  // MAX_LENGTH 20-ға тең болады

void row(void);
/* Бұл жол нөмірін шығару функциясының прототипі. Программа
орындалуы барысында осы арқылы әрбір жол нөмірі квадрат жақша
ішіне шығарылып отырады */
void  main()
{
 char c = 'A', psz[] = "It is a string for experiments ",
 strbuffer[MAX_LENGTH]; // жадты бөлу (резервте)
 int ivalue = 1234, w;
 double dPi = 3.14159265;

// c символын шығару
row();                  // [1]
cout << c;

// c символының ASCII-кодын шығару
row();                  // [2]
cout << (int)c;

// ASCII-коды 90 болып келген символды шығару
  row();                // [3]
  cout << (char)90;

// ivalue айнымалысының мәнін сегіздік жүйеде шығару
row();                  // [4]
cout << oct <<  ivalue;

// ivalue айнымалысының мәнін төменгі регистр әріптерімен
// он алтылық жүйеде шығару
  row ();               // [5]
  cout << hex << ivalue;

// ivalue айнымалысының мәнін жоғарғы регистр әріптерімен
// он алтылық жүйеде шығару
  row ();               // [6]
  cout.setf(ios::uppercase);    // он алтылық жүйе үшін
                                // жоғарғы регистрді қосу
  cout << hex << ivalue;
```

```
  cout.unsetf(ios::uppercase); // жоғарғы регистрді алу
  cout << dec;                 // ондық жүйеге қайту

// бір символ шығару, ерістің ең кіші ені 5-ке тең,
// босорындармен толтыра отырып оңға туралау
/* width()функциясы шығарылатын еріс енін береді. Ол тек
келесі шығару операторына ғана әсер етеді,ал, мысалы,
precision()функциясының әсері ол қайта қолданылғанша немесе
setprecision() функциясы шақырылғанша сақталады. */
  row();                   // [7]
  cout.width(5);
  cout << c;

// бір символ шығару, ерістің ең кіші ені 5-ке тең,
// босорындармен толтыра отырып солға туралау
  row();                   // [8]
  cout.width(5);
  cout.setf(ios::left);
  cout << c;
  cout.unsetf(ios::left);

// сөз тіркесін шығару, 24 символ бейнеленеді
  row ();                  // [9]
  cout << psz;

// кем дегенде, бес символ шығару
  row ();                  // [10]
  cout.width (5);
  cout << psz;

// кем дегенде, тіркестің 38 символын шығару
// вывод минимум 38 символов строки,
// босорындармен толтыра отырып оңға туралау
  row();                   // [11]
  cout.width(38);
  cout << psz;
// кем дегенде, тіркестің 38 символын шығару
// босорындармен толтыра отырып солға туралау
  row();                   // [12]
  cout.width(38);
  cout.setf(ios::left);
  cout << psz;
  cout.unsetf(ios::left);

// ivalue мәнін шығару,келісім бойынша, 4 цифр шығады
  row();                   // [13]
  cout << ivalue;
```

```
// ivalue мәнін таңбасымен бірге шығару
row();                  // [14]
cout.setf(ios::showpos);
cout << ivalue;
cout.unsetf(ios::showpos);
// ivalue мәнін,кем дегенде,3 цифрмен шығаруды керсету
// негізінде,сіздің талап орындалмай,4 цифр шығарылады
row ();                 // [15]
cout.width (3);
cout << ivalue;

// ivalue мәнін,кем дегенде,10 цифрмен шығару
// босорындармен толтыра отырып оңға туралау
row();                  // [16]
cout.width(10);
cout << ivalue;

// ivalue мәнін,кем дегенде,10 цифрмен шығару
// босорындармен толтыра отырып солға туралау
 row();                 // [17]
 cout.width(10);
 cout.setf(ios::left);
 cout << ivalue;
 cout.unsetf(ios::left);

// ivalue мәнін,кем дегенде,10 цифрмен шығару
// нөлдермен толтыра отырып оңға туралау
 row();                 // [18]
 cout.width(10);
 cout.fill('0');
 cout << ivalue;
 cout.fill(' ');

// dPi мәнін,келісім бойынша,форматтап шығару
  row();                // [19]
  cout << dPi;

// Экранды аялдату!
/* Бұл экранға шыққан мәліметтердің алғашқы бөлігін көріп
үлгеру үшін керек. Экранды аялдатуды төмендегідей түрде
орындауға болады: */
cout << "\nPlease,input any letter or number to continue\n";
cin >> w;
cout << "\n";

// dPi мәнін шығару,ерiс ені,кем дегенде,20-ға тең,
// босорындармен толтыра отырып оңға туралау
```

С.Торайғыров
атындағы ПМУ-дің
академик С.Бейсембаев
атындағы ғылыми
КІТАПХАНАСЫ
688041

```
row();          // [20]
cout.width(20);
cout << dPi;

// dPi мәнін шығару,ерiс ені,кем дегенде,20-ға тең,
// нөлдермен толтыра отырып оңға туралау
row();          // [21]
cout.width(20);
cout.fill('0');
cout << dPi;
cout.fill(' ');

// dPi мәнін шығару,ерiс ені,кем дегенде,20-ға тең,
// босорындармен толтыра отырып солға туралау
 row ();              // [22]
  cout.width(20);
  cout.setf(ios::left);
  cout << dPi;

// dPi мәнін шығару,ерiс ені,кем дегенде,20-ға тең,
// нөлдермен толтыра отырып солға туралау
   row();             // [23]
  cout.width(20);
  cout.fill('0') ;
  cout  << dPi;
  cout.unsetf(ios::left);
  cout.fill(' ');

// тіркестің 19 символын шығару,ерiс ені,кем дегенде,19-ға тең
row();          // [24]
ostrstream(strbuffer, 20).write(psz, 19) << ends;
/*  Мұнда класс және класс конструкторы түсініктері
қолданылады, олар туралы кейін айтылады. Өзірге мынадай
түсінік береміз.Мұндай жазу қажеттілігінің туындаған себебі
форматтау жалаушалары < операторына әсер етеді,бірақ керсе-
тілген символдар санын жазатын write() функциясына емес.Оның
шығаратын мәндері әрқашанда сол жақ шетке тураланады.Мұны
өзгерту үшін буферлік тіркесті анықтап, сонымен әрекеттер
орындайды. */
  cout.width(19);
  cout << strbuffer;
/* тіркестің алғашқы екі символын шығару, өйткені үшіншісі
   \0 - тіркес соңы белгісі болады */
  row ();       // [25]
  ostrstream(strbuffer,3).write(psz,2) << ends;
  cout << strbuffer;
```

```
// тіркестің алғашқы екі символын шығару, ерістің ең кіші
ені
// 19-ға тең,босорындармен толтыра отырып оңға туралау
  row ();          // [26]
  cout.width(19);
  cout << strbuffer;

// тіркестің алғашқы екі символын шығару, ерістің ең кіші
ені
// 19-ға тең,босорындармен толтыра отырып солға туралау
  row ();          // [27]
  cout.width(19);
  cout.setf(ios::left);
  cout << strbuffer;
  cout.unsetf(ios::left);

// dPi мәнін оның бөлшек бөлігінің тоғыз цифрымен шығару
  row ();          // [28]
  cout.precision(9);
  cout << dPi;

// dPi мәнін оның бөлшек бөлігінің екі цифрымен шығару,
// өріс ені,кем дегенде,20-ға тең
// босорындармен толтыра отырып оңға туралау
  row();           // [29]
  cout.width(20);
  cout.precision(2);
  cout << dPi;

// dPi мәнін оның бөлшек бөлігінің төрт цифрымен шығару
  row();           // [30]
  cout.precision(4);
  cout << dPi;

// dPi мәнін оның бөлшек бөлігінің төрт цифрымен шығару
// өріс ені,кем дегенде,20-ға тең
// босорындармен толтыра отырып оңға туралау
  row();           // [31]
  cout.width(20);
  cout << dPi;

// dPi мәнін шығару,өріс ені,кем дегенде,20-ға тең
// ондық нүктеден кейін төрт цифрмен шығару,
// босорындармен толтыра отырып оңға туралау
// ғылыми формат (экспонентамен)
```

```
row();              // [32]
cout.setf(ios::scientific);
cout.width(20);
cout << dPi;
cout.unsetf(ios::scientific);
cout << "\n";
}

void row(void)
{
  static int ln = 0; /* ln айнымалысы static түрінде
                     жарияланған,сондықтан row()функциясын
                     қайтадан шақырғанда,ln айнымалысының соңғы
                     мәні сақталады.Бұл алдыңғы жолдың нөмірін
                     программаның "есте сақтауы" үшін жасалған
                     */

   cout << "\n [";
   cout.width(2);
   cout << ++ln << "] ";
}
```

Программа орындалған кездегі оның алғашқы 19 жолы суретте көрсетілген.

```
[ 1] A
[ 2] 65
[ 3] Z
[ 4] 2322
[ 5] 4d2
[ 6] 4D2
[ 7]     A
[ 8] A
[ 9] It is a string for experiments
[10] It is a string for experiments
[11]         It is a string for experiments
[12] It is a string for experiments
[13] 1234
[14] +1234
[15] 1234
[16]         1234
[17] 1234
[18] 0000001234
[19] 3.14159
Please,input any letter or number to continue
```

1.5 сурет. C++ тіліндегі экранға шыққан нәтижелердің алғашқы беті

Программаға, мысалы, "y" әріпін енгізіп, Enter пернесі басылғанда, экранға калған жолдар жұмысының нәтижесі шығарылады.

```
 [13] 1234
 [14] +1234
 [15] 1234
 [16]         1234
 [17] 1234
 [18] 0000001234
 [19] 3.14159
Please,input any letter or number to continue
y

 [20]                  3.14159
 [21] 00000000000003.14159
 [22] 3.14159
 [23] 3.14159000000000000
 [24] It is a string for
 [25] It
 [26]                      It
 [27] It
 [28] 3.14159265
 [29]                      3.1
 [30] 3.142
 [31]                    3.142
 [32]             3.1416e+000
Press any key to continue
```

1.6 сурет. C++ тіліндегі экранға шыккан нәтижелердің екінші беті

С тілінің енгізу және шығару операторларын C++ тілінің енгізу және шығару операторларымен бірге пайдалануға болмайды.

C++ тілінде бүтін сандармен арифметикалық амалдар орындайтын тағы бір мысал карастырайық.

C++ тілінде бүтін сандармен арифметикалық амалдар орындайтын тағы бір мысал карастырайық.

```
/* ia және ib бүтін сандарын қосу, азайту, көбейту және бөлу
операцияларын орындау */
#include <iostream.h>
void main()
{         // мәліметтерді сипаттау
int ia, ib, iplus, iminus;
float del, mult;
          // сандық  мәндерді енгізу
cout << "Input ia, ib: \n";  //"ia, ib енгіз:" сөзін шығару
cin >> ia >> ib; // енгізілген мәндерді ia, ib-ға меншіктеу
          // есептеулер
```

```
iplus = ia + ib;
iminus = ia - ib;
mult = ia * ib;
del = ia / ib;
// нетижелерді шығару
cout << "plus = " << iplus << "\n";
cout << "minus = "<< iminus <<"\n";
cout << "del = " << del  <<"\n";
cout << "mult = " << mult  <<"\n";
}
```

Программа нәтижесі:

```
Input ia, ib:
4 5
Plus = 9
Minus = -1
Del = 0
Mult = 20
```

Пернетақтадан **ia** мен **ib** мәндерін енгізу кезінде арасында бос орын болуы тиіс. Мысалда 4 цифрын енгізіп, бос орынды басып, 5 цифрын теріп, Enter пернесін басу қажет.

Бөлу кезінде шығатын нәтижеге назар аударыңдар, ол дұрыс болмайды. Бұл бүтін санды бүтін санға бөлгендегі бөлінді де бүтін сан болатындығынан шығып отыр. Мұндайда дұрыс нәтиже алу үшін, **ia**, немесе **ib** айнымалыларының бірінің типін **float** деп көрсету керек.

Келесі бөлімдерде осы C және C++ тілдерінің мүмкіндіктерінің екеуінен де мысалдар келтіріледі, бірақ бір программада бұл екеуін араластырмай, жеке-жеке карастыру ұсынылады.

Бақылау сұрақтары

1. Формат тіркесі дегеніміз не? `printf` функциясындағы формат тіркесінің құрамы қандай? `scanf` функциясындағы формат тіркесінің құрамы қандай?
2. Түрлендіру спецификациясы дегеніміз не? Мәліметтердің әр түрлі типтері үшін түрлендіру спецификацияларынан мысалдар келтіріңдер.
3. Келесі функция жұмысы нәтижесінде экранға не шығады?
   `printf("\nСандар тізбегінің арифметикалық ортасы: %10.5f \`
   `\n Сол тізбектің жұп элементтерінің саны: %10.5d ",S/n,k);`
4. Алдынғы сұрақтың мәліметтерін экранға шығаруды << операциясы арқылы қалай жазамыз?
5. Егер x айнымалысы типі `long int`, ал y айнымалысы типі `double` болса, оларды `scanf()` функциясының көмегімен экранға шығару қалай орындалады? Осы әрекеттерді >> операциясы көмегімен қалай жүзеге асыруға болады?

1.5 C/C++ тілінің қарапайым операторлары

Бос оператор. Тек `';'` (нүктелі үтір) таңбасынан тұрады. Өрнек түріндегі операторлардың соңын білдіреді. Күрделірек операторлар ішінде де (`if, for, while, do..while, switch`) ешқандай операция орындалмайтын бос әрекетті білдіреді.

Оператор-өрнек. С тілінің ережелерін сақтай отырып жазылған кез келген күрделі өрнек (формула). Ол нүктелі үтір таңбасымен аяқталады.

`Break, continue` операторлары. Әсер ету аймағы шектелген шартсыз көшу операторлары болып табылады. Басқа операторлар (`for, while, do..while, switch`) ішінде пайдаланылып, олардың орындалу реттілігін өзгерту үшін қолданылады. `Break` цикл немесе `switch` операторларының жұмысын бірден тоқтатады, ал `continue` операторы циклдің келесі қадамына көшіп кетуді жүзеге асырады.

Операторлар блогы мәліметтер хабарламалары мен соларды анықтаулардан және де жүйелі жақшаға алынған операторлар тізбегінен тұрады. Тіл ережесіне сәйкес бір оператор тұратын орында орналасып, бірнеше әрекеттер тізбегін орындай алады.

`Return` операторы функция орындалуын аяқтайтын оператор ретінде пайдаланылады. Екі түрлі форматта қолданылады:

```
return;
```

немесе

```
return (өрнек) ;
```

Алғашқы түрі осы сәтте жұмыс істеп тұрған функциядан оны шақырған программаға кері қайту ісін орындайды, бұл оператор болмауы да мүмкін, ондайда функция операторларын орындау аяқталған соң, кері қайту жүзеге асырылады.

Ал екінші түрі тек кері қайту әрекетін ғана орындамай, оны шақырған функцияға `өрнек` мәнін де қайтарады. Бұл мән шақырған функциядағы белгілі бір объектіге (өз типіне сәйкес) меншіктеледі.

`Goto` операторы. Әсер ету аймағы шектелген шартсыз көшу операторларына (`break, continue`) қарағанда, `goto` операторы басқаруды *белгісі* бар кез келген жолға бере алады. Оның жазылуы:

```
goto белгі;
...
белгі: оператор;
```

Белгі қарапайым айнымалы атауы түрінде болады да, одан кейін кос нүкте қойылады. Бұл операторды қолдану қажет болып саналмайды, құрылымдық программалау талаптарына сай оны пайдаланбауға тырысу керек, өйткені ол программа жұмысын түсінуді қиындатады және жөнделуі қиын қателердің туындауына себепші болады.

Дегенмен, кейде оны қолдану қажеттілігі де кездесіп қалады. Көбінесе ол қате шыққан сәттерде функция соңына көшу ісін атқарады, мұндай қателер жиі кезесіп жатса, оны қолдану ыңғайлы болып жатады. Мысалы:

```
void func1()
{  // функция операторлары
   if (do_1()==FAILURE) goto error;
   if (do_2()==FAILURE) goto error;
   if (do_3()==FAILURE) goto error;
    // функция операторлары
   return SUCCESS;
error:
    // функция операторлары
   return FAILURE
};
```

Goto операторы қабаттасқан цикл операторлардың ішкі бөлігінен бірден ең сыртына шығу үшін де қолданылады (break операторы тек бір цикл сыртына шығарады). Мысалы:

```
while(1-шарт)
{  while(2-шарт)
      {  while(3-шарт)
            {  // операторлар
               if (қате_шарты) goto loop_end
                  // операторлар
            }
      }
}
loop_end: // программа жалғасы
```

Goto операторының белгісі ағымдағы осы функцияда болуын қадағалау қажет. Басқаша айтқанда, goto операторын басқа функцияда жарияланған (немесе керісінше) белгіге көшу үшін қолдануға болмайды.

Жаңа үйреніп жүрген программалаушылар өз программаларында бұл операторды жиі қолданады. Goto операторын колдануға тырыспай, оның орына if, if...else немесе while конструкциясын пайдалану дұрысырақ болып саналады.

1 саны бүтін сандар ішіндегі ерекше сан болып саналады. Бірден үлкен және өзінен басқа ешқандай бөлгіштері жоқ бүтін натурал сан қарапайым (жай) сан деп аталады.

1.6 Шартты операторлар

If шартты операторының екі түрлі жазылу форматы бар:

```
if (шарт-өрнек)
  1-оператор;
[else
  2-оператор;]
```

Тік жақшаға алынған бөлікті жазбай кетуге болады. Бұл бөлік жазылмаған жағдайда, оператордың бірінші – *қысқаша түрі* қолданылады деп айтылады. Операторды орындау барысында **шарт-өрнек** есептеледі. Егер оның мәні «ақиқат» (≠ 0, «иә») болса, онда **1-оператор** орындалады. Ал егер ол «жалған» (= 0, «жоқ») болса, онда **2-оператор** атқарылады. **Else** бөлігі болмаған жағдайда, **if** операторынан кейін орналасқан операторлар орындалады. Мысалы:

$$y = \begin{cases} sin\sqrt{x}, & егер\ x \geq \pi \\ x \cdot e^{x+1}, & егер\ x < \pi \end{cases}$$

өрнегін есептеу қажет болса, оның алгоритмі схемасына (1.7 сурет) сәйкес шартты оператор былай жазылады:

```
if (x >= pi)
  y=sin(sqrt(x));
else
  y=x*exp(x+1);
```

Оператордың *қысқаша түріне* мысал ретінде айнымалы мәні теріс болған жағдайда оның абсолюттік шамасын табатын өрнекті келтіруге болады:

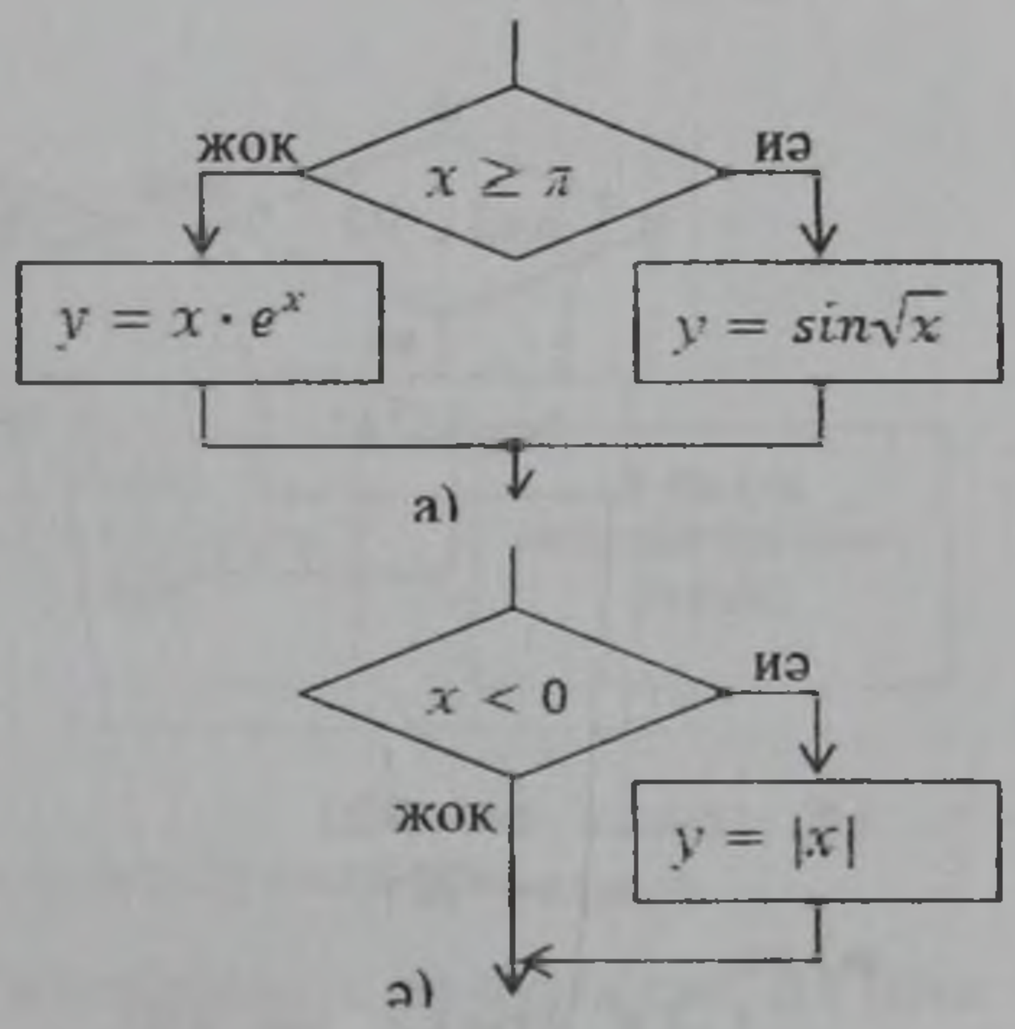
```
if (x<0)
   y=abs(x);
```

If операторын қабаттастыра отырып, күрделі өрнек мәндерін анықтайтын жағдайлар да жиі кездесіп отырады, мысалы:

```
if (c == 1)
  kun="duisenbi";
else if (c == 2)
       kun="seisenbi";
     else if (c == 3)
            kun="sarsenbi";
          else if (c == 4)
                 kun="beisenbi";
               else if (c == 5)
                      kun="juma";
                    else if (c == 6)
                           kun="senbi";
                         else if (c == 7)
                                kun="jeksenbi";
                              else puts("kate");
```

Мынадай да мысалдар кездеседі:

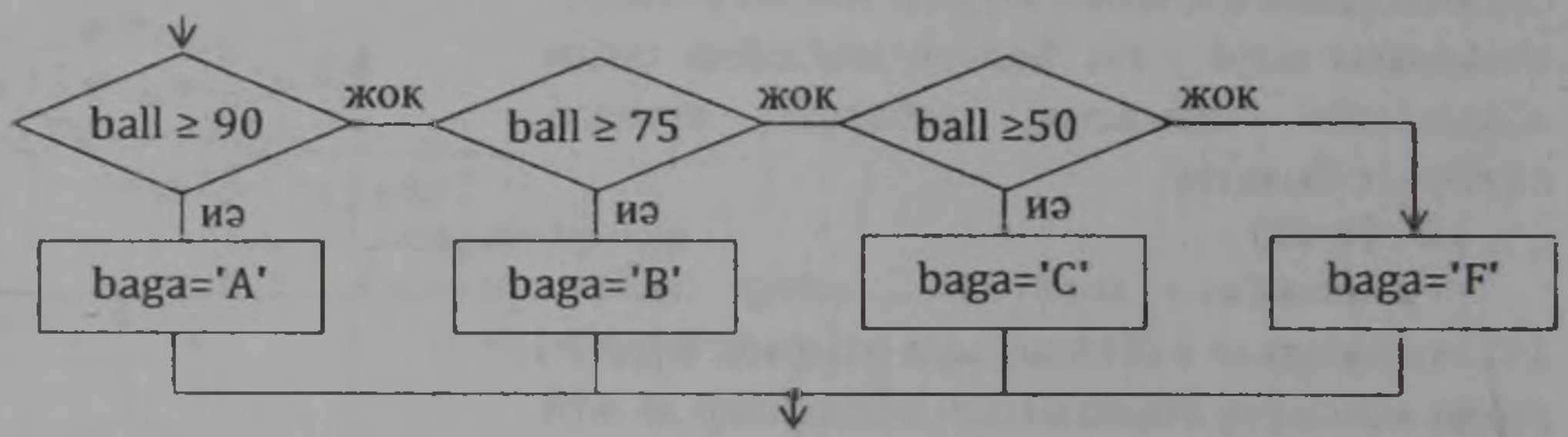
```
if (ball >= 90)
     baga='A';
else
   if (ball >= 75)
```



1.7-сурет. Тармақты алгоритм схемасы: а) толық түрі; ә) қысқа түрі

```
if (ball < 90)
      baga='B';
else
   if (ball >= 50)
      if (ball < 75)
         baga='C';
      else
         baga='F';
```

Мұндай кездерде компилятор **else** түйінді сөзін онын алдындағы ең соңғы **if** операторымен байланыстырады. Сондықтан **else** түйінді сөзін соған сәйкес **if** сөзімен бір деңгейде жазған ыңғайлы болады деп саналады. Бірақ ең ыңғайлысы қабаттаскан **if** операторларында олардың дұрыс орындалу реттілігін көрсететін операторлар блогын пайдалану болып табылады (1.8 сурет):



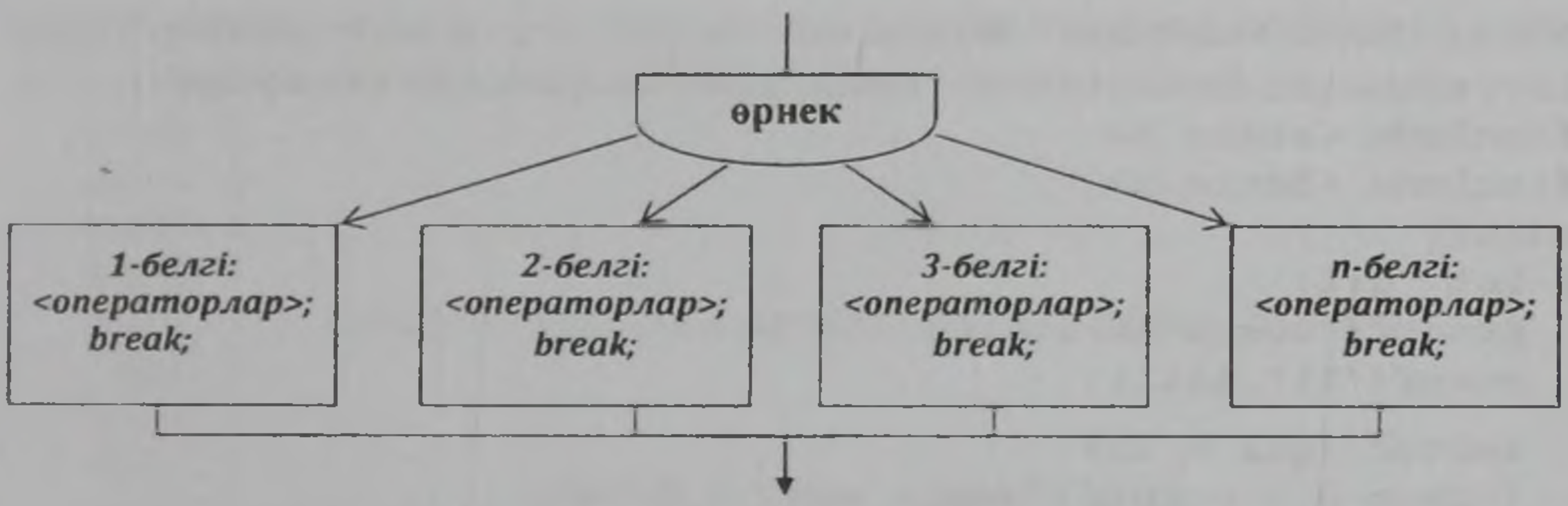
1.8-сурет. Қабаттасқан тармақты алгоритмнің бейнеленуі

```
if (ball >= 90)
      baga='A';
else
   { if (ball >= 75)
         baga='B';
      else
         { if (ball >= 50)
              baga='C';
           else
              baga='F';
         }
   }
```

1.7 Switch операторы

Жоғарыда көрсетілген қабаттасқан **if** операторларын **switch** ауыстыргыш операторын пайдаланып қысқаша жазу мүмкіндігі бар. Бұл операторды есептердің тармақталу жолдары бірнешеу (екеуден артық) болған кездерде, солардың бірін таңдау үшін қолданған ыңғайлы болып келеді. **Switch** операторының жазылу синтаксисі (алгоритмі 1.9 суретте):

```
switch (бүтін_сандық_өрнек)
{
  case 1_тұрақты_өрнек :
       1-оператор; [break;]
  case 2_тұрақты_өрнек :
       2-оператор; [break;]
  case 3_тұрақты_өрнек :
       3-оператор; [break;]
    . . .
  case n_тұрақты_өрнек :
       n-оператор; [break;]
  default :
       оператор;
}
```



1.9-сурет. Switch операторының орындалу схемасы

Мұнда **switch** сөзінен кейінгі өрнек мәні есептеледі, ол бүтін сандық (char типі де) типте болуы тиіс. Сол мән **case** сөздерінен кейін жазылған тұрақтылар мәндерімен салыстырылады. Егер тұрақты мәні солардың біріне тең болса, сол жол орындалады да, жол соңында міндетті емес **break** операторы болмаса, келесі жолдар толық орындалады. Ал бір жолды орындап болған соң, **switch** операторынан шығу үшін **break** операторы қолданылады. Егер **switch** сөзінен кейінгі өрнек мәні ешбір тұрақтымен сәйкес келмесе, онда **default** сөзінен кейінгі оператор аткарылады. Кейде **default** сөзі болмауы да мүмкін.

Default сөзі болмаса, онда **switch** операторынан кейінгі келесі операторлар орындала береді. **Switch** операторындағы өрнек түрінде нақты типтегі мәліметтерді, сөз тіркестерін (жолдарды) пайдалануға болмайды. Кейде бүтін мәндермен үйлестірілген мәліметтердің құрылымдық (структуралық) элементтері қолданылуы мүмкін.

Мысал. Екі бүтін сан енгізіп, олармен арифметикалық 4 амалдың бірін орындау.

```
#include <stdio.h>
#include <conio.h>
char symbol;
```

```
int x,y,z;
clrscr();
printf ("Екі бүтін сан енгізіңіз: ");
scanf ("%i%i",&y,&z);
printf ("Қандай амал орындау керек: ");
scanf ("%s",symbol);
switch (symbol)
{
 case '-' : x=y-z; break;
 case '+' : x=y+z; break;
 case '*' : x=y*z; break;
 case '/' : x=y/z; break;
 default:  printf ("белгісіз операция\n");
 printf("\nНәтижесі - %d\n",x);
 getch();
}
```

Мысал. Шығыс календары бойынша жылға сәйкес жануар атын анықтау. Мұнда 12-ге қалдықсыз бөлінетін жыл – мешін жылы болатыны негізге алынған.

```
#include <stdio.h>
#include <conio.h>
main()
{ int  gil;
  printf("Жылды енгізіңіз: \n");
  scanf("%i",&gil);
switch (gil % 12)
{ case 0 : printf("Мешін жылы");break;
  case 1 : printf("Тауық жылы"); break;
  case 2 : printf("Ит жылы"); break;
  case 3 : printf("Доңыз жылы"); break;
  case 4 : printf("Тышқан жылы"); break;
  case 5 : printf("Сиыр жылы"); break;
  case 6 : printf("Барыс жылы"); break;
  case 7 : printf("Қоян жылы"); break;
  case 8 : printf("Ұлу жылы"); break;
  case 9 : printf("Жылан жылы"); break;
  case 10: printf("Жылқы жылы"); break;
  case 11: printf("Қой жылы"); break;
   default : printf("Таңбасыз бүтін сан
                     енгізіңіз");
  }
  printf("\nENTER басыңыз");
  getch();
}
```

Мысал. Апта күнінің реттілік саны бойынша күннің атын анықтау.

```
#include <stdio.h>
#include <conio.h>
```

```
main()
{ int  day_num;
  printf ("Enter day number: ");
  scanf("%i",&day_num);
  switch (day_num)
  {
    case 1 : printf("Monday"); break;
    case 2 : printf("Tuesday"); break;
    case 3 : printf("Wednesday"); break;
    case 4 : printf("Thursday"); break;
    case 5 : printf("Friday"); break;
    case 6 : printf("Saturday"); break;
    case 7 : printf("Sunday"); break;
   default : printf("Enter an integer (1..7)");
  }
  getch();
}
```

Егер өрнек мәнінің бірнеше мәнінде тек бір оператор орындалуы қажет болса, онда `case` тармақтарын біріктіріп кою керек:

```
case 1 :
case 3 :
case 5 :
case 7 :
case 9 : printf("Odd numbers"); break;//тақ сандар
case 2 :
case 4 :
case 6 :
case 8 :
case 10 : printf("Even numbers"); break; //жұп сандар
. . .
```

1.8 C/C++ тілінің операторларын қолдану арқылы есептер шығару

«Жұмысты жаңа бастаған» программалаушылар, әсіресе студенттер программа жазуды былай бастайды: тапсырманы алған соң, бірден компьютерге отырады да, сол алгоритмнің өздері білетін фрагменттерін кодтай бастайды. Айнымалыларға бірден қарапайым атаулар (x және y сияқты) береді. Компьютер жұмыс істемей тұрып қалған жағдайда, жазылған программаның барлығы өшеді де, үзіліс жасалынып, барлық жұмыс басынан қайталанады. Компилятордың, компьютердің немесе операциялық жүйенің дұрыс жұмыс істемей жатқандығы жайлы жаңсақ күмәнді пікірлер де айтылып жатады. Программаны орындау кезіне жеткенде, оған кез келген мәндер енгізіледі де, экранға шыққан нәтижелерге таңданып қараумен болады. Бұндай программа мәліметтердің тек бір енгізілген мәндері үшін ғана жұмыс істейді, ал өзгерістер енгізілсе, ойға симайтын нәтижелер шығып, автордың өзіне деген сенімсіздігі пайда болып, тіпті ол программалау ісін жек көріп кететін жағдайларға да тап болуы мүмкін.

Студенттің мақсаты программалауды кәсіби деңгейде үйрену болып табылады. Жақсы маман әрбір программаны онда қойылған талаптарға сәйкес жазуға канша

уақыт кететінін алдын ала анықтай алатын дәрежеде болуы тиіс. Мұндай жұмысқа «білім, шыдам және төзімділіктен» басқа, тәжірибе өте қажет нәрсе болып саналады. Оған қоса, программалау пәні шыққаннан бергі жарты ғасырдан аса мерзімде мамандардың жинақтаған негізгі қағидаларын да меңгеру қажет. Тіпті ең қарапайым программа жазуда да біртіндеп тізбекті түрде атқарылатын ережелер тәрізді әрекеттерді орындау керек екендігін білген артық болмайды.

Программа жазып есеп шығаруда мынадай жұмыс кезеңдері атқарылады:

1. **Есептің математикалық моделін жасау**. Бұл кезеңде берілген мәліметтер мен алынатын нәтижелер және де мәліметтерден нәтижеге қол жеткізуді жүзеге есыратын математикалық формулалар анықталады.

2. **Есептің шығару алгоритмін құру**. Берілген мәліметтерден қажетті нәтижелер алу жолында атқарылатын әрекеттер толық анықталуы тиіс. Олар табиғи тіл немесе графикалық алгоритмді бейнелеу тәсілдерімен өрнектелуітиіс.

3. **Белгілі бір программалау тілінде алгоритмге сәйкес программа жазу**. Бұл кезеңде алгоритмнің әрбір қадамына сәйкес таңдап алынған программалау тілінің операторлары (конструкциясы) жазылады.

4. **Программаны орындау немесе атқару** (бастапқы модуль → компилятор → объектілік модуль → құрастырушы → атқарылу модулі тізбегін алу).

5. **Программаны тексеріп (тесттен өткізіп), оны қателерін түзету, жөндеу**. Программаны орындау барысында үш түрлі қате кетуі мүмкін:

а) синтаксистік қате – компиляциядан өткізу кезеңінде түзетіледі.

ә) программаның атқарылуы кезінде анықталатын қателер (сандардың 0-ге бөлінуі, теріс саннан түбір, логарифм алу, т.с.с.) – программаны алғашқы орындауда анықталады.

б) семантикалық (логикалық) қателер – тапсырманы дұрыс түсінбеген жағдайда, алгоритм дұрыс құрылмаған кездерде болады.

Мұндай қателерді анықтау үшін программаны белгілі бір сандарды бастапқы мәліметтер түрінде енгізіп орындау қажет, яғни тесттен өткізу керек. Тесттен өткізудің мақсаты – қате барын анықтау, қате кеткен орынды тауып, оның себептерін анықтау, соларға сәйкес программаны түзету. Тест дегеніміз – алдын ала нәтижелері белгілі болып келетін бастапқы мәліметтер жиыны. Қатені анықтауға көмектескен тест тиімді болып саналады. Программаны жөндеу ісі тесттердің барлығы (немесе қажетті бөлігі) қатесіз орындалғанда аяқталады, яғни олардың бәрі де дұрыс нәтиже бергенде тоқтатылады.

Тесттердің жеткілікті (қажетті) санын анықтаудың екі жолы бар: бірінші тәсілде программа «қара жәшік» ретінде қарастырылады да, оған бастапқы мәлімет енгізіліп, нәтиже алынады. Жәшіктің ішкі құрылымы бізге белгісіз. Программаны мұндай жолмен толық тесттен өткізу үшін оны барлық берілген мәліметтер жиынында тексеріп шығу керек, ал ол мүмкін жұмыс емес. Сондықтан программаны толық тексеру үшін қажетті тесттер санын анықтайтын арнайы критерийлер құрастырылады. Осы тәсілде алдымен келесі критерийлер қолданылады:

1) Бастапқы мәліметтер кластарын тесттен өткізу, яғни тесттер жиыны мәліметтердің мүмкін болатын мәндерінің әрбір класынан бір-бір мән алуы тиіс.

2) Нәтижелік мәліметтер кластарын тесттен өткізу, яғни тесттер жиыны нәтижелік мәліметтердің әрбір класынан тексеруге қажет бір-бір мән алынатындай дәрежеде болуы тиіс. Мұндайда толық тесттен өткізу – программа басынан онын соңына алып баратын барлық тармақтарды тексеру арқылы орындалады. Бірақ бұны да орындау өте қиынға соғады, өйткені циклдері мен тармақтары бар программалардың басынан соңына алып баратын жолдар саны өте көп болып келеді. Сондықтан мұндайда келесі критерийлер қолданылады:

а) Командаларды тесттен өткізу. Тесттер жиыны әрбір команданың бірнеше рет (бірден артық) орындалуын қамтамасыз етуі тиіс.

ә) Тармақтарды тесттен өткізу. Тесттер жиыны әрбір тармақтың бірнеше рет (бірден артық) орындалуын қамтамасыз етуі тиіс. Программалау практикасында осы критерийді пайдалану ең кең таралған тәсілдерге жатады.

2 СЫЗЫҚТЫҚ ЖӘНЕ ТАРМАҚТЫ АЛГОРИТМДЕРГЕ БЕРІЛГЕН ЕСЕПТЕР

2.1 есеп. А) Нақты *a* және *b* сандары берілген. Олардың қосындысын *s*, айырмасын *r*, көбейтіндісін *p* және *a*-ны *b*-ға бөлгендегі бөліндіні *h* табу керек.

Бұл алгоритм қарапайым болып келеді: программаға a және b сандарын енгізу, $s=a+b$, $r=a-b$, $p=a\cdot b$, $h=a/b$ мәндерін есептеу және экранға нәтижелерін шығару. Бірақ $b=0$ болған кезде бұл алгоритм қате нәтиже береді. Мұндайда $h=a/b=\infty$ деп математикалық түрде экранға жазып шығаруға болады. Бірақ компьютерде нөлге бөлу операциясы орындаларда қате шығып, нәтижесінде экран бетінде «нөлге бөлу» туралы хабарлама пайда болады да, программаның орындалуы тоқтатылады. Сондықтан b нөлге тең екендігін тексеру алгоритмде көрсетілуі тиіс. Осы мүмкіндікті тексеретін алгоритм 2.1-суретте көрсетілген. Төменде алгоритмнің мәтіндік түрде жазылған нұсқасы берілген.

1-қадам. a, b енгізу.

2-қадам. s=a+b, r=a-b, p=a·b есептеу.

3-қадам. s, r, p мәндерін экранға шығару.

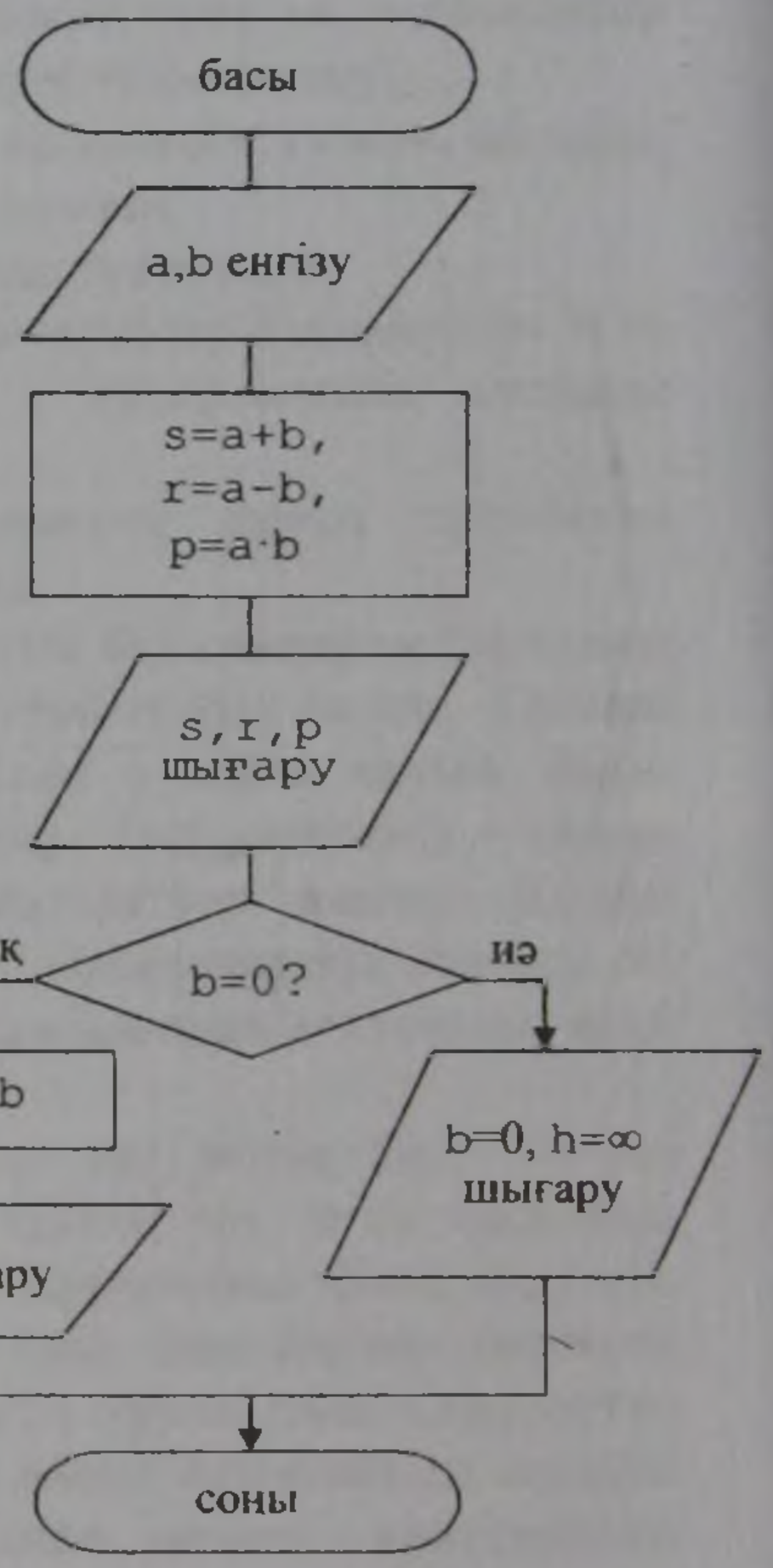
4-қадам. Егер b=0 болса, 7-қадамға көшу.

5-қадам. h=a/b мәнін есептеу.

6-қадам. h-ты шығаружәне 8-қадамға көшу.

7-қадам. b=0, h=∞ хабарламасын шығару.

8-қадам. Соны.



2.1-сурет. Нақты сандармен арифметикалық амалдар орындау

Алгоритмнің C тіліндегі программасының бір нұсқасы :

```
#include <stdio.h>
#include <conio.h>
main()
{float a,b,s,r,p,h;
 printf("a,b: ");
 scanf("%f%f",&a,&b);
 s=a+b; r=a-b; p=a*b;
 printf("s=%5.2f\nr=%5.2f\n
         p=%7.3f\n",s,r,p);
if(b==0) puts(" b=0,h=шексіздік");
   else
   { h=a/b;
     printf("\nh=%6.3f",h);
   }
 getch();
}
```

a=8,75 және b=3,12 болған кездегі программа жұмысы нәтижесі:

```
a,b: 8.75 3.12
s=11.87
r= 5.63
h= 2.804
```

Ал b=0 болған кездегі программа жұмысы нәтижесі:

```
a,b: 4.35 0
s= 4.35
r= 4.35
p=  0.000
b=0, h=шексіздік
```

Ә) Нақты *a*, *b*, *c*, *d* сандары берілген. Солардың қосындысын *sum*, айырмасын *airma* және көбейтіндісін *kobeit* табу керек.

Есептің С тіліндегі программасының коды:

```
#include <stdio.h>
#include <conio.h>
main()
{
  float a,b,c,d;
  float sum,kobeit,airma;
  clrscr();
  printf("a,b,c,d : ");
  scanf("%f%f%f%f",&a,&b,&c,&d);
  sum=a+b+c+d;
  printf("sum = %7.2f",sum);
  kobeit=a*b*c*d;
  printf("\nkobeit = %7.2f",kobeit);
  airma = a-b-c-d;
  printf("\nairma = %7.2f",airma);
  getch();
}
```

Программа жұмысы нәтижесі:

```
a,b,c,d : 3.45 -5.1 6.45 2.3
sum =  7.10
kobeit = -261.02
airma =  -0.20
```

Б) Нақты *x*, *y* сандары берілген. Келесі өрнекті есептеу программасын жазуды қарастырайық:

$$a = 2^{-x}\sqrt{x + \sqrt[3]{|y|}} \cdot \sqrt{e^{x-1/\sin x}}$$

С тіліндегі программасы коды:

```
// Бұл программада a-ның мәні берілген x,y үшін есептеледі
#include <stdio.h>
#include <math.h>
void main()
{    double a, a1, a2, a3, x, y;
     puts("Enter x:");
     scanf("%lf", &x) ; // %lf - double типінің форматы
     printf("x = %7.4lf\n", x);
     puts("Enter y:");
     scanf("%lf", &y);
     printf("y = %7.4lf\n", y) ;
     a1 = pow(2,-x);
     a2 = sqrt(x+pow(fabs(y),(1./4.)));
     a3 = sqrt(exp(x-1/sin(x)));
     a = a1 * a2 * a3;
     printf("\nResult = %lf\n",a);
}
```

Берілген $x = 4{,}1$ және $y = 5{,}2$ мәндері үшін программа жұмысы нәтижесі:

```
Enter x:
4.1
x =  4.1000
Enter y:
5.2
y =  5.2000
Result = 1.976681
```

Келесі x, y мәндерін енгізіп, нәтижені анықтайық: `3.981E-2` `-1.625E3`

```
Enter x
3.981E-2
x =  0.0398
Enter y
-1.625E3
y = -1625.0000
Result = 0.000009
```

В) Нақты x, y сандары берілген. Келесі өрнекті есептеу программасын жасауды қарастырайық:

$$\frac{|x|+|y|}{1+|xy|}$$

Есептің С тіліндегі программасының бір нұсқасы:

```
/* программада берілген x,y үшін өрнек мәні есептеледі */
#include <stdio.h>
#include <conio.h>
#include <math.h>
```

```
main()
{
  float x,y,rez;
  printf("x,y : ");
  scanf("%f%f",&x,&y);
  printf("x= %6.2f y=%6.2f",x,y);
  rez=(fabs(x)+fabs(y))/(fabs(x*y)+1);
  printf("\nrez = %f",rez);
  getch();
}
```

Программаны орындау нәтижесі:

```
x,y : -23.56 41.34
x= -23.56  y= 41.34

rez = 0.066566
```

Г) Нақты *x*, *y* және *z* сандары берілген. Олардың мәндерін пернетақтадан енгізе отырып, төмендегі *b* өрнегінің мәнін табу керек.

$$b = y^{\sqrt[3]{|x|}} + \cos^3 y \cdot \frac{|x-y|\left(1+\frac{\sin^2 z}{\sqrt{x+y}}\right)}{e^{|x-y|}+x/2},$$

осы өрнекті $x = 5{,}324$, $y = 0{,}5281$, $z = 23{,}654$ болған кезде есептеу керек болсын.

```
/*   b өрнегін x, y, z мәндері арқылы есептеу   */
#include  <stdio.h>
#include  <math.h>

void main ()
{ double x, y, z, a1, a2, a3, a4, b;
puts( "input x, y, z: " );      /* бұл функция сөз тіркесін
  (put string) экранға шығарады, тақырыптық файлы stdio.h */
scanf ("%lf%lf%lf",&x,&y,&z);   // %lf - double форматы
a1 = pow(y,pow(fabs(x),1./3));
a2 = pow(cos(y),3) ;
a3 = fabs(x - y)*(1 + pow(sin(z),2)/sqrt(x + y));
a4 = exp(fabs(x - y)) + x/2;
b = a1 + a2 * a3 / a4;
printf ("\n for x = %lf, y = %lf, z = %lf, b = %lf\n", x, y,
z, b);
}
```

```
input x, y, z:
5.324 0.5281 23.654

for x = 5.324000, y = 0.528100, z = 23.654000, b = 0.363196
```

Д) Келесі $f(x)$ функциясын пайдаланып, *t* өрнегінің мәнін анықтайтын программа жазу керек.

$$f(x) = sinx \cdot cos2x - sin2x \cdot cos\frac{2}{x}; \quad t = f(a+b) - \frac{f(a)}{f(a-b)}$$

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
/* f  функциясының сипатталуы */
float f(float x)
{ float f1;
  f1=sin(x)*cos(2*x)-sin(2*x)*cos(2/x);
  return f1;
}
main()
{ float a,b,t;
  clrscr();
  printf("\n a,b : ");
  scanf("%f",&a); // a=b болмауы тиіс
  scanf("%f",&b);
  t=f(a+b)-f(a)/f(a-b);
  printf("\nt = %5.2f",t);
}
```

2.2 есеп. Нақты a және b оң сандары берілген. Осы сандардың арифметикалық ортасы мен геометриялық ортасын табу керек.

Бұл сандарының арифметикалық ортасы $s = (a + b)/2$, ал геометриялық ортасы $g = \sqrt{a \cdot b}$ болып табылады. Сондықтан оның алгоритмі 2.2-суретте көрсетілгендей түрде болады. Ал мәтін түрінде берілген шығару жолы мынадай:

1-қадам. a, b енгізу.

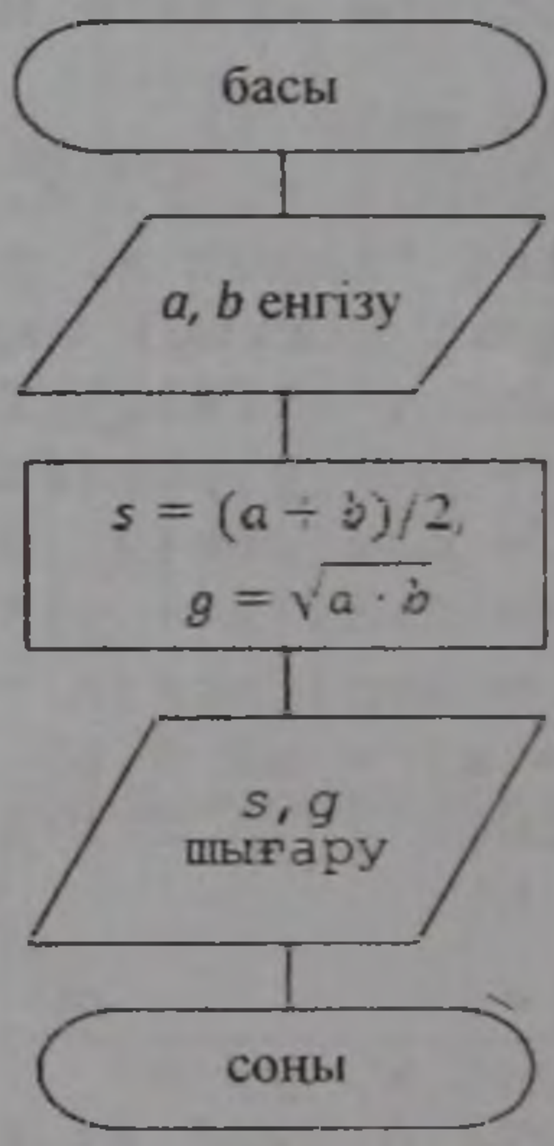
2-қадам. $s = (a + b)/2$, $g = \sqrt{a \cdot b}$ есептеу.

3-қадам. s, g шығару.

4-қадам. Соңы.

Осы алгоритмнің C тіліндегі программасы :

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
main()
{float a,b,s,g;
 clrscr();
 printf("a,b: ");
 scanf("%f%f",&a,&b);
 printf("a=%5.2f b=%5.2f",a,b);
 s=(a+b)/2; g=sqrt(a*b);
 printf("\ns=%5.2f\ng=%5.2f",
        s,g);
 getch();
}
```



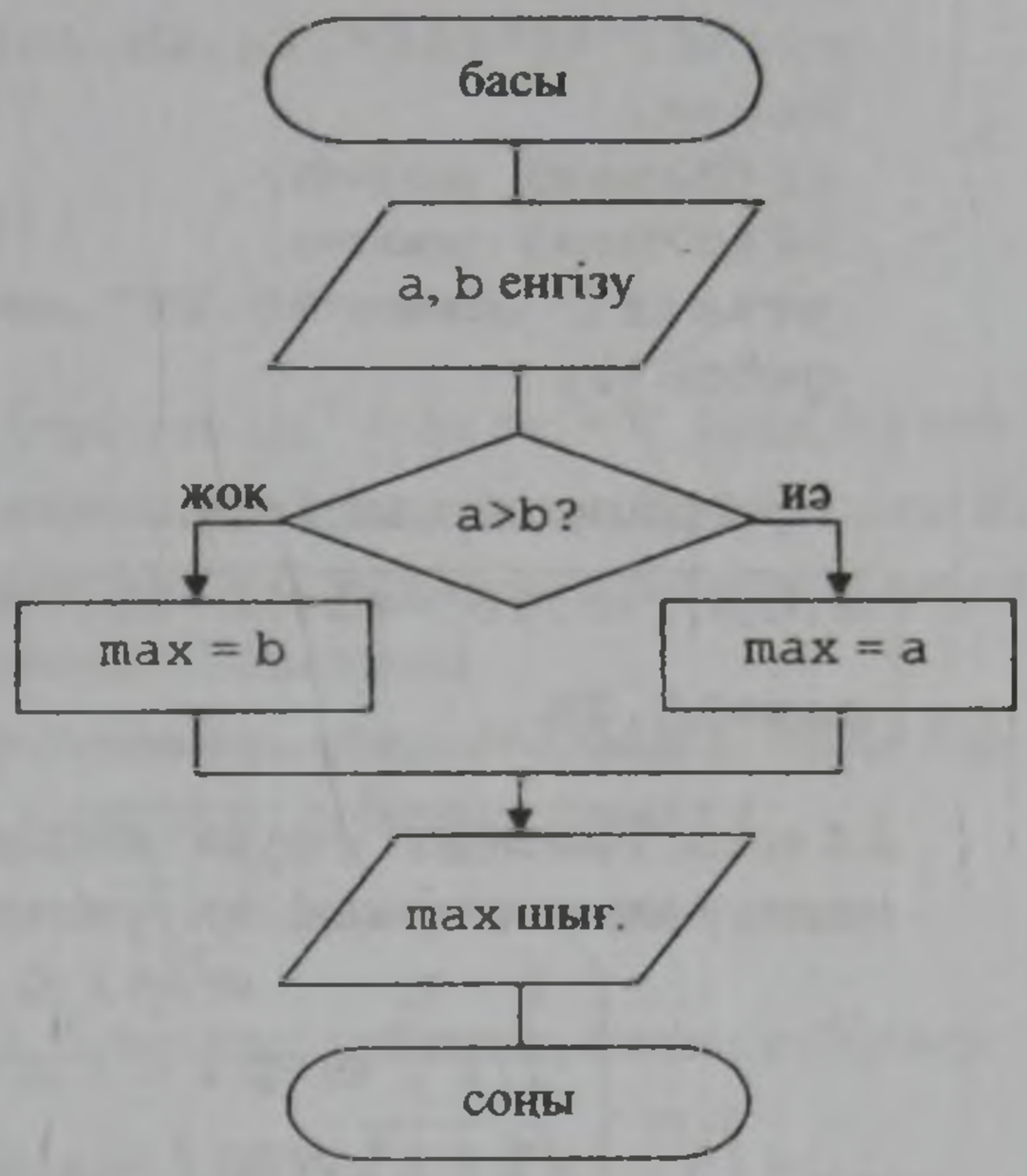
2.2-сурет. Екі нақты санның арифметикалық және геометриялық ортасын табу алгоритмі

Программа жұмысының нәтижесі:

```
a,b: 5.34 7.78

a= 5.34 b= 7.78
s= 6.56
g= 6.45
```

2.3 есеп. а) Нақты a және b сандары берілген. Сандардың үлкенін анықтайтын есептің алгоритм схемасы мен программасын құру керек. Есепті шығару алгоритмі 2.3 суретте көрсетілген.



2.3-сурет. Екі санның үлкенін табу алгоритмі

Оның С тіліндегі программасының бірінші нұсқасы:

```
#include <stdio.h>
#include <conio.h>
main()
{float a,b,max;
 printf("a,b: ");
 scanf("%f%f",&a,&b);
  if(a>b) max=a;
     else max=b;
  printf("\nmax=%5.2f",max);
}
```

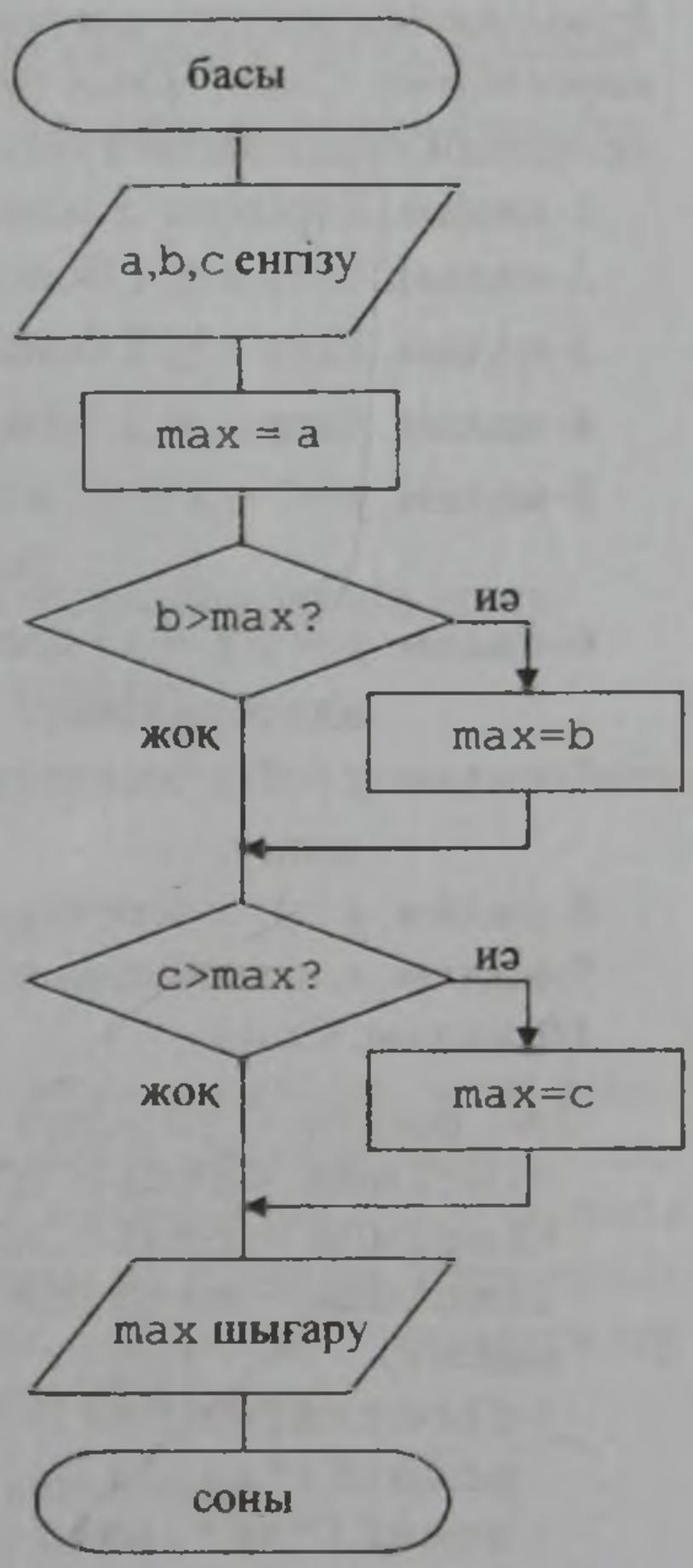
Ос есепті шығару программасының тернарлық (үш орынды) операцияны пайдалану жолымен берілген екінші нұсқасының соңғы бөлігі:

```
...
printf("a,b: ");
scanf("%f%f",&a,&b);
max=a>b?a:b;
printf("\nmax=%5.2f",max); ...
```

ә) Нақты a, b, c сандары берілген. Солардың үлкенін анықтайтын есептің алгоритм схемасы мен программасын құру керек. Есепті шығару алгоритмі 2.4 суретте көрсетілген.

Осы алгоритмнің С тілінде жазылған программасы кодының нұсқасы:

```
#include <stdio.h>
#include <conio.h>
main()
{float a,b,c,max;
```



2.4-сурет. Үш санның үлкенін табу алгоритмі

```
    printf("a,b,c: ");
    scanf("%f%f%f",&a,&b,&c);
    max=a;
    if(b>max) max=b;
    if(c>max) max=c;
    printf("\nmax=%5.2f",max);
    getch();
}
```

Программа жұмысы нәтижесі:

```
a,b,c: 3.46 -45.12 15.75

max=15.75
```

2.4 есеп. Төмендегі *y* функциясының мәнін есептеу алғоритмі мен программасын құру кажет.

$$y = \begin{cases} 1 - x, & \text{егер } x \le 1; \\ 1/x^2, & \text{егер } 1 < x \le 2; \\ \sqrt{1 + x^3}, & \text{егер } 2 < x \le 3; \\ 1 + x + x^2, & \text{егер } x > 3. \end{cases}$$

функциясын есептеу алгоритмі 2.5 суретте көрсетілген. Оның мәтін түріндегі алгоритмін келесідей түрде жазуға болады.

1-қадам. Берілген x мәнін енгізу.

2-қадам. Егер $x \le 1$ болса, 8-қадамға көшу.

3-қадам. Егер $x \le 2$ болса, 7-қадамға көшу.

4-қадам. Егер $x \le 3$ болса, 6-қадамға көшу.

5-қадам. $y = 1 + x + x^2$ мәнін есептеу және 9-қадамға көшу.

6-қадам. $y = \sqrt{1 + x^3}$ есептеу және 9-қадамға көшу.

7-қадам. $y = 1/x^2$ есептеу және 9-қадамға көшу.

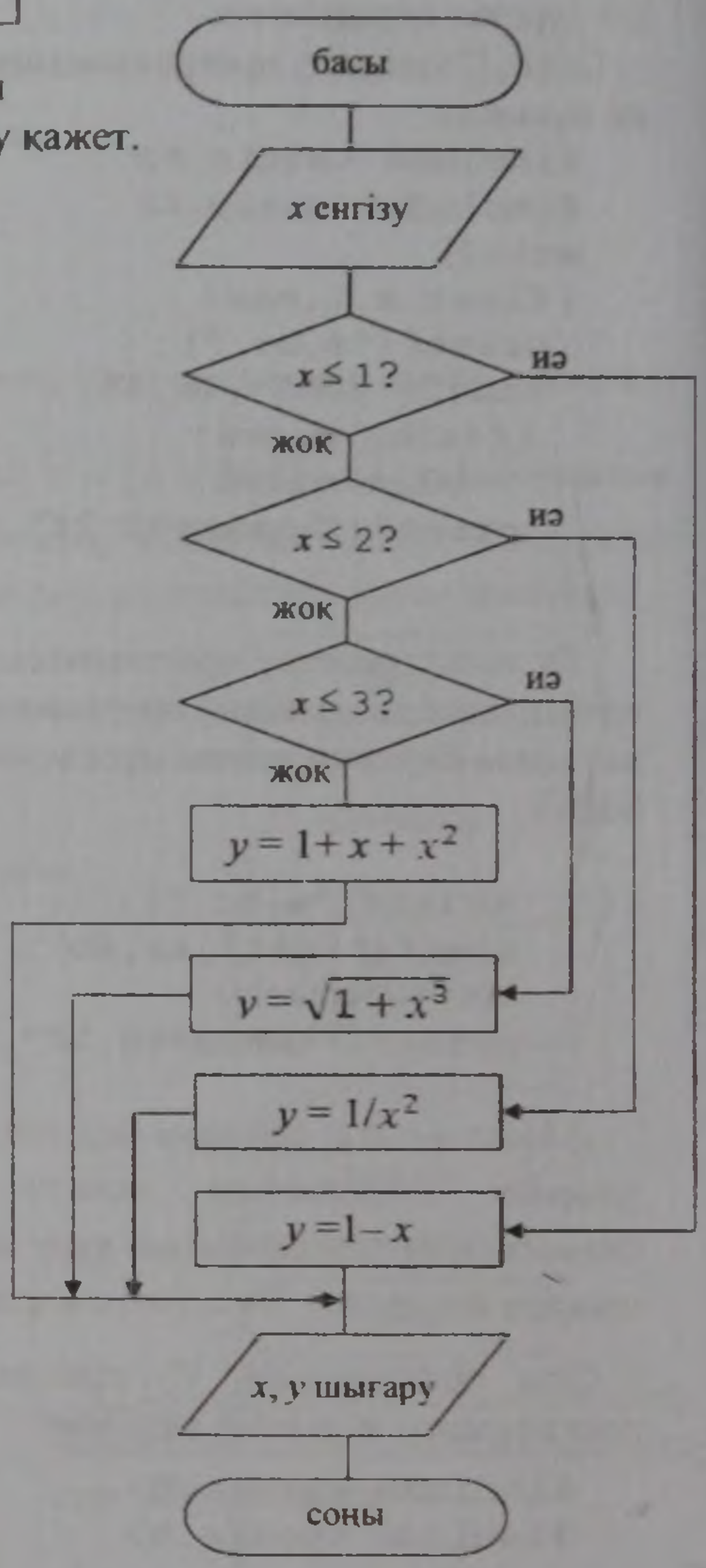
8-қадам. $y = 1 - x$ есептеу.

9-қадам. x, y мәндерін шығару.

10-қадам. Соңы.

Осы есептің С тіліндегі программасы:

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
main()
{float x,y;
 printf("x: ");
 scanf("%f",&x);
```



2.5-сурет. Тармақты алгоритм схемасы

```
    if(x<=1) y=1-x;
    else if(x<=2) y=1/x/x;
          else if(x<=3)
                y=sqrt(1+pow(x,3));
                else y=1+x+x*x;
    printf("x=%5.2f\ty=%5.2f",x,y);
    getch();
}
```

2.5 есеп. Нақты a, b, c коэффициенттері берілген $ax^2 + bx + c = 0$ квадрат теңдеуінің түбірлерін есептеу алгоритмін құру керек ($a \neq 0, b \neq 0, c \neq 0$). Мұндағы *a, b, c* нақты коэффициенттеріне байланысты квадрат теңдеудің дискриминантының $d = b^2 - 4ac$ таңбасына қарай үш түрлі түбірлердің болатыны белгілі: егер $d>0$ болса, әртүрлі екі нақты түбірлер, егер $d=0$ болса, онда бір-біріне тең нақты екі түбір, ал егер $d<0$ болса, комплекс түбірлер шығады.

Әр түрлі нақты түбірлер $x_1 = \frac{-b+\sqrt{D}}{2a}$, $x_2 = \frac{-b+\sqrt{D}}{2a}$ формулаларымен, тең түбірлер $x_1 = x_2 = -\frac{b}{2a}$ болып анықталады. Комплекс түбірлер, нақты түбірлер сияқты есептеледі, тек сандар қосарланып былай жазылады $-\frac{b}{2a} \pm i\frac{\sqrt{|D|}}{2a}$. Бірінші кезеңде *d* дискриминантының мәні есептеледі де, оның таңбасы тексеріледі. Есептеу операцияларының аздаған көлемі орындалатын алгоритм нұсқаларының бір түрі 2.6-суретте келтірілген. Төменде алгоритмнің мәтіндік формасы берілген.

1-қадам. *a, b, c* енгізу.

2-қадам. $d = b^2 - 4ac, h = 2a, k = -b/h$ есептеу.

3-қадам. Егер $d<0$ болса, 7-қадамға көшу.

4-қадам. Егер $d>0$ болса, 6-қадамға көшу.

5-қадам. $x_1 = x_2 = k$ қойып, 9-қадамға көшу.

6-қадам. $g = \sqrt{d}, r = g/h, x_1 = k + r, x_2 = k - r$ есептеп, 9-қадамға көшу.

7-қадам. $g = \sqrt{|d|}, r = g/h, x_1 = k + ir, x_2 = k - ir$ есептеу

8-қадам. Экранға: «Комплекс түбірлер $x_1 = k + ir, x_2 = k - ir$» деген мәлімет шығарып, 10-қадамға көшу.

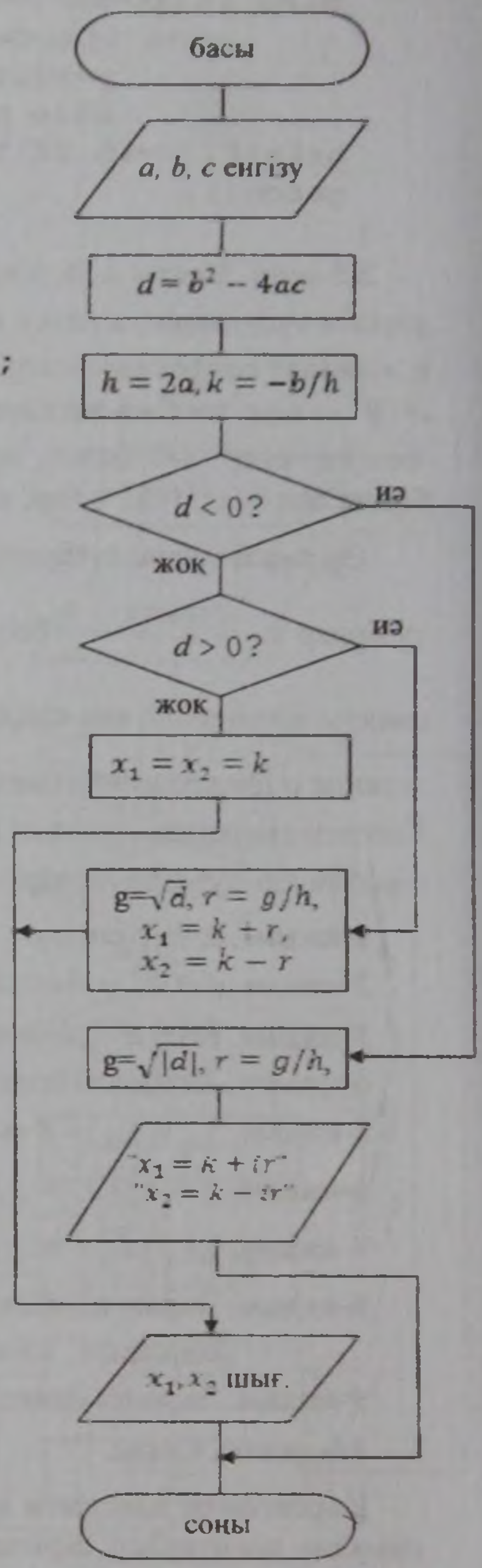
9-қадам. Экранға «нақты түбірлер» x_1, x_2 шығару.

10-қадам. Соңы.

Көрсетілген алгоритм негізінен тармақталу әрекеттерін қамтиды, яғни блок-схемада есептеулер бірнеше жолдар арқылы тармақталып шығарылады. Бұлар салыстыру блоктарымен анықталады. Әр салыстыру блогының екі шығу сызығы бар, оның біреуі шарт орындалғанда («иә») атқарылады, ал екіншісі шарт орындалмаған («жоқ») жағдайда жүзеге асырылады. Мұндай алгоритмдер тармақталған құрылымды алгоритмдер болып табылады (2.6 сурет).

Енді осы есептің С тіліндегі программалық кодын келтірелік.

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
main()
{ float a,b,c,x1,x2,h,k,g,r,d;
  printf("a,b,c: ");
  scanf("%f%f%f",&a,&b,&c);
  d=b*b-4*a*c; h=2*a; k=-b/h;
  if (d<0)
   { g=sqrt(fabs(d)); r=g/h;
     printf("complex:\tx1=%f+i*%f",k,r);
     printf("\tx2=%f-i*%f",k,r);
   }
  else
   if (d>0)
    { g=sqrt(d); r=g/h;
      printf("real:\tx1=%f",k+r);
      printf("\tx2=%f",k-r);
    }
   else
     printf("x1 = x2 = %f",k);
}
```



Программа жұмысы нәтижелері:

$a = 4{,}56$; $b = 5{,}23$; $c = -8{,}12$ болған кездегі экран бейнесі:

```
a,b,c: 4.56 5.23 -8.12
real: x1=0.878969  x2=-2.025899
```

Ал тең түбірлер шығатын $a = 1{,}0$; $b = -2{,}0$; $c = 1{,}0$ болғандағы нұсқасы:

```
a,b,c: 1.0 -2.0 1.0
real: x1=1.000000  x2=1.000000
```

Комплекс түбірлер шығатын $a = 1{,}25$; $b = 2{,}56$; $c = 45{,}8$ болған кездегі нұсқасы:

```
a,b,c: 1.25 2.56 45.8
complex: x1=1.024 + i*5.965855
         x2=1.024 - i*5.965855
```

2.6-сурет. Квадрат теңдеу түбірлерін табу алгоритмі

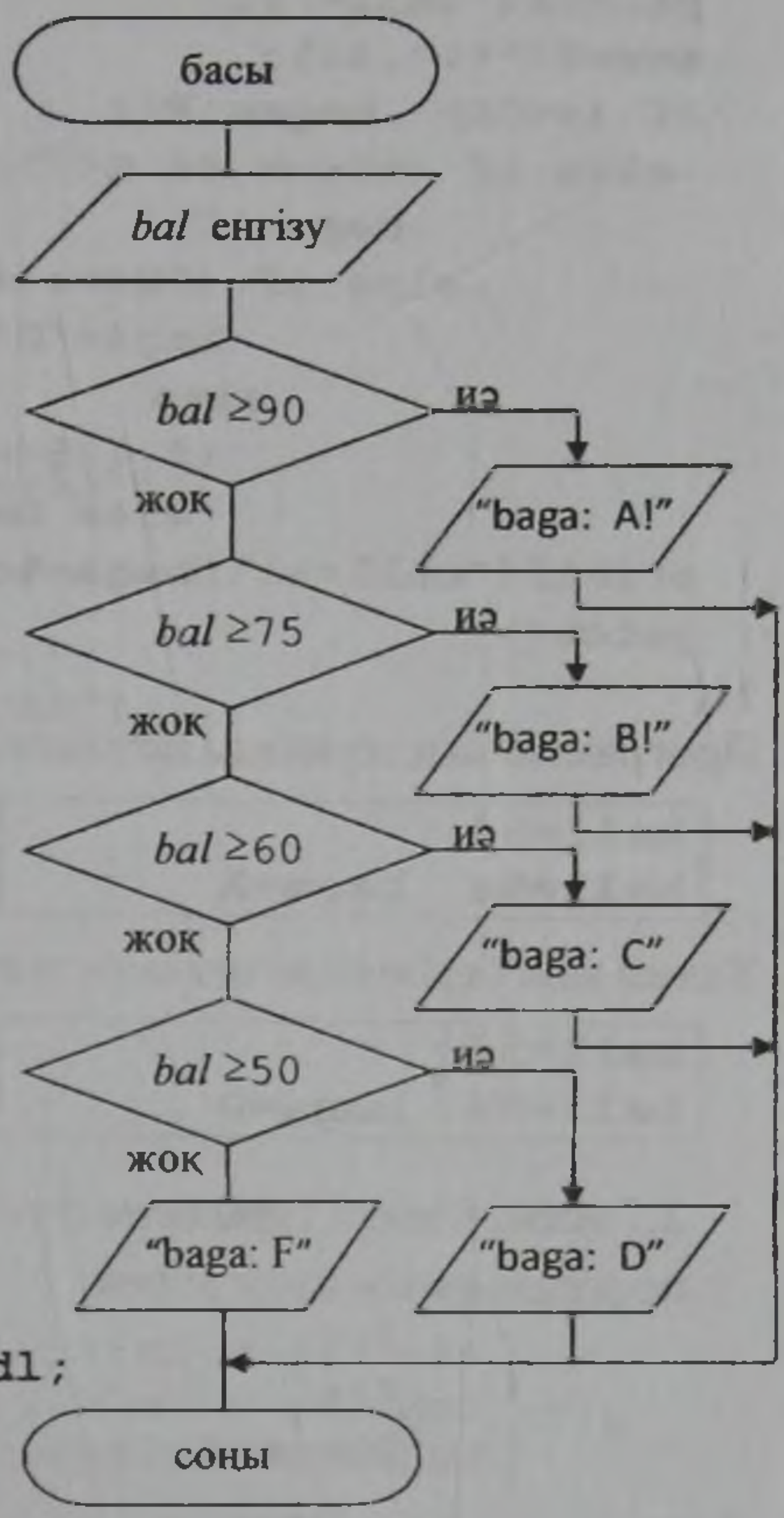
2.6 есеп. Студенттердің тест тапсыруы нәтижесінде алған ұпайлары (баллдары) 0-мен 100 аралығындағы бүтін сандар түрінде берілген. Сол сандарды негізге ала отырып, әрбір студенттің әріп жүйесіндегі бағасын анықтау алгоритмі мен программасын құру керек. Бұл алгоритм төмендегі тармақталған өрнек бойынша анықталады.

$$baga = \begin{cases} A, & егер\ 90 \le балл \le 100; \\ B, & егер\ 75 \le балл < 90; \\ C, & егер\ 60 \le балл < 75; \\ D, & егер\ 50 \le балл < 60; \\ F, & егер\ балл < 50; \end{cases}$$

Есептің алгоритмі 2.7-суретте көрсетілген.

Алгоритмнің C++ тілінде жазылған нұсқасы:

```
/* балл енгізіп, A B C D F түріндегі
   бағаны анықтау */
#include <iostream.h> // C++ тілінде
void main(void)
{ int bal;
  cout<<"Ball: ";
  cin>>bal;
  if(bal>=90)
   cout<<"Baga: A!"<<endl;
  else
    if(bal>=75)
      cout<<"Baga: B!"<<endl;
    else if(bal>=60)
           cout<<"Baga: C"<<endl;
         else if(bal>=50)
                cout<<"Baga: D" <<endl;
              else
                cout<<"Baga: F"
                    << endl;
}
```



2.7-сурет. Тест бағасын анықтау алгоритмі

Программаның жұмыс істеу нәтижелері:

```
Ball: 96
Baga: A!
```

```
Ball: 64
Baga: C
```

Алгоритмнің C тілінде жазылған екінші нұсқасы:

```
/* Балл енгізіп, бағаны анықтау */
#include <stdio.h>  // C тілінде жазылған нұсқасы
#include <conio.h>
main()
{ int x;
  char baga;
  textcolor(RED);
```

```
textbackground(WHITE);
clrscr();
printf("ball=");
scanf("%i",&x);
if (x<50)  baga='F';
 else if (60<=x && x<75)
          baga='C';
        else if (50<=x && x<60)
                 baga='D';
               else
                 if (75<=x && x<90) baga='B';
                          else baga='A';
   printf("ball=%i\tbaga=%c",x,baga);
   getch();
}
```

Программаның жұмысы нәтижесі:

```
ball=94
ball=94   baga=A
```

Келесі мән берілгенде шыккан нәтиже:

```
ball=56
ball=56   baga=D
```

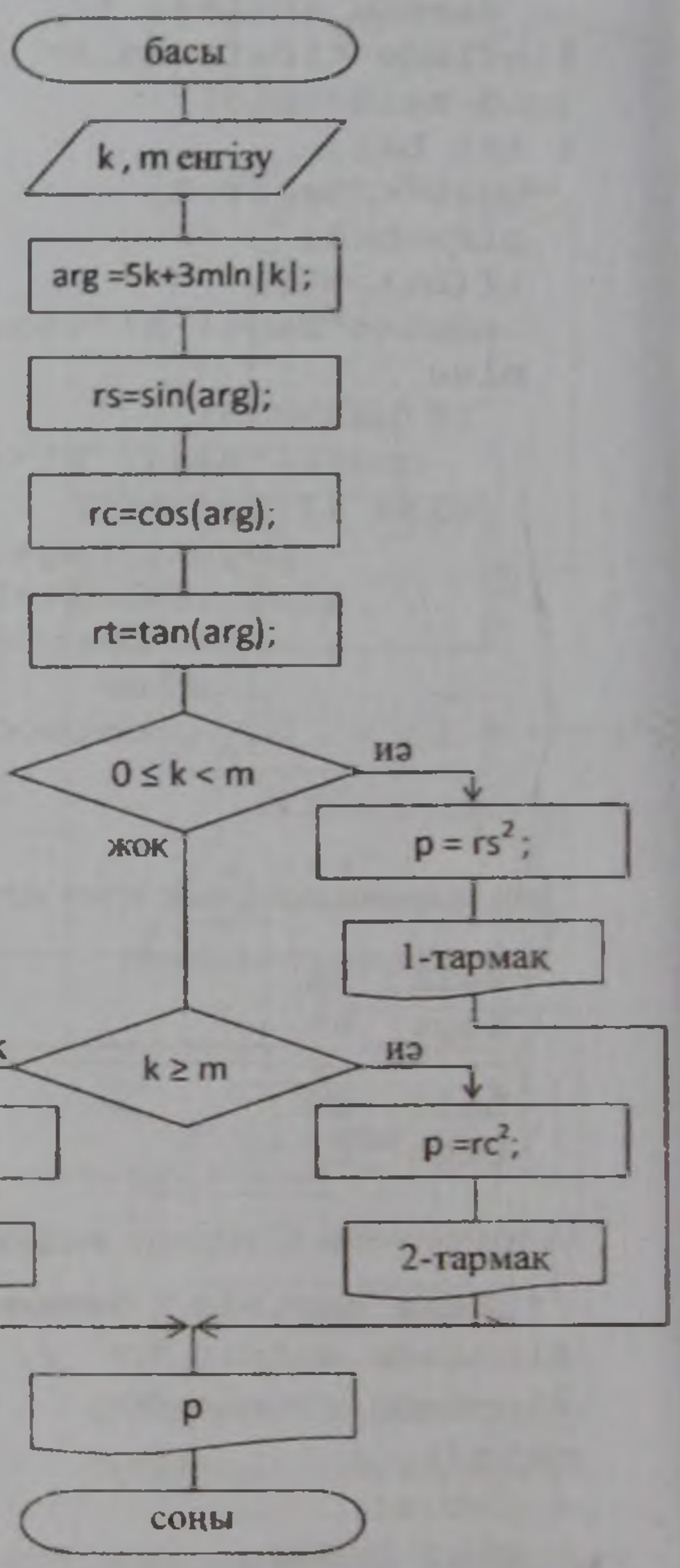
2.7 есеп. Келесі тармакты функцияны есептеу программасын құру керек:

$$p=\begin{cases}\sin^2(5k+3m\ln|k|), & 0\le k<m\\ \cos^2(5k+3m\ln|k|), & k\ge m\\ tg^2(5k+3m\ln|k|), & k<0\end{cases}$$

Есептің шығару алгоритмі оң жактағы суретте көрсетілген. Үш рет кайталанатын аргументтің және функциялардың да кыскаша түрде жазылғандарына назар аударыңдар. Мұнда есептің кай тармакпен шығарылғаны жайлы мәлімет беру керек (2.8 сурет).

С тілінде жазылған программа нұскасы:

```
#include  <stdio.h>
#include  <math.h>
  void main ()
  { float p,k,m;
    float rsin,rcos,arg,rtg;
    puts ("Enter k:" );
     scanf( "%f",&k);
    puts( "Enter m:" );
    scanf ("%f",&m);
```



2.8-сурет. Тармакты функцияны есептеу алгоритмі

```
arg = 5 * k + 3 * m * log(fabs(k));
rs = sin(arg);
rc = cos(arg);
rt = tan(arg);
if((0<=k) && (k<m))
   { p = rs*rs;
     puts("The first line works");
   }
 else if(k >= m)
         { p = rc*rc;
           puts("The second line works");
         }
        else
         { p = rt*rt;
           puts("The third line works");
         }
  printf( "Result %f", p);
}
```

Программаны орындау нәтижесі:

```
Enter k
 2
Enter m
 5
 The first line works
Result 0.999464
```

Берілген басқа мәндер үшін программаны орындау нәтижелері:

```
Enter k
5
 Enter m
4
The second line works
Result    0.894408
```

```
Enter k
-5
 Enter m
 1
 The third line works
Result    15.509675
```

2.8 есеп. Программаға бір жыл нөмірін енгізіп, сол жылдың кәбисә жыл немесе қарапайым жыл екендігін анықтау керек. Бұл алгоритмде 100-ге және 4-ке қалдықсыз бөлінетін жылдың кәбисә жыл екендігі анықталады (ақпан айы 29 күннен тұратын жыл). Мұнда C/C++ тілдерінде тек нөлге тең сан логикалық тексеру кезінде `false` (ақиқат) мәнін беретіндігі қолданылған.

```
/* Кәбисә жылды анықтау */
#include <conio.h>
#include <stdio.h>
main()
{int year, r;
 clrscr();
 printf("\nEnter year number -> ");
 scanf("%d",&year); // жылды енгізу
 if (year%100)
 { if (year%4)
      printf("It is a standard year \n"); // қарапайым жыл
   else
      printf("It is a bissextile year (leap-year)\n");
 }             // кәбисә жыл
 else
 { r=year/100;
   if (r%4)
      printf("It is a standard year \n");
   else
      printf("It is a bissextile year (leap-year)\n");
 }
 getch();
}
```

Программа жұмысының нәтижесі:

```
Enter year number -> 2012
It is a bissextile year (leap-year)
```

2.1 Горнер тәсілі бойынша көпмүшелікті есептеу алгоритмі

Көптеген математикалық есептерді шығару кезінде алгоритм түсінігі маңызды рөл атқарады. Мұндайда құрастырылған алгоритм сипаттайтын әрекеттер әртүрлі объектілермен орындала береді. Есептің шығарылу алгоритмін құрастырып, онын программасын жазып шыккан соң, есептің шарттарына көңіл бөлмей-ақ, программаны көптеген мысалдарға пайдалана беруге болады.

Бір ғана есепті шығару үшін бірнеше түрлі алгоритм құрастыруға болады. Олардың кейбірі компьютердің бірсыпыра ресурстарын (уақытын, жады көлемін, т.с.с.) пайдаланса, екіншілері оның аз ғана бөлігімен жұмыс істеуі мүмкін. Сондықтан алгоритмдерді құрастыру кезінде ең тиімді әрі тез істейтін жолды табуға тырысқан жөн. Бірнеше мысалдар карастырайық.

Көпмүшелік мәнін есептеуді оңайлатуға болады. Ол үшін берілген көпмүшелікті мынадай түрде қайта жазып шығамыз: $x^4 + 2x^3 + 3x^2 - 5x - 15 = (x^3 + 2x^2 + 3x - 5)\,x - 15 = ((x^2 + 2x + 3)\,x - 5)\,x - 15 = (((x + 2)\,x + 3)\,x - 5)\,x - 15$, содан кейін амалдарды жақшалардың аныктауы бойынша рет-ретімен біртіндеп орындаймыз. Мұнда тек жеті арифметикалық амал орындалады екен. Көпмүшелік мәнін осылай есептеу тәсілі *Горнер схемасы* деп аталады. Кез келген көпмүшелікке оны қалай пайдалануға болатынын көрсетейік.

$a_0x^3 + a_1x^2 + a_2x + a_3$ көпмүшелігі берілген делік. Оны мынадай түрге $((a_0x + a_1)x + a_2)x + a_3$ келтіреміз де, біртіндеп төменгі өрнектерді есептейміз:

$a_0 x$

$a_0 x + a_1$

$(a_0 x + a_1) x$

$(a_0 x + a_1) x + a_2$

$((a_0 x + a_1) x + a_2) x$

$((a_0 x + a_1) x + a_2) x + a_3$

Соңғы шама көпмүшеліктің керекті мәні болып шығады. Осы тәсілмен төмендегідей кез келген көпмүшеліктің мәнін табуға болады:

$$a_0x^n + a_1x^{n-1} + \ldots + a_{n-1}x + a_n$$

ол үшін оны мынадай түрге келтіріп алу керек

$$(\ldots(a_0x + a_1)x + \ldots + a_{n-1})x + a_n$$

Жаттығулар

1. $x = 2$ болғанда, $2x^3 - 3x + 5$ көпмүшелігінің мәнін Горнер схемасын пайдаланбай анықтау керек.

2. $x = 0{,}1$ болғанда, $3x^4 + 2x^2 + x + 1$ көпмүшелігінің мәнін «тікелей» алгоритммен және Горнер схемасы арқылы анықтау керек. Алынған нәтижелерді салыстырыңдар. Осы екі тәсілдің де есептеу дәлдігін анықтаңдар.

3. Горнер схемасын пайдаланып, төмендегі көпмүшеліктердің мәндерін кесте түрінде құрастырып шығындар:

а) $2x^4 - 1{,}2x^3 - 3{,}5x^2 + 4$; [1; 2] кесіндісі аралығында, қадамы – 0,2 (осы аралықта берілген функцияның экстремумдік нүктелері бар ма?);

ә) $4x^3 - 0{,}7x^2 + x - 2{,}18$; [0; 4] кесіндісі аралығында, қадамы – 0,5;

б) $x^3 - 2{,}1x^5 + 0{,}2x^2 - 4$; [-2; 2] кесіндісі аралығында, қадамы – 0,4.

№ 1 зертханалық жұмыс

Тақырыбы: Сызықтық алгоритмдер құру арқылы формулалар бойынша есептеу.

Тапсырма: Берілген формула (нұсқаларды қ.) бойынша өрнектер мәнін есептейтін программа жазып шығу керек.

Нәтижесі: Программа мәтіні мен бірнеше бастапқы мәліметтер үшін компьютерден алынған есептеу нәтижелері.

Ескерту: Формулалар бойынша есептеу барысында С тілінің төмендегі математикалық функциялары (тақырып файлы – `<math.h>`) қажет болуы мүмкін (А қосымшасын қ.).

Стандартты математикалық функциялардың сипаттамалары **1 *кесте***

Математикалық функциялар	*С тілі функциялары*	*Типтері*
e^x	`exp(x)`	`double exp(double x)`
`sin x`	`sin(x)`	`double sin(double x)`
`arcsinx`	`asin(x)` `-1 < x < 1`	`double asin(double x)`
`cos x`	`cos(x)`	`double cos(double x)`
`arccos(x)`	`acos(x)` `-1 < x < 1`	`double acos(double x)`
`tg x`	`tan(x)`	`double tan(double x)`
`arctgx`	`atan(x)` `-1 < x < 1`	`double atan(double x)`
`| x |`	`abs(x)`	`int abs(int x)`
`| x |`	`fabs(x)`	`double abs(double x)`
`ln x`	`log(x)` `x > 0`	`double log(double x)`
`lg x`	`log10(x)` `x > 0`	`double log10(double x)`
$\sqrt{x}$	`sqrt(x)` `x > 0`	`double sqrt(double x)`
x^y	`pow(x,y)`	`double pow(double x, double y)`
10^n	`pow10(n)`	`double pow10(int n)`
`x бүтін бөлігі`	`floor(x)`	`double floor(double x)`
`кездейсоқ сан алу`	`rand()`	`int rand()`

Экранға шығарылатын символдар түсін өзгерту – `textcolor (n);`
Символдар шығару кезінде экран фоны түсін өзгерту – `textbackground(n);`
мұндағы `n` – кез келген бір түске сәйкес келетін таңбасыз бүтін сан немесе түстің ағылшын тіліндегі атауы. Экранды тазалау операторы – `clrscr();`

`<conio.h>` – пернетақта мен экранға (консоль) арналған тақырып файлы; `n(түс)` параметрі төмендегі атаулы константалардың бірін пайдаланады.

Экрандағы түстер атаулары мен кодтары ***2 кесте***

Түс	*Константа*	*Константа мәні*
Қара – Черный	**BLACK**	**0**
Көк – Синий	**BLUE**	**1**
Жасыл – Зеленый	**GREEN**	**2**
Көкшіл – Бирюзовый	**CYAN**	**3**
Қызыл – Красный	**RED**	**4**
Күлгін – Сиреневый	**MAGENTA**	**5**
Қоңыр – Коричневый	**BROWN**	**6**
Боз – Светлосерый	**LIHGTGRAY**	**7**

		2 кестенің жалғасы
Сұр – Серый	**DARKGRAY**	**8**
Көгілдір – Голубой	**LIHGTBLUE**	**9**
Ақ жасыл – Светлозеленый	**LIHGTGREEN**	**10**
Ақшыл көк– Светлобирюзовый	**LIHGTCYAN**	**11**
Қызғыш – Алый	**LIHGTRED**	**12**
Ақ күлгін – Светлосиреневый	**LIHGTMAGENTA**	**13**
Сары – Желтый	**YELLOW**	**14**
Ақ (аппақ) – Белый (яркий)	**WHITE**	**15**

Жұмысты қорғауға арналған сұрақтар:

1. Алгоритм, программа, оператор ұғымдары.
2. Алгоритм қасиеттері, оның мысалдары.
3. Сызықтық алгоритмдер ұғымы, оның ерекшелігі.
4. Си тілінің алфавитіне қандай символдар кіреді?
5. Си тілінің функциялары, олардың типтері.
6. Арифметикалық өрнек ұғымы.
7. Мәліметтерді енгізу/шығару операторлары.
8. Меншіктеу операторлары, олардың жазылу тәсілдері.
9. Инкремент, декремент операциялары, жазылу форматтары.
10. Бүтін және нақты типтегі айнымалыларды сипаттап, соларды енгізіп, мәндерін есептейтін операторларды жазу жолдары.

Тапсырмалар

1. Кубтың қыры берілген. Кубтың көлемі мен бүйір бетінің ауданын анықтаңыздар.
2. Екі санның арифметикалық және геометриялық ортасын есептеп, нәтижесін шығаратын операторлар тізбегін жазыңыздар (оң сандар берілген деп есептейміз). x, y сандарының геометриялық ортасы:

$\sqrt{x \cdot y}$, ал $a_1, a_2, a_3, \ldots, a_n$ сандарының геометриялық ортасы: $\sqrt[n]{a_1 a_2 a_3 \ldots a_n}$

3. Екі санның қосындысын, айырмасын және көбейтіндісін есептейтін программа жазыңдар.
4. Берілген екі катеті бойынша тікбұрышты үшбұрыштың гипотенузасы мен ауданын есептейтін программа құрыңдар.
5. Координаттары x_1, y_1 және x_2, y_2 болып келген екі нүкте ара қашықтығын табыңдар.
6. x саны берілген. $2x^4 - 3x^4 - 3x^3 + 4x^2 - 5x + 6$, $1 - 2x + 3x^2 - 4x^3$ және

және $1 + 2x + 3x^2 + 4x^3$ өрнектерін есептейтін программа құрыңдар. Мұнда

көбейту, қосу және алу амалдарын ғана қолдануға (Горнер схемасы) болады. Орындалатын операциялар саны мүмкіндігінше аз болатын болсын.
7. Тең қабырғалы үшбұрыштың қабырғасы берілген. Осы үшбұрыштың ауданын табатын программа құрыңыздар.

8. x, y, z берілген. a, b-ны есептеу керек:

$$a = \frac{\sqrt{|x-1|}-\sqrt{|y|}}{1+\frac{x^2}{2}+\frac{y^2}{2}}; \quad b = x \cdot (arctgz + e^{-x-3}).$$

9. x, y, z *берілген*. a, b-ны есептеу керек:

$$a = \frac{3 + e^{y-1}}{1 + x^2|y - tgz|}, \qquad b = 1 + |y - x| + \frac{(y-x)^2}{2} + \frac{|y-x|^3}{3}.$$

10. x, y, z берілген. a, b-ны есептеу керек:

$$a = (1 + y) \cdot \frac{x+\frac{y}{x^2+4}}{e^{-x-2+\frac{1}{x^2+4}}}, \quad b = \frac{1+\cos(y-2)}{\frac{x^4}{2}+\sin^2 z}.$$

11. x, y, z берілген. a, b-ны есептеу керек:

$$a = y + \frac{x}{y^2 + \left|\frac{x^2}{y+\frac{x^3}{3}}\right|}, \quad b = 1 + tg^2\frac{z}{2}.$$

12. x, y, z берілген. a, b-ны есептеу керек:

$$a = \frac{2\cdot\cos\left(x-\frac{\pi}{6}\right)}{\frac{1}{2}+\sin^2 y}, \quad b = \cos^2\left(arctg\frac{1}{z}\right).$$

13. x, y, z берілген. a, b-ны есептеу керек:

$$a = \frac{1+\sin^2(x+y)}{2+|x-2x/(1+x^2y^2)}, \quad b = 1 + \frac{z^2}{3+z^2/5}.$$

14. x, y, z берілген. a, b-ны есептеу керек:

$$a = \ln\left|\left(y - \sqrt{|x|}\right)\cdot\left(x - \frac{x}{2+\frac{x^2}{4}}\right)\right|, \quad b = x - \frac{x^2}{3!} - \frac{x^5}{5!}.$$

15. Екі нақты сан берілген. Солардың арифметикалық ортасын және модульдерінің геометриялық ортасын табыңдар.
16. Үшбұрыш төбелерінің координаттары берілген. Сол үшбұрыштың периметрі мен ауданын анықтаңдар.
17. Арифметикалық прогрессия мүшелерінің $a, a + d, \ldots, a + d(n-1)$ қосындысын берілген a, d, n мәндері арқылы анықтаңдар.
18. Тікбұрышты үшбұрыштың гипотенузасы мен бір катеті берілген. Соның екінші катеті мен іштей сызылған шеңбер радиусын табыңдар.
19. Массалары m_1 және m_2, ара қашықтықтары r болып келген денелердің бір бірін тарту күшін F анықтаңдар.
20. x, y нақты сандары берілген. Көбейту, косу және алу амалдары арқылы ғана мына өрнекті $3x^2y^2 - 2xy^2 - 7x^2y - 4y^2 + 15xy + 2x^2 - 3x + 10y + 6$

есептеңдер. Амалдар саны сегіз көбейтуден және сегіз қосу мен алудан аспайтын болсын.

21. h биіктіктен жерге құлайтын тастың түсу уақытын табыңдар .
22. x, y, z берілген. a, b-ны есептеу керек:

$$a = \frac{e^{x+y} + sinx + cosy}{3x^2 + 2x + 6}, \qquad b = \ln\left|\frac{z+1}{x}\right| + z.$$

23. Берілген қабырғаларының ұзындығы a, b, c нақты оң сандар болып келген үшбұрыш тұрғызуға болатыны белгілі болса, сол үшбұрыштың ішкі бұрыштарын табыңдар.
24. Үшбұрыштың бұрыштары мен сырттай сызылған шеңбер радиусы берілген, үшбұрыш қабырғаларын анықтаңдар.
25. Шеңбер ұзындығы берілген, онын ауданын табыңдар.
26. Ішкі радиусы 20 болып келген, ал сырткы радиусы — берілген санға r ($r > 20$) тең сақина ауданын табу керек.
27. Үш R_1, R_2, R_3 кедергілері параллель қосылған. Жалпы кедергі мәнін табыңдар.
28. a саны берілген деп есептеп, төмендегі өрнек мәнін табу керек:

$$\frac{\sqrt{cos\sqrt[3]{sina}}}{2} + \frac{\sqrt{sin\sqrt[3]{cosa}}}{2}$$

29. Радиусы R берілген деп есептеп, шеңбер ұзындығын, дөңгелектің ауданын және шардың көлемін табу керек.
30. Цельсий градустарымен берілген температураны Фаренгейт градустарына төмендегі формула бойынша түрлендіретін программа құру қажет:

$$F = \frac{9}{5}C + 32,$$

мұндағы F – Фаренгейт бойынша температура; C– Цельсий бойынша берілген температура (градус).

31. Терезенің сол жақ жоғарғы бұрышына, ортасына, содан соң терезенің оң жақ төменгі бұрышына өзіңіздің аты жөніңізді үш рет шығарыңыз.Мысалы:

Burybaev Bakhyt Burybaev Bakhyt Burybaev Bakhyt

33. Берілген аракашықтықты шақырымнан километрге (1 шақырым = 1,0668 км), салмақты фунттан килограммға (1 фунт = 0,4059 кг) ауыстырып есептейтін программа құрыңыз. Программаны келесі мәліметтермен шығарыңыз: 100 шақырым және 5 фунт.
34. Пернетақтадан енгізілген нақты санды ақша форматына ауыстыратын программа жазыңыз. Мысалы, нәтижесі келесідей болу керек:

```
Input the number -> 25.8
25.8 tenge - 25 tenge 80 tyin
```

35. Цифрлар саны белгісіз бүтін сан берілген, осы санның цифрларының қосындысын есептейтін программа құрыңыз.
36. Төмендегі программа коды қандай мәндерді қайтарады? Ауызша анықтауға тырысыңыз.

```
/* Инкремент пен декременттің постфикстік және
   префикстік формалары  */
#include <iostream.h>
void main()
{
  int ik =8, il = 4, im, in;
  im = (++ik)+ (il++);
  in = (ik--) + (--il) ;
  cout<< "\n im = "    << im;
  cout<< "\n in = "    << in;
  cout<< "\n ik = "    << ik;
  cout<< "\n il = "    << il << "\n";
}
```

Енді өзіңіздің жауабыңыз бен компьютер берген жауапты салыстырыңыз.

37. Төмендегі программа орындалғаннан кейін қандай нәтиже береді?

```
// Операциялардың орындалу реттілігін тексеру
#include <iostream.h>
void main()
{    int  ii = 2;
     double dk, da = 8.2, dd =0.781;
     dk = da + (ii++)- dd * (--ii);
     cout << " k = " << dk;
     cout << " i = " << ii;
     dk = da + (++ii) - dd * (ii--);
     cout << " k = " << dk << "\n";
}
```

Өзіңіздің жауабыңыз бен компьютер берген жауапты салыстырыңыз.

Зертханалық жұмыстарға арналған тапсырмалар нұсқалары

1 нұсқа

1. C/C++ ортасындағы келесі үш математикалық функциясының `pow`, `sin`, `log10` қалай жұмыс істейтінін талдап, жазып шығындар.
2. Мына өрнекті есептеу программасын құрындар:

$$t = \frac{2\cos(x - \frac{\pi}{6})}{0{,}5 + \sin^2 y}(1 + \frac{z^2}{3 - z^2/5})$$

мұндағы $x = 14{,}26$, $y = -1{,}22$, $z = 3{,}5 \cdot 10^{-2}$.

3. Қырларының ұзындықтары a, b, c болып берілген тікбұрышты параллелепипедтің толық бетінің ауданын табындар. Қырлар ұзындықтары a, b, c пернелерден енгізіледі, мысалы: $a = 9$, $b = 7{,}5$, $c = 5$.

2 нұска

1. C/C++ ортасының келесі үш математикалық функциясының abs, acos, asin калай жұмыс істейтінін талдап, жазып шығындар.
2. Мына өрнекті есептеу программасын құрындар:

$$u = \frac{\sqrt[3]{8 + |x - y|^2 + 1}}{x^2 + y^2 + 2} - e^{|x-y|}(tg^2 z + 1)^x$$

мұндағы $x = -4{,}5$, $y = 0{,}75 \cdot 10^{-4}$, $z = 0{,}845 \cdot 10^{2}$.

3. Параллелепипедтің берілген ұзындығы (9) , ені (7,5) мен биіктігі (5) арқылы көлемін табу керек.

3 нұска

1. C/C++ ортасының келесі үш математикалық функциясының atan, ceil, cos калай жұмыс істейтінін талдап, жазып шығындар.
2. Мына өрнекті есептеу программасын құрындар:

$$v = \frac{1 + \sin^2(x + y)}{\left| x - \frac{2y}{1 + x^2 y^2} \right|} x^{|y|} + \cos^2\left(arctg \frac{1}{z} \right)$$

мұндағы $x = 3{,}74 \cdot 10^{-2}$, $y = -0{,}825$, $z = 0{,}16 \cdot 10^{-2}$.

3. Пернетактадан үшбұрыштың қабырғаларын a, b, c енгізіп және келесі формула бойынша оның медианасының ұзындығын табыңыз

$$m_b = \frac{1}{2}\sqrt{2a^2 + 2c^2 - b^2},$$

мұнда m_b – бұл B төбесінен түсірілген медиана.

4 нұска

1. C/C++ ортасының келесі үш математикалық функциясының cosh, exp, fabs калай жұмыс істейтінін талдап, жазып шығындар.
2. Мына өрнекті есептеу программасын құрындар:

$$w = \frac{1 + \sin^2(x + y)}{\left| x - \frac{2y}{1 + x^2 y^2} \right|} x^{|y|} + \cos^2\left(arctg \frac{1}{z} \right)$$

мұнда $x = 0{,}4 \cdot 10^{4}$, $y = -0{,}875$, $z = -0{,}475 \cdot 10^{-3}$.

3. Үшбұрыш қабырғалары ұзындықтарын *a*, *b*, *c* пернетақтадан енгізіп, онын биіктіктерінің ұзындықтарын төмендегі формула бойынша анықтаңдар

$$h_b = \frac{2\sqrt{p(p-a)(p-b)(p-c)}}{b}$$

мұндағы h_b — бұл *B* төбесінен жүргізілген биіктік, *p* — үшбұрыштың жарты периметрі.

5 нұска

1. C/C++ ортасының келесі үш математикалық функциясының `fmod`, `log`, `log10` қалай жұмыс істейтінін талдап, жазып шығындар.
2. Мына өрнекті есептеу программасын құрындар:

$$\alpha = \ln(y^{-\sqrt{|x|}})(x - \frac{y}{2}) + \sin^2 arctgz$$

мұнда $x = -15{,}246$, $y = 4{,}642 \cdot 10^{-2}$, $z = 20{,}001 \cdot 10^{2}$.

3. Үшбұрыш қабырғалары ұзындықтарын *a*, *b*, *c* пернетақтадан енгізіп, онын биссектрисасының ұзындықтарын төмендегі формула бойынша анықтаңдар

$$l_a = \frac{2\sqrt{bcp(p-a)}}{b+c},$$

мұндағы l_a — бұл *A* төбесінен жүргізілген биссектриса, *p* — үшбұрыштың жарты периметрі.

6 нұска

1. C/C++ ортасының келесі үш математикалық функциясының pow, pow10, sin.
2. Мына өрнекті есептеу программасын құрындар:

$$\beta = \sqrt{10(\sqrt[3]{x} + x^{y+2})}(\arcsin^2 z - |x - y|)$$

мұндағы $x = 16{,}55 \cdot 10^{-3}$, $y = -2{,}75$, $z = 0{,}15$.

3. Шар бетінде жатқан $A(x_1, y_1, z_1)$ нүктесі мен шар центрі $B(x_2, y_2, z_2)$ координаталары арқылы шар көлемін табындар. Берілген $x_1, y_1, z_1, x_2, y_2, z_2$ мәндері пернетақтадан енгізіледі.

7 нұска

1. C/C++ ортасының келесі үш математикалық функциясының sinh, sqrt, tan қалай жұмыс істейтінін талдап, жазып шығындар.
2. Мына өрнекті есептеу программасын құрындар:

$$\gamma = 5arctg(x) - \frac{1}{4}\arccos(x)\frac{x + 3|x - y| + x^2}{|x - y|z + x^2}$$

мұндағы $x = 0{,}1722$, $y = 6{,}33$, $z = 3{,}25 \cdot 10^{-4}$.

3. Төбелері берілген $A(x_1, 0)$, $B(x_2, 0)$, $C(x_2, y_3)$, $D(x_1, y_4)$ нүктелерде орналасқан трапеция периметрін табындар. Берілген x_1, x_2, y_3, y_4 мәндері пернетақтадан енгізіледі.

8 нұска

1. C/C++ ортасының келесі үш математикалық функциясының `pow10`, `exp`, `asin` қалай жұмыс істейтінін талдап, жазып шығындар.
2. Мына өрнекті есептеу программасын құрындар:

$$\varphi = \frac{e^{|x-y|}\,|x-y|^{x+y}}{arctg(x)+arctg(z)} + \sqrt[3]{x^6+\ln^2 y}$$

мұндағы $x = -2{,}235\cdot10^{-2}$, $y = 2{,}23$, $z = 15{,}221$.

3. Берілген $A(x_1, y_1)$ нүктесі арқылы өтетін және центрі $B(x_2, y_2)$ нүктесінде орналасқан шеңбер ұзындығын табындар. Берілген x_1, y_1, x_2, y_2 мәндері пернетақтадан енгізіледі.

9 нұска

1. C/C++ ортасының келесі үш математикалық функциясының `floor`, `ceil`, `tanh` қалай жұмыс істейтінін талдап, жазып шығындар.
2. Мына өрнекті есептеу программасын құрындар:

$$\Psi = \left| x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}} \right| + (y-x)\frac{\cos y - \dfrac{z}{y-x}}{1+(y-x)^2}$$

мұндағы $x = 1{,}825\cdot10^{2}$, $y = -18{,}225$, $z = -3{,}298\cdot10^{-2}$.

3. радиусы мен биіктігі берілген цилиндрдің бүйір бетінің ауданын тбындар. Берілген мәндер пернетақтадан енгізіледі.

10 нұска

1. C/C++ ортасының келесі үш математикалық функциясының `exp`, `fabs`, `pow` қалай жұмыс істейтінін талдап, жазып шығындар.
2. Мына өрнекті есептеу программасын құрындар:

$$c = 2^{(y^x)} + (3^x)^y - \frac{y(arctgz - \dfrac{\pi}{6})}{|x| + \dfrac{1}{y^2+1}}$$

мұндағы $x = 3{,}251$, $y = 0{,}325$, $z = 0{,}466\cdot10^{-4}$.

3. радиусы мен биіктігі берілген цилиндрдің толық бетінің ауданын тбындар. Берілген мәндер пернетақтадан енгізіледі.

11-нұска

1. C/C++ ортасының келесі үш математикалық функциясының `ceil`, `cos`, `fmod` қалай жұмыс істейтінін талдап, жазып шығындар.
2. Мына өрнекті есептеу программасын құрындар:

$$f = \frac{\sqrt[4]{y} + \sqrt[3]{x} - 1}{|x-y|(\sin^2 z + \operatorname{tg} z)}$$

өрнекті берілген $x = 17{,}421$, $y = 10{,}365\cdot10^{-3}$, $z = 0{,}828\cdot10^{5}$ мәндері үшін есептеңдер.

3. Радиусы мен биіктігі берілген цилиндрдің көлемін табыңыз. Радиус пен биіктіктің мәндерін пернетакта арқылы енгізу керек.

12-нұска

1. C/C++ ортасының келесі үш математикалық функциясының sin, log10, abs қалай жұмыс істейтінін талдап, жазып шығындар.
2. Мына өрнекті есептеу программасын құрындар:

$$g = \frac{y^{x+1}}{\sqrt[3]{|y-2|+3}} + \frac{x+\frac{y}{2}}{2|x+y|}(x+1)^{-1/\sin z}$$

өрнекті берілген $x = 12{,}3\cdot 10^{-1}$, $y = 15{,}4$, $z = 0{,}252\cdot 10^{3}$ мәндері үшін есептеңдер.

3. шардың радиусы берілген, шардың көлемін табу керек. Радиустың мәні пернетакта арқылы енгізіледі.

13-нұсқа

1. C/C++ ортасының келесі үш математикалық функциясының cos, atan, ceil қалай жұмыс істейтінін талдап, жазып шығындар.
2. Мына өрнекті есептеу программасын құрындар:

$$h = \frac{x^{y+1}+e^{y-1}}{1+x|y-tgz|}(1+|y-x|) + \frac{|y-x|^2}{2} - \frac{|y-x|^3}{3}$$

өрнекті берілген $x = 2{,}444$, $y = 0{,}869\cdot 10^{-2}$, $z = -0{,}13\cdot 10^{3}$ мәндері үшін есептеңдер.

3. Шардың радиусы берілген, шар бетінің ауданын табу керек. Радиустың мәні пернетакта арқылы енгізіледі.

№ 2а зертханалық жұмыс

Тармақталған алгоритмдерді программалау

Тақырыбы: Тармақты алгоритмдер құру арқылы есептер шығару.

Тапсырма: Берілген нұскалар бойынша алгоритмдер құрып, C/C++ тілінде программа жазып шығу керек.

Нәтижесі: Есепті шығару алгоритмі мен программасы мәтіні және компьютерден алынған есептеу нәтижелері.

Тапсырмалар

1. Тармақты алгоритмдерді программалау тәсілдерін пайдаланып, төмендегі функциялардың мәндерін есептейтін программа жазындар:

а) $$y = \begin{cases} \dfrac{e^{\sin x/2}+\sin e^{x/4}}{x^2+2}, & \text{егер } x < 2 \\ 2x^2 + \dfrac{1}{\sqrt{3x}}, & \text{егер } x \geq 2 \end{cases}$$

ә) $$y = \begin{cases} \sin^2 2x + 5x^2, & \text{егер } x > 1{,}66 \\ \dfrac{61\sqrt{x}-17}{\sqrt{4+x^2+\cos^2 4x}}, & \text{егер } x \leq 1{,}66 \end{cases}$$

б) $$y=\begin{cases}\dfrac{e^x+1}{e^{x^2}}, & \text{егер} \quad x<-2{,}4\\ (x+\sin 4x)+\lg x^2, & \text{егер} \quad x\ge -2{,}4\end{cases}$$

в) $$y=\begin{cases}\sqrt{(\sin x+1)}+\lg x^2, & \text{ерер} \quad x<-1\\ \dfrac{\sin x+\cos x}{\cos x}, & \text{ерер} \quad -1\le x<3\\ e^{-\sin x}+\sin x, & \text{ерер} \quad x>3\end{cases}$$

г) $$z=\begin{cases}5\sin y+\cos y, & \text{ерер} \quad y<1\\ \dfrac{y^2-2y-5}{e^y}, & \text{ерер} \quad 1\le y<4\\ \sqrt{y^2+5}+\lg y, & \text{ерер} \quad y\ge 4\end{cases}$$ мұндағы $$y=\begin{cases}\lg x+\sqrt{x}, & x\le 2\\ \lg 2x+\sqrt{3x}, & x>2\end{cases}$$

д) $$y=\begin{cases}\sqrt{x}+\sin x, & \text{ерер} \quad x>0{,}4\\ \sqrt{2-x}+\cos x, & \text{ерер} \quad -1<x\le 0{,}4\\ e^{5x-1}+\lg x, & \text{ерер} \quad x\le -1\end{cases}$$

2. Жазықтағы нүктенің координаталары бойынша оның қай ширекте жататындығын анықтайтын программа құрындар.
3. Жазықтықта үш нүкте берілген: A(*x*1,*y*1),B(*x*2,*y*2) және C(*x*3,*y*3). Осы үш нүкте арқылы үшбұрыш құруға бола ма? Егер болса, Герон формуласын пайдаланып оның ауданын есептейтін программа құрындар.
4. Квадрат теңдеуді шешу программасын құрындар ($ax^2+bx+c=0$; $a\neq 0$).
5. Үшбұрыштың төбелері координаталар арқылы берілген: A(*x*1,*y*1), B(*x*2,*y*2), және C(*x*3, *y*3). Бұл үшбұрыш тең қабырғалы, тең бүйірлі болатынын анықтау программасын құрындар.
6. Жазықтықта екі нүкте *N* (*x1,y1*) және *M*(*x2,y2*) берілген. Бұлардың қайсысы координаталар басына жақын болатынын анықтайтын программа құрындар.
7. Бүтін *n* санын берілген, ол бүтін *m* (*m*<*n*) санына қалдықсыз бөлінетінін анықтайтын программа құру керек.
8. Егер берілген *a* саны жұп болса, онда *p* атауына true, ал тақ болса false мәнін меншіктейтін программа құрындар.
9. Кез келген айдың бірінші жұлдызы аптаның қай күні екені белгілі болғанда, сол айдың енгізілген кез келген күнінің аптаның қандай күні болатынын анықтау программасын құру кажет.
10. Апта күндерінің реттік нөмірі бойынша олардың аттарын анықтайтын программа құру керек.

11. Нақты x, y ($x \neq y$) берілген. Кішісін олардың жарты қосындысымен, ал үлкенін – екі еселенген көбейтіндісімен алмастырыңыз.
12. Үш нақты сан берілген. Теріс емес сандарды квадраттаңыздар.
13. Егер берілген нақты x, y, z сандарының қосындысы 1-ден кем болса, онда бұл үш санның ең кішісін қалған екі санның жарты қосындысымен алмастырыңыз, кері жағдайда x және y-тің кішісін қалған екеуінің жарты қосындысымен алмастырыңыз.
14. Нақты a, b, c, d сандары берілген. Егер $a<=b<=c<=d$ болса, онда әр санды ең үлкен санмен алмастырыңыз, егер $a>b>c>d$ болса, сандарды өзгеріссіз қалдырыңыз, кері жағдайда барлық сандарды олардың квадратымен алмастырыңыз.
15. Нақты x, y, z сандары берілген. Егер x y-ке қалдықсыз бөлінсе және y z-ке қалдықсыз бөлінсе, онда барлық сандарға 1-ді қосыңыз, кері жағдайда барлық сандарды нөлге теңестіріңіз.
16. Нақты a, b, c, d сандары берілген. Осы сандардың терістерін квадраттап, ал оң сандарын түбірден шығарыңыз.
17. Нақты a, b, c, d сандары берілген. Егер кем дегенде бір сан нөлге тең болса, ол жайлы экранға мәлімет шығарыңыз, кері жағдайда a-ның b-ға және c-ның d-ға қалдықсыз бөлінетіндігін тексеріңіз.
18. Бүтін a, b, c сандары берілген. Егер $a<=b<=c$ болса, онда барлық сандарды олардың квадратымен алмастырыңыз; егер $a>b>c$ болса, онда әр санды ең үлкен санмен алмастырыңыз, кері жағдайда барлық сандардың таңбасын кері таңбаға ауыстырыңыз.
19. Нақты x, y, z сандары берілген. max ($x + y + z$, $x*y*z$) + 10 өрнегін есептейтін программа жазыңыз.
20. Нақты x, y, z сандары берілген. max($x^2 + y^2$, $y^2 + z^2$) – 1 өрнегін есептейтін программа жазыңыз.
21. Бүтін k, l, m сандары берілген. Нөлге тең сандардың санын анықтаңыз.
22. Бүтін k, l, m сандары берілген. Оң сандардың квадратының қосындысын есептеңіз. Егер бір де бір оң сан жоқ болса, ол жайлы экранға мәлімет шығарыңыз.
23. Бүтін x және y сандары берілген. Егер екі санда жұп болса оларға 1-ді қосыңыз; егер тек біреуі жұп болса, онда олардың көбейтіндісін табыңыз; қалған жағдайда сандарды өзгеріссіз қалдырыңыз.
24. Үш сан берілген. Солардың ішінен [0;1] аралығына кіретіндерін анықтау керек.
25. Нақты x, y, z оң сандары берілген. Қабырғаларының ұзындығы x, y, z-ке тең үшбұрыш тұрғызуға болатынын тексеріңіз. Сандарды енгізгенде олардың теріс емес және нөлге тең емес екендігін де тексеру қажет.
26. Нақты x, y, z сандары берілген. min2($x + y + z/2$, $x*y - z$) + 1 өрнегін есептеңіз.
27. Бүтін a, b, c, d сандары берілген. Нөлге тең емес сандардың көбейтіндісін табыңыз. Егер барлық сандар нөлге тең болса, экранға мәлімет шығарыңыз.
28. Бүтін a, b, c сандары берілген. Тақ сандардың қосындысын табыңыз. Егер барлық сандар жұп болса, экранға мәлімет шығарыңыз.

29. Бүтін *a, b, c, d* сандары берілген. Егер *a>=b>=c>=d* болса, онда барлық сандарды нөлге теңестіріңіз; егер *a<b<c<d* болса, онда әр санды 1-ге өсіріңіз; қалған жағдайда әр санды 1-ге кемітіңіз.
30. Бүтін *x, y, z* ($x \neq y, x \neq z, y \neq z$) сандары берілген. Осы сандардың ең кішісін тауып, онын жұп екендігін тексеріңіз.
31. Келесі функцияны шығару алгоритмі мен программасын құрыңдар:

$$a = \begin{cases} (f(x)+y)^2 - \sqrt{f(x)y}, & xy > 0 \\ (f(x)+y)^2 + \sqrt{|f(x)y|}, & xy < 0 \\ (f(x)+y)^2 + 1, & xy = 0 \end{cases}$$

Алгоритмнің барлық тармақтарының жұмыс істейтінін тексеретін бірнеше бастапқы мәндер енгізіп, нәтижелерін жазып алу қажет. Соңғы нәтижені шығару алдында программа оның қай тармақ арқылы шығарылғанын мәлімдеуі керек. Өрнектегі f(*x*) орнына: x^2 немесе e^x екеуінің бірін қою керек (немесе екіншісін де тексеріп шығындар).

Екі өлшемді жиымдар

Бұл тақырып бойынша лабораториялық жұмыстар тапсырмаларын жүзеге асыратын программалар практикумнын [11] 71 б. келтірілген.

1-нұсқа

Берілген тікбұрышты бүтін сандық матрицада:

1) нөлдік элементі жоқ жолдар санын;
2) берілген матрицада бір реттен артық кездесетін сандардың ішіндегі ең үлкенін анықтандар.

2-нұсқа

Тікбұрышты бүтін сандық матрица берілген. Нөлдік элементі жоқ бағаналар санын анықтау керек.

Бүтін сандық матрица жолының сипаттамасы деп оның оң жұп элементтерінің қосындысын атайық. Берілген матрицаның жолдарының орнын ауыстыра отырып, оларды сипаттамаларының артуы бойынша орналастырындар.

3-нұсқа

Берілген тікбұрышты бүтін сандық матрицада:

1) кемінде бір нөлдік элементі бар бағаналар санын;
2) бірдей элементтердің ең ұзақ тізбегі орналасқан жолдың нөмірін анықтандар.

4-нұсқа

Берілген квадратты бүтін сандық матрицада:

1) теріс элементтері жоқ жолдардағы элементтер көбейтіндісін;
2) матрицаның бас диагоналына параллель орналасқан диагональдердің элементтерінің қосындыларының арасындағы ең үлкенін анықтандар.

Ескерту:

3) Егер матрицаның A_{ij} элементі *i*-ші жолдағы ең кіші және *j*-ші бағанадағы ең үлкен элементі болса, онда ол А матрицасының қайқы нүктесі болып табылады.

5-нұсқа

Берілген квадратты бүтін сандық матрицада:

1) теріс элементтері жоқ бағандардағы элементтердің қосындысын;
2) матрицаның қосалқы диагоналына параллель орналасқан диагональдарының элементтері модульдерінің қосындыларының ішіндегі ең кішісін анықтандар.

6-нұсқа

Берілген бүтін сандық тікбұрышты матрицада:

1) кемінде бір теріс элементі бар жолдардағы элементтерінің қосындысын;
2) матрицаның барлық қайқы нүктелерінің жолдары мен бағандарының нөмірлерін анықтандар.

7-нұсқа

Берілген 8x8 өлшемді матрица үшін оның k нөмірлі жолы k нөмірлі бағанасымен бірдей болып келетін k мәндерін табыңдар.

Матрицаның кемінде бір теріс элементі бар жолдарындағы элементтерінің қосындысын табыңдар.

8-нұсқа

Бүтін сандық матрица бағанасының сипаттамасы деп теріс тақ элементтерінің модульдерінің қосындысын атайық. Берілген матрица бағаналарының реттілігін сипаттамаларының артуы бойынша орналастырыңдар.

Матрицаның кемінде бір теріс элементі бар бағаналарындағы элементтерінің қосындысын табыңдар.

9-нұсқа

Матрицадағы A_{ij} элементінің көршілері деп $i-1 \le k \le i+1$, $j-1 \le l \le j+1$, $(k, l) \ne (i, j)$ теңсіздіктері орындалатын A_{kl} элементтерін атаймыз. Матрицаны тегістеу операциясы деп өлшемі бастапқы матрицамен бірдей, бірақ оның әрбір элементі бастапқы матрицаның сәйкес элементі көршілерінің арифметикалық ортасы ретінде анықталатын жаңа матрицаны айтады. Берілген 10x10 өлшемді нақты матрицаға тегістеу операциясын қолданып, жаңа матрица құру керек.

Тегістелген матрицадағы бас диагональдан төмен орналасқан элементтер модульдерінің қосындысын табыңдар.

10-нұсқа

Егер матрицаның элементі оның барлық көршілерінен міндетті түрде кіші болып келетін болса, онда ол жергілікті минимум деп аталады. Берілген 10x10 өлшемді матрицаның жергілікті минимумдарының санын табыңдар.

Матрицаның бас диагоналынан жоғары орналасқан элементтері модульдерінің қосындысын табыңдар.

11-нұсқа

Сызықтық теңдеулер жүйесінің коэффициенттері тікбұрышты матрица түрінде берілген. Орындауға болатын түрлендірулер көмегімен жүйені үшбұрышты түрге келтіріңдер.

Матрица элементтерінің арифметикалық ортасы берілген шамадан кіші болатын жолдар санын табыңдар.

12-нұсқа

Берілген матрицадағы нөлдерден тұратын жолдар мен бағаналарды алып тастау арқылы оны тығыздап ықшамдау керек.

Кемінде бір оң элементі бар алғашқы жолдың нөмірін табыңдар.

13-нұсқа

Тікбұрышты матрица элементтерін *n* элементке оңға немесе төмен қарай (көрсетілген бағытқа байланысты) циклдік түрде ығыстыру керек. *n* санының мәні жолдағы немесе бағанадағы элементтер санынан артық болуы мүмкін.

2ә зертханалық жұмыс. Көпнұсқалы таңдау операторы

Тапсырма: Қойылған есепті шешуге арналған программа жазыңыздар.

Есеп беру: Блок-схема және программа мәтіні.

Нәтижесі: Берілген аралықтарды толық тексеруге арналған мәндерді қамтитын мәліметтер үшін программа нәтижелерін келтіру.

Бақылау сұрақтары

1. Таңдау (ауыстырғыш) операторының жазылуы.
2. If операторының switch операторынан айырмашылығы.
3. Көшу операторларының пайдаланылуы.
4. Бір нұсқаның бірнеше тұрақты мәндерді қамтитын мүмкіндіктері.
5. Есептелген өрнек мәнінің берілген тұрақты мәндеріне сәйкес келмеген жағдайда switch операторының орындалуы.

Таңдау операторына (switch) берілген есептер

1. 1 мен 7 аралығындағы кез келген бүтін сан берілген. Сол санға сәйкес апта күнінің аттарын ағылшынша және қазақша жазып шығаратын программа құру керек. (1 – «Дүйсенбі – Monday», 2 – «Сейсенбі … » және т.с.с.). Апта күндерінің нөмірлеріне сәйкес келмейтін сандар енгізілсе, «Қате» деп мәлімет шығару керек.
2. 1 мен 5 аралығындағы бүтін сан берілген. Осы санға сәйкес бағаның сипаттамасын шығарыңыз (1 – «нашар», 2 – «канағаттанарлықсыз», 3 – «канағаттанарлық», 4 – «жақсы», 5 – «өте жақсы»). Егер бүтін сан 1–5 аралығында жатпаса, экранға «Қате» деген мәлімет шығарыңыз.
3. Айлардың реттік нөмірін (1-12) енгізіп, соған сәйкес ай аттарын қазақша (ағылшынша) жазып шығаратын программа құру керек.
4. Айдың реттік нөмірі 1–12 арасындағы бүтін сан арқылы берілген (1 – қаңтар, 2 – ақпан және т.с.с.). Соларға сәйкес жыл мезгілдерінің атын шығарыңыз («қыс», «көктем», «жаз», «күз»).
5. Айдың реттік нөмірі 1–12 арасындағы бүтін сан арқылы берілген (1 – қаңтар, 2 – ақпан және т.с.с.). Сол айларға сәйкес айдағы күндер санын анықтаңыз.

6. Уақытты (тек сағатты) енгізіп, сол уақытқа сәйкес: "Қайырлы таң", "Қайырлы күн", "Кеш жарық", "Ұйқың тәтті болсын!" деген мәліметтер шығарындар.
7. Арифметикалық амалдар келесі ретте нөмірленген: 1 – қосу, 2 – азайту, 3 – көбейту, 4 – бөлу. Амал нөмірі – N (1–4 аралығындағы бүтін сан) және нақты A мен B ($B \neq 0$) сандары берілген. Осы сандармен нөмірге сәйкес амалды орындап, нәтижесін шығарыңыз.
8. Ұзындық өлшем бірліктері келесі ретте нөмірленген: 1 – миллиметр, 2 – сантиметр, 3 – дециметр, 4 – метр, 5 – километр. Ұзындық бірлігі (1–5 аралығындағы бүтін сан) мен сол бірлікте берілген кесіндінің ұзындығын енгізіп, нәтижесін метрмен жазып шығыңыз.
9. Салмақ өлшем бірліктері келесі ретте нөмірленген: 1 – миллиграмм, 2 – грамм, 3 – килограмм, 4 – центнер, 5 – тонна. Салмақ бірлігі (1–5 аралығындағы бүтін сан) мен сол бірлікте берілген дененің салмағын енгізіп, нәтижесін килограммен жазып шығыңыз.
10. Отыратын орындықтарының санын программаға енгізіп, соған сәйкес көлік атын көрсетіңіз, мысалы: "велосипед", "мотоцикл", "жеңіл автомобиль", "микроавтобус", "автобус" (басқа да варианттары болуы мүмкін).
11. Кәбисә емес (невисокосный) жылдың датасын анықтайтын екі сан: D (күн) және M (ай) берілген. Осы датаға сәйкес соның алдындағы D мен M –ді шығарыңыз.
12. Кәбисә емес жылдың датасын анықтайтын екі сан: D (күн) және M (ай) берілген. Осы датаға сәйкес сонан кейінгі D мен M –ді шығарыңыз.
13. Робот 4 бағытта («С» – солтүстік, «Б» – батыс, «О» – оңтүстік, «Ш» – шығыс) қозғала алады және үш санмен берілген командаларды: 0 – қозғалысты жалғастыру, 1 – солға бұрылу, 1 – оңға бұрылу орындай алады. Бастапқы берілген бағыт – C символына, команда – N бүтін санына сәйкес келеді. Осы командадан кейінгі роботтың бағытын анықтаңыз.
14. Локатор 4 бағыттың («С» – солтүстік, «Б» – батыс, «О» – оңтүстік, «Ш» – шығыс) бірін таңдай алады және үш санмен берілген командаларды: 1 – солға бұрылу, 2 – оңға бұрылу, 3 – 180°-қа бұрылу орындай алады. Локатордың бастапқы бағыты – C, N1 және N2 бүтін сандарына сәйкес екі команда берілген. Осы командалар орындалғаннан кейінгі локатордың бағытын анықтаңыз.
15. Шеңбер элементтері келесі түрде нөмірленген: 1 – радиус (R), 2 – диаметр ($D = 2 \cdot R$), 3 – ұзындық ($L = 2 \cdot \pi \cdot R$), 4 – шеңбер ауданы ($S = \pi \cdot R^2$). Осы элементтердің бірінің нөмірі және оның мәні берілген. Шеңбердің қалған элементтерінің мәнін берілген реті бойынша есептеп шығарыңыз.
16. Тең бүйірлі тікбұрышты үшбұрыштың элементтері келесі түрде нөмірленген: 1 – катет (a), 2 – гипотенуза ($c = a \cdot (2)^{1/2}$), 3 – гипотенузаға түсірілген биіктік ($h = c/2$), 4 – аудан ($S = c \cdot h/2$). Осы элементтердің бірінің нөмірі және оның мәні берілген. Қалған элементтердің мәнін берілген реті бойынша есептеп шығарыңыз.

17. Тең бүйірлі үшбұрыштың элементтері келесі түрде нөмірленген: 1 – қабырғасы (а), 2 – іштей сызылған шеңбердің радиусы ($R_1 = a\cdot(3)^{1/2}/6$), 3 – сырттай сызылған шеңбердің радиусы ($R_2 = 2\cdot R_1$), 4 – ауданы ($S = a^2\cdot(3)^{1/2}/4$). Осы элементтердің бірінің нөмірі және оның мәні берілген. Қалған элементтердің мәнін берілген реті бойынша шығарыңыз.
18. Шығыс календарында 12 жылдық ішкі циклден тұратын 60-жылдық цикл бар. Ішкі цикл белгілі бір түспен (жасыл, қызыл, көк, ақ және қара), ал оның әр жылы жануарлар атымен аталған (тышқан, сиыр, барыс, қоян, ұлу, жылан, жылқы, қой, мешін, тауық, ит, доңыз). Жыл нөмірі бойынша оның аты мен түсін анықтаңыз, егер циклдегі жыл басы 1984 болса, ол – «жасыл тышқан жылы» болады.

3 ЦИКЛДІК АЛГОРИТМДЕР МЕН ПРОГРАММАЛАР ҚҰРУ

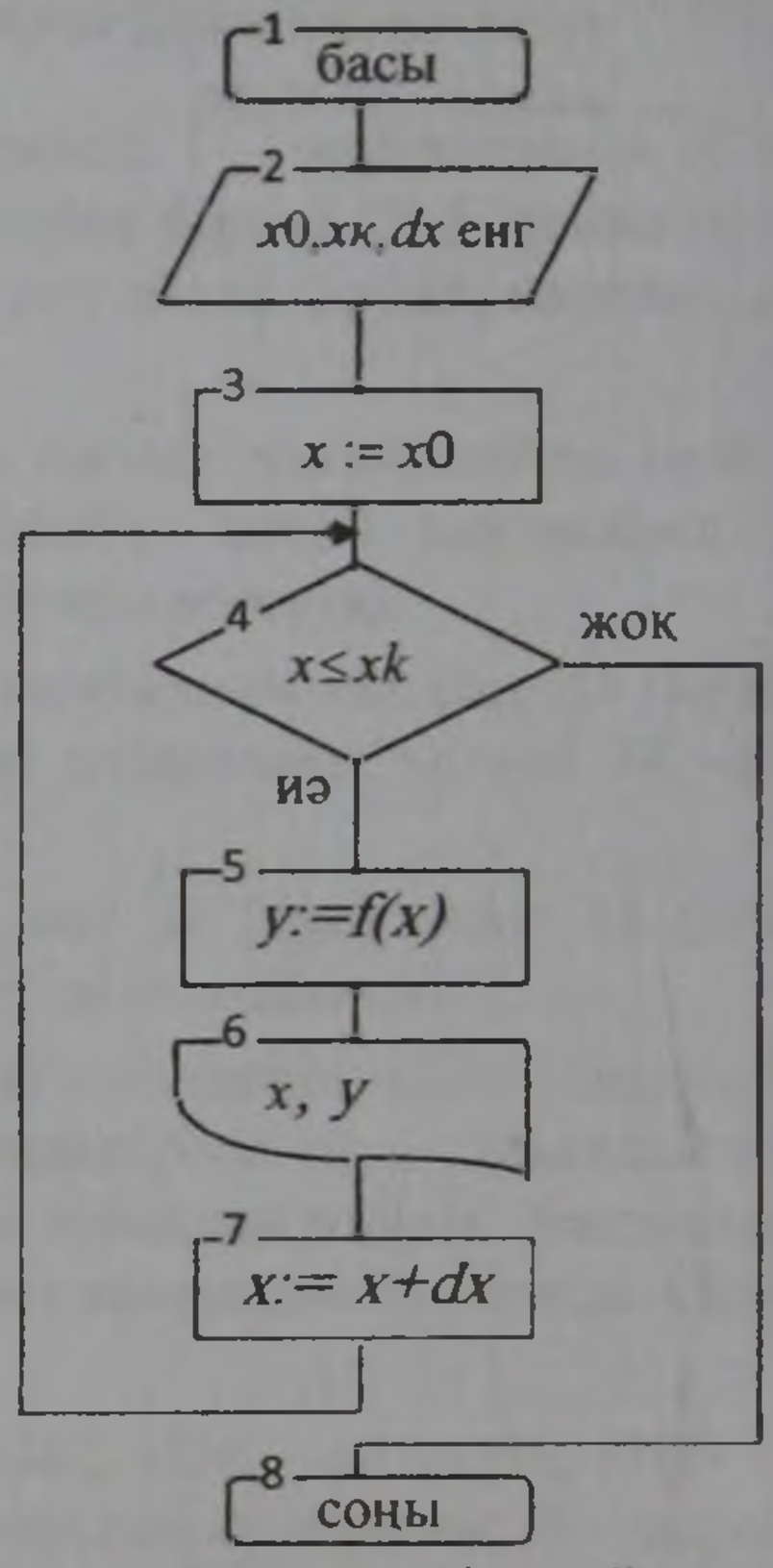
Енді құрамында есептеудің қайталау фрагменттері болатын циклдік алгоритмдерді карастырайық. Циклдік алгоритмдерді пайдалану оларды кейіннен программаларда цикл операторы түрінде қыскартып жазу мүмкіндігін береді. Циклдер қайталану санының алдын ала белгілі және белгісіз болуына байланысты екі топка бөлінеді. Қайталану сандары алдын ала белгілі болып келген циклдер тобы *арифметикалық цикл* болып есептеледі, ал орындалу саны белгісіз циклдер – *қадамдық (итерациялық) цикл* болып аталады.

Практикада белгілі бір айнымалының сандық мәніне байланысты орындалатын арифметикалық циклдер жиі кездеседі. Мұнда арифметикалық прогрессияға ұқсас болып келетін циклдер ең карапайым арифметикалық цикл болып табылады. Оны баскару кайталану кезеңінде прогрессияның заңына сәйкес тұрақты шамаға өзгеріп отыратын цикл параметрінің сандық мәнімен байланысты болуы тиіс.

Цикл орындалуы алдында оның айнымалы аргументі – параметрі алғашқы мәнге ие болуы керек, сонан кейін қайталану кезеңінде цикл параметрі белгілі бір шамаға (кадамға) өзгере отырып, ол алдын ала берілген ең соңғы мәнге дейін жетуі кажет.

Алгоритмнің орындалу барысында цикл параметрі, мысалы, x өзінің ең алғашқы $x0$ мәнінен ең соңғы xk мәніне дейін тұрақты шамага (dx) өзгеріп отырады. Осының нәтижесінде x мынадай мәндерді кабылдайды: $x0$, $x0+dx$, $x0+2dx$, ..., $x0+(n-1)\,dx$, xk, мұндағы n – циклдің қайталану саны, ол былай анықталады:

$$n = \left[\frac{xk - x0}{dx} \right] + 1 ,$$



3.1-сурет. Карапайым циклдік алгоритм

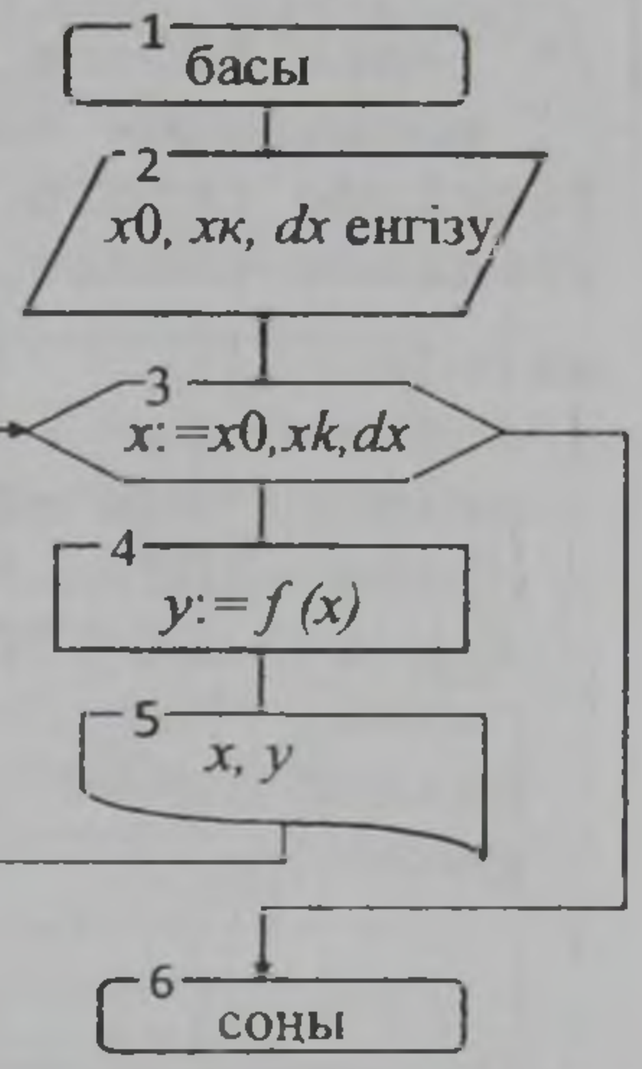
мұнда [...] – өрнектің бүтін бөлігі алынатынын көрсетеді. n әрқашанда бүтін сан болуы тиіс, егер ол аралас сан болса, онда оның бөлшегі алынып тасталады, өйткені циклдің қайталану саны бүтін натуралдық сан болуы тиіс.

Арифметикалық цикл үшін $y=f(x)$ функциясын есептеу алгоритмі 3.1-суретте көрсетілген. Мұндағы 3-ші, 4-ші, 7-блоктар циклді ұйымдастыру үшін кажет. Олар цикл параметрінің алғашқы мәнін, өзгеру кадамын белгілеп және оның ең соңғы мәніне жеткен-жетпегенін тексереді. Ал 5- және 6-блоктар бірнеше рет қайталанып циклдің өзін құрайды. 4-блок шартты тексеріп қайталану процесін ұйымдастырады.

Алгоритм схемасын салу мен программаны жазуды жеңілдету үшін цикл алгоритмдері ықшамдалған түрде "модификатор" немесе "цикл басы" блогын пайдалану арқылы жазылады (3.2-сурет). Онда 3.1-суретте көрсетілген 3-ші, 4-ші, 7-блоктардың орнына "цикл басы" блогы орналасады.

Ол алтыбұрыш тәрізді геометриялық фигурадан тұрады да, оның міндетті түрде екі кіру және екі шығу сызығы болуға тиіс. Осы блокты пайдалану арқылы жоғарыда келтірілген алгоритм 3.2-суреттегідей түрде кескінделеді. Параметрдің алғашқы x мәні оның соңғы x мәнінен кем болса, онда оның қадамы dx оң сан болады. Керісінше, параметрдің алғашқы мәні оның соңғы мәнінен артық болса, онда қадам теріс сан болады.

Программалауда арифметикалық прогрессияға ұқсас болып келетін параметрлі цикл **for** операторы арқылы және қадамдық циклдер **while**, **do..while** операторлары арқылы жүзеге асырылады.



3.2-сурет. Модификаторлы циклдік алгоритм

3.1 For операторы

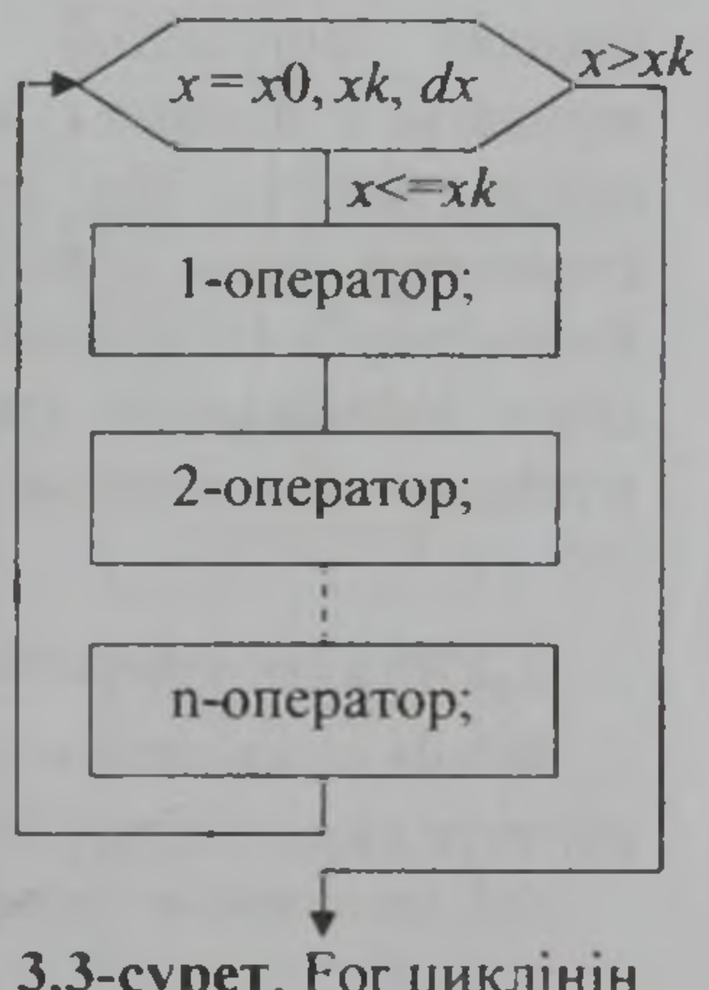
For операторы айнымалы ретінде берілген цикл параметрінің алғашкы, соңғы мәні мен өзгеру қадамы белгілі болғанда, соған сәйкес бір немесе бірнеше операторларды қайталап орындау кезінде колданылады. Бұл оператор *параметрлі цикл* операторы немесе *арифметикалық цикл* деп аталады.

Параметрлі циклдің орындалу схемасы 3.3-суретте көрсетілген.

For цикл операторының жалпы жазылу түрі:

```
for(x=x0; x<=xk; x=x+dx)
     {
       1-оператор;
       2-оператор;
          . . .
       n-оператор;
     }
```

Мұнда **x=x0** – цикл айнымалысының бастапкы мәні, **x<=xk** – циклдің орындалу шарты, **x=x+dx** – цикл айнымалысының қадамы. **x=x0** цикл операторы орындаларда бір рет есептеледі, **x<=xk** акикат болса немесе 0-ге тең болмаса, цикл тұлғасы ретіндегі **операторлар** аткарылады. Содан соң **x=x+dx** есептеледі және **x<=xk** мәні кайта анықталады. **x<=xk** мәні жалған болса немесе жалпы жағдайда ол 0-ге тең болса, **for** операторының жұмысы аяқталады. Сонымен, цикл тұлғасының келесі орындалуы немесе орындалмауы оның аткарылуы алдында анықталады.

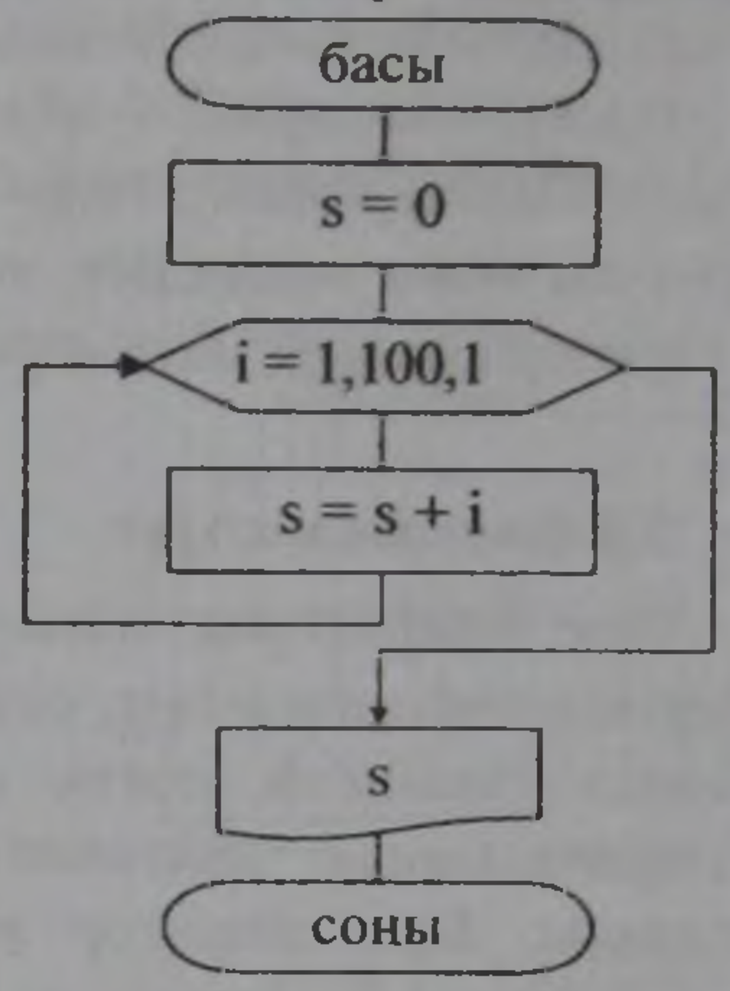


3.3-сурет. For циклінің орындалу алгоритмі

Арифметикалық циклдерде, циклдің санауышы рөлін аткаратын, яғни оны баскаратын айнымалы (**x**) көмегімен есептеулердің қайталану саны анықталады. Әрбір келесі есептеуде санауыш мәні белгілі бір шамаға (**dx**) өзгертіледі де, ол алдын ала берілген айнымалының соңғы мәнімен (**xk**) салыстырылады. Егер айнымалының жаңа мәні осы қайталану санымен дәл келсе, цикл аяқталады да,

келесі әрекеттер орындала бастайды, яғни санауыш мәні бойынша циклден шығу ісі жүзеге асады. Егер циклдің басында берілген санауыш мәні оның көрсетілген соңғы мәнінен артық болса, цикл мүлде орындалмайды. Сонымен қатар циклден белгілі бір шарт арқылы оның аяғына жетпей, шығып кету мүмкіндігі де (**break** операторы) қарастырылған. Мысалы, 1-ден 100-ге дейінгі сандар қосындысын табу алгоритмі (3.4-сурет):

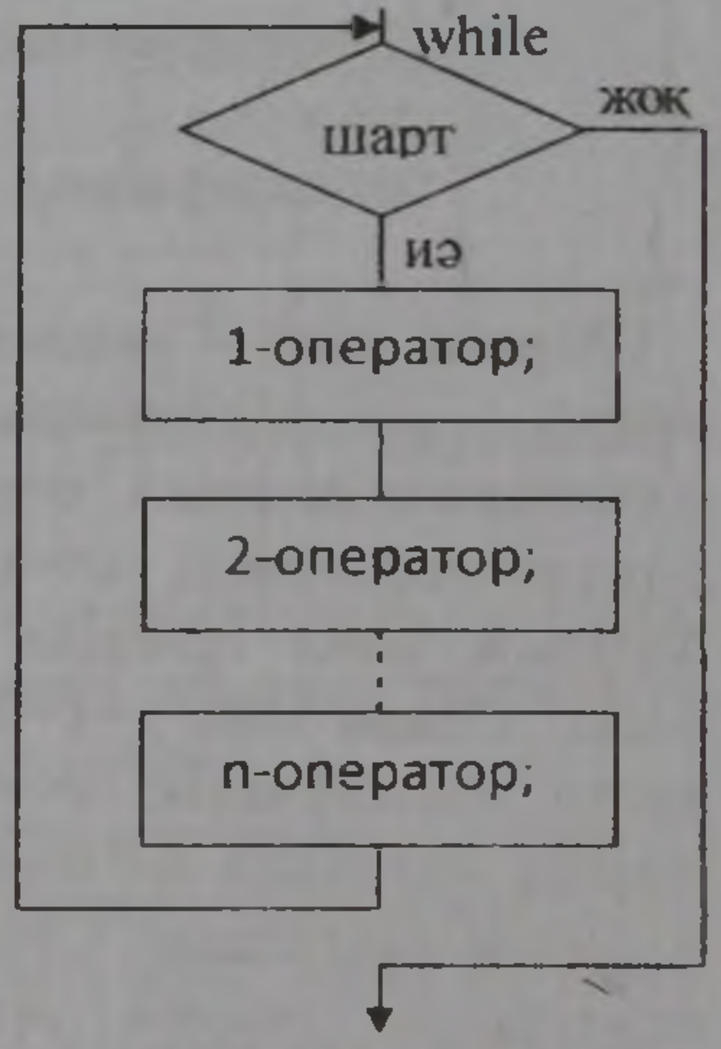
```
/* 1-ден 100-ге дейінгі сандар
   қосындысын анықтау */
#include <stdio.h>
#include <conio.h>
main()
{ int s=0,i;
  printf("sum of numbers
  (1+2+…+100): ");
  for(i=1;i<=100;i++)
   s+=i;
  printf("s=%d",s);
  getch();
}
```



3.4-сурет. Қосынды табу алгоритмі

Итерациялық типтегі циклдерде, оның қайталану саны бастапқы кезде белгісіз болып табылады. Оның жұмысының аяқталуы (циклден шығу) белгілі бір шарт орындалғанда ғана жүзеге асады.

Әдетте бұл шарт, мысалы, мынадай теңсіздік: $x \leq c$ орындалып тұрғанша (ақиқат болғанша) қайталану жүргізіле береді, мұндағы *c* тұрақты, ал *x* циклдің әрбір қайталануы сайын белгілі бір шамаға өзгеріп отыратын цикл параметрі, яғни санауыш рөлін атқаратын айнымалы. Қайталану шарты цикл басында тексерілетін жағдайда, цикл алғышартты (шарты алдын ала тексерілетін) **while** циклiне жатады.



3.5-сурет.Шарты алдын ала тексерілетін цикл

3.2 While операторы

While цикл операторының орындалу схемасы 3.5-суретте көрсетілген. Оның жазылуы:

```
while (шарт-өрнек)
     {
        1-оператор;
        2-оператор;
           . . .
        n-оператор;
     }
```

Мұнда шарт ретінде шартты өрнек немесе кез келген типтегі өрнек пайдаланылуы мүмкін. Оператор қарапайым немесе құрама болуы мүмкін. Ол құрама

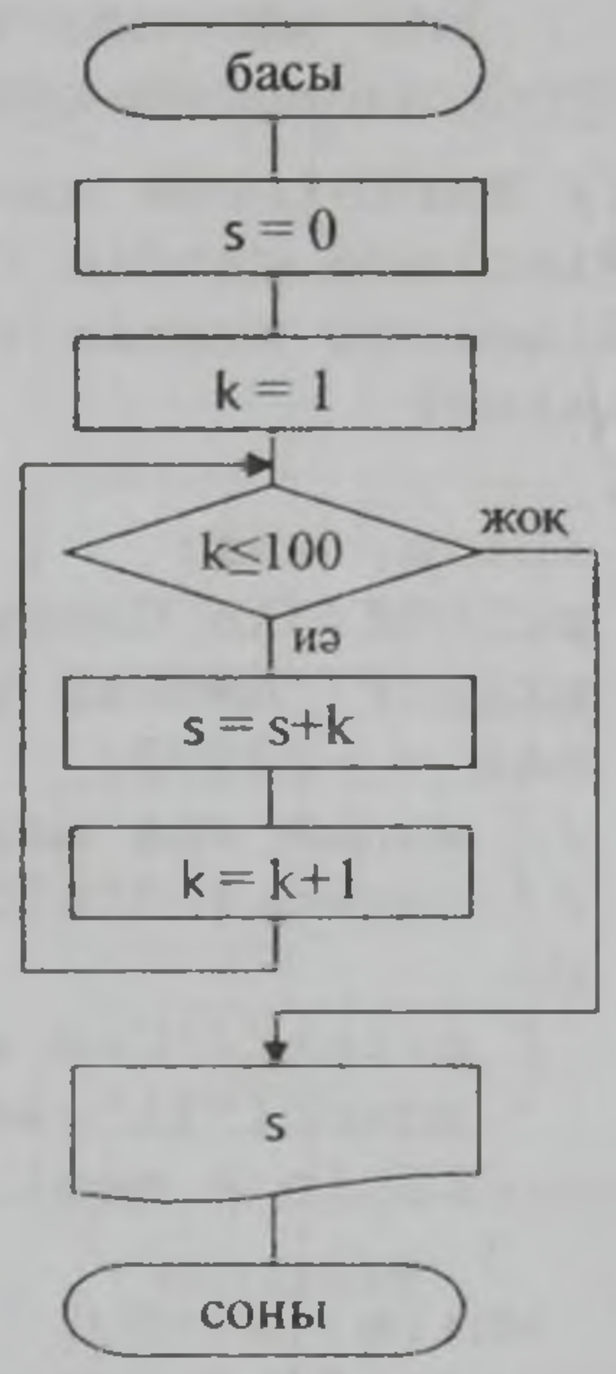
оператор болса, онда операторлар жиыны жүйелі жақшаға алынып жазылады. **While** операторы орындалғанда, алдымен, жақша ішіндегі өрнек есептеліп тексеріледі. Егер өрнек мәні ақиқат болса немесе жалпы жағдайда 0-ге тең болмаса, онда **оператор** атқарылады. Содан соң жақшадағы өрнек тағы да есептеледі. Егер өрнек мәні жалған болса (немесе жалпы жағдайда 0-ге тең болса), онда **while** цикл операторы өз жұмысын аяқтайды.

While операторының орындалу саны шексіз болып кетпес үшін, мұнда шарт-өрнек құрамына кіретін айнымалы цикл ішінде өзгеріп отырады.

Мысал (3.6-сурет).

```
/* 1-ден 100-ге дейінгі бүтін
сандар қосындысы */
#include <stdio.h>
#include <conio.h>
main()
{
  int s,k;
  s=0; k=1;
  while (k<=100)
   { s+=k;
     k++;
   }
  printf ("s= %d",s);
  printf("\nPress any key\n");
  getch();
}
```

Ал егер циклдің қайталану шарты итерация соңында тексерілетін болса, онда ол соңғышартты циклге жатқызылады.



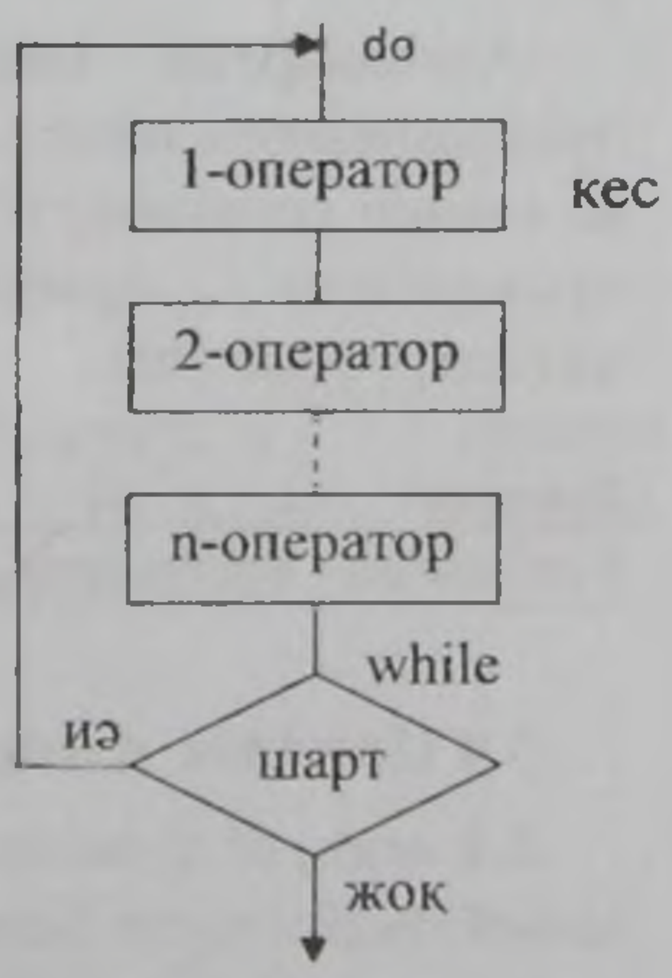
3.6-сурет. Бүтін сандарды қосу алгоритмі

3.3 Do..while операторы

Шарты соңынан тексерілетін **do ... while** циклінің орындалу схемасы 3.7-суретте көрсетілген. Осыған сәйкес оператордың жалпы жазылу түрі:

```
do
  {
    1-оператор;
    2-оператор;
     ... ... ...
    n-оператор;
  }
while (өрнек);
```

Цикл тұлғасы ретінде карапайым немесе құрама оператор колданылуы мүмкін. Жақшадағы өрнек



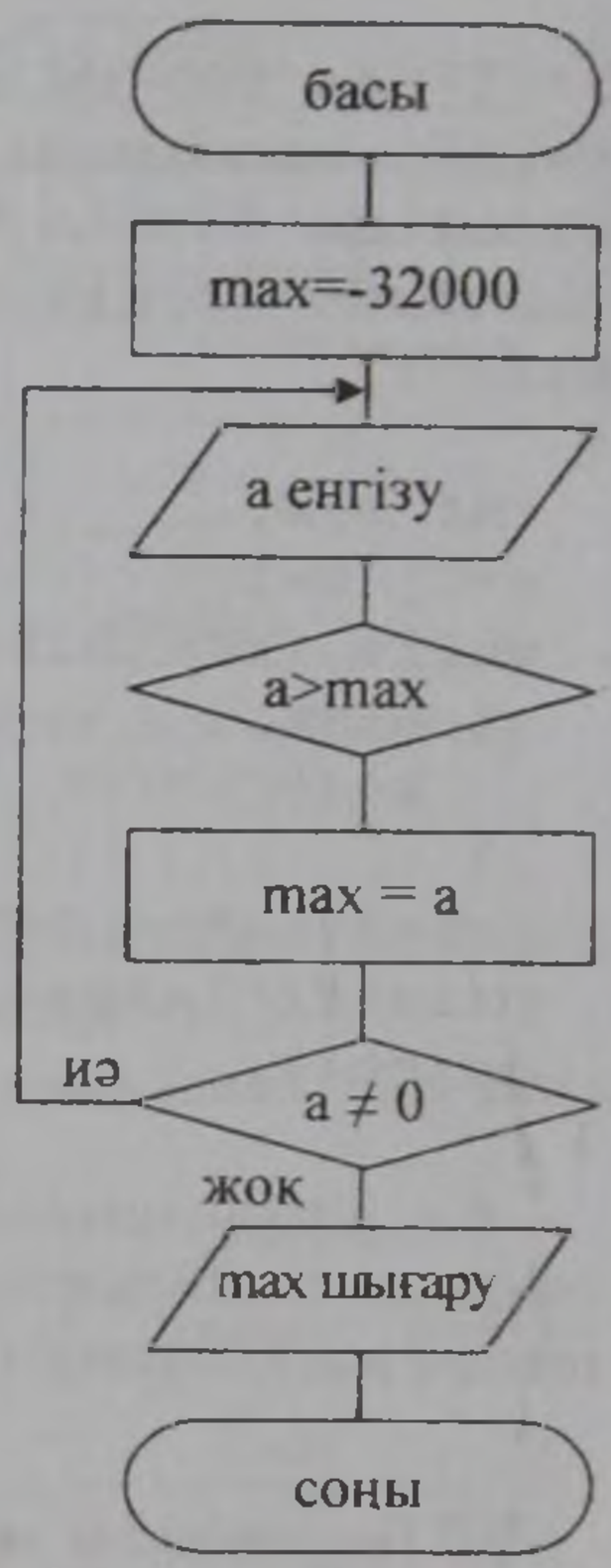
3.7-сурет. Шарты соңынан тексерілетін цикл

цикл тұлға- сынан кейін тексеріледі. Сондықтан **do while** цикл тұлғасы ең болмағанда бір рет орындалады. Цикл тұлғасынан кейін жазылған өрнек ақиқат болса (немесе жалпы
жағдайда ол 0-ге тең болмаса), цикл тұлғасы қайтадан орындалады. Ал өрнек жалған болса (немесе 0-ге тең болса), цикл аяқталады.

do ... while циклінің соңында міндетті түрде нүктелі үтір қойылатынына назар аудару керек.Цикл ішіндегі операторлардың орындалу реттілігін өзгерту мақсатында `break` және `continue` операторларын пайдалану қажет.

Енді мысалдар келтірейік. Сандардың үлкенін табу алгоритмін карастырайық (3.8 сурет).

```
// Енгізілген сандардың үлкенін табу
#include <stdio.h>
#include <conio.h>
main()
{
 int a, max;
 printf("\n Сандар максимумын табу \n");
 printf("Аяқтау үшін 0 енгізіңіз \n");
 max = -32000;
 // алдын ала максимумды ең кіші бүтін
 // санға (-32000) теңейміз
 do
  { printf("Сан енгізіңіз: ");
    scanf("%i",&a);
    if (a > max) max = a;
  }
 while (a!=0);
 printf("Сандардың максимумы: %i",m);
 getch();
}
```



3.8-сурет. Сандардың үлкенін анықтау

Алғышартты циклдер үшін, шарт жалѓан болса, программаның цикл ішіндегі бөлігі мүлде орындалмайды, ал екінші сонғышартты цикл үшін, ол кем дегенде, бір рет орындалады. Арифметикалық циклдер алғышартты циклдерге жатады.

Ескерту:

f әні жылдам өсетіндіген, оны нақты сан ретінде алған дұрыс болып табылады.

3.4 Циклдік алгоритмдер мен программалар құруға берілген есептер

3.1 есеп. n! функциясын есептеу алгоритмін құру. n! – бұл n натурал сандардың көбейтіндісі екені белгілі 1·2·3· ... · n. Мұнда 1!=1 деп аламыз. Сондықтан n! мәнін есептеуді F факториалының алдынғы мәнін келесі натурал санға көбейту операциясын қайталау арқылы жүзеге асыруға болады.

n! есептеу алгоритмі 3.9 суретінде көрсетілген.

Онын мәтіндік формасы мынадай:

1-қадам. n енгізу

2-қадам. f = 1 деп алу

3-қадам. i=2 деп меншіктеу

4-қадам. Егер i>n болса, онда 7-қадамға көшу.

5-қадам. F=F·i есептеу

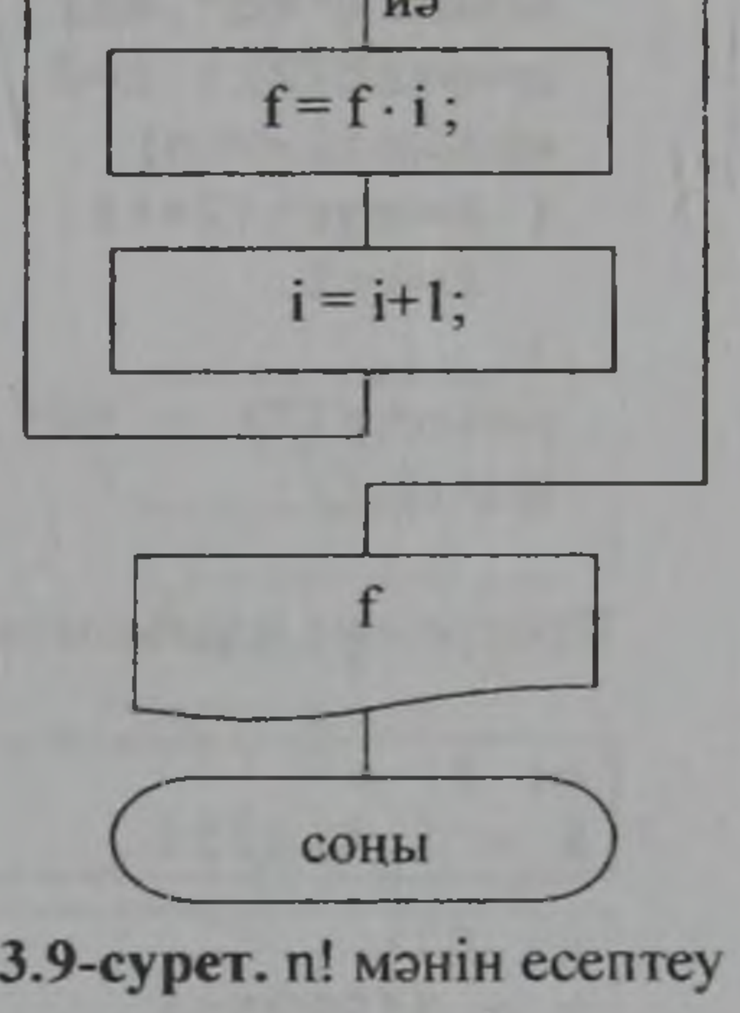
6- қадам. i=i+1 деп алып, 4-қадамға көшу

7-қадам. f мәнін шығару

8-қадам. Соңы

Осы алгоритмнің C тілінде жазылу нұсқасы:

```
#include <stdio.h>
main()
{
   int i,f,n;
   puts("n:");
   scanf("%d",&n);
   f=1; i=2;
   while(i<=n)
   { f=f*i;
     i=i+1;
   }
   printf("%d!=%d",n,f);
}
```



3.9-сурет. n! мәнін есептеу алгоритмі

Осыған карап 2^n, a^n мәндерін шығару алгоритмдерін құруға да болады, мұнда a – бөлшек, ал n – бүтін сандар.

3.2 есеп. $k=\underbrace{\sqrt{2+\sqrt{2+\sqrt{2+\cdots+\sqrt{2}}}}}_{n \text{ түбір}}$

өрнегін есептеу алгоритмін құру керек.

Егер бастапқы мәнін $k=\sqrt{2}$ деп алсақ, онда қайталанатын әрекет $\sqrt{2+k}$ өрнегін есептеу болады. Сондықтан бұл есепті шешу алгоритмі 3.10 суретте көрсетілгендей түрде бейнеленеді. Онын мәтіндік формасы мынадай болады:

1- **қадам**. n енгізу.

2- **қадам**. $k = \sqrt{2}$ есептеу.

3- **қадам**. i =2 деп алу.

4- **қадам**. Егер $i > n$ болса, 7- қадамға көшу

5- **қадам**. $k = \sqrt{2+k}$ есептеу

6- **қадам**. $i = i + 1$ деп алып, 4-қадамға көшу.

7- **қадам**. k шығару

8- **қадам**. Соңы.

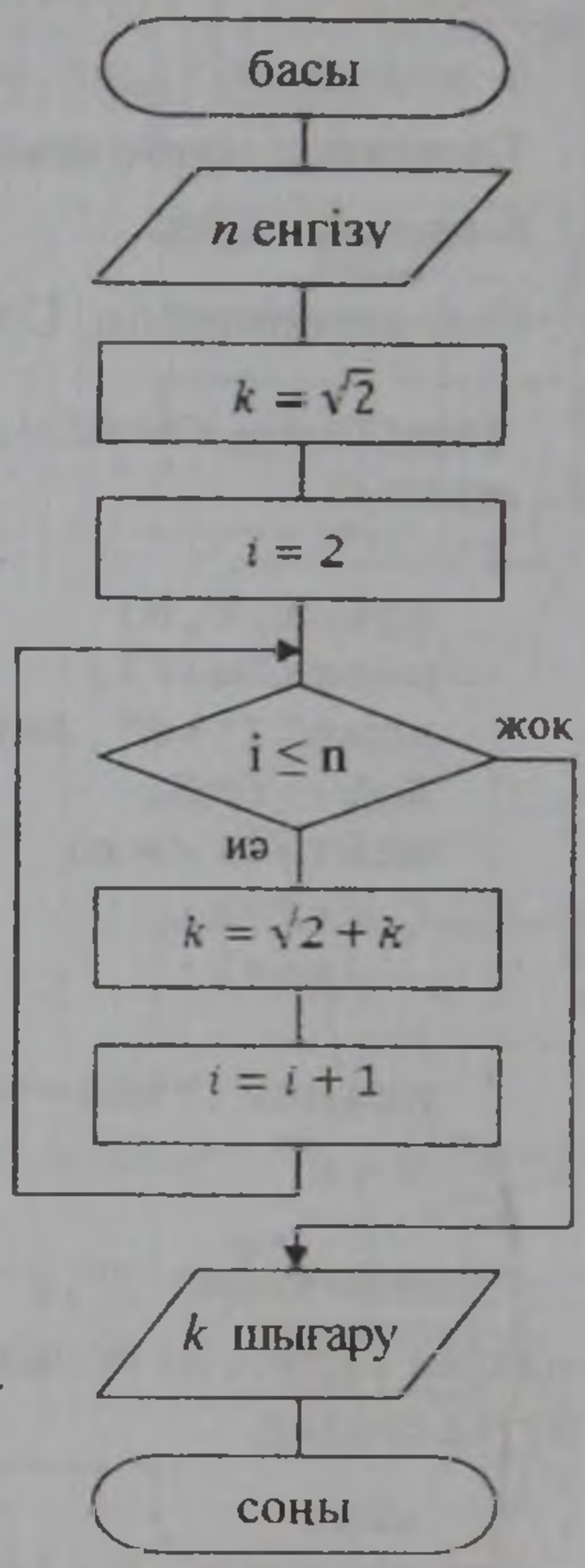
Құрылған алгоритмге сәйкес есептің С тілінде жазылған программасы:

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
main()
{ int i,n;
  float k;
  clrscr();
  printf("n: ");
  scanf("%d",&n);
  k=sqrt(2); i=2;
  while(i <= n)
  { k=sqrt(2+k);
    i=i+1;
  }
  printf("k = %f",k);
  getch();
}
```

Программа жұмысының нәтижелері:

```
n: 1
k = 1.414214
```

```
n: 5
k = 1.997591
```



3.10 сурет. k өрнегін есептеу алгоритмі

3.3 есеп. Енді Фаренгейт градустарын Цельсий градустарына ауыстыратын программа мәтінін келтірейік,

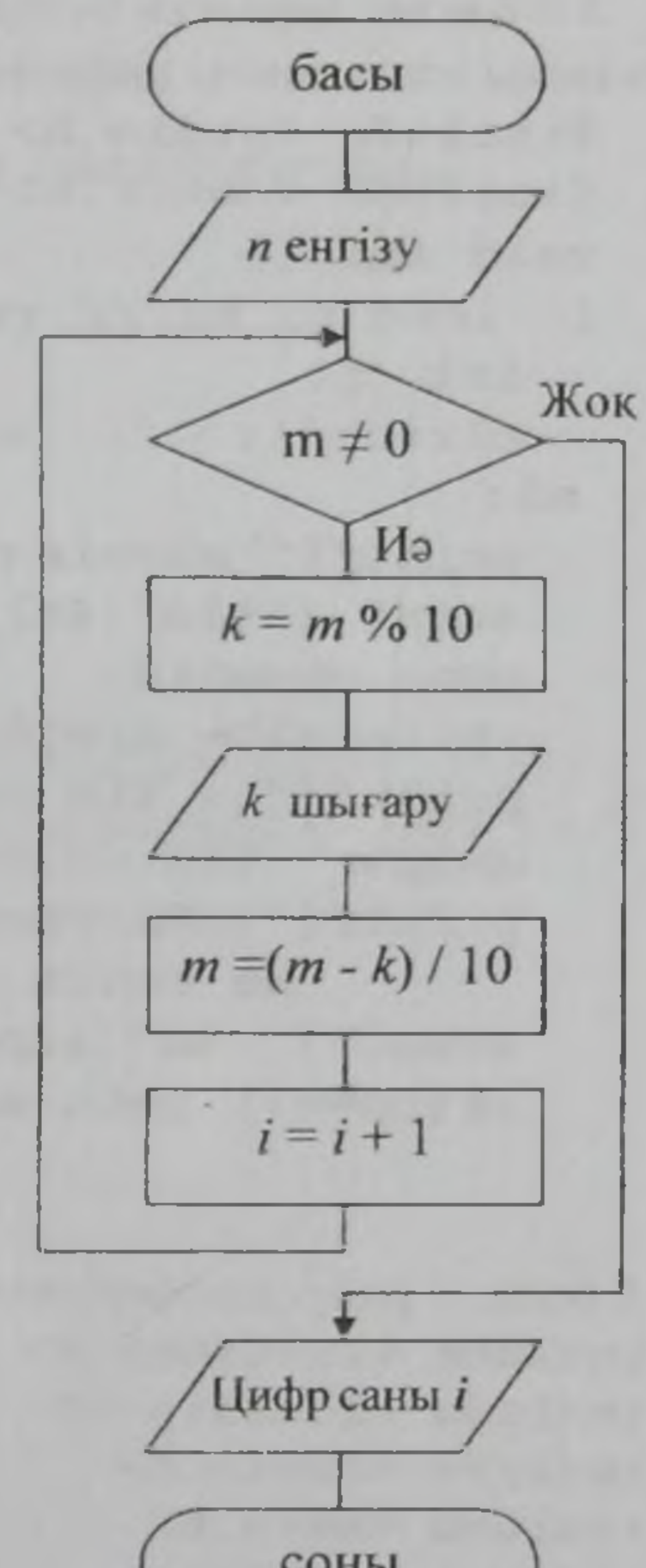
мұндағы $F = \frac{9}{5}C + 32$.

```
/* Градустарды Фаренгейт бірлігінен
   Цельсий бірлігіне алмастыру,
   f = 0, 20, ...,300 */
#include <stdio.h>
#include <conio.h>
main()
{  int t0, tn, step;
   float f,c;
   t0 = 0;       /* төменгі температура */
   tn =300;      /* жоғарғы температура */
   step  = 20; /* өзгеру қадамы */
   clrscr(); f = t0;
   printf("ГрадФ  ГрадЦ\n");
   while (f <= tn)
```

```
    { c = (5.0/9.0) * (f -32.0);
      printf("%4.0f %6.1f\n",f,c);
      f = f + step;
    }
    printf("\nАяқтау үшін ENTER басыңыз");
    getch();
  }
}
```

3.4а есеп. Берілген бүтін сан (*n*) таңбаларын кері бағытта экранға шығарып, онын неше цифрдан тұратынын аныктау қажет (3.11а сурет).

```
/ Сан цифрларын кері бағытта жазу және цифрлар санын анықтау
#include <conio.h>
#include <stdio.h>
#include <math.h>
main()
{
  int i=0,n,m,k=0;
  clrscr();
  printf("\nБерілген сан: \n");
  scanf("%d",&n);
  m=n;
  while(m!=0)
  { k=m%10;
    printf("Кері жазылуы: %i",k);
    m=(m-k)/10;
    i++;
  }
  printf("\nЦифрлары саны: %i",i);
  getch();
}
```



3.11а-сурет. Бүтін сан цифрларын өңдеу

3.4ә есеп. Берілген бүтін сан цифрларын кері бағытта жазу және цифрлар саны мен косындысын анықтаудың келесі нұсқасы (3.11ә сурет).

```
#include <iostream.h> // C++ тілінде
void main(void)
{
  int k,s,n;
  long a;
  cout << "Берілген бүтін сан: ";
  cin >> n;
  k=s=a=0;
  do
  {
    k++;s=s % 10;
    a=a*10 % 10;
```

```
    n=n/10;
  }
  while (n==0);
  cout << "цифрлар саны: " << k;
  cout << "цифрлар қосындысы: " << s;
  cout << "Сан кері бағытта: "  << a;
}
```

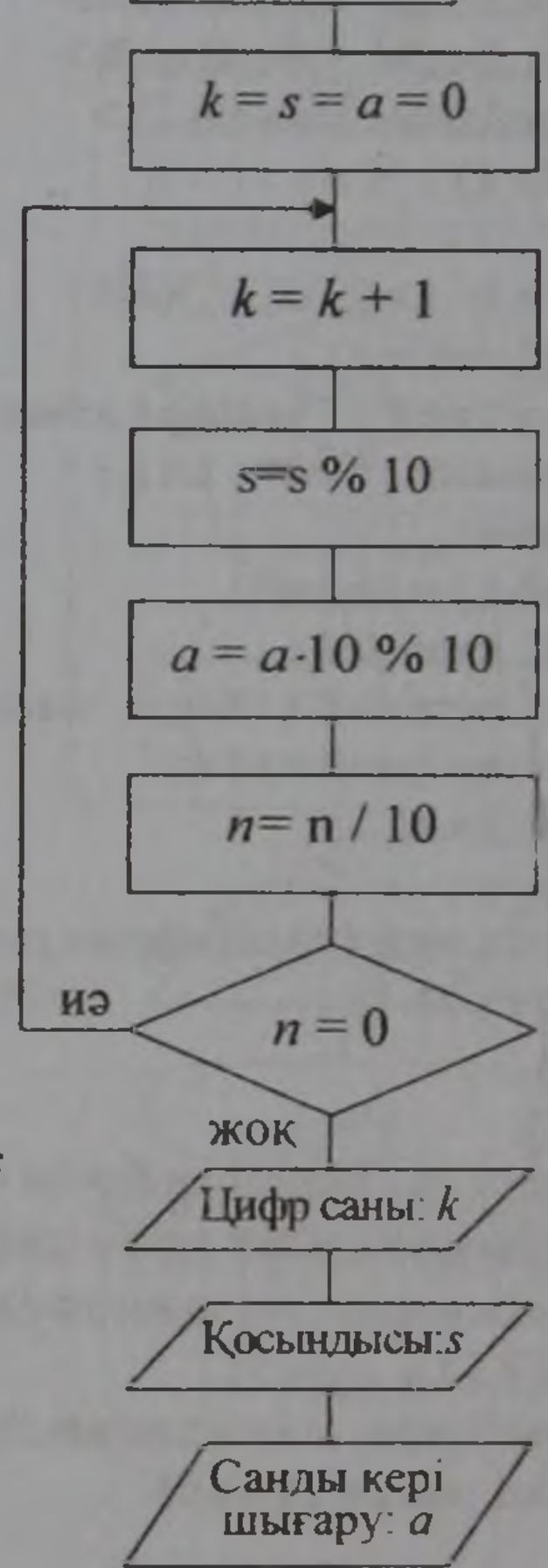
Программа жұмысы нәтижесі:

```
Берілген бүтін сан: 7614
цифрлар саны: 4
цифрлар қосындысы: 18
Сан кері бағытта: 4167
```

Осы тәрізді есепті функция арқылы шығару 4.9 есепте көрсетілген.

3.4 б есеп. Берілген бүтін санның (n) алғашқы және соңғы цифрын анықтау қажет.

```
#include <stdio.h>
#include <conio.h>
void main()
{  long n, m; // ұзын бүтін сан
   int cp;
   clrscr();  //  экранды тазалаймыз
 m1:
   printf("\nБүтін сан енгізіңіз: ");
   scanf ("%ld",&n) ;
   m=n; cp=m%10;
   while(m>9) m/=10;
   printf("\n %ld санының алғашқы
   цифры - %ld,ал соңғысы - %d",n,m,cp);
   printf("\nЖалғастырамыз ба ?
          Иә болса - 7 енгізіңіз: ");
   scanf (" %d",&cp);
   if(cp==7) goto m1;
}
```



3.11ә-сурет. Бүтін сан цифрларын өңдеу

3.5 есеп. Градуспен берілген бұрыш косинусын табу

```
#include <iostream.h>
#include <iomanip.h>
#include <conio.h>
#include <math.h>
void main(void)
{
   const eps=1e-7; // есептеу дәлдігі
   float x,s,u;
   int n;
```

```
clrscr();
cout << "Градуспен берілген бұрыш: ";
cin >> x;
x=x*M_PI/180; // бұрышты радианға ауыстыру
u=s=1; n=0;
do
{ n=n+2;
  u=-u*x*x/((n-1)*n);
  s=s+u;
}
while (fabs(u)>eps);
cout<<"cos("<<setprecision(2) <<x*180/M_PI;
  cout<<")=" << setprecision(4) << s << endl;
  getch();
}
```

Программа жұмысы нәтижесі:

```
Градуспен берілген бұрыш: 45
cos(45)=0.7071
```

3.6 есеп. π санын жуық шамамен есептеу мысалы. Ол мына формуланы пайдалану арқылы жүзеге асырылады: $\frac{\pi}{4}=1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\ldots$

```
#include <conio.h>
#include <stdio.h>
int main()
{
 double x,sum;
 int n,i;
 clrscr();
 printf("Calculation of the series sum 1-1/3+1/5-1/7+...\n");
 printf("Enter quantity of series members to be summed (>1000)->");
 scanf("%d", &n);
 sum=0;
 for(i=1; i<=n; i++)
 {
   x=(double)1/(2*i-1);
   if((i%2)==0) x=-x;
   sum+=x;
 }
 printf("Series sum is %2.15f\n",sum);
 printf(" PI  number = %2.15f\n",sum*4);
 getch();
 return 0;
}
```

Программа жұмысы нәтижесі:

```
Calculation of the series sum 1-1/3+1/5-1/7+...
Enter quantity of series members to be summed (>1000) -> 1200
Series sum is 0.785189830100284
 PI  number =3.140759320401137
```

3.7 есеп. Анықталған интегралды тіктөртбұрыштар, трапециялар және Симпсон тәсілдерімен жуық шамамен есептеу мысалы.

```
// интегралды сандық тәсілдермен есептеу
// берілген функция y=3*x*x
#include <iostream.h>
#include <iomanip.h>
#include <conio.h>
main()
{float a,b,        // кесіндінің төменгі шегі
       h,          // интервал мәні
       s1,s2,s3,   // интегралдың жуық мәні
       x,x1,x2;    // аргумент
 int n,            // интервалдар саны
     i;            // цикл санауышы
 clrscr();
 cout << "Интегралды жуықтап есептеу\n";
 cout << "кесіндінің төменгі шегі: ";
 cin >> a;
 cout << "кесіндінің жоғарғы шегі: ";
 cin >> b;
 cout << "кесіндіні неше бөлікке бөлесіз: ";
 cin >> n;
 h=(b-a)/n;
 x1=a+h/2; x2=a;
 s1=3*x1*x1;
 s2=((3*x2*x2)+(3*b*b))/2;
 for(i=1;i<n;i++)
  { x1=x1+h;x2=x2+h;
    s1=s1+(3*x1*x1); s2=s2+(3*x2*x2);
  }
 s1=h*s1; s2=h*s2; s3=(2*s1+s2)/3;
 cout << "Интегралды формулалар бойынша есептеу:\n";
 cout << "тіктөртбұрыштар\t трапеция\t Симпсон тәсілдері\n";
 cout << setprecision(4); // шығару дәлдігі 4 таңба
 cout << "s1 = "<<s1<<"\t s2 = "<<s2<<"\t s3 = "<<s3;
 getch();
}
```

Программа жұмысы нәтижесі:

```
Интегралды жуықтап есептеу
кесіндінің төменгі шегі: 0
кесіндінің жоғарғы шегі: 4.5
кесіндіні неше бөлікке бөлесіз: 25
Интегралды формулалар бойынша есептеу:
тіктөртбұрыш  трапеция      Симпсон тәсілдері
s1 = 91.0886  s2 = 91.1979  s3 = 91.125
```

Осы программаны функция арқылы құру мысалы 4.6 есепте келтірілген.

3.8 есеп. Көбейту кестесін тексеруге арналған тест программасын құру.

```
// Көбейту бойынша тест
#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
void main(void)
{  int s1,s2,otvet,kol=0,prav=0;
   char yn;
   randomize(); // кездейсоқ сандарды өзгертіп отыру
   clrscr();
   cout << "Тестті бастаймыз ба? (Y/N)"; // Иә - Yes, Жоқ - No
   cin >> yn;
   while(yn!='n' && yn!='N')
   { kol++;
     s1=random(18)+2; // 0-19 арасындағы кездейсоқ сан
     s2=random(18)+2;
     cout << "Нәтижесі қанша: " << s1 << "*" << s2 << "? ";
     cin >> otvet;
     if (otvet==s1*s2)
     { cout << "Дұрыс.\n";
       prav++;
     }
     else cout << "Дұрыс емес.\n";
     cout << "Тестті жалғастырамыз ба? (Y/N)\n";
     cin >> yn;
   }
  if (kol>0)
     { clrscr();
       cout << "Тест нәтижелері: \n";
       cout << "-------------------- \n";
       cout << kol << " сұрақ қойылды ";
       cout << "\nДұрыс жауаптар саны: " << prav;
       getch();
     }
}
```

Программа жұмысы нәтижесі:

```
Тестті бастайыңыз ба? (Y/N)y
Нәтижесі қанша: 16*4? 64
Дұрыс.
Тестті жалғастырамыз ба? (Y/N)
Y
Нәтижесі қанша: 10*12? 120
Дұрыс.
Тестті жалғастырамыз ба? (Y/N)
Y
Нәтижесі қанша: 19*19? 371
Дұрыс емес.
Тестті жалғастырамыз ба? (Y/N)
N
Тест нәтижелері:
------------------
3 сұрақ қойылды
Дұрыс жауаптар саны: 2
```

3.9 есеп. Натурал санды рим цифрларымен жазу.

```
// Натурал санды рим цифрларымен жазу
#include <iostream.h> // C++ тілі стилінде
#include <conio.h>
void main(void)
{  int i;
   clrscr();
   cout << "Бүтін сан енгізіңіз(<900): ";
   cin >> i;
   cout << i << " -> ";
while (i>=500)
{
  cout << 'D';
  i-=500;
}
while (i>=100)
{
  cout << 'C';
  i-=100;
}
while (i>=50)
{
  cout << 'L';
  i-=50;
}
while (i>=10)
{
  cout << 'X';
  i-=10;
}
while (i>=5)
{
  cout << 'V';
```

```
      i-=5;
    }
    switch(i)
    { case 0: break;
      case 1: cout << "I"; break;
      case 2: cout << "II"; break;
      case 3: cout << "III"; break;
      case 4: cout << "IV"; break;
    }
    getch();
}
```

Программа жұмысының орындалу нәтижесі:

```
Бүтін сан енгізіңіз(<900): 758
758 -> DCCLVIII
```

3.10 есеп. Сандар тізбегінің ең үлкенін (**max**), ең кішісін (**min**), арифметикалық ортасын (**average**) табу.

```
// 3.28 Сандар тізбегі үшін max, min, average табу
#include <iostream.h>
#include <conio.h>
main()
{float a,          // тізбектің кезекті саны
sum, average,  // қосынды мен арифметикалық орта
min,max;       // ең үлкен және ең кіші сан
int n,         // тізбектегі сандар санауышы
    i;         // цикл санауышы
clrscr();
cout << "Тізбекте қанша сан болады: ";
cin >> n;
cout << "Сандарды енгізіңіз:\n";
cout << "->"; cin >> a; // 1-санды енгізу
min=a; // уақытша 1-сан max болсын делік
max=a; // уақытша 1-сан min болсын делік
sum=a; // 1-санды қосындыға меншіктеу
       // қалған сандарды енгізуді бастау
for(i=1;i<n;i++)
 { cout << "->"; cin >> a; // келесі санды енгізу
   sum+=a;              // келесі санды қосындылау
   if (a<min) min=a; // келесі сан алдыңғылардың кішісі
   if (a>max) max=a; // келесі сан алдыңғылардың үлкені
 }
average=sum/n;
cout << "Енгізілген сандар:" << n << endl;
cout << "Арифметикалық ортасы:" << average << endl;
cout << "Максимал сан:" << max << endl;
cout << "Минимал сан:" << min << endl;
```

```
 getch();
}
```

Программа жұмысының орындалу нәтижесі:

```
Тізбекте қанша сан болады: 7
Сандарды енгізіңіз:
->34
->23
->9
->12
->9
->12
->15
Енгізілген сандар:7
Арифметикалық ортасы:16.285715
Максимал сан:34
Минимал сан:9
```

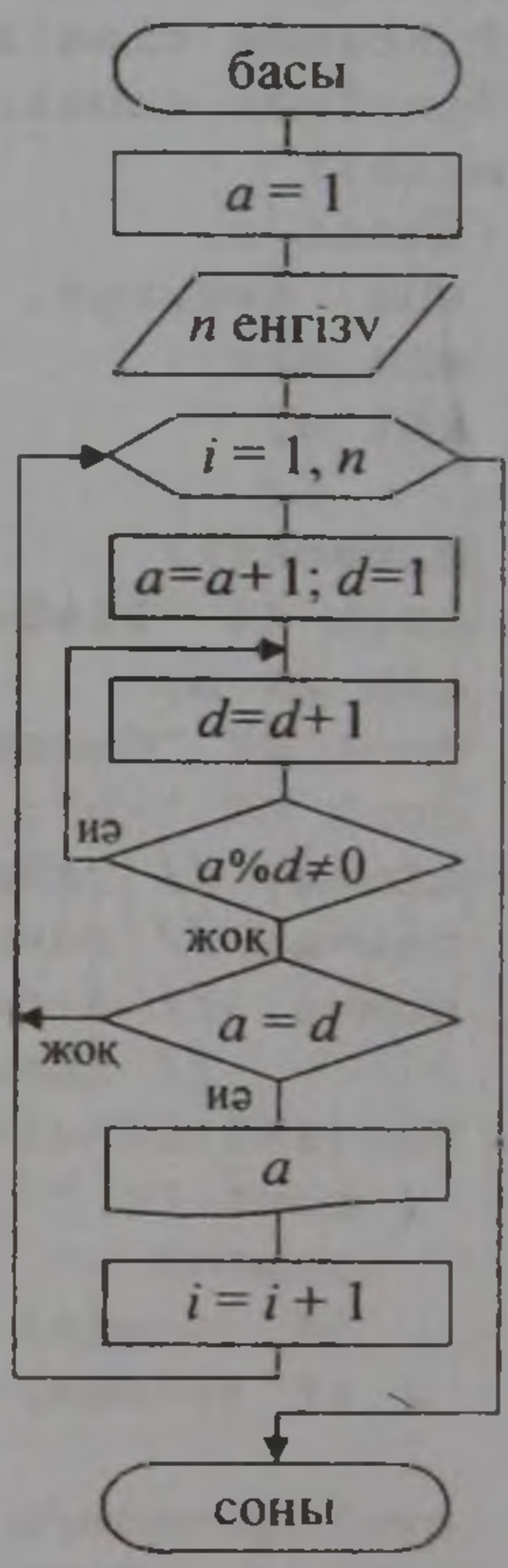
3.11 есеп. Бөлшектер дәрежелерінің қосындысын табу.

$s = 1^n + \frac{1}{2^n} + \frac{1}{3^n} + \frac{1}{4^n} + \ldots$, мұндағы *n* берілген натурал сан.

```
// (1/1)^n+(1/2)^n+ ... + (1/n)^n
//  n - натурал сан
#include <iostream.h>
#include <conio.h>
main()
{
 int i,j,n;
 float a,s,p;
 clrscr();
 cout << "Дәреже көрсеткішін (n) енгізіңіз: ";
 cin >> n;  // n енгізу
 s=0;       // бастапқы қосынды нөлге тең
 for(i=1;i<=n;i++) // сыртқы цикл басы
  { a=1./i; p=a;
    for(j=2;j<=n;j++) p*=a;
// қосылатын мүшені есептеу
    s+=p; // қосындыны жинақтау
  }
 cout << "Дәрежелер қосындысы: " << s << endl;
 getch();
}
```

Программа жұмысының орындалу нәтижесі:

```
Дәреже көрсеткішін (n) енгізіңіз: 17
Дәрежелер қосындысы: 1.000008
```



3.12-сурет. Жай сандарды анықтау

3.12 есеп. Алғашқы n жай сандарды анықтау керек (3.12 сурет).

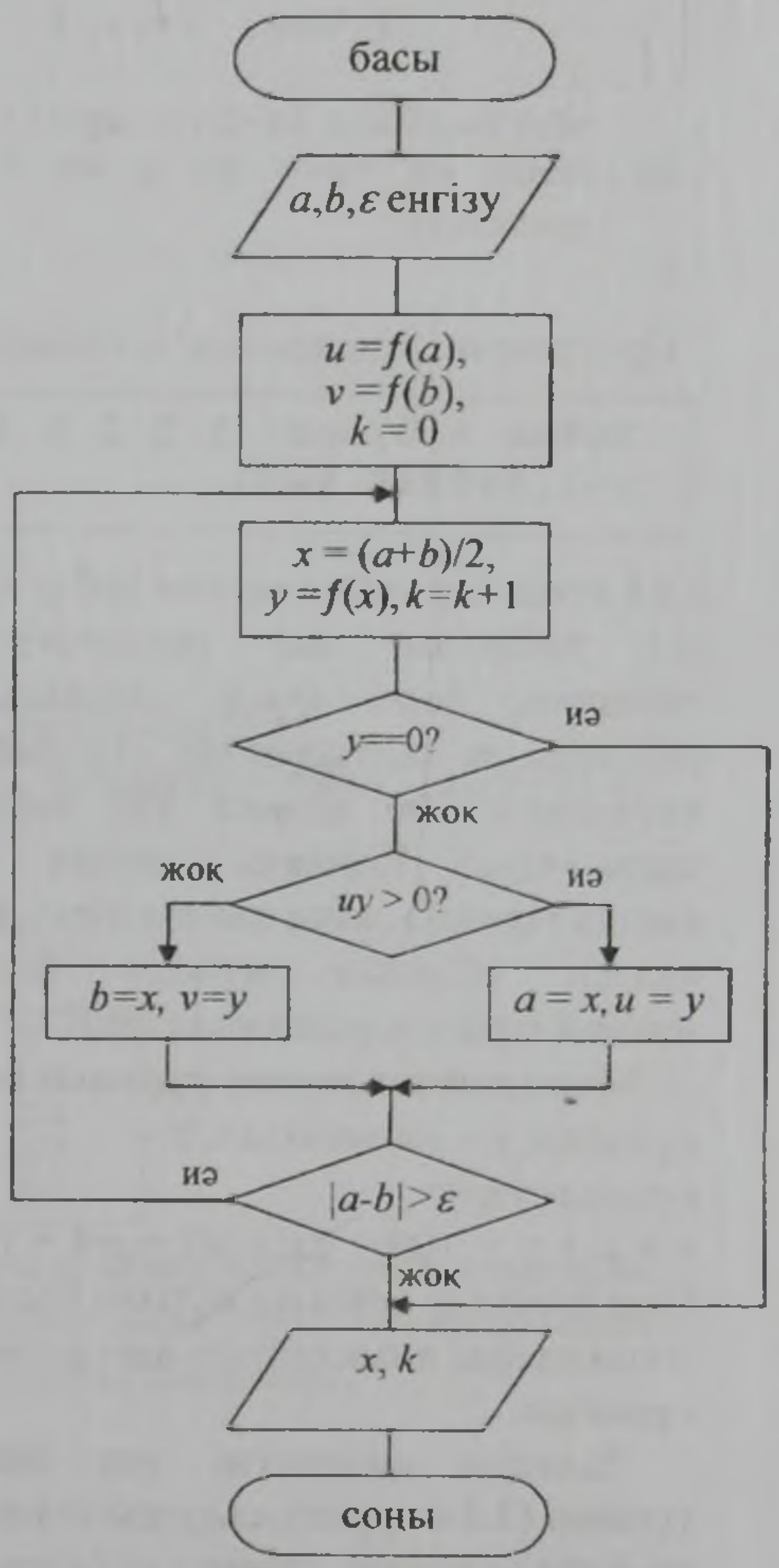
```
#include <iostream.h>
void main()
{    int a=1,n,d; cout << "\nEnter n: ";
     cin >> n;
     for(int i=0;i<n;) // сыртқы цикл
     {  a++; d=1;
        do                  // ішкі цикл
          { d++; }
        while(a%d!=0);// ішкі цикл аяғы
        if(a==d)
          { cout << a << " ";
            i++; }
     }          // сыртқы цикл аяғы
}
```

Программа жұмысының нәтижесі:

```
Enter n: 8
2 3 5 7 11 13 17 19
```

3.13 есеп. Бір белгісізі бар теңдеу түбірін кесіндіні қақ бөлу (дихотомия) тәсілімен анықтау керек (3.13 сурет). Трансценденттi $f(x) = x^2 - 5\sin x$ теңдеуінің түбірі $[a,b]$ интервалында, яғни [1,5; 2,5] кесіндісінде жатқаны белгілі, өйткені $f(a)\cdot f(b)<0$. Сол түбірді табу керек. Ол үшін түбір жатқан интервалды екіге бөлеміз де, түбірдің қай бөлікте жатқанын сол функцияның мәндерін кесінді шеттеріндегі және оның ортасындағы нүктелерде анықтаймыз. Кесіндінің екі шетінде функцияның әртүрлі таңбалар қабылдайтын,түбір жатқан кесінді бөлігін тағы екіге бөліп, ары қарай функция мәндерін салыстыра отырып, біртіндеп түбірдің нақты мәніне жақындауға тырысамыз. Осы тәсілді кесіндіні қақ бөлу немесе дихотомия тәсілі деп атайды.

Осы тәсілмен теңдеу түбірін табу алгоритмі
3.13 суретте көрсетілген. Есептің программасы
мәтінін төмендегідей түрде жазуға болады:



3.13 сурет. $f(x)=0$ теңдеуі түбірін дихотомия тәсілімен анықтау алгоритмі

```
#include <iostream.h>
#include <conio.h>
```

```
#include <math.h>
float f(float x)
{ return x*x-5*sin(x); }
void main()
{   float a,b,x,y,u,v,eps;
    int k=0; // қадамдар саны
    cout<<"\nEnter a,b,eps:";
    cin >> a >> b >> eps;
    u=f(a); v=f(b);
    do
    { x=(a+b)/2; y=f(x); k++;
      if(y==0) goto m1;
      if(u*v>0)
            { a=x; u=y; }
      else
            { b=x; v=y; }
    }
    while(fabs(a-b)> eps);
m1: cout << "x=" << x << " k=" << k;
    getch();
}
```

Программа жұмысының нәтижесі:

```
Enter a,b,eps: 1.5 2.5 1e-6
x=1.999999 k=20
```

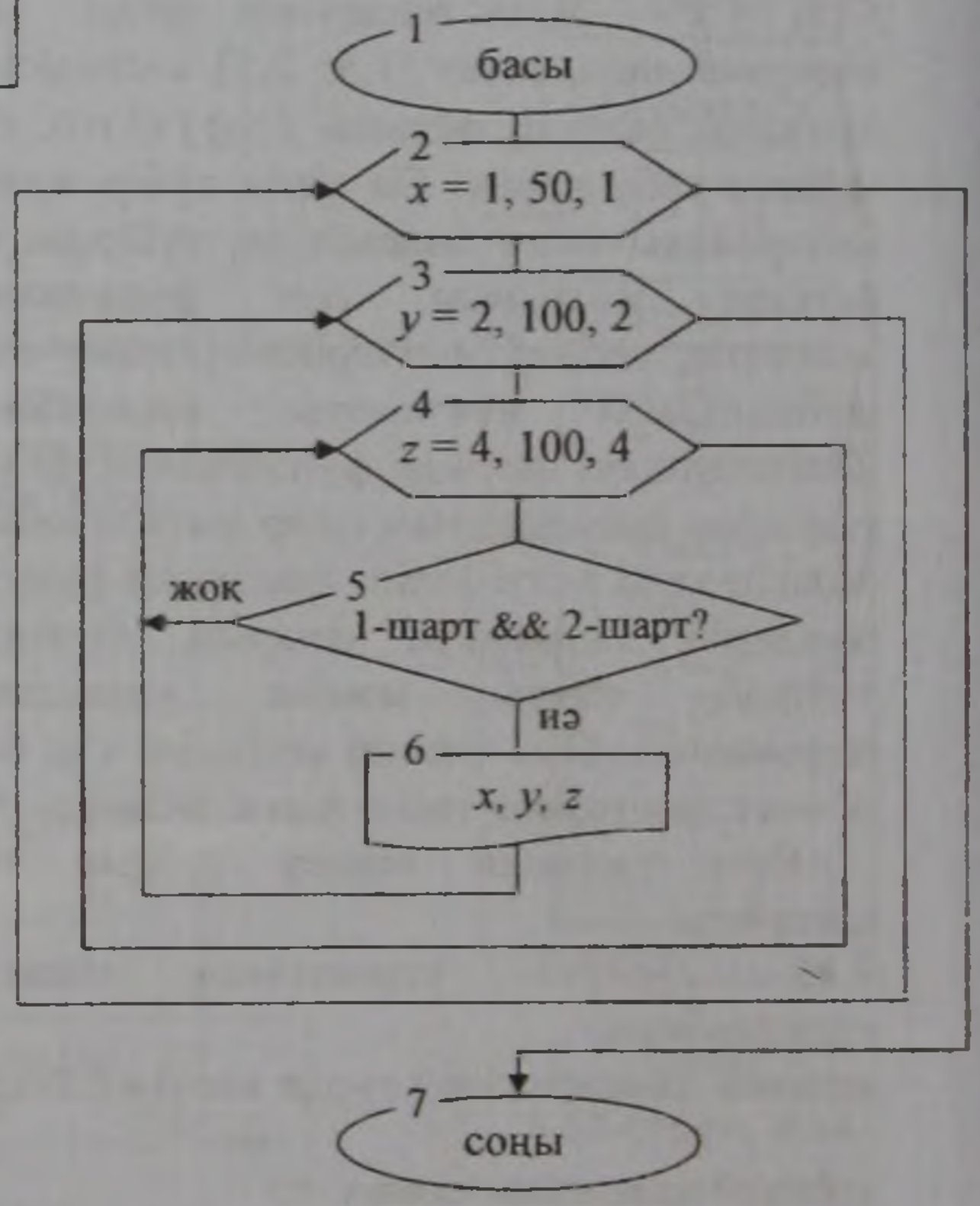
3.14 есеп. Бір тойға келген әрбір құдаға екі табактан, екі нағашыға бір табактан, төрт ауыл адамына бір табактан ет тартылыпты. Нәтижесінде конактағы 100 адамға 100 табак ет тартылады. Конакка канша құда, канша нағашы және канша ауыл адамы келген (барлық мүмкін болатын жағдайларды карастырындар)?

Мынадай теңдеулер құрамыз (*x* – құдалар, *y* – нағашылар, *z* – ауылдастар):

$x + y + z = 100$; $2x + y/2 + z/4 = 100$;

Енді циклдер аркылы мүмкін болатын шешімдерді анықтайтын алгоритм құрайык.

Есептің алгоритмі үш циклден тұрады (3.14-сурет), сырткы циклде (2-ші блок) құдалар санын (*x*) 1-ден 50-ге дейін қадамын 1 етіп өзгертеміз, өйткені 50 құданың өзі 100 табакка ие



3.14-сурет. Адамдар санын табу алгоритмі

бола алады. Келесі циклде нағашылар санын (*y*) 2-ден 100-ге дейін қадамын 2 етіп өзгертеміз (3-ші блок). Соңғы циклде ауылдастар санын (*z*) 4-тен 100-ге дейін қадамын 4 етіп аламыз (4-ші блок).

5-ші блокта адамдар санының 100 екендігі – 1-ші шарт:

$$x + y + z = 100$$

және (&&) табақтар саны да 100 болатындығы – 2-ші шарт:

$$2x + y/2+z/4 = 100$$

қатар тексеріледі де (5-ші блок), шарт орындалатын жағдайларда ғана құдалар, нағашылар және ауылдастар саны экранға шығарылады (6-шы блок). Осы алгоритмге сәйкес программа мәтіні мынадай болады:

```
/*   100 қонақ, 100 табақ
     1 құдаға 2 табақ
     2 нағашыға 1 табақ
     4 ауылдасқа 1 табақ  */
#include <stdio.h>
#include <conio.h>
main()
{int x,y,z; clrscr();
 for(x=1; x<50; x++)
   for(y=2; y<100; y+=2)
      for(z=4; z<100; z+=4)
       { if ((2*x+y/2+z/4==100) && (x+y+z==100))
         printf("құда = %d   нағашы = %d   ауылдас = %d\n",x,y,z);
       }
 getch();
}
```

Программа жұмысының орындалу нәтижесі:

```
құда = 34   нағашы = 62   ауылдас = 4
құда = 36   нағашы = 48   ауылдас = 16
құда = 38   нағашы = 34   ауылдас = 28
құда = 40   нағашы = 20   ауылдас = 40
құда = 42   нағашы = 6    ауылдас = 52
```

3.15 есеп. «Бақытты» билет нөмірлерін экранға шығару керек. Алты орынды билет нөмірлерінің алғашқы үш цифрының қосындысы соңғы үш цифрының қосындысына тең болса, ол «бақытты» билет болып саналады.

```
// Бақытты билеттер нөмірлерін шығару
#include <stdio.h>
#include <conio.h>
main()
{
  int a,b,c,x,y,z;
```

```
   clrscr();
   for (a=0; a<9; a++)
    for (b=0; b<9; b++)
     for (c=0; c<9; c++)
      for (x=0; x<9; x++)
       for (y=0; y<9; y++)
        for (z=0; z<9; z++)
          if (a+b+c==x+y+z)
            printf("%i%i%i%i%i%i\n",a,b,c,x,y,z);
}
```

Бақылау сұрақтары

1. Параметрлі циклдің орындалу схемасы мен жазылуы.
2. Параметрлі цикл бір де бір рет орындалмауы мүмкін бе?
3. for операторының параметрі қандай типтерде бола алады?
4. for операторы параметрінің алғашқы мәні оның соңғы мәнінен кіші бола ма?
5. Параметрлі цикл операторының неше рет қайталанатынын алдын ала білуге бола ма, болса – ол қалай анықталады?
6. for цикл операторындағы жақша ішіндегі бір немесе бірнеше өрнектерді жазбауға бола ма?
7. for цикл операторы нүктелі үтірмен аяқтала ала ма?
8. for цикл операторында қай кезде құрама операторлар қолданылады?
9. for цикл операторы қай кезде шексіз циклге айналады?
10. for цикл операторында бірнеше құрама өрнектерді үтір аркылы қалай жазуға болады?
11. Шарты алдын ала тексерілетін цикл операторының орындалу схемасы мен оның жазылуы.
12. while цикл операторының ішкі тұлғасы бір де бір рет орындалмауы мүмкін бе?
13. while цикл операторының шарты қатынас таңбаларысыз жазыла ма?
14. while цикл операторы қай кезде шексіз циклге айналады?
15. while цикл операторының тұлғасындаонын шартына әсер ететін өрнектер жазыла ма?
16. Шарты соңынан тексерілетін do ... while циклінің орындалу схемасы мен жазылуы.
17. do ... while цикл операторының ішкі тұлғасы бір де бір рет орындалмауы мүмкін бе?
18. do ... while цикл операторының шартында қатынас таңбасы болмаса, оның ақиқат немесе жалған екенін қалай аныктауға болады?
19. do ... while цикл операторының ішкі тұлғасында шартсыз көшу операторын қолдануға бола ма?

20.do … while цикл операторы қай кезде шексіз орындалады?

Тапсырмалар

1. 1-ден *N*-ге дейінгі сандардың қосындысын және арифметикалық ортасын есептейтін программа құрындар. *N*-нің мәні пернетақтадан енгізіледі.
2. 1-ден *N*-ге дейінгі сандардың көбейтіндісін және геометриялық ортасын есептейтін программа құрындар. *N*-нің мәні пернетақтадан енгізіледі.
3. Пернетақтадан *N* сан енгізіледі. Енгізілген сандардың ішіндегі теріс, оң сандардың және нөлдердің санын анықтайтын программа құрындар.
4. Ұзындықтың 1-ден 20 дюймге дейінгі мәндерін сантиметрге (1 дюйм = 2,54 см) айналдыратын және оны экранға шығаратын программа құрындар.
5. Банктегі жылдық өсімі 9 пайыздық (проценттік) салымға *S* теңге салынды. *N* жылдан кейін салынған ақша неше теңгеге жетеді?
6. Пернетақтадан 10 бүтін сан енгізіп, солардың квадраты мен кубын анықтаңыз.
7. 20-дан 50-ге дейінгі натурал сандар берілген. Олардың ішіндегі 3-ке бөлінетін, бірақ 5-ке бөлінбейтін сандарды анықтандар.
8. 35-тен 87-ге дейінгі натурал сандар берілген. Олардың ішінде 7-ге бөлгенде, 1-ге, 2-ге немесе 5-ке тең калдық қалатын сандарды табындар.
9. 1-ден 50-ге дейінгі натурал сандар берілген. Олардың ішіндегі 5-ке немесе 7-ге бөлінетін сандардың қосындысын табындар.
10. Пернетақтадан 10 сан енгізіндер. Егер олардың ішінде 15-тен асқаны бар болса, онда оларды 15-пен алмастырындар. Сандарды экранға шығарындар.
11. Пернетактадан он теріс және оң сан енгізіндер. Барлық теріс сандарды олардың модульдерімен алмастырып сандарды экранға басып шығарындар.
12. Екі орынды сандардың ішіндегі 4-ке бөлінетінін, бірақ 6-ға бөлінбейтінін табындар.
13. 13-ке қалдықсыз бөлінетін екі орынды тақ сандардың көбейтіндісін табындар.
14. 100-ден 200-ге дейінгі сандардың ішіндегі 17-ге қалдықсыз бөлінетін сандардың қосындысын табындар.
15. Пернетактадан 10 сан енгізіндер. Егер сан 100-ден кем болса, онда осы санды және оның квадратынтабындар.
16. 1-ден бастап өздерің енгізген бүтін *n* санына дейінгі сандардың квадраттарының қосындысын есептейтін программа құрындар.
17. 200-ге дейінгі 5-ке бөлгенде 4 қалдық қалатын сандар нешеу екенін табу керек.
18. 200-ге дейінгі бүтін сандардын 25-ке қалдықсыз бөлінетін сандары нешеу екенін анықтандар.
19. 20-дан үлкен және 100-ден кіші 3-ке қалдықсыз бөлінетін оң сандардың қосындысын табу керек.
20. Берілген *a* нақты және *b* бүтін сандарының мәндері бойынша a^b өрнегінің мәнін pow() функциясын пайдаланбай табатын программа жазындар.

21. Төмендегі есептерде бір өлшем бірлігін екінші өлшем бірлігіне түрлендіру қажет (цикл санауышы мәні 1-ден 20-ға дейін өзгереді):

а) футпен берілген ұзындық өлшемін метрге (1 фут = 0,3048 м);

ә) драхмды граммға (1 драхм = 3,7325 г);

б) унцияны граммға (1 унция = 29,86 г);

в) фунтты килограммға (1 фунт = 0,40951 кг);

г) аршынды метрге (1 аршын = 0,7112 м);

д) қадақты граммға (1 қадақ = 400 г);

е) қарысты сантиметрге (1 қарыс = 18 см);

ж) дюймді миллиметрге (1 дюйм = 25,3995 мм).

22. Төмендегі функциялар мәндерін x айнымалысы көрсетілген кесіндіде dx қадамымен өзгерген кезде анықтандар.

1) $y = \ln(1+\sqrt{x})$, $x\in[2;3]$, dx=0,1;

2) $y = 2^x \sin 2x$, $x\in[0;5]$, dx=0,5;

3) $y = 3\cdot\ln x - \frac{\sqrt[5]{x}}{5}$, $x\in[2;5]$, dx=0,25;

4) $f(x)=\begin{cases}5x^2+6, \text{егер} & x<5;\\ x^3+7, \text{егер} & x\ge 5;\end{cases}$ $x\in[0;10]$, dx=0,75;

5) $f(x)=\begin{cases}\sqrt{3x^3+4}+\cos, \text{егер} & 0<x<1;\\ 5-\sin^2 x^9, \text{егер} & x\ge 1 \text{ немесе } x\le 0;\end{cases}$ $x\in[-5;15]$, dx=0,75;

6) $f(x)=\begin{cases}\sqrt[3]{x^2+3}+6x^2, \text{егер} & x\le 5;\\ x^5+35, \text{егер} & x>5;\end{cases}$ $x\in[-2;7]$, dx=0,5;

7) $f(x)=\begin{cases}x^3\cos x, \text{егер} & 0<x<2\\ 3x^4+7, \text{ қалған жағдайларда} & \\ \sqrt{5x^2+1.6} \text{ егер} & 2\le x\le 9;\end{cases}$ $x\in[-2;15]$, dx=0,75;

8) $f(x)=\begin{cases}xtgx-\sin x, \text{егер} & x<1.5;\\ x^3+\sin x, \text{егер} & 1.5\le x<2.5;\\ 3x^3+5, \text{егер} & x\ge 2.5;\end{cases}$ $x\in[-5;15]$, dx=0,75

9) $f(x)=\begin{cases}xtgx^2-\sin x, & \text{егер} \quad x<1.5;\\ x^3+\sin x, & \text{егер} \quad 1.5\le x<2.5;\\ 3x^3+5, & \text{егер} \quad x\ge 2.5;\end{cases}$ $x\in[-2;5]$, dx=0,75

23. 9 санының көбейту кестесін шығаратын программа құрындар:

```
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27   т.с.с.
```

24. Енгізілген санның жай сан екенін тексеретін программа құру керек.

25. Келесідей символдар тізбегін шығаратын программа құру қажет:

```
25.1. *          25.2. *******   25.3. *******   25.4. *******
      **                 ******          *****          ******
      ***                *****            ***           *****
      ****               ****              *            ****
      *****              ***              ***            ***
      ******             **              *****           **
      *******            *              *******          *
```

26. Ең соңында 9999 саны тұрған бүтін сандар тізбегі берілген. Солардың:
    - жалпы санын;
    - арифметикалық ортасын;
    - максимал мәнін;
    - теріс және оң элементтері санын;
    - минимал элементінің нөмірін;
    - тізбек ішіндегі жұп сандар санын;
    - жұп сандары ішіндегі ең кіші элементін анықтау керек.

27. x нақты саны және n натурал саны берілген. Төмендегі өрнектерді есептеу керек:

    а) $\sin x + \sin^2 x + \sin^3 x + \ldots + \sin^n x$;

    ә) $\sin x + \sin x^2 + \sin x^3 + \ldots + \sin x^n$;

    б) $\sin x + \sin\sin x + \sin\sin\ldots\sin\sin x$, соңғы өрнек n рет қайталанады.

    в) $1+\frac{1}{x}+\frac{1}{2x}+\frac{1}{3x}+\ldots+\frac{1}{nx}$;

    г) $\frac{x}{2}+\frac{2x}{3}+\frac{3x}{4}+\frac{4x}{5}+\ldots+\frac{(n-1)x}{n}$;

28. a, h нақты сандары және n натурал саны берілген. Төмендегі өрнекті есептеңдер:

$$f(a)+2f(a+h)+2f(a+2h)+\ldots+2f(a+(n-1)h)+f(a+nh);$$

мұндағы $f(x)=(x^2+1)\cos^2 x$.

29. n натурал саны берілген.

    а) n санында қанша цифр бар?

    ә) Осы сан цифрларының қосындысы неге тең?

    б) Осы санның ең алғашқы тұрған цифрын анықтау керек.

    в) n саны цифрларының ауыспалы таңбалы қосындысын табу керек (n санының ондық жүйеде жазылуы $a_k a_{k-1}..a_0$ болсын, сонда: $a_k - a_{k-1} + \ldots + (-1)^k a_0$ табу керек).

30. *n, m* натурал сандары берілген. *n* санының соңғы *m* цифрлары косындысын табу керек.

31. Натурал *n* саны берілген.

а) n^2 санының жазылуында 3 санының бар екенін анықтау керек.

ә) *n* санының цифрлар қатарын керісінше жазып шығу керек.

б) *n* санының алғашқы және соңғы цифрларының орнын ауыстыру керек.

в) *n* санының алғашқы цифрының алдына және соңғы цифрынан кейін 1-ді қосып жазу қажет.

32. Теріс емес бүтін сандардың ең үлкен ортақ бөлгішін табу керек, ол үшін, Евклид алгоритмінің шарттары қолданылады. Ол шарт бойынша, *n* және *m* он бүтін сандар және $m \geq n$. Егер $n=0$ болса, онда ЕҮОБ $(n, m)=m$, ал $n\neq 0$ болса, онда *m, n* және *r* сандары арқылы, мұндағы $r - m$-ді *n*-ға бөлгендегі қалдық, ЕҮОБ (n, m) табу кезінде мынадай теңдік қолданылады: ЕҮОБ(n, m) = ЕҮОБ(n, r). Мысалы: ЕҮОБ(6, 15) = ЕҮОБ(6, 3) = ЕҮОБ(3, 0) = 3.

33. *n* натурал саны берілген. Есептеңіздер:

а) $\sum_{k=1}^{n} \frac{1}{k}$

ә) $\sum_{k=1}^{n} \frac{1}{k^5}$

б) $\sum_{k=1}^{n} \frac{1}{(2k+1)^2}$

в) $\sum_{k=1}^{n} \frac{(-1)^k}{(2k+1)k}$

г) $\sum_{k=1}^{n} \frac{(-1)^{k+1}}{k(k+1)}$

д) $\sum_{k=0}^{n} \frac{(-1)^k (k+1)}{k!}$

е) $\sum_{k=1}^{n} \frac{k!}{\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{k+1}}$

34. *x* нақты саны және *n* натурал саны берілген. Есептеу керек:

а) $\sum_{i=1}^{n} \frac{x^i}{i!}$;

ә) $\sum_{i=1}^{n} \left(\frac{1}{i!}+\sqrt{|x|}\right)$;

б) $\sum_{i=1}^{n} \frac{x+\cos(ix)}{2^i}$;

в) $\prod_{k=1}^{n} \left(1+\frac{\sin(kx)}{k!}\right)$;

г) $\prod_{k=1}^{n} \left(\frac{k}{k+1}-\cos^k|x|\right)$;

д) $\prod_{k=1}^{n} \frac{(1-x)^{k+1}+1}{((k-1)!+1)^2}$.

35. *n* натурал саны берілген. *n* мүшелі тізбектің көбейтіндісін табыңыздар.

а) $\frac{1}{2}\cdot\frac{3}{4}\cdot\frac{5}{6}\cdot\ldots$;

ә) $\frac{1}{1}\cdot\frac{3}{2}\cdot\frac{5}{3}\cdot\ldots$

36. *n* натурал саны берілген. Келесі өрнекті есептеңіз.

$$\sqrt{3+\sqrt{6+\cdots+\sqrt{3(n-1)+\sqrt{3n}}}}$$

37. Кассада тек үш және бес теңгелік ақша (купюралар) ғана қалған (бұл 1998 жылдары болған). 7 теңгеден артық кез келген соманы осындай екі ғана ақша түрімен "төлейтін" программа құру керек.

38. Бүтін *M* саны енгізілген (M<25), цифрларының қосындысы *M*–ге тең болатын барлық үш таңбалы ондық сандарды табу керек. Сонымен қатар осындай сандардың қосындысын есептеу немесе ондай сандар болмаса экранға мәлімет шығару қажет.

39. Программаға өзіңіздің туған айыңыз бен күніңізді енгізіңіз. Келе жатқан жылдардың қайсысы сіз үшін бақытты жыл болатынын анықтаңыз.

«Бақытты жыл» деп жылдың цифрлар қосындысын 10-ға бөлгендегі қалдығы мен туған ай мен күніңіздің цифрлары қосындысының 10-ға бөлгендегі қалдығы тең болатын жылды айтады.

№ 3-4 зертханалық жұмыстар
Циклдік алгоритмдерді программалау

Тақырыбы: Қайталау (цикл) алгоритмдерін программалау.

Тапсырма: Берілген есептерді шығару алгоритмдерінің блок-схемасы мен програмаларын құрыңдар.

Нәтижесі: Орындалған жұмыс бойынша есеп беру алгоритм блок-схемасы, программа мәтіні және есептеу нәтижелерінен тұрады (бастапқы мәндер: теріс, он сандар, нөл, бір мәндері болуы тиіс).

Зертханалық жұмысты қорғау сұрақтары

1. Арифметикалық цикл алгоритмі, for операторы, параметрлі цикл.
2. Қадамдық цикл алгоритмі, while және do .. while операторлары.
3. While, do .. while және for операторларын салыстыру.

Тапсырмалар нұсқалары

1 нұсқа

1. *n* натурал саны берілген. Дәрежелеу (pow) функциясын колданбай, 2^n есептеңіз.
2. Нақты *a* саны берілген. Төмендегі қатар сандарының ішінен *a*-дан үлкен бірінші санды табыңыз.

$$1,1+\frac{1}{2},1+\frac{1}{2}+\frac{1}{3},\ldots$$

2 нұсқа

3. *n* натурал саны берілген. *n*! есептеңіз.
4. Нақты *a* саны берілген. $1+\frac{1}{2}+\cdots+\frac{1}{n}>a$ тенсіздігі орындалатын ең кіші *n*-ді табыңыз.

3 нұсқа

5. *n* натурал саны берілген. Төмендегі өрнекті есептеңіз.

$$\left(1+\frac{1}{1^2}\right)\left(1+\frac{1}{2^2}\right)\ldots\left(1+\frac{1}{n^2}\right)$$

6. Натурал *n*, нақты *x* саны берілген. Төмендегі өрнекті есептеңіз:

$$sinx + sin^2x + \cdots + sin^nx$$

4 нұсқа

7. *n* натурал саны берілген. Төмендегі қосындыны есептеңіз:

$$\frac{1}{sin1} + \frac{1}{sin1+sin2} + \cdots + \frac{1}{sin1+\cdots+sin\,n}$$

8. Натурал *n*, нақты *x* саны берілген. Төмендегі қосындыны есептеңіз:
$sinx + sinx^2 + \cdots + sinx^n$

5 нұсқа

9. *n* натурал саны берілген. Төмендегі өрнекті есептеңіз:

$$\cfrac{1}{1+\cfrac{1}{3+\cfrac{1}{5+\cfrac{1}{\cdots}}}}$$

$$\cdots + \frac{1}{101+\frac{1}{103}}$$

10. Натурал *n*, нақты *x* саны берілген. $\sin x + \sin\sin x + \ldots + \sin\sin\ldots\sin x$ өрнегін есептеңіз, соңғы көбейтінді *n* рет қайталанады.

6 нұсқа

11. *n* натурал саны берілген. Төмендегі өрнекті есептеңіз.

$$\frac{\cos 1}{\sin 1}\cdot\frac{\cos 1+\cos 2}{\sin 1+\sin 2}\cdot\ldots\cdot\frac{\cos 1+\ldots+\cos n}{\sin 1+\ldots+\sin n}$$

12. Мына өрнекті есептеңіз. $\sum_{i=1}^{100}\frac{1}{i^2}$

7 нұсқа

13. Бүтін *n*, *k* сандары берілген $(n \geq k \geq 0)$. Есептеңіз: $\frac{n(n-1)\ldots(n-k+1)}{k!}$

14. Төмендегі өрнекті есептеңіз. $\sum_{i=1}^{50}\frac{1}{i^3}$

8 нұсқа

15. *n* натурал және *a* нақты саны берілген. Дәрежелеу функциясын пайдаланбай, a^n мәнін есептеңіз.

16. Төмендегі өрнекті есептеңіз. $\sum_{i=1}^{10}\frac{1}{i!}$

9 нұсқа

17. *n* натурал және *a* нақты саны берілген. $a(a+1)\dots(a+n-1)$ есептеңіз.

18. Мына өрнекті есептеңіз. $\sum_{i=1}^{128}\frac{1}{(2i)^2}$

10 нұсқа

19. *n* натурал және *a* нақты саны берілген. Төмендегі өрнекті есептеңіз.

$$\frac{1}{a}+\frac{1}{a(a+1)}+\dots+\frac{1}{a(a+1)\dots(a+n)}$$

20. Төмендегі өрнекті есептеңіз, n натурал сан. $\sum_{k=1}^{n}\frac{1}{k}$

11 нұсқа

21. *n* натурал және *a* нақты саны берілген. Төмендегі өрнекті есептеңіз.

$$\frac{1}{a}+\frac{1}{a^2}+\frac{1}{a^4}+\dots+\frac{1}{a^{2^n}}$$

22. Мына өрнекті есептеңіз. *n* натурал сан. $\sum_{k=1}^{n}\frac{1}{k^5}$

12 нұсқа

23. *n* натурал және *a* нақты саны берілген.

$a(a-n)(a-2n)\dots(a+n^2)$ есептеңіз.

24. Төмендегі өрнекті есептеңіз. n натурал сан. $\sum_{k=1}^{n}\frac{1}{(2k+1)^2}$

13 нұсқа

25. $(1+\sin 0.1)(1+\sin 0.2)\dots(1+\sin 10)$ өрнегін есептеңіз.

26. Төмендегі өрнекті есептеңіз. n натурал сан. $\sum_{k=1}^{n}\frac{(-1)^k}{(2k+1)k}$

14 нұсқа

27. Нақты *x* саны берілген. Төмендегі өрнекті есептеңіз:

$$x-\frac{x^3}{3!}+\frac{x^5}{5!}-\frac{x^7}{7!}+\frac{x^9}{9!}-\frac{x^{11}}{11!}+\frac{x^{13}}{13!}$$

28. Төмендегі өрнекті есептеңіз. n натурал сан. $\sum_{k=1}^{n}\frac{(-1)^{k+1}}{k(k+1)}$

15 нұсқа

27. Нақты *x*, *a*, натурал *n* саны берілген. Есептеу керек:

$((\dots((x+a)^2+a)^2+\dots+a)^2+a)^2+a$; *n* жақшалы;

28. Нақты *x* саны берілген. Төмендегі өрнекті есептеңіз.

$$\frac{(x-2)(x-4)(x-8)\dots(x-64)}{(x-1)(x-3)(x-7)\dots(x-63)}$$

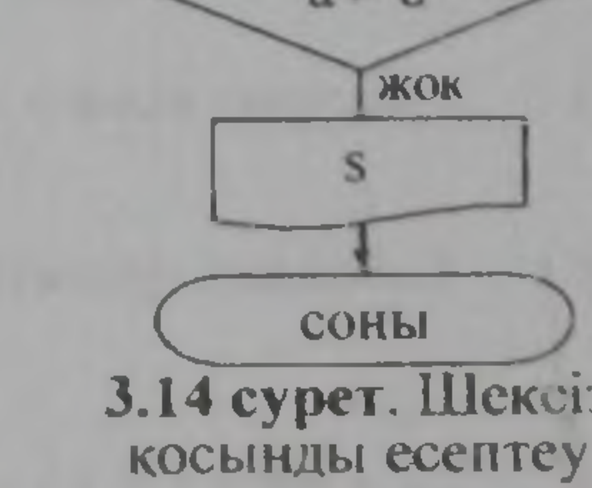
3.5 Шексіз қосындыларды есептеу

3.14 есеп. Біртіндеп кішірейетін сандар қосындысын

$$s = \sum_{i=1}^{\infty} \frac{1}{i^2} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{5^2} + \ldots + \frac{1}{n^2} + \ldots$$

берілген дәлдікпен $\varepsilon=10^{-5}$ анықтау керек. Бұл дәлдікке қол жеткізу үшін берілген қатардың бірсыпыра мүшелерін есептеп қосындылау керек. Келесі мүше берілген дәлдіктен (Е) кем болмағанда, қосындылау тоқтатылады.

Мүшелер саны көбейген сайын қосындыны есептеу дәлдігі артып отырады. Сондықтан қатардың келесі мүшесі берілген ε мәнінен кіші болған сәтте есептеу тоқтатылады. Осындай тәртіпті сақтай отырып, жоғарыдағы қосындыны анықтау алгоритмі 3.15 суретте келтірілген. Алгоритмге сәйкес программа мәтіні төмендегідей түрде болады.



3.14 сурет. Шексіз қосынды есептеу

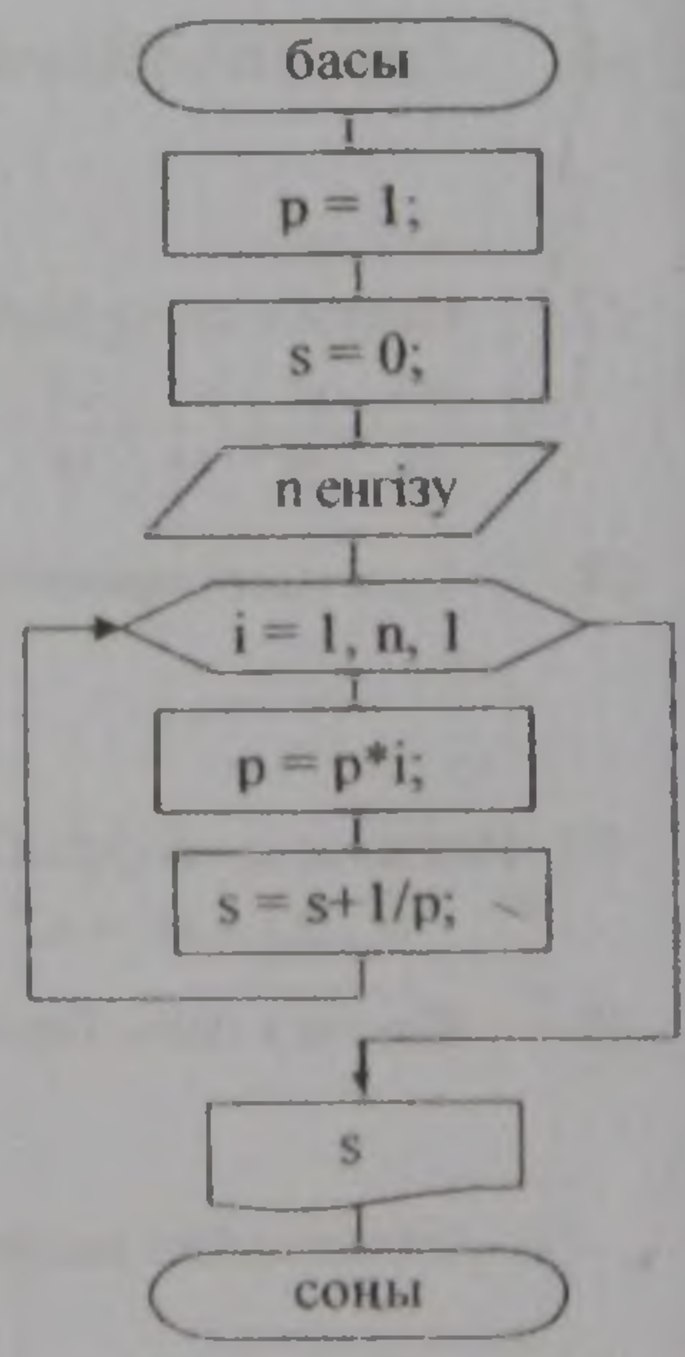
```
#include <stdio.h>
#include <conio.h>
#define epsilon 1e-5
main ()
{ int i;
  float a,s;
  clrscr();
  s=0; i=1;
  do
    {a=1.0/i/i;
     s+=a;
     i++;
    }
   while (a>epsilon);
   printf("s=%7.4f",s);
   getch();
}
```

Көрсетілген дәлдік үшін программа жұмысының нәтижесі: `S= 1.6418`

3.15 есеп. Берілген қосындыны

$$s = \sum_{i=1}^{n} \frac{1}{i!} = 1 + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \ldots + \frac{1}{n!}$$

енгізілетін *n* натурал саны үшін табу керек. Мұнда `for` циклін пайдалана отырып, факториалды және қосындыны цикл ішінде табамыз (3.16 сурет).



3.15 сурет. Қосынды табу алгоритмі

```
// s = 1+1/2!+1/3!+...+1/n!
#include <stdio.h>
#include <conio.h>
void main(void)
{  int p=1,i,n;
```

```
        float s=0;
        clrscr();
        printf("n = ");
        scanf("%d",&n);
        for (i=1; i<=n; i++)
         {   p *= i;
             s += 1.0/p;
         }
        printf("\ns=%f",s);
        getch( );
    }
```

Мүшелер саны n=10 болған кездегі программа нәтижесі:

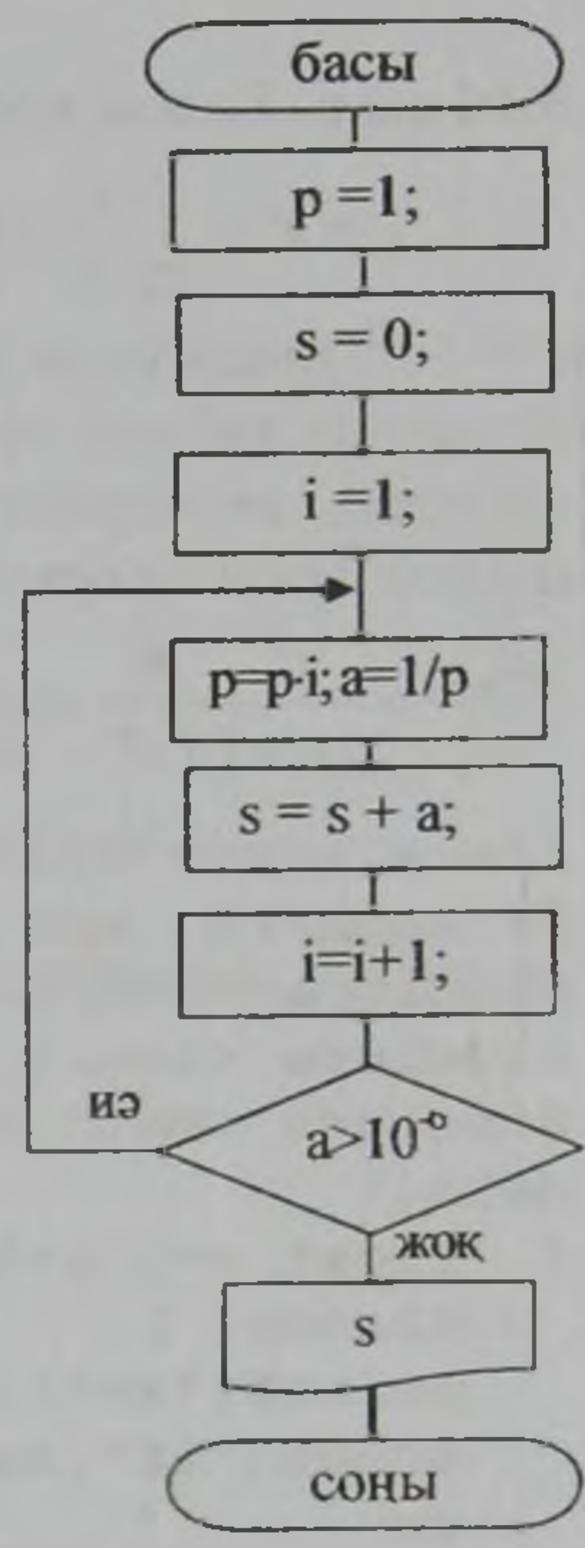
```
S=1.718222
```

Егерде n 18-ден артық болса, 1/n! мәні машиналық нөл мәніне (бөлімі өте үлкен сан) айналып, санның нөлге бөлінгені жайлы мәлімет шығады.

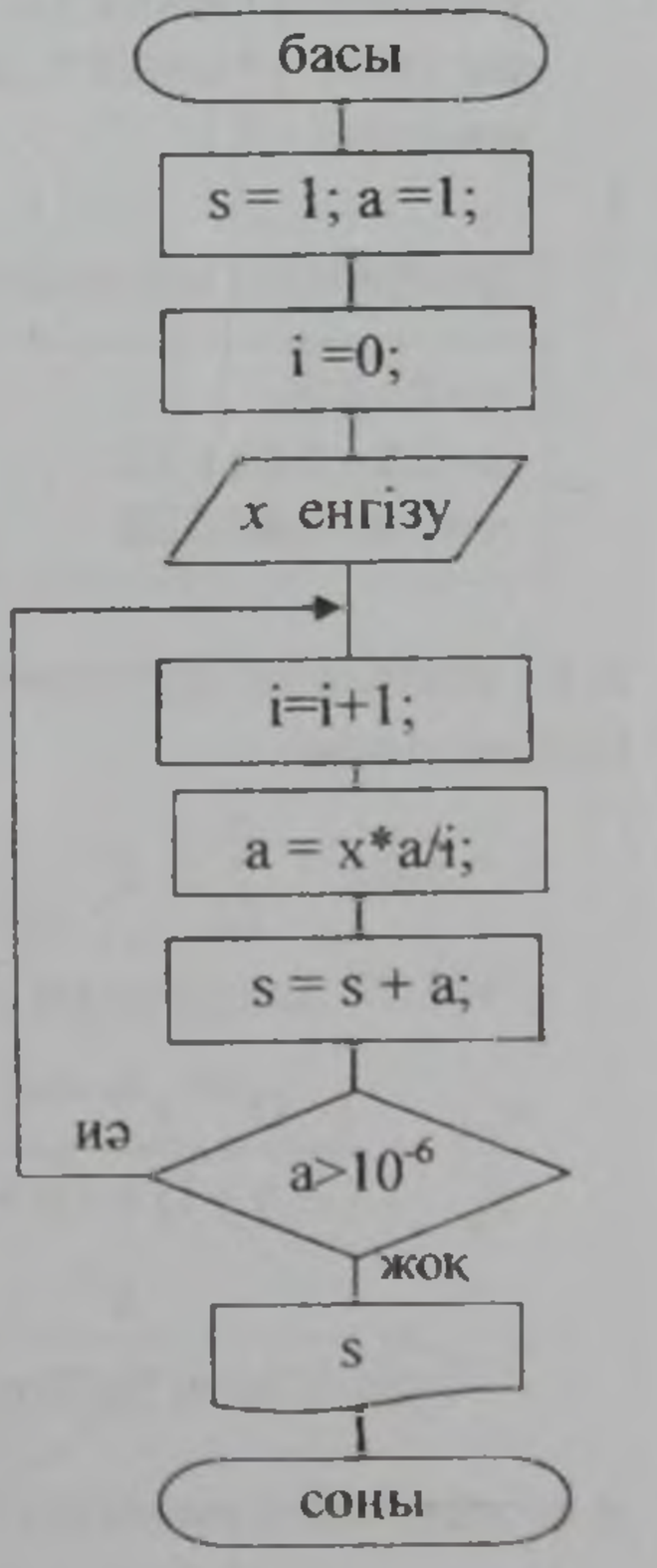
Осы қатарда n шексіздікке ұмтылғанда, қосындыны берілген $\varepsilon = 10^{-6}$ дәлдігімен анықтау керек болсын делік. Қатардың алғашқы бірнеше мүшесі есептеліп, оның келесі мүшесінің модулі ε санынан аз болғанда, қажетті дәлдікке қол жеткізілді деп есептеледі (3.16 сурет).

```
// s=1+1/2!+1/3!+...+1/n!+... дәлдігі 1e-6
#include <stdio.h>
#include <conio.h>
 void main(void)
{  int p=1,i=1,n;
   float a,s=0;
    clrscr( );
   do
   { p *= i++;
     a=1.0/p;
     s += a;
   }
   while(a > 1e-6);
    printf("\ns=%f",s);
    getch( );
}
```

Әртүрлі есептерді шығару кезінде sinx, cosx, logx, e^x, т.с.с. элементар функцияларды есептеу қажет болып жатады. Бұларды шығаруды жеңілдету үшін алдын ала олардың қатарға жіктелуі программаланып, жүйелік кітапханаларда сақталады да, керек кезінде сол программалар пайдаланылады. Төменде бірнеше функцияларды қатарға жіктеу арқылы есептеу программасы келтірілген.



3.18 сурет. Шексіз қосынды табу алгоритмі



3.17 сурет. e^x функциясын есептеу алгоритмі

3.16 есеп. Келесі қосындыны

$$s = e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^i}{i!} + \ldots$$

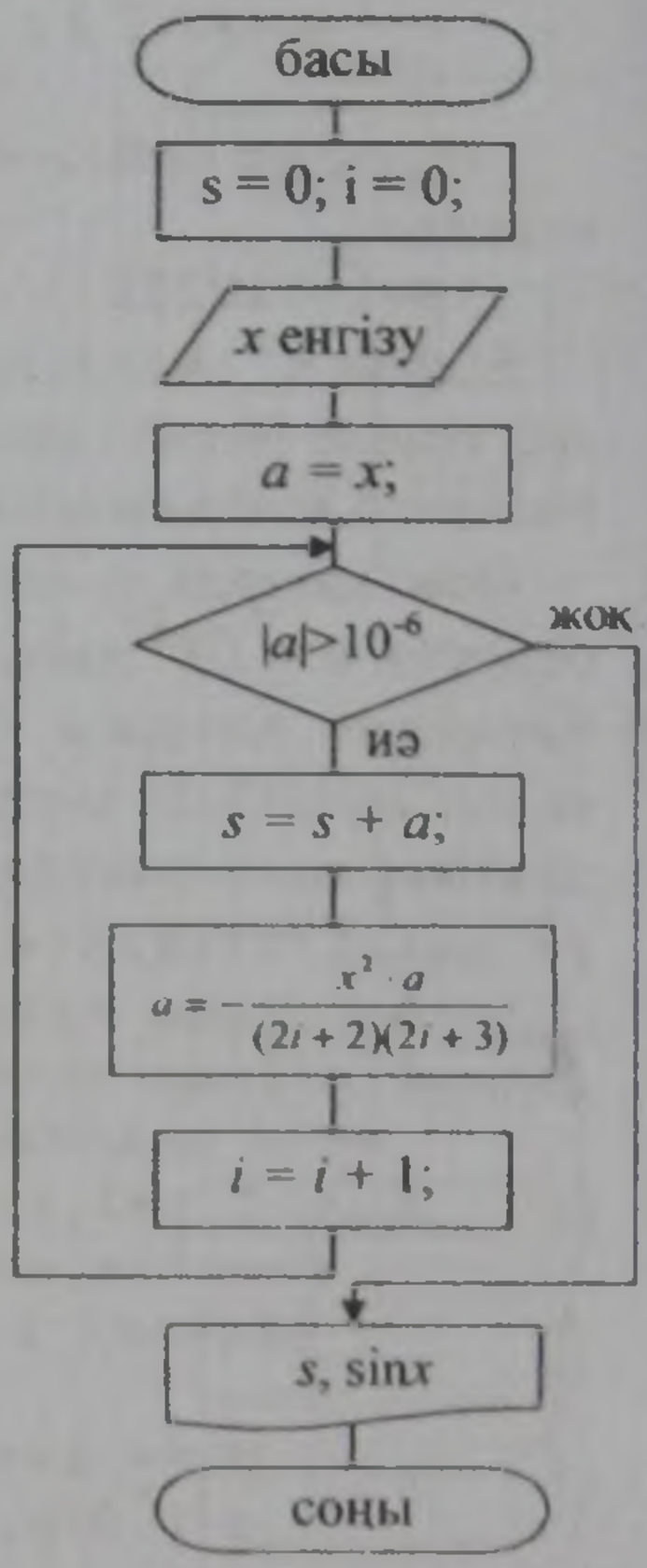
$\varepsilon = 10^{-6}$ дәлдігімен табу керек болсын. Мұндағы x берілген кез келген нақты сан. Қосынды табуды жеңілдету үшін төмендегі қатынас арқылы қатардың еселігін анықтап, реккуренттік формуланы пайдаланып, оның келесі мүшесін алдыңғы мүшесі арқылы өрнектеп алу қажет (3.17 сурет).

$$\frac{a_{n+1}}{a_n} = \frac{x^{n+1} \cdot n!}{(n+1)!x^n} = \frac{x}{n+1}, \quad a_{n+1} = \frac{x}{n+1} \cdot a_n$$

```
//s = 1+x+x^2/2!+x^3/3!+x^4/4!...+x^n/n!+...
// дәлдігі  eps = 1e-6, exp(x) функциясы
#include <stdio.h>
#include <conio.h>
#include <math.h>
main()
{   float s=1,a=1,x; int i=0;
    clrscr( );
    printf("x=");
    scanf("%f",&x);
    do
    {   a*=x/++i; s+=a; }
    while(a > 1e-6);
    printf("s=%f\n",s);
    printf("s=%f",exp(x));
    getch( );
}
```

Программа нәтижесі:

```
x=2.75
s=15.642632
s=15.642632
```



3.18 сурет. $\sin(x)$ функциясын есептеу алгоритмі

3.17 есеп. $\sin x$ функциясын жуықтап есептейтін шексіз қосындыны

$$s = \sin x = \sum_{i=0}^{\infty} (-1)^i \cdot \frac{x^{2i+1}}{(2i+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$$

$\varepsilon = 10^{-6}$ дәлдігімен табу керек. Мұнда да қатар еселігін анықтаймыз:

$$\frac{a_{n+1}}{a_n} = \frac{(-1)^{n+1} x^{2(n+1)+1} \cdot (2n+1)!}{(2(n+1)+1)! \cdot (-1)^n x^{2n+1}} = -\frac{x^{2n+3}(2n+1)!}{(2n+3)!x^{2n+1}} = -\frac{x^2}{(2n+2)(2n+3)},$$

$$a_{n+1} = -\frac{x^2}{(2n+2)(2n+3)} \cdot a_n;$$

Алгоритмнің мәтіндік түрде жазылуы:

1-қадам. $s = 0$; $i = 0$; деп қабылдау.

2-қадам. Берілген x мәнін енгізу.

3-қадам. $a = x$; деп қабылдау.

4-қадам. Егер $|a| \le 10^{-6}$ болса, 8-қадамға көшу.

5-қадам. s = s + a; деп қабылдау.

6-қадам. i = i + 1; деп қабылдау.

7-қадам. $a = -\dfrac{x^2 \cdot a}{(2i+2)(2i+3)}$; деп қабылдап, 4-қадамға көшу.

8-қадам. s және sinx мәндерін шығару.

9-қадам. Соны.

Алгоритмнің блоктар тізбегі түрінде жазылуы 3.18-суретте келтірілген. Ал оның С тіліндегі программасы:

```
//s=x-x^3/3!+x^5/5!-x^7/7!+...
// дәлдігі  eps = 1e-6, sin(x) функциясы
#include <stdio.h>
#include <conio.h>
#include <math.h>
main()
{  float s=0,a,x; int i=0;
   clrscr( );
   printf("x=");
   scanf("%f",&x);
   a=x;
   while(fabs(a) > 1e-6)
   {  s+=a;
      a*=-x*x/(2*i+2)/(2*i+3);
      i++;
   }
   printf("s=%f\n",s);
   printf("sin(%4.2f)=%f",x,sin(x));
   getch( );
}
```

Программа жұмысының нәтижесі:

```
x=0.85
s=0.751280
sin(0.85)=0.75280
```

Тапсырмалар

Берілген есептердің блок-схемасы мен есептеу программасын жазыңдар.

Келесі есептеулерді қатардың кезекті мәні абсолюттік шамасы бойынша 10^{-6} санынан кем болғанда тоқтату керек.

1. Қосындыны $\mathbf{s} = \sum_{i=1}^{\infty} \frac{i}{(i+1)(i+2)} = \frac{1}{2\cdot 3} + \frac{2}{3\cdot 4} + \ldots + \frac{i}{(i+1)(i+2)} + \ldots$ табу керек.

2. Қосындыны $\mathbf{s} = \sum_{i=1}^{\infty} \frac{\cos ix}{i^2} = \cos x + \frac{\cos 2x}{4} + \frac{\cos 3x}{9} + \ldots + \frac{\cos ix}{i^2} + \ldots$ кез келген нақты x үшін анықтау керек.

3. Қосындыны $\mathbf{s} = \cos x = \sum_{i=0}^{\infty} (-1)^i \cdot \frac{x^{2i}}{(2i)!} = x - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots + (-1)^i \frac{x^{2i}}{(2i)!} + \ldots$

x-тің мәні $\pi/10$-нан π-ге дейін қадамы $\pi/10$ болып өзгергенде анықтау қажет (cos(x) функциясының жіктелуі).

4. Қосындыны $s = \sin x = \sum_{i=0}^{\infty} (-1)^i \cdot \frac{x^{2i+1}}{(2i+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$

x-тің мәні $-\pi/2$-ден 2π-ге дейін қадамы $\pi/15$ болып өзгергенде анықтау қажет (sin(x) функциясының жіктелуі).

5. Қосындыны $\mathbf{s} = e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^i}{i!} + \ldots$ x-тің мәні -2-ден 2-ге дейін қадамы 0,25 болып өзгергенде анықтау қажет (exp(x) функциясының жіктелуі).

6. Қосындыны

$$\mathbf{s} = \sum_{i=0}^{\infty} \frac{(x-1)^{2i+1}}{(2i+1)(x+1)^{2i+1}} = \frac{x-1}{x+1} + \frac{(x-1)^3}{3(x+1)^3} + \frac{(x-1)^5}{5(x+1)^5} + \ldots + \frac{(x-1)^{2i+1}}{(2i+1)(x+1)^{2i+1}} + \ldots$$

x-тің мәні 1-ден 2-ге дейін қадамы 0,1 болып өзгергенде анықтау қажет.

7. Төмендегі қосындыларды $\varepsilon = 10^{-6}$ дәлдігімен табу керек.

а) $S = \sum_{k=1}^{\infty} \left(\frac{1}{k}\right)$; ә) $S = \sum_{k=1}^{\infty} \left(\frac{1}{k^5}\right)$; б) $S = \sum_{k=1}^{\infty} \frac{1}{(2k+1)^2}$.

8. Төмендегі шексіз қосындыларды да берілген дәлдікпен ε ($\varepsilon = 10^{-7}$) есептеңдер.

а) $S = \sum_{i=1}^{\infty} \frac{1}{i^2} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{5^2} + \ldots + \frac{1}{i^2} + \ldots$

ә) $S = \sum_{i=1}^{\infty} \frac{1}{i\cdot(i+1)} = \frac{1}{1\cdot 2} + \frac{1}{2\cdot 3} + \frac{1}{3\cdot 4} + \ldots + \frac{1}{i\cdot(i+1)} + \ldots$

б) $S = \sum_{i=0}^{\infty} \frac{(-1)^i}{i!} = 1 - \frac{1}{2!} + \frac{1}{3!} + \ldots + \frac{(-1)^i}{i} + \ldots$

в) $S = \sum_{i=1}^{\infty} \frac{(-2)^i}{i!} = -2 + \frac{4}{2!} - \frac{8}{3!} + \ldots + \frac{(-2)^i}{i!} + \ldots$

г) $S = \sum_{i=1}^{\infty} \frac{(-1)^{i+1}}{i\cdot(i+1)\cdot(i+2)} = \frac{1}{1\cdot 2\cdot 3} - \frac{1}{2\cdot 3\cdot 4} + \frac{1}{3\cdot 4\cdot 5} + \ldots + \frac{(--1)^{i+1}}{i\cdot(i+1)\cdot(i+2)} + \ldots$

д) $S=\sum_{i=0}^{\infty}\frac{1}{(4^i+5^{i+2})}=\frac{1}{1+5^2}+\frac{1}{4+5^3}+\frac{1}{4^2+5^4}+\ldots+\frac{1}{(4^i+5^{i+2})}+\ldots$

9. Төмендегі шексіз қосындыларды x айнымалысын енгізу арқылы берілген дәлдікпен ε ($\varepsilon=10^{-6}$) есептеңдер.

а) $S=\sum_{i=0}^{\infty}\frac{(x-1)^{2i+1}}{(2i+1)\cdot(x+1)^{2i+1}}=\frac{x-1}{x+1}+\frac{(x-1)^3}{3(x+1)^3}+\ldots+\frac{(x-1)^{2i+1}}{(2i+1)(x+1)^{2i+1}}+\ldots$

ә) $S=\exp x=\sum_{i=0}^{\infty}\frac{x^i}{i!}=1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+\ldots+\frac{x^i}{i!}+\ldots$

б) $S=\sum_{i=0}^{\infty}\frac{(x-1)^{2i+1}}{(2i+1)\cdot(x+1)^{2i+1}}=\frac{x-1}{x+1}+\frac{(x-1)^3}{3(x+1)^3}+\ldots+\frac{(x-1)^{2i+1}}{(2i+1)(x+1)^{2i+1}}+\ldots$

в) $S=\ln(1+x)=\sum_{i=0}^{\infty}\frac{(-1)^i x^{i+1}}{i+1}=x-\frac{x^2}{2}+\frac{x^3}{3}-\frac{x^5}{5}+\ldots+\frac{(-1)^i x^{i+1}}{i+1}+\ldots$

г) $S=arctgx=\sum_{i=0}^{\infty}\frac{(-1)^i x^{2i+1}}{2i+1}=x-\frac{x^3}{3}+\frac{x5}{5}-\frac{x^7}{7}+\ldots+\frac{(-1)^i x^{2i+1}}{2i+1}+\ldots$

Тапсырмалар нұсқалары

1 нұсқа

1) 200-ге дейінгі 13-ке қалдықсыз бөлінетін бүтін сандардың қосындысын табу керек.
2) Жалпы мүшесі төменгі өрнекке сәйкес қатар қосындысын $\varepsilon=10^{-4}$ дәлдігімен анықтау керек.

$$a_n=\frac{(-1)^{n-1}}{n^n}$$

2 нұсқа

1) 100-ге дейінгі жұп он сандардың қосындысын табу керек.
2) Жалпы мүшесі төменгі өрнекке сәйкес қатар қосындысын $\varepsilon=10^{-4}$ дәлдігімен анықтау керек.

$$a_n=\frac{1}{2^n}+\frac{1}{3^n}$$

3 нұсқа

1) 200-ге дейінгі тақ он сандардың қосындысын табу керек.
2) Жалпы мүшесі төменгі өрнекке сәйкес қатар қосындысын $\varepsilon=10^{-4}$ дәлдігімен анықтау керек.

$$a_n=\frac{1}{((3n-2)(3n+1))}$$

4 нұсқа

1) 200-ге дейінгі 17-ге бөлінетін сандар қосындысын табу керек.
2) Жалпы мүшесі төменгі өрнекке сәйкес қатар қосындысын $\varepsilon=10^{-4}$ дәлдігімен анықтау керек.

$$a_n = \frac{(2n-1)}{2^n}$$

5 нұсқа

1) 200-ге дейінгі 19-ға бөлінетін сандар нешеу екенін табу керек.
2) Жалпы мүшесі төменгі өрнекке сәйкес қатар қосындысын $\varepsilon=10^{-4}$ дәлдігімен анықтау керек.

9) $a_n = \frac{10^n}{n!}$

6 нұсқа

1) 200-ге дейінгі 21-ге бөлінетін сандар қосындысын табу керек.
2) Жалпы мүшесі төменгі өрнекке сәйкес қатар қосындысын $\varepsilon=10^{-4}$ дәлдігімен анықтау керек.

$$a_n = \frac{n!}{(2n)!}$$

7 нұсқа

1) 200-ге дейінгі 21-ге бөлінетін сандар нешеу екенін табу керек.
2) Жалпы мүшесі төменгі өрнекке сәйкес қатар қосындысын $\varepsilon=10^{-4}$ дәлдігімен анықтау керек.

$$a_n = \frac{n!}{n^n}$$

8 нұсқа

1) 200-ге дейінгі 5-ке бөлгенде 2 қалдық қалатын сандар нешеу екенін табу керек.
2) Жалпы мүшесі төменгі өрнекке сәйкес қатар қосындысын $\varepsilon=10^{-4}$ дәлдігімен анықтау керек.

$$a_n = \frac{2^n n!}{n}$$

9 нұсқа

1) 200-ге дейінгі 5-ке бөлгенде 4 қалдық қалатын сандар нешеу екенін табу керек.
2) Жалпы мүшесі төменгі өрнекке сәйкес қатар қосындысын $\varepsilon=10^{-4}$ дәлдігімен анықтау керек.

$$a_n = \frac{3^n n!}{(3n)!}$$

10 нұсқа

1) 200-ге дейінгі бүтін сандардың 25-ке қалдықсыз бөлінетін сандары нешеу екенін табу керек.

2) Жалпы мүшесі төменгі өрнекке сәйкес қатар қосындысын $\varepsilon=10^{-4}$ дәлдігімен анықтау керек. $a_n = \dfrac{n!}{3n^n}$

11 нұсқа

1) 200-ге дейінгі 10-ға бөлінетін сандар нешеу екенін табу керек.

2) Жалпы мүшесі төменгі өрнекке сәйкес қатар қосындысын $\varepsilon=10^{-4}$ дәлдігімен анықтау керек.

$$a_n = \frac{(n!)^2}{2^{n^2}}$$

12 нұсқа

1) 200-ге дейінгі 20-ға бөлінетін сандар нешеу екенін табу керек.

2) Жалпы мүшесі төменгі өрнекке сәйкес қатар қосындысын $\varepsilon=10^{-4}$ дәлдігімен анықтау керек.

$$a_n = \lg(n!)e^{-n\sqrt{n}}$$

13 нұсқа

1) 200-ге дейінгі 25-ке бөлінетін сандар нешеу екенін табу керек.

2) Жалпы мүшесі төменгі өрнекке сәйкес қатар қосындысын $\varepsilon=10^{-4}$ дәлдігімен анықтау керек.

$$a_n = 10^{-n}(n-1)!$$

4 C/C++ ТІЛДЕРІНДЕГІ ҚОЛДАНУШЫ ФУНКЦИЯЛАРЫ

Күрделі әрі көлемді программалар жазу кезінде, программаны *функцияларға* – шағын бөліктерге бөлу арқылы жұмысты жеңілдетуге және бастапқы код көрнекілігін арттыруға болады. Мысалы, ағымдағы бухгалтерлік шоттармен жұмыс істейтін программа жазу қажет болсын делік. Мұнда кітап операцияларын орындайтын басты бір функция жазып, шоттарды төлеу ісін аткаратын тағы бір функция, сонан соң қабылдап алынатын шоттарды реттейтін функция және баланстық есеп беру жұмысын орындайтын функциялар карастырылса, программаны оқу, түсіну жеңіл болар еді. Егер осы әрекеттерді орындайтын барлық операторлар **main** ішінде орналаскан болса, онда программа өте көлемді болып кетеді және оны түсіну де қиын болар еді. Оның үстіне, программа көлемі мен күрделілігі арткан сайын, онда қате жіберу мүмкіндігі де ұлғаяды.

Берілген бір тапсырманы аткаратын атаулы (өзіндік аты бар) операторлар жиыны *қолданушы функциясы* болып табылады. Мысалы, **hello_world** деп аталатын келесі функция **printf()** операторы арқылы экранға мәлімет жазып шығарады:

```
void hello_world(void)
{
  printf("Hello world!!!\n");
}
```

Көбінесе функциялар программаға өз есептеулерінің нәтижесін қайтарады. Егер функция ешқандай да мән қайтармайтын болса, онда ол функция атының алдында **void** түйінді сөзі көрсетілуі қажет. Функция атынан кейін жақша ішінде оның параметрлері көрсетіледі. *Параметр – бұл функцияға өңдеу үшін берілетін мәлімет*. Егер функцияның параметрлері болмаса, онда дөңгелек жақшалар ішіне **void** түйінді сөзі жазылады немесе ештеңе көрсетілмейді. Функцияны пайдалануды функцияны шақыру деп атайды. Келесі программада **hello_world** функциясы пайдаланылады.

```
#include <stdio.h>
void hello_world()
{
  printf("Hello_world()!!!\n");
}
void main()
{
  hello_world();
}
```

Программаны орындау нәтижесі:

```
Hello world!!!
```

Сонымен, программаны іске коскан кезде бірінші болып **main** функциясы орындалады. Жоғарыдағы программада **hello_world** функциясын шақыру **main** функциясында аткарылады. Егер программаны орындау кезінде функцияны шақыру әрекеті кезіксе, онда басқару бірден функцияға беріледі де, онын

операторлары ең алғашқы жолдан бастап орындала бастайды. Функцияның ең соңғы операторы орындалып болған соң, басқару негізгі программадағы функцияны шақыру жолынан кейін тұрған операторға беріледі.

Функцияларды пайдалану программалаушыларға модульдік программалауды пайдалану мүмкіндігін береді, яғни программаны модульдерден құрастыруға болады. Әрбір функция белгілі бір тапсырманы (есепті) орындауы тиіс және функция аты сол тапсырманың мазмұнына сәйкес келуі керек. Сонда ғана бұл функцияны басқа программаларды жазу кезінде де қолдануға болады.

4.1 Функцияларда айнымалыларды пайдалану

Программалаушы өз функцияларын жазғанда, онда айнымалыларды пайдалану қажеттілігі туындайды. Айнымалылар `main` функциясындағы тәрізді функцияда жариялануы, яғни алдын ала сипатталуы қажет.

Келесі программада функция параметрлерін қолдану мысалы қарастырылады.

```
#include <iostream.h>
void hello_world()
{
   cout << "Hello world!!!" << "\n";
}
void hello_number(int nh)
{
   int counter;
   for (counter = 1; counter <= nh; counter++)
   hello_world();
}
void main()
{
   int kol;
   cout << "How much time?" << "\n";
   cin >> kol;
   hello_number(kol);
   cout << "How much time?" << "\n";
   cin >> kol;
   hello_number(kol);
}
```

Программаны орындау нәтижесі төменде көрсетілген.

```
How much time?
2
Hello world!!!
Hello world!!!
How much time?
5
Hello world!!!
Hello world!!!
Hello world!!!
Hello world!!!
Hello world!!!
```

4.2 Функцияның мән қайтаруы

Үнемі функция нақты бір жұмысты атқаруды (мысалға, қандай да бір хабарламаға қорытынды), және де шақыру функциясының есептеу нәтижесін қайтаруды талап етеді. Шақыру кодының нәтижесін қайтаруды қамтамасыз ету үшін функцияда `return` операторын пайдалану керек.

Функцияның типі оның (`int`, `float`, `char`, т.б.) мағынасының қайтарылу типімен анықталады. Мысалы, егер функция `int` типін қайтаратын болса, онда сол типті функция идентификаторы алдында көрсету қажет:

```
int some_function(int value)
{
   // функция операторлары
}
```

Келесі `icube` функциясы параметр ретінде берілген саннын кубын қайтарады. Мысалы, функцияға 5 саны берілсе, онда `icube` функциясы 5·5·5 мәнін, яғни 125 мәнін қайтарады:

```
int icube(int value)
{
  return(value * value * value);
}
```

Сонымен, функция шақырылған соң, онын жұмыс нәтижесін қайтару үшін `return` операторы қолданылады (төмендегіні қараңыз). Программада функцияның қайтарған мәні айнымалыға меншіктеледі немесе басқа бір функцияның параметрі ретінде (`printf` немесе `cout`) келесідей түрде көрсетілуі мүмкін:

```
   result = icube(5);
```

немесе

```
   printf("5 cubed is %d\n", icube(5));
```

Енді толық программа карастырайык.

```
#include <iostream.h>
int icube (int value)
{
   return (value * value * value);
}
void main ()
{
   cout << "3 cubed is " << icube(3) << "\n";
   cout << "5 cubed is " << icube(5) << "\n";
   cout << "7 cubed is " << icube(7) << "\n";
}
```

Программаны орындау нәтижелері:

```
3 cubed is 27
5 cubed is 125
7 cubed is 343
```

Функцияға берілетін мән онын параметрі типіне сәйкес болу керек. Егер, мысалы, нақты (жылжымалы нүктелі сан) санның үшінші дәрежесін есептеу керек болса, онда типі басқа `fcube` функциясын құру керек:

```
float fcube(float value)
{
  return(value * value * value);
}
```

C/C++ тілдері `return` сөзі кездескенде, бірден функция жұмысын аяқтап, нәтижесін негізгі программаға (оны шақырушы функцияға) қайтарады. Функцияны орындау кезінде `return` операторынан кейін тұрған кез келген операторлар еленбей қалады. Программа өз жұмысын функцияны шақырған жолдан ары қарай жалғастыра береді.

Кей жағдайда функция тұлғасы (операторлары) ішінде бірнеше `return` операторлары болуы мүмкін, мұндайда функция программаға өзіндегі әрбір шарт үшін жеке есептелген мәндерді қайтарады.

Мүмкіндігіне қарай функцияда тек қана бір `return` операторын қолдану ұсынылады. Функцияның көлемді және күрделі болып, онда бірнеше `return` операторларының тұруы оның жұмысын түсінуді қиындатады. Көбінесе, функцияны онда тек бір `return` операторы қолданылатындай етіп түрлендіріп жазуға болады, мысалы, екі мәнді салыстыратын бір функция жазайық:

```
int compare_values(int first, int second)
{
    int result;
    if (first == second)
       result = 0;
    else if (first > second)
       result = 1;
    else if (first < second)
       result = 2;
  return (result);
}
```

Бұл программа тым қарапайым болғандықтан, бір `return` операторын қолдану артықшылығын түсіну қиындау шығар.

Бірақта кейбір жағдайларда кодтың көрнекілігін арттыру үшін бірнеше `return` операторын пайдалану да дұрыс болып саналады.

C/C++ компиляторлары функцияның қайтарылатын мәнінің типі мен оның параметрлері типі жайлы мәліметтің программада функцияны шақыруға дейін тұруын талап етеді. Компилятордың функцияны анықтауды оны бірінші пайдаланудан бұрын коюды талап етуі компьютер жадында программа мәліметтерін дұрыс орналастыруды қамтамасыз етеді. Бірақ программаның күрделілігінің артуы нәтижесінде бұл талап орындалмай қалуы мүмкін. Сондықтан мұндай жағдайда функция параметрлері мен қайтарылатын мән типтерін алдын ала қысқаша сипаттап кететін функция прототипін қолдану қарастырылған.

Мысалы, `i_cube` және `f_cube` функцияларын қолданатын программаны прототиптерді көрсетіп қайта қарастырайық:

```
#include <stdio.h>
int i_cube (int);      // i_cube функциясының прототипі
float f_cube (float); // f_cube функциясының прототипі
void main ()
 {
  printf("3 cubed is %d\n",i_cube(3)); //i_cube функциясын шақыру
  printf("3.7 cubed is %f\n", f_cube(3.7));//f_cube функциясын
                                          // шақыру
 }
int i_cube (int value) // icube функциясын сипаттау
 {
  return (value * value * value);
 }
float f_cube (float value) // f_cube функциясын сипаттау
{
  return (value * value * value);
}
```

Программаны орындау нәтижелері:

```
3 cubed is 27
3.7 cubed is 50.653002
```

Функция прототипі нүктелі үтірмен аяқталады, ал функцияны сипаттау кезінде тақырып жолынан соң функция операторлары жазылып, тек солардан кейін ғана нүктелі үтір орналасады.

Өз (тұтынушы) функцияларыңызды құрғаннан соң, бір программа үшін жазылған функцияны басқа бір программада да колдануға болады. Бір рет жазылған функцияны бірнеше программаларда пайдалану мүмкіндігі программалау және оны түзету кезеңдерін қысқартуға мүмкіндік береді.

4.3 Нақты және формальды (көрсетпелі) параметрлер

Формальды параметрлер деп функция анықталуында көрсетілетін параметрлер атауларын айтамыз. Мысалы, мынадай функцияда:

```
void job_information (int age, float salary, int number_job)
{
   // функциялар операторлары
}
```

`age, salary, number_job` атаулары берілген функция үшін формальды немесе көрсетпелі параметрлер болып саналады.

Функцияны программадан немесе басқа бір функциядан шақырған кезде, функцияға берілетін мәндер нақты немесе қойылмалы параметрлер болып табылады.

Төмендегі мысалда берілген 30, 42000.00 және 321 мәндері `job_information` функциясын шақыру кезіндегі көрсетпелі параметрлерді алмастырып солардың орнына қойылатын нақты параметрлер болып табылады:

```
job_information (30,42000.00,321);
```

Функцияға тасымалданып қойылатын параметрлер сандық мән немесе айнымалы түрінде болуы мүмкін. Оларға қойылатын жалғыз талап – қойылмалы нақты параметрдің (немесе айнымалының) типі көрсетпелі параметр типімен бірдей болуы тиіс. Келесі код фрагменті қойылмалы параметр орнына айнымалыны пайдалану тәсілін көрсетеді:

```
int workers_age = 30;
float workers_salary = 42000.00;
int job_number = 321; ,
job_information (workers_age, workers_salary, job_number);
```

4.4 Функциялар мен айнымалылар атауларының әрекет ету аймағы

Программадағы функциялар мен айнымалылар атауларының өз әрекет ету (немесе көріну) аймағы, яғни оларды пайдалануға болатын, мәндері белгілі болып келетін программаның жұмыс аймағы болады.

Келесі программаны қарастырайық.

```
#include <stdio.h>
void pr(int pr_count)
{
    int count;
    for (count = 1; count<=pr_count; count++)
        putchar('A');
}
void main(void)
{
  int count;
  for (count = 1; count <= 3; count++)
    {
      printf("\n Your computer should print %d time(s)\n",
                                                    count);
      pr(count);
    }
  printf("\n");
}
```

Программаның орындалу нәтижелері:

```
Your computershould print 1 time(s)
A
Your computer should print 2 time(s)
AA
Your computer should print 3 time(s)
AAA
```

Мұнда `pr` және `main` функцияларының әрқайсысында `count` айнымалысы пайдаланылады, бірақ бұл айнымалылардың екеуі екі түрлі болып табылады, өйткені олардың әрқайсысының өз жұмыс аймағы бар. `Pr` функциясындағы

`count` айнымалысы тек функцияның орындалуы кезінде ғана белгілі (көріну аймағы функция ішінде ғана) болып саналады. Сол сияқты екінші `count` айнымалысы тек `main` функциясының орындалуы кезінде ғана жұмыс істей алады. Сондықтан `for` циклiндегі `count` айнымалысының өзгеру нәтижесі `main` функциясындағы `count` айнымалысына ешқандай әсерін тигізбейді.

Айнымалының жұмыс істеу аймағын қарастыру кезінде локальдық және глобальдық айнымалылар түсінігі жиі пайдаланылады.

Локальдық (жергілікті) айнымалы – бұл белгілі бір функцияның ішкі операторларында ғана белгілі болып табылатын, әрекет ету аймағы шектеулі айнымалы.

Глобальды (ауқымды) айнымалы – бұл программаның барлық аймақтарында толық белгілі болып саналатын айнымалы.

Жоғарыла мысалда көрсетілген `count` айнымалысы тек қарастырылған функция ішінде ғана анықталған локальдық айнымалы болатын.

4.5 Функция прототиптері

Функция прототипі компиляторға функция қайтаратын мәлімет типін, функцияда қолданылатын параметрлер санын, параметрлер типтерін және олардың орналасу реттілігін (мысалдағы екінші параметр `void`, яғни ештеңе жоқ) көрсетеді. Компилятор негізгі программадан (одан кейін жазылған) функцияны шақырудың дұрыстығын тексеру үшін функция прототипін қолданады. Прототип жазылған кезде функцияның сипатталуы негізгі программадан соң орнасады.

Прототиптің жазылу түрі: `void row(void);`

Бірінші `void` компиляторға функция қайтаратын мәннің типін (мұнда «ештеңе» қайтарылмайды) көрсетеді. Екінші `void` параметрлердің саны мен типін және солардың реттілігін көрсетуі (мұнда да «ештеңе») тиіс.

Компилятор функция прототипінде көрсетілген айнымалы атауларын елемейді. Мысалы, берілген үш бүтін санның максималды мәнін табу функциясының прототипі мынадай

```
int maximum (int a, float b, int c);
```

немесе

```
int maximum (int, float, int);
```

болуы да мүмкін.

Қысқаша жазу түрін пайдаланған дұрысырақ болып саналады.

4.6 Рекурсивті функциялар

Келесі функция void типіндегі мән қайтарады, яғни жұмыс аткарады, бірақ ешқандай мән бермейді:

```
void print(int gg, int mm, int dd)
{ printf("\n жыл:%d",gg);
  printf("\t ай:%d",mm);
  printf("\t күн:%d",gg);
}
```

Енді программада осы функцияны төмендегідей түрде шақырсақ,

```
printf(2012,10,15);
```

Оның нәтижесі:

```
жыл:2012 ай:10 күн:15
```

түрінде болады.

Рекурсивті функциялар. C/C++ тілінде функция өзін өзі шақырып та пайдалана алады, мұндай функция ***рекурсивті*** деп аталады.

Берілген n бүтін санының факториалын есептейтін (n!=1·2·3·...·n) рекурсивті функция мысалы:

```
fakt (int n)
{
   int b;
   if (n == 0) return 1;
      b = fakt(n-1)*n;
   return n;
}
```

Осы функцияны енгізілген n мәні үшін кез келген бүтін санның факториалын есептеуде қолданудың толық программасы төмендегі 4.1 мысалда келтірілген.

Функцияларды пайдалану есептері

4.0 есеп. x^n өрнегін есептеу функциясын құрып, оны тексеріп (тесттен өткізіп) шығу қажет.

```
#include <stdio.h>
#include <conio.h>
void main()
{
int n=-4;
float power(float, int), x=2.0;
clrscr () ;
printf("-3-тің 3 дәрежесі =%f\n", power(-3.0, 3));
printf("%4.2f-нің %3d дәрежесі=%6.2f\n",x,n,power(x,n));
printf("0-ді -2 дәрежесіне шығару: %6.2f\n",power(0.0, -2));
getch();
}
float power (float x, int n)
{
float step;
if(n>0)
{
for(step=1.0; n>0; n--) step*=x;
return step;
}
   else if(x!=0)
{
for( step=1.0; n<0; n++) step/=x;
```

```
return step;
}
else
 {
printf("Қате!!! ");
printf("%d-ті 0 дәрежеге шығаруға болмайды!\n",n);
return 0;
      }
}
```

Программа нәтижесі:

```
-3-тің 3 дәрежесі =-27.000000
2.00-нің -4 дәрежесі= 0.06
Қате!!! -2-ні 0 дәрежеге шығаруға
болмайды!
0-ді -2 дәрежесіне шығару: 0.00
```

`power()` функциясы оны шақырған функцияға бір ғана мән беруі керек. Бұл үшін `return` операторын қолдану ыңғайлы. Осындай тәсілмен қайтарылған мән функция жұмысының нәтижесі болып табылады және оны өрнектерде қолдануға болады.

Функция аргументтері оның жұмысынан кейін өзгермеуі керек, сондықтан ол функцияға *мәні арқылы* беріледі.

Жазылған функцияның жұмысының дұрыстығын тексеру үшін оны бірнеше берілген мәндер арқылы тесттен өткізу керек. Тест ретінде алынатын мәнді құрылған алгоритмнің барлық тармақтарын қадағалап шығатындай етіп таңдап алу қажет. Біздің мысалымызда дәрежені есептеу `main()` функциясында үш түрлі типтегі мәліметтер арқылы тексеріліп шықты, осылай тексеру жеткілікті болып саналады.

4.1 есеп. Бүтін сандардың факториалын анықтау программасы.

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
float long fakt(int m)
{
 if (m==0)
    return 1;
 else
    return((float)m*fakt(m-1));
}
main ()
{
 float p;
 int n;
```

```
 printf("Бүтін сан енгізіңіз n : ");
 scanf("%d",&n);
 for (int k=0; k<=n; k++)
  { p=fakt(k);
    printf(" %i!=  %-15.0f\n",k,p);
  }
 puts("Аяқтау үшін Enter\n");
 getch();
}
```

Келесі функция да ешқандай мән қайтармайды және оның ешқандай параметрлері де жоқ.

4.2 есеп. Ағымдағы уақытты бейнелеу.

```
#include <stdio.h>
void real_time(void)
{
 printf("\nҚазіргі уақыт: %s",TIME "(сағ:мин:сек.)");
}
main()
{ real_time(); }
```

Программа нәтижесі:

```
Қазіргі уақыт: 14:16:25 (сағ:мин:сек.)
```

Прототипті көрсетуді осы мысал аркылы былай көрсетуге болады:

```
#include <stdio.h>
void real_time(void);
main()
{ real_time(); }
void real_time(void)
{
  printf("\n Қазіргі уақыт: %s",TIME "(сағ: мин: сек.)");
}
```

4.3 есеп. Берілген екі бүтін санның кішісін анықтайтын функцияның ішкі мәтіні мынадай болуы мүмкін:

```
  int  mini(int x, int y)
{ int t;
   if (x<y) t=x;
      else t=y;
   return t;
}
```

Бұл функцияны төмендегідей түрде басқаша да жазуға болады

```
mini(int x, int y)
{ return (x<y)? x:y; }
```

Қайтарылатын нәтиже типі көрсетілмесе, ол келісім бойынша бүтін сан (int) типі болып саналады.

4.4 есеп. Берілген он бүтін санның факториалын анықтау функциясын жазу:

```
#include <conio.h>
#include <stdio.h>
long fakt(int k)
      { if (k<0) return 0;
        if (k==0) return 1;
        long p=1;
        for (int j=1; j<=k;j++)
           p*=j;
        return (p);
      }
main ()
{int n; float c;
 printf("n = ");
 scanf("%d",&n);
 c = fakt(n);
 printf("%d!=%lf\n",n,c);
 getch();
}
```

Теріс сандардың факториалы болмайды, мұндайда функция 0 мәнін береді, ал 0!=1, қалған жағдайларда бұл функция цикл ішінде факториалды есептейді.

4.5 есеп. Берілген m, n бүтін сандары үшін мынадай өрнекті есептеу керек:

$$c = \frac{m! \cdot n!}{(m+n)!}$$

```
 // c=( m!*n!)/(m+n)! есептеу
#include <conio.h>
#include <stdio.h>
long fakt(int k)
      { if (k<0) return 0;
        if (k==0) return 1;
        long p=1;
        for (int j=1; j<=k;j++)
           p*=j;
        return (p);
      }
main ()
{int m,n; float c;
 printf("m,n = ; ");
 scanf("%d%d",&m,&n);
 c=fakt(m)*fakt(n)/fakt(m+n);
 printf("c=%e\n",c);
 getch();
}
```

Функция ішінде орналасқан айнымалылар типтері сипатталуы осы функцияның ішінде ғана анықталған болып саналады, (мұнда тек қана **main** функциясы ерекше жағдайда болып саналады). Программа мәтінінде функциялар кез келген ретпен орналасуы мүмкін. Функцияны шақыру үшін функцияның аты және жақшада нақты аргументтері көрсетіледі.

4.6 есеп. Анықталған интегралды тіктөртбұрыштар, трапециялар және Симпсон тәсілдерімен жуық шамамен есептеуді функция арқылы орындау мысалы. Бұл есепті функциясыз қарапайым жолмен есептеу тәсілі 3.7 есепте келтірілген.

```
// интегралды қолданушы функциясы арқылы есептеу
// берілген интегралдағы өрнек y=3*x*x
// функция арқылы есептеу 2-тәсіл
// төртбұрыштар, трапеция және Симпсон тәсілдері
#include <iostream.h>
#include <iomanip.h>
#include <conio.h>
float f(float x)
{ return 3*x*x;} //интегралдағы функция
main()
{float a,b,        // кесінді шекаралары
      h,           // интервал мәні
      s1,s2,s3,    // интегралдың жуық мәндері
      x,x1,x2;     // аргумент
 int n,            // интервалдар саны
     i;            // цикл санауышы
clrscr();
cout << "Интегралды жуықтап есептеу\n";
cout << "кесіндінің төменгі шегі: ";
cin >> a;
cout << "кесіндінің төменгі шегі: ";
cin >> b;
cout << "кесінді неше бөлікке бөлінеді: ";
cin >> n;

h=(b-a)/n;
x1=a+h/2; x2=a;
s1=f(x1); // функцияны шақыру
s2=(f(x2)+f(b))/2;
for(i=1;i<n;i++)
 {
   x1=x1+h;x2=x2+h;
   s1=s1+f(x1); s2=s2+f(x2);
 }
s1=h*s1; s2=h*s2; s3=(2*s1+s2)/3;
cout << "Формулалар бойынша интеграл мәндері\n";
cout << "төртбұрыштар\t трапециялар\t Симпсон тәсілдері\n";
cout << setprecision(5); // шығару дәлдігі 5 таңба
```

```
 cout << "s1 = "<<s1<<"\t s2 = "<<s2<<"\t s3 = "<<s3;
 getch();
}
```

Программа жұмысы нәтижесі:

```
Интегралды жуықтап есептеу
кесіндінің төменгі шегі:
кесіндінің төменгі шегі:
кесінді неше бөлікке бөлінеді:
Формулалар бойынша интеграл мәндері
тертбұрыштар      трапециялар       Симпсон тәсілдері
s1 = 91.08859     s2 = 91.19794     s3 = 91.12504
```

4.7 есеп. Келесі программа фукция арқылы латынның бас әріпін кіші әріпке түрлендіреді.

```
/* Латынша бас әріпті кіші әріпке айналдыру */
#include <stdio.h>
#include <conio.h>
#include <ctype.h> //әріпті түрлендірудің прототипі
char clowercase(char c);
int main()
{char clowchar, chichar;
  clrscr();
  printf("Shift-ті басып тұрып латын әрпін енгізіңіз: ");
  chichar = getchar();
  clowchar = clowercase(chichar);
  printf ("%c\n", clowchar);
  getch();
  return(0);
}
char clowercase(char c)
{ return(tolower(c)); } // бұл функция прототипі <ctype.h>
```

Программа жұмысы нәтижесі:

```
Shift-ті басып тұрып латын әрпін енгізіңіз: G
g
```

4.8а есеп. Келесі программа функция арқылы енгізілген кез келген ондық бүтін санды екілік жүйеге түрлендіреді.

```
/* Енгізілген ондық санның екілік баламасын табу */
#include <stdio.h>
void vbinary (int idata);
int main()
{
int ivalue;
printf ("Ондық сан енгізіңіз: ");
```

```
scanf("%d", &ivalue) ;
vbinary (ivalue) ;
return(0) ;
}
void vbinary (int idata)
{ int t = 0;
   int iyourarray[50];
   while (idata != 0) {
   iyourarray[t] = (idata % 2);
   idata /= 2;
   t++; }
   t--;
      printf("\n");
      for( ; t >= 0; t--)
      printf("%d", iyourarray[t]);
      printf("\n");
   }
```

4.8ә есеп.Пернелерден енгізілген ондық санды екілік санға түрлендіру есебінің функциясыз жазылған нұсқасы:

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
main()
{
 unsigned dec,v,i;
clrscr();
printf("Enter an integer number up to 4 294 967 295 -> ");
scanf("%u", &dec);
printf("Decimal number %u in binary format is: ", dec);
v=1;
for(i=1; i<=15; i++) v*=2;
// 2-ні 31-ші дәрежеге былай да шығаруға болады
// v=(unsigned int)pow(2,31);
for(i=1; i<=16; i++)
{
   if (dec >= v)
   {
      printf("1");
      dec-=v;
   }
   else
      printf("0");
   v=v/2;
 }
 printf("\n");
 getch();
 return 0;
}
```

Программа жұмысы нәтижесі:

```
Enter an integer number up to 4 294 967 295 -> 28494
Decimal number %u in binary format is: 0110111101001110
```

4.9 есеп. Санды кері бағытта функция аркылы жазып шығу.

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
 void cifr(int m)
  ( while(m!=0)
    { int k=m%10;
      printf("%d",k);
      m=(m-k)/10;
    }
 }
main()
(
 int n;
 clrscr();
 printf("\nБерілген сан: \n");
 scanf("%d",&n);
 printf("\nКері жазылуы: ");
 cifr(n);
 getch();
 }
```

4.10 есеп. Сандарды көбейту кестесін – Пифагор кестесін шығару.

```
// Пифагор квадратын шығару 3.32
#include <iostream.h>
#include <iomanip.h>
#include <conio.h>
line(int k) // k келденең сызықтар сызу функциясы
  { for(int i=1;i<=k;i++)
      cout << "-";
    cout << endl;
  }
main()
{const n=50;
 int i,j;  // кестедегі жол және баған немірлері
 clrscr();
 cout << "\t    Кебейту (Пифагор) кестесі\n";
 line(n);
 for(i=1;i<=10;i++)
  { for(j=1;j<=10;j++)
      cout << setw(5) << i*j;
```

```
    cout<< endl;
  }
 line(n);
 getch();
}
```

Программа жұмысының нәтижесі:

```
                Көбейту (Пифагор) кестесі
-----------------------------------------------------
   1    2    3    4    5    6    7    8    9   10
   2    4    6    8   10   12   14   16   18   20
   3    6    9   12   15   18   21   24   27   30
   4    8   12   16   20   24   28   32   36   40
   5   10   15   20   25   30   35   40   45   50
   6   12   18   24   30   36   42   48   54   60
   7   14   21   28   35   42   49   56   63   70
   8   16   24   32   40   48   56   64   72   80
   9   18   27   36   45   54   63   72   81   90
  10   20   30   40   50   60   70   80   90  100
-----------------------------------------------------
```

4.11 есеп. Он үштен артық натурал сан (n>13) берілген, n-нен кіші барлық «егіз» сандарды анықтап, экранға шығару керек («егіздер» – айырмалары 2-ге тең жай сандар).

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
void main ()
{
 int n,a,k,simple(int);
 clrscr();
m1:
 printf("\nНатурал сан енгізіңіз N>13: ");
 scanf("%d", &n);
 printf("%d санынан кіші \"егіздер\" жұбы: ",n);
 for(k=0,a=3; a<n-2; a+=2) // жай сандар ішінде
                           // жұп сандар жоқ
 {
   if(simple(a) && simple(a+2))
   {                       // "егіздер"айырмасы 2-ге тең
     if(k%8==0) putchar('\n'); // 1 жолда 8 жұп
        k++;
     printf("%d-%d ", a, a+2);
   }
 }
 printf("\n\n Жалғастырамыз ба? Иә болса,7 енгізіңіз: ");
 scanf ("%d" , &n);
 if(n==7) goto m1;
}
```

```
int simple(int n)
{
 int i,im=sqrt(n);
 for(i=2; i<=im; i++)      // санның мүмкін болатын
                           // бөлгіштерін іздеу
  if (n%i==0) return 0;    // бөлгіш табылса -
                           // ол жай сан емес
 return 1;                 // бөлгіштер табылмады
}
```

Программа жұмысының нәтижесі:

```
Натурал сан енгізіңіз N>13: 50
50 санынан кіші "егіздер" жұбы:
3-5 5-7 11-13 17-19 29-31 41-43
Жалғастырамыз ба? Иә болса,7 енгізіңіз: 0
```

Бақылау сұрақтары

1. Қолданушы (тұтынушы) функциясын сипаттау дегеніміз не?
2. Егер функцияның параметрлері жоқ болса, ол қалай сипатталады?
3. Тұтынушы функциясы мен стандартты функциялардың айырмашылығы неде ?
4. Функция аргументтерін беру жолдарын атаңыз?
5. Нақты және формальды (көрсетпелі) параметрлер түсінігі.
6. “Жергілікті” және “ауқымды” айнымалылардың айырмашылығы.
7. “Жергілікті” айнымалылардың әрекет ету аймағы ұғымы.
8. “Ауқымды” айнымалылардың әрекет ету аймағын түсіндіріңіз.
9. Return операторының қолданылуы және аткаратын қызметі.
10. Функция прототиптері түсінігі

Тапсырмалар

1. Үш натурал сандар берілген. Олардың ең үлкен ортақ бөлгішін (ЕҮОБ) анықтайтын функцияны құру керек.
2. *a,b,c,d* кесінділері берілген. Осы кесінділердің кез келген үшеуін қарастыра отырып, олардан үшбұрыш тұрғызуға болатын/болмайтынын анықтап, үшбұрыш тұргызылатын жағдайда, оның ауданын табу керек. Үшбұрыштың ауданы $S=\sqrt{p*(p-x)(p-y)(p-z)}$, мұндағы $p=(x+y+z)/2$.
3. Натурал N саны берілген. Ол екі x және y бүтін сандарының квадраттарының қосындысына тең болатын болса $N=x^2+y^2$, онда x, y сандарын анықтайтын функцияны құру керек.
4. Нақты x, y ($x>0$, $y>1$) сандары берілген, $y^{k-1} \le x < y^k$ шартын қанағаттандыратындай k бүтін санын (оң, теріс немесе нөлге тең) табу керек.
5. Натурал N саны берілген ($N > 99$). Ондағы жүздіктер санын анықтаңыз.
6. Натурал N саны берілген ($N \le 99$). N^2 саны N санының цифрларының қосындысының кубына тең екендігін тексеру керек.

7. Натурал N саны берілген ($N > 10000$). N санының алғашқы K разрядтарының қосындысын анықтаңыз ($K \le 4$).

8. Натурал n, m сандары берілген, n санының соңғы m цифрларының көбейтіндісін табу керек.

9. Натурал N саны берілген. N санының жазбасынан басқа цифрлардың ретін өзгертпей 0 және 5 цифрларын алып тастау керек. Мысалы, 59015509 санынан 919 саны шығады.

10. Натурал N және K (K<N/2) сандары берілген. K^2-қа бөлінетін және K^3-қа бөлінбейтін барлық натурал K-ларды табу керек.

11. N саннан тұратын бүтін сандар тізбегі берілген. Олардың арасында бірдей екі сан кездеспейді. Қалған сандар өсу ретімен орналасатындай етіп, сол реттілікке көнбейтін сандарды өшіріп тастау керек.

12. Егер N цифрдан тұратын натурал саннның цифрларының қосындысын n-ші дәрежеге шығарғанда, сол саннның өзіне тең болатын болса, ондай сан Армстронг саны деп аталады (мысалы, $153=1^3+5^3+3^3$). Екі, үш және төрт цифрдан тұратын барлық Армстронг сандарын табу керек.

13. 1-ден n-ға дейін нөмірленген (n=10) n елді мекен бар. Кейбір қос елді мекендер жолдармен қосылған. Осы жолдар арқылы бірінші елді мекеннен n-шісіне жетуге болатынын немесе болмайтынын анықтау керек. Жолдар туралы ақпарат i-ші және j-ші елді мекендердің жолдармен қосылғанын білдіретін i және j (i<j) қос сандарының тізбегі түрінде берілген. Ол тізбектің аяқталғанының белгісі – қос нөл (00).

14. Екі үшбұрыштың төбелерінің координаталары берілген. Олардын қайсысының ауданы үлкен екенін анықтау керек.

15. Жазықтықтағы үш түзу $a_kx+b_ky=c_k$ (k = 1,2,3) теңдеулерімен берілген. Егер ол түзулер қос-қостан қиылысып, үшбұрыш құрайтын болса, сол үшбұрыштың ауданын табу керек.

16. Екі жай саннын бір-бірінен айырмашылығы 2-ге тең болса, олар **"егіздер"** деп аталады (мысалы, 41 және 43 сандары). *[*n, *2*n*]* аралығындағы барлық "егіздерді" анықтау керек, мұндағы n – 2-ден үлкен бүтін сан.

17. Натурал N санын енгізіңіз. Келесі алгоритм бойынша Паскаль үшбұрышын құру керек:

$$C_0^0$$
$$C_1^0\ C_1^1$$
$$C_2^0\ C_2^1\ C_2^2$$
$$C_3^0\ C_3^1\ C_3^2\ C_3^3$$
$$C_4^0\ C_4^1\ C_4^2\ C_4^3\ C_4^4$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

18. $C_n^m = \frac{n!}{m!(n-m)!}$ анықтау керек, мұндағы $n!=1\cdot2\cdot3\cdot\ldots\cdot n$, n – санының факториалы.

19. **"8 ферзі туралы есеп".** Шахмат тақтасында 8 ферзіні олар бір-бірін "жемейтіндей" етіп орналастыру керек (барлығы 92 орналастыру бар).

20. **"фон Нейман әдісі".** Нақты N саны берілген. Оларды кемімейтіндей етіп төмендегідей түрде реттеу керек: А және В екі жиымын құрып, бастапқы сандарды А-ға жазу керек; көрші екі санды реттеп (A_1 және A_2, A_3 және A_4 және т.с.с.) оларды В-ға жазу керек; В-дан екі-екіден реттелген көрші сандарды алып, оларды реттелген төрттіктерге жинақтап, қайтадан А-ға жазу керек; одан кейін әрбір көрші реттелген төрттіктерді В-дан алып, оларды реттелген сегіздіктерге жинақтап А-ға көшіру керек, т.с.с.

21. Әрқайсысы 100 элементтен тұратын X және Y жиымдары берілген. Осы екі жиымның әрбір элементін бір-бірімен орын алмастыруды $(x_k \leftrightarrow y_k)$ функция түрінде ұйымдастыратын программа құру керек, мұндағы k=1,2,...,100. Аралық мәндерді сақтайтын қосымша айнымалыларды қолдануға болмайды.

22. 1,2,3,4,5,6,7,8,9 цифрлары жазылған 9 кәртішке бар (әрбіреуіне бір цифрдан жазылған). Осы кәртішкелерден екі сан құрылған. Бірінші сан – бөлшектің алымын, ал екіншісі – бөлшектің бөлімін құрайды. Әрбір кәртішке міндетті түрде пайдаланылуы тиіс. Бөлшектің бөлімі 5 цифрдан тұрады, олай болса бөлшекті алымын қалған төрт цифр құрайды. Бөлшектің мәні дәл 1/2-ге тең болатындай барлық комбинацияларды табыңыз. Мысалы, бөлшектің алымы – 6729, бөлімі – 13458 (барлық кәртішкелер пайдаланылған), 6729/13458 =1/2.

23. 1,2,3,4,5,6,7,8,9 цифрлары жазылған 9 кәртішке бар (әрбіреуіне бір цифрдан жазылған). Екі натурал N және M сандарын енгізіңіз. Мысалы N=2, M=3. N/M бөлшегін (2/3) дәл құрайтын кәртішкелердің комбинацияларын құру керек, мысалы, 4/6, 12/36, 123/369, т.с.с. (сондайлар бар болатын болса). Егер ондай комбинация саны бірден көп болса, олардың бәрін шығару қажет.

24. Цилиндрдің бүйір беті ауданын есептейтін функцияға оның аргументтерін беруді көрсететін программаны келесі формула бойынша анықтау керек:

$$S_6 = 2\pi RH.$$

25. **Main** функциясынан **kvadrat** функциясына шаршы (квадрат) қабырғасының ұзындығы – **side** параметрі берілетін программа жазыңыз. Функция сол квадратты экраннын сол жақ жоғарғы бұрышында жұлдызшалар арқылы бейнелеуі керек. Мысалы, егер **side** = 4 болса, программаның нәтижесі төмендегідей болуы тиіс:

```
* * * *
* * * *
* * * *
* * * *
```

26. төмендегі есепті шығару үшін функция құрып, оны тексеріп шығу керек

$$f(x,n,m) = \sum_{i=m}^{n} \frac{x^{2i}}{(2i)!}$$

27. Мына есептің мәнін табу керек

$$z = \frac{v_1 + v_2 + v_3}{3}$$

мұнда v_i - r_i радиусымен берілген шардың көлемі (функцияны шардың көлемін табу үшін пайдалану).

28. Өзінің төбелерінің координаталарымен берілген үшбұрыштың ауданын мына формуламен табу қажет

$$S = 0,5 * |x_1y_2 + x_2y_3 + x_3y_1 - x_1y_3 - x_2y_1 - x_3y_2|.$$

Функцияны үшбұрыштың ауданын табу үшін қолдана отырып өзінің төбелерінің координаталарымен берілген ABCD төртбұрышының ауданын табу керек.

29. Он алтылық санау жүйесіндегі тек цифрлардан тұратын санды оған эквивалентті бүтін ондық санға түрлендіретін функцияны жазып тексеріп шығыңыздар.

30. k натурал санымен берілген $\sqrt[k]{x}$ $(x > 0)$ өрнегінің мәнін табу үшін функция жазу керек. Ол үшін келесі итерациялық формула қолданылады:

$$y_0 = x; \quad y_{i+1} = y_i + \frac{1}{k}(\frac{x}{y_i^{k-1}} - y_i).$$

Есептің шешімін табу үшін $| y_{i+1} - y_i | < \varepsilon$ түрінде алынатын y_{i+1} мәнін пайдаланыңыздар, мұндағы есептеу дәлдігі $0 < \varepsilon < 10^{-6}$ аралығында болады.

4 зертханалық жұмыс.

Қолданушы анықтаған функцияларды пайдалану

Тақырыбы: Қолданушы функцияларын пайдалану.

Тапсырма: Берілген есептеулерді орындайтын программа жазу.

Есеп беру: Программа мәтіні мен бірнеше бастапқы мәндерге сәйкес нәтижелер алу.

Жұмысқа арналған тапсырмалар варианттары

1 нұсқа

1. *n* натурал саны берілген. Экранға сол *n* саны цифрларының кері бағытта

жазылған нұсқасын шығару керек. Жаңа санды табу үшін функция жазу қажет.

2. Цилиндр көлемін есептейтін Volume атты функция қолданылатын программа жазыңыздар. Функция параметрлері ретінде цилиндрдің радиусы мен биіктігі алынады.

2 нұсқа

1. Мүмкін болатын алмастырулар санын төмендегі формула бойынша есептейтін программа құрыңыздар:

$$C_n^k = \frac{n!}{k!(n-k)!}.$$

Факториал есептеуді функция түрінде жазу керек.

2. Берілген санның белгілі бір пайызын (процентін) қайтаратын **Payiz** функциясын пайдаланатын программа жазыңыздар. Функция аргументі ретінде берілген сан мен оның қайтарылатын пайыз мөлшері беріледі.

3 нұсқа

1. Екілік санды соған парапар (эквивалентті) ондық санға түрлендіретін программа жазу керек. Түрлендіру әрекеті функция түрінде жазылуы тиіс.
2. Салынған ақшаның пайыздық өсімін (процентін) есептейтін **Osim** функциясын пайдаланатын программа жазыңыздар. Функция үшін бастапқы мәлімет ретінде салынған ақша мөлшері, өсім пайызы (бір жылдық) және салынатын мерзімі (күндер саны) көрсетіледі.

4 нұсқа

1. Он алтылық санды соған парапар (эквивалентті) ондық бүтін санға түрлендіретін программа жазу керек. Түрлендіру әрекеті функция түрінде жазылуы тиіс.
2. Аргумент ретінде дауысты әріп (символ) берілсе – 1-ді, ал дауыссыз әріп берілсе –
0-ді қайтаратын Dauys функциясын пайдаланатын программа жазыңыздар.

5 нұсқа

1. Екілік санды соған парапар (эквивалентті) он алтылық санға түрлендіретін программа жазу керек. Түрлендіру әрекеті функция түрінде жазылуы тиіс.
2. Латынның кіші әріптерін бас әріптерге және керісінше түрлендіретін **Turlen** функциясын пайдаланып, сөз тіркестерін өңдейтін программа жазу керек.

6 нұсқа

1. Он алтылық санды соған парапар (эквивалентті) екілік бүтін санға түрлендіретін программа жазу керек. Түрлендіру әрекеті функция түрінде жазылуы тиіс.
2. Экранға бірдей символдардан тұратын сөз тіркесін шығаратын **Output** функциясы бар программа жазыңыздар. Функция үшін параметр ретінде символ таңбасы мен тіркес ұзындығы көрсетіледі.

7 нұсқа

1. Келесі өрнекті есептейтін программа жазыңыздар: $f(x,n,m)=\sum_{i=m}^{n}\frac{x^{2i}}{(2i)!}$

   Факториалды есептеу функция түрінде берілуі тиіс.
2. Параллелепипед көлемі мен толық беті ауданын **Volume** және **Space** функциялары арқылы есептейтін программа құрыңыздар.

8 нұсқа

1. Берілген натурал сан бойынша ондағы цифрлар саны мен қосындысын табатын программа құрыңыздар.Сандағы цифрлар саны мен қосындысын функция ретінде жазыңыздар.
2. Экранга квадрат түріндегі төртбұрыш шығару үшін **Sharshy** функциясын колданатын программа құрыңыз.Функцияның параметрлері ретінде сол жак жоғарғы бұрыштың координаттары, квадрат кабырғасы мен сол кабырғаны салатын символ бейнесі берілуі тиіс.

9 нұсқа

1. Берілген екі натурал сандардың ең үлкен ортақ бөлгішін (ҮОБ) есептейтін программа жазыңыз.Мектеп программасынан белгілі Евклид алгоритмін колданып ҮОБ -тың табылу жолын функция ретінде жазыңыздар.
2. Пернетактадан енгізілген символдарды экранда көрсетпейтін **San1** функциясы программасын жазыныздар. Enter пернесін басканда, экранға сол сиволдар ішіндегі тек 1-ден 9-ға дейінгі цифрлар ғана шығуы керек.

10 нұсқа

1. Берілген натурал сан бойынша оның бастапкы және соңғы цифрын табатын программа құрыңыз. Санды анықтау мен оны экранға шығаруды функция ретінде жазу керек.
2. **Perim** функциясы бар программа жазыныз. Бұл функцияға программадан *A(x1, y1), B(x2, y2),* C(*x*3, *y*3) үшбұрыш төбелерінің координаттары беріледі, ал ол үшбұрыштың периметрін есептеуі тиіс. Егер ондай үшбұрыш тұрғызуға болмаса, онда соған сәйкес хабарлама шығару керек.

11 нұсқа

1. *n* және *m* натурал сандары берілген. *n* санының *k* кіші разрядтарынан тұратын сан мен *m* санының *k* үлкен разрядтарынан тұратын екі саннын қосындысын есептейтін программа жазыңыз.
2. *(x, y)* нүктелерінің координаталарын программадан ала отырып, сол нүктенің центрі (0,0) нүктесінде, радиусы бірге тең шенбер ішінде жататынын/жатпайтынын анықтайтын **Orin** функциясын колданатын программа құрыныз. Жұмыс нәтижесі мәтіндік түрде экранға шығарылуы тиіс.

12 нұсқа

1. Келесі өрнекті *n* мәні 12-ден 24-ке дейін қадамы 4 болып өзгерген кезде, есептейтін

программа жазыңыздар. Факториалды есептеуді функция ретінде беру керек.

$$S(n) = \sum_{i=m}^{n} \frac{(2i)!}{(n+i)!}$$

3. F функциясының мәндерін оның аргументі x_0 мен x_n аралығында dx қадамымен өзгерген кезде анықтайтын **Esepteu1** функциясын қолданатын келесі өрнек үшін программа құрыңыздар, x, a, b, c мәндері программадан функцияға беріледі.

$$F = \begin{cases} \dfrac{1}{ax} - b, & \text{егер } x + 5 < 0 \text{ және } c = 0 \\ \dfrac{x - a}{x} - b, & \text{егер } x + 5 > 0 \text{ және } c \neq 0 \\ \dfrac{10x}{c - 4}, & \text{қалған жағдайларда} \end{cases}$$

13 нұсқа

1. n және m натурал сандары берілген. n санының k кіші разрядтарынан және m санының k кіші разрядтарынан тұратын екі саннның қосындысын есептейтін программа жазыңыз.
2. **Dauis** функциясы бар программа жазыңыз. Бұлфункция, егер аргумент ретінде оған берілген символ ағылшын алфавитінің дауысты дыбысы болса, 1 деген мәнді, кері жағдайда 0 мәнін қайтаруы тиіс.

5 ЖИЫМДАР. БІРӨЛШЕМДІ ЖИЫМДАРДЫ ӨҢДЕУ

Жиым (немесе массив) деп ат қойылған біртекті шамалар тізбегін айтады.

Жиым элементін немесе компьютер жадының бір ұясын көрсету үшін жиымның атын және оның индексін көрсету қажет. Индекс бүтін сан немесе бүтін өрнек болуы мүмкін. Жиым элементтері мәнін оларды хабарлау кезінде меншіктеу арқылы немесе енгізу жолымен беруге болады. Егер алғашқы инициалдау кезінде жиым элементтерінің көрсетілген мәндері жалпы элементтер санынан аз болса, онда қалған жиым элементтері нөлдік мән қабылдайды.

Мысалы, мынадай тізбек

0 1 1 2 3 5 8 13 21

Фибоначчи тізбегінің 9 элементін құрайды (алғашқы екі санды таңдап алып, келесі санды алдыңғы екеуін қосу жолымен алады). Ал мынау өзіне және бірге бөлінетін жай сандар тізбегінің алғашқы 7 элементі:

1 3 5 7 11 13 17

Осындай бір текті тізбектерді жиым түрінде C/C++ тілінде сипаттап, оған бастапқы мән беріп инициалдау үшін былай жазамыз:

```
int fib[8]={0, 1, 1, 2, 3, 5, 8, 13, 21};
```
немесе
```
int fib[]={0, 1, 1, 2, 3, 5, 8, 13, 21};
```
деп көрсетеміз, мұндағы `fib` – жиым аты, оның элементтерінің типі `int`, ал ені, яғни ұзындығы – 9, жиым элементтерінің индекстері 0-ден бастап нөмірленеді, сол себепті 9 элемент 8 индекспен көрсетіледі. Мәндері толық көрсетілсе, индексті жазбаса да болады. Ал былай болса,

```
int fib[8]={0, 1, 2, 3};
```
қалған элементтері 0 болып саналады.
```
n=10; k=2; fib[n-k] = {0, 1, 2, 3};
```
десе де болады.

Жоғарыдағы тізбектің 7-ші элементін бір бүтін айнымалыға меншіктеу үшін былай жазамыз.

```
int a = fib[6];      // a = 8
```

Жиымды сипаттау кезінде оның ені нақты санмен көрсетіледі, мысалы, `a[20]`, `a[n]` деп жазу үшін алдын ала `#define n 20` жолы көрсетіледі немесе `const n=20;` болып жазылады.

Жиым элементтерін енгізу немесе оларды түрлендіру үшін цикл операторлары қолданылады. Төменде 10 элементі бар жиымды 0-ден 9-ға дейінгі сандармен толтырып, содан кейін оларды кері бағытта экранға шығару мысалы көрсетілген:

```
main()
{  int ia[10];
   int index;
  for(index = 0; index <10; index ++)
     ia[index] = index;
  for(index = 9; index >=0; index --)
    printf(" %i", ia[index]); }
```

C тілінде жиымды жиымға бірден теңестіруге болмайды, мысалы, $a_0, a_1, a_2, ..., a_9$ және $c_0, c_1, c_2, ..., c_9$ жиымдары үшін `a=c` деп жазуға

рұқсат етілмейді. Олардың элементтерін цикл ішінде бір-біріне біртіндеп теңестіру керек.

Мысалы, мынадай цикл жазылуы тиіс:

```
int a[9], c[9];
for(int i=0; i<9; ++i)
    a[i]=c[i];
```

Жиым элементтері бастапқы нөлдік мәнді автоматты түрде ала алмайды, сондықтан программалаушылар жиым элементерінің барлығын автоматты түрде нөлге айналдыру үшін, кем дегенде оның бірінші элементіне нөлді меншіктеуі керек.

Жиым элементерін нөмірлеу нөлден басталады:

```
intc[12]; //c[0]–c[11] жиымын хабарлау
```

Жиым элементінің позициялық (реттік) нөмірі – индексі – квадрат (тік) жақша ішіне жазылады.

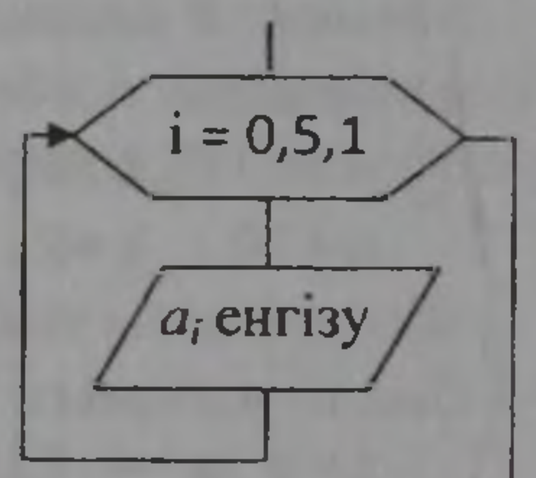
5.1 Жиымды мәндермен толтыру

Программада сипатталған жиымдармен жұмыс істеу алдында оларға мәндер берілуі тиіс.

Жиымдар (массивтер) элементтерінің мәндерін программаға енгізу тәсілдері:

- пернетақтадан мән енгізу;
- кездейсоқ сандар арқылы мәндер беру;
- берілген мәндерді меншіктеу аркылы енгізу;
- элементтер мәндерін файлдан енгізу.

Жиымдар мәндері бірнешеу болатындықтан, олар цикл аркылы жүзеге асырылады. Ең ыңғайлысы for циклі болып табылады. Екі өлшемді жиымдар үшін қабаттасқан циклдер қолданылады.



5.1-сурет. 5 элементті пернелерден енгізу

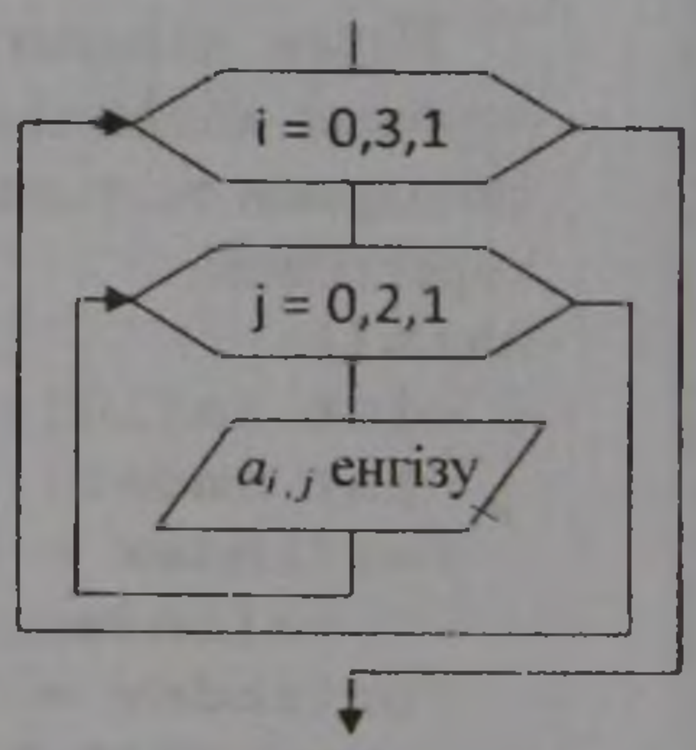
Пернетактадан енгізу мысалы:

- 5 элементтен тұратын жиым үшін (5.1-сурет):

```
for(i=0; i<=5; i++)
    cin >> a[i];
```

- көлемі 3x2 элементтен тұратын матрица түріндегі жиым үшін (5.2-сурет):

```
for(i=0; i<3; i++)
  for(j=0; j<2; j++)
     cin >> a[i][j];
```



5.2-сурет. Матрицаны пернелерден енгізу

N элементтен тұратын бір өлшемді жиымды 0-ден 99-ға дейінгі кездейсоқ сандармен толтыру мысалы:

```
for(i=0; i<n; i++)
  x[i]=random(100);
```

Көбінесе жиым элементтеріне программа басында нөл мәнін беру керек болып жатады, n жолдан және m бағанадан тұратын екі өлшемді жиымды 0-дермен толтыру мысалы (5.3-сурет):

```
for(i=0; i<n; i++)
  for(j=0; i<m; j++)
    a[i][j]=0;
```

5.2 Жиым мәндерін экранға (файлға) шығару

Жиым элементтері мәндерін экранға шығару үшін де циклдер қолданылады. 5 элементтен тұратын жиым мәндерін экранға шығару мысалы:

- бағаналар бойынша:

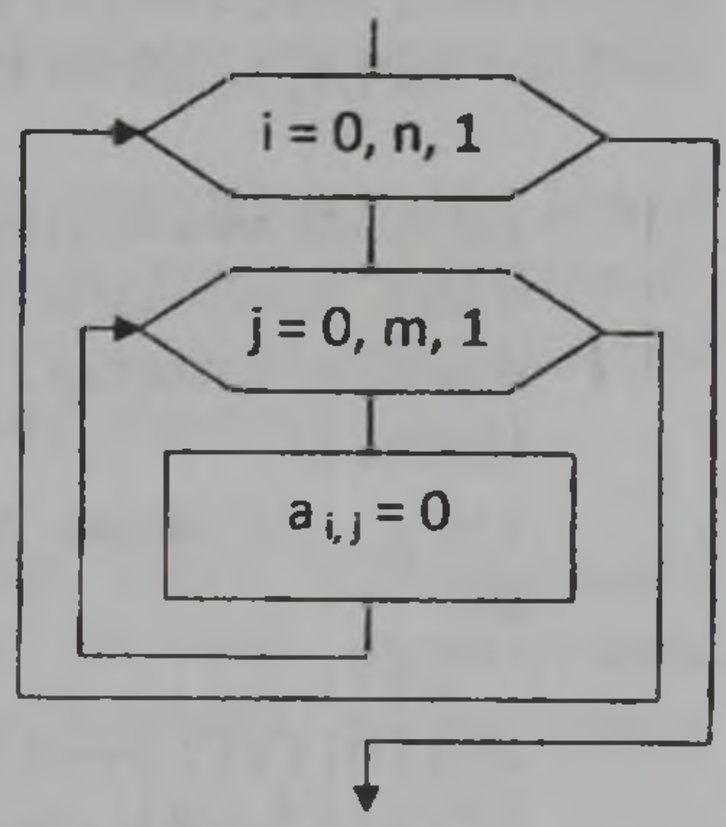
```
for(i=0; i<=5; i++)
   cout >> a[i] >> endl;
```

- араларын бос орындармен бөле отырып, жолдар бойынша:

```
for(i=0; i<=5; i++)
   cout >> a[i] >> " ";
```

- формат арқылы әрбір элементке 4 орын бере отырып, жолдар бойынша:

```
for(i=0; i<=5; i++)
   cout >> setw(4) >> a[i];
```



5.3-сурет. Матрицаны нөлдермен толтыру

Матрицаны стандартты формада – жолдар мен бағаналар бойынша – cout операторы арқылы шығару, бір жолды шығарған соң, келесі жолға көшу үшін, endl немесе "\n" тіркестерін пайдалану арқылы іске асыру:

```
for(i=0; i<n; i++)
  { for(j=0; j<m; j++)
      cout >> setw(4) >> a[i][j];
    cout >> endl;
  }
```

5.3 Жиымдарды өңдеу

Көбінесе жиым элементтерінің қосындысын, арифметикалық ортасын, ең үлкен немесе ең кіші мәндерін, олардың индекстерін анықтау және де жиым элементтерінің мәндерін өзгерту сиякты есептер жиі кездеседі. Мұндайда бір өлшемді және екі өлшемді жиымдар үшін ұқсас алгоритмдер қолданылады, тек екі өлшемді жиымдарда қабаттасқан циклдер қолданылады.

Бір өлшемді жиымдармен жұмыс істеу

Жиымның n элементі бар болсын делік, енді төмендегі әрекеттерді орындау есептерін карастырайық.

- *Элементтер қосындысын табу*:

```
s=0;
for(i=0;i<n;i++)
   s+=a[i]; // s айнымалысында қосындыны жинақтау
```

- *Элементтер көбейтіндісін табу*:

```
s=1;
for(i=0;i<n;i++)
   s*=a[i]; // s айнымалысында көбейтіндіні жинақтау
```

- *Белгілі бір шартты қанағаттандыратын элементтер санын анықтау*. Мысалы, бүтін сандар жиымындағы жұп сандардың қанша екенін анықтау мысалы:

```
k=0;
for(i=0;i<n;i++)
   if(a[i]%2 == 0) k++;  //  егер сан 2-ге қалдықсыз бөлінсе,
                         //  санауышты 1-ге арттыру
```

- *Берілген мәнге тең жиым элементін анықтау*. Элементті анықтау дегеніміз – оның жиымдағы нөмірін анықтау. Жиымның нөлге тең алғашқы элементінің нөмірін анықтау, егер ондай элемент болмаса, соған сәйкес мәлімет беру мысалы:

```
i=0;  // жиым элементінің нөмірі
do
  i++;
while((a[i]==0) || (i==n));
if(a[i]==0) cout << "Алғашқы нөлдік элемент нөмірі=" << i;
  else cout << "Ондай элементтер жоқ";
```

Егер нөлге тең элементтердің барлығының да нөмірін анықтау керек болса, онда for циклін пайдаланып, мынадай түрде программа құруға болады:

```
cout << "Нөлге тең элементтер нөмірлері:"
for(i=0;i<n;i++)
  if(a[i]==0) cout << i << " "; // нөмірлерді шығару
```

- *Жиымның ең үлкен (максимал) элементі мен оның нөмірін анықтау*. Max айнымалысы максимал элементтің мәнін, ал k – оның индексі нөмірін сақтайды. Мысалы:

```
max=a[0]; k=0;    // іздеуді алғашқы элементтен бастаймыз
for(i=1;i<n;i++) // келесіден соңына дейін қарап шығу
  if(a[i]>max)    // алдыңғысынан үлкен элементтің
    { max=a[i];  // мәнін және нөмірін (индексін)
      k=i;       // есте сақтаймыз
    }
```

Осы тәрізді a[i]<min болған кезде ең кіші (минимал) элемент анықталады.

- *Жиым элементтерінің мәндерін өзгерту*. Мысалы, a_n жиымы жұмысшылар жалақыларын сақтайды делік. Жалақысы ең кіші айлық мөлшерінен (minzp) төмен жұмыскерлерге осы minzp мөлшерінде жалақы тағайындау керек.

```
minzp=40000;
for(i=0;i<n;i++)
  if(a[i]<minzp)
     a[i]=minzp;
```

5.3.1 Жиым элементтерін сұрыптау

Мәліметті іздеу мен сұрыптау информатиканың ең маңызды түсініктерінің бірі болып табылады. Сұрыптау (реттеу) – бір типтегі мәліметтер жиынтығын сондағы сандардың өсуі немесе кемуі бойынша реттеп орналастыру. Программалау тұрғысынан алғанда, маңызды болып жиымды сұрыптау, жолдарды сұрыптау, файл элементтерін сұрыптау әрекеттері есептеледі. Осы тәрізді сұрыптаулар түрлері компиляторларда, интерпретаторларда, мәліметтер базасында, статистикалық мәліметтерді, анықтамалық материалдарды қарастыруда жиі қолданылады.

Сұрыптау кезінде жиым элементтері олардың мәндерінің өсуі не кемуі бойынша реттеліп орналасатындай түрде алмастырылып, индекстері өзгертіледі. Көбінесе бір өлшемді жиымды сұрыптау туралы айтылады, мысалы, оқушыларды бойлары бойынша сапқа тұрғызу кезіндегі сиякты реттеулер қарастырылады.

Төмендегідей ұғымдарды есте сақтаған жөн:

- *жиымдарды сұрыптау* – ақпаратты сақтау және іздеу жүйелеріндегі маңызды жұмыстардың бірі болып саналады, өйткені сұрыпталған мәліметтер ішінен керектісін тауып алу, сұрыпталмаған мәліметтерге қарағанда, өте жылдам орындалады;
- бір-бірінен *жылдамдығы* бойынша әртүрлі болып келетін сұрыптау алгоритмдері бар;
- жиымдарды *«жылдам» сұрыптау* тәсілдері мыңдаған элементтерден тұратын өте көлемді мәліметтер үшін аса ыңғайлы болып келеді де, олар уақыт үнемдеуге келгенде көптеген артықшылықтар береді, бірақ шағын көлемдегі жиымдар үшін қарапайым сұрыптау жолдарын пайдалана беруге болады.

Қалыптағы сұрыптау есептерінде – берілген жиым элементтерін экранға шығарып, сонан соң оларды өсуі (кемуі) бойынша сұрыптап алып, реттелген мәндерді экранға қайта шығару керек.

8 бүтін саннан тұратын m жиымы үшін оның элементтерін кемуі бойынша реттеп орналастыратын 3 сұрыптау тәсілін қарастырайық. Олардың тиімділігін салыстыру үшін сұрыптау кезіндегі *итерациялар* (жиымды қарастыру, қайталап көрсету қадамдары саны) санын көрсететін a айнымалысын енгіземіз. Жиымның әр уақыт кезеңіндегі қалпын бақылау мақсатында әрбір алмастыру әрекетінен кейін жиым элементтерін экранға шығарып отыру керек. Программаны түзету ыңғайлы болуы үшін жиым элементтерін типі берілген (типтелген) константа ретінде қарастырамыз.

Сызықтық сұрыптау (таңдау арқылы сұрыптау)

Мәндердің кемуі бойынша сызықтық сұрыптау тәсілінде жиым элементтері толық қарастырылып, ең үлкен мәні анықталады да, оны алғашқы элементпен алмастырамыз. Содан кейін екінші элементтен бастап, қалған элементтер толық қарастырылып, олардың да үлкені екінші элементпен орнын алмастырады, т.с.с. болып кете береді.

Мәндерді кемуі бойынша реттейтін сызықтық сұрыптау тәсілінің программасы:

```
/* Жиымды кемуі бойынша сызықты түрде сұрыптау */
#include <iostream.h>
#include <iomanip.h>
#include <conio.h>
main()
{ const n=4; // элементтер саны
  int m[n]={11,3,19,9},
      i,j,buf,l,a;
clrscr();
cout << "Array elements:\n";
for(i=0;i<n;i++)
  cout << setw(4) << m[i];
cout << endl;
a=0; // итерациялар санауышын нөлге теңеу
     // сұрыпталмаған жиым бөлігінің мөлшерін өзгерту
for(i=0;i<n-1;i++)
   // сұрыпталмаған жиым бөлігінің аяғына дейін оның
   // i-элементін i+1-элементімен салыстыру
  for(j=i+1;j<n;j++)
     { a++;
       if(m[i]<m[j])
         {buf=m[i]; m[i]=m[j]; m[j]=buf;}
       for(l=0;l<n;l++)
          cout << setw(4) << m[l]; // сұрыптаудан кейін
       cout << "; iteration # " << a << endl;
     }
   getch();
}
```

Программа жұмысы нәтижесі:

```
Array elements:
  11   3  19   9
  11   3  19   9; iteration # 1
  19   3  11   9; iteration # 2
  19   3  11   9; iteration # 3
  19  11   3   9; iteration # 4
  19  11   3   9; iteration # 5
  19  11   9   3; iteration # 6
```

«Көпіршік» тәсілімен сұрыптау

Ең кең тараған сұрыптау тәсіліне жататын «көпіршік» тәсілінде – алгоритмнің орындалуы барысында – жиымның «жеңілірек» элементтері біртіндеп «қалқып»

жоғары шығады. Бұл тәсілдің ерекшелігі – қатар тұрған екі элемент салыстырылып, қажет болса, олар орын алмастырады.

«Көпіршік» тәсілі арқылы сұрыптау программасы:

```
/*  "Көпіршік" тәсілімен сұрыптау */
#include <iostream.h>
#include <iomanip.h>
#include <conio.h>
main()
{
  const n=4; // элементтер саны
int  m[n]={11,3,19,9},i,j,buf,k,a;
clrscr();
cout << "Берілген жиым:\n";
for(i=0;i<n;i++)
  cout << setw(4) << m[i];
cout << endl << endl;
a=1; // итерациялар санауышы
for(i=1;i<n;i++)
 {
   for(j=n-1;j>=i;j--)
   {if(m[j-1]<m[j]) /* егер оң жақтағы элемент
     сол жақтағы элементтен артық болса,оны сол
     жаққа жылжытамыз - көпіршік «жоғары» кетеді */
       { buf=m[j-1]; m[j-1]=m[j]; m[j]=buf;
         for(k=0;k<n;k++)
           cout << setw(4) << m[k]; //сұрыптау итерациясынан соң
         cout << "; итерация # " << a++ << endl;
       }
   }
 }
getch();
}
```

Программа жұмысының нәтижесі:

```
Берілген жиым:
  11   3  19   9

  11  19   3   9; итерация # 1
  19  11   3   9; итерация # 2
  19  11   9   3; итерация # 3
```

Бөле отырып жылдам сұрыптау

Жоғарыдағы екі тәсіл қарапайым, әрі көрнекі, бірақ олар онша тиімді емес. Олардан жылдам істейтін К.Хоор алгоритмі *бөле отырып сұрыптау* немесе

«*жылдам сұрыптау*» деген атпен кең таралған. Бұл алгоритм негізіне жиымды бірнеше бөліктерге бөліп, солардың арасында мәлімет алмасу тәсілі жатыр. Алгоритм күрделі болып саналады, ол рекурсияны және функцияны пайдалануды керек етеді, сондықтан оны карапайым түрде аздаған комментарилермен келтіреміз.

Жылдам сұрыптау тәсілінің программасы:

```
/*  Хоор тәсілімен жылдам сұрыптау  */
#include <iostream.h>
#include <iomanip.h>
#include <conio.h>
 const n=4; // элементтер саны
int   m[n]={11,3,19,9},i,a;
void quicks(int first,int last)
{ int i,j,x,buf,l;
  i=first;
  j=last;
  x=m[(first+last)/2]; // жиым ортасы
  do
  { while (m[i]>x) i++;
    while (m[j]<x) j--;    //
    a++;
    if(i<=j)
    { buf=m[i]; m[i]=m[j]; m[j]=buf;
      i++;j--;
      for(l=0;l<n;l++)
         cout << setw(4) << m[l];
      cout << "; iteration # " << a << endl;
    }
  }
  while(i<=j);  //
  if(first<j)  quicks(first,j);
  if(i<last)   quicks(i,last);
 }
main()
{
clrscr();
cout << "Array elements:\n";
for(i=0;i<n;i++)
  cout << setw(4) << m[i];
cout << endl << endl;
a=0;
quicks(0,n);
cout << endl << "The sorted array:\n";
for(i=0;i<n;i++)
```

```
    cout << setw(4) << m[i];
  cout << endl;
  getch();
}
```

Программа жұмысы нәтижесі:

```
Array elements:
  11   3  19   9

  19   3  11   9; iteration # 1
  19  11   3   9; iteration # 3
  19  11   9   3; iteration # 4
  19  11   9   3; iteration # 5

The sorted array:
```

5.3.2 Реттелген жиымдардағы мәліметтерді жылдам іздеп табу

Реттелмеген жиымдардан мәлімет іздеу кезінде берілген санды тізбекті түрде барлық элементтермен біртіндеп салыстыра отырып, нәтижесі «ақиқат» болғанша, іздей береміз. Кейде ондай сан жиым ішінде болмауы да мүмкін. Егер жиым көлемді (элементтер саны өте көп) болса, іздеу процесі ұзаққа созылып кетеді. Осындай іздеу жұмыстары алдын ала реттелген жиымдарда тез нәтиже береді де, олар жиымдардан мәлімет іздеу операцияларын тиімді түрде жүзеге асыра алады. Сонымен жиымдардағы керекті элементті жылдам іздеп табу үшін, оларды алдын ала өсуі (не кемуі) бойынша сұрыптап алу керек екен.

Сұрыпталған көлемді жиымдардан мәлімет іздеудің ең тиімді жолдарының бірі *бинарлық тәсіл* немесе жиымды *екіге бөлу арқылы* жүргізілетін іздеу тәсілі болып саналады.

Бұл тәсілдің ерекшелігі – жиым элементтерін түгел карастырмай, оны екіге бөледі де, керекті мәнді ортадағы мәнмен салыстырып, жиым сұрыпталғандықтан, оның қай бөлігінде екенін бірден анықтауға болады. Содан кейін керекті элемент орналаскан бөлікті тағы да екіге бөлеміз де, қажетті мәннің қай бөлікте екенін анықтау жолымен карастырылатын бөлік бір элементтен тұратын болганша, іздеуді жалғастыра береміз.

Төменде бүтін сандардан тұратын сұрыпталған жиымнан керекті элементті бинарлық тәсілмен іздеу жүргізетін программа мәтіні берілген.

```
  // бинарлық іздеу
#include <iostream.h>
#include <iomanip.h>
#include <conio.h>
main()
{
 const count=20; // элементтер саны
 int m[count]={20,20,19,19,19,18,17,17,12,12,11,10,9,9,5,5,3,3,2,1},
     n,i,a,first,last,found;
```

```
clrscr();
cout << "Берілген жиым:\n";
for(i=0;i<count;i++)
   cout << setw(4) << m[i];
cout << endl;
cout << "Іздеетін санды енгізіңіз: ";
cin >> n;
a=0; first=0;last=count-1;
found=0;     // әлі элемент табылмаған
do           // іздеуді қайталау
{  i=(first+last)/2; // жиымды 2 бөлікке бөлу
   if(m[i]==n)
     found=1;
   else
     if(m[i]>n)
       first=i+1;  // сан оң жақ бөлікте
     else
       last=i-1;   // сан сол жақ бөлікте
   a++;
} // керек элемент табылса немесе жиым толық қарастырылса, аяқтау
while(!(found || first>last));

 if (found)
    { cout << "Керекті элемент " << n <<" жиымдағы ";
      cout << i << " позицияда орналасқан\n";
    }
  else cout << "Жиымда керекті элемент жоқ..." << n << endl;
  cout << "Іздеу " << a << " итерацияда орындалды\n";
 getch();
}
```

Программа жұмысының нәтижесі:

```
Берілген жиым:
  20 20 19 19 19 18 17 17 12 12 11 10 9 9 5 5 3 3 2
1
Іздеетін санды енгізіңіз: 10
Керекті элемент 10 жиымдағы 11 позицияда орналасқан
Іздеу 3 итерацияда орындалды
```

5.3.3 Жиым элементтерін жою және жиымға жаңа элемент кірістіру

Жиым элементтері бірінен соңы бірі тізбектеліп катар тұрған жады ұяларында орналасады. Сондықтан еш киындықсыз соңғы элементті жоюға (өшіріп тастауға) немесе жиым соңына жаңа элемент қосуға болады. Қалған жағдайларда, бір элемент өшірілсе, одан кейінгілерін алға бір орынға жылжытуға тура келеді. Ал

бір элемент қосу керек болса, оған орын босатып алу үшін, элементтер арасын «ашу» керек. Мысал ретінде жиымнан берілген мәнге тең элементтерді өшіретін программа мәтінін қарастырайық.

```
//Программа берілген мәні бар элементті тауып өшіреді
#include <iostream.h>
#include <iomanip.h>
#include <conio.h>
main()
{const count=10; // элементтер саны
 int m[count]={33,43,2,33,90,78,60,33,33,21},
     n,  // эл-ттер саны айнымалысы
     i,  // санауыш
     j,  // эл-ттерді жылжыту кезіндегі цикл параметрі
     x;  // өшірілетін элемент
clrscr();
cout << "Берілген массив:\n";
for(i=0;i<count;i++)
   cout << setw(4) << m[i];
cout << endl;
cout << "Өшірілетін элементті енгізіңіз: ";
cin >> x;
n=count; i=j=0; // инициалдау
do                  // іздеуді қайталау
 {  while(m[i]==x)
    { j=i;
      do
       { m[j]=m[j+1];
         j++;
       }
      while(j<n);
      n--;  // жиым эл-ттерін азайту
    }
   i++;
 }
while(i<n);
 for(i=n;i<count;i++) m[i]=0; // соңғы жақты нөлге теңеу
 cout << "Алынған жиым:";
 // егер соңғы нөлдерді шығару керек болмаса, циклдегі count
 // айнымалысын n айнымалысына өзгерту керек
 for(i=0;i<count;i++)
    cout << setw(4) << m[i];
 cout << endl;
 getch();
}
```

Программа жұмысы нәтижесі:

```
Берілген массив:
  33  43   2  33  90  78  60  33  33  21
Өшірілетін элементті енгізіңіз: 60
Алынған жиым:  33  43   2  33  90  78  33  33  21   0
```

5.4 Адрестік операциялар

Адрестік операциялар үшін C/C++ тілінде екі арнайы оператор қолданылады.

& – адресті анықтау (алу) үшін қолданылатын операция;

* – адрес аркылы қатынас жасау үшін қолданылатын операция (көрсетілген адрестегі мәнді пайдалану).

& операциясы берілген айнымалының адресін қайтарады. Мысалы, программа мәтінінде p айнымалысы былай сипатталған болсын:

```
int p;
```

онда &p жазуы осы айнымалының компьютер жадында орналаскан ұясының адресі болып табылады.

Нұсқауыштар. Нұсқауыш мәліметтердің адресін сақтайтын айнымалы болып табылады. Жалпы алғанда, нұсқауыш адрестің символдық кескінделуі болып саналады.

Қарастырылатын мысалда &p – p атты айнымалыға сілтейтін нұсқауыш болып табылады. Нақты адрес ретінде белгілі бір сан тұрады, ал &p нұсқауыш типті константа болып табылады. Басқаша айтар болсақ, егер ptr нұсқауыш атауы болса, онда ptr=&p өрнегі p адресін ptr айнымалысына меншіктейді. & – адрес алу операциясы. Ptr көрсетіп тұрған мәнді анықтау үшін келесідей жанамалы адрестеу операциясын пайдалану қажет:

```
V=*ptr;
```

C/C++ тілінде нұсқауыш типті айнымалылар да бар. Нұсқауыш типті айнымалылар мәні болып белгілі бір шаманың адресі саналады.

Мысалы, нұсқауыш типті айнымалы ptr идентификаторы аркылы белгіленген болсын, онда төмендегі оператор p айнымалысының адресін ptr атты нұсқауыш типті айнымалыға меншіктейді. Ptr атты нұсқауыш типті айнымалы баска да объектіге сілтеуі мүмкін.

Мысалы:

```
int *ptr;
ptr=&p; ptr=&max;
```

* операциясы – адрес аркылы қатынас жасау үшін пайдаланылатын операция, ол одан кейін тұрған айнымалының нұсқауыш екенін білдіреді. Мысалы, ptr нұсқауыш типті айнымалысында жоғарыда көрсетілгендей max айнымалысының адресі сақталған болсын. Осы айнымалының мәнін білу үшін * адресі бойынша

катынас жасау операциясын колдануға болады. Ptr нұсқауышы мәнін анықтау үшін келесі операцияны орындау қажет:

```
res=*ptr; /* res айнымалысына ptr айнымалысында сақталған адрестегі
             мәнді меншіктеу */
```

Төмендегі екі меншіктеу операциялары

```
ptr=&x;
y=*ptr;
```

мынадай бір оператор орындайтын

```
y=x;
```

әрекетті гана жүзеге асырады. Бірақ жиымдарды және құрылымдарды пайдалан-ған кезде нұсқауыштың атқаратын ісі көптеген әрекеттерді орындауды жеңіл-детеді.

Нұсқауышты сипаттау. Нұсқауыш типті айнымалыны сипаттағанда берілген нұсқауыш қандай типтегіі айнымалыға сілтейтінін көрсету қажет. Өйткені әртүрлі типті айнымалыға компьютер жадында ұялардың да әртүрлі саны бөлініп беріледі. Мысалы:

```
int *pi;        // бүтін типті айнымалыға нұсқауыш
char *pc;       // символдық айнымалыға нұсқауыш
float *pf,*pg; // накты типтегі айнымалыға нұсқауыш
```

Нұсқауыштармен орындалатын қарапайым әрекеттерді келесі мысалдан көруге болады.

```
#include <stdio.h>
#include <conio.h>
main()
{
  float x=15.15,y,*ptr;
  ptr=&x; clrscr();
  y=*ptr;
  printf("x=%f y=%f\n",x,y);
  *ptr++;
  printf("x=%f y=%f\n",x,y);
  y=1+*ptr*y;
  printf("x=%f y=%f\n",x,y);
  getch();
  return 0;
}
```

```
x=15.150000 y=15.150000

x=15.150000 y=15.150000
```

Бұл программа нәтижесі төртбұрыш ішінде көрсетілген.

Нұсқауыштарға меншіктеу операциясын қолдана аламыз. Бір типтегі нұсқауыштар меншіктеу операцияларында қарапайым айнымалылар сиякты пайдаланыла алады. Тағы бір мысал қарастырайық.

```
#include <stdio.h>
#include <conio.h>
main()
{ int x=10,*g,*p;
```

```
    clrscr();
    p=&x;
    g=p;
    printf("p=%p\n",p); //  p ішкі мәнін шығару
    printf("g=%p\n",g); //  g ішкі мәнін шығару
    printf("x=%d *g=%d\n",x,*g); //  x-ті және g адресіндегі
                                 //  шаманы шығару
    getch();
    return 0;
}
```

Бұл мысалда printf() функциясының тағы бір %p форматы көрсетілген. Ол жады адресін оналтылық түрде экранға шығарады. Мысал нәтижесі:

```
p=FFF4

g=FFF4
```

Нұсқауышқа кез келген жады адресін меншіктеуге болады. Бірақ бүтін типке мынадай нұсқауыш сипатталса, `int *p;`
Ал, бұл нұсқауышқа берілген адресте нақты `float` типіндегі x айнымалысы тұрса, оны орындау кезінде

```
p=&x;
```

жолында қате шығады. Мұндай қатені int типіндегі нұсқауышты `float` типіне төмендегідей түрде әдейілеп түрлендіру арқылы дұрыстай аламыз:

```
p=(int*)&x;
```

Мұнда бірақ бастапқы нұсқауыштың қай типті көрсеткені жайлы мәлімет жоғалады. Осыған мысал келтірейік.

```
#include <stdio.h>
#include <conio.h>
// нұсқауыш типтерінің сәйкессіздігі
main ()
{ float x=10.1,y,*p;
  clrscr();
  p=&x;// мұны p=(int)&x; деп өзгерту керек
  y=p; // осы жолда қате шығады
  printf("x=%f y=%f\n",x,y);
  getch();
  return 0;
}
```

Басқа айнымалылармен жұмыс істеу кезіндегі тәрізді нұсқауыштармен де арифметикалық операцияларды: қосу мен азайтуды орындауға болады (++ және -- амалдары қосу/алудың бір түрі болып табылады). Нұсқауыштармен орындалатын операциялардың өз өзгешеліктері бар. Оған мысал келтірейік.

```
#include <stdio.h>
#include <conio.h>
main ()
{ int *p,*g,x;
  clrscr();
  //x=5;
  p=&x;
  g=p+1;
  printf("p=%p g=%p\n",p,g);
  getch();
  return 0;
}
```

```
P=FFF2 g=FFF4
```

Программа нәтижесінен p мәнін бірге арттырғанда, g нұсқауыш мәні екіге өзгергенін байқаймыз. Бұл дұрыс, өйткені нұсқауыштың жаңа мәні жады ұясының келесі адресін емес, келесі бүтін санның адресін көрсетуі тиіс.

Ал бүтін санның компьютер жадында екі байт орын алатыны белгілі ғой. Егер нұсқауыштың типі `float` болса, онда адрес 4-ке, ал `double` болса, 8-ге өзгеретін еді, яғни адрес берілген типтің жадыда алатын орнына (ұзындығына) сәйкес артып отырады.

Нұсқауыштарға тек бірді ғана қосу/азайту емес, оларға шағын бүтін сандарды қосып алуға болады. Мысалы бүтін типтегі p нұсқауышының мәні 2000 болатын болса, онда g=p+3 нәтижесінде g мәні 2006 болып өзгереді. Ал егерде float типіндегі нұсқауыш мәні p1=2000 болса, онда мынадай амал
g1=p1+10; нәтижесінде g1 мәні 2040 болар еді. Жалпы нұсқауыш мәнін мынадай p+n амалынан соң есептеу өрнегі:

<p>=<p>+n*<нұсқауыштың базалық типі алатын байттар саны>;

Осындай жолмен бір нұсқауыш мәнінен екінші нұсқауыш мәнін азайта аламыз. Мысалы p мен p1 бір жиым элементтеріне нұсқауыш болатын болса, онда p-p1 сол жиым элементтерінің нұсқауыштарға сәйкес индекстерінің айырмасын береді.

Нұсқауыштармен басқа арифметикалық операцияларды орындай алмаймыз, мысалы, екі нұсқауышты қосуға болмайды, нұсқауышты бір санға көбейтудің де мағынасы жоқ.

Нұсқауыштарды салыстыруға болады, мұнда барлық 6 операцияны да қолдана аламыз: <, >, <=, >=, ==, !=.

Мысалы, p<g операциясы p-да сақталған адрестің g-да сақталған адрестен кіші екенін білдіреді. Егер бұлар бір жиым элементтерін нұсқап тұрса, онда p көрсетіп тұрған элемент индексі g көрсетіп тұрған элемент индексінен кіші болады.

5.4.1 Функциялар арасында байланыс жасау үшін нұсқауыштарды пайдалану

Кейде функцияны нұсқауышсыз пайдалана алмайтын кездер болады. Нұсқауышсыз қолданылатын функция мысалын келтірейік, мұнда екі айнымалының мәндерін бір-бірімен алмастырамыз.

```
#include <iostream.h>
void change(int u,int v);
void main()
{
  int x=5,y=10;
  cout << x << " " << y << endl;
  change(x,y);
  cout << x << " " << y << endl;
}
change(int u,int v)
{
  int temp;
  temp=u;
  u=v;
  v=temp;
}
```

Программаны орындаған соң, оның нәтижесі мынадай болады:

```
5 10
```

Мұнда change функциясы ішінде u,v айнымалылары мәндерін алмастырады, бірақ ол негізгі программадағы x,y мәндеріне әсер етпейді.

Функция (return операторы да) негізгі программаға тек бір шаманы ғана қайтарады, ал оған екі мән қайтару керек. Функция дұрыс жұмыс істеуі үшін нұсқауыштарды пайдалану керек.

1-мысал. Бұл программада айнымалының мәндерін ауыстыру үшін нұсқауыштар пайдаланылған:

```
#include <iostream.h>
void change(int *u,int *v);
void main()
{
int x=5,y=10;
cout << x << " " << y << endl;
change(&x,&y);
cout << x << " " << y << endl;
}
change(int *u,int *v)
{
int temp;
```

```
temp=*u;
*u=*v;
*v=temp;
}
```

Нәтижесі:

```
x=5  y=10
x=10 y=5
```

Функцияны шақырғанда, айнымалы туралы ақпарат функцияға екі түрде беріле алады екен.

Егер function1(x) түрі пайдаланылса, x мәні беріледі. Ал егер function2(&x) түрі пайдаланылса, онда x адресі беріледі.

Нұсқауыштарды инициалдау

Нұсқауыштарды инициалдауды оларды хабарлау кезінде жасауға болады. Мысалы:

```
int ir;
int *pr=&ir;
```

Нұсқауышты *pr деп жарияласақ, оған ir айнымалысының адресі меншіктеледі, ал ir айнымалысы әлі инициалданбаған. Адресті алу операциясын кез келген өрнекпен колдана алмаймыз.

5.4.2 Нұсқауыштардың жиымдарға қолданылуы

Жиымдарды белгілеп жазу нұсқауыштарды қолданудың жасырын бір түрі болып табылады. Жиым аты оның алғашқы элементін көрсетіп тұрған нұсқауыш болып саналады, мысалы,

f[] – жиым болса, онда

f=&f[0]; болады,

соңғы теңдіктің екі жағы да жиымның алғашқы элементін анықтайды. Екі белгілеу де нұсқауыш типіндегі тұрақты (константа) болып саналады, өйткені олар программа жұмысы барысында ешқандай өзгеріске ұшырамайды. Бірақ оларды нұсқауыш типіндегі айнымалыға меншіктеп, сол айнымалының мәнін өзгертуге болады. Мысал келтірейік.

```
#include <iostream.h>
void main()
{
    int d[4]={1,2,3,4};
    int *p,i,g;
    float b[4]={1,2,3,4};
    float *p1;
    p=d; g=*(p+2);
    p1=b;i=*(p1+1);
    cout << g << " " << i;
}
```

Бұл программа нәтижесі: 3 2

Келесідей тендіктер дұрыс болып саналады:
d+2==&d[2] – екеуі де бір адресті көрсетіп тұр.
*(d+2)==d[2] – бұлар да бірдей.
Мынадай өрнек *(d+2) – жиымның 3-ші элементінің мәнін береді, ал мына өрнек (*d+2) – жиымның алғашқы элементі мәніне 2-ні қосады.
Келесі операторлар бір-бірімен синоним болып есептеледі.

```
int y[];
int *y;
```

Жиымдармен жұмыс істеу кезінде нұсқауышты пайдалану мысалдары:

```
// жиым элементтерінің арифметикалық ортасын табу
#include <iostream.h>
void main()
{
     int d[]={1,2,3,4},*p,i,n;
     float s=0,sr;
     n=sizeof d/(sizeof(int));
     cout << n << endl;
     p=d;
     for(i=0;i<n;i++)
          s+=*(p+i);
     sr=s/n;
     cout << sr << endl;
}
```

```
4
2.5
```

Алдымен жиымды, сонан соң нұсқауышты пайдаланып, арифметикалық ортаны анықтайтын функция жазайық.

```
// жиым арқылы арифметикалық ортаны анықтау
#include <iostream.h>
int mn(int p[],int n);
int main()
{
     int d[]={1,2,3,4},r;
     r=mn(d,4);

     cout<<r;
}
int mn(int p[],int n)
{
     int i; long s;
     if(n>0)
     {
          for(i=0,s=0;i<n;i++)
               s+=p[i];
          return((int)(s/n));}
}
```

```
// нұсқауыш арқылы арифметикалық ортаны анықтау
#include <iostream.h>
int mn(int *p,int n);
void main()
{
     int d[]={1,2,3,4},r;
     r=mn(d,4);
     cout<<r;
}
int mn(int *p,int n)
{
     int i; long s;
     if(n>0)
     {
          for(i=0,s=0;i<n;i++)
               s+=*(p+i);
          return((int)(s/n));}
}
```

Функцияны пайдаланбай, жиым элементтерінің ең кіші (min) және ең үлкен (max) мәндерін анықтайтын тағы бір мысал келтірейік.

```
#include <iostream.h>
void main()
{
     int d[]={1,2,3,4},*p,i,n,min,max;
     float s=0,sr;
     n=sizeof d/(sizeof(int));
     p=d;
     min=*p;max=*p;
     for(i=0;i<n;i++)
          if(*(p+i)<min)
               min=*(p+i);
          else
               if(*(p+i)>max)
                    max=*(p+i);
     cout << min << endl << max << endl;
}
```

Осы мысалдың тағы бір нұсқасы:

```
#include <iostream.h>
void main()
{
     int d[]={1,2,3,4},*p,i,n,min,max;
     float s=0,sr;
     n=sizeof d/(sizeof(int));
     p=d;
     min=*p;max=*p;
     for(i=0;i<n;i++)
```

```
            if(*(p+i)<min)
                  min=*(p+i);
            else
                  if(*(p+i)>max)
                        max=*(p+i);
      cout << min << endl << max << endl;
}
```

Енді функция арқылы орындайық.

```
#include <iostream.h>
void minmax(int *p,int *min,int *max,int n);
void main()
{
      int d[]={1,2,3,4},*p,n;
      int min,max;
      n=sizeof d/(sizeof(int));
      p=d;
      minmax(d,&min,&max,n);
      cout << min << endl << max << endl;
}
void minmax(int *p,int *min,int *max,int n)
{
      int i;
      *min=*p;*max=*p;
      for(i=0;i<n;i++)
            if(*(p+i)<*min)
                  *min=*(p+i);
            else
                  if(*(p+i)>*max)
                        *max=*(p+i);
}
```

S жиымының индекстері екінің дәрежелері болып келетін элементтерін экранға шығару есебін карастырайық.

```
#include <stdio.h>
#include <conio.h>
#include <malloc.h>
void main ()
{  int i, n, *s;
m1: clrscr() ;
  printf("\n Жиым элементтері санын енгіз (<360): ");
  scanf ("%d",&n);
  s=(int*)malloc(n*sizeof(int));  // жады бөлеміз
  puts("\n\t Берілген жиым ");
  for(i=0; i<n; i++)
   {
     s[i]=i;  // жиым элементтері өз индекстеріне тең
```

```
   printf ("%4d", s[i]);
}
printf("\n Керекті жиым элементтері: ");
for(i=1;i<n; printf("*d ",s[i]), i*=2);
free(s);  // жадыны босатамыз
printf("\nЖалғастырамыз ба ? Иә болса - 7 енгіз:");
scanf ("%d", &i);
if(i==7) goto m1;
}
```

Нұсқауыштарды көпөлшемді жиымдарда пайдалану

Мынадай түрде сипатталу болсын делік:

```
int z[4][2]; int *p;
```

онда,

`p==&z[0][0]` – 1-ші жолдың 1-ші элементін көрсетеді.
`p+1==&z[0][1]` – 1-ші жол, 2-ші бағана.
`p+2 ==&z[1][0]` – 2-ші жол, 1-ші бағана.
`p+3 ==&z[1][1]` – 2-ші жол, 2-ші бағана.

```
#include <iostream.h>
void main()
{
  int d[2][2]={ {1,2},{3,4} };
  int *p,*p1,g,g1;
  p=&d[0][0];p1=p+1;
  g=*p;
  g1=*p1;
  cout<<g<<g1;
}
```

Екіөлшемді жиымды функцияға беру

```
#include <iostream.h>
int mn(int p[2][2],int m,int n);
int main()
{
  int r;
  int a[2][2] = {
                   {1,2},
                   {3,4}
                 };
  r = mn(a,2,2);
  cout << r << endl;
}
  int mn(int p[2][2],int m,int n)
```

```
{
   int i,j,s=0;
   for(i=0;i<m;i++)
   for(j=0;j<m;j++)
   s+=p[i][j];
   return(s);
}
```

Жиымдарды сипаттағанда, элементтер типі және жалпы жағдайда компьютер жадының қажетті класы көрсетіледі. Қарапайым айнымалыларда қарастырылатын қасиеттер жиымдарды да сипаттау кезінде қолданылуы мүмкін. Мысалы:

```
int b[30];
main()
{ float a[30];
      static char c[20];
      extern b[ ];
      ......
}
```

Жиымдарды инициалдауды қарастырайық. Жиым сипатталуында тек сыртқы немесе статистикалық жиымдар ғана инициалдануы мүмкін. Мысалы, 8-топтағы студенттер саны `stud[8]` жиымы ретінде көрсетілген:

```
#include <conio.h>
#include <stdio.h>
int stud[8]={15,16,14,18,15,20,17,19};
main()
{ int i;
   clrscr();
   extern int stud[ ];
   for(i=0; i<8; i++)
   printf("N %i тобында %i студент\n",  i+1,stud[i]);
}
```

Жиымның нұсқауыштарын қарастырайық. Мысалы: `int *a;`
Жиым аты нұсқауышты қолданған жағдайда, жиымның 0-інші элементін анықтайды, яғни a жиымы сипатталған болса, программа мәтініндегі a идентификаторы 0-інші элементті көрсетеді деп саналады:

```
a == &a[0];
```

бұл теңдеудің екі бөлігі де – a да және &a[0]-де жиымның 0-ші элементінің адресін анықтайды. Осы екі белгілеу де нұсқауыш типті константа болып табылады. Сондықтан оларды мән ретінде нұсқауыш типті айнымалыға меншіктеуге болады немесе қажет болса, нұсқауыш типті айнымалының мәнін өзгертуге болады.

1-мысал. Нұсқауыштың мәніне санды қосуға болатынын көрсететін программа қарастырайық.

```
main()
```

```
{int a[4], *pti, i;
 float b[4], *ptf;
 pti=a;
 ptf=b;
 for(i=0; i<4; i++)
 printf("нұсқауыштар +% d: %8u %10u\n",i,pti+i,ptf+i);
}
```

Мұның нәтижесі:

```
нұсқауыштар + 0:   65518   65498
нұсқауыштар + 1:   65520   65502
нұсқауыштар + 2:   65522   65506
нұсқауыштар + 3:   65524   65510
```

Келесі мысалды қарастырайық.

```
(a+2)==&a[2];
*(a+2)==a[2];
```

Бұлар нұсқауыштар мен жиымның арасындағы байланысты анықтайды, яғни жиымның жеке элементін анықтау үшін немесе оның мәнін пайдалану үшін нұсқауышты қолдануға болады.

5.4.3 Нұсқауыштарды пайдаланып жиымдармен жұмыс істеу

Жиымдарды функция аркылы карастырып, содан кейін осы функцияны нұсқауыштарды пайдаланып жазып шығу керек болсын.

1-мысал. Функция аркылы жиым косындысын табу.

```
// функция аркылы жиым косындысын табу
#include <conio.h>
#include <stdio.h>
int f1(int a[], int t)
{
 int i, sum=0;
 for(i=0; i<t; i++)
    sum+=a[i];
 return (sum);
}
main()
{ int i,s,b[10]={5,6,14,12,30,5,9,7,15,5};
   clrscr();
   s=f1(b,10);
   printf("s=%d", s);
   getch();
}
```

Негізгі программада f1 функциясын шакыру үшін нақты параметрлерді жазып, функция келесі түрде шакырылып отыр:

```
f1(b,10);
```

2-мысал. Нұсқауышты функцияда (f1) пайдалану программасын жазайық.

```
// функцияда нұсқауыш арқылы жиым қосындысын табу
#include <conio.h>
#include <stdio.h>
int f1(int *pa, int t)
{int i,sum=0;
 for(i=0; i<t; i++)
    sum += *(pa+i);
 return (sum);
}
main()
{ int i,s,b[10] = {5,6,14,12,30,5,9,7,15,5};
  clrscr();
  s=f1(b,10);
  printf("s=%d",s);
  getch();
}
```

программада осы f1 функциясын шақыру үшін нақты параметрлер бұрынғыдай жазыла береді: `f1(b,10);`

5.4.4 Нұсқауыштарға қолданылатын операциялар

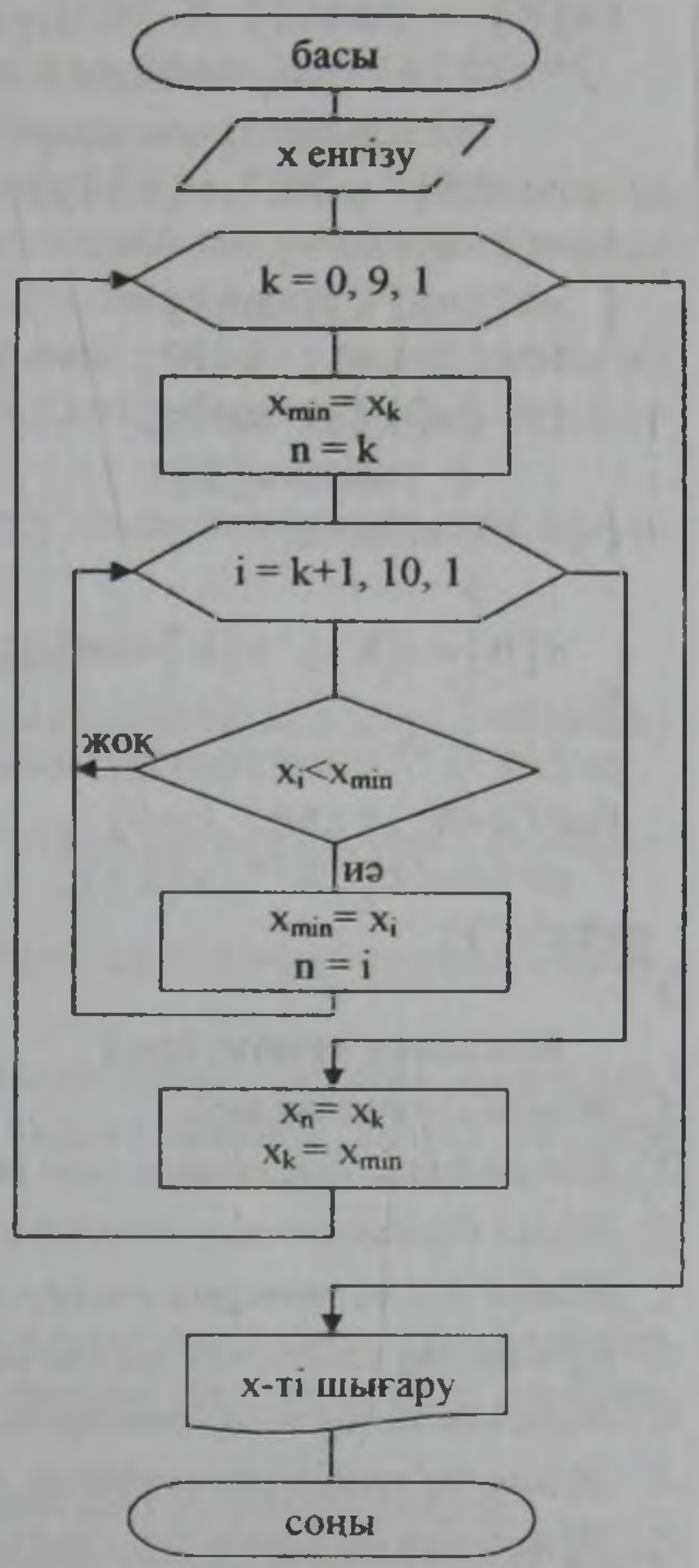
С тілінде нұсқауыш типті айнымалыларға бес түрлі оператор қолдануға болады:

1) меншіктеу операциясы. Нұсқауышқа адресті меншіктеуге болады. Жиымның атын қолданып немесе адресті анықтайтын & операторын пайдаланып, әдетте адресті меншіктеуге болады;
2) мәнді анықтау. Берілген адрес бойынша кейбір ұяшықта сақталатын мәнді анықтау үшін * операциясы қолданылады;
3) нұсқауыштың адресін анықтау. Кез келген айнымалылар сияқты нұсқауыш типті айнымалылар мәні немесе адресі болуы мүмкін. & операциясы арқылы нұсқауыштың адресін анықтауға болады;
4) нұсқауыштарды арттыру. Бұл амал әдеттегі + операциясы көмегімен немесе арттыру операциясы арқылы орындалуы мүмкін. Нұсқауышты арттырып, жиымның келесі элементіне өтуге болады (кажет болса, нұсқауыш мәнін кемітуге де болады);
5) нұсқауыштардың айырмасы. Бір жиымның элементіне сілтейтін нұсқауыштың айырмасын табуға болады. Жиым элементінің арасындағы ара қашықтығын анықтау үшін нұсқауыштың айырмасын есептеуге болады.

1-мысал.

```
/* Жиымның максимумын тауып, одан кейінгі элементтерін кему
бойынша реттеп орналастыру */
```

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
#define  n  10
main()
{ int i, j, t, c;
  int a[n]={6,5,9,8,7,4,1,2,3,0};
  int *pa, max;
  clrscr();
  printf("a[10] элементтері : ");
  for(i=0; i<n; i++)
     printf(" %d ",a[i]);
  pa=a; max=*pa; t=0;
/* максимумды және оның  индексін
табу  */
for(i=1; i<n; i++)
 if((*(pa+i)) > (max))
  {max = *(pa+i); t=i; }
 printf("\nmax = %d оның индексі =
%d\n", max, t);
for(i=t; i<n-1; i++)
/* элементтерді кемуі бойынша */
for(j=i+1; j<n; j++)  // реттеу
if(*(pa+i) < *(pa+j))
 { c=*(pa+i); /* жиымның екі элементін
алмастыру */
   *(pa+i)=*(pa+j);
   *(pa+j)=c;
 }
 printf("\n нәтиже : ");
 for(i=0; i<n; i++)
   printf("%d ",*(pa+i));
getch();
}
```



5.3-сурет. Жиым элементтерін өсуі бойынша реттеу

2-мысал. Берілген x_{10} жиым элементтерін сол жиымда өсу реті бойынша орналастыру керек (5.1-сурет).

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
main()
{
  int xmin,x[10];
  int n,k,i;
  clrscr();
  printf("\nБерілген жиым элементтері:");
```

```
for(k=0; k<10; k++)
{x[k] = rand() % 100;
 /* 32767-ге дейінгі кездейсоқ сандарды 100-ге бөліп,
      қалдықтарын алу */
 printf("  %i",x[k]);}
for(k=0; k<9; k++)
{ xmin=x[k]; n=k;
  for(i=k+1; k<10; k++)
  if (x[i] < xmin)
      { xmin=x[i];
        n = i;
      }
    x[n]=x[k]; x[k]=xmin;
  }
  printf("\nРеттелген жиым элементтері:");
  for(k=0; k<10; k++)
    printf(" %i",x[k]);
 getch();
}
```

Бақылау сұрақтары

1. *Жиым дегеніміз не?*
2. *Жиымдарға бастапқы мәндер қалай тағайындалады?*
3. *Жиымды сипаттау тәсілдері.*
4. *Жиым элементтерін енгізу және экранға шығару жолдары.*
5. *Кездейсоқ сандарды қалай шығаруға болады?*
6. *Жиымға кездейсоқ сандарды меншіктеу қалай орындалады?*
7. *Жиымды өңдеу есептерінің түрлері (кластары).*
8. *Жиымның ең үлкен (ең кіші) элементін анықтау.*
9. *Жиым элементтері қосындысын табу.*
10. *Жиым ішіндегі екі элементтің бір-бірімен орнын алмастыру.*
11. *Жиым элементтерін кері бағытта орналастыру.*
12. *Жиымның көрсетілген элементтерін өңдеу тәсілдері.*
13. *Жиымды сұрыптау (сорттау, реттеу) жолдары.*

ТАПСЫРМАЛАР

Жиымдарды өңдеу кезінде төмендегі ережелерді есте сақтаған жөн.

1. Жиым өлшемі (ені, көлемі) тек константа немесе константалық өрнек бола алады. Жиым өлшемін идентификатор түріндегі (ат қойылған) константа аркылы беру дұрыс деп саналады.
2. Жиым элементтері нөлден бастап нөмірленеді, сол себепті элементтің ең үлкен нөмірі элементтер санынан бірге кем болады.

3. Индекс нөмірлерінің жиым шекарасынан асып кетпеуін программа қадағаламайды, оны программалаушы өзі бақылауы тиіс.
4. Нұсқауыш – жедел жады аймағының адресін сақтайтын айнымалы.
5. Жиым аты оның 0-элементіне нұсқауыш болып табылады.
6. Егер программаға енгізілетін элементтер мәні алдын ала белгілі болса, онда жиымды сипаттау кезінде оның мәндерін көрсетіп (инициалдау) кету керек. Жиымды глобальді айнымалы емес, локальді айнымалы етіп сипаттаған дұрыс.
7. Жиым элементтерін пернеден енгізгеннен гөрі rand() және random() функцияларын пайдаланып, кездейсоқ сандар тізбегін беру арқылы енгізген ыңғайлы болады.
8. Жиымдарды сұрыптау алгоритмдерінің жұмыс істеу жылдамдығы, алатын көлемі жәнеколданылу аймағы әртүрлі болып келеді.

Тапсырмаларды орындауға арналған нұсқаулар

а) Әрбір студент журналдағы өз нөміріне сәйкес (мұғалімнің көрсетуі бойынша) төрт есеп шығаруы тиіс.

ә) Әр есептің блок-схемасы және соған сәйкес программасы құрылып, компьютерде сол программаның нәтижесін алу керек.

б) Екі блок-схема Word программасындағы *Сурет салу-Автофигуралар-Блок-схемалар* мүмкіндігімен сызылып көрсетілуі тиіс.

в) Берілген жиым элементтері мен нәтижелік элементтерді экранға шығарып, нәтижелерді жазып алу қажет немесе нәтижені бірден файлға жазуға тырысу керек.

Есептер

1. Кез келген сандардың бірөлшемді *A*(10) жиымындағы оң элементтерді екі есе кемітіндер, ал теріс элементтері болса, онда оларды индекстерінің мәнімен ауыстырыңдар.
2. Бірөлшемді *A*(10) жиымындағы теріс элементтердің ең үлкенін табыңдар.
3. Бірөлшемді *A*(20) жиымындағы -5-тен кіші элементтердің қосындысын және олардың жалпы санын, сонымен бірге 5 санына қалдықсыз бөлінетін элементтердің нөмірлерін анықтаңдар.
4. Бірөлшемді *A*(10) жиымындағы оң элементтердің квадраттарының арифметикалық ортасын есептеңдер.
5. Бірөлшемді *A*(100) жиымындағы теріс элементтердің санын анықтаңдар.
6. Бірөлшемді жиым мәндері берілген диапазонда жататын элементтерінің нөмірін басып шығарыңдар.
7. Бірөлшемді *A*(10) жиымындагы оң элементтердің ішінде мәні ең кішісін және оның индексін (нөмірін) табыңдар.
8. Бірөлшемді жиымда 2-элементті бірінші орынға, 3-ні 2-шімен және т.б. алмастырулар орындай отырып, 1-элементі соңғы орынға қойыңдар.
9. Берілген оң сандар тізбегіндегі қосындысы берілген саннан асып кетпейтін элементтердің санын табыңдар.
10. Ұтыс билетінің нөмірі алты орынды сан. Билет “бақытты” (алдыңғы және соңғы үш цифрдың қосындысы өзара тең) немесе “табысты” (жұп орындарда

тұрған цифрлардың қосындысы тақ орында тұрғандардың қосындысына тең) болатындығын анықтандар.

11. Оң және теріс сандар тізбегіндегі бірінші теріс санға дейінгі орналаскан сандар тізбегінің өсу ретімен орналасатындығын анықтандар.
12. *N* кәсіпорынның бір жылғы электр энергиясын тұтынуы туралы дерек бар. Осы бойынша энергия тұтынудың арифметикалық ортасын және энергияны ең көп үнемдеген кәсіпорынды анықтандар.
13. 14 аудан бойынша жанармай қорының мөлшері белгілі. Жанармаймен ең жақсы қамтылған үш ауданды анықтандар.
14. Берілген *A*(14) жиымы бойынша мына шарттарды қанағаттандыратын *B* жиымын құрындар: $B(0)=A(1)$; $B(1)=A(1)*A(3)$; ... $B(4) = A(1)*A(3)* \ldots *A(13)$
15. Берілген *A*(13) жиымы бойынша мына шарттарды қанағаттандыратын *B* жиымын құрындар: $B(0)=A(0)$;$B(1)=A(0) * A(2)$; ... $B(6) = A(0)*A(2) * \ldots *A(12)$
16. Берілген *A*(14) жиымы бойынша мына шарттарды қанағаттандыратын B жиымын құрындар: $B(0) = A(0) - A(13)$; $B(1) = A(1) - A(12)$; ... $B(6) = A(6) - A(7)$
17. Берілген *A*(14) жиымы бойынша мына шарттарды қанағаттандыратын B жиымын құрындар: $B(0) = A(0)/A(13)$; $B(1) = A(1)/A(12)$, ... $B(6) = A(6)/A(7)$
18. *A*(14) жиымы берілген. Жиым элементтерінің арифметикалық ортасын және ол мәнге ең жақын элементтің нөмірін анықтандар.
19. *A*(14) жиымы берілген. Екі жиым құрындар: оның біріншісі индекстері жұп сандар, ал екінші жиым индекстері тақ сан болатын элементтер болсын.
20. *A*(14) жиымы берілген. Ең кіші элементтің алдында орналаскан элементтерді жою арқылы сызықтық жиымды қыскартындар.
21. Өлшемдері бірдей екі сызықтық жиым берілген. Осы жиымдардың индекстері бірдей орындарында орналасқан элементтердің көбейтіндісінен тұратын үшінші жиым құрындар.
22. Бүтін сандардың бірөлшемді жиымы берілген. Ондағы ең көп кездесетін санның неше рет қайталанатынын табындар.
23. *A*(17) жиымы берілген. 7-ден үлкен барлық элементтерді берілген жиымның ең үлкен элементімен ауыстырындар.
24. *A*(16) жиымы берілген. Алғашқы оң элементке дейінгі теріс элементтердің санын анықтандар.
25. *A*(18) жиымы берілген. 2-ден кіші барлық элементтерді 3 санымен ауыстырындар.
26. *A*(17) жиымы берілген. 0-ден үлкен барлық элементтерді 5-ке, ал қалғандарын 0-ге ауыстырындар.
27. *A*(14) жиымы берілген. *A* жиымының барлық тақ элементтері индекс нөмірлерімен ауыстырылатын *B* жиымын құрындар.
28. *A*(34) жиымы берілген. Тақ нөмірлі индекстерде тұрған элементтердің ең үлкенін анықтандар.
29. *A*(*N*) жиымы берілген. Алғашқы ең кіші (минимал) элементтің орнын (индексін) анықтандар.
30. *A*(*N*) жиымы берілген. Соңғы ең кіші элементтің орнын анықтандар.

31. *A*(*N*) жиымы берілген. Соңғы ең үлкен (максимал) элементтің орнын анықтаңдар.
32. *A*(*N*) жиымы берілген. Алғашқы ең үлкен элементтің орнын анықтаңдар.
33. *A*(*N*) жиымы берілген. Мәндері 3 пен 11 саны интервалында жататын элементтердің арифметикалық ортасын анықтаңдар.
34. *A*(12) = {2,5; 4,3; –0,57; 10,45; 1,5; –7,1; 11,4; 5,12; 4,9; 7,7; –12,3; 0,031} жиымы берілген. *B*(*I*) = SIN(*A*(*I*)) формуласы бойынша *B*(12) жиымын құрастырыңдар және *P* = *A*(0) * *B*(11) + *A*(1) * *B*(10) + ... + *A*(11) * *B*(0) өрнегін есептеп шығарыңдар.
35. Пернетақтадан жиымға бес бүтін мән енгізіңдер. Олар бір жолға үтір арқылы жазылады. Жиымның арифметикалық ортасын табыңдар.
36. Пернетақтадан *X* жиымының бес бүтін мәнін енгізіңдер. Жиым элементтерінің әрқайсысының түбірінің мәнін және квадратын экранға шығарыңдар.
37. Бүтін сандар жиымындағы жұп нөмірлі элементтердің қосындысын табыңдар. Жиымның өлшемі – 20. Жиымды 100 бен 200 арасындағы кездейсоқ сандармен толтырыңдар.
38 Бүтін сандар жиымындағы 7-ге қалдықсыз бөлінетін элементтердің көбейтіндісін табыңдар. Жиым өлшемі – 15. Жиымды 10 мен 50 арасындағы кездейсоқ сандармен толтырыңдар.
39. Кездейсоқ сандар генераторының көмегімен *A*[0..7] жиымын құрыңдар және оны экранға шығарыңдар. Оның барлық элементтерін 2 есеге арттырыңдар.
40. Кездейсоқ сандар генераторының көмегімен элементтері –10 мен 10 сан аралығында болатын *A*[0..8] жиымын құрыңдар және оны экранға шығарыңдар. Жиымның теріс элементтерінің санын есептеңдер.
41. Кездейсоқ сандар ғенераторының көмегімен элементтері –20 мен 10 саны аралығында болатын A[0..11] жиымын құрыңдар және оны экранға шығарыңдар. Жиымның барлық теріс элементтерін 0-ге ауыстырыңдар.
42. Кездейсоқ сандар генераторы аркылы элементтері –15 пен 30 саны аралығында болатын бүтін саннан A[0..14] жиымын құрастырыңдар және оны экранға шығарыңдар. Жиымның ең үлкен элементін және оның индексін анықтаңдар.
43. 8 сағаттан 20 сағатқа дейінгі уақыт аралығында ауа температурасы сағат сайын өлшенеді. Осы аралықта температураның төмендегені мәлім. Температураның алғашқы теріс мәні қай сағатта пайда болғандығын анықтаңдар.
44. Қараша айында он күн бойына ауа температурасы туралы деректер жиымда сақталған. Температураның -5 градустан канша рет төмен болғанын анықтаңдар.
45. Балқаш көлі жағалауының температурасы туралы мәлімет қыркүйек айында он күн бойы жиымда сақталған. Осы уақыт ішінде неше күн шомылуға колайлы болатынын анықтаңдар.
46. Сәуір айының онкүндік ауа температурасы мен жауын-шашын мөлшері туралы мәлімет жиымда сақталады. Осы он күнде жауған жаңбыр және кардың мөлшерін анықтаңдар.

47. Желтоқсан айының онкүндік ауа температурасының мәліметі жиымда сақталған. Онкүндік температура мәліметі бойынша қанша рет орта температурадан жоғары ауытқу болатындығын анықтаңдар.
48. Қараша айының он күн ішіндегі желдің (солтүстік, оңтүстік, шығыс, батыс) бағыты және соғу күшінің мәліметтері жиым түрінде сақталған. Неше күн желдің жылдамдығы 8 м/с-тан артық жылдамдықпен соққандығын анықтаңдар?
49. Бүтін саннан тұратын 15 элементті жиым құрыңдар және олардың арасындағы ең кіші элементті анықтаңдар.
50. Жерге еркін түсу кезінде дененің 1, 2, ... , 10 с ішінде жүріп өткен қашықтықтарынан құралатын нақты сандардың сызықтық жиымын құрыңдар.
51. Бүтін сандардың сызықтық жиымы берілген. Оның кему ретімен орналасқандығын тексеріңдер.
52. Бүтін сандардың сызықтық жиымындағы оң элементтердің қосындысын табыңдар. Жиымның өлшемі – 10. Жиым элементтерін пернетақта арқылы енгізіңдер.

Бірөлшемді жиымдар

Бұл тақырып бойынша зертханалық жұмыстар тапсырмаларын жүзеге асыратын программалар құрыңдар.

1 нұсқа

n нақты элементтерден тұратын бірөлшемді жиымда:

1) жиымның теріс элементтерінің қосындысын;
2) жиымның ең үлкен (максимал) және ең кіші (минимал) элементтерінің арасында орналасқан элементтерінің көбейтіндісін анықтаңдар;

Жиым элементтерін өсуі бойынша реттеңдер.

2 нұсқа

n нақты элементтерден тұратын бірөлшемді жиымда:

1) жиымның оң элементтерінің қосындысын;
2) жиымның модулі бойынша ең үлкен (максимал) және модулі бойынша ең кіші (минимал) элементтерінің арасында орналасқан элементтерінің көбейтіндісін анықтаңдар;

Жиым элементтерін кемуі бойынша реттеңдер.

3 нұсқа

n бүтін элементтерден тұратын бірөлшемді жиымда:

1) жиымның жұп нөмірлі элементтерінің көбейтіндісін;
2) жиымның алғашқы және екінші нөлге тең екі элементінің арасында орналасқан элементтерінің қосындысын анықтаңдар;

Жиымның барлық оң элементтері оның алдыңғы жағында, ал теріс элементтері соңғы жағында орналасатындай етіп түрлендіріңдер.

4 нұсқа

n нақты элементтерден тұратын бірөлшемді жиымда:

1) жиымның тақ нөмірлі элементтерінің қосындысын;
2) жиымның алғашқы және ең соңғы теріс екі элементінің арасында орналасқан элементтерінің қосындысын анықтаңдар;

Жиымның модульдері 1-ден кіші элементтерін жойып, қысқартындар да, жиымның соңында бос қалған элементтерін нөлдермен толтырындар.

5 нұсқа

n нақты элементтерден тұратын бірөлшемді жиымда:

1) жиымның ең үлкен элементін;
2) жиымның алғашқы теріс элементіне дейін орналасқан элементтерінің қосындысын анықтандар;

Жиымның модульдері [a, b] аралығында орналасқан элементтерін жойып, қысқартындар да, жиымның соңында бос қалған элементтерін нөлдермен толтырындар.

6 нұсқа

n нақты элементтерден тұратын бірөлшемді жиымда:

1) жиымның ең кіші элементі мен оның нөмірін;
2) жиымның ең алғашқы және ең соңғы оң элементтері арасында орналасқан элементтерінің қосындысын анықтандар;

Жиымның алдыңғы жағында тек нөлге тең элементтері, сонан кейін барып басқалары орналасатындай етіп түрлендіріңдер.

7 нұсқа

n бүтін элементтерден тұратын бірөлшемді жиымда:

1) жиымның ең үлкен элементі мен оның нөмірін;
2) жиымның ең алғашқы және ең соңғы нөлге тең элементтері арасында орналасқан элементтерінің көбейтіндісін анықтандар;

Жиымның бірінші жартысында – оның тақ нөмірлі элементтері, ал екінші жартысында – жұп нөмірлі элементтері орналасатындай етіп түрлендіріңдер.

8 нұсқа

n нақты элементтерден тұратын бірөлшемді жиымда:

1) жиымның ең кіші элементінің нөмірін;
2) жиымның ең алғашқы және одан кейінгі теріс элементтері арасында орналасқан элементтерінің қосындысын анықтандар;

Жиымның алғашқы жағында – оның модульдері 1-ден кіші элементтері, ал соңында – қалған элементтері орналасатындай етіп түрлендіріңдер.

9 нұсқа

n нақты элементтерден тұратын бірөлшемді жиымда:

1) жиымның модулі бойынша ең үлкен элементін;
2) жиымның ең алғашқы және одан кейінгі оң элементтері арасында орналасқан элементтерінің қосындысын анықтандар;

Жиымның нөлге тең элементтері оның соңғы жағында орналасатындай етіп түрлендіріңдер.

10 нұсқа

n бүтін элементтерден тұратын бірөлшемді жиымда:

1) жиымның модулі бойынша ең кіші элементін;
2) жиымның нөлге тең алғашқы элементінен кейінгі элементтерінің модульдерінің қосындысын анықтандар;

Жиымның бірінші жартысында – оның жұп нөмірлі элементтері, ал екінші жартысында – тақ нөмірлі элементтері орналасатындай етіп түрлендіріңдер.

11 нұсқа

n нақты элементтерден тұратын бірөлшемді жиымда:

1) жиымның модулі бойынша ең кіші элементі нөмірін;
2) жиымның алғашқы теріс элементінен кейін орналасқан элементтерінің модульдері қосындысын анықтандар;

Жиымның мәндері [a, b] аралығында орналасқан элементтерін жойып, қысқартындар да, жиымның соңында бос қалған элементтерін нөлдермен толтырындар.

12 нұсқа

n нақты элементтерден тұратын бірөлшемді жиымда:

1) жиымның модулі бойынша ең үлкен элементі нөмірін;
2) жиымның алғашқы оң элементінен кейін орналасқан элементтерінің қосындысын анықтандар;

Жиымның мәндерінің бүтін бөлігі [a, b] аралығында орналасқан элементтерін алдыңғы жағына, қалғандарын – соңғы жағына орналасатындай етіп түрлендіріңдер.

13 нұсқа

n нақты элементтерден тұратын бірөлшемді жиымда:

1) жиымның мәндері A мен B аралығында жатқан элементтері санын;
2) жиымның ең үлкен элементінен кейін орналасқан элементтерінің қосындысын анықтандар;

Жиым элементтерін оның модульдерінің кемуі бойынша реттендер.

6 ЕКІ ӨЛШЕМДІ ЖИЫМДАР

Екі өлшемді жиымды – матрицаны пайдалану үшін тік жақшалар ішінде олардың екі өлшемінің де енін көрсету керек. Мысалы: `int a[4][3];`

алғашқы сан жолдар санын, ал екінші сан бағаналар санын көрсетеді, *a* жиымы 12 элементтен тұрады. Оларға бастапқы мәнді былай береміз:

```
int a[4][3]={{0,1,2},
             {3,4,5},
             {6,7,8},
             {9,10,11}
            };
```

ішкі жүйелі жақшаларды қоймаса да болады:

```
int a[4][3]={0,1,2,3,4,5,6,7,8,9,10,11};
```

Келесі түрде сипаттау жолдардың тек бірінші элементтерін ғана анықтайды, қалған элементтер 0-ге тең болып саналады:

```
int a[4][3]={ {0},{3},{6},{9} };
```

Егер ішкі жүйелі жақшалар алынып тасталса, онда мағынасы өзгереді.

```
int a[4][3]={ 0,3,6,9 };
```

мұнда бірінші жолдың 3 элементі мен екінші жолдың бірінші элементі анықталады да, қалғандары 0 болып саналады.

Екі өлшемді жиымды инициалдау қабаттасқан циклдер арқылы орындалады. Енді жиымды кездейсок сандармен толтыратын бір *мысал* карастырайык.

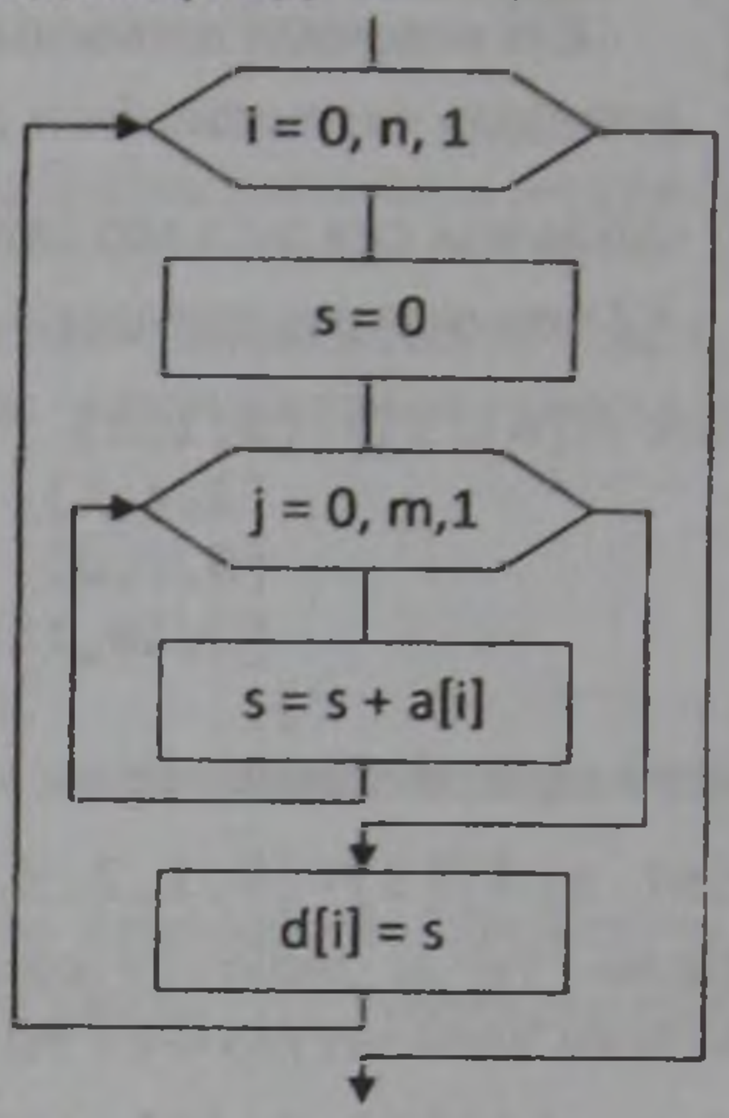
```
/* a[3][4] жиымы  элементтерін rand() арқылы енгізу және экранға
шығару */
#include <stdio.h>
#include <stdlib.h> // rand() функциясы үшін керек
#include <conio.h>
main()
{
 const int row=3, col=4;
 int a[row][col];
 clrscr();
 for(int i=0; i<row; i++)
 for(int j=0; j<col; j++)
       a[i][j]=rand()%100-50;
 printf("\na[3][4] жиым элементтері мәндері:");
 for(i=0; i<row; i++)
 for(j=0; j<col; j++)
       printf(" %i",a[i][j]);
 getch();
}
```

Екі өлшемді жиымдармен жұмыс істеу

Жиым n жолдан және m бағанадан тұратын болсын делік, енді төмендегі әрекеттерді орындау есептерін қарастырайық.

- *Жиымның әр жолдарында орналасқан элементтерінің қосындысын анықтау*. Қосындылар ретінде n элементтен тұратын бір өлшемді d жиымын құрастырамыз. Мысалы (6.1-сурет):

```
for(i=0;i<n;i++)
   {
      s=0;
      for(j=0;j<m;j++)
        s+=a[i][j];
     d[i]=s;
  }
```

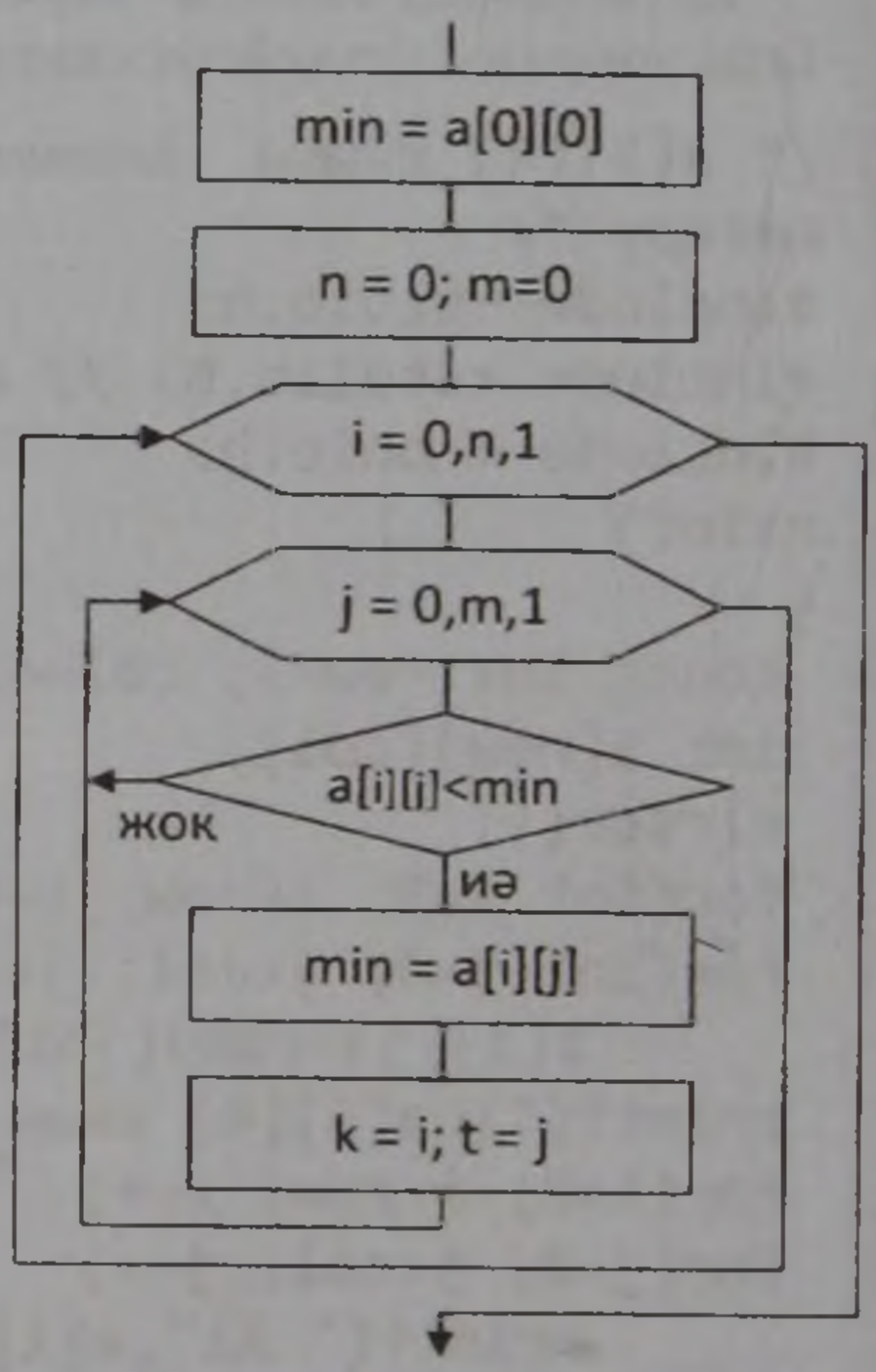


6.1-сурет. Екі өлшемді жиым қосындысын табу

Жиымның әр бағаналарында орналаскан элементтерінің қосындысын анықтау да осы тәсілмен іске асырылады. Мұнда сырткы циклді j айнымалысы, ал ішкі циклді i айнымалысы бойынша орындау керек.

- *Матрицаның $a_{n,m}$ ең кіші элементін табу* кезінде min айнымалысы ең кіші элементтің мәнін сақтайды да, k айнымалысы оның жолы нөмірін, ал t айнымалысы – бағанасы нөмірін есте сақтайды (6.2-сурет).

```
min=a[0][0]; k=0; t=0;
for(i=0;i<n;i++)
  for(j=0;j<m;j++)
    if(a[i][j]<min)
      { min = a[i][j];
        k = i; t = j;
      }
```



6.2-сурет. Матрицаның ең кіші мәнін анықтау

- *a матрицасын x векторына көбейту* кезінде жаңа *y* векторы шығады.

```
for(i=0;i<n;i++)
  { s=0;
    for(j=0;j<m;j++)
      s+=a[i][j]*x[j];
    y[i]=s;
```

Енді тағы бір *мысал* қарастырайық.

Квадрат түріндегі матрица a[3,3] берілген. Онын

элементтерін пернетақтадан енгізіп, одан екі бірөлшемді жиым құру керек, алғашқысына – матрицаның бас диагоналында және одан жоғары орналасқан элементтерін, ал екіншісіне – бас диагональдан төмен орналасқан элементтерін жазып шығу керек.

```
#include <iostream.h>
#include <conio.h>
main()
{ const row=3, // жолдар саны
       col=row; // бағаналар саны
int i,j,m,n,a[row][col],
    b[row*col],
    c[row*col];
clrscr();
cout << "Жиымды жолдар бойынша енгіз:\n";
for(i=0;i<row;i++)
{ cout << "жол #" << i << " ";
  for(j=0;j<col;j++)
     cin >> a[i][j];
}
n=0;
m=0;
for(i=0;i<row;i++)
  for(j=0;j<col;j++)
    if(j>=i)
      {
       b[n]=a[i][j];
       n++;
      }
      else
      {
       c[m]=a[i][j];
       m++;
      }
cout << "Бас диагональда және одан жоғары
        орналасқан элементтер:\n";
for(i=0;i<n;i++)
  cout << b[i] << "  ";
cout << endl;
cout << " Бас диагональдан төмен орналасқан
       элементтер:\n";
for(i=0;i<m;i++)
  cout << c[i] << "  ";
cout << endl;
```

```
  getch();
 }
```

Программаны орындау нәтижесі:

```
Жиымды жолдар бойынша енгіз:
жол #0 4 5 6
жол #1 8 6 4
жол #2 2 6 9
Бас диагональда және одан жоғары орналасқан элементтер:
```

Жиымдағы элементтердің орындарын алмастыру

Жиымдағы элементтердің орындарын алмастыру кезінде олардың мәндері өзгермейді де, тек *элементтердің реттілігі*, яғни жиым ішіндегі индекстері өзгереді.

Екі элементті бір-бірімен орын алмастыру үшін үшінші бір айнымалы қажет болады, ол элементтердің бірін уақытша есте сақтай тұрады да, *буферлік айнымалы* рөлін атқарады. Мысалы, a жиымындағы үшінші және төртінші элементтердің орнын алмастырайық.

```
buf = a[3]; a[3] = a[4]; a[4] = buf;
```

Матрица жолдарын алмастыру кезінде екі жолдың барлық элементтерін алмастырып шығу керек. Мысалы, матрицаның екінші және үшінші жолдарын алмастыру үшін, буфер ретінде жиымды емес, тек бір айнымалыны ғана пайдалануға болады, өткені оларды кезекпен, бірінен соң бірін ала отырып алмастырамыз.

```
for(j=0;j<m;j++)
  { buf = a[1][j]; a[1][j]=a[2][j]; a[2][j]=buf; }
```

Екі өлшемді матрица жолдарын алмастырудың көрнекі мысалы ретінде *матрицаны транспонирлеуді* айтуға болады. Мұнда қарастырылатын матрицалардың бірінің жолын екіншісінің бағанасы ретінде түрлендіруге тура келеді, бірақ екеуінің де бас диагоналындағы элементтері өзгермейді. Транспонирлеу әрекеті бас диагонал бойынша симметриялы түрде орналаскан матрица элементтерін алмастыруға келіп тіреледі.

Келесі есепте матрица элементтері кездейсок сандармен толтырылады да, оны транспонирлеу әрекеті жүзеге асырылады.

```
/* Кездейсоқ сандар матрицасын транспонирлеу */
#include <iostream.h>
#include <iomanip.h>
#include <conio.h>
#include <stdlib.h>
main()
{
  const row=3, // жолдар саны
      col=row; // бағаналар саны
```

```
int i,j,buf,a[row][col];
clrscr(); randomize();
cout << "Кездейсоқ сандар матрицасы \n";
for(i=0;i<row;i++)
{
  for(j=0;j<col;j++)
     {
       a[i][j]=random(100); // элементтің кездейсоқ мәні
       cout << setw(4) << a[i][j]; // элементті шығару
     }
  cout << endl;
}
    // матрицаны транспонирлеу
for(i=0;i<row;i++);
{ // бас диагоналдан жоғары тұрған жолдар эл-терін қарау
    for(j=i+1;j<col;j++)
     { // бас диагонал бойынша симметриялы түрде
       // орналасқан элементтерді алмастыру
     buf=a[i][j]; a[i][j]= a[j][i]; a[j][i]=buf; }
  }
cout << "Матрицаны транспонирлеу нәтижесі:\n";
for(i=0;i<row;i++)
{
  for(j=0;j<col;j++)
     cout << setw(4) << a[i][j];
     cout << endl;
  }
  getch();
}
```

Матрицаларды өңдейтін негізгі алгоритмдер ретінде бір өлшемді жиымдарды өңдеу кезінде колданылған алгоритмдер саналады. Жалпы матрицаларды өңдейтін барлық алгоритмдерді екі топка бөліп карастыруға болады, олар:

1) матрицаның барлық элементтерін өңдейтін алгоритмдер;
2) матрицаның әр жолы немесе әр бағанасы элементтерін жеке-жеке өңдейтін алгоритмдер.

6.1 Матрицаның барлық элементтерін өңдейтін алгоритмдер

1-мысал. Нақты сандардан тұратын $a_{4,6}$ матрицасы берілген. Мынадай өрнекті $Z=\frac{P1}{|P2|}$ есептеу керек, мұндағы P1 және P2 – сәйкесінше алынған матрицаның оң және теріс элементтерінің көбейтіндісі.

```
/*  a[4][6] матрицасы берілген. z=p1/|p2|мәнін есептеу керек, p1
және p2 - матрицаның  оң және теріс элементтерінің көбейтіндісі
*/
#include <math.h>
```

```
#include <conio.h>
#include <stdio.h>

void  line()   // көлденең сызық сызу функциясы
 {printf("-----------------------------\n");
return;
}
main()
{ static int a[4][6]={ {5,-11,4,-2,5,6},
                       {3,3,-12,-5,7,8},
                       {2,3,-3,14,-9,-3},
                       {-9,3,-6,14,9,-3}
                     };
int i, j;
float p1,p2,z;
clrscr();
printf("Берілген матрица :\n");
line();
for(i=0; i<4; i++)
 {for(j=0; j<6; j ++)
    printf(" %3i ", a[i][j]);
  printf("\n");}
  line();
   /* Матрицаны өңдеу */
  p1 = 1; p2 = 1;
  for(i=0; i<4; i++)
   for(j=0; j<6; j ++)
    {  if (a[i][j]>0)  p1 = p1*a[i][j];
       if (a[i][j]<0)  p2 = p2*a[i][j];
    }
     z = p1/fabs(p2);
  printf("   z = %f\n",z);
  line();
  getch();
}
```

Программаны орындау нәтижесі:

```
Берілген матрица :
-----------------------------
   5  -11    4   -2    5    6
   3    3  -12   -5    7    8
   2    3   -3   14   -9   -3
  -9    3   -6   14    9   -3
-----------------------------
   z = 554.343384
-----------------------------
```

2-мысал.Бүтін сандардан тұратын квадрат $b_{5,5}$ матрицасы берілген. Оның бас диагоналының сол жағында және оң жағында орналасқан нөлге тең элементтері санын анықтап, солардың айырмасының модулін табу керек.

Мынадай белгілеулер енгізейік:

L1 – бас диагоналдың сол жағында (төменінде) орналасқан элементтер саны;

L2 – бас диагоналдың оң жағында (жоғарысында) орналасқан элементтер саны;

L= |L1-L2| – солардың айырмасының модулі.

```
#include <math.h>
#include <conio.h>
#include <stdio.h>
void line()
 {printf("--------------------------\n");
 return;}

main()
{ static int b[5][5]={   {5,0,0,0,0},
                         {0,3,12,0,0},
                         {0,33,13,14,0},
                         {0,23,0,14,0},
                         {35,0,13,14,9},
                      };
  int i,j;
  int L1,L2,L;
  clrscr();
  printf("Берілген матрица:\n");
  line();
  for(i=0; i<5; i++)
   {for(j=0; j<5; j ++)
      printf(" %3i ", b[i][j]);
    printf("\n");}
  line();
L1 = L2 = 0;
for(i=0; i<5; i++)
 for(j=0; j<5; j ++)
  if (b[i][j]==0)
   {if (i>j)  L1 = L1+1;
    if (i<j)  L2 = L2 +1;
   }
L= abs(L1 - L2);
printf(" L = %i ",L);
getch();
}
```

6.2 Екінші типтегі есептер алгоритмдері

1-мысал. Бүтін сандардан тұратын $a_{3,6}$ матрицасы жолдарының алғашқы элементін осы жолдың минималды элементімен алмастыру керек. Нәтижелік $a_{3,6}$ матрицасы элементтерін экранға шығару қажет.

```
/*  a[3][6] матрицасы жолдарының алғашқы элементін осы жолдың
минималды элементімен алмастыру керек. Нәтижелік X матрицасы
элементтерін экранға шығару қажет.*/
#include <math.h>
#include <conio.h>
#include <stdio.h>
void line()
{ printf("-----------------------------\n");
   return; }
main()
{ static int a[3][6]={ {5,-11,4,-2,5,6},
                       {2,3,-3,14,-9,-3},
                       {-9,3,-6,-14,9,-3}
                     };
  int i,j,jmin,min;
  clrscr();
  printf("Берілген матрица:\n");line();
  for(i=0; i<3; i++)
    {for(j=0; j<6; j ++)
       printf(" %3i ", a[i][j]);
     printf("\n");
    }
  line();
for(i=0; i<3; i++)
   {min=+1E6;
for(j=0; j<6; j ++)
     if (a[i][j]<min)
{min=a[i][j];
     jmin=j;
}
 a[i][jmin]=a[i][0]; a[i][0]=min;
}
printf("Өңделген матрица:\n"); line();
for(i=0; i<3; i++)
{for(j=0; j<6; j ++)
  printf(" %3i ", a[i][j]);
 printf("\n");
} line();
```

```
  getch();
}
```

2-мысал. Бүтін сандардан тұратын $a_{3,4}$ матрицасының әрбір бағаналарының арифметикалық орташа мәнін анықтап, оларды бір өлшемді s_4 жиымы ретінде бейнелеу керек.

```
/*  a[3][4] матрицасының әрбір бағаналарының арифметикалық
орташа мәнін анықтап,оларды бір өлшемді s[4] жиымы ретінде
бейнелеу керек. */
//әрбір бағана қосындысы және олардың орташа мәні
#include <conio.h>
#include <stdio.h>
main()
{
  static int a[3][4]={      {5,11,4,2},
                            {3,3,12,5},
                            {2,3,3,14}
                      };
  int i,j;
  float s[4];
  clrscr();
  printf("Берілген матрица :\n");
  printf("-----------------------\n");
for(i=0; i<3; i++)
   {for(j=0; j<4; j ++)
      printf(" %3i ", a[i][j]);
    printf("\n");}
    /* матрицаны өңдеу */
  printf("-----------------------\n");
  printf("");
  for(j=0; j<4; j++)
   { s[j]=0;
for(i=0; i<3; i ++) s[j]+=a[i][j];
printf(" %4.2f",s[j]/3);
   }
 printf("\n-----------------------");
getch();
}
```

Программаны орындау нәтижесі:

```
Берілген матрица :
-----------------------
  5   11    4    2
  3    3   12    5
  2    3    3   14
-----------------------
 3.33 5.67 6.33 7.00
-----------------------
```

Бұл программада s[j] жиымының әрбір элементін есептеуде қабаттасқан екі цикл қолданылған, онда j индексі – сырткы цикл параметрі, ал і индексі – ішкі цикл параметрі. Осы тәсіл матрица элементтеріи бағаналар бойынша өңдеу ісін жүзеге асырады.

3-мысал. Берілген жиымның әрбір жолындағы элементтері қосындыларын және сол қосындылардың орташа мәнін анықтау керек.

```
//әр жол қосындысы және олардың арифметикалық ортасы
#include <conio.h>
#include <stdio.h>
main()
{ static int a[3][4]={
                        {5,3,4,2},
                        {3,3,4,5},
                        {2,3,3,4}
                       };
int i,j,s=0;
float c=0;
clrscr();
for(i=0; i<3; i++)
 {for(j=0; j<4; j ++) s+=a[i][j];
     printf("%i-жол косындысы:%i\n",i+1,s);
  c+=s;
 }
printf("----------------------");
printf("\парифм.ортасы %5/2f", c/4) ;
}
```

4-мысал. Берілген жиымның әрбір бағанадағы элементтері қосындыларын және сол қосындылардың орташа мәнін анықтау керек.

```
// әрбір бағана қосындылары мен олардың орташа мәнін анықтау
#include <conio.h>
#include <stdio.h>
main()
{
  static int a[3][4]={
                        {5,11,4,2},
                        {3,3,12,5},
                        {2,3,3,14}
                      };
int i,j,s; float c;
clrscr();  printf("Берілген матрица :\n");
printf("-----------------------\n");
for(i=0; i<3; i++)
 {for(j=0; j<4; j++)
```

```
    printf(" %3i ", a[i][j]);
  printf("\n");}
  /* Матрицаны өңдеу */
printf("-----------------------\n");
printf("s=");
for(j=0; j<4; j++)
 { s=0;
   for(i=0; i<3; i ++) s+=a[i][j];
     printf("%3i  ",s); // косынды s
   c+=s;
   }
 printf("\n-----------------------");
 printf("\n арифм. ортасы %5.2f",c/4);
 getch();
}
```

5-мысал. Берілген a_{44} жиымының бас диагоналындағы элементтерді нолге, ал қосалқы диагоналындағы элементтерді бірге теңестіру программасы.

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
main()
{
  int a[4][4]={
                {6,8,9,2},
                {5,3,4,2},
                {3,3,4,5},
                {2,3,3,4}
               };
  int i,j;
  clrscr();
  printf("Берілген матрица :\n");
  for(i=0; i<4; i++)
   {for(j=0; j<4; j ++)
      printf(" %2i ", a[i][j]);
    printf("\n");}
    /* Матрицаны өңдеу */
  for(i=0; i<4; i++)
       a[i][i]=0;
  for(i=0; i<4; i++)
    for(j=0; j<4; j ++)
       if (i+j==3)  a[i][j]=1;
   /* Өңделген матрица элементтерін шығару */
  printf("Нәтижелік матрица :\n");
```

```
   for(i=0; i<4; i++)
    {for(j=0; j<4; j ++)
      printf(" %2i ", a[i][j]);
     printf("\n");}
getch();
}
```

Программаны орындау нәтижесі:

```
Берілген матрица :
  6   8   9   2
  5   3   4   2
  3   3   4   5
  2   3   3   4
Нәтижелік матрица :
  0   8   9   1
  5   0   1   2
  3   1   0   5
  1   3   3   0
```

6-мысал. Берілген $a_{3,3}$ жиымының бас диагоналынан төмен орналасқан элементтер қосындысын анықтау программасы.

```
#include <conio.h>
#include <stdio.h>
main()
{ static int a[3][3]={ {0,1,2},
                       {3,4,5},
                       {6,7,8}
                     };
  int i,j,r,s;
  clrscr();
   printf("Енгізілген a[3][3] жиым элементтері:\n");
  for(i=0; i<3; i++)
   {for(j=0; j<3; j++)
      printf(" %2i",a[i][j]);
   printf("\n");
  }
  /* бас диагоналдан төмен орналасқан элементтер
     қосындысын анықтау */
  s=0;
  for(i=0; i<3; i++)
  for(j=0; j<3; j++)
    if (j<i) s+=a[i][j];
```

```
    printf("\ns=%i",s);
    getch();
}
```

7-мысал. Берілген $a_{3,3}$ жиымының әрбір жолындағы элементтерді өсуі бойынша реттеп орналастыру программасы.

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
main()
{ static int a[3][3]={{8,7,6},
                      {5,4,3},
                      {2,1,0}
                      };
  int i,j,r,s,n=3,amin,m,k;
  clrscr();
  printf("Берілген матрица:\n");
  for(i=0; i<3; i++)
   {for(j=0; j<3; j++)
     printf(" %2i",a[i][j]);
    printf("\n");
   }
   for(i=0; i<n; i++)    //жолды таңдау
  {                      //мин тауып алмастыру
   for(k=0; k<n-1; k++)
    { amin=a[i][k];m=k;
     for(j=k+1; j<n; j++)
       if (a[i][j] < amin){amin=a[i][j]; m = j;}
     a[i][m]=a[i][k]; a[i][k]=amin;
    }
  }
printf("\nНәтижелік матрица:\n");
for(i=0; i<3; i++)
   {for(j=0; j<3; j++)
     printf(" %2i",a[i][j]);
    printf("\n");
   }
getch();
```

8-мысал. Берілген $a_{3,3}$ жиымының әрбір бағанасындағы элементтерді өсуі бойынша реттеп орналастыру программасы.

```
#include <stdio.h>
#include <stdlib.h>
```

```
#include <conio.h>
main()
{
  static int a[3][3]={ {8,7,6},
            {5,4,3},
            {2,1,0}
                       };
  int i,j,r,s,n=3;
  int amin;
  int m,k;  clrscr();
  printf("Берілген матрица:\n");
  for(i=0; i<3; i++)
   {for(j=0; j<3; j++)
      printf(" %2i",a[i][j]);
    printf("\n");
   }

for(j=0; j<n; j++)    //бағананы таңдау
{for(k=0; k<n-1; k++) //мин тауып алмастыру
{ amin=a[k][j];m=k;
 for(i=k+1; i<n; i++)
if (a[i][j] < amin)
     {amin=a[i][j]; m = i;}
a[m][j]=a[k][j]; a[k][j]=amin;
   }
 }
printf("\nСұрыпталған матрица элементтері:\n");
for(i=0; i<3; i++)
   {for(j=0; j<3; j++)
     printf(" %2i",a[i][j]);
    printf("\n");
   }
getch();
}
```

Программаны орындау нәтижесі:

```
Берілген матрица:
  8  7  6
  5  4  3
  2  1  0

Сұрыпталған матрица элементтері:
  2  1  0
  5  4  3
  8  7  6
```

9 мысал. Келесі түрдегі A(10, 10) матрицасын құрастыру керек:

```
1    0    0    .    .    0
2    1    0    .    .    0
3    2    1    .    .    0
.    .    .    .    .    0
.    .    .    .    .    0
10   9    8    .    .    1
```

```
#include <stdio.h>
#include <conio.h>
#define N 10
void main ()
{
  static int a[N][N]; // статикалық жиым нөлдермен
                      // инициалданады
  int i,j;
  clrscr();
  for(i=0; i<N; i++)
    for(j=0; j<N; j++)
      if(i>=j) a[i][j]=i-j+1; // бас диагоналдан төмен
                              // болу шарты
     // матрицаны шығару
  for(i=0; i<N; i++)
   {
     for(j=0; j<N; j++) printf("%3d",a[i][j]);
     putchar('\n');
   }
}
```

Егер матрицаларды толтыру әрекеті келесі түрде жазылса, циклдің қайталану санын азайтуға және IF операторында тексеруден құтылуға болады:

```
for (i=0; i<N;    i++)
for (j=0; j<=i;   j++)
a[i][j]=i-j+1;
```

Матрицаларды көбейту

Екі өлшемді жиымдарды көбейту есептерін карастырайық. Келесі мысалда үш сатушы төрт түрлі тауар сатады. Сатылатын тауарлар саны 1-кестеде көрсетілген.

1 кесте - Сатушылар мен тауарлар

Сатушы	Тауар			
	№ 1	№ 2	№ 3	№ 4
№ 1	5	2	0	10
№ 2	3	5	2	5
№ 3	20	0	0	0

2-кестеде әрбір тауар бағасы мен комиссияға бөлінетін қаржы мөлшері көрсетілген .

2 кесте - Тауар бағасы мен комиссияға шығатын қаржы мөлшері

Тауар	Бағасы	Комиссиялығы
№ 1	1,5	0,2
№ 2	2,8	0,4
№ 3	5	1
№ 4	2	0,5

3-кестеде сатылған тауардан түскен қаржы мөлшерін есептеу жолы көрсетілген.

3 кесте – Сатудан түскен пайда

Са-	№ 1	5 x 1,5 + 2 x 2,8 + 0 x 5 + 10 x 2 = 33,1
ту-	№ 2	3 x 1,5 + 5 x 2,8 + 2 x 5 + 5 x 2 = 38,5
шы	№ 3	20 x 1,5 + 0 x 2,8 + 0 x 5 + 0 x 2 = 30

Комиссияға шығатын қаржы мөлшерін есептеу тәсілі 4 кестеде берілген. Осы есептеулер матрицаларды көбейту аркылы 5 кестеде көрсетілген түрде орындалады.

4 кесте – Комиссияға шығатын қаржы мөлшері

Са-	№ 1	5 x 0,2 + 2 x 0,4 + 0 x 1 + 10 x 0,5 = 6,8
ту-	№ 2	3 x 0,2 + 5 x 0,4 + 2 x 1 + 5 x 0,5 = 7,1
шы	№ 3	20 x 0,2 + 0 x 0,4 + 0 x 1 + 0 x 0,5 = 4

5 кестедегі А бағаналарының саны В жолдарының санына тең болуы тиіс, ал шығатын нәтижедегі жолдар саны Ажолдарымен бірдей болады да, бағаналар саны В бағаналары санымен бірдей болады.

5 кесте – Матрицаларды көбейту

		1	2	3	4				1	2				1	2
A	1	5	2	0	10		B	1	1,5	0,2		C	1	33,1	6,8
A	2	3	5	2	5	x	B	2	2,8	0,4	=	C	2	38,5	7,1
A	3	20	0	0	0		B	3	5	1		C	3	30	4
							B	4	2	0,5					

Келесі программада бастапқы берілген «Сатушы-Тауар» және «Бағасы-Комиссиялығы» жиымдары енгізіледі де, олар көбейтіледі, соңында үш матрица да экранға шығарылады.

```
// матрицаларды көбейту
```

```
#include <iostream.h>
#include <iomanip.h>
#include <conio.h>
main()
{
 const mmax=5,nmax=5; // көбейткіш-жиымдардың макс.мөлшерлері
 float a[mmax][nmax],b[mmax][nmax],c[mmax][nmax]; // берілген
                                                  // жиым және нәтиже
int m1,n1,m2,n2,                          // жиымның нақты мөлшерлері
     i,j,k;
 clrscr();
cout << "Сатушылар-тауарлар матрицасын енгізіңіз:\n";
do
 {
    cout << "Сатушылар санын енгізіңіз <=" << mmax << ": ";
    cin >> m1;
    cout << "Тауарлар санын енгізіңіз <=" << nmax << ": ";
    cin >> n1;
 }
while(!(m1<=mmax)&&(n1<=nmax));
for(i=0;i<m1;i++)
  for(j=0;j<n1;j++)
    { cout << "Тауарлар саны "<< j <<" сатушы: "<< i <<" ";
      cin >> a[i][j];
    }
cout << "\nЕнгізілген сатушылар-тауарлар матрицасы:\n";
            // 1-көбейткішті шығару
for(i=0;i<m1;i++)
  { for(j=0;j<n1;j++)
      cout << setprecision(3) << setw(7) << a[i][j];
    cout << endl;
  }
cout << "Бағасы-комиссиясы матрицасын енгізіңіз\n";
m2=n1;
n2=2;
for(i=0;i<m2;i++)
  { cout << "тауар бағасы мен комиссиясы "<< i <<" босорын арқылы ";
    for(j=0;j<n2;j++)
      cin >> b[i][j];
  }
cout << "\nЕнгізілген баға-комиссиялар матрицасы:\n";
 // 2-көбейткішті шығару
for(i=0;i<m2;i++)
```

```
  { for(j=0;j<n2;j++)
      cout << setprecision(3) << setw(7)<<b[i][j];
    cout << endl;
  }
 // матрица көбейтіндісін есептеу
for(i=0;i<m1;i++)
   for(j=0;j<n2;j++)
     { c[i][j]=0.0;
       for(k=0;k<n1;k++)
         c[i][j]=c[i][j]+a[i][k]*b[k][j];
     }
 cout << "\nНәтиже: сатушылар пайдасы мен комиссиясы:\n";
  // нәтижені шығару
 for(i=0;i<m1;i++)
   { for(j=0;j<n2;j++)
       cout << setprecision(3)<<setw(7)<<c[i][j];
     cout << endl;
   }
 getch();
}
```

Программаны орындау нәтижесі:

```
Сатушылар санын енгізіңіз <=5: 3
Тауарлар санын енгізіңіз <=5: 4
Тауарлар саны 0 сатушы: 0 5
Тауарлар саны 1 сатушы: 0 2
Тауарлар саны 2 сатушы: 0 0
Тауарлар саны 3 сатушы: 0 10
Тауарлар саны 0 сатушы: 0 3
Тауарлар саны 1 сатушы: 0 5
Тауарлар саны 2 сатушы: 0 2
Тауарлар саны 3 сатушы: 0 5
Тауарлар саны 0 сатушы: 0 20
Тауарлар саны 1 сатушы: 0 0
Тауарлар саны 2 сатушы: 0 0
Тауарлар саны 3 сатушы: 0 0

Енгізілген сатушылар-тауарлар матрицасы:
      5       2       0      10
      3       5       2       5
     20       0       0       0
Бағасы-комиссиясы матрицасын енгізіңіз тауар бағасы мен
комиссиясы 0 босорын арқылы
```

```
Сатушылар санын енгізіңіз <=5: 3
Тауарлар санын енгізіңіз <=5: 4
Тауарлар саны 0 сатушы: 0 5
Тауарлар саны 1 сатушы: 0 2
Тауарлар саны 2 сатушы: 0 0
Тауарлар саны 3 сатушы: 0 10
Тауарлар саны 0 сатушы: 0 3
Тауарлар саны 1 сатушы: 0 5
Тауарлар саны 2 сатушы: 0 2
Тауарлар саны 3 сатушы: 0 5
Тауарлар саны 0 сатушы: 0 20
Тауарлар саны 1 сатушы: 0 0
Тауарлар саны 2 сатушы: 0 0
Тауарлар саны 3 сатушы: 0 0

Енгізілген сатушылар-тауарлар матрицасы:
       5       2       0      10
       3       5       2       5
      20       0       0       0
Бағасы-комиссиясы матрицасын енгізіңіз
тауар бағасы мен комиссиясы 0 босорын арқылы 1.5 0.2
тауар бағасы мен комиссиясы 0 босорын арқылы 2.8 0.4
тауар бағасы мен комиссиясы 0 босорын арқылы 5 1
тауар бағасы мен комиссиясы 0 босорын арқылы 2 0.5

Енгізілген баға-комиссиялар матрицасы:
     1.5     0.2
     2.8     0.4
       5       1
       2     0.5

Нәтиже: сатушылар пайдасы мен комиссиясы:
    33.1     6.8
    38.5     7.1
      30       4
```

Мұндағы mmax және nmax тұрақтылары матрицалардың ең үлкен өлшемін енгізіп, олар өзгерген жағдайда бір-ақ жерде мәндерін беріп, программа бойынша пайдалана беруге болады. Программа кемшілігі болып оның ұқсас матрицаларды үш рет экранға шығару әрекетінің қайталатыны саналады, бірақ оны келесі функция тақырыбынан соң бір-ақ рет жазып ықшамдау мүмкіндігі бар.

6.3 Екі өлшемді жиымдармен жұмыс істеу кезінде нұсқауыштарды қолдану

$A_{3,2}$ жиым берілген болсын. Олар – бүтін сандар, яғни

```
int a[3][2];
int *pri;
```

```
pri = a; // бұл pri=a[0][0] деген сөз
```

а – жиымның аты немесе a[0][0] элементінің адресі a=&a[0][0];
pri нұсқауышына 1-ді қоссақ, pri+1 деген нұсқауыш a[0][1] элементіне сілтейді. Бұл жиымды қарастырсақ, келесі теңдеулер дұрыс болып шығады:

```
pri   == &a[0][0];
pri+1== &a[0][1];

pri+2== &a[1][0];
pri+3== &a[1][1];

pri+4== &a[2][0];
pri+5== &a[2][1];
```

Екі өлшемді жиым бір өлшемді жиымдардан құрастырылған жиым ретінде қарастырылуы мүмкін. Яғни берілген екі өлшемді жиым үш жолдан тұрады, ал әрбір жол екі элементтен тұратын жиым болып табылады.

Бірінші жолдың аты – a[0],
екінші жолдың аты – a[1],
үшінші жолдың аты – a[2].

Жиымның аты берілген жиымның нұсқауышы болып табылады, яғни ол жиымның 1-элементіне сілтейді.

Екі өлшемді жиымның осы қасиеті бір өлшемді жиымға арналған функцияны екі өлшемді жиыммен жұмыс істеуге мүмкіндік береді.

1-мысал. Төмендегі b[3][4] матрицасы жолдарының арифметикалық ортасын табатын программада функция қолданылған.

```
/* b[3][4]  матрицасы жолдарының қосындысы */
#include <conio.h>
#include <stdio.h>
float f1(int x[], int n)
 { int k; float s;
   for(k=0,s=0;k<n; k++)
     s+=x[k];
   return(s/n);
 }
main()
{ int i,b[3][4]={
                 {6,4,3,3},
                 {7,5,3,3},
                 {8,4,2,6}
               };
  clrscr();
  for(i=0;i<3;i++)
     printf("%d жолының орташа мәні: %f\n",f1(b[i],4));
  getch();
}
```

2-мысал. Төменде көрсетілген түрдегі A(10, 10) матрицасын құрастырып шығару керек:

```
1   0   0   .   .   0
2   1   0   .   .   0
3   2   1   .   .   0
.   .   .   .   .   0
.   .   .   .   .   0
10  9   8   .   .   1
```

```
#include <stdio.h>
#include <conio.h>
#define N 10
void main ()
{
  staticint a[N][N]; // статикалық жиымды нөлдермен
                     // инициалдау
  int i, j;
  clrscr();
  for(i=0; i<N; i++)
  for(j=0; j<N; j++)
  if(i>=j) a[i][j]=i-j+1; // Бас диагоналдан төмен
                          // түсу шарты
          // Матрицаны шығару
 for(i=0; i<N; i++)
  {
 for(j=0; j<N; j++) printf("%3d",a[i][j]);
 putchar('\n');
  }
}
```

Бұл программаны орындау нәтижесі:

```
  1   0   0   0   0   0   0   0   0   0
  2   1   0   0   0   0   0   0   0   0
  3   2   1   0   0   0   0   0   0   0
  4   3   2   1   0   0   0   0   0   0
  5   4   3   2   1   0   0   0   0   0
  6   5   4   3   2   1   0   0   0   0
  7   6   5   4   3   2   1   0   0   0
  8   7   6   5   4   3   2   1   0   0
  9   8   7   6   5   4   3   2   1   0
 10   9   8   7   6   5   4   3   2   1
```

Егер матрицаларды толтыру фрагменті келесі түрде жазылса, циклдің қайталану санын азайтуға және IF операторында тексеруден құтылуға болады: for

```
(i=0; i<N;    i++)
            for (j=0; j<=i;   j++)
                a[i][j]=i-j+1;
```

3-мысал. Екiөлшемдi $C_{10,10}$ жиымын кездейсоқ сандармен толтырып, жиымның әрбір жолын өсу ретімен орналастыратын программа құрындар.

```
#include<iostream.h>
#include<math.h>
#include<stdlib.h>
#define n 10
void main()
{int i,j,k,t,a[n][n];
 cout<<"Kezdeisok engizilgen [10;10] olwemdi massiv:\n\n";
 for(i=0;i<n;i++)
 { for(j=0;j<n;j++)
   {   a[i][j]=rand()%100-50;
       cout<<a[i][j]<<"\t"; }
   cout<<endl;
 }
cout<<endl<<endl;
for(i=0;i<10;i++)
 { for(j=0;j<n;j++)
   { for(k=0; k<n-j-1; k++)
       if(a[i][k]>=a[i][k+1])
         { t=a[i][k];
           a[i][k]=a[i][k+1];
           a[i][k+1]=t;      }
   }
 }
cout<<"Suriptalgan massiv:\n\n";
for(i=0;i<n;i++)
```

```
{for(j=0;j<n;j++)
cout<<a[i][j]<<"\t";
cout<<endl;
} }
```

Бақылау сұрақтары

1. *Екі өлшемді жиымды – матрицаны сипаттау, бастапқы мәндерді тағайындау тәсілдері.*
2. *Екі өлшемді жиымды инициалдауды қабаттасқан циклдер арқылы орындау.*
3. *Матрица элементтерін толық өңдейтін алгоритмдер.*
4. *Матрицалардың көрсетілген элементтерін өңдеу тәсілдері.*
5. *Матрицалардың диагоналына байланысты орналасқан элементтерін өңдеу жолдары.*
6. *Матрицалардың жолдарының және бағаналарының ең үлкен (кіші) элементтерін, қосындыларын, көбейтінділерін табу.*
7. *Матрицалардың жолдарында және бағаналарында орналасқан элементтерді өсуі (кемуі) бойынша реттеу алгоритмдері.*

Тапсырмалар

1. Екіөлшемді $A(10,10)$ жиымда әрбір жол үшін оң элементтердің қосындысын есептеп шығарындар.
2. Екіөлшемді $A(10,10)$ жиымындағы оң элементтердің санын есептеп шығарындар.
3. Екіөлшемді $A(10,10)$ жиымындағы ең үлкен элементті анықтаңдар.
4. Екіөлшемді $A(10,10)$ жиымы берілген. A жиымының әрбір жолындағы элементтердің көбейтіндісіне тең болатын бірөлшемді B жиымын алыңдар.
5. Екіөлшемді $A(10,10)$ жиымы берілген. A жиымының әрбір жолындағы ең үлкен элементтің мәндеріне тең болатын бірөлшемді B жиымын алыңдар.
6. Екіөлшемді $A(10,10)$ жиымы берілген. A жиымының әрбір жолындағы теріс элементтің мәндеріне тең болатын бірөлшемді B жиымын алыңдар.
7. Екіөлшемді $A(10,10)$ жиымында теріс элементтері бар жолдардың санын есептеп шығарындар.
8. Екіөлшемді жиым жолдарындағы ең үлкен элементтердің қосындысын есептеп шығарындар.
9. Екіөлшемді жиым әрбір бағанадағы (тік жолдағы) элементтердің қосындысын есептеп шығарындар.
10. Екіөлшемді жиымның әрбір жолындағы теріс сандардың санын, қосындысын және арифметикалық ортасын есептеп шығарындар.
11. Екіөлшемді $A(4,4)$ жиымындағы индекстерінің қосындысы 4-ке тең болатын элементтерінің қосындысын есептеп шығарындар.
12. Екіөлшемді $A(4,7)$ жиымындағы оң элементтердің арифметикалық ортасын және нөлге тең элементтердің санын есептеп шығарындар.

13. Екіөлшемді *A*(10,10) жиымының бас диагоналы бойындағы элементтердің ең үлкенін табыңдар.
14. Екіөлшемді *A*(10,10) жиымының қосымша диагоналы бойындағы ең үлкен элементті табыңдар.
15. Екіөлшемді *A*(10,10) жиымының қосымша диагоналы бойындағы ең үлкен элементті табыңдар.
16. Екіөлшемді *A*(4,7) жиымының әрбір жолындағы берілген диапазонда (a;b) жататын элементтердің арифметикалық ортасын есептеп шығарыңдар.
17. Екіөлшемді *A*(10,10) жиымында элементтердің арифметикалық ортасы жиымның барлық элементтерінің арифметикалық ортасынан кем болатын бағанның нөмірлерін анықтаңдар.
18. Екіөлшемді *A*(10,10) жиымында 3- және 1-жолдардың орындарын ауыстырыңдар.
19. Екіөлшемді *A*(7,7) жиымында бас диагоналдағы элементтерді әрбір жолдың ең үлкен мәнімен алмастырыңдар.
20. Екіөлшемді *A*(10,10) жиымында бас диагоналдан жоғары және төмен орналасқан элементтердің қосындысын есептеп шығарыңдар.
21. Екіөлшемді жиымда берілген жолдың ең үлкен элементінің мәнін және нөмірін анықтаңдар.
22. Екіөлшемді жиымда әр баған үшін берілген бағанның мәнінен кіші болатын элементтердің арифметикалық ортасын есептеп шығарыңдар.
23. Бүтін сандардан құралған 10X10 матрицасының ең үлкен элементінің жолы мен баған нөмірін шығарып беретін программа жазыңдар.
24. *A*(5,5) жиымы және *k* саны берілген. Әрбір жолдың элементтерін осы жолдағы бас диагоналда орналасқан диагоналдық элементке бөліңдер.
25. *A*(10,10) жиымы берілген. Осы жиымның бас диагоналының элементтерінен тұратын бірөлшемді жиым құрыңдар.
26. *A*(10,10) жиымы берілген. *k*- және 1-жолдардың орындарын ауыстырыңдар.
27. *A*(10,10) жиымы берілген. Одан бірөлшемді жиым құрастырыңдар.
28. Бүтін санды *x* [0...5, 0...4] жиымы берілген. Оның 5-тен кіші барлық элементтерін 111 санымен ауыстырыңдар.
29. Бүтін санды *B* [0...4, 0...3] жиымы берілген. Оның элементтерін олардың квадраттарымен ауыстырыңдар.
30. Нақты сандар жиымы *A*[0...5, 0...3] берілген. Оның теріс элементтерінің индекстерін басып шығарыңдар.
31. Екіөлшемді бүтін санды *A*[0..10, 0..7] жиымын құрыңдар және онын екі так санды индекстерінің барлық элементтерінің қосындысын табыңдар.
32. Бүтін санды *A* [0..10, 0..7] жиымы берілген. Жиымның 5-ке қалдықсыз бөлінетін элементтерінің қосындысын табыңдар.
33. 3x3 матрицаның бүтін сандық элементтерін пернетақтадан енгізіңдер және әрбір баған элементтерінің қосындысын есептеп шығарыңдар.
34. Бүтін санды *B*[0..5,0..5] жиымы берілген. Оның диагоналдарынан сол және оң жақта орналасқан элементтерін аныктаңдар.
35. Бүтін санды *B* [0..5, 0..5] жиымы берілген. Диагоналдың оң жақтағы элементтерінің қосындысын, сол жақтағы элементтерінің көбейтіндісін табыңдар.

36. Бүтін санды B [0..5, 0..5] жиымы берілген. Жиымның ең үлкен элементін табындар және оның диагоналының қай жағында орналасқаны туралы хабарды экранға шығарындар.
37. Бүтін санды B [0..5, 0..5] жиымы берілген. Жиымның ең кіші элементін табындар және оның диагоналдың қай жағында орналасқаны туралы хабарды экранға шығарындар.
38. Бүтін санды B [0..5,0..5] жиымы берілген. Жиымның басдиагоналынан жоғары орналасқан элементтердің қосындысын табындар.
39. Бүтін санды B [0..5,0..5] жиымы берілген. Жиымның басдиагоналынан төмен орналасқан элементтердің көбейтіндісін есептеп шығарындар.
40. Бүтін санды B [0..5,0..5] жиымы берілген. Жиымның басдиагоналан төмен орналасқан теріс таңбалы элементтерінің санын табындар.
41. Бүтін санды B[0..5,0..5] жиымы берілген. Жиымның басдиагоналынан жоғары орналасқан оң таңбалы элементтерінің санын табындар.
42. Бес цехтың әркайсысының 4 бөлімшесіндегі барлық шикізат туралы мәлімет кестесі берілген. Шикізаты ең аз цехтың нөмірін анықтандар.
43. A[0..3, 0..15] жиымы берілген. Оның ішінде өзара тең екі элементтің индекстерін басып шығарындар.
44. a_1; a_2, a_3 сандары берілген. Элементтері $B[i,j] = a_i - 3a_j$ болатын бүтін санды B [0..3, 0..3] жиымын анықтандар.
45. Нақты a_1, a_2, ... , a_{10}, b_1, b_2, ... , b_{20} сандары берілген. Элементтері $a_{i,j} = i + 2j$ болатын бүтін санды A [0..10, 0..12] жиымын алындар.
46. Өлшемдері 5*5 матрицаның ең кіші элементінен кейін орналасқан барлық элементтерінің қосындысын есептеңдер.
47. Нақты [0..7, 0...7] жиымын алындар, оның 1-жолы $a_{1j} = 2j + 3$ формуласымен, 2-жолы $a_{2j} = j + 3/(2+j)$ формуласымен беріліп, содан кейінгі әрбір жол алдыңғы екі жолдың қосындысына тең болатын болсын.
48. Натурал n саны берілген. Егер $a_{i,j} = \sin(i+j/2)$ болса, A[0..n, 0..n] жиымында канша оң элемент болатындығын анықтандар.
49. Бүтін санды A[0..4, 0..5] жиымы берілген. Әрбір бағананың арифметикалық ортасын табындар.
50. Барлық элементтері нөлге тең емес $n..m$ өлшемді нақты жиым берілген. Бұл жиымның модулі бойынша ең үлкен элементіне басқа барлық элементтерін бөлу арқылы жаңа жиым алындар.
51. Бүтін санды A[0..4, 0..5] жиымы берілген. Соңғы жолдан басқа жолдарды әрбір элементі бойынша азайту аркылы матрицаны түрлендіріңдер.
52. Екіөлшемді C жиымының әрбір жолын өсу ретімен орналастыратын программа құрындар.
53. $m..n$ өлшемді матрицаның әрбір жолын кему ретімен орналастыратын программа құрындар.
54. Бүтін санды A[0..4, 0..5] жиымы берілген. Құрылымында кем дегенде бір рет 10-ға тең элементі бар жолдардың нөмірін анықтандар.
55. $m..n$ өлшемді матрицаның әрбір бағанасын өсу ретімен орналастыратын программа құрындар.

56. *A*[0..5, 0..5] жиымы берілген. Бұл жиымның әрбір жолының элементтерін диагоналдың сол жағында орналасқан элементтерге бөлу жолымен алынған жаңа жиым құрындар.
57. *A*[0..5, 0..6] жиымы берілген. Оның бірінші және соңғы жолдарының орындарын алмастырыңдар.
58. A[0..5, 0..6] жиымы берілген. Оның бірінші және соңғы бағаналарының орнын алмастырындар.
59. Тікбұрышты матрица берілген. Элементтерінің қосындысы ең үлкен болатын жолды табыңдар.
60. Тікбұрышты матрица берілген. Элементтерінің қосындысы ең үлкен болатын бағананы табыңдар.
61. Өлшемі 4x8 болатын бүтін сандар жиымындағы барлық жұп нөмірлі элементтерінің қосындысын табыңдар.
62. Бүтін сандардың 5x5 өлшемді жиымындағы бас диагоналда орналасқан барлық элементтердің қосындысын табыңдар.
63. Бүтін сандардың 7x4 өлшемді жиымындағы максимал элементтің жолы мен бағана нөмірлерін табыңдар.
64. Бүтін сандардың 6x5 екіөлшемді жиымы бар. Элементтерінің арифметикалық орта мәні максимал болатын жолдың нөмірін табыңдар.
65. Бүтін сандардын 5x9 өлшемді жиымында бірдей нөмірлі жол мен бағананың орнын алмастырыңдар.
66. Екіөлшемді жиымның элементтері сиқырлы квадрат (сиқырлы квадратта барлық вертикаль, горизонталь және екі диагональ бойынша сандардың қосындысы бірдей болады) құрайтындығын тексеретін программа құрындар.
67. Матрицаның элементтері қосалқы диагональ бойынша симметрия құра отырып орын алмастыратын программа құрындар .
68. Екіөлшемді *k* жиымынын бағаналарын циклді түрде, оның *i*-ші бағанасын *i* + 1 бағанасымен алмастыра отырып, соңғы бағана бірінші болып орналасатындай деңгейге жеткізетін программа құрындар.
69. Екіөлшемді А жиымының нөлге тең элементтері жоқ жолдарының оң элементтерінің қосындысын есептеп шығаратын программа құрындар.
70. Квадрат пішінді кестенің ең кіші элементін анықтап, соған сәйкес жол мен бағана элементтерінің орындарын алмастырыңдар.
71. Келесі түрдегі өлшемі (10,10) квадрат матрицаны құрып экранға шығару керек:

0	0	.	.	0	1
0	0	.	.	2	0
.	.	.	.	.	.
.	.	.	.	.	.
0	9	.	.	0	0
10	0	.	.	0	0

72. 15-ке қалдықсыз бөлінетін, бірақ 30-ға бөлінбейтін үш таңбалы сандардың қачшаекенін тауып, соларды экранға шығару.
73. 70 студенттің бойы өлшеніп алынған мәліметтер В жиымында жазылған. Бойлары 180 см-ден аспайтын студенттердің санын анықтап, олардың бойларын С жиымына орналастырындар.
74. Кассада тек үш және бес теңгелік қағаз ақшалар ғана бар (бұл 2000 жылдары болған). 7 теңгеден артық кез келген соманы осындай қағаз ақшалармен "төлейтін" программа құру керек.
75. 0 мен 1 сандары BIT(N) жиымының элементтері болып табылады. Жиымның барлық нөлдерін жиымның басына, ал бірлерін жиымның соңында орналастырып реттейтін программа құрындар. Қосымша жиым ашуға болмайды.
76. Элементтері келесі сандар болатын IM(100) жиымын құру қажет.

1, -1,2,-2, ..., 50,-50.

77. Матрицаның қайқы (ершік) нүктелерін табыңыз (матрица жолының минимумы, ал бағанының максимумы болып табылатын элемент қайқы нүкте деп аталады).
78. Төртбұрыштың төбелері мен тағы бір нүктенің координаттары берілген. Осы нүктенің төртбұрыш ішінде немесе сыртында орналасқандығын анықтау керек.
79. Берілген күн-ай мерзімі (дата) бойынша апта күнінің атын (жұма, сенбі, ...) анықтайтын бір алгоритм төмендегідей түрде жүзеге асырылады:

1) жыл санын екіге *N*1 және *N*2 бөлу керек:

*N*1–жылдың екі алдыңғы цифры, *N2* –жылдың екі соңғы цифры;

2) жылдың кәбисә (егер*N2=0*және *N*1 4-ке бөлінетін болса немесе *N2*≠0 және 4-ке бөлінетін болса, жыл кәбисә болып табылады да,қалған жағдайларда–жыл кәбисә емес) немесе карапайым жыл екендігінанықтау.

3) <жыл, ай кодын> есептеу қажет:

$$jakb = \begin{cases} 1, \text{ егер ай – кәбисә жылдың қаңтар немесеақпан айы болса;} \\ 2, \text{егер ай – кәбисә емес жылдың қаңтар немесеақпан айы болса;} \\ 0 – \text{қалған барлық жағдайларда;} \end{cases}$$

4) <күн кодын> есептеу:

$kk = [365.25 * N2] + [30.56 * m] + id + jak$,

мұнда[..] –санның бүтін бөлігін алу операциясы; *m*–айнөмірі; *id*–күні.

5)(*kk*+3) мәнін 7-ге бөлгендегі калдықты – *dn* есептеу керек.

Егер калдык *dn* = 0 болса, онда күн–жексенбі,

dn = 1болса, күн–дүйсенбі,

...

dn= 6болса, күн– сенбі.

Осы алгоритмді пайдаланып, М1-жылдан М2-жылға(М2≥М1) дейінгі жұма күндерінің тізімін беретін программа құру қажет.

80. *a(n)* жиымының элементтерін *k* позицияға оңға карай жылжыту қажет. Жиымның шекарасынан шығып кеткен элементтер жоғалып кетеді. Жиымның бос орындарын нөлдермен толтыру керек.
81. Өлшемдері k<20 квадрат матрицасы берілген. Бас диоганальда орналаскан элементтерінің қосындысын табу қажет.

Зертханалық жұмысқа арналған тапсырмалар

А. Әр есептің блок-схемасын және соған сәйкес программасын құрып, компьютерде сол программаның нәтижесін алу керек.

В. Екі блок-схема Word программасындағы *Сурет салу-Автофигуралар-Блок-схемалар* мүмкіндігімен сызылып көрсетілуі тиіс.

С. Берілген жиым элементтері мен нәтижелік элементтерді экранға шығарып, нәтижелерді жазып алу қажет немесе нәтижені бірден файлға жазу керек.

1 нұсқа

1. Екіөлшемді жиымның – $a_{2,5}$ ең үлкен элементін және оның индексін табыңдар.
2. *A*[5][4] жиымы берілген. Оның бірінші және соңғы жолдарының орындарын алмастырыңдар.
3. A[5][5] жиымы берілген. Бұл жиымның әрбір элементін бас диагональдің арифметикалық ортасына бөлу жолымен алынған жаңа жиым құрыңдар.

2 нұсқа

1. Екіөлшемді жиымның – $c_{4,5}$ әрбір жолдарының қосындыларын табыңдар.
2. C[5][5] өлшемді матрицаның жолдарының минимумдарының арифметикалық ортасын табатын программа құрыңдар.
3. Бүтін санды A[4][5] жиымы берілген. Оның 10-ға тең элементтерінің нөмірлерін анықтаңдар.

3 нұсқа

1. Екіөлшемді *A*(4,7) жиымындағы оң элементтердің арифметикалық ортасын есептеп шығарындар.
2. Бүтін санды *B*[5][5] жиымы берілген. Жиым бас диагоналінің сол жағынан жоғары орналаскан оң таңбалы элементтерінің санын табыңдар.
3. Бүтін санды B[4][3] жиымы берілген. Оның бірінші және және үшінші жолдары элементтерін солардың квадраттарымен ауыстырыңдар.

4 нұсқа

1. Екіөлшемді *A*(4,4) жиымының бас диагоналі бойындағы элементтердің қосындысын табыңдар.
2. Бүтін сандардың екі өлшемді C(3,8) жиымындағы жолдардың максимал элементтерінің арасындағы минимал элементті табыңдар.
3. Бүтін санды A[4][5] жиымы берілген. Әрбір бағананың арифметикалық ортасын табыңдар.

5 нұсқа

1. Екіөлшемді *A*(4,4) жиымының қосымша диагоналі бойындағы ең үлкен элементті табыңдар.
2. Екі өлшемді *C*(4,5) жиымының әрбір бағанасының арифметикалық ортасын табатын программа құрыңдар.
3. Бүтін санды A[4][4] жиымы берілген. Жиымның 5-ке қалдықсыз бөлінетін элементтерінің қосындысын табыңдар.

6 нұсқа

1. Екіөлшемді *A*(4,4) жиымының қосымша диагоналі бойындағы ең кіші элементті табыңдар.
2. Екіөлшемді *A*(6,4) жиымының берілген диапазонда (a;b) жататын элементтерінің арифметикалық ортасын есептеп шығарыңдар.
3. Бүтін санды A[3][3] матрицасы берілген. Оның әрбір бағанасы элементтерінің қосындыларын есептеп шығарыңдар.

7 нұсқа

1. Екіөлшемді *A*(5,10) жиымында оң сан көп пе, әлде теріс сан көп пе, соны анықтаңдар.
2. Екіөлшемді *A*(7,7) жиымында бас диагональдағы элементтердің арифметикалық ортасын табыңдар.
3. Бүтін санды B[5][5] жиымы берілген. Онын екі диагональдарында орналаскан элементтерінің көбейтіндісін табыңдар.

8 нұсқа

1. Бүтін санды *B*[5][5] жиымы берілген. Онын диагональдарында орналаскан элементтерінің ен кішісін табыңдар.
2. Екіөлшемді *A*(4,4) жиымында бас диагональдан жоғарғы орналаскан элементтердің косындысын есептеп шығарыңдар.
3. Екіөлшемді A(2,4) жиымындағы элементтердің арифметикалық ортасынын таңбасын анықтаңдар.

9 нұсқа

1. Екіөлшемді *A*(4,4) жиымында бас диагональдан төмен орналаскан элементтердің көбейтіндісін анықтаңдар.
2. Екіөлшемді *A*(4,4) жиымынын берілген диапазонда (a;b) жататын элементтерінің максимумын табыңдар.
3. Бүтін санды B[5][5] жиымы берілген. Онын диагональдарында орналаскан элементтерінің арифметикалық ортасын табыңдар.

10 нұсқа

1. Екіөлшемді *A*(4,4) жиымында 5-тен артық элементтердің қосындысын есептеп шығарыңдар.
2. Екіөлшемді *A*(4,4) жиымынын бас диагоналі бойындағы элементтердің ең үлкенін табыңдар.

3. A[5][5] жиымы берілген. Бұл жиымның әрбір элементін бас диагональдің арифметикалық ортасына бөлу жолымен алынған жаңа жиым құрындар.

11 нұсқа

1. Екіөлшемді *A*(4,4) жиымының бас диагоналі бойындағы элементтердің ең кішісін табындар.
2. Екіөлшемді *A*(4,4) жиымының қосымша диагоналі бойындағы ең үлкен элементті табындар.
3. Бүтін санды A[4][5] жиымы берілген. Оның 10-ға тең элементтерінің нөмірлерін анықтаңдар.

12 нұсқа

1. Бес цехтың әркайсысының 4 бөлімшесіндегі барлық шикізат туралы мәлімет кестесі берілген. Шикізаты ең аз цехтың нөмірін анықтаңдар.
2. Екіөлшемді *A*(4,4) жиымындағы элементтердің арифметикалық ортасының таңбасын анықтаңдар.
3. Өлшемі 4x8 болатын бүтін сандар жиымындағы барлық жұп нөмірлі элементтердің қосындысын табындар.

13 нұсқа

1. Екіөлшемді *A*(7,7) жиымында бас диагональдағы элементтердің арифметикалық ортасын табындар.
2. Екіөлшемді *A*(4,4) жиымында бас диагональдан жоғарғы орналасқан элементтердің максимумын табындар.
3. Екіөлшемді *A*(4,4) жиымында бас диагональдан төмен орналасқан элементтердің көбейтіндісін анықтаңдар.

7. СӨЗ ТІРКЕСТЕРІН ӨҢДЕУ

Pascal тілінде сөз тіркестерін өңдеу кезінде қолданылатын арнайы тип – string бар. Ал С тілінде мұндай арнайы тип жоқ. Сөз тіркестері char типті бір өлшемді жиым ретінде қарастырылады, яғни сөз тіркесі – нөлдік байтпен аяқталатын char типті бір өлшемді жиым. Нөлдік байт – барлық биттері де нөлге тең байт, ол '\0' символдық константасымен анықталады (тіркес соңы белгісі немесе нөл-терминатор). Сондықтан егер тіркесте k символ болса, онда жиымды сипаттауда k+1 элемент көрсетілуі тиіс.

Мысалы, `char a[7]` деген сипаттау тіркестің 6 символдан тұратынын, ал соңғы байт нөлдік екенін білдіреді. С тіліндегі тіркестік (жолдық) константа – қос тырнақшаға алынған символдар жиыны. Мысалы, "Берілген тапсырма" тіркесі, оның соңына нөлдік байтты компилятор автоматты түрде өзі жазады.

Айнымалы мәні болатын сөз тіркесін сипаттау кезінде бірден көрсетуге болады, мысалы, `char S1[10]="123456789", S2[]="Болат";`
Соңғы сөз ұзындығы тіркестің символдары санымен анықталады.

Символдар тіркесін пернелерден енгізу үшін екі стандартты функция – `scanf()` немесе `gets()` қолданылады, ал олардың прототиптері `stdio.h` тақырыптық файлында болады.

7.1 Символдық таңбаларды енгізу/шығару

Символдарды біртіндеп енгізу/шығару үшін `printf()` және `scanf()` функцияларының %c форматы қолданылады.

`getch()` – параметрсіз функция, басылған перненің кодын (int) береді, экранға ешқандай символ шығармайды.

`getchar()` – параметрсіз функция. Пернеден символдарды бір-бірлеп енгізеді. Сөз тіркесі `<Enter>` пернесі басылғанша енгізіле береді, оған дейін оны өзгертуге де болады.

`putch(c)` – бір символды (c – символдық айнымалы немесе константа), яғни бір таңбаны ғана экранға шығарады.

`putchar(c)` – бұл да тек бір таңбаны экранға шығарады.

Бұлар `conio.h` тақырып файлы бойынша жұмыс істейді.

Мысалы, латын әліпбиі әріптерін экранға шығару программасы төмендегідей болады:

```
#include <conio.h>
#include <stdio.h>
void main()
{ char z;
  clrscr();
  for(z='A';z<='Z';z++)
    putchar(z);
```

```
  getch();
}
```

Нәтижесі:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Ал енді осы символдарды ASCII-кодтарымен бірге шығаратын мына программаны көрейік.

```
/* латын әліпбиі*/
#include <conio.h>
#include <stdio.h>
void main()
{
 char z;
 clrscr();
 for(z='A';z<='Z';z++)
 {
   if (z=='K' || z=='U') printf("\n");
   printf(" %c-%d ",z,z);
 }
 getch();
 }
```

Программа жұмысының нәтижесі:

```
A-65 B-66 C-67 D-68 E-69 F-70 G-71 H-72 I-73 J-74
K-75 L-76 M-77 N-78 O-79 P-80 Q-81 R-82 S-83 T-84
U-85 V-86 W-87 X-88 Y-89 Z-90
```

Келесі программа 0 мен 9 арасындағы цифрлық символдарды және олардың ASCII кодтарын басып шығарады:

```
#include <conio.h>
#include <stdio.h>
void main()
{
 char z;
 clrscr();
 for(z='0';z<='9';z++)
  {
   if (z=='0' || z=='5') printf("\n");
   printf (" %c-%в ", z, z);
  }
getch();
}
```

Жұмыс нәтижесі:

```
0 - 48  1 - 49  2 - 50  3 - 51  4 - 52
5 - 53  6 - 54  7 - 55  8 - 56  9 - 57
```

Сонымен, жеке берілген символдық мәліметтер (константалар мен айнымалылар) компьютер жадында сол символдардың ASCII-кодтарына сәйкес бүтін сандық мәндер жазылған бір-бір байт орын алады екен. Соларды экранға шығаруға арналған мәліметтерді Ә қосымшасынан қарауға болады.

7.2 Символдық тіркестер

Символдық жолдарды немесе тіркестерді бірнеше тәсілмен өңдеуге болады, олардың негізгілері:

1) тіркестік константаларды қолдану;
2) char типті жиымды қолдану;
3) char типіне сілтейтін нұсқауыштарды пайдалану;
4) символдық тіркестерден тұратын жиымдарды қолдану.

Сөз тіркестері немесе тіркестік (жолдық) константа қостырнақшаға алынып жазылады. Тырнақшаға алынатын символдар тізбегінің ең соңына автоматты түрде '\0' символы жазылады. Компилятор жолдық символдарды компьютер жадына жазғанда, олардың көлемін анықтау үшін сол символдар санын есептейді. Символдық константа осы сөз тіркесі жазылған жады аймағына сілтейтін нұсқауыш болып табылады. Символдық тіркестер жиымын (массивін) беру кезінде компилятор компьютер жадының қажетті көлемін анықтау үшін жиымды сипаттағанда, оны тіркестік константа арқылы инициалдауға болады. Мысалы:

```
char c[]="Атырау, Алтай - жеріміз";
```

Әдеттегі жиым қолданылатын жағдайлар сияқты бұл жиым аты с осы жиымның 1-элементіне сілтейтін нұсқауыш болып табылады.

```
c ==&c[0];
*c =='0';
*(c+1) == c[1] == 'n';
```

Сөз тіркестерін анықтау үшін нұсқауыштарды мынадай түрде сипаттауға болады:

```
char *c1 = "\n студенттер саны";
```

осы сипаттауға эквивалентті болып келесі сипаттау есептеледі:

```
static char c1[]="\n студенттер саны";
```

Осы қарастырылған екі сипаттау да c1 тіркесінің нұсқауыш екенін білдіреді. Компьютер жадының қажетті көлемін айқын көрсетуге де болады. Мысалы, сырткы сипаттауда келесі жолдың мынадай түрде жазылғаны көрсетілген.

```
char c[25]= "Білім - өмір шырағы";
```

Элементтердің саны жолдың ұзындығынан бір символ артық болуы керек, яғни оның ең соңында '\0' символы болуы тиіс.

Статикалық немесе сырткы жиымдағы бұрынғы қарастырылған әдеттегі жиымдар оларды қолдану кезінде автоматты түрде 0-мен инициалданған болатын. Ал сөз тіркестерін пайдалану кезінде де статистикалық немесе сырткы жиымдар солар тәрізді 0 символымен инициалданады.

Келесі мысалды қарастыралық:

```
#include <stdio.h>
#include <string.h>
 main()
{char msg[30];
```

```
    strcpy(msg, "Сәлем, Азат!");
    puts(msg);
  }
```

Мұндағы msg сөзінен соң тұрған [30] саны компиляторға 29 символ үшін, яғни char типіндегі 29 айнымалыдан тұратын жиым үшін жады бөлуді камтамасыз етеді (30-орын нөлдік символмен – \0 толтырылады). msg айнымалысының символдық мәні жоқ; ол тек char типіндегі 29 айнымалының алғашқысының адресін (компьютер жадындағы белгілі бір орын адресі) сақтайды.

Компилятор strcpy(msg, "Сәлем, Азат!") операторын кездестіргенде, екі түрлі әрекет орындайды:

- "Сәлем, Азат!" тіркесі соңына (\0) символын (ASCII коды 0) қосады.
- strcpy функциясын орындап, msg айнымалысы нұсқап тұрған жады аймағына сол сөз тіркесі символдарын біртіндеп көшіреді. Ол тіркесті көшіруді сөз соңындағы нөлдік символдан кейін барып аяқтайды.

puts(msg) функциясын орындағанда, оған msg мәні, яғни тіркес құрамындағы бірінші символ адресі беріледі. Одан кейін puts сол символдың нөлдік символ емес екенін анықтап, ары қарай адреске бірді қосып, келесі символды оқиды, т.с.с. тіркес соңына дейін жетеді. Нөлдік символға жеткен соң, puts жұмысты аяқтайды;

Осындай тәсіл тіркес ұзындығына шек қоймай, нөлдік символға дейінгі символдарды біртіндеп оқуды жүзеге асырады.

7.3 Символға нұсқауышты пайдалану

Екінші тәсіл – символдарға нұсқауыш жасау. Программаны келесі түрге келтірейік:

```
#include <stdio.h>
#include <string.h>
 main()
{
 char *msg;
 msg = "Сәлем, Азат!";
 puts(msg);
}
```

msg алдындағы жұлдызша (*) компиляторға оның символға нұсқауыш екенін білдіреді, яғни msg белгілі бір символ адресін сақтай алатын айнымалы. Бірақ мұнда компилятор символдар үшін ешқандай орын бөлмейді және msg да ешқандай мәнге ие болмайды.

Компилятор strcpy(msg, "Сәлем, Азат!") операторын кездестіргенде, ол тағы екі түрлі әрекет орындайды:

- объектілік код файлы ішіндегі бір орынға соңына (\0) символы қосылған "Сәлем, Азат!" тіркесін (ASCII коды 0) жазып қояды.
- сол тіркестің алғашқы символы адресін msg айнымалысына меншіктейді.

strcpy функциясын орындалып, puts(msg) командасы бұрынғыша нөлдік символға дейінгі мәліметті көшіреді.

Енді *символдық тіркестерден тұратын жиымдарды* қарастыралық. Бұл жиымдардың әрбір жолы символдық жиым болып табылады. Мысалы, статикалық жиымның сипатталуы келесідей түрде жазылуы мүмкін:

```
static char *m[4]={"регистр","жады","курсор","элемент"};
```

бұл жиым символдық тіркестерге сілтейтін 4 нұсқауыш болып табылады. Сонымен, символдық тіркестер жиымдар болып табылатын болса, онда осы жиымдарға сілтейтін 4 нұсқауыш қарастырылады. 1-жолға сілтейтін 1-нұсқауыш болып m[0] есептеледі, m[1]2-жолға сілтейтін 2-нұсқауыш болып табылады. Сонымен, әрбір нұсқауыш соған сәйкес жолдың немесе қатардың ең бірінші символына сілтейді

```
*m[0]=='р'; *m[1]=='ж'; *m[2]=='к'; *m[3]=='э';
```

Тіркестерден құрылған жиымдарды сипаттаганда, символдық тіркестер көлемін көрсетуге де болады және бұл сипаттауда тіркестердің ұзындығын келесідей түрде көрсетуге болады:

```
static char m[10];
```

Символдар тіркестерін енгізу/шығару үшін printf() және scanf() функцияларының %s форматы қолданылады.

Тағы мысалдар келтірейік.

```
/* символдық тіркесті шығару */
#include <conio.h>
#include <stdio.h>
void main()
{char b[]="Сезам, ашыл!";
 clrscr();
 printf("%s",b);
 getch();
}
```

Мұндағы b жиымының ұзындығы – 13 символ, яғни сөздер ұзындығынан 1-ге артық.

Енді бір сөйлем енгізіп, соның соңғы сөзін экранға шығарайық.

```
#include <conio.h>
#include <stdio.h>
void main()
{
 char s,ss; // s - енгізілетін символ
// ss - алдыңғы символ
 char a[80];  // сөз жиымы
 int i,k;     // k - сөз ені
 clrscr();
 printf("Соңында нүктесі бар сөйлем енгізіңдер:\n");
for(i='0',s=' ',k=0;i<=79;i++)
    {
     ss=s; s=getchar();
     if (s==' ') continue;
```

```
      if (s=='.') break;
      if (ss==' ') k=0;
      a[k]=s; k++;
     }
 //нүктеден кейін шығу немесе тіркес біткен соң шығу
if (i==80 || k==0)
  printf("сөйлем дұрыс емес \n");
else
 {
  a[k]='\0';   // жол соңы
  printf("ең соңғы сөз: %s", a);
 }
  getch();
}
```

Мұнда символдар біртіндеп getchar() функциясы арқылы енгізіледі. Егер бос орын енгізілсе continue операторы келесі қадамға көшіреді. Нүкте енгізілсе цикл тоқталады, бірақ алыңгы k символда соңғы сөз сақталады. Егер символ нүкте де, бос орын да емес болса, онда алдыңғы символ қарастырылады. Егер ол бос орын болса, онда келесі сөз енгізіліп, k нөлге тең болады. Циклден шығу нүкте арқылы немесе 80 символ енгізілген соң орындалады.

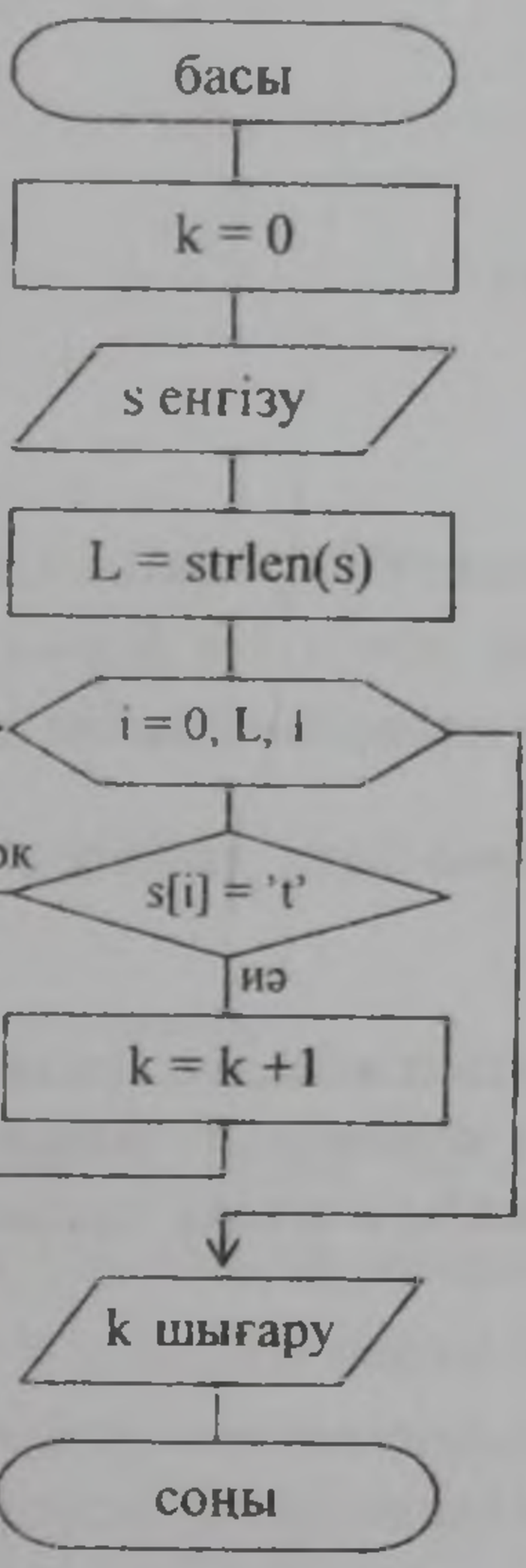
Келесі мысалда сөз тіркесінің ұзындығы екі тәсілмен анықталады.

```
#include <conio.h>
#include <stdio.h>
#include <string.h>
void main()
{char st[80];
 int i;
 clrscr();
 puts("Сөз тіркесін енгізіп, Enter басыңыз:");
 gets(st);
 i=0;
 while (st[i++])
    ;
 printf("Енгізілген тіркес ұзындығы: %i\n",i-1);
 puts("Сөз тіркесін енгізіп, Enter басыңыз:");
 gets(st);
 printf("Енгізілген тіркес ұзындығы: %i\n",
         strlen(st));
 getch();
}
```

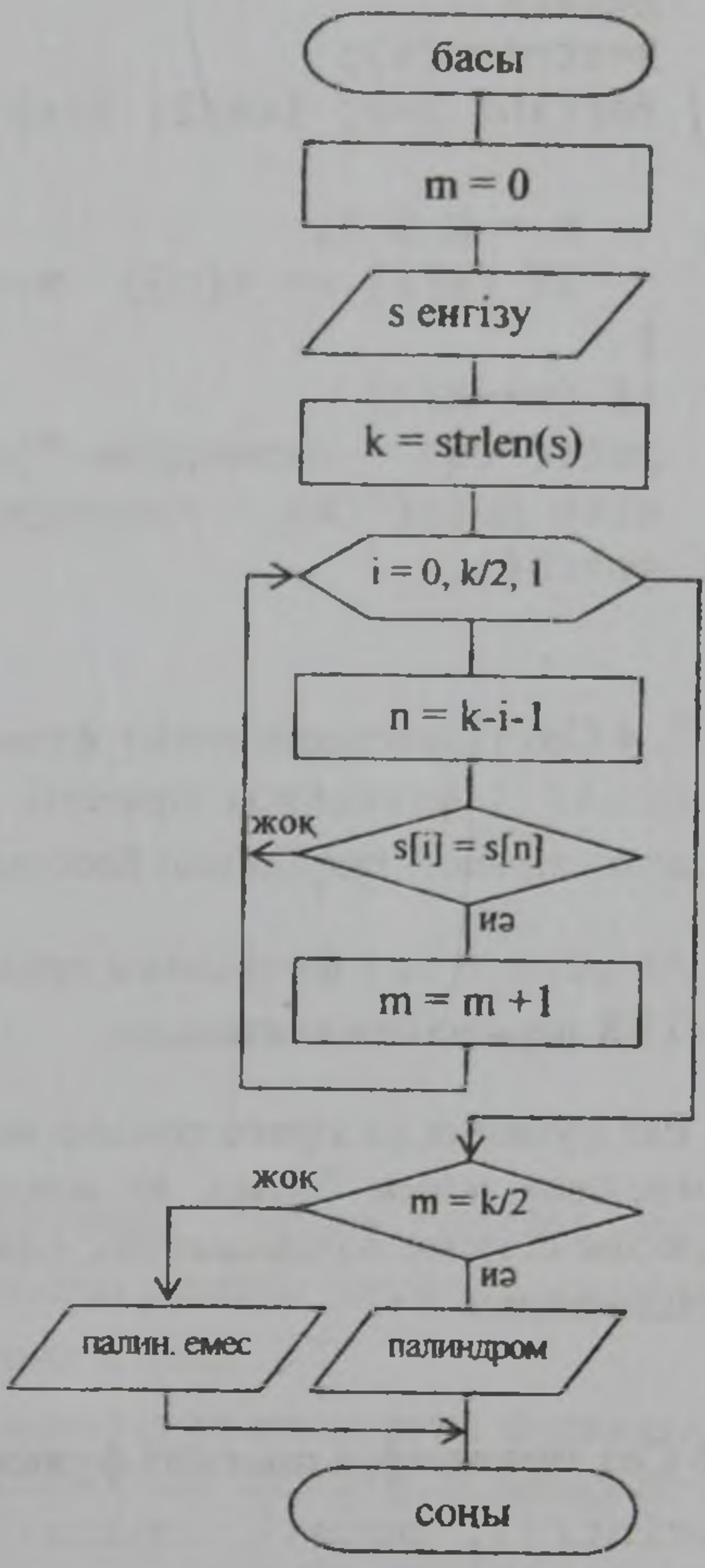
Енді бір сөз тіркесін енгізіп, оның ішінде "t" символының неше рет кездесетінін табайық.

```
#include <conio.h>
```

```
#include <stdio.h>
#include <string.h>
void main()
{ char s[80];
  int k=0,L;
  clrscr();
  puts("Сөз тіркесін (сөйлем) енгізіңіз:");
  gets(s);
  L = strlen(s);
  for(int i=0; i<L; i++)
     if (s[i] == 't')  k++;
  printf("'t' символы %i рет
          кездеседі\n",k);
  getch();
}
```



7.1-сурет. Символ санын анықтау



7.2-сурет. Палиндром сөзді анықтау

Ендігі мысалда енгізілген сөздің палиндром (алды-артынан оқығанда,
мәні бірдей – *керек, қазақ* сияқты сөздер)
екенін анықтайық.

```
#include <conio.h>
#include <stdio.h>
#include <string.h>

void main()
{
    char s[80];
    int k,n,m=0;
    clrscr();
    puts("Бір сөз (палиндром) енгізіңіз:");
    gets(s);
    k=strlen(s);
    for(int i=0; i<k/2; i++)
    {
       n = k-i-1;
       if (s[i] == s[n])  m++;
    }
    if (m==k/2)
    puts("Сөз - палиндром.");
    else puts("Сөз - палиндром емес.");
    getch();
}
```

7. 4 Сөз тіркестерін енгізу функциялары `scanf()`, `gets(str)`

`Scanf()` функциясы тіркестік айнымалылар мәнін %s форматымен енгізеді, бірақ ол тіркесті тек бірінші босорын таңбасына дейін ғана енгізе алады.

Ал `gets(str)` функциясы арасында босорыны бар тіркестерді енгізеді, енгізу ENTER пернесімен аяқталады.

Екі функция да тіркес соңына нөлдік байт қосып жазады. Оның үстіне тіркес – символдық жиым болып, ал жиым аты оның компьютер жадындағы алғашқы адресіне сілтеме болғандықтан, тіркестік айнымалы атының алдына «&» символы жазылмайды.

7.5 Сөз тіркестерін шығару функциялары

`cprintf(), puts(), cputs()`

`printf()` – экранға формат арқылы сөз тіркесін шығарады;

`cprintf()` – экранға `printf()` сияқты формат арқылы сөз тіркесін шығарады, тек олардың түстерін `textcolor()` және `textbackground()` функциялары арқылы өзгертуге мүмкіндік береді;

puts(str) – экранға сөз тіркесін шығарып, курсорды бірден келесі жолдың басына алып барады, мұндағы str – тіркестік константа немесе тіркестік айнымалы. Бұлар stdio.h тақырып файлы бойынша жұмыс істейді.

Екі функция да символдық жиымды нөлдік байтқа дейін шығарады. printf() функциясы символ тіркесі шығарылған соң курсорды келесі жолға көшірмейді, ол үшін арнайы формат (\n) жазылуы тиіс. Ал puts() функциясы символдар шығарылған соң, автоматты түрде курсорды келесі жол басына көшіреді.

```
//puts функциясын пайдаланудың мысалы

#include <stdio.h>

#include <conio.h>

main()

{ char str1[]="abc";

  char str2[]="def\nghi\n";

  char str3[]="jkl";

  puts(str1);

  puts(str2);

  puts(str3);

}
```

Нәтижесі:

```
abc
def
ghi
jkl
```

cputs(str) – экранға сөз тіркестерін шығарып, олардың түстерін **textcolor()** және **textbackground()** функциялары арқылы өзгертуге мүмкіндік береді, conio.h тақырып файлы бойынша жұмыс істейді.

Сөз тіркестерімен орындалатын басқа операциялар да стандартты функциялар арқылы аткарылады. Ол функциялар жұмыс істеуі үшін **string.h** тақырыптық файлы қажет.

Жалпы сөз тіркестеріне қолдану үшін **stdlib.h** немесе **string.h** тақырыптық файлдары қолданылады.

7.6 Сөз тіркестерімен жұмыс істейтін өзге функциялар

1) **strlen(str)** функциясы **str** сөз тіркесіндегі символдар санын (соңғы нөлді есепке алмайды), яғни жолдың ұзындығын анықтайды, оның типі **int**, тақырыптық файлы **<string.h>** (*Б қосымшасын* қараңыз).

Мысалы. Бірнеше сөз тіркестерінің ұзындығын анықтайтын программа құру керек.

```
// strlen(str) функциясын пайдалану

#include <conio.h>

#include <stdio.h>

#include <string.h>

main()

{

static char t[] = "Студенттер жайлы хабарлама.";

clrscr();

printf("%d\n",strlen(t));

printf("%d\n",strlen("Студенттер жайлы хабарлама."));

printf("%d\n",strlen("әл-Фараби ат.КазҰУ"));

printf("%d\n",strlen(""));

getch();

}
```

Мұның нәтижесі:

```
27

27

19

0
```

2) strcat(stroka1, stroka2) функциясы қатарларды біріктіру үшін колданылады. Ол stroka1 және stroka2 тіркестерін біріктіріп, нәтижені stroka1 айнымалысына меншіктейді, stroka2 тіркесінің мәні өзгермейді

Мысалы:

```
// strcat(str1, str2) функциясын пайдалану
```

```
#include <conio.h>
#include <stdio.h>
#include <string.h>
main()
{
char str1[50]="С тілін оқимыз, ";
char str2[]="жақында емтихан тапсырамыз.";
clrscr();
printf("%s\n",strcat(str1,str2));
puts(str1);    // катарды экранға шығару
puts(strcat("Егер жақсы оқысақ, ",str2));
getch();
}
```

Мұның нәтижесі:

```
С тілін оқимыз, жақында емтихан тапсырамыз.
С тілін оқимыз, жақында емтихан тапсырамыз.
Егер жақсы оқысақ, жақында емтихан тапсырамыз.
```

Келесі мысалда студенттің аты пернелерден енгізіліп, ол екінші тіркеспен біріктіріледі.

```
// strcat(str1, str2) функцияларын пайдалану

#include <conio.h>
#include <stdio.h>
#include <string.h>
main()
{
char name[80];
```

```
char stud[]= " - КазҰУ студенті";
clrscr();
puts("атын енгіз:");
gets(name);
strcat(name,stud);puts(name);
getch();
}
```

3)strcmp(stroka1,stroka2) функциясы екі сөз тіркесін салыстыру үшін қолданылады. Егер олар бірдей болса, функцияның мәні 0-ге тең болады, әйтпесе ол екі тіркестің айырмасын береді. Егер stroka1<stroka2 болса, нәтиже 0-ден кіші, ал stroka1>stroka2 болса, нәтиже 0-ден артық болады. Көбінесе бұл тәсіл екі тіркестің бірдей еместігін анықтау үшін ғана қолданылады.

Мысалы:

```
main()
{
 printf("%d\n",strcmp("Сәлем","Сәлем"));
 printf("%d\n",strcmp("Azat","Izat"));
 printf("%d\n",strcmp("Абайда","Абайла"));
 getch();
}
```

Мұның нәтижесі:

```
0
-8
-7
```

Алғашқы екі сөз бірдей, нәтижесі – 0, келесі екі сөздің алғашқы әрпі әртүрлі, олардың ASCII-кодтарының айырмасы – -8 (А - 65, І - 73), ал 3-жолы -7 (д – 164, л – 171, олардың кодтарының айырмасы 164-171=-7).

```
// strcmp(str1, str2) функциясын пайдалану
#include <conio.h>
#include <stdio.h>
#include <string.h>
#define NAME "Ритчи"
main()
{
 char f[20];
 puts("С тілінің авторы кім?");
 gets(f);
```

```
 while(strcmp(f,NAME)!=0)
  {puts("басқа кім болуы мүмкін?");
   gets(f);
  }
puts("Жауап дұрыс!");
getch();
}
```

Нәтижесі:

```
С тілінің авторы кім?
Керниган
басқа кім болуы мүмкін?
Ритчи
Жауап дұрыс!
```

4)strcpy(str1,str2) функциясы сөз тіркесінің көшірмесін алу үшін қолданылады, мұнда str2 айнымалысындағы сөз тіркесі str1 айнымалысына көшіріледі. Мысалы:

```
// strcpy(str1, str2) функциясын пайдалану
#include <conio.h>
#include <stdio.h>
#include <string.h>
main()
{
 char str1[21];
 strcpy(str1,"Хал қалай, Азат?");
 puts(str1);
 strcpy(str1,"Тамаша!");
 puts(str1);
 getch();
}
```

Нәтижесі:

```
Хал қалай, Азат?
Тамаша!
```

2-мысал:

```
// strcpy(str1, str2) функциясын пайдалану
#include <conio.h>
#include <stdio.h>
#include <string.h>
#define stroka "көшіру функциясы"
main()
{
 char *ptr=stroka;
```

```
  char res[25];
  clrscr();
  puts(ptr);
  puts(res);
  strcpy(res,ptr);
  puts(ptr);
  puts(res);
  getch();
}
```

Нәтижесі:

```
көшіру функциясы
көшіру функциясы
көшіру функциясы
```

Мұнда ptr айнымалысы көшіру функциясысөзін береді, resайнымалысы бос жол береді, ал келесі жолы екеуі де көшіру функциясысөзін береді.

5) strstr(str1,str2) функциясы 2-ші көрсетілген жолды 1-ші жолдың ішінен іздейді.

6) strset(str,ch) функциясы берілген катардағы барлық символдарды көрсетілген символға (char ch) ауыстырады.

7) strtod(str1,str2) функциясы берілген катарды double типті санға ауыстырады.

8) strchr(str,c) функциясы берілген катардағы коды көрсетілген символдың позициясын анықтайды.

9) strrev(str) функциясы берілген катардың барлық символдарын керісінше бейнелейді.

10) strpbrk(str1,str2) функциясы 2-ші катардың кез келген символын 1-ші катардан іздейді. Басқа функцияларды *Б қосымшасынан* қараңыздар.

Есептер

1. Бір символ енгізіп, соның ASCII-кодын экранға шығару.

```
#include <iostream.h>
int main()
{
  int number;
  char character;

  cout << "Type in a character:\n";
  cin >> character;

  number = character;
  cout << "The character '" << character;
  cout << "' is represented as the number ";
```

```
  cout << number << " in the computer.\n";

  return 0;
}
```

Программа жұмысы нәтижесі:

```
Type in character:
G
The character 'G' is represented as the number 71 in the
computer.
```

2. Берілген сөз тіркесінің соңғы сөзін табу керек.

Бірінші нұсқасы:

```
//songi cozdi tabu
#include<conio.h>
#include<string.h>
#include<stdio.h>
void main()
{
char str[80];
int L,i,n;
textbackground(LIGHTGRAY);
textcolor(BLUE);
clrscr();
puts("soz tirkesi:");
gets(str);
L=strlen(str);
for(i=L-1;;i--)
   if(str[i]==' ') {n=i;break;}
puts("songi soz:");
for(i=n+1;i<L;i++)
   putchar(str[i]);
getch();
}
```

Программа жұмысы нәтижесі:

```
Soz tirkesi:
Мен университетке келдім
Songi soz:
келдім
```

Екінші нұсқасы:

```
/* соңғы сөзді шығару */
#include <conio.h>
#include <stdio.h>
void main()
{
 char s,ss; // s - енгізілетін символ
  // ss - алдыңғы символ
 char a[80];   // сөз массиві
int i,k;       // k - сөз ені
clrscr();      // сөз тіркесі нүктемен аяқталуы тиіс
printf("Soz tirkesi:\n");
  for(i='0',s=' ',k=0;i<=79;i++)
 {
  ss=s; s=getchar();
  if (s==' ') continue;
  if (s=='.') break;
  if (ss==' ') k=0;
  a[k]=s; k++;
   }
    //нүктеден кейін шығу не тіркес біткен соң шығу
if (i==80 || k==0)
  printf("soylem durys emes \n");
else
{
 a[k]='\0';    // жол соңы
 printf("Eng songy soz: %s",a);
}
getch();
}
```

Программа жұмысы нәтижесі:

```
Soz tirkesi:
Azat, kitap alyp kel.
```

3. Берілген сөз тіркесіндегі символдар санын анықтау.

```
// символдар санын есептеу
#include <conio.h>
#include <stdio.h>
void main()
```

```
{char z;    // z - енгізілетін символ
 int k;     // k - символдар саны
 clrscr(); // сөз тіркесі нүктемен аяқталуы тиіс
 printf("Soz tirkesi:\n");
 for(k=0;(z=getchar())!='.';)
   if (z!=' ') k++;
 printf("Symvoldar sany=%d",k);
 getch();
 }
```

Программа жұмысы нәтижесі:

```
Soz tirkesi:
Марат аудиторияда отыр.
```

4. Бір-бірінен бос орын таңбасымен бөлінген сөздерден тұратын сөйлем берілген. Сөйлемде неше сөз бар екенін анықтау керек.

```
// тіркесте неше соз бар екенін табу
#include<conio.h>
#include<string.h>
#include<stdio.h>
void main()
{
char str[80];
int L,i,j=1,m=1,k[10],n;
clrscr();
puts("soz tirkesi:");
gets(str);
L=strlen(str);
for(i=0;i<L;i++)
   if(str[i]==' ')  j++;
printf("tirkeste %i coz bar",j);
getch();
}
```

Программа жұмысы нәтижесі:

```
soz tirkesi:
Бекзат пен Азат базардан келді
```

5. Берілген сөз тіркесін кері бағытта шығаратын программа құру керек.

Бірінші нұсқасы:

```
#include <conio.h>
#include <stdio.h>
```

```
#include <string.h>
void main()
{
  char str[80];
  int s=0;
  clrscr();
  puts("soz tirkesi:");
  gets(str);
  puts(strrev(str));
  getch();
 }
```

Программа жұмысы нәтижесі:

```
soz tirkesi:
Марат Лекеров осында
```

Екінші нұсқасы:

```
/* Енгізілген бір сөзді ғана кері жазу, 2-нұсқа */
#include <stdio.h>
#include <string.h>
#include <conio.h>
void main()
{
int top,bot;
char string[20],temp; /* сөз символдар жиымы ретінде */
clrscr();
printf("Input a word: "); // бос орыны жоқ бір сөз енгізу
scanf("%s",string);
/* сөз тіркесін енгізуде & символы қолданылмайды, өйткені жиым аты
оның басына нұсқауыш болып саналады */
bot=strlen(string);
for(top=0;top<bot--; top++)
{  // 1-символ мен соңғы символ орын ауыстырады
  temp=string[top];
  string[top]=string[bot];
  string[bot]=temp;
}
  printf("%s\n",string); //немесе  puts(string);
  getch();
}
```

Программа жұмысы нәтижесі:

```
Input a word: University
ytisrevinU
```

Үшінші нұсқасы:

```
/* Пернелерден енгізілген сөз тіркесін кері жазу,3-нұсқа */
  #include <stdio.h>
```

```
#include <string.h>
#include <conio.h>
void main(void)
{
   char s[100];     // символдық жиымды сипаттау
   int i, k;
   clrscr();
   puts("Input the string: ");
   gets(s);
   k=strlen(s);
   puts("  Result of the program \n");
   for (i=k; i>=0; i--)
      printf("%c",s[i]);
      /* жиым элементтерін кері бағытта шығару */
  getch();
}
```

6. Сөз тіркестерін салыстыратын стандартты функцияларды пайдалану

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
void main(void)
{
   char  s1[]="aaa",s2[]="aaaa",s3[]="aaa ";
   clrscr();
   printf("\n %d",strcmp(s1,s2));
   // салыстыру нәтижесі -97  - 'a' символы  коды
   printf("\n %d",strcmp (s1,s3));
  // нәтижесі -32  - 'босорын' символы коды
  printf("\n %d",strcmp (s2,s1));                  // 97
  printf("\n %d",strcmp (s3,s1));                  // 32
  printf("\n %d",atoi("a"));     // нәтижелері: 0
  printf("\n %d",atoi("12a"));                     //  12
  printf("\n %d",atoi("5aa")*atoi("2a34")); //  10
  getch();
}
```

7. Берілген сөз тіркесінің оның алғашқы нүктесіне дейінгі бөлігін экранға шығаратын программа құру керек.

```
#include<stdio.h>
#include<conio.h>
#define STOP '.'
void main()
{
 char ch;
 clrscr();
 puts("Input a string:");
 while((ch=getchar())!=STOP)
   putchar(ch);
}
```

Программа жұмысы нәтижесі:

```
Input a string:
Мен осындамын. Сен ше?
```

8. Берілген сөз тіркесінің оның алғашқы нүктесіне дейінгі символдар санын анықтайтын программа құру керек.

```
// символдар санын есептеу
#include <conio.h>
#include <stdio.h>
void main()
{char z; // z - енгізілетін символ
 int k;  // k - символдар саны
 clrscr();
 printf("Input a string:\n");
 for(k=0;(z=getchar())!='.';)
   if (z!=' ') k++;
 printf("Quantity of symbols=%d",k);
 getch();
}
```

Программа жұмысы нәтижесі:

```
Input a string:
Мен осындамын. Сен ше?
```

9. Берілген сөз тіркесінің неше сөзден тұратынын және ондағы ең ұзын сөзде неше символ бар екенін анықтайтын программа құру керек.

```
#include<conio.h>
#include<string.h>
#include<stdio.h>
```

```
void main()
{
char str[80];
int L,i,j=1,m=0,k[10],n;
clrscr();
puts("soz tirkesi:");
gets(str);
L=strlen(str);
for(i=0;i<L;i++)
   if(str[i]==' ') {k[j]=i; j++;}
k[j]=L;
n=j;
printf("Tirkeste %i soz bar",n);
k[0]=-1;
for(i=1; i<=n; i++)
   { n=k[i]-k[i-1];
     if(n>m) m=n;
   }
   printf("\neng uzin coz %i simboldan turadi\n",m-1);
getch();
}
```

Программа жұмысы нәтижесі:

```
soz tirkesi:
Мен осында отырып, сен туралы ойладым.
Tirkeste 6 soz bar
eng uzin coz 7 simboldan turadi
```

10. Енгізілген символдар тіркесіндегі '*' таңбаларын өшіріп және әрбір 'A' әрпін екеуден жазып экранға шығару.

```
#include <stdio.h>
void main ()
{
  int i, k=0;
  char str[80], ch;
```

```
puts("soz tirkesi:");
while((ch=getchar())!='\n')
   {
    if(ch=='*') continue;
    if(ch=='A' || ch=='a') str[k++] = ch;
    str[k++]=ch;
   }
  // өңделген жиым элементтерін шығару
  for(i=0; i<k; i++)
     putchar(str[i]);
}
```

Программа жұмысы нәтижесі:

```
soz tirkesi:
Alfa* beta* *gamma
```

Программаның бұл нұсқасының негізгі циклінде мәліметтерді енгізумен бірге str[ ] нәтижелік жиымы калыптасады. Егер WHILE циклінде бірден керекті символдар шығарылатын болса, есепті мүлдем жиымсыз да шығаруға болады.

```
while((ch=getchar())!='\n')
{
 if(ch=='*')    continue;
 if(ch=='a' || ch='A') putchar(ch);
 putchar(ch);
}
```

Бақылау сұрақтары

1. *Тіркестік айнымалылардың сипатталу тәсілдері қандай?*
2. *Тіркестік айнымалы қандай идентификатормен және қалай анықталады?*
3. *Бір тіркестік айнымалыға немесе тұрақтыға қанша символ жазуға болады?*
4. *Тіркестік айнымалының ұзындығы қалай анықталады?*
5. *Тіркестік өрнектер дегеніміз не?*
6. *Тіркестік айнымалылар мен тұрақтыларға қандай амалдар қолданылады?*
7. *Тіркестің ішкі символдарын қалай бөліп алуға болады?*

8. *С тілінде сөзтіркестерін өңдейтін қандай функциялар бар? Оларды қалай пайдаланады және олар қалай жазылады?*

ТАПСЫРМАЛАР

1. Бір топтағы фамилиялары бірдей студенттерді табыңдар.
2. Топ студенттері фамилиялары мен аттарының бірінші әрпін шығарыңдар.
3. Әрбір сөзден кейін бір бос орын қалдырылған сөйлемдер берілген. Құрамында дауысты әріптер ең көп кездесетін сөзді табыңдар.
4. Әрбір сөзден кейін бір бос орын қалдырылған сөйлемдер берілген. Құрамында берілген әріптен басталатын сөздердің санын анықтаңдар.
5. Әрбір сөзден кейін бір бос орын қалдырылған сөйлем берілген. Сөйлемдегі ең ұзын сөзді табыңдар.
6. Әрбір сөзден кейін бір бос орын қалдырылған сөйлем берілген. Сөйлем ішіндегі жақшалардың дұрыс қойылғанын тексеріңдер.
7. Әрбір сөзден кейін бір бос орын қалдырылған сөйлемдер берілген. Сөйлем ішіндегі жақшаға алынған мәтіндерді өшіріңдер.
8. Берілген сөз тіркесіндегі әрбір нүктені көп нүктемен (яғни үш нүктемен) алмастырыңдар.
9. Алдыңғы есеп шартындағы қатар келген нүктелердің әрбір тобын бір нүктемен алмастырыңдар.
10. Дүкендегі кассир көмекшісіне арналған программа құрыңдар. Программа тауардың бағасын, мөлшерін, сатып алынған тауарлар бағасының қосындысын есептеп, сатып алушының берген ақшасының мөлшерін сұрап, оған қайтарылатын соманы да анықтайды.
11. Әрбір сөзден кейін бір бос орын қалдырылған сөйлем берілген. Барлық сөздерді керісінше жазып шығыңдар.
12. Пернетақтадан енгізілген сөздердегі әріптердің санын есептейтін программа жазыңдар. Тапсырманы do … while циклінде орындаңдар.
13. Натурал n саны және S_1, S_2, ... S_n символдары берілген. Осы символдардың арасында неше рет + символы кездесетінін анықтаңдар.
14. Натурал n саны және S_1, S_2, ... S_n символдары берілген. Осы символдардың арасында * символдарының санын есептеңдер.
15. Натурал n саны және S_1, S_2, ... S_n символдары берілген. Осы символдардың арасында қандай символдар көп: + немесе * символы ма?
16. Сөз тіркесі енгізілгеннен кейін оның құрамында бір символ қалғанша, тіркесті цикл сайын бір символға қысқарта отырып, сөздердің барлық нұсқаларын экранға шығаратын программа жазыңдар.
17. Енгізілген сөз тіркесіндегі сөздердің санын анықтайтын программа жазыңдар. Бір сөз екіншісінен бір бос орын арқылы айырылады деп санау керек.
18. Мәтіні пернетақтадан енгізілген телеграмманың бағасын есептейтін программа жазыңдар.

19. Берілген сөзде бірінші және соңғы әріптердің қайсысы көп кездесетінін анықтайтын программа жазыңдар.
20. "*а*" әрпімен аяқталатын атау септігіндегі зат есім берілген. Осы сөзді септеп, басып шығарыңдар.
21. Берілген сөздің жұп нөмірлі орындарында канша "а" әрпі бар екендігін анықтайтын программа жазыңдар.
22. Студенттің фамилиясы, есімі және әкесінің аты бос орындармен бөлініп берілген. Студенттің аты-жөнінің инициалдарын (алғашқы әріптерін) басып шығаратын программа жазыңдар.
23. Сөз тіркесіндегі *а* әрпін өшіретін программа жазыңдар.
24. Мәтіндегі соңғы әріппен бірдей әріптерді жоятын программа жазыңдар.
25. *Z, X* сөздері берілген. *Z* сөзінде кездесетін барлық әріптерді *X* сөзінен өшіріп тастайтын программа жазыңдар.
26. Берілген сөздегі әртүрлі әріптердің санын есептейтін программа жазыңдар.
27. Сөздердің әрбір үшінші әрпін жоятын программа жазыңдар.
28. Берілген мәтіндегі "*Айна*" сөзін "*Асыл*" сөзіне өзгертетін программа жазыңдар.
29. Пернетақтадан енгізілген символды жазылған сөз тіркесінен өшіретін программа жазыңдар. Өшіру процесін жеке функция етіп қарастырыңдар.
30. Берілген мәтіндегі кездесетін "*а*" әрпін "*о*" әрпіне ауыстырыңдар.
31. Енгізілген сөз тіркесін керіге айналдыратын, яғни символдарды кері тәртіпте орналастыратын программа құрыңдар.
32. Енгізілген сөздің палиндром болатынын/болмайтынын анықтайтын программа құрыңдар.
33. Пернетақтадан енгізілген символдарды ASCII кестесінде нөмірлерінің өсу реті бойынша сұрыптайтын программа жазыңдар.
34. Бос орындармен бөлініп жазылған үш сөзден тұратын сөйлемдегі ең қысқа сөздің ұзындығын есептеп шығаратын программа жазыңдар.
35. Экранға жылжымалы жолды шығаратын программа жазыңдар.
36. Сөйлемдегі жүйелі жақшаға алынған мәтіндерді және жақшаның өзін өшіретін программа құрыңдар.
37. Ұзындығы 25 символдан артпайтын сөз тіркесін алып, одан мүмкіндігінше бірнеше жаңа сөз құрастырыңдар.
38. Сөз тіркесіндегі кездесетін "а" әрпін "ә" әрпімен ауыстыратын программа құру керек.
39. Пернелерден енгізілген сөздің ұзындығын анықтайтын программа жазу қажет. Программаны while do циклі арқылы ұйымдастырып, программа жұмысын аяқтауды '999' тіркесін енгізу арқылы жүзеге асыру керек.
40. Берілген тіркес құрамындағы сөздер бос орын арқылы бөлініп жазылған деп есептеп, олардың ішіндегі ең ұзын сөзді табу керек.

Зертханалық жұмысқа арналған тапсырмалар

А. Әр есептің блок-схемасын және соған сәйкес программасын құрып, компьютерде сол программаның нәтижесін алу керек.

В. Екі блок-схема Word программасындағы *Сурет салу-Автофигуралар-Блок-схемалар* мүмкіндігімен сызылып көрсетілуі тиіс.

С. Берілген сөз тіркестері мен нәтижелерді экранға шығарып, оларды жазып алу қажет немесе нәтижені бірден файлға жазу керек.

ЕСЕПТЕР

1 нұсқа

1. Құрамына сандар кіретін сөз тіркесінің ұзындығын – L анықтап, егер L жұп сан болса, онда тіркестегі барлық екілік сандарды өшіріп тастаңдар.
2. Берілген сөз тіркесінің дәл ортасында тұрған сөзді керісінше жазып шығындар.

2 нұсқа

1. Берілген сөз тіркесінің ұзындығын – L анықтап, егер L жұп сан болса, онда тіркестегі алғашқы сөзді өшіріңдер, ал тақ сан болса, соңғы сөзді өшіру керек.
2. Берілген сөз тіркесінде палиндром сөз бар екенін анықтап, ол жайлы мәлімет беру керек.

3 нұсқа

1. Берілген сөз тіркесінің ішінде ДОС сөзіне кіретін символдардың нешеу екенін анықтаңдар.
2. Берілген сөз тіркесінің ұзындығын – L анықтап, егер L >10 болса, соңғы сөзді өшіріп тастаңдар.

4 нұсқа

1. Берілген сөз тіркесінің ұзындығын – L анықтап, егер L тақ сан болса, онда тіркестің дәл ортасындағы символды анықтау керек, ал жұп болса – тіркес ортасындағы екі символды анықтау керек.
2. Берілген сөз тіркесінің ішіндегі барлық ! белгісін ? белгісіне алмастырыңдар.

5 нұсқа

1. Берілген тіркес ішіндегі бос орын таңбаларын астын сызу (_) таңбасына алмастырыңдар.
2. Берілген сөз тіркесінің ұзындығын – L анықтап, егер L >10 болса, соңғы сөзді өшіріп тастаңдар.

6 нұсқа

1. Берілген сөз тіркесінің ұзындығын – L анықтап, егер L 3-ке қалдықсыз бөлінетін болса, тіркестегі екінші сөзді өшіріп тастаңдар.
2. Берілген сөз тіркесінің ұзындығын – L анықтап, егер L 5-ке қалдықсыз бөлінетін болса, тіркестегі барлық жақша түрлерінің санын анықтаңдар.

7 нұсқа

1. Екі сөйлемнен тұратын сөз тіркесінің сөйлемдерінің орындарын алмастырыңдар.
2. Берілген сөз тіркесінің ұзындығын – L анықтап, егер L 3-ке қалдықсыз бөлінетін болса, тіркестегі екінші сөзді өшіріп тастаңдар.

8 нұсқа

1. Берілген сөз тіркесінің алғашқы нүктесіне дейінгі символдарды “с” әрпіне алмастырыңдар.

2. Берілген сөз тіркесіндегі бірінші және соңғы сөзді керісінше жазып шығындар.

9 нұсқа

1. Берілген сөз тіркесінің алғашқы сөзі мен екінші сөзін керісінше жазып шығындар.
2. Бірнеше сөйлемнен тұратын сөз тіркесіндегі екінші сөйлем ішіндегі "Е" әрпінің санын анықтаңдар.

10 нұсқа

1. Берілген сөз тіркесінде жақшалар бар. Алғашқы жақшалар ішіндегі сөзді анықтаңдар.
2. Берілген сөз тіркесіндегі ең ұзын сөздің енін тауып, оны керісінше жазып шығындар.

11 нұсқа

1. Сөз тіркесі берілген. Оның ішіндегі леп белгілерін нүктемен алмастырып, нүктелер санын анықтаңдар. Леп белгісі жоқ болса, ол туралы мәлімет беріңдер.
2. Құрамына сандар кіретін сөз тіркесінің ұзындығын – L анықтап, егер L жұп сан болса, онда тіркестегі барлық екілік сандарды өшіріп тастаңдар.

12 нұсқа

1. Сөз тіркесі берілген. Оның ішіндегі нүктелерді үш нүктемен алмастырып, нүктелер санын анықтаңдар, нүкте жоқ болса, ол жайлы мәлімет беріңдер.
2. Берілген сөз тіркесінің ұзындығын – L анықтап, егер L тақ сан болса, онда тіркестің соңғы сөйлемін анықтау керек, ал жұп болса – тіркес ортасындағы символды анықтау керек.

13 нұсқа

1. Сөз тіркесі берілген. Оның ішінде үтірлер бары белгілі. Алғашқы үтір мен соңғы үтірдің қай позицияда тұрғанын және олардың арасында неше символ бар екенін анықтаңдар.
2. Бірнеше сөйлемнен тұратын сөз тіркесіндегі соңғы сөйлем ішіндегі "А" әрпінің санын анықтаңдар.

8 ҚОЛДАНУШЫ АНЫҚТАЙТЫН МӘЛІМЕТТЕР ТИПТЕРІ МЕН ҚҰРЫЛЫМДАРДЫ ПАЙДАЛАНУ

Практикалық есептерде өңделуге тиіс мәліметтердің құрылымы күрделі де болуы мүмкін. Соларды айқын, әрі анық бейнелеу үшін қарапайым типтер, жиымдар және нұсқауыштар негізінде жасалған мәліметтер типтері қолданылады. C++ тілі программалаушыға өз типін анықтап, солармен жұмыс істеу ережелерін де анықтауға мүмкіндік береді. Бұлар қолданушы немесе программалаушы анықтайтын типтер деп аталып жүр.

8.1 Типтердің атын өзгерту (typedef)

Программа жұмысын түсінуді жеңілдету үшін, `typedef` түйінді сөзі арқылы бұрыннан белгілі типке жаңа ат беріледі:

```
typedef тип жаңа_ат [ өлшемі ];
```

`typedef` сөзі тип спецификаторын анықтайды. Мұндай сипаттама (хабарлау) қолданушы анықтаған типтерге қысқа және мағыналы аттар беру үшін қолданылады. Мысалы:

```
typedef char name [100];
```

мұндағы тік жақша синтаксис элементі болып табылады. Өлшемі көрсетілмеуі де мүмкін. Мысалдар:

```
typedef unsigned int UINT;
typedef char sg[10];
```

Осылай енгізілген атаулар стандарттық типтердің аттары сияқты қолданылады:

```
UINT i, j;      // unsigned int типті 2 айнымалы
sg a;           // символдардан тұратын a[10] жиымы
```

Тағы бір мысал:

```
typedef char Massiv[20];
typedef struct {
 char fio[30];
 int date, code;
 double zarplata;} Worker;
```

Осылай енгізілген тип атын кейіннен пайдалануға болады:

```
Massiv str[10]; /* әрқайсысы 20 символдан тұратын 10 жолы бар массив*/
Worker sotrudniki[50]; /* 50 құрылымнан тұратын массив*/
```

8.2 Тізбелер (перечисления – enumeration)

Программа жазу кезінде бір типтегі әр түрлі мәндер қабылдай алатын шамаларды анықтау да кездесіп жатады. Ол үшін тізім түріндегі мәліметтер типін – тізбелерді қолданған жөн. **Enum** түйінді сөзі арқылы жасалған тізбелер

программаны оқып түсінуді жеңілдетеді. Тізбелерге кіретін константалар int типінде болады. Келісім бойынша бірінші константа 0-ге тең, келесісі – 1-ге, сонан соң – 2-ге, т.с.с. бола береді. Бір тізбедегі барлық константалардың аттары әр түрлі болуы тиіс. Тізбенің жазылу форматы:

```
enum [тип_аты] {константалар_тізімі};
```

Мысалы:

```
enum seasons {win,spr,sum,aut};
```

егер

```
printf("%d%d",win,aut);
```

деп жазатын болсақ, оның нәтижесі 0 және 3 болады.

Тип аты программада осы типтегі айнымалыларды анықтау керек болған жағдайда беріледі. Компилятор осындай айнымалылардың константалар тізіміндегі мәндерді ғана қабылдауын қамтамасыз етеді. Константалар бүтін санды типте болуы тиіс және олар алғашқы мәндерді кәдімгідей түрде қабылдауы (инициалдануы) керек. Инициалдану болмаса, бірінші константа нөл болып саналады да, келесісі алдыңғысынан бірге артық мән қабылдайды. Мысалы:

```
enum {two=2,three,four,ten=10,eleven,fifty=ten+40};
```

мұнда three және four константаларына 3 және 4 мәндері, ал eleven константасына – 11 мәні меншіктеледі.

```
enum color {r,g,b}; // r=0, g=1, b=2
enum color {r=2,g=4,b=6}; // типті жариялауда инициалдау
enum color {r,g=3,b}; // r=0, g=3, b=4
```

Компилятор int және enum типтерінің арасында айырмашылық бар деп есептемейді, сондықтан тізбе типіндегі айнымалыларға программада бүтін санды мәндер меншіктеле береді. Бірақ C++ тілінде мұндай меншіктеу типті нақты түрде келтіру арқылы орындалады, әйтпесе компилятор ескертпе жасайды. Мысалы:

```
enum weekdays { /* жұмыс күндер:*/ Monday, Tuesday,
          Wednesday, Thursday, Friday};
          /* Monday -0 , Tuesday -1, т.с.с. */
weekdays today; today = (weekdays)1;
```

Тізбедегі константалар аттары бірегей (қайталанбайтын) болуы тиіс, ал мәндері бірдей де бола береді. Тізбелерді қолданудың жай константаларды және #define директивасын пайдаланудан артықшылығы байланысқан константалардың көрнекілігінде жатыр және компилятор константаларға бастапқы мән бергенде олардың типін тексере алады.

Мысалы:

```
enum {one=1,two,three,four,ten=10,thousand=1000,next};
```

болса, онда келесі жолды орындасақ,

```
printf("%d%d%d%d%d",one,two,ten,thousand,next);
```

экранға мынадай сандар шығады: 1 2 10 1000 1001.

Келесі программада С тілінде тізбемен жұмыс істеу мысалы көрсетілген.

```
#include <stdio.h>
```

```
enum Months { /* айлар*/
   January = 1, February, March, April, May, June, July,
August, September, October, November, December } months;
/* мұнда тізбе түрінде 1-ден 12-ге дейінгі сан тізбегі алынған,
өйткені January айнымалысына  1 меншіктелген */
void main ()
{int present_month;
intdiff;
/* үстіміздегі ай немірін (1 ... 12) енгізу*/
printf ("Input the present month number (1-12): ");
scanf ("%d",&present_month);
months = December;
diff  = (int) months - present_month;
/* жыл соңына дейін ... ай қалды*/
printf ("There are  %d months till the end of the year\n",
diff);
}
```

Программаны орындау нәтижесі:

```
Input the present month number (1-12):   5
There are 7 months till the end of the year
```

8.3 Құрылымдарды пайдалану

С тіліндегі жиымдар бір типтегі мәліметтерді сақтайтын болса, құрылымдар өзара логикалық байланысқан әртүрлі типті мәліметтерді байланыстырады.

Мысалы, бір жиымда 50 қызметкердің жалақысын сақтай аламыз. Егер солармен байланыстыра отырып, қызметкерлердің аты-жөнін, жасын, реттік (табельдік) нөмірін сақтайтын болсақ, мынадай мәліметтер типтерін жазуға тура келеді:

```
char name;                          // фамилиясы
int age;                            // жасы
float salary;                       // жалақысы
unsigned employee_number;   // реттік нөмірі
```

Мұндайда байланысқан әртүрлі мәліметтерді бір атаумен сақтау мүмкіндігін беретін құрылымды пайдаланады. Мысалы:

```
struct student
{
  char name[30];                    // фамилиясы
  int kurs;                         // курсы
  char group[3];                    // тобы
  float stip;                       // стипендиясы
};
```

Бұл құрылымның аты **student** идентификаторы болып табылады. Оны құрылымның *тәгі* деп, ал оның ішкі элементтерін құрылым *өрістері* деп те атайды. Өрістер кез келген типте немесе соларга нұсқауыш түріндегі құрылымның адресі арқылы катынасу амалы ретінде де бола береді.

8.4 Құрылымдарды сипаттау

Құрылымдар әртүрлі типтегі бірнеше айнымалыларды біріктіре алады. Мәліметтердің құрылымдық типі келесі сипаттаумен анықталады:

```
struct құрылым_аты
{элементтерді сипаттау;};
```

Құрылымның атауын, яғни тәгті жазу міндетті емес. Егер құрылым атаусыз болса, өнда оны сипаттау кезінде осы типке кіретін бірнеше айнымалылардың аты бірден көрсетіледі:

```
// Құрылымдар жиымын және құрылымға нұсқауышты анықтау
struct {
char name[64];                    // фамилиясы
int age;                          // жасы
float salary;                     // жалақысы
unsigned employee_number; // қызметкердің реттік нөмірі
} staff[50], *ps; /* бұл құрылымды жариялаудан кейінгі
                     айнымалыларды анықтау */
```

Құрылымдар типіндегі айнымалыларды сипаттау екі жолмен жүзеге асырылады. Алғашқы тәсіл:

```
struct Shape {      /* фигуралар*/
int type;           //0 - шеңбер, 1 - квадрат, 2 - үшбұрыш
int color;          // түсі
float radius; // радиусы
float area;         // ауданы
float perimeter;  // периметрі
};
Shape new_shape, old_shape;
```

Екінші тәсіл:

```
struct Shape {      /* құрылым аты - фигуралар*/
int type;           // 0 - шеңбер, 1 - квадрат, 2 - үшбұрыш
int color;          // түсі
float radius; // радиусы
float area;         // ауданы
float                    perimeter;                    //периметрі
} new_shape, old_shape;
```

немесе мынадай түрде де беруге болады:

```
struct gr           /* құрылым аты - топ */
{char fio[30];      /* құрылым элементі - аты-жөні */
charfak[25];       /* құрылым элементі */
 int nomer;         /* құрылым элементі */
}
gruppa1;            /* құрылымдық айнымалы аты */
struct gr gruppa2; /* құрылымдық айнымалыны хабарлау */
```

Құрылымға компьютер жадынан орын бөлу үшін құрылымдық айнымалыны сипаттау керек:

```
struct құрылым_аты айнымалы_аты;
```

Мұндайда құрылым жарияланған соң оның атын бірден пайдалануға болады (анықтауды кейінірек беруге болады), бірақ мұндай тәсіл компиляторға құрылым мөлшерін білу қажет етілмейтін кездерде қолданылады:

```
struct First;
struct Second {
  First *p;
  Second *prev, *suc;
}; struct First { /* First құрылымын анықтау */ };
```

Бұл құрылымның байланысқан тізімдерін құру мүмкіндігін береді.

Құрылымды инициалдаудың (құрылым өрістеріне мән берудің) бір жолы – оның элементтерін сипатталу реттілігімен жүйелі жақша ішінде біртіндеп беру болып табылады:

```
struct Employee
{
 char name[64];                    // фамилиясы
 int age;                          // жасы
 float salary;                     // жалақысы
 unsigned employee_number; //   реттік нөмірі
}  new_employee = {"Kadyrov",   32, 300.5, 1122};
```

Құрылымдағы жиымдарды инициалдау кезінде олардың элементтерін жүйелі жақшалар аркылы бөліп,құрылым өрістеріне мән беруге болады, мысалы:

```
struct complex{
 unsigned code;
 float salary;
} comp [2][3]={{{1021, 301}, {1031, 289}, {1041, 250}},
               {{1121, 300}, {1131, 222}, {1141, 220}}
              };
```

немесе төмендегідей түрде де инициалдау мүмкіндіктері бар:

```
struct date { int day, month, year; };
d[5]={ {1,3,1980},
       {5,1,1990},
       {1,1,2002} };
```

Құрылымдарды анықтау барысында олардың элементтеріне бастапкы мәндерді пернелерден енгізіп те меншіктеуге болады. Құрылым элементтерінің мәндерін енгізу үшін ағымдык енгізу операторы *cin*>>(C++ стилінде) немесе форматпен енгізу операторы – *scanf* колданылады.

Егер құрылымды сипаттау жолы программадағы барлық функциялардың алдында орналасса, онда ол құрылымды осы программадағы барлық функциялар пайдалана алады.

8.5 Құрылым өрістерін пайдалану

Құрылымның элементін, яғни өрістерін пайдалану үшін нүкте (.) кою жолымен (тікелей ену) немесе -> таңбалары аркылы нұскауыш бойынша ену амалы колданылады, мысалы:

```
struct  fruit {
```

```
 char name[15] ;
 int calories;
};
struct vegetable {
 char name[15];
 int calories;
};
fruit a;
vegetable b;
```

Құрылымды осылай жариялап алған соң оның өрістерін a.calories және b.calories деп, келесісін a.name и b.name деп пайдалана береміз.

Енді бір программаны толығырақ карастырайық.

1 мысал. new_employee атты құрылым өрістерін пайдалану

```
#include <iostream.h>
struct Employee {
 char name[64];
 int age;
 float salary;
 unsigned employee_number;
}  new_employee = {"Kadyrov", 32, 300.5, 1122};
void main()
{
 cout << new_employee.name <<endl <<
      new_employee.age    << endl <<
      new_employee.salary << endl <<
      new_employee.employee_number << endl;
}
```

Программа жұмысы нәтижесі:

```
Kadyrov
32
300.5
1122
```

Тағы бір мысал:

```
/* ойын карталарын сипаттайтын құрылымды анықтау */
enum suit {clubs, diamonds, hearts, spades};
              /* шыбын, қиық, табан, қарға*/
typedef int pips;/* pips(ұпайлар)int синонимі ретінде*/

struct card {
 suit s;
 pips   p;    /*   1-ден   13-ке   дейін   сандар   реттілігі
   Туз,2,3,...,10,Валет,Дама,Корольдерге сәйкес келеді*/
};
```

Келесі анықтаулар card типіндегі m1, m2 идентификаторларына орын бөледі:

```
card m1,m2;
```

Құрылымның m1 және m2 өрістерін пайдалану үшін нүкте арқылы жазылатын таңдау жолын қолданамыз, m1 өрісіне екілік қиық, ал m2 өрісіне қарғаның дамасын тағайындағымыз келеді делік:

```
m1.s = diamonds;
m1.p = 2;
m2.s = spades;
m2.p = 12;
```

Келесідей анықтаулар мен меншіктеу амалдарын карастырайық:

```
card cd, *pc = &cd;
cd.p = 2;
cd.s = spades;
```

Құрылым элементтерін пайдалануды нұсқауыш түріндегі -> операторы арқылы да орындауға болады.

Құрылым элементтерін пайдалану 8.1-кесте

Өрнек	Оның басқаша жазылуы	Мәні
cd.p	pc ->p	2
cd.s	pc ->s	spades(3) enum аркылы шығады.
(*pc).s	pc ->s	spades(3)

Құрылымдар объектіге бағытталған программалаудың іргелі ұғымдарының бірі – кластар құру негізі болып саналатын маңызды мәліметтер типіне жатады.

8.6 Құрылымдар жасау

Құрылымдар жасауға толығырақ бір мысал келтірейік

2 мысал. Қайық атын, шыққан жылын және бағасын құрылым түрінде жазу

```
/* struct_2
Құрылым жасау мысалы */
#include <stdio.h>
# define STR15 16     // аяқтау нөлі үшін бір символ қосу
struct stboat         // Қайық құрылымы
{ char model[STR15]; // моделі
  int year;          // шығарылған жылы
  float price;       // бағасы
} boat;              // қайық

void main()
{  /* қайық моделін енгізу*/
printf("\n Input a model of the boat: ");
gets(boat.model);
```

```
   /* шығарылған жылын енгізу */
printf("\n Input the year of the creation of the boat: ");
scanf("%d",&boat.year);
  /* бағасын енгізу */
printf("\n Input the price of the boat: ");
scanf("%f",&boat.price);

/* барлық енгізілген мәліметтерді шығару*/
printf("\n\nThe boat %s, the year of the creation %d\n",
      boat.model, boat.year);
printf("The boat was sold for %8.2f tenge.\n",
                                        boat.price);
}
```

Программаны орындау жұмысының нәтижесі:

```
Input a model of the boat: Tulpar
Input the year of the creation of the boat: 2009
Input the price of the boat: 168000
The boat Tulpar, the year of the creation 2009
The boat was sold for 168000.00 tenge.
```

Құрылымды функция аргументі ретінде беру тәсілін карастырайык. Мұнда құрылым аргумент ретінде мәні бойынша функцияға беріледі, яғни функцияға тек мәліметтер көшірмесі ғана жіберіледі.

3 мысал. Геометриялық фигуралар түрлерін құрылым түрінде жазу

```
/*  struct_3
Құрылымды функция аргументі ретінде қолдану */
#include
<stdio.h>
#define      PI
3.14
struct Shape {    // фигура
   int type;      // 0 - шеңбер,1 - квадрат,2 - үшбұрыш
   char color;    // түсі
   float radius; // радиусы
   double area; // ауданы
 };
void show_struct (Shape );/* show_struct (құрылымды көрсету)
                             функциясы прототипі */
void main()
{
```

```
  Shape circle;
  circle.type = 0;
  circle.color = 'r';
  circle.radius =5.0;
  circle.area = PI * circle.radius * circle.radius;
  show_struct (circle ); /* функцияны шақыру */
}
  /* функцияны сипаттау*/
  void show_struct (Shape shape)
  {
  printf("shape.type  %d\n", shape.type);
  printf("shape.color  %c\n", shape.color);
  printf("shape.radius %f\n", shape.radius);
  printf("shape.area  %f\n", shape.area);
  }
```

Программа жұмысының нәтижесі:

```
shape.type  0
shape.color r
shape.radius  5.000000
shape.area  78.500000
```

Құрылымдарға нұсқауыш ретінде берілетін мәліметтерді карастырып, функция ішінде құрылымды өзгерту тәсіліне программа құрып көрейік.

4 мысал. Геометриялық фигуралар түрлерін құрылым түрінде жазу

```
/* struct_4
Құрылым өрістерінің мәндерін өзгерту керек болса, онда құрылым
ретіндегі айнымалы мән бойынша емес, сілтеме бойынша берілуі
тиіс. Функция ішінде құрылым өрістерін пайдалану -> амалы
арқылы орындалады */
#include <stdio.h>
#define PI 3.14
struct Shape {
   int type; /* 0 - шеңбер,1 - квадрат,2 - теңбүйірлі үшбұрыш
                */
   char color;     // түсі
   float radius;   // радиусы немесе қабырғасы
   double area;    // ауданы
   } circle = {1,'b',2,4}; /* бастапқы мән беріп инициалдау */

void  change_struct  (Shape  *)  ;  /*  change_struct  функциясы
                прототипі (ағылш.құрылым өзгерту) */
void main ()
{/* функцияға мәлімет беруге дейін */
 printf("Before\n");
```

```
printf("circle.type  %d\n", circle.type);
printf("circle.color  %c\n", circle.color);
printf("circle.radius  %f\n", circle.radius);
printf("circle.area    %f\n", circle.area);
change_struct (&circle ); /* функцияны шақыру */
/* функцияға мән берген кейін */
printf("After\n") ;
printf("circle.type  %d\n", circle.type);
printf("circle.color  %c\n", circle.color);
printf("circle.radius  %f\n", circle.radius) ;
printf("circle.area    %f\n", circle.area);
}
/* функцияны сипаттау*/
void change_struct (Shape *shape)
{
  shape -> type  = 0;
  shape -> color = 'r';
  shape -> radius =5.0;
  shape -> area = PI * shape -> radius * shape -> radius;
}
```

Программа жұмысы нәтижесі:

```
Before
circle.type 1
circle.color b
circle.radius  2.000000
circle.area    4.000000
After
circle.type  0
circle.color  r
circle.radius  5.000000
circle.area    78.500000
```

Құрылымдардың өрістерінің өздері де құрылым бола алады. Енді соған бір мысал келтірейік.

5 мысал. Қызметкерлер туралы мәліметтерді құрылым түрінде жазу

```
struct    Employee    {              //    "Қызметкер"    құрылымы
char name[64];       //  фамилиясы
int age;             //  жасы
unsigned employee_number; // реттік нөмірі
struct Date {        // датасы
  int day;           //күні
  int month;         //  айы
  int year;          //  жылы
  } hiredate;        //  жұмысқа алынған мерзімі
  float salary;      //  жалақысы
} new_employee;      //  жаңа қызметкер
```

Ішкі құрылым элементін пайдалану үшін . (нүкте) операциясы қолданылады. Алдымен сыртқы құрылым элементі, сонан соң ішкіқұрылым элементі көрсетіледі.Мысалы:

```
new_employee.hire_date.month = 12;
```

Құрылымдар жиымын құруға да бір мысал келтірейік.

6 мысал. Құрылымдар жиымы

```
/*  struct_6    Құрылымдар жиымын құру */
#include <stdio.h>
#define STR15 16
#define MAX 50
struct boat    // "қайық" құрылымы
{
  char model[STR15];
  int year;
  float price;
};
void main ()
{ int i, k;
  boat Boats[MAX];
        /* Қанша қайық бар? */
  printf("How many boats? ");
  scanf("%d", &k);
  for (i = 0; i < k; i++)
  {
    flushall() ; /* буферді тазалау алдыңғы scanf() функциясын
орындаудан қалған кіріс ағынындағы жаңа жолға көшу символын
алып тастау үшін керек */
    /* қайық моделін енгізу */
    printf("\n Input a model of the boat: ");
    gets(Boats[i].model);
    /* қайықтың шыққан жылын енгізу */
    printf("\n Input the year of the creation of the boat: ");
    scanf("%d",&Boats[i].year);
    /* қайықтың бағасын енгізу*/
    printf("\n Input the price of the boat: ");
    scanf("%f",&Boats[i].price);
    }
    printf("\n\n");
/* циклде барлық қайықтар туралы мәлімет шығару*/
for (i = 0; i < k; i++)
{
  printf("The boat %s, the year of the creation %d\n",
          Boats[i].model, Boats[i].year);
  printf("This boat was sold for %8.2f $.\n",
          Boats[i].price);
}
}
```

Программаны орындау нәтижелері:

```
How many boats? 3
Input a model of the boat: Guldyz
Input the year of the creation of the boat:
2006
Input the price of the boat: 142200
Input a model of the boat: Tulpar
Input the year of the creation of the boat:
2009
Input the price of the boat: 162000
Input a model of the boat: Superi
Input the year of the creation of the boat:
2011
Input the price of the boat: 284000
The boat Guldyz, the year of the creation 2006
This boat was sold for 142200.00 tenge.
The boat Tulpar, the year of the creation 2009
This boat was sold for 162000.00 tenge.
The boat Superi, the year of the creation 2011
 This boat was sold for 284000.00 tenge.
```

8.7 Құрылым жиымдарын функция аргументі ретінде пайдалану

Егеркұрылымдар орнына оларға арналған нұсқауышты жазатын болсақ, онда құрылымдар функцияларға мәні бойынша беріледі де, программаның орындалуы жылдамдайды, өйткені мұндайда бастапқы мәліметтердің көшірмелері жасалмайды. Келесі мысалда құрылымдар орнына нұсқауышты пайдалану көрсетілген.

```
//  struct_5
құрылымдар орнына оған арналған нұсқауышты қолдану мысалы
#include <iostream.h>
#define STR15 16
#define MAX 50

int k;

struct boat
{ char model[STR15];
  int year;
  float              price;
};
void print_data(boat *);

void main()

{ int i;
boat Boats[MAX], *pastBoats;
```

```
pastBoats = &Boats[0];
cout << "How many boats? ";
cin  >> k;
for (i = 0; i < k; i++)
{
  cout << "\n Input a model of the boat: ";
  cin  >> pastBoats -> model;
  cout << "\n Input the year of the creation of the boat: ";
  cin  >> pastBoats -> year;
  cout << "\n Input the price of the boat: ";
  cin >> pastBoats ->"price;
    pastBoats++;
   }
  pastBoats = &Boats[0];
  print_data(pastBoats);
}
void print_data(boat *ptr_boat)
{
  int i;
  cout << "\n";
  for (i = 0; i < k; i++) /* k айнымалысының көріну аймағына
                             назар салыңдар */
  {
   cout<<"Theboat"<<ptr_boat->model<<"theyearofthe    creation"<<
       ptr_boat -> year << endl;
 cout<<"wassoldfor"<<ptr_boat -> price<<"$"<<endl;
   ptr_boat++;
 }
}
```

Программаның орындалу нәтижесі:

```
How many boats? 3

Input a model of the boat: Yakor

Input the year of the creation of the boat: 1987

Input the price of the boat: 18234

Input the price of the boat: 44321

Input a model of the boat: Volna

Input the year of the creation of the boat: 2000

Input the price of the boat: 39876

The boat Yakor the year of the creation 1987

was sold for 18234$
```

8.8 Біріктірмелер (объединение – union)

Біріктірмелер – бұл өрістерінің барлығы да бір адресте орналасатын құрылымның бір түрі. Оны сипаттау форматы да құрылымдардікі сиякты, тек `struct` түйінді сөзі орнына `union` сөзі қолданылады. Біріктірме ұзындығы онын өрістерінің ең көлемдісінің ұзындығына тең болады. Әрбір уақыт кезеңінде «Біріктірмелер» типті айнымалыда тек бір мән ғана сақталады. Біріктірмелерді бір сәтте бір өріс қана пайдаланылатыны белгілі жағдайларда, тек компьютер жадын үнемдеу үшін қолданады.

Басқаша айтқанда, біріктірмелер – бұл әр түрлі уақыт кезеңдерінде әр түрлі типтегі объектілер сақталатын жады аймағы. Кез келген сәтте біріктірмеде көп дегеннің өзінде, тек бір объект сақталады, өйткені оның элементтері кезектесе отырып, жадының тек бір аймағын пайдаланады. Программалаушы біріктірмедегі мәліметті соның типіне сәйкес элементтің аты арқылы пайдаланылуын қадағалап отыруы тиіс. Біріктірмеге ең соңғы жазылған элементтен басқаларын пайдалану нәтижесі анықталмаған болып саналады. Енді бір мысал карастырайық.

1-мысал

```
#include <iostream.h>// C++ тілі стилінде
void main ()
{/*  "карточка" немесе "чек" сиякты мәндері бар жаңа pt
     типін(төлем типі) сипаттау */
enum pt {CARD, CHECK};
pt type;
union payment
{char card[25];
  long check;
} info;
  type = CARD;
  cout << "Input info.card: ";
  cin.getline(info.card, 25);
  cout << "Input info.check: ";
  cin >> info.check;
  switch (type)
  {case CARD : cout << info.card << endl;break;
   case CHECK: cout << info.check << endl; break;
  }
}
```

Прогамманың орындалу нәтижесі:

```
Input info.card : My bank
Input info.check : 12345
90
```

Алынған мәннің 90 болуы біріктірме элементін пайдалану нәтижесінің анықталмағанын көрсетеді. Келесі мысалдағы біріктірмеде ең соңғы орналаскан элемент пайдаланылады.

```
#include <stdio.h>  // C тілі стилінде
void main ()
{enum pt {CARD, CHECK};
  pt type;
  union payment
  {
    char card[25];
  long                                    check;
  } info;
type = CARD;
  info.check = 12345;
  /* Мұнда компьютер жадыма соңғы болып орналаскан біріктірме
     элементі пайдаланылып отыр: */
  puts("Input info.card:");
  gets(info.card);
  switch (type)
  {case CARD : puts(info.card); break;
    case CHECK: printf ("%d",info.check); break;
  }
}
```

Программаның орындалу нәтижесі:

```
Input info.card:
My bank
My bank
```

Бұл программала дұрыс нәтиже алынды.

Келесі мысал біріктірмелерді дұрыс пайдалану жауапкершілігі толығынан программалаушыда жатқанын көрсетелі.

```
#include <iostream.h> // C++ тілі стилінде
union Number
{
  int x;
  float y;
};
void main()
{Number znachenie;
  /* соңғы болып znachenie элементі мен кабылдайды*/
  znachenie.x = 100;
  znachenie.y = 50.;
  cout << "znachenie.x:   " << znachenie.x
```

```
        << "   znachenie.y:   " << znachenie.y<< endl;
        /* соңғы болып znachenie.хэлементі мән қабылдайды*/
    znachenie.y = 50;
    znachenie.x = 100;
    cout << "znachenie.x:   " << znachenie.x
        << "   znachenie.y:   " << znachenie.y<< endl;
}
```

Программа нәтижесі:

```
znachenie.x:  1112014848   znachenie.y:  50
znachenie.x:  100   znachenie.y:  1.4013e-043
```

Программа орындалуы барысындағы мынадай мәліметтер

znachenie.x: 1112014848 znachenie.y: 50

соңғы элемент znachenie.y болғанын көрсетеді.

Ал программадағы мынадай мәліметтер

znachenie. x: 100 znachenie. y: 1.4013e-043

соңғы элемент znachenie.x болғанын көрсетеді.

Біріктірмелер келесі программада көрсетілгендей компьютер жадын үнемді пайдалану мүмкіндігін береді.

```
#include <iostream.h> // C++ тілі стилінде
union many_types
{
 char c;          //    1    байт
 int ivalue;      //    4    байт
 float fvalue;    //    4    байт
 double dvalue;   //    8    байт
} my_union;
void main()
{
// дұрыс орындалатын енгізу-шығару операциялары
my_union.c = 'b';
cout << my_union.c << endl;

my_union.ivalue =1990;
cout << my_union.ivalue << endl;
my_union.fvalue = 19.90;
 cout << my_union.fvalue << endl;
 my_union.dvalue = 987654.32E+13;
 cout <<my_union.dvalue << endl;
 // дұрыс орындалмайтын енгізу-шығару операциялары
cout << my_union.c << endl;
cout << my_union.ivalue << endl;
cout << my_union.fvalue << endl;
cout << my_union.dvalue << endl;
```

```
    /* Программаның бірінші бөлігі дұрыс орындалды, өйткені
    біріктірмеге әр түрлі типтегі мәліметтер бірден емес, біртіндеп
    меншіктелді.
    Ал екінші бөлікте тек double типті мән ғана дұрыс
    көрсетілді, өйткені ол соңында енгізілген болатын. */
    /*біріктірменіңеңкөлемдіэлементінетеңұзындығынесептеу*/
    cout << "Size of the union = "
       << sizeof (many_types) << "  bytes" << endl;
}
```

Программаны орындау нәтижесі :

```
b

1990

19.9

9.87654e+018

Ш

-154494568

-2.05461e+033

9.87654e+018

Size of the union = 8 bytes
```

8.9 Биттік өрістер

Биттік өрістер – бұл құрылым өрістерінің ерекше бір түрі. Олар мәліметтерді жинақтап, мысалы, «иә/жоқ» сиякты жалаушаларды ықшамдап орналастыру үшін кажет. Мұндағы компьютер жадының ең кіші адрестелетін ұяшығы – 1 байт, ал жалаушаны сақтау үшін бір-ак бит жеткілікті. Биттік өрістерді сипаттау кезінде оның атынан соң қос нүкте қойылады да, өріс ұзындығы битпен (бүтін он константа) көрсетіледі:

```
struct Options{
   bool centerX:1;
   bool centerY:1;
   unsigned int shadow:2;
   unsigned int palette:4;
};
```

Биттік өрістер кез келген бүтін типте бола береді. Өріс аты жазылмауы да мүмкін, мұндай өрістер аппараттық шекараға туралау үшін қолданылады. Өрісті пайдалану әдеттегідей оның атын көрсету арқылы орындалады. Өріс адресін алуға болмайды, бірақ жалпы биттік өрістерді кәдімгі құрылым өрістері сиякты пайдалана беруге болады. Жеке биттермен амалдар орындау байт пен машиналық сөздерге қарағанда, онша үнемді орындалмайды, өйткені ол үшін компилятор арнайы кодтар жасап, программа кодының көлемін тым арттырып жібереді.

Биттік өрістертерді компьютер жадында орналастыру компиляторға және аппаратураға байланысты болады.

Келесі құрылымды сипаттау мысалын карастырайық:

```
struct BitCard {
/* көрсетілген цифрлар - бұл биттер саны */
  unsigned face : 4; /*0 –туз, ... ,12 –король; 4 бит 0 ... 15 сандарын сактай
  алады*/
 ,unsigned suit : 2; /* 0 – киык, 1 – табан, 2 – шыбын, 3 – карға;
                        2 бит 0 ... 3 сандарын сактайды */
  unsigned color : 1; /* 0 - қызыл, 1 - қара;
                         мұнда 1 бит жеткілікті */
};
```

Биттік өрістерді карта элементтерін сактау үшін қолдану мысалы.

```
#include <iostream.h>// C++ тілі стилінде
#include <iomanip.h>
#include <stdio.h>
#include <math.h>

struct BitCard {
  unsigned face:4; /* ағылш. face - карта;0 - туз,...,
                      12 - король */
  unsigned suit:2; /* ағылш. suit - карта түрі;0 - киык,
                      1 - табан,2 - шыбын,3 - карға */
  unsigned color:  1; /* color-түсі;0-қызыл,1-қара */
};

void fillDeck(BitCard *); /*deck жиымын 52 картамен толтыру*/
void deal (BitCard *);     /* deal - карта тарату;
                              экран 52 карта шығару */

void main ()
{
 BitCard deck[52];  /* 52 картадан тұратын жиым */
 fillDeck(deck) ;
 deal(deck);
}

void fillDeck(BitCard *temp_deck)
{ for ( int i = 0; i < 52; i++)
   { temp_deck[i].face  =  i % 13;
     temp_deck[i].suit  =  i / 13;
     temp_deck[i].color  = i / 26;
   }
}
 void deal (BitCard  *temp_deck)
 { int l =0; /* бөліп шығару үшін */
 /* ыңғамды болуы үшін карталарды екі бағанға бөлеміз:
    0-25 карталар - бірінші баған, 26-51 карталар -
```

```
    екінші баған */
for (int k1 = 0, k2 = k1 + 26; k1 <= 25; k1++, k2++) {
cout<<"Card:"<<setw(3)<<temp_deck[k1].face<<" Suit: "
    <<setw(2)<<temp_deck[k1].suit<<" Color: "<<setw(2)
    <<temp_deck[k1].color;
cout<<" Card: "<<setw(3)<<temp_deck[k2].face<<" Suit: "
      <<setw(2)<<temp_deck[k2].suit<<" Color: "<<setw(2)
      <<temp_deck[k2].color<<endl;
l++;
if (l > 12) { /* жалғастыру үшін Enter басу керек */
                printf("Press Enter for continuing ") ;
                getchar(); l = 0;}
}
}
```

Программаның орындалу нәтижелері:

```
Card: 0 Suit:0 Color:0Card:0 Suit:2Color:1
Card: 1 Suit:0 Color:0Card:1 Suit:2Color:1
Card: 2 Suit:0 Color:0Card:2 Suit:2Color:1
Card: 3 Suit:0 Color:0Card:3 Suit:2Color:1
Card: 4 Suit:0 Color:0Card:4 Suit:2Color:1
Card: 5 Suit:0 Color:0Card:5 Suit:2Color:1
Card: 6 Suit:0 Color:0Card:6 Suit:2Color:1
Card: 7 Suit:0 Color:0Card:7 Suit:2Color:1
Card: 8 Suit:0 Color:0Card:8 Suit:2Color:1
Card: 9 Suit:0 Color:0Card:9 Suit:2Color:1
Card:10Suit:0 Color:0Card:10Suit:2Color:1
Card:11Suit:0 Color:0Card:11Suit:2Color:1
Card:12Suit:0 Color:0Card:12Suit:2Color:1
Press Enter for continuing
Card: 0 Suit:1 Color:0Card:0 Suit:3Color: 1
Card: 1 Suit:1 Color:0Card:1 Suit:3Color: 1
Card: 2 Suit:1 Color:0Card:2 Suit:3Color: 1
Card: 3 Suit:1 Color:0Card:3 Suit:3Color: 1
Card: 4 Suit:1 Color:0Card:4 Suit:3Color: 1
Card: 5 Suit:1 Color:0Card:5 Suit:3Color: 1
Card:6Suit:1 Color:0 Card: 6  Suit:  3  Color:1
Card:  7 Suit:  1 Color:0Card:   7Suit: 3Color:  1
Card:  8 Suit:  1 Color:0Card:   8Suit: 3Color:  1
Card:  9 Suit:  1 Color:0Card:   9Suit: 3Color:  1
Card: 10 Suit: 1 Color:0Card:   10Suit: 3Color:  1
Card: 11 Suit: 1 Color:0Card:   11Suit: 3Color:  1
Card: 12 Suit: 1 Color:0Card:   12Suit: 3Color:  1
Press Enter for continuing
```

Биттік өрістің адресін алуға болмайды, ал бұдан басқасында олар құрылым өрістері сияқты пайдаланыла береді. Енді карастырылған құрылымдардың бірсыпыра ерекшеліктерін атап өтейік.

- Құрылым анықталғанда, компьютер жадында орын бөлінбейді.
- Құрылымды жариялауды аяқтаған соң, нүктелі үтір қоюды ұмытпау керек.
- Біріктірмелерде биттік өрістер болмайды.

Құрылымдар тек айнымалылар емес функцияларды да қамти алады (тек C++ тілінде). Құрылым функциялары өз ішіндегі айнымалылармен ғана жұмыс істейді. Құрылым мүшелерінің барлығы да ашық болып саналады.

Бақылау сұрақтары

1. Жиымдар мен құрылымдардың қандай айырмашылығы бар?
2. С тілінде құрылымдар қандай тәсілдермен сипатталады? Олардың бір-бірінен айырмашылықтарын түсіндіріңдер.
3. Құрылым элементтерін қалай инициалдауға болады?
4. "Құрылымға нұсқауыш" термині нені білдіреді?
5. Құрылымның элементін пайдалану кезінде нүкте (.)немесе -> таңбалары не үшін қолданылады?
6. Құрылымдардың өрістерінің өздері құрылым бола ала ма?
7. Нұсқауыштарды құрылымдарға қатысты қолдану ерекшеліктері қандай?
8. Құрылымдарды функция аргументі ретінде пайдалануға бола ма?
9. Құрылымдармен жұмыс істеуде қандай амалдар қолданылады?
10. Құрылымдар жиымдарын қалай құруға болады?
11. Құрылым элементтерін экранға шығару мүмкіндіктерін көрсетіңдер.
12. Біріктірмелердегеніміз не? Олар қалай сипатталады?
13. Биттік өрістер не үшін қажет? Олар компьютер жадынан қанша орын алады?
14. Биттік өрістерді сипаттау қалай атқарылады?
15. **`Typedef`** *түйінді сөзі не үшін керек?* **`Enum`** *түйінді сөзі ше?*
16. Тізбе түріндегі айнымалылар қалай сипатталып, олар қалай инициалданады?

ТАПСЫРМАЛАР

Келесі мәліметтерден құралған төрт студент туралы ақпаратты енгізіңіз:
- тегі және инициалы;
- туған жылы;
- оқуға түскен жылы;
- бірінші семестрдің бағасы:
- физика;
- жоғарғы математика;
- информатика;

1. Әліпби бойынша реттелген студенттердің тізімін шығару керек.
2. Туған жылы бойынша реттелген студенттердің тізімін көрсету қажет.
3. Оқуға түскен жылы бойынша реттелген озат студенттердің тізімін шығару керек.
4. Сессияны 4 пен 5-ке тапсырған студенттердің анкеталық мәліметтері шығарылуға тиіс.
5. Тегі (фамилиясы) Б әрпінен басталатын студенттердің тізімін және олардың барлық пәндер бойынша бағаларын көрсету керек.

6. Оқу озаттарының анкеталық мәліметтерін шығару қажет.
7. Тегі А әріпінен басталатын студенттердің тізімін және олардың туған жылдары көрсетілуі тиіс.
8. Сессияда «3» деген бағалары бар студенттердің анкеталық мәліметтерін шығару керек.
9. Тегі В және Г әрпінен басталатын студенттердің тізімін және олардың бағаларын шығаратын программа құру керек.
10. Сессияны «үштік» баға алмай, емтихан тапсырған студенттердің тізімін және олардың туған жылдарын шығару керек.
11. Барлық студенттердің жалпы орта балын және осы орта балдан жоғары балл жинаған студенттердің тізімі көрсетілуі тиіс.
12. Барлық студенттердің жалпы орта балын және осы орта балға тең балл жинаған студенттердің тізімін шығару керек.
13. Сессияда "екілік" алған студенттердің анкеталық мәліметтерін шығару керек.
14. Информатика пәнін 5-ке тапсырған студенттердің анкеталық мәліметтерін экранға шығару қажет.
15. Физиканы 4-ке және жоғарғы математиканы 5-ке тапсырған студенттердің анкеталық мәліметтері шығарылуы тиіс.

Зертханалық жұмысқа арналған тапсырмалар

Берілген мәліметтер мен нәтижелерді экранға шығарып, оларды жазып алу қажет немесе нәтижені бірден файлға жазу керек.
Төмендегідей іс-әрекеттерді зертханалық жұмыстар кезінде орындау керек.

1 нұсқа

Flat (бөлмелер саны, көлемі, қабаты, мекен жайы, бағасы) құрылымын сипаттаңыз. Пернетақтадан жиымға бес элементтен тұратын Flat типіндегі мәндер енгізіңіз. Экранға бағалары (пернетақтадан) енгізілген саннан аспайтын пәтерлер жайлы ақпаратты шығару керек. Егер, ондай пәтер болмаса, программа экранға сәйкесінше хабарламаны шығаруы тиіс.

2 нұсқа

PEREVOZKI (ұшақ типі, рейстер саны, осы кезге дейінгі сағатпен берілген ұшқан мерзімі, мың километрмен берілген ұшқан қашықтығы) құрылымын сипаттаңыз. Пернетақтадан жиымға алты элементтен тұратын PEREVOZKI типіндегі мәндер енгізіңіз. Экранға ұшқан мерзімі(пернетақтадан) енгізілген саннан артық ұшақтар жайлы ақпаратты шығару керек. Егер, ондай ұшақболмаса, программа экранға сәйкесінше хабарлама шығарылуы тиіс.

3 нұсқа

AVTO (машина маркасы, түсі, бағасы, максималды (максимум) жылдамдығы) құрылымын сипаттаңыз. Пернетақтадан жиымға алты элементтен тұратын AVTO типіндегі мәндер енгізіңіз. Экранға жылдамдығы (пернетақтадан) енгізілген саннан (аспайтын) кем болатын машиналар жайлы ақпаратты шығару керек. Егер,

ондай машина болмаса, программа экранға сәйкесінше хабарламаны шығаруы тиіс.

4 нұсқа

TELEFON (абоненттің ЖАТ, телефон нөмірі, телефонға төлеген төлемі, орнатқан жылы) құрылымын сипаттаңыз. Пернетақтадан жиымға бес элементтен тұратын TELEFON типіндегі мәндер енгізіңіз. Экранға орнатылған жылы (пернетақтадан) енгізілген саннан (аспайтын) артық болатын телефондар жайлы ақпаратты шығару керек. Егер, ондай телефон болмаса, программа экранға сәйкесінше хабарламаны шығаруы тиіс.

5 нұсқа

DET_SAD (бала бақша нөмірі, бүлдіршіндер саны, қала ауданы, бір айлық төлемі) құрылымын сипаттаңыз. Пернетақтадан жиымға алты элементтен тұратын DET_SAD типіндегі мәндер енгізіңіз. Экранға көрсетілген аудандағы (ауданның аты пернетақтадан енгізіледі) бала бақшалар жайлы ақпаратты шығару керек. Егер, бұл ауданда балабақшалар әлі болмаса, программа экранға сәйкесінше хабарлама шығарылуы тиіс.

6 нұсқа

MUSEUM (мұражай аты, жұмыс жасау уақыты, билеттің құны, мекен жайы) құрылымын сипаттаңыз. Пернетақтадан жиымға алты элементтен тұратын MUSEUM типіндегі мәндер енгізіңіз. Экранға билетінің құны (пернетақтадан) енгізілген саннан (аспайтын) аспайтын мұражайлар жайлы ақпаратты шығару керек. Егер, ондай мұражай болмаса, программа экранға сәйкесінше хабарлама шығаруы тиіс.

7 нұсқа

TOUR (елі, турдың ұзақтығы, транспорт түрі, бағасы) құрылымын сипаттаңыз. Пернетақтадан жиымға жеті элементтен тұратын TOUR типіндегі мәндер енгізіңіз. Экранға билет құны (пернетақтадан) енгізілген саннан (аспайтын) аспайтын елдер жайлы ақпаратты шығару керек. Егер, ондай ел болмаса, программа экранға сәйкесінше хабарламаны шығаруы тиіс.

8 нұсқа

PRICE (тауардың аты, құны, дүкеннің аты) құрылымын сипаттаңыз. Пернетақтадан жиымға жеті элементтен тұратын PRICE типіндегі мәндер енгізіңіз. Экранға пернетақтадан енгізілген дүкендегі тауарлар жайлы барлық ақпаратты шығару керек. Егер, ондай дүкен болмаса, программа экранға сәйкесінше хабарлама шығаруы тиіс.

9 нұсқа

FRIDGE (тауардың аты, дайындаған фирмасы, қоймадағы саны, бағасы) құрылымын сипаттаңыз. Пернетақтадан жиымға бес элементтен тұратын FRIDGE типіндегі мәндер енгізіңіз. Экранға бағасы (пернетақтадан) енгізілген саннан (аспайтын) аспайтын тоңазытқыштар жайлы ақпаратты шығару керек. Егер, ондай тоңазытқыш болмаса, программа экранға сәйкесінше хабарламаны шығаруы тиіс.

10 нұсқа

ZODIAK (жұлдыз бойынша зодиак белгісі, аты және жөні, туылған күні) құрылымын сипаттаңыз. Пернетақтадан жиымға жеті элементтен тұратын ZODIAK типіндегі мәндер енгізіңіз. Экранға жұлдызы «мерген» болатын адамдардың

жөнін, атын және туылған күнін шығару керек. Егер, ондайлар жоқ болса, программа экранға сәйкесінше хабарламаны шығаруы тиіс.

11 нұсқа

STUDENT (оқу орнының аты, студенттің тегі және инициалдары, топ нөмірі, бес бағадан тұратын жиым-үлгерімі) құрылымын сипаттаңыз. Пернетақтадан жиымға алты элементтен тұратын STUDENT типіндегі мәндер енгізіңіз. Экранға орта балы (пернетақтадан) енгізілген саннан артық болатын студенттер жайлы ақпаратты шығару керек. Егер, ондай студент болмаса, программа экранға сәйкесінше хабарламаны шығаруы тиіс.

12 нұсқа

WORKER (жұмысшының тегі және инициалдары, қызметі, жұмысқа орналасқан жылы, үш бағадан тұратын жиым – аттестатция нәтижесі) құрылымын сипаттаңыз. Пернетақтадан жиымға бес элементтен тұратын WORKER типіндегі мәндер енгізіңіз. Экранға аттестациясының орта балы (пернетақтадан) енгізілген саннан (аспайтын) төмен болмайтын жұмысшылар жайлы ақпаратты шығару керек. Егер, ондай жұмысшы болмасы, программа экранға сәйкесінше хабарламаны шығаруы тиіс.

9 ФАЙЛДАРДЫ ПАЙДАЛАНУ

Файл – сыртқы есте сақтау құрылғыларында (магниттік дискілерде) орналастырылған және мәлімет өңдеу, тасымалдау кездерінде біртұтас күйде қарастырылатын мәліметтер жиыны.

Файлдармен жұмыс істеу үшін оларды, алдымен, ашу керек, яғни файл туралы мәліметті – атын, адресін программаға белгілі ету қажет.

С тілінде файл ашу *fopen*() функциясы арқылы орындалады. Ол сыртқы құрылғыдағы физикалық файлды, мысалы, D:\NUR.DAT, программадағы оның логикалық атымен байланыстырады. Логикалық ат дегеніміз – файлға нұсқауыш, яғни файл туралы мәлімет сақталатын жады аймағына сілтеме жасау. Файлға нұсқауыш сипатталуы тиіс.

Сонымен, файлдармен жұмыс істегенде нұсқауыштар қолданылады. Файлға нұсқауыш мынадай түрде сипатталады:

FILE *fp;

мұндағы FILE типі – <stdio.h>тақырыптық файлында анықталған құрылым. Бұл нұсқауышты көрсетілген файлмен сол файл ашылғаннан бастап, байланыстырып қоюға болады. Ол үшін келесі функция пайдаланылады:

fopen ("файл адресі/аты","қатынасу типі");

ол файлға нұсқауыш мәнін қайтарады немесе қате болса, NULL мәнін береді.

Мысалы, мынадай оператор орындалуы нәтижесінде

fp = fopen ("ex1.txt", "w");

жұмыс бумасындағы ex1.txt файлы оған мәлімет жазу (write) үшін ашылады, ал программада бұл файлды **fp** нұсқауышы арқылы пайдаланамыз (яғни *fopen()* функциясы файлдың сыртқы атын оның программада қолданылатын ішкі логикалық атымен байланыстырады).

Сонымен, файлды ашатын fopen() функциясының жалпы жазылуы:

fp = fopen(name,mode)

мұнда **fp** – файлға сілтейтін нұсқауыш; **name** – файлдың адресін де көрсетуге болатын аты, көбінесе символдық тіркес арқылы жазылады; **mode** – файл қандай режимде қолданылатынын көрсететін параметр, бұл да символдар тіркесімен төмендегідей түрде жазылады:

"r" – файлды оқу үшін ашу (файл бұрын ашылған болуы тиіс);
"w" – бос файлды информация жазу үшін ашу;
"a" – файл соңына мәлімет қосып жазу үшін оны ашу;
"r+"– файлдан информация оқу және оған информация жазу үшін ашу);
"w+"– бос файлдан информация оқу және оған мәлімет жазу үшін файл ашу (бұрын файл болса, ол өшіріледі);
"a+"– файлдан информация оқу және оның соңына информация қосып жазу үшін ашу).
"t" – файл мәтіндік (текстік) режимде ашылады, ол rt, wt, at, т.с.с. болып көрсетіле береді.
"b" – файл екілік режимде ашылады, rb, wb, ab, т.с.с. болып көрсетіле береді.

Келісім бойынша файл көбінесе мәтіндік режимде ашылады.

Файлмен жұмыс істеп болған соң, оны жабу үшін келесі функция қолданылады:

fclose(файл_нұсқауышы).

Файлға мәлімет жазу/оқу үшін жалпы енгізу/шығару функциялары тәріздес бірсыпыра функциялар пайдаланылады:
`fprintf()`,`fscanf()`,`fputs()`,`fgets()`,`getc()`,`putc()`,`fgetc()`, `fputc()`.

Бұлардың ішіндегі `getc()/fgetc()`,`putc()/fputc()` функциялары әрекеттері ұқсас, айырмашылықтары тек `getc()` пен `putc()` макроанықтаулар да, ал `fgetc()` және `fputc()` – нағыз функциялар болып табылады.

Барлық файлдық функциялар прототиптері <stdio.h> файлында орналасқан.

Файлдардан мәлімет жазу/оқу әрекеттерін үш топқа бөлуге болады:
- символдарды енгізу/шығару операциялары;
- сөз тіркестері жолдарын енгізу/шығару операциялары;
- блок бойынша енгізу/шығару операциялары.

Символдарды енгізу/шығару операцияларында файлдан бір ғана символ оқылады немесе оған бір символ жазылады. Мысалы:

`fgetc(FILE *fp);` ашылған файлдан символ окиды.

`fputc(int ch,FILE *fp);` ch символы кодын файлға жазады.

Сөз тіркестері жолдарын енгізу/шығару операцияларында бір мәлімет алмасуы кезінде файлдан сөз тіркесі жолдары оқылады немесе оған сөз тіркесі жазылады. Мысалы:

`gets(char *S);` файлдан мәлімет байттарын '\n' символы кездескенше біртіндеп окиды да, оларды S тіркесіне орналастырып, '\n' символын '\0' (нөл-терминатор) белгісіне айналдырады.

`fgets(char *S,int m,FILE *fp);` fp түрінде сипатталған файлдан байттарды оқып, оларды '\n' символы кездескенше немесе m байт окылып біткенше S тіркесі мәні ретінде жазып шығады.

Блок бойынша енгізу/шығару операцияларында мәліметтер алмасу олардың бір блогын толық қамтиды. Мысалы:

```
fread (void *ptv, int size, int n, FILE *fp);
```

мұнда fp файлынан әрқайсысы size байттан тұратын мәліметтің n блогы ptv нұсқауышы көрсетіп тұрған жады аймағына окылады (оқылатын блок үшін алдын ала орын дайындап қою қажет).

```
fwrite (void *ptv, int size, int n, FILE *fp);
```

мұнда ptv нұсқауышы көрсетіп тұрған жады аймағында орналасқан әркайсысы size байттан тұратын мәліметтің n блогы ашық тұрған fp файлына жазылады.

9.1 fprintf және fscanf функцияларын пайдалану

`fprintf`файлға информация жазу үшін, ал `fscanf` файлдан информация оқу үшін қолданылады. Олардың жалпы жазылу түрі:

```
fprintf(fp,"спецификация шаблоны",p);
```

мұндағы p– өрнек;

```
fscanf(fp,"спецификация шаблоны",адрес);
```

Мысал: Бүтін сандар тізбегін (n=5) пернелерден енгізіп, оларды файлға жазу керек болсын.

```
// файл ашып, оған 5 сан жазу
#include <stdio.h>
#include <conio.h>
main()
{ int k,i,n=5;
  char fname[]="d:\\num.txt\0";
  clrscr();
  FILE *fp;
  fp=fopen(fname,"w");
printf("Енгізілген сандар %s файлына жазылады\n",fname);
puts("Әр сан енгізілген соң, Enter басыңыз\n");
  for(i=0; i<n;i++)
  {scanf("%i",&k);
   fprintf(fp,"%d ",k);
  }
  fclose(fp);
printf("Енгізілген сандар %s файлына жазылды\n",fname);
getch();
}
```

Нәтижесі:

Енгізілген сандар a:\num.txt файлына жазылады
Әр сан енгізілген соң, Enter басыңыз
1
2
3
4
5
Енгізілген сандар d:\num.txt файлына жазылды
Файлдағы мәлімет: 1 2 3 4 5

Бес бүтін санды n=5 бұрыннан бар file1.txt файлына жазу керек, сол файлды мәлімет қосу үшін ашып, оған 5 сан жазып, сонан соң ондағы мәліметті оқып тексеру керек.

```
/* файлға мәліметтер қосу */
#include <math.h>
#include <stdio.h>
#include <conio.h>
main()
{ int i, k=0, s=0, n=5, y, x;
  char fname[]="file1.txt";
  clrscr();
  FILE *fp;
```

```
fp=fopen("file1.txt", "at");
if ((fp=fopen(fname,"at")) == NULL)
 { printf("\nФайлды ашуда қате болды");
   getch();
   return(0);
 }
puts("Enter арқылы бөліп, 5 сан енгізіңіз:");
for(i=1; i<=n;i++)
 {scanf("%i", &x);
  fprintf(fp,"%d ", x);}
fclose(fp);
   getch();
}
```

Нәтижесі:

Enter аркылы бөліп, 5 сан енгізіңіз:
11
22
33
44
55

Файлдағы мәлімет: 1 2 3 4 5 11 22 33 44 55

Дискідегі `file1.txt` файлын мәліметтер оқу үшін ашып, ондағы мәліметті оқып, файлдағы тақ сандардың қанша екенін және қосындысын табу керек.

```
/* файлдан мәліметтер оқу */
#include <math.h>
#include <stdio.h>
#include <conio.h>

main()
{ int i,k=0,s=0,n=5,y,x;
  char fname[]="file1.txt";
  clrscr();
  FILE *fp;
  fp=fopen(fname,"r");

  /* Оқу үшін файл ашу */
fp=fopen("fp","rt");

 if ((fp=fopen(fname,"rt")) == NULL)
{  printf("\n Файл ашуда қате болды");
   getch();
   return(0);
}
puts("Файлдан оқылған сандар:");
while(!feof(fp))
 {fscanf(fp,"%i",&y);
```

```
    printf(" %i ",y);
    if (y%2==0)   (s+=y;k=k+1;)
   }
  fclose(fp);
  printf("\nТақ сандар қосындысы s = % i,
            олардың саны k=%i", s, k);
  getch();
}
```

9.2 fgets және fputs функцияларын пайдалану

Сөз тіркестерімен жұмыс істегенде **fgets** және **fputs** функциялары қолданылады:

fgets функциясының жалпы жазылу түрі:

fgets(нұсқауыш,MAXLEN,fp);

нұсқауыш – компьютер жадындағы орынға сілтейтін нұсқауыш; **MAXLEN** – оқылатын тіркестің максимальды ұзындығы; **fp** - файл нұсқауышы.

Мысалы:

```
#define L 20
main()
{
  FILE *fp;
  char *st[n];
  fp=fopen("stroka","r");
  while(fgets(st,L,fp)!=NULL)
puts (st);
}
```

Файлдарға сөз тіркестерін жазу үшін **fputs** функциясы келесі түрде қолданылады:

status=fputs(қатар нұсқауышы, fp);

status – бүтін сан, оның мәні**eof**функциясында жазылады,
егер**fputs()** функциясы файлдың соңына шыққан болса немесе қате тапса,
fputsфункциясы жазылатын жолдың соңына /0 символын жазбайды.

Төменде осы функцияларды пайдалану мысалы келтірілген.

```
#include <stdio.h>
void main()
{
 int n;
 char str[50],str1[50],ch;
 FILE *fp;
 // Файлға мәлімет жазу
 fp = fopen("ex.txt","w");
 puts("Бүтін сан енгізіңіз: "); scanf("%d",&n);
  fprintf(fp,"%d\n",n);
  puts ("Символ енгізіңіз: "); ch=getchar();
  putc (ch, fp);
```

```
puts ("Сөз тіркесін енгізіңіз: "); gets(str);
fputs(str,fp);
fclose(fp);
// Файлдан мәлімет оқу
if((fp = fopen("ex.txt","r")) != NULL)
 {
 fscanf (fp, "%d", &n); printf ("n=%d\n", n) ;
 ch = getc (fp); putchar (ch);
 fgets(str1, 50, fp); puts (str1);
 fclose (fp);
 } else printf ("\nФайлдан мәлімет оқылмайды!");
}
```

Мұндағы **fgets()** функциясының екінші параметрі N – оқылатын символдар саны, оған '\0' белгісі де қосылады. Бұл функция өз жұмысын N-1 символын оқығаннан кейін немесе '\0' белгісі кездескенде аяқтайды. Екеуінде де сөз тіркесі соңына '\0' белгісі қосылады.**fgets()** функциясы оқылған сөз тіркесі адресін қайтарады немесе файл оқылып болғанда (не қате шыкса), NULL белгісін береді.

fputs() функциясы әрекет дұрыс орындалса, соңғы оқылған символ кодын қайтарады, ал қате болса, EOF (файл соңы) белгісін береді. Бұл функция курсорды автоматты түрде келесі жолға көшірмейді.

Жоғарыдағы функциялар файл мәліметтерін біртіндеп, символдан соң келесі символды карастыра отырып өңдейді. С тілі файлдармен жиым сиякты жұмыс істеуге де мүмкіндік береді, яғни кез келген байтты жеке өңдеуге де болады. Файл ішіндегі белгілі бір орынды айқындау үшін мына функция колданылады:

fseek(файлға нұскауыш,бастапқы нүктеден ығысу, бастапқы нүкте);

Екінші аргумент типі *long*, оның мәні оң да, теріс те болуы мүмкін. Ол бастапқы нүктеден канша орынға (байтпен) ығысу керек екендігін көрсетеді. Үшінші аргумент файлдағы бастапқы нүкте орнын аныктайтын код болып табылады. Осы код үшін мынадай мәндер тағайындалған:

0 - файл басы;

1 - ағымдағы позиция;

2 - файл соңы.

Дұрыс орындалғанда, *fseek()* функциясы 0 мәнін береді, ал егер қате (мысалы, файлдын сол жак шекарасынан ары аспакшы болғанда) болса, онда 1 береді.

9.3 fwrite және fread функцияларын пайдалану

Құрылымдарды пайдаланатын файлдармен жұмыс істеу кезінде **fread()/fwrite()** функцияларын пайдаланған ыңғайлы. Олардың жазылуы:

```
fread(ptr, size, n, fp);
```

мұнда **fp** файлынан әркайсысы **size** байттан тұратын мәліметтің **n** блогы **ptv** нұскауышы көрсетіп тұрған жады аймағына оқылады (оқылатын блок үшін алдын ала орын дайындап кою қажет).

```
fwrite(ptr,size,n,fp);
```

мұнда **ptr** нұскауышы көрсетіп тұрған жады аймағында орналаскан әркайсысы **size** байттан тұратын мәліметтің **n** блогы ашык тұрған **fp** файлына жазылады.

Осы функцияларды қолданудың бір мысалын келтірейік:

```
typedef  struct
{
  char author [30];
  char title [50];
int pages;
} BOOK;
BOOK  b1={ "Kernighan", "C Language", 256 }, b2;
FILE *fp;
void main()
{...
 fp=fopen("struct.txt", "w+");
// Файл әрі оқу, әрі жазу үшін ашылды
 fwrite(&b1, sizeof(BOOK), 1, fp);
 fseek(fp,0,0); // маркер файл басына
 fread(&b2, sizeof(BOOK), 1, fp);
 printf("Авторы - %s,аты - %s,беттер саны - %d\n",b2.author,
         b2.title, b2.pages);
}
```

Файлдың соңы белгісімен аяқталып енгізілген мәтіндегі ең ұзын сөздің символдары санын анықтау қажет (сөздер бір-бірінен бос орын таңбасымен бөлініп жазылған.)

```
#include <stdio.h>
#include <conio.h>
void main()
{
 int c, k, min;
 clrscr();
 k=0; min=80;
 puts("EOF белгісімен аяқталатын мәтін енгізіңіз:");
 c=getchar();
 while(c!=EOF)
  {
   while(c==' ') c=getchar();
   //Сөз алдындағы бос орынды аттап өтеміз
   if(c==EOF) break;
   while(c!=' ' && c!=EOF)
   { c=getchar(); k++;
   }
   // Ағымдағы сөздің ұзындығын табамыз
   if(k<min) min=k;
   // оның ұзындығын қысқа сөзбен салыстыру
   k=0; // Ағымдағы сөз ұзындығын 0-ге теңеу
  }
 printf("Ең қысқа сөзде %d символ бар",min);
 getch();   // Тұтынушы экранын тоқтата тұру
}
```

Бақылау сұрақтары

1. *Файл дегеніміз не? Ол не үшін пайдаланылады?*
2. *Файл қайда орналасады және қалай белгіленеді?*
3. *Файл ашу функциясы қалай жазылады?*
4. *Файл атын программада қалай анықтаймыз?*
5. *Файлдың қолданылу режимдері қалай көрсетіледі?*
6. *Файлға мәлімет жазу/оқу функциялары.*
7. *Файлға жаңа элемент қалай қосылады?*
8. *Мәлімет оқылған файлға мәлімет жазуға бола ма?*
9. *Файлдан ақпарат оқу үшін не істеу керек?*
10. *Файлға ақпарат жазу үшін не істеу керек?*
11. *Символдар мен сөз тіркестерін файлға жазу үшін не істеу керек?*
12. *Құрылымдарды пайдаланатын файлдар қандай функцияларды пайдаланады?*

ТАПСЫРМАЛАР

1. Файлдан сандар оқып, солардың ішіндегі теріс сандар қанша екенін анықтайтын программа құру керек.
2. Файлдан сөз тіркесін оқып, солардың ішіндегі 6 символдан артық сөздерді экранға шығарып, басқа файлға жазып шығындар.
3. Файлдан сандар оқып, солардың арифметикалық ортасын файл соңына қосып жазу керек.
4. Файлдан сөз тіркесін оқып, солардың керісінше жазылған нұсқасын басқа бір файлға жазып шығындар.
5. Файлдан сандар оқып, солардың максимумын анықтайтын программа құру керек.
6. Файлдан сөз тіркесін оқып, солардың ішіндегі бос орын орнына сызықша жазып оны басқа бір файлға жазып шығу керек.
7. Файлдан сандар оқып, солардың минимумын анықтайтын программа құру керек.
8. Бір файлда екі сөйлем жазылған, соның екінші сөйлемін басқа файлға жазып шығу керек.
9. Файлдан сандар оқып, солардың көбейтіндісін анықтайтын программа құру керек.
10. Сөз тіркесінен тұратын файлдағы бас әріптерді кіші әріптерге айналдырып, басқа файлға жазып шығу керек.
11. Файлдан сандар оқып, солардың нешеуі 5-тен артық екенін анықтау керек
12. Сөз тіркесінен тұратын екі файл берілген. Осы екі файлдағы сөз тіркестерін біріктіріп, үшінші файлға жазып шығындар.
13. Файлдан сандар оқып, солардың қосындысын анықтау керек
14. Сөз тіркесінен тұратын екі файл берілген. Осы екі файлдағы сөз тіркестерінің орнын ауыстырып жазып шығу керек.

15.Файлдан сөз тіркесін оқып, солардың ішіндегі ең ендісін анықтап, соның неше символдан тұратынын анықтау қажет.

16.Файлдан сандар оқып, солардың тақтарын бір файлға, жұптарын екінші файлға жазып шығындар.

17.Файлдан бірнеше сөйлем оқып, сол сөйлемдердің "м" әрпінен басталатын бір сөйлемін ғана экранға шығарып, сонын ұзындығын анықтайтын программа құру керек.

18.Файлдан сандар оқып, солардың барлығын да бірге арттырып, шыққан сандарды басқа бір файлға жазып шығындар.

19.Файлдан бірнеше сөйлем оқып, сол сөйлемдердің ең соңғы сөйлемін ғана экранға шығарып, сонын ұзындығын анықтайтын программа құру керек.

20.Файлдан сандар оқып, соларды керісінше тәртіппен екінші бір файлға жазып шығындар.

21.Файлдан сандар оқып, солардың цифрларын экранға сөзбен шығаратын программа құру керек, мысалы, 0 орнына "нөл", 1 орнына "бір", т.с.с. 9 орнына "тоғыз" деп жазатын болуы тиіс.

22.Файлдан сөз тіркесін оқып, солардың бірінші сөзі мен соңғы сөзін алмастырып, екінші бір файлға жазып шығындар.

23.Файлдан сөз тіркесін оқып, соларды керісінше жазып шығатын программа құру керек.

24.Файлдан сандар оқып, солардың алғашқы жартысы мен соңғы жартысының орнын ауыстырып, нәтижесін жаңа файлға жазып шығу керек.

25.10 бүтін саннан тұратынфайл жасайтын программа құру керек. Сол файлдағы сандарды окып, олардың қосындысын анықтандар.

26.While операторы аркылы Char типті элементтерден тұратын файл жасау кажет. Циклден шығу шарты –z әрпін енгізу. Сол файлдың көшірмесін екінші бір файлға жазып, жазылған мәліметтерді экранға да шығару керек.

27.Integer типті N саннан тұратын файл жасап, сол файлдағы жұп сандарды экранға шығару қажет.

28. Бүтін сандардан тұратын файл жасап, сол сандарды басқа бір файлға кері тәртіппен жазып шығу керек.

29.Мынадай құрылымдағы бірнеше катарлары бар файл жасау керек:

a. реттік нөмірі;
b. фамилиясы, аты-жөні;
c. жалақысы.

30.Осы файлға бес адам туралы мәліметтер енгізіп, басқа файлға осылардың ішіндегі ең көп жалақы алатын адам туралы мәліметті көшіріп жазу керек.

31.Файлдағы мәтінді түгел оқып шығып, ондағы "о" әрпін "а" әрпімен алмастыратын программа құрындар.

32.Файлдағы мәтіндік ақпаратты экранға және қағазға шығаратын программа жасау керек.

33 Файлда N бүтін сан жазылған. Соларды өсуі бойынша реттеп, екінші файлға жазып шығару қажет.

34. Нақты сандардан тұратын бір өлшемді жиым элементтерін пернелерден енгізе отырып, бір файлға жазып шығып, сол жиымның жұп индексті элементтерін екінші файлға жазатын программа құрыңыздар.

35. Топтағы студенттердің үлгерімін бір файлға мынадай түрде жазу керек: рет нөмірі, аты-жөні, 5 сабақтың аты, әр сабақтан алған 3 бағасы. Программаға енгізілетін мәліметтер пернелерде теріліп, файлға жазылуы қажет. Жақсы оқитын студенттер тізімін екінші бір файлға бөлек жазыш шығу керек.

Зертханалық жұмысқа арналған тапсырмалар

Берілетін мәліметтер алдын ала файлға жазылып қойылуы тиіс. Алынған нәтижелерді файлға жазылса, соларды оқу арқылы тексеріп тапсырманың дұрыстығын тексеріп шығу керек.

ТАПСЫРМАЛАР

1 нұсқа

1. Файлдан сандар окып, солардың ішіндегі теріс сандар қанша екенін анықтайтын программа құру керек.

2. Файлдан сөз тіркесін оқып, солардың ішіндегі 6 символдан артық сөздерді экранға шығарып, басқа файлға жазып шығыңдар.

2 нұсқа

1. Файлдан сандар оқып, солардың арифметикалық ортасын файл соңына қосып жазу керек.

2. Файлдан сөз тіркесін оқып, солардың керісінше жазылған нұсқасын басқа бір файлға жазып шығыңдар.

3 нұсқа

1. Файлдан сандар оқып, солардың максимумын анықтайтын программа құру керек.

2. Файлдан сөз тіркесін оқып, солардың ішіндегі бос орын орнына сызықша жазып оны басқа бір файлға жазып шығу керек.

4 нұсқа

1. Файлдан сандар оқып, солардың минимумын анықтайтын программа құру керек.

2. Бір файлда екі сөйлем жазылған, соның екінші сөйлемін басқа файлға жазып шығу керек.

5 нұсқа

1. Файлдан сандар оқып, солардың көбейтіндісін анықтайтын программа құру керек.

2. Сөз тіркесінен тұратын файлдағы бас әріптерді кіші әріптерге айналдырып, басқа файлға жазып шығу керек.

6 нұсқа

1. Файлдан сандар оқып, солардың нешеуі 5-тен артық екенін анықтау керек

2. Сөз тіркесінен тұратын екі файл берілген. Осы екі файлдағы сөз тіркестерін біріктіріп, үшінші файлға жазып шығындар.

7 нұсқа

1. Файлдан сандар оқып, солардың қосындысын анықтау керек

2. Сөз тіркесінен тұратын екі файл берілген. Осы екі файлдағы сөз тіркестерінің орнын ауыстырып жазып шығу керек.

8 нұсқа

1. Файлдан сөз тіркесін оқып, солардың ішіндегі ең ендісін анықтап, соның неше символдан тұратынын анықтау қажет.

2. Файлдан сандар оқып, солардың тақтарын бір файлға, жұптарын екінші файлға жазып шығындар.

9 нұсқа

1. Файлдан бірнеше сөйлем оқып, сол сөйлемдердің "м" әрпінен басталатын бір сөйлемін ғана экранға шығарып, соның ұзындығын анықтайтын программа құру керек.

2. Файлдан сандар оқып, солардың барлығын да бірге арттырып, шыққан сандарды басқа бір файлға жазып шығындар.

10 нұсқа

1. Файлдан бірнеше сөйлем оқып, сол сөйлемдердің ең соңғы сөйлемін ғана экранға шығарып, соның ұзындығын анықтайтын программа құру керек.

2. Файлдан сандар оқып, соларды керісінше тәртіппен екінші бір файлға жазып шығындар.

11 нұсқа

1. Файлдан сандар оқып, солардың цифрларын экранға сөзбен шығаратын программа құру керек, мысалы, 0 орнына "нөл", 1 орнына "бір", т.с.с. 9 орнына "тоғыз" деп жазатын болуы тиіс.

2. Файлдан сөз тіркесін оқып, солардың бірінші сөзі мен соңғы сөзін алмастырып, екінші бір файлға жазып шығындар.

12 нұсқа

1. Файлдан сөз тіркесін оқып, соларды керісінше жазып шығатын программа құру керек.

2. Файлдан сандар оқып, солардың алғашқы жартысы мен соңғы жартысының орнын ауыстырып, нәтижесін жаңа файлға жазып шығу керек.

10. ГРАФИКАЛЫҚ РЕЖИМДЕ ЖҰМЫС ІСТЕУ

С тілінде растрлық графика жұмыс істейді, мұнда бірсыпыра функциялар пайдаланылады, олардың прототиптері `graphics.h` тақырыптық файлында сақталады. График нүктелерден – пиксельдерден тұрады. *Пиксель – экранның адрестелетін ең кіші элементі.* Алдымен, графика шығара алатын бейнережимді іске қосу керек. Мұнда экраннның пиксельмен берілген мөлшері және түстер саны беріледі.

Мәтіндік режимдегі сиякты мұнда да жұмыс терезелер арқылы жүргізіледі.

Графикада үш координаталық жүйе: *абсолюттік*, *салыстырмалы* және *масштабталған* жүйе қолданылады. Тіке және көлденең өстер бойынша пиксельдер саны экран типіне байланысты болады.

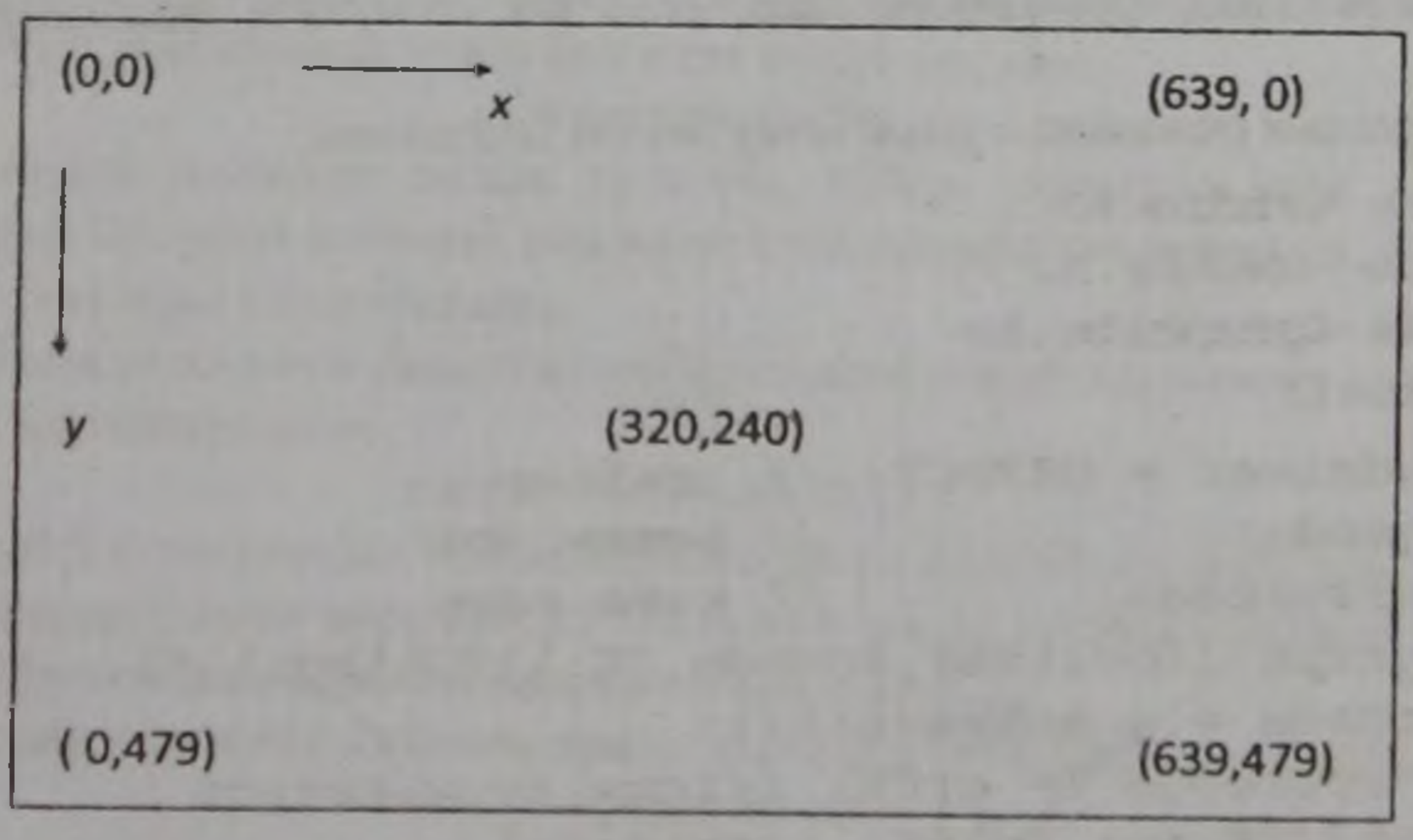
Абсолюттік координатада координаталар басы – (0;0) нүктесі сол жақ жоғарғы бұрышта болып саналады, *x* координатасы солдан оңға қарай, *y* координатасы жоғарыдан төмен қарай өседі.

Салыстырмалы режимде координаталар басы экраннның кез келген нүктесіне ауыстырыла алады.

Масштабталатын режимде экран бетінде масштабталған координаталар беруге болады, онда *x* пен *y* өстері бойынша минимум және максимум мәндер енгізіп, жұмыс істеуге мүмкіндік бар.

Жалпы абсолюттік графикалық режимде (10.1 сурет)

- әрбір пиксель берілген 16 түстің біріне боялады;
- (0,0) – экраннын сол жақ жоғарғы бұрышы координатасы;
- (639,479) – оң жақ төменгі бұрышының координатасы болады;
- әртүрлі фигураларды экранға шығару үшін алдын ала графикалық режимді іске қосып алу керек.



10.1-сурет. Экрандағы нүктелердің координаталары

Жалпы, дисплей адаптерлері графикалық режимде 200, 350, 480, 600 нүктелерден тұратын экран жолдарының әркайсысында 640, 720, 800 нүктелер тізбегін

бейнелей алады. Мұндағы нүкте деп отырғанымыз – көлемі 0,8x1 мм^2 шамасында болып келген (CGA) кішкентай тіктөртбұрыш, яғни пиксель.

Әрбір нүктенің координаталары екі бүтін санмен (*x*, *y*) анықталады. Дисплей экранына график салу үшін оның нүктелерінің координаталарын көрсету қажет. Координаталар басы (0,0) болып экранның сол жақ жоғарғы бұрышы есептеледі. *X* координаталары (бағаналар немесе позициялар нөмірлері) солдан оңға қарай, ал *y* мәндері (жолдар немесе қатарлар) жоғарыдан төмен қарай өсіп отырады. Мысалы, VGA адаптерінің экран бұрыштарының координаталарын *x*=0..799, *y*=0..599 аралығында көрсету қажет. Экрандағы шеткі нүктелер координаталары суретте көрсетілген.

Олардың ең жоғарғы мәндері пайдаланылған экран адаптеріне тәуелді болады, яғни (0,0)..(320x200), (0,0)..(640x480), (0,0)..(800x600) аралықтарында және т.б. болуы мүмкін. Сонымен, графикалық режимде экрандағы кез келген объект көрініп тұрған нүктелер тобынан тұрады. Мәтіндік режимнен графикалық режимге көшкенде экран тазартылады. Графикалық режимде экраннан курсор көрінбейді. Дисплей экранының бетіне (кейін қағазға) нүкте, түзу немесе қисық сызық, шеңбер, эллипс және кез келген тұйық сызық сызып шығаруға болады. Сонымен қатар тұйық сызықтардың ішін әртүрлі түске бояп қою мүмкіндіктері де бар. Сызықтарды жылжыту, айналдыру және басқа орынға көшіру арқылы көрнекі бейнелер мен мультфильмдер жасауға болады.

График тұрғызу үшін, алдымен графикалық режимге көшу керек, сонан соң барып, оны шығару немесе бастау нүктесін көрсету қажет. Мәтіндік режимде ол курсор позициясы болып саналады, ал графикалық режимде көрініп тұратын курсор жоқ, бірақ экранда көрінбейтін курсор тәрізді сілтеме белгі CP (current pointer) бар. Негізінде оны да курсор деп қабылдауға болады.

Графикалық режимдегі жұмыстарды атқаратын Турбо С нұсқасында бірсыпыра графикалық функциялар бар, енді біз солардың негізгілеріне тоқталып өтеміз.

Графикалық режимде жұмыс істеу былай басталады:

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
void main()
{ int gdriver = DETECT; // драйвер
int gmode;              // режим аты
int errorcode;          // қате коды
initgraph (&gdriver,&gmode,"C:\\TC\\bgi\\");
errorcode = graphresult();
if (errorcode != grOk) //іске қосу қатесі
{printf("Қате: %d\n", errorcode);
 puts("Аяқтау үшін ENTER басыңыз");
 getch ();
 return;
}
```

```
    ... Ары қарай программа мәтіні...
    getch();
    closegraph ();
}
```

10.1. Графикалық режим орнату, одан шығу, мәтін жазу, сызық салу функциялары

Графикалық режим драйверін іске қосу функциясы

```
initgraph(&Driver,&Mode,Path);
```

Driver параметрі бейнелік жүйе драйверін анықтайды, *Mode* параметрі бейнелік жүйе жұмыс режимін береді, ал *Path* параметрі драйвер файлының орнын көрсетеді. Көбінесе *Driver* параметрі мәніретінде *DETECT* болатын бүтін константа колданылады. Мұндайда *initgraph* функциясының өзі графикалық драйвер типін анықтап, ең дұрыс режимді таңдап алады. Тақырыптық файлы: `<graphics.h>`

Драйвер дұрыс оқылған соң, *initgraph()* функциясы 4К көлемінде (келісім бойынша) ішкі графикалық буфер ұйымдастырады да, экран түсі, сызықтар түсі анықталып, дисплей адаптері графикалық режимге кіреді. Экран тазаланып, курсор сол жак жоғарғы бұрышка орнатылады.

Егер BGI-файлдар ағымдағы бумада болса, *initgraph()* функциясының үшінші параметрі ретінде бос орын мәнін беруге болады

```
initgraph(&Driver,&Mode," ");
```

немесе олар C: дискісінің TC каталогындағы BGI бумасында орналасса,

```
initgraph(&Driver,&Mode,"C:\\TC\\BGI");
```

түрінде жазылады немесе оның баска нақты орны көрсетіледі.

Графикалық режимнен шығу және оған бөлініп берілген жады бөлігін босату үшін, яғни бейнеадаптер буферін тазартып, бұған дейінгі мәтіндік режимді калпына келтіру максатында мына функция колданылады

```
closegraph();
```

Графикалық режимде сызық түстерін, тұйық сызықтар ішін түрлі түске бояуға болады. Ол үшін мәтіндік режимдегідей түстердің кодтары және олардың ағылшынша атаулары колданылады.

Графикалық режимнен уакытша мәтіндік режимге ауысу максатында мынадай функция карастырылған

```
restorecrtmode();
```

Ол *initgraph()* функциясын пайдаланғанға дейін болған мәтіндік режимді қайта орнатады,буферде (экран көрінісі де) сакталған мәтін калпына келмейді, өйткені ол *initgraph()*функциясы арқылы өшірілген болатын.

Графикалық режимге қайтып оралу мына функция арқылы аткарылады

```
setgraphmode(gm);
```

Функция аргументі ретінде қолданылған драйверге қатысты режимнің бүтін сан түріндегі нөмірі карастырылады.

Курсор орнын анықтау функциялары – `getx();` `gety();`
Фуикциялары курсор тұрған орын координаталарын (*x* және *y*) береді.

```
maxx(); getmaxy();
```

getmaxx() функциясы экранның оң жақтағы ең шеткі нүктесінің *x* координатасын анықтайды, ал **getmaxy()** функциясы — экранның ең төменгі *y* нүктесінің координатасын анықтайды.

Сызықтар түсін тағайындау функциясы `setcolor(col);`

мұндағы `col` – түс атауы немесе түс коды. Сызық және мәтіндер түсі осы функциямен беріледі. Ал сызық типі **setlinestyle()** функциясы арқылы тағайындалады.

Түстер бүтін сан түріндегі кодпен немесе константа түрінде бас әріппен жазылатын ағылшынша түс атауларымен беріледі.

Түстердің стандартты белгіленулері төмендегідей:

Көмескі түстер	Кодтары	Ашық түстер	Кодтары
Қара (BLACK)	0	Қара қошқыл (DARKGRAY)	8
Көк (BLUE)	1	Көкшіл (LIGHTBLUE)	9
Жасыл (GREEN)	2	Ақ жасыл (LIGHTGREEN)	10
Көгілдір (CYAN)	3	Ақшыл көк (LIGHTCYAN)	11
Қызыл (RED)	4	Қызғылт (LIGHTRED)	12
Күлгін (MAGENTA)	5	Қызғыш (LIGHTMAGENTA)	13
Қоңыр (BROWN)	6	Сары (YELLOW)	14
Боз (LIGHTGRAY)	7	Ақ (WHITE)	15

Мысалы: `setcolor(YELLOW); setcolor(3); setcolor(5);`

Экранның фоны түсін өзгерту функциясы

```
setbkcolor (түсі);
```

Мысалы: `setbkcolor(BLUE); setbkcolor(14);`

Мәтін шығару функциялары

Графикалық режимде экранға мәтін шығарарда символдар көлемін, бағытын (тік, көлденең), бекітілген бірнеше қаріп түрінің бірін таңдау мүмкіндіктері бар. Осындай символдар параметрлері мынадай функция арқылы беріледі:

```
settextstyle(Қаріп,Бағыты,Көлемі);
```

outtextxy және outtext функциялары арқылы шығарылатын мәтіндердің қаріп түрін, көлемін және бағытын тағайындайды. **Қаріп** *(шрифт)* параметрі ретінде төмендегі константалардың бірін пайдалануға болады.

Константа	*Мәні*	*Қаріп типі*
DEFAULT_FONT	0	Стандартты. Әрбір символ 8 х 8 пиксел көлемдегі квадрат ішіне шығарылады.
TRIPLEX_FONT	1	Triplex шрифті (TRIP.CHR файлы)
SMALL_FONT	2	Майда шрифт (LITT.CHR файлы)
SANSSERIF_FONT	3	SansSerif тік шрифті (SANS.CHR файлы)
GOTHIC_FONT	4	Готикалық шрифт (GOTH.CHR файлы)

Орыс каріптері тек стандартты түрде ғана (default_font) шығарылады. **Бағыты параметрі берілген мәтінді тік немесе көлденең беруді тағайындайды, оның мәндері:**

HORIZ_DIR	0	Символдар солдан оңға қарай жазылады
VERT_DIR	1	Символдар төменнен жоғары қарай жазылады

Көлемі 1-ден 10-ға дейін өзгере алады. Бұл шама DEFAULT_FONT стандартты қарпі (8x8 пиксел) үшін әр символды неше есе үлкейту керек екендігін көрсетеді(стандартты матрица 8*8 болғандықтан, егер көлемі 4 болса, онда символдар 32*32 пикселдік матрицаға дейін үлкейеді). Ал қалған каріптер үшін бұл параметр сызықтық емес, экспоненциалдық масштабтау шкаласын көрсетеді. Символдың негізгі нұсқасы көлемі 4-ке тең болып саналады, сондықтан ол 7 болса, онда символдар 2 есе өседі; егер 8 болса - 3 есе; ал 9 болса – онда 4 есе ұлғаяды. Символдар әрқашанда үздіксіз жіңішке сызықтармен жазылады.

Экранға мәтін шығару функциясы – `outtext("Мәтін");`

қостырнақшаға алынған мәтінді курсор тұрған орыннан бастап экранға шығарады. Мәтін ішінде басқару символдары болмауы тиіс, мысалы, \n. Шығарылатын символдар түсі setcolor, шрифт типі — `settextstyle` функцияларымен беріледі. Мысалы:

```
setcolor(4); moveto(250,10);outtext("Омаров Марат");
```

Экрандағы көрсетілген орынға мәтін шығару функциясы –

`outtextxy(x,y,"мәтін");`

Курсорды алдымен x, y нүктесіне орналастырып алып барып, мәтінді экранға шығарады. Мәтін шығарылған соң, курсор бұрынғы орнында (x,y)қалады.

Шығарылатын символдар түсін – `setcolor()`, каріп типін — `settextstyle()` функциясымен беруге болады. Мысалы:

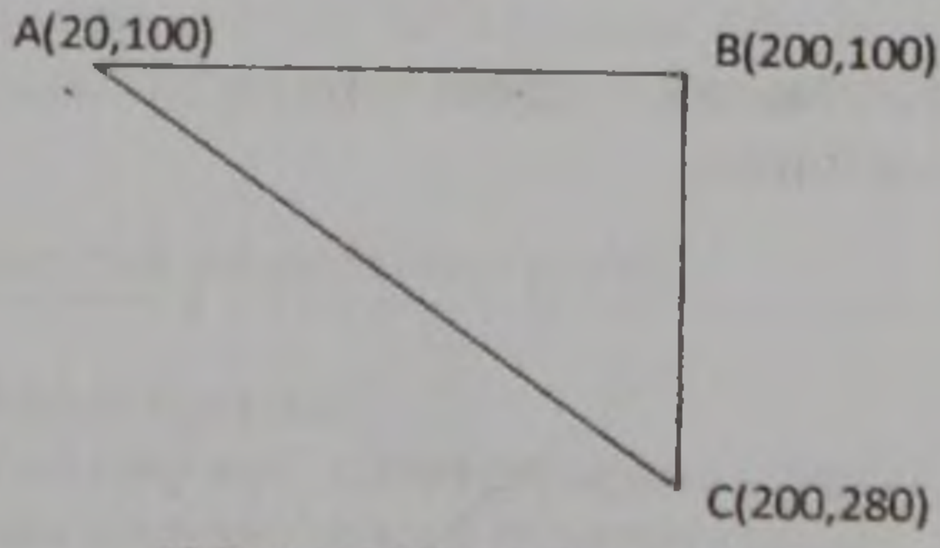
```
setcolor(4); outtextxy(250,2,"Омарова Айман");
```

Компьютер бейнежадына пиксел жазу функциясы –

`putpixel(x,y,Түсі);`

координатасы (x,y) пиксел-нүктені *Түсі* санына байланысты бояп шығады. *Түсі* ағылшынша сөзбен немесе кодпен беріледі. Бұл функцияны цикл ішіне орнатып, сызықтар сызуға болады. Мысалы:

```
setcolor(BLUE);
for(int x=20; x<=200; x++)
   putpixel(x,100,BLUE);
for(int y=100; y<=280;y++)
   putpixel(200,y,BLUE);
line(200,280,20,100);
```



10.2 сурет. Түзу сызықтар салу

Түрлі сызықтар салу үш функция арқылы орындалады:

1) `line(x1,y1,x2,y2);`

x1,y1 нүктесінен *x2,y2* нүктесіне дейін сызық сызады, бірақ мұнда курсор орны өзгермейді.

2) `lineto(x,y);`

Курсор тұрған нүктеден x,y нүктесіне дейін сызық салады, мұнда курсор бұрынғы орнынан жаңа орынға ауысады.

3) `linerel(dx,dy);`

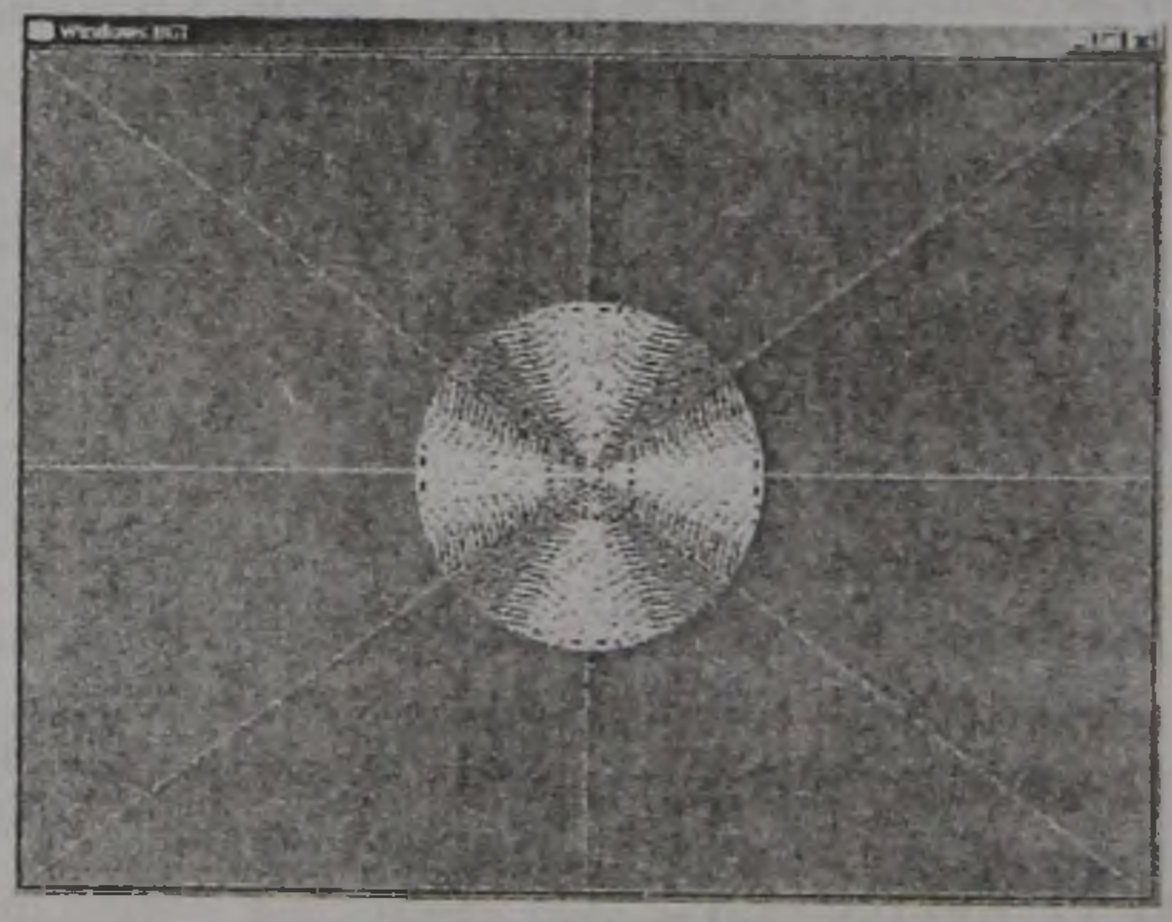
Курсор тұрған x,y нүктесінен x+dx,y+dy нүктесіне дейін сызық салады, яғни сызық координаталары салыстырмалы түрде беріледі. Мұнда да курсор бұрынғы орнынан жаңа орынға ауысады.

Енді бірнеше программалар қарастырайық.

1 мысал. Келесі программа экранға әртүрлі сызықтар салады:

```
// координаталар x=0..639, y=0..479
#include <graphics.h>
#include <conio.h>
#include <stdio.h>
int main ()
{ int gdriver=DETECT;    //драйверді анықтау
  int gmode;             //режим аты
  int xmin=1,ymin=1,xmax=637,ymax=477,R;
  initgraph (&gdriver,&gmode,"D:\\TC\\bgi");
  setbkcolor(BLUE);                  // фон көк түсті
  setcolor(YELLOW);                  // сызықтар сары түсті
  line (xmin,ymin,xmax,ymin);   // жоғарғы горизонталь сызық
  line (xmax,ymin,xmax,ymax);   // оң жақтағы тік сызық
  line (xmax,ymax,xmin,ymax);   // төменгі горизонталь сызық
  line (xmin,ymax,xmin,ymin);   // сол жақтағы тік сызық
  for(R=1; R<100;R++)           // ортада 100 шеңбер
     circle(320,240,R);
  // екі диагональ мен көлденең және тік сызықтар
  line (xmin,ymin,xmax,ymax);
  line (xmin,ymax,xmax,ymin);
  line (xmin,ymax/2,xmax,ymax/2);
  line (xmax/2,ymin,xmax/2,ymax);
  getch();
  closegraph();
}
```

Программа нәтижесі суретте көрсетілген.



10.3 сурет. Түзу сызықтар салу

Сызықтар осыған дейінгі *setcolor()* функциясы тағайындаған түспен және *setlinestyle()* функциясы арқылы орнатылған стильмен сызылады.

2 мысал. Ромб сызудың екі тәсілін келтірейік.

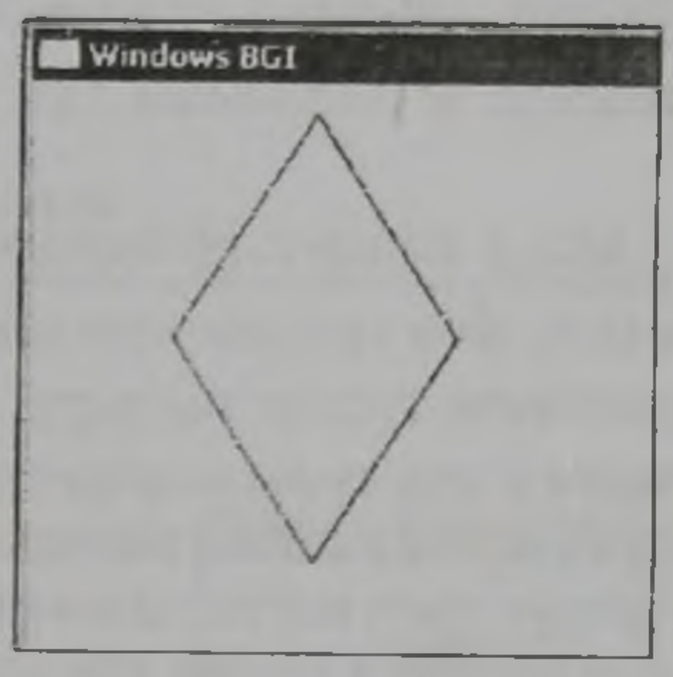
1 тәсіл

```
setcolor (GREEN);
moveto(100,10); lineto(50,90);  lineto(100,170);
setcolor(RED);  lineto(150,90); lineto(100,10);
```

2 тәсіл

```
setbkcolor (WHITE); setcolor (GREEN);
line(100,10,50,90); line(50,90,100,170);
setbkcolor (RED);
line(100,170,150,90); line(150,90,100,10);
```

Екінші тәсілдегі программаның нәтижесі 10.4-суретте көрсетілген.



10.4 сурет. Ромб салу

10.2. Сызық стильдерін беру

Геометриялық фигуралар сызықтарының қалыңдығы мен түр сипатын беру функциясы

```
setlinestyle(tip,obr,tol);
```

сызықтар стилін береді, мұндағы `tip` – алдын ала анықталған константа, ол сызық ти-пін береді. `tol` – сызық қалыңдығын анықтайтын константа (**NORM_WIDTH**– қалыпты, **THICK_WIDTH** – қалыңырақ). Егер программалаушы анықтаған сызық типі қолданылатын болса, онда **obr** мәні төрт таңбалы оналтылық константа және ұзындығы 16 пиксел сызық кесіндісін кодтау үлгісі болуы тиіс.

`tip` константасының мүмкін мәндері (сызықтық элементтер үшін)

0	SOLID_LINE	(жай сызық)
1	DOTTED_LINE	(пунктир сызық – нүктелерден тұрады)
2	CENTER_LINE	(штрих-пунктир сызық – нүктелер мен сызықшалардан тұрады)
3	DASHED_LINE	(DOTTED_LINE сызығынан ұзындау келген пунктир сызық)
4	USERBIT_LINE	(программалаушы анықтаған сызық)

`tol` – *қалыңдық* параметрінің мүмкін мәндері

1	NORM_WIDTH	(қалыңдығы бір пиксел)
3	THICK_WIDTH	(қалыңдығы үш пиксел)

obx параметрі **tip** константасымәндері 4-ке тең болғанда ғана жұмыс істейді (қалған жағдайларда ол қарастырылмайды, сондықтан оны 0 деп алуға болады). Оның көмегімен кез келген периодты түрде қайталанатын сызық түрін бере аламыз, оның периоды 16 пиксел болуы тиіс. Егер сызықта жарықтанатын пиксел керек болса, үлгіде 1-ге тең бит, болмаса – 0-ге тең бит алу керек. Сонымен, **obx** мәні төрт таңбалы оналтылық константа болып, ол ұзындығы 16 пиксел сызық кесіндісін кодтау үлгісі болуы тиіс.

Мысалы, пунктир сызық үшін алынған үлгі мынадай болуы мүмкін: 0x3333, бұл мынадай биттер тізбегіне сәйкес келеді 0011 0011 0011 0011.

Графикалық режимде экрандағы тұйық сызық іші белгілі бір түспен боялуы мүмкін.

Штрихтау сызықтарын тағайындау және тұйық аймақты бояу функциясы – setfillstyle(stil,col);

аймақты бояу, толтыру стилін береді, мұндағы **stil** – алдын ала мәні анықталған константа, стильді береді, оның мәндері төменгі кестеде көрсетілген; **col** – түс кодына сәйкес бүтін сан немесе ағылшынша түс аты;ол **setcolor** функциясын анықтауда көрсетілген.

Бояу тәсілін анықтайтын **stil** параметрінің мүмкін мәндері

EMPTY_FILL	0	аймақты фон түсімен толтыру
SOLID_FILL	1	көрсетілген түспен толтыру
LINE_FILLL	2	көлденең штрих сызықтармен толтыру
LTSLASH_FILL	3	45 градус оңға қиғаш жіңішке штрих сызықтармен толтыру
SLASH_FILL	4	45 градус оңға қиғаш қалың штрих сызықтармен толтыру
BKSLASH_FILL	5	45 градус солға қиғаш жіңішке штрих сызықтармен толтыру
LTBKSLASH_FILL	6	45 градус солға қиғаш қалың штрих сызықтармен толтыру
HATCH_FILL	7	торсызықты штрихтармен толтыру
XHATCH_FILL	8	45 градус оңға қиғаш сирек штрих сызықтармен толтыру
INTERLEAVE_FILL	9	45 градус қиғаш жиі штрих торсызықтармен толтыру
WIDE DOT_FILL	10	сирек нүктелермен толтыру
CLOSE_DOT_FILL	11	жиі нүктелермен толтыру

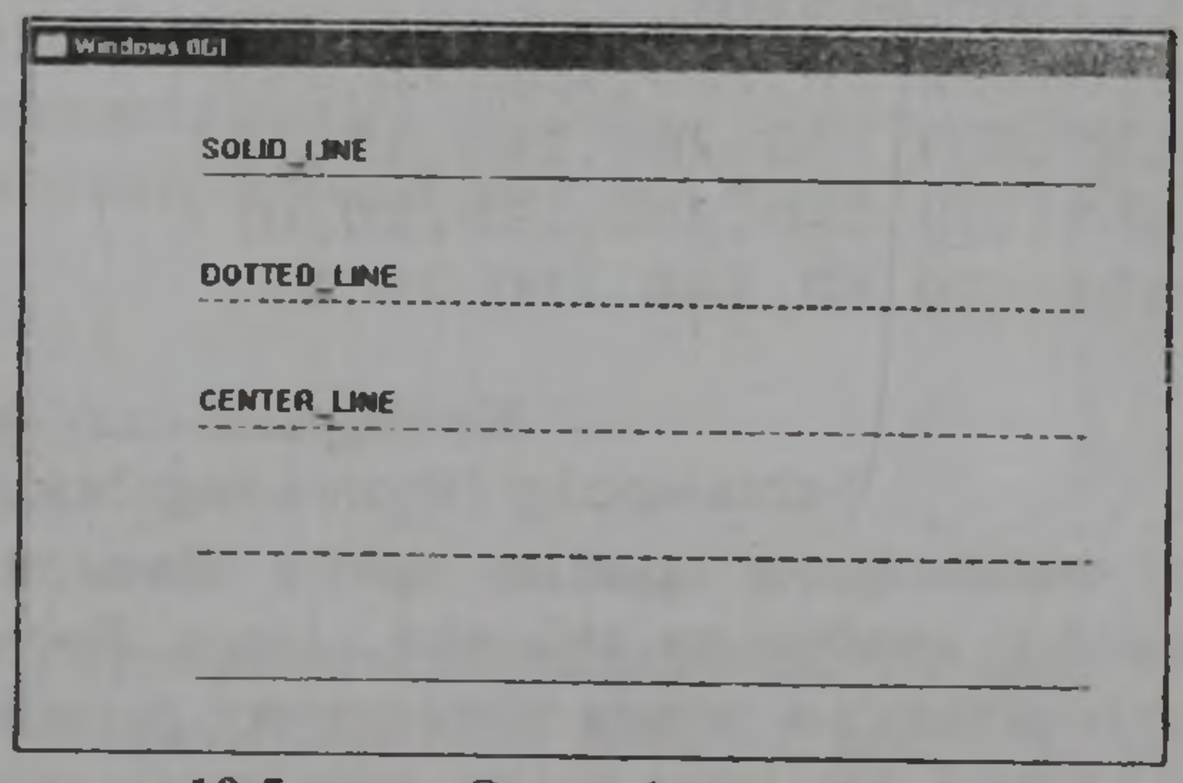
Сызық стилі мен түсі көптеген функцияларда (**bar, bar3d, sector**, т.б.) колданылады. Енді бір мысал келтірейік.

3 мысал. Әртүрлі сызықтар салу.

```
// Әртүрлі сызықтар салу мысалы
#include <graphics.h>
#include <conio.h>
#include <stdio.h>
main()
{
int gdriver=DETECT;
int gmode;
int x,y,xk=500;
initgraph (&gdriver,&gmode,"C:\\TC\\bgi");
```

```
setcolor(RED);  x=80; y=30;
moveto(x,y); outtext("SOLID_LINE"); // жай сызық
setlinestyle (SOLID_LINE,0,NORM_WIDTH);  y+=20;
line(x,y,xk,y);
y+=40; moveto(x,y);
outtext("DOTTED_LINE");   // үзік сызық
setlinestyle (DOTTED_LINE,0,NORM_WIDTH);
y+=20;  line(x,y,xk,y);
y+=40;  moveto(x,y);
outtext("CENTER_LINE"); // штрих-пунктир сызық
setlinestyle (CENTER_LINE,0,NORM_WIDTH);
y+=20;  line(x,y,xk,y);
y+=40;  moveto(x,y);
// келесі - штрихі ұзынша пунктир сызық
setlinestyle (DASHED_LINE,0,NORM_WIDTH);
y+=20;  line(x,y,xk,y);
y+=40;  moveto(x,y);
// келесі - қалың сызық
setlinestyle (SOLID_LINE,0,THICK_WIDTH);
y+=20; line(x,y,xk,y);
getch();
closegraph();
}
```

Программа жұмысының нәтижесі 10.5 суретте көрсетілген.



10.5 сурет. Әртүрлі сызықтар салу

10.3. Тұйық сызықтар салу

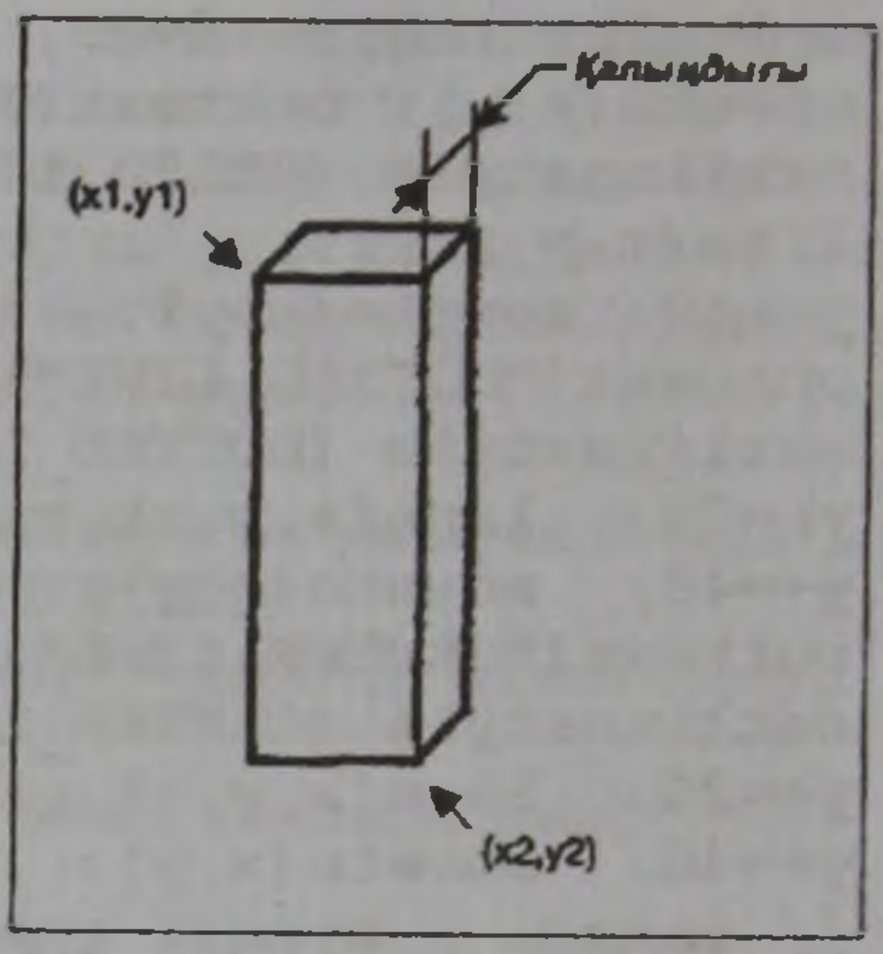
Тіктөртбұрыш контурын салу үшін `rectangle(x1,y1,x2,y2);` функциясы қолданылады, ол сол жақ жоғарғы бұрышы – *x1,y1*, оң жақ төменгі бұрышы – *x2,y2* болып келетін төртбұрыш сызады. Сызық типі setlinestyle функциясымен, ал түсі – setcolor функциясымен анықталады.

Іші боялған тіктөрбұрыш салу үшін

```
bar(x1,y1,  x2,y2);
```

функциясы пайдаланылады, *x1, y1* сол жақ жоғарғы бұрыштың, ал *x2* , *y2* – оң жақ төменгі бұрыштың координаталары.

Мұның қабырғалары өстерге параллель болады, контуры көрсетілмейді. Оны бояу стилі *setfillstyle()* функциясымен анықталады.



10.6 сурет. Параллелепипед салу

Алдыңғы жақ беті боялған параллелепипед салу функциясы

```
bar3d(x1,y1, x2,y2,
      Қалыңдығы, Ж_Беті);
```

сыртқы контуры сызылған паралелепипед салады (10.6 сурет), мұндағы *x1,y1* параметрлері сол жақ жоғарғы бұрыштың, ал *x2,y2*– оң жақ төменгі бұрыш координаталары.

Қалыңдығы параметрі – алдыңғы және арткы жақтарының ара қашықтығы, егер ол 0 болса, онда айналасы сызылған төртбұрыш салынады.

Ж_Беті параметрі – жоғарғы жақ шеттерін сызуды (1 – TOP_ON) көрсетеді. Егер ол 0 (TOP_OFF) болса, сызықтар сызылмайды (бірінің үстіне бірі қойылған бірнеше параллелепипед салу кезінде қажет болады).

Мысалы, бірінің үстіне бірі қойылған бірнеше паралелепипед салу:

```
setbkcolor (WHITE);
setcolor(GREEN);
bar3d(230,50,250,150,15,1);
bar3d(220,150,260,180,15,1);
bar3d(300,150,340,180,15,0);
bar3d(300,50,340,150,15,1);
```

Көпбұрыш салу функциясы

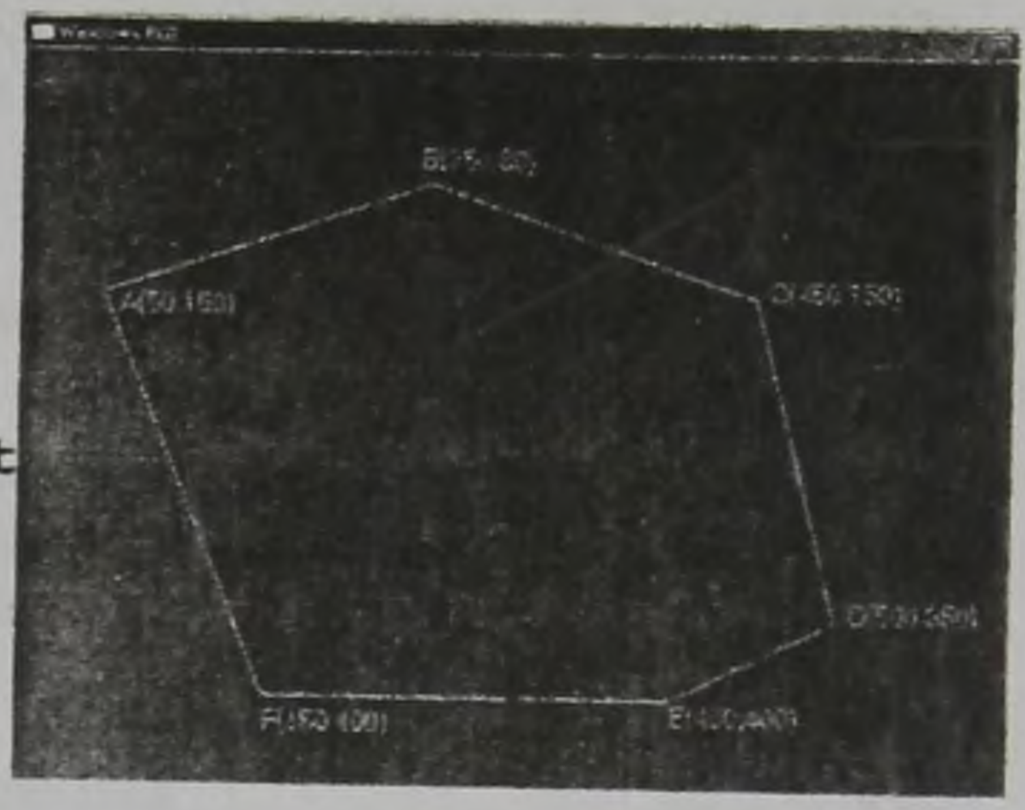
```
drawpoly(НүктелерСаны,Координаталары);
```

түзу сызықтардан тұратын тұйық аймақты көпбұрыш сызады. *НүктелерСаны* параметрі көпбұрыш төбелері санын, *Координаталары* параметрі сол төбелер координаталарын жиым элементтері ретінде береді. Жиымның 0-ші және 1-ші элементтері алғашқы нүкте координаталары, 2-ші, 3-ші элементтері – екінші нүкте координаталары, т.с.с. Енді осы функцияны пайдаланып, алты бұрыш салайық.

4 мысал. Экранға көпбұрыш бейнесін салу.

```
#include <graphics.h>
#include <conio.h>
#include <stdio.h>
main()
{
int gdriver=DETECT, gmode;
initgraph (&gdriver,&gmode,"C:\\TC\\bgi");
int x,y,t[14]={450,150, 500,350, 400,400,150,400,
50,150,250,80, 450,150};
setcolor(WHITE);
```

```
setbkcolor(BLUE);drawpoly(7,t);
settextstyle(3,HORIZ_DIR,1);
outtextxy(458,135,"C(450,150)");
outtextxy(508,335,"D(500,350)");
outtextxy(400,400,"E(400,400)");
outtextxy(150,400,"F(150,400)");
outtextxy(60,147,"A(50,150)");   out
getch();
closegraph();
}
```



10.7 сурет. Көпбұрыш салу

Программа жұмысының нәтижесі 10.7-суретте көрсетілген.

Іші боялған көпбұрыш салу үшін қолданылатын функция

`fillpoly(НүктелерСаны,координаталары);`

мұндағы **НүктелерСаны** – төбелердің саны, **координаталары** – жиым элементтері түрінде берілген төбе координаталары. Әрбір төбе координатасы екі бүтін санмен беріледі. Бұл функция төбелердің алғашқы нүктесін соңғы нүктесімен қосып, сызықтарды тұйықтап, ішін бояйды. Сызық типі мен бояу түсі *setfillstyle()* және *setfillpattern()* функцияларымен анықталады.

Шеңбер сызу функциясы – `circle(x,y,r);`

радиусы *r* (бүтін сан), центрінің координаталары (x, y)болатын шеңбер сызады. Сызық түсі `setcolor()` функциясымен беріледі. Мысалы, қызыл түсті 5 шеңбер сызайық:

```
...
setcolor(RED);
for(r=5; r<=25; r+=5)
  circle(320,240,r);   ...
```

Доға сызу функциясы – `arc(x,y,БұрышБасы,БұрышСоңы,Радиус);`

центрінің координатасы *(x, y)*, *радиусы* берілген доға сызады. **БұрышБасы,*БұрышСоңы*** параметрлері бұрышты градуспен горизонталь *x* өсінен бастап, сағат тіліне қарсы бағытта береді. Радиуспараметрідоға радиусын бүтін санмен береді. Бұрыштар мәні периодына сәйкес эквивалентті түрде [0..360] интервалындағы мәндерге келтіріледі. Мысалы, **arc(x,y,-45,45,r)** және **arc(x,y,675, -315,r)** шеңбердің бір ширегіне сәйкес бір доғаның екі түрде берілуін көрсетеді.

Эллипс доғасын сызу функциясы –

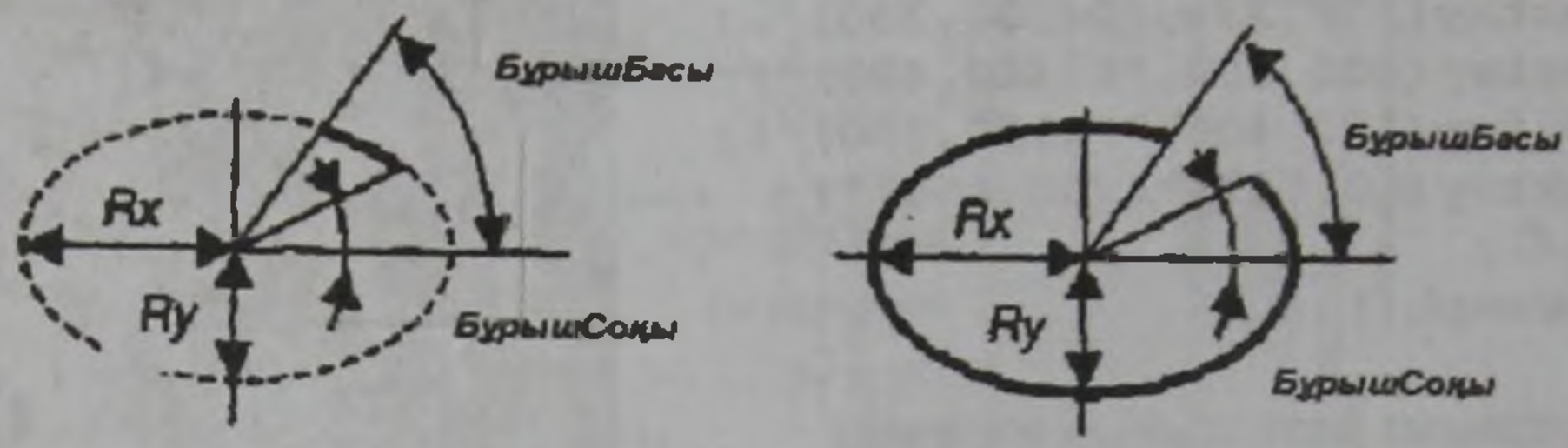
`ellipse(x,y,БұрышБасы,БұрышСоңы,rx,ry);`

центрінің координаталары *(x, y)* эллипс немесе эллипс доғасын (10.4 сурет) сызады. **БұрышБасы,БұрышСоңы** параметрлері доғаның басы мен соңын градуспен сағат тіліне қарсы береді. `rx`, `ry` параметрлері эллипстің көлденең және тік радиустарын береді. Эллипс өстері координаталар өстеріне параллель болады.

Іші боялған контурлы эллипс салу функциясы –

`fillellipse (x,y, rx,ry);`

мұндағы *x, y* – центр координаталары; *rx, ry* – пикселмен берілген эллипс жарты өстері радиустары. Эллипс өстері координат өстеріне параллель болады. Эллипс ағымдағы түспен боялып шығады.



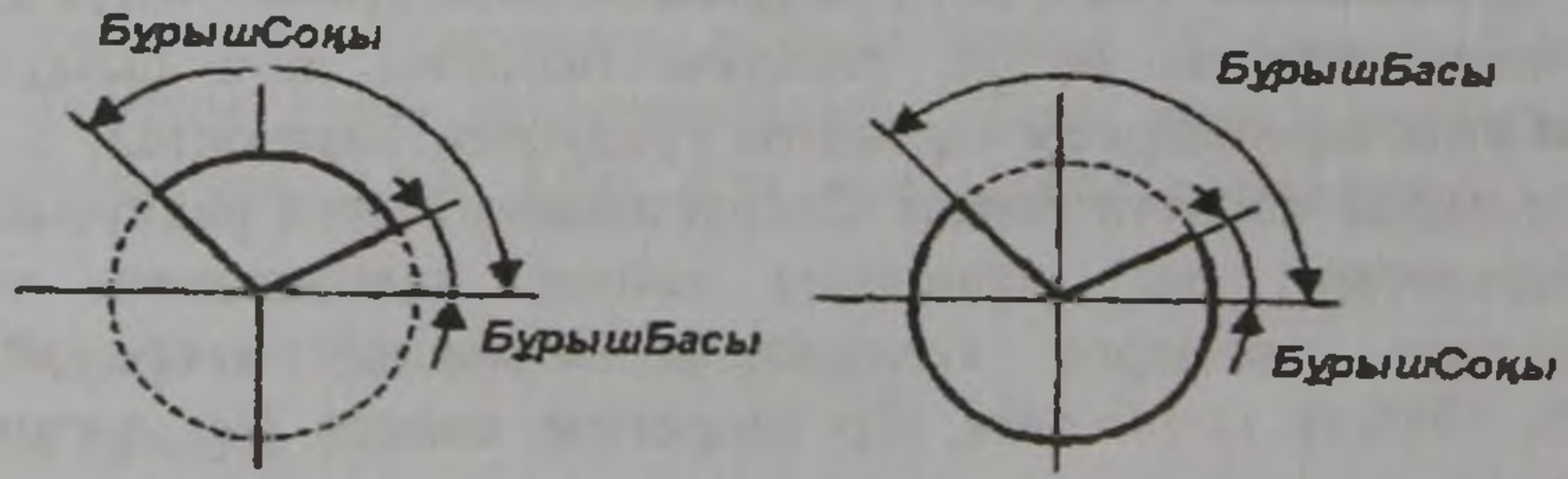
10.8 сурет. Эллипс доғасын салу

Іші боялған дөңгелек сектор сызу функциясы

```
pieslice(x,y,БұрышБасы,БұрышСоңы,Радиус);
```

радиусы *Радиус*, центрі (x,y) нүктесіндегі дөңгелек сектор сызады. **БұрышБасы, БұрышСоңы**параметрлері шеңбер секторының бастапқы және соңғы бұрыштарын градуспен сағат тіліне қарсы анықтайды. Егер **БұрышБасы** = 0, ал **БұрышСоңы** = 360 болса, онда **pieslice** функциясы шеңбер сызып шығады. Бұрыштарды [0..360] шегіне (диапазонына) келтірген соң, сектор мәні кіші бұрыштан мәні үлкен бұрышқа қарай сызылады, сол себепті OX өсінің оң жақтағы бағытын кесіп өтетін сектор салуға болмайды. Сектор контуры (доға мен екі радиус) сектор боялған соң сызылады, ал сызық типі мен қалыңдығы *setlinestyle()* функциясымен анықталады. Егер контурсыз сектор салу керек болса, мынадай тәсілді пайдалануға болады (10.5 сурет):

```
setcolor (BLACK); setbkcolor (BLUE);
setwritemode (XOR_PUT);
setfillstyle (WIDE_DOT_FILL, RED);
pieslice(200,100,45,90,50);
```



10.9 сурет. Секторлар салу

Іші боялған контурлы эллипс секторын салатын функция

```
sector (x,y,бұрыш_басы,бұрыш_соңы,rx,ry);
```

Бұл функция *pieslice()* функциясы тәрізді жұмыс істейді.

5 мысал. Экранға секторлар шығару

```
// Жазуы бар секторлар
#include <stdio.h>
```

```
#include <stdlib.h>
#include <conio.h>
#include <graphics.h>
void main()
{
   int gd=DETECT,gm,i,x,y;
   initgraph (&gd,&gm,"c:\\TC\\bgi");
   setcolor(BLACK); setbkcolor(BLUE);
   setwritemode(XOR_PUT);
   setfillstyle(WIDE_DOT_FILL,RED);
   setbkcolor(BLUE); setcolor(RED);
   setfillstyle(1,3);
   x=getmaxx()/2; y=getmaxy()/2;
   pieslice(x,y,270,360,100);
   setfillstyle(1,2);
   pieslice(x,y,0,270,100);
   settextstyle(1,0,2);
   moveto(x-20,y-40);
   outtext("75%");
   moveto(x+20,y+20);
   outtext("25%");
   getch();
   closegraph();
}
```



10.10 сурет. Секторлар салу

Программа жұмысының нәтижесі 10.10 суретте көрсетілген.

Тұйық сызықпен қоршалған аймақтың ішін бояу функциясы

```
floodfill(x,y,шекара);
```

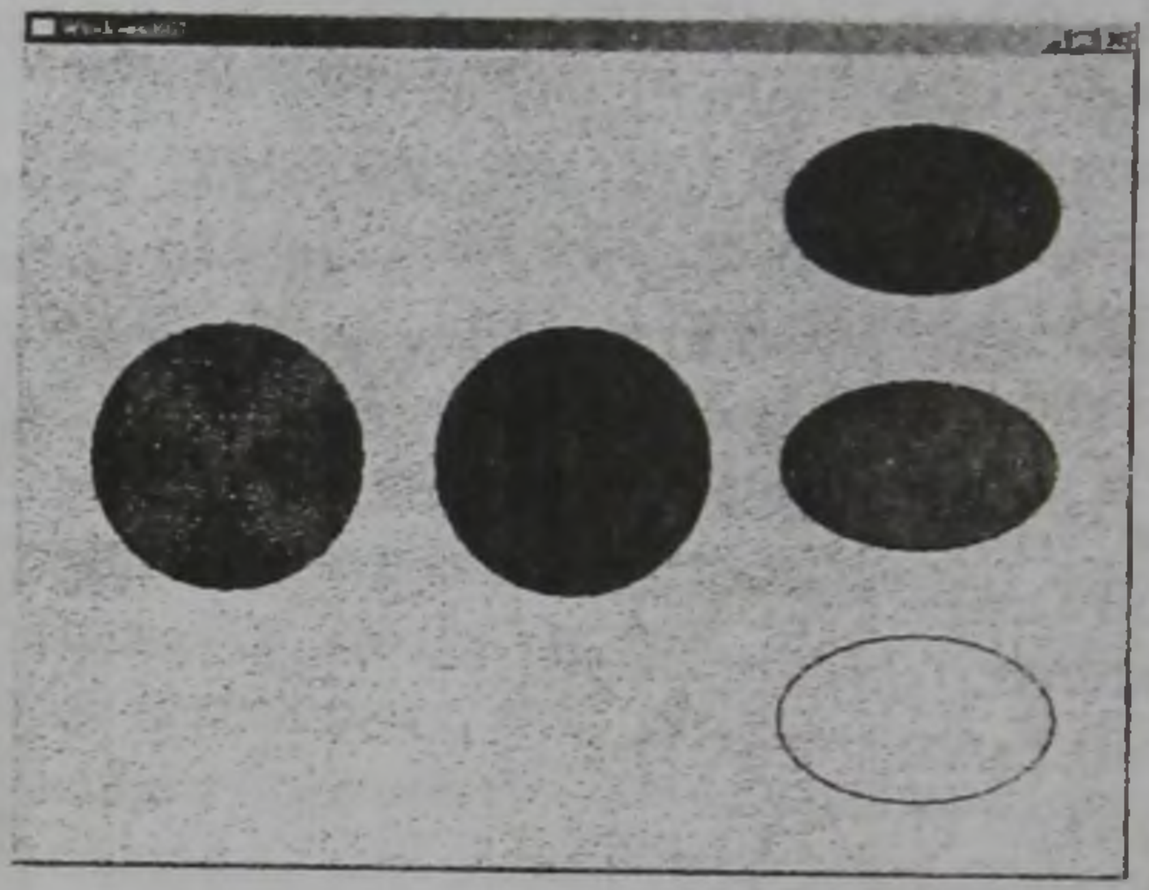
мұндағы *x, y* – боялатын аймақ ішіндегі нүкте координатасы. Тұйық аймақты қоршаған сызық контурында тесік болмауы тиіс, әйтпесе бояу бүкіл экранды сол түске бояп жібереді. Контур түсі *шекара* түсімен бірдей болуы тиіс. Бояу түсі мен типі *setfillstyle()* функциясымен орнатылады. Енді бір мысал келтірейік.

6 мысал. Экранға эллипстер салу

```
//боялған шеңберлер, эллипстер салу
#include <conio.h>
#include <graphics.h>
void main()
{ int gd=DETECT,gm,r,x=120,y=240;
  initgraph(&gd, &gm,"C:\\TC\\BGI");
  setcolor(RED);          //сызықтар қызыл түсті
  setbkcolor(YELLOW); //фон сары түсті
  for(r=0; r<80; r++) //концентрлі
    circle(x,y,r);        //80 қызыл шеңбер салу
  setfillstyle(SOLID_FILL,RED);
```

```
    x+=200;  circle(x,y,80);  // қызыл шеңбер
    floodfill(x,y,RED);       // іші де қызыл
    x+=200;  fillellipse(x,y-150,80,50); //қызыл эллипс
    setfillstyle(SOLID_FILL,GREEN);
    fillellipse(x,y,80,50);   //жасыл эллипс
    setfillstyle(SOLID_FILL,YELLOW);
    fillellipse(x,y+150,80,50); //сары эллипс
    getch();
    closegraph();
}
```

Программа жұмысы нәтижесі 10.11 суретте көрсетілген.



10.11 сурет. Эллипстер салу

Терезе ашу функциясы

Графикалық режимде экран ішінде өз координаталық жүйесі бар төртбұрышты басқа терезені ашуда қолданылатын келесі функция колданылады:

```
setviewport(x1,y1,x2,y2,clip);
```

Мұндағы *x1, y1* – терезенің сол жақ жоғарғы бұрышы координаталары; *x2, y2* – оң жақ төменгі бұрышының координаталары; *clip* – кию параметрі. Егер *clip* параметрі 1 болса, онда терезеге сыймайтын бейне элементтері киылып алынып тасталады; ал егерде ол 0 болса, терезе шекаралары есепке алынбай, бейне толық экранда көрсетіледі. Бұл функция дұрыс орындалса, графикалық курсор терезенің координаталар басына орналасады.

Терезені тазалау функциясы

```
clearviewport();
```

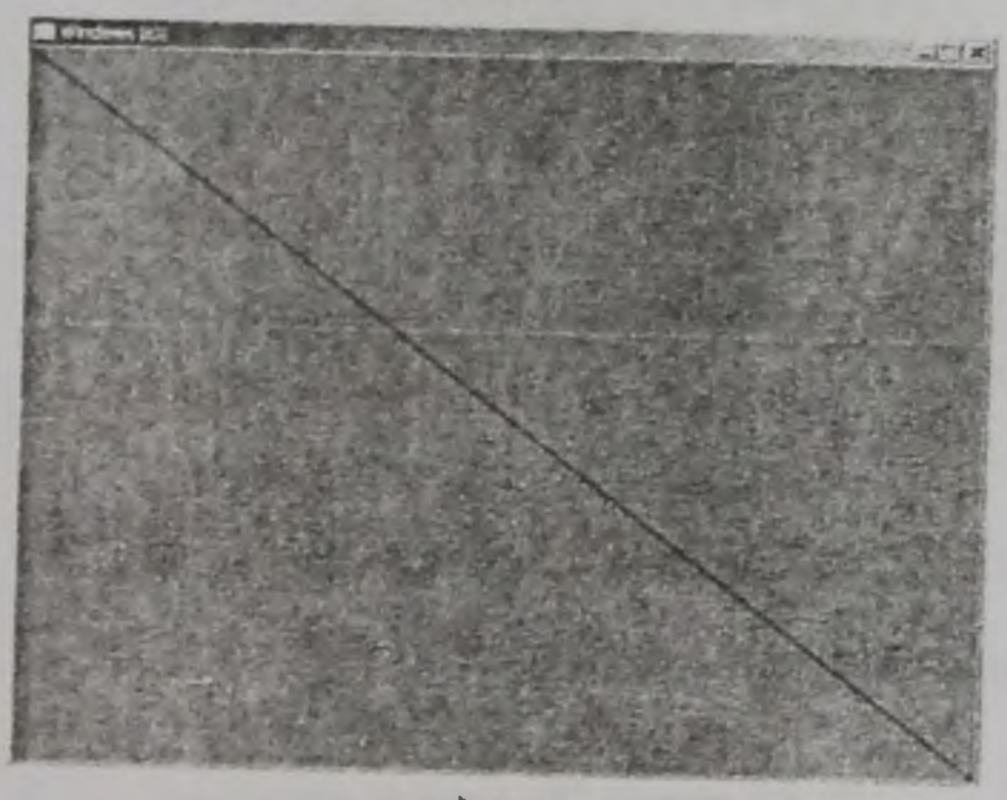
Енді осы функцияларды пайдаланып, бір мысал келтірейік.

7 мысал. Экранда терезе ашу

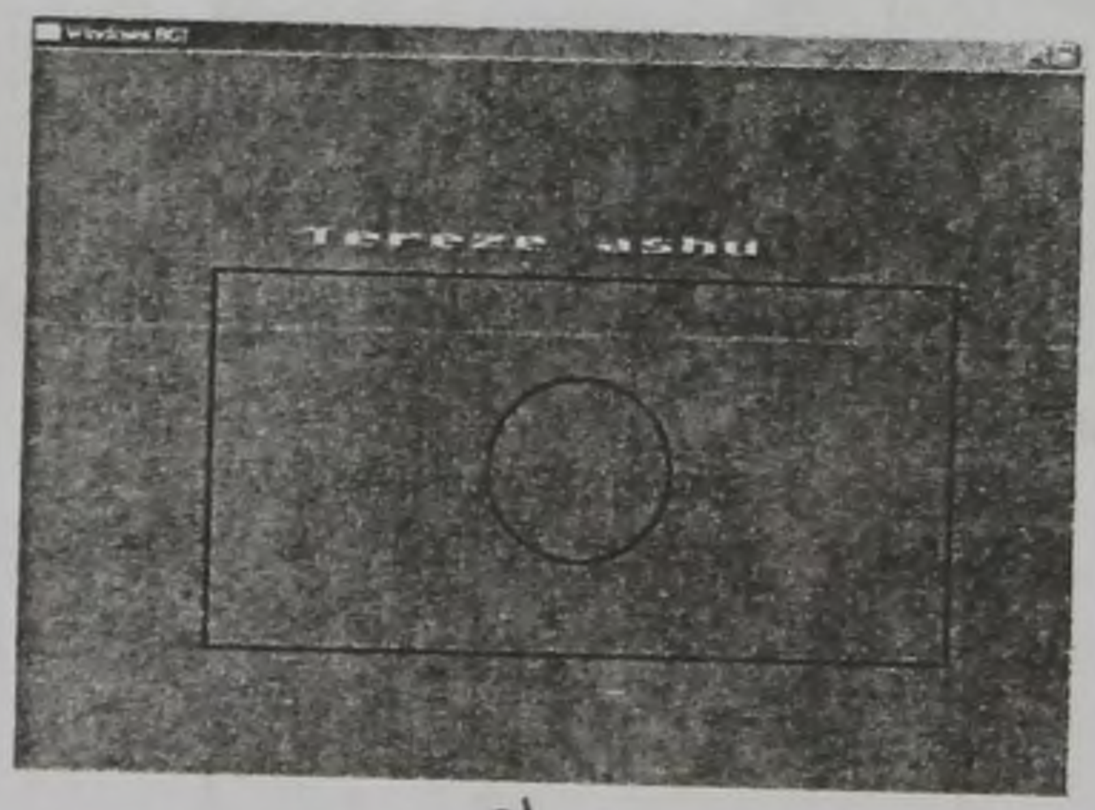
```
//Диагональ сызу,жаңа терезелер ашу
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <graphics.h>
void main()
{
int gd=DETECT,gm,x,y;
```

```
initgraph (&gd,&gm,"C:\\TC\\bgi");
//жасыл экранда қалыңдығы 3 пиксел диагональ сызу
setbkcolor(GREEN); setcolor(RED);
setlinestyle(SOLID_LINE,0,3);
x=getmaxx(); y=getmaxy();
line(0,0,x,y); // бір перне басып,экран тазалау
getch();
cleardevice(); // терезе ашу, оны тазалау
setviewport(100,100,600,400,1);
clearviewport();  setcolor(GREEN);
rectangle(10,30,450,255);
circle(230,145,55);
settextstyle(DEFAULT_FONT,HORIZ_DIR,3);
//орыс әріптері тек DEFAULT_FONT қарпінде
setbkcolor(GREEN);
setcolor(BLUE);
//Мәтін терезенің сол жақ жоғары бөлігінде
outtextxy(1,1,"Tereze ashu");
getch();
closegraph();
}
```

Программаны орындау нәтижесі 10.12 суретте көрсетілген, онда алдымен сол жақтағы сурет шығады, сонан соң Enter пернесін басқаннан кейін, оң жақтағы сурет пайда болады.



а)



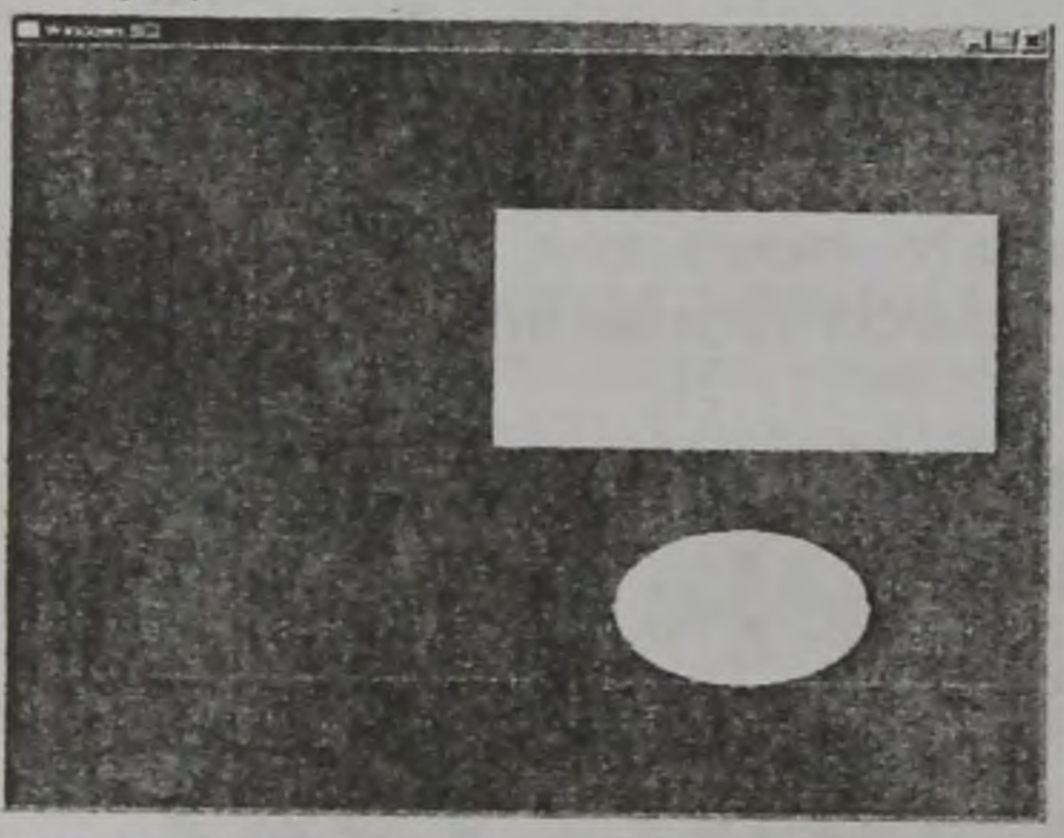
ә)

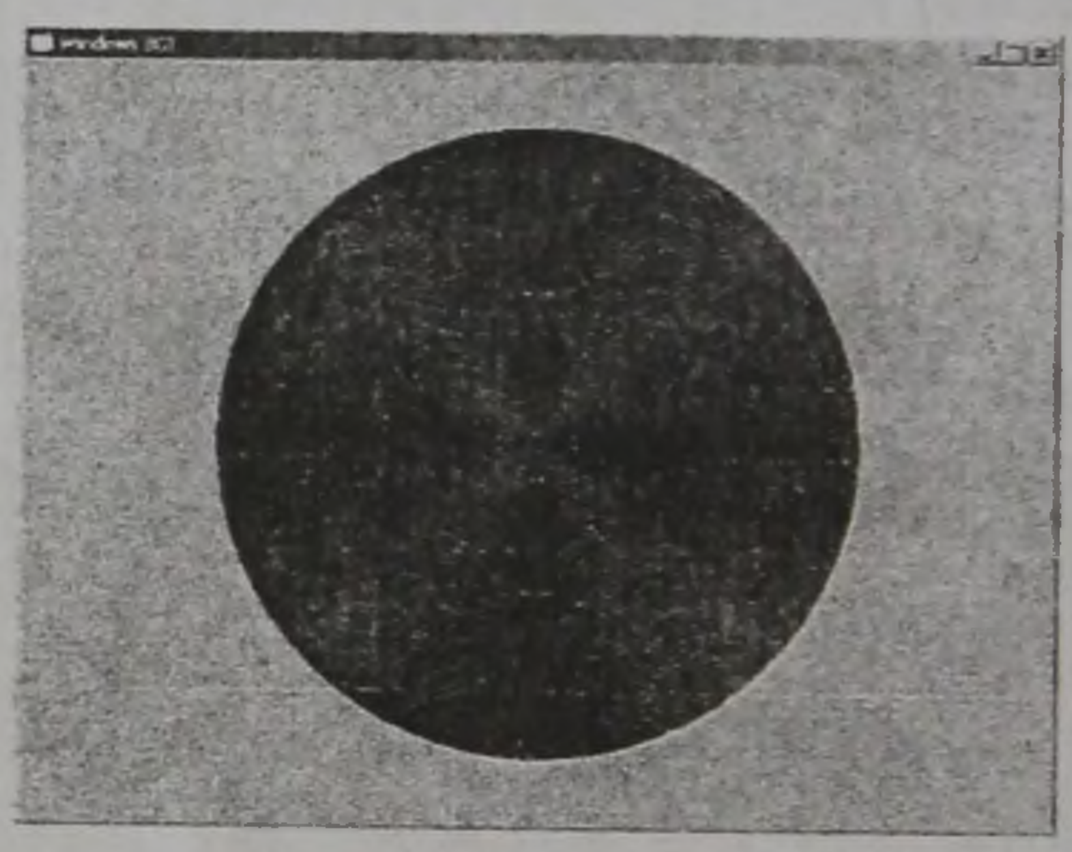
Келесі мысалда түстерді өзгерту тәсілдері көрсетілген.

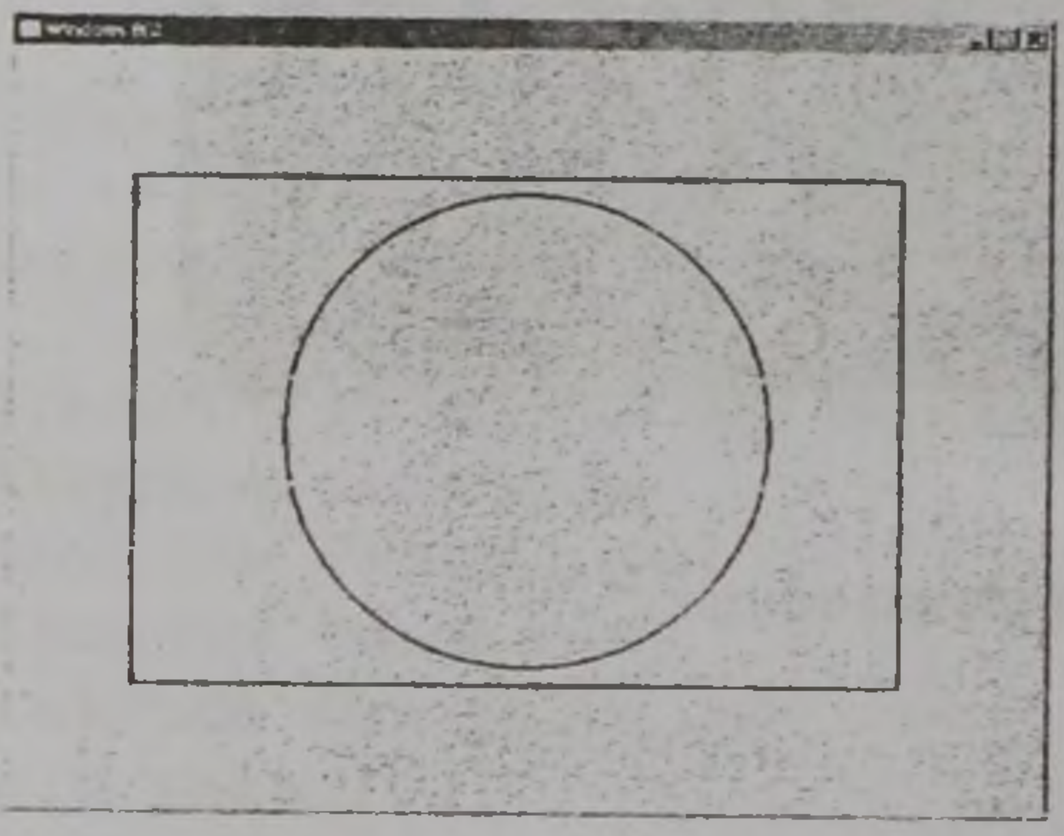
8 мысал. Экрандағы түстерді өзгерту

```
// Түстерді өзгерту мысалдары
#include <conio.h>
#include <graphics.h>
void main ( )
```

```
{   int gd=DETECT,gm,r;
    initgraph(&gd, &gm,"C:\\TC\\BGI");
    setbkcolor(GREEN); setcolor(RED);
    rectangle(100,100,250,250);
    bar(300,100,610,250);
    circle(175,350,95);
    fillellipse(455,350,80,50);
   getch(); cleardevice();
   setbkcolor(LIGHTGRAY);
   setcolor(BLUE);
   for(r=1; r<200; r++)
    circle(320,240,r);
   getch(); cleardevice();
   setbkcolor(YELLOW);
   setcolor(BLUE);
   rectangle(75,80,550,400);
   circle(320,240,150);
   getch();
   closegraph();
   }
```







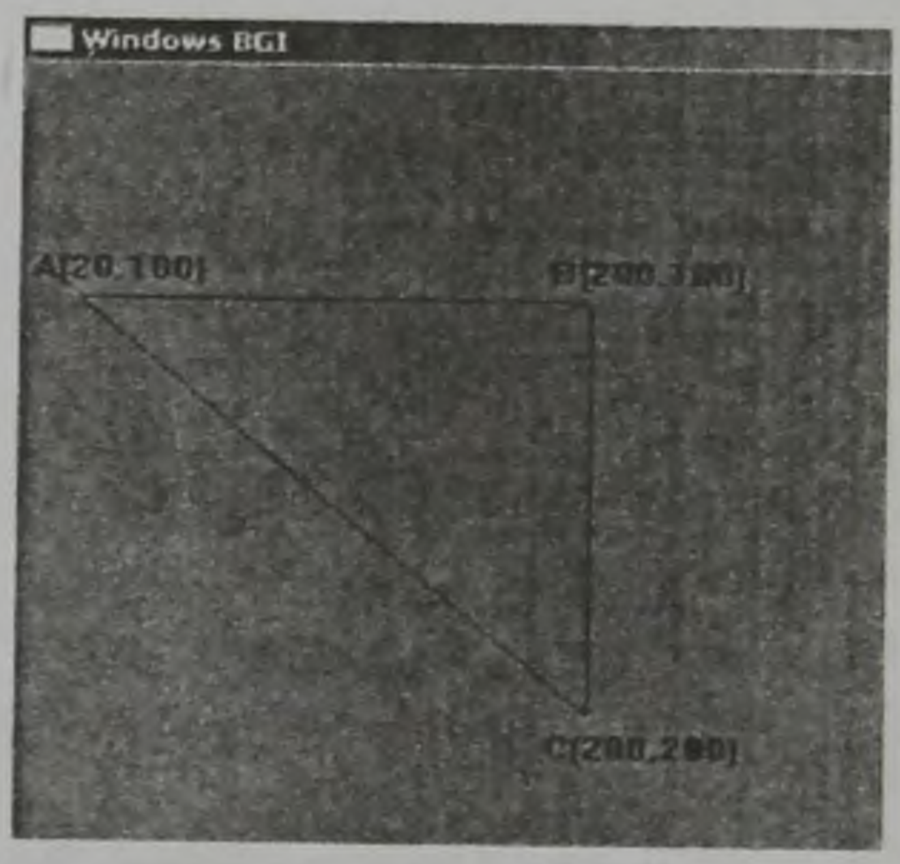
10.13 сурет. Түстерді өзгерту мысалдары нәтижесі

8 мысал. Жоғарыдағы 10.2 суреттегі үшбұрышты салу программасының толықтырылған нұсқасын келтірейік:

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
void main()
{ int gdriver = DETECT;   // драйвер
  int gmode;              // режим аты
  int errorcode;          // қате коды
  initgraph (&gdriver,&gmode,"D:\\TC\\bgi\\");
  errorcode = graphresult();
  if (errorcode != grOk) //іске қосу қатесі
  { printf("Error: %d\n", errorcode);
    puts("Press ENTER for END");
    getch ();
    return 0;
  }
setbkcolor(GREEN);
setcolor(BLUE);
for(int x=20; x<=200; x++)
  putpixel(x,100,BLUE);
for(int y=100; y<=280;y++)
  putpixel(200,y,BLUE);
line(200,280,20,100);

 putpixel(20,100,GREEN);
 putpixel(200,100,GREEN);
 putpixel(200,280,GREEN);

 outtextxy(5,90,"A(20,100)");
 outtextxy(185,90,"B(200,100)");
 outtextxy(185,286,"C(200,280)");
 getch();
 closegraph ();
}
```



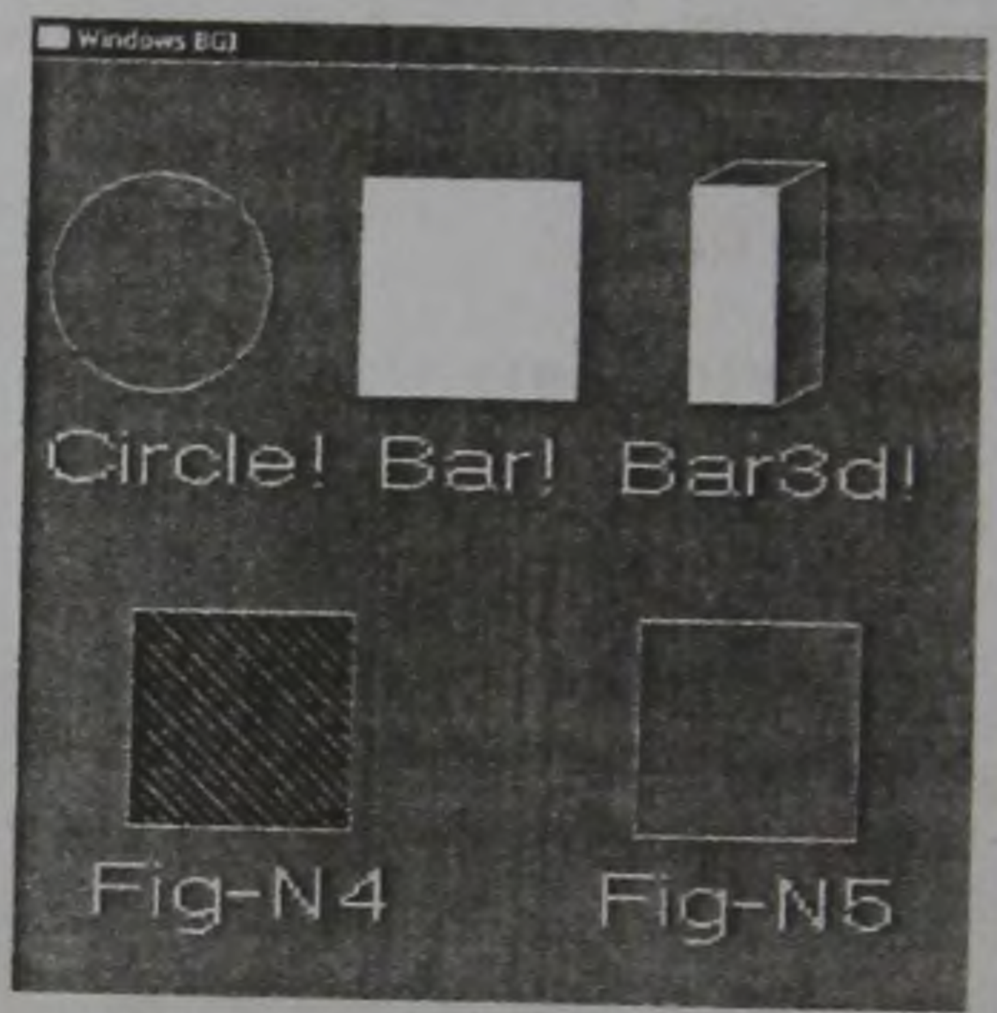
10.14 сурет. Экранға үшбұрыш салу мысалының нәтижесі

Программа жұмысының нәтижесі 10.14 суретте көрсетілген.

9 мысал. Экранға бір фигура шығарылады, сонан соң Enter пернесін бассақ, келесі фигура шығарылады, т.с.с.

```
#include <stdio.h>
#include <graphics.h>
#include <conio.h>
#include <process.h>
void main(void)
{ int g_driver,g_mode,g_error;
  int x,y;
  g_driver=9;g_mode=2;
  initgraph(&g_driver,&g_mode,"c:\\tc\\bgi");
```

```
g_error=graphresult();
if(g_error!=grOk)
    {puts("error");
     printf("\n
     error=%d,reason=%s\n",g_error,
                grapherrormsg(g_error));
     getch();
     exit(1);
    }
setbkcolor(1);
setcolor(14);
settextstyle(3,0,5);
x=60;y=100;
circle(x,y,50);
outtextxy(8,160,"Circle!");
getch();
bar(150,50,250,150);
outtextxy(162,160,"Bar!");
getch();
bar3d(300,50,340,150,20,1);
outtextxy(280,160,"Bar3d!");
getch();
setfillstyle(6,12);
rectangle (50,250,150,350);
floodfill(100,300,14);
outtextxy(40,360,"Fig-N4");
getch();
setfillstyle(8,12);
rectangle (280,250,380,350);
floodfill(350,250,14);
outtextxy(270,360,"Fig-N5");
getch();
closegraph();
}
```



10.15 сурет. Әртүрлі фигураларды экранға шығару нәтижесі

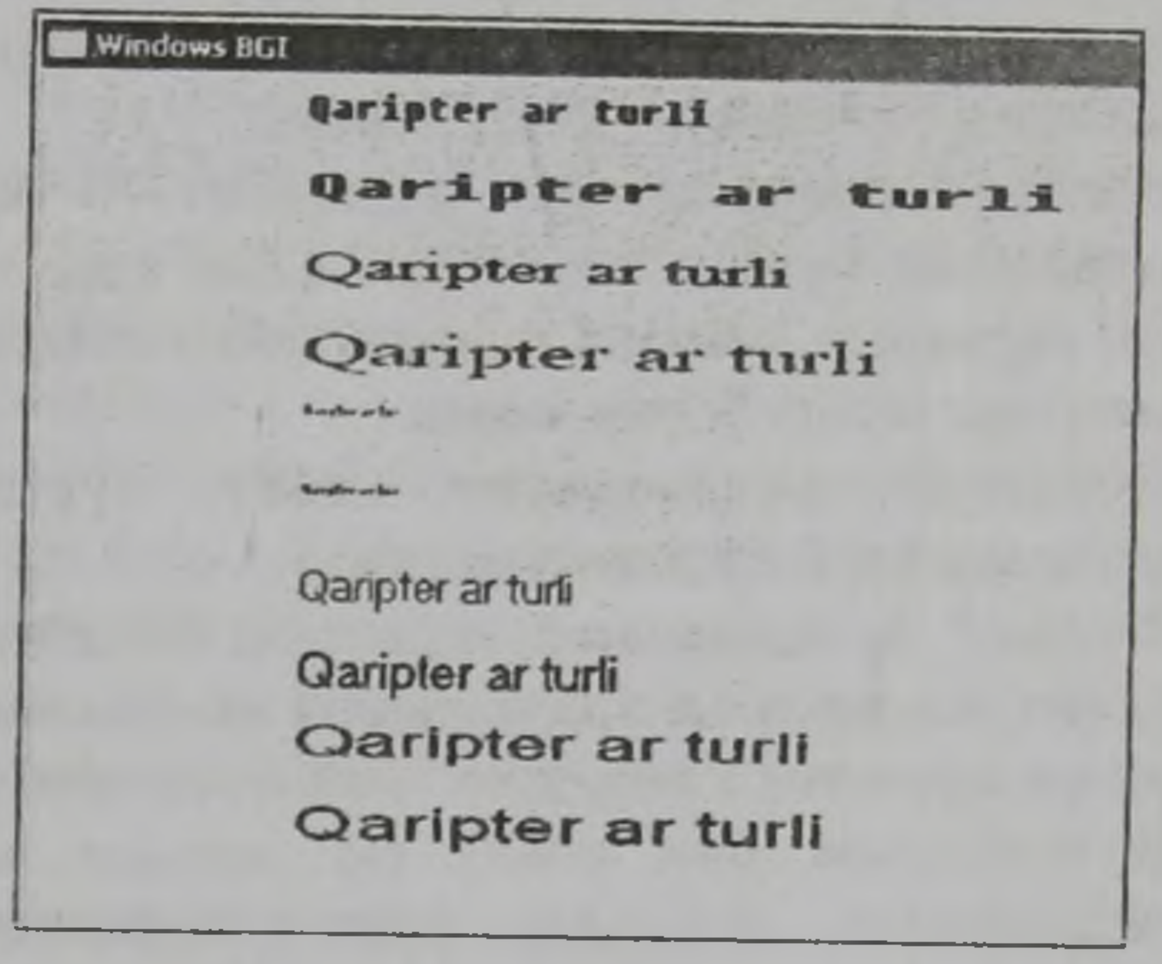
Программа жұмысының нәтижесі 10.15 суретте көрсетілген.

10 мысал. Экранға әртүрлі қаріптерді (шрифтерді) шығару

```
// координаталар x=0..639, y=0..479
// Көлденең жазылған қаріп түрлері
#include <graphics.h>
#include <conio.h>
#include <stdio.h>
int main ()
{ int gdriver=DETECT;              // драйверді анықтау
  int gmode,i,j,k=5;               // режим аты
  initgraph (&gdriver,&gmode,"C:\\TC\\bgi");
  setcolor(RED);
  for(i=0; i<=4; i++)      // i- qarip turi
    for(j=1; j<=2; j++)    // j- qarip kolemi
```

```
    { moveto(100,k);
      settextstyle(i,0,j);
      outtext("Qaripter ar turli");
      k+=30;
    }
  getch();
  closegraph();
}
```

Программа жұмысының нәтижесі 10.15 суретте көрсетілген.



10.15 сурет. Әртүрлі қаріптерді экранға шығару нәтижесі

Бақылау сұрақтары

1. *IBM PC компьютерлері бейнемониторының қандай түрлері бар? Олардың айырмашылығы неде?*
2. *Графикалық режимнің атқаратын қызметі қандай? Оны программада қалай іске қосады?*
3. *Графикалық режим қандай функция арқылы орнатылады?*
4. *Адаптерлердің мынадай типтерінің CGA, EGA, VGA бір-бірінен айырмашылығы неде?*
5. *Монитордың мәтіндік және графикалық режимдерінің мүмкіндіктері неге әртүрлі болады?*
6. *Драйвер деп нені айтады? Графикалық драйвер ше? Ол қандай қызмет атқарады?*
7. *Экранның түсін және оған шығарылатын жол символдарының түсін қалай өзгертуге болады?*
8. *Графикалық режимде курсор бола ма?*
9. *Символдарды қалай жыпылықтатып қоюға болады?*
10. *Курсорды экранның кез келген нүктесіне қалай көшіреміз?*
11. *Графикалық режимде экранда қалай терезе құруға болады?*
12. *Графикалық режимде экранды қалай тазартуға болады?*
13. *Clrscr, Clreol, Delline және Insline функцияларының ортақ қасиеттері мен айырмашылықтары.*

ТАПСЫРМАЛАР

1. Уақытты есептеп отыратын сықыршымен кездестіңдер делік және ол сендерге жұлдыздар сырын ашып береді деп ойлайық. Неге аспанда жұлдыздар жымыңдайды? Аспандағы әрбір жұлдыз — бір адамның өмірі. Сендерге көне жұлдыздарды дәл уақытында өшіріп, жаңасын дер кезінде жағып отыру керек. Уақыт есептеушінің жұмысын көрсететін программа жазу керек. Есеп модель құрудан басталады: экрандағы әрбір жұлдыз нүктемен көрсетіледі және барлығы 20 жұлдыз жанып тұрсын делік (кездейсоқ түрде таңдап алынған): жанып тұрған бірінші нүктені өшіріп, оны кез келген басқа орыннан жандыру керек, содан кейін екіншісін, т.с.с. Жиырмасыншы жұлдыз өшкеннен кейін қайтадан біріншісіне көшу керек.
2. "Жаңбыр" программасын жазыңдар — экранда (кездейсоқ күйде) 50 нүкте жоғарыдан төменге қарай қозғалып келе жатқандай болуы тиіс. Нүктелердің бірі экранның төменгі шекарасына жеткенде, ол қайтадан экраннның жоғарғы бөлігінде пайда болуы керек.
3. "Жаңбыр" программасын кейбір нүктелер шапшаң, кейбіреулері баяу қозғалатындай етіп өзгертіңдер.
4. "Жаңбыр" программасын нүктелер өз қозғалысын төменгі сол жақ бұрыштан бастап, жоғарғы оң жақ бұрышта аяқталатындай етіп өзгертіңдер.
5. Қағазға оралған "кәмпиттің" бейнесін: диагональдары жүргізілген горизонталь тіктөртбұрыш пен оның екі жағына жалғасып жатқан тең қабырғалы үшбұрыштар салыңдар. Тіктөртбұрыштың сол жақ төменгі төбесінің координатасы (x, y), биіктігі a-ға, ұзындығы $2a$-ға тең болсын. Үшбұрыштың биіктіктері $a/2$. Тікбұрыштың диагональдары мен қабырғаларынан пайда болған қарама-қарсы екі үшбұрышты бояңдар.
6. Үш тісті және жарты дөңгелек пішінді аркасы бар мұнара салыңдар. Мұнараның сол жақ төменгі төбесінің координатасы (x, y), табанының ұзындығы a. Қалған өлшемдерін a арқылы өрнектеп табыңдар. Мұнараны бояңдар.
7. Куб салыңдар. Кубтың алдыңғы сол жақ төменгі төбесінің координатасы (x, y), бүйір қыры a. Жоғарғы бүйір бетін бояңдар, оң жақ бүйір бетінде диагональ жүргізіңдер.
8. Алтыбұрышты тік пирамида салыңдар. Табанының алдыңғы сол жақ төбесінің координатасы (x, y), табанындағы қыры a. Қалған өлшемдерін a арқылы өрнектеп табыңдар. Пирамиданың екі бүйір жақтарын бояңдар.
9. Тіктөртбұрышты қиық пирамида салыңдар. Пирамида табанының төменгі сол жақ төбесінің координатасы (x,y), табанындағы қыры a, жоғарғы жақтың қыры — $a/2$. Қалған өлшемдерін a арқылы өрнектеп табыңдар. Пирамиданың оң жақ бүйір бетін бояңдар.
10. Қарайтылған "Е" әрпі тәрізді фигура салатын программа құрыңдар. Әріп құрайтын нүктелердің абсциссалары мен ординаталары жиым элементтері ретінде берілген. Сурет салу үшін циклге енгізілетін тек қана бір LINE операторын пайдаланыңдар.
11. Экранда боялған шырша салатын программа жазыңдар. Сурет элементтерін жиымда сақтап, өшіріп және содан кейін қайтадан экранға шығарыңдар.

12. Боялған 7 шыршаның суретін салатын программа жазыңдар. Шыршалар бір-біріне ұқсас, горизонталь бойында бір қалыпта орналасады, шыршалардың биіктігі солдан оңға қарай сызықтық түрде өсуі керек. Шыршаларды бойлап Қызыл Телпек оларды уақытша "үсіп қалмасын" деп бір затпен жауып шығатын болсын.
13. Графикалық режімде ESC пернесін басқанша монитор экранына кездейсоқ күйде бірсыпыра нүктелер шығаратын программа құрындар.
14. Графикалық режімде ESC пернесін басқанша монитор экранында (100, 100) — (300, 200) тіктөртбұрышының нүктелерін кездейсоқ күйде жоятын программа құрындар.
15. Графикалық режімде ESC пернесін басқанша монитор экранының центрі (200, 200) нүктесімен дәл келетін және радиусы 80 болатын дөңгелек нүктелерін кездейсоқ күйде жоятын программа құрындар.
16. Шахмат тақтасының бейнесін салыңдар.
17. Экранда қарайтылған **М** әрпі пішіндес фигура салатын және бағыттауыш тілсызық (→, ←, ↑, ↓) пернелер арқылы басқарылып, фигураның көлемін + және – пернелерінің көмегімен өзгертетін программа құрындар.
18. Циклдік операторды пайдаланып 10 басқышы бар саты салыңдар, оның жұп және тақ нөмірлі басқыштарының көлемі әртүрлі болатын программа құрындар.
19. Шеңберге іштей бесбұрышты жұлдыз салыңдар.
20. Қызыл экранға 0..9 аралығындағы 1000 кездейсоқ санды ақ түспен шығару қажет. Мұнан соң экран түсін жасылға бояп, сандарды сары түспен шығарындардар.
21. Экранды барлық (8) фон түстеріне 5 секунд кідіріспен бояп шығыңыздар. Экраннын сол жақ жоғарғы бұрышына оның нөмірін жазып кою керек.
22. Қара экранға 1-ден 16-ға дейінгі сандарды 16 түрлі түске бояп шығару қажет. Әрбір сан бөлек жолда орналасатын болсын.
23. Алдын ала тазартылған экранды ақ түске бояп, жеті атаңыздын аттарын жеті түрлі түспен шығарындардар.
24. Қара экранға өзгеріп отыратын кездейсоқ түстер аркылы 200 "+" белгісін кездейсоқ түрде берілетін координаталық нүктелерге басып шығару керек.
25. Экраннын жоғарғы жағына екі терезе салып, оның біріншісінің ішіне өз атыңызды, екіншісіне – фамилияңызды жазып қойыңыз. 10 секундтан кейін терезелер ішін тазартып, өз аты-жөніңіздің орнына досыңыздын аты мен фамилиясын орналастырыңыз.
26. Экраннын сол жақ шетіне көк түсті терезе салып, оны оң жакка карай қадамын бір символ етіп жылжытып отыратын программа құру керек.
27. Экранға біртіндеп үлкейе бастайтын шағын кызыл терезе салу қажет. Үлкею қадамын *X* координатасы үшін 3, ал *У* координатасы үшін 1 етіп алыңыздар.

11. ПРОГРАММАЛАУ ТІЛДЕРІНІҢ ҚЫСҚАША ДАМУ ТАРИХЫ

Компьютердің машиналық тілінде жұмыс істеу программалаушының өте жоғары деңгейін талап етеді. Есептеу машиналарының бірте-бірте көбеюіне байланысты онымен қарым-қатынас жасауды онайлату қажеттілігі пайда болды. Біріншіден істелген іс – компьютер жадын бөлу жұмысы автоматтандырылды және машиналық командалар үшін адамға түсінікті белгілеулер енгізілді. Осылай пайда болған компьютермен қатынасу тілі **Ассемблер тілі** деп аталды. Кейіннен көптеген тілдер дүниеге келіп, олар уақыт талабына сай мүмкіндігінше жетілдіріліп отырылды.

Программалау тілдерін жасау және дамыту жұмысының басында АҚШ-тағы атақты IBM фирмасы тұрды. Бұл фирма кәсіби программалаушылардан басқа да мамандарға алгебралық формулалармен жұмыс істеуге дағдыланған ғылыми қызметкерлер мен инженерлерге жеңіл қолдануға болатын компьютерлер шығаруға тырысты.

1953 жылы Джон Бэкус компьютерлердің IBM-704 атты нұсқасына арнап программалауды оңайлататын жаңа тіл ойлап шығаруды ұсынды. Осыған орай жаңа тілдің сөйлемдерін (операторларын) машиналық кодқа түрлендіретін программаны – трансляторды да жасап шығару керек болды.

Алдыңғы кезде мамандар Бэкустың жұмыс тобына үлкен сенімсіздікпен қараған болатын, өйткені машиналық кодты жетілдіру істерінің алғашқы ұсыныстары айналасында, «адам тілінде сөйлейтін» компьютерлер болады деген әңгіме-сөз көп болған еді.

Әйтсе де, FORTRAN *(FORmula TRANslator* -- формулаларды түрлендіргіш) деп аталған тіл 1957 жылдың сәуірінде дайын болды және ол машина тіліне тек «формулаларды аударып» қана қоймай, циклдерді ұйымдастыру істерін де автоматтандыра білді.

Бұл жүйенің табысы күткен мүмкіндіктен бірнеше есе асып түсті, 1958 жылдың өзінде-ақ IBM-704 компьютерлеріндегі машиналық командалардың жартысынан астамы қолмен емес, осы Фортран тілінің трансляторы арқылы алынды.

Негізінде, Фортран жаңа программалау құралы емес, бұрынгы нақты машинаға көмекші ретінде, оны сату ісін арттыратын мүмкіндіктің бірі ретінде ғана жасалған жұмыс болатын.

Бірсыпыра кемшіліктерінің болуына қарамастан, мысалы, кейбір программалау идеологтары Фортранды «зердені тұйыққа тірейтін, синтаксис ережелерімен біріккен бұтақшалардың тізбегі, оны оқып үйренуге тиым салу» керек деп шулағанымен, бұл тіл көпшілікке кең таралып кетті және оны басқа машиналарға да икемдеп енгізу ісі жүргізіле бастады.

Икемдеу ісі (адаптация), Фортран тілінен басқа машина тілдеріне транслятор программалар жасау арқылы жүргізілді, әрине, ол оңай іс емес болатын.

Дегенмен, программалау тарихында тұңғыш рет программаны жақсы қойылған тапсырма бойынша (Фортран тілінің ережелерімен) жасау, «не жазарынды білмейтін» тілде жазғаннан гөрі әлдеқайда жеңіл, әрі арзан екенін көрсете білді. Басқаша айтқанда, ешқандай программалаусыз-ақ есепті ықшам

түрге келтіріп, оны түсінікті етіп қоя білу де үлкен өнер саналып, жоғары баға ала бастады.

Фортран тілі басқа да бір маңызды рөл атқарды: ол әртүрлі компьютерлер бір-бірімен ортақ тіл таба алатын тамаша құралға айналды.

Фортран бірнеше рет толықтырылып, өткен ғасырдың сексенінші жылдарына дейін кең пайдаланылды. Сонан кейін оны жаңа тілдер ығыстыра бастады. Соңғы кездерге дейін ол АҚШ-тың Аэронавтика мен ғарыштық зерттеулер жүргізетін ұлттық басқармасының (NASA) компьютерлерге арналған программалар жасайтын негізгі тілі болды. Оның әртүрлі нұсқалары осы кездерде де қуатты компьютерлер үшін есептеу программаларын жасау мақсатында кең қолданылып келеді.

Фортраннның осындай кенеттен ашылған қасиеттері, оның әртүрлі компьютерлер үшін әмбебап тіл бола алатын мүмкіндіктері программалау идеологтарын қарапайым ойын программаларын жазудан босатып, оларды өмірдің кез келген кезеңінде пайдаға асатын «әмбебап, жақсы, қолайлы тамаша» тіл жасау ісіне бет бұрғызды. Оның үстіне трансляторларды жазатын олар емес, басқалар екені етенеден белгілі еді.

Программа жасаушылардың «Ең жақсы – жақсының жауы» деген ұранына қарамастан, Алгол *(ALGOrithmic Language)* деп аталған жаңа тілді жасау ісі екі жылдан аса уақыт алды және ол негізінен Европада ғана кең қолданыста болды. Бұл, сірә, Европада да программалау тілдерінің мамандары бар дегенді дәлелдеу үшін жасалған сияқты болды.

Алгол туралы белгілі информатик мамандардың бірі Грейс Хоппер былай деген болатын: «Ол үлкен поэмаға ұқсас: математика тұрғысынан қарапайым, әрі түсінікті, бірақ практикалық тұрғыдан олай дей алмаймыз».

Осылай программалау саласындағы европалық аналитиктер тілдің әмебаптығына көңіл бөлсе, біртіндеп компьютер шығарушылардың әлемдегі көш басшысына айналған IBM фирмасы одан да мықты тіл шығаруға бет алды.

IBM фирмасы 1964 жылы PL/1 *(Programming Language One* — Нөмірі Бірінші Программалау Тілі) тілін дүниеге келтірді. Сол кездегі АҚШ мамандары осы тілді – программалау тілдерін жасаушылар ойлап тапқан мүмкіндіктердің аскар шыңы болар деп үміттенді. Солай болды да. Бұл тілге көптеген жаңалықтар енгізілген еді.

Әйтсе де, супер әмбебап программалау тілдерін жасау идеясы кір жуғыш машинасы бар теледидар жасаған сияқты біраз жерде артық кетіп жатты, оның көптеген мүмкіндіктерін пайдалану оңай болмады.

Программалау тілдерінің ең жақсысын жасаймыз деп зерделерін жұқартып жатқан теоретиктер жұмысы алаңдағы жалғыз допты қуалаған футбол командасы тәрізді болды – солардың жасаған тілін ең жақсы, әрі әмбебап деп айтсын деуге ұмтылған жарысушылар күйінде жүрді.

Ал бұдан шығатын жол бұрыннан-ақ белгілі болатын. Балаларға арналған әңгімедегі «Хоттабыч шалдың» әрбір футболшыға бір-бір доп сыйлағаны тәрізді әрбір тілдің өз қолданылу аясы болуы керек еді.

Осындай «доптардың» алғашқысы қазіргі программалау ісінің әжесі – Грейс Мюррей Хоппер атымен тығыз байланыста болды.

Жеті жасында-ақ Грейс суперпрограммалаушы қасиеттерінен белгі берді: ол үйіндегі барлық оятқыш сағаттардың қалай істелгенін білмек болып бұзып, артынан жинай алмай қойыпты.

Екінші дүниежүзілік соғыс қызып тұрған кезде математика докторы Грейс Хоппер АҚШ-тың соғыс-теңіз флотына кіріп, 1944 жылғы маусымда офицер атағын алады. Оның әскерде адмирал шеніне дейін жеткенін айта отырып, программалау саласындағы табыстарына тоқталайық.

Хоппер сол кездердегі «Марк-1» машинасына арнап программа жазумен шұғылданған. Ол сегіздік санау жүйесінде жұмыс істеген.

Бірде өзінің банктегі есебін анықтағысы келіп (математика докторы үшін қиын мәселе), Хоппер өзінің банкте істейтін бауырынан көмек сұрайды. Банкир інісі Грейстің жазбаларын карап отырып, оның барлық есептеулерді сегіздік жүйеде орындағанын көріп таң қалады. Ал банк болса, бұрынғыша ондық жүйені пайдаланатын еді.

Осы кездерден бастап, Хоппер машинамен адам тіліне жақын тілмен қатынас құруға болатын мүмкіндіктерді зерттеумен айналыса бастайды. Оның еңбегінің нәтижесі ретінде А-1, А-2, А-3, В-0, т.б. трансляторларды келтіруге болады. Бірақ осы трансляторлардың барлығында да қолданылған тіл адам тілінен әлдекайда алшақ болатын.

Грейс Хоппердің мақсаты ағылшын тілінде жасалған программалау мүмкіндігін табу болатын.

Банк есептерімен болған жағдай оның программалаудағы жұмысын физиктерге арнауға немесе әмбебап программалау тілдерін жасауға емес, карапайым экономикалық есептерді шығаруды жеңілдетуге арнауға бет бұрғызды.

Осы жобаға байланысты болған жайттардың барлық бүге-шігесіне тоқтамайақ, тек Хоппердің соғыс-теңіз флотымен байланысы арқасында оның АҚШ әскери министрлігін қолданбалы іс жүйесінде пайдаланылатын тілдермен айналысуға көндіргенін айту жеткілікті. Ол кезде бұл министрліктің қол астында көбінесе экономикалық есептеулермен айналысатын мыңнан аса әртүрлі компьютерлер бар болатын.

Осының нәтижесінде 1959 жылы жаңа тіл жасау комитеті құрылды. 1960 жылдың сәуірінде сол комитет тілдің сипаттамаларын құрастырып, жариялаған болатын. Ал сол жылдың аяғында бірнеше фирма бұл тілдің трансляторларын да жасап үлгерді.

COBOL *(COmmon Business Oriented Language* — бизнеске арналған әмбебап тіл) тілі бірден Американың іскерлері ортасынан өз бағасын алды. Әрине, программалау ісінің мықтылары оны көп сөзді және өте көлемді деп санайды, бірақ онда ағылшын тілінің карапайым сөздері қолданылатындықтан, программа коды кез келген мамандар мен менеджерлерге түсінікті деңгейде болды.

Бүгіннің өзінде, шыққанына ширек ғасырдан асса да, ол әлі кең қолданыс табуда. Бұл тілде жазылған программалардың жалпы бағасы 50 млрд доллар шамасында деп саналады. Экономикалық есептер шығарудағы оның тиімділігі

әлі де жоғары деңгейде. Кобол тілінің негізінде қазіргі кездегі мәліметтер базасымен жұмыс істеуге арналған Clarion тілі жасалып шықты.

Егер алғашқы программалау мамандары тіл және оған транслятор жасау жұмыстарын бөле жарып қарамайтын болса, кейіннен теоретиктердің күш салуы арқасында бұл екі жұмыс бір-бірінен ажыратылды да, теоретиктерге де арнайы жұмыс табылды. Тілді біреулер жасап, трансляторларды басқалар жазатын болды.

Дегенмен компьютер тұтынушылары үшін программалардың бағалы қасиеті – олардың әмбебаптығы сақталғаны дұрыс болар еді. Мысалы, бір типтегі компьютерлерге арналып Коболда жазылған программалар басқа компьютерлерде де жақсы жұмыс істеуі тиіс.

Осының бәрі тілдердің сипаттамаларын стандарттау қажеттілігін туғызды, өйткені кейіннен оларға арнап басқа фирмалар трансляторлар жазуы керек болатын.

Бұрынғы бір бағыт орнына енді үш бағытта жұмыс істеуге тура келді:

1. Тілді жасау.
2. Тілдің стандартын анықтау.
3. Программалау тіліне транслятор жазу.

Оның үстіне ақпарат өңдеуге арналған адамзат қызметінің әр саласының өз программалау тілі болғаны дұрыс екенін өмір дәлелдеді (міне, әр ойыншыға бір-бір доп деген осы шығар), мысалы:

- қолданбалы іскерлік аймақтарға арналған тіл (COBOL);
- ғылыми-техникалықесептерге арналған тіл (FORTRAN);
- кестелерді өңдеуге арналған тіл (APL–*A Programming Language* – программалау тілі – жай ғана атау);
- металл өңдеуге арналған қондырғыларға (станоктарға) арналған программалау тілі (APT–*Automatically Programmed Tools* – автоматты түрде программаланатын аспаптар);
- мамандардың айтуы бойынша, ми жұмысын модельдейтін және жасанды зерде (интеллект) жасауға болатын тіл (IPL – *Information Processing Langauges* – ақпарат өңдеу тілдері);
- әмбебаптықтан да аздап үміті бар, нақты уақыт режимінде жұмыс істейтін объектілерді басқару тілі (ADA);
- компьютер жедел жадынан өте аз орын алып, барынша жылдам істейтін программалар алуға мүмкіндік беретін жүйелік программалаушыларға арналған «орта» деңгейдегі тіл (C/C++);
- нақты уақыт режимінде жұмыс істейтін «сындарлы» (критикалық) есептерге және борттағы компьютерлерге арналған тіл (FORTH);
- программалауды үйренуге арналған тіл (PASCAL);
- студенттерге арналған программалау тілі (LOGO);
- программалауды үйренуге қабілеті болмаса да, өте программалағысы келетіндерге арналған тіл (BASIC), тағы да сол сияқты жалғасып кете береді.

АҚШ әскери министрлігінде өткен ғасырдың 70 жылдары 450-дей жоғары деңгейдегі тілдер мен солардың диалектілері қолданыста болғанын айту жеткілікті шығар. Осы жайт әмбебаптықтан үміткер ADA тілінің шығуына себепкер болған.

Компьютерлердің қуаты артқанмен, олардың бағалары тұрақты түрде төмен түсіп жатқан заманда, программалаудың мықты теоретигі Эдсгер Дийкстра айтқандай, трансляторлар көптеген «ысқырықтар мен сылдырмақтармен» толықтырылып, жаңа мүмкіндіктерге ие болып жатыр. Сол сылдырмақтар мен ысқырықтар программалаушылардың жап-жақсы өмірін одан да көріктендіре түспек. Ол қандай болмақ – бұл жағынан пікір айтушылар аспан мен жердей алшақтықпен айтысып келе жатыр.

Программалау тілдері туралы ой өрбіткенде, құрылымдық программалау идеологиясы жайлы айтпауға болмайды, бұл идеологияны алғаш ұсынған да сол Дийкстра болатын. Осы құрылымдық программалау ұғымы оқу құралымыздың негізінде жатыр. Ал Паскаль тілінің конструкциясы Дийкстра ұсынған, қазіргі кездегі кең қолданыста жүрген барлық процедураға бағытталған құрылымдық тілдердің ядросы болып саналады. Жақсы ұйымдастырылған құрылымдық тілдердің басты белгісі – онда GOTO операторының, яғни шартсыз басқаруды беру операторының болмауы немесе оның сирек колданылуы болып есептеледі.

Дийкстра программаларда нақты математикалық логикалық құрылымның жоқ екеніне көңіл бөліп, зерделі мамандардың жиі қайталайтын пікіріне көбірек назар аудару қажеттілігін басып айтқан болатын. Оның идеяларын жүзеге асыра отырып, IBM фирмасы «Нью-Йорк таймс» газетінің мәліметтер банкін жасау кезінде жұмыс бағасының әжептеуір арзандағанын және оны орындау мерзімінің де біраз қысқарғанын анықтады.

Бір қарағанға өте қарапайым көрінетін Дийкстраның осы идеясының өзі көптеген елдерде миллиардтаған долларды үнемдеуге көмектесті. Қазіргі кезде жоғарыдан төмен қарай жүргізілетін құрылымдық бағыттағы жобалауды жүзеге асыра алмайтын бірде-бір программалау тілі қолданысқа кіріп, өмірден өз орнын таба алмайды.

Сонымен, жүйелік программалау маманы Харлан Миллс айтқан: «Ғарыш кемелерінің ұшуының өзі де құрылымдық программалауды пайдаланбай жүзеге асқан деп ойламаймын», – деген сөзінде шындық бар.

Паскаль тілі де құрылымдық программалау тәсілдерін үйретететін оқыту құралы ғана болуға тиіс еді. Бірақ бірсыпыра себептерге байланысты ол бұл аядан шығып, әлемдегі ең негізгі тілдердің біріне айналды.

Құрылымдық программалау тілдері табиғи тілдің сөздерін және грамматикасын кеңінен пайдалану жолымен дамыды, бірақ оның негізінің түп тамыры болатын программалаудың процедуралық стиліне тиіскен жоқ. Осы *процедуралық стиль* сөзі компьютердің командаларды атқарушы рөлінде болатынын және оған нәтиже алуға қажетті барлық әрекеттер (процедуралар) тізбегін толық әрі дәлме-дәл сипаттап беру керек екендігін білдіреді. Оған қоса, компьютердің фон-Нейман архитектурасының өзі де оларды қуатты, бірақ ақылсыздау робот ретінде қарастыруды ұсынады.

1972 жылы шотланд ғалымы Р.Ковальский осындай мүмкіндігі бар тіл ретінде тұжырымдамалар және олармен орындалатын операцияларды өңдей алатын математикалық логика тілін жетілдіріп пайдалануды ұсынды. Бұл идеяны Франция, Португалия, Ресей, т.б. елдердің ғалымдары жалпы түрде қарастырып, практикалық түрде жүзеге асырмақшы болды.

Осындай мақсатта жасалған программалау тілі PROLOG (французша – *PROgramming et LOGicque)* деп аталды. Оны жасауға және дамытуға жаңа ғылыми бағыттың нақты түрдегі қорытындысы (прологы) ретінде пайда болған **логикалық программалау** ғылымы үлес қосып келеді.

Логикалық программалау компьютер есепті адам сияқты жолмен шығаруы керек деген ойды негізге алады. Логикалық программалаумен қатар пайда болып, функцияның математикалық ұғымдарына негізделген екінші бір сала **функционалдық программалау** деп аталады.

Функционалдық программалаудың алғашқы тілі 1960 жылдардың басында шыккан LISP тілі болып табылады. LISP сөзі – ағылшынның *LISt Processing* деген сөз тіркесінің қыскартылған нұсқасы. Бұл да кең таралған тілдердің бірі. Ол жасанды зерде мәселелерін шешу жолында, әсіресе АҚШ-та көп қолданылады.

Қазіргі кезде компьютерлердің көбісіне арналған LISP тілінің бірнеше нұсқалары бар. Бірақ бұл тіл функционалдық программалаудың жалғыз ғана құралы емес. Оған ұқсас тілдерге Miranda, Hope, FP тілдері жатады.

Өткен ғасырдың 80-жылдары программалауда тағы да бір жаңа ағым пайда болды. Енді компьютерге бастапқы мәліметтер мен нәтиже арасында байланыс орнату ісі жүктелді. Біз компьютерге ол жұмыс істеуге тиіс объектілер сипаттамасын енгіземіз, ал компьютердің өзі сол объектілердің қалай байланысқанын анықтауы керек болды.

Объектілер қасиеттерін сипаттаудың осы түрі **объектіге бағытталған программалау** деп аталды. Объектіге бағытталған программалау тілдеріне C++, C#, Java, PHP және Smalltalk тілдері жатады.

Бұлардың бәрі бүгінгі күн тұрғысынан туындаған көзқарастар болатын. Ал тарихи тұрғыдан объектіге бағытталған программалау компьютердің мүмкіндіктерінің жетілдірілуі мен ол шығара алатын есептер аймағының кеңуі арқасында пайда болды.

Алғашқы программалау тілдері тек бір санды ғана өңдей алатын еді. Ал сан жиымдары цикл арқылы өңделетін.

Кейіннен машина бір оператор арқылы жүздеген және мыңдаған сандарды қоса алатын, жиым ішінен максимумын табатын, оларды реттейтін, т.с.с. мүмкіндіктерге ие болды Сандар жиымы, программалаушы тұрғысынан қарағанда, әртүрлі амалдар орындауға болатын бір ғана объект түрінде карастырылатын кезге жеттік.

Бірақ нақты өмірде біз тек сандық ақпаратпен ғана жұмыс істемейміз. Мысалы, студентті тек оның жасы емес, аты, фамилиясы, тобы, үй адресі, т.б. сипаттайды. Әртүрлі типтегі мәліметтерді біріктіріп, **мәліметтер құрылымын** алуға мүмкіндік (қажеттілік) туды. Осындай әрбір құрылымды *біртұтас күйде* өңдеуге болатын болды.

Осындай құрылымдар олармен орындауға болатын әрекеттермен бірге сипаттауға арналған жасанды тіл объектілері мен олардың қасиеттері атауларынан тұрады. Мұндай қасиеттер компьютерге осы объектілерді қалай түрлендіруге болатындығы жайлы сигнал береді. Бұл тілде программалау

объектілерді сипаттау және бастапқы мәліметтер мен нәтижелерді осындай объектілерге құрылымдау ісіне келіп тіреледі.

Құрылымдар мен объектілердің күрделілігін, практикалық тұрғыдан алғанда, шектеусіз арттыру ісі (мысалы, құрылымдар объектісі болатын жаңа объект жасау ісі) ақпараттың өте үлкен көлемімен жұмыс істеуге мүмкіндік береді. Сол себепті компьютерлер де күннен-күнге күрделіленіп келе жатқан есептерді шығаруға пайдаланылып келеді.

Бақылау сұрақтары

1. *Программалауды оңайлататын жаңа тіл ойлап шығаруды алғаш рет кім ұсынды?*
2. *Фортран тілі қай жылы шықты және оның ерекшелігі қандай болды? Алгол, PL/1, Кобол тілдері ше?*
3. *Программалау тілдерінжасауды стандарттау қажеттілігінеге байланысты туындады?*
4. *Грейс Хоппердің программалау саласындағы еңбектері.*
5. *Программалау тілдерінің саны неге көп болып кетті?*
6. *Құрылымдық программалау тілдерінің ерекшелігі неде?*
7. *Программалаутеоретигі Эдсгер Дийкстра қандай бағытты ұстану қажеттілігі туралы айтты?*
8. *Объектіге бағытталған программалау тілдері қандай болады?*

Зертханалық жұмысқа арналған тапсырмалар

Берілетін мәліметтер алдын ала файлға жазылып қойылуы тиіс. Алынған нәтижелерді файлға жазылса, соларды оқу арқылы тексеріп тапсырманың дұрыстығын тексеріп шығу керек.

12 С ПРОГРАММАСЫН ОРЫНДАУ ОРТАСЫ

12.1. Турбо С редакторының терезесі

Turbo C программалау жүйесі C/C++ тілінде программа құрастырып, оны орындауға мүмкіндіктер береді. Жалпы, тұтынушы мынадай әрекеттерді орындай білуі керек:

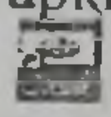
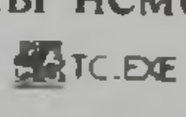
- программаның мәтінін жазып, дискіде программа файлы ретінде сақтау;
- программаны ***компиляциядан*** өткізіп, егер синтаксистік қателері бар болса, оларды түзету;
- программаны орындап, нәтижесін алу.

Сонымен, кез келген программа оны теру, компиляциялау, құрастыру, атқарылушы модульді жасау және орындап нәтиже алу сатыларынан өтуі тиіс.

С тілінің біріктірілген (интегралданған) ортасында программа орындау төмендегідей қадамдардан тұрады:

1) компилятор кажет файлдарды іздеп тауып алуы үшін ортаның параметрлерін тағайындау;
2) программалық файлды редактор ортасына жүктеу немесе теру;
3) атқарылатын модульді (орындалатын файлды) жасау;
4) программаны іске косу және орындау;
5) программада қате болса, оны жөндеп түзету.

Программа құру үшін, ен алдымен, Турбо С біріктірілген ортасымен жұмыс істей білу керек.

Турбо С ортасы – арнайы көп терезелі мәтіндік редактор. Ол көбінесе **C:\TC\BIN\tc.exe** файлын іске қосу арқылы немесе сонын жарлыктарынын бірін іске қосу жолымен жүктеледі:  

TC.EXE файлының жұмыс істеу барысында экранда Турбо С ортасының негізгі терезесі ашылады (12.1-сурет).

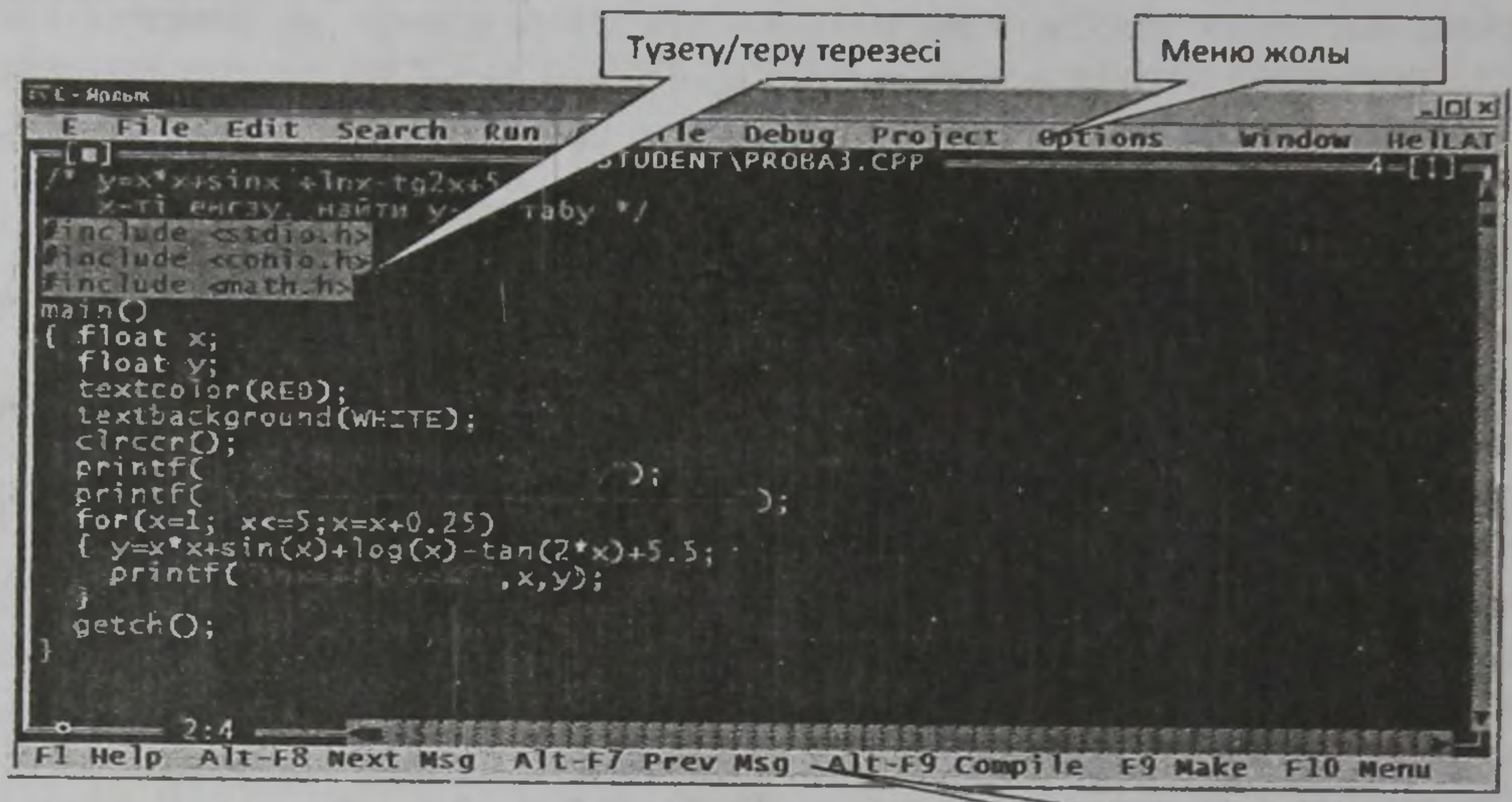
Турбо С ортасын автоматты түрде жүктеп, орыс (қазақ) әріптерін теру мүмкіндігін беретін C.bat пакеттік файлының мәтінін мынадай түрде құрастыруға болады:

```
C:
cd TC\BIN
rk.com
tc.exe
```

C.bat файлын немесе оның жарлығын шерту арқылы да Турбо С ортасы ашылады. Ол іске қосылған соң, орыс әріптеріне және ағылшын әріптеріне ауысу кос Shift пернелерін (ол басқаша да болуы мүмкін) қатар басу арқылы орындалады. Турбо С редакторы DOS ортасында жұмыс істейтін көкшіл экранға шығады. Ол экран төрт бөліктен тұрады (12.1-сурет):

- меню жолы,
- түзету/теру терезесі,
- мәлімдеме хабарлар терезесі,
- қалып қатары.

Терезені үлкейту үшін **Alt+Enter** пернелерін басу керек, редактор терезесін экранды толық алатындай етіп үлкейту үшін, оның оң жақ жоғарғы бұрышындағы тілсызық [↑] батырманы шерту керек. Ал терезені жабу үшін – сол жақ жоғарғы шеттегі төртбұрышты батырманы [■] шертеміз. Турбо С-ден шығу үшін *Alt* және *X* (латын) пернелерін қатар басу керек.



12.1-сурет. Турбо С-дің біріктірілген ортасы

Терезе нөмірі оның он жақ жоғары бұрышында орналасады. Керекті терезеге (1 – 9) көшу үшін:

- Alt+0 пернелерін басқанда шығатын терезелер тізімінен керектісін таңдау арқылы;
- Alt+5 деп терезе нөмірін 5-ті енгізіп көшу;
- F6 пернесі арқылы терезелердің бірінен біріне көшуге болады.

F5 пернесі арқылы терезені үлкейтуге немесе аздап кішірейтуге болады.

Меню жолында 11 команда бар, олар әртүрлі қызметтер атқарады, әр менюді таңдағанда, оның ішкі командалары ашылады. Солардың кез келгенін таңдап орындай аламыз.

12.2 Меню командалары

Программа іске қосылып, терезе ашылғаннан кейін, курсор жұмыс алабында тұрады.Меню қатарына F10 пернесі арқылы шығып, ESC арқылы жұмыс алабына ораламыз. Меню қатарының командаларын және төменгі сатылы командаларының қажеттісін ← ↑ → ↓ бағыттауыштары арқылы таңдай аламыз. Команданы орындау үшін **Enter** пернесін басамыз.

Осы әрекеттерді тышқан қолтетігімен де қалыптағыдай етіп орындауға болады.

Командалар тізімі

- **Ё**– қосымша әрекеттер орындау.
- **File** (файл) – файлдармен жұмыс істеу.

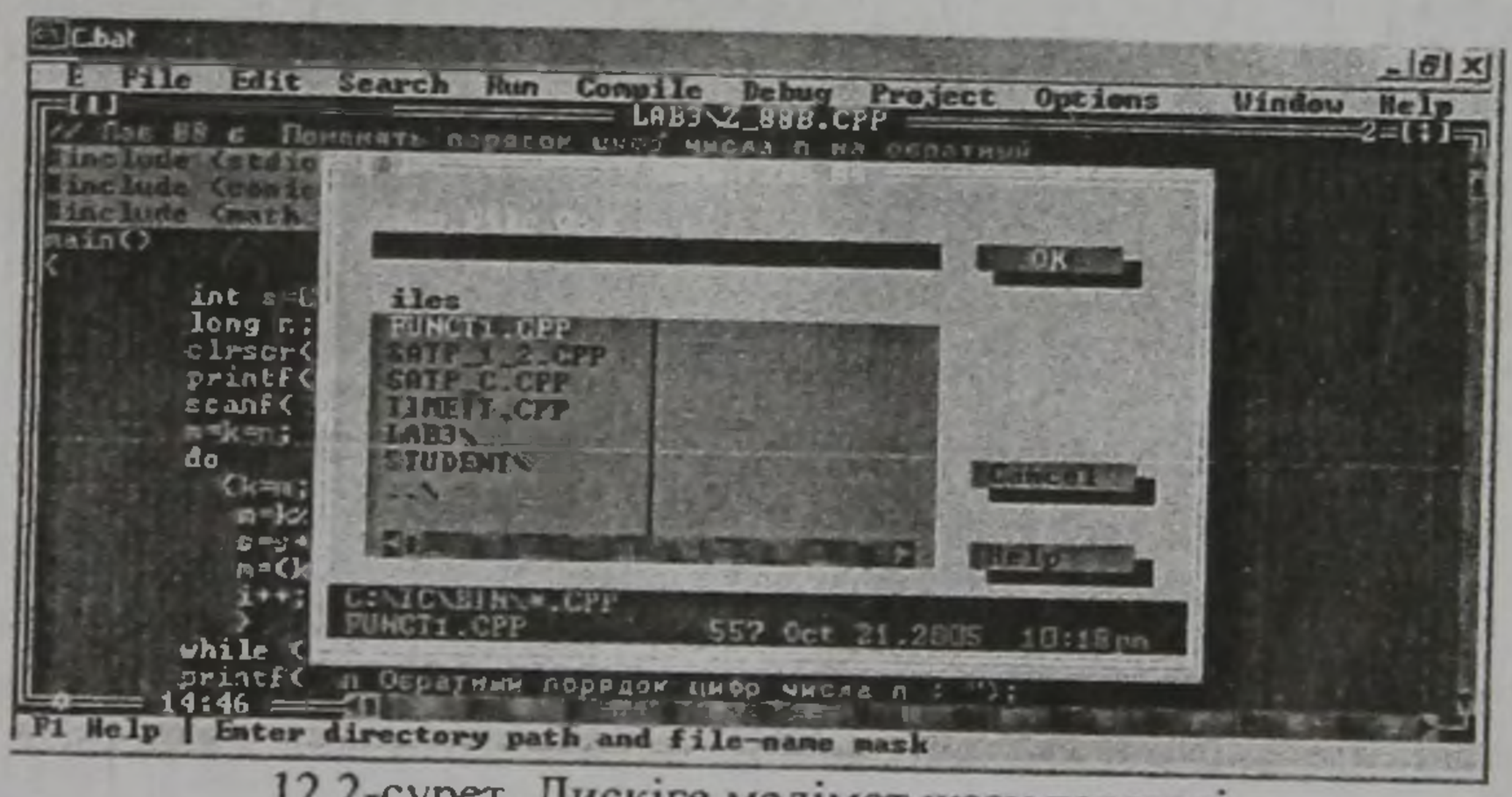
•**Edit** (түзету) – ашық терезедегі мәтінді түзету режимдерін орындау.
•**Search** (іздеу) – іздеп табу әрекеттерін орындау.
•**Run** (атқару) – программаны орындау.
•**Compile** (компиляция) – программаны компиляциядан өткізу.
•**Debug** (отладка) – программаны жөндеу.
•**Options** (нұсқалар) – орта параметрлерін тағайындау.
•**Windows** (терезе) – тереземен жұмыс істеу.
• **Help** (көмек) – анықтамалық жүйеден көмек алу.

Қосымша әрекеттер жұмыс істеп тұрған файлдарды анықтау, ішкі Ассемблерді іске қосу сияқты әрекеттерді орындайды.

File менюінің ішкі командалары (12.2-сурет):

Open F3 – бұрын жазылған файлды ашу {D:\LB1.C}
New – жаңа файл ашу
Save F2 – файлды дискіге жазып сақтау
Save as ... – файлды басқаша түрде жазып сақтау
Save all – барлығын да жазып сақтау
Change dir– директорийды өзгерту
Print – ашық терезедегі файлды қағазға басу
Dos shell – DOS-қа (Turbo C-ден) уақытша шығу
Exit Alt-X – Turbo C-ден шығып кету

Файлды дискіге өзіңіз қойған атпен сақтауға арналған *Басқаша сақтау* (Save as – Сохранить как...) терезесі төмендегі 12.2-суретте көрсетілген.



12.2-сурет. Дискіге мәлімет жазу терезесі

Edit менюінің ішкі командалары (12.3-сурет):

Undo (болдырмау) Alt + Bksp – соңғы орындалған команданың әрекетін болдырмай алып тастайды.

Redo Shift + Alt + Bksp – Undo командасының кері қайтарған командасы әрекетін қайтадан іске қосады.

Cut (қиып алу) Shift+Del – белгіленген бөлікті буферге қиып алады (бұрынғы орнында қалмайды).

Copy (көшіру) Ctrl+Ins – белгіленген бөлікті буферге көшіреді (бұрынгы орнында сақталады).

Paste (кірістіру) Shift+Ins – курсор орналасқан жерге буфердегі ақпаратты кірістіріп қояды, яғни енгізеді.

Clear (өшіру) Ctrl+Del – белгіленген бөлікті өшіру.

Copy Examples – мысалды көшіру.

Show Clipboard (буферді ашу) – редактор терезесінен буферге алынған мәтінді сақтайтын терезені ашады.

Search менюінің ішкі командалары:

Find (табу) – табуға қажетті сөзді енгізу мүмкіндігін беретін сұхбат терезені ашады.

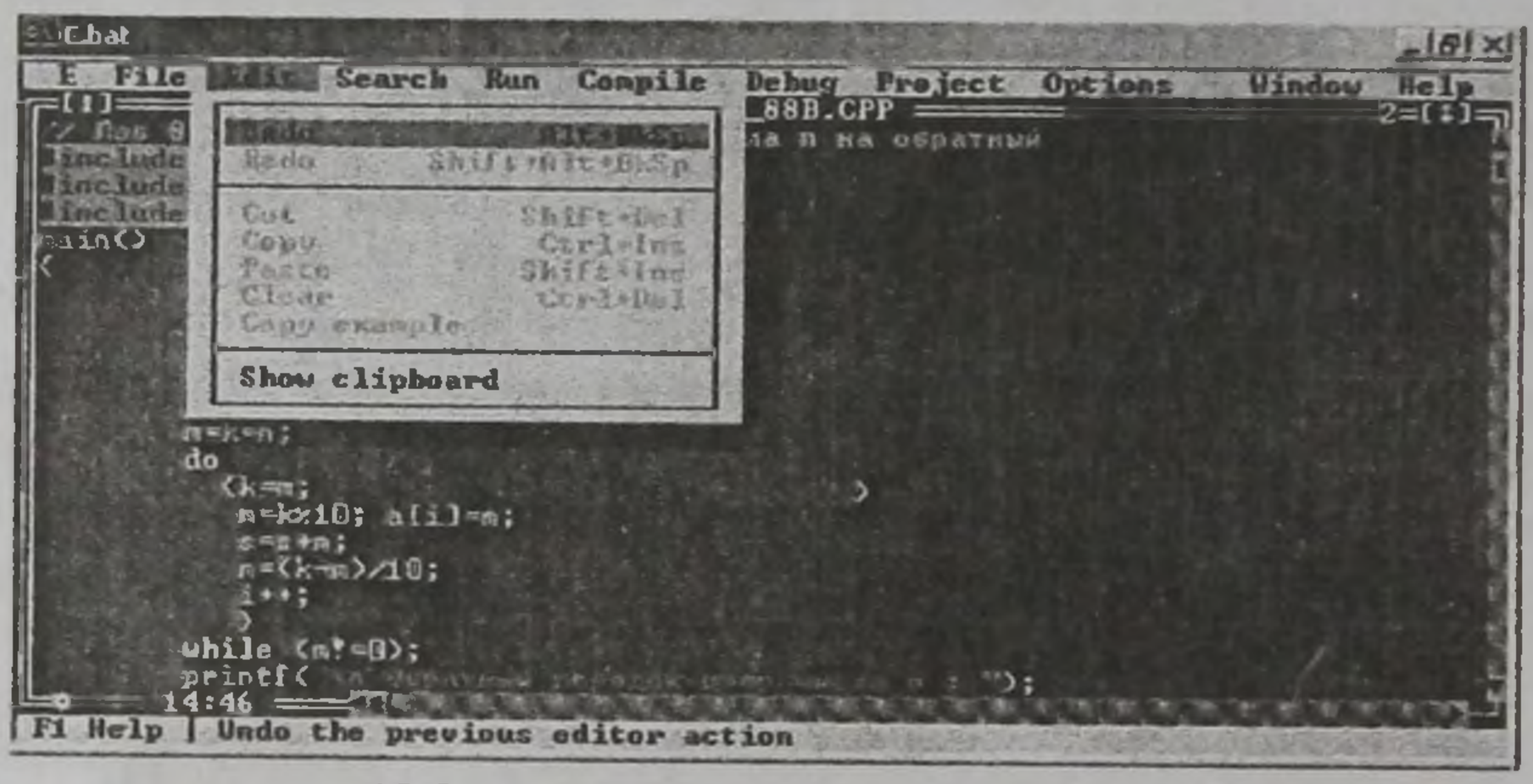
Replace (орын алмастыру) Alt+S+R – іздейтін мәтін мен оны алмастыратын мәтінді енгізу мүмкіндігін беретін сұхбат терезені ашады.

Search Again (қайтадан іздеу) Ctrl+L – **Find** немесе **Replace** командаларының соңғы әрекетін қайталайды.

Go to line number (нөмір қатарына бару) – курсорды нөмірі көрсетілген қатарға орналастырады.

Previous error Alt+F7 – алдыңғы қате орнына бару.

Next error Alt+F8 – келесі қате орнына бару.



12.3-сурет. Edit менюінің ішкі командалары

Run менюінің ішкі командалары:

Run(орындау) Ctrl+F9 – орнатылған параметрлерді қолдана отырып, редактор терезесіндегі екпінді программаны орындайды.

Program reset Ctrl+F2(сброс программы) – жөндеушінің орындап жаткан әрекетін тоқтатып, программаға бөлінген орынды босатып, барлық ашық файлдарды жабады.

Go to cursor(курсорға өту) F4 – екпінді терезедегі программаны курсор тұрған орындағы қатарға дейін орындайды.

Trace into (қосалқы программаға кіріп, қадамдық орындау) F7 – программадағы операторларды қадам бойынша көрсете отырып орындайды.

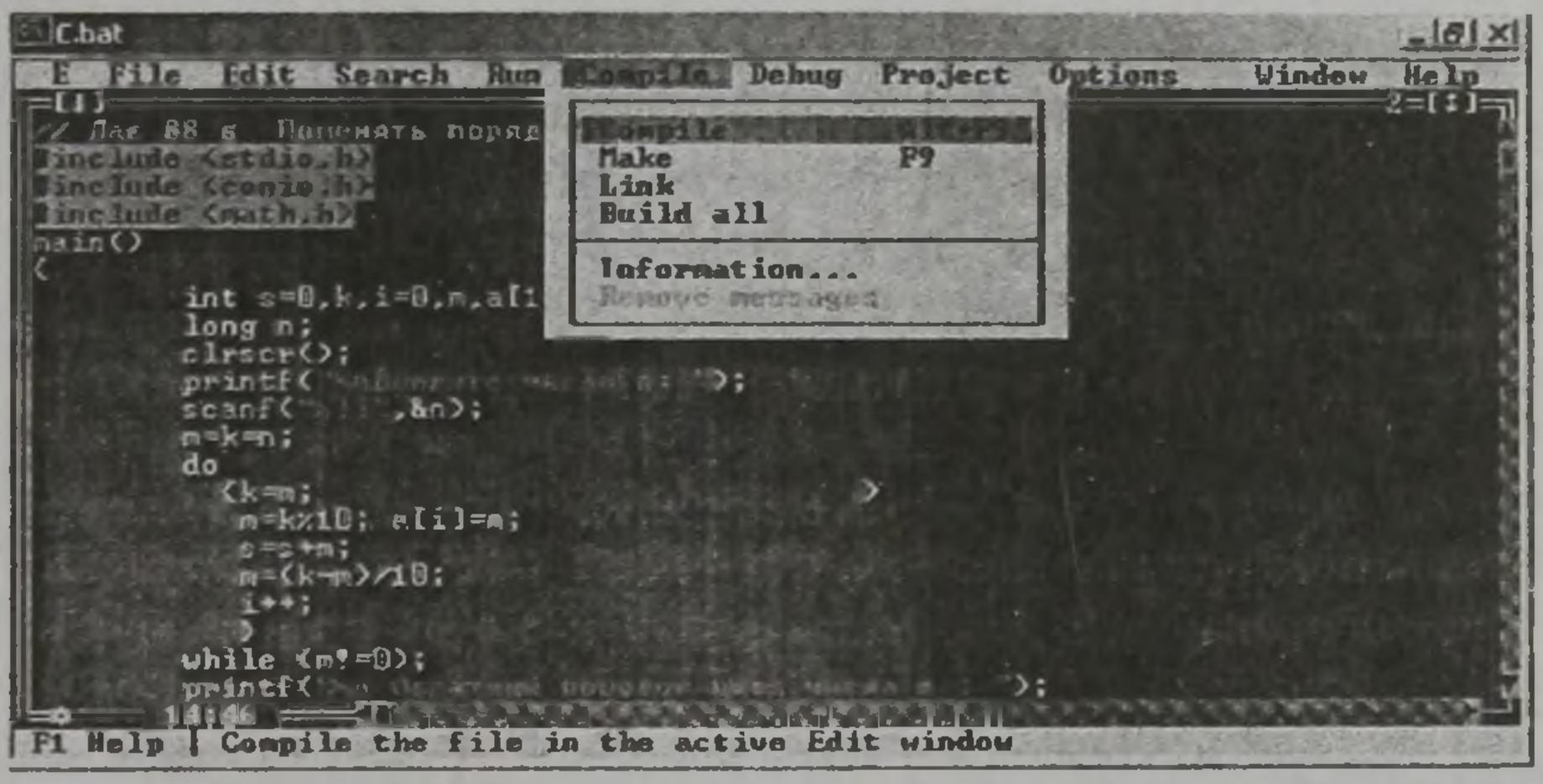
Step over (программаны қадам бойынша орындау) – программа мәтінінің бір жолына сәйкес келетін кезекті операторды біртіндеп орындайды.

Compile менюінің ішкі командалары (12.4-сурет):

Compile Alt+F9 – екпінді терезедегі программаның қатесін тексереді. Синтаксистік қате жөнінде хабарлама береді де, курсор қате жіберілген орынға орналасады. Қате жоқ болса, компилляцияның сәтті болғаны жөнінде хабарлама береді.

Make (программаны жинақтау). Егер негізгі программада немесе негізгі модульде қолданылатын жеке модульдердің мәтінінде объектілік файлды алғаннан кейін өзгеріс болса, соған сәйкес модульдер қайта тексеріледі, одан кейін негізгі программа немесе негізгі модульден тұратын файл қайта құрылады.

Build all (Программа құру) – бұл команда орындалғанда, негізгі программа және негізгі модульде қолданылатын барлық модульдер қайта компиляциядан өткізіледі.



12.4-сурет. Compile менюінің ішкі командалары

Debug (жөндеу) менюі ішкі командалары:

Inspect... (Alt+F4) – Inspector терезесін ашады, ол объектілер мәнін талдауға көмектеседі.

Evaluate/modify... (Ctrl+F4) үш өрісі бар терезе ашады: Expression, Result, New value — олар айнымалы мәндерін көріп, оларды өзгерту мүмкіндігін береді.

Call stack (Ctrl+F3) – программада қолданылған функциялар тізбегін – стекті көрсететін қосалқы программа терезесін көрсетеді.

Watches 8 суырылып шығатын менюді ашып, жаңа өрнектер енгізіп, оның нәтижесін көрсете алады.

Breakpoints... – түзету режимінде тоқтау нүктесімен жұмыс істеу мүмкіндігін беретін терезе ашады.

Project командалары қажет болғанда, жобалар ашу, толықтыру және жабу ісін атқарады.

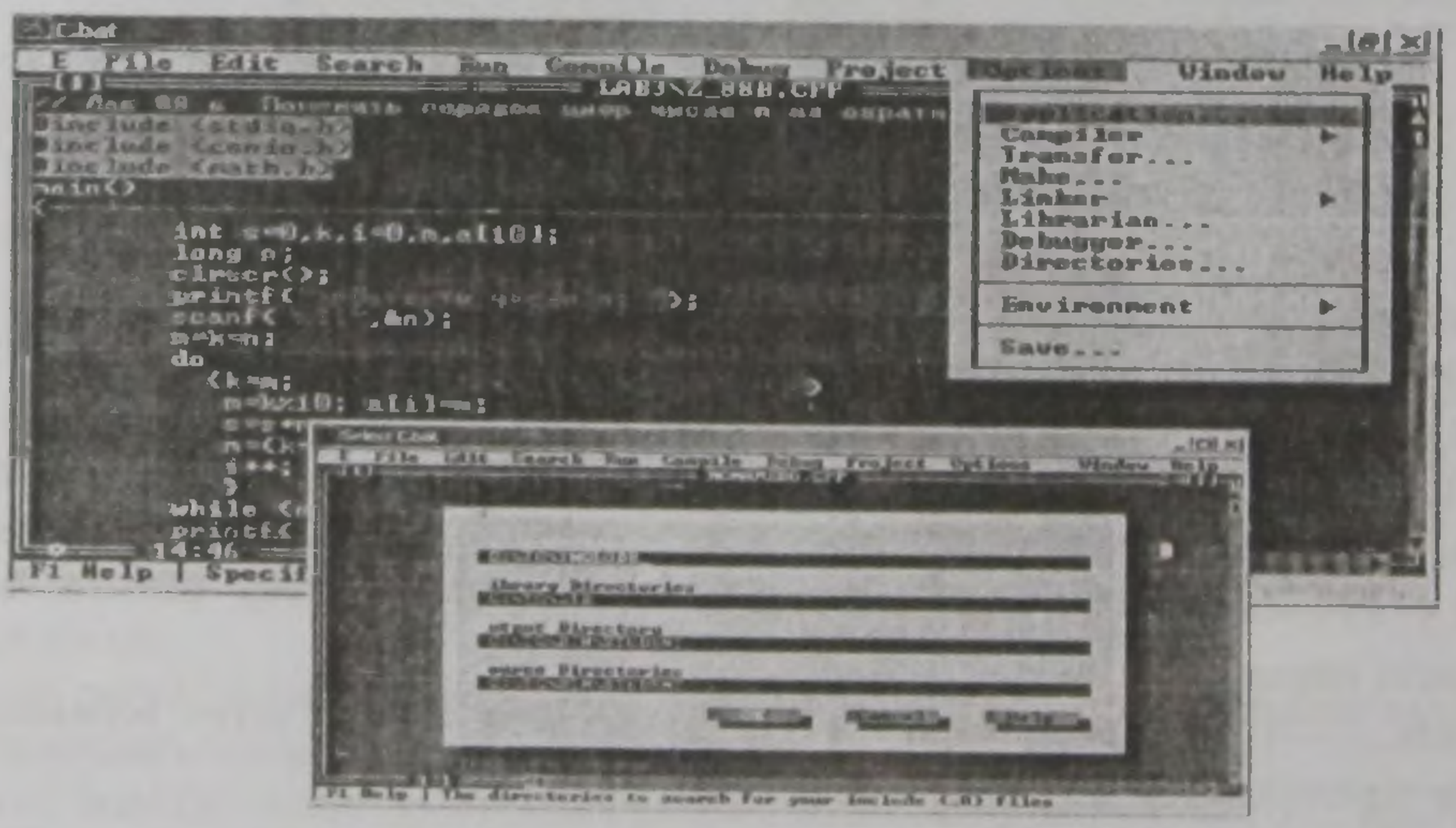
Жоба – бір-бірімен байланысты файлдар жиыны, бірнеше объектілік файлдар бірден компиляциядан өткізіліп, атқарылатын бір программа жасайды. Жоба көп

файлды программалар кезінде керек. Кейде бір файлмен жұмыс істегенде де қолданылады.

Модульдік программалау барысында көпфайлдық компиляциялаусыз жұмыс істеу мүмкін емес. Көлемі үлкен программалармен жұмыс істеу кезінде ол программаның бөліктерін бірнеше файлдарға сақтау анағұрлым ыңғайлы. Әрбір файл бүтіндей бір немесе бірнеше функцияны қосуға тиіс. Ол файлдардың аттары арнайы файл-жобаға жазылады, IDE ол ақпараттан мәтіндік файлдардың қайсысын орындалатын файлға (.EXE) біріктіру керек екендігінен хабардар болады.

Файл-жобалармен жұмыс істеуге қажетті бұйрықтардың барлығы *Project* мәзіріне қосылған.

Файл-жобаларды ұйымдастыру үшін ол файл-жобаны ашу керек. Ол үшін **Project\Open Project**... командасы орындалады және IDE экранның төменгі жағындағы арнайы *Project* терезесін іске қосады. Қажетті файл-жобаны жүктейтін немесе берілген атпен жаңа файл-жоба құратын диалог терезесі ашылады. Жаңа файл-жоба құрылған болса, *Project* терезесі бастапқыда бос болады. Жобаға файлдарды қосу немесе оларды жою **Project\Add item**... және **Project\Delete item** бұйрықтарымен орындалады. Меңзер (курсор) *Project* терезесінде орналасқан жағдайда осы мақсатта Ins және Del батырмаларын басса да жеткілікті. Файлдарды жобаға қосу кезінде қажетті файлды таңдауға мүмкіндік беретін диалог терезесі ашылады. *Project* терезесі жобаға қосылған бір файлдан оларды редакциялау барысында келесісіне көшуді жеңілдетеді. Ол үшін *Project* терезесіндегі қажетті жолға меңзерді (курсорды) апарып ENTER пернесін басса жеткілікті. Borland C++ ортасында жұмыс істеу кезінде программа бір ғана файлдан тұрғанның өзінде жобаны қолданған ыңғайлы.



12.5-сурет. Option менюінің ішкі командалары

Options менюінің командалары (12.5-сурет) Турбо С ортасының келісім бойынша тағайындалатын параметрлерін көру және оларды өзгерту мүмкіндігін

береді. Олардың көптеген мәндерін өзгертпей, қалдыруға болады. Мұнда түйінді сөздер түсін (16 түс) өзгерту мүмкіндігі бар (Options/Enviroment/Colors/Syntax).

Directories командасы тақырыптық файлдар каталогын (*Include Directories*), кітапханалық функциялар (*Library Directories*) каталогын және терілген файлдар мен олардың нәтижелік файлдары (*Output Directory*) қайда орналасатынын көрсетіледі.

Мысалы, егер TC программалары C:\TC каталогында орналасса, онда **Include Directories** өрісінде **C:\TC\INCLUDE** деп көрсеткен дұрыс болады, ал Library Directories өрісіне — **C:\TC\LIB** деп жазу керек. **Output Directory** жолына нәтижелік файлдар орналасатын каталогты, мысалы, **C:\TC\BIN\STUDENT** деп көрсеткен ыңғайлы болып саналады немесе **C:\TC\BIN** каталогын қалдыру үшін — нүкте "." енгізе салу керек. Керекті параметрлер енгізілген соң, оларды **Options – Save...** командасымен есте сақтап қою қажет.

Window (Терезе) меню командалары терезені ашу, жабу, экранды жылжыту әрекеттерін орындау мүмкіндігін береді.

Size/Move	Ctrl+F5	– терезе көлемі мен орны
Zoom	F5	– терезені ұлғайту
Tile		– сатылы (каскадты)
Next	F6	– келесі терезе
Close	Alt+F3	– терезені жабу
Output		– нәтижелік терезе
User screen	Alt+ F5	– тұтынушы экраны
List All	Alt+0	– басқа терезелерді ашу

Help (көмек) меню командалары жүйедегі анықтамалық ақпаратты оқу мүмкіндігін береді.

Contents (экранға шығарылған ақпарат жөнінде мәлімет) ағымдағы уақытта экранға шығарылған мәлімет жөнінде (екпінді терезе, таңдалған меню командасы, жіберілген қате) мәліметті сұхбат терезеге шығарады.

Index (түйінді сөздер) Shift+F1 – жүйеде бар барлық анықтамалық ақпарат тізімін әліпбилік ретімен түйінді сөздер бойынша шығарады. **Topic search** (сөз бойынша іздеу) Alt+F1 – курсор орналасқан сөз жөнінде анықтамалықты шығарады. Егер сол сөз жөнінде анықтамалық жоқ болса, алғашқы символдарының саны көп сәйкес келетін түйінді сөздер тізімін береді.

Prevoius topic (алдыңғы тақырып) алдыңғы сұранысқа сәйкес келетін анықтамалықты шығарады. Жүйе 20 сұранысты сақтай алады.

12.3 Қателер коды және олардың мәліметтері

Турбо С ортасы программа компиляциядан өткен кезде пайда болған қателер жайлы толық мәлімет береді. Қате кездескен кезде орта автоматты түрде бастапқы программа мәтінін экранға шығарып, курсорды қате табылған орынға орналастырады және редактордың жоғарғы жолында диагностикалық мәлімет пайда

болады. F1 пернесінен басқа кез келген пернеге бассаңыз, жоғарғы жол бастапқы қалпына келіп, интегралданған орта редакциялау режиміне ауысады. Егер қате жайлы мәлімет шыққаннан кейін F1 пернесін басқанда, экранда қатені түзету жайлы нұсқаулар жазылған анықтама қызметінің терезесі пайда болады. Кейбір қателер бірден емес, программа мәтінін талдау барысында анықталады.

Қате нөмірі	Қате түсінігі
1	*Out of memory (Жады шекарасынан асып кету)*
2	*Identifier expected (Идентификатор көрсетілмеген)*
3	*Unknown identifier (Белгісіз идентификатор)*
4	*Duplicate identifier (Идентификатор қайталанған)*
5	*Syntax error (Синтаксистік қате)*
6	Error in real constant (*Нақты тұрақтыда қате бар*)
7	*Error in integer constant (Бүтін тұрақтыда қате бар)*
8	*String constant exceeds line (Тіркестік тұрақты берілген аймақтан тыс жатыр)*
9	*Too many nested files (Кіріктірілген файлдар саны тым көп)*
10	*Unexpected end of file (Файл соңы табылмады)*
11	*Line too long (Тым ұзын жол)*
12	*Type identifier expected (Тип идентификаторы қажет)*
13	*Too many open files (Ашылған файлдардың саны көп)*
14	*Invalid file name (Файл аты дұрыс емес)*
15	*File not found (Файл табылмады)*
16	*Disk full (Диск толып кеткен)*
17	*Invalid compiler directive (Компилятор директивасы қате)*
18	*Too many files (Файлдардың саны тым көп)*
19	*Undefined type in pointer definition (Көрсеткіш сипаттауында тип көрсетілмеген)*
20	*Variable identifier expected (Айнымалының идентификаторы жоқ)*

12.3.1 Программа орындалу кезінде шығатын қателер

Программа орындалу кезінде анықталатын кейбір қателер экранда Runtime error nnn at xxxx:yyyy (xxxx:yyyy адресі бойынша nnn кезенін орындаудағы қате) мәлімдемесінің шығуына әкеледі, мұнда nnn — қате нөмірі; xxxx:yyyy — адрес (сегмент немесе жылжу). Бұл мәлімдемеден кейін программа өз жұмысын тоқтатады. Программа орындалу кезінде шығатын қателер төртке бөлінеді: СОЖ (сұхбатты операциялық жады) анықтайтын қателер (1-ден 99-ға дейінгі қателер), енгізу-шығару қателері (100-ден 149-ға дейінгі қателер), дағдарысты қателер (критические ошибки) (150-ден 199-ға дейін) және фаталды қателер (200-ден 255-ке дейінгі қателер).

12.3.2 Операциялық жүйе анықтайтын қателер

Қате №	*Аты*
1	*Invalid function number (Функция нөмірі дұрыс жазылмаған)*
2	*File not found (Файл табылмады)*
3	*Path not found (Жол табылмады)*
4	*Too many open files (Ашылған файлдардың саны көп)*
5	*File access defined (Файлға қол жеткізуге рұқсат жоқ)*
6	*Invalid file handle (Ретсіз файлдық канал)*
12	*Invalid file access code (Файлға қол жеткізудің коды нақты емес)*
15	*Invalid drive number (Дискжетек нөмірі ретсіз)*
16	*Cannot remove current directory (Ағымдағы каталогты өшіруге болмайды)*
17	*Cannot rename across drives (Атты ауыстырғанда әртүрлі дискжетек аттарын көрсетуге болмайды)*

12.3.3 Енгізу-шығару қателері

Егер операторлардың біреуі {$I+} директивасымен компиляциядан өтсе, онда енгізу-шығару катесі программаның орындалуын тоқтатады. {$I–} калпында программа орындалуын жалғастырып, қате IORESULT функциясымен қайтарылады.

Қате №	*Аты*
100	*Disk read error (Дискіден оқу кезінде қате кетті)*
101	*Disk write error (Дискіге жазу кезінде қате кетті)*
102	*File not assigned (Файлға ат берілмеген)*
103	*File not open (Файл ашылмаған)*
104	*File not open for input (Файл енгізу үшін ашылмаған)*
105	*File not open for output (Файл жазу үшін ашылмаған)*
106	*Invalid numeric format (Сан форматы дұрыс емес)*

12.3.4 Дағдарысты қателер

Қате №	*Аты*
150	*Disk is write protected (Диск жазудан сақталған)*
151	*Unknown unit (Белгісіз модуль)*
152	*Drive not ready (Дискенгізгіш «дайын емес» қалып күйінде)*
153	*Unknown command (Бейтаныс команда)*
154	*CRC error in data (Бастапқы мәндерде қате бар)*
155	*Bad drive request structure length (Дискіден мәлімет аларда құрылым ұзындығы дұрыс көрсетілмеген)*
156	*Disk seek error (Дискіден оқитын құрылғы бастарын дискке орнату операциясы кезінде қате кетті)*
157	*Unknown media type (Тасымалдауыш типі белгісіз)*
158	*Sector not found (Сектор табылмады)*

159	*Printer out of paper (принтерде қағаз бітті)*
160	*Device write fault (Құрылғыға жазу кезінде қате кетті)*
161	*Device read fault (Құрылғыдан оқу кезінде қате кетті)*
162	*Hardware failure (Аппаратура жұмыс істемей тұр)*

12.3.5 Фаталды апатты қателер

Бұл қаталер программа жұмысының бірден тоқталуын жүзеге асырады.

Қате №	*Аты*
200	*Division by zero (Нөлге бөлу)*
201	*Range check error (Шекараларды тексеру кезінде қате шықты)*
202	*Stack overflow error (Стек толып кетті)*
203	*Heap overflow error (Мәлімет толып кетті)*
204	*Invalid pointer operation (Көрсеткішпен орындалатын белгісіз операция)*
205	*Floating point overflow (Жылжымалы нүктемен жұмыс жасау кезінде мәндер берілген аймақтан шығып кетті)*
206	*Floating point underflow (Жылжымалы нүктемен жұмыс жасау кезінде реті бұзылды)*
207	*Invalid floating point operation (Жылжымалы нүктемен жұмыс жасауға болмайтын операция)*
208	*Overlay manager not installed (Оверлейді басқарудың ішкі жүйесі орнатылмаған)*
209	*Overlay file read error (Оверлейлік файлды оқу кезінде қате жіберілген).*
210	*Object not initialized (Объект инициалданбаған)*
211	*Call to abstract method (Абстракты ережені шақыру)*
212	*Stream registration error (Тіркеуге алу ағынында қате бар)*
213	*Collection index out of range (Терілген индекс диапазон шекарасынан тыс жатыр)*
214	*Collection overflow error (Коллекция толып кеткен)*

Блокнотта немесе Word программасында терілген мәтіндерді пайдалану

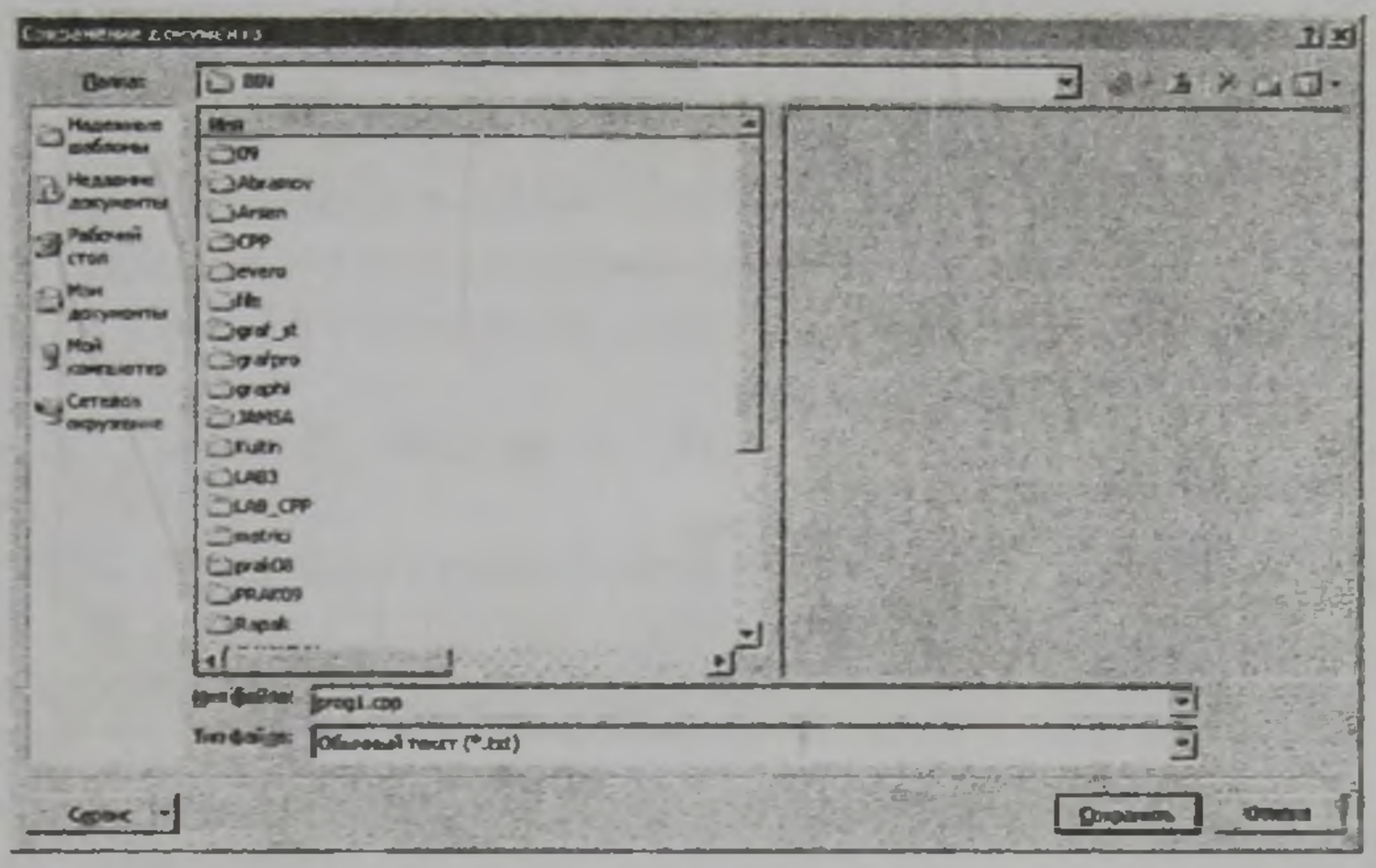
Word программасында немесе Блокнотта терілген программаны Turbo C ортасында пайдалану үшін келесі әрекеттерді орындау керек:

1. программа мәтінін Word (Блокнот) программасында теріп шығу;
2. *File – Save as…* (*Файл – Сохранить как...– Басқаша сақтау)* командасын орындап, *Құжатты сақтау* терезесіндегі *Файл типі...* өрісінен *қарапайым мәтін* (*обычный текст* (*.txt)) түрінде C: немесе D: дискісінің TC каталогындағы BIN бумасындағы (немесе басқа бір дискідегі) керекті орынды таңдау;
3. *Құжатты сақтау* терезесіндегі *Файл аты...* (*Имя файла...*) өрісіне керекті атауды енгізіп, *.cpp* типімен сақтау, мысалы, *prog1.cpp* (12.6а сурет);
4. Келесі шығатын *Файлды түрлендіру* (*Преобразование файла*) сұхбат терезесінде MS-DOS өрісі алдына нүкте (⊙) қою (12.6ә сурет);

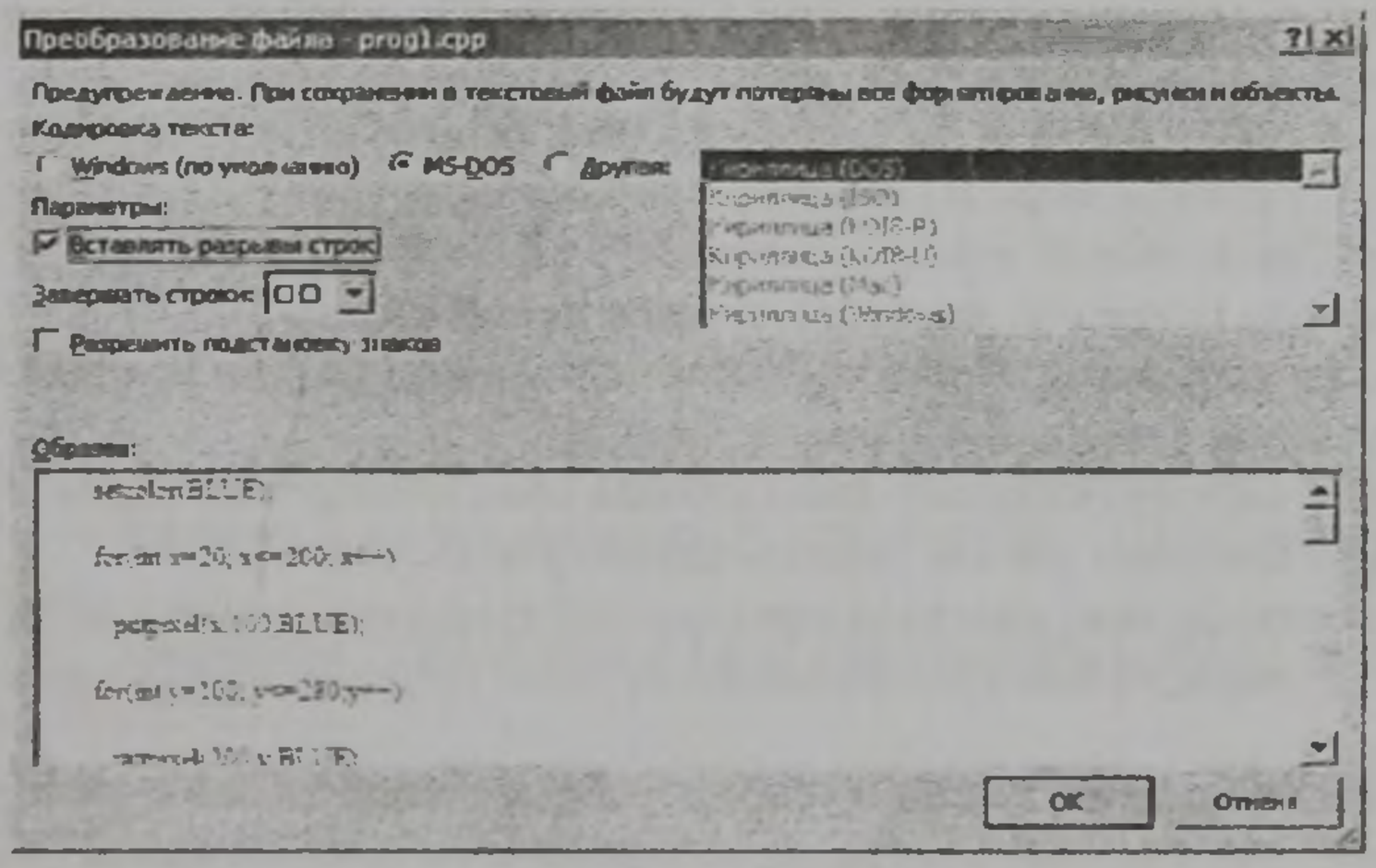
5. *Жол аралықтарын бөлу (Вставлять разрывы строк)* өрісі алдына қанатбелгі (✔) орнату (12.6ә сурет);

6. *ОК* батырмасын шерту.

Осыдан кейін Turbo C (Borland C) программалау ортасында *File – Open* командаларын орындап, өзіміз жазған бума ішінен керекті файлды (prog1.cpp) тауып, оның атын шерту қажет.



а)



ә)

12.6-сурет. Word ортасында терілген программа мәтінін сақтау кезінде *Құжатты сақтау* (а) және *Файлды түрлендіру* (ә) терезелерін пайдалану

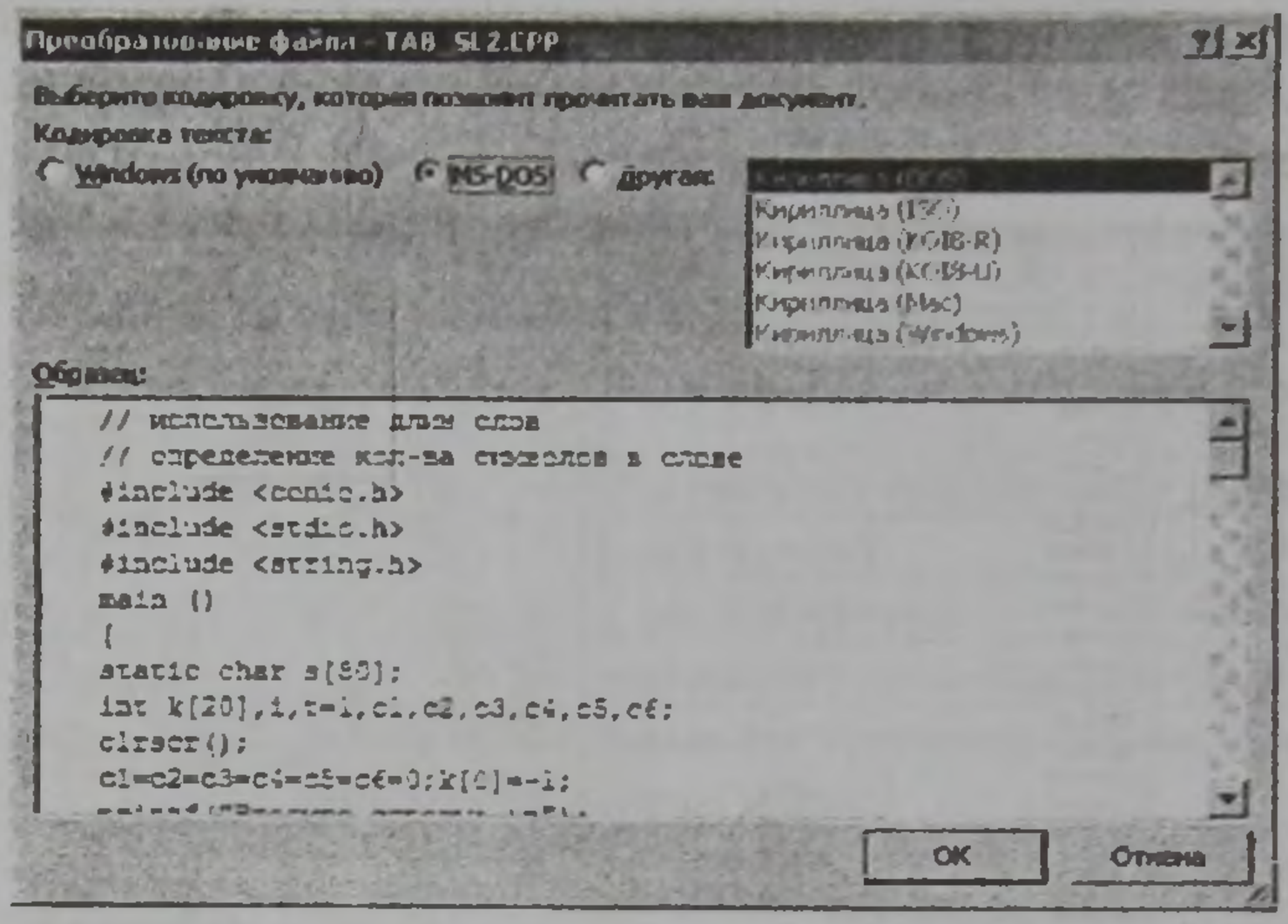
Енді Turbo C (Borland C) программалау ортасында терілген программа мәтінін Word программасына көшіріп алу үшін төмендегі әрекеттер орындалады:

1. TC каталогындағы BIN бумасы ішіндегі немесе басқа орындағы керекті программалық файл атын (мысалы, prog1.cpp) тауып алып, тышқанның оң жақ батырмасын шертіп, сол сәтте шыққан терезеден *Открыть с помощью* ▸ *Microsoft Office Word* (немесе *Блокнот*) жолдарын таңдау керек;

2. Келесі шығатын *Файлды түрлендіру* (*Преобразование файла*) сұхбат терезесінде MS-DOS өрісі алдына нүкте (⊙) кою (12.7 сурет);

3. *ОК* батырмасын шерту.

Осыдан соң Word (немесе Блокнот) терезесі ашылып, оған таңдап алған файлдағы программа мәтіні шығады.



12.7-сурет. Word ортасына C/C++ программасын оқу

Бақылау сұрақтары

1. *Turbo C біріктірілген программалау жүйесі қандай қызмет атқарады және оның терезесі неше бөліктен тұрады?*
2. *Turbo C ортасында программа орындау қандай қадамдардан тұрады?*
3. *Turbo C ортасы терезесін үлкейту/кішірейту үшін не істеу керек?*
4. *Turbo C ортасында программа теру үшін неше терезе пайдаланылады? Керекті терезеге қалай көшуге болады?*
5. *Меню қатарына және жұмыс алабына ауысу үшін не істеуге болады?*
6. ***File** менюінің ішкі командалары қандай?*
7. *Файлды дискіге жазу/оқу үшін қандай командалар мен пернелер қолданылады?*
8. *Файлды басқаша атпен сақтау қалай орындалады?*
9. ***Edit** менюінің ішкі командалары қандай қызметтер атқарады?*
10. *Turbo C терезесінде ерекшеленген бөлікті буферге қалай қиып (көшіріп) алады?*
11. *Курсор орналасқан жерге буфердегі ақпаратты қалай кірістіріп қояды?*
12. *Turbo C терезесінде ерекшеленіп белгіленген бөлікті қалай өшіріп тастайды?*
13. *Программа мәтіні ішінен тауып алуға қажетті сөзді қандай сұхбат терезеге және қалай енгізуге болады?*
14. ***Run** менюінің ішкі командалары қандай қызметтер атқара алады?*
15. *Орнатылған параметрлерді қолдана отырып, редактор терезесіндегі екпінді программаны қай команда орындайды?*
16. ***Options** менюінің командалары қандай?*
17. ***Window** (Терезе) меню командалары қандай қызметтер атқарады?*
18. *Программаны компиляциялауда шыққан қателер жайлы мәлімет қалай алынады?*
19. *Программа орындалу кезінде шығатын қателер қандай топтарға бөлінеді?*
20. *Қатені түзету жайлы нұсқауларды қайдан алуға болады?*

ҚОЛДАНЫЛҒАН ӘДЕБИЕТТЕР

1. Павловская Т.А. С/С++. Жоғарғы деңгейлі тілде программалау. -Алматы: ЖШС "Дәуір", 2012. -461 б.
2. Павловская Т.А. Щупак Ю.А. С/С++. Структурное и объектно-ориентированное программирование: Практикум.-СПб.: Питер, 2011. -352 с.
3. Сэмюел П. Харбисон, Гай Л. Стил. Язык программирования Си. Пер. с англ. – М.: ООО "Бином-Пресс", 2004. -528 с.
4. Липпман С., Лажойе Ж. Язык программирования С++. Вводный курс, 3-е изд. Пер с англ. –СПб. –М.: Невский диалект – ДМК Пресс, 2003. -1104 с.
5. Шиманович Е.Л. С/С++ в примерах и задачах.-Мн.: Новое знание,2004. -528с.
6. Щупляк В.И. С++. Практический курс: уч. пособие. –Минск: Новое знание, 2008. -576 с.
7. Керниган Б., Ритчи Д., Фьюэр А. Язык программирования Си. -М. : Финансы и статистика, 2004. -271с.
8. Страуструп Б. Язык программирования С++, спец. изд. Пер. с англ. -М.: БИНОМ; -СПб.: Невский диалект "БИНОМ", 2008. -288 с.
9. Вирт Н. Алгоритмы и структуры данных: Пер. с англ. 2-е изд., испр. -СПб.: Невский диалект,2008. -352с.
10. Канцедал С.А. Алгоритмизация и программирование: уч. пособие. -М.: ИД «Форум»: ИНФРА-М, 2008. -352 с.
11. Абрамов С.А., Гнездилова Г.Г., Капустина Е.И., Селюн М.И. Задачи по программированию. -М.: Наука, Гл. ред. физ.-мат. лит. 1988. -224 с.
12. Культин Н.Б. С/С++ в задачах и примерах. 2-е изд., перераб. и доп. -СПб.: БХВ-Петербург, 2009. -368 с.
13. Березин Б.И., Березин С.Б. Начальный курс С и С++ . -М: ДИАЛОГ-МИФИ, 2004. -288с.
14. Подбельский В.В., Фомин С.С. Программирование на языке Си. -М.: ФиС, 2004. – 600 с.
15. Костюкова Н.И., Калинина Н.А. Язык Си и особенности работы с ним. –М.: «Интернет-университет информационных технологий - ИНТУИТ.ру», 2006. -208 с. // www.intuit.ru
16. Рапаков Г.Г., Ржеуцкая С.Ю. Turbo Pascal для студентов и школьников. –СПб.: БХВ-Петербург, 2007. -352 с.
17. Бөрібаев Б., Дүйсебекова К. Си тілінде программалау: Оқу-әдістемелік құрал. – Алматы: Қазақ университеті, 2007. -208 б.
18. Бөрібаев Б. Алгоритмдеу, мәліметтер құрылымы және программалау тілдері: оқулық. –Алматы: Қазақ университеті, 2012. -235 б.
19. Құралбаев З.Қ. Алгоритмдеу және программалау тілдері. –Алматы: «TST-company» баспасы, 2008. -354 б.
20. Бөрібаев Б. Компьютердің арифметикалық негіздері: Оқу құралы. -Алматы: Қазақ университеті, 2009. -80 б.
21. Балапанов Е., Бөрібаев Б., Бекбаев А.,т.б. Информатика терминдерінің қазақша-ағылшынша-орысша, орысша-қазақша-ағылшынша, ағылшынша-орысша-қазақша сөздігі. -Алматы: Сөздік-Словарь,1998. -176 б.

ИНТЕРНЕТ РЕСУРСТАРЫ

1. **http://www.research.att.com/~bs/homepage.html** – C++ тілінің авторы Бьярн Страуструп сайты.
2. **http://anubis.dkuug.dk/jtcl/sc22/wg21/** – C++ тілі стандарты туралы.
3. **http://msdn.microsoft.com** – Microsoft компаниясы анықтамалық мәліметтері сайты.
4. **http://www.firststeps.ru/** – "Алғашқы қадамдар " сайты (жаңа үйренушілерге арналған сайт).
5. **http://www.cppwmeste.r2.ru/** – бұл да жаңа үйренушілерге арналған сайт.
6. **http://rsdn.ru** – Ресей программалаушылары сайты. Сайттың өз форумы бар.
7. **http://staff.develop.com/slip/** – C++ тілінде бірсыпыра кітаптар авторы Стэнли Липпманның сайты.
8. Программалаушыларға арналған сайттар:
   - **http://www.codetools.com/**
   - **http://www.codeproject.com/**
   - **http://www.codenet.ru/**
   - **http://www.codeguru.com/**
   - **http://www.doc.ic. ac. uk/lab/cplus/c++. rules/**
   - **http://www.helloworld.ru/**
   - **http://www.emanual.ru**
   - **http://www.sourcesl.ru**
   - **http://www.rusdoc.ru**
9. **http://gcc.gnu.org** — C++ компиляторының тегін таратылатын нұскасын алуға болатын сайт.
10. **http://www.digitalmars.com/** – бұл да тегін компилятор сайты.

Стандартты математикалық функциялар

Math.h тақырып файлында стандартты математикалық функциялардың анықтаулары берілген, оларды программа құру кезінде дұрыс жаза білу керек (Қ1.1-кестені қ.).

Қ1.1-кесте

Функция прототипі	Қайтаратын мәні
double acos(double x)	arccos x
double asin(double x)	arcsin x
double atan(double x)	arctg x
double atan2(double x,double y)	arctg (y/x)
double ceil(double x)	"жоғары" қарай дөңгелектеу
double cos(double x)	cos x
double cosh(double x)	ch x
double exp(double x)	e^x
double fabs(double x)	| x |
double floor(double x)	"төмен" қарай дөңгелектеу
double log(double x)	ln x
double log10(double x)	lg x
max(a,b)	максимум (a,b) типі үлкен аргумент типімен бірдей болады
min(a,b)	минимум (a,b) типі кіші аргумент типімен бірдей болады
double pow(double x, double y)	x^y
double pow10(int p)	10^p
double sin(double x)	sin x
double sinh(double x)	sh x
double sqrt(double x)	x-тің квадрат түбірі
double tan(double x)	tg x
double tanh(double x)	th x
double hypot((double x, double y)	x^2+y^2 квадрат түбірі
double poly(double x, int n, double *a)	полинөм мәні
double ldexp(double x, int n)	$x*2^n$

Бұл кестедегі max және min функцияларының прототиптері, негізінде, функция емес, өздеріне сәйкес макроанықтаулар түрінде stdlib.h тақырып файлында берілген.

Аргументтері типі де, өз типі де double болып келген функциялардың long double типіндегі аналогтары бар. Олардың аттары жоғарғы кестеде көрсетілген функция аттарынан l әрпін қосу арқылы алынады, мысалы, (fabs, fabsl), (acos, acosl), т.с.с.

Ceil(x) және floor(x) функцияларының айырмашылығы – алғашқысы берілген нақты санды жоғарғы жақтағы бүтін санға, ал екіншісі – төменгі жақтағы

бүтін санға қарай дөңгелектейді. Екі функция да double форматындағы бүтін санды қайтаратынын есте сақтаған жөн. Мысалы:

```
ceil(0.1) =1.0          floor(0.1) = 0.0
ceil(0.5) =1.0          floor(0.5) = 0.0
ceil(0.9) =1.0          floor(0.9) = 0.0
ceil(-0.9)=0.0          floor(-0.9)=-1.0
ceil(-0.5)=0.0          floor(-0.5)=-1.0
ceil(-0.1)=0.0          floor(-0.1)=-1.0
```

Көбінесе программалаушылар дөңгелектеу үшін floor(x+0.5) функциясын пайдаланады. Бірақ ол кейде математикадағы ұғымнан алшақ кетіп жатады, мысалы,

floor(-0.5+0.5)=0.

Әрине, C/C++ тілдерінде Паскаль тіліндегідей round функциясының баламасы жоқ, дегенмен ондай функцияны колдан жасап алуға да болады:

```
int round(double x)
{ int res;
  res=(x<0)? x-0.5 : x+0.5;
  return res;
}
```

Егер нақты санның бірнеше бөлшек таңбаларын (k) алып дөңгелектеу керек болса, онда ол санды 10^k-ға көбейтіп алып, бүтін санға дейін дөңгелектеп, қайтадан 10^k-ға бөлу керек.

Math.h тақырып файлында стандартты константалардың да аныктаулары берілген, оларды да программа құру кезінде дұрыс жазып пайдалана білу керек (К1.2-кестені к.).

К1.2-кесте

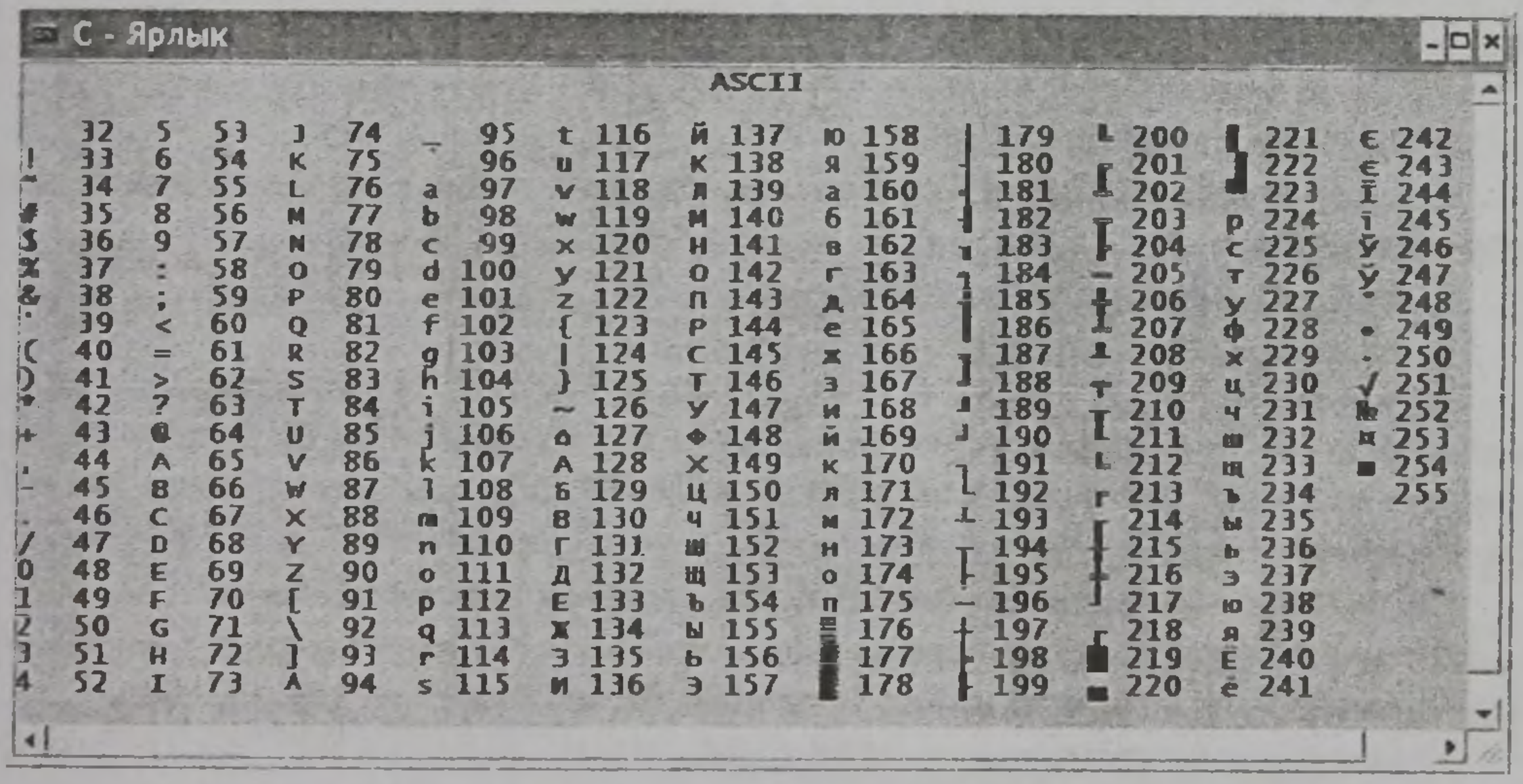
Константа	Мәні	Константа	Мәні
M_PI	π	M_E	e=2.71828...
M_PI_2	$\pi/2$	M_LOG2E	$\log_2 e$
M_PI_4	$\pi/4$	M_LOG10E	log e
M_2_PI	$2/\pi$	M_LN2	ln 2
M_1_SQRTPI	$1/\sqrt{\pi}$	M_LN10	ln 10
M_2_SQRTPI	$2/\sqrt{\pi}$	M_SQRT2	$\sqrt{2}$=1.414...
		M_SQRT_2	$\sqrt{2}/2$

Символдық мәліметтер және оларды компьютерде бейнелеу

Жеке берілген символдық мәліметтер (константалар мен айнымалылар) компьютер жадында сол символдардың ASCII-кодтарына сәйкес бүтін сандық мәндер жазылған бір-бір байт орын алады. Соларды экранға шығару үшін келесі программаны орындауға болады:

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int i,j;
  gotoxy(37,1);
  printf("ASCII");
  for(i=32; i<=52; i++)
    {  gotoxy(1,i-29);
       for(j=i; j<=255; j+=21)
         printf("%c %3d  ",j,j);
    }
  getch();
}
```

Бұл программа жұмысының нәтижесі:



Қ1.2-сурет. Жеке символдық мәліметтердің ASCII-кодтық кестесі (code page 866)

Кестедегі символдар "босорын" таңбасы кодынан – 32-ден (0x20) басталады, оған дейінгі символдар басқару кодтарына сәйкес келеді. Орыс әліпбиінің бас әріптері 128 (0x80) кодынан басталады ('Ё' әрпі басқа орында тұр), ал кіші

әріптері 160-тан 175-ке (а-дан п-ға) дейін, сонан соң 224-тен 239-ға (р-ден я-ға) дейін орналасқан. Олардың арасында кестелер салу кезінде қолданылатын графикалық символдар орналасқан. Қазақ әріптерін пайдалану үшін Unicode кестесіне көшу керек, сондықтан C/C++ тілдерінде экранға қазақ әріптері шығарылмайды.

Бірбайттық символдық константалардың мәндері жалқы тырнақшаға (‘ – апострофқа) алынып жазылады да, үш түрлі тәсілмен беріледі. Олардың алғашқысы экранда бейнеленетін ASCII-код кестесінің символдарына (Қ1.2-суретті қ.) қатысты айтылады, бұлар апострофқа алынып жазылады:

```
'F', '%', 'Ş', 'ы', '5', '+', '"', 'q', 'Я', ' '
```

Соңғы константа «босорын» таңбасы. Апостроф таңбасын мұндай тәсілмен жаза алмаймыз. Сондықтан басқа тәсілді қолданамыз.

Екінші тәсіл кері қиғаш сызық (слеш) таңбасынан кейін керекті символдың оналтылық кодын жазу арқылы беріледі:

`'\x27'` ("апостроф" символының коды, $39_{10} = 27_{16}$)

`'\x5C'` (кері қиғаш сызық \ символының коды, $92_{10} = 5C_{16}$)

Он алтылық кодтың алдыңғы нөлсіз жазылатынына назар аударыңдар (оналтылық сандар түрінде берілген константаларды жазу кезінде \0x болып бейнеленеді). Осы тәсілді ASCII-код кестесінің бас жағында (1-31 кодтары) орналасқан басқару кодтарының кез келгенін жазу үшін қолдануға болады. C/C++ тілдеріндегі Escape-тізбектері болып табылатын басқару кодтары (4.6-кестені қ.) да осы тәсілмен жазылады. Мысалы:

`'\''` – "апостроф" символы

`'\\'` – "кері слеш" символы

Үшінші тәсіл "кері слеш" таңбасынан соң, символдың сегіздік кодын жазу болып табылады:

`'\47'` ("апостроф" символының коды, $39_{10} = 47_8$)

`'\134'` (\символының коды, $92_{10} = 134_8$)

Мұндағы сегіздік кодтың да алдыңғы нөлсіз жазылатынына назар аударыңдар (сегіздік сандар түрінде берілген константаларды жазу кезінде олардың алдында \0 таңбасы жазылады). Апостроф ішіндегі сегіздік код [0, 377] аралығында болуы тиіс.

Символдық айнымалылар олардың типін көрсететін `char` немесе `unsigned char` спецификаторы (түйінді сөзі) аркылы жарияланады. Оларға бірден мән беруге, яғни айтылған тәсілдердің біоімен инициалдауға болады:

```
char ch1='Я',ch2='\x9F',ch3='\237',ch4=
 0x9F,ch5=0237,ch6=159;
```

Осылай сан арқылы мән тағайындау тәсілі DOS-қосымшаларында да және Windows ортасында да символдардың дұрыс бейнеленуіне кепілдік бере алады. Windows ортасында консолдық программалар құру кезінде символдық мәндер тағайындау тәсілі ASCII-код кестесінің тек бірінші жартысында орналасқан символдар үшін ғана дұрыс жұмыс істей алады.

Б ҚОСЫМШАСЫ

Сөз тіркестерінен тұратын мәліметтермен операциялар орындау

Бірөлшемді символдық жиымдар (массивтер) түрінде берілетін сөз тіркестерін өңдеу үшін C/C++ тілдерінің жүйелік функциялар кітапханасында бірсыпыра амалдар (операциялар) карастырылған. Олардың прототиптері `string.h` тақырыптық файлына жинақталып, көбісінің атаулары `str` (ағылш. `string` – `тіркес` сөзінің алғашқы әріптері) таңбаларынан басталады. Тіркестерді өңдейтін функцияларды карастыру үшін, қолданылатын аргументтер мен олардың типтері жайлы ортақ белгілеулер туралы бірсыпыра түсініктер бере кетейік:

`S, S1,S2` – символдық жиымға нұсқауыш (көбінесе жиым аты);

`CS` – `const char *` типіндегі нұсқауыш (яғни өзгермейтін жиым немесе бастапқы мәліметтер түрінде берілген тіркестік константа);

`ch` – символ коды, көбінесе `int` типіндегі сандық мән;

`k` – символдар саны.

Қ1.3-кесте

Функция	Атқарылатын операция
Сөз тіркесінің (тіркестің) ұзындығын анықтау	
`strlen(CS)`	`S` тіркесіндегі символдар санын береді
Тіркестерді қалыптастыру	
`strcpy(S1,CS2)`	`CS2` тіркесін `S1`-ге көшіреді, нұсқауышты `S1`-ге қайтарады
`strncpy(S1,CS2,k)`	`CS2` тіркесінің алғашқы `k` символын `S1`-ге көшіреді, нұсқауышты `S1`-ге қайтарады
`strset(S,ch)`	`S` тіркесін `ch` символдарымен толтырады, нұсқауышты `S`-ке қайтарады
`strnset(S,ch,k)`	`S` тіркесінде `ch` символын `k` рет қайталайды, нұсқауышты `S`-ке қайтарады
Тіркестерді біріктіру (конкатенация)	
`strcat(S1,CS2)`	`CS2` тіркесін `S1` тіркесінің соңына косып жазады, нұсқауышты `S1`-ге қайтарады (`S1` тіркесінің ұзындығы екі тіркес сиятындай көлемді болуы тиіс)
`strncat(S1,CS2,k)`	`CS2` тіркесінің алғашқы `k` символын `S1`тіркесіне косып жазады да, нұсқауышты `S1`-ге қайтарады
Регистрді ауыстыру	
`strlwr(S)`	`S` тіркесі символдарын кіші әріптерге ауыстыру, тек латын әліпбиі әріптеріне әсер етеді
`strupr(S)`	`S` тіркесі символдарын бас әріптерге ауыстыру, тек латын әліпбиі әріптеріне әсер етеді
Тіркесті керісінше жазып шығу	
`strrev(S)`	`S` тіркесіндегі символдарды кері бағытта жазып шығады
Сандық мәліметке түрлендіру	
`strtol(CS,ptr,r)`	`CS` тіркесінде символдық түрде бейнеленіп, негізі `r` болып келетін санау жүйесінде жазылған сан `long` типіндегі машиналық форматтағы санға түрлендіріледі. `ptr` нұсқауышына түрлендіруді токтаткан (үзген) символ адресі жазылады. Қайтарылатын мән – түрлендіру нәтижесі
`strtoul(CS,ptr,r)`	Жоғарыдағыдай түрлендіру ісі екі еселенген танбасыз бүтін сан үшін атқарылады
`strtod(CS,ptr)`	Нақты санды символдық түрден `double` типіндегі машиналық форматқа түрлендіру

Тіркестерді салыстыру	
`strcmp(CS1,CS2)`	Егер `CS1=CS2` болса, функция 0 мәнін қайтарады, егер `CS1>CS2` болса, функция 0-ден артық мән, ал егер `CS1<CS2` болса, функция 0-ден кіші мән қайтарады
`strncmp(CS1,CS2,k)`	`CS1` және `CS2` тіркестерінің алғашқы k символдары ғана салыстырылады
`stricmp(CS1,CS2)`	Салыстыру кезінде бас әріптер мен кіші әріптер кодтарының әртүрлілігі есепке алынбайды
`strcmpi(CS1,CS2)`	Жоғарыдағы операция сияқты, тек функция аты ғана басқаша
`strnicmp(CS1,CS2,k)`	Екі тіркестің алғашқы k символдарын бас әріп пен кіші әріп кодтарының айырмашылығын есепке алмай салыстырады
`strncmpi(CS1,CS2,k)`	Жоғарыдағы операция сияқты, тек функция аты ғана басқаша
Символды іздеу	
`strchr(CS,ch)`	`CS` тіркесі солдан оңға қарай `ch` символы табылғанша біртіндеп қарастырылады. Егер ол табылса, нұсқауыш `CS` тіркесіндегі осы символды (позициясын) көрсетіп тұрады, егер ондай символ табылмаса, онда нұсқауыш `null` (яғни 0) мәнін қайтарады
`strrchr(CS,ch)`	Жоғарыдағы операция сияқты, тек символды іздеу `CS` тіркесінің соңынан басына қарай жүргізіледі
Ішкі тіркесті іздеу	
`strstr(CS1,CS2)`	`CS2` тіркесін `CS1` тіркесінің құрамынан іздеу (алғашқы кіруін анықтау). Егер ол табылса, нұсқауыш табылған тіркестің алғашқы символын көрсетеді, табылмаса, ол `null` мәнін береді
Арнайы іздеу	
`strpbrk(CS1,CS2)`	`CS2` символының `CS1` тіркесіндегі алғашқы кездесуі ізделеді Табылған символға нұсқауыш немесе null қайтарылады
`strspn(CS1,CS2)`	Толығынан `CS2` символдарынан тұратын `CS1` тіркесінің бастапқы фрагментінің ұзындығы анықталады (символдар реттілігі ешкандай рөл атқармайды)
`strcspn(CS1,CS2)`	`CS2` символдарының бір де бірі құрамына кірмейтін `CS1` тіркесінің бастапқы фрагментінің ұзындығы анықталады
`strtok(S1,CS2)`	`S1` тіркесі ішінен `CS2` символдарымен бөлінген лексемдерді іздеу

Қ1.3-кестесінде көрсетілген функцияларға аздаған қосымша түсініктер бере кетейік.

Саннын символдық бейнеленуін соған сәйкес машиналық форматқа түрлендіретін `strtol` және `strtoul` функцияларында оның негізін `r=0` деп те беруге болады. Мұндайда санау жүйесінің негізі саннын символдық жазылуымен анықталады. Егер тіркес `'0'` символынан басталып, ары қарай 7-ден аспайтын цифрлар орналасатын болса, онда ол сегіздік сан болып саналады. Ал егер сөз тіркесі `'0x'` немесе `'0X'` символдарынан басталып, ары қарай оналтылық цифрлар орналасатын болса, онда `r=16` болып есептеледі.

`Strtok` функциясында лексем болып ажыратқыш символдарының бірімен (босорын, үтір, нүкте, т.с.с.) аяқталатын символдар тіркесі саналады. Ол функцияны алғашқы рет пайдалану кезінде `S1` тіркесінде бастапқы лексем орналасып, қайтарылатын мән оның бірінші символына нұсқауыш болып саналады. Осымен қатар `S1` тіркесіндегі анықталған ажыратқыш символ орнына нөлдік байт жазылады. Бұл табылған лексеммен кейіннен сөз тіркесі ретінде жұмыс істеуге мүмкіндік береді. Мұнан кейінгі `strtok` функциясын пайдалану кезінде келесі лексемдерді іздеу үшін бірінші аргумент орнына нөлдік аргумент

жазуға болады. Сонда функция енгізілген нөлдік байттан оң жақта орналасқан келесі лексемді іздеп табуға кіріседі. Осы тәсілмен S1 тіркесіне кіретін барлық лексемдерді біртіндеп тауып алуға болады. Түсіндіру үшін бір мысал келтірейік:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{ char *ptr;
  ptr=strtok("FEB.14,2009",".,-/");
  while(ptr!=NULL)
  { printf("ptr=%s\n",ptr);
    ptr=strtok(NULL, ".,-/");
  }
  getch();
}
//=== Бұл программа нәтижесі ===
ptr=FEB
ptr=14
ptr=2009
```

В ҚОСЫМШАСЫ

Глоссарий

ASCII-кодтар (American Standart Code for Information Interchange) – символдар кодтарының негізі болып табылатын американдық кодтар стандарты

Айнымалы – программа орындалуы барысында өз мәнін өзгерте алатын объект

Айнымалылар – идентификаторлармен белгіленіп, программаның орындалу барысында әртүрлі мәндерді қабылдай алатын шамалар

Айнымалының қасиеттері:

1) кез келген сәтте айнымалының белгілі бір мәні болады немесе ол анықталмаған болып есептеледі;
2) айнымалыға соңғы берілген мән оның алдыңғы мәнін өшіріп жібереді.
3) айнымалыны таңдау (оқу) және оны пайдалану айнымалының мәнін өзгертпейді

Айнымалылардың әрекет ету аймағы – программаның ағымдағы әрекеттеріне қандай мәліметтердің қатынасуға болатынын анықтау аймағы

Айыру белгілері – бос орын, барлық басқару символдары, ENTER (келесі жолға көшіру) пернесін басу белгісі және түсініктемелер

Алгоритм – алға қойған мақсатқа жету немесе берілген есепті шешу бағытында атқарушыға біртіндеп қандай әрекеттер жасау қажеттігін әрі түсінікті әрі дәл етіп көрсететін нұсқаулар жиыны

Алгоритм – берілген есептің шығару жолын реттелген амалдар тізбегі түріне келтіру

Алгоритм – берілген мәндерді пайдаланып қажетті нәтижеге жетуді жүзеге асыратын әрекеттердің орындалу ережесі

Алгоритм атқарушы – көрсетілген іс-әрекеттер тізбегін бұлжытпай орындай отырып, керекті нәтиже алуды жүзеге асыратын машина, құрылғы немесе адам

Алгоритмді өрнектеу түрлері:

1) табиғи тіл арқылы жазу;
2) белгілі бір түйінді сөздер – терминдер (псевдокодтар – жалған кодтар) арқылы қысқаша тізбекті түрде жазу (карапайым алгоритмдік тілде);
3) график жолымен (блок-схема арқылы) жазу;
4) программалау тілдерінде жазу

Алгоритмдік тіл немесе программалау тілі – жазу ережелері карапайым болып келетін жасанды тіл

Алгоритмнің қасиеттері:

1) алгоритмнің айқын, дәл өрнектелу қасиеті;
2) алгоритмнің үздіктілік қасиеті;
3) алгоритмнің нәтижелілік қасиеті;
4) Алгоритмнің жалпылық немесе ортақтық қасиеті

Алгоритмнің формалды орындалуы – орындаушы алгоритм командаларын

орындай отырып, өзі атқарып отырған істің мән-жайына көңіл аудармай, әрекеттерді ретімен атқара отырып, белгілі бір нәтиже алуы

Атқарушының командалар жүйесі – алгоритм атқарушы орындай алатын командалар жиыны

Ауыстырғыш оператор (switch) – берілген өрнек мәні бойынша программада кездесетін бірнеше нұсқаның бірін таңдап алуға мүмкіндік беретін шартты оператор

Басқару операторлары – шартсыз және шартты көшу операторлары, цикл (қайталау) ұйымдастыру операторлары

Басқару тізбектері – мәліметтер енгізу мен шығаруда қолданылатын арнайы символдар тіркесі (C/C++ тілінде)

Бейнемонитор (немесе дисплей) – электрондық сәулелік түтікшеден немесе сұйық кристалды тақтадан тұратын мәтін және графикалық бейнелер шығарылатын компьютер құрылғысы

Бос оператор – ешқандай да әрекеттің орындалмайтынын көрсететін оператор

Вирт синтаксистік диаграммасы – тіл ережелерін графикалық түрде бейнелейтін синтаксистік диаграмма (Н. Вирт көп қолданған)

Драйверлер – дербес компьютерлердің техникалық құрылғыларын басқаратын арнайы программа

Енгізу/шығару операторы – адам мен компьютер арасында мәлімет алмасу мақсатында алдын ала анықталған Input, Output файлдарын (Паскаль тілінде) қолданып, берілген мәндерді пернетақтадан енгізіп, алынған нәтижені дисплей экранына шығару операторлары

Жазбалар (record) – әртүрлі типтегі элементтерден құралған күрделі типтегі объектілер (Паскаль тілінде).

Жиым (массив) –бір атаумен белгіленіп, реттеліп біріктірілген бір типтегі мәліметтер жиыны

Жиын (set) – бір бірімен логикалық түрде байланыскан бір типтегі объектілер (элементтер) тобы.

Идентификатор – міндетті түрде әріптен басталатын сандар мен әріптер (астын сызу белгісі де) тізбегі

Көшу (немесе шартсыз көшу) *операторы* – операторлардың рет-ретімен орналасуын бұзып, келесі атқарылуды белгісі бар жолға беру операторы

Қабатталған циклдер – бір цикл тұлғасы ретінде (цикл ішінде) басқа циклдік құрылым тұратын күрделі циклдерді ұйымдастыру тәсілі

Қосалқы программа (подпрограмма) – белгілі бір ат қойылып, жеке программа түрінде бөлек жазылған, қажет кезінде оны қайталап пайдаланып отыруға болатын негізгі программаның арнайы бөлігі

Қосымша программа – алдына ала ат қойылған командалар тобын программаның кез келген жерінен оның атын көрсету аркылы шақырып орындау

Құрама оператор – бір-бірінен нүктелі үтір арқылы бөлінген бірнеше операторларды begin және end түйінді сөздерімен (Паскаль тілінде) немесе жүйелі жақшалармен ({ және } С тілінде) шектей отырып, оларды бір оператор тәрізді орындалатын етіп біріктіру жолы

Құрылымдар – C/C++ тіліндегі өзара логикалық байланысқан әртүрлі типті мәліметтерді байланыстыру

Құрылымдық операторлар – басқа операторларды белгілі бір ережелер бойынша біріктіру жолымен құрастырылған операторлар тобы. Олар үш топқа – құрама, шартты және қайталау операторларына жіктеледі

Логикалық амалдар:

AND - және (логикалық көбейту);
OR - немесе (логикалық қосу);
NOT - емес (терістеу немесе жоққа шығару);
XOR - арифметикалық *немесе* амалы;
SHL - биттер тіркесін солға ығыстыру;
SHR - биттер тіркесін оңға ығыстыру

Машинаға бағытталған тілдер компьютердің ерекшеліктерін есепке ала отырып әріптерді де пайдаланатын тілдер

Мәлімет енгізу – бастапқы мәндерді пернетақтадан, дискілерден немесе енгізу-шығару порттарынан оқу арқылы жүзеге асыру

Мәліметтер – белгілі бір процесс көмегімен тасымалдап, өңдеуге болатын, формальды түрде бейнеленген фактілер мен идеялар

Мәліметтер – сан мәндерін, мәтін ретіндегі сөз тіркестерін де мән ретінде қабылдай алатын тұрақтылар (константалар), айнымалылар, тағы осылар тәрізді құрылымдар немесе солардың адрестері

Меншіктеу операторы – жазылған өрнектердің мәнін есептеп, оны айнымалыға (функцияға) беру операторы

Нақты сандар – аралас сандардың бүтіні мен бөлшегін нүкте (немесе Е) арқылы бөліп жазу тәсілі

Нәтиже алу (шығару) – аралық немесе қорытынды мәліметтерді экранға, дискіге немесе енгізу-шығару порттарына жазу

Нұсқауыш – С тіліндегі мәліметтердің адресін сақтайтын айнымалы, яғни адрестің символдық түрде кескінделуі

Оператор – операциялар мен мәндерді көрсететін немесе солардың элементтерінің қай жерде орналасқанын білдіретін символдар жиыны

Оператор – программалау тілінің белгілі бір іс-әрекетті орындай алатын тиянақты мағынасы бар ең карапайым сөйлемі

Операциялар немесе амал-әрекеттер – берілген, есептелген мәндерді меншіктеу, соларды өңдеу, салыстыру істерін орындау

Өрнек – арифметикалық немесе логикалық амалдар таңбасымен біріктірілген

айнымалылар, атаулар, функциялар, жиымдар тағы да басқа мағынасы бар сөздер тізбегі

Параметрлі цикл операторы (for) – параметр өзінің алғашқы мәнінен соңғы мәніне дейін берілген қадаммен өзгеру барысында циклге кіретін бір немесе бірнеше әрекеттерді қайталап орындау ісін атқаратын оператор

Паскаль программалық блогы – программаның тақырыбы, сипаттау бөлімі және *begin* және *end* сөздерімен қоршалған операторлар бөлімінің жиыны

Паскаль тілі – 1968-1971 жылдары Швейцарияда профессор Никлаус Вирт жасап шығарған оқып үйренуге қолайлы программалау тілі

Пиксель – экранның адрестелетін ең кіші элементі

Проблемаға бағытталған тілдер шығарылатын есептердің ерекшеліктерін еске ала отырып, есептің математикада жазылу тіліне жақындастырылған тілдер

Программа – алгоритмді машинаға түсінікті нұсқаулар тізімі ретінде жазу

Программалау тілінің түйінді сөздері – бір-бірімен айыру белгілерімен бөлініп, программада алдын ала анықталған белгілі бір мағынасы бар арнайы идентификаторлар.

Программаны орындау ортасы – программалау тілдерінде программа кодын құрастырып, оны орындауға мүмкіндік беретін біріктірілген орта

Процедура – программадағы бірнеше операторларды олардың бастапқы параметрлерін өзгерте отырып, бірнеше рет қайталап орындауды жүзеге асыратын қосалқы программалар түрі

Рекурсия – процедура немесе функция операторларының орындалуы барысында оның өзін-өзі, яғни осы қосалқы программаның өзін өзінің ішінде пайдалану тәсілі

С – программалық жабдықтамалар жасауға және әртүрлі есептер шығаруға арналған құрылымдық программалау тілі

С++ – бұл С тілінің мүмкіндіктерін кеңейтетін тіл. 1980-жылдың басында жасалып шықты (Б.Страуструп). Объектіге бағытталған программалау қағидаларын қамтамасыз етіп, жаң идеяларды жүзеге асыра алатын тіл. Қазіргі көптеген заманауи тілдер синтаксисінің негізгі бастамасы осы тіл болып табылады

Стандартты функциялар – жиі кездесетін математикалық және басқа да функциялардың программалау тілінің ішкі объектісі ретінде берілуі

Сызықтық құрылым – бірінен кейін бірі орындалып тізбектеле орналасқан бірнеше әрекеттер (операторлар) жиыны

Таңдау операторы (switch) – бірнеше тармақты операторлардың ішінен берілген өрнектің мәніне байланысты біреуін таңдауды жүзеге асыратын оператор

Тармақты құрылым – шартқа байланысты екі операторлың бірінің орындалуы

Тип (мәліметтердің немесе *шамалардың типі)* – программалау оюъектілерінің қабылдай алатын мәндерінің және солармен орындауға болатын амалдардың жиыны, яғни шамалардың қабылдайтын мәндеріне берілетін сипаттама

Транслятор – программалау тілін машина тіліне аударатын программалар тобы

Тұрақты (немесе *константа*) – программаның орындалу барысында мәндері өзгеріссіз қалатын шамалар

Түйінді (өзекті, кілтті) сөздер – бұл компилятор үшін арнайы мәні бар, алдын ала аңықталған белгілі бір мағынасы бар идентикаторлар. Оларды тек анықталған мағынасы бойынша ғана пайдалануға болады. Мысалы: float, if, break, т.с.с

Тілдің әліпбиі – программаның элементтерін құруда қолдануға болатын символдар жиыны

Тілдің қарапайым объектілері – сан, идентификатор, константа, айнымалы, функция, өрнек ұғымдары

Тіркестік (жолдық) константа – С тіліндегі қос тырнақшаға алынған символдар жиыны

Тіркестік айнымалы – мән ретінде апострофтармен қоршалған сөз тіркестерін қабылдайтын айнымалы

Тіркестік өрнек – амал белгілері, тіркестік тұрақтылар, айнымалылар және функция атауларынан құралған, ұзындығы 256 символдан артпайтын сөз тіркестерінен құралған тізбек

Файл – сыртқы есте сақтау құрылғыларында (магниттік дискілерде) орналастырылған және мәлімет өңдеу, тасымалдау кездерінде біртұтас күйде қарастырылатын мәліметтер жиыны

Файл (file) – магниттік дискілерде тізбектеле орналасқан жеке элементтерден тұратын белгілі бір ат қойылған ақпараттар тобы.

Функция – жұмыс нәтижесінде бір ғана мәнді анықтайтын қосалқы программалар (операторлар тізбегі) түрі

Цикл – белгілі бір шарт орындалса (кейде орындалмаса), көрсетілген командалардың бірнеше рет қайталанып атқарылуы немесе шарт көрсетілмей-ақ алдын ала олардың неше рет қайталанатынын бүтін санмен беру

Цикл – операторлар бөлігінің бірнеше рет қайталана орындалуы

Шартты оператор (if) – тармақталу процестері бар алгоритмдерді ұйымдастыратын, белгілі бір шарттың орындалуы немесе орындалмауына байланысты екі мүмкіндіктің бірін таңдау операторы

Шартты операция (? :) – берілген шартқа байланысты әртүрлі мән қабылдай алатын *шартты өрнектер* құратын үшорынды операция (C/C++ тілінде).

Шартын алдын ала тексеретін цикл операторы (while) – орындалар алдында қайталану шартының мәні есептеліп, соның ақиқат/жалған болуына байланысты циклді ұйымдастыру операторы.

Шартын соңынан тексеретін цикл операторы (do..while) – цикл тұлғасы орындалғаннан кейін қайталану шартының мәні есептеліп, соның ақиқат/ жалған болуына байланысты циклді ұйымдастыру операторы.

C/C++ ТІЛІНДЕ ПРОГРАММАЛАУДАН ТЕСТ СҰРАҚТАРЫ

1. Идентификатор дегеніміз не?
   1) **программадағы объектінің аты**
   2) динамикалық жады
   3) жиыннан тұратын массив
   4) программаның берілу жолы
   5) компиляторға арналған сөздер

2. Символдық типтегі шамаларға компилятор жадында канша байт орын бөлінеді?
   1) 1
   2) 2
   3) 10
   4) 8
   5) 27

3. Жиым (массив) дегеніміз не?
   1) ұяшықтарда орналасқан мәліметтер
   2) **бір атаумен аталған бір типті мәндер тізбегі**
   3) өлшемді анықтайтын тип
   4) динамикалық жадыны пайдаланатын көрсеткіш
   5) сілтемені анықтайтын типтер жиынтығы

4. Қандай да бір әрекеттер тізбегін орындайтын операциялар мен сипаттамалардың айқындалғаи тізбегін ... деп атайды.
   1) процедура
   2) рекурсия
   3) **функция**
   4) дұрыс жауап жоқ
   5) 1,2

5. Функцияның сипаттамасы басындағы параметрлер қалай аталады?
   1) **формальды**
   2) фактілі
   3) аргументті
   4) 2,3 жауап дұрыс
   5) дұрыс жауап жоқ

6. С тілінде кез келген программаның орындалуы қай функциядан басталады?
   1) random
   2) **main()**
   3) randomize()
   4) clrscr()
   5) Әр программада әртүрлі

7. Өзін-өзі шақыратын функцияны не деп атаймыз?

1) процедура
2) функция
3) **рекурсия**
4) 1,2
5) дұрыс жауап жоқ

8. Кітапханалық функциялар қайда орналасқан?

1. кітапханалық файлдарда

1) С ішінде
2) компьютерде
3) дұрыс жауап жоқ
4) **интернетте**

9. С әліпбиі:

1) араб цифрі: 0-9
2) арнайы белгілер
3) түйінді сөздер
4) операциялар таңбалары
5) **барлығы дұрыс**

10. Латын әріптерінен, цифрлардан тұратын тек әріптерден басталуы тиіс таңбалар тізбегін ... деп атайды.

1) тұрақты
2) айнымалы
3) **идентификатор**
4) дұрысы жоқ
5) функция

11. a[3] = {1,3,5} жиымы берілсін делік. Осы жиымды экранға шығаратын программаны көрсетіңіз...

1) int i, a[3] = {1,3,5};
for(i=0;i<3;i++)
printf(" %d",a[i]);

2) int a[3] = {1,3,5}, i;
for(i=0; i<3;i++)
printf(" %f",a[i]);

3) int a[i] = {1,3,5}, i;
for(i=0;i<3,i++)
printf (" %k",a[i]);

4) int a[3], i;
for(i=0;i<3;i++)
printf(" %i",a[i+1]);

5) int a[3], i;
for(i=0;i<3;i++)
printf(" %c",a[i]);

12. С тілінде тұрақтыларды сипаттау үшін қандай түйінді сөз қолданылады?
1) int
2) var
3) const
4) procedure
5) дұрысы жоқ

13. return (өрнек) операциясының қызметі:
1) функцияны шақырған операторға өрнектің мәнін қайтарады
2) С тілінде ондай оператор қолданылмайды
3) функция мән қайтармаған кезде қолданылады
4) программаға әсемдік береді
5) қызметі анықталмаған, заң бойынша программа соңында тұруы керек

14. rand() функциясының тақырыптық файлда орналасқан прототипі қандай?
1) <math>
2) <string>
3) <stdlib>
4) <cmath>
5) rand

15. Функцияның ішінде ғана белгілі, соның ішінде анықталатын айнымалы қалай аталады
1. ауқымды (глобальді) айнымалы
2. жергілікті айнымалы
3. белгісіз айнымалы
4. белгілі айнымалы
5. рекурсивті айнымалы

16. Егер float пен double типтері араласса, нәтижесі ... болады;
1. float;
2. long;
3. **double**;
4. int;
5. short;

17. Кез келген блоктан немесе функциядан тыс хабарланған айнымалы...
1. ауқымды (глобальді) айнымалы
2. жергілікті айнымалы
3. программалық айнымалы
4. базалық айнымалы
5. унарлық айнымалы

18. ... компиляторға функция арқылы берілетін аргументтердің санын, типін, ретін анықтауға көмектеседі.
1. функцияның идентификаторы
2. функцияның прототипі
3. файлдың прототипі

4. жергілікті прототип
5. аумақтық идентификатор

19. Функцияның тақырыбында void түйінді сөзі не үшін қолданылады?
1. функция мән қайтаратынын көрсету үшін
2. функцияның тақырыбында void түйінді сөзі қолданылмайды
3. дұрыс жауабы жоқ
4. функцияның ешқандай мән қайтармайтынын көрсету үшін
5. функция мән сақтайтынын көрсету үшін

20. Қандай айнымалылар өзі сипатталған функцнядан шыққаннан кейін де өз мәнін сақтайды?
1. static секілді хабарланған жергілікті айнымалылар
2. аумақты айнымалылар
3. extern секілді хабарланған жергілікті айнымалылар
4. static секілді хабарланған аукымды айнымалылар
5. жергілікті айнымалылар

21. Блок дегеніміз не?
1. функцияның екінші аты
2. айнымалыларды сипаттауы бар құрама оператор
3. кездейсоқ оператор
4. қатені анықтайтын оператор
5. дұрыс жауап жок

22. Айнымалы дегеніміз не?
1. белгілі бір атауы бар жады аймағы, онда анықталған типтің мәліметтері сақталады
2. анықталмаған тип мәліметтерінің жадыда сақталуы
3. дұрыс жауабы жок
4. жады кластарының бірнеше бөліктерге бөлінуі
5. операндадан, операция белгісінен, жақшадан тұратын, анықталған типтің мағынасын білдіруі

23. Шартты операция форматын көрсетіңіз:
1. операнд_1? операнд_2 : операнд_3
2. операнд_1 := операнд_2
3. 1, 2
4. if (өрнек)оператор_1:[else оператор_2;]
5. if (тұрақты өрнек)1:[оператор тізбегі]

24. Жады кластарын беру үшін қандай спецификатор қолданылады?
1. auto
2. extern
3. static
4. register
5. барлығы дұрыс

25. С тілінде қайталау операторының қандай түрлері бар?
1. while (өрнек) оператор;
2. do ... while (өрнек)
3. дұрыс жауап жоқ
4. 1, 2, 5
5. for(инициализациялау; өрнек; модификация) оператор;

26. Тармақталу операторы дегеніміз не?
1. Алгоритмнің белгілі бір шарттың орындалуына немесе орындалмауына байланысты тармақталып, бірнеше жолдарға бөлінуі
2. дұрыс жауап жоқ
3. бір ғана шарттың орындалуын тексеретін оператор
4. шарттардың орындалмауын тексеретін оператор
5. операторлардың тармақталуын тексеретін оператор

27. #include директивасын қолданған кезде бұрыштық ("<",">") жақшаның орнына тырнақшаларды қолдануға болады ма, болса оның мәні неде?
1. болмайды
2. болады, бірақ одан ештеңе өзгермейді
3. болады, бұл уақытта файлды іздеу әрекеті бастапқы файлы бар каталогта жүргізіледі, сосын стандартты каталогтан іздейді
4. болады, онда файлды стандартты каталогтан іздейді
5. болады, бұл жағдайда файлды стандартты каталогтан іздейді, сосын бастапқы файлы бар каталогты қарастырады

28. Бастапқы ".h-файлдың" құрамына не кіреді?
1. типтің, тұрақтылардың, кіргізілген функциялардың, шаблондардың, тізімдердің анықталуы
2. функцияның, мәліметтің, шаблонның, аттың сипатталуы
3. атау кеңістігі
4. процедуралардың директивалары
5. көрсетілген жауаптың бәрі дұрыс

29. #define директивасы не үшін қолданылады?
1. тұрақтының мәнін беру үшін
2. макростар үшін
3. шартты компиляторды басқаруға арналған символ үшін
4. символдық константа үшін
5. берілген жауаптың бәрі дұрыс

30. Құрамында функциялары немесе анықталған мәліметтері бар бастапқы (тақырыптық) файлдардың кеңейтілуі қандай және ол файлдар қалай аталады?
1. ".htp", ".htp-файлдары"
2. ".h", ".h-файлдары"
3. ".c" ".c-файлдары"
4. ".f" ".f- файлдары"
5. ".exe", ".exe- файлдары"

32. Сөз тіркесінің (жолдың) ұзындығы қай функцияның көмегімен анықталады?

1. strlen
2. src
3. return
4. template
5. main()

32. 10 нақты саннан тұратын массивті сипаттау қай нұскада дұрыс берілген?

1. int a[10];
2. float a(10);
3. float a[10];
4. float a[1..10];
5. int a[1..10];

33. Операндтардан, операция таңбаларынан, жақшалардан тұратын .... мәндерін есептеу үшін қолданылады. Көп нүктенің орнындағы сөзді табыңдар.

1. өрнектер
2. айнымалылар
3. тұрақтылар
4. логикалық типтер
5. жады класы

34. Айнымалы дегеніміз не?

1. программа орындалуы барысында әртүрлі мәндер қабылдайтын шама
2. жады класының бірнеше аймақтарға бөлінуі
3. программалар тізбегі
4. алгоритмнің белгілі бір шартының орындалуы
5. дұрыс жауабы жоқ

35. Таңдау операторы (нұсқауы) қай сөз арқылы беріледі?

1) **switch**
2) return
3) for
4) if
5) class

36. Егер long пен float типтері араласса, нәтижесі ... болады.

1) char;
2) short;
3) int ;
4) **float;**
5) double.

37. Бөлгендегі қалдық табу операциясы қалай беріледі?

1. **%**
2. /
3. *

4. <<
5. &

38. Блок ішінде анықталған айнымалы қалай аталады?
1. ауқымды
2. тұрақты
3. жергілікті
4. тұрақсыз
5. сілтеуіш

39. Блоктың ішінде сипатталған идентификатор қалай аталады?
1. глобальды көрініске ие
2. локальді идентификатор
3. оператордың белгісі
4. функцияның прототипінде көрсетілген параметр
5. көрсетілген типтің айнымалысы құрылғаннан бастап, ол жойылғанша өмір сүреді

40. *a* атты бүтін айнымалының дұрыс сипатталуы:
1. **int *a*;**
2. *a* int;
3. integer *a*;
4. float *a*;
5. char *a*.

41. Блоктан тыс анықталған айнымалы қалай аталады?
1. ауқымды (глобальды)
2. жергілікті
3. символды
4. тұрақты
5. жергіліксіз

42. Қай нұсқада sizeof жазылуы дұрыс?
1. sizeof тип
2. sizeof [тип]
3. sizeof (тип)
4. sizeof: тип
5. sizeof_ тип

43. Логикалық терістеу қай нұсқада дұрыс берілген?
1. -
2. ~
3. =
4. !
5. --

44. extern спецификаторы нені білдіреді?
1. айнымалы программа ішінде анықталғанын
2. айнымалы программадан тыс анықталғанын

3. айнымалы анықталмағанын
4. айнымалы автоматты түрде анықталғанын
5. дұрыс жауабы жоқ

45. 12 бүтін саннан тұратын *a* массивінің дұрыс сипатталуы:
1. float *a*[12];
2. int *a*[12];
3. float *a*[10];
4. char *a*[12];
5. int *a*(12);

46. Функция тұлғасы қандай таңбалармен коршалып тұрады?
1. ()
2. / /
3. { }
4. * *
5. " "

47. 0...255 сандар диапазоны қай типке жатады?
1. double
2. unsigned char
3. bool
4. float
5. signed char

48. Goto нұсқауының дұрыс жазылуы:
1. Goto 65
2. Goto m65
3. Goto белгі
4. Goto ?65
5. Goto printf

49. Келесі операциялардың қайсысы тренарлы (үш операндты)?
1. !
2. &
3. <=
4. !=
5. ?:

50. С тілінде комментарийді қандай таңбалар ішінде жазады?
1. // //
2. (* *)
3. /* */
4. < >
5. { }

51. Программаның барлық объектілері үшін ортақ мәліметтерді сақтау үшін қандай класс қолданылады?

1. статикалық
2. тұрақты
3. рекурсивті
4. айнымалы
5. **аукымды (глобальді)**

52. Егер short пен int типтері араласса, нәтижесі – ... болады

1. char;
2. short;
3. **int;**
4. float;
5. double.

53. Нұсқауыштарды сипаттау форматы қай нұсқада дұрыс көрсетілген?

1. аты *типі
2. *аты типі
3. **типі *аты**
4. & аты типі
5. тип_аты

54. float *a*[10]; нені білдіреді?

1. 1-ден 10-ға дейінгі бүтін сандар тізбегін сипаттау
2. 1-ден 10-ға дейінгі нақты сандар тізбегін сипаттау
3. **10 нақты саннан тұратын *a* массивін сипаттау**
4. 10 мәнді қабылдай алатын *a* жиымын сипаттау
5. 10x10 өлшемді *a* матрицасын сипаттау

55. switch операциясынан шығу үшін қай нұсқау қолданылады?

1. do
2. **break**
3. case
4. return
5. default

56. Типтерді түрлендіру мысалы. Программа орындалғанда экранға не шығады:

```
main()
{ char ch;
  int i; float fl;
  fl=i=ch='A';
  printf(" %c  %d  %6.2f\n",ch,i,fl);}
```

1. **A 65 65.00 ;**
2. B 66 65.000;
3. B 63 63.000;
4. B 62 62.000;
5. A A 65.00.

57. *= операциясы нені білдіреді?
1. көбейту
2. меншіктеу
3. әр элементін көбейту
4. көбейтіп барып меншіктеу
5. комментарий

58. Лексем дегеніміз не?
1. сөз тіркестері
2. сөйлемдер
3. тілдің өзіндік мағынасы бар ең кіші бірлігі
4. программаның тұлғасы
5. бетті ауыстыру

59. " int n;" жазуы нені білдіреді?
1. n класы берілгенін
2. n спецификаторы берілгенін
3. n айнымалысы нақты мән қабылдайды
4. n айнымалысы бүтін мән қабылдайды
5. дұрыс жауабы жоқ

60. string.h – бұл мынадай функциялардың кітапханасы
1. сөз тіркесі функциялары
2. стандартты функциялар
3. енгізу/шығару базалық кітапханасы
4. графикалық функциялар
5. буфермен жұмыс істеу функциялары

61. graphics.h - бұл мынадай функциялардың кітапханасы
1. сөз тіркесі функцияларының
2. стандартты функциялардың
3. енгізу/шығару базалық кітапханасының
4. графикалық функциялардың
5. буфермен жұмыс істеу функцияларының

62. С тілінде айнымалылар программаның мынадай аймағында сипатталады
1. басында
2. аяғында
3. ортасында
4. кез келген жерінде
5. еш жерінде сипатталмайды

63. С тілінде бір таңбамен белгіленген бас және кіші әріптер мәні...
1. әртүрлі болып саналады
2. бірдей болып саналады
3. компиляторға байланысты өзгереді
4. тілдің нұсқасына байланысты өзгереді
5. барлығы да дұрыс

64. Айнымалыларға, тұрақтыларға, мәліметтер типіне және функцияларға қойылған атау былай аталады...

1. идентификаторлар
2. түйінді сөздер
3. директивалар
4. нұсқауыштар
5. дұрыс жауап көрсетілмеген

65. Келесі сөздер ішіндегі нақты сандар типтерін анықтайтын түйінді сөздерді көрсету керек:

1) char, 2) int, 3) float, 4) double, 5) long, 6) long int, 7) short, 8) signed, 9) unsigned

1. 1,4,6,7
2. 2,3,4,8
3. 1,7,8,9
4. 3,4,6
5. 3,4

66. Келесі сөздер ішіндегі бүтін сандар типтерін анықтайтын түйінді сөздерді көрсету керек:

1) char, 2) int, 3) float, 4) double, 5) long, 6) long double, 7) short, 8) signed, 9) unsigned

1. 1,2,5,7,8,9
2. 5,6,7,9
3. 1,2,3,6,9
4. 2,5,7,9
5. 2,7,8,9

67. Директиваның дұрыс жазылған жолын көрсетіңіз:

1. #define PI = 3.1415
2. #define PI == 3.1415
3. #DEFINE = 3.1415
4. #define PI = 3.1415;
5. #define PI 3.1415

68. sizeof() операциясы мынаны анықтайды:

1. литерлік өрнекті
2. константалық тұрақты өрнекті
3. операнд типіне бөлініп берілетін байттар санын
4. операнд орналасатын байттар санын
5. логикалық өрнекті

69. Программа нәтижесі нешеге тең болады?

```
void main( )
{int i = 5;
 switch (i++) {case 5: printf("%d %d", 1 , i); break;
               case 6: printf("%d %d", 2 , i); break;
               default: printf("%d %d", 3 , i); }}
```

1. 1 5
2. 2 6

3. 2 5
4. 3 6
5. 1 6

70. Программа нәтижесі нешеге тең болады?

```
void main()
{int i = 5;
 switch (++i) {case 5: printf("%d%d", 1 , i); break;
               case 6: printf("%d%d", 2 , i); break;
              default: printf("%d%d", 3 , i); }}
```

1. 1 5
2. 2 6
3. 2 5
4. 3 6
5. 1 6

71. int a[5] массивінде қандай элемент жоқ?

1. a[1]
2. a[2]
3. a[3]
4. a[4]
5. a[5]

72. Төмендегі қай функция файлға мәлімет жазады?

1. fread()
2. fscanf()
3. gets()
4. **fprintf()**
5. seek()

73. Төмендегі қай функция файлдан мәлімет оқиды?

1. fprintf()
2. **fscanf()**
3. fread()
4. fwrite()
5. seek()

74. Файлды одан мәлімет оқу ұшін ашу (режимін корсету керек)

1. r
2. w
3. a
4. r+
5. w+

75. Мәлімет жазатын файл ашу қажет, егер файл бұрыннан бар болса, ондағы мәлімет жойылады (режимін көрсету керек).

1. r
2. w
3. a
4. r+
5. w+

76. Мәлімет қосу: файл соңына мәлімет жазу үшін оны ашу қажет (режимін көрсету керек).
1. r
2. w
3. a
4. r+
5. w+

77. Файлдағы мәліметті оқып, оған мәлімет жазу: оқу және жазу (режимін көрсету керек).
1. r
2. w
3. a
4. r+
5.w+

78. Файлды жаңарту үшін оны ашу, егер файл бұрыннан бөлса, оның мәліметі жойылады (режимін көрсету керек)
1. r
2. w
3. a
4. r+
5. w+

79. Мәлімет қосу: файлды ашып, оны жаңартады, мәлімет бұрнғының соңына жазылады (режнмін көрсету керек).
1. r
2. w
3. a
4. r+
5. a+

80. Төмендегі функция файлдан сөз тіркесін окиды
1. fread()
2. fgets()
3. fgetc()
4. fputs()
5. fputc()

81. Төмендегі функция файлға сөз тіркесін жазады
1. fwrite()
2. fgets()
3. fgetc()

4. fputs()
5. fputc()

82. printf() функциясының сөз тіркесін шығару кезіндегі форматы қандай?
1. %d
2. %c
3. %s
4. %f
5. %x

83. printf() функциясының нақты сан шығару кезіндегі форматы қандай?
1.%d
2.%c
3.%s
4.%f
5.%x

84. printf() функциясының таңбасыз ондық бүтін сан шығару кезіндегі форматы:
1. %d
2. %u
3. %o
4. %x
5. %f

85. printf() функциясының таңбасыз сегіздік бүтін сан шығару кезіндегі форматы қандай?
1. %d
2. %o
3. %x
4. %c
5. %f

86. printf() функцнясының таңбасыз он алтылық бүтін сан шығару кезіндегі форматы қандай?

1. %d
2. %o
3. %x
4. %c
5. %f

87. printf() функциясының экспоненциал нақты сан шығару кезіндегі форматы қандай?
1. %d
2. %o
3. %e
4. %c
5. %f

88. Егер char мен short типтері араласса, нәтижесі қандай болады?
1. char;
2. **short ;**
3. float;
4. long ;
5. int;

89. Мәліметтердің бір еселік дәлдікпен берілетін нақты санды типі қалай сипатталады?
1. float
2. double
3. char
4. int
5. struct

90. Мәліметтердің екі еселік дәлдікпен берілетін нақты санды типі қалай сипатталады?
1. float
2. double
3. char
4. int
5. struct

91. Мәліметтердің символдық типі қалай сипатталады?
1. float
2. double
3. char
4. int
5. struct

92. Тақырып файлдары нұсқаулары қай жерде орналасады?
1. main() функциясы алдында
2. main() функциясы артында
3. main() функциясы ішінде (тұлғасында)
4. жеке файлда
5. керекті функция шақырылатын блокта

93. < және > символдарымен қоршалып тұратын тақырып файл аттары компиляторға бұл файл қандай каталогта орналасқанын білдіреді?
1. LIB
2. BIN
3. INCLUDE
4. BGI
5. TVISION

94. " және " символдарымен қоршалып тұратын тақырып файл аттары компиляторға бұл файл қандай каталогта орналасқанын біллдіреді?
1. LIB
2. BIN

3. INCLUDE
4. жұмыс істеп тұрған каталогта
5.TVISION

95. С программасы мәтіндері қандай типті файлдарда сақталады?
1. .txt, .doc
2. .cpp, .c
3. .pas, .bas
4. .h, .htm
5. .html, .xtml

96. Программалардың объектілік кодтары қандай типті файлдарда сақталады?
1. .exe
2. .obj
3. .cpp, .c
4. .com
5. .txt

97. Программалардың атқарылатын екілік кодтары қандай типті файлдарда сақталады?
1. .exe
2. .com
3. .obj
4. .cpp, .c
5. .txt

98. Түйінді сөздер (ключевые слова) дегеніміз не?
1. қарапайым идентификаторлар
2. арнайы қорға енгізілген сөздер
3. операциялар
4. функциялар
5. макрокомандалар

99. С тілінің int типті бүтін саны 16 разрядты процессорда қандай орын алады?
1. 2 байт;
2. 4 байт;
3. 8 байт;
4. 6 байт;
5. 10 байт;

100. С тілінің int типті бүтін саны 32 разрядты процессорда қандай орын алады?
1. 2 байт;
2. 4 байт;
3. 8 байт;
4. 6 байт;
5. 10 байт;

Қолданылған терминдердің қысқаша орысша-қазақша сөздігі

	А
Аварийное завершение	Апаттық (қатемен) аяқталу
Альтернативный способ	Баламалы тәсіл
Аргумент заданный по умолчанию	Келісім бойынша берілген аргумент
Арифметика указателей	Нұсқауыштар арифметикасы
Арифметические операции	Арифметикалық амалдар
	Б
База данных	Мәліметтер базасы
Библиотека	Кітапхана
Бинарные деревья	Бинарлы бұтақтар
	В
Ввод	Енгізу
Ввод данных	Мәліметтерді енгізу
Ввод текста	Мәтінді енгізу
Внутреннее представление данных	Мәліметтердің ішкі бейнесі
Восстановить crt режим – *restore crt mode*	crt режиміне қайтып оралу
Время жизни переменной	Айнымалының өмірлік кезеңі
Встроенные функции	Құрамдас функциялар
Вывод	Шығару
Вывод данных	Мәліметтерді шығару
Вывод текста	Мәтінді шығару
Выражение сравнения	салыстыру өрнегі
	Г
Графический режим – *graph mode*	Графикалық режим
	Д
Данные	Мәліметтер
Двоичное число	Екілік сан
Двоичные деревья	Екілік бұтақтар
Двоичный поиск	Екілік түрде іздеу
Диапазон	Аукым; диапазон
Динамическая память	Динамикалық жады
Динамическое распределение памяти	Жадты динамикалық түрде бөлу
Драйвер – *driver*	Драйвер, жетектеуіш программа
	Е
«Заглушка» программы	Программа «жапсырмасы»
Закрыть графический режим – *close graph*	Графикалық режимді жабу (аяқтау)
Заполнить эллипс – *fill ellipse*	Эллипсті толтыру
Зарезервированные слова	Кордағы сөздер
Знаки	Белгілер (таңбалар)
Знаки операций	Операциялар таңбалары (белгілері)
	И
Идентификатор	Идентификатор, атау, аты

Именование переменной	**Айнымалы атауы**
Имя переменной	**Айнымалы аты, атауы**
Инициализация	**Бастапқы мән беру, инициалдау**
Инкапсуляция	**Инкапсуляция; жасыру**
Инстанцирование шаблона - *instantiation*	**Үлгіні инстанциялау**
Интегрированная среда	**Біріктірілген (интегралданған) орта**
Информация	**Ақпарат**
Исключения	**Аластаулар; аластамалар; ерекше жағдайлар**
	К
Классы	Кластар, топтар
Классы памяти	Жады кластары
Ключ	Кілт (кілтипан), түйін
Ключевые понятия	Өзекті ұғымдар (түсініктер)
Ключевые слова	Түйінді (өзекті) сөздер
Колонка таблицы	Кесте бағаны
Комментарий	Түсініктеме, комментарий
Компилятор	Компилятор, түрлендіргіш
Компоновщик	Жинақтауыш
Конкатенация строк	Сөз тіркестерін біріктіру
Константы	Тұрақтылар
Константы с плавающей точкой	Жылжымалы нүктелі константалар (тұрақтылар)
Константы символьные	Символдық константалар, тұрақтылар
Константы символьные	Символдық константалар (тұрақтылар)
Константы строковые	Тіркестік константалар, тұрақтылар
Константы целочисленные	Бүтін константалар, тұрақтылар
Куча – heap	Үйінді
	Л
Логика	Логика, қисын
Логические выражения	Логикалық өрнектер
	М
Мантисса и порядок числа	Санның мантиссасы мен дәрежесі
Массив – array	Жиым, массив
Массив двумерный	Екіөлшемді жиым
Массив динамический	Динамикалық жиым
Метки оператора *label*	Оператор белгілері
Метод	Тәсіл, әдіс
Множество	Жиын
	Н
Наименование данных	Мәліметтер атаулары
Наследование	Мұралау
Начинать графический режим – *initialization graph* (*initgraph*)	Графикалық режимді бастау (инициалдау)
Недоступные ячейки памяти – «Мусор»	Қол жетімсіз (пайдаланылмайтын) жады ұялары – «Қоқыс»
	О

Область имен	Атаулар аймағы
Обработка исключительных ситуаций	Ерекше жағдайларды өңдеу
Образец заполнения – *set fill pattern*	Толтыру үлгісі
Объединение	Біріктіру
Объединения	Біріктірмелер
Объектно-ориентированное программирование	Объектіге бағытталған программалау
Объектный модуль	Объектілік (нысандық) модуль
Окружность, круг – *circle*	Шеңбер, дөңгелек
Оператор исполняемый	Аткарылатын (орындалатын) оператор
Оператор неисполняемый	Аткарылмайтын оператор
Оператор присваивания	Меншіктеу (тағайындау) операторы
Оператор составной	Құрама оператор
Оператор условный	Шартты оператор
Оператор цикла	Цикл операторы
Оператор цикла с постусловием	Соңғы шартты цикл операторы
Оператор цикла с предусловием	Алғы шартты цикл операторы
Операция декремента (--)	Декремент (1-ге кеміту) операциясы (--)
Операция инкремента (++)	Инкремент (1-ге арттыру) операциясы (++)
Операция приведения типов	Типтерді келтіру операциясы
Операция разадресации	Адрессіздендіру операциясы
Опертор выбора	Тандау операторы
Освобождение памяти	Жадты босату
Основание	Негіз
Основание системы счисления	Санау жүйесінің негізі
Отношение	Қатынас

П

Память автоматическая	Автоматты жады
Память статическая	Статикалық жады
распределение памяти	жадты бөлу (үлестіру)
Панель – *bar*	Тақта, панель
Перегрузка функций	Функцияларды асыра жүктеу
Переименование типов – *typedef*	Типтердің аттарын өзгерту
Переключатель	Ауыстырғыш (switch операторы)
Переменная автоматическая	Автоматты айнымалы
Переменная внешняя	Сыртқы айнымалы
Переменная временная	Уақытша айнымалы
Переменная глобальная	Ауқымды (глобальдік) айнымалы
Переменная локальная	Жергілікті (локальдік) айнымалы
Переменная ссылочная	Сілтемелі айнымалы
Переместить к... – *move to*	...-ға (-ге) жылжыту, ығыстыру
Перестановка	Орын ауыстыру (алмастыру)
Перечисления – *enumeration*	Тізбелер, ретімен санау
Поместить пиксель – *put pixel*	Пиксель орналастыру
Поразрядное исключающее ИЛИ (арифметическое ИЛИ)	Разрядтар бойынша аластамалы НЕМЕСЕ (арифметикалық НЕМЕСЕ)
Порт представления – *view port*	Бейнелеу порты

Последовательность	Тізбек
Постепенное заполнение (водой) – *flood fill*	Біртіндеп (сумен) толтыру
Пошаговая детализация программы	Программаны біртіндеп байыптау
Преобразование	Түрлендіру
Преобразование типов	Типтерді түрлендіру
Преобразование типов данных	Мәліметтер типтерін түрлендіру
Приведение типов	Типтерді келтіру
Принципы программирования	Программалау қағидалары
Приоритет операции	Амалдар үстемділігі (приоритеті)
Приоритеты операций	Операциялар приоритеттері (басымдылықтары)
Присваивание	Меншіктеу, тағайындау
Программа	Программа (бағдарлама)
Программа вложенная	Қабаттасқан (кіріктірілген) программа
Программа линейная	Сызықтық программа
Программа разветвленная	Тармақталған программа
Программа циклическая	Циклдік (қайталау) программа
Программирование	Программалау (бағдарламалау)
Программирование восходящая	Төменнен жоғары қарай программалау
Программирование нисходящая	Жоғарыдан төмен қарай программалау
Программное обеспечение	Программалық жабдықтама; Программамен қамтамасыз ету
Программы отладка – *debug*	Программаны жөндеп түзету
Программы фрагмент	Программа үзіндісі (фрагменті)
Процесс программирования	Программалау үрдісі (процесі)
Прямоугольник – *rectangle*	Тіктөртбұрыш
Пунктирная линия – *dashed* (*dotted*) *line*	Пунктир (үзік) сызық
	Р
Разветвление	Тармақталу
Размер типа	Тип мөлшері (көлемі, ені)
Размерность массива	Жиым өлшемі
Разрешение перегрузки – resolution	Асыра жүктеудің рұқсатнамасы
Разыменование	Атаусыздандыру
Режим (способ работы) – *mode*	Режим, жұмыс тәртібі, тәсілі
Рекурсия	Рекурсия, функцияның өзін-өзі шақыруы
	С
Символ константы	Константа (тұрақты) символы
Символ новой строки (\n)	Жаңа жолға көшу символы
Символические	Символдық
Система счисления	Санау жүйесі
Следование	Тізбектелу
Случайные	Кездейсоқ
Сортировка	Сұрыптау, іріктеу, реттеу
Составной оператор	Құрама оператор
Составные типы данных	Мәліметтердің құрамалы типтері
Сравнение строк	Тіркестерді салыстыру

Средства программирования	Программалау құралдары
Ссылка	Сілтеме, сілтеу
Стиль заполнения – *fill style*	Толтыру стилі
Стиль линии – *line style*	Сызық стилі
Столбец матрицы или таблицы	Матрица немесе кесте бағанасы
Строка – *row*	Жол, қатар
Строка – *string*	Тіркес, сөз тіркесі
Строка матрицы или таблицы	Матрица немесе кесте жолы
Строковая переменная	Тіркестік айнымалы
Структура	Құрылым
Структурное программирование	Құрылымдық программалау
Структурный подход	Құрылымдық көзқарас (тәсіл)
	Т
Таблица	Кесте
Твердая линия – *solid line*	Қалың (қатты) сызық
Тело оператора	Оператор тұлғасы (ішкі операторлары)
Термины	Атаунамалар, терминдер
Тип возвращаемого значения	Қайтарылатын мәннің типі
Тип данных	Мәліметтер типтері
Тип данных без знака	Таңбасыз мәліметтер типі
Тип данных основной	Негізгі мәліметтер типі
Тип данных с плавающей точкой	Жылжымалы нүктелі
Тип данных символьный	Символдық мәліметтер типі
Тип данных целочисленный	Бүтін мәліметтер типі
Толстая ширина – *thick width*	Қалың енді
Точка заполнения – *dot fill*	Толтыру нүктесі
Точки в круге – *pies lice*	Дөңгелек ішіндегі нүктелер
	У
Угловые скобки	Бұрыштық жақшалар
Указатели на ссылки	Сілтемелерге нұсқауыштар
Указатель – *pointer*	Нұсқауыш
Указатель на функцию	Функцияға нұсқауыш
Унаследованные функции из языка С	С тілінен мұраланған функциялар
Управляющие последовательности языка С	С тілінің басқару тізбектері
Условный оператор	Шартты оператор
Установить цвет – *set color*	Түсті орнату (тағайындау)
Установка	Орнату (тағайындау)
Установочные параметры	Тағайындау параметрлері
	Ф
Фактический параметр	Нақты параметр
Формальный параметр	Формалды (көрсетпелі) параметр
Форматирование	Пішімдеу, форматтау
Форматированный ввод/вывод	Форматталған енгізу/шығару
Функции вызов	Функцияны шақыру
Функции объявление	Функцияны жариялау
Функции определение	Функцияны анықтау

Функции с переменным числом параметров	Параметрлерінің саны айнымалы функциялар
Функция со многими аргументами	Көп аргументті функция
Функция сравнения	Салыстыру функциясы
Функция, определенная пользователем	Тұтынушы анықтаған функция
	Х
Хаос алгоритмов	Алгоритмдер аласапыраны (хаосы)
	Ц
Целочисленная константа	Бүтін мәнді тұрақты
Цикл	Цикл, қайталау
Цикл вложенный	Қабаттасқан (кіріктірілген) цикл
Цикл с параметром	Параметрлі цикл
	Ч
Числа вещественные	Нақты сандар
Числа с фиксированной точкой	Бекітілген нүктелі сандар
Числа целые	Бүтін сандар
Число восьмеричное	Сегіздік сан
Число конечное	Ақырлы (белгілі) сан
Число ограниченное	Шектеулі сан
Число с плавающей точкой	Жылжымалы нүктелі сан
Число с фиксированной точкой	Бекітілген (тұрақты) нүктелі сан
Число шестнадцатеричное	Он алтылық сан
	Ш
Шаблоны функций	Функция үлгілері (шаблондары)
Шаг цикла	Цикл қадамы, адымы
Шрифт – *font*	Қаріп
Шрифт по умолчанию – *default font*	Келісім (үнсіз) бойынша қаріп
	Э
Электронная почта	Электрондық пошта
Элемент картинки – *picture element* (*pixel*)	Сурет элементі (пиксел)
	Я
Язык программирования	Программалау тілі

МАЗМҰНЫ

КІРІСПЕ 3
1. АЛГОРИТМДЕР ЖӘНЕ ПРОГРАММАЛАР. СЫЗЫҚТЫҚ ЖӘНЕ ТАРМАҚТЫ АЛГОРИТМДЕРДІ ПРОГРАММАЛАУ 5
1.1 Алгоритмдерді жазу жолдары 5
1.2 Алгоритмдерді графикалық түрде жазу 5
1.3 Алгоритмдердің бірыңғай құрылымы 7
1.4 Мәліметтерді енгізу және шығару 8
1.5 С тілінде мәліметтерді форматталған түрде шығару 9
1.6 C++ тілінде мәліметтерді ағындар арқылы шығару 13
1.6 Шартты операторлар 24
1.7 Switch операторы 26
1.8 C/C++ тілінің операторларын қолдану арқылы есептер шығару 29
СЫЗЫҚТЫҚ ЖӘНЕ ТАРМАҚТЫ АЛГОРИТМДЕРГЕ БЕРІЛГЕН ЕСЕПТЕР 32
2.1 Горнер тәсілі бойынша көпмүшелікті есептеу алгоритмі 44
3 ЦИКЛДІК АЛГОРИТМДЕР МЕН ПРОГРАММАЛАР ҚҰРУ 62
3.1 For операторы 63
3.2 While операторы 64
3.3 Do..while операторы 65
3.4 Циклдік алгоритмдер мен программалар құруға берілген есептер 66
4. C/C++ ТІЛДЕРІНДЕГІ ҚОЛДАНУШЫ ФУНКЦИЯЛАРЫ 96
4.1 Функцияларда айнымалыларды пайдалану 97
4.2 Функцияның мән қайтаруы 98
4.3 Нақты және формальды (көрсетпелі) параметрлер 99
4.4 Функциялар мен айнымалылар атауларының әрекет ету аймағы 100
4.5 Функция прототиптері 102
4.6 Рекурсивті функциялар 102
5. ЖИЫМДАР. БІРӨЛШЕМДІ ЖИЫМДАРДЫ ӨҢДЕУ 119
5.1 Жиымды мәндермен толтыру 120
5.2 Жиым мәндерін экранға (файлға) шығару 121
5.3 Жиымдарды өңдеу 123
5.3.1 Жиым элементтерін сұрыптау 128
5.3.3 Жиым элементтерін жою және жиымға жаңа элемент кірістіру 130
5.4 Адрестік операциялар 134
5.4.1 Функциялар арасында байланыс жасау үшін нұсқауыштарды пайдалану 135
5.4.2 Нұсқауыштардың жиымдарға қолданылуы 135
5.4.3 Нұсқауыштарды пайдаланып жиымдармен жұмыс істеу 141
5.4.4 Нұсқауыштарға қолданылатын операциялар 142
6 ЕКІ ӨЛШЕМДІ ЖИЫМДАР 151
6.1 Матрицаның барлық элементтерін өңдейтін алгоритмдер 155
6.2 Екінші типтегі есептер алгоритмдері 158

6.3 Екі өлшемді жиымдармен жұмыс істеу кезінде нұсқауыштарды қолдану 169
7. СӨЗ ТІРКЕСТЕРІН ӨҢДЕУ 181
7.1 Символдық таңбаларды енгізу/шығару 181
7.2 Символдық тіркестер 183
7.3 Символға нұсқауышты пайдалану 184
7. 4 Сөз тіркестерін енгізу функциялары scanf(),gets(str) 188
7.5 Сөз тіркестерін шығару функциялары cprintf(), puts(), cputs() 188
7.6 Сөз тіркестерімен жұмыс істейтін өзге функциялар 190
8 ҚОЛДАНУШЫ АНЫҚТАЙТЫН МӘЛІМЕТТЕР ТИПТЕРІ МЕН ҚҰРЫЛЫМДАРДЫ ПАЙДАЛАНУ 207
8.1 Типтердің атын өзгерту (typedef) 207
8.2 Тізбелер (перечисления – enumeration) 207
8.3 Құрылымдарды пайдалану 209
8.4 Құрылымдарды сипаттау 210
8.5 Құрылым өрістерін пайдалану 211
8.6 Құрылымдар жасау 213
8.7 Құрылым жиымдарын функция аргументі ретінде пайдалану 218
8.8 Біріктірмелер (объединение – union) 220
8.9 Биттік өрістер 223
9 ФАЙЛДАРДЫ ПАЙДАЛАНУ 230
9.1 fprintf және fscanf функцияларын пайдалану 231
9.2 fgets және fputs функцияларын пайдалану 234
9.3 fwrite және fread функцияларын пайдалану 235
10. ГРАФИКАЛЫҚ РЕЖИМДЕ ЖҰМЫС ІСТЕУ 240
10.1. Графикалық режим орнату, одан шығу, мәтін жазу, сызық салу функциялары 243
10.2. Сызық стильдерін беру
Геометриялық фигуралар сызықтарының қалыңдығы мен түр сипатын беру функциясы 247
10.3. Тұйық сызықтар салу 249
Тіктөртбұрыш контурын салу үшін
ПРОГРАММАЛАУ ТІЛДЕРІНІҢ ҚЫСҚАША ДАМУ ТАРИХЫ 262
12 С ПРОГРАММАСЫН ОРЫНДАУ ОРТАСЫ 269
12.1. Турбо С редакторының терезесі 269
12.2 Меню командалары 270
12.3 Қателер коды және олардың мәліметтері 270
12.3.1 Программа орындалу кезінде шығатын қателер 276
12.3.2 Операциялық жүйе анықтайтын қателер 277
12.3.3 Енгізу-шығару қателері 277
12.3.4 Дағдарысты қателер 277
12.3.5 Фаталды қателер 278
ҚОЛДАНЫЛҒАН ӘДЕБИЕТТЕР 281
ИНТЕРНЕТ РЕСУРСТАРЫ А ҚОСЫМШАСЫ 282

Стандартты математикалық функциялар 283
Ә ҚОСЫМШАСЫ 285
Символдық мәліметтер және оларды компьютерде бейнелеу
Б ҚОСЫМШАСЫ 287
Сөз тіркестерінен тұратын мәліметтермен операциялар орындау
В ҚОСЫМШАСЫ Глоссарий 290
C/C++ ТІЛІНДЕ ПРОГРАММАЛАУДАН ТЕСТ СҰРАҚТАРЫ 293
ҚОЛДАНЫЛҒАН ТЕРМИНДЕРДІҢ ҚЫСҚАША ОРЫСША-ҚАЗАҚША СӨЗДІГІ 311

Б. БОРІБАЕВ

С/С++ ТІЛДЕРІНДЕ ПРОГРАММАЛАУ

(практикалық курс)

Жоғары оқу орындарына арналған оқу құралы

Пішімі 60x100 1/16
Тығыздығы 80 гр./м^2. Қағаздың ақтығы 95%.
Қағазы офсеттік. РИЗО басылымы.
Көлемі 320 бет. Шартты баспа табағы 13.75

«Эверо» баспасында басылымға
дайындалды және басып шығарылды
ҚР, Алматы, Байтұрсынұлы к., 22.
тел.: 8 (727) 233 83 89, 233 83 43,
233 80 45, 233 80 42
e-mail: evero08@mail.ru

ЭВЕРО

"Эверо" ЖШС баспаханасында басылып шыгарылды.
ҚР, Алматы, Байтұрсынұлы к., 22.
Тел.: 8 (727) 2 33 80 45, 2 33 80 42, тел./факс 2 33 83 43
E-mail: evero08@mail.ru