



**Коммерциялық емес
акционерлік
қоғам**

**АЛМАТЫ
ЭНЕРГЕТИКА
ЖӘНЕ
БАЙЛАНЫС
УНИВЕРСИТЕТІ**

Ақпараттық жүйелер
кафедрасы

ТІЛДЕР ЖӘНЕ ПРОГРАММАЛАУ ТЕХНОЛОГИЯСЫ

5B060200 – Информатика мамандығының студенттері үшін зертханалық жұмыстарды орындауға арналған әдістемелік нұсқаулықтар

Алматы 2015

ҚҰРАСТЫРУШЫЛАР: А.Г. Ни, С.А. Адилгажинова. Тілдер және программалау технологиясы. 5В060200 – Информатика мамандығының студенттері үшін зертханалық жұмыстарды орындауға арналған әдістемелік нұсқаулықтар. – Алматы: АЭЖБУ, 2015. – 52 б.

Әдістемелік нұсқаулық зертханалық жұмыстарды орындауға арналған нұсқаулықтар мен олардың тапсырмаларынан, зертханалық жұмысқа есеп берудің талаптары мен мазмұнынан, бақылау сұрақтары мен әдебиеттер тізімінен тұрады.

Әдістемелік нұсқаулық 5В060200 – Информатика мамандығының барлық оқу түрінің студенттеріне арналған.

Кесте – 1, ил. – 5, әдебиеттер көрсеткіші – 15 атау.

Рецензент: АЭБ кафедрасының аға оқытушы Калиева С.

«Алматы энергетика және байланыс университеті» коммерциялық емес акционерлік қоғамының 2014 жылғы жоспары бойынша басылады.

© КЕАҚ «Алматы энергетика және байланыс университеті», 2015 ж.

Мазмұны

Кіріспе.....	4
1 Зертханалық жұмыстарды орындаудағы жалпы ұсыныстар.....	5
1.1 Зертханалық жұмысты орындаудағы әдістемелік ұсыныстыр.....	5
1.2 Есеп беруді дайындаудың талаптары.....	6
2 Зертханалық жұмыстарға тапсырмалар нұсқасы.....	7
2.1 Зертханалық жұмыс №1. Сызықты алгоритмдерді программалау.....	7
2.2 Зертханалық жұмыс №2. Тармақталған алгоритмді программалау.....	12
2.3 Зертханалық жұмыс №3. Циклдік алгоритмдерді программалау.....	18
2.4 Зертханалық жұмыс №4. Бір өлшемді жиымдар және нұсқауыштар.....	22
2.5 Зертханалық жұмыс №5. Жиымдар мен нұсқауыштар.....	30
2.6 Зертханалық жұмыс №6. Функцияларды пайдаланып программалау. Рекурсивті функция.....	38
2.7 Зертханалық жұмыс №7. Сөз тіркестері және файлдармен жұмыс.....	44
Қорытынды.....	50
Әдебиеттер тізімі	51

Кіріспе

Ұсынылған «Тілдер және программалау технологиясы» пәні бойынша зертханалық жұмыстарды орындауға арналған әдістемелік нұсқаулық 5B060200 – Информатика мамандығының бірінші курс студенттеріне көмекші құрал ретінде С тілінде программалаудың тәжірибелік қолданысын игеру үшін құрастырылған.

С программалау тілін «жүйелік программалау тілі» деп атайды, өйткені осы тілдің көмегімен компиляторларды және операциялық жүйелерді жазған ыңғайлы. Дегенмен, осы тіл әртүрлі қолдану аймақтарына байланысты үлкен қолданбалы программаларды жазуға да ыңғайлы. С тілі типтердің көп түрін ұсынады. Іргелі типтері символдық, бүтін және нақты (жылжымалы нүктелі сандар) типтердің әртүрлі өлшемдегі бірнеше түрінен тұрады. Сонымен қатар, нұсқауыштың, массивтің, құрылымның және біріктірулердің көмегімен құралатын әртүрлі типтері де бар.

С тілінде жақсы құрылымдық программаны жазу үшін қажетті барлық негізгі басқарылатын конструкциялары бар: операторларды блокқа топтастыру, шарт бойынша шешім қабылдау (if-else), бірнеше мүмкін болатын нұсқалардың біреуін таңдау (switch), шартты басында (while, for) және соңында (do) тексеретін циклдік операторлар, сонымен қоса циклдан мәжбүрлеп шығару операторы (break). С тілі өзінің тиімді және кең ауқымды қосымшалар үшін қолайлы екендігін дәлелдеді.

Тапсырманы орындау кезінде студентке қойылатын негізгі мақсат – блок-сұлбаларды құруға және тәжірибелік программалауға икемделу, С тілінің операторларын тиімді қолдану және де алынған нәтижелерді талдауға үйрену.

Осы әдістемелік нұсқаулықта, кредиттік технология жүйесімен оқытуды ескере отырып, «Тілдер және программалау технологиясы» пәнінің барлық тақырыптарын қамти отырып, жеті зертханалық жұмыс келтірілген.

Зертханалық жұмыстар екі тапсырмадан тұрады: А және В, олар бір-бірінен күрделілігі жағынан айырмашылығы бар. Бірінші тапсырманы таңдағанда студент қорғау кезінде ең үлкен 80% ала алады, екінші тапсырманы қорғаса - 100% ала алады. Студент екінші тапсырманы қорғауға бірінші тапсырманы жоғарғы балға қорғаса ғана жіберіледі. Сонымен қатар, зертханалық жұмыстарды орындау кезінде студент пререквизиттер жиынтығына кіретін пәндерден алған білімдерін де пайдалана ала білуі қажет:

- 1) информатика
- 2) алгебра
- 3) геометрия
- 4) алгоритмдер және деректер құрылымы.

Зертханалық жұмыстардың тапсырмаларын студенттер орындай отырып, алған білімін постреквизит кешеніне кіретін «Нысанды-бағытталған программалау» пәніне қолдана алатындай етіп құрастырылған.

Зертханалық жұмыстарда оның тақырыбы, мақсаты, зертханалық жұмыс тақырыбы бойынша қысқаша теориялық материал, ұқсас есепті шешу мысалы және әрбір студент үшін жеке тапсырмалар нұсқасы келтірілген (әртүрлі 20 нұсқа). Тапсырма нұсқасын оқытушы береді.

Осы жұмыста зертханалық жұмысты орындауға және қорғауға жалпы талаптар құрастырылған.

1 Зертханалық жұмыстарды орындаудағы жалпы ұсыныстар

1.1 Зертханалық жұмыстарды орындаудағы әдістемелік ұсыныстар

Зертханалық жұмысты орындамас бұрын студент онымен толық танысып шығуы керек, берілген теориялық мағлұматтарды оқып, есеп беру қағазын дайындауы керек, оның құрамына сыртқы беті, жұмыс мақсаты, тапсырма нұсқасының мәтіні, сәйкес блок-сұлбасы, программа листингі және бақылау сұрақтарына жауаптары кіреді. Тапсырма нұсқасын оқытушы береді. Жұмысқа дайын емес студенттер, зертханалық жұмысты орындауға жіберілмейді. Студент жұмысты компьютерде орындап болған соң, шыққан нәтижелерді бекітілген кестеге енгізіп, қорытынды есеп беруге жазуы керек. Есеп беруді оқытушы қабылдайды, егер ол дұрыс болмаса, жұмысты қайта толықтыру үшін студентке қайтарылады.

Әрбір зертханалық жұмыс 2 тапсырмадан және келесі жұмыс түрлерінен тұрады:

- тапсырманың талаптарын сараптау және оны шешуді игеру;
- шешімін табу алгоритмін қадаммен қарастыру және оны сипаттау;
- алгоритмнің негізі;
- алгоритмнің блок-сұлбасын жасау;
- кіріс, шығыс және аралық мәліметтерді таңдау және негіздеу;
- алгоритмді кодтау, яғни оның С тілінде жазылуы;
- программаны тексеретін тесттер жинағын таңдау;
- программаны дұрыстау және таңдалынған тесттер жинағында оның дұрыс жұмыс істеуін көрсету.

Алгоритмді құрудың тиімділігін арттыру үшін үлкен программаларды жазған кезде программалаудың құрылымдық түрін қолдануға назар аударған жөн. Бұл программаны құру шығындарын азайтуға және оны ары қарай пайдалануға әсерін тигізеді.

Программалаудың құрылымдық түрі алдымен есептің жалпы түрін, сонан соң есеп бірнеше қарапайым ішкі есептерге бөлінеді.

Мәліметтерді көрсетуді таңдаған кезде студенттің басшылыққа алатын негізгі түсініктері:

- берілген мәліметтердің табиғи ішкі түрі, оларды адам қабылдауы үшін қалыптылығы. Шығатын мәліметтерге де осылар әсер етеді. Құрастырылған программаның пайдасы ең төменгі деңгейге жетеді, егер

жазылған жауапты түсіну үшін адамнан есептің берілгенін түсінікті түрге аударуға байланысты қосымша күрделі жұмыс талап етілсе;

- есепті шешудің алгоритмін тиімді құрастырудың мүмкіндігі. Бұл мүмкіндік есептің берілген және аралық мәліметтерін көрсетуді дұрыс таңдау арқылы іске асатын мүмкіндік. Тиімді алгоритмді құрастыру үшін берілген мәліметтерден басқа да мәліметтер қажет болуы мүмкін.

Программаның орындалу барысында ұсынылатын экран түрі:

- зертханалық жұмыстың аты;
- зертханалық жұмыстың тақырыбы;
- нұсқаның нөмірі;
- берілген мәліметтерді енгізу үшін шақыру;
- берілген мәліметтерді енгізу;
- программа жұмысының нәтижесі;
- программа авторының тобы және аты-жөні.

Программаны құру кезінде программаны тексеріп отыратын тесттер жинағын құруға назар аударған маңызды. Таңдалынған тесттер жинағына негізгі бір талап – ол оның толық түрі, яғни программаның әрбір тармағы қандай да бір тесттер жинағында орындалуы қажет. Осы кезде басты назарды программа көрсететін қателерге, яғни олардың орындалмауын көрсететін диагностикалық хабарламаларына назар аударған жөн.

Сонымен қатар, мысалдардың соңғы нәтижесін қосымша Microsoft Excel программасының көмегімен есептеу қажет. Тестілеудің нәтижелері экранның скриншот түрінде есептің шешімдерінің нәтижесі ретінде көрсетілуі керек.

1.2 Есеп беруді дайындаудың талаптары

Зертханалық жұмысты орындап болғаннан кейін студент, келесі бөлімдерден тұратын есеп беруді дайындайды:

- а) зертханалық жұмыстың №, тақырыбы;
- ә) жұмыстың мақсаты және тапсырма шарттары;
- б) тапсырманы шешудің алгоритмінің сұлбасын дайындау:
 - тапсырманың математикалық моделі;
 - алгоритмнің блок-сұлбасы;
- в) программа мәтіні;
- г) программаның орындалуының нәтижелері;
- ғ) құрылған программаның дұрыстығын тексеру;
- д) қорытынды.

А қосымшасында титул парағының үлгісі көрсетілген.

2 Зертханалық жұмыстарға тапсырмалар нұсқасы

2.1 Зертханалық жұмыс №1. Сызықты алгоритмдерді программалау

Жұмыстың мақсаты: диалогтық режимде Си тіліндегі қарапайым программаларды жөндеп және орындап, компьютерге енгізіп, құруды үйрену, сызықты алгоритмді жүзеге асыратын программаның орындалуы кезінде болатын компилятордың диагностикалық қателерімен танысу.

Жалпы мағлұмат.

Сызықтық алгоритм дегеніміз – берілген мәліметтердің кез келген мәндерінде тізбектелген әрекеттердің бір рет ғана орындалғанда алынатын нәтиже. Программаның операторлары тізбектеліп, программада орналасқан ретімен бірінен кейін бірі орындалады.

Си тіліндегі қарапайым программа тақырыбынан және негізгі функция *main()* денесінен тұрады. Тақырыбында программада қолданылатын функциялардың бас файлдары көрсетіледі. Си тіліндегі программаның негізгі бөлігі программаның *старттық нүктесін* көрсететін *void main ()* операторынан басталады. Си тіліндегі программаның құрылымы келесі түрде көрсетіледі:

```
#include <stdio.h>
#include <math.h>
void main()
{ тұрақтылар мен айнымалыларды сипаттау;
  өрнектер;
  функциялар;
  операторлар;
}
```

Форматты шығару printf() функциясы

Функцияның жазылу синтаксисі:

```
printf(<басқаратын жол>, <айнымалылар тізімі>);
```

мұнда, <айнымалылар тізімі> - экранға шығарылатын айнымалылар идентификаторының тізімі;

<басқаратын жол> - қос тырнақшада қоршалып, аргументтердің қалай бейнеленетінін көрсететін символдар тіркесі: %i, %d – барлық бүтін ондық санды енгізу үшін қолданылатын белгі, %u – таңбасыз ондық санды енгізетін белгі, %f – жылжымалы нүктелі ондық санды енгізу үшін, %c - символды енгізу үшін, %s – сөз тіркесін енгізу үшін.

Форматты енгізу scanf() функциясы.

Жазылу синтаксисі:

```
scanf(<басқаратын жол>, <айнымалы адрестерінің тізімі>);
```

мұнда, <басқаратын жол> - формат тіркесі, спецификация мәндері дәл *printf()* функциясымен бірдей.

<айнымалы адрестерінің тізімі> - функция арқылы енгізілетін айнымалылардың адрестері үтір арқылы көрсетіледі. Айнымалы адресі & символымен көрсетіледі және бұдан әрі айнымалы идентификаторы көрсетіледі, мысалы, *stud* айнымалысының адресі &*stud* түріндегі символмен көрсетіледі.

Си тілінде мәліметтердің 5 негізгі түрі бар: *char* – символдық тип; *int* – бүтін тип; *float* – жылжымалы нүктелі сандар типі; *double* – ұзындығы екі еселенген жылжымалы нүктелі тип; *void* – ешқандай мәні жоқ бос тип.

Тапсырма А. Өрнектің оң жағындағы есептеулерге программа құру қажет. Есептеулердің барлық мәндерін пернетақтадан енгізу керек.

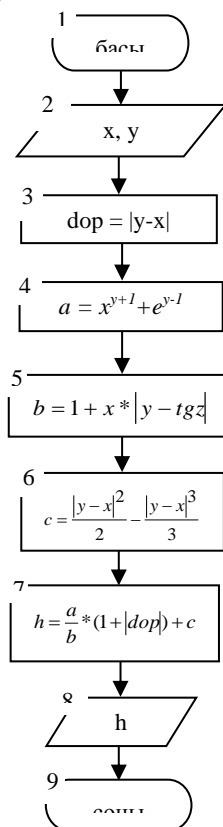
Мысал.

Тапсырма: $h = \frac{x^{y+1} + e^{y-1}}{1 + x * |y - \text{tgz}|} * (1 + |y - x|) + \frac{|y - x|^2}{2} - \frac{|y - x|^3}{3}$ өрнегін есептеу.

Математикалық моделі.

Жаңа айнымалыларды еңгізе отырып, өрнекті қарапайым бірнеше өрнектерге бөлеміз:

$$\begin{aligned} dop &= |y - x|, \\ a &= x^{y+1} + e^{y-1}, \\ b &= 1 + x * |y - \text{tgz}|, \\ c &= \frac{|y - x|^2}{2} - \frac{|y - x|^3}{3}, \\ h &= \frac{a}{b} * (1 + |dop|) + c. \end{aligned}$$



2.1 сурет – Есепті шешудің блок-сұлбасы

Блок-сұлбаны Си тіліне аударамыз.

Программа мәтіні келесі түрде болуы мүмкін:

```
#include <stdio.h> /* бас файлдарды қосу */
#include <math.h>
void main(void)
{ int x,y; /* бүтін типтес айнымалыларды хабарлау */
float dop,a,b,c, rezult; /* нақты типтес айнымалыларды хабарлау */
scanf("%d",&x); /* x айнымалысын енгізу */
scanf("%d",&y); /* y айнымалысын енгізу */
    dop=fabs(y-x);
    a=pow(x,y+1)+exp(y-1);
    b=1+x*fabs(y-tan(z));
    c=0.5*pow(dop,2)-pow(dop,3)/3;
    rezult=a/b*(1+dop)+c;
    printf("%lf", rezult);
    getch( ); /* кез келген батырманы басқанға дейін экранды тоқтату */
}
```

Программаның интерфейсін өзгертіп, қайтадан көрсетейік:

```
#include <stdio.h>
#include <math.h>
void main(void)
{
    int x,y;
    float dop,a,b,c, rezult;
    scanf("%d",&x);
    scanf("%d",&y);
    puts("Зертханалық жұмыс № 1 – Сызықтық программа");
    puts("=====");
    dop=fabs(y-x);
    a=pow(x,y+1)+exp(y-1);
    b=1+x*fabs(y-tan(z));
    c=0.5*pow(dop,2)-pow(dop,3)/3;
    rezult=a/b*(1+dop)+c;
    printf("\a\n Жауабы: rezult=%lf", rezult);
    printf ("Аяқтау үшін <Enter> батырмасын басыңыз ");
    getch( ); /*Кез келген батырманы басқанға дейін тоқтату */
}
#include <stdio.h>
#include <math.h>
int main(void)
{
```

```

int x,y;
float dop,a,b,c, rezult,z;
puts("Зертханалық жұмыс № 1 – Сызықтық программа");
puts("=====");
printf ("енгізіңіз x\npx=");
scanf("%d",&x);
printf ("енгізіңіз y\npy=");
scanf("%d",&y);
printf ("енгізіңіз z\npz=");
scanf("%d",&z);
dop=fabs(y-x);
a=pow(x,y+1)+exp(y-1);
b=1+x*fabs(y-tan(z));
c=0.5*pow(dop,2)-pow(dop,3)/3;
rezult=a/b*(1+dop)+c;
printf("\a\n Жауабы: rezult =%lf", rezult);
printf ("Аяқтау үшін <Enter> батырмасын басыңыз");
getch( ); /* Кез келген батырманы басқанға дейін тоқтату */
return 0;
}

```

Программаның орындалу нәтижесі:

Зертханалық жұмыс № 1 – Сызықтық программа

```

=====
енгізіңіз x
x=2
енгізіңіз y
y=3
енгізіңіз z
z=1
Жауабы: rezult =6.849254

```

А тапсырмасының нұсқасы.

1. $Y = (\pi x + \arctg|x| - \text{sign}(x)) \frac{1}{3}$.
2. $Y = \lg|a^7| + \frac{\pi}{e|a|^{1/3}}$
3. $Y = e^x + \ln|x^{1/4}| + \text{ctg}(x)$
4. $Y = \frac{\lg|x| - |\ln(x)|}{\arctg(x) + 1}$
11. $Y = \frac{bx}{|x|} + \frac{x^5}{\log_2 e^8}$.
12. $Y = (\arctg x^2 + \cos \sqrt{(\sin(x))^{2x}})$.
13. $Y = \sqrt{\frac{ax+b}{cx-d}}$.
14. $Y = (x^{1/3} + \text{sign}(x^{1/5}))^{\sin(x)}$.

$$5. Y = \ln|x^{\cos(x)}| + \operatorname{tg}ax^{\sin(x)}$$

$$6. Y = (a + b - \operatorname{ctg}(x))^{\frac{1}{|x|}}$$

$$7. Y = \frac{3x^2 + 15x^3 + 25x^8}{|x^{10}| + \ln(199)}$$

$$8. Y = \arcsin(\cos x + \sqrt{|\operatorname{tg}(x)|})$$

$$9. Y = \frac{\sin^2(5x+2)}{5,3x+9} + \frac{7x-e^{2x}}{\sqrt{|5x+2|}}$$

$$10. P = \frac{y \sin^2 x}{y^{x-2} + e^{xy}} - \frac{\operatorname{tg}x - 3y}{x-2}$$

$$15. Y = \log_2(x + \cos \sqrt{|\sin(\operatorname{tg}(x))|})$$

$$16. Y = b + e^{\sqrt{|x + \cos(x) - \operatorname{tg}x^3|}}$$

$$17. Y = \sqrt{\operatorname{tg}^2(x) + \lg|\operatorname{ctg}(x)|}$$

$$18. Z = \frac{\sin^2(x+y)}{(x+y)^2} - \frac{\sqrt{|3x-4y|}}{\cos x + e^{5x}}$$

$$19. Y = \frac{7 \cos a}{\sqrt{|5a+2b|}} - \frac{\sin^2(5a+2b)}{e^{\cos a}}$$

$$20. P = \frac{\sin a + \cos b}{x + \operatorname{tg}(ax+b)} - \frac{a^3 + \sin b}{\sqrt{x^2 + 7b^2}}$$

Б тапсырмасы.

Есепті шешудің прогаммасын құру. Есептеулер үшін барлық мәндерді пернетақтадан енгізу.

Б тапсырмасының нұсқасы.

1. Студент программалау сабағы бойынша электрондық сағаттың $h1$ сағат және $min1$ минутында есепті шеше бастады, $h2$ сағат және $min2$ минутында аяқтады. Осы есепті шешуге қанша уақыт кеткенін анықтайтын программа құр (есепті шешуге бір сөткеден артық уақыт кетпеген деп есептейік).

2. a нақты саны берілген. Көбейту амалынан басқа, ешқандай амалды қолданбай: а) a^4 екі операция арқылы; б) a^6 үш операция арқылы; в) a^7 төрт операция арқылы есептеңіз.

3. Екі нөлдік емес сан берілген. Олардың айырмасын, қосындысын, бөліндісін, көбейтіндісін табу керек.

4. Екі сан берілген. Олардың квадраттарының орташа арифметикасын және модульдерінің орташа арифметикасын тап.

5. Тоған судағы қайық жылдамдығы V км/сағ, өзен ағысының жылдамдығы U км/сағ ($U < V$). Қайықтың көлде қозғалу уақыты $T1$ сағ, ал өзенмен қозғалу уақыты (ағысқа қарсы) - $T2$ сағ. Қайықтың жүріп өткен жолы S -ты анықта.

6. Бірінші автокөліктің жылдамдығы $V1$ км/сағ, екінші автокөліктің жылдамдығы - $V2$ км/сағ, екеуінің ара қашықтығы S км. Егер автокөліктер бір бірінен алшақтап кетсе, екеуінің арақашықтығын T сағаттан кейін анықта.

7. Бірінші автокөліктің жылдамдығы $V1$ км/сағ, екінші автокөліктің жылдамдығы - $V2$ км/сағ, екеуінің ара қашықтығы S км. Егер автокөліктер бір біріне қарама-қарсы қозғалса, екеуінің арақашықтығын T сағаттан кейін анықта.

8. Егер a және b катеттерінің ұзындығы белгілі болса, онда тікбұрышты үшбұрыштың ауданы мен периметрін тап.
9. Кубтың қабырғасының ұзындығы берілген. Қырларының ауданы, жоғарғы бөлігінің толық ауданын және кубтың көлемін тап.
10. Радиусы R берілген шеңбердің ұзындығы мен дөңгелектің ауданын табыңыз. Pi -дің мәні ретінде 3.14 қолданылады.
11. Ішкі радиусы $R1$ -ге тең, ал сыртқы радиусы $R2$ -ге тең ($R1 < R2$) сақинаның ауданын тап. Pi -дің мәні ретінде 3.14 пайдалан.
12. Тең қабырғалы үшбұрыштың жақтары берілген. Осы үшбұрыштың ауданы мен сол аймаққа тиісті және тиісті емес дөңгелектің радиусын табыңыз.
13. Шеңбердің ұзындығы берілген. Осы шеңбермен шектелген дөңгелектің ауданын табу керек. Pi -дің мәні ретінде 3.14 қолданылады.
14. Дөңгелектің ауданы берілген. Осы дөңгелекті шектейтін шеңбердің ұзындығын табу керек. Pi -дің мәні ретінде 3.14 қолданылады.
15. Теңбүйірлі трапецияның ауданы мен периметрін табу керек, оның негізі a және b ($a > b$) берілген және негізі үлкен болған кезде $alpha$ бұрышы берілген (бұрышы радианмен берілген).
16. Негізі a және b болатын ($a > b$) және $alpha$ бұрышы берілген (бұрышы радианмен берілген) тікбұрышты трапецияның ауданын және периметрін табу керек.
17. Координаталары $(x1, y1)$ және $(x2, y2)$ берілген екі нүктенің ара қашықтығын табу керек.
18. Кубтың қабырғаларының ұзындығы берілген. Кубтың көлемі мен оның үстіңгі бетінің ауданын табыңыз.
19. Үш кедергі $R1, R2, R3$ параллель біріккен. Бірігудің кедергісін табыңыз.
20. a нақты саны берілген. Көбейту амалынан басқа, ешқандай операция қолданбай: а) a^{10} төрт операция арқылы; б) a^{13} бес операция арқылы; в) a^{64} алты операция арқылы табыңыз.

Бақылау сұрақтары:

- 1) Си тілінің тағайындалуы мен мүмкіндіктері қандай?
- 2) Программаны қалай трансляцияға және орындауға жібереді?
- 3) Программаның басы және соңын қандай операторлармен жазылады?
- 4) Си тіліндегі программа қандай бөлімдерден тұрады?
- 5) Си тіліндегі программа бөлімдері қандай ретпен жазылуы керек?
- 6) Си тілінде экранға шығару операторлары қалай жазылады?
- 7) <Басқаратын жол> қандай мағына білдіреді?
- 8) Мәліметтер типін атап шық.

2.2 Зертханалық жұмыс №2. Тармақталған алгоритмді программалау

Жұмыстың мақсаты: тармақталған алгоритмді жүзеге асыратын блок-сұлба және программа құруды үйрену, шартты өту операторлары мен таңдау операторын дұрыс пайдалануды үйрену.

Жалпы мағлұмат.

Егер есептеу мазмұны бір-бірінен өзгеше болса, және ол бірнеше тармақтан тұрса, онда мұндай алгоритмді тармақталған алгоритм деп атайды. Алгоритмнің оң жақ немесе сол жақ тармағына есептеу үрдісінің шығуы есептің бастапқы берілгендеріне байланысты анықталады.

Си тілінде программаның сызықты емес болу мүмкіндігі бар, яғни операторлары жазылған тәртіп бойынша орындалмауы мүмкін. Осындай мүмкіндікті бізге тармақталған алгоритм береді. Олар келесі үш әдістің біреуімен іске асуы мүмкін:

- өту операторы арқылы;
- шартты оператор арқылы;
- таңдау операторы арқылы.

GOTO өту операторы.

Өту операторы мынандай түрде болады:

GOTO <белгі>;

Ол басқаруды программаның қажетті операторына беруге мүмкіндік береді. Осы оператордың алдында үтірлі нүктемен бөлінген белгі болуы қажет. Си тілінде белгі ретінде бүтін сандар немесе әріптік өрнектер де болуы мүмкін.

Шартты оператор.

IF шартты операторының 2 түрлі жазылу форматы бар:

а) толық түрі

if (шарт) оператор 1 else оператор 2;

б) қысқартылған түрі

if (шарт) оператор 1;

мұнда, *оператор 1, оператор 2* – құрамдас және шартты операторларды қоса алғандағы кез-келген операторлар; *шарт* – жалпы жағдайда логикалық өрнек.

Егер шарт ақиқат болса, онда оператор 1 орындалады, егер шарт жалған болса, онда оператор 2 орындалады.

SWITCH CASE таңдау операторы.

Параметрдің мәніне байланысты белгіленген нұсқалардың бір әрекетін (операторын) таңдау үшін қызмет ететін оператор.

Таңдау операторы келесі түрде болады:

switch (бүтін_сандық_өрнек)

{

case 1_тұрақты_өрнек: оператор 1; break;

case 2_тұрақты_өрнек: оператор 2; break;

... ..

```
case n_тұрақты өрнек: оператор n; break;  
default: оператор;  
}
```

мұнда, *өрнек* – бүтін санды айнымалы немесе арақатынас;
n_тұрақты өрнек – тұрақты өрнек түріндегі белгі немесе тұрақты өрнек;
default – егер өрнек тұрақты өрнектердің ешқайсысымен сәйкес келмесе орындалатын оператор белгісі;
break – switch операторынан шығу үшін қолданылатын оператор.

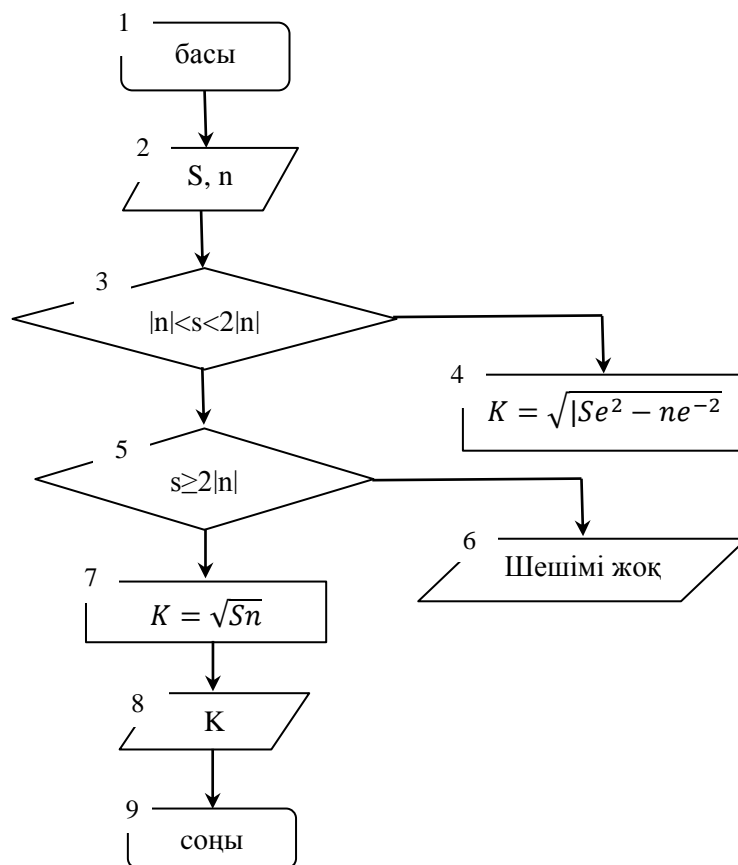
А тапсырмасының нұсқасы.

Өрнектің оң жақ бөлігін орындайтын программа құрыңыз. Есептеуге қажет барлық мәндерді пернетақтадан енгізіңіз.

Мысалы.

Тапсырма: есептеу программасын құру:
$$K = \begin{cases} \sqrt{|Se^2 - ne^{-2}|}, & |n| < s < 2|n| \\ \sqrt{Sn}, & s \geq 2|n|. \end{cases}$$

2.2 - суретте тапсырма шешімінің блок-сұлбасы көрсетілген.



2.2 сурет – Тапсырма шешімінің блок-сұлбасы

Блок-сұлбаны СИ тіліне аударамыз.

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

```

#include<conio.h>
#include<math.h>
void main()
{ int S,n;
float K;
printf("\n Берілгендерді енгізіңіз");
printf("\n S=");
scanf("%d",&S);
printf("\n n=");
scanf("%d",&n);
if (fabs(n)<S && S<2*fabs(n)) K=sqrt(fabs(S*exp(2)-n*exp(-2)));
else
if (S>=2*fabs(n)) K=sqrt(fabs(S*n));
else
printf("\n Шешімі жоқ");
printf("\n Шешімі K=%f",K);
getch();
}
Программаның орындалу нәтижесі.

```

Берілгендерді енгізіңіз.

S=7

n=1

Шешімі K=2.645751.

A тапсырмасының нұсқасы.

$$1. S = \begin{cases} \frac{\sin(n_1 - 2n_2)}{1 + 2n_1}, & \text{егер } p < 2n_2 \leq 1; \\ 5(n_1 - 3\cos(n_2)), & \text{егер } n_1 - 2n_2 > 1. \end{cases}$$

$$2. y = \cos \begin{cases} b^m - ax \sin x, & \text{егер } a^m > bx \cos x; \\ \log \left| \operatorname{tg}^m \left(\cos(ab) - x^b \right)^m \right|, & \text{егер } a \geq b. \end{cases}$$

$$3. S = \begin{cases} \frac{n_1 - 2n_2}{1 + 2n_1}, & \text{егер } p < n_1 - 2n_2 \leq 1; \\ 2(n_1 - 2n_2), & \text{егер } n_1 - 2n_2 > 1. \end{cases}$$

$$4. y = \sin \cos \begin{cases} a^m - bxtgx, & \text{егер } a^m > bxtgx; \\ \log \left| \operatorname{tg}^m \left(a - x^b \right)^m \right|, & \text{егер } a \geq b. \end{cases}$$

$$5. S = \begin{cases} \frac{x - 2y}{1 + 2y}, & \text{егер } 5 < x - 2y \leq 1; \\ 2(x - 2y), & \text{егер } x - 2y > 10. \end{cases}$$

$$6. f(x) = \begin{cases} \sqrt{3x^3 + 4 + \cos x}, & \text{егер } 0 < x < 1; \\ 5 - \sin^2 x, & \text{егер } x \geq 1. \end{cases}$$

$$7. f(x) = \begin{cases} x \operatorname{tg} x - \sin x, & \text{егер } x < 1.5; \\ x^3 + \sin x, & \text{егер } 1.5 \leq x \leq 2.5; \\ 3x^3 + 5, & \text{егер } x \geq 2.5; \end{cases}$$

9.

$$f(x) = \begin{cases} x \cos^2 x + \sin x, & \text{егер } 0 \leq x < 1; \\ \sqrt[3]{x^2 + 6 \sin x}, & \text{егер } 1 \leq x \leq 2; \\ 1.7x^3 + 7, & \text{егер } x > 2; \end{cases}$$

11.

$$f(x) = \begin{cases} x^2 \sin x + \sqrt{x^2 + 1.2}, & \text{егер } x > 0; \\ \operatorname{tg}^2 x + \cos x + 3.5, & \text{егер } 2 < x \leq 6; \end{cases}$$

13.

$$f(x) = \begin{cases} \sqrt[3]{1.7x + \sin x}, & \text{егер } 0 < x \leq 2; \\ \cos \pi x + \operatorname{tg} x, & \text{егер } 2 < x \leq 6; \end{cases}$$

15.

$$f(x) = \begin{cases} x^3 \cos x, & \text{егер } 0 < x < 2; \\ 3x^4 + 7, & \text{егер } x \geq 2; \\ \sqrt{5x^2 + 1.6}, & \text{егер } 5 < x \leq 9; \end{cases}$$

17.

$$f(x) = \begin{cases} \sqrt{3x^3 + 4} + \cos, & \text{егер } 0 < x < 1; \\ 5 - \sin^2 x, & \text{егер } x \geq 1; \end{cases}$$

19.

$$f(x) = \begin{cases} \sqrt{x^2 + |x|}, & \text{егер } -5 < x < 0; \\ 5x^3 + \cos x, & \text{егер } 0 \leq x < 2. \end{cases}$$

$$8. f(x) = \begin{cases} x^3 \cos x, & \text{егер } 0 < x < 2; \\ 3x^4 + 7, & \text{егер } x \geq 2; \\ \sqrt{5x^2 + 1.6}, & \text{егер } 5 < x \leq 9; \end{cases}$$

10.

$$f(x) = \begin{cases} x \cos^2 x + \sin x, & \text{егер } 0 \leq x < 1; \\ \sqrt[3]{x^2 + 6 \sin x}, & \text{егер } 1 \leq x \leq 2; \\ 1.7x^3 + 7, & \text{егер } x > 2; \end{cases}$$

12.

$$f(x) = \begin{cases} \sin \pi x + \sqrt[5]{x^2 + 6}, & \text{егер } 0 < x \leq 4; \\ \ln x + \sqrt{3x + 7}, & \text{егер } x > 4; \end{cases}$$

$$14. f(x) = \begin{cases} \sqrt[3]{|x| + x^4}, & \text{егер } -1 < x \leq 1; \\ \operatorname{tg}^2 3x + \ln 5x, & \text{егер } 1 < x \leq 4; \end{cases}$$

16.

$$f(x) = \begin{cases} x \operatorname{tg} x - \sin x, & \text{егер } x < 1.5; \\ x^3 + \sin x, & \text{егер } 1.5 \leq x < 2.5; \\ 3x^3 + 5, & \text{егер } x \geq 2.5; \end{cases}$$

$$18. f(x) = \begin{cases} \sqrt{x^2 + |x|}, & \text{егер } -5 < x < 0; \\ 5x^3 + \cos x, & \text{егер } 0 \leq x < 2; \end{cases}$$

20.

$$f(x) = \begin{cases} x^2 \sin x + \sqrt{x^2 + 1.2}, & \text{егер } x > 0; \\ \operatorname{tg}^2 x + \cos x + 3.5, & \text{егер } 2 < x \leq 6; \end{cases}$$

Б тапсырмасы.

Есепті шешудің программасын құрыңыз. Есептеулер үшін барлық мәндерді пернетақтадан енгізіңіз.

Б тапсырмасының нұсқасы.

1. x , y , z нақты оң сандары берілген. Жақтарының ұзындықтары x , y , z болатын үшбұрыштың бар екендігін анықтаңыз.

2. x , y нақты сандары берілген. Егер x , y теріс болса, онда оларды оның модулімен ауыстырыңыз; егер де олардың біреуі ғана теріс болса, онда екеуінің мәнін 0.5-ке арттыру керек; егер екі мәні де теріс болмаса және олардың біреуі

[0.5, 2.0] аралығына кірмесе, онда екі мәнді де 10 есе кішірейту керек; басқа жағдайларда x , y -ті өзгеріссіз қалдыру қажет.

3. $M(x,y)$ нүктесі орналасқан квадраттың нөмірін анықтап, оны баспаға шығарыңыз, мұндағы x , y – берілген нақты сандар.

4. x -тің берілген мәнінде $a=\sin x$, $b=\cos x$, $c=\ln|x|$ өрнектерінің мәндерінің арасынан ең кішісін тауып, оны экранға шығару қажет.

5. Екі санның қосындысын табатын программа жазыңыз, егер қосынды 100-ден үлкен болса, онда шыққан шешімді 2-ге көбейтіп, шарт орындалмаса, онда шешімді 3-ке көбейтіңіз.

6. Берілген екі $M1(x1,y1)$ және $M2(x2,y2)$ нүктелерінің қайсысы координата басына жақын орналасқанын анықтаңыз. Экранға осы екі нүктенің координаталарын шығарыңыз.

7. Пернетақтадан енгізілген теріс сандардың тізбегінің арасынан ең кішісін анықтайтын программаны жазыңыз.

8. a , b , c , x , y нақты сандары берілген. x және y жақтары бар тіктөртбұрышты ойыққа a , b , c қабырғалары бар кірпіш өте ме соны анықтаңыз. Кірпіштің әрбір қабырғасы ойықтың әрбір жақтарына параллель немесе перпендикуляр болған жағдайда ғана кірпішті ойыққа кіргізуге болады.

9. Пернетақтадан енгізілген теріс сандардың тізбегінің арасынан ең үлкенін анықтайтын программаны жазыңыз.

10. Енгізілген сандар қатарынан «нуль»-ді және оның санын анықтайтын программаны жазыңыз.

11. Сөткенің уақытын анықтайтын программаны жазыңыз. Егер сағат 12^{00} – ге дейін – «таң», 18^{00} – ге дейін – «күн», 0^{00} – ге дейін – «кеш», 6^{00} –ге дейін – «түн» деген сәйкес хабарламаларын көрсетіп тұрсын.

12. Жазықтықта екі нүкте өзінің координатасымен берілген, олар полярлық немесе декарттық болуы мүмкін. Осы екі нүктенің арасындағы қашықтықты есептейтін программа құрыңыз.

13. Желі бойынша ақпаратты беру жылдамдығы 14 кбит/сек және 58 кбит/сек диапазон аралығында өзгереді. Орташа жылдамдық пен қазіргі жылдамдықтың ең үлкен айырмашылығын анықтайтын программа жазыңыз. Программада сәйкес хабарламалары да болуы керек.

14. R радиусы бар шар ромб түріндегі ойыққа жақтары P және Q бұрышымен өте ала ма? Соны анықтайтын программа жазыңыз.

15. Келесі шарттарды қанағаттандыратын болса, *True* немесе *False* хабарламасын экранға шығаратын программа жазыңыз:

- берілген үш орынды санның квадраты оның цифрларының қосындысының кубына тең болса;

- берілген үш орынды санның арасында бірдей цифрлар болса.

16. 13-тен басқа бүтін сандарды сұрайтын және оның мәнін баспаға шығаратын программаны жазыңыз. Егер берілген сан 13 болса, онда оның орнына 77 саны басылып шықсын.

17. Екі фигураның (шеңбер немесе квадрат) қайсысының ауданы үлкен екенін анықтаңыз. Квадраттың қабырғалары a -ға тең, шеңбердің радиусы r -ге тең екені белгілі. Экранға ауданы үлкен фигураның мәні мен атын шығарыңыз.

18. a , b , c үш кесіндінің ұзындығы берілген. Егер осы үш кесінді арқылы үшбұрыш құруға болса, онда оның периметрі мен ауданын есептеңіз.

19. радиусы r , ортасы (x_0, y_0) нүктесі болатын шеңберге $M(x, y)$ нүктесі кіретінін анықтау керек.

20. Берілген төрт кесінді арқылы параллелограмм құруға бола ма, соны анықтаңыз. Координаттары (x, y) болатын нүкте штрихталған аймаққа жата ма соны анықтайтын программаны жазыңыз.

Бақылау сұрақтары:

- 1) Шартты оператордың қандай түрлерін білесіз?
- 2) Таңдау операторы қалай орындалады?
- 3) Тармақталған программалау үшін қандай оператор қолданылады?
- 4) Өту операторы қалай орындалады?
- 5) Өту операторының толық формасы қалай жазылады?
- 6) Өту операторының қысқартылған формасы қалай жазылады?
- 7) *Break* операторы не үшін қолданылады?
- 8) *Default* операторы не үшін қолданылады?

2.3 Зертханалық жұмыс №3. Циклдік алгоритмдерді программалау.

Жұмыстың мақсаты: циклдік алгоритмдерді жүзеге асыратын программаларды және блок-сұлбаларды құрып үйрену, циклдік операторлар *for*, *while*, *do – while* – мен жұмыс жасап үйрену.

Жалпы мағлұмат.

Циклдік алгоритм дегеніміз – аралық мәндері әртүрлі болған жағдайда бір оператордың көп рет орындалуы. Осы операторлардың қайталану саны нақты (қайталану саны алдын-ала белгілі цикл) немесе нақты емес (қайталану саны алдын-ала белгісіз цикл) формада беріледі.

Жұмысты орындаудың алдында циклды ұйымдастыратын әртүрлі сұлбаларды және параметрлі циклдік оператор, шартты алдын-ала тексеретін циклдік оператор, шартты соңынан тексеретін циклдік операторды білу қажет.

Параметрлерінің мәні бірнеше рет өзгерген кезде көп рет орындалатын іс-әрекеттердің тізбегін цикл деп атайды.

Циклдың түрлері:

- шартты алдын-ала тексеретін;
- шартты соңынан тексеретін;
- параметрлі.

С тілінің циклдік операторлары.

Шартты алдын-ала тексеретін циклдік оператор:

```
while (шарт)
{
    //цикл денесінің операторлары
}
```

Егер циклда бір ғана оператор болса, онда операторлық жақшаларды жазбауға да болады.

Шартты алдын-ала тексеретін цикл келесі түрде жұмыс жасайды: алдымен логикалық шарт тексеріледі, егер ол ақиқат болса, онда цикл денесінің операторлары орындалады, егер шарт жалған болса, онда циклдан кейінгі операторлар орындалады.

Шартты соңынан тексеретін циклдік оператор:

```
do {
    //цикл денесінің операторлары
} while (шарт);
```

Мұнда, егер де циклда бір ғана оператор болса, онда операторлық жақшаларды жазбауға болады. Алдымен цикл денесінің операторлары орындалады, сонан соң логикалық шарт тексеріледі, егер шарт жалған болса, онда цикл денесінің операторлары қайта орындалады, ал егер шарт ақиқат болса, онда циклдан кейінгі операторлар орындалады.

Параметрлі циклдік оператор:

```
for (цикл басы; цикл соңының шарты; цикл қадамының өзгеруі)
{
    //цикл денесінің операторлары
}
```

Параметрлі циклдік оператор келесі түрде жұмыс істейді: басқарушы айнымалының (j – циклдың санауышы) мәні әзірге қандай да бір n -нің мәнінен кіші (үлкен) болса, онда циклдың операторлары орындалады, сол кезде циклдың әрбір айналымында басқарушы айнымалы өзінің мәнін циклдың қадамы деп аталатын қандай да бір анықталған санға өзгертеді.

Ішкі циклдар. Басқа операторлар сияқты цикл да басқа циклдардың ішінде болуы мүмкін, бұл жағдайда, әзірге ішкі операторлар орындалмайынша, сыртқы операторлар орындалмайды.

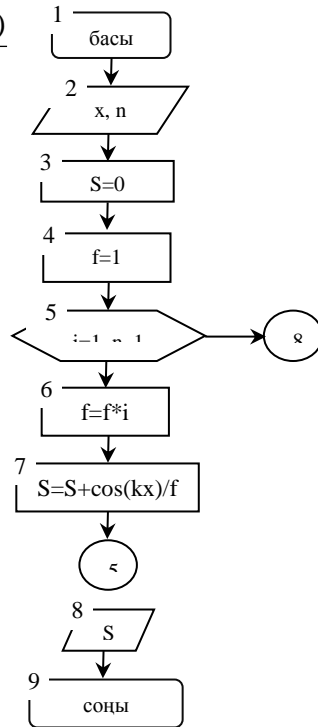
Циклды басқаратын операторлар.

break операторы. Компилятор *break* операторын кездестірген сәттен бастап ол өзінің қалған операторлары мен циклдың айналу санына қарамастан өзінің орындалуын аяқтайды да, циклдан кейін тұрған операторларды орындап бастайды.

continue операторы. Компилятор *continue* операторын кездестірген сәттен бастап, цикл айналымының соңына дейін қанша оператор қалғанына қарамастан берілген цикл айналымын орындауды аяқтайды және бірден циклдың келесі айналымына өтеді.

Мысалы. Келесі қосындыны есептейтін программаны құрыңыз:

$$S = \sum_{k=1}^n \frac{\cos(kx)}{k!}$$



2.3 сурет – Есептің шешімінің блок-сұлбасы

Программа мәтіні келесі түрде болады:

```

#include <stdio.h>
#include <math.h>
int main()
{int n, x, f, i;
float S;
printf ("\n n-нің мәнін енгізіңіз\n n=");
scanf ("%d", &n);
printf ("\n x-тің мәнін енгізіңіз \n x=");
scanf ("%d", &x);
S=0;
f=1;
for (i=1; i<=n; i++)
{f=f*i;
S=S+(cos(i*x)/f);}
printf ("\n Қосындысы S=%f", S);
getch();
return 0;
}

```

Программа жұмысының нәтижесі.

n-нің мәнін енгізіңіз

n=4

x-тің мәнін енгізіңіз

$x=1$

Қосындысы $S=0.139995$

А тапсырмасының нұсқасы.

Блок-сұлбасын құрып және қосындыны есептейтін программаны жазыңыз

$$1. S = \sum_{k=0}^n \frac{\ln^k 3}{k!} x^k$$

$$2. S = \sum_{k=0}^n (-1)^k \frac{x^{2k+1}}{(2k+1)!}$$

$$3. S = \sum_{k=0}^n (-1)^k \frac{x^{2k}}{(2k)!}$$

$$4. S = \sum_{k=0}^n \frac{x^{4k+1}}{(4k+1)!}$$

$$5. P = \prod_{k=1}^n \left(\frac{k}{k+1} - \cos^k |x| \right)$$

$$6. P = \prod_{k=1}^n \frac{(1-x)^{k+1} + 1}{((k-1)!+1)^2}$$

$$7. P = \prod_{k=1}^n \frac{1+(x+1)^k}{((k-1)!+3)^2}$$

$$8. S = \sum_{i=1}^n \frac{1 + \sqrt{|x|} + x^3}{i!}$$

$$9. S = \sum_{i=1}^n \left(\frac{1}{i!} + \sqrt{|x|} \right)$$

$$10. S = \sum_{k=1}^n \frac{(-1)^k x^k}{k!}$$

$$11. S = \sum_{k=1}^n \frac{\sin(kx)}{k!}$$

$$12. S = \sum_{k=0}^n \frac{x^k x}{k!}$$

$$13. S = \sum_{k=1}^n \frac{(-1)^{k+1}}{(2k+1)!} * \left(\frac{x}{3} \right)^{4k+3}$$

$$14. S = \sum_{k=1}^n \frac{\sqrt{|x^3|}}{k!^3}$$

$$15. S = \sum_{k=1}^n \frac{(-1)^k (k+1)x^k}{3^k}$$

$$16. S = \sum_{k=1}^n \frac{x}{k! \sqrt{|x|} + 1}$$

$$17. S = \sum_{k=1}^n \frac{1}{\sqrt{|x|} + k!^2}$$

$$18. S = \sum_{k=1}^n \frac{(-1)^k x^k}{(k+1)!^2}$$

$$19. S = \sum_{k=1}^n \frac{(-1)^k (k+1)x^k}{3^k}$$

$$20. S = \sum_{k=1}^n (-1)^{k+1} \frac{\sin(kx)}{k!}$$

Б тапсырмасы.

Есептің шешімінің программасын құрыңыз. Есептеулер үшін барлық мәндерді пернетақтадан енгізіңіз.

Б тапсырмасының нұсқасы.

1. 100 теңгеге 100 бас мал алу керек болса, егер жылқы 10 теңге, сиыр 5 теңге, бұзау 0,5 теңге тұратын болса, қанша жылқы, сиыр, бұзау алуға болады? (Жауабы: сиыр – 9, жылқы – 1, бұзау – 90).

2. Компьютерге N нүктенің өлшемдері кезекпен енгізіледі. Ортасы (a, b) нүктесінде болатын, радиусы R шеңберге нүктелердің қаншасы түсетінін анықтаңыз.

3. Қояндар мен үйректердің барлығын қосқанда 24 аяқ бар. Қанша қоян және үйректер болуы мүмкін (барлық жағдайды көрсетіңіз)?

4. Компьютерге топтың N оқушысының бойы туралы мәліметтер енгізіледі. Топтағы оқушылардың орташа бойын анықтаңыз.

5. Бір клеткалы амеба әрбір 3 сағат сайын 2 клеткаға бөлінеді. 3, 6, 9, 12, ... , 24 сағаттан кейін қанша амеба болатынын есептеңіз (Жауабы: 256).

6. A санының дәрежесі натурал N саны болғандағы мәнін есептейтін программаны құрыңыз (программаны 3 түрлі циклды пайдаланып құрыңыз: *for, while, do...while*).

7. Енгізілген бүтін санды мәндердің кез келгенінің қосындысын есептейтін программаны жазыңыз. Егер 999 саны енгізілген болса, онда экранға қосындының нәтижесін шығарыңыз.

8. *while* циклдік операторының көмегімен кез келген 10 санның көбейтіндісін есептеп, экранға шығаратын программаны құрыңыз.

9. Спортшы шынығудың алғашқы күні 10 км жүгіріп өтті. Күн сайын алдыңғы күнгі норманы 10 пайызға өсіріп отырды. 7 күннің ішінде спортшы барлығы қанша шақырым жүгіріп өтті?

10. *while* циклдік операторын пайдаланып, 22-ден 100-ге дейінгі жұп сандарды экранға шығаратын программаны жазыңыз.

11. $[a,b]$ аралығындағы h қадамымен $F(x)$ функциясының мәндерін есептейтін программа жазыңыз. Нәтижесін кесте түрінде көрсетіңіз, бірінші бағанда – аргументтің мәні, екінші бағанда – сәйкес функция мәндерін көрсетіңіз. $F(x)=\cos x + \operatorname{ctg} x$.

12. *while* циклдік операторының көмегімен 1-ден бастап өзіңіз енгізген санға дейінгі бүтін сандардың квадраттарының қосындысын есептейтін программаны жазыңыз.

13. 1-ден 99-ға дейінгі сандардың ішінен 3-ке бөлінетін тақ сандардың қосындысын есептейтін программаны құрыңыз.

14. 133-ке бөлгенде қалдығы 125 болатын, ал 134-ке бөлгенде 111 қалдық қалатындай төрт орынды сандарды табатын программаны құрыңыз.

15. Шартты соңында тексеретін циклдік оператордың көмегімен кез келген символды енгізіп, бірақ орыс алфавитіне ғана комментарий жасайтын программаны жазыңыз. «*Я*» әріпін басқанда программа жұмысын аяқтауы қажет.

16. Екі орынды сандардың ішінен, цифрлар санының қосындысы 2, 3, 4, 5, 6, 7, 8, 9-ға көбейткенде өзгермейтін сандарды табу программасын жазыңыз.

17. Үш орынды сандардың ішінен, цифрлар санының қосындысы берілген бүтін санға тең болатын сандарды анықтайтын программа жазыңыз.

18. Үш орынды сандардың ішінен, ортасындағы саны алғашқы және соңғы санының қосындысына тең болатын санды анықтайтын программа жазыңыз.

19. Үш орынды сандардың ішінен, алғашқы екі санының квадраты мен үшінші санның квадратының айырмасына тең санды анықтайтын программа жазыңыз.

20. Екі орынды сандардың ішінен цифрларының квадраттарының қосындысы 17-ге бөлінетін сандарды анықтайтын программа жазыңыз.

Бақылау сұрақтары:

- 1) *FOR* операторы қалай жазылады және қалай жұмыс істейді?
- 2) *FOR* операторы қандай циклдарды құрастыру үшін қажет?
- 3) *WHILE* операторының *DO - WHILE* операторынан айырмашылығы неде?
- 4) Циклдік қайталану саны белгілі циклдік алгоритмдер қалай программаланады?
- 5) Циклдік қайталану саны белгісіз циклдік алгоритмдер қалай программаланады?
- 6) Бірде-бір рет орындалмайтын циклдік операторды жазыңыз.
- 7) Операторлық жақшалар қандай жағдайларда қолданылады?
- 8) Циклды басқару операторлары қалай жұмыс істейді?

2.4 Зертханалық жұмыс №4. Бір өлшемді жиымдар және нұсқауыштар

Жұмыстың мақсаты: есеп шығару кезінде жиым түріндегі мәліметтер құрылымын пайдаланып программалауды игеру, бір өлшемді жиымдарды дұрыс сипаттап үйрену, бір өлшемді жиымдарды өңдейтін алгоритмдерді сызба түрінде көрсету, яғни блок-сұлбасын құрып үйрену, нұсқауыш түріндегі айнымалыларды қолданып программалауды игеру.

Жалпы мағлұмат.

Жиымдар – бірдей атауларға ие, жады ұяшығындағы тізбектелген топ. Жиымның жеке элементтеріне қол жеткізу жиым атауы мен оның реттік нөмірі (индексі) арқылы ұйымдастырылады. Индекс элементтің орнын жиымның басынан бастап анықтайды.

Жиымды сипаттау кезінде мыналарды көрсету қажет:

- элемент типін;
- жиым атауын;
- жиым өлшемін.

Жиымды сипаттаудың жалпы түрі мынадай:

типі жиым_атауы [өлшем 1][өлшем 2]....;

Бір өлшемді жиымдарды сипаттау кезінде тік жақшада жиым элементтерінің санын білдіретін бір ғана индекс көрсетіледі. Мысалы:

float A[5], B[25];

Сипаттау кезінде жиым элементтерін берілген мәндермен инициалдауға болады. Мысалы:

int D[5]={23, 45, 32, 12, 88};

Си тілінде жиым элементтері нөлден бастап нөмірленеді, яғни бірінші элементтің индексі 0-тең, ал соңғы элементтің индексі – жиым өлшемінен бірге кем болады.

Жиым мен нұсқауыштың арасындағы байланыс. Си тілінде жиымдар мен нұсқауыш бір-бірімен тығыз байланысқан.

Нұсқауыш – басқа айнымалының адресі немесе жады бөлігі сақталынатын айнымалы.

Нұсқауыш келесі түрде жазылады:

типі **<айнымалы атауы>*;

Нұсқауыштар айнымалылар тізімінде хабарланады, бірақ олардың атының алдына * белгісі қойылады. Нұсқауыш өзі үшін хабарланған типтегі айнымалыға нұсқайды.

Кейбір айнымалыларға қолданылатын & *унарлы операциясы* бізге осы айнымалының мәні емес, адресі керек екенін көрсетеді. Егер *uk* айнымалысы нұсқауыш ретінде хабарланса, онда меншіктеу операторы *uk=&x* мынаны білдіреді: «*x* айнымалысының адресін алып және оның мәнін айнымалы-нұсқауыш *uk*-ға меншіктеу».

Нұсқауышқа қолданылатын * *унарлы операциясы* нұсқауыш сілтеме жасайтын жады ұяшығына қол жеткізуді қамтамасыз етеді. Басқаша айтқанда, адресі арқылы қатынас жасау үшін қолданылатын операция (көрсетілген адрестегі мәнді пайдалану). Мысалы, **uk*-ны сөзбен былай сипаттауға болады: «*uk* нұсқайтын адрестегі бар мәлімет».

Жиым элементінің кез келгеніне қол жеткізу нұсқауыш арқылы орындалуы мүмкін. Егер *uk* – *int* типінде сипатталған *int *uk* нұсқауышы болса, онда *uk=&a[0]* операциясы орындалғаннан кейін *uk – a[0]* адресіне, ал *uk+i* жиымның *i*-нші элементіне нұсқайды. Осылайша, *uk+i – a[i]* -дің адресі болып табылады. Программада жиым аты оның бірінші элементінің адресіне тең болса, онда *uk=&a[0]* өрнегі мынаған тең: *uk=a*. Сондықтан, *a[i]*-дің мәнін былайша жазуға болады *(*a+i*). Жиымның осы екі элементіне адресті алу операциясын қолданып, *&a[i]* және *a+i* бірдей (тепе-тең) екенін көреміз.

Символдар жолын кері ретпен баспаға шығаратын программаның мысалын қарасырайық.

```
#include <stdio.h>
main()
{
char *uk1,*uk2;
uk1=uk2="Тілдер және программалау технологиясы";
while(*uk2!='\0')
putchar(*uk2++);
putchar('\n');
while(--uk2 >= uk1)
putchar(*uk2);
putchar('\n');
}
```

Ең басында *uk1* және *uk2* нұсқауыштарына "Тілдер және программалау технологиясы" жолының бастапқы адресі меншіктеледі. Содан кейін, жол әр символы бойынша басылады және бір уақытта *uk2* нұсқауышы жол бойымен жылжиды. Шығарудың соңында *uk2* берілген мәтіннің соңғы символын көрсетеді. Екінші *while* циклында сол *uk2* нұсқауышы кері бағытта өзгере

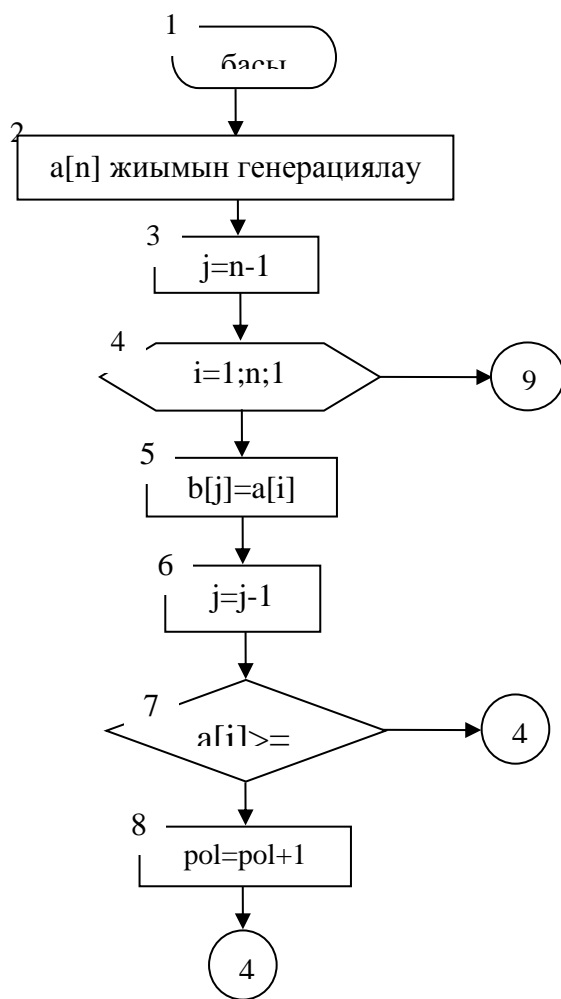
бастайды, жолды кері ретпен беру үшін, жиымның нөлдік элементіне дейін кему арқылы өзгеріп отырады.

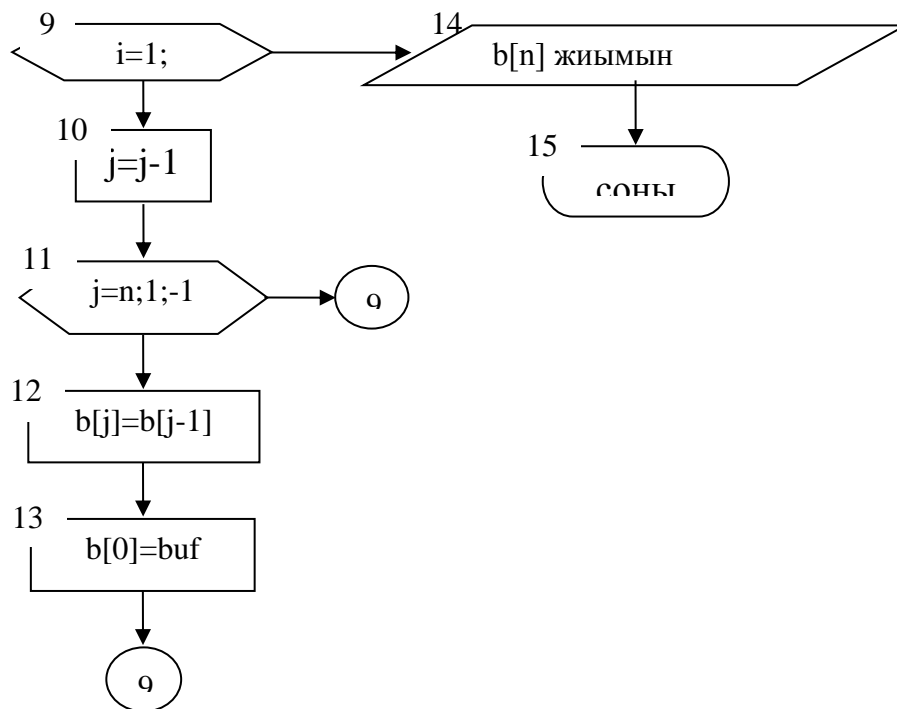
А тапсырмасы.

Бір өлшемді жиымдарды өңдейтін программалар және блок-сұлбасын құру. Жиым элементтерін кездейсоқ сандар генераторының көмегімен толтыру керек.

Мысалы. $A[n]$ жиымы берілген. $B[n]$ жиымын $A[n]$ жиымының элементтерімен соңынан бастап толтыру (кері бағытта) және K позициясына оңға қарай циклдік жылжыту жасау керек, K – оң элементтерінің саны.

2.4 суретінде есептің шешімінің блок-сұлбасы келтірілген.





2.4 сурет – Тапсырма шешімінің блок-сұлбасы

Программа келесі түрде болады:

```

#include <stdlib.h>
# include <conio.h>
# include <stdio.h>
#define n 10
int main ()
{ int buf,j,pol,i,b[n], a[n];
  for (i = 0; i < n; i++)
    a[i]=rand()% 10-5;
  pol=0;
  printf ("\n \n Генерацияланған жиым a[n] \n");
  for (i=0;i< n;i++)
    printf ("%d ",a[i]);
  j=n-1;
  for (i=0;i< n;i++)
    { b[j]=a[i];
      j=j-1;
      if (a[i]>=0)pol++;}
  printf ("\n Оң элементтерінің саны \n pol=%d",pol);
  printf ("\n \n b[n] жиымын шығару \n");
  for (i=0;i< n;i++)
    printf ("%d ",b[i]);
  for (i=0;i<pol;i++)
    { buf=b[n-1];
      for(j=n-1;j>0;j--)
  
```

```

        b[j]=b[j-1];
        b[0]=buf;  }
printf ("\n %d позициясына b[n] жиым элементтерін жылжыту \n", pol);
for (i=0;i< n;i++)
printf ("%d ",b[i]);
printf ("\n");
system("PAUSE");
return 0;}

```

Программа жұмысының нәтижесі.

```

Генерацияланған жиым a[n]
1 2 3 -4 -5 -1 -2 -3 -7 9
Оң элементтерінің саны
pol=4
b[n] жиымын шығару
9 -7 -3 -2 -1 -5 -4 3 2 1
4 позицияға b[n] жиым элементтерін жылжыту
-4 3 2 1 9 -7 -3 -2 -1 -5

```

A тапсырмасының нұсқасы.

Тапсырманы орындау кезінде жиым элементтерін кездейсоқ сандар генераторының көмегімен толтыру керек.

1 $A[N]$ жиымы берілген. $B[N]$ жиымын $A[N]$ жиымының элементтерімен келесі түрде: алдымен индексі жұп элементтерімен, содан соң индексі тақ элементтерімен толтыру қажет. Оңға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – оң элементтерінің саны.

2 $A[N]$ жиымы берілген. $B[N]$ жиымын $A[N]$ жиымының элементтерімен келесі түрде: алдымен индексі жұп элементтерімен, содан соң индексі тақ элементтерімен толтыру қажет. Оңға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – теріс элементтерінің саны.

3 $A[N]$ жиымы берілген. $B[N]$ жиымын $A[N]$ жиымының элементтерімен мына қос теңсіздікті қанағаттандыратындай етіп толтырыңыз: $A[1] < A[i] < A[10]$. $B[N]$ жиымының толтырылмаған элементтерін $A[N]$ жиымынан қалған элементтермен толтыру қажет. Солға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – $A[N]$ жиымынан қалған элементтердің саны.

4 $A[N]$ жиымы берілген. $B[N]$ жиымын $A[N]$ жиымының элементтерімен мына қос теңсіздікті қанағаттандыратындай етіп толтырыңыз: $A[1] < A[i]$ немесе $A[i] < A[10]$. $B[N]$ жиымының толтырылмаған элементтерін $A[N]$ жиымынан қалған элементтермен толтыру қажет. Оңға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – $A[N]$ жиымынан қалған элементтердің саны.

5 N өлшемді бүтінсанды жиым берілген. Оның жұп сандарына бірінші элементті қосу арқылы түрлендіріңіз. Жиымның бірінші элементін өзгеріссіз

қалдырыңыз. Оңға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – тақ элементтерінің саны.

6 N өлшемді бүтінсанды жиым берілген. Оның тақ сандарына бірінші элементті қосу арқылы түрлендіріңіз. Жиымның бірінші элементін өзгеріссіз қалдырыңыз. Оңға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – жұп элементтерінің саны.

7 N өлшемді бүтінсанды жиым берілген. Оның жұп сандарына соңғы элементті қосу арқылы түрлендіріңіз. Жиымның соңғы элементін өзгеріссіз қалдырыңыз. Солға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – тақ элементтерінің саны.

8 N өлшемді бүтінсанды жиым берілген. Оның жұп сандарына бірінші элементті қосу арқылы түрлендіріңіз. Жиымның бірінші элементін өзгеріссіз қалдырыңыз. Оңға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – тақ элементтерінің саны.

9 10 элементтен тұратын жиымның ең кіші элементімен ең үлкен элементінің орындарын ауыстырыңыз. Оңға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – ең үлкен және ең кіші элементтерінің арасында орналасқан элементтер саны.

10 $A[N]$ жиымы берілген. Оң элементтердің барлығын ең кіші элементке азайту керек. Оңға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – оң элементтерінің саны.

11 $A[N]$ жиымы берілген. Теріс элементтердің барлығын ең үлкен элементке азайту керек. Солға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – теріс элементтерінің саны.

12 10 элементтен тұратын жиым берілген. Ең кіші элементімен ең үлкен элементінің арасында орналасқан жиым элементтерін кері бағытта орындарын ауыстыру керек. Солға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – ең кіші элементімен ең үлкен элементінің арасында орналасқан элементтер саны.

13 N өлшемді жиым берілген. Солға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – ең кіші элементтің индексі.

14 N өлшемді жиым берілген. Оңға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – ең үлкен элементтің индексі.

15 $A[N]$ жиымы берілген. Оңға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – $A[N]$ жиымының оң элементтерінің орташа арифметикасының бүтін бөлігі.

16 $A[N]$ жиымы берілген. Оңға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – $A[N]$ жиымының теріс элементтерінің орташа арифметикасының бүтін бөлігі.

17 $A[N]$ жиымы берілген. Оңға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – $A[N]$ жиымының жұп мәндерінің орташа арифметикасының бүтін бөлігі.

18 $A[N]$ жиымы берілген. Оңға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – $A[N]$ жиымының тақ мәндерінің орташа арифметикасының бүтін бөлігі.

19 $A[N]$ жиымы берілген. Жиымның барлық теріс элементтерін оның ең кіші элементіне азайту қажет. Оңға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – оң элементтерінің саны.

20 $A[N]$ жиымы берілген. Жиымның барлық оң элементтерін оның ең үлкен элементіне азайту қажет. Солға қарай k позициясына циклдік жылжыту жасаңыз, мұндағы k – теріс элементтерінің саны.

Б тапсырмасы.

Бір өлшемді жиымды өңдейтін программаны құрыңыз. Жиым элементтеріне қол жеткізуді нұсқауыштың көмегімен жүзеге асыру қажет. Жиым элементтерін кездейсоқ сандар генераторының көмегімен толтырыңыз.

Мысалы. $A[N]$ жиымы берілген. $B[N]$ жиымын $A[N]$ жиымының элементтерін соңынан бастап оңға қарай k позицияға циклдік жылжыту жасап толтырыңыз, мұндағы k – оң элементтерінің саны.

```
#include <stdlib.h>
# include <conio.h>
# include <stdio.h>
#define n 10
int main ()
{ int buf,j,pol,i,b[n], *pa,*pb,*p, a[n];
  for (i = 0; i < n; i++)  a[i]=rand()%10-5;
  pol=0;
  pa=&a[0];
  pb=&b[9];
  printf ("\n \n Генерацияланған жиым a[n] a[n] \n");
  for (i=0;i<n;i++)
    printf ("%d ",*pa++);
  pa=&a[0];
  for (i=0;i<n;i++)
  { *pb=*pa;
    if (*pa>=0)pol++;
    --pb;
    ++pa;  }
  printf ("\n Оң элементтерінің саны \n pol=%d",pol);
  printf ("\n \n b[n] жиымын шығару \n");
  pb=&b[0];
  for (i=0;i<n;i++)
    printf ("%d ",*pb++);
  pb=&b[9];
  p=&b[0];
```

```

for (i=0;i<pol;i++)
{ buf=*pb;
  for(j=0;j<n;j++)
  *pb=* (--pb);
  *p=buf;
  p=&b[0];
  pb=&b[9];  }
printf ("\n %d позициясына b[n] жиым элементтерін жылжыту\n", pol);
pb=&b[0];
for (i=0;i< n;i++)
  printf ("%d ",*pb++);
printf ("\n");
system("PAUSE");
return 0;
}

```

Б тапсырмасының нұсқасы.

Б тапсырмасының нұсқалары осы зертханалық жұмыстың А тапсырмасының нұсқаларына сәйкес келеді.

Бақылау сұрақтары:

- 1) Жиым түріндегі айнымалылар қалай анықталады?
- 2) Жиым элементтерін өңдеу кезінде Си тілінде қандай операторлар қолданылады?
- 3) Бір өлшемді жиымның жеке элементтеріне нұсқауыштың көмегімен қалай қол жеткізуге болады?
- 4) Жиым элементтері экранға қалай шығарылады?
- 5) & унарлы операциясының нәтижесі қандай?
- 6) * унарлы операциясының нәтижесі қандай?
- 7) $uk = \&a[0]$ өрнегі $uk = a$ өрнегіне эквивалентті ме?
- 8) $\&a[i]$ және $a+i$ өрнектері ұқсас па?

2.5 Зертханалық жұмыс №5. Жиымдар мен нұсқауыштар

Жұмыстың мақсаты: Си программалау тіліндегі жиымдарды өңдеудің алгоритмдерін жүзеге асыруды үйрену. Жұмысты орындамас бұрын жиым типтес айнымалылар мен нұсқауышты сипаттаудың ережелерін білу қажет.

Жалпы мағлұмат.

Жиымдар мен нұсқауыштарды сипаттауға қатысты жалпы мәліметтер №4 зертханалық жұмыста көрсетілген. Си тілінде тек бір өлшемді жиымдар анықталған, алайда жиым элементі ретінде жиымның өзі болса, онда көп өлшемді жиымдарды да анықтауға болады, олардың қарапайым түрі ретінде

екі өлшемді жиымдарды қарастырамыз. Екі өлшемді жиымдарды матрица деп те атайды. Жиымды сипаттау кезінде бірінші өлшемі жолдарының санын, ал екіншісі бағандарының санын анықтайды. Екі өлшемді жиым $\text{int } a[2][4]$

$a[0][0] \ a[0][1] \ a[0][2] \ a[0][3]$
 $a[1][0] \ a[1][1] \ a[1][2] \ a[1][3]$

Бірінші индекс – жолдың нөмірін, екінші индекс – бағанның нөмірін көрсетеді.

$a[n][n]$ жиымының элементтерін кездейсоқ сандармен толтырып және оны жиым түрінде шығаратын программаның мысалын қарастырамыз.

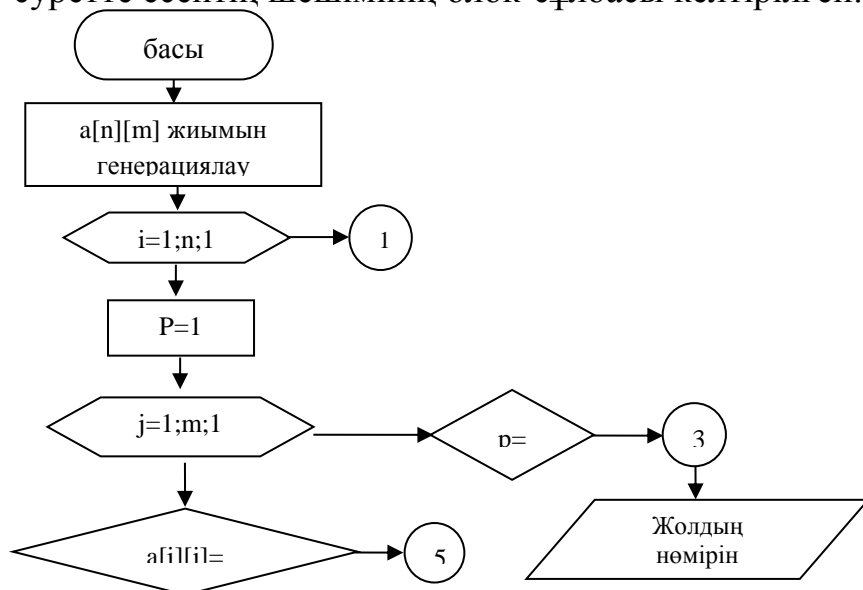
```
#include<stdio.h>
#include<stdlib.h>
#define n 10
void main()
{ int a[n][n],i,j,k,*pa,s1;
  pa=&a[0][0];
  for(i=0;i<n;i++)
  {for(j=0;j<n;j++)
   { *pa=rand()%9;
     printf("%d ",*pa);
     pa++;
   } printf("\n"); }
  getch();
}
```

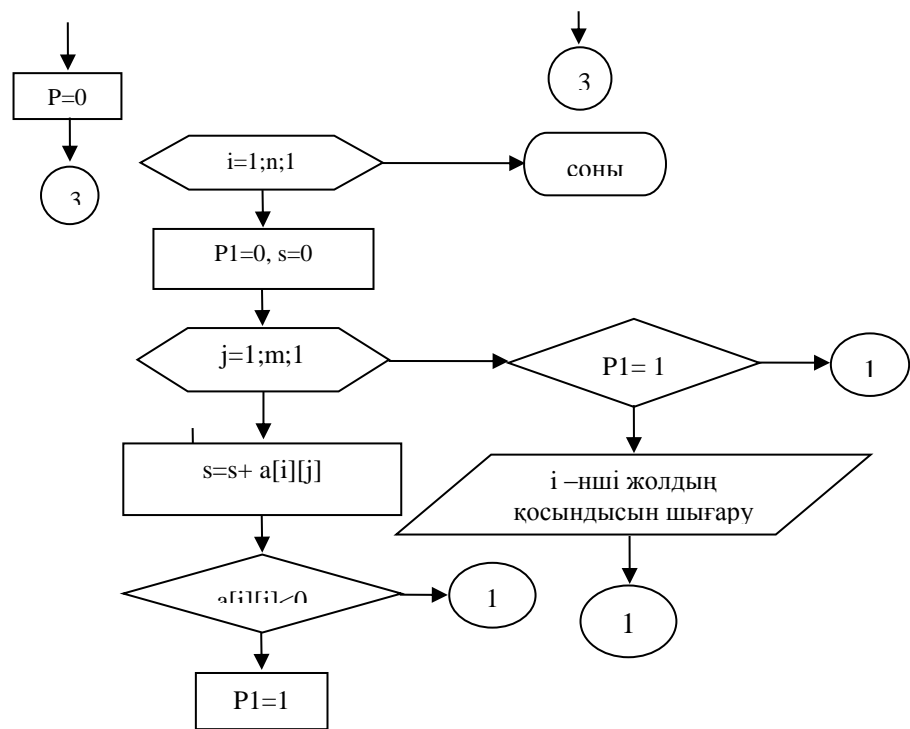
А тапсырмасы.

Блок-сұлбасын және жиымды өңдейтін программаны құрыңыз. Жиым элементтерін кездейсоқ сандар генераторының көмегімен толтыру керек.

А тапсырмасының мысалы. Бүтін санды 8 жолдан, 8 бағаннан тұратын жиым берілген. Табыңыз: 1) жиымның k-ншы жолы k-ншы бағанмен сәйкес келетін k-ны; 2) теріс элементі бар жолдардағы элементтердің қосындысын.

2.5 суретте есептің шешімінің блок-сұлбасы келтірілген.





2.5 сурет – Есептің шешімінің блок-сұлбасы

Программа келесі түрде болады:

```

#include <stdio.h>
#include <stdlib.h>
#define m 8
#define n 8
int main()
{
    int matrix[n][m]; /* nхm өлшемді жиымды хабарлау */
    int i, j, p; /* сәйкес келу санауышы және белгісі */
    int p1, iSumm; /* теріс элементін табу белгісі және қосындыны сақтайтын айнымалы */
    /* жиымды кездейсоқ сандар генераторының көмегімен толтырамыз */
    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            matrix[i][j]=rand()%10-1;
    printf("\n Генерацияланған жиымның түрі:\n ");
    /* жиымды шығару */
    for (i = 0; i < m; i++)
    {
        for (j = 0; j < n; j++)
            printf("%d ", matrix[i][j]);
        printf("\n ");
    }
    printf("\n \t Тапсырманың бірінші бөлімінің нәтижесі");
}
  
```



```

for (i = 0; i < m; i++)
{
    p=1;
    for (j = 0; j < n; j++)
    { /* i-нші жолдың j-нші бағанындағы элементті j-нші жолдың i-нші
бағанымен салыстырамыз. Олар сәйкес келмесе p-ға нөл мәнін меншіктеп, j
бойынша циклды break операторымен үземіз */
        if (matrix [i][j] != matrix [j][i])
        {
            p=0;
            break;
        }
    }
    /* егер p=1 болса, онда экранға сәйкес жолдың нөмірін шығарамыз */
    if (p==1) printf("\n k = %d ", i);
}
if (p==0) printf("\n бірдей жол мен баған жоқ ");
printf("\n \t\t\ Тапсырманың екінші бөлімінің нәтижесі ");
printf("\n\n");
for (i = 0; i < m; i++)
{
    /* айнымалыларға бастапқы мәндерді меншіктейміз */
    iSumm = 0;
    p1=0;
    for (j = 0; j < n; j++)
    { /* i-нші жолдың элементтерінің қосындысын табамыз */
        iSumm += matrix [i][j];
        /* егер де бір ғана теріс элементі табылса, қосындыны экранға
шығаруды белгілейтін, p1=1 меншіктейміз */
        if (matrix [i][j] < 0) p1=1;
    }
    /* егерде жолда бір ғана теріс элемент болса, онда экранға i-нші
жолдың элементтерінің қосындысын шығарамыз */
    if (p1==1) printf("жолдағы элементтердің қосындысы #%d = %d\n",
i+1, iSumm);
}
system("PAUSE");/* экранды тоқтату */
return 0;
}

```

Программаның орындалу нәтижесі:

Генерацияланған жиымның түрі мынадай:

2 7 2 7 1 4 0 3

1 0 7 6 0 -8 3 5

2 7 1 8 1 4 9 3

9 2 8 5 2 0 0 6
7 1 1 3 9 3 9 1
8 2 4 9 1 -6 4 9
0 3 9 0 9 4 8 8
1 8 3 2 8 2 8 0

Тапсырманың бірінші бөлімінің нәтижесі
Сәйкес келген жолдар мен бағандар
 $k = 2$
 $k = 6$

Тапсырманың екінші бөлімінің нәтижесі
жолдағы элементтердің қосындысы #1 = 14
жолдағы элементтердің қосындысы #5 = 31

А тапсырмасының нұсқасы.

1. Бүтінсанды квадрат матрица берілген. Анықтау керек:
 - а) нөлдік элементі жоқ жолдар санын;
 - б) берілген матрицада бірнеше рет кездесетін ең үлкен мәнін.
 2. Бүтінсанды квадрат матрица берілген.
 - а) нөлдік элементі жоқ бағандар санын анықтау;
 - б) бүтінсанды матрицаның жолдарындағы оң тақ элементтерінің қосындысын табыңыз. Берілген матрицаның жолдарын алмастыра отырып, оны өсу реті бойынша орналастырыңыз.
 3. Бүтінсанды квадрат матрица берілген. Анықтау керек:
 - а) бір ғана нөлдік элементі бар бағандар санын;
 - б) ең ұзын бірдей элементтер сериясы орналасқан жолдарды.
 4. Бүтінсанды квадрат матрица берілген. Анықтау керек:
 - а) теріс элементтері жоқ жолдардағы элементтердің көбейтіндісін;
 - б) матрицаның бас диагоналына параллель орналасқан диагоналдағы элементтердің қосындысының ең үлкенін.
 5. Бүтінсанды квадрат матрица берілген. Анықтау керек:
 - а) теріс элементтері жоқ бағандардағы элементтердің көбейтіндісін;
 - б) матрицаның бүйір диагоналына параллель орналасқан диагоналдағы элементтердің қосындысының ең кішісін.
 6. Бүтінсанды квадрат матрица берілген. Анықтау керек:
 - а) теріс элементі бар жолдардағы элементтердің қосындысын;
 - б) матрицаның барлық ершік нүктесі орналасқан жолдар мен бағандардың нөмірін.
- Ескерту - А матрицасының A_{ij} ершік нүктесі дегеніміз i -нші жолдағы ең кіші элементі және j -нші бағандағы ең үлкен элементін айтады.
7. 8-де 8 өлшемнен тұратын матрицаның:
 - а) k -ншы жолымен k -ншы бағаны сәйкес келетін k -ны табу;

б) бір ғана теріс элементі бар жолдардағы элементтердің қосындысын табу.

8. Бүтін санды матрицаның бағандарының мінездемесі деп оның теріс жұп элементтерінің модулінің қосындысын атайды:

а) берілген матрицаның бағандарын ауыстырып, оны өсу реті бойынша орналастыр;

б) ең болмағанда бір ғана теріс элементі бар бағандардағы элементтердің қосындысын тап.

9. Матрицадағы A_{ij} элементінің көршілері деп мына элементтерді айтамыз $A_{i-1,j-1}$, $A_{i-1,j}$, $A_{i-1,j+1}$, $A_{i,j-1}$, $A_{i,j+1}$, $A_{i+1,j-1}$, $A_{i+1,j}$, $A_{i+1,j+1}$. Жиымды тегістеу операциясы берілген жиымның көрші элементтерінің орташа арифметикалық мәнімен толтырылған сол өлшемдегі жаңа жиымды береді.

а) өлшемі 10-да 10 болатын нақты сандардан тұратын жиымды тегістеу нәтижесін құрыңыз;

б) тегістелген жиымда бас диагоналдың төмен жағында орналасқан элементтерінің модулінің қосындысын табыңыз.

10. Жиымның элементі локальды минимум деп аталады, егер ол жанындағы көрші элементтерінен кіші болса.

а) өлшемі 10-да 10 болатын жиымның локальды минимум санын есептеңіз;

б) бас диагоналдың жоғарғы жағында орналасқан элементтерінің модулінің қосындысын табыңыз.

11. Сызықты теңдеулер жүйесінің коэффициенттері тік төртбұрышты жиым түрінде берілген.

а) түрлендірудің көмегімен жүйені үшбұрышты түрге келтіріңіз;

б) элементтерінің орташа арифметикалық мәні берілген мәннен кіші болатын жолдарының санын табыңыз.

12. n ретті нақты квадрат жиым берілген.

а) нөлдермен толтырылған баған мен жолды алып тастау арқылы жиымды ығыстыру;

б) оң элементі бар алғашқы жолдың нөмірін табыңыз.

13. n ретті нақты квадрат жиым берілген.

а) тік төртбұрышты жиымның элементтерін n элементке оңға немесе төмен циклдік жылжыту жасаңыз (енгізілген режимге байланысты). n жолдағы немесе бағандағы элементтер санынан үлкен болуы мүмкін;

б) нөлдік емес элементі бар соңғы жолдың нөмірін табыңыз.

14. n ретті нақты квадрат жиым берілген.

а) $M \times N$ өлшемді тіктөртбұрышты жиым элементтерін k элементке оңға циклдік жылжытуды былай орындаңыз: 1-ші жолдың элементтерін соңғы бағанға жоғарыдан төмен, одан соңғы жолға оңнан солға, одан – бірінші бағанға төменнен жоғары, одан – бірінші жолға, қалған элементтері үшін де – дәл солай жылжыту қажет;

б) тақ жолдардағы оң элементтердің қосындысын тап.

15. Тіктөртбұрышты бүтін сандардан тұратын жиым берілген.

а) бір ғана нөлдік элементі бар алғашқы бағанның нөмірін анықтау;
б) бүтінсанды жиымның жолдарындағы элементтері оның жұп теріс элементтерінің қосындысы болсын. Берілген жиымның жолдарының орнын оның кему ретімен орналастыру қажет.

16. Тіктөртбұрышты бүтін сандардан тұратын жиым берілген.

а) тіктөртбұрышты бүтін сандардан тұратын жиымның жолдарындағы элементтерді өсу ретімен орналастыру;

б) бірде бір теріс элементі жоқ алғашқы бағанды табыңыз.

17. Нақты сандардан тұратын квадрат жиым берілген.

а) элементтерінің орнын ауыстыру арқылы оның ең үлкен элементі сол жақ жоғарғы бұрышта орналасқан болсын, келесі үлкен мәні (2,2) позициясында, келесісі - (3,3) позициясында және т.с.с., осылайша бас диагональ элементтерін толтырыңыз;

б) бірде бір оң элементі жоқ алғашқы жолдың нөмірін табыңыз.

18. Тіктөртбұрышты бүтін сандардан тұратын жиым берілген.

Анықтаңыз:

а) тек бір ғана нөлдік элементі бар жолдың санын;

б) бірдей элементтері бар бағанның нөмірін.

19. Бүтін санды квадрат жиым берілген. Анықтаңыз:

а) теріс элементтері жоқ жолдарының элементтерінің қосындысын;

б) бас диагональға параллель орналасқан диагональдың элементтерінің қосындысын.

20. Тіктөртбұрышты бүтін сандардан тұратын жиым берілген.

Анықтаңыз:

а) тек бір ғана нөлдік элементі бар жолдардағы теріс элементтер санын;

б) жиымның барлық ершік нүктесі бар жолдар мен бағанның нөмірін.

Ескерту - A жиымында A_{ij} ершік нүктесі болады, егер A_{ij} i -ші жолдағы ең кіші элемент болса және j -нші бағандағы ең үлкен мәні болса.

Б тапсырмасы.

Жиымды өңдеу программасын құрыңыз. Жиым элементтеріне қол жеткізуді нұсқауыштың көмегімен жүзеге асырыңыз. Жиым элементтерін кездейсоқ сандар генераторын пайдаланып толтырыңыз.

Мысалы. 8-де 8 өлшемнен тұратын бүтінсанды квадрат матрица берілген. Табыңыз:

а) матрицаның k -ншы жолымен k -ншы бағаны сәйкес келетін k -ны табу;

б) бір ғана теріс элементі бар жолдардың элементтерінің қосындысын.

Есепті шешудің мысалы:

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

```
#include <stdlib.h>
```

```
#define m 8
```

```
#define n 8
```

```

void main()
{ int  matrix[n][m]; /* nхm матрицасын хабарлау */
  int  i, j, p; /* Санауыш және сәйкес келу белгісі */
  int *mat; /* matrix[n][m] жиымына нұсқауышты хабарлаймыз */
  mat=&matrix[0][0]; /* mat нұсқауышына matrix[n][m] жиымының нөлдік
элементінің адресін меншіктейміз */
  /* жиымды кездейсоқ сандар генераторымен толтырамыз */
  for (i = 0; i < m; i++)
  for (j = 0; j < n; j++)
    { *mat=rand()%10-1;
      mat++;
    }
  mat-= n*m; /* mat нұсқауышына алғашқы адресі меншіктейміз*/
  printf("\n Генерацияланған жиым: \n ");
  /* жиымды шығару */
  for (i = 0; i < m; i++)
    { for (j = 0; j < n; j++)
      {printf("%d ",*mat);
        mat++;
      }
      printf("\n ");
    }
  mat-= n*m;
  printf("\n \t\t\t Тапсырманың бірінші бөлігінің нәтижесі");
  printf("\n Сәйкес келген жол мен баған");
  for (i = 0; i < m; i++)
    { p=1;
      for (j = 0; j < n; j++)
        { /* j-нші бағанның i-нші жолын j-нші жолдың i-нші бағанымен
салыстырамыз. Егер олар сәйкес келмесе p-ға ноль мәнін меншіктеп, циклды j
бойынша break операторымен үземіз */
          if (*(mat+i * m + j) != *(mat+j * m + i))
            { p=0;
              break;
            }
        }
      /* егер p=1 болса, онда экранға сәйкес жолдың нөмірін шығарамыз */
      if (p==1) printf("\n k = %d ", i);
    }
  if (p==0) printf("\n бірдей жол мен баған жоқ ");
  printf("\n \t\t\t Тапсырманың екінші бөлігінің нәтижесі ");
  mat=&matrix[0][0];
  int p1, iSumm; /* теріс элементті табу белгісі және қосындыны
сақтайтын айнымалы */

```

```

printf("\n\n");
for (i = 0; i < m; i++)
{
    /* Айнымалыларға бастапқы мәндерін меншіктейміз */
    iSumm = 0;
    p1=0;
    for (j = 0; j < n; j++)
    { /* i-нші жолдың элементтерін қосамыз */
        iSumm += *mat;
        /* ең болмағанда бір теріс элементті табу үшін, Қосындыны экранға
шығару қажеттілігін білдіретін p1=1 меншіктейміз */
        if (*mat < 0) p1=1; mat++;
    }
    /* жолда тым болмағанда бір теріс элементті тапса, онда экранға i-
нші жолдың элементтерінің қосындысын шығарамыз */
    if (p1==1) printf("Жолдағы элементтердің қосындысы  %#d = %d\n",
i+1, iSumm);
    }
    system("PAUSE");/* экранды тоқтату */
    return 0;
}

```

Программаның орындалу нәтижесі.

Есептің шешімі А тапсырмасындағыдай нәтижеге тең.

Б тапсырмасының нұсқасы.

Б тапсырмасының нұсқалары зертханалық жұмыстың А тапсырмасының нұсқаларына сәйкес келеді.

Бақылау сұрақтары:

- 1) Жиым түріндегі айнымалылар қалай анықталады?
- 2) Жиым элементтерін өңдеу кезінде Си тілінің қандай операторлары қолданылады?
- 3) Нұсқауыштың көмегімен жиымның әрбір элементіне қол жеткізу қалай жүзеге асады?
- 4) Жиым элементтері экранға қалай шығарылады?
- 5) & унарлы операциясының нәтижесі қалай?
- 6) * унарлы операциясының нәтижесі қалай?
- 7) $uk = \&a[0][0]$ өрнегі $uk = a$ өрнегіне тепе-тең бе?
- 8) $\&a[i][j]$ және $a+i+j$ өрнектері ұқсас па?

2.6 Зертханалық жұмыс №6. Функцияларды пайдаланып программалау. Рекурсивті функция.

Жұмыстың мақсаты: Си программалау тіліндегі функция, рекурсивті функция түсініктерімен танысу, функцияның, рекурсивті функцияның көмегімен жүзеге асатын алгоритмдердің мысалымен программалау дағдыларын бекіту, тәжірибелік программалауда тізбектеп тексеру әдісін қолдануды үйрену, есептерді шешуде функцияларды қолдану.

Осы жұмысты орындамас бұрын функцияны сипаттау ережелерін, параметрлерді беру механизмін, локальді және глобальді айнымалылар түсінігімен танысып, оқу қажет.

Жалпы мағлұмат.

Си тілінде программалаудың ерекшелігі, ол Си тіліндегі программалар функция деп аталатын блоктардан тұрады.

Функция қандай да бір нақты спецификалық есепті орындайтын C тіліндегі программаның атауы бар жеке фрагменті болып табылады. Функцияның өзі C тіліндегі жеке программаны білдіреді.

Функциялардың барлығы екі үлкен категорияға бөлінеді: программалаушылардың өздері жазған функциялар және C тілінің компиляторлар пакетіне кіретін функциялар. Біріншісі қолданушының анықтаған функциялары, екіншісі – кітапханалық функциялар деп аталады. Қайталанатын программаның кодын өңдеу қажет болса немесе көлемді есепті шешу қажет болса, көлемді есептерді шешу кезінде оларды бірнеше қарапайым кішкентай бөліктерге бөлуге болады, сол кезде функцияларды пайдаланған жөн.

Функцияларды хабарлау үшін келесі синтаксисті пайдаланған жөн:

<тип> <функцияның аты> ([параметрлерінің тізімі]) { <функцияның денесі> }

Тип функцияның қайтарылатын типін анықтайды. Функцияның аты оны программада шақыру үшін қолданылады және оны анықтау ережесі айнымалыларды анықтау ережесімен сәйкес келеді. Параметрлер тізімі функцияны шақырған кезде қандай да бір мәліметтерді беруге қажет. Функцияның денесі – функцияны шақырған кезде орындалатын операторлар жиыны.

Функцияны пайдаланудың мысалы.

```
double square(double x)
{ x = x*x;
return x;
}
int main()
{ double sq1, sq2, arg = 5;
  sq1=square(arg);
  sq2=square(3);
return 0;
```

```
}
```

Бұл мысалда *square* атты функция берілген, ол *double* типтес бір енгізу параметрін қабылдайды, оны квадраттап, есептелінген мәнді шақырылатын программаға *return* операторының көмегімен қайтарады.

Функцияның жұмысы *return* операторын шақырған кезде аяқталатынын ескерген жөн. Егер де осы оператордан кейін тағы да бір операторлар болса да, олар орындалмайды.

C тілінде функция өзін өзі шақыра алады. Бұл үрдісті рекурсия деп атайды. Программалаудың кейбір есептерінде мұндай әрекеттер құрылатын программалық кодты анағұрлым қысқартуға мүмкіндік береді. Осы үрдісті келесі мысалда қарастырамыз.

```
#include <stdio.h>
void up_and_down(int );
int main(void)
{ up_and_down(1);
  return 0;
}
void up_and_down(int n)
{ printf("Төменгі деңгей %d ",n);
  if(n < 4) up_and_down(n+1);
  printf("\n");
  printf("Жоғарғы деңгей %d\n",n);
}
```

Осы программаның нәтижесі экранға келесі жолдарды шығарады:
Төменгі деңгей 1 Төменгі деңгей 2 Төменгі деңгей 3 Төменгі деңгей 4
Жоғарғы деңгей 4 Жоғарғы деңгей 3 Жоғарғы деңгей 2 Жоғарғы деңгей 1

Программа жұмысының нәтижелері былай түсіндіріледі. Басында *main()* функциясы *up_and_down()* функциясын 1 аргументімен шақырады. Нәтижесінде осы функцияның *n* аргументі 1 мәнін қабылдайды және *printf()* функциясы бірінші жолды басып шығарады. Сонан соң тексеру орындалады, егер $n < 4$, онда қайтадан *up_and_down()* функциясын 1 аргументімен $n+1$ -ден үлкен мәнімен шақырылады. Нәтижесінде қайтадан шақырылған функция екінші жолды баспаға шығарады. Осы үрдіс аргументтің мәні 4-ке тең болғанша жалғаса береді. Бұл жағдайда *if* операторы жұмыс істемейді және бесінші жол «Жоғарғы деңгей 4»-ті баспаға шығаратын *printf()* функциясы шақырылады. Сонан соң функция өзінің жұмысын аяқтайды және басқару осы функцияны шақыратын функцияға беріледі. Аргументі $n=3$ -ке тең *up_and_down()* функциясы өзінің жұмысын жалғастыра отырып, «Жоғарғы деңгей 3» 6-шы жолды баспаға шығаратын *printf()* операторына көшеді. Бұл үрдіс бастапқы деңгейге жеткенге дейін жалғаса береді, яғни басқару *up_and_down()* функциясын алғашқы рет шақырған *main()* функциясына беріледі де, программаның жұмысы аяқталады.

Функция күрделі жұмыстарды орындау кезінде олар өздерінің есептерін іске асыру үшін өзінің айнымалыларын қолдануы тиіс.

Функцияның ішінде хабарланатын айнымалылар локальды айнымалылар деп аталады. Локальды (жергілікті) айнымалының аты мен мәні тек функцияның ішінде ғана хабарланған айнымалы ретінде белгілі.

Локальды айнымалыларға қосымша C тілінде сіздің программаңыздың барлық аймақтарында толық белгілі болатын глобальды (ауқымды) айнымалыларды хабарлауға мүмкіндік береді (барлық функциялар үшін глобальды). Глобальды айнымалы функциядан тыс қандай да бір файлда хабарланады.

А тапсырмасы.

Функцияны пайдаланып жиымды өңдеу программасын құрыңыз. Есептің әрбір бөлігін функция түрінде жазыңыз.

Мысалы.

Өлшемі 8-де 8 бүтін санды жиым берілген. Табыңыз:

а) жиымның k -ншы жолымен k -ншы бағаны сәйкес келетін k -ларды;

б) тым болмағанда бір теріс элементі бар жолдың элементтерінің қосындысын тап.

Есепті шешудің блок-сұлбасы №5 зертханалық жұмыста берілген.

Программа келесі түрде болады:

```
#include <stdio.h>
#define m1 8
#define n1 8
/* Функция прототиптерін хабарлау */
void func(int *mat, int m, int n);
void func2(int *mat, int m, int n);
int main(int argc, char* argv[])
{int i, j, matrix[n1][m1];
 /* 8x8 жиымды хабарлап, инициализациялаймыз */
/* жиымды кездейсоқ сандар генераторымен толтырамыз */
for (i = 0; i < m1; i++)
for (j = 0; j < n1; j++)
matrix[i][j]=rand()%10-1;
printf("\n Генерацияланған жиым мынандай болады:\n ");
/* жиымды шығару */
for (i = 0; i < m1; i++)
{ for (j = 0; j < n1; j++)
printf("%d ",matrix[i][j]);
printf("\n ");
}
func1(&matrix[0][0], 8, 8);
func2(&matrix[0][0], 8, 8);
```

```

system("PAUSE");/* экранды тоқтату*/
return 0;
}
void func1(int *mat, int m, int n)
{ int i, j, p; /* Санауыш және сәйкес келу белгісі */
for (i = 0; i < m; i++)
{ p = 1;
for (j = 0; j < n; j++)
{
/* i-нші жолдың j-нші бағанындағы элементті j-нші жолдың i-нші
бағанымен салыстырамыз. Олар сәйкес келмесе Флагқа Жалған мәнін
меншіктейміз және j бойынша циклды break операторымен үземіз */
if (mat[i * m + j] != mat[j * m + i])
{ p =0;
break;
}
}
printf("\n \t\t\t №1 Функцияның нәтижесі");
/* экранға сәйкес жолдың нөмірін шығарамыз */
if (p==1) printf("i=%d ", i);
} if (p==0) printf("\n бірдей жол мен бағандар жоқ ");
}
void func2(int *mat, int m, int n)
{
int i, j; /* Санауыш және қосындыны сақтайтын айнымалы */
int p1, iSumm; /* теріс элементті табатын белгі және қосындыны
сақтайтын айнымалы */
printf("\n\n");
printf("\n \t\t\t №2 Функцияның нәтижесі \n");
for (i = 0; i < m; i++)
{ /* Айнымалыларға бастапқы мәндерді меншіктейміз */
p1 = 0;
iSumm = 0;
for (j = 0; j < n; j++)
егер де бір ғана теріс элементі табылса, қосындыны экранға шығаруды
қажет деп белгілейтін, p1=1 меншіктейміз
{ /* егер де бір ғана теріс элементі табылса, қосындыны экранға
шығаруды қажет деп белгілейтін, p1=1 меншіктейміз */
if (mat[i * m + j] < 0) p1 = 1;
/* i-нші жолдың элементтерінің мәнін қосындылаймыз */
iSumm += mat[i * m + j];
}
/* егерде жолда бір ғана теріс элемент болса, онда экранға i-нші
жолдың элементтерінің қосындысын шығарамыз */
}

```

```

        if (p1) printf("Summ of elements of row #%d = %d\n", i, iSumm);
    }
}

```

Программаның орындалу нәтижесі:

Генерацияланған жиымның түрі мынадай:

```

2 7 2 7 1 4 0 3
1 0 7 6 0 -8 3 5
2 7 1 8 1 4 9 3
9 2 8 5 2 0 0 6
7 1 1 3 9 3 9 1
8 2 4 9 1 -6 4 9
0 3 9 0 9 4 8 8
1 8 3 2 8 2 8 0

```

№1 функцияның нәтижесі

Сәйкес жолдар мен бағандар

k = 2

k = 6

№2 функцияның нәтижесі

#1 жолдың элементтерінің қосындысы = 14

#5 жолдың элементтерінің қосындысы = 31

А тапсырмасының нұсқалары.

А тапсырмасының нұсқалары №5 зертханалық жұмыстың А тапсырмасының нұсқаларымен сәйкес келеді.

Б тапсырмасы.

«Зертханалық жұмыс №3. Циклдік алгоритмдерді программалау» бөлімінен факториалды есептейтін, рекурсивті функцияны пайдаланып жаттығуларды орындаңыз.

Тапсырма: келесі қосындыны есептейтін программаны құрыңыз:

$$S = \sum_{k=1}^n \frac{\cos(kx)}{k!}.$$

Зертханалық жұмыс №3-тегі есептің шешімінің блок-сұлбасы.

Программа келесі түрде болады:

```

#include <stdio.h>
#include <math.h>
long fact( int m)
{ if (m<=1) return (1);

```

```

        else return (m * fact ( m -1 )); // fact функциясы өзін-өзі шақырады
    }
int main()
{
    int n, x, f, i;
    float S;
    printf ("\n Мәнді енгізіңіз  n\n n=");
    scanf ("%d", &n);
    printf ("\n Мәнді енгізіңіз  x\n x=");
    scanf ("%d", &x);
    S=0;
    for (i=1; i<=n; i++)
    {
        f=fact(i);
        S=S+(cos(i*x)/ f);}
    printf ("\n Қосынды S=%f", S);
    getch();
    return 0;
}

```

Программаның орындалу нәтижесі:

```

Мәнді енгізіңіз n
n=4
Мәнді енгізіңіз x
x=1
Қосынды S=0.139995

```

Б тапсырмасының нұсқасы.

Б тапсырмасының нұсқалары №3 зертханалық жұмыстағы А тапсырмасының нұсқаларына сәйкес келеді.

Бақылау сұрақтары:

- 1) Программада процедуралар мен функциялар не үшін қажет?
- 2) Процедуралар мен функциялардың арасында қандай айырмашылық бар?
- 3) Формальды және фактілі параметрлерінің айырмашылығы неде?
- 4) Айнымалы-мән мен параметры-айнымалының айырмашылығы неде?
- 5) Глобальді және локальді айнымалылар қалай хабарланады? Осы айнымалылардың көріну ережесі қандай?
- 6) Процедураға назар аударылған кезде параметр-айнымалыға берілетін аргумент, тұрақты немесе өрнек емес, тек ғана айнымалы бола алады?

7) *return* операторын шақырған кезде функция өз жұмысын аяқтай ала ма?

8) Өзін-өзі шақыратын функция қалай аталады?

2.7 Зерханалық жұмыс №7. Сөз тіркестері және файлдармен жұмыс

Жұмыстың мақсаты: сөз тіркестері және файлдармен жұмыс жасайтын негізгі операцияларды үйрену.

Жалпы мағлұмат.

С тілінде жолдық айнымалыларға арнайы берілгендер жоқ. Бұл мақсаттар үшін жиым символдары қолданылады (тип *char*). Келесі көрсетілген мысалда сөз тіркесін пайдалану көрсетілген:

```
char str_1[100] = {'П', 'р', 'и', 'в', 'е', 'т', '\0'};
```

```
char str_2[100] = "Привет";
```

```
char str_3[] = "Привет";
```

```
printf("%s\n%s\n%s\n", str_1, str_2, str_3);
```

Келтірілген мысалда сөз тіркесін пайдаланудың үш тәсілі көрсетілген. Алғашқы тәсіл жиымның классикалық жариялануы, екінші және үшіншісі арнайы сөз тіркестері үшін қолданылады. Сондай-ақ, соңғы жағдайда компилятор сөз тіркесін жазу үшін жиым ұзындығын өзі анықтайды.

'\0' символы С тілінде сөз тіркесінің соңын білдіреді және одан кейінгі барлық таңбалар сөз тіркесінің символдары ретінде ескерілмейді.

Сөз тіркестерін енгізу функциясы

```
scanf("%s", сөз тіркесінің айнымалысының аты);
```

```
gets(сөз тіркесінің айнымалысының аты);
```

Символды енгізу функциясы

```
printf("%s", сөз тіркесінің айнымалысының аты);
```

```
puts(сөз тіркесінің айнымалысының аты);
```

Сөз тіркестерімен жұмыс жасау үшін `<string.h>` стандартты функциялар кітапханасын пайдалануға болады, онда келесі сөз тіркестерімен жұмыс жасайтын функциялар бар:

1) конкатенация немесе сөз тіркесін біріктіру:

```
strcat(сөз тіркесінің аты1, сөз тіркесінің аты2);
```

2) бір сөз тіркесін екінші сөз тіркесіне көшіру :

```
strcpy(сөз тіркесінің аты1, сөз тіркесінің аты2);
```

3) сөз тіркесіндегі таңбалар санын табу:

```
strlen(сөз тіркесінің аты);
```

4) сөз тіркесі1 мен сөз тіркесі2-ні салыстыру:

```
strcmp(сөз тіркесінің аты1, сөз тіркесінің аты2);
```

Егер сөз тіркестері тең болса, сан нөлден кіші болса 0-ді қайтарады, егер *Сөз тіркесі1* < *Сөз тіркесі2* және сан нөлден үлкен болса, *Сөз тіркесі1* > *Сөз тіркесі2*;

5) сөз тіркесінің кіші әріптерін үлкен әріптерге айналдырады (латын алфавитінің әріптерін ғана өңдейді):

strlwr(сөз тіркесінің аты);

6) сөз тіркесінің үлкен әріптерін кіші әріптерге айналдырады (латын алфавитінің әріптерін ғана өңдейді):

strupr(сөз тіркесінің аты);

7) функцияны шақырған кезде көрсетілген символдармен сөз тіркесін толтыру:

strset(сөз тіркесінің аты, символдың аты);

8) сөз тіркесінен символды іздеу (нұсқауышты алғашқы табылған символға қайтарады, немесе, егер символ табылмаса – NULL мәнін қайтарады):

strchr(сөз тіркесінің аты, символдың аты).

Программа аяқталғаннан кейін барлық мәліметтер жоғалады. Мәліметтерді сақтау үшін файлдар қолданылады.

Файл – бұл әдетте винчестерде (қатқыл диск) сақталынатын мәліметтер бірлігі. Біз файлға қол жеткізудің еркін және тізбектелген түрін қарастырамыз. Енді *файлдармен жұмысты* қарастырайық. Файл - әдетте винчестерде (қатқыл диск) сақталынатын мәліметтер бірлігі. Си тілінде кез келген файл байттар ағыны ретінде қарастырылады. Кез келген файлдың соңы болып арнайы символ табылады.

Файлдармен жұмыс жасағанда енгізу-шығару операциялары орындалады. Енгізу операциясы мәліметтерді ішкі құрылғыдан файлдық буфер арқылы ЭЕМ-нің негізгі жадына жазуды білдіреді (енгізу файлы), шығару операциясы – бұл файлдық буфер арқылы ішкі құрылғыға негізгі жадыдан жіберу (FILE – құрылымы немесе файлдың дескрипторы).

Файлдық буферге қол жеткізу файлдық буфердің адресі бар нұсқауыш арқылы жүзеге асады.

Файлға нұсқауышты анықтау

*FILE *fp;*

Файлдармен жұмыс жасауға қажетті төрт қадам:

- 1) файлды ашу;
- 2) файлды жабу;
- 3) файлды енгізу;
- 4) файлды шығару.

Файлды ашу:

fp=fopen(“*физикалық файлдың аты*”, “*режим*”).

Файлды жабу:

fclose(*сілтеме аты*).

Файлдармен жұмыс жасайтын режимдер:

Режим	Тағайындалуы
r	Файлды оқу үшін ашамыз (файлдан файлдық буферге оқу)
w	Файлды жазу үшін құрамыз (файлдық буферден файлға жазу)
a	Мәліметтерді енгізу режимі (файлдық буферден файлға қосу)

r+	Құрылған мәтіндік файл файлдың кез келген жерінде оқу үшін, жазу үшін ашылады; файлды үлкейтуге рұқсат етілмейді
w+	Файлды жаңарту үшін құрамыз. Егер файл құрылған болса, онда мәліметтер жойылады
a+	Файлды жаңарту үшін құрады. Егер файл бар болса, онда мәліметтер файлдың соңына жазылады

Файлды енгізу функциялары (мәліметтерді файлдан оқу):

1) бір символды енгізу

$айнымалының_аты=gets(нұсқауыштың_аты);$

2) сөз тіркесін енгізу

$fgets(pa, n, fp),$

мұндағы, pa – сөз тіркесінің айнымалысына $нұсқауыштың_аты$ немесе файлдан сөз тіркесін оқитын символдар жиымының аты;

n – файлдан оқылатын символдар саны;

fp – файлдық буфердің адресі бар $нұсқауыштың_аты$.

3) тізбектелген файлдан мәліметтерді оқу

$fscanf(fp, "f", a),$

мұндағы, f – форматты сөз тіркесі;

a – айнымалының адресі.

Мысалы: $fscanf(fp, "%d", &a);$

4) еркін қол жетімді файлдан мәліметтерді оқу

$fread(a, m, n, fp),$

мұндағы, m – мәліметтер блогының өлшемі;

n – мәліметтер блогының саны.

Мысалы. $fread(&a, sizeof(int), 10, fp);$

Файлдарды шығару функциялары (файлдарға мәліметтер жазу):

1) бір символды шығару (бір символды жазу)

$putc(айнымалының_аты, нұсқауыштың_аты);$

2) сөз тіркесін шығару (сөз тіркесін файлға жазу)

$fputs(pa, fp),$

мұндағы, pa – сөз тіркесінің айнымалысына $нұсқауыштың_аты$ немесе файлдан сөз тіркесін оқитын символдар жиымының аты;

fp – файлдық буфердің адресі бар $нұсқауыштың_аты$.

3) тізбектелген қол жетімді файлға мәліметтер жазу

$printf(fp, "f", a),$

мұндағы, f – форматты сөз тіркесі,

a – айнымалының аты.

Мысалы. $fprintf(fp, "%d ", a);$

4) еркін қол жетімді файлға мәліметтер жазу

$fwrite(a, m, n, fp),$

мұндағы, m – мәліметтер блогының өлшемі;

n – мәліметтер блогының саны.

Мысалы: $fwrite(a, sizeof(int), 10, fp);$

А тапсырмасы.

Мысалы. Енгізілген сөз тіркесіндегі «а» әріпінің орнына «+» символымен алмастыратын программаны жазыңыз.

```
#include<stdio.h>
#include<string.h>
int main()
{ int i;
char st[10];
gets(st);
for (i=0;i<10;i++)
if (st[i]!='a')st[i]='+';
printf("\n st=%s\n",st);// puts(st);
system("PAUSE");
return 0;
```

А тапсырмасының нұсқасы.

1. «Тілдер және программалау технологиясы» сөз тіркесінің әр элементін басқа символдық жиымға көшіріп жазу программасын құрыңыз.

2. Үш сөз тіркесін «Зертханалық», «жұмыс», «№6» біріктіріп, төртінші жолға «Зертханалық жұмыс №6» деген сөз тіркесін жазыңыз.

3. Енгізілген сөз тіркесінде кіші әріппен жазылған «а» әрібін үлкен әріппен жазылған түріне ауыстыратын программаны жазыңыз.

4. Енгізілген сөз тіркесінен «н» әріпін алып тастайтын программаны құрыңыз.

5. Енгізілген сөз тіркесінде «е» әріпінің санын санайтын программаны құрыңыз.

6. Енгізілген сөз тіркесінің бірінші сөзінен кейін «салют» сөзін қосатын программаны жазыңыз.

7. Енгізілген сөз тіркесінен «о» әріпін алып тастайтын программаны құрыңыз.

8. Екі жолдағы сөз тіркесін біріктіріп жазатын программаны құр.

9. Енгізілген сөз тіркесіндегі әрбір «а» әріпінен кейін пробел қоятын программаны жаз.

10. Енгізілген сөз тіркесінде үлкен әріппен жазылған «О» әрібін кіші әріппен жазылған түріне ауыстыратын программаны жазыңыз.

11. Енгізілген сөз тіркесіндегі сөздер санын санайтын программаны жаз.

12. Енгізілген сөз тіркесіндегі «и» әріпін санайтын программаны құр.

13. Енгізілген сөз тіркесінің бірінші сөзін «*» символдар тізбегіне ауыстыратын программаны жазыңыз.

14. Енгізілген сөз тіркесіндегі барлық пробелдарды жоятын программаны құр.

15. Енгізілген сөз тіркесінің соңғы сөзін «*» символдар тізбегіне ауыстыратын программаны жазыңыз.

16. Енгiзiлген сөз тiркесiндегi сөздердiң жартысын басқа сөз тiркесiне көшiретiн программаны құрыңыз.

17. Енгiзiлген сөздi соңынан алдына қарай жазатын программаны құр.

18. Енгiзiлген екi сөз тiркесiнiң алғашқы жартысын алмастыратын программаны құр.

19. Сөз тiзбегiндегi соңғы сөздi жоятын программаны құр.

20. Енгiзiлген сөз тiзбегiнiң алғашқы жартысын екiншi жартысымен алмастыратын программаны құр.

Б тапсырмасы.

Орындалатын мәтiндi файлға енгiзу. Файлдан мәлiметтердi программаға енгiзiп, оны өңдеп, файлға жазу және файлдан мәлiметтердi баспаға шығару.

Мысалы. Енгiзiлген сөз тiркесiндегi «а» әрiпiнiң орнына «+» символымен алмастыратын программаны жазыңыз.

```
#include <conio.h>
#include <stdio.h>
int main()
{int i;
 char st[10];
 FILE *fp; /* File құрылымына нұсқауыш хабарлаймыз*/
 fp = fopen("1.txt","w"); /*жазу үшін файлды ашамыз*/
 if (fp == NULL)
 printf("Файлды ашуға немесе құруға мүмкіндік жоқ\n");
 else{
 scanf ("%s", st); /*мәліметтер енгізу*/
 fprintf(fp, "%s\n", st); /* st айнымалысынан мәліметтерді файлға жазу*/
 }
 fclose(fp); /*файлды жабу*/
 fp = fopen("1.txt","r"); /*оқу үшін файлды ашу*/
 if ((fp = fopen("1.txt","r")) == NULL)
 printf("Файлды ашуға немесе құруға мүмкіндік жоқ\n");
 else{
 fscanf(fp," %s", st); /*файлдан мәліметтерді оқу және оларды st айнымалысына жазу */
 for (i=0;i<10;i++)
 if (st[i]=='a')st[i]='+';
 }fclose(fp);
 fp = fopen("1.txt","w"); /*жазу үшін файлды ашу*/
 if (fp == NULL)
 printf("Файлды ашу немесе құру мүмкін емес\n");
 else{
 fprintf(fp, "%s\n", st); /* st айнымалысынан мәліметтерді файлға жазу */
 }
 }
```

```

fclose(fp);
fp = fopen("1.txt","r"); /*файлды оқу үшін ашу*/
if ((fp = fopen("1.txt","r")) == NULL)
    printf("Файлды ашу немесе құру мүмкін емес \n");
else{
    fscanf(fp," %s", st);
    printf (" %s\n", st); /*мәліметтерді шығару*/

}
fclose(fp); /*файлды жабу*/
system("PAUSE");
return 0;
}

```

Б тапсырмасының нұсқасы.

Тапсырма нұсқалары А тапсырмасының нұсқаларына сәйкес келеді.

Бақылау сұрақтары:

- 1) Файл дегеніміз не?
- 2) Файлдың қандай түрлері бар?
- 3) Файлды жасаудағы программаның құрамы қандай болуы керек?
(элементтерді ауыстыру, қосу, жою).
- 4) Файлмен жұмыстың қандай режимдері бар?
- 5) Файлдармен жұмыс жасауға қажетті қандай қадамдар бар?
- 6) Си тіліндегі сөз тіркесінің айнымалылары қалай жарияланады?
- 7) Сөз тіркесінің айнымалыларын инициалдау әдістері?
- 8) С тілінде "\0" символы нені білдіреді?

Қорытынды

Бұл әдістемелік нұсқаулықта негізгі акцент студенттердің зертханалық жұмыстарды орындауда өзіндік жұмыс жасауына қойылған. Теориялық материал және есептердің толық шешімі студенттің материалды өз бетімен оқып және есептерді сәтті шешуіне мүмкіндік береді.

Студенттердің зертханалық жұмыстарды орындау нәтижесінде блок-сұлбаларды құру, Си тілінде тәжірибелік программалаудың операторларын тиімді пайдалануды үйреніп, дағдыланады.

Әдебиеттер тізімі

- 1 Эпштейн М.С. Программирование на языке С.- М.: «Академия», 2011.
- 2 Эпштейн М.С. Практикум по программированию на языке С.- М.: «Академия», 2007.
- 3 Уэйт М и др. Язык Си. Руководство для начинающих.- М.: «Мир», 1988.
- 4 Керниган Б. Язык программирования Си.- М.: «Финансы и статистика», 1992.
- 5 Вирт Н. Алгоритмы и структуры данных.- М.: «Мир», 1989.
- 6 Березин Б.И. Начальный курс С и С++.- М.: «Диалог-Мифи», 2004.
- 7 Архангельский А.Я. Язык С++ в С++ BUILDER.- М., 2008.
- 8 Ишкова Э.А. С++ начала программирования.- М.: «Бином», 2011.
- 9 Коплиен Дж. Программирование на С++.- СПб., 2005.
- 10 Либерти Дж. Освой самостоятельно С++ за 21 день.- М., 2007.
- 11 Павловская Т.А. С/С++. Структурное программирование. - СПб.: «Питер», 2010.
- 12 Подбельский В.В. Программирование на языке Си. – М., 2000.
- 13 Полубенцева М. С/С++ процедурное программирование.- СПб., 2008.
- 14 Культин Н. С/С++ в задачах и примервах.- СПб.: «БХВ-Петербург», 2011.
- 15 Ни А.Г. Языки и технология программирования. Методические указания к выполнению лабораторных работ (для студентов специальности 5В060200 - Информатика). - Алматы, АУЭС, 2013.

Ни Анна Геннадьевна
Адилгажинова Сайран Адилгажыкызы

ТІЛДЕР ЖӘНЕ ПРОГРАММАЛАУ ТЕХНОЛОГИЯСЫ

5B060200 – Информатика мамандығының студенттері үшін зертханалық жұмыстарды орындауға арналған әдістемелік нұсқаулықтар

Редактор Ж.Н. Изтелеуова
Стандарттау бойынша маман Н.Қ. Молдабекова

Басуға __.__.__.қол қойылды
Таралымы 150 дана
Көлемі 3,6 есептік-баспа табақ

Пішімі 60x84 1/16
Баспаханалық қағаз №1
Тапсырыс _____Бағасы 1800 теңге

«Алматы энергетика және байланыс университеті»
коммерциялық емес акционерлік қоғамының
көшірмелі-көбейткіш бюросы
050013, Алматы қаласы, Байтұрсынұлы көшесі, 126