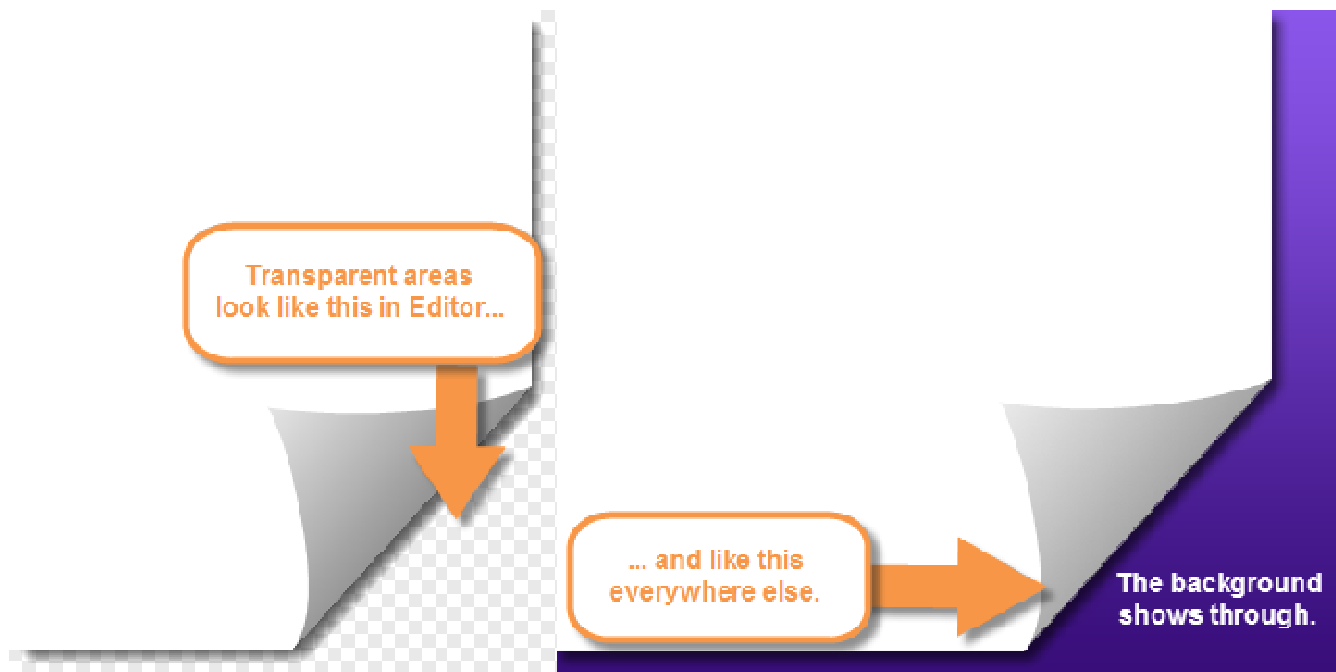


III. Есептің программасы:

```
#include<stdio.h>
#include<conio.h>
main()
{
    int l, w, s;
    clrscr();
    printf("Үшбұрыш ауданын есептеу\n");
    printf("Бастапқы мәндерді енгізіңіз: \n");
    printf("Ұзындығы (см)→");
    scanf("%d", &l);
    printf("Ені (см)→");
    scanf("%d", &w);
    s=l*w;
    printf("\n\nАуданы = %d",s);
    getch();
    return 0;
}
```

Студентке орындатуға арналған тапсырмалар:

1. Екі сан берілген. Осы сандар кубының арифметикалық ортасы мен модульдерінің геометриялық ортасын табыңыз.
2. (x_1, y_1) және (x_2, y_2) координаттарымен берілген екі нүктенің ара қашықтығын есептеңіз.
3. Куб қабырғасының ұзындығы берілген. Кубтың толық бетінің ауданы мен көлемін есептеңіз.
4. Тең қабырғалы үшбұрыш қабырғасы берілген. Осы үшбұрыштың ауданын, биіктігін, іштей және сырттай сы



зылған шеңберлердің радиусын табыңыз.

5. a, b, c коэффициенттерімен берілген $ax^2 + bx + c = 0$ квадрат теңдеудің түбірлерін табыңыз (мұндағы $a \neq 0$ және теңдеу дискриминанты теріс емес).
6. Үшбұрыштың екі қабырғасы a және b белгілі. Олардың арасындағы бұрыш α -ға тең. Ауданын есептеңіз.
7. a шамасы берілген. Көбейтуден басқа ешқандай амалдарды және функцияларды пайдаланбай, a^8 мәнін үш операциядан, ал a^{10} мәнін төрт операциядан кейін алу керек.
8. R_1, R_2, R_3 үш кедергі параллель жалғанған. Жалғау кедергісін табыңыз.

Практикалық жұмыстың есебін дайындау формасы:

Берілген тапсырманың әрқайсысы үшін төмендегілерді орындау:

1. Есептің математикалық моделін анықтау;
2. Есептің алгоритмінің блок-схемасын сызу;
3. Есептің алгоритміне сәйкес программасын жазу.

Блиц-тест

1. C++ тіліндегі негізгі типтер қайсы?
 - a) Таңбалы, таңбасыз
 - b) long, char, int, unsigned
 - c) char, int, float, double
 - d) символдық, жолдық
 - e) ұзын, қысқа
2. int i=3, j, k=0 түрінде сипатталса, k=++i өрнегінің мәндері не болады?
 - a) i=4, k=3
 - b) i=2, k=2
 - c) i=0, k=0

d) i=4, k=4

e) i=4, k=0

3. Енгізу-шығару стандартты функциялары орналасқан тақырыптық файлды таңдаңыз:

a) string.h

b) conio.h

c) math.h

d) ctype.h

e) stdio.h

Бақылау сұрақтары:

1. C/C++ тілдері қандай программалау тілеріне жатады?
2. Универсалды тілдерден айырмашылығын ата?
3. C/C++ тіліндегі сипатталған айнымалыларға жадыдан қанша орын бөлінетіні неден тәуелді?
4. Оперативті жадыдан тіл компиляторы қанша орынды қажет етеді?
5. Оператор деген не?
6. Енгізу-шығару функцияларының түрлерін атаңыз?

Глоссарий

Егер программадағы барлық нұсқаулар бірінен соң бірі тізбектеліп орындалатын болса, ондай программа *сызықтық* программа деп аталады.

C/C++ тіліндегі программа *main функциясы түрінде құрылады*. Оның денесін анықтау коды мынадай:

```
main()
{
/* Кодтар */
}
```

C/C++ тілінде жазылған программа текстінде кездесетін инструкциялар (нұсқаулар) препроцессор директивасы (**ПД**) деп аталады.

#include директивасы.

Бұл программаға көрсетілген файлдағы мәліметтерді қосады. Бұл директиваның екі формасы бар:

#include “файл аты”

include <файл аты>

#define директивасы.

#define – директивасы жиі қолданылатын тұрақтыларды, қызметші сөздерді, операторларды және өрнектерді кейбір идентификаторлармен ауыстыруға мүмкіндік береді.

#undef – директивасы

Бұл директива **#define** директивасы қызметін қайтаруға қолданылады. Яғни, көрсетілген идентификаторларға қолданылатын **#define** анықтамасы қызметін болдырмайды.

Тақырыптық файлдар атаулары төмендегідей:

**assert.h, float.h, math.h, ctype.h,
limits.h, setjmp.h, errno.h, locale.h,
signal.h, stdarg.h, stddef.h, stdio.h,
stdlib.h, string.h, time.h.**

Кейбір тақырыптық файлдар мазмұны Стандартты кітапханалар ретінде жеке қарастырылды, сәйкес бөлімді қараңыз.

С немесе C++ тілдерінде мәндер төмендегідей базалық типтер бірімен бейнеленеді. Олар:

1. char - бірлік байт, қолданылатын литерлер жиынтығындағы бір литерді меншітей алады.
2. int – бүтін.
3. float - 3.4E- 38 және 3.4E+38 аралығындағы нақты сандар.
4. double – 1.7E-308 –1.7E+308 аралығындағы жылжымалы нүктелі сандар.
5. Void типті, ешқандай мән қайтармайтын функцияларды қолданылады.
6. bool типті айнымалы C++-те true немесе false мәнін қабылдайды.
7. Санылымды тип.
8. Көрсеткіштер.

Басқарушы символдар шығарылатын белгілердің экранда орналасуына әсер етеді. Басқарушы символдардың белгісі ретінде \ беріледі. Төменде олардың тізімі келтірілген:

\n – жаңа жолға көшіру;
\t – горизонталды табуляция;
\r – курсорды жаңа жолдың басына қайтару;
\a – қоңырау дыбысы;
\b – бір символ кері қайтару (бір позиция);
\f – жаңа параққа көшу;
\v - вертикалды табуляция.

Формат спецификаторы шығарылатын өлшембірліктің сырт көрінісінің түрін анықтайды. Кейбір формат спецификаторларын келтірейік:

%c – символ;
%s – жол;
%d – бүтін ондық сан (типі int);
%u – белгісі жоқ бүтін ондық сан (типі unsigned);
%f – бекітілген нүкте түріндегі нақты сандар;
%e – қалқымалы нүкте түріндегі нақты сандар (мантиссамен, ретпен берілген)

Кесте №1.-Есептеулер реті мен приоритеттері кестесі

Операторлар	Реті
() [] ! ~ ++ -- + - * & size of * / %	солдан оңға оңнан солға солдан оңға

+	-	солдан оңға
<<	>>	солдан оңға
<	<=	солдан оңға
>	>=	солдан оңға
=	!=	солдан оңға
&		солдан оңға
^		солдан оңға
!		солдан оңға
&&		солдан оңға
::		солдан оңға
?:		оңнан солға
=+=	-=	оңнан солға
*=	/=	оңнан солға
%=	&=	солдан оңға
^=	:=	солдан оңға
<<=	>>=	солдан оңға
,		солдан оңға

Әдебиеттер:

1. Керниган Б., Ритчи Д.,- Язык программирования Си. – М.: Финансы и статистика, 1985 г.
2. Вирт Н. Алгоритмы и структуры данных: Пер. с англ.- М.: Мир, 1989 г.
3. Д.Кнут Искусство программирования для ЭВМ. Основные алгоритмы. М.: Мир, 1976. 3.
4. Н.Культин., С/С++ в задачах и примерах – БХБ-Петербург, 2002 г.
5. Крис Паппас., Программирование на С и С++ - ВНУ, Киев, 2000 г.
6. Т. А. Павловская, А. Ю. Щупак С/С++ Структурное программирование: Практикум- СПб.: Питер, 2005
7. Уэйт М . Язык Си. Руководство для начинающих: Пер. с англ. – М., Мир, 1988 г.

Практикалық жұмыс №2.

Тақырыбы: Тармақталу және таңдау алгоритмдері

Сабақтың мақсаты: Тармақталу конструкцияларын пайдаланып программалауға машықтандыру. Тармақталу алгоритмдерін C/C++ тілдерінде жүзеге асыру жолдарын үйрету.

Қажетті материалдар мен жабдықтар: практикалық сабақтарды орындауға арналған нұсқаулар, тақта.

Жұмыстың мазмұны және орындалу реті:

1. Тапсырмаларды орындау мысалдарын қарастыру.
2. Ретімен берілген тапсырмаларды орындау.
3. Практикалық жұмысты талапқа сай орындап тапсыру.

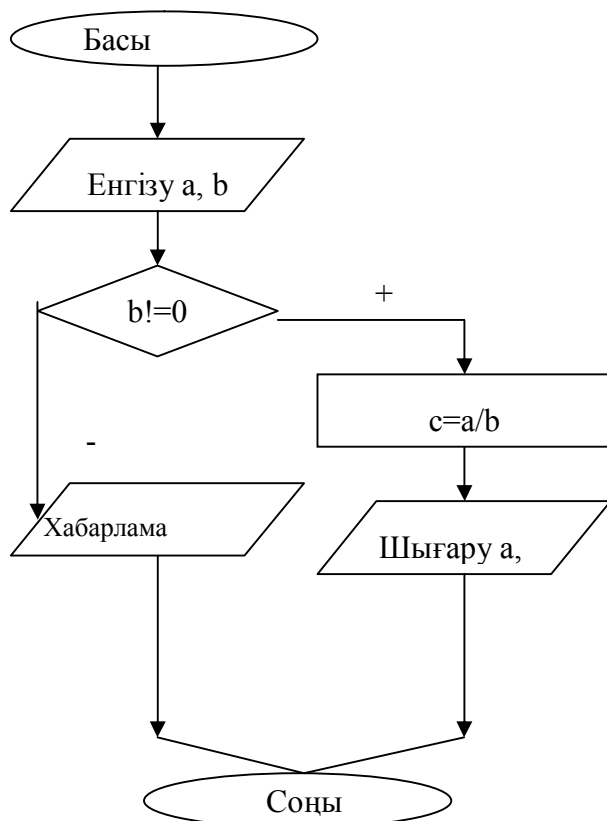
Тапсырманы орындау мысалдары

Екі санды бөлгендегі бөлшек бөлігін есептеу программасын құру. Программада бөліндінің нөлге тең еместігі тексерілуі тиіс.

C/C++ I. Есептің математикалық моделі:

Формуласы: $c = a / b$, a - бөлінгіш; b - бөлгіш.

II. Есептің блоу-схемасы:



II. Есептің программасы:

```
#include<stdio.h>
#include<conio.h>
main()
{
    float a,b, c;
    clrscr();
    printf(" Бастапқы мәндерді енгізіңіз: \n");
    printf("→");
    scanf("%f%f", &a, &b);
    if (b!=0)
    {
        c=a/b;
        printf ("“%5.2f -ны %5.2f – ға өлгендегі бөлшек бөлігінің ", a, b );
        printf ("мәні: %5.2f", c);
    }
    else
    {
        printf ("Бөлімі нөлге тең бола алмайды! \n" );
    }
    getch();
    return 0;
}
```

Студентке орындатуға арналған тапсырмалар:

1. Бір шартты конструкция түрінде жазыңыз:

a.
$$y = \begin{cases} \cos^2 x, & \text{егер } 0 < x < 2, \\ 1 - \sin x^2 & \text{аал аал } a \text{ айда.} \end{cases};$$

b. $\arcsin(1 + \ln x) = a$, теңдеуінің түбірі бар болса, онда x айнымалысына меншіктетіңіз;

c. $d = \max(a, b, c);$

d.
$$z = \begin{cases} \max(x, y) & \text{при } x < 0, \\ \min(x, y) & \text{при } x \geq 0. \end{cases}.$$

2. $A(x_1, y_1)$ және $B(x_2, y_2)$ екі нүкте берілген. Олардың қайсысы координат басына жақын екендігін анықтайтын программа құрыңыз.

3. Өзара тең емесе x және y нақты сандары берілген. Олардың ең кішісін қосындысының жартысымен, ал үлкенін екі еселенген көбейтіндісімен ауыстырыңыз.

4. m, n бүтін сандары берілген. Егер олар өзара тең болмаса, әрқайсысын осы сандардың үлкенімен ауыстырыңыз, ал тең болса 0-мен ауыстырыңыз.

5. a, b, c сандарының ішіндегі теріс сандардың санын анықтаңыз.

6. Экранда “Сен кімсің: қыз немесе ер бала. Д немесе М енгіз”. Қайтарылған жауапқа сәйкес, “Маған қыздар ұнайды” немесе “Маған балдар ұнайды” деген текст экранға шығарылуы керек.
7. Күріш екі пакетке салынған. Біріншісінің массасы – m кг, екіншісінікі – n кг.
а) қай пакет ауыр – біріншісі немесе екіншісі; б) ауыр пакеттің салмағын анықтайтын программа құрыңыз.
8. Ең үздік кәсіпкер сайысының финалына Иванов, Петров, Сидоров өтті. Сайыс үш турдан тұрды. Иванов бірінші турда m_1 , екіншіде – n_1 , үшіншіде – p_1 балл жинаған. Петров сәкесінше- m_2, n_2, p_2 ; Сидоров – m_3, n_3, p_3 . Жеңімпаз қанша балл жинағандығын анықтайтын программа құрыңыз.

Практикалық жұмыстың есебін дайындау формасы:

Берілген тапсырманың әрқайсысы үшін төмендегілерді орындау:

1. Есептің математикалық моделін анықтау;
2. Есептің алгоритмінің блок-схемасын сызу;
3. Есептің алгоритміне сәйкес программасын жазу.
- 4.

Блиц-тест

1. C/C++ тіліндегі «кіші немесе тең» қатынас операторы қайсысы?:
 - a) !=
 - b) <>
 - c) <=
 - d) ==
 - e) >=
2. Қате жазылған жолдың нөмірін көрсетіңіз.
 1. #include<stdio.h>
 2. main()
 3. { int x,y;
 4. scanf("%d %d",&x,&y);
 5. if(x=y) printf("тең");
 6. else printf("тең емес");}
 - a) 5
 - b) 2
 - c) 1
 - d) 6
 - e) 3
3. Қай конструкция таңдау таңдау алгоритмін жүзеге асырады?
 - a) if
 - b) break
 - c) goto
 - d) switch
 - e) for

Бақылау сұрақтары:

1. if конструкциясының толық және толық емес түрлерінің блок –схемасы қандай?
2. Таңдау алгоритмінің қолданылу ерекшелігі неде?
3. Іштестірілген тармақталу алгоритмін құру барысында басты назар аударатын жағдайды түсіндіріңіз№

Глоссарий

if-else конструкциясы.

Төмендегі конструкция программадағы бірнеше жолдың біреуін таңдауға қолданылады:

Ол келесі түрде жазылады:

If (өрнек)

инструкция1

else

инструкция2

Мұнда else тармағы болуы да болмауы да мүмкін. Алдымен «өрнек» есептеледі, егер ол ақиқат болса, онда инструкция1 орындалады. Егер өрнек жалған және else тармағы бар болса, онда инструкция2 орындалады.

If өрнектің сандық мәнін ғана тексеретіндіктен, шартты кейде қысқартылған түрде жазуға болады.

Программалауда екі немесе оданда көп тармақтық жағдайларды ұйымдастыруда тармақталу алгоритмдері құрылады. Ол үшін арнайы инструкция қарастырылған:

Switch (өрнек) {

Case <тұрақты-өрнек> : <инструкциялар>

Case <const-өрнек> : <инструкциялар>

Default : <инструкциялар>

}

Switch инструкциясы өрнектің мәні берілген тұрақтылар жиынынан алынған мәндердің біріне тең болса, онда оған сәйкес тармақты, ал ешқайсысына тең болмаса, онда default сөзінен кейінгі тармақ орындалады, егер ол жоқ болса, ештеңе орындалмайды.

case және default тармақтарын әртүрлі ретпен қоюға болады. Әрбір тармақтың соңына break инструкциясын қойған дұрыс. Тұрақты мәндер қайталанбау керек.

Әдебиеттер:

1. Керниган Б., Ритчи Д.,- Язык программирования Си. – М.: Финансы и статистика, 1985 г.
2. Вирт Н. Алгоритмы и структуры данных: Пер. с англ.- М.: Мир, 1989 г.

3. Д.Кнут Искусство программирования для ЭВМ. Основные алгоритмы. М.: Мир, 1976. 3.
4. Н.Культин., С/С++ в задачах и примерах – БХБ-Петербург, 2002 г.
5. Крис Паппас., Программирование на С и С++ - BHV, Киев, 2000 г.
6. Т. А. Павловская, А. Ю. Щупак С/С++ Структурное программирование: Практикум- СПб.: Питер, 2005
7. Уэйт М . Язык Си. Руководство для начинающих: Пер. с англ. – М., Мир, 1988 г.

Практикалық жұмыс №3.

Тақырыбы: Циклдық процестерді ұйымдастыру

Сабақтың мақсаты: С/С++ программалау тілдерінде циклдық процестерді программалау жолдарын үйрету және цикл конструкцияларын тиімді пайдалануға машықтандыру.

Қажетті материалдар мен жабдықтар: практикалық сабақтарды орындауға арналған нұсқаулар, «math.h» тақырыптық файлының элементтерінің кестесі, тақта.

Жұмыстың мазмұны және орындалу реті:

1. Тапсырмаларды орындау мысалдарын қарастыру.
2. Ретімен берілген тапсырмаларды орындау.
3. Практикалық жұмысты талапқа сай орындап тапсыру.

Тапсырманы орындау мысалдары

Пернетақтадан тізбектеп енгізілген оң сандардың қосындысы мен арифметикалық ортасын есептеу программасын құрыңыз.

```
#include<stdio.h>
#include<conio.h>
main()
{
int a; // пернетақтадан енгізілетін сан меншіктелетін айнымалы
int n=0; // енгізілетін сандар санағышы
int s=0; // сандар қосындысы
float m; // арифметикалық орта мәні
clrscr();
printf(" Сандарды енгізіңіз. Енгізуді аяқтау үшін нөлді басыңыз \n");
do {
printf("→");
scanf("%i", &a);
```

```

if (a>0) { s+=a;  n++; }
    } while (a>0);
printf ("Сан енгізіңіз:  %i\n", n );
printf ("Сандар қосындысы:  %i\n", s );
m= (float) s/n;
printf ("Арифметикалық ортасыh:  %3.2f\n", m );
printf("\n Жұмысты аяқтау үшін енгізу пернесін басыңыз: ");
getch();
return 0;
    }

```

Орындауға берілетін тапсырмалар

1. N натурал саны берілген. Төмендегілерді есептеңіз:

- 2^n ;
- $n!$;
- $(1 + \frac{1}{1^2})(1 + \frac{1}{2^2}) \dots (1 + \frac{1}{n^2})$;
- $\frac{1}{\sin 1} + \frac{1}{\sin 1 + \sin 2} + \dots + \frac{1}{\sin 1 + \sin 2 + \dots \sin n}$;

2. a-нақты саны және n натуралы берілген. Есептеңіз:

- a^n ;
- $a(a+1) \dots (a+n-1)$;
- $\frac{1}{a} + \frac{1}{a(a+1)} + \dots + \frac{1}{a(a+1) \dots (a+n)}$;
- $a(a-n)(a-2n) \dots (a-n^2)$.

X –нақты саны берілген. Есептеңіз:

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

6. n-натурал саны берілген. $f_0 f_1 \dots f_n$ -ді есептеңіз. Мұндағы

$$f_i = \frac{1}{i^2 + 1} + \frac{1}{i^2 + 2} + \dots + \frac{1}{i^2 + i + 1}.$$

7. Іштестірілген қосындыларды есептеңіз: $\sum_{k=1}^{10} k^3 \sum_{l=1}^{15} (k-l)^2$.

Практикалық жұмыстың есебін дайындау формасы:

Берілген тапсырманың әрқайсысы үшін төмендегілерді орындау:

- Есептің математикалық моделін анықтау;
- Есептің алгоритмінің блок-схемасын сызу;
- Есептің алгоритміне сәйкес программасын жазу.

Блиц-тест

1. Циклдың келесі итерациясына басқару билігін беру инструкциясын көрсетіңіз.
 - a) if
 - b) goto
 - c) continue
 - d) switch
 - e) for
2. Шексіз цикл құрайтын жолды таңдаңыз.
 - a) for (i=0;i<=1;i++); printf("цикл\n");
 - b) for (i=1;i<6;i++); printf("цикл\n");
 - c) for (i=5;i<15;i++); printf("цикл\n");
 - d) for (i=1;i>=1;i++); printf("цикл\n");
 - e) for (i=10;i<=1;i--); printf("цикл\n");

Бақылау сұрақтары:

1. Цикл дегеніміз не?
2. Іштестірілген цикл дегеніміз не?
3. C/C++ тілдерінде қандай цикл конструкциялары қолданылады?
4. Циклдың базалық структуралы қандай?

Глоссарий

While циклы

Жалпы форматы:

While <өрнек> инструкция

Мұндағы, <Өрнек> есептеледі, егер мәні 0-ге тең емес болса, онда инструкция орындалады да өрнек қайта есептеледі және ол өрнек мәні 0-ге дейін қайталанады.

for циклы

For (өрнек1; өрнек2; өрнек3) инструкция

Мұндағы: өрнек1 және өрнек3 көбінесе меншіктеу немесе функцияны шақыру, ал өрнек2 қатынас өрнегі болып табылады. Өрнек1 цикл орындалғанға дейін қабылданғаннан кейін қатынас өрнегі тексеріледі. Егер қатынас өрнегі ақиқат мән қабылдаса, онда циклдың денесі ретінде анықталған инструкция орындалады. Цикл денесі орындалған соң, яғни итерация аяқталғаннан кейін өрнек3 орындалады да қатынас өрнегі қайта тексеріледі. Осы процесс оның мәні жалған мән қабылдағанға дейін жалғасады.

Бұл өрнектердің кез-келгені болмаса да ; қойылуы міндетті.
Мысалы,

```
for(;;) {
    ...
}
```

Мұнда өрнек2 мәні барлық уақытта ақиқат деп есептелінеді, бұл шексіз циклды көрсетеді. Мұның жұмысы break немесе return инструкцияларының көмегімен тоқтатылуы мүмкін.

Цикл мысалы:

```
For(i=0,i<n;i++) ...
```

Do-while циклы

Жалпы форматы:

```
Do
    инструкция
while (өрнек);
```

Инструкция орындалған соң өрнек есептеледі, ақиқат болса онда инструкция қайта орындалады.

Break және Continue

Break-циклдан немесе ажыратқыштан уақытынан бұрын шығару.

Continue-циклды келесі итерациясына көшіру

Return инструкциясы-программаның орындалу барысында жұмысын тоқтату үшін қолданылады,аяқталу кодын меншіктейді.

Goto инструкциясы және белгі

Іштестірілген күрделі құрылымдарды ұзу үшін кейде goto қолданылады. Ол көрсетілген белгіге көшеді. Белгі - айнымалы атауы секілді, белгілі бір атпен аталады да соңына : қойылады.

```
Мысалы,      for(...)
                For(...) { ...
                If() goto error;
```

Әдебиеттер:

1. Керниган Б., Ритчи Д.,- Язык программирования Си. – М.: Финансы и статистика, 1985 г.
2. Вирт Н. Алгоритмы и структуры данных: Пер. с англ.- М.: Мир, 1989 г.
3. Д.Кнут Искусство программирования для ЭВМ. Основные алгоритмы. М.: Мир, 1976. 3.
4. Н.Культин., С/С++ в задачах и примерах – БХБ-Петербург, 2002 г.
5. Крис Паппас., Программирование на С и С++ - ВНУ, Киев, 2000 г.
6. Т. А. Павловская, А. Ю. Щупак С/С++ Структурное программирование: Практикум- СПб.: Питер, 2005
7. Уэйт М . Язык Си. Руководство для начинающих: Пер. с англ. – М., Мир, 1988 г.

Практикалық жұмыс №4.

Тақырыбы: Массивтер. Массивтерді өңдеу.

Цель занятия: Массивтерді сипаттау, элементтерін қолдану және өңдеу тәсілдерін үйрету. Массивтерді пайдаланып алгоритмдеу дағдыларын қалыптастыру.

Қажетті материалдар мен жабдықтар: практикалық сабақтарды орындауға арналған нұсқаулар, «math.h» тақырыптық файлының элементтерінің кестесі, тақта.

Жұмыстың мазмұны және орындалу реті:

1. Тапсырмаларды орындау мысалдарын қарастыру.
2. Ретімен берілген тапсырмаларды орындау.
3. Практикалық жұмысты талапқа сай орындап тапсыру.

Тапсырманы орындау мысалдары

Клавиатурадан енгізілген 5 бүтін саннан тұратын бір өлшемді массив құрыңыз. Сол массивтің нөлдік емес элементтерін сәйкес индексін көрсету арқылы экранға шығарыңыз.

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

#define SIZE 5 // массив өлшемі
void main()
{
    int a[ SIZE ]; // массив
    int n=0; // санағыш
    int i; // индекс
    clrscr();
    puts(" \n Бүтін сандар массивін анықтау ");
    puts(" Әр санды енгізген сайын Enter пернесін басыңыз ");
    for (i=0; i< SIZE; i++)
    {
        printf("a[%i]=", i+1);
        scanf("%i", & a[i]);
        if (a[i]!=0) n++;
    }
    printf(" массивте %i нөлдік емес элемент бар\n", n);
    puts("\nЖұмысты аяқтау үшін Enter пернесін басыңыз ");
    getch();
}
```

Орындауға берілетін тапсырмалар

- 1) Он шеңбердің радиустары берілген. Ең үлкен радиуска ие шеңбердің номерін шығарыңыз.
- 2) Фирма қызметкерлерінің жасы белгілі. 20 мен 30 жас аралығында қызметкерлердің санын анықтаңыз.
- 3) Дүкендегі конфеттердің бағасы берілген. Бағасы 500 – ден төмен конфеттердің орташа бағасын есептеңіз.
- 4) Бір аптада шығарылған детальдар саны берілген. Барлығы қанша деталь шығарылған?
- 5) Квадрат матрицаның элементтерінің қосындысы максималды болатын жолының номерін көрсететін программа жазыңыз.
- 6) Екі өлшемді массив берілген. Осы матрицаның төрт бұрышындағы элементтердің қосындысын есептейтін программа жазыңыз.
- 7) Үш дүкенде бұйымдардың төрт түрі бойынша бір күнде түскен пайда берілген. Әрбір дүкен бойынша қайсы бұйым жақсы өткенін анықтау қажет.
- 8) 0, 1 және 2 - ден бөтен әртүрлі сандардан тұратын квадрат массив берілген. Оның бас диагоналдарының элементтерін 0 – ге, бас диагоналдың үстінде тұрған элементтерді 1 –ге, ал бас диагоналдың астында орналасқан элементтерді 2 – ге ауыстыратын программа құру керек.

Практикалық жұмыстың есебін дайындау формасы:

Берілген тапсырманың әрқайсысы үшін төмендегілерді орындау:

1. Есептің математикалық моделін анықтау;
2. Есептің алгоритмінің блок-схемасын сызу;
3. Есептің алгоритміне сәйкес программасын жазу.

Блиц-тест

1. C/ C++ тілінде 10 бүтін элементі бар массив қалай сипатталады?
 - a) `int a(10)`
 - b) `int a{10}`
 - c) `int a[10]`
 - d) `float a[10]`
 - e) `char a[10]`
2. Массивті сипаттау кезінде қолданылатын тұрақтыны қандай директивада сипаттауға болады?
 - a) `# define;`
 - b) `# conio.h;`
 - c) `# stdio.h;`
 - d) `# math.h.`
 - e) `# math`

3. C/C++ тілдерінде массив элементтері нешеден бастап нөмірленеді?

- a) 1
- b) 2
- c) 3
- d) 4
- e) 0

Глоссарий

Массив дегеніміз - бір атпен аталған, әртүрлі индекстелген бір типті элементтер жиынтығы.

Қасиеттері:

Массивтің қасиеттерін анықтайтын 4 негізгі принципі бар:

- массивте элементтер деп аталатын мәндер сақталады;
- массивтің барлық элементтері бір типті болуы керек;
- жадыда массивтің барлық элементі тізбектеліп орналасады және бірінші элемент адрестің нөлдік индексін білдіреді;
- массив атауы тұрақты болып табылады және массивтің бірінші элементінің адресінен тұрады.

Массивтің сипатталуы:

<элементтің типі> <массив аты> [<элементтер саны>];

Мысалы, `int a[12];` {12 элементтен тұратын массив}
`char s[20];` {20 элементтен тұратын массив}
`#define KOL 10 {}`
`int [KOL];`

Массивті инициализациялау 3 тәсілмен анықталады:

- массивті құру барысында;
- массивті құру барысында бастапқы тұрақтыларды көрсету;
- программаның орындалу барысында;
- құру барысында массивтің бастапқы мәндері берілмесе, онда автоматты түрде массив 0-мен толтырылады(NULL).

Массив элементін қолданғанда индекс ретінде үш түрлі мән беруге болады:

1. Сандық тұрақты.
2. Айнымалы.
3. Өрнек.

Екі өлшемді массив екі квадрат жақшамен анықталып төмендегідей сипатталады.

Әдебиеттер:

1. Керниган Б., Ритчи Д.,- Язык программирования Си. – М.: Финансы и статистика, 1985 г.
2. Вирт Н. Алгоритмы и структуры данных: Пер. с англ.- М.: Мир, 1989 г.

3. Д.Кнут Искусство программирования для ЭВМ. Основные алгоритмы. М.: Мир, 1976. 3.
4. Н.Культин., C/C++ в задачах и примерах – БХБ-Петербург, 2002 г.
5. Крис Паппас., Программирование на C и C++ - BHV, Киев, 2000 г.
6. Т. А. Павловская, А. Ю. Щупак C/C++ Структурное программирование: Практикум- СПб.: Питер, 2005
7. Уэйт М . Язык Си. Руководство для начинающих: Пер. с англ. – М., Мир, 1988 г.

Практикалық жұмыс №5.

Тақырыбы: Символдық массивтерді өңдеу

Сабақтың мақсаты: символдық және жолдық мәліметтермен жұмыс кезінде алгоритмдеу тәсілдерін меңгерту; «қадамдап орындау» тәсілін қолдану және көмекші алгоритмдерді құру; қолданушымен функцияны құру және оны Си тілінде шақыру; Си тілінде мәтіндерді өңдеудегі программалау ерекшеліктері; символдар мен жолдарды өңдеуде Си тілінің стандарт кітапханаларының функцияларын пайдалану.

Материалдар және құрылғылар: тақта, бор, «жолдармен жұмыс жасауға арналған функциялар – string.h, stdlid.h файлдары» «символдарды түрлендіру және тексеру функциялары – ctype.h, stdlid.h файлдары» кестелері.

Жұмыстың мазмұны және орындалу тәртібі

1. Тапсырманы орындау мысалын қарау.
2. Ретімен келесі тапсырмаларды орындау.
3. Орындалған практикалық жұмыс туралы есепті дайындау.

Тапсырманы орындау мысалы

Клавиатурадан енгізілген жолда орыс алфавитінің кіші әріптерін бас әріпке ауыстыратын программа жазыңыз.

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

```
void main()
{
    unsigned char st[ 80 ]; // мәтіндік жол
    int i; //қарастырылатын символ нөмірі
    clrscr();
    puts(" \n Жолды енгізіңіз және Enter пернесін басыңыз");
    gets (st);
```

```

i=0;
while (st[i])
{
    if ((st[i]>='a' && st[i]<='z' ) || (st[i]>='а' && st[i]<='я' ) ) st[i]-=32;
    else if (st[i]>='p' && st[i]<='я' ) st[i]-=80;
    i++;
}
printf("\n %s\n", st);
puts("\n Аяқтау үшін Enter пернесін басыңыз ");
getch();
}

```

Студенттердің өз бетімен орындауға арналған тапсырмалары

1. Сіз енгізген жолда b әрпінің ену санын есептейтін программа жазыңыз.
2. Жолды енгізетін және 1 символ қалғанша жолды 1 символға қысқартып шығаратын программа құрыңыз.
3. «а» әрпінен аяқталатын бірінші септіктегі зат есім берілген. Осы сөзді барлық септікте шығаратын программа құрыңыз.
4. Берілген сөзден оның соңғы әрпімен сәйкес келетін әріптерді алып тастайтын программа құрыңыз.
5. Екі сөз берілген. А сөзінің әріптерінен В сөзін құрауға болатынын немесе болмайтынын анықтайтын программа құрыңыз.
6. X және Y сөздерінде бірдей орында тұрған бірдей әріптердің санын есептейтін программа құрыңыз.
7. Клавиатурадан енгізілген жолдан алғашқы бос орындарды жоятын программа құрыңыз.
8. Нүктемен аяқталған, орыс алфавитінің бас әріптерінен тұратын бос емес жол берілген. Осы әріптер алфавит бойынша реттелген бе, жоқ па, анықтайтын программа құру керек.
9. Мәтін берілген. Егер мәтінде * символы болса, оны өзгеріссіз қалдырып, әйтпесе бірінші кездескен * символының алдындағы латын кіші әріптерін 3 цифрына ауыстыратын программа құру керек.
10. n (n <= 1000) натурал саны берілген. Осы санды сөзбен жазып шығаратын программа құрыңыз.

Практикалық жұмыстың орындалуы туралы есептің формасы

Есепте болуы қажет:

Кез-келген орындалған тапсырма үшін

1. Есептің шешімінің математикалық моделі;
2. Есептің шешімінің алгоритмі;
3. Есеп шешімінің алгоритмі орындалған программа.

Блиц-тест

1. Жолдармен жұмыс жасау функциялары анықталған кітапхананы көрсетіңіз
 - a) conio.h
 - b) string.h
 - c) math.h
 - d) stdio.h
 - e) ctype.h
2. Қандай функция жолдың ұзындығын соңғы нөлдік байтсыз қайтарады?
 - a) strlen
 - b) strlwr
 - c) strcmp
 - d) strcpy
 - e) strset
3. Қандай формат командасы жолды экранға шығаруды анықтайды?
 - a) %c
 - b) %i
 - c) %d
 - d) %s
 - e) %f

Бақылау сұрақтары:

1. Оперативті жадыда символдық мәліметтерді бейнелеу тәсілдері.
2. Оперативті жадыда жолдық тұрақтыларды бейнелеу тәсілдері.
3. Оперативті жадыда жолдық айнымалыларды бейнелеу тәсілдері.
4. Си тілінде жолдарды сипаттау тәсілдері.
5. Си тіліндегі символдармен жұмыс жасауға арналған функциялар.

Глоссарий

Егер программада жолдық мәліметтерді өңдеу қажет болса, онда оны символдар массив ретінде қарастыру қажет. Мұндай массивте жолдың әр элементі үшін жеке ұяшықтар бөлінеді және ең соңғы элементі ретінде жолдың соңы таңбасы (\0) қойылуы тиіс.

Символдық массивті енгізу және инициализациялау мысалдарына программа келтірейік:

```
#include <stdio.h>
main ()
{
    char pan[12], razm[5];

    static char opr[11]="информация";

    razm[0]='б';
    razm[1]='а';
    razm[2]='й';
    razm[3]='т';
```

```

razm[4]='\o';

printf("\n информатика сөзін енгізіңіз:");
scanf("%s", pan);

printf("\n %s \n", pan);
printf("\n %s \n", razm);
printf("\n %s \n", opr);

}

```

Көріп отырғанымыздай символық массивке енгізілетін жолдық мәннің ұзындықтары бірге арттырылып алынып отыр. Ол орын жолдың соңы таңбасына арналады. Инициализациялау барысында ол таңб арнайы қойылады.

Әдебиеттер

1. Керниган Б., Ритчи Д.,- Язык программирования Си. – М.: Финансы и статистика, 1985 г.
2. Вирт Н. Алгоритмы и структуры данных: Пер. с англ.- М.: Мир, 1989 г.
3. Д.Кнут Искусство программирования для ЭВМ. Основные алгоритмы. М.: Мир, 1976. 3.
4. Н.Культин., С/С++ в задачах и примерах – БХБ-Петербург, 2002 г.
5. Крис Паппас., Программирование на С и С++ - ВНУ, Киев, 2000 г.
6. Т. А. Павловская, А. Ю. Щупак С/С++ Структурное программирование: Практикум- СПб.: Питер, 2005
7. Уэйт М . Язык Си. Руководство для начинающих: Пер. с англ. – М., Мир, 1988 г.

Практикалық жұмыс №6.

Тақырыбы: Ішкі программалар. Функциялар.

Сабақ мақсаты: Ішкі программаларды ұйымдастыру принциптері мен қажеттілігін түсіндіру. С/С++ тіліндегі функцияларды қолдану дағдысы қалыптастыру. Рекурсивті алгоритмдерді құру машығын қалыптастыру.

Қажетті материалдар мен жабдықтар: практикалық сабақтарды орындауға арналған нұсқаулар, тақта.

Жұмыстың мазмұны және орындалу реті:

1. Тапсырмаларды орындау мысалдарын қарастыру.
2. Ретімен берілген тапсырмаларды орындау.
3. Практикалық жұмысты талапқа сай орындап тапсыру.

Тапсырманы орындау мысалдары

“Факториалды” есептеу функциясын құру факториалдар кестесін экранға шығару үшін осы функцияны қолдану программасын жазу.

Факториалды есептеу формуласы : $n! = 1 * 2 * 3 * \dots * n$.

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

unsigned int factor(int x)
{ unsigned int f=1;for (int i=2; i<=x; i++)  f*=i; return(f);  }
void main()
{
unsigned int f;
clrscr();
puts(" \n Факториалдар кестесі");
for (int n=1; n<=8; n++)
{
    f=factor(n);
    printf("%2i  %u\n", n, f);
}
    getch();
}
```

Практикалық жұмыстың есебін дайындау формасы:

Берілген тапсырманың әрқайсысы үшін төмендегілерді орындау:

4. Есептің математикалық моделін анықтау;
5. Есептің алгоритмінің блок-схемасын сызу;
6. Есептің алгоритміне сәйкес программасын жазу.

Блиц-тест

1. Функция жұмысын аяқтау функциясы?
 - a) return
 - b) exit
 - c) go to
 - d) continue
 - e) break.

Бақылау сұрақтары:

1. Ішкі программа дегеніміз не?
2. Функция аргументі қандай типті бола алады?
3. Функция қандай типті мәндер қайтара алады?
4. Рекурсияның қандай түрлері бар?
5. Функцияның жергілікті аргументі қайда сипатталады?

Глоссарий

Си-дегі кез-келген программа - бұл айнымалылар сипаттамасы мен функциялар жиынтығы.

Функциялар арасындағы байланыс мән қайтаратын аргументтер мен сыртқы айнымалылар арқылы орнатылады.

Return инструкциясы нәтижені шақыратын функциядан шақырған функцияға қайтару механизмін орындайды және return сөзінен кейін кез келген өрнек тұруы мүмкін.

Return <өрнек>;

Си-гі кез келген функцияның прототипі болуы тиіс, яғни функция аты, аргументтерінің типі және қайтарылатын нәтиже типі көрсетілген тақырыбы хабарланады. Прототипі программа денесінде немесе бөлек тақырыптар файлында айқындалады. Функция прототипі компиляторға функция туралы информация береді де, сәйкесінше компилятор типтер сай келуін тексереді. Функция әртүрлі типті мән қайтара алады.

Кез келген функцияны анықтау үшін форматы:

<тип нәтиже> <функция аты> (параметр тізімі)

{

<функция денесі>

Кез келген функцияның формалды аргументтер тізімі болады (болмауы да мүмкін).

Программадан функция шақырғанда аргумент тізімі беріледі.

С++-те аргументтерді беру барысында көрсеткіш орнына тікелей *сілтеме* беруге болады. Сілтеме де аргументке сілтейді, бірақ көрсеткішті айқындау операторын қолдануды қажет етпейді.

Аргументтерінің айнымалы-мән ретінде берілу тәсілі:

Аргументтер мән ретінде берілгенде функцияның ішінде аргумент атас жергілікті айнымалы құрылады да оның мәні функцияда өзгеруі мүмкін, бірақ аргументтің мәні өзгермейді.

Айнымалы-сілтеме арқылы берілу тәсілі:

Аргументтегі айнымалы сілтеме арқылы берілгенде функция аргументтің мәнін емес, *оның адресін* алады. Ал мұнда құрылған жергілікті айнымалы *көрсеткіш* болады. Бұл, жадыны үнемдеуге мүмкіндік береді және функцияның

орындалу барысында айнымалы аргумент мәні өзгереді. Бұл жағдайда функция аргумент айнымалы сілтеме арқылы бірнеше мән қайтара алады.

Өзін өзі шақыратын функция рекурсия деп аталады.

Рекурсия тереңдігі дегеніміз – функция мәнін есептеуде өзін-өзі шақыру саны.

Рекурсивті программалау *стек принципіне* сүйенеді.

Рекурсия түрлері мынадай:

- сызықтық рекурсия;
- параллель рекурсия;
- қосалқы рекурсия;
- жоғары ретті рекурсия.

Әдебиеттер: [

1. Керниган Б., Ритчи Д.,- Язык программирования Си. – М.: Финансы и статистика, 1985 г.
2. Вирт Н. Алгоритмы и структуры данных: Пер. с англ.- М.: Мир, 1989 г.
3. Д.Кнут Искусство программирования для ЭВМ. Основные алгоритмы. М.: Мир, 1976. 3.
4. Н.Культин., С/С++ в задачах и примерах – БХБ-Петербург, 2002 г.
5. Крис Паппас., Программирование на С и С++ - ВНУ, Киев, 2000 г.
6. Т. А. Павловская, А. Ю. Щупак С/С++ Структурное программирование: Практикум- СПб.: Питер, 2005
7. Уэйт М . Язык Си. Руководство для начинающих: Пер. с англ. – М., Мир, 1988 г.

Практикалық жұмыс №8.

Тақырыбы: Көрсетекіштер. Көрсеткіштерге қолданылатын амалдар

Сабақтың мақсаты: Динамикалық құрылымды мәліметтермен жұмыс жасау дағдыларын қалыптастыру, көрсеткіштерге қолданылатын арифметикалық амалдарды үйрету.

Материалдар және құрылғылар: тақта, бор.

Жұмыстың мазмұны және орындалу тәртібі:

1. Тапсырманы орындау мысалын қарау.
2. Ретімен келесі тапсырмаларды орындау.
3. Орындалған практикалық жұмыс туралы есепті дайындау.

Тапсырманы орындау мысалы

Нақты типті элементтерден тұратын массивті тез сұрыптау әдісімен реттейтін программа құру.

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

```
#include<math.h>
```

```
int main()
```

```
{
```

```
    const int n=20;
```

```
    float arr[n], middle, temp;
```

```
    int *stackl=new int [n], *stackr=new int [n], sp=0;
```

```
    int i,j, left, right;
```

```
    clrscr();
```

```
    cout <<" массив элементтерін енгізіңіз ";
```

```
    for (i=0, i<n;i++) cin>>arr[i];
```

```
    // Сортировка
```

```
    sp=1; stackl [1]=0; stackr [1]=n-1;
```

```
    while (sp>0)
```

```
{
```

```
    // Соңғы сұраныс стегінен таңдау
```

```
    left= stackl [sp];
```

```
    right= stackr [sp];
```

```
    sp--;
```

```
    while (left< right) {
```

```
    // Разделение
```

```
    i= left; j= right;
```

```
    middle= arr[( left+ right)/2];
```

```
    while (i< j) {
```

```
    while (arr[i]< middle) i++;
```

```
    while (middle <arr[j]) j--;
```

```
    if (i <=j) {
```

```
    temp= arr[i]; arr[i]= arr[j]; arr[j]=temp;
```

```
    i++; j--;
```

```
    }
```

```
}
```

```
    if (i < right) {
```

```
    // Оң жақтағы сұраныс стегін жазу
```

```
    sp++;
```

```
    stackl [sp]=i; stackr [sp]= right;
```

```
}
```

```
    right=j;
```

```
    // Енді left және right сол жақ бөлікті шектейді
```

```
}
```

```
}
```



```
// Нәтижені шығару
for (i=0, i<n;i++) cout<<arr[i]<<' ';
cout<<endl;
return 0;
}
```

Студенттердің өз бетімен орындауға арналған тапсырмалары

1. Нақты типті n элементтен тұратын массивтегі соңғы теріс элементтің оң жағында орналасқан элементтердің қосындысын есептейтін программа жазыңыз.
2. Нақты типті n элементтен тұратын массивтегі алғашқы оң элементтен кейін орналасқан элементтердің қосындысын есептейтін программа жазыңыз.
3. Тізім берілген. Оның ақпараттық бөлігі оң сандардан тұрады. Ең үлкен санды табу керек.
4. Ақпараттық бөлігі студенттердің фамилиясынан тұратын тізім құру керек. Тегі «Петров» болып келген студенттер санын табу керек.
5. Ақпараттық бөлігі нақты сандардан тұратын тізім құру керек. Тізімнің оң элементтерінің көбейтіндісін табу керек.
6. Ақпараттық бөлігі бүтін сандардан тұратын тізім құру керек. Олардың қосындысын табу керек.
7. Ақпараттық бөлігі бүтін сандардан тұратын тізім құру керек. Тізімнің 2-ге еселі элементтерінің санын табу керек.
8. Кітап атауларының реттелген тізімі берілген. Тізімдегі реттілікті сақтай отырып жаңа кітап туралы ақпарат қосу керек.

Практикалық жұмыстың орындалуы туралы есептің формасы

Есепте болуы қажет:

Кез-келген орындалған тапсырма үшін

1. Есептің шешімінің математикалық моделі;
2. Есептің шешімінің алгоритмі;
3. Есеп шешімінің алгоритмі орындалған программа.

Блиц-тест

1. C++ тілінде көрсеткіш дегенімі не?
 - a) мәліметтер регистріні нөмірі
 - b) оперативті жадыда стек сегментін анықтаушы
 - c) басқа айнымалының адресі бар айнымалы
 - d) жіберуді басқаруды анықтайтын белгі
 - e) кейбір пернетақтада жоқ пернелерді бейнелейтін символдар тізбегі
2. Стек- бұл бір бағытталған тізімнің дербес түрі, мұнда:
 - a) элементті басына қосуға, ал соңынан кез-келген элементті жоюға рұқсат етіледі;
 - b) шыңы деп аталатын бір шетіне элементтерді қосуға және жоюға рұқсат етіледі;
 - c) бірінші және соңғы элементтер арасында байланыс анықталған;

- d) элементті соңына қосуға және басынан жоюға рұқсат етіледі;
 - e) аталған нұсқалардың кез-келгені;
3. Сызықты байланысқан структураға жатады:
- a) Стектер, ширеттер, тізімдер
 - b) Ағаштар, желілер
 - c) Стектер, ағаштар
 - d) Ширеттер, тізімдер, желілер
 - e) Тізімдер, желілер

Бақылау сұрақтары

1. Статикалық айнымалылар мен динамикалық айнымалылар арасындағы айырмашылықты көрсетіңіз.
2. Көрсеткіштермен жұмыс жасау кезінде базалық типтің рөлін түсіндіріңіз.
3. Динамикалық айнымалылардың қолданылу себептерін көрсетіңіз.
4. Қолданушы анықтаған типті базалық тип ретінде қолдануға бола ма?
5. Динамикалық жады шектелген бе?

Глоссарий

Көрсеткіштер дегеніміз - басқа айнымалыларының адресінен тұратын айнымалы.

Көрсеткіш - адрес сақтаған ұяшықтар тобы (2 н/се 4 ұяшық).

Ұяшықтардың нақты адрестерін қолдану үшін арнайы операторлар қарастырылған:

& унарлық операторы объектінің адресін береді.

R=&C - инструкциясы C ұяшығының адресін R айнымалысынаа меншіктейді немесе R айнымалысы C-ға сілтейді деп аталады.

*** унарлық операторы** көрсеткішті сипаттауда қолданылады. Көрсеткіш бұл операторды қолданылғанда осы көрсеткіш сілтейтін объектіні береді. Си тілінде функция өз аргументі ретінде параметр-мәнді қабылдайды, сондықтан шақырған функцияның айнымалысын шақырылған функция денесінде отырып өзгерту мүмкін емес. Мұны орындау үшін мәнге көрсеткіштерді қолдану керек.

Көрсеткіш – бұл айнымалы, сондықтан оны $r=a$ немесе $r++$ деп жазуға болады. Ал, массив аты тұрақты ретінде қатысады, яғни айнымалы емес, сондықтан оны $a=r$ немесе $a++$ деп жазуға болмайды.

Көрсеткіштерді инициализациялау

Көрсеткіштерді сипаттау барысында инициализациялауға болады.

- Көрсеткіштегі адресі кез-келген бүтін санды қосу және алу арқылы өзгертуге де болады. Мысалы, адресі 14 ұяшыққа жылжыту үшін $prb=prb+14$;
- Көрсеткіштен бүтін мәнді шегеруге де болады, бұл жағдайда ағымды ұяшықта тұрған көрсеткіш солға қарай көрсетілген шамаға ығысып, сондағы элементке сілтейді.
- Бір көрсеткішті екінші көрсеткіштен шегеруге болады (2кКөрсеткіш те 1 массивке сілтеуі тиіс) мұның нәтижесі олар сілтеп тұрған ұяшықтар

арасындағы элементтер санына тең болады. Егер бұл өрнекте екі әртүрлі типті көрсеткіш қарастырылса және олар әртүрлі массивке сілтесе амал орындалғанда нәтижесі мағынасыз (түсініксіз) мән қайтарады.

- Көрсеткіштерді салыстыру амалы қолданылады. Бұл \leq , $=$, \geq операторлары көмегімен салыстырылады нәтижесінде false немесе true мәндерін қабылдайды. Бұл мәнді шартты өрнектерде қолдануға болады және бүтін типті айнымалыға меншіктетуге болады.
- Көрсеткішті нөлмен салыстыруға болады. Егер нөлге тең болса, онда оның ешқандай объектімен байланыспағанын білдіреді. Нөлді кезкелген типті көрсеткішке меншіктетуге болады.
- Көрсеткішті нәтижесі нөлге тең тұрақты өрнекпен салыстыруға болады.
- Сол секілді void типті басқа көрсеткішпен де салыстыруға болады.
- Көрсеткіштерді бір-біріне қосуға, бөлуге және көбейтуге болмайды.

Әдебиеттер

1. Керниган Б., Ритчи Д.,- Язык программирования Си. – М.: Финансы и статистика, 1985 г.
2. Вирт Н. Алгоритмы и структуры данных: Пер. с англ.- М.: Мир, 1989 г.
3. Д.Кнут Искусство программирования для ЭВМ. Основные алгоритмы. М.: Мир, 1976. 3.
4. Н.Культин., С/С++ в задачах и примерах – БХБ-Петербург, 2002 г.
5. Крис Паппас., Программирование на С и С++ - BHV, Киев, 2000 г.
6. Т. А. Павловская, А. Ю. Щупак С/С++ Структурное программирование: Практикум- СПб.: Питер, 2005
7. Уэйт М . Язык Си. Руководство для начинающих: Пер. с англ. – М., Мир, 1988 г.

Практикалық жұмыс №8.

Тақырыбы: Структуралар. Бірігулер.

Сабақтың мақсаты: Мәліметтердің қосымша типтерімен жұмыс жасау дағдыларын қалыптастыру.

Материалдар және құрылғылар: тақта, бор.

Жұмыстың мазмұны және орындалу тәртібі:

1. Тапсырманы орындау мысалын қарау.
2. Ретімен келесі тапсырмаларды орындау.
3. Орындалған практикалық жұмыс туралы есепті дайындау.

Тапсырманы орындау мысалы

Келесі өрістерден тұратын STBOAT структурасын беретін программа жазыңыз:

- модель
- тіркеу нөмірі
- жасалған жылы
- мотосағаттар саны
- құны.

```
#include<stdio.h>
#include<conio.h>
struct stboat
{
char szmodel [15];
char szserial [20];
int iyear;
long lmotor_hours;
float fsaleprice;
} stused_boat;
int main(void)
{
clrscr();
printf("\n Қайықтың моделін енгізіңіз- ");
gets (stused_boat. szmodel);
printf("\n Қайықтың тіркеу нөмірін енгізіңіз - ");
gets (stused_boat. szserial);
printf("\n Қайықтың жасалған жылын енгізіңіз - ");
scanf ("%d", &stused_boat.iyear);
printf("\n Қозғалтқыштың жұмыс жасаған мотосағаттар санын енгізіңіз - ");
scanf ("%d", &stused_boat. lmotor_hours);
printf("\n Қайықтың құнын енгізіңіз - ");
scanf ("%a", &stused_boat. fsaleprice);
printf("\n\n");
printf("\n Қайық %s%d жылы дайындалған тіркеу нөмірі № %s, \n ", stused_boat.
szmodel, stused_boat. iyear, stused_boat. szserial );
printf("Орындалған мотосағатта %d p", stused_boat. lmotor_hours);
printf("%08.2f $ сатылды \n", stused_boat. fsaleprice);
puts("\Аяқтау үшін Enter басыңыз ");
getch();
return (0)
}
```

Студенттердің өз бетімен орындауға арналған тапсырмалары

1. Келесі өрістерден тұратын DOC тип атаулы структураны сипаттаңыз:
 - Құжат жолының нөмірі;
 - Жол мәтіні.

Структураны анықтайтын айнымалыны S деп атаңыз.

2. Мектеп мәліметтер қорына арналған ақпараттан тұратын BAZA тип атаулы структураны сипаттаңыз:
 - Оқушының жеке нөмірі ;
 - ТАӘ;
 - Туған жылы;
 - Мекен-жайы.
- Структураны анықтайтын айнымалыны Inf деп атаңыз.
3. Топ оқушыларының геометриядан алған бағаларынан тұратын Geometr тип атаулы структураны сипаттаңыз:
 - ТАӘ;
 - Тоғыз айда алған бағалары, бір айда ең көбі 20 баға.

Структураны анықтайтын айнымалыны Dig деп атаңыз.

4. Қоймада 30 күннің орташа температурасы туралы ақпараттан тұратын Data тип атаулы структураны сипаттаңыз:
 - Ай нөмірі;
 - Температура.

Структураны анықтайтын айнымалыны Zamer деп атаңыз. Структураға келесі бастапқы мәндерді меншіктеңіз: ай аты «маусым», бірінші күннің температурасы 9,5.

5. Пошта мәліметтер қорындағы газет-журналдарға жазылушылар туралы ақпараттан тұратын Post тип атаулы структураны сипаттаңыз:
 - ТАӘ;
 - Мекен-жайы;
 - Газет-журналдар атауынан тұратын 10 жол.

Структураны анықтайтын айнымалыны G деп атаңыз. Өрістерге келесі мәндерді меншіктетіңіз: ‘Петров И.В.’, ‘Мәскеу қ., Горький к., 5’, ‘Московский комсомолец’, ‘Спорт’

6. Достардың телефон-анықтамалығын сипаттайтын структуралар массивінен тұратын программа құрыңыз. Онда мәліметтерді енгізуге, фамилиясы бойынша телефон нөмірін іздеуге, «компьютер ойындарымен әуестенетін» барлық абоненттерді есептеп, тізімін шығаруға болады. Структурада әрбір дос үшін келесі мәліметтер берілген:тегі, аты, телефоны, хоббиі.

7. Химиялық элементтер кестесін сипаттайтын программа құрыңыз. Онда келесі мәліметтерді көрсетіңіз: атауы, символдық белгіленуі, атом массасы, атомдық ядроның заряды, негізгі химиялық қасиеттер тізімі. Программа көрсетілген символдық белгіленуіне байланысты химиялық элемент туралы мәліметті шығаруды, ең үлкен массалы, ең кіші ядро заряды бар элементті табуды орындауы қажет.

8. Үй тұрғындары структурасы массивін сипаттайтын программа құрыңыз. Онда әрқайсысы үшін келесі ақпараттар берілген: пәтер нөмірі, тегі, аты, жасы, 18 жастан асқан тұрғындар үшін қызмет түріне байланысты (оқу, жұмыс, зейнет) – жұмыс орны және еңбек өтілі туралы, зейнеткерлер үшін зейнетке шыққан жылы туралы жазба болуы керек. Программа мәліметтерді енгізуді, тұрғындар саны ең

көп пәтерді іздеуді, ең жас және ең кәрі тұрғынды табуды, студенттер мен зейнеткерлерді іздеуді жүзеге асыруы керек.

Практикалық жұмыстың орындалуы туралы есептің формасы

Есепте болуы қажет:

Кез-келген орындалған тапсырма үшін

1. Есептің шешімінің математикалық моделі;
2. Есептің шешімінің алгоритмі;
3. Есеп шешімінің алгоритмі орындалған программа.

Блиц-тест

1. Қандай көмекші сөздің көмегімен біріктіру құрылады?
 - a) struct;
 - b) swith;
 - c) union;
 - d) enum;
 - e) record;
2. Қандай көмекші сөздің көмегімен саналым құрылады?
 - a) struct;
 - b) swith;
 - c) union;
 - d) enum;
 - e) record;
3. Қандай көмекші сөздің көмегімен структура құрылады?
 - a) struct;
 - b) swith;
 - c) union;
 - d) enum;
 - e) record;

Бақылау сұрақтары

1. C/C++ тілдерінде қандай қосымша типтер қолданылады?
2. Саналымды типті айнымалылар қалай сипатталады?
3. Қандай көмекші сөздің көмегімен біріктіру құрылады?
4. Структура дегеніміз не?
5. typedef құралы не үшін қолданылады?

Глоссарий

Структура – бір атпен біріктірілген бір немесе бірнеше әр түрлі типті айнымалылар тобы. Структура **struct** қызметші сөзімен құрылады. Структураның жалпы құрылымы төмендегідей:

struct meg {

```

mun1 иден1;
mun2 иден2;
...
munN иденN;
};

```

мұндағы *тег* мәліметтердің жаңа типі болып саналады және оны айнымалыны сипаттауда қолдануға болады.

Сипатталған структура негізінде айнымалы құру үшін

```
struct <тег> <айнымалы>
```

C++ тілінде *struct* қызметші сөзін көрсетпесе де болады:

```
<тег> <айнымалы>
```

Айнымалыны структураны құру кезінде де анықтауға болады:

```

struct тег {
    mun1 иден1;
    mun2 иден2;
    ...
    munN иденN;
} <айнымалы>;

```

Егер структура типті басқа айнымалылар құрылмайтын болса, структура тегін көрсетпеуге болады. Мұндай структура *атаусыз (безымянный)* деп аталады.

Структуралар іштестірілген де болуы мүмкін:

```

struct тег1 {
    mun1 иден1;
    mun2 иден2;
    mun3 иден3;
};
struct тег2 {
    mun1 иден1;
    mun2 иден2;
    struct тег1 иден3;
};

```

Коріп отырғанымыздай структура элементтерін **нүкте** операторының көмегімен алуға болады, форматы мынадай:

```
<айнымалы>.<элемент>
```

Структура элементтері циклда енгізілгенде `_flusshall()` –функциясы кіріс легіндегі жаңа жол символын өшіруге, яғни буферді тазалауға қолданылады. Жаңа жол символы `scanf()` функциясын қолданғанда қойылады. Ал, `gets()` функциясы бұл таңбаға дейінгі символдарды ғана оқиды.

Функцияға структура типті объектіні беру тәсілдері үш түрлі:

1. Компоненттерін жеке-жеке беру.

2. Структураны толығымен беру.
3. Структураға көрсеткішті беру.

Көрсеткіш келесі түрде сипатталуы мүмкін:

*struct <тег> *<айнымалы>*

Бұндай жағдайда структуралық айнымалының кез-келген элементін мына әдістердің бірімен алуға болады:

(*<айнымалы>).<элемент> немесе <айнымалы> -> <элемент>

Әртүрлі уақытта әрүрлі типті және әртүрлі өлшемді объектілерді сақтай алатын айнымалы –*біріктіру* деп аталады. Өлшемдеріне және реттеуге қатысты талаптарды компиляция орындайды. Жадының бір аймағында әртүрлі мәндерді сақтауға мүмкіндіктері бар.

Біріктіру синтаксисі структура синтаксисі секілді:

```
Union unprim {
    Int iman;
    Float fman;
    Char *sman;
} mainun;
```

Си тілінде мәндер типіне жаңа атау беруге болатын құрал - typedef құралы бар.

Әдебиеттер

1. Керниган Б., Ритчи Д.,- Язык программирования Си. – М.: Финансы и статистика, 1985 г.
2. Вирт Н. Алгоритмы и структуры данных: Пер. с англ.- М.: Мир, 1989 г.
3. Д.Кнут Искусство программирования для ЭВМ. Основные алгоритмы. М.: Мир, 1976. 3.
4. Н.Культин., С/С++ в задачах и примерах – БХБ-Петербург, 2002 г.
5. Крис Паппас., Программирование на С и С++ - ВНУ, Киев, 2000 г.
6. Т. А. Павловская, А. Ю. Щупак С/С++ Структурное программирование: Практикум- СПб.: Питер, 2005
7. Уэйт М . Язык Си. Руководство для начинающих: Пер. с англ. – М., Мир, 1988 г.

Практикалық жұмыс №9.

Тақырыбы: С++ тіліндегі графика

Сабақтың мақсаты: Графикалық режимде жұмыс жасау дағдыларын қалыптастыру.

Материалдар және құрылғылар: тақта, бор, “графикалық жүйені басқаруға арналған функциялар” кестесі.

Жұмыстың мазмұны және орындалу тәртібі:

1. Тапсырманы орындау мысалын қарау.
2. Ретімен келесі тапсырмаларды орындау.
3. Орындалған практикалық жұмыс туралы есепті дайындау.

Тапсырманы орындау мысалы

Графикалық программа шаблоны:

```
#include<graphics.h>
```

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

```
#define PATHODRIVER "c:\\\\borlandc\\bgi\\"
```

```
void main(void)
```

```
{
```

```
    int gdriver=DETECT; // Драйвер
```

```
    int gmode; // режим
```

```
    int errorcode; //қате коды
```

```
initgraph(&gdriver, &gmode, PATHODRIVER);
```

```
errorcode =graphresult();
```

```
if (errorcode!= grOk) //графикалық режимді инициализациялау қателігі
```

```
{
```

```
    printf("қате:%d \n", errorcode);
```

```
    puts("\n Аяқтау үшін  Enter-ді басыңыз ");
```

```
    getch();
```

```
}
```

```
    // әрі қарай программа инструкциясы
```

```
    closegraph(); // графикалық режимді аяқтау
```

```
}
```

Студенттердің өз бетімен орындауға арналған тапсырмалары

1. Экранда үйді сызатын программа құрыңыз.
2. Экранға Олимпиада ойындарының жалауын шығаратын программа құрыңыз.
3. Экранға диаметрі және түсі кездейсоқ болып келген 100 шеңберден тұратын өрнек сызатын программа құрыңыз.
4. Экранға өлшемі және түсі кездейсоқ болып келген 50 тіктөртбұрыштан тұратын өрнек сызатын программа құрыңыз.
5. Экранда алтыбұрыш сызатын программа құрыңыз.
6. Экранда шахмат тақтасының бейнесін сызатын программа құрыңыз.
7. Экран бойымен қозғалатын дөңгелек сызатын программа құрыңыз.
8. Экранда $y = x^2 + 5$ функциясының графигін шығаратын программа құрыңыз.

Практикалық жұмыстың орындалуы туралы есептің формасы

Есепте болуы қажет:

Кез-келген орындалған тапсырма үшін

1. Есептің шешімінің математикалық моделі;
2. Есептің шешімінің алгоритмі;
3. Есеп шешімінің алгоритмі орындалған программа.

Блиц-тест

1. C/C++-те `closegraph` функциясы:
 - a) TurboC ортасындағы программаның ағымдық терезесін жабу.
 - b) Программаны жабу және TurboC ортасынан шығу.
 - c) Программаны орындау режиміне көшу.
 - d) Жүйені мәтіндік режимге ауыстыру (графикалық режимнен).
 - e) Графיקаны инициализациялау.
2. C/C++-те `setcolor` функциясы
 - a) Фон түсін орнатады.
 - b) Бейне нүктесінің түсін орнатады.
 - c) Стилді орнатады.
 - d) Экранды фон түсімен тазартады.
 - e) Графיקаны инициализациялау.
3. C/C++-те `cleardevice` функциясы:
 - a) Экранды фон түсімен тазартады.
 - b) Фон түсін орнатады.
 - c) Стилді орнатады.
 - d) Жүйені мәтіндік режимге ауыстыру (графикалық режимнен).
 - e) Графיקаны инициализациялау.

Бақылау сұрақтары

1. Графикалық режимді баптауға арналған енгізгі процедураларды атаңыз.
2. Фигураларды тұрғызу үшін қандай процедуралар қолданылады?
3. Графикалық режимде мәтінді шығару үшін қандай процедуралар қолданылады?

Глоссарий

C/C++ тіліндегі графика – бұл графикалық процедуралар мен функциялардың тізімі. Барлық аргументтер графикалық процедуралар мен функциялар үшін . тек бүтін типті бола алады. *Драйвер* – бұл ДК техникалық құрылғыларын басқаруды жүзеге асыратын арнайы программа. Графикалық режимді *инициализациялау* – бұл адаптер (экран) жұмысының мәтіндік режимнен графикалық режимге ауысуы.

Қазіргі ДК-де, негізінен, растрлық дисплейлер қолданылады. Олардағы бейненің ең кіші элементі болып нүкте – **pixel** (ағылш., picture element) табылады.

Дисплейдің мүмкіндігі – бұл көлденең және тік орналасқан пикселдер саны (стандартты разрешение – 640*480 нүкте).

Адаптерлердің барлық түрлері үшін графикалық драйверлер Borland International фирмасымен құрастырылған. Олар BGI (Borland Graphics Interface) кеңеймесімен жеке файлдарда орналасқан. Графикалық драйверді қосу үшін арнайы *initgraph()* функциясы қолданылады.

Драйверді және графикалық режимді таңдау
detectgraph (&gd, &gm)

функция арқылы орындалады.

2. Драйверді жүктеу, графикалық жүйені инициализациялауды
initgraph (&gd, &gm, "BGI-файлдың жолы")

функциясы атқарады, мұндағы *gd* және *gm* айнымалылары қажет драйвер мен графикалық режимнің номерлері.

Егер BGI-файлдар ағымдағы директорияда орналасса, онда *initgraph()* функциясының үшінші параметрі ретінде бос жолды беруге болады
initgraph (&gd, &gm, " ");

Режимдерді ауыстыру. Дисплейлік адаптердің текстік режиміне уақытша ауысу үшін мына функция қолданылады:
restorecrtmode ();

Қайтадан графикалық режимге ауысу
setgraphmode (gm);
функциясымен жүзеге асырылады.

Ағымдағы драйвер үшін графикалық режимнің максимал мәнінің номерін
getmaxmode ();
функциясы арқылы анықтауға болады.

Графикалық жүйемен жұмыс жасап болған соң, графикаға бөлінген жадыны босату, бейнеадаптер буферін тазарту, алдыңғы текстік режимді қалпына келтіру керек. Барлық әрекеттерді
closegraph ();
функциясы жүзеге ауыстырады.

5. Түстерді, шрифттерді сызықтардың стилін және бояу стилін тағайындау. Түстерді көрсету үшін констандаларды пайдалануға болады.

Әдебиеттер

1. Керниган Б., Ритчи Д.,- Язык программирования Си. – М.: Финансы и статистика, 1985 г.
2. Вирт Н. Алгоритмы и структуры данных: Пер. с англ.- М.: Мир, 1989 г.
3. Д.Кнут Искусство программирования для ЭВМ. Основные алгоритмы. М.: Мир, 1976. 3.
4. Н.Культин., С/С++ в задачах и примерах – БХБ-Петербург, 2002 г.
5. Крис Паппас., Программирование на С и С++ - ВНУ, Киев, 2000 г.
6. Т. А. Павловская, А. Ю. Щупак С/С++ Структурное программирование: Практикум- СПб.: Питер, 2005
7. Уэйт М . Язык Си. Руководство для начинающих: Пер. с англ. – М., Мир, 1988 г.

Практикалық жұмыс №10. Файлдар

Сабақтың мақсаты: C/C++ тілдерінде файлды құру, өңдеу, оқу, жазу дағдыларын қалыптастыру.

Материалдар және құрылғылар: тақта, бор.

Жұмыстың мазмұны және орындалу тәртібі:

1. Тапсырманы орындау мысалын қарау.
2. Ретімен келесі тапсырмаларды орындау.
3. Орындалған практикалық жұмыс туралы есепті дайындау.

Тапсырманы орындау мысалы

Бірнеше студент туралы ақпаратты файлда сақтауға мүмкіндік беретін программа құру. Кейін бұл ақпарат өңдеу үшін қайтадан оперативті жадыға оқыла алады.

```
#include <stdio.h>
struct STUDENT
{
char name[20];
int age;
};
void writeStudInfo(FILE *fp,struct STUDENT *st,int num)
{
int i;
for(i=0; i<num; i++)
fwrite(&st[i],sizeof(STUDENT),1,fp);
}
void readStudNum(FILE *fp,struct STUDENT *st,int n)
{
int a=fseek(fp,0L,0);
a=fseek(fp,(long)(n-1)*sizeof(STUDENT),0);
fread(st,sizeof(STUDENT),1,fp);
}
void readStudInfo(FILE *fp,struct STUDENT *st,int num)
{
int i;
for(i=0; i<num; i++)
fread(&st[i],sizeof(STUDENT),1,fp);
}
void main(void)
{
struct STUDENT st[5]={ {"name1",17},
{"name2",19},
```

```

{"name3",20},
{"name4",21},
{"name5",22} };
struct STUDENT stnew[5], stn;
FILE *fp=fopen("l9_2.dat","w");
writeStudInfo(fp,st,5);
fclose(fp);
fp=fopen("l9_2.dat","r");
readStudInfo(fp,stnew,5);
readStudNum(fp,&stn,2);
fclose(fp);
}

```

Студенттердің өз бетімен орындауға арналған тапсырмалары

1. Файл құратын программа жазыңыз, оның компоненттері келесі структурадан тұрады:

- Табель нөмірі
- ТАӘ
- Жалақы мөлшері

Файлға бес жұмысшы туралы мәлімет енгізіңіз, басқа файлға ең жоғары жалақы алатын жұмысшы туралы мәліметті енгізіңіз.

2. ZARPL.DAT файлын құратын және экранға шығаратын программа құрыңыз, оның компоненттері келесі структурадан тұрады:

- Табель нөмірі
- ТАӘ
- Жалақы мөлшері

Енгізуден шығу – табель нөмірі=999. Жалақысы 45000 теңгеден асатын жұмысшылардың табель нөмірін, ТАӘ және жалақысын экранға шығарыңыз.

3. Файл құратын программа жазыңыз, оның компоненттері келесі структурадан тұрады:

- Табель нөмірі
- аванс

Енгізуден шығу – табель нөмірі=999. Экранға келесі құрылымды ведомость шығарыңыз:

Табель нөмірі	аванс
.....
.....

Барлығы:

4. Бүтін сандар файлын құрыңыз. Файлдағы оң сандардың арифметикалық ортасын табыңыз.
5. Бүтін сандар файлын құрыңыз. Файлдағы ең кіші элементтің реттік нөмірін табыңыз. Егер мұндай сандар бірнешеу болса, онда біріншісінің нөмірін табыңыз.

6. Элементтері жеке символдар болып келген файл құрыңыз. Алғашқы екі символы цифр болып табыла ма, анықтаңыз. Егер цифр болса, онда осы цифрлардан құралған сан тақ бола ма, анықтаңыз.
7. Элементтері жеке сөздер болып келген файл құрыңыз. Реттік нөмірі тақ болып келген сөздерден сөйлем құраңыз.
8. Элементтері «киллобайт» сөзінің жеке әріптері болып келген файл құрыңыз. Қатесі жоқ жаңа файл құрыңыз.
9. Текстік файл берілген. 30-дан аса символы бар жолдарды басқа файлға жазу керек.
10. Текстік файл берілген. Барлық тақ нөмірлі жолдарды екінші файлға, ал жұп нөмірлі жолдарды үшінші файлға жазу керек. Жолдардың жалғасу реті сақталады.

Практикалық жұмыстың орындалуы туралы есептің формасы

Есепте болуы қажет:

Кез-келген орындалған тапсырма үшін

1. Есептің шешімінің математикалық моделі;
2. Есептің шешімінің алгоритмі;
3. Есеп шешімінің алгоритмі орындалған программа.

Блиц-тест

1. C/C++-те fopen функциясы не үшін арналған?
 - a) Файлды ашуға.
 - b) Файлды жабуға.
 - c) Файлды құруға және ағымдық каталогта уақытша файл ашуға.
 - d) Ашылған файлдан символды оқуға.
 - e) Графיקаны инициализациялауға.
2. C/C++-те fclose функциясы не үшін арналған?
 - a) Файлды ашуға.
 - b) Файлды жабуға.
 - c) Файлды құруға және ағымдық каталогта уақытша файл ашуға.
 - d) Ашылған файлдан символды оқуға.
 - e) Графיקаны инициализациялауға.
3. C/C++-те tmpfile функциясы не үшін арналған:
 - a) Файлды ашуға.
 - b) Файлды жабуға.
 - c) Файлды құруға және ағымдық каталогта уақытша файл ашуға.
 - d) Ашылған файлдан символды оқуға.
 - e) Графיקаны инициализациялауға.

Бақылау сұрақтары

1. Файл дегеніміз не? Файлдар қандай мақсатта қолданылады?
3. Файл атауларына қойылатын талаптар қандай?
4. Файлдармен қандай амалдар орындалады?
5. Файлдан мәліметтерді оқу үшін қандай функциялар қолданылады?

Глоссарий

С тілінде:

Файлдан оқу және файлға енгізу үшін ең алдымен файл fopen функциясының көмегімен ашылуы тиіс. Бұл функция операциялық жүйе арқылы орындалатын әрекеттерді ұйымдастыру жұмысын орындайды және файлмен мәлімет алмасуға арналған **көрсеткішті** қайтарады.

Ал, файлға көрсеткіш файл туралы информациялардан тұратын құрылымға (жазбаға) сілтейді. Мұндағы информация мынадай сұрақтарының жауабынан тұрады:

- буфер адресі,
- буфердегі ағымды литердің күйі,
- файлдан оқуға немесе жазуға ашық па?,
- файлдың соңғы таңбасы кездесті ме?

Мұндай құрылым сипаттамасы **<stdio.h>** кітапханасындағы FILE типінде беріледі. Қолдану үшін мынадай декларация берілсе жеткілікті:

FILE * fp;

FILE * fopen (char * name, char * m);

Мұндағы: fp – FILE типіндегі көрсеткіш, ал fopen FILE – ге көрсеткішті қайтарады. Fopen функциясы мына түрде қолданылады:

fp = fopen (name, m);

Мұндағы: name – файлдың атын меншіктейтін жол;

ал, m – файлды қолдану режимі; яғни бұл да жол, қолданушы файлды қалай қолданатынын білдіреді, төмендегідей мәндердің бірін иелене алады:

" r " – (read) оқу режимі;

" w " - (write) жазу;

" a " - (apprnd) толықтыру;

Кейбір жүйеде тексттік және бинарлық файлдар болып жіктеледі, бұл жағдайда режим жолына "b" (binary – бинарлық) немесе "t"(тексттік) таңбасы тіркеледі.

Файлмен жұмыс жасау барысында қате кездесе, онда fopen функциясы NULL мәнін қайтарады.

Мысалы,

FILE* fin, * fout;

fin=fopen ("PRIMER.dat","r");

fout=fopen ("RESULT.dat", "w");

1-ші жолда 2:fin,fout – файл көрсеткіші құрылады, ал төменгі жолдарда сәйкесінше оқуға және жазуға арналған файлдар ашылады.

Fopen() функциясы файлдың аталған көрсеткішін қабылдайды, программаның орындалу барысында олардың мәндері жасанды өзгертілмеуі тиіс. Мұндай файлдағы 2-ші параметр файлдарымен мәлімет алмасу режимін анықтайды. Файл мынадай режимдерде ашылуы мүмкін:

- тексттік;
- екілік;

Текстік режимде ашу үшін режимді көрсеткенде қасына «t» символын тіркеу арқылы жүзеге асады;

Режимдер кестесі:

Режим	Сипаты
a	Файл мәндерімен толықтырылуы үшін ашылады. Егер файл жоқ болса, ол құрылады. Жаңа мән соңына тіркеледі;
a+	Жоғарыдағы секілді, тек оқуға да болады;
r	Файлды тек оқу үшін ашады, файл жоқ болса, ашылмайды;
r+	Файлды оқуға да, жазуға да болады;
w	Жаңа файл ашады, бұрын бар болса, мәнін өшіреді.
w+	Жаңа файлды мән жазуға да, одан оқуға да мүмкіндік береді. Файл болса, мәні тазартылады.

R+,w+,a+ режимдерін қолдануда, яғни оған оқуды және жазуды бір уақытта орындалу барысында файл көрсеткішінің fsetpos(), fseek(), немесе rewind() файлдары көмегімен ағымды позицияларын модификациялау керек.

Порграмма жұмысы аяқталғанда Си-де автоматты түрде барлық ашық файлдар жабылады, ал файлды жабу үшін fclose(fin) қолданылады.

C++ тілінде:

Файлмен мәлімет алмасу мақсатында ifstream және ofstream кластарын қолданған тиімді. Ол үшін < **fstream. h** > тақырыптық файлы ашылуы тиіс.

Мұнда, сәйкесінше, **ifstream** класының объектісі **myof** және **ostream** класының **mygf** лектері құрылып көрсетілген файлмен байланысады.

Мысалы, ifstream myof(“prim.in”,ios::in);

Ofstream mygf(“prim.out”,ios::out);

- Бір объектімен тізбектеп бірнеше лекті байланыстыру қажет болса, онда ол төмендегідей ретпен орындалады:

```
ifstream myof;
```

```
...  
myof(“prim.in”);
```

```
...  
myof.close();  
myof.open (“prim2.in”);
```

```
...  
myof.close();
```

- Файлмен мәлімет алмасу мүмкіндігінің режимін өзгерту – файлдық объектінің сипатындағы екінші аргументін өзгерту арқылы жүзеге асады.

Мысалы, ofstream myof (“prim.in”,ios::app | ios::nocreate); - мұндағы файл бар болса ғана myof объектісі құрылып, байланыс орнатылады, себебі **ios::nocreate** аргументі берілген, яғни файл болмаса, объект құрылмайтынын білдіреді. Ал,

ios::app аргументі файлға шығарылатын мәндер файлдың соңына қосылатынын анықтайды.

Флаг-аргументтер биттік | - «немесе» амалының көмегімен біріктіріліп анықтала алады.

Файлдық лекті анықтау барысында қолданылатын аргументтерді анықтайтын флагтар төмендегідей:

флаг	Мағынасы
ios::in	Файл оқуға ашылады
ios::out	Файл жазуға ашылады
ios::ate	Объект құрылған соң, ағымды көрсеткіш файлдың соңына орнатылады
ios::app	Мәндер файлдың соңына жазылады
ios::trunc	Егер файл бар болса, ол тазартылады
ios::nocreate	Файл болмаса, онда объект құрылмайды
ios::noreplace	Файл бар болса, онда объект құрылмайды
ios::binary	Файл екілік режимде ашылады (қалыпты жағдайда текстік режим)

feof (FILE * fp) – функциясы файлдың соңын анықтайды.

Файлды жабу үшін:

Int fclose(FILE *fp) функциясы қолданылады, яғни файлдық көрсеткішпен орнатылған байланысты үзеді.

- Файлмен мәлімет алмасу үшін **fstream** класының объектісі де қолданылады:
Fstream io(“update.dat”, ios::in|ios::app); - бұл жағдайда көрсетілген файл мәліметті оқуға және жазуға ашылады.
- **Seekg()** файлан оқу ал, **seekp()** файлға жазу барысында функциялары файлдың ағымды позициясының маркерін басқаруға қолданылады.

Мысалы,

io.seekg(5, ios::cur) – оқу барысында маркерді ағымды орынынан 5 байтқа жылжыту;

io.seekg(-7, ios::end); - маркерді соңынан бастап 7 байтқа жылжыту керектігін білдіреді.

Аргументтер:

Io::cur – ағымды орынынан; **Io::beg** – **файл басынан**;

Io::end - файл соңынан жылжытылатынын білдіреді.

- *Файлдық лектің күйін анықтау функциялары:*

Функция	Түсініктеме
eof()	Файл соңын кездестірсе, 0-дік емес мән қайтарады
fail()	Лекте қате кездессе 0- ден өзге мән қайтарады
bad()	Енгізу-шығару барысында мағыналы қате кетсе, нөлден өзге мән қайтарады, онда лекпен жұмыс тоқтатылғаны дұрыс. Лек күйінің биттік мәндері анықталмаса, 0-ге тең емес мән береді.
rdstate()	Мына тұрақтылардың бірін, яғни лектің ағымды күйін анықтайтын мәнді қайтарады:

	Ios::goodbit - қате жоқ Ios::eofbit – файл соңына жетті Ios::failbit – форматтау немесе түрлендіру қатесі Ios::badbit - маңызды қате
clear()	0-ге тең болса тазарту

12.2. Файлдық енгізу-шығару

Файлдан оқығанда Pascal-дағы секілді,

```
fprintf(fin, 'формат', <айнымалы>);
```

```
fscanf(fin, 'формат', <айнымалы>);
```

fseek() функциясы:

Жалпы форматы:

Int fseek (FILE *str, long offset, int orig)

M, fseek (f, K, r)

Fseek функциясы f Көрсеткіші арқылы берілген файлдағы ағымды көрсеткішті K байтқа жылжытады. Жылжыту:

- файлдың басынан (r=0)
- ағымды көрсеткіш тұрған (r=1)
- файлдың аяғынан (r=1) басталуы мүмкін.

Си- де r-дің мәні ретінде көрсету үшін қолданылатын 3 const қарастырылған:

SEEK_SET (файл басынан жылжу)

SEEK_CUR (ағымдағы позициядан бастау)

SEEK_END (файл соңынан)

Fseek() функциясы қызметі сәтті аяқталса 0-ді, әйтпесе EOF мәнін қайтарады.

- Ftell()-функциясы файл басынан бастап ығысқан шамамен анықталады, яғни long типті мән қайтарады, ағымды көрсеткіштің рынын анықтайды.

Ж.Т: long ftell(FILE*str)

Ftell(f)

- Rewind функциясы ағымды көрсеткішті файлдың басынан орындайды.
rewind(FILE*str);
rewind(f) мынаған сәйкес мәнді қайтарады;
fseek(f, ol, seek_set);

Әдебиеттер

1. Керниган Б., Ритчи Д.,- Язык программирования Си. – М.: Финансы и статистика, 1985 г.
2. Вирт Н. Алгоритмы и структуры данных: Пер. с англ.- М.: Мир, 1989 г.
3. Д.Кнут Искусство программирования для ЭВМ. Основные алгоритмы. М.: Мир, 1976. 3.
4. Н.Культин., С/С++ в задачах и примерах – БХБ-Петербург, 2002 г.
5. Крис Паппас., Программирование на С и С++ - ВНУ, Киев, 2000 г.
6. Т. А. Павловская, А. Ю. Щупак С/С++ Структурное программирование: Практикум- СПб.: Питер, 2005
7. Уэйт М . Язык Си. Руководство для начинающих: Пер. с англ. – М., Мир, 1988 г.

Қазақстан Республикасы Білім және Ғылым министрлігі
Қ.Жұбанов атындағы Ақтөбе мемелететтік университеті

«050704-ЕТ және бағдарламалық қамтамасыз ету» мамандығы
күндізгі бөлімінің 1 курс студенттеріне арналған
«Программалау технологиялары» пәнінен

ЛАБОРАТОРИЯЛЫҚ ЖҰМЫСТАРДЫ ОРЫНДАУҒА АРНАЛҒАН
ӘДІСТЕМЕЛІК НҰСҚАУЛАР

Ақтөбе, 2011

I. ЛАБОРАТОРИЯЛЫҚ ЖҰМЫСТАРҒА АРНАЛҒАН ӘДІСТЕМЕЛІК НҮСҚАУДЫ ҚҰРАСТЫРҒАН ЖӘНЕ ЕНГІЗГЕН

Орындаушы

Информатика және ЕТ кафедрасының
Доценті, п.ғ.к.

_____ Байбақтина А.Т.

«__» _____ 2011 ж.

Жауапты орындаушы

Информатика және ЕТ кафедра
меңгерушісі

_____ Сарсимбаева С.М.

«__» _____ 2011 ж.

II. КАФЕДРАНЫҢ ӘДІСТЕМЕЛІК СЕКЦИЯСЫНЫҢ ОТЫРЫС БАСЫНДА ТАЛҚЫЛАНДЫ

Хаттама № __ «__» _____ 2011 ж.

III. СЫН-ПІКІР БЕРУШІ

_____ АЖ кафедрасының аға оқытушысы _____ Жахина Р.У.

«__» _____ 2011 ж.

IV. ФИЗИКА-МАТЕМАТИКА ФАКУЛЬТЕТІ ӘДІСТЕМЕЛІК КОМИССИЯСЫНА ҰСЫНЫЛДЫ

Хаттама № __ «__» _____ 20__ ж.

V. Қ.ЖҰБАНОВ АТЫНДАҒЫ АМУ-НІҢ ОҚУ-ӘДІСТЕМЕЛІК СЕКЦИЯСЫНЫҢ ОТЫРЫСЫНДА МАҚҰЛДАНҒАН

Хаттама № __ «__» _____ 20__ ж.

VI. АЛҒАШҚЫ ТЕКСЕРУ МЕРЗІМІ

2012 жыл

ҚАЙТАЛАП ТЕКСЕРУ МЕРЗІМІ

2 жыл

ЛАБОРАТОРИЯЛЫҚ ЖҰМЫС №1

Тақырыбы: C/C ++ ТІЛДЕРІНДЕ ПРОГРАММАЛАУ ОРТАСЫНЫҢ НЕГІЗГІ КОМПОНЕНТТЕРІ

Мақсаты: - IDE (interated development environment) біріктірілген құралдар ортасының жұмысын түсіндіру;
- оның негізгі компоненттерімен таныстыру;
- C/C ++ тіліндегі программаның өңделуін: енгізу, тестілеу, компиляциялау реттерін үйрету.

Қажетті материалдар мен жабдықтар: ДК, C/C++ тілінің IDE ортасы, лабораториялық жұмысты орындауға арналған әдістемелік нұсқаулар.

Лабораториялық жұмыстың мазмұны және орындалу реті:

1. C/C ++ тілінде программалау ортасының негізгі компоненттерімен танысу және жұмысын меңгеру.
2. Ұсынылған мысалды орындау және талқылау.
3. Қарастырылған жеке тапсырмалардың оқытушы ұсынған нұсқасын орындау.
4. Лабораториялық жұмыстың есебін (отчет) дайындап тапсыру.

Негізгі әдістемелік нұсқаулар:

Программалау ортасы – IDE–біріктірілген программалау ортасы – құралдар жиынтығынан тұрады. Ол құралдар: программаны жазуға, өңдеуге, компиляциялауға, құрастыруға (компановка) және жөндеуге (отладка) арналған.

Сонымен қатар, ортада мынадай қосымша мүмкіндіктері арастырылған:

- Монитор экранында бірнеше терезені бейнелеп және оның өлшемін қажетінше өзгертіп, орнынан жылжытып қарауға болады;
- «Мышь» манипуляторын қолдану;
- Диалогтар блогы бар;
- Текст фрагментін өшіру және орналастыру мүмкіндігі бар;
- Басқа программаларды шақырып қолдану мүмкіндігі;
- Редакторда макротіл қолданылады.

IDE ортасының негізгі 3 компоненті бар: C/C ++тілдерінің кең тарағаны turbo C++ (Borland фирмасынан шыққан) ортасына қатысты сипаттау беріледі:

- Экранның жоғарғы бөліміндегі менюлер жолы;
- Экранның ортасындағы жұмысшы алаңы;

- Экранның төменгі бөлігінде қалыпты-күй жолы.

Менюлер жолы:

Менюден кейбір элементті таңдағанда диалогты блоктар экранға шығарылады. Менюлер жолы менюдегі барлық командаларды қолдану жолын алады. Ондағы қажетті менюді мышпен шерту немесе F10 тетігін басып қажетіне жылжып бару арқылы екпінді етуге болады. Екпінді менюден басқару тетіктері мен Enter пернесін басу арқылы команда таңдалады.

Команда таңдалғанда ішкі меню (команда қасында үшбұрышты символ болса) немесе диалогты блок (команда қасында көп нүкте тұрса) ашылуы мүмкін, ал команда қасында ештеңе болмаса ол тікелей орындала бастайды.

Диалогты блокта параметрлерді анықтау мынадай негізгі басқару құралдарының: таңдау көрсеткіштері, күйін ауыстырғыштар, әрекеттер түймесі, енгізу блогы, тізімдер блогы арқылы жүзеге асырылады.

Жүйеде меню екі түрде қатысуы мүмкін: толық немесе азайтылған. Мұның қажетін орындау үшін:

Options\Full Menus командасы қолданылады, сәйкесінше OFF немесе ON күйлері орнатылады.

Жүйеде кез-келген әрекет нақты бір терезеде орындалады. Бір уақытта бір терезе екпінді бола алады, бірақ бірнеше терезе ашық болуы мүмкін. Терезенің бірнеше түрі бар, оның әрқайсысының мынадай элементтері болады:

- тақырыптар жолы;
- терезені жабу маркері;
- айналдырғыш сызғыш;
- терезе өлшемін өзгерту бұрышы;
- экранға терезені толығымен ашу маркері;
- терезе мөлшері (1-ден 9-ға дейін).

Қалыпты күй жолы:

Экранның төменгі бөлігінде орналасқан. Төмендегідей функцияларды атқарады:

- Екпінді терезедегі орындалып отырған әрекетке байланысты қолдануға болатын негізгі пернелер туралы мәлімет беріп отырады;
- Нақты көрсетіліп тұрған әрекетті орындау үшін менюден команданы таңдаудың орнына перне белгісіне мышь Көрсеткішін қойып, сосын шерту арқылы жеңілдету мүмкіндігі бар.
- Программада қандай әрекет орындалып отырғандың хабарлайды.
- Менюден таңдалған команда немесе диалогты блок элементтері туралы қосымша бір жолдық кеңестер мен ұсыныстар береді.

Терезенің жүйелік менюі:

ALT – пробел пернелер комбинациясын басу арқылы терезенің сол жақ жоғарғы бұрышында орналасқан жүйелік меню ашылады. Мұнда төмендегідей командалар орналасқан:

About - бұл команда таңдалғанда диалогты блок ашылады, мұнда TURBO C++ версиясы туралы және авторлық құқығы жайында информация беріледі. Бұл терезе ESC немесе ENTER пернелерін басу арқылы жабылады.

Clear Desktop - командасы арқылы ашық терезелердің бәрі жабылып, тізімдегі барлық анықтама өшіріледі. Жаңа проект құру барысында қолданған тиімді.

Repaint Desktop - экрандағы бейнені регенерациялауды жүзеге асырады.

Transfer ішкі менюінің элементтері:

Options/Transfer командасының көмегімен шақырылып ашылған диалогты терезеден анықталып, орнатылған барлық программалардың аттары көрсетіледі. Қажет программаның атын жүйелік менюден таңдап жүктеу керек.

File(ALT F) менюі терезелерге программаларды ашуға, енгізуге және өңдеп сақтауға, сонымен қатар файлдарға әрекет қолдануға және DOS операциялық жүйесі ортасына уақытша көшіп жұмыс жасауға, TURBO C++ ортасының жұмысын аяқтауға мүмкіндік береді.

FILE\OPEN командасын таңдағанда диалогта блок ашылады да, мұнда ашылатын файл көрсетіледі.

Ашу үшін мынадай әрекеттің бірі орындалады:

- 1) Файлдың толық атын көрсетіп, REPLACE немесе OPEN түймесінің бірін басу керек.

Open түймесі басылса, онда файл Edit редакторының жаңа терезесінде ашылады, ал Replace түймесін басу үшін екпінді терезе болуы тиіс және нәтижесінде таңдалған файл сол терезедегі программа орынына ашылады.

- 2) Файл аттарын метасимволдармен беру. Бұл жағдайда файлдар тізімін спецификациясына байланысты елеп алуға болады.

Бұған дейін енгізілген файлдар спецификациясы тізімінен таңдау үшін ? таңбасын қолдану керек.

- 3) Басқа файлдар мазмұнын қарауға болады, ол үшін файлдар тізімінен басқа каталог атын таңдау қажет.

Енгізу блогында файл атын нақты немесе шаблондар арқылы енгізуге болады, нәтижесінде файл бар болса ашады, әйтпесе сол атпен жаңа файл құрады. Курсор

енгізу блогында тұрған сәтте ? пернесі басылса, онда бұған дейін қолданылған сегіз файлдың тізімі беріледі.

File\ New командасы NONAMExx. C (xx дегеніміз 00 -ден 99-ға дейінгі сан). NONAME атауымен аталған файл редакциялауға арналған уақытша буфер болып табылады, файлды сақтарда файлға нақты атау беріледі.

File\ Save командасы редактордың екпінді терезесіндегі файлды дискіге жазады. Егер нақты атауы болмаса (NONAMEOO.Cжәне т.с.с.) TurboC++ Save Editor File диалогты блогын ашады да, сол блокта атауы мен сақталатын орыны көрсетілуі тиіс.

File\Save As... екпінді терезедегі файлды басқа атпен сақтауға мүмкіндік береді.

File\Change Dir жұмысшы каталогты өзгерту, яғни атауы мен жолын көрсетуге мүмкіндік береді. Жұмысшы каталог дегеніміз - Turbo C++ файлды сақтағанда немесе іздегенде қарайтын каталогы.

File\Print Turbo C++ - дегі екпінді терезенің мазмұнын баспаға шығарады. Текстің тек белгіленген бөлігін Ctrl-K P пернелер комбинациясын басу арқылы баспаға шығаруға болады.

File\Get Info командасы ағымдағы каталог туралы информация блогынан тұрады. Get Info блогының мәндері кестесі:

МӘНІ	МАҒЫНАСЫ
Current directory	- Каталогтың қалыпты аты
Current file	- Екпінді терезедегі файл аты
Extended memory usage	- Turbo C++ -ге бөлінген қосымша жады мөлшері
Expanded memory usage	- Turbo C++ - ге арналған кеңейтілген жады мөлшері
Lines compiled	- Компиляцияланған жолдар саны
Total warnings	- Жүйенің ескертпе хабарламаларының саны
Totals errors	- Генерацияланған қателер саны.
Total time	- Программаның соңғы рет орындалған уақыты.
Program loaded	- Өңдеу статусы
Program exit	- Соңғы аяқалған программадан қайту коды.
Available memory	- Қолданылатын жады көлемі DOS (640 K)
Last step time	- Өңдеу барысындағы соңғы қадамның орындалған уақыты

File\ DOS Shell Turbo C++ ортасынан уақытша шығып DOS командаларын орындауға мүмкіндік береді. Turbo C++ ортасына оралу үшін клавиатурадан EXIT командасын енгізіп Enter басу керек.

Берілген команданы орындау барысында жады мөлшері жетпей қалса, онда **Run\Program Reset** (Ctrl-F2) командасының көмегімен өңдеуді тоқтату қажет.

File\Quit командасы Turbo C++ жүйесінен шығып, оны жадыдан өшіріп, басқару билігін DOS ОЖ-не береді. Егер терезелерде сақталмаған өзгертулер жасалған болса, оларды сақтау туралы сұранымдар жасалады.

Edit(Alt-E) менюі:

Редактор терезелеріндегі текстке өзгертулер, енгізулер, өшірулер жүргізеді. Сонымен қатар тексттік буфер мазмұнын қарауға да болады. Текстті белгілеу дегеніміз - оны өзге түске бояу. Ол үшін:

1) Shift пернесін стрелкалармен бірге басу;

2) Ctrl-K B пернелерін қатар басу арқылы белгіленетін текст блогының басын көрсету, сосын курсорды белгіленетін блок соңына қойып Ctrl-K K басу керек.

3) Жолды белгілеу үшін Ctrl-K L басу қажет.

Фрагмент белгіленген соң ғана Edit менюіндегі командалар қолданылады және Clipboard буферін пайдаланады.

Restore Line командасы соңғы орындалған команданы кері қайтарады.

Cut (Shift-Del) командасы белгіленген фрагментті құжат текстінен өшіреді де оны буферге орналастырады.

Copy (Ctrl-Ins) командасы белгіленген фрагменттің көшірмесін буферге орналастырады. Оны Paste командасының көмегімен басқа текстке қоюға болады. Сол секілді Help терезесінен фрагмент алуға болады.

Paste (Shift-Ins) командасы буфердегі текстті жұмысшы терезенің курсоры тұрған жеріне орналастырады.

Show Clipboard командасы басқа терезелерден көшірілген және өшірілген текст фрагменттері орналасқан Clipboard терезесін ашады.

Clear (Ctrl-Del) командасы тексттегі белгіленген фрагментті өшіреді.

Search (Alt-S) менюі:

Search менюі текстті іздеу, функцияны сипаттау және файлдағы қате жіберілген орында іздеу әрекеттерін орындайды.

Search\Find (Ctrl Q F) командасы Find диалогты блогын ашады, мұнда іздеу үлгілерін және параметрлерін беруге болады.

Replace (Ctrl Q A):

Search Replace командасы тексттегі ізделінді фрагментті және оның орынына қойылуға тиіс фрагментті көрсетуге мүмкіндік беретін диалогты терезені ашады.

Search\Again (Ctrl L) командасы Find немесе Replace командаларының соңғы әрекетін қайталайды. Мұнда соңғы орнатылған параметрлері қолданылады.

Run (Alt-R) менюі:

Бұл меню командалары программаны орындайды және өңдеу сеансын инициализациялайды, аяқтайды.

Run(Ctrl-F9)

Run командасы Run Arguments командасы көмегімен берілген аргументтерді пайдаланып программаны орындайды.

Trace Into(F7) командасы программаны операторлап орындайды. Және функция шақырылса, оны толығымен бірден орындаудың орынына оның да операторларын біртіндеп орындайды.

Run\Program Reset(Ctrl-F2) командасы программаның ағымды өңдеу сеансын тоқтатады, программа жадысын тазалайды және программада қолданылған барлық файлдарды жабады.

Run\Step Over командасы ағымды функциядағы келесі операторды орындайды, Step Over командасын функцияны операторлап орындату барысында басқа қолданылатын функцияларға кірмей орындату керек болған жағдайда пайдаланыңыз.

Run\Arguments командасы орындалатын программада командалық жол аргументтерін DOS ОЖ-де қандай түрде берілсе, сондай түрде анықтауға мүмкіндік береді. Мұнда DOS енгізу-алу командалары қарастырылмайды.

Compile (C) менюі командалары екпінді терезедегі программаны және проектін толығымен немесе таңдап компиляциялауға қолданылады.

Compile\ Make EXE File - EXE-файл құруға арналған проектилер менеджерін шақырады.

Compile\Link EXE File командасы ағымды OBJ и LIB-файлдарын қолданады және оларды и жинақтайды.

Debug (Alt F9) менюі командалары біріктірілген өңдегіштің барлық мүмкіндіктерін басқарады.

Debug Inspect (Alt F4) командасы арқылы мәліметтер элементтерін талдап, жөндеуге мүмкіндік беретін Inspector терезесін ашуға болады.

Options(Alt-O) менюі командалары Turbo C++ ортасының негізгі функцияларының стандартты параметрлерін қарауға және өңдеуге мүмкіндік беретін командалардан тұрады.

Көп файлдық проекттерді басқару

Қазіргі программалау технологиясы бойынша бір тұтас программалық жабдық – паралелль дайындалған бірнеше программалық файлдардан тұрады. Сондықтан бірнеше файлан проект құру мүмкіндігі болуы керек. Бұл үшін C ++ ортаснда қосалқы **проектилер менеджері (ПМ)** қолданылады. ПМ құрылатын

проектіні құрайтын файлдарды анықтайды және проект рекомпиляцияланғанда проект файлында сақталған информацияны жаңартады. Проект файлында мынадай информациялар сақталады:

- проектке енетін барлық файл аттары;
- олардың дискідегі маршруттар;
- алдымен компиляцияланатын файлдармен олардың басқа файлдармен байланысы;
- программаның әрбір бөлігінің құрылуы барысында қажетті компиляторлар түрлері мен командалық жолдарының параметрлері анықталады;
- нәтижелік программа қайда орналастырылатыны;
- соңғы рет компиляцияланғанда анықталған код өлшемі, мәліметтер өлшемі мен жолдар саны.

Проект құру үшін мынадай әрекеттер орындалады:

- 1) Проект файл атын таңдау, ол үшін: **Project / Open Project** командасы орындалады;
- 2) Проектіге құрамына енетін файл аттарын қосу, яғни **Project / Add Item** командасын таңдау арқылы;
- 3) Жүйеге файлды компиляциялау командасын беру, ол үшін: **compile\Make EXE** немесе F9 басылады;
- 4) Соңынан программаны орындауға жіберу керек, яғни **Run\Run** немесе CTRL+F9 пернелер комбинациясы басылады.

Кейін де мынадай әрекеттерді орындауға болады:

- Проектіге тағы да файлдар қосып немесе алып тастау;
- Проектіге енгізілген файлдарға қатысты параметрлер беруге;
- Проектіге нақты бір файлға енгізілетін файлдар мазмұнын қарау.

Өзіндік жұмыс т а п с ы р м алары:

Қарапайым программа енгізіп, оны еске сақтап, оның текстін арнайы командалар арқылы редакциялап көріңіз. Turbo C++ ортасының қызыметін толығымен меңгеруге арналған мысалдар қараңыз.

Блиц- тест:

1. C++ тілінің IDE ортасында төмендегі компоненттердің қайсысы жоқ?

- a. қалыпты күй жолы.
- b. менюлар жолы.
- c. саймандар тақтасы.
- d. терезе.

2. C++ тілінің **IDE** ортасында бір терезелердің бірінен біріне көшу үшін мына перделер комбинациясын басу керек:

- a. shift + a
- b. ctrl+a
- c. alt+1
- d. alt+терезе нөмірі
- e. ctrl+ терезе нөмірі

3. C программалау ортасында текст блогын өшіріп тастау үшін қандай пернелер комбинациясы қолданылады?:

- a. Ctrl+del
- b. Ctrl+shift**
- c. Ctrl+ins
- d. Ctrl+end
- e. Shift+del

C программалау ортасында текст блогын өшіріп тастау үшін қандай пернелер комбинациясы қолданылады?:

- a. Ctrl+del
- b. Ctrl+shift
- c. Ctrl+ins
- d. Ctrl+end
- e. Shift+del

Бақылау сұрақтары:

1. Turbo C++ ортасының негізгі элементтері қандай?
2. Диалогты блок дегеніміз не?
3. Файлды ашу диалогты блогын қандай командамен ашамыз?
4. Екпінді терезені қандай команданың көмегімен экранға толық жайуға болады?
5. Change Dir командасы не үшін қолданылады?
6. Ашық тұрған терезелердің бірінен біріне қалай өтуге болады?
7. Программаны қадамдап орындау қалай жүзеге асырылады?
8. Inspect командасының қызметі қандай?
9. Программа текстінің белгіленген фрагментін экраннан өшіріп және оны буферге орналастырмайтын команда қандай?
10. Arguments командасының қызметі қандай?
11. Trace Into командасының қызметі қандай?
12. Белгіленген текст фрагментін буферге көшіру командаы қандай?

13. Алдыңғы қатеге көшу командасы қандай?
14. программаны курсор орналасқан жерге дейін орындау командасы қандай?
15. DOS Shell командасының қызметі?

Глоссарий:

- a. **IDE–біріктірілген программалау ортасы** - программаны жазуға, өңдеуге, компиляциялауға, құрастыруға (компиляция) және жөндеуге (отладка) арналған құралдар жиынтығынан тұрады.
- b. **Менюлер жолы** – программалау ортасында орындалатын әрекеттерге қатысты топтастырылған командалар жиынтығы.
- c. **Қалыпты күй жолы**- ортадағы іс-әрекеттерді түсіндіру жолы.
- d. **Терезенің жүйелік менюі** – программалау ортасындағы терезелерді басқару командаларынан тұрады.
- e. **Проектілер менеджері** - құрылатын проектінінің құрамына енетін файлдарды анықтайды және проект рекомпиляцияланғанда проект файлында сақталған информацияны жаңартады.

Әдебиеттер:

[1],[5],[3],[14].

ЛАБОРАТОРИЯЛЫҚ ЖҰМЫС №2

Тақырыбы: C/C++ СЫЗЫҚТЫҚ АЛГОРИТМДЕРДІ ПРОГРАММАЛАУ

Мақсаты: C/C++ тіліндегі программаның жалпы құрылымын көрсету және сызықтық программау элементтерін бекіту;
Енгізу-шығару мүмкіндіктерінің ерекшелігін көрсету.

Қажетті материалдар мен жабдықтар: ДК, C/C++ тілінің IDE ортасы, лабораториялық жұмысты орындауға арналған әдістемелік нұсқаулар.

Лабораториялық жұмыстың мазмұны және орындалу реті:

1. C/C ++ тілінде сызықтық алгоритмдерді программалауға қатысты әдістемелік нұсқау – материалдарды алдын-ала танысып меңгеру.
2. Ұсынылған мысалды орындау және талқылау.
3. Өзіндік жеке тапсырмалардың оқытушы ұсынған нұсқасын орындау.
4. Лабораториялық жұмыстың есебін (отчет) дайындап тапсыру.

Әдістемелік нұсқау

Программаның құрылымы:

Си/Си++ тіліндегі программалардың жалпы құрылымы:

```
Препроцессор_директивалары  
1_функциясының_анықталуы  
2_функциясының_анықталуы  
.....  
N_функциясының_анықталуы
```

Функциялардың ішінде міндетті түрде *main* атты негізгі функция болуы қажет. Қарапайым программа тек негізгі функциядан тұрады және оның құрылымы мынадай:

```
Препроцессор_директивалары  
void main()  
{объектілердің_анықтамалары;  
орындалатын_операторлар;  
}
```

void сөзі *main* негізгі функциясының ешқандай мән қайтармайтынын білдіреді, ал бос жақшалар оның аргументтерінің жоқтығын көрсетеді. Функцияның тақырыбынан кейін, фигуралық жақшалардың ішінде функцияның денесі орналасқан.

Нүктелі үтірмен аяқталған кез-келген өрнек компиляторда жеке оператор ретінде қабылданады. Программадағы

```
i++
```

конструкциясы да оператор болып табылады. Оны *оператор-өрнек* деп атайды. Егер өрнектің есептелуі меншіктеумен аяқталса, онда оны *меншіктеу операторы* деп атауға болады.

Экранға мәліметтерді форматтап шығару. `printf()` функциясын шақыру операторы келесі құрылымға ие:

```
printf(формат түріндегі жол, аргументтер_тізімі);
```

Форматты жол екі тырнақшамен шектелген (яғни, тексттік константа болып табылады) және өзінің құрамына *кез-келген текстті, басқару символдарын және формат спецификаторларын* қамти алады. Аргументтер тізімі келтірілмеуі мүмкін немесе мәндері экранға шығарылатын өрнектерден тұра алады. Мысалы, `printf("\na=");` операторы ("a=") текстін және (\n) басқару символын қамтиды. Текст жазылған түрінде экранға шығарылады. Басқарушы символдар шығарылатын белгілердің экранда орналасуына әсер етеді. Басқарушы символдардың белгісі ретінде \ беріледі. Төменде олардың тізімі келтірілген:

\n – жаңа жолға көшіру;

\t – горизонталды табуляция;

\r – курсорды жаңа жолдың басына қайтару;

\a – қоңырау дыбысы;

\b – бір символ кері қайтару (бір позиция);

\f – жаңа параққа көшу;

\v - вертикалды табуляция.

`printf("\nүшбұрыш ауданы=%f",s);` операторын қарастырайық. Аргументтер тізімі бір *s* айнымалысынан тұрады. Оның мәні экранға шығарылады. %f символдар жұбы шығарылатын *s* айнымалысының мән форматының спецификациясы болып табылады. % - формат белгісі, ал *f* әрпі шығарылатын санның нақты типті екенін көрсетеді.

Формат спецификаторы шығарылатын өлшембірліктің сырт көрінісінің түрін анықтайды. Кейбір формат спецификаторларын келтірейік:

%c – символ;

%s – жол;

%d – бүтін ондық сан (типі int);

%u – белгісі жоқ бүтін ондық сан (типі unsigned);

%f – бекітілген нүкте түріндегі нақты сандар;

%e – қалқымалы нүкте түріндегі нақты сандар (мантиссамен, ретпен берілген)

Мысалы, келесі операторлардың орындалуынан кейін
`float m, p;`

```
int k;  
m=84.3; k=-12; p=32.15;  
printf("\nm=%f\tk=%d\tp=%e", m, k, p);
```

экранға мына жол шығады:

```
m=84.299999      k=-12      p=3.21500e+01
```

Формат спецификаторына сандық параметрлер қосыла алады: жолдың ені мен дәлдігі. Ені - өлшемге экранда берілетін позициялар саны, ал дәлдік – бөлшек бөлігіне берілетін позициялар саны (нүктеден кейін). Параметрлер % пен формат символының арасында жазылады және бір бірінен нүкте арқылы ажыратылады.

Мысалы,

```
printf("\nm=%5,2 ", m); нәтижесінде экранға  
m=84.30 шығады.
```

Көрсетілген жолдың еніне шығарылатын мән симай қалған жағдайда да, өлшем толық шығарылады.

Формат спецификаторларына келесі түрдегі модификаторлар да қосылуы мүмкін:

%ld – long int шығару;

%hu – short unsigned шығару;

%lf – long double шығару.

Клавиатурадан форматты түрде енгізу. scanf() функциясын шақыру операторы келесі құрылымға ие:

scanf(формат түріндегі жол, аргументтер_тізімі);

Бұл функция клавиатурадан енгізілген символдардың оқылуын қамтамасыз етеді. scanf() функциясында формат жолы мен аргументтер тізімі міндетті түрде болуы қажет. Келесі операторды қарастырайық:

```
scanf("%f",&a);
```

Мұндағы "%f" - формат жолы, &a – енізілетін мәндерге сәйкес аргументтер тізімі. Бұл оператор а айнымалысына сандық мәннің енгізілуін қамтамасыз етеді.

Клавиатурадан енгізілетін және scanf() функциясыменен қабылданатын символдық тізбекті *кіріс ағымы* деп атайды. Аргументтер тізімі – енгізілетін айнымалылар және әр айнымалының алдына & белгісі қойылады.

Формат жолы тырнақшаға алынады және спецификациялар тізімінен тұрады. Әр спецификация % белгісінен басталады, одан кейін

*жолдың_ені модификатор спецификатор

тұруы мүмкін. Соның ішінде спецификатор міндетті элемент болып табылады. * белгісі кіріс ағымында белгілі бір символдар санын жіберуге мүмкіндік береді.

Өртүрлі мәндерді енгізу ағымында ажыратушы ретінде пробелдердің, табуляция белгісінің кез-келген саны бола алады. Тек **Enter**'ді басқаннан кейін ғана енгізілген мәндер сәйкес айнымалыларға меншіктеледі.

Мысалы, 5 3.2 2.4 <Enter>

Си++ тіліндегі ағымдық енгізу-шығару. Си++ тілінде программа құру барысында, stdio.h тақырыптық файлының көмегімен қосылатын Си стандарт кітапханасының енгізу-шығару құрылғыларын пайдалануға болады, бірақ Си++

тің өз енгізу-шығару құрылғылары бар. Ол кластар кітапханасы және программаға `iostream.h` файлының көмегімен қосылады. Бұл кітапханада объектілер ретінде келесі есімді стандартты символдық ағымдар анықталған:

`cin` – клавиатурадан енгізу стандартты ағымы;

`cout` – экранға шығару стандартты ағымы.

Мәліметтердің енгізілуі `cin` ағымынан шығарылуы және сәйкес айнымалыларға мәндердің меншіктелуі деп интерпретацияланады. Си++ стандартты ағымнан шығару операциясы ретінде `>>` белгісі анықталған. Мысалы, `x` айнымалысына мән енгізу операторы мына түрде беріледі:

```
cin>>x;
```

Мәліметтердің экранға шығарылуы шығарылатын мәндерді `cout` стандарт ағымына орналастыру ретінде интерпретацияланады. Ағымға орналастыру белгісі `<<`. Ағымдық шығаруды пайдаланудың үлгілері:

```
cout<<a+b;
```

```
cout<<"\nНәтиже="<<Y;
```

```
cout<<"x="<<x<<" y="<<y<<" z="<<z<<endl;
```

Келтірілген мысалдан `printf()` функциясындағыдай басқару символдарын пайдалануға болатыны және әр шығару элементінің алдында `<<` операция белгісін қою қажеттілігі көрінеді. `endl` шығару элементі курсорды жаңа жолға көшіруді білдіретін манипулятор болып табылады (`\n` аналогі).

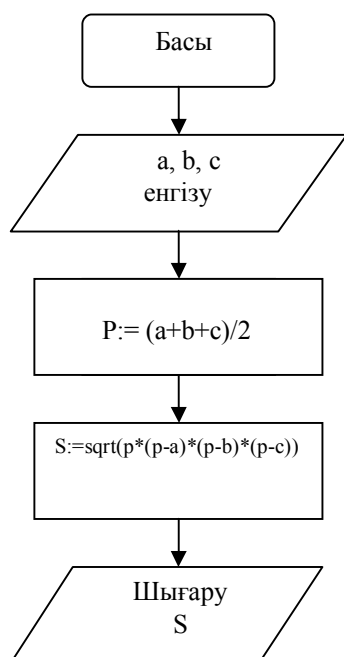
Мысал 1

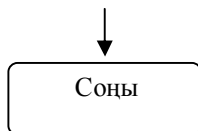
1. Есептің қойылуы: Герон формуласы бойынша үшбұрыш ауданын есептеу. `a`, `b`, `c` - үшбұрыш қабырғалары, `S` – ауданы.

2. Математикалық моделі:

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

3 Блок – схемасы:





4. Программа листингі:

```
#include <stdio.h>
#include <math.h>
void main()
{float a, b, c, p, s
printf("\na="); scanf("%f",&a);
printf("\nb="); scanf("%f",&b);
printf("\nc="); scanf("%f",&c);
p=(a+b+c)/2;
s=sqrt(p*(p-a)*(p-b)*(p-c));
printf("\nүшбұрыш ауданы=%f",s);
}
```

Есеп 1-дегі программаны Си++ тегі ағымдық енгізу-шығаруды пайдалану арқылы орындайық

```
#include <iostream.h>
#include <math.h>
void main()
{float a, b, c, p, s
cout<<"\na="; cin>>a;
cout<<"\nb="; cin>>b;
cout<<"\nc="; cin>>c;
p=(a+b+c)/2;
s=sqrt(p*(p-a)*(p-b)*(p-c));
cout<<"\nүшбұрыш ауданы=",s);
}
```

5. Программаны өңдеу:

Программаны өңдеуден өткізу үшін CTRL+F9, ал нәтижені қарау үшін ALT+F5 клавишалар комбинациясы қолданылады.

6. Программаны талдау:

Қарастырылған программадағы printf(...); және scanf(...); операторлары сәйкесінше экранға шығаруды және мәліметтерді клавиатурадан енгізуді жүзеге асырады. Олар тақырыптық файлының есімі stdio.h болып келетін Си енгізу-шығару стандарт кітапханасының сәйкес функцияларына жүгінеді.

Мұндағы a, b, c үш өлшемін бір оператор арқылы енгізуге болады:

```
scanf("%f%f%f",&a, &b, &c);
```

Қарастырылған программада екі меншіктеу операторы келтірілген: (p) жарты периметрінің есептеулері және (S) үшбұрыш ауданының есептеулері.

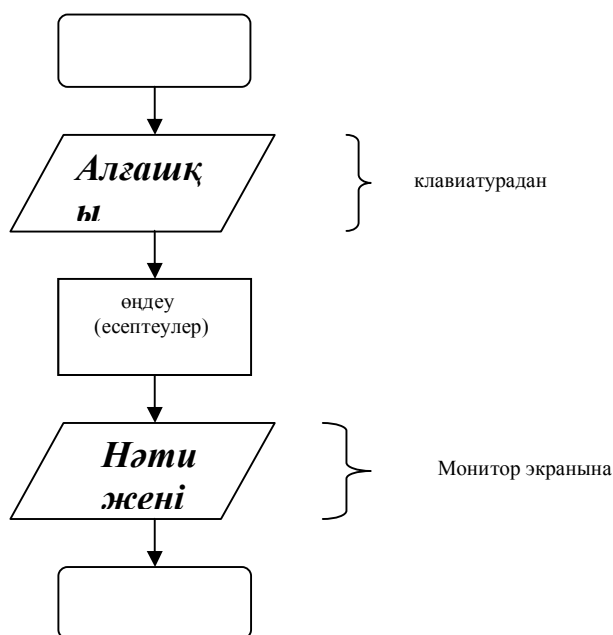
Өрнек ішінде ауданды есептеу үшін `sqrt()`- түбір асты кітапханалық функциясы қолданылған. Бұл функция *математикалық функциялардың кітапханасына* кіреді. Оны біздің программамызға қосу үшін `#include <math.h>` препроцессорлық директивасы қолданылады. Мұндағы `math.h` – сол кітапхананың тақырыптық файлының есімі. Келесі №1 кестеде Си/Си++ математикалық кітапханасының кейбір неғұрлым жиі қолданылатын функцияларының сипаттамалары берілген.

Кесте №1

Математикалық функциялар (тақырыптық файлы `math.h`)

Атауы	Аргумент типі	Нәтиже типі	Функция
<code>abs(x)</code>	Int	int	бүтін санның абсолюттік мәні
<code>acos(x)</code>	double	double	арккосинус (радиандар)
<code>asin(x)</code>	double	double	арксинус (радиандар)
<code>atan(x)</code>	double	double	арктангенс (радиандар)
<code>ceil(x)</code>	double	double	x-тен кіші емес ең жақын бүтін
<code>cos(x)</code>	double	double	косинус (x радианмен берілген)
<code>exp(x)</code>	double	double	e^x – x-тен берілген экспонента
<code>fabs(x)</code>	double	double	нақты x-тің абсолюттік мәні
<code>floor(x)</code>	double	double	x-тен аспайтын ең үлкен бүтін
<code>fmod(x,y)</code>	double double	double	x ті y-ке бөлгендегі қалдық
<code>log(x)</code>	double	double	$\ln x$ – натурал логарифм
<code>log10(x)</code>	double	double	$\lg x$ – ондық логарифм
<code>pow(x,y)</code>	double double	double	x^y – y дәрежелі x
<code>sin(x)</code>	double	double	синус (x радианмен берілген)
<code>sinh(x)</code>	double	double	гиперболалық синус
<code>sqrt(x)</code>	double	double	түбір асты (оң мән)
<code>tan(x)</code>	double	double	тангенс (x радианмен берілген)
<code>tanh(x)</code>	double	double	гиперболалық тангенс

Сызықтық программаның алгоритмі



№2 лабораториялық жұмыстың өзіндік тапсырмалары

1. a, b, c, d нақты сандары берілген. Олардың қосындысын, алымын және көбейтіндісін табу керек.
2. x және y нақты сандары берілген. Алу керек: $\frac{|x| + |y|}{1 + |xy|}$
3. Екі нақты оң сан берілген. Осы сандардың арифметикалық және геометриялық ортасын табыңыз.
4. Тік үшбұрыштың катеттері берілген. Оның гипотенузасы мен ауданын табыңыз.
5. x, y, z берілген. Экранға a, b -ның мәнін шығарыңыз, егер:

$$a) \quad a = \frac{\sqrt{|x-1| - \sqrt[3]{|y|}}}{1 + \frac{x^2}{2} + \frac{y^2}{4}}, \quad b = x(\arctg z + e^{-(x+3)});$$

$$b) \quad a = \frac{3 + e^{y-1}}{1 + x^2|y - \lg Z|}, \quad b = 1 + |y - x| + \frac{(y-x)^2}{2} + \frac{|y-x|^3}{3};$$

$$c) \quad a = (1+y) \frac{x + y/(x^2 + 4)}{1/(x^2 + 4)}, \quad b = \frac{1 + \cos(y-2)}{x^4/2 + \sin^2 z};$$

$$d) \quad a = y + \frac{x}{y^2 + \left| \frac{x^2}{y - x^3/3} \right|}, \quad b = (1 + \lg^2 \frac{z}{2});$$

$$e) \quad a = \frac{2 \cos(x - \pi / 6)}{1/2 + \sin^2 y}, \quad b = 1 + \frac{z^2}{3 + z^2 / 5};$$

6. Параллелограммның ауданын есептеу программасын жазыңыз.
7. Цилиндр көлемін есептеу программасын құрыңыз.
8. Тік үшбұрыштың гипотенузасы мен катеті берілген. Екінші катеті мен іштей сызылған шеңбердің радиусын анықтаңыз.
9. x_1, y_1 және x_2, y_2 координаталарымен берілген екі нүкте арасындағы арақашықтықты есептеңіз.
10. Куб қабырғасының ұзындығы берілген. Кубтың көлемі мен оның бір жағының ауданын есептеңіз.
11. Тең қабырғалы үшбұрыштың бір қабырғасы берілген. Осы үшбұрыштың ауданын есептеңіз.
12. Клавиатурадан енгізілген бөлшек санды ақшалай форматқа келтіретін программа құрыңыз. Мысалы, 125 саны 12 теңге 50 тиын түріне келтіріңіз.
13. Үшбұрыш бұрыштарының координаталары белгілі. Оның ауданын есептеу программасын құрыңыз.
14. Арақашықтықты версттан километрге аударатын программа жазыңыз (1 верст 1066,8 м-ге тең).
15. Көлікпен дачаға барып келудің құнын есептейтін программа құрыңыз. Кіріс мәліметтері ретінде: дачаға дейінгі арақашықтық (км); көліктің 100 км-ге жұмсайтын бензинінің саны; 1 литр бензиннің құны берілген.
16. Егер *идеалды салмақ (кг) = бойы (см) – 100* белгілі болса, адамның идеалды салмағын оның бойына байланысты есептеу программасын құрыңыз.
17. Мектептен стадионға дейінгі қашықтық S км, ал оқушы жүрісінің орташа жылдамдығы – V км/сағ. жолда оқушы қанша уақыт (минутпен) жұмсайды.

Блиц- тест:

1. Математикалық функцияларды қамтитын тақырыптық файлдың есімі:

A) Math.h
 B) Stdio.h
 C) Conio.h
 D) Graphics.h
 E) String.h

2. Бүтін санның абсолюттік мәнін есептейтін функция:

A) Abs(x).
 B) Pow(x,y).
 C) Sqrt(x).
 D) Tan(x).
 E) Log(x).

3. Горизонталды табуляцияны білдіретін эскейп-тізбек:

A) \t.
 B) \n.

- C) \r.
- D) \a.
- E) \v.

4. Бүтін ондық санды (типi int) білдіретін формат спецификаторы:

- A) %d.
- B) %c.
- C) %f.
- D) %e.
- E) %s.

5. C++ тегі x айнымалысына мән енгізу операторы:

- A) Cin>>x.
- B) Scanf(“%f”, &x).
- C) Cout<<x.
- D) Printf(“\nx=”,x).
- E) Cin<<x.

Бақылауға арналған сұрақтар

1. Сызықтық программа дегеніміз не?
2. void қызметші сөзі нені білдіреді?
3. Оператор дегеніміз не?
4. Математикалық функцияларды программаға қосу үшін қандай препроцессорлық директива қолданылады.
5. printf() және scanf() операторларының қызметі?
6. Басқару символдары не үшін пайдаланылады?
7. \t басқару символы нені білдіреді?
8. row(x, y) функциясының қызметі?
9. Формат спецификаторлары не анықтайды?
10. Модификатор дегеніміз не?
11. Си және Си++ тілдерінің енгізу-шығару құрылғыларының айырмашылығы.

Глоссарий:

- a. **Программа** – мақсатқа жету барысында орындалатынын әрекеттер тізбегінің – алгоритмнің машинаға түсінікті тілдегі нұсқасы.
- b. **Сызықтық программа** – Алгоритмнің «Ілесеу» базалық структурасының машинаға түсінікті программалау тіліндегі нұсқасы.
- c. **Форматтап енгізу-шығару** - клавиатурадан енгізілетін немесе экранға, лекке шығарылатын мәліметтердің орналасу ретін, типін анықтап орындау.
- d. **Формат спецификаторлары** – (әдістемелік нұсқауға қара)

Әдебиеттер:

[1],[3],[4],[14].

ЛАБОРАТОРИЯЛЫҚ ЖҰМЫС № 3

Тақырыбы: C/C++ тілінде көшу операторлары көмегімен программалау

Мақсаты: Таңдау инструкцияларын программалауды үйрету. Көшу операторларының түрлерімен таныстыру және логикасын түсіндіру.

Қажетті материалдар мен жабдықтар: ДК, C/C++ тілінің IDE ортасы, лабораториялық жұмысты орындауға арналған әдістемелік нұсқаулар.

Лабораториялық жұмыстың мазмұны және орындалу реті:

1. C/C++ тілінде таңдау және тармақталу алгоритмдерін программалауға катысты әдістемелік нұсқау – материалдарды алдын ала танысып меңгеру.
2. Ұсынылған мысалды орындау және талқылау.
3. Өзіндік жеке тапсырмалардың оқытушы ұсынған нұсқасын орындау.
4. Лабораториялық жұмыстың есебін (отчет) дайындап тапсыру.

Әдістемелік нұсқау:

If инструкциясы

- If инструкциясы, программаның орындалуы барысында екі бағыттың бірін таңдауға арналған.
- Инструкция тізбегін таңдау, *if* операторынан кейін жақшаның ішіне жазылған шарттың мәнінен тәуелді.
- Егер шарт жалған болса, онда *else* қызметші сөзінен кейін орналасқан оператор орындалады.
- Егер шарт орындалу немесе орындалмау кезінде бірнеше операторлар орындалатын болса, оларды фигуралық жақшаға алу керек.

Switch инструкциясы.

- Switch инструкциясы программаның орындалу кезінде бірнеше мүмкін болатын бағыттардың біреуін таңдауға арналған.
- Тандалатын инструкция тізбегі, селектор-айнымалының мәні, *case* сөзінен кейін орналасқан тұрақтыға теңдігіне байланысты.
- Егер селектор-айнымалының мәні *case* инструкциясынан кейін жазылған бірде бір тұрақтыға тең болмаса, онда *default* сөзінен кейін тұрған инструкциялар орындалады.
- Селектор-айнымалы ретінде бүтін (*int*) және символдық (*char*) типті айнымалыларды пайдалануға болады.

Көшу операторларының форматы.

If инструкциясы:

Қысқартылған түрі:

```
If (<шарт>)  
{
```

```
//Мұнда Шарт өрнегінің мәні ақиқат болған жағдайда,  
//орындалатын инструкциялар орналасады.  
}
```

Толық түрі:

```
If (<шарт>)  
{  
// Мұнда Шарт өрнегінің мәні ақиқат болған жағдайда,  
// орындалатын инструкциялар орналасады.  
}  
Else  
{  
// Мұнда Шарт өрнегінің мәні жалған болған жағдайда,  
// орындалатын инструкциялар орналасады.  
}
```

Switch инструкциясы:

1 вариант

```
Switch (<өрнек>)  
{  
Case 1тұрақты: 1инструкция; break;  
Case 2тұрақты: 2инструкция; break;  
.....  
Default: break;  
}
```

2 вариант

```
Switch (<өрнек>)  
{  
Case 1тұрақты: 1инструкция; break;  
Case 2тұрақты: 2инструкция; break;  
.....  
}
```

1 - мысал:

Қолданушыдан оқушының қай сыныпта оқитынын сұрап, сәйкесінше «бастауыш сынып», «орта сынып» немесе «жоғарғы сынып» деген хабарлама шығаратын программаны құрыңыз. Егер қолданушы мүмкін емес санды енгізсе, «Мәліметтерді енгізу барысында қате жіберілді!» деген хабарлама шығуы тиіс.

C тілінде

```
#include <stdio.h>  
#include <conio.h>  
void main ()  
{  
int klass;  
puts("\n Сыныптың номерін енгіз (1-ден 11 дейін)");
```



```

printf("- ");
scanf("%i", &klass);
if (klass<1 && klass>11)
printf("Мәліметтерді енгізу барысында қате жіберілді!");
else if (klass>=1 && klass<=4)
printf("Бастауыш сынып");
else if (klass>4 && klass<=9)
printf("Орта сынып");
else if (klass>9 && klass<=11)
printf("Жоғарғы сынып");
printf("\n\nПрограмма жұмысын аяқтау үшін <Enter> пернесін басыңыз");
detch();
}

```

C++ тілінде

```

#include <iostream.h>
#include <conio.h>
int main()
{
int klass;
cout<< "Сыныптың номерін енгіз (1 ден 11 дейін) \n"
cout<< "- "
cin>>klass
if (klass<1 && klass>11)
cout<< "Мәліметтерді енгізу барысында қате жіберілді! \n ";
else if (klass>=1 && klass<=4)
cout<< "Бастауыш сынып\n ";
else if (klass>4 && klass<=9)
cout<< "Орта сынып\n ";
else if (klass>9 && klass<=11)
cout<< "Жоғарғы сынып\n ";
cout<< "\n\nПрограмма жұмысын аяқтау үшін <Enter> пернесін басыңыз";
detch();
}

```

2 - мысал: Клавиатурадан енгізілген жыл мезгілінің номеріне (1-қыс, 2-көктем, 3-жаз, 4-күз) сәйкес, осы маусымдағы ай аттарын шығаратын программа құру.

```

#include <stdio.h>
#include <conio.h>
void main()
{
int season;
cout<< "\n Жыл мезгілін енгіз (1..4)";
cout<< "- ";
cin>>season;

```

```

switch (season);
case 1: cout<<“\nЖелтоқсан \nҚаңтар \nАқпан”; break;
case 2: cout<<“\nНаурыз \nСәуір \nМамыр”; break;
case 3: cout<<“\nМаусым \nШілде \nТамыз”; break;
case 4: cout<<“\nҚыркүйек \nҚазан \nҚараша”; break;
default: cout<<“Жыл мезгілінің номерін енгізу барысында қате жіберілді”;

```

Программаларды компьютерде теріп, нәтижесін талдаңыз.

Жеке тапсырмалар:

1. Квадрат тендеуді шешетін программа құрыңыз. Программа бастапқы мәліметтердің дұрыстығын тексеруі қажет және егер 2-ші дәрежелі белгісіздің коэффициенті нольге тең болса, онда сәйкесінше хабарлама шығаруы керек.
2. Клавиатурадан енгізілген 2 санды салыстыратын программа жазыңыз. Программа қай санның үлкен екендігін көрсетуі қажет, егер олар тең болса сәйкесінше хабарлама шығару керек.
3. Қолданушының енгізген санының жұп немесе тақтығын тексеретін программа құрыңыз.
4. Телефонмен сөйлесу ақысын есептейтін программа құрыңыз, сенбі және жексенбі күндері 20% жеңілдік болатындығын еске алыңыз.
5. Келесі күннің датасын есептейтін программа жазыңыз. Датаны толығымен енгізіңіз. Мысалы: 31.12.2004
6. Қолданушыдан апта күнін сұрап, соған сәйкес келесі мәліметтердің: «Жұмыс күні», «Сенбі», «Жексенбі» біреуін шығаратын программа жазыңыз.
7. Клавиатурадан енгізілген бүтін санның (1-ден 100 дейін) 3-ке және 5-ке қалдықсыз бөлінетіндігін тексеретін программа құрыңыз. Тексеріп болған соң бөлінетіндігі немесе бөлінбейтіндігі жайында хабарлама шығуы қажет.
8. Сатып алынған заттарға төленетін ақша көлемін жеңілдікпен есептейтін программа құрыңыз. Егер ақша көлемі 500 теңгеден асса, 3% жеңілдік, ал 1000 теңгеден асса, 5% жеңілдік жасалады. Ақша саны енгізілгеннен кейін, қанша ақша төлеу қажеттілігі айтылып, экранға шығуы қажет.
9. Қолданушының оптимальді салмағын есептейтін және оны негізгі салмағымен салыстырып, ұсыныс беретін программа жазыңыз. Оптимальді салмақ = Бойы (см) -100 формуласымен есептелінеді.

(Қолданушының енгізген мәліметтері қою түспен көрсетілген.)

Программа орындалу барысында экран кескіні төмендегідей болуы керек:

Бір жолға, арасына бос орын тастап бой (см) және салмақ (кг) мәндерін енгізіп, <Enter> пернесін басыңыз:

168 55

Сізге 13.00 кг толықтау қажет.

10. Клавиатурадан енгізілген бүтін санның үшке бөлінетін-бөлінбейтіндігін тексеретін программа жазыңыз.

(Қолданушының енгізген мәліметтері қою түспен көрсетілген.)

Программа орындалу барысында экран кескіні төмендегідей болуы керек

Бүтін санды енгізіп, <Enter> пернесін басыңыз:

523

523 саны үшке толық бөлінбейді.

11. Экранға екі санның қосындысын табуға арналған есепті шығарып, қолданушыдан жауабын сұрайтын программа құрыңыз. Егер қолданушының енгізген жауабы дұрыс болса, «Дұрыс» деген хабарлама, ал қате енгізілген жағдайда, дұрыс жауабын шығарылуы керек.

(Қолданушының енгізген мәліметтері қою түспен көрсетілген.)

Программа орындалу барысында экран кескіні төмендегідей болуы керек:

12+45=?

Жауабын жазып <Enter> пернесін басыңыз

45

Қателестіңіз ! 12+45=57

12. Қолданушының Ақтөбе қаласының қаланған жылын білетіндігін тексеруге арналған программа жазыңыз. Қате жауап берілген жағдайда, программа дұрыс жауапты шығару тиіс.

(Қолданушының енгізген мәліметтері қою түспен көрсетілген.)

Программа орындалу барысында экран кескіні төмендегідей болуы керек:

Ақтөбе қаласының іргетасы қай жылы қаланды?

Санды енгізіп, <Enter> пернесін басыңыз.

1875

Дұрыс емес, Ақтөбе қаласының іргетасы 1869 жылы қаланды.

13. Клавиатурадан енгізілген ай номеріне сәйкес жыл мезгілінің атын шығаратын программа құрыңыз.

Егер қолданушы 1-мен 12 аралығына жатпайтын санды енгізсе, «Мәліметтерді енгізу барысында қате жіберілді» деген хабарлама шығарылуы керек.

(Қолданушының енгізген мәліметтері қою түспен көрсетілген)

Программа орындалу барысында экран кескіні төмендегідей болуы керек:

Ай номерін енгіз (1 ден 12-ке дейін):

11

Қыс.

14. Switch инструкциясын пайдаланып қолданушыдан апта күнін сұрап, соған сәйкес келесі мәліметтердің «Жұмыс күні», «Сенбі», «Жексенбі» біреуін шығаратын программа жазыңыз. Егер қате мәліметтер енгізілсе «Қате!» деген хабарлама шығару керек.

15. **Switch** инструкциясын пайдаланып есепті шешіңіз.

Қалааралық телефон арқылы сөйлесудің бағасын есептейтін программа жазыңыз (1 минут сөйлесу бағасы абоненттің қаламен ара қашықтығымен анықталады).

Берілген мәліметтер: қаланың коды және сөйлесу ұзақтығы.

(Қолданушының енгізген мәліметтері қою түспен көрсетілген.)

Программа орындалу барысында экран кескіні төмендегідей болуы керек:

Телефон арқылы сөйлесу бағасын есептеу:

Берілген мәліметтерді енгізіңіз:

Қала коды- **120**

Ұзақтығы (минут)- **5**

Қала: Ақтөбе

1 минутқа төленетін баға- 7.8 тнг.

Сөйлесу құны: 39.00 тнг.

Қала	Код	1 минутқа төленетін бағасы (теңге)
Ақтөбе	120	7.8
Астана	145	11.5
Алматы	254	9.78

16. Қолданушының Қазақстан тарихынан білімін тексеру программасын құрыңыз.

Программа «Сырым Датұлы бастаған отарлау саясатына қарсы ұлт-азаттық көтеріліс қай жылдары болған?» деген сұрақты экранға шығарып, үш нұсқа жауабын береді. Қолданушы дұрыс нұсқаны тандап, сонын номерін енгізуі керек.

(Қолданушының енгізген мәліметтері қою түспен көрсетілген.)

Программа орындалу барысында экран кескіні төмендегідей болуы керек:

Сырым Датұлы бастаған отарлау саясатына қарсы ұлт-азаттық көтеріліс қай жылдары болған ?

1.1783-1797 жылдары.

2.1835-1547 жылдары.

3.1750-1780 жылдары

Дұрыс жауаптың номер енгіз:

2

Дұрыс емес.

Дұрыс жауап: 1

17. Экранға екі санның көбейтіндісін табуға арналған есепті шығарып, қолданушыдан жауабын сұрайтын программа құрыңыз. Егер қолданушының енгізген жауабы дұрыс болса, «Дұрыс» деген хабарлама, ал қате енгізілген жағдайда, дұрыс жауабын шығарылуы керек.

(Қолданушының енгізген мәліметтері қою түспен көрсетілген.)

Программа орындалу барысында экран кескіні төмендегідей болуы керек:

*12*8=?*

Жауабын жазып <Enter> пернесін басыңыз

80

*Қателестіңіз ! 12*8=96*

18. Кез-келген a , b , c саны үшін $ax^2+bx+c = 0$ теңдеуінің ең болмағанда бір нақты шешімінің бар/жоқтығын анықтаңыз.

19. Алты орынды n наурал саны берілсін. Оның құрамындағы жүздер мен мыңдардың санын анықтаңыз.

Блиц- тест:

1. if ($x>1.5$) $s=78$; $s:=-5$ жолының нәтижесі неге тең?

A) 78

B) -5

C) 0

2. Таңдау инструкциясы қайсысы?

- A) If
- B) Switch
- C) For
- D) while

Бақылау сұрақтар:

1. **if** инструкциясының атқаратын қызметі қандай?
2. Инструкциялар тізбегін таңдау қалай жүзеге асырылады?
3. *Else* қызметші сөзінен кейін жазылған инструкция қай жағдайда орындалады?
4. Егер шарт орындалған немесе орындалмаған жағдайда бірнеше инструкциялар орындалатын болса, оларды не істеу керек?

Глоссарий:

Тармақталу алгоритмі – қандайда бір шарттың орындалу немесе орындалмауына байланысты анықталатын екі жағдайдың бірін таңдау мүмкіндігі ескерілген алгоритм.

Таңдау алгоритмі – мүмкін болатын бірнеше жағдайлардың бірін таңдау мүмкіндігі берілген алгоритм.

Инструкция- компиляторға берілетін нұсқау

Әдебиеттер:

[3],[4],[5],[13],[15].

ЛАБОРАТОРИЯЛЫҚ ЖҰМЫС №4

Тақырыбы: Циклдық процестерді ұйымдастыру.

Мақсаты: C/C++ тілінде циклдарды ұйымдастыру, цикл көмегімен күрделі функцияларды шешу.

Қажетті материалдар мен жабдықтар: ДК, C/C++ тілінің IDE ортасы, лабораториялық жұмысты орындауға арналған әдістемелік нұсқаулар.

Лабораториялық жұмыстың мазмұны және орындалу реті:

1. C/C++ тілінде циклдық алгоритмдерді программалауға қатысты әдістемелік нұсқау – материалдарды алдын ала танысып меңгеру.
2. Ұсынылған мысалды орындау және талқылау.
3. Өзіндік жеке тапсырмалардың оқытушы ұсынған нұсқасын орындау.
4. Лабораториялық жұмыстың есебін (отчет) дайындап тапсыру.

Әдістемелік нұсқау:

for do цикл операторы

- программаны өңдеу кезінде цикл денесінің қайталану саны белгілі циклдардық процестерді ұйымдастыру кезінде қолданылады;
- циклдың қайталану саны цикл параметрінің бастапқы мәнімен және циклдың аяқталу шартымен анықталады;
- айнымалы параметр бүтін типті (int) болуы керек және тікелей цикл құрылымында сипатталуы мүмкін.

Екі цикл ерекшелігі

do while цикл операторы

- **do while** циклы құрылымының қайталану саны программаның орындалу барысында анықталады;
- Цикл денесі кем дегенде бір рет орындалады;

while do цикл операторы

- **while** циклының құрылымы while сөзінен кейінгі жазылған өрнектің мәні нөлге тең болғанша орындала береді;
- **while** сөзінен кейін цикл құрылымы орындалу шарты міндетті түрде жазылуы керек;

- **while** циклының аяқталуы үшін цикл денесінде міндетті түрде орындалуы циклдың аяқталу шартына әсер ететін нұсқау болуы керек;

Мысалы:

а) Есеп: Сыйымдылығы x кг – дық қалта және азық – түліктердің салмағы берілген. Азық – түліктерді осы қалтаға сыятындай етіп салып, жалпы салмағы қанша азық – түлік салынғанын экранға шығаратын программа құрыңыз.

ә) Есептің қойылымы: n_1, n_2, n_3, \dots параметрлері арқылы қалтаға салынатын заттардың салмағын белгілейміз, ал азық – түліктердің қосынды салмағын s арқылы белгіленеді, яғни $s = n_1 + n_2 + n_3 + \dots$. s -тің мәні экранға шығарылады.

в) Есептің шығарылу алгоритмі:

г) Есептің программасы:

С – дегі нұсқасы:	C++тегі нұсқасы:
<pre>#include<stdio.h> #include<conio.h> void main() { clrscr(); int n; float x,s; printf("Қалтаның сыйымдылығын енгізіңіз"); scanf("%f",&x); s=0; do {scanf("%i",&n); s=s+n;} while (s<x); printf("s=%f",s); getch(); }</pre>	<pre>#include<stdio.h> #include<conio.h> void main() { clrscr(); int n; float x,s; cout<< "\n Қалтаның сыйымдылығын енгізіңіз.""; cin >>x; s=0; do {cin >>n; s=s+n;} while (s<x); cout << "s= " <<s; getch(); }</pre>

Өз бетінше орындайтын тапсырмаларға түсініктеме

Есепті шығару барысында оның берілгенін талдап, математикалық моделін құру керек. Берілген тапсырмаларды орындау барысында цикл операторының үш түрімен де көрсетуге болады. Егер берілген есеп циклдың үш түрінің қайсібіреуімен шығаруға келмейтін болса, оның неліктен бұл циклмен шығаруға келмейтінін дәлелдеңіз. Есеп программасын құрғанда С және С++ тілдеріндегі нұсқаларын және блок – схемасы мен нәтижесі көрсетілуі қажет.

Өзіндік жеке тапсырмалар:

- 1) Клавиатурадан енгізілген сандардың қосындысын экранға шығаратын программа құрыңыз.
- 2) Клавиатурадан енгізілген сандар тізбегінің арифметикалық ортасын есептейтін программа жазу керек. Тізбек элементтерінің саны программаның орындалуы барысында беріледі.
- 3) Қолданушы енгізген ондық санау жүйесіндегі санды екілік санау жүйесіне ауыстыратын программа жазыңыз.
- 4) Алғашқы он оң бүтін жай сандардың квадратын шығаратын программа құрыңыз.
- 5) Қолданушы кез келген пернені басқанға дейін, мысалы үш минут бойы жұмыс жасап тұратын “электронды сағатты” экранға шығаратын программа құру керек.
- 6) Қолданушы енгізген сан жай сан екенін тексеретін программа құрыңыз.
- 7) Оқушылар бой көрсеткішінен тәуелсіз екі қатарлы сапқа сапқа тұрғызылды. Мұғалім бұйрығымен бірінші қатарға екінші қатар жалғасып тұрғызылады. Сол кезде пайда болған бір қатарды экранға шығаратын программа құрыңыз.
- 8) Екі бүтін санның ең үлкен ортақ бөлгішін есептейтін программа жазыңыз.
- 9) Клавиатурадан енгізілген оң сандар тізбегіндегі максималды санды анықтайтын программа құру керек.
- 10) Клавиатурадан енгізілген оң сандар тізбегінің ортақ арифметикалық ортасын есептейтін программа құру керек.

- 11) Екі санының нөлден бастап онға дейінгі дәрежесінің кестесін экранға шығаратын программа жазыңыз.
- 12) Клавиатурадан енгізілген санның факториалын есептейтін программа құрыңыз.
- 13) Пифагор шаршысын – көбейту кестесін экранға шығаратын программа жазыңыз.
- 14) Көбейту кестесін, мысалыға 7 санының, экранға шығаратын программа жазыңыз.
- 15) Экранға шахмат тақтасын шығаратын программа жазу керек. Тақтаның қара шаршыларын жұлдызшалармен, ал ақ шаршыларын астын сызу белгісімен белгілейміз.
- 16) Экранға бағалар кестесін, мысалы алманың 100г – нан 1 кг – ға дейінгі бағасын, 100г қадаммен шығаратын программа жазыңыз.
- 17) Бүтін сандардан тұратын тізбек берілген. Оның алғашқы элементі мен соңғы элементінің орнын ауыстырып жататын программа жазыңыз.

Блиц- тест:

1. for инструкциясының жазылу түрі

- A) for(i=1,i<=n,i++)
- B) for(i>10,i=i+1)
- C) for(i++,i<10)
- D) for()
- E) for (I,i<n,i++)

2. C/C++ тілінде циклдың қандай түрлері бар? Төменде көрсетілген нұсқалардың ішінен дұрыс емесін көрсетіңіз.

- A) repeat until;
- F) for ();
- G) while ;
- H) do while;
- I) Do while ; while;

Бақылау сұрақтары:

- 1) Цикл операторының қандай түрлері бар.
- 2) **for do** операторы қай жағдайда қолданылады.
- 3) **do while** операторының қолданылу жағдайы.

4) **while do** цикл операторында while циклының құрылымы қай уақытқа дейін орындала береді?

Глоссарий:

1. **Цикл** – бір немесе бірнеше әрекеттер тізбегінің әртүрлі мәндер үшін бірнеше рет қайталануы.
2. **Цикл параметрі**- (әдістемелік нұсқауды қара)
3. **Іштестірілген циклдар** – бірінің денесінде бірі орналасқан циклдар.
4. **Дейін-циклы, Әзір-Циклы**- алгоритмнің базалық структуралары.

Әдебиеттер:

[1-6],[13],[15].

ЛАБОРАТОРИЯЛЫҚ ЖҰМЫС №5

Тақырыбы: МАССИВ. МАССИВТІ ӨНДЕУ.

Мақсаты: C/C++ тілінде бір өлшемді және екі өлшемді массивтерді енгізу, шығару, өңдеуді ұйымдастыру. Массив көмегімен қолданбалы есептерді шешу.

Қажетті материалдар мен жабдықтар: ДК, C/C++ тілінің IDE ортасы, лабораториялық жұмысты орындауға арналған әдістемелік нұсқаулар.

Лабораториялық жұмыстың мазмұны және орындалу реті:

1. C/C++ тілінде массивтерді сипаттау және өңдеуге қатысты әдістемелік нұсқау – материалдарды алдын ала танысып меңгеру.
2. Ұсынылған мысалды орындау және талқылау.
3. Өзіндік жеке тапсырмалардың оқытушы ұсынған бір және екі өлшемді массивтерді өңдеу тапсырмаларының нұсқасын орындау.
4. Лабораториялық жұмыстың есебін (отчет) дайындап тапсыру.

Әдістемелік нұсқау:

Мысалы:

а) Есептің берілуі: Аптаның жеті күніндегі таңғы, түскі және кешкі уақыттағы температура мәні берілген. Аптаның ең жылы күнінің температура көрсеткішін табыңыз.

ә) Есептің математикалық моделі: Бұл есепті шығару барысында екі өлшемді массивті қолданамыз, яғни $apta[n][m]$ өлшемді массив аламыз. Массив жолы ретінде апта күндерін, ал бағаны ретінде уақыт мезгілдерін (таңғы, түскі және кешкі) аламыз. Аптаның әрбір күні $apta[n][m]$ массивінің бағандарына сәйкес келетіндіктен, аптаның әрбір күніндегі температураның орташа мәнін $s[n]$ өлшемді массивке меншіктеп, осы массивтің ең үлкен мәнін, яғни аптаның ең жылы күнінің температура көрсеткішін баспаға шығарамыз. Программаның мүмкін мәндері төменде көрсетілген:

$apta[n][m]$	18	20	19	20	21	18	17
	20	21	20	22	23	20	19
	17	19	18	20	21	17	16
$s[n]$	18,3	20	19	20,7	21,7	18,3	17,3

Аптаның ең жылы күнінің температура көрсеткіші- 21.7градус

б) Есептің шығарылу алгоритмі:

в) Есептің программасы:

C – дегі нұсқасы:	C++тегі нұсқасы:
<pre>#include<stdio.h> #include<conio.h> #include<math.h> #define n 7 #define m 3 void main() { clrscr(); int apta[n][m], i,j; float s[n], max; printf (“\n Апта күндеріндегі таңертеңгі, түскі және кешкі уақыттағы температура мәндерін енгізіңіз.”); for (i=0; i<n; i++) { for (j=0; j<m; j++) scanf (“%i”, &apta[i][j]); } for (i=0; i<n; i++) { s[i] = 0; for (j=0; j<m; j++) s[i] = (s[i] + apta[i][j])/n; } max=s[0]; for (i=0; i<n; i++) if (s[i]>max) max=s[i]; printf (“\n Аптаның ең жылы күнінің температура көрсеткіші -”); printf (“\n %f”, max); getch(); }</pre>	<pre>#include<iostream.h> #include<stdio.h> #include<conio.h> #include<math.h> #define n 7 #define m 3 void main() { clrscr(); int apta[n][m], i,j; float s[n], max; cout<< “\n Апта күндеріндегі таңертеңгі, түскі және кешкі уақыттағы температура мәндерін енгізіңіз.”; for (i=0; i<n; i++) { for (j=0; j<m; j++) cin >> apta[i][j]; } for (i=0; i<n; i++) { s[i] = 0; for (j=0; j<m; j++) s[i] = (s[i] + apta[i][j])/n; } max= s[0]; for (i=0; i<n; i++) if (s[i]>max) max=s[i]; cout << “\n Аптаның ең жылы күнінің температура көрсеткіші -”; cout << “\n” << max; getch(); }</pre>

Өзіндік жұмыс тапсырмалары:

Өз бетінше орындайтын тапсырмаларға түсініктеме

Берілген есепті талдап, оның математикалық моделін құрыңыз. Есеп программасын құрғанда оның C және C++ тілдеріндегі нұсқалары көрсетілуі керек.

а) Жеке тапсырмалар: (Бір өлшемді массив үшін)

- 1) Бүтін сандардан тұратын бір өлшемді массив берілген. Оның элементтерін кему ретімен орналастыру керек.

- 2) Клавиатурадан енгізілген 5 бүтін саннан тұратын бір өлшемді массив берілген. Осы массивке оның нөлдік емес элементтерін жалғастыра енгізетін программа жазу керек.
- 3) Топ студенттерінің ішіндегі бойы ең үлкен және ең кіші студенттің бой көрсеткішінің экранға шығаратын программа құрыңыз.
- 4) Клавиатурадан енгізілген сан берілген массивте бар жоғын анықтайтын программа құрыңыз. Массив программа орындалуы барысында енгізілуі керек.
- 5) Клавиатурадан енгізілген сан берілген массивте неше рет кездесетінін анықтайтын программа құрыңыз.
- 6) Массивте бірдей элементтердің барын анықтайтын программа жазу керек.
- 7) Сыныптағы оқушылардың ішінен бойы орташа бой көрсеткішінен асатын оқушылардың бойларының ұзындығын шығаратын программа жазыңыз.
- 8) Бүтін сандардан тұратын массив берілген. Осы массивтегі 5 – ке еселі элементтердің қосындысын табу керек.
- 9) Қала дүкендеріндегі сүттің бағалары берілген. Қала дүкендері бойынша сүттің орташа бағасы нормадан аса ма, соны тексеріңіз. Нормалы бағаны қолданушы өзі енгізеді.
- 10) Қала дүкендеріндегі еттің бағалары берілген. Қала дүкендері бойынша орташа бағаны анықтап, егер орташа баға нормадан асатын болса, оны нормалы бағаға ауыстыру керек. Нормалы бағаны қолданушы өзі енгізеді.
- 11) Массив берілген. Оның алғашқы элементі мен соңғы элементінің орнын ауыстыратын программа құрыңыз.
- 12) Екі массив берілген. Осы екі массивтің элементтерінің қосындысын экранға шығаратын программа құрыңыз.
- 13) Студенттің барлық оқу жылы бойынша «Математика» пәнінен алған бағалары берілген. Оның осы пән бойынша қорытынды бағасын есептейтін программа құрыңыз.
- 14) Студенттің 1 – курстағы қысқы және жазғы сессиясының бағалары берілген. Оның үлгерімі қалай өзгергенін есептейтін программа жазыңыз. Егер оның жазғы сессиядағы үлгерімі қысқы сессияға қарағанда артса, онда “артты” деген, ал керісінше болса, “төмендеді” деген хабарландыру шығуы керек.
- 15) Бір сыныптағы оқушылардың білімінің орташа балы берілген. Озат оқушылар сыныптың неше пайызын құрайтынын есептейтін программа құрыңыз.
- 16) Массив берілген. Оның алғашқы және соңғы екі элементінің қосындысын есептейтін программа құрыңыз.
- 17) Оқушылар бой көрсеткішінен тәуелсіз екі қатарлы сапқа тұрғызылды. Мұғалім бұйрығымен саптың бірінші қатарында тұрғандар бойларының өсу ретімен, ал саптың екінші қатарында тұрғандар бойларының кему ретімен тұруының программасын жазыңыз.

- 18) Солдаттар екі қатарлы сапқа бой – бойымен реттеліп тұрғызылған. Командирдің бұйрығымен олардың бой – бойымен бір қатарлы сапқа реттеліп тұрғызылуын көрсететін программа құрыңыз.
- 19) Сыныптағы оқушылардың орташа бой көрсеткішін анықтайтын және бойы орташа бой көрсеткіштен төмен оқушылардың бойының ұзындығын көрсететін программа жазыңыз.
- 20) Бір өлшемді массивтің тақ позициялы элементтерінің ішіндегі ең кіші элементін, ал жұп позициялы элементтерінің ішіндегі ең үлкен элементін экранға шығаратын программа жазыңыздар.

б) Жеке тапсырмалар: (Екі өлшемді массив үшін)

- 1) Екі өлшемді массив берілген. Осы массивтегі ең кіші элементті табуып, оның реттік номерін анықтау керек. Егер олар бірнешеу болса, олардың реттік номерін әрқайсысы үшін анықтау қажет.
- 2) Екі өлшемді массивтің бағандары бойынша элементтерінің қосындысын анықтайтын программа құрыңыз.
- 3) Студенттердің 8-ші семестрдегі 5 түрлі пән бойынша алған бағалары берілген. Студенттердің үлгерімін анықтайтын программа жазыңыз.
- 4) Екі өлшемді квадрат массив берілген. Оның бас диагоналының элементтерінің қосындысын экранға шығаратын программа жазыңыз.
- 5) Екі өлшемді квадрат массив берілген. Оның элементтерін спираль бойымен шығаратын программа құрыңыз.
- 6) Екі өлшемді квадрат массив берілген. Осы матрицаның анықтауышын табатын программа құрыңыз.
- 7) Екі өлшемді квадрат массив берілген. Оның бас диагоналының үстінде орналасқан элементтерінің қосындысын есептейтін программа құру керек.
- 8) Енгізілген матрица сиқырлы шаршы болуын тексеретін программа құрыңыз.
- 9) Екі өлшемді екі массив берілген. Осы екі массивтің элементтерінің қосындысын экранға шығаратын программа құрыңыз.
- 10) Бензин құятын жерде бензиннің үш түрі бар. Үш күн бойы сатылған бензин мөлшері литрмен берілген. Бензиннің қай түріне сұраным көп екенін анықтайтын программа құрыңыз.
- 11) Бастауыш сыныптың оқушылары сыныптары бойынша бой – бойымен сапқа тұрғызылған. Әрбір сынып бойынша оқушылардың орташа бой көрсеткішін анықтайтын программа жазыңыздар.
- 12) Квадрат матрицаның элементтерінің қосындысы максималды болатын жолының номерін көрсететін программа жазыңыз.
- 13) Екі өлшемді массив берілген. Осы матрицаның төрт бұрышындағы элементтердің қосындысын есептейтін программа жазыңыз.
- 14) Үш дүкенде бұйымдардың төрт түрі бойынша бір күнде түскен пайда берілген. Әрбір дүкен бойынша қайсы бұйым жақсы өткенін анықтау қажет.
- 15) 0, 1 және 2 - ден бөтен әртүрлі сандардан тұратын квадрат массив берілген. Оның бас диагоналының элементтерін 0 – ге, бас диагоналының

үстінде тұрған элементтерді 1 –ге, ал бас диагональдың астында орналасқан элементтерді 2 – ге ауыстыратын программа құру керек.

- 16) Оқушылар үш қатарлы сапқа тұрғызылған. Жұп позицияда тұрған оқушылар 0 – мен, ал тақ позицияда тұрған оқушылар 1 – мен белгіленген. Тақ позицияда тұрған оқушыларды экранға шығаратын программа жазыңыз.
- 17) Оқушылар бой көрсеткішінен тәуелсіз екі қатарлы сапқа сапқа тұрғызылды. Мұғалім бұйрығымен бірінші қатарға екінші қатар жалғасып тұрғызылады. Сол кезде пайда болған бір қатарды экранға шығаратын программа құрыңыз.
- 18) Оқушылар бір қатарлы сапқа тұрғызылған. Жұп позицияда тұрған оқушылар 0 – мен, ал тақ позицияда тұрған оқушылар 1 – мен белгіленген. Жұп позицияда тұрған оқушыларды экранға шығаратын программа жазыңыз.

Бақылау сұрақтары:

- 1) Массив дегеніміз не?
- 2) Бір өлшемді массив дегеніміз не, екі өлшемді массив дегеніміз не?
- 3) Массив элементтерін қандай операторлар көмегімен енгізіп, шығаруға және өңдеуге болады?
- 4) Массив элементтерінің индексі қандай аралықта өзгереді?
- 5) Массив қандай типке жатады?
- 6) Массивті сипаттау кезінде тұрақты қолданылса, қандай директива қолданылады?

Блиц- тест:

1. C/C++ тілінде бір өлшемді массивті қалай сипаттайды? Төменде көрсетілген нұсқаларының ішінен дұрысын көрсетіңіз:
 - A) <типi> <массив аты> [өлшемі];
 - B) var <массив аты>:array[1..n] of <типi>, мұндағы n – массив элементтерінің саны;
 - C) <массив аты> [өлшемі] <типi>;
 - D) void <массив аты> [өлшемі] <типi>.
 - E) Массив қолданылмайды
2. A[n] массиві элементтерінің индексі қай аралықта өзгере алады?
 - A) 0 – ден бастап n – 1 ге дейін;
 - B) 1 – ден бастап n – 1 - ге дейін;
 - C) n – нен бастап 1 –ге дейін;
 - D) 0 – ден бастап 255 арасында.
 - E) 1 – ден бастап 255 арасында.
3. C/C++ тілінде екі өлшемді массивті сипаттаудың төменде көрсетілген нұсқаларының ішінен дұрысын көрсетіңіз:
 - A. <типi> <массив аты> [жолының өлшемі] [бағанының өлшемі];

- B. var <массив аты>:array[1..n,1..m] of <типi>, мұндағы n – массив жолдарының саны, m – массив бvгвндарының саны;
- C. <массив аты> [өлшемі] <типi>;
- D. void <массив аты> [өлшемі] <типi>.
- E. <массив аты> [өлшемі] <типi>.

Глоссарий:

- **Массив** – (әдістемелік нұсқауды қараңыз)
- **Массивті сұрыптау**- массив элементтерін өсу немесе кему реті бойынша реттеп орналастыру.
- **Массив индексі** – бір атпен аталған бірдей типті элементтердің реттік нөмірі.
- **Массивті сипаттау**- компиляторға бірдей типті элементтің атауын, қандай ретпен индекстелетінін және ОЖ-дан қанша байт орын берілуі жөнінде мәліметтердің берілу формасы.

Әдебиеттер:

[1-6],[13],[15].

ЛАБОРАТОРИЯЛЫҚ ЖҰМЫС №6

Тақырыбы: C/C++ ТІЛІНДЕ ЖОЛДЫҚ ҚАТАРЛАРДЫ ӨНДЕУ

Мақсаты: Студенттерді C/C++ тіліндегі жолдық қатарларды өндеуге арналған негізгі функциялармен таныстыру.

Қажетті материалдар мен жабдықтар: ДК, лабораториялық жұмысты орындауға арналған әдістемелік нұсқаулар.

Лабораториялық жұмыстың мазмұны және орындалу реті:

1. C/C++ тілінде жолдармен жұмыс жасауға қатысты келтірілген әдістемелік нұсқау – материалдарды алдын-ала танысып меңгеру.
2. Ұсынылған мысалды орындау және талқылау.
3. Өзіндік жеке тапсырмалардың оқытушы ұсынған нұсқасын орындау.
4. Лабораториялық жұмыстың есебін (отчет) дайындап тапсыру.

Әдістемелік нұсқау:

C/C++ тілдерінде Турбо Паскаль тіліндегідей жолдық тип жоқ. Жолдар соңғы элементі \0 литеры болатын литерлер массиві ретінде ұйымдастырылады. C тіліндегі литерлік массивтің ұзындығы шектелмеген.

Жол литерлік массив түрінде сипатталады.

Мысалы:

```
char STR[20];
```

Жолдарды енгізу-шығару функциялары

gets ()*, *puts ()*, *fgets ()*, *fputs ()*, *sprintf ()

gets () функциясы литерлерді енгізу құрылғысынан қабылдап, аргумент ретінде алынған массивке орналастырады. Енгізуді аяқтаған (Enter пернесін басқан) кезде, жаңа жол литері (\n) алынады. *gets ()* функциясы оны нольдік литерге (\0) айналдырады.

gets () функциясын қолданған кезде енгізілген литерлер саны массив өлшемінен асып кететіндігін анықтау мүмкіндігі жоқ екендігін есте сақтаған жөн.

puts () функциясы *gets ()* функциясының көмегімен енгізілген жолды экранға шығарады. Ол нольдік литерді, керісінше, жаңа жол литеріне ауыстырады.

fgets () функциясы *gets ()* функциясына ұқсас, бірақ массив өлшеміне сәйкес клавиатурадан енгізілетін литерлер санын бақылап отыруға мүмкіндік береді. Литерлер қандай да бір файлдан немесе стандартты енгізу құрылғысынан (*stdin*) оқылады. Енгізілген литерлер саны массив өлшемінен бір бірлікке аз болуы керек, өйткені соңғы позицияға автоматты түрде нольдік литер орналасады. *fgets ()* функциясымен бірге *fputs ()* функциясы қолданылады. Бұл функция литерлерді файлға немесе стандартты шығару құрылғысына (*stdout*) бағыттайды.

sprintf() (string printf(), яғни *printf()* жолдық) функциясы *printf()* функциясындағыдай форматтау спецификаторлары қолданылады. Айырмашылығы *sprintf()* функциясы нәтижені экранға емес, берілген массивке орналастырады. Бұндай жағдай нәтижені, айталық экранға және принтерге бірнеше рет шығару кезінде тиімді болуы мүмкін.

Жолдармен жұмыс істеуге арналған негізгі функциялар (*string.h*, *stdlib.h*)

Функция	Прототипі және атқаратын қызметі
Atof	double atof (char *str); str жолын double типті нақты санға түрлендіреді
Atoi	int atoi (char *str); str жолын ондық бүтін санға түрлендіреді
Itoa	char *itoa (int v, char *str, int baz); v бүтін санын str жолына түрлендіреді. Санды бейнелеу кезінде baz ($2 \leq baz \leq 36$) негізі қолданылады.
Strcat	char *strcat (char *sp, char *si); si жолын sp жолына тіркеп жазады (конкатенация)
Strncat	char *strncat (char *sp, char *si, int k); si жолының k литерін sp жолына тіркеп жазады (конкатенация)
Strchr	char *strchr (char *str, int c); str жолында c литерінің алғашқы енуін анықтайды
Strrchr	char *strrchr (char *str, int c); str жолында c литерінің соңғы енуін анықтайды
Strcmp	int strcmp (char *str1, char *str2); str1 және str2 жолдарын салыстырады. Егер str1 < str2 болса, нәтиже теріс; егер str1 = str2 болса, нәтиже нольге тең және егер str1 > str2 – нәтиже оң.
Strncmp	char *strncmp (char *str1, char *str2, int k); str1 және str2 жолдарының алғашқы k литерін салыстырады. Егер str1 < str2 болса, нәтиже теріс; егер str1 = str2 болса, нәтиже нольге тең және егер str1 > str2 – нәтиже оң.
Strcpy	char *strcpy (char *sp, char *si); si жолын байттарын sp жолына көшіреді.
Strncpy	char *strncpy (char *sp, char *si, int k); si жолының k литерін sp жолына көшіреді.
Strlen	unsigned strlen (const char *str); str жолының ұзындығын анықтайды
Strlwr	char *strlwr (char *str); Жолдағы жоғарғы регистр әріптерін сәйкесінше төменгі регистрге ауыстырады.
Strupr	char *strupr (char *str); Жолдағы төменгі регистр әріптерін сәйкесінше жоғарғы регистрге ауыстырады.


```
dt=strlen(t); //Сөздің ұзындығы
```

```
//Сөзді кері ретпен жазу циклы
```

```
for(r=0;r<=(dt-1)/2;r++)
```

```
{ c=t[r]; t[r]=t[dt-r-1]; t[dt-r-1]=c;}
```

```
printf("%s\n",t);
```

```
for (r=0;r<dt;r++) t[r]=' ';
```

```
m=0;}
```

```
}
```

```
//Соңғы символ әріп болған жағдайда
```

```
if(m!=0) printf("\n%s - ",t); dt=strlen(t);
```

```
for(r=0;r<=(dt-1)/2;r++)
```

```
{ c=t[r];
```

```
t[r]=t[dt-r-1];
```

```
t[dt-r-1]=c;}
```

```
printf("%s\n",t);
```

```
return(0);
```

```
}
```

Өз бетінше орындалатын тапсырмаға түсініктеме

Есепті шығару барысында оның қойылымын, математикалық моделін, блок-схемасын құрып, программасын С және С++ тілдерінде жазып, нәтижесін көрсетіңіз.

Жеке тапсырмалар

1. Арасында қос нүкте (:) белгісі бар литерлер тізбегі берілген. Оған дейінгі литерлер санын анықтау.
2. Нүктемен (.) аяқталған текст берілген. Үш әріптен тұратын сөзді экранға шығару.
3. Жол берілген. Осы жолға 'abc' әріптерінің тобы қанша рет енетінін анықтау.
4. Жол берілген. Оның соңғы сөзіндегі 'k' әріпінің санын есептеу.
5. Жол берілген. Қатар тұрған 'a' әріптерінің ең ұзын тізбегін есептеу.

6. Бір жабылатын және бір ашылатын жақшалары бар литерлер тізбегі берілген. Экранға осы жақшалардың ішіндегі литерлерді шығару.
7. Нүктелі үтір (;) литерімен бөлінген сөздер тізбегі берілген. Тізбек қос нүктемен (:) аяқталады. 'a' әріпімен аяқталатын қанша сөздің бар екендігін анықтау.
8. Жол берілген. Ең болмағанда бір 'k' әріпінен тұратын сөздерді табу.
9. Жол берілген. Бір әріптен басталып, сол әріппен аяқталатын сөздерді табу.
10. Жолдағы барлық нүктелі үтірлерді (;) қос нүктемен (:) ауыстыру. Ауыстырулар санын есептеу.
11. Жолдағы барлық қос нүкте (:) литерлерін өшіру және өшірілген литерлер санын есептеу.
12. Жолдағы сөздердің арасындағы бос орындардың орнына үтір (,) және бос орын қою.
13. Жақшаға алынған жолдың бөлігін жақшасымен бірге өшіру.
14. Берілген сөздің жолда қанша рет кездесетінін анықтау.
15. Жолда бір нүктелі үтір (;) бар. Оған дейінгі және одан кейінгі литерлер санын есептеу.
16. Берілген сөздегі әріптердің санына байланысты '!' литерін тіркеу. (Мысалы, «УРА» жолынан «УРА!!!» жолын алу).
17. Жол берілген. Осы жолдағы «a» әріптерін «b» әріптеріне ауыстыратын программа жазу.
18. Жол берілген. Жолдағы тыныс белгілер санын есептеу.
19. Текст берілген. Ондағы ең қысқа сөздің неше рет кездесетінін анықтау.
20. Текст берілген. Осы тексте кездесетін барлық сөздерді алфавиттік ретпен шығару.
21. Литерлер массиві берілген. Әріптен басқа барлық литерлерді жойып және барлық кіші әріптерді үлкен әріптермен алмастыр.

Блиц- тест:

1. c-ны жоғарғы регистрге келтіретін функция
A) toupper(c)
B) isalnum(c)

- C) tolower(c)
- D) case(c)
- E) Upcase (c)

2. Литерлік шамаларға қолданылатын функциялар қай кітапханада орналасқан?

- A) < string . h >
- B) < stdio. h >
- C) < conio. h >
- D) < ctype. h >
- E) < iostream. h >

3. Жолдың ұзындығын табу функциясын көрсетіңіз:

- A) str len
- B) strstr
- C) Strcmp
- D) strcpy
- E) Length

Глоссарий:

- **Жолдар-** символдар тізбегі немесе символдық массив.
- **Жол ұзындығы-** жол құрамындағы символдар саны.

Әдебиеттер:

[1-6],[13],[15].

ЛАБОРАТОРИЯЛЫҚ ЖҰМЫС №7

Тақырыбы: C/C++ ТІЛІНДЕГІ ІШКІ ПРОГРАММАЛАР - ФУНКЦИЯЛАР.

Мақсаты: Функция туралы түсінік беріп, функциялар құруды және программада қолдана білуге үйрету.

Қажетті материалдар мен жабдықтар: ДК, лабораториялық жұмысты орындауға арналған әдістемелік нұсқаулар.

Лабораториялық жұмыстың мазмұны және орындалу реті:

1. C/C++ тілінде ішкі программаларды функция түрінде ұйымдастыруға қатысты әдістемелік нұсқау – материалдарды алдын ала танысып меңгеру.
2. Ұсынылған мысалды орындау және талқылау.
3. Өзіндік жеке тапсырмалардың оқытушы ұсынған нұсқасын орындау.
4. Лабораториялық жұмыстың есебін (отчет) дайындап тапсыру.

I. Әдістемелік нұсқау:

Функция – бұл аяқталған әрекетті орындайтын операторлар тобы. Олар программаны жеке блоктарға немесе модульдерге бөліп, басқаруға мүмкіндік береді. Бұл программа құрылымын жеңілдетеді. Функцияны қолдану үшін оның қалай жұмыс жасайтынын білу міндетті емес, тек қалай шақыру керектігін білу жеткілікті. Функция программада тек бір рет қана жазылып, бірнеше рет шақырылуы мүмкін. C/C++ тілінің негізгі программалық кодын осы функциялар құрайды. Бұл тілдердегі кез келген программада ең кемінде бір функция - `main()` функциясы болады. Ол программаның аяқталу кодын қайтарады.

Әрбір функцияның прототипі, яғни алдын ала хабарланған функция тақырыбы болуы керек. Тақырыбында функция аты, қайтарылатын нәтиже типі, сонымен қатар қанша және қандай типті аргументтер берілу керектігі көрсетіледі. Функцияның жалпы сипатталу синтаксисі төмендегідей:

```
нәтиже_типі функция_аты (аргумент_типі аргумент_аты,... )
{
    .
    .
    функция денесі
    .
    .
}
```

Мұндағы:

нәтиже_типі – функцияның қайтаратын мәнінің типі;

функция_аты – функцияның аты;

аргумент_типі – функцияда қолданылатын формальді аргумент типі;

аргумент_аты- функцияда қолданылатын формальді аргумент аты.

Функция туралы сөз болғанды екі түрлі аргумент, біріншісі – нақты аргумент, екіншісі –формальді аргумент туралы айтылады. Нақты аргумент функцияны шақырған уақытта беріледі, ал формальді аргумент функция сипаттамасында ғана қолданылады.

Формальді аргументтер әр түрлі типті болуы мүмкін: void, char, int, float, double және массив. void типі функцияда формальді аргументтер көрсетілмеген уақытта қолданылады.

Функция нәтижесінде void, bool, int, long, float, double типті айнымалылар қайтаруы мүмкін. Функция ешқандай мәлімет қайтармайтын болса, void типімен сипатталады.

II. Мысал.

Берілгені: Үшбұрыш төбелерінің координаттары: (X_1, Y_1) , (X_2, Y_2) , (X_3, Y_3) белгілі. Ауданын есептеу керек.

Математикалық моделі: Үшбұрыш ауданын табу үшін Герон формуласын пайдалануға болады. Бірақ, алдымен қабырғаларының ұзындығын табу керек. Оны екі нүктенің ара-қашықтығын есептеу формуласымен

$$D = (X_2 - X_1)^2 + (Y_2 - Y_1)^2$$

анықтауға болады. Үшбұрыш қабырғаларының ұзындығын есептеуді функция түрінде ұйымдастыруға болады.

Программасы:

С тілінде:	C++ тілінде:
<pre>#include <stdio.h> #include <math.h> float dlina(int x,int y,int xx,int yy); main() { int x1,x2,x3,y1,y2,y3; float ab,bc,ac,p,s; printf("\n x1="); scanf("%d",&x1); printf("\n y1="); scanf("%d",&y1); printf("\n x2="); scanf("%d",&x2); printf("\n y2="); scanf("%d",&y2); printf("\n x3="); scanf("%d",&x3); printf("\n y3="); scanf("%d",&y3); ab=dlina(x1,y1,x2,y2); bc=dlina(x2,y2,x3,y3); ac=dlina(x3,y3,x1,y1); p=(ab+bc+ac)/2; s=sqrt(p*(p-ab)*(p-bc)*(p-ac)); printf("\n s=%f",s); return (0);</pre>	<pre>#include <iostream.h> #include <math.h> float dlina(int x,int y,int xx,int yy); main() { int x1,x2,x3,y1,y2,y3; float ab,bc,ac,p,s; cout<<"x1="; cin>>x1; cout<<"y1="; cin>>y1; cout<<"x2="; cin>>x2; cout<<"y2="; cin>>y2; cout<<"x3="; cin>>x3; cout<<"y3="; cin>>y3; ab=dlina(x1,y1,x2,y2); bc=dlina(x2,y2,x3,y3); ac=dlina(x3,y3,x1,y1); p=(ab+bc+ac)/2; s=sqrt(p*(p-ab)*(p-bc)*(p-ac)); cout<<"s="<<s; return (0);</pre>

<pre> } float dlina(int x,int y,int xx,int yy) { float d; d=sqrt(pow(xx-x,2)+pow(yy-y,2)); return (d); } </pre>	<pre> } float dlina(int x,int y,int xx,int yy) { float d; d=sqrt(pow(xx-x,2)+pow(yy-y,2)); return (d); } </pre>
---	---

Өзіндік жеке тапсырмалар:

1. Сыныптағы әрбір оқушының бойы мен салмағы белгілі. Әрқайсысының салмағы бойынан қаншалықты ауытқитынын, яғни қаншаға артық не кем екенін анықтаңыз.
2. Екі бөлшекті қосу керек. Ең үлкен ортақ бөлгішін табуды функция түрінде жаз.
3. Автопаркте белгілі госномерлі 5 машина жұмыс жасайды. Әрбір жүргізуші күнде түскен табысты автошаруашылыққа тапсырады. Бір аптадағы күнделікті түсіп отырған табыс мөлшері белгілі. Қай машина осы аптада ең көп табыс түсіргенін анықтаңыз.
4. Клавиатурада енгізілген сандардың жәй сан ба екенін тексеріңіз.
5. Апта басында дүкенге 5 түрлі конфет әкелінді. Дүкенші күнде әрбір конфет түрінен қанша табыс түскенін жазып отырады. Апта соңында қай конфет түрінен көп пайда түскен?
6. Клавиатурадан 5 адамның аты енгізіледі. Әрбір адамның атындағы дауысты дыбыстар санын анықтаңыз.
7. Қаланың 5 дүкенінде сатылатын қаймақ бағасы белгілі. Қай дүкенде қаймақ арзан, ал қай дүкенде қымбат екенін анықтаңыз.
8. Бір аптаның әрбір күніндегі ауа температурасы белгілі. Әр аптаның орташа температурасын анықтаңыз.
9. Фирма қызметкерлерінің жастары белгілі. Жасы орта жастан жоғары қызметкерлердің санын анықтаңыз.
10. Компьютерлік кластағы мониторлардың шығарылған жылы белгілі. Ең көне монитордың шығарылған жылын анықтаңыз.
11. Бес бөлменің әрқайсысын сырлау үшін қажетті ақ, қызыл, көк, түсті бояу саны белгілі. Ақ, қызыл, көк бояулардың әрқайсысынан барлығы қаншасы қажет.
12. Топтағы студенттердің туған айы белгілі. Олардың қаншасы қыста, қаншасы көктемде, қаншасы жазда, қаншасы күзде туғанын анықтаңыз.
13. Кассадағы 200-к, 500-к және 1000-к купюралар саны белгілі. Барлығы кассада қанша ақша бар?

14. Топтағы әрбір студенттің себепті және себепсіз жіберген сағаттар саны белгілі. Жалпы топтағы себепті және себепсіз жіберілген сағаттар саны қанша? Қайсысы көп?
15. Театрда спектакльге сатылған билеттер саны және әрбір қатардағы билет құны белгілі. Осы спектакльден театрға қандай мөлшерде табыс түсті.

Бақылау сұрақтары:

1. Ішкі программалар дегеніміз не?
2. Функция қандай типті мәндер қайтара алады?
3. Функция параметрлерінің берілу механизмі қандай?
4. Қай уақытта функцияларды қолданған дұрыс?

Блиц-тест:

1. Функцияның ішінде ғана пайдаланылатын аргументтер қалай аталады:
A) Формальді;
B) Глобальді;
C) Нақты;
D) Уақытша;
E) Локальді;
2. Ешқандай мән қайтармайтын функцияны көрсетіңіз:
A) `void kos(int x, int y);`
B) `float kos(float y);`
C) `bool kos(int x, int y);`
D) `float kos(char x);`
E) `double kos(float *p);`
3. Аргументтері бүтін тип ретінде сипатталған функцияны көрсетіңіз:
A) `bool kos(int x, int y);`
B) `float kos(float y);`
C) `void kos(int x, int y);`
D) `float kos(char x);`
E) `double kos(float *p);`

Глоссарий:

- **Функция параметрлері** – ішкі программаға берілетін бастапқы мәндер мен нәтижелік мәндердің көшірмесі немесе оларға Көрсеткіштер тізімі.

- **Жергілікті объектілер-** функция денесінде сипатталып сонда ғана қоланылатын элементтер.
- **функцияның прототипі** – (әдістемелік нұсқауды қараңыз)
- **Формальды аргумент** – (әдістемелік нұсқауды қараңыз)
- **Нақты аргумент** – (әдістемелік нұсқауды қараңыз)

Әдебиеттер:

[1-6],[13],[15].

Лабораториялық жұмыс №8

Тақырыбы: АЛГОРИТМДЕРДІ ҚҰРУДЫҢ РЕКУРСИЯЛЫҚ ӘДІСТЕРІ

Мақсаты: Студенттерді рекурсиялық функциялармен және рекурренттік тізбектермен таныстыру.

Қажетті материалдар мен жабдықтар: ДК, лабораториялық жұмысты орындауға арналған әдістемелік нұсқаулар.

Лабораториялық жұмыстың мазмұны және орындалу реті:

1. C/C ++ тілінде рекурсивті алгоритмдерді программалауға қатысты әдістемелік нұсқау – материалдарды алдын ала танысып меңгеру.
2. Ұсынылған мысалды орындау және талқылау.
3. Өзіндік жеке тапсырмалардың оқытушы ұсынған нұсқасын орындау.
4. Лабораториялық жұмыстың есебін (отчет) дайындап тапсыру.

Әдістемелік нұсқау:

Рекурренттік тізбек түсінігі математика курсынан белгілі. Мысалы, a_1, \dots, a_k тізбегінің k саны белгілі болсын. Бұл сандар сандық тізбектің алғашқы сандары болып табылады. Тізбектің келесі элементтері былай есептеледі:

$$a_{k+1}=F(a_1, \dots, a_k); \quad a_{k+2}=F(a_2, \dots, a_{k+1}); \quad a_{k+3}=F(a_3, \dots, a_{k+2}); \dots$$

Мұндағы F дегеніміз k аргументтен тұратын функция.

$a_i=F(a_{i-1}, a_{i-2}, \dots, a_{i-k})$ формуласы *рекурренттік формула* деп аталады. Ал k өлшемі *рекурсия тереңдігі* деп аталады.

Яғни, рекурренттік тізбек дегеніміз – бұл шексіз сандар тізбегі және оның әр мүшесі алдыңғысы арқылы есептеледі, бастапқы k -ны санамағанда.

Рекурренттік тізбек ретінде арифметикалық (1) және геометриялық (2) тізбектерді қарастыруға болады:

$$a_1 = 1, \quad a_2 = 3, \quad a_3 = 5, \quad a_4 = 7, \quad a_5 = 9, \dots \quad (1)$$

$$a_1 = 1, \quad a_2 = 2, \quad a_3 = 4, \quad a_4 = 8, \quad a_5 = 16, \dots \quad (1)$$

Келтірілген арифметикалық прогрессияның рекурренттік формуласы: $a_i = a_{i-1} + 2$.

Келтірілген геометриялық прогрессияның рекурренттік формуласы: $a_i = 2 * a_{i-1}$.

Екі жағдайда да рекурсия тереңдігі 1-ге тең (мұндай тәуелділікті бір қадамды рекурсия деп те атайды). Тармақталған формулаға келтірілген жағдайда, арифметикалық прогрессия үшін:

$$a_i = \begin{cases} 1, & \text{егер } i = 1, \\ a_{i-1} + 2, & \text{егер } i > 1. \end{cases}$$

Геометриялық прогрессия үшін:

$$a_i = \begin{cases} 1, & \text{егер } i = 1, \\ 2 * a_{i-1}, & \text{егер } i > 1. \end{cases}$$

Есеп 1. Арифметикалық прогрессияның (1) n – ші элементін есептеңіз.

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

```
int recurs_1( int n ) {  
    if ( n == 1 ) {  
        return 1;  
    }  
    return recurs_1( n - 1 ) + 2;  
}
```

```
void main() {  
    int number;  
    cout << "САН енгізіңіз: ";  
    cin >> number;  
    cout << "Арифм. прогр. элементі = " << recurs_1( number ) << '\n';  
}
```

Есеп 2. Геометриялық прогрессияның (2) алғашқы n элементін қосындылаңыз (прогрессияның алғашқы n мүшесінің қосындысын есептеу формуласын пайдаланбаңыз).

1 – ші шешу жолы сілтеме арқылы орындалған:

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

```
int recurs_2( int n, int& s ) {  
    if ( n == 1 ) {  
        s = 1;  
        return 1;  
    }  
    int b = recurs_2( n - 1, s ) * 2;  
    s += b;  
    return b;  
}
```

```
void main() {  
    int number;  
    cout << "Санды енгізіңіз: ";  
    cin >> number;  
    cout << '\n';  
    int summa;  
    cout << "Нәтижесі = " << recurs_2( number, summa );  
    cout << " сумма = " << summa << '\n';  
}
```

2 – ші шешу жолы глобалды айнымалыны қолдану арқылы орындалған:

```

#include <iostream.h>

int summa_2;

int recurs_2_2( int n ) {
    if ( n == 1 ) {
        summa_2 = 1;
        return 1;
    }
    int b = recurs_2_2( n - 1 ) * 2;
    summa_2 += b;
    return b;
}

void main() {
    int number;
    cout << "Санды енгізіңіз: ";
    cin >> number;
    cout << '\n';
    summa_2 = 0;
    cout << "Нәтижесі. = " << recurs_3_2( number );
    cout << " сумма = " << summa_2 << '\n';
}

```

Келесі сандық тізбек математикада Фибоначчи сандары ретінде танымал:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

Бұл сандардың ерекшелігі үшінші элементтен бастап, әр сан оның алдындағы екі санның қосындысына тең, яғни рекурренттік тізбектің тереңдігі 2-ге тең (екі қадамды рекурсия). Оны тармақталған түрде сипаттайық:

$$a_i = \begin{cases} 1, & \text{егер } i = 1, i = 2, \\ f_{i-1} + f_{i-2}, & \text{егер } i > 2. \end{cases}$$

Есеп 3. Алғашқы n Фибоначчи сандарын баспаға шығару.

```

#include <iostream.h>

int recurs( int n, int& s ) {
    if ( n == 2 ) {
        s = 1;
        cout << " 1, 1, ";
        return 1;
    }
    int p;
    s = recurs( n - 1, p );
    cout << s + p << ", ";
}

```

```

return s + p;
}

void main() {
    int number, temp;
    cout << "Введите число: ";
    cin >> number;
    cout << '\n';
    recurs( number, temp );
}

```

№8 лабораториялық жұмыстың тапсырмалары

1. Геометриялық прогрессияның (2) n – ші элементін есептеңіз.
2. Арифметикалық прогрессияның (1) алғашқы n элементін қосындылаңыз (прогрессияның алғашқы n мүшесінің қосындысын есептеу формуласын пайдаланбаңыз).
3. Алғашқы n Фибоначчи сандарының ішінен жұп сандардың санын анықтаңыз.
4. Рекурренттік тізбек келесі түрде анықталған:

$$a_i = \begin{cases} 1, & \text{әдетте } i = 0, \\ a_{i-1} * i + \frac{1}{i}, & \text{әдетте } i > 0. \end{cases}$$

Берілген нақты n үшін a_n мәнін есептеңіз.

5.

$$a_i = \begin{cases} 1, & \text{егер } i = 0, i = 1, \\ a_{i-2} + \frac{a_{i-1}}{i-1}, & \text{егер } i > 1. \end{cases}$$

1-20-ші элементтердің көбейтіндісін есептеңіз.

6. Рекурренттік әдісті пайдаланып, Горнер формуласы бойынша 10 дәрежелі көпмүшенің қосындысын есептеңіз. Мұндағы x – берілген нақты сан:
 $10x^{10} + 9x^9 + 8x^8 + \dots + 2x^2 + x = ((\dots((10x + 9)x + 8)x + \dots + 2)x + 1)x.$
7. Берілген нақты x пен натурал N үшін келесі тізбекті есептеңіз:
 $x/(1 + x/(2 + x/(3 + x/(\dots/(N +)) \dots).$
8. $1, \frac{1}{2!}, \frac{1}{3!}, \dots, \frac{1}{N!}$, сандық қатарының 10^{-5} мәнінен асатын барлық мүшелерін есептеп, шығарыңыз.
9. Берілген натурал N мен нақты x ($x > 0$) үшін келесі өрнектің мәнін есептеңіз:
 $\sqrt{x + \sqrt{x + \dots + \sqrt{x}}}$
10. Факториалды есептеудің рекурсивті функциясын құрыңыз.
11. Екі бүтін санның ең үлкен ортақ бөлгішін табу (ЕҮОБ) Евклид алгоритмін рекурсивті функцияның көмегімен программалаңыз.
12. $F(x)=x^n$ функциясының есептелуін рекурсияны пайдаланып шығарыңыз.

13. $F(x)=x$ теңдеуін шешу үшін рекурсивті функция құрыңыз. Оның жұмысын $\cos(x)$ және $\sqrt{x+1}$ функциялары арқылы тексеріңіз.
14. x нүктесіндегі n ретті Лежандр полиномының мәнін есептеу үшін рекурсивті функцияны құрыңыз. Лежандр полиномы келесі түрде анықталады:
- $$P_0(x) = 1,$$
- $$P_1(x) = x,$$
- $$P_n(x) = ((2n-1)P_{n-1}(x) - (n-1)P_{n-2}(x))/n.$$
15. Нүктемен аяқталған текст жолы берілген. Осы текстті кері ретпен экранға шығарыңыз.
16. Берілген нақты x пен бүтін n бойынша x^n мәнін келесі формулаға сәйкес есептейтін рекурсиялық функцияны құрыңыз:
- $$x^n = \begin{cases} 1, & n = 0, \\ \frac{1}{x^{|n|}}, & n < 0, \\ x * (x^{n-1}), & n > 0. \end{cases}$$
17. Нақты n және m сандары берілген. $EYOB(n, m) = EYOB(m, r)$ (мұндағы r – n -ді m -ге бөлгендегі қалдық) қатынасына негізделген ең үлкен ортақ бөлгішті есептеу рекурсиялық функциясын пайдаланып, $EYOB(n, m)$ табыңыз.

Блиц- тест:

1. Функция ешқандай мән қайтармаса қандай типпен сипатталады?
 - A) Void;
 - B) Int;
 - C) Double;
 - D) Char;
 - E) Bool;
2. Рекурсияның түрлері:
 - A) Тіке, жанама.
 - B) Тізбекті, тізбексіз.
 - C) Статикалық, динамикалық.
 - D) Терең, тайыз.
 - E) Шексіз, шектеулі.

Бақылауға арналған сұрақтар

1. Функциялардың рекурсивті пайдаланылуы дегеніміз не?
2. Рекурренттік тізбек түсінігі?
3. Рекурсия тереңдігі нені білдіреді?
4. Тіке және жанама рекурсияның айырмашылығы?
5. Рекурсияны қай жағдайда қолданған жөн?
6. return операторының қызметі?

7. Стек дегеніміз не?
8. Қай тізбекте стектің толтырылуы және стектен элементтердің таңдалуы орындалады?
9. Рекурсивті алгоритмде рекурсиядан шығу шарты әрдайым болу керек пе?
10. Рекурсивті алгоритм өзін-өзі «шексіз» рет шақырған жағдайда не болады?
11. Рекурсивті алгоритм арқылы орындалған есепті, рекурсивті емес алгоритммен шығаруға болады ма?

Глоссарий:

- **Рекурренттік тізбек-** (әдістемелік нұсқауды қараңыз)
- **Рекурсивті алгоритм-** (әдістемелік нұсқауды қараңыз)

Әдебиеттер:

[6],[8], [10],[12],[13],[15].

ЛАБОРАТОРИЯЛЫҚ ЖҰМЫС №9

Тақырыбы: C/C++ ТІЛІНДЕ ГРАФИКАНЫ ПРОГРАММАЛАУ

Мақсаты: - C/C++ тілінің графикалық мәліметтерді өңдеу мүмкіндіктерімен танысу.

- <graphics.h>, <conio.h> файлдарында сипатталған негізгі графикалық функциялардың прототиптерін үйрену.

Қажетті материалдар мен жабдықтар: ДК, лабораториялық жұмысты орындауға арналған әдістемелік нұсқаулар.

Лабораториялық жұмыстың мазмұны және орындалу реті:

1. C/C++ тілінің графиктік мүмкіндіктері мен графиктік режимді орнатуға қатысты әдістемелік нұсқау – материалдарды алдын ала танысып меңгеру.
2. Ұсынылған мысалды орындау және талқылау.
3. Өзіндік жеке тапсырмалардың оқытушы ұсынған нұсқасын орындау.
4. Лабораториялық жұмыстың есебін (отчет) дайындап тапсыру.

Негізгі әдістемелік нұсқаулар:

C/C++ тіліндегі графиканың негізгі элементтері

1. Драйверді және графикалық режимді таңдау
detectgraph (&gd, &gm)

функция арқылы орындалады.

2. Драйверді жүктеу, графикалық жүйені инициализациялауды
initgraph (&gd, &gm, "BGI-файлдың жолы")

функциясы атқарады, мұндағы *gd* және *gm* айнымалылары қажет драйвер мен графикалық режимнің номерлері.

Егер BGI-файлдар ағымдағы директорияда орналасса, онда *initgraph()* функциясының үшінші параметрі ретінде бос жолды беруге болады

initgraph (&gd, &gm, " ");

3. Графикамен жұмыс жасағанда қате жағдайларды өңдеу үшін
graphresult ();

функциясы қолданылады. Ол соңғы қолданылып отырған графикалық функцияның аяқталу кодын қайтарады. Қате болмаса, «0» қайтарылады.

Фон түсін мына функция арқылы басқаруға болады:

setbkcolor (Color);

Ағымдағы түсті мына функциямен тағайындауға болады:

setcolor (Color);

Графикалық режимде шрифт түрін, символдар өлшемін және бағытын тағайындау мына функция арқылы жүзеге асырылады:

settextstyle (шрифт, бағыты, өлшемі);

Шрифт параметрлері:

- | | | |
|---|-----------------|-------------------------------------|
| 0 | DEFAULT_FONT | (стандартты) |
| 1 | TRIPLEX_FONT | (триплекс типті; TRIP.CHR файлында) |
| 2 | SMALL_FONT | (кішірейтілген; LITT.CHR файлында) |
| 3 | SANS_SERIF_FONT | (түзу; SANS.CHR файлында) |
| 4 | GOTHIC_FONT | (готикалық; GOTH.CHR файлында) |

Бағыт параметрлері:

- | | | |
|---|-----------|-------------------------|
| 0 | HORIZ_DIR | (солдан оңға қарай) |
| 1 | VERT_DIR | (төменнен жоғары қарай) |

Шрифт размерін басқаратын аргумент 1 мен 10 аралығында өзгереді. Ол әрбір символының қаншаға үлкейетінін көрсетеді.

Геометриялық объектілер сызықтарының сипаттамасы мен қалыңдығын тағайындау үшін

setlinestyle (түрі, үлгісі, қалыңдығы);

функциясы қолданылады.

Қалыңдық параметрлері:

- | | | |
|---|-------------|---------------------|
| 1 | NORM_WIDTH | (бір пикселі сызық) |
| 3 | THICK_WIDTH | (үш пикселді сызық) |

Түр параметрі үшін кодтар:

- | | | |
|---|--------------|---------------------------|
| 0 | SOLID_LINE | (тұтас) |
| 1 | DOTTED_LINE | (нүктелерден) |
| 2 | CENTER_LINE | (нүктелер мен тирелерден) |
| 3 | DASHED_LINE | (пунктирлі) |
| 4 | USERBIT_LINE | (қолданушымен анықталады) |

Үлгі параметрі түр 4 тең болғанда тағайындалады. Сондықтан оны 0-ге тең деп алуға болады.

Графикалық режимде экрандағы тұйық облысты бояу мүмкіндігі бар. Бояу стилін тағайындау үшін

setfillstyle (бояу түні, түсі);

функциясы қолданылады.

Бояу түні параметрлерінің мәндері:

- | | | |
|---|----------------|---|
| 0 | EMPTY_FILL | фон түсімен штрихтау |
| 1 | SOLID_FILL | көрсетілген түспен тұтас бояу |
| 2 | LINE_FILL | горизонталь сызықтармен штрихтау |
| 3 | LTSLASH_FILL | //// сызықтармен штрихтау |
| 4 | SLASH_FILL | ///// қалыңдатылған сызықтармен штрихтау |
| 5 | BKSLASH_FILL | \\\\\\\\ қалыңдатылған сызықтармен штрихтау |
| 6 | LTBKSLASH_FILL | \\\\\\\\ сызықтармен штрихтау |
| 7 | HATCH_FILL | тікбұрыштры горизонталь бойынша штрихтау |

- 8 XHATCH_FILL қисайтып штрихтау
- 9 INTERLEAVE_FILL қисайтып штрихтау
- 10 WIDE_DOT_FILL сирек орналасқан нүктелемен толтыру
- 11 CLOSE_DOT_FILL жиі орналасқан нүктелемен толтыру

Терезелер және координаттармен жұмыс

1. Экранды тазалау мына функция арқылы жүзеге асырылады:

cleardevice ();

2. Берілген графикалық режимде горизонталь және вертикаль бойынша нүктелер координаттарының максимальді мәнін

getmaxx ();

(горизонталь бойынша максимальді координатаны қайтарады),

getmaxy ();

(вертикаль бойынша максимальді координатаны қайтарады)

функцияларының көмегімен анықтауға болады.

3. Графикалықэкрандағы терезені ашу. Экран нүктелерінің бәр массиві ішінен ішкі массив - өзінің координаттары бар тікбұрышты терезе ашуға болады. Ол төмендегі функция арқылы жүзге асырылады:

setviewport (xl, yl, x2, y2, clip);

мұндағы xl, yl – терезенің жоғары сол жақ бұрышының координаттары; x2, y2 – терезенің төменгі оң жақ бұрышының координаттары; clip – кима (отсечка).

Егер clip параметрі 1-ге тең болса, терезеде симай тұрған сурет элементтері қиылып тасталады, ал егер 0-ге тең болса терезе шекарасы өзгертіледі.

4. Графикалық терезені тазартуды төмендегі функция орындайды:

clearviewport ();

5. Ағымдағы көрсеткіш координаттарын

getx ();

(горизонталь координатты қайтарады),

gety ();

(вертикаль координатты қайтарады)

функция арқылы алуға болады.

6. Позиция көрсеткішінің орнын ауыстыруды мына функциялар жүзеге асырады:

moveto (x, y);

moverel (dx, dy);

мұндағы x, y – терезенің координаттар жүйесіндегі жаңа координаттар;

dx, dy – терезедегі көне координаттарға қатысты өсімше.

7. Параметрлер қалпына келтіру үшін

graphdefaults ();

функциясы қолданылады.

Initgraph() тағайындағандай терезе, позици көрсеткіші, түсі, сызық стилі, шрифттер қалпына келтіріледі.

8. Пикселді бейне жадыға жазуды

putpixel (x, y, цвет);

функциясы жүзеге асырады.

Мұндағы x, y – пиксел координаттары. Мысалы мына фрагменттің орындалуы нәтижесінде

```
for(i=0; i<160; i++)  
{  
    putpixel (i, 10, GREEN);  
    putpixel (160+i, 10, RED);  
}
```

жасыл-қызыл түзу сызық шығарылады.

9. Текстті шығару.

Шығарудің стандартты printf(), puts() функциялары графикалық режимде де жұмыс жасайды. Бірақ онда символдар түрі, өлшемі шектеулі. Шығарылатын текст стилі settextstyle() функциясымен беріледі. Графикалық текстті шығарудың екі функциясы бар:

outtext (sp);

ажол ғымдағы көрсеткіштен бастап шығарылады; sp – шығарылатын жолға Көрсеткіш;

outtextxy (x, y, sp);

жол (x, y) позициясынан бастап шығарылады.

Графикалық объектілерді салу.

1. Сызықтарды үш түрлі функция көмегімен салуға болады:

line (x1, y1, x2, y2);

мұндағы x1, y1 – кесінді басының координаттары; x2, y2 – кесінді аяғының координаттары.

linerel (dx, dy);

мұндағы dx, dy – ағымдағы нүкте координаттарының өсімшесі. Кесінді ағымдағы нүктеден жаңа координатты нүктеге дейін салынады. Көрсеткіш бұрынғы нүктеден жаңасына ығысады.

lineto (x, y);

мұндағы x, y – кесінді аяғының жаңа координаттары. Көрсеткіш бұрынғы нүктеден жаңасына ығысады.

Кесінділер setlinestyle() функциясы арқылы тағайындалған ағымдағы стильмен, және setcolor() функциясы арқылы тағайындалған ағымдағы түспен салынады.

3. Тіктөртбұрыш контурын мына функциямен салуға болады:

rectangle (x1, y1, x2, y2);

мұндағы x1, y1 – жоғары сол жақ бұрыш координаттары; x2, y2 – төменгі оң жақ бұрыш координаттары.

Егер квадрат салу керек болса, горизонталь және вертикаль бойынша пиксельдер санын коррекциялау керек. Пиксельдер пропорциясын

getaspectratio (&xasp, &yasp);

функциясының көмегімен анықтауға болады.

xasp/yasp қатынасы пиксель өлшемдрінің горизонталь және вертикаль бойынша қатынасы.

4. Сынық сызықты

drawpoly (төбелер саны, бүтін массивке Көрсеткіш);

функциясы салуға мүмкіндік береді.

Массивтің әрбір екі саны сынық сызықтың кезектегі қос координатын білдіреді.

5. Қисықтарды салу үшін мынадай функциялар қолданылады:

Вычерчивание окружности

circle (x, y, радиус);

мұндағы x, y – центр координаттары; радиус – горизонталь бойынша шеңбердің пиксельдегі радиусы.

Шеңбердің доғасын салу

arc (x, y, бұрыш_басы, бұрыш_соңы, радиус);

Бұрыштар градуспен өрнектеледі және сағат тіліне қарсы есептеледі. Бұрыш мәндері [0..360] интервалынан сәйкес эквивалентті мәндерге түрленеді. Сонымен, arc (x, y, -45, 45, r) және arc (x, y, 675, -315, r) бірдей шеңбердің төрттен бір бөлігін салады.

Эллипстің доғасын сызу

ellipse (x, y, бұрыш_басы, бұрыш_соңы, rx, ry);

мұндағы rx, ry – пиксельмен берілген эллипстің жарты осьтері.

6. Тіктөртбұрышты бояуды

bar (x1, y1, x2, y2);

функциясы жүзеге асырады. мұндағы x1, y1 – жоғары сол жақ бұрыш координаттары; x2, y2 – төменгі оң жақ бұрыш координаттары. Бояу стилі setfillstyle() функциясымен тағайындалады.

7. Көпбұрыш салу және бояуға

fillpoly (n, указатель на массив целых);

функциясы мүмкіндік береді. Мұндағы n – төбелер саны. Әрбір төбенің координаттары бүтін типті екі шамамен беріледі. Бұл функция төбелер тізімінің бірінші нүктесін соңғысымен байланыстырады, тұйықтайды және оны бояйды. Бояу шаблон мен түсі setfillstyle() и setfillpattern() функцияларымен берілуі мүмкін.

8. Контурлы боялған эллипсті мына функция арқылы салуға болады:

fillellipse (x, y, rx, ry);

мұндағы x, y – центр координаттары; rx, ry – пиксельмен берілген эллипстің жарты осьтері

9. Контурлы боялған шеңбер секторын

pieslice (x, y, бұрыш_басы, бұрыш_соңы, радиус);

функциясы арқылы салуға болады.

10. Контурлы боялған эллипстік секторды

sector (x, y, нач_угол, кон_угол, gx, gy);
функциясын пайдаланып салуға болады.

11. Тұйық облысты бояу

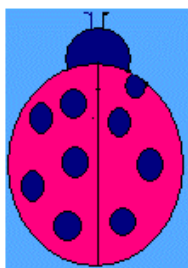
floodfill (x, y, граница);

функциясымен жүзеге асырылады. мұндағы x, y – облыс ішіндегі нүкте координаттары.

Контур түсі шекара параметрінің мәнімен сәйкес болуы керек. Бояу (түсі мен типі) setfillstyle() функциясымен тағайындалады.

Тапсырмалар

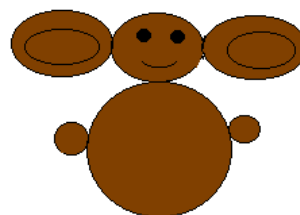
Вариант1



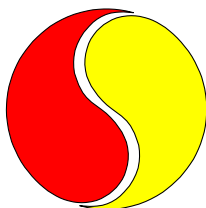
Вариант 2



Вариант 3



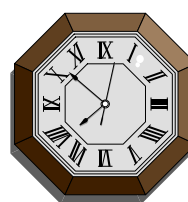
Вариант 4



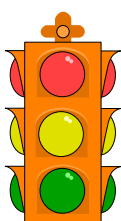
Вариант 5



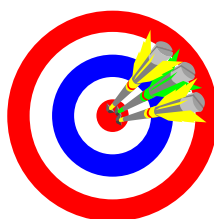
Вариант 6



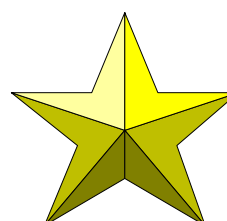
Вариант 7



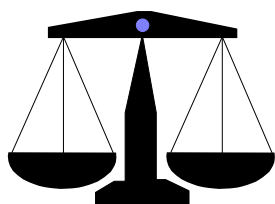
Вариант 8



Вариант 9



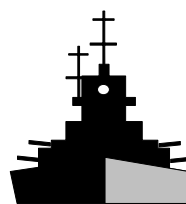
Вариант 10



Вариант11



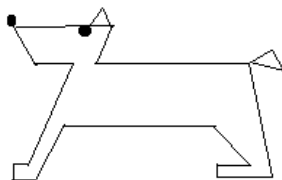
Вариант 12



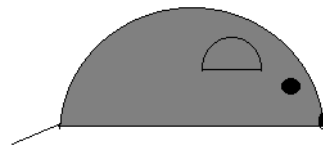
Вариант 13



Вариант 14



Вариант 15



Блиц- тест:

1. Ағымдағы драйвер үшін графикалық режимнің максимал мәнін қайтаратын функция:

- A) getmaxmode ();
- B) getmaxxy ();
- C) maxx ();
- D) getmax ();
- E) maxmode ();

2. Тексттік режимнен графиктік режимге қайта ауысу функциясы:

- A) setgraphmode ();
- B) restorecrtmode ();
- C) detectgraph ();
- D) initgraph ();
- E) grapherrormsg ();

3. Тұйық облысты бояу функциясы:

- A) floodfill (x,y, шекара);
- B) floodfill (x,y);
- C) fillpoly(x,y, шекара);
- D) fillellips(x,y, rx, ry);
- E) settextfill (x,y, шекара);

Бақылау сұрақтары

1. Пиксел дегеніміз не?
2. Графикалық ақпарат қалай сақталады?
 1. Графикалық режимге қалай ауысуға болады?
 2. Тексттік режимге қалай оралуға болады?
 3. Шрифттерді қалай таңдауға болады?
6. Сызық сипаттамасы мен қалыңдығын қалай тағайындауға болады?
7. Бояу стилін қалай тағайындауға болады?
8. Графикалық экранда терезені қалай ашуға және қалай жабуға болады?
9. Горизонталь және вертикаль бойынша нүктелердің максимальді координат мәндерін қалай алуға болады?
10. Терезенің координаттар жүйесінде ағымдық координаттарды алуға болады?
11. Терезеге текстті қалай шығаруға болады?
12. Тұйық фигураны қалай бояуға болады?

Глоссарий:

- **Пиксель-** (әдістемелік нұсқауды қараңыз).
- **Графитик режим** - (әдістемелік нұсқауды қараңыз)
- **Түстер-** (әдістемелік нұсқаудағы түстер кестесін қараңыз)
- **Бейне адаптер-** құрылғы
- **Координаталық жүйе** – экранның пиксельдер жиынтығын реттеп санауға қатысты жүйесі.
- **Графика** - (әдістемелік нұсқауды қараңыз)

Әдебиеттер:

[14],[13],[15].

ЛАБОРАТОРИЯЛЫҚ ЖҰМЫС №10

Тақырыбы : КӨРСЕТКІШТЕР.

Мақсаты:

Көрсеткіштер түсінігін тереңірек зерттеу,
Көрсеткіштерді қолдану машығын қалыптастыру;
Мысалдар арқылы бекіту.

Қажетті материалдар мен жабдықтар: ДК, лабораториялық жұмысты орындауға арналған әдістемелік нұсқаулар.

Лабораториялық жұмыстың мазмұны және орындалу реті:

1. C/C ++ тілінің көрсеткіштер мен сәйкес типтері, адрестер ұғымдарына қатысты әдістемелік нұсқау – материалдарды алдын ала танысып меңгеру.
2. Ұсынылған мысалды орындау және талқылау.
3. Өзіндік жеке тапсырмалардың оқытушы ұсынған нұсқасын орындау.
4. Лабораториялық жұмыстың есебін (отчет) дайындап тапсыру.

Әдістемелік нұсқау:

Көрсеткіш- бұл компьютердің жадысын басқаруға қосымша мүмкіндік беретін конструкция.

Көрсеткіш- бұл айнымалының жадыдағы адресі деп түсінуге болады.

Компьютер жадысын нөмерленген ұяшықтарға бөлінетінін еске түсірсек (байттар), онда программаның айнымалылары тізбектелген ұяшықтар арқылы жүзеге асырылады. Айнымалаға мәлімет алу мүмкіндігін атын атамай оның адресін пайдалануға болады, адрес дегеніміз айнымалының жадыда орналасқан ұяшықтардың біріншісінің адресі. Осы мақсатпен пайдаланған айнымалының адресі Көрсеткіш деп аталады.

Көрсеткіш айнымалыда сақталу мүмкін. Он жадыдағы адрес деп анықталғасын, оны сақтау үшін `int` немесе `double` типті айнымалылар сәйкес келетін сияқты. Бірақ негізінде олай емес. Көрсеткіштер үшін ерекше типті айнымалылар қажет. Ол айнымалыларды хабарлау үшін арнаулы `*` символы пайдаланады. Мысалы : `double *p;` Бұл оператор `p` айнымалыны `double` типті деп хабарлап тұр. Келесі оператор

`int *r, *p, v1,v2;` `p` және `r` айнымалыларды `int` типті Көрсеткіштер, ал `v1,v2` айнымалыларды `int` типті деп хабарлағаны.

Айнымалының адресін алу үшін `&` операторын пайдаланады: `p1=&v1;`

`p1` айнымалыға `v1` айнымалысына көрсеткіш меншіктенелі.

C++ тілінде `p1` айнымалысы нұсқайтын айнымалы қатынас жасау былай орындалады: `*p1`.

Келесі кодты қарастырайық:

```

v1=0;
P1=&v1;
*p1=42;
Cout <<v1<< endl;
Cout <<*p1<<endl;

```

Орындалу нәтижесі экранға шығарылған кезде:

42

42

Көрсеткіш типтері:

Егер функцияда `int` типті айнымалыға көрсеткіш болатын, типті параметр қажет болса, ол Көрсеткіш типін `int*` деп көрсетеді. Мысалы:

```
Void manipulatePointer(int* p);
```

Ал егер сол типті айнымалыны хабарлағыңыз келсе, онда `*` символын типіне емес, айнымалының атына қосып жазу керек:

```
Int *p, *p2;
```

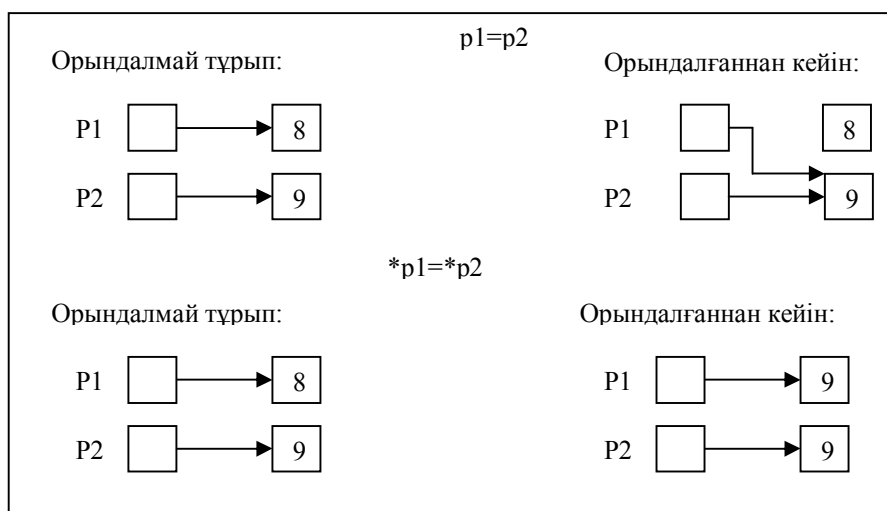
Бір айнымалы-Көрсеткіштың мәнін екінші айнымалы-Көрсеткішқа меншіктеуге болады, нәтижесінде бірінші айнымалыда сақталған адрес екінші айнымалыға көшіріледі:

Мысалы: Егер `p1` айнымалыда `v1` айнымалының адресі болса, онда оператор `p2=p1`; бұл адресі `p2` айнымалыға меншіктейді де, ол да `v1` айнымалыға Көрсеткіш болады. Егер `v1` айнымалысы ішінде 42 саны болса, оператор:

```
cout<<*p2;
```

экранға 42 санын шығарады:

Көрсеткіш-айнымалылары бар, меншіктеу операторын пайдалану.



Мысал:

```

Cin >>*p1;
*p1=*p2+7;
Cout <<*p1;

```

пернетақтадан `int` типті мәнді орналастырады де, 7 санын қосады және нәтижесін шығарады.

Мысал: Көрсеткіштерге қолданылатын негізгі операциялар:

//Көрсеткіштермен және динамикалық айнымалылармен орындалатын операцияларды демонстрациялау программасы//

```
#include <iostream>
```

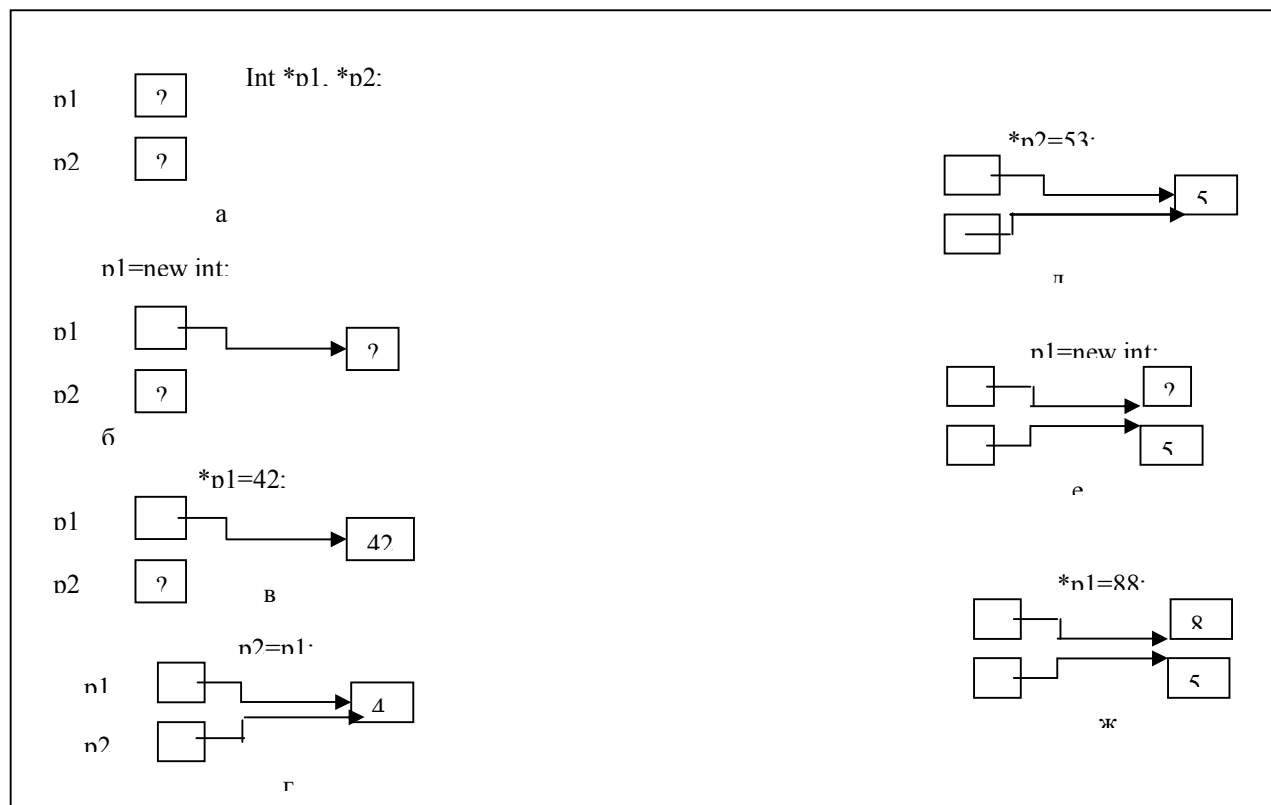
```
Using std:: cout;
```

```
Using std::endl;
```

```
Int main()
```

```
{
Int *p1, *p2;
p1=new int;
*p1=42;
p2=p1;
cout << "*p1=="<<*p1<<endl; //Жаңа
                               жолға көшу үшін endl немесе "\n" жазылуы мүмкін, ол C++ версиясына
                               байланысты
cout << "*p2=="<<*p2<<endl;
*p2=53;
Cout << "*p1=="<<*p1<<endl;
Cout << "*p2=="<<*p2<<endl;
p1=new int;
*p1=88;
Cout<< "*p1=="<<*p1<<endl;
Cout<< "*p2=="<<*p2<<endl;
Cout<< "Hope you got the point of this example!\n";
Return 0;
}
```

Диалог мысалы:



*p1=
=42
*p2=
=42
*p1=
=53
*p2=
=53
*p1=
=88
*p2=
=53
Hope
you
got
the
point
of
this

example!

2.Массивтер-айнымалылар және көрсеткіштер- аynamалылар.

Массив типті аynamалы жадыдағы массивтің бірінші элементінің адресі деп түсіндірілді. Енді сіздің біліміңіз аynamалы басқа аynamалының адресін іште сақтайтын болса ол Көрсеткіш деген анықтамамен толықтырылды. Сондықтан C++-те массив типті аynamалы, массивтің бірінші индекстелген аynamалысына Көрсеткіш деп түсінуге де болады.

Мысал:

```
//Массив типті аynamалы Көрсеткіш екенің көрсететін программа//
#include <iostream>
Using std::cout;
Using std::endl;
typedef int* IntPtr;
Int main()
{
IntPtr p;
Int a[10];
Int index;
For (index=0; index<10;index++)
a[index]=index;
p=a;
for (index=0; index<10;index++)
cout<<<p[index]<<" ";
cout<<"\n";
for (index=0; index<10; index++)
p[index]=p[index]+1;           // P массивінің элементтерің өзгерткенде
for (index=0; index<10; index++) //a массивінің элементтері өзгереіне
cout<<a[index]<<" ";         // назар аударыңыз
cout<<"\n";
return 0;
}
```

Диалог мысалы:

0 1 2 3 4 5 6 7 8 9

1 2 3 4 5 6 7 8 9 10

A int типті аynamалыға Көрсеткіш болғандық (a[0]) , a аynamалының мәнін p Көрсеткіш аynamалыға меншіктеуге болады: p=a;

Онда p[0],p[1],... өрнегі ,бұл a массив элементтеріне a[0],a[1],... сілтеме болып табылады.

3. Көрсеткіштермен орындалатын арифметикалық операциялар:

Программада төменде берілген код бар болсын делік:

```
Typedef double* DoublePtr;
```

```
DoublePtr d;
```

```
D= new double[10];
```

Орындалу нәтижесінде d аynamалысы d[0] элементінің адресін сақтайды. d+1 өрнегі d[1] элементінің адресін, d+2 - d[2] элементінің т.б. сақтайды. Мысалы: егер

double типті айнымалыға 8 байт (жадының сегіз ұяшығы) қажет болса және d 2001 адресің іште сақтаса, онда d+1 өрнегі 2009 адресін қайтарады. Түсіндірілген арифметикалық операциялар бұл массив элементтеріне альтернативтік мәлімет алу мүмкіндігі. Сонымен, arraySize-d динамикалық массивтің өлшемін нұсқайтын айнымалы, циклі

```
For (int i=0; i<arraySize; i++)
```

```
Cout << *(d+i)<<" ";
```

Массивтің ішіндегі мәліметтерді экранға шығарады. Бұл цикл төмендегі циклге эквивалентті:

```
for (int i=0; i<arraySize; i++)
```

```
cout <<d[i]<<" ";
```

Көбейту және бөлу операциялары Көрсеткіштерға қолданылмайды..

d++ өрнегі, d Көрсеткіш-айнымалын мәнің көбейтеді, d-- өрнегі, Көрсеткіш-айнымалыға, алдында тұрған элементтің адресің кіргізіп, берілген мәнді азайтады.

Өзіндік жеке тапсырмалар:

1. Оперативтік жадыда вектор Int X[10] векторы b7f0 адресінен бастап орналасқан. Мына өрнектердің мәнін табыңыз:

а) X+1;

б) X+5;

в) X-4;

2. Программада Int P[]={0.2.4.5.6.7.9.12} массиві хабарланған. Мына өрнектердің мәнін табыңыз:

а) P[3];

б) *P;

в) *[P+4];

г) *(P+P[2]);

3. Өсу тәртібімен, үш айнымалының мәнің сұрыптау функциясың жазыңыз:

4. [0,1] диапазонына түсетін кездейсоқ сандармен, бүтін санды, бір өлшемді массивті толтыру функциясың жазыңыз.

5. Заттық, бір өлшемді массивтің элементтерінің орта мәнің есептейтін функцияны жазыңыз. Осы функцияны, ең үлкен орта мәнді жолды анықтау үшін пайдаланыңыз.

Блиц-тест:

1. Int *p;

Int a;

...

p=&a;

p++;

p--;

...

фрагментінің орындалу нәтижесінде p қайда сілтейді?

A) A мәні орналасқан орынға;

B) Ешқайда;

C) A орналасқан орыннан кейінгі ұяшыққа;

D) A орналасқан орынға дейінгі ұяшыққа;

Е) Р ұяшығына.

2.

...

```
int *p;
```

```
int x[]={5,4,7};
```

```
p=x;
```

```
p++;
```

```
printf("%i",*p);
```

...

Орындалу нәтижесінде экранға не шығады?

A) 4

B) 5

C) 7

D) 6

E) 8

Бақылау сұрықтары:

1. C++ тілінде Көрсеткіш деп нені атайды ?

2. * операторын пайдалану үш мысалын келтіріп және әр қайсысы қалай жұмыс жасайтының түсіндіріңіз.

3. Келесі код нені шығарады ?

```
Int *p1, *p2;
```

```
p1=new int;
```

```
p2=new int;
```

```
*p1=10;
```

```
*p2=20;
```

```
cout<<*p1<<" "<<*p2<<"\n";
```

```
p1=p2;
```

```
cout <<*p1<<" "<<*p2<<"\n";
```

```
*p1=30;
```

```
cout<<*p1<<" "<<*p2<<"\n";
```

Программа нәтижесі қалай өзгертінеді, егер p1=30; жолы *p2=30; жолына ауыстырылса.

4. Төменгі код нені шығарады?

```
Int *p1,*p2;
```

```
p1=new int;
```

```
p2=new int;
```

```
*p1=10;
```

```
*p2=20;
```

```
cout <<*p1<<" "<<*p2<<"\n";
```

```
*p1=*p2;//бұл жол 3 сұрақтағы дай емес.
```

```
cout<<*p1<<" "<<*p2<<"\n";
```

```
*p1=30;
```

```
cout<<*p1<<" "<<*p2<<"\n";
```

Әдебиеттер:

[1],[4],[7],[8],[10-13],[15].

ЛАБОРАТОРИЯЛЫҚ ЖҰМЫС №11

Тақырыбы: СТРУКТУРАЛАР

Мақсаты: Студенттерді C/C++ тіліндегі

- структуралық тип ұғымымен таныстыру, структура типті берілгендерді енгізу, шығару және структуралық мәліметтерді қолданып программалау дағдыларын қалыптастыру;
- біріктіру ұғымымен таныстыру, біріктіруді қолданып программалау дағдыларын қалыптастыру;
- программалауда *Typedef* құралын қолдану.

Қажетті материалдар мен жабдықтар: ДК, C/C++ тілінің IDE ортасы, лабораториялық жұмысты орындауға арналған әдістемелік нұсқаулар.

Лабораториялық жұмыстың мазмұны және орындалу реті:

1. C/C++ тілінде циклдық алгоритмдерді программалауға қатысты әдістемелік нұсқау – материалдарды алдын ала танысып меңгеру.
2. Ұсынылған мысалды орындау және талқылау.
3. Өзіндік жеке тапсырмалардың оқытушы ұсынған нұсқасын орындау.
4. Лабораториялық жұмыстың есебін (отчет) дайындап тапсыру.

Әдістемелік нұсқаулар:

Структура

Структура – бір атпен біріктірілген бір немесе бірнеше әр түрлі типті айнымалылар тобы. Структура ***struct*** қызметші сөзімен құрылады. Структураның жалпы құрылымы төмендегідей:

```
struct тег {  
    тип1 иден1;  
    тип2 иден2;  
    ...  
    типN иденN;  
};
```

мұндағы *тег* мәліметтердің жаңа типі болып саналады және оны айнымалыны сипаттауда қолдануға болады.

Сипатталған структура негізінде айнымалы құру үшін

```
struct <тег> <айнымалы>
```

C++ тілінде *struct* қызметші сөзін көрсетпесе де болады:

<тег> <айнымалы>

Айнымалыны структураны құру кезінде де анықтауға болады:

```
struct тег {  
    mun1 иден1;  
    mun2 иден2;  
    ...  
    munN иденN;  
} <айнымалы>;
```

Егер структура типті басқа айнымалылар құрылмайтын болса, структура тегін көрсетпеуге болады. Мұндай структура *атаусыз (безымянный)* деп аталады:

```
struct {  
    mun1 иден1;  
    mun2 иден2;  
    ...  
    munN иденN;  
} <айнымалылар тізімі>;
```

Структуралар іштестірілген де болуы мүмкін:

```
struct тег1 {  
    mun1 иден1;  
    mun2 иден2;  
    mun3 иден3;  
};  
struct тег2 {  
    mun1 иден1;  
    mun2 иден2;  
    struct тег1 иден3;  
};
```

Структура элементтерін **нүкте** операторының көмегімен алуға болады:

<айнымалы>.<элемент>

Кейбір жағдайларда структураны функция денесінде қолдану қажеттігі туындайды. Ондай жағдайда функция прототипінде структураны функция параметрі ретінде беру керек (C++ тілінде *struct* қызметші сөзін көрсету міндетті емес):

```
// C және C++ тілдерінде  
<функция туні> <функция аты> (struct <тег> <айнымалы>)  
// Тек қана C++ тілінде  
<функция туні> <функция аты> (<тег> <айнымалы>)
```

Структуралар массиві. Структуралар массивін сипаттау структура типті айнымалыны сипаттауға ұқсас. Мысалы, *struct <тег> <айнымалы-массив>*

Структураларға көрсеткіш. Көрсеткіш келесі түрде сипатталуы мүмкін:

*struct <тез> *<айнымалы>*

Бұндай жағдайда структуралық айнымалының кез-келген элементін мына әдістердің бірімен алуға болады:

(<айнымалы>).<элемент>* немесе *<айнымалы> -> <элемент>*

Біріктіру

Біріктіру структураған ұқсас. Жазылу ерекшелігі: **struct** қызметші сөзінің орнына **union** сөзі қолданылады.

```
union тез {
    тун1 иден1;
    тун2 иден2;
    ...
    тунN иденN;
};
```

Біріктірудің структурадан басты айырмашылығы жадыны ұйымдастыру тәсілінде, яғни біріктірудің барлық элементтері жадыда бір байттан басталады. Айталық, программада төмендегідей структура сипатталған болсын:

```
struct P
{ int i;
  char ch;
  long int L;
};
```

Бұл элементтердің жадыда орналасуы келесі түрде болады:

байт	байт	байт	байт	байт	байт	байт
i		ch	L			

Структураның жалпы өлшемі оның өрістерінің ұзындықтарының қосындысына тең.

Енді, дәл осындай біріктіруді қарастырайық:

```
union P
{ int i;
  char ch;
  long int L;
};
```

Бұл элементтердің жадыда орналасуы келесі түрде болады:

байт	байт	байт	байт
------	------	------	------

ch	
i	
L	

Біріктірудің жалпы өлшемі оның ең үлкен өрісінің өлшеміне тең.

Typedef қызметші сөзінің көмегімен бұрыннан бар типтердің негізінде жаңа тип құруға болады. Бұл программа текстін ықшамдау үшін қажет. Мысалы,

Typedef int butin;

өрнегі *butin* сөзін *int* типінің синонимі ретінде қабылдауға мүмкіндік береді.

Мысал.

Мекеме қызметкерлері туралы мынадай мәліметтер берілген: аты-жөні, мекен-жайы, еңбек стажы, еңбек ақысы. Қызметкердің мекен-жайы көшенің атынан, үйінің және пәтерінің номерінен тұрсын.

Еңбек стажы 10 жылдан артық қызметкерлердің еңбек ақысын 20%-ке арттырып, фамилиясымен қоса экранға шығаратын программа жазу.

Есептің қойылымы. Программада іштестірілген структура қолданамыз. *Work* структурасы қызметкерлер туралы мәліметтерден тұрсын. Структура элементтері: *fam*, *address*, *stazh*, *zarp*. *Address* — *ul*, *dom*, *kv* элементтерінен тұратын структуралық айнымалы. *n* — қызметкерлер саны.

Математикалық моделі.

$$zarp = zarp * 1.2, \text{ егер } stazh \geq 10$$

Программасы.

```

#include<stdio.h>
struct adr {
    char ul[15];
    int dom;
    int kv;
};
struct work {
    char fam[25];
    struct adr address;
    int stazh;
    float zarp;
} rab[10];
void main()
{ int i,n;
  printf("\n Қызметкерлер саны n=");
  scanf("%i",&n);
  for(i=0;i<n;i++)
  { printf("%i- қызметкердің фамилиясы: ",i); scanf("%s",rab[i].fam);
    printf("%i- қызметкердің тұратын көшесі: ",i);
    scanf("%s",rab[i].address.ul);
    printf("%i- қызметкердің үйінің номері: ",i);
    scanf("%i",&rab[i].address.dom);
    printf("%i- қызметкердің пәтерінің номері: ",i);
    scanf("%i",&rab[i].address.kv);
    printf("%i- қызметкердің еңбек стажы: ",i); scanf("%i",&rab[i].stazh);
    printf("%i- қызметкердің еңбек ақысы: ",i); scanf("%f",&rab[i].zarp);
  }
  printf("\n Еңбек стажы 10 жылдан асатын қызметкерлер");
  printf("\n Фамилиясы \t Еңбек ақысы");
  for(i=0;i<n;i++)
  if (rab[i].stazh>=10)
  { rab[i].zarp*=1.2;
    printf("\n %s \t %f\n Мекен-жайы: %s \t %i-%i ", rab[i].fam,
    rab[i].zarp,
    rab[i].address.ul, rab[i].address.dom, rab[i].address.kv);
  }
}

```

C++ тілінде

```
#include<iostream.h>
struct adr {
    char ul[15];
    int dom;
    int kv;
};
struct work {
    char fam[25];
    struct adr address;
    int stazh;
    float zarp;
};

void main()
{ work rab[10];
  int i,n;
  cout<<"\nҚызметкерлер саны n=";   cin>>n;
  for(i=0;i<n;i++)
  { cout<<i<<"-қызметкердің фамилиясы: "; cin>>rab[i].fam;
    cout<<i<<"-қызметкердің тұратын көшесі: "; cin>>rab[i].address.ul;
    cout<<i<<"-қызметкердің үйінің номері: "; cin>>rab[i].address.dom;
    cout<<i<<"-қызметкердің пәтерінің номері: "; cin>>rab[i].address.kv;
    cout<<i<<"-қызметкердің еңбек стажы: "; cin>>rab[i].stazh;
    cout<<i<<"-қызметкердің еңбек ақысы: "; cin>>rab[i].zarp;
  }
  cout<<"Еңбек стажы 10 жылдан асатын қызметкерлер";
  cout<<"Фамилиясы \t Еңбек ақысы";
  for(i=0;i<n;i++)
  if (rab[i].stazh>=10)
  { rab[i].zarp*=1.2;
    cout<<"\n"<<rab[i].fam<<"\t"<<rab[i].zarp<<"\nМекен-жайы: "
    <<rab[i].address.ul<<"\t"<<rab[i].address.dom<<"-"<<rab[i].address.kv;
  }
}
```

Өзіндік жеке тапсырмалар:

1. Информатикадан олимпиадаға қатысқан студенттердің мәліметтері белгілі. 30 баллдан жоғары алған студенттердің тізімін шығару.
2. Кітапхананың оқырмандары туралы мәліметтер белгілі: фамилиясы, мекен-жайы, жұмыс орны, кітапты алған уақыты, кітапты тапсыру уақыты. Кітапханаға қарыз оқырмандардың фамилиясын, мекен-жайын және жұмыс орнын көрсету.

3. Тіс дәрігеріне келушілердің тізімінен фамилиясының бас әрпі "Б"-дан "Л"-ға дейінгі аралықта жатқан адамдардың жасы мен диагнозын шығару программасын құрыңыз.
4. Темір жол кассасында төмендегідей мәліметтер белгілі: поездың номері, баратын жері, кету уақыты, бос орындар саны. Астана қаласына баратын поездың номері мен кету уақытын шығару, егер поезде бос орын болса.
5. Музыка мектебінің скрипкада ойнайтын оқушыларының тізімін шығару. Олардың музыкамен неше жыл айналысатындығын және конкурстарға қатысқан-қатыспағандығын көрсету.
6. Оқушылар туралы келесі мәліметтер берілген: аты-жөні, бойы. Кластағы ең ұзын және ең қысқа оқушыны, сонымен қатар неше оқушының баскетболға қатысатындығын анықтау, егер баскетболисттің бойы 170 см.-ден артық болуы керек болса.
7. Автомобильдер туралы мәлімет берілген: автомобиль иесінің аты-жөні, автомобиль бағасы, маркасы және номері. Автомобильдер туралы мәліметті бағасының өсу реті бойынша баспаға шығару қажет.
8. Фирма қызметкерлері туралы мәлімет берілген: қызметкердің аты-жөні, туған жылы, адресі, телефон номері. Жасы қызметкерлердің орташа жасынан аспайтын қызметкерлер туралы мәліметті баспаға шығар.
9. Экспортқа шығарылатын тауарлар туралы мәлімет берілген: тауардың аты, тауарды импортқа шығаратын елдің аты және әкелінетін партияның көлемі. Берілген тауардың қай елге экспортқа шығарылатындығын және оның жалпы көлемін тап.
10. Тұрғындар санағының нәтижелері берілген болсын. 1990 ж. кейін туылған тұрғындардың фамилиясын, атын және жалпы санын анықтайтын программа құрыңыз.
11. Дәріханадағы дәрілер туралы келесі мәліметтер белгілі болсын: дәрілік препараттың аты, бағасы, саны, сақтау мерзімі. Дәріханадағы барлық дәрілердің санын, қанша тұратынын және сақтау мерзімі 3 айдан артық препараттарды анықтау программасын құрыңыз.
12. Қысқы сессия нәтижесі бойынша ешбір сабақтан “үштік” баға алмаған студенттердің фамилиясын, тобын және орта балын анықтау.
13. Кітаптар туралы мәліметтер белгілі: авторы, кітаптың аты және шыққан жылы. Берілген автордың 1960ж бастап шыққан кітаптарының атын анықтау.
14. Ойыншықтың аты, құны және қанша жастағы балаға арналғандығы белгілі. 6 жастағы балаларға арналған қуыршақтың құнын анықтау.

15. Автосалондағы автомобильдер туралы мәліметтер: автомобильдің моделі, шыққан жылы, түсі және бағасы. Қызыл түсті Toyota Camri автомобилінің бағасы мен шыққан жылын анықтау.
16. Туристік фирма туралы төмендегідей мәліметтер берілген: фирманың аты, құрылған жылы, мекен-жайы, телефоны, спецификасы, яғни қай елдерге баруға болатындығы және жолдама (путевка) құны. Европа елдеріне баруды ұйымдастыратын барлық туристік фирмалардың мәліметтерін экранға шығару.
17. «Қазақтелеком» телефон станциясының абоненттері туралы мәліметтер белгілі: абоненттің аты-жөні, мекен-жайы, телефонды қондырған жылы, телефон номері. Соңғы 5 жыл ішінде телефон қондырған абоненттердің санын анықтап, экранға аты-жөнін және мекен-жайын шығару.
18. Әуежай кассасында төмендегідей мәліметтер белгілі: рейстің номері, баратын жері, ұшу уақыты, ұшатын күндері (күн сайын, жұп күндері, тақ күндері). Клавиатурадан енгізілген күні Лондон қаласына ұшатын рейстің номері мен ұшу уақытын анықтайтын программа құру.
19. Фирма қызметкерлері туралы мәліметтер белгілі: аты-жөні, туған жылы, қызметі, оклады, стажы. Зейнетақы қорына (10%), кәсіподақтық төлемге (15%) және стажына байланысты төленетін қосымша ақыны ескеріп, әрбір қызметкердің жалақысын есептеу.

$$Doplata = \begin{cases} 7000, & \text{егер стаж} \geq 10 \\ 5000, & \text{егер } 2 \leq \text{стаж} \leq 10 \\ 2000, & \text{егер стаж} \leq 2 \end{cases}$$

20. Төмендегі мәліметтерден тұратын *Сабақ кестесі* айнымалысын сипаттау:

- апта күні;
- сол күндегі парлар саны;
- пардың басталу және аяқталу уақыты;
- пән аты;
- оқытушының аты-жөні.

“Информатика” пәніне қатысты сабақтар туралы толық информацияны экранға шығару. **Бақылау сұрақтары**

1. C/C++ тілінің қандай қосымша типтерін білесіңдер?
2. Структура дегеніміз не?
3. Тегі көрсетілмеген структура қалай аталады?
4. Структураны қандай түрде сипаттауға болады?
5. Біріктірудің структурадан айырмашылығы неде?
6. Жадыны үнемдеу кезінде нені қолданған тиімді? (структураны, әлде біріктіруді)

7. Біріктіруді сипаттау үшін қандай қызметші сөз қолданылады?
8. *Typedef* қызметші сөзі не үшін қолданылады?

Блиц- тест:

1. Атаусыз структура деп ...

- A) Анықтау барысында тегі көрсетілмеген структураны айтады.
- B) Бірдей типті айнымалылардан тұратын структураны атайды.
- C) Элементтерінің бірі көрсеткіш болатын структураны атайды.
- D) Элементтерінің бірі структура болатын структураны атайды.
- E) Дұрыс жауап жоқ.

2. *Иштестірілген структура* дегеніміз не?

- A) Элементтерінің бірі структура болатын структура
- B) Элементтерінің бірі көрсеткіш болатын структура
- C) Элементтерінің бірі символдық массив болатын структура
- D) Тегі көрсетілмеген структура
- E) Дұрыс жауап жоқ

Глоссарий:

- **Структура-** (әдістемелік нұсқауды қара)
- **Структура тегі-** (әдістемелік нұсқауды қара)
- **Структураға көрсеткіш-** (әдістемелік нұсқауды қара)
- **Атаусыз структура-** (әдістемелік нұсқауды қара)

Әдебиеттер:

[1-6],[12],[15].

Лабораториялық жұмыс №12

Тақырыбы: C/C++ ТІЛІНДЕГІ ФАЙЛДЫҚ ЕНГІЗУ-ШЫҒАРУ МҮМКІНДІКТЕРІ

Мақсаты: C/C++ тіліндегі файлдан оқу және файлға жазу мүмкіндіктерін үйрету.

Қажетті материалдар мен жабдықтар: ДК, C/C++ тілінің IDE ортасы, лабораториялық жұмысты орындауға арналған әдістемелік нұсқаулар.

Лабораториялық жұмыстың мазмұны және орындалу реті:

1. C/C++ тілінде мәліметтерді файлдан оқу немесе файлға жазу мүмкіндіктеріне қатысты әдістемелік нұсқау – материалдарды алдын ала танысып меңгеру.
2. Ұсынылған мысалды орындау және талқылау.
3. Өзіндік жеке тапсырмалардың оқытушы ұсынған нұсқасын орындау.
4. Лабораториялық жұмыстың есебін (отчет) дайындап тапсыру.

Әдістемелік нұсқау:

C тілінде:

Файлдан оқу және файлға енгізу үшін ең алдымен файл fopen функциясының көмегімен ашылуы тиіс. Бұл функция операциялық жүйе арқылы орындалатын әрекеттерді ұйымдастыру жұмысын орындайды және файлмен мәлімет алмасуға арналған **көрсеткішті** қайтарады.

Ал, файлға көрсеткіш файл туралы информациялардан тұратын құрылымға (жазбаға) сілтейді. Мұндағы информация мынадай сұрақтарының жауабынан тұрады:

- буфер адресі,
- буфердегі ағымды литердің күйі,
- файлдан оқуға немесе жазуға ашық па?,
- файлдың соңғы таңбасы кездесті ме?

Мұндай құрылым сипаттамасы **<stdio.h>** кітапханасындағы FILE типінде беріледі. Қолдану үшін мынадай декларация берілсе жеткілікті:

FILE * fp;

FILE * fopen (char * name, char * m);

Мұндағы: fp – FILE типіндегі көрсеткіш, ал fopen FILE – ге көрсеткішті қайтарады. Fopen функциясы мына түрде қолданылады:

fp = fopen (name, m);

Мұндағы: name – файлдың атын меншіктейтін жол;

ал, m – файлды қолдану режимі; яғни бұл да жол, қолданушы файлды қалай қолданатынын білдіреді, төмендегідей мәндердің бірін иелене алады:

" r " – (read) оқу режимі;

" w " – (write) жазу;

" a " - (apprnd) толықтыру;

Кейбір жүйеде тексттік және бинарлық файлдар болып жіктеледі, бұл жағдайда режим жолына "b" (binary – бинарлық) немесе "t"(тексттік) таңбасы тіркеледі. Файлмен жұмыс жасау барысында қате кездессе, онда fopen функциясы NULL мәнін қайтарады.

Файлдан оқу - жазу әрекетінің ең қарапайым тәсілі – **getc** және **putc** функцияларын қолдану.

int getc (FILE * fp) - файлға көрсеткіштен тәуелді және файлдан кезекті литерді қайтарады.

Ал, **int putc (int c, FILE * fp)** – функциясы c литерін fp файлына жазады және жазылған литерді немесе EOF мәнін қайтарады (егер қате болса).

Файлға форматтап енгізу – шығару функциялары:

int fscanf (FILE * fp, char * format,...)

int fprintf (FILE * fp, char * format,...)

Бұлар scanf және printf функциялары секілді, тек бірінші аргументтері – файлға көрсеткіш.

Жолдарды оқып – жазуға мынадай функциялар қолданылады:

char * fgets (char * l, int maxl, FILE * fp);

int fputs (char * w, FILE * fp)

мұндағы **fgets** – fp файлынан келесі жолды оқып алып, w литерлер массивіне орналастырады, көшірілген жол "o" литерімен толықтырылады.

fputs – шығару функциясы жолды файлға жазады.

Мұндағы: w – жол тұрған массив адресі;

maxl – жолдағы символдардың максимум саны;

fp – файл көрсеткіші;

feof (FILE * fp) – функциясы файлдың соңын анықтайды.

Файлды жабу үшін:

Int fclose(FILE *fp) функциясы қолданылады, яғни файлдық көрсеткішпен орнатылған байланысты үзеді.

C++ тілінде:

Файлмен мәлімет алмасу мақсатында ifstream және ofstream кластарын қолданған тиімді. Ол үшін < **fstream. h** > тақырыптық файлы ашылуы тиіс.

Мұнда, сәйкесінше, **ifstream** класының объектісі **myof** және **ostream** класының **mygf** лектері құрылып көрсетілген файлмен байланысады.

Мысалы, **ifstream myof("prim.in",ios::in);**

Ofstream mygf("prim.out",ios::out);

• Бір объектімен тізбектеп бірнеше лекті байланыстыру қажет болса, онда ол төмендегідей ретпен орындалады:

ifstream myof;

...

myof("prim.in");

...

myof.close();

myof.open ("prim2.in");

...

myof.close());

- Файлмен мәлімет алмасу мүмкіндігінің режимін өзгерту – файлдық объектінің сипатындағы екінші аргументін өзгерту арқылы жүзеге асады.

Мысалы, `Ofstream myof ("prim.in", ios::app | ios::nocreate);` - мұндағы файл бар болса ғана `myof` объектісі құрылып, байланыс орнатылады, себебі **ios::nocreate** аргументі берілген, яғни файл болмаса, объект құрылмайтынын білдіреді. Ал, **ios::app** аргументі файлға шығарылатын мәндер файлдың соңына қосылатынын анықтайды.

Флаг-аргументтер биттік | - «немесе» амалының көмегімен біріктіріліп анықтала алады.

Файлдық лекті анықтау барысында қолданылатын аргументтерді анықтайтын флагтар төмендегідей:

флаг	Мағынасы
<code>ios::in</code>	Файл оқуға ашылады
<code>ios::out</code>	Файл жазуға ашылады
<code>ios::ate</code>	Объект құрылған соң, ағымды көрсеткіш файлдың соңына орнатылады
<code>ios::app</code>	Мәндер файлдың соңына жазылады
<code>ios::trunc</code>	Егер файл бар болса, ол тазартылады
<code>ios::nocreate</code>	Файл болмаса, онда объект құрылмайды
<code>ios::noreplace</code>	Файл бар болса, онда объект құрылмайды
<code>ios::binary</code>	Файл екілік режимде ашылады (қалыпты жағдайда текстік режим)

- Файлмен мәлімет алмасу үшін `fstream` класының объектісі де қолданылады: `Fstream io("update.dat", ios::in|ios::app);` - бұл жағдайда көрсетілген файл мәліметті оқуға және жазуға ашылады.

- **Seekg()** файлдан оқу ал, **seekp()** файлға жазу барысында функциялары файлдың ағымды позициясының маркерін басқаруға қолданылады.

Мысалы,

io.seekg(5, ios::cur) – оқу барысында маркерді ағымды орынынан 5 байтқа жылжыту;

io.seekg(-7, ios::end); - маркерді соңынан бастап 7 байтқа жылжыту керектігін білдіреді.

Аргументтер:

Io::cur – ағымды орынынан; **Io::beg** – **файл басынан**;

Io::end - файл соңынан жылжытылатынын білдіреді.

- *Файлдық лектің күйін анықтау функциялары:*

Функция	Түсініктеме
<code>eof()</code>	Файл соңын кездестірсе, 0-дік емес мән қайтарады
<code>fail()</code>	Лекте қате кездессе 0- ден өзге мән қайтарады
<code>bad()</code>	Енгізу-шығару барысында мағыналы қате кетсе, нөлден өзге мән қайтарады, онда лекпен жұмыс тоқтатылғаны дұрыс. Лек күйінің биттік мәндері анықталмаса, 0-ге тең емес мән береді.

rdstate()	Мына тұрақтылардың бірін, яғни лектің ағымды күйін анықтайтын мәнді қайтарады: Ios::goodbit - қате жоқ Ios::eofbit – файл соңына жетті Ios::failbit – форматтау немесе түрлендіру қатесі Ios::badbit - маңызды қате
clear()	0-ге тең болса тазарту

Файлдан оқу және файлға шығару әрекеттерін қолдану **мысалы**:

Есеп:

Білім деңгейін тексеруге арналған тестілеу программасын құрыңыз. Мұндағы тест материалдары текстік файлда мынадай ретпен берілсін:

- Алдыңғы жолда тест сұрағы;
- келесі бес жолда сол сұрақтың мүмкін болатын жауаптар нұсқасы;
- одан кейінгі жолда дұрыс жауабының нөмірі орналастырылады;
- келесі сұрақтар осындай ретпен файлға жазылады.

Тестілеу нәтижесін дұрыс жауаптар саны мен сәйкес алған бағасын жариялау арқылы хабарлау керек.

Есеп қойылымын нақтылау: Белгілеулер енгізейік:

F – файлдық айнымалы;

Kb – варианттар саны, жалпы жағдайда 5-ке тең;

Sur_san – Файлдағы сұрақтар біткенше оқылып анықталатын сұрақтар саны;

Dur_san – қолданушының жауаптарына қарай анықталатын дұрыс жауаптар саны;

Otb - қолданушы енгізетін ағымды жауаптарын меншіктейтін айнымалы;

Dg – файлдан оқылып анықталатын әр сұрақтың ағымды дұрыс жауабы;

Бага – дұрыс жауаптарының саны арқылы анықталатын баға;

P – дұрыс жауаптардың пайыздық мөлшерін меншіктейтін айнымалы;

Gol – сұрақтар мен варианттарын меншіктейтін ағымды жол айымалысы;

Fname – Тест материалдарынан тұратын файл аты енгізілетін айнымалы.

Математикалық моделі:

$$P = \text{dur_san} * 100 / \text{sur_san};$$

$$baga = \begin{cases} 5, & \text{егер } p > 91 \\ 4, & \text{егер } 76 \leq p \leq 91 \\ 3, & \text{егер } 56 \leq p \leq 90 \\ 2, & \text{қалған жағдайда} \end{cases}$$

Есептің программасы (C тілінде):

```
#include "stdio.h"
#include "conio.h"
#include "string.h"
#define kb 5
void main()
{char fname[40]; FILE* f;
int sur_san=0;
```

```

int dur_san=0;
int dg,otb,baga,p;
char gol[80];
int i;
puts("Файлдың атын енгізіңіз=?"); gets(fname);
if ((f=fopen(fname,"rt"))==NULL) {
printf("Файл аты дұрыс емес - %s",fname);
getch();
return;
}
clrscr();
puts("\nТестілеуді бастау үшін :");
printf("Enter басыңыз");
getch();
textbackground(RED);
clrscr();
    while ( !feof (f) ){
sur_san++;
fgets (gol,80,f);
printf("\n%i.%s\n",sur_san,gol);
for (i=1; i<=kb;i++){
fgets(gol,80,f);
printf("\n%i.%s",i,gol);
}
fscanf(f,"%i",&dg);
fgets(gol,80,f);
printf("\n Select=?"); scanf("%i",&otb);
if (otb==dg) dur_san++;
}
printf("Test dur san=%i\t",dur_san);
fclose(f);
p=100*dur_san/sur_san;
if (p>=91) baga=5;else
    if (p<=90 && p>=76) baga=4;
    else
        if (p<=75 && p>=56) baga=3;else baga=2;
printf("\a Baga=%i",baga);
getch();
}

```

Есептің программасы (C++ тілінде):

```

#include "iostream.h"
#include "conio.h"
#include "string.h"
#include<fstream.h>

```

```

#define kb 5
void main()
{ char fname[40];
int sur_san=0;
int dur_san=0;
int dg,otb;
char gol[40];
int i;
cout<<"fname=?";
cin>>fname;
ifstream fil(fname,ios::in);
if (!fil) {
cout<<"error %s"<<fname;
getch();
return;
}
clrscr();
textbackground(RED);
sur_san=0;
while(!fil.eof()){
sur_san++;
fil>>gol;
cout<<"\n"<<sur_san<<". "<<gol;
for (i=1; i<=kb;i++){
fil>>gol;
cout<<"\n"<<i<<")"<<gol;
}
fil>>dg;
cout<<"\n d="<<dg;
cout<<"\n Select=?"; cin>>otb;
if (otb==dg) dur_san++;
}
cout<<"Test dur san="<<dur_san;
fil.close();
}

```

Өзіндік жеке тапсырмалар:

1. Текстік файлдың әрбір жолындағы қысқа сөздерден тұратын файл құрыңыз.
2. Файлдағы сандар тізбегі геометриялық прогрессияны құрай ма?
3. Файлға кездейсоқ сандар тізбегін шығарыңыз. Файлдағы оң сандардың санын, ал теріс сандардың қосындасан анықтаңыз.
4. Файлда емхана пациенттерінің аты-жөні, диагонызы тіркелген. Қан қысымы жоғарыларының тізімін анықтаңыз.

5. Файлдағы сандар тізбегі арифметикалық тізбек прогрессия құрай ма?
6. Тексттік файлдың әрбір жолындағы символдар саны мен кері оқығанда өзі шығатын сөздерінен тұратын файл құрыңыз.
7. Төрт орында кездейсоқ n саннан тұратын файл құрыңыз. Олардың 3-кеселілерінің санын анықтаңыз.
8. Тексттік файлдың әрбір жолындағы цифрлардан тұратын сандар файлын құрыңыз.
9. Файлда бүтін сандар тізбегі берілген. Тақ сандар мен жұп сандарды екі файлға бөліңіз.
10. Студенттердің аты-жөні, туылған жылдары файлда сақталған. Бірдей жылдары туылған студенттерді анықтаңыз.
11. Студенттердің аты-жөні, туылған жылдары файлда сақталған. Атас студенттер бар ма?
12. Банк салымшыларының тізімі мен салым мөлшері файлда тіркелген. Бір айдан соң жалпы салынған ақша мөлшері мен әр салымшыға төленетін пайыздық(7%) ақша мөлшерін жаңа файлға сақтаңыз.
13. Файлда әр күні түскен өнім мөлшері берілген. Ең көп және ең аз өнім алынған күннің нөмірін анықта.
14. Файлда әр күні түскен өнім мөлшері анықталған. Барлық жиналған өнімді, орташа бір күндік мөлшерін және өнімнің неше күнде жиналғанын анықтаңыз.
15. Берілген файл студенттің аты-жөні, үш пәннен алған бағасы көрсетілген жолдардан тұрады. Әр студенттің фамилиясы мен орташа үлгерімі мәндерінен тұратын шығыс файлын құру программасын жазыңыз.
16. Топ студенттері туралы мәліметтерден тұратын файл құрыңыз. Топтағы ең жоғары және ең төмен үлгерімді студентті анықтау программасын құрыңыз.
17. Цех жұмысшыларының әр айдағы табыс мөлшерінен тұратын файл берілген. Барлық жұмысшының орташа табыс мөлшерін анықтаңыз.
18. Бір айдағы ауа қысымынан тұратын файл құрыңыз. Айдағы орташа қысым мөлшерін және ауа қысымы ең жоғары күнді анықтаңыз.
19. Кітапханадағы кітаптар туралы мәліметтер қорын құрыңыз. Мұндағы сатып алуға ең көп ақша жұмсалған кітаптың аты мен авторын, бір данасының құны мен жалпы санын экранға шығарыңыз.
20. Банктен несие алып, төлегендер мен уақытында төлегендер тізімі файлда аралас орналасқан. Төлемегендер тізімі мен жалпы қайтарылуы тиіс ақша мөлшерін анықтаңыз. Қарыз мөлшері теріс санмен анықталған.

Блиц-тест:

1. Файлдан оқу немесе файлға жазу үшін файлды ашу функциясын көрсетіңіз:
 - A) fopen
 - B) FILE
 - C) open
 - D) EOF
 - E) Fscanf

2. FILE структурасы қандай тақырыптық файлда сипатталған?

- A) Stdio.h;
- B) Iostream.h;
- C) Stdlib.h;
- D) Math.h;
- E) Strstrea.h.

3. Жаңа файлды әрі оқуға әрі жазға ашу режимі қайсысы?

- A) W+;
- B) W;
- C) R+;
- D) R;
- E) A+;

Бақылау сұрақтары:

1. Файлмен мәлімет алмасу мүмкіндіктері қандай?
2. Файлдық лектен оқу немесе шығару барысында қолданылатын функцияларды атаңыз, олардың қызметтерін меңгердіңіз бе?
3. C++ тіліндегі объект-лек құрылымы туралы не айтасыз?
4. C/C++ тіліндің файлдық енгізіп-шығару мүмкіндіктерін басқа тілдер мүмкіндіктерімен салыстырыңыз.

Глоссарий:

- **Файл-** жадыдағы аты аталған орын.
- **Файлмен мәлімет алмасу мүмкіндігі-** жадыдағы аты аталған орыннан мәліметтерді оқу немесе жазу реті.
- **Файлдық тип** - әдістемелік нұсқауды қараңыз
- **Файлдық лек-** әдістемелік нұсқауды қараңыз

Әдебиеттер:

[1],[4],[7],[8],[10-13],[15].

Глоссарий

СТАНДАРТТЫ КІТАПХАНАЛАР

C/C++ тілдерінің стандартты функциялары, типтері және макростары төмендегідей тақырыптық файлдарда орналасқан. Олар функционалдық қызметтеріне қарай топтастырылған:

<assert.h>, <float.h>, <math.h>, <ctype.h>, <limits.h>, <setjmp.h>, <errno.h>, <locale.h>, <signal.h>, <stdarg.h>, <stddef.h>, <stdio.h>, <stdlib.h>, <string.h>, <time.h>, <conio.h>, <graph.h>, <dos.h>.

<string.h> файлы

Кестедегі **s** және **t** айнымалылары **char *** типті, ал **cs** және **ct** **const char *** типті, **n** – **size_t** типті айнымалы, **c** айнымалысының мәні **int** - **char** типті мәніне келтірілген.

функция	Сипаттамасы
Char *strcat(s,ct)	S стрингінің соңына ct стрингін жалғайды, S стрингін қайтарады.
Char *strncat(s,ct,n)	S стрингінің соңына ct стрингінің n литерін жалғайды, S стрингін қайтарады және '\0' литерімен аяқтайды.
Char strcmp(cs,st)	s<t болса, теріс сан; s==t болса, 0; s>t болса оң сан;
Char strncmp(cs,ct,n)	strcmp секілді, бірақ салыстырылатын литерлер n-нен аспайды.
strcpy(s,t:string)	t-ны s-ке көшіреді.
strncpy(s,t:string,n:int)	t-нің n-нен көп емес литерін s-ке көшіреді.
strlen(s:string)	s-тің ұзындығын береді.
strchr(s,c:string)	s-тегі алғашқы кездескен c литеріне көрсеткішті береді, әйтпесе NULL.
strrchr(s,c:string)	s-тегі соңғы кездескен c литеріне көрсеткішті береді, әйтпесе NULL.

<ctype.h> файлы

<ctype.h> кітапханасындағы бірнеше функциялар литерлер тексеру жғне түрлендіру жұмысын атқарады. Бұл функциялардың қайтаратын мәні int типті.

функция	Сипаттамасы
isalpha(c)	c-әріп болса, 0-ге тең емес, кері жағдайда 0.
isupper(c)	c-жоғары регистрдегі әріп болса, 0-ге тең емес; кері жағдайда 0.
islower(c)	c-төменгі регистрдегі әріп болса, 0-ге тең емес; кері жағдайда 0.
isdigit(c)	c-цифр болса, 0-ге тең емес; кері жағдайда 0.
isalnum(c)	0-ге тең емес, егер isalpha(c) немесе isdigit(c) ақиқат болса; кері жағдайда 0.
isspace(c)	0-ге тең емес, егер c-пробел, табуляция, жаңа жол, парақаудару, тік-табуляция литерлері болса.
toupper(c)	жоғарғы регистрге аударылған c-ны қайтарады.
tolower(c)	төменгі регистрге аударылған c-ны қайтарады.

<stdlib.h> файлы

функция	Сипаттамасы
double atof(const char*s)	s-ті double-ге келтіреді
int atoi(const char *s)	s-ті int-ке келтіреді
int atol(const char *s)	s-ті long-ке келтіреді
int rand(void)	0-ден RANDMAX дейінгі диапазонындағы кездейсоқ сандар шығарады.
void *calloc(size_t nobj, size_t size)	Көрсеткішті size-өлшемді nobj объектер массивіне бөлінген жадыдағы орнына қайтарады
void *malloc(size_t size)	Көрсеткішті size өлшемді объектінің жадыдағы орнын қайтарады.
void *realloc(void	p сілтейтін объектінің өлшемін size-ға өзгертеді

*p,size_t size)	
void free(void *p)	p сілтейтін жады аймағын босатады.
void exit(int status)	Программаның дұрыс аяқталуын қамтамасыз етеді.

<stdio.h> файлы

функция	Сипаттамасы
FILE *fopen(const char *fname,...)	Берілген атты файлды ашады және ағынды қайтарады.
FILE freopen(const char *fname,...)	көрсетілген режимдегі файлды ашады және оны stream ағынымен байланыстырыды.
int fflush(FILE *stream)	Шығыс ағынына қолданылатын fflush буферде қалған берілгендерді толықтырып жазады.
int remove(const char *fname)	Көрсетілген атты файлды өшіреді.
int rename(const char *o,const char n)	Файл атын өзгертеді.
FILE *tmpfile(void)	Уақытша файл құрады.
int fprintf(FILE *s,const char *f,...)	Шығаруды түрлендіріп f-тің басқаруымен s ағынына жазады.
int printf(const char formate,...)	Толығымен fprintf функциясына эквивалентті.
int sprintf(char *s,const char f,...)	printf секілді, тек шығаруды '\0' литерімен аяқтап, s жолына жазады.
int fscanf(FILE *s,const char *f,...)	Берілгендерді f-тің басқаруымен s ағынына оқиды.
int scanf(const char *formate,...)	fscanf секілді.
int sscanf(char *s,const char *f,...)	scanf секілді айырмашылығы литерді енгізу s-мен жүргізіледі.
int fgetc(FILE *stream)	Келесі литерді stream ағынынан unsigned

	char түрінде қайтарады.
char *fgets(char *s,int n,FILE *stream)	s массивіне n-1 литерден аспайтын литер оқиды.
int fputc(int c,FILE *stream)	c литерін stream ағынына жазады.
int fputs(const char *s,FILE *stream)	s-стрингісін stream-ге жазады.
int getc(FILE stream)	fgetc секілді, айырмашылығы егер ол-макрос болса, stream 1-ден астам рет есептейді.
int getchar(void)	getc(stdin) секілді.
int ungetc(int c,FILE *stream)	c литерін stream-ге қайтарады.
char *gets(char *s)	Келесі енгізу жолын жана жол литерін '\0' литеріне ауыстырып s массивіне жазады.

<math.h> файлы

функция	Сипаттамасы
Double sin(double x)	синус x
Double cos(double x)	косинус x
Double tan(double x)	тангенс x
Double asin(double x)	арксинус x
Double acos(double x)	арккосинус x
Double atan(double x)	арктангенс x
Double atan2(double x,y)	арктангенс y/x
Double sinh(double x)	гиперболалық синус x
Double cosh(double x)	гиперболалық косинус x
Double tanh(double x)	гиперболалық тангенс x
Double exp(double x)	экспоненциалдық функция e

Double log(double x)	натурал логарифм
Double log10(double x)	ондық логарифм
Double pow(double x,y)	х-тің у дәрежесі
Double sqrt(double x)	түбір астындағы х
Double ceil(double x)	double үріндегі ең кіші бүтін
Double floor(double x)	double түріндегі ең үлкен бүтін
Double fabs(double x)	абсолютті мән х
Double modf(double x;double *ip)	х бүтін жғне бөлшек бөліктерге бөлінеді ;бүтін бөлігі *ip-ге,бөлшек бөлігі нәтиже ретінде беріледі.
Double fmod(double x,y)	х-ті у-ке бөлгендегі қалдық.

Стандартты кітапханада ungetc атты функция бар.

Int ungetc(int c,FILE *fp)

Бұл функция с литерін Fp файлына жібереді. Функция мәні ретінде EOF қайтарылады. ungetc функциясын енгізу функцияларының кез келгенімен бірге scanf, getc, getchar және т.б қолдануға болады.

Кездейсоқ сандар генераторы.

<stdlib.h> кітапханасында анықталған RAND_MAX тұрақтысы мүмкін болатын кездейсоқ сандар диапазонының ең үлкен мәні rand() RAND_MAX

#define frand() ((double) rand()/RAND_MAX+1.0))

Қазақстан Республикасы Білім және Ғылым министрлігі
Қ.Жұбанов атындағы Ақтөбе мемелететтік университеті

«050704-ЕТ және бағдарламалық қамтамасыз ету» мамандығы
күндізгі бөлімінің 1 курс студенттеріне арналған
«Программалау технологиялары» пәнін

ӨЗІНДІК ЖҰМЫСТАРДЫ ОРЫНДАУҒА АРНАЛҒАН
әдістемелік нұсқаулар

Ақтөбе, 2011

I. Программалау тілінің мүмкіндіктерін тереңірек зерттеуге арналған тапсырмалар

Мақсаты: C/C++ тілдерінің мүмкіндіктері туралы теориялық білімдерін тереңдету және әдебиеттермен жұмыс жасауға машықтандыру.

Өз бетімен орындауға арналған тапсырмалар:

1.1. Төмендегі тақырыптарды өз бетінше толықтырып конспектілеу:

- Си тілінің «жоғары деңгейлі» тілдерден айырмашылығы.
- Си және Си++ арасындағы салыстырмалы түрдегі айырмашылықтары.
- Квалификаторлар: const, volatile.
- Мәліметтер типін түрлендіру.
- Операторлардың орындалу приоритеттері.
- Айнымалылардың көріну аймақтарын кеңейту операторы.
- Енгізу-шығару мүмкіндіктері.
- Енгізу-шығару кластарының иерархиясы.
-

1.2. Тақырыптық файлдар мазмұнымен танысу және конспектілеу:

1. <assert.h>, <float.h>,
2. <math.h>, <ctype.h>,
3. <limits.h>, <setjmp.h>,
4. <errno.h>, <locale.h>,
5. <signal.h>, <stdarg.h>,
6. <stddef.h>, <stdio.h>,
7. <stdlib.h>, <string.h>,
8. <time.h>, <conio.h>,
9. <graph.h>, <dos.h>.

Өз бетімен орындауға арналған тапсырмаларға әдістемелік нұсқау:

1. Қажетті тақырыпқа қатысты дәрістегі материалдарды қайта оқып шығыңыз.
2. Аталған тақырыпқа қатысты берілген бағдар бойынша ойыңызды жинақтап, қарастыратын сұрақтарды топшалаңыз.

3. Сол сұрақтар бойынша әдебиеттерді қарастырып, қажетті материалдармен танысыңыз.
4. Толық түсінген соң негізгі элементтерін конспектілеп алыңыз.

II. Өз бетімен орындауға арналған мәнжазбалық тақырыптар

Мақсаты: Программалау мүмкіндіктерін зерттеуге машықтандыру және ғылыми зерттеу жұмыстарын жүргізуге бейімдеу.

Өз бетімен орындауға арналған тапсырмалар:

Мәнжазбалар тақырыптары:

1. Программалау тілдерінің жіктемесі.
2. Программалау технологиялары.
3. Машиналық бағытталған тілдер.
4. Жүйелік программалау барысындағы Си тілінің орыны.
5. Программалау негіздері.
6. C++ және объектілі-бағытталған программалау.

Өз бетімен орындауға арналған тапсырмаларға әдістемелік нұсқау:

1. Мәнжазба тақырыбына қатысты ұсынылған әдебиеттерді қарап материалдармен танысыңыз.
2. Нақты мағұлматтар алған соң тақырыпты толық қамтитын жоспар құрып, оқытушыға көрсетіп, талқылап алыңыз.
3. Қабылданған жоспар бойынша сұрақтар мазмұнын айқындайтын теориялық материалдарды топтастырып жазыңыз.
4. Әр түрлі әдебиеттерден материалдар алу барысында сәйкесінше сол әдебиетке сілтеме жасау қажет.
5. Мәнжазбаға кіріспе жазу барысында құрылған жоспардың мазмұнына түсініктеме беріңіз.
6. Қорытындылау барысында әдебиеттерді зерттеу барысында туындаған ойларыңызды, тиімді әдістеріңізді және ұсыныстарыңызды келтіріңіз.
7. Мәнжазбаны қорғауға арналған қысқаша баяндамаңызды дайындаңыз.

III. Программалауға арналған тапсырмалар

Мақсаты: C/C++ тілдерінің программалау ортасында жұмыс жасау икемділіктерін қалыптастыру және практикалық машықтарын бекіту. Сұрыптау алгоритмдерін меңгерту.

Өз бетімен орындауға арналған тапсырмалар:

1. Сұрыптау алгоритмдерін меңгеру және C/C++ тілдерінде программалау.

Сұрыптау әдістерінің жіктемесі:

- 1) Таңдау арқылы сұрыптау әдістері;
- 2) Енгізу арқылы сұрыптау әдістері;
- 3) Алмастыру арқылы сұрыптау әдістері;
- 4) Үлестіру арқылы сұрыптау әдістері;
- 5) Санау арқылы сұрыптау әдістері;
- 6) Біріктіру арқылы сұрыптау әдістері;
- 7) Құрылымдық мәліметтерді сұрыптау мысалдары;
- 8) Сыртқы сұрыптаулар.

Ескерту. Аталған сұрыптау тақырыптарының әрқайсысында қарастырылатын әдістерді анықтап, теориялық материалдарын конспектілеңіз және төмендегі әдістемелік нұсқа бойынша сұрыптау алгоритмдерін программалап жұмысты орындаңыз.

Әдебиеттер:[18]

2. Жеке тапсырмалар кестесі (Әр студентке жеке нұсқалар беріледі):

Тақырыптарына сәйкес есептер нөмірлері			
Варианттар	1. Нақты сандарға амалдар қолдану. 2. Формулаларды есептеу. 3. Тармақталу. 4. Қарапайым бүтін сандар арифметикасы. 5. Қарапайым цикл.	1. Мәліметтерді біртіндеп енгізу және шығару. 2. Цикл және тармақталуға аралас есептер. 3. Символдар тізбегін өңдеу. 4. Матрицалар, циклдар (іштестірілген)	Функция
1.	1, 21, 41, 80, 100, 101	136(а), 179(а), 251, 367	424, 453
2.	2, 22, 42, 79, 99, 102	136(б), 179(б), 252(а), 368	425, 454
3.	3, 23, 43, 78, 98, 103	136(в), 179(в), 252(а), 369	426, 455
4.	4, 24, 44, 77, 97, 104	136(г), 180, 253(а), 370(а)	427, 456
5.	5, 25, 45, 76, 96, 105	136(д), 181(а), 253(б), 370(б)	428, 457
6.	6, 26, 46, 75, 95, 106	136(е), 181(б), 253(в), 371	429, 458
7.	7, 27, 47, 74, 94, 107	136(ж), 181(в), 253(г), 372	430, 459
8.	8, 28, 48, 73, 93, 108	136(з), 182, 254, 373(а)	431, 460(а)
9.	9, 29, 49, 72, 92, 109	136(а), 183, 255, 373(б)	432, 461
10.	10, 30, 50, 71, 91, 110	136(к), 184, 256(а), 374(а)	433, 462
11.	11, 31, 51, 70, 91, 111	136(м), 185, 256(б), 374(б)	434, 463
12.	12, 32, 52, 69, 89, 112	136(н), 186, 257(а), 374(в)	435, 460(б)
13.	13, 33, 53, 68, 88, 113	136(л), 187, 257(б), 375	436, 464
14.	14, 34, 54, 67, 87, 114	136(о), 188, 257(в), 376(а)	437, 465
15.	15, 35, 55, 66, 86, 115	137(а), 189, 257(г), 376(б)	438, 466
16.	16, 36, 56, 65, 85, 116	137(б), 190, 257(д), 377	439, 467
17.	17, 37, 57, 64, 84, 117	137(в), 191, 257(е), 378(а)	440(а,б), 468
18.	18, 38, 58, 63, 83, 118	137(г), 192, 2558, 378(б)	440(в), 469
19.	19, 39, 59, 62, 82, 119(а, в, д)	137(д), 193, 259, 379(а)	441, 470
20.	20, 40, 60, 61, 81, 119(б, г, е)	137(е), 194, 260(а), 379(б)	445, 456
21.	9, 29, 49, 71, 91, 110	138, 195, 260(б), 379(в)	446, 460(а)
22.	6, 26, 46, 76, 96, 105	139(д), 196, 260(в), 379(г)	447, 460(б)
23.	4, 24, 44, 80, 100, 101	139(ж), 198, 261(а), 380	448(а), 459
24.	10, 30, 50, 77, 97, 104	139(и), 199, 261(б), 381	448(б), 470

Әдебиет: С. А. Абрамов и другие Задачи по программированию, М: Наука гл. ред. физ. мат. лит. 1989-224с.

Өз бетімен орындауға арналған тапсырмаларға әдістемелік нұсқау:

1. С. А. Абрамов батаған авторлар тобы құрастырған «Задачи по программированию» кітабына өзіңізге берілген нұсқа есептерін алыңыз.
2. Қарастырылып отырған тақырыпқа сәйкес есептің қойылымын анықтаңыз, яғни не берілгендігін және нені табу керектігін белгілеулер енгізе отырып анықтаңыз.
3. Есептің математикалық моделін құрыңыз.
4. Есептің алгоритмін құрыңыз. Дәлірек айтқанда есептің блок-схемасын жазу керек.
5. Құрылған алгоритмге сәйкес есептің Си және С++ тілдеріндегі программаларын құрыңыз.
6. Құрылған программаны енгізіп, өңдеңіз және есептеулер жүргізіңіз.
7. Алынған нәтижелерді талдаңыз.
8. Жоғарыдағы кезеңдер нәтижелерін арнайы дәптерге түсіріп, семестрлік жұмысты тапсыру есебін(отчет) толтырыңыз.

СТУДЕНТТІҢ БІЛІМ ДЕҢГЕЙІН БАҚЫЛАУҒА АРНАЛҒАН МАТЕРИАЛДАР

Бақылау жұмысы №1

Бақылау жұмысы тармақталу және циклдық алгоритмдерді және массивтерді өңдеу тақырыбын меңгергендігін тексеруге арналған.

Нұсқа I

1. Үш санның ең кішісі мен ең үлкенінің қосындысын анықтаңыз.
2. 1 мен 20-ның аралығындағы тақ сандардың арифметикалық ортасын есептеңіз.
3. Студенттердің 8-ші семестрдегі 5 түрлі пән бойынша алған бағалары берілген. Студенттердің үлгерімін анықтайтын программа жазыңыз.

Нұсқа II

1. Телефон желісінің қызметі келесі түрде төленеді: A минутке дейін сөйлескені үшін айына – B тенге, одан артық болса – C тенге төленеді. Бір айда сөйлескен уақыт үшін қанша тенге төленетіндігін есептейтін программа құрыңыз.
2. $\frac{(X-1)(X-3)(X-7)\dots(X-63)}{(X-2)(X-4)(X-8)\dots(X-64)}$ мәнін есептеу программасын жазыңыз.
3. Бастауыш сыныптың оқушылары сыныптары бойынша бой – бойымен сапқа тұрғызылған. Әрбір сынып бойынша оқушылардың орташа бой көрсеткішін анықтайтын программа жазыңыздар.

Бақылау жұмысы №2

Бақылау жұмысы жолдық мәліметтерді өңдеу және ішкі программаларды құру тақырыбын меңгергендігін тексеруге арналған.

Нұсқа I

Жол берілген. Жолдағы тыныс белгілер санын есептеу.

«Қазақтелеком» телефон станциясының абоненттері туралы мәліметтер белгілі: абоненттің аты-жөні, мекен-жайы, телефонды қондырған жылы, телефон номері. Соңғы 5 жыл ішінде телефон қондырған абоненттердің санын анықтап, экранға аты-жөнін және мекен-жайын шығару.

Нұсқа II

1. Жол берілген. Осы жолдағы «a» әріптерін «b» әріптеріне ауыстыратын программа жазу.
2. Әуежай кассасында төмендегідей мәліметтер белгілі: рейстің номері, баратын жері, ұшу уақыты, ұшатын күндері (күн сайын, жұп күндері, тақ күндері). Клавиатурадан енгізілген күні Лондон қаласына ұшатын рейстің номері мен ұшу уақытын анықтайтын программа құру.

Емтихан сұрақтары:

№	Сұрақтар мәтіні
1.	Мәліметті бейнелеу моделдері мен алгоритмдік концепциялары негіздері.
2.	Программалау тілдері жіктелімі.
3.	Программалау технологиялары.
4.	Алгоритмдер құру және оны талдау түсініктері.
5.	Программалау стилдері.
6.	Си программалау тілі туралы жалпы мағлұматтар. Өзге тілдерден ерекшелігі.
7.	Си тіліндегі программа құрылымы.
8.	Стандартты енгізу-шығару мүмкіндіктері.
9.	Форматтап шығару функциясы. Мысалдар.
10.	Форматтап енгізу функциясы. Мысалдар.
11.	Форматтау тегтері.
12.	Жолдарды енгізіп-шығару функциялары.
13.	Айнымалылар аттары. Мәліметтер типі мен өлшемі.
14.	Тұрақтылар. Декларациялар.
15.	Арифметикалық операторлар.
16.	Логикалық және реттік қатынас операторлары.
17.	Типтерді түрлендіру.
18.	Инкременттік және декременттік операторлар.
19.	Биттік операторлар.
20.	Меншіктеу және өрнек операторлары.
21.	Шартты өрнектер.
22.	Есептеулер реті мен приоритеттері.
23.	Инструкциялар мен блоктар түсінігі.
24.	Шартты конструкциялар.
25.	Ажыратқыш (таңдау) инструкциясы.
26.	Циклдар инструкциялары.
27.	Циклдың жұмысын үзу инструкциялары мен басқаруды шартсыз көшіру.
28.	Функциялар туралы негізгі мәліметтер. Функциялардың қайтаратын мәндері.
29.	Сыртқы және ішкі айнымалылары.
30.	Файлдармен жұмыс жасау ерекшеліктері, режимдері
31.	Файлдармен жұмыс жасау функциялары.
32.	Тексттік файлдармен жұмыс жасау мүмкіндіктері.
33.	Екілік файлдармен жұмыс жасау мүмкіндіктері.
34.	Рекурсия. Түсінігі мен түрлері.
35.	Рекурсивті алгоритмдер.
36.	Массивтер. Типінің сипаттамасы. Инициализациялау тәсілдері.
37.	Бір өлшемді массивтерді өңдеу мүмкіндіктері.

38.	Екі өлшемді массивтерді өңдеу мүмкіндіктері.
39.	Массивтерді сұрыптау алгоритмдері.
40.	Массивтерден іздеу алгоритмдері.
41.	“string.h” тақырыптық файлының мазмұны.
42.	Массивтерді енгізу-шығару мүмкіндіктері.
43.	Литерлер массиві.
44.	Көрсеткіштер.
45.	Адресер. Адрестік арифметика.
46.	Көрсеткіштер мен массивтер.
47.	Көрсеткіштер массиві.
48.	Функцияларға көрсеткіштер.
49.	көрсеткіштер мен функция аргументтері.
50.	Структура туралы жалпы түсініктер.
51.	Структуралар массиві.
52.	Структуралар мен функциялар.
53.	Структураларға көрсеткіштер .
54.	typedef құралы.
55.	Бірігу. Синтаксисі, бірігу элементтерімен мән алмасу мүмкіндіктері.
56.	Символдарды түрлендіруге және тексеруге арналған <ctype.h> файлындағы функциялар.
57.	Си тілде қолданылатын математикалық функциялар.
58.	<stdlib.h> бас файлындағы жалпы жағдайларға арналған функциялар.
59.	ОБП негіздері, принциптері.
60.	Программа сапасын бағалау критерийлері.
61.	Программаны жобалау тәсілдері.
62.	Программаны жобалау кезеңдері.
63.	Компоненттік технология.

Тест тапсырмалары:

Функцияның қайтаратын мәнінің жоқтығын қандай сөз білдіреді:

Void.

Main.

Function.

Procedure.

Sql.

Горизонталды табуляцияны білдіретін эскейп-тізбек:

\t.

\n.

\r.

\a.

\v.

Бүтін ондық санды (типі int) білдіретін формат спецификаторы:

%d.

%c.

%f.

%e.

%s.

C++ тегі x айнымалысына мән енгізу операторы:

Cin>>x.

Scanf("%f", &x).

Cout<<x.

Printf("\nx=", x).

Cin<<x.

Санды белгілі бір дәрежеге келтіру функциясы:

Pow().

Cos().

Sqrt().

Log().

Abs().

Айнымалының адресін операторы:

&

%

\

*

=

Жаңа жолға көшіруді басқаратын символ:

\n.
\t.
\v.
\a.
\r.

Рекурсияның түрлері:

Тіке, жанама.

Тізбекті, тізбексіз.

Статикалық, динамикалық.

Терең, тайыз.

Шексіз, шектеулі.

Сегіздік кодты бейнелейтін эскейп тізбегі

\ooo

\b

\xhh

\i

|ooi

c-ны жоғарғы регистрге келтіретін функция

toupper(c)

isalnum(c)

tolower(c)

case(c)

Uppcase (c)

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

```
main ( ) {
```

```
long nc;
```

```
nc =0;
```

```
while ( getchar ( ) != EOF ) ++ nc;
```

```
printf ( “ %d \n “ , nc ) ;
```

```
} программасы нені анықтайды ?
```

литерлер санын есептейді

литерлер кодын шығарады

литерлерді экранға шығарады

файлдан оқиды

литерлер кодын қосады

Функция мәнін қайтару инструкциясын таңдаңыз:

Return

End

Printf

Define

Typedef

Мәліметтерді шығару барысында дыбыс шығарту эскейп -тізбегі:

\a
\b
\f
\n
\?

Файлдан оқу немесе файлға жазу үшін файлды ашу функциясын көрсетіңіз:

Fopen
FILE
Open
EOF
Fscanf

Қолданушы файлмен байланыс орнату үшін қандай декларацияны анықтауы тиіс:

FILE *fp;
file fp;
FILE open (fn);
fopen file
file *open

FILE құрылымы (структурасы) қандай тақырыпты файлда сипатталған?

<stdio.h>
<stdlib.h>
<file.h>
<tostream.h>

Сипатталмаған

Файлдық объектінің конструкторындағы екінші аргумент құрамындағы флаг қандай болғанда файл екілік режимде ашылады?

ios :: binary
ios :: noreplace
ios :: nocreate
ios :: app
ios :: in

A[n] массиві элементтерінің индексі қай аралықта өзгере алады?

0 – ден бастап n – 1 ге дейін;

1 – ден бастап n – 1 - ге дейін;

n – нен бастап 1 –ге дейін;

0 – ден бастап 255 арасында.

1 – ден бастап 255 арасында.

Төменде бір өлшемді массивтің максимал элементін табу программасының фрагменті келтірілген. Төмендегі нұсқалардың ішінен дұрысын көрсетіңіз?

```
int a[5], max, I;
{
    <Массив элементтерін енгізу>
    max[0];
    for (i=1; i>n; i++)
        if (a[i]>max) max=a[i];
    ...
}
int a[5], max, I
{
    <Массив элементтерін енгізу>
    max[0];
    for (i=1; i>n; i++)
        if (a[i]<max) max:=a[i];
    ...
}
float a[5], max, I;
{
    <Массив элементтерін енгізу>
    max[0];
    for (i=1; i>n; i++)
        if (a[i]>max) max=a[i];
    ...
}
int a[5], max, I;
{
    max[0];
    for (i=1; i>n; i++)
        if (a[i]>max) max=a[i];
    <Массив элементтерін енгізу>}
int a[5], max, I;
{
    max[0];
    for (i=2; i>n; i++)
        if (a[i]<max) max=a[i];
    <Массив элементтерін енгізу>
}
```

Графикамен жұмыс жасағанда қате кодын экранға шығару үшін қандай функция пайдаланылады:

```
grapherrormsg (қате коды);  
detectgraph(қате коды);  
graphresult (қате коды);  
error (қате коды);  
closegraph(қате коды);
```

Тексттік режимнен графиктік режимге қайта ауысу функциясы:

```
setgraphmode( );  
restorecrtmode( );  
detectgraph( );  
initgraph( );  
grapherrormsg ();
```

settextstyle () функциясы қандай қызмет атқарады?

Шрифт түрін, бағытын, өлшемін тағайындайды;
Шрифт түсін тағайындайды;
Шрифт өлшемін тағайындайды;
Шрифт сызылымын тағайындайды;
Шрифт бағытын тағайындайды;

cleardevice () функциясы қандай қызмет атқарады?

Экранды тазалайды;
Графикалық режимді инициализациялайды;
Графикалық режиммен жұмысты аяқтайды;
Текстті экранға шығарады;
Текстті өшіреді;

moveto (x, y) функциясы қандай қызмет атқарады?

Позиция көрсеткішін x,y жаңа координаттарға орын ауыстырады;
Ағымдағы позиция көрсеткішіне x,y өсімшелерін қосады ;
x,y координатты нүктені салады;
Басы x,y координаттарында орналасқан түзу кесінді тұрғызады;
x,y позициясындағы символд өшіреді;

x,y позициясынан бастап sp жолын экранға шығару функциясы:

```
outtextxy(x,y,sp);  
outtextxy(sp);  
text(x,y, sp);  
outtext(x,y);  
outtext(sp);
```

Тұйық облысты бояу функциясы:

```
floodfill (x,y, шекара );  
floodfill (x,y);  
fillpoly(x,y, шекара );  
fillellips(x,y, rx, ry);  
settextfill (x,y, шекара);
```

Графигтік режимнен тексттік режимге уақытша ауысу функциясы:

```
restorecrtmode( );  
setgraphmode( );  
detectgraph( );  
initgraph( );  
grapherrormsg ( );
```

Ешқандай мән қайтармайтын функцияны көрсетіңіз:

```
void kos(int x, int y);  
float kos(float y);  
bool kos(int x, int y);  
float kos(char x);  
double kos( float *p);
```

Растрлы бейненің ең кіші элементі не болып табылады?

Пиксель;
Түзу;
Кесінді;
Квадрат;
Шеңбер;

Seekp () функциясының қызметі қандай?

Файлдан оқу барысындағы маркердің (файлдың ағымды) орнын анықтау
Файлға жазу барысындағы маркердің (файлдың ағымды) орнын анықтау
Маркердің позициясын қайтарады
кез келген файлдың маркерін басына қою
кез келген файлдың маркерін соңына қою

Seekg () және seekp () функцияларының екінші аргументтері ретінде алынған қай флаг маркерді файлдың басынан бастап көрсетілген байтқа жылжыту әрекетін орындайды?

ios:: beg
ios:: cur
ios:: end
ios:: app
Жылжытпайды

C/C++ тілінде бір өлшемді массивті қалай сипаттайды? Төменде көрсетілген нұсқаларының ішінен дұрысын көрсетіңіз:

<типi> <массив аты> [өлшемі];

var <массив аты>:array[1..n] of <типi>, мұндағы n – массив элементтерінің саны;

<массив аты> [өлшемі] <типi>;

void <массив аты> [өлшемі] <типi>.

Массив қолданылмайды

Массивті сипаттау кезінде қолданылатын тұрақтыны қандай директивада сипаттауға болады?

define;

conio.h;

stdio.h;

math.h.

math

Жолдың ұзындығын табу функциясын көрсетіңіз:

str len

Strstr

Strcmp

Strcpy

Length

Strstr (S1,S2) Функциясы ...

S1 жолына S2 жолының енуін анықтайды

S1 жолының соңына S2 жолын тіркеп жазады

S1, S2 жолдарын салыстырады

S1 жолының әріптерін жоғарғы регистрге ауыстырып S2 жолына жазады

S1 жолының әріптерін төменгі регистрге ауыстырып S2 жолына жазады

Си тілінде мәндер типіне жаңа атау беруге мүмкіндік беретін құрал:

Typedef

Tupe

Union

Struct

Элементтерінің бірі көрсеткіш болатын структураны атайды.

C++ тілінде файлмен жұмыс жасау барысында қолданылатын классты көрсетіңіз:

Ifstream

Stream

Iostream

Print

Input

Файлға мәліметті форматтап шығару үшін қандай функция қолданылады?

fprintf,
fscanf,
fopen,
fwritef,
freadf,

for do цикл операторында айнымалы параметр қандай типті болуы керек?

бүтін типті;
нақты типті;
жолдық типті;
логикалық типті.
Массив типті

C/C++ тілінде циклдың қандай түрлері бар? Төменде көрсетілген
нұсқалардың ішінен дұрыс емесін көрсетіңіз.

repeat until;
for ();
while ;
do while;
Do while ; while;

c литерін түменгі регистрге келтіретін функция

tolower(int c)
free(void *c)
toupper(int c)
case(c)
Uppcase(c)

if (a>b) z=a; else z=b инструкциясын басқаша қалай жазуға болады?

z=(a>b)?a:b
a>b:z=a:z=b
z=a>a>b:z=b
if (a>b) z=a else z=b
(a>b)?a:b

Түзету үшін тексттік файлды ашу?

"r+"
"w"
"r"
"a"
"b"

C/C++ тілінде екі өлшемді массивті сипаттаудың төменде көрсетілген нұсқаларының ішінен дұрысын көрсетіңіз

<типi> <массив аты> [жолының өлшемі] [бағанының өлшемі];

var <массив аты>:array[1..n,1..m] of <типi>, мұндағы n – массив жолдарының саны, m – массив бағандарының саны;

<массив аты> [өлшемі] <типi>;

void <массив аты> [өлшемі] <типi>.

<массив аты> [өлшемі] <типi>.

C тілінде екі өлшемді массивті шығарудың төмендегі нұсқаларының қайсысы дұрыс?

```
for (i=0; i< <массив өлшемі>; i++) {  
    for (j=0; j< <массив өлшемі>; j++)  
        printf('%<массив типінің форматы>', <массив аты> [i] [j]); }  
for (i=0; i<= <массив өлшемі>; i++) {  
    for (j=0; j<= <массив өлшемі>; j++)  
        printf('&<массив типінің форматы>', %<массив аты> [i] [j]); }  
for (i=0; i<= <массив өлшемі>; i++)  
    for (j=0; j<= <массив өлшемі>; j++)  
        scanf('%<типінің форматы>', &<массив аты> [i] [j]);  
for (i=0; i< <массив өлшемі>; i++)  
    for (j=0; j< <массив өлшемі>; j++)  
        printf('%<типінің форматы> &<массив аты> [i] [j]");  
for (i=0; i< <массив өлшемі>; i++)  
    for (j=0; j< <массив өлшемі>; j++)  
        printf('%<типінің форматы> &<массив аты> [i] [j]");
```

Аргументтері бүтін тип ретінде сипатталған функцияны көрсетіңіз:

bool kos(int x, int y);

float kos(float y);

void kos(int x, int y);

float kos(char x);

double kos(float *p);

Мәндер енгізу барысында буферді тазалау үшін қандай функция қолданылады?

flushall ()

Flush

Dispose

Clear

clear bus

Атаусыз структура деп ...

Анықтау барысында тегі көрсетілмеген структураны айтады.

Бірдей типті айнымалылардан тұратын структураны атайды.

Элементтерінің бірі көрсеткіш болатын структураны атайды.
Элементтерінің бірі структура болатын структураны атайды.
Дұрыс жауап жоқ.

Иштестірілген структура дегеніміз не?

Элементтерінің бірі структура болатын структура
Элементтерінің бірі көрсеткіш болатын структура
Элементтерінің бірі символдық массив болатын структура
Тегі көрсетілмеген структура

Дұрыс жауап жоқ

Төмендегі программа нені есептейді?

```
#include <stdio.h>
main()
{
long nc;
nc=0;
while(getchar()!=EOF)
++nc;
printf("%ld\n",nc);
}
```

Литерлер санын есептейді;
Нөлдік литерлер санын есептейді;
Ештеңе есептемейді;
'\0' литерінің санын есептейді;
Файлдың соңын анықтайды.

Төмендегі фрагмент нені анықтайды?

```
If (c>='0' && c<='9') ++nc[c-'0'];
```

Литер цифр болса, онда әр цифрдың санын анықтайды;
Литер цифр болса, онда әр цифрдың кодын анықтайды;
Литер цифр болсамаса, онда әр цифрдың санын анықтайды;
Әр символдың санын анықтайды;
Литерлер санын анықтайды;

Төмендегі функция нені анықтайды?

```
...
int step(int b,int n){
int p;
for(p=1; n>0;--n)
p=p*b;
return p;
}
```

В – ның N – дәрежесін анықтайды;
Өртүрлі N санның көбейтіндісін есептейді;
В-ны N рет қосады;
В – ны N – ге көбейтеді;
Ештеңе.

Спецификатордың алдындағы қандай таңба санды барлық уақытта таңбасымен шығару қажеттігін білдіреді?

+

-

#

%

&

Fmod(x,y) функциясы нені анықтайды?

X –ті Y –ке бөлгендегі қалдықты бөлшек сан түрінде анықтап қайтарады;

X –ті Y –ке бөлгендегі қалдықты бүтін сан түрінде анықтап қайтарады;

X –ті Y –ке бөлгендегі бүтін бөлігін қайтарады;

X –ті Y –ке бөлгендегі қалдықты қиып алып қайтарады;

Ештеңе.

Төмендегі фрагмент нені анықтайды?

Printf(“\n[%d] %-15s”,++I,gol);

Жолдың кем дегенде 15 символын шығарады және сол жағынан пробелмен толтырып тегістейді;

Жолдың кем дегенде 15 символын шығарады және оң жағынан пробелмен толтырып тегістейді;

Жолдың кем дегенде 15 символын шығарады ;

Жолдың кем дегенде 15 символын шығарады және ортасынан пробелмен толтырып тегістейді;

Жолдың 15 символын ғана шығарады және сол жағынан пробелмен толтырып тегістейді

Төмендегі фрагмент нені анықтайды?

Cout<<oct<<k;

K мәнін сегіздік санау жүйесінде шығарады;

K мәнін екілік санау жүйесінде шығарады;

K мәнін оналтылық санау жүйесінде шығарады;

K мәнін ондық санау жүйесінде шығарады;

K мәнін шығарады.

Функция қандай типті мән қайтара алмайды?

Массив типті;

Бүтін типті;

Нақты типті;

Символдық;
Булевті;

Графикалық режим дұрыс инициализацияланған жолды көрсетіңіз:

```
initgraph(&gd, &gm, "BGI файлының жолы");  
detectgraph(&gd, &gm, "BGI файлының жолы");  
initgraph(gd, &gm, "BGI файлының жолы");  
detectgraph(gd, &gm, "BGI файлының жолы");  
initgraph(&gd, gm, "BGI файлының жолы");
```

```
void salistr (int a, int b)  
{  
    int c;  
    if (a>b) c=a;  
    else c=b;  
    printf('c=%i', c);  
}
```

функциясының орындалуы нәтижесінде не қайтарылады?

Ешқандай мән қайтарылмайды

Бүтін типті мән

Нақты типті мән

Символдық типті мән

Булевті типті мән

Жаңа файлды әрі оқуға әрі жазға ашу режимі қайсысы?

W+;

W;

R+;

R;

A+;

FILE структурасы қандай тақырыптық файлда сипатталған?

Stdio.h;

Iostream.h;

Stdlib.h;

Math.h;

Strstrea.h.

Int *p;

Int a;

...

p=&a;

p++;

p--;

...

фрагментінің орындалу нәтижесінде р қайда сілтейді?

А мәні орналасқан орынға;

Ешқайда;

А орналасқан орыннан кейінгі ұяшыққа;

А орналасқан орынға дейінгі ұяшыққа;

Р ұяшығына.

Текст шығарылатын орынның фонының түсін анықтау функциясын көрсетіңіз:

Textbackground;

Textcolor;

Color;

Window;

Text;

X^Y мәнін есептеу функциясы қайсы?

Pow;

Sqrt;

Exp;

Expl;

Rand;

Fabs(double x) функциясы нені есептейді?

X аргументінің нақты абсолюттік мәнін есептейді;

X аргументінің бүтін абсолюттік мәнін есептейді;

X аргументінің мәнін есептейді;

X аргументінің дәрежелік мәнін есептейді;

X аргументінің мәнін математикалық дөңгелектеу орындалады.

Тандау операторының барлықтармағына қойылуы мүмкін операторды көрсетіңіз:

Break;

Exit;

Close;

Switch;

Ондай мүмкін емес.

Графиктік режимде курсор көрсеткішін берілген координаталы нүктеге көшіру функциясын көрсетіңіз:

Moveto;

Moverel;

Linerel;

Outtext;

Out;

Setfillstyle функциясының қызметі қандай?

Берілген аймақты бояу стилі мен түсін орнатады.

Сызықтың түсі мен стилін анықтайды.

Тіктөртбұрыштың шекараларының түсі мен стилін анықтайды;

Тексттің түсін анықтайды;

Түзудің стилін анықтайды.

Белгілі бер секундтар мөлшерінде үзіліс жасау әрекетін орныдайтын функцияны таңдаңыз:

Delay;

Pause;

Sound;

Nosound;

Next.

Параллелепипед сызу функциясы:

Bar3d;

Bar;

Arc;

Rectengle;

Paral;

Таңдау операторында мәні анықталмаған жағдайлар үшін анықталған тармақ қандай қызметші сөзден басталады?

Default;

Case;

Switch;

Else;

Then;

Функция ешқандай мән қайтармаса қандай типпен сипатталады?

Void;

Int;

Double;

Char;

Bool;

++*Р өрнегі орындалу нәтижесі қандай?

Р сілтейтін мән бірге арттырылады;

Р көрсеткішінің мәні арттырылады;

Көрсеткіштің де оның сілтейтін мәні де бірге артады;

Р көрсеткіші бір байтқа жылиды;

Р көрсеткіші жойылады;

```
Int x=1, y=2.z[10];
```

```
Int *p;
```

```
Ip=&x;
```

```
Y=*p;
```

Фрагменті орындалу нәтижесінде у нешеге тең?

1

2

0

-1

-2

++*P өрнегіне эквивалентті өрнекті көрсетіңіз:

*p+=1

*(p++)

*p++

*p+1

*(p+)

Жұмысшы алаңдағы программаны қадамдап орындауға арналған перде қандай?

F8

F7

F9

CTRL

ALT+F5

С программалау ортасында текст блогын өшіріп тастау үшін қандай пернелер комбинациясы қолданылады?:

Ctrl+del.

Ctrl+shift.

Ctrl+ins.

Ctrl+end.

Shift+del.

Программалау тілдеріндегі типтер жүйесі туралы қай тұжырым дұрыс емес?

Типі анықталған айнымалы программаның орындалу барысында мәнін өзгертпейді.

Типі мәнінің жадыда бейнелену форматын анықтайды.

Типін анықтау арқылы айнымалыға бастапқыда жадыдан қанша орын бөлінетіндігі анықталады.

Айнымалыға меншіктеу барысында сәйкес емес мәнді меншіктету қателігін бақылау барысында типі қолданылады.

Тип арқылы мәніне қолданылатын амалдар тобы анықталады.

ҰСЫНЫЛАТЫН ӘДЕБИЕТТЕР:

1. Б. Керниган, Д. Ритчи Язык программирования Си: Пер. С англ. – М.: Финансы и статистика, 1992.-272с.
2. Крис Паппас, Уильям Мюррей Программирование на С и С++: -К.: Изд. Группа BHV,2000. -320 с.
3. Вирт Н. Алгоритмы и структура данных: Пер. с англ.-М.: Мир, 1989.
4. А.Г.Юркин. Задачник по программированию. Питер, 2002 г.
5. Н.Культин., С/С++ в задачах и примерах – БХБ-Петербург, 2002 г.
6. Уэйт М . Язык Си. Руководство для начинающих: Пер. с англ. – М., Мир, 1988 г.
7. Ален И. Голуб. Правила программирования на С и С++: М., БИНОМ, 1996.
8. Белецкий Я. Энциклопедия языка Си. М.: Мир, 1992.
9. Б.И.Березин, С.Б.Березин Начальный курс С и С++, Москва, 999 г.
10. Уолтер Савич С++ во всей полноте. Москва,Киев,2005.
11. Дэвис, Стефан, Р. С++ для «чайников», Москва, Сант-Петербург, Киев 2001.- 336с.
12. Гудман С. и др. Введение в разработку и анализ алгоритмов. – М.: Мир, 1981.
13. Марков А.С. и др. Программное обеспечение ЭВМ. – М.: Выс. Шк., 1990.
14. Болски. Язык программирования Си. Справочник пер. с английского, М., Радио и связь, 1988.
15. Бондарев В.М. Основы программирования. Харьков: ФОЛИО, Ростов-на-Дону: ФЕНИКС.
16. И.Ю. Каширин, В.С. Новичков От Си к Си++, Москва:Горячая линия – Телеком, 2005. – 334 с.
17. Уэйт М., Прата С., Мартин Д. Язык Си. Руководство для начинающих: Пер с англ.-М.:Мир,1988.
18. Климова Л.М. Pascal 7.0 Практическое программирование. Решение типовых задач.-М.: КУДИЦА - ОБРАЗ, 2000. - 496 с.

Мазмұны

Пәннің типтік бағдарламасы	3
Оқу пәнінің жұмыс бағдарламасы	6
Силлабус	21
Пән бойынша тапсырмаларды орындау және тапсыру кестесі	29
Пәннің оқу-әдістемелікпен қамтамасыз ету картасы	30
Теориялық материалдарға шолу	32
Практикалық сабақтар жоспары	99
Пәнді оқуға арналған әдістемелік нұсқаулар	100
Практикалық сабақтарға әдістемелік нұсқау	110
Лабораториялық сабақтарға әдістемелік нұсқау	154
Глоссарий	240
Өзіндік жұмыстарды орындауға арналған әдістемелік нұсқау	245
Студенттің білім деңгейін бақылауға арналған материалдар	251
Емтихан сұрақтары	253
Тест тапсырмалар тізімі	255
Мазмұны	271

КЕЛІСІМ БЕТІ

ПӘННІҢ ОҚУ-ӘДІСТЕМЕЛІК КЕШЕНІ

Мамандығы 5B070400-ЕТ және бағдарламамен қамтамасыз ету

Пәні ТР 1301 программалау технологиялары

Білім деңгейі жоғары білім (бакалавр)

Оқу формасы жалпы орта білім негізіндегі күндізгі бөлім

Оқу жылы 2010-2011жж.

Оқу ісі жөніндегі проректор

б.ғ.д., профессор

«___» _____ 2011 ж.

Айпеисова С.А.

Байбақтина А.Т.

**«050704-ЕТ және бағдарламалық қаматамсыз ету»
мамандығы күндізгі бөлімінің
1 курс студенттеріне арналған
«Программалау технологиялары» пәнінен
ОҚУ ӘДІСТЕМЕЛІК КЕШЕН**

Тапсырыс № 808
Таралымы 20 дана

Қ. Жұбанов атындағы Ақтөбе мемлекеттік университетінің
Редакциялық-баспа бөлімі
(Ақтөбе қ., Ағ.Жұбановтар к-сі, 263)