

**AXBOROT TEXNOLOGIYALARI VA
KOMMUNIKATSIYALARINI RIVOJLANTIRISH
VAZIRLIGI
TOSHKENT AXBOROT TEXNOLOGIYALARI
UNIVERSITETI FARG'ONA FILIALI**

“Kompyuter injiniring” fakulteti

“Axborot texnologiyalari” kafedrası

“Fizika” fanidan

REFERAT

Bajardi:

Umarova D

2015- Farg'ona

Kalendar

Reja :

1. Kirish

2. Dev C++ ob'yekli dasturlash tilining muhitida ishlash.

3. Dasturiy qism

4. Xulosa

Kirish

Xar bir ishning maqsadi bo'lganidek bizning bajarayotgan kurs ishimiz ham maqsad va vazifalari mavjud. Ular bilan sizni tanishtirib o'tsak. Ishimizdan maqsad eng avvalo har bir talabada C++ dasturlash tilini o'rganish va undan foydalanish ko'nikmasini hosil qilish. Ushbu kursishini vaifalari talabalarni o'zlari o'rgangan dasturlash tili haqida qay darajada bilimga ega bo'lganligini sinovdan o'tkazish hisoblanadi. Kurs ishimiz C++ dasturlash tilida borar ekan biz C++ dasturlash tili haqida malumotga ega bo'lishimiz kerak.

C++ dasturlash tili C tiliga asoslangan. C esa o'z navbatida B va BCPL tillaridan kelib chiqqan. BCPL 1967 yilda Martin Richards tomonidan tuzilgan va operatsion sistemalarni yozish uchun mo'ljallangan edi. Ken Thompson o'zining B tilida BCPL ning ko'p hossalarni kiritgan va B da UNIX operatsion sistemasining birinchi versiyalarini yozgan.

BCPL ham, B ham tipsiz til bo'lgan. Yani o'zgaruvchilarning ma'lum bir tipi bo'lmagan - har bir o'zgaruvchi kompyuter hotirasida faqat bir bayt yer egallagan. O'zgaruvchini qanday sifatda ishlatish esa, yani butun sonmi, kasrli sonmi yoki harfdekmi, dasturchi vazifasi bo'lgan.

C tilini Dennis Ritchie B dan keltirib chiqardi va uni 1972 yili ilk bor Bell Laboratories da, DEC PDP-11 kompyuterida qo'lladi. C o'zidan oldingi B va BCPL tillarining juda ko'p muhim tomonlarini o'z ichiga olish bilan bir qatorda o'zgaruvchilarni tiplashtirdi va bir qator boshqa yangiliklarni kiritdi. Boshlanishda C asosan UNIX sistemalarida keng tarqaldi. Hozirda operatsion sistemalarning asosiy qismi C/C++ da yozilmoqda. C mashina arhitekturasiga bog'langan tildir. Lekin yahshi rejalashtirish orqali dasturlarni turli kompyuter platformalarida ishlaydigan qilsa bo'ladi.

1983 yilda, C tili keng tarqalganligi sababli, uni standartlash harakati boshlandi. Buning uchun Amerika Milliy Standartlar Komiteti (ANSI) qoshida X3J11 texnik komitet tuzildi. Va 1989 yilda ushbu standart qabul qilindi. Standartni dunyo bo'yicha keng tarqatish maqsadida 1990 yilda ANSI va Dunyo Standartlar Tashkiloti (ISO) hamkorlikda C ning ANSI/ISO 9899:1990 standartini qabul qilishdi. Shu sababli C da yozilgan dasturlar kam miqdordagi o'zgarishlar yoki umuman o'zgarishsiz juda ko'p kompyuter platformalarida ishlaydi.

C++ 1980 yillar boshida Bjarne Stroustrup tomonidan C ga asoslangan tarzda tuzildi. C++ juda ko'p qo'shimchalarni o'z ichiga olgan, lekin eng asosiysi u ob'ektlar bilan dasturlashga imkon beradi.

Dasturlarni tez va sifatli yozish hozirgi kunda katta ahamiyat kasb etmoda. Buni ta'minlash uchun ob'ektki dasturlash g'oyasi ilgari surildi. Huddi 70-chi yillar boshida strukturali dasturlash kabi, programmalarni hayotdagi jismlarni modellashtiruvchi ob'ektki orqali tuzish dasturlash sohasida inqilob qildi.

C++ dan tashqari boshqa ko'p ob'ektki dasturlshga yo'naltirilgan tillar paydo bo'ldi. Shulardan eng ko'zga tashlanadigani Xerox ning Palo Altoda joylashgan ilmiy-qidiruv markazida (PARC) tuzilgan Smalltalk dasturlash tilidir. Smalltalk da hamma narsa ob'ektki asoslangan. C++ esa gibrid tildir. Unda C ga o'hshab strukturali dasturlash yoki yangicha, ob'ektki bilan dasturlash mumkin. Yangicha deyishimiz ham nisbiydir. Ob'ektki dasturlash falsafasi paydo bo'lganiga ham yigirma yildan oshayapti.

C++ funksiya va ob'ektkiarning juda boy kutubhonasiga ega. Yani C++ da dasturlashni o'rganish ikki qismga bo'linadi. Birinchisi bu C++ ni o'zini o'rganish, ikkinchisi esa C++ ning standart kutubhonasidagi tayyor ob'ekt/funksiyalarni qo'llashni o'rganishdir.

C++ sistemasi asosan quyidagi qismlardan iborat. Bular dasturni yozish redaktori, C++ tili va standart kutubhonalardir. C++ dasturi ma'lum bir fazalardan o'tadi. Birinchisi dasturni yozish va tahrirlash, ikkinchisi preprocessor amallarini bajarish, kompilyatsiya, kutubhonalardagi ob'ekt va funksiyalarni dastur bilan bog'lash (link), hotiraga yuklash (load) va bajarish (execute).

Dev C++ ob'yektki dasturlash tilining muhitida ishlash.

Alfavit, identifikator, xizmatchi so'zlar.

Alfavit. C++ alfavitiga quyidagi simvollar kiradi.

- Katta va kichik lotin alfaviti xarflari (A,B,...,Z,a,b,...,z)
 - Raqamlar: 0,1,2,3,4,5,6,7,8,9
 - Maxsus simvollar: “ , { } | [] () + - / % \ ; ‘ . : ? < = > _ ! & * # ~ ^
 - Ko'rinmaydigan simvollar (“umumlashgan bushliq simvollar”).
- Leksemalarni uzaro ajratish uchun ishlatiladigan simvollar (misol uchun bushlik, tabulyatsiya, yangi qatorga o'tish belgilari).
- Izohlarda, satrlarda va simvolli konstantalarda boshqa literalalar, masalan rus xarflarini ishlatilishi mumkin.

C++ tilida olti khil turdagi leksemalar ishlatiladi: ehrkin tanlanadigan va ishlatiladigan identifikatorlar, khizmatchi suzlar, konstantalar(konstanta satrlar), amallar(amallar belgilari), azhratuvchi belgilar.

Identifikator. Identifikatorlar lotin xarflari, ostki chiziq belgisi va sonlar ketma ketligidan iborat buladi. Identifikator lotin xarfidan yoki ostki chizish belgisidan boshlanishi lozim.

Misol uchun:

A1, _MAX, adres_01, RIM, rim

Katta va kichik xarflar farklanadi, shuning uchun ohirgi ikki identifikator bir biridan farq qiladi.

Borland kompilyatorlaridan foydalanilganda nomning birinchi 32 xarfi ,ba'zi kompilyatorlarda 8 ta xarfi inobatga olinadi. Bu holda NUMBER_OF_TEST va NUMBER_OF_ROOM identifikatorlari bir biridan farq qilmaydi.

Xizmatchi so'zlar. Tilda ishlatiluvchi ya'ni dasturchi tomonidan uzgaruvchilar nomlari sifatida ishlatish mumkin bulmagan identifikatorlar xizmatchi so'zlar deyiladi.

C ++ tilida quyidagi xizmachi so'zlar mavjud:

int	extern	else
char	register	for
float	typedef	do
double	static	while
struct	goto	switch
union	return	case
long	sizeof	default
short	break	entry
unsigned	continue	
auto	if	

O'zgaruvchilar. (VARIABLES)

O'zgaruvchilar ob'ekt sifatida. Ci++ tilining asosiy tushunchalaridan biri nomlangan hotira qismi – ob'ekt tushunchasidir. Ob'ektning xususiy holi bu o'zgaruvchidir. O'zgaruvchiga qiymat berilganda unga ajratilgan hotira qismiga shu qiymat kodi yoziladi. O'zgaruvchi qiymatiga nomi orqali murojaat qilish mumkin, hotira qismiga esa faqat adresi orqali murojaat qilinadi. O'zgaruvchi nomi bu erkin kiritiladigan identifikator. O'zgaruvchi nomi sifatida xizmatchi so'zlarni ishlatish mumkin emas.

O'zgaruvchilar tiplari. O'zgaruvchilarning qo'yidagi tiplari mavjuddir:

char – bitta simvol;

long char – uzun simvol;

int – butun son;
short yoki short int – qisqa butun son;
long yoki long int – uzun butun son;
float - haqiqiy son;
long float yoki double – ikkilangan haqiqiy son;
long double – uzun ikkilangan haqiqiy son;

Butun sonlar ta'riflanganda ko'rilgan tiplar oldiga unsigned (ishorasiz) ta'rifi kushilishi mumkin. Bu ta'rif qushilgan butun sonlar ustida amallar mod $2n$ arifmetikasiga asoslangandir. Bu erda n soni int tipi hotirada egallovchi razryadlar sonidir. Agar ishorasiz k soni uzunligi int soni razryadlar sonidan uzun bulsa, bu son qiyjmati $k \bmod 2n$ ga teng bo'ladi. Ishorasiz k son uchun ga $-k$ amali $2n - k$ formula asosida hisoblanadi. Ishorali ya'ni signed tipidagi sonlarning eng katta razryadi son ishorasini ko'rsatish uchun ishlatilsa unsigned (ishorasiz) tipdagi sonlarda bu razryad sonni tasvirlash uchun ishlatiladi.

O'zgaruvchilarni dasturning ixtiyoriy qismida ta'riflash yoki qayta ta'riflash mumkin.

Misol uchun:

```
Int a, b1, ac; eki
```

```
Int a;
```

```
int b1;
```

```
int ac;
```

O'zgaruvchilar ta'riflanganda ularning qiymatlari aniqlanmagan bo'ladi. Lekin o'zgaruvchilarni ta'riflashda initsializatsiya ya'ni boshlang'ich qiymatlarini ko'rsatish mumkin.

Misol uchun:

```
Int I=0;
```

```
Char c='k';
```

Typedef ta'riflovchisi yangi tiplarni kiritishga imkon beradi.

Misol uchun yangi COD tipini kiritish:

```
Typedef unsigned char COD;
```

```
COD simbol;
```

KONSTANTALAR. (CONSTANTS)

Konstanta bu o'zgartir ish mumkin bulmagan qiymatdir. C++ tilida besh turdagi konstantalar ishlatilishi mumkin: butun sonlar, haqiqiy sonlar, simvollar, sanovchi konstantalar va nul kursatkich.

1. Ma'lumotlarning butun son turi.

Butun sonlar o'nlik, sakkizlik yoki un oltilik sanoq sistemalarida berilishi mumkin. O'nlik sanoq sistemasida butun sonlar 0-9 raqamlari ketma ketligidan iborat bo'lib, birinchi raqami 0 bulishi kerak emas.

Sakkizlik sanoq sistemasida butun sonlar 0 bilan boshlanuvchi 0-7 raqamlaridan iborat ketma ketlikdir.

O'n oltilik sanoq sistemasida butun son 0x eki 0X bilan boshlanuvchi 0-9 raqamlari va a-f yoki A-F xarflaridan iborat ketma ketlikdir.

Masalan 15 va 22 o'nlik sonlari sakkizlikda 017 va 026, un oltilikda 0xF va 0x16 shaklda tasvirlanadi.

Ma'lumolarning uzun butun son turi.

Oxiriga l eki L harflari quyilgan o'nlik, sakkizlik yoki o'n oltilik butun son.

Ma'lumotlarning ishorasiz (unsigned) butun son turi:

Ohiriga u yoki U harflari quyilgan o'nlik, sakkizlik yoki o'n oltilik oddiy yoki uzun butun son.

2. Ma'lumotlarning haqiqiy son turi:

Olti qismdan iborat bulishi mumkin: butun qism, nuqta, kasr qism, yoki E belgisi, o'nlik daraja, F eki f suffikslari.

Masalan: 66. .0 .12 3.14F 1.12e-12

Ma'lumolarning uzun haqiqiy son turi:

Ohiriga L eki l suffikslari quyilgan haqiqiy son.

Masalan: 2E+6L;

3. Simvolli konstanta.

Bittalik qavslarga olingan bitta yoki ikkita simvol. Misol uchun 'x', '*', '\012', '\0', '\n' - bitta simvolli konstanta; 'dd', '\n\t', '\x07\x07' ikki simvolli konstantalar.

'\` simvolidan boshlangan simvollar eskeyp simvollar deyiladi. Simvolli konstanta qiymati simvolning kompyuterda qabul qilingan sonli kodiga tengdir.

ESC (eskeyp) simvollar jadvali:

Yozilishi	Ichki kodi	Simvoli (nomi)	Ma'nosi
\a	0x07	bel (audible bell)	Tovush signali
\b	0x08	Bs (backspace)	Bir qadam qaytish
\f	0x0C	Ff (form feed)	Sahifani qaytarish
\n	0x0A	lf (line feed)	Qatorni o'tkazish
\r	0x0D	Cr (carriage return)	Karetkani qaytarish
\t	0x09	Ht (horizontal tab)	Gorizontal tabulyatsi
\v	0x0B	Vt (vertical tab)	Vertikal tabulyatsi
\\	0x5C	\ (backslash)	Teskari chiziq
\'	0x27	' (single quote)	Apostrof (oddiy qavs)
\"	0x22	" (double quote)	Ikkilik qavs
\?	0x3F	? (question mark)	Savol Belgisi
\000	000	Любой (octal number)	Simvol sakkizlik kodi
\xhh	0xhh	Любой (hex number)	Simvol o'n oltilik kodi

Satrlı konstanta.

Satrlı konstantalar C++ tili konstantalariga kirmaydi, balki leksemalari alohida tipi hisoblanadi. Shuning uchun adabiyotda satrlı konstantalar satrlı leksemalar deb ham ataladi..

Satrlı konstanta bu ikkilik qavslarga olingan ihtiyoriy simvollar ketma ketligidir. Misol uchun “ Men satrlı konstantaman”.

Satrlar orasiga eskeyp simvollar ham kirishi mumkin. Bu simvollar oldiga \ belgisi quyiladi. Misol uchun :

“\n Bu satr \n uch katorga \n zhoyjlashadi”.

Satr simvolları hotirada ketma-ket joylashtiriladi va har bir satrlı konstanta ohiriga avtomatik ravishda kompilyator tomonidan ‘\0’ simvoli qo’shiladi. Shunday satrning hotiradagi hazhmi simvollar sonı+1 baytga tengdir.

Ketma-ket kelgan va bushlik, tabulyatsiya yoki satr ohiri belgisi bilan ajratilgan satrlar kompilyatsiya davrida bitta satrga aylantiriladi. Misol uchun:

“Salom” “Toshkent ”

satrlari bitta satr deb qaraladi.

“Salom Toshkent”

Bu qoidaga bir necha qatorga yozilgan satrlar ham buysinadi. Misol uchun :

“O’zbekistonga ”

“bahor ”

“keldi”

qatorlari bitta qatorga mos:

“O’zbekistonga bahor keldi”

Agar satrda ‘\’ belgisi uchrasa va bu belgidan so’ng to ‘\n’ satr ohiri belgisigacha bushlik belgisi kelsa bu bushlik belgilari ‘\’ va ‘\n’ belgisi bilan birga satrdan uchiriladi. Satrning uzi keyingi satrda kelgan satr bilan qo’shiladi.

“Ozbekistonga \

“ bahor\

“ keldi”

qatorlari bitta qatorga mos:

“Uzbekistonga bakhor keldi”

Sanovchi konstanta.

Sanovchi konstantalar enum hizmatchi so’zi yordamida kiritilib, int tipidagi sonlarga qulay suzlarni mos quyish uchun ishlatiladi.

Misol uchun:

```
enum{one=1,two=2,three=3};
```

Agar son qiymatlari ko’rsatilmagan bulsa eng chapki so’zga 0 qiymati berilib qolganlariga tartib buyicha usuvchi sonlar mos quyiladi:

```
Enum{zero,one,two};
```

Bu misolda avtomatik ravishda konstantalar quyidagi qiymatlarni qabul qiladi:

```
Zero=0, one=1, two=2;
```

Konstantalar aralash ko’rinishda kiritilishi ham mumkin:

```
Enum(zero,one,for=4,five,seeks).
```

Bu misolda avtomatik ravishda konstantalar quyidagi qiymatlarni qabul qiladi:

Zero=0, one=1, for=4;five=5,seeks=6;

Yana bir misol:

```
Enum BOOLEAN {NO, YES};
```

Konstantalar qiymatlari:

NO=0, YES=1;

Nomlangan konstantalar.

CI ++ tilida o'zgaruvchilardan tashqari nomlangan konstantalar kiritilishi mumkin. Bu konstantalar qiymatlarini dasturda o'zgartirish mumkin emas. Konstantalar nomlari dasturchi tomonidan kiritilgan va hizmatchi so'zlardan farqli bo'lgan identifikatorlar bulishi mumkin. Odatda nom sifatida katta lotin harflari va ostiga chizish belgilari kombinaciyasidan iborat identifikatorlar ishlatiladi. Nomlangan konstantalar quyidagi shaklda kiritiladi:

```
Const tip konstanta_nomi=konstanta_kiyjmati.
```

Misol uchun:

```
Const double EULER=2.718282;
```

```
Const long M=999999999;
```

```
Const R=765;
```

Ohirgi misolda konstanta tipi kursatilmagan, bu konstanta int tipiga tegishli deb hisoblanadi.

Nul ko'rsatkich.

NULL- ko'rsatkich yagona arifmetik bulmagan konstantadir. Konkret realizatsiyalarda null ko'rsatkich 0 eki 0L eki nomlangan konstanta NULL orqali tasvirlanishi mumkin. Shuni aytish lozimki bu konstanta qiymati 0 bo'lishi eki '0' simvoli kodiga mos kelishi shart emas.

Quyidagi jadvalda konstantalar chegaralari va mos tiplari ko'rsatilgan:

Ma'lum otlar turi	Ha jm , bit	Qiymatlar chegarasi	Tip vazifasi
Unsigne d char	8	0...255	Kichik butun sonlar va simvollar kodlari
Char	8	-128...127	Kichik butun sonlar va ASII kodlar
Enum	16	- 32768...327 67	Butun sonlar tartiblangan katori
Unsigne d int	16	0...65535	Katta butun sonlar
Short int	16	-	Kichik butun sonlar,

		32768...32767	tsikllarni boshqarish
Int	16	-32768...32767	Kichik butun sonlar, tsikllarni boshqarish
Unsigned long	32	0...4294967295	Astronomik masofalar
Long	32	-147483648...2147483647	Katta sonlar
Float	32	3.4E-32...3.4E+38	Ilmiy hisoblar (7 raqam)
Double	64	1.7E-308...1.7E+308	Ilmiy hisoblar(15 raqam)
Long double	80	3.4E-4932...1.1E+4932	Moliyaviy hisoblar (19 raqam)

TANLASH OPERATORLARI.

Shartli operator. Shartli operator ikki ko'rinishda ishlatilishi mumkin:

```
If (ifoda)
1- operator
Else
2- operator
eki
If (ifoda)
1-operator
```

Shartli operator bajarilganda avval ifoda hisoblanadi ; agar qiymat rost ya'ni nol'dan farqli bo'lsa 1- operator bajariladi. Agar qiymat yolg'on ya'ni nol' bo'lsa va else ishlatilsa 2-operator bajariladi. Else qism har doim eng yaqin if ga mos qo'yiladi.

```
if( n>0)
  if(a>b)
    Z=a;
  else
    Z=b;
```

Agar else qismni yuqori if ga mos quyish lozim bo'lsa, figurali qavslar ishlatish lozim.

```
if( n>0) {
    if(a>b)
        z=a;
}
else
    z=b;
```

Misol tariqasida uchta berilgan sonning eng kattasini aniqlash dasturini ko'ramiz:

```
#include <iostream.h>
void( )
{
    float a,b,c,max;
    Cout <<"\n a="; Cin>>a;
    Cout <<"\n b="; Cin>>b;
    Cout <<"\n c="; Cin>>c;
    if (a>b)
        if (a>c) max=a else max=c;
    else
        if b>c then max=b else max=c;
    Cout <<"\n" <<max;
}
```

Keyingi misolda kiritilgan ball va maksimal ball asosida baho aniqlanadi:

```
#include <iostream.h>
void main( )
{
    float ball,max_ball,baho;
    Cout<<"\n ball="; Cin>>("%f",&ball);
    Cout<<"\n max_ball="; Cin>>max_ball;
    d=ball/max_ball;
    if (d>0.85) baho=5 else
        if (d>0.75) baho=4 else
            if (d>0.55) then baho=3 else baho=2;
    Cout<<"\n baho;
}
```

Kalit bo'yicha tanlash operatori. Kalit bo'yicha o'tish switch operatori umumiy ko'rinishi qo'yidagicha

```
Switch(<ifoda>) {
    Case <1-kiymat>:<1-operator>
        ...
    break;
    ...
    default: <operator>
```

```

...
case: <n-operator>;
}

```

Oldin qavs ichidagi butun ifoda hisoblanadi va uning qiymati hamma variantlar bilan solishtiriladi. Biror variantga qiymat mos kelsa shu variantda ko'rsatilgan operator bajariladi. Agar biror variant mos kelmasa default orqali ko'rsatilgan operator bajariladi. Break operatori ishlatilmasa shartga mos kelgan variantdan tashqari keyingi variantdagi operatorlar ham avtomatik bajariladi. Default; break va belgilangan variantlar ixtiyoriy tartibda kelishi mumkin. Default yoki break operatorlarini ishlatish shart emas. Belgilangan operatorlar bo'sh bo'lishi ham mumkin.

Misol tariqasida bahoni son miqdoriga qarab aniqlash dasturini ko'ramiz.

```

Include <iostream.h>

```

```

Int baho;

```

```

Cin>> baho;

```

```

Switch(baho)

```

```

{case 2:Cout <<“\n emon”;break;
case 3:Cout <<“\n urta”;break;
case 4:Cout <<“\n yahshi”;break;
case 5:Cout <<“\n a'lo”;break;
default: Cout <<“\n baho notugri kiritilgan”;
};
}

```

Keyingi misolimizda kiritilgan simvol unli harf ekanligi aniqlanadi:

```

Include <iostream.h>

```

```

Int baho; Char c; Cin >> c;

```

```

Switch(c)

```

```

{case 'a':
case 'u':
case 'o':
case 'i':
Cout <<“\n Kiritilgan simvol unli harf”;break;
default: Cout <<“\n Kiritilgan simvol unli harf emas”;
};
}

```

SIKL OPERATORLARI.

While operatori. While operatori quyidagi umumiy ko'rinishga egadir:

```

While(ifoda)

```

```

Operator

```

Bu operator bajarilganda avval ifoda hisoblanadi. Agar uning qiymati 0 dan farqli bo'lsa operator bajariladi va ifoda qayta hisoblanadi. To ifoda qiymati 0 bo'lmaguncha tsikl qaytariladi.

Agar dasturda while (1); satr quyilsa bu dastur hech qachon tugamaydi.

Misol. Berilgan n gacha sonlar yigindisi.

```
Void main()
{
long n,i=1,s=0;
cin >>n;
while (i<= n )
    s+=i++;
Cout<<"\n s="<< s;
};
```

Bu dasturda s+=i++ ifoda s=s+i; i=i+1 ifodalarga ekvivalentdir.

Quyidagi dastur to nuqta bosilmaguncha kiritilgan simvollar va qatorlar soni hisoblanadi:

```
Void main()
{
long nc=0,nl=0;
char c="";
while (c!= '.')
{++nc;
if (c =='\n') ++nl;
};
Cout<<("% 1d\n", nc);
Cout <<"\n satrlar="<< nl<<"simvollar="<< nc;
};
```

Do-While operatori. Do-While operatori umumiy ko'rinishi qo'yidagicha:

do

Operator

While(ifoda)

Tsikl operatorining bu ko'rinishida avval operator bajariladi so'ngra ifoda hisoblanadi. Agar uning qiymati 0 dan farqli bo'lsa operator yana bajariladi va hokazo. To ifoda qiymati 0 bo'lmaguncha tsikl qaytariladi.

Misol. Berilgan n gacha sonlar yigindisi.

```
Void main()
{
long n,i=1,s=0;
cin >>n;
do
    s+=i++;
```

```
while (i<= n );
Cout<<"\n s="<< s;
};
```

Bu dasturning kamchiligi shundan iboratki agar n qiymati 0 ga teng eki manfiy bo'lsa ham, tsikl tanasi bir marta bajariladi va s qiymati birga teng bo'ladi.

Keyingi misolimizda simvolning kodini monitorga chiqaruvchi dasturni kuramiz. Bu misolda tsikl to ESC (kodi 27) tugmasi bosilmaguncha davom etadi. Shu bilan birga ESC klavishasining kodi ham ekranga chiqariladi.

```
# include <iostream.h>;
main ()
{
char d; int I;
do
cin>>d;
i=c;
Cout<<"\n "<<i;
while(i!=27);
};
```

For operatori. For operatori umumiy ko'rinishi qo'yidagicha:

```
For( 1-ifoda;2- ifoda; 3-ifoda)
Operator
```

Bu operator qo'yidagi operatorga mosdir.

```
1-ifoda;
while(2-ifoda) {
operator
3-ifoda
}
```

Misol. Berilgan n gacha sonlar yigindisi.

```
# include <iostream.h>;
void main {
int n;
Cin>>n;
for(int i=1,s=0;i<=n; i++, s+=i);
Cout<<"\n",s;
};
```

FOR operatori tanasi bu misolda bush, lekin C ++ tili grammatikasi qoidalari FOR operatori tanaga ega bo'lishini talab qiladi. Bush operatorga mos keluvchi nuqta vergul' shu talabni bajarishga hizmat qiladi.

Keyingi dasturda kiritilgan jumlada satrlar, so'zlar va simvollar sonini hisoblanadi.

```
# include <iostream.h>;
#define yes 1
#define no 0
```

```

void main()
{
int c, nl, nw, inword;
inword = no;
nl = nw = nc = 0;
for(char c='';c!='.';cin>> c)
{++nc;
if (c == '\n')
++nl;
if (c==' ' ||c=='\n' ||c=='\t')
inword = no;
else if (inword == no)
inword = yes;
++nw;
}
Cout <<"\n satrlar="<< nl<<"suzlar="<< nw<<"simvollar="<< nc;
}

```

Programma har gal so'zning birinchi simvolini uchratganda, mos o'zgaruvchi qiymatini bittaga oshiradi. INWORD o'zgaruvchisi programma so'z ichida ekanligini kuzatadi. Oldiniga bu o'zgaruvchiga so'z ichida emas ya'ni NO qiymati beriladi. YES va NO simvolik konstantalardan foydalanish dasturni o'qishni engillashtiradi.

NL = NW = NC = 0 katori kuyidagi katorga mos keladi;

NC = (NL = (NW = 0));

Dasturiy qism.

Kalendar

Ushbu dasturni yozishni boshlaymiz, demak biz eng avval C++ dan yangi **project** ochib olamiz. Unga 3 dona o'zgaruvchi kiritamiz ya'ni **“int kun, oy, yil”** deb olamiz. Keyin esa **oylar[]** degan bir o'lchamli massiv ochib olamiz. Uni elementlari yil oylarining kunlar soni bilan belgilanadi ya'ni **“int oylar[] = {0,31,((yil%4==0,yil%100!=0)?29:28),31,30,31,30,31,31,30,31,30,31}”**, bu yerda 0-element 0 kundan iborat, 1- element yanvar oyi kunlari bo'lib u 31 kundan iborat, 2- element fevral oyi kunlari bo'lib u kabisa yilida 29 kundan boshqa yillarda 28 kundan iborat (*tartib raqami 4 ga bo'linib, 100 ga bo'linmaydigan yillar kabisa yili hisoblanadi.*), 3-element mart oyi kunlari bo'lib u 31 kundan iborat, 4- element aprel oyi kunlari bo'lib u 30 kundan iborat, 5- element may oyi kunlari bo'lib u 31 kundan iborat, 6- element iyun oyi kunlari bo'lib u 30 kundan iborat, 7- element iyul oyi kunlari bo'lib u 31 kundan iborat, 8- element avgust oyi kunlari bo'lib u 31 kundan iborat, 9- element sentabr oyi kunlari bo'lib u 30 kundan iborat, 10- element oktabr oyi kunlari bo'lib u 31 kundan iborat, 11- element noyabr oyi kunlari bo'lib u 30 kundan iborat va nihoyat 12- element dekabr oyi kunlari bo'lib u 31 kundan iborat. Keyin **oykunlari** degan o'zgaruvchi kiritib uni 0 ga tenglab olamiz **“int oykunlari = 0”**. Chunki bu oyning kiritilgan kunigacha bo'lgan kunlari sonini belgilaydi. Bu yerda oy uchun sikl ochib olamiz

for (int i=0; i<oy; i++). So'ng bu oyning kiritilgan kunigacha bo'lgan kunlari sonini bundan avvalgi oylarning kunlari soni jamlanmasiga qo'shamiz **"oykunlari += oylar[i]"**. Endi natijani biror narsa bilan belgilab olamiz, masalan **"natija"** degan o'zgaruvchi ochib olamiz va uni 0 ga tenglaymiz **"int natija = 0"**. Bizda chiqishi kerak bo'lgan natija bu kiritilgan oygacha bo'lgan oylarning kunlari sonini yig'indisiga kiritilgan kunni qo'shishdan chiqqan son hisoblanadi buni quidagicha belgilaymiz **natija = oykunlari + kun**. Nihoyat ekranga kiritilgan kun yilning qaysi kuni ekanligini chiqaruvchi kodni yozamiz u quidagicha bo'ladi **"cout<<"Ushbu kun yilning "<< natija<<"-kuni."<<endl"**. Kiritilgan yil kabisa yili bo'lsa uni ekranga chiqarish uchun bitta shart qo'yib olamiz **if(yil%4==0, yil%100!=0)**, bu degani agar yil 4 ga bo'linganda qoldiq 0 ga teng bo'lsa va yil 100 ga bo'linganda qoldiq 0 ga teng bo'lmasa u holda ushbu yil kabisa yili hisoblanadi va ekranda buni chiqarish uchun **"cout<<yil<<"-yil KABISA yili"<<endl"** ushbu kod kiritiladi.

C++ dasturida kodi.

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int kun , oy , yil;
    int dx, xkun;
    int firstday = 3 - 2 ;//birinchi kun chorshanba
    int firstyear = 1980;

    cout << "Kunni kiriting: "; cin >> kun;
    cout << "Oyni kiriting: "; cin >> oy;
    cout << "Yilni kiriting: "; cin >> yil;

    cout << yil << "-yil ";
    cout << kun << "-";
```

```

switch ( oy ){
    case 1: cout << "Yanvar"; break; //31 kun
    case 2: cout << "Fevral"; break; //28 kun
    case 3: cout << "Mart"; break; //31 kun
    case 4: cout << "Aprel"; break; //30 kun
    case 5: cout << "May"; break; //31 kun
    case 6: cout << "Iyun"; break; //30 kun
    case 7: cout << "Iyul"; break; //31 kun
    case 8: cout << "Avgust"; break; //31 kun
    case 9: cout << "Sentabr"; break; //30 kun
    case 10: cout << "Oktabr"; break; //31 kun
    case 11: cout << "Noyabr"; break; //30 kun
    case 12: cout << "Dekabr"; break; //31 kun
}

cout << " ( ";
xkun = 0;
for(int k = firstyear, l = 365; k <= yil; k++){
    if(k == yil)
        for(int i = 1 ; i < oy ; i++){
            switch (i){
                case 1:
                case 3:
                case 5:
                case 7:
                case 8:
                case 10:
                case 12: dx = 31; break;
                case 4:
                case 6:
                case 9:
                case 11: dx = 30; break;
                case 2: dx = 28; if( yil % 4 == 0 ) dx++; break;
            }
            xkun = xkun + dx;
        }//end for2
    else {
        xkun = xkun + l; if( k % 4 == 0 ) xkun++;
    }
}

switch ( (firstday + kun + xkun)%7 ){
    case 0: cout<<"Yakshanba"; break;
    case 1: cout<<"Dushanba"; break;

```

```

        case 2: cout<<"Seshanba"; break;
        case 3: cout<<"Chorshanba"; break;
        case 4: cout<<"Payshanba"; break;
        case 5: cout<<"Juma"; break;
        case 6: cout<<"Shanba"; break;
    }
    cout<<" )";

int oylar[] =
    {0,31,((yil%4==0,yil%100!=0)?29:28),31,30,31,30,31,31,30,31,30,31};
int oykunlari = 0;
for(int i=0; i<oy; i++)
    oykunlari += oylar[i];
int natija = 0;
natija = oykunlari + kun;
cout<<"Ushbu kun yilning " << natija<<"-kuni." << endl;
if(yil%4==0, yil%100!=0)
    {cout<<yil<<"-yil KABISA yili" << endl;}

system("PAUSE");
return EXIT_SUCCESS;
}

```

Dasturda ko'rinishi

```

#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int kun , oy , yil;
    int dx, xkun;
    int firstday = 3 - 2 ;//birinchi kun chorshanba
    int firstyear = 1980;

    cout << "Kunni kiriting: "; cin >> kun;
    cout << "Oyni kiriting: "; cin >> oy;
    cout << "Yilni kiriting: "; cin >> yil;

    cout << yil << "-yil ";
    cout << kun << "-";
    switch ( oy ){

```

```
case 3: cout << "Mart"; break; //31 kun
case 4: cout << "Aprel"; break; //30 kun
case 5: cout << "May"; break; //31 kun
case 6: cout << "Iyun"; break; //30 kun
case 7: cout << "Iyul"; break; //31 kun
case 8: cout << "Avgust"; break; //31 kun
case 9: cout << "Sentabr"; break; //30 kun
case 10: cout << "Oktabr"; break; //31 kun
case 11: cout << "Noyabr"; break; //30 kun
case 12: cout << "Dekabr"; break; //31 kun
}

cout << " ( ";
xkun = 0;
for(int k = firstyear, l = 365; k <= yil; k++){
    if(k == yil)
        for(int i = 1 ; i < oy ; i++){
            switch (i){
                case 1:
```

```
switch (i){
    case 1:
    case 3:
    case 5:
    case 7:
    case 8:
    case 10:
    case 12: dx = 31; break;
    case 4:
    case 6:
    case 9:
    case 11: dx = 30; break;
    case 2: dx = 28; if( yil % 4 == 0 ) dx++; break;
}
xkun = xkun + dx;
} //end for2
else {
    xkun = xkun + 1; if( k % 4 == 0 ) xkun++;
}
}

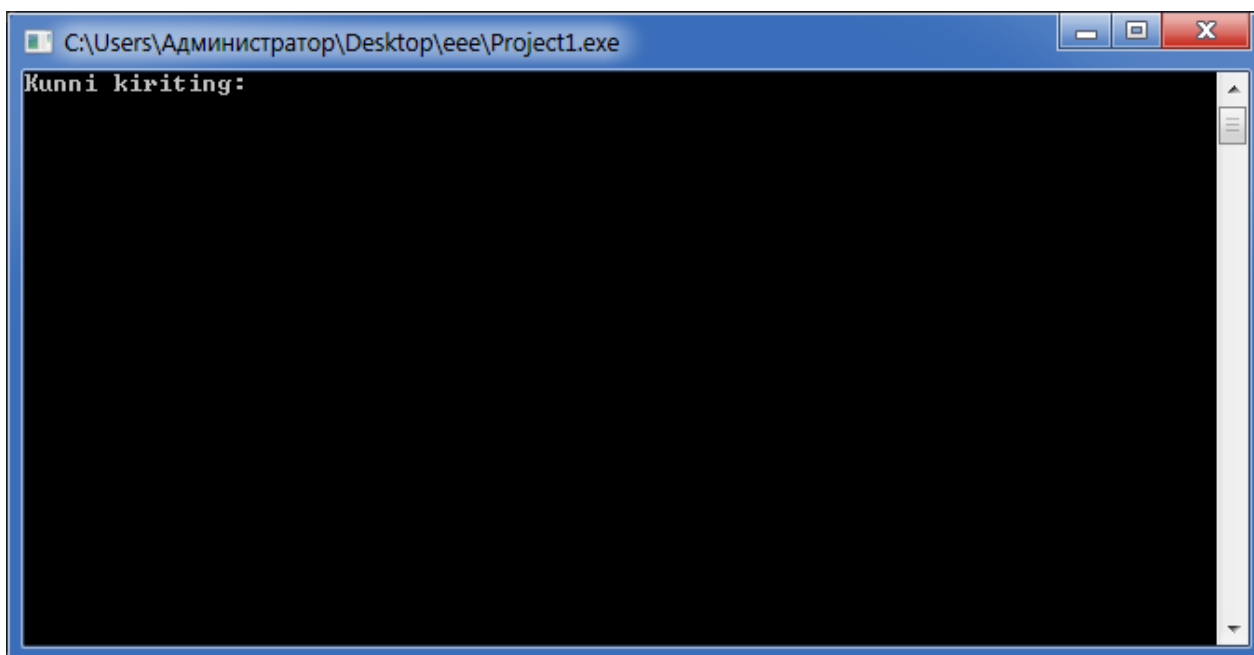
switch ( (firstday + kun + xkun)%7 ){
    case 0: cout<<"Yakshanba"; break;
    case 1: cout<<"Dushanba"; break;
```

```
case 1: cout<<"Dushanba"; break;
case 2: cout<<"Seshanba"; break;
case 3: cout<<"Chorshanba"; break;
case 4: cout<<"Payshanba"; break;
case 5: cout<<"Juma"; break;
case 6: cout<<"Shanba"; break;
}
cout<<" )"<<endl;

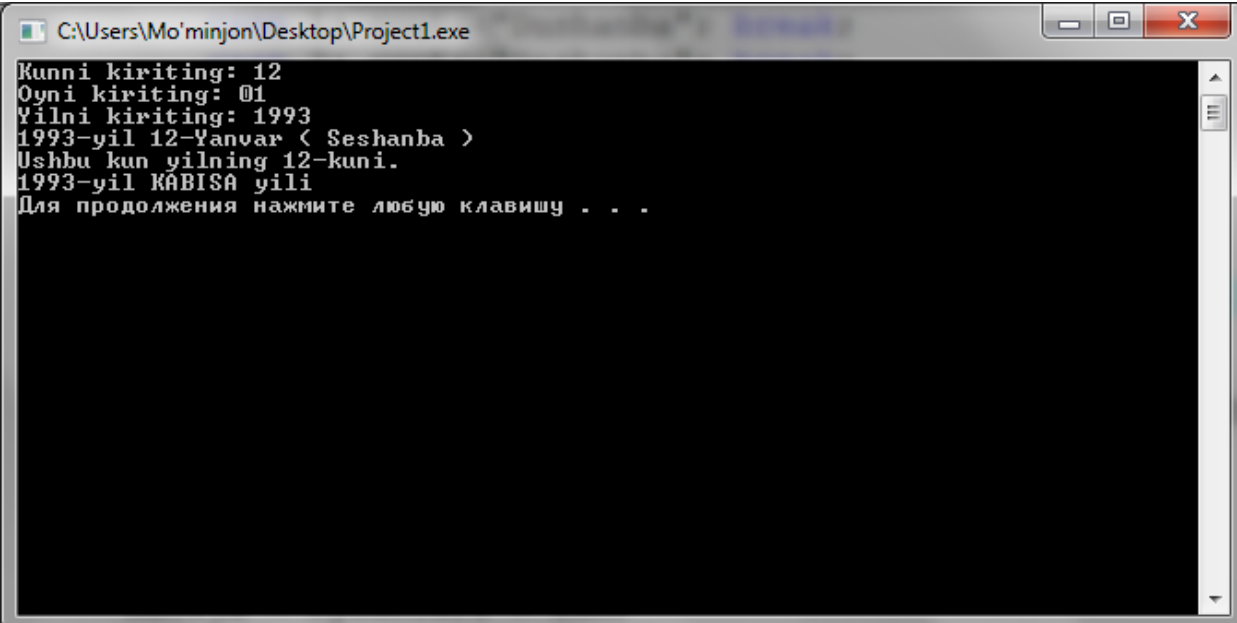
int oylar[] =
{0, 31, ((yil%4==0,yil%100!=0)?29:28), 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
int oykunalari = 0;
for(int i=0; i<oy; i++)
oykunalari += oylar[i];
int natija = 0;
natija = oykunalari + kun;
cout<<"Ushbu kun yilning "<< natija<<"-kuni."<<endl;
if(yil%4==0, yil%100!=0)
{cout<<yil<<"-yil KABISA yili"<<endl;}

system("PAUSE");
return EXIT_SUCCESS;
}
```

Kunni ,oyni va yilni kiritamiz.



Natija



```
C:\Users\Mo'minjon\Desktop\Project1.exe
Kunni kiriting: 12
Oyni kiriting: 01
Yilni kiriting: 1993
1993-yil 12-Yanvar < Seshanba >
Ushbu kun yilning 12-kuni.
1993-yil KABISA yili
Для продолжения нажмите любую клавишу . . .
```

Xulosa.

Aholi malakaviy darajasi tabiiy holda mamlakat iqtisodiyoti rivojlanishida, shuning bilan birga demokratik jamiyatning shakllanishida, millatning ijtimoiy-ma'naviy kamol topishida muhim rol o'ynaydi. Biz talabalarning bilim saviyasi, fanni nechog'li o'zlashtirganligimiz esa, bizning bajargan mustaqil topshiriqlarimizda o'z ifodasini topadi.

Xulosa qilib aytadigan bo'lsak, kurs ishlaridan asosiy maqsad: biz ta'lim bo'yicha nazariy va amaliy bilimlarni mustahkamlash va kengaytirish, olingan bilimlarni muayyan ilmiy, texnikaviy, ishlab chiqarish, iqtisodiy, ijtimoiy, madaniy vazifalarni hal etishda qo'llashdan iboratdir.

Kurs ishlarining biz talabalar uchun ahamiyatli tomoni shundaki: ijodiy ishlash, ishlab chiqilayotgan masalaning qo'yilish jarayonidan boshlab uni to'la nihoyasiga yetkazish bo'yicha qaror qabul qilishda bo'lgan mas'uliyatni his etishni o'rganish, zamonaviy ishlab chiqarish, iqtisodiyot, texnika va madaniyatning rivojlanishi sharoitida biz talabalarni mustaqil ishlashga tayyorgarligini ta'minlashga imkon beradi.

Kurs ishini bajarishda men o'z imkoniyatimdan kelib chiqib, o'zim xohlagan tarzda dasturlar bog'lamini tuzish imkoniga ega bo'ldim.

Adabiyotlar

1. Gulomov S.S. va boshkalar, Iktisodiy informatika. Toshkent 1999 yil.
2. Gulomov S.S. va boshkalar. «Axborot texnologiyalari va tizimlari» T.2000 y.
3. A.T. Karimov va boshk. Xalk xujaligida AS yaratish asoslari. T., «Uzbekiston» - 1992 yil.
4. M.Ikromov. avtomatlashtirilgan boshkarish sistemalari T. «Ukituvchi» - 1989y.
5. ASU na promyshlennom pred-ti: Metodi sozd-ya.Spravochnik S.B. Mixaylev i dr. Energoizdat. 1989.
6. Organizatsiya vzaimodeystviya cheloveka s texnicheskimi sredstvami ASU M. Vysshaya shkola, 1990-91.
7. Dopil'chenko I.A. i dr. Avtomatizirovannye sistemy upravleniya predpriyatiyami. M. Mashinostroenie 1984.
8. SHixayev K.N. Protsessy integratsii v ASU M. Finansy i statistika 1982.
9. Informatika fanidan maxruza matnlari. – 2002y. Fargona

QO‘SHIMCHA ADABIYOTLAR.

1. Skuruxin V.I. i dr. Optimizatsii funktsionirovaniya ASU predpriyatiem. Kiev. Nauka.
2. Mamikonov A.G. Osnovy postroenie ASU M. 1981.
3. Pavlidenie G. Algoritmy mashinnoy grafiki. 1985.