

## 4-Ma'ruza. Grafika. C++ da kiritish chiqarish operatorlari va arifmetik amallar

### Ma'ruza rejasi:

- 4.1. Ma'lumotlarni kiritish va chiqarish;
- 4.2. Amallar.

### Kalit so'zlar:

#### 4.3. Ma'lumotlarni kiritish va chiqarish

**Formatli chiqarish – printf.** Chiqarish printf funksiyasi ko'rsatilgan parametrlarni standart oqimga chiqarish uchun ishlatiladi. Standart oqim tushunchasi keyingi boblarda yoritiladi. Xozircha standart oqim sifatida monitor tushunilishi yetarlidir.

Funksiya **stdio.h** modulida joylashgan bo'lib, umumiy ko'rinishi quyidagichadir:

**printf(control,arg1,arg2,...)**

Bunda control boshqaruvchi qator deb atalib ikki turdagi simvollardan iborat bo'ladi: oddiy chiqariluvchi simvollar va navbatdagi parametрни o'zgartirib chiqaruvchi spesifikasiyalar.

Har bir spesifikasiya % simvolidan boshlanib o'zgartirish turini ko'rsatuvchi simvol bilan tugaydi.

O'zgartirish simvollari quyidagilardan iborat.

Butun sonlar uchun:

d– parametr ishorali o'nlik butun songa aylantiriladi.

u - parametr ishorasiz o'nlik butun songa aylantiriladi.

o – parametr ishorasiz va birinchi raqami 0 bo'lmagan sakkizlik songa aylantiriladi.

x – parametr ishorasiz va 0x belgisiz o'n oltilik songa aylantiriladi.

X – parametr xuddi x kabi. Faqat harf bilan ko'rsatiluvchi raqamlar katta harf ya'ni A,B,C,D,E,F sifatida yoziladi.

Haqiqiy sonlar uchun:

e – parametr float yoki double turidagi son deb qaraladi va ishorali m.nnnnnne+-xx ko'rinishidagi o'nlik songa keltiriladi.

E – parametr xuddi e kabi. Faqat mantissa belgisi katta harf ya'ni E sifatida yoziladi.

f - parametr float yoki double turidagi son deb qaraladi va ishorali m.nnnnnn ko'rinishidagi o'nlik songa keltiriladi.

g – parametr berilgan son qiymati va aniqligi uchun eng ixcham %e yoki %f tanlaydi.

G – parametr xuddi g kabi. Faqat mantissa belgisi katta harf ya'ni E sifatida yoziladi.

Simvol va satr uchun:

c – parametr bitta simvol deb qaraladi.

s – parametr satr simvollar no'linchi simvol uchramaguncha yoki ko'rsatilgan sondagi simvollar bosiladi.

Misol:

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

```
int main()
```

```
{
```

```
int num = -27; int number = 27; float f = 123.456;
```

```
char r = 'a'; char str[4] = "abc";
```

```
printf("%d\n",num); /* -27 */
```

```
printf("%u\n",number); /* 27 */
```

```
printf("%o\n",number); /* 33 */
```

```
printf("%x\n",number); /* lb */
```

```
printf(" %f\n", f); /* 123.456001 */
```

```
printf("%e\n", f); /* 1.23456e+02 */
```

```
printf("%E\n", f); /* 1.23456E+02 */
```

```
printf("%c\n",r); /* a */
```

```
printf("%s\n", str); /* abc */
```

```
return 0;
```

}

Prosent % belgisi va o'zgartirish simvoli orasiga quyidagi simvollarni qo'yish mumkin.

Chiqarilayotgan argument chapga tekislash lozimligini ko'rsatuvchi minus belgisi.

Maydon minimal uzunligini ko'rsatuvchi raqamlar qatori.

Maydon uzunligini keyingi raqamlar qatoridan ajratuvchi nuqta.

Biror qatordan qancha simvol ajratib olish lozimligini hamda float yoki double turidagi sonlarda nuqtadan keyin qancha kasr raqamlari bosib chiqarilishini ko'rsatuvchi raqamlar ketma-ketligi.

Chiqarilayotgan son long turiga tegishli ekanligini ko'rsatuvchi uzun o'nlik markeri l.

% dan keyingi simvol o'zgartirish simvoli bo'lmasa u bosmaga chiqariladi.

% simvolini o'zini bosmaga chiqarish uchun %% belgisini berish lozim.

Quyidagi jadval har xil spesifikasiyalarni "HELLO, WORLD" (12 simvolli) so'zini bosishga ta'sirini ko'rsatadi. Bu yerda har bir maydon uzunligini ko'rsatish uchun maydon oxiriga ikki nuqta qo'yilgan.

:%10S:	:HELLO, WORLD:
--------	----------------

:%10-S:	:HELLO, WORLD:
---------	----------------

:%20S:	: HELLO, WORLD:
--------	-----------------

:%-20S:	:HELLO, WORLD :
---------	-----------------

:%20.10S:	: HELLO, WOR:
-----------	---------------

:%-20.10S:	:HELLO, WOR :
------------	---------------

:%.10S:	:HELLO, WOR:
---------	--------------

**Turlar maksimal va minimal qiymatlari.** Turli turlar maksimal va minimal qiymatlari <LIMITS.H> faylidagi konstantalarda saqlanadi.

Quyidagi dasturda char turidagi bitlar soni va char turi maksimal va minimal qiymati ekranga chiqariladi

**#include<stdio.h>**

```
#include<limits.h>
```

```
int main()
```

```
{
```

```
printf("CHAR_BIT = %d\n",CHAR_BIT);
```

```
printf("CHAR_MIN = %d CHAR_MAX = %d\n",CHAR_MIN,CHAR_MAX);
```

```
return 0;
```

```
}
```

Quyidagi dasturda short, int, long turlari maksimal va minimal qiymati ekranga chiqariladi

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

```
#include<limits.h>
```

```
int main()
```

```
{
```

```
printf("SHRT_MIN = %d SHRT_MAX = %d\n",SHRT_MIN,SHRT_MAX);
```

```
printf("INT_MIN = %d INT_MAX = %d\n",INT_MIN,INT_MAX);
```

```
printf("LONG_MIN = %ld LONG_MAX = %ld\n",LONG_MIN,LONG_MAX);
```

```
return 0;
```

```
}
```

Quyidagi dasturda unsigned char, unsigned short, unsigned int, unsigned long turlari maksimal va minimal qiymati ekranga chiqariladi

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

```
#include<limits.h>
```

```
int main()
```

```
{
```

```
printf("UCHAR_MAX = %u\n",UCHAR_MAX);
```

```
printf("USHRT_MAX = %u\n",USHRT_MAX);
```

```
printf("UINT_MAX = %u\n",INT_MAX);
```

```
printf("ULONG_MAX = %ul\n",ULONG_MAX);
```

```
return 0;
```

}

Formatli kiritish Scanf. Scanf funksiyasi stdio.h modulida joylashgan bo'lib, umumiy ko'rinishi quyidagichadir:

Scanf(control, arg1, arg2,...)

Funksiya standart oqimdan simvollarini o'qib boshqaruvchi qator asosida formatlab mos parametrlarga yozib qo'yadi. Parametr ko'rsatkich bo'lishi lozim.

Boshqaruvchi qator quyidagi o'zgartirish spesifikasiyalaridan iborat:

Bo'shliq, tabulyasiya, keyingi qatarga o'tish simvollarini;

Oddiy simvollar (% dan tashqari) kiritish oqimidagi navbatdagi simvollar bilan mos kelishi lozim;

% simvolidan boshlanuvchi spesifikasiya simvollarini;

% simvolidan boshlanuvchi qiymat berishni ta'qiqlovchi \* simvoli;

% simvolidan boshlanuvchi maydon maksimal uzunligini ko'rsatuvchi son;

quyidagi spesifikasiya simvollarini ishlatish mumkin:

d – ishorali o'nli butun son kutilmoqda.

o – ishorali sakkizlik butun son kutilmoqda.

x – ishorali o'n oltilik butun son kutilmoqda.

h - ishorasiz o'nlik son kutilmoqda.

c – bitta simvol kutilmoqda.

s – satr kutilmoqda.

f - float turidagi son kutilmoqda. Kiritilayotgan sonning butun raqamlari va nuqtadan so'ng kasr raqamlari soni va E yoki e belgisidan so'ng mantissa raqamlari soni ko'rsatilishi mumkin.

**#include <stdio.h>**

**int main(void)**

{

**unsigned width, precision;**

```

int number = 256;

double weight = 242.5;

printf("What field width?\n");

scanf("%d", &width);

printf("The number is :%*d:\n", width, number);

printf("Now enter a width and a precision:\n");

scanf("%d %d", &width, &precision);

printf("Weight = %*.*f\n", width, precision, weight);

printf("Done!\n");

return 0;

}

```

**Lokal va global o'zgaruvchilar.** C tilida o'zgaruvchi ta'rif albatta blok boshida joylashishi lozim.

O'zgaruvchi mavjudlik sohasi deb shu o'zgaruvchiga ajratilgan xotira mavjud bo'lgan dastur qismiga aytiladi. O'zgaruvchi ko'rinish sohasi deb o'zgaruvchi qiymatini olish mumkin bo'lgan dastur qismiga aytiladi. Biror blokda ta'riflangan o'zgaruvchi lokal o'zgaruvchi deyiladi. Har qanday blokdan tashqarida ta'riflangan o'zgaruvchi global o'zgaruvchi deyiladi.

Lokal o'zgaruvchi mavjudlik va ko'rinish sohasi ta'rifdan to shu ta'rif joylashgan blok oxirigachadir.

Tashqi blokda o'zgaruvchi nomi shu blokda joylashgan yoki shu blokda ichki blokda o'zgaruvchi nomi bilan bir xil bo'lmasligi kerak.

Global o'zgaruvchi mavjudlik sohasi ta'rifdan to dastur oxirigachadir.

Agar ichki blokda o'zgaruvchi nomi global o'zgaruvchi nomi bilan bir xil bo'lsa lokal o'zgaruvchi ko'rinish sohasida global o'zgaruvchi ko'rinmay qoladi.

Misol:

<b>#include&lt;stdio.h&gt;</b>	<b>int main()</b>
<b>int i = 5;</b>	<b>{</b>
<b>int k = 6;</b>	<b>int i = 9;</b>

```
printf("%d\n",i);           return 0;
printf("%d\n",k);           }
```

Natija:

9

6

## 4.2. Amallar

**Arifmetik amallar.** Amallar odatda unar, ya'ni bitta operandga qo'llaniladigan amallarga va binar, ya'ni ikki operandga qo'llaniladigan amallarga ajratiladi.

Binar amallar additiv ya'ni + qo'shish va – ayirish amallariga, hamda multiplikativ, ya'ni \* ko'paytirish, / bo'lish va % modul olish amallariga ajratiladi.

Butun sonni butun songa bo'lganda natija butun songacha yaxlitlanadi. Misol uchun,  $20/3 = 6$ ;  $(-20)/3 = -6$ ;  $20/(-3) = -6$ .

Modul amali butun sonni butun songa bo'lishdan hosil bo'ladigan qoldiqqa tengdir. Agar modul amali musbat operandlarga qo'llanilsa, natija ham musbat bo'ladi, aks holda natija ishorasi kompilyatorga bog'liqdir.

Unar amallarga ishorani o'zgartiruvchi unar minus – va unar plyus + amallari kiradi. Bundan tashqari inkrement ++ va dekrement -- amallari ham unar amallarga kiradi.

Inkrement ++ unar amali qiymatni 1 ga oshirishni ko'rsatadi. Amalni prefiks, ya'ni ++i ko'rinishda ishlatish oldin o'zgaruvchi qiymatini oshirib, so'ngra foydalanish lozimligini, postfiks esa i++ ko'rinishda ishlatish oldin o'zgaruvchi qiymatidan foydalanib, so'ngra oshirish kerakligini ko'rsatadi. Misol uchun, i ning qiymati 2 ga teng bo'lsin, u holda  $3+(++i)$  ifoda qiymati 6 ga,  $3+i++$  ifoda qiymati 5 ga teng bo'ladi. Ikkala holda ham i ning qiymati 3 ga teng bo'ladi.

Dekrement -- unar amali qiymatni 1 ga kamaytirishni ko'rsatadi. Bu amal ham prefiks va postfiks ko'rinishda ishlatilishi mumkin. Bu ikki amalni faqat o'zgaruvchilarga qo'llash mumkin.

**Amallar ustivorligi.** Murakkab ifodalarda qaysi amal birinchi navbatda bajarilishi operator prioritetiga bog'liq.

Masalan:  $x = 5+3*8$ .

Ko'paytirish qo'shishga nisbatan yuqoriroq prioritetga ega. Shuning uchun bu ifoda qiymati 29 ga teng bo'ladi.

Agarda ikkita matematik ifodaning prioriteti teng bo'lsa, ular chapdan o'ngga qarab ketma-ket bajariladi.

Masalan:  $x = 5 + 3 + 8 * 9 + 6 * 4$ .

Bu ifodada birinchi ko'paytirish amallari chapdan o'ngga qarab bajariladi  $8 * 9 = 72$  va  $6 * 4 = 24$ . Keyin qo'shish amallari bajariladi. Natijada  $x = 104$  qiymatga ega bo'ladi.

Lekin, barcha amallar ham bu tartibga amal qilmaydi. Masalan, o'zlashtirish amali o'ngdan chapga qarab bajariladi.

Additiv amallarining ustivorligi multiplikativ amallarining ustivorligidan pastrokdir.

Unar amallarning ustivorligi binar amallardan yuqoridir.

**Razryadli amallar.** Razryadli amallar natijasi butun sonlarni ikkilik ko'rinishlarining har bir razryadiga mos mantiqiy amallarni qo'llashdan hosil bo'ladi. Masalan, 5 kodi 101 ga teng va 6 kodi 110 ga teng.

$6 \& 5$  qiymati 4 ga, ya'ni 100 ga teng.

$6 | 5$  qiymati 7 ga, ya'ni 111 ga teng.

$6 \wedge 5$  qiymati 3 ga, ya'ni 011 ga teng.

$\sim 6$  qiymati 4 ga, ya'ni 010 ga teng.

Bu misollarda amallar ustivorligi oshib borishi tartibida berilgandir.

Bu amallardan tashqari  $M \ll N$  chapga razryadli siljitish va  $M \gg N$  o'ngga razryadli siljitish amallari qo'llaniladi. Siljitish  $M$  butun sonning razryadli ko'rinishiga qo'llaniladi.  $N$  nechta pozitsiyaga siljitish kerakligini ko'rsatadi.

Chapga  $N$  pozitsiyaga surish bu operand qiymatini ikkining  $N$  chi darajasiga ko'paytirishga mos keladi. Misol uchun  $5 \ll 2 = 20$ . Bu amalning bitli ko'rinishi:  $101 \ll 2 = 10100$ .

Agar operand musbat bo'lsa,  $N$  pozitsiyaga o'ngga surish chap operandni ikkining  $N$  chi darajasiga bo'lib kasr qismini tashlab yuborishga mosdir. Misol uchun  $5 \gg 2 = 1$ . Bu amalning bitli ko'rinishi  $101 \gg 2 = 001 = 1$ . Agarda operand qiymati manfiy bo'lsa ikki variant mavjuddir: arifmetik siljitishda bo'shatilayotgan razryadlar ishora razryadi qiymati bilan to'ldiriladi, mantiqiy siljitishda bo'shatilayotgan razryadlar nullar bilan to'ldiriladi.



Razryadli surish amallarining ustivorligi o'zaro teng, razryadli inkor amalidan past, qolgan razryadli amallardan yuqoridir. Razryadli inkor amali unar amalga qolgan amallar binar amallarga kiradi.

**Nisbat amallari.** Nisbat amallari qiymatlari 1 ga teng agar nisbat bajarilsa va aksincha 0 ga tengdir. Nisbat amallari arifmetik turdagi operandlarga yoki ko'rsatkichlarga qo'llaniladi.

Misollar:

$1! = 0$  qiymati 1 ga teng;

$1 == 0$  qiymati 0 ga teng;

$3 > = 3$  qiymati 1 ga teng;

$3 > 3$  qiymati 0 ga teng;

$2 < = 2$  qiymati 1 ga teng;

$2 < 2$  qiymati 0 ga teng;

Katta  $>$ , kichik  $<$ , katta yoki teng  $> =$ , kichik yoki teng  $< =$  amallarining ustivorligi bir xildir.

Teng  $=$  va teng emas  $!=$  amallarining ustivorligi o'zaro teng va qolgan amallardan pastdir.

**Mantiqiy amallar.** C tilida mantiqiy tur yo'q. Shuning uchun mantiqiy amallar butun sonlarga qo'llanadi. Bu amallarning natijalari quyidagicha aniqlanadi:

$x||y$  amali 1 ga teng agar  $x>0$  yoki  $y>0$  bo'lsa, aksincha 0 ga teng

$x\&\&y$  amali 1 ga teng agar  $x>0$  va  $y>0$  bo'lsa, aksincha 0 ga teng

$!x$  amali 1 ga teng agar  $x>0$  bo'lsa, aksincha 0 ga teng

Bu misollarda amallar ustivorligi oshib borish tartibida berilgandir.

Inkor  $!$  amali unar qolganlari binar amallardir.

**Qiymat berish amali.** Qiymat berish amali = binar amal bo'lib chap operandi odatda o'zgaruvchi o'ng operandi esa ifodaga teng bo'ladi. Misol uchun

$z = 4.7 + 3.34$

Bu qiymati 8.04 ga teng ifodadir. Bu qiymat Z o'zgaruvchiga ham beriladi.

Bu ifoda oxiriga nuqta vergul (;) belgisi qo'yilganda operatorga aylanadi.

$$z = 4.7 + 3.34$$

Bitta ifodada bir necha qiymat berish amallari qo'llanilishi mumkin. Misol uchun:

$$c = y = f = 4.2 + 2.8;$$

Bundan tashqari C tilida murakkab qiymat berish amali mavjud bo'lib, umumiy ko'rinishi quyidagichadir:

O'zgaruvchi\_nomi **amal** = ifoda;

Bu yerda **amal** quyidagi amallardan biri \*, /, %, +, -, &, ^, |, <<, >>.

Misol uchun:

$x + = 4$  ifoda  $x = x + 4$  ifodaga ekvivalentdir;

$x * = a$  ifoda  $x = x * a$  ifodaga ekvivalentdir;

$x / = a + b$  ifoda  $x = x / (a + b)$  ifodaga ekvivalentdir;

$x >> = 4$  ifoda  $x = x >> 4$  ifodaga ekvivalentdir;

**Imlo belgilari amal sifatida.** C tilida ba'zi bir imlo belgilari ham amal sifatida ishlatilishi mumkin. Bu belgilar oddiy () va kvadrat [] qavslardir. Oddiy qavslar binar amal deb qaralib ifodalarga yoki funksiyaga murojaat qilishda foydalaniladi. Funksiyaga murojaat qilish quyidagi shaklda amlga oshiriladi:

<funksiya nomi> (<argumentlar ro'yxati>). Misol uchun  $\sin(x)$  yoki  $\max(a, b)$ .

Kvadrat qavslardan massivlarga murojaat qilishda foydalaniladi. Bu murojaat quyidagicha amalga oshiriladi:

<massiv nomi>[<indeks>]. Misol uchun  $a[5]$  yoki  $b[n][m]$ .

Vergul simvolini ajratuvchi belgi sifatida ham amal sifatida ham qarash mumkin. Vergul bilan ajratilgan amallar ketma-ketligi bir amal deb qaralib, chapdan o'ngga hisoblanadi va oxirgi ifoda qiymati natija deb qaraladi. Misol uchun:

$$d = 4, d + 2 \text{ amali natijasi } 6 \text{ ga teng.}$$

**Shartli amal.** Shartli amal ternar amal deyiladi va uchta operanddan iborat bo'ladi:

<1-ifoda>?<2-ifoda>:<3-ifoda>

Shartli amal bajarilganda avval 1- ifoda hisoblanadi. Agar 1-ifoda qiymati 0 dan farqli bo'lsa 2- ifoda hisoblanadi va qiymati natija sifatida qabul qilinadi, aks holda 3-ifoda hisoblanadi va qiymati natija sifatida qabul qilinadi.

Misol uchun modulni hisoblash:  $x < 0 ? -x : x$  yoki ikkita son kichigini hisoblash  $a < b ? a : b$ .

Shuni aytish lozimki shartli ifodadan har qanday ifoda sifatida foydalanish mumkin. Agar F FLOAT turpga, a N – INT turga tegishli bo'lsa,

$(N > 0) ? F : N$  ifoda N musbat yoki manfiyligidan qat'iy nazar DOUBLE turiga tegishli bo'ladi.

Shartli ifodada birinchi ifodani qavsga olish shart emas.

### Amallar ustivorligi jadvali

Rang	Amallar	Yo'nalish
1	() [] -> :: .	Chapdan o'ngga
2	! ~ + - ++ -- & * (tur) sizeof new delete tur()	O'ngdan chapga
3	. * ->*	Chapdan o'ngga
4	* / % (multiplikativ binar amallar)	Chapdan o'ngga
5	+ - (additiv binar amallar)	Chapdan o'ngga
6	<< >>	Chapdan o'ngga
7	< < = > = >	Chapdan o'ngga
8	= !=	Chapdan o'ngga
9	&	Chapdan o'ngga
10	^	Chapdan o'ngga
11		Chapdan o'ngga

12	&&	Chapdan o'ngga
13		Chapdan o'ngga
14	?:(shartli amal)	O'ngdan chapga
15	= * = / = % = + = - = & = ^ =   = << = >> =	O'ngdan chapga
16	, (vergul amali)	Chapdan o'ngga

### Nazorat savollari

1. Butun sonli va haqiqiy turlarni qanday farqi bor?
2. Ishorasiz unsigned turining xossalarini ko'rsating.
3. Ishorasiz unsigned short int va long int turlarining o'zaro farqi nimada?
4. Birinchi qaysi funksiya bajariladi?
5. Simvolli kiritish funksiyalari.
6. Shartli amal umumiy ko'rinishi.
7. Turlarni keltirish qoidalari.
8. Quyidagi #include direktivasi qanday vazifani bajaradi.
9. Bosh main() funksiyasining o'ziga xos xususiyati nimadan iborat?
10. Izohlar bir necha qatorda yozilishi mumkinmi?