

## 5- laboratoriya ishi

### Dasturlash tilining peratorlari, massivlar, o'zgaruvchilar, konstanta, prosedura va funksiyalar.

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

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

**char** – bitta belgi;

**long char** – uzun belgi;

**int** – butun son;

**short**– qisqa butun son;

**long**– uzun butun son;

**float** - haqiqiy son;

**double** – ikkilangan haqiqiy son;

**long double** – uzun ikkilangan haqiqiy son;

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 initsializatsiyalash 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 turini kiritish:

***Typedef unsigned char COD;***

***COD simbol;***

## 2. O'zgarmaslar.

**O'zgarmas-** bu dastur ishlashi davomida qiymati o'zgarmaydigan kattalikdir. C++ tilida besh turdagi o'zgarmaslar ishlatilishi mumkin: butun , haqiqiy , belgi, sanovchi o'zgarmaslar va nul kursatkich. Butun sonlar o'nlik, sakkizlik yoki o'n 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 yoki 0X bilan boshlanuvchi 0-9 raqamlari va a-f yoki A-F harflaridan iborat ketma- ketlikdir.

Masalan 15 va 22 o'nlik sonlari sakkizlikda 017 va 026, o'n oltilikda 0xF va 0x16 shaklida tasvirlanadi. Ma'lumotlarning ishorasiz (unsigned) butun son turi:

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:

Oxiriga L yoki l suffikslari qo'yilgan haqiqiy son.

Masalan: 2E+6L;

### Belgili o'zgarmas.

Apostrof ichiga olingan bitta yoki ikkita belgi. Misol uchun 'x', '\*', '\012', '\0', '\n'- bitta belgili o'zgarmas; 'dd', '\n\t', '\x07\x07' ikki belgili o'zgarmaslar. Belgili o'zgarmas qiymati belgining komp'yuterda qabul qilingan sonli kodiga tengdir.

ESC (eskeyp) belgilar jadvali:

Yozilishi	Ichki kodi	Belgii (nomi)	Ma'nosi
\a	0x07	bel (audible bell)	Tovush signali
\b	0x08	Bs (bascpase)	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	\ (bacslash)	Teskari chiziq
\'	0x27	' (single out)	Apostrof
\"	0x22	“ (double quote)	Qo'shtirnoq
\?	0x3F	? (question mark)	Savol belgisi

\000	000	Ëpáîé(octal number)	Belgi sakkizlik kodi
\xhh	0xhh	Ëpáîé(hex number)	Belgi o'n oltilik kodi

### Satrlı o`zgarmas.

Satrlı o`zgarmaslar C++ tili o`zgarmaslariga kirmaydi, balki alohida tur hisoblanadi. Satrlı o`zgarmaslar satrlı leksemalar deb ham ataladi.

Satrlı o`zgarmas bu qo`shtrinoqqa olingan ihtiyorıy belgılar ketma -ketligidir. Misol uchun “ Men satrlı o`zgarmasman”.

Satrlar orasiga eskeyp belgılar ham kirishi mumkin. Bu belgılar oldiga \ belgisi qo`yiladi. Misol uchun : “\n”.

Satr belgilari xotirada ketma-ket joylashtiriladi va har bir satrlı o`zgarmas oxiriga avtomatik ravishda kompilyator tomonidan ‘\0’ belgii qo`shiladi. Shunday satrning xotiradagi xajmi belgılar sonı+1 baytga tengdir.

Ketma-ket kelgan va bo`shlik, tabulyatsiya yoki satr oxiri 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 bo`ysinadi. Misol uchun :

***“O`zbekistonga ”***

***“bahor ”***

***“keldi”***

qatorlari bitta qatorga joylashadi: *“O`zbekistonga bahor keldi”*.

Agar satrda ‘\’ belgisi uchrasa va bu belgidan so`ng to ‘\n’ satr oxiri belgisigacha bo`shlik belgisi kelsa, bu bo`shlik belgilari ‘\’ va ‘\n’ belgisi bilan birga satrdan o`chiriladi. Satrning uzi keyingi satrda kelgan satr bilan qo`shiladi.

***“O`zbekistonga \***

***“ bahor\***

***“ keldi”***

qatorlari bitta qatorga mos: *“O`zbekistonga bahor keldi”*.

### Sanovchi o`zgarmas.

Sanovchi o`zgarmaslar enum hizmatchi so`zi yordamida kiritilib, int turidagi sonlarga qulay so`zlarni mos qo`yish uchun ishlatiladi.

Misol uchun:

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

Agar son qiymatlari ko`rsatilmagan bo`lsa, eng chapki so`zga 0 qiymati berilib qolganlariga tartib bo`yicha o`suvcıi sonlar mos quyiladi:

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

Bu misolda avtomatik ravishda o`zgarmaslar quyidagi qiymatlarni qabul qiladi:

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

O'zgarmlar aralash ko'rinishda kiritilishi ham mumkin:

***Enum(zero,one,for=4,five,six}.***

Bu misolda avtomatik ravishda o'zgarmlar quyidagi qiymatlarni qabul qiladi:

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

Yana bir misol:

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

O'zgarmlar qiymatlari:

***NO=0, YES=1;***

**Nomlangan o'zgarmlar.**

C++ tilida o'zgaruvchilardan tashqari nomlangan o'zgarmlar kiritilishi mumkin.

Bu o'zgarmlar qiymatlarini dasturda o'zgartirish mumkin emas. O'zgarmlar nomlari dasturchi tomonidan kiritilgan va xizmatchi so'zlardan farqli bo'lgan identifikatorlar bo'lishi mumkin. Odatda nom sifatida katta lotin harflari va ostiga chizish belgilari kombinatsiyasidan iborat identifikatorlar ishlatiladi. Nomlangan o'zgarmlar quyidagi shaklda kiritiladi:

***Const tur o'zgarmlar\_nomi=o'zgarmlar\_qiymati.***

Misol uchun:

***Const double EULER=2.718282;***

***Const long M=99999999;***

***Const R=765;***

Oxirgi misolda o'zgarmlar turi ko'rsatilmagan, bu o'zgarmlar int turiga tegishli deb xisoblanadi.

**Nul ko'rsatkich.**

NULL- ko'rsatkich yagona arifmetik bo'lmagan o'zgarmlardir. Null ko'rsatkich 0 yoki 0L yoki nomlangan o'zgarmlar NULL orqali tasvirlanishi mumkin. Quyidagi jadvalda o'zgarmlar chegaralari va mos turlari ko'rsatilgan:

Ma'lumotlar turi	Xajm, bit	Qiymatlar chegarasi	Tur vazifasi
Unsigned char	8	0...255	Kichik butun sonlar va belgilar kodlari
Char	8	-128...127	Kichik butun sonlar va ASII kodlar
Enum	16	-32768...32767	Butun sonlar tartiblangan katori
Unsigned int	16	0...65535	Katta butun sonlar

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

### 3. C++ da amallar

**Arifmetik amallar.** Ko'p dasturlar ijro davomida arifmetik amallarni bajaradi. C++ da amallar quyidagi jadvalda berilgan. Ular ikkita operand bilan ishlatildi.

Amal Arifmetik operator Algebraik ifoda C++dagi ifodasi:

Qo'shish	+	$h+19$	$h+19$
Ayirish	-	$f-u$	$f-u$
Ko'paytirish	*	$s1$	$s*1$
Bo'lish	/	$v/d$	$v/d$
Modul olish	%	$k \bmod 4$	$k\%4$

Bularning ba'zi birlarinig xususiyatlarini ko'rib chiqaylik. Butun sonli bo'lishda, yani bo'luvchi ham, bo'linuvchi ham butun son bo'lganda, javob butun son bo'ladi. Javob yaxlitlanmaydi, kasr qismi tashlanib yuborilib, butun qismining o'zi qoladi.

$x\%y$  ifodasi  $x$  ni  $y$  ga bo'lgandan keyin chiqadigan qoldiqni beradi. Demak,  $7\%4$  ifoda natijasi 3 bo'ladi. % operatori faqat butun sonlar bilan ishlaydi. Vergulli (real) sonlar bilan ishlash uchun "math.h" kutubxonasidagi fmod funksiyasini qo'llash kerak. C++ da qavslarning ma'nosi huddi matematikadagidekdir. Undan tashqari boshqa boshqa algebraik ifodalarning ketma-ketligi ham odatdagidek. Oldin ko'paytirish, bo'lish va modul olish operatorlari bajariladi. Agar bir necha operator ketma-ket kelsa, ular chapdan boshlab o'nga bajariladi. Bu operatorlardan keyin esa qo'shish va ayirish bajariladi. Misol uchun,  $k = m * 5 + 7 \% n / (9 + x)$ ;

Birinchi bo'lib  $m * 5$  xisoblanadi. Keyin  $7 \% n$  topiladi va qoldiq  $(9 + x)$  ga bo'linadi. Chiqqan javob esa  $m * 5$  ning javobiga qo'shiladi.

### Amallar jadvali

Arifmetik amallar	Razryadli amallar	Nisbat amallari	Mantiqiy amallar
+ qo'shish	& va	= = teng	&& va
- bo'lish	yoki	!= teng emas	yoki
* ko'paytirish	^ yoki inkori	> katta	! inkor
/ bo'lish	<< chapga surish	>= katta yoki teng	
% modul olish	>> o'ngga surish	< kichik	
- unar minus	~ inkor	<= kichik yoki teng	
+ unar plyus			
++ dekrement			
-- inkrement			

### Amallar jadvali (davomi)

Imlo amallar	Qiymat berish va shartli amallar	Tipli amallar	Adresli amallar
() – doirali qavs	= - oddiy qiymat berish	(tip) – tipni o'zgartirish	& - manzilni aniqlash
[] – kavadrat qavs	op= - murakkab qiymat berish	sizeof- xajmni hisoblash	* - manzil bo'yi-cha qiymat aniqlash yoki joylash
, - vergul	? – shartli amal		

**Arifmetik amallar.** Amallar odatda unar, ya'ni bitta operandga qo'llaniladigan amallarga va binary, ya'ni ikki operandga qo'llaniladigan amallarga ajratiladi. Binar amallar additiv ya'ni + qo'shuv va – ayirish amallariga , hamda multiplikativ ya'ni \* ko'paytirish, / bo'lish va % modul olish amallariga ajratiladi. Additiv amallarining ustivorligi multiplikativ amallarining ustivorligidan pastroqdir.

Butun sonni butun songa bo'lganda natija butun songacha yahlitlanadi. 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'likdir.

Binar arifmetik amallar bajarilganda turlarni keltirish quyidagi qoidalar asosida amalga oshiriladi: short va char turlari int turiga keltiriladi;

Agar operandlar biri long turiga tegishli bo'lsa, ikkinchi operand ham long turiga keltiriladi va natija ham long turiga tegishli bo'ladi;

Agar operandlar biri float turiga tegishli bo'lsa, ikkinchi operand ham float turiga keltiriladi va natija ham float turiga tegishli bo'ladi;

Agar operandlar biri double turiga tegishli bo'lsa ikkinchi operand ham double turiga keltiriladi va natija ham double turiga tegishli bo'ladi;

Agar operandlar biri long double turiga tegishli bo'lsa ikkinchi operand ham long double turiga keltiriladi va natija ham long double turiga tegishli bo'ladi;

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

++ 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, postfix, ya'ni i++ ko'rinishda ishlatish oldin o'zgaruvchi qiymatidan foydalanib, so'ngra oshirish kerakligini ko'rsatadi. Misol uchun i 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 qiymati 3 ga teng bo'ladi. -- unar amali qiymatni 1 ga kamaytirishni ko'rsatadi. Bu amal ham prefiks va postfix ko'rinishda ishlatilishi mumkin. Bu ikki amalni faqat o'zgaruvchilarga qo'llash mumkindir. 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.

$6^5$  qiymati 3 ga ya'ni 011 ga teng.

Bu misollarda amallar ustivorligi oshib borishi tartibida berilgandir.

Bu amallardan tashqari  $M<<N$  chapga razryadli siljitish va  $M>>N$  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  $5<<2=20$ . Bu amalning bitli ko'rinishi:

**$101<<2=10100$ .**



Agar operand musbat bo'lsa N pozitsiyaga o'ngga surish chap operandni ikkinchi N darajasiga bo'lib, kasr qismini tashlab yuborishga mosdir. Misol uchun  $5 \gg 2 = 1$ . Bu amalning bitli kurinishi  $101 \gg 2 = 001 = 1$ . 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 tuldirladi. Razryadli surish amallarining ustivorligi o'zaro teng, razryadli inkor amalidan past, qolgan razryadli amallardan yuqoridir. Razryadli inkor amali unar qolgan amallar binar amallarga kiradi.

**Munosabat amallari.** Nisbat amallari qiymatlari 1 ga teng agar nisbat bajarilsa va aksincha 0 ga tengdir. Nisbat amallari arifmetik tipdagi 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 hildir. Teng  $==$  va teng emas  $!=$  amallarining ustivorligi uzaro teng va qolgan amallardan pastdir.

**Mantiqiy amallar.** Bu amallarning natijalari quyidagicha aniqlanadi:

$(x, y)$  nuqta koordinatalar tekisligining 1-choragida joylashganligini tekshirmoqchimiz.  $x || y$  amali 1 ga teng agar  $x > 0$  va  $y > 0$  shartning kamida bittasi bajarilsa, aksincha 0 ga teng.  $x \&\&y$  amali 1 ga teng agar  $x > 0$  va  $y > 0$  bo'lsa, aksincha 0 ga teng.  $!x$  amali 0 ga teng agar  $x > 0$  bo'lsa, aksincha 1 ga teng

Bu misollarda amallar ustivorligi oshib borish tartibida berilgandir.

Inkor  $!$  amali unar kolganlari binar amaldir.

Bu amallardan tashqari quyidagi amallar ham mavjuddir:

**Qiymat berish amali.** Qiymat berish amali = binar amal bo'lib, chap operand odatda o'zgaruvchi, o'ng operandi odatda ifoda 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 ++ tili da 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 belgilardan oddiy () va kvadrat [] qavslardir. Oddiy qavslar binary amal deb qaralib, ifodalarda yoki funktsiyaga muroaat qilishda foydalaniladi. Funktsiyaga murojaat qilish quyidagi shaklda amalga oshiriladi:

<funktsiya 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  $b[n][m]$ . Vergulni ajratuvchi belgi deb ham qarash mumkin amal sifatida ham qarash mumkin. Vergul bilan ajratilgan amallar ketma-ketligi bir amal deb qaralib, chapdan o'ngga xisoblanadi va oxirgi ifoda qiymati natija deb qaraladi. Misol uchun:  $d=4, d+2$  amali natijasi 6 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 turga, N – INT turga tegishli bo'lsa,  $(N > 0) ? F : N$  ifoda N musbat yoki manfiylikidan qat'iy nazar DOUBLE turiga tegishli bo'ladi.

Shartli ifodada birinchi ifodani qavsga olish shart emas.

**Turlar bilan ishlovchi amallar.** Turlarni o'zgartirish amali quyidagi ko'rinishga ega: (tur\_nomi) operand; Bu amal operandlar qiymatini ko'rsatilgan turga keltirish uchun ishlatiladi. Operand sifatida o'zgarmas, o'zgaruvchi yoki qavslarga olinga ifoda kelishi mumkin. Misol uchun (long)6 amali o'zgarmas qiymatini o'zgartirmagan holda operativ xotirada egallagan baytlar sonini oshiradi. Bu misolda o'zgarmas turi o'zgarmagan bo'lsa, (double) 6 yoki (float) 6 amali o'zgarmas ichki ko'rinishini ham o'zgartiradi. Katta butun sonlar haqiqiy turga keltirilganda sonning aniqligi yuqolishi mumkin. sizeof amali operand sifatida ko'rsatilgan ob'ektning baytlarda xotiradagi hajmini hisoblash uchun ishlatiladi. Bu amalning ikki ko'rinishi mavjud:

sizeof ifoda sizeof (tur) Misol uchun:

**Sizeof 3.14=8**

*Sizeof 3.14f=4*

*Sizeof 3.14L=10*

*Sizeof(char)=1*

*Sizeof(double)=8.*

### Amallar ustivorligi

Rang	Amallar	Yo'nalish
1	() [] -> :: .	Chapdan o'ngga
2	! ~ + - ++ -- & * (tur) sizeof new delete tur()	O'ngdan chapga
3	. * ->*	Chapdan o'ngga
4	* / % (multurlikativ 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)	Chapdan o'ngga
15	= *= /= %= += -= &= ^=  = <<= >>=	Chapdan o'ngga
16	, (vergul amali)	Chapdan o'ngga

### 4. Dastur tuzilishi.

Dastur preprocessor komandalari va bir necha funktsiyalardan iborat bo'lishi mumkin. Bu funktsiyalar orasida **main** nomli asosiy funktsiya bo'lishi shart.

Quyidagi dastur yozaylik:

```
#include <iostream>
using namespace std;
void main ()
{
    cout << "Salom \n";
}
```

*#Include* buyrug'i kompilyatorga *iostream* faylidagi standart kirish va chiqish oqimlaridan foydalanishni buyuradi. Ushbu tavsiflarsiz, *cout << "Salom \n"* amali bajarilmaydi. *using namespace std*- std nomlar fazosodan foydalanishni bildiradi. Ushbu buyruq berilmasa ma'lumotlarni kiritish va chiqarish operatorlarida *std* so'zini yozishga to'g'ri keladi. *void main ()* –dasturdagi main nomli asosiy funktsiya bo'lib, dasturda albatta yozilishi kerak. Har bir dasturda *main* deb nomlangan funktsiya bo'lishi kerak va dastur shu funktsiya bilan boshlanadi. *void- main* funktsiyasining turini bildirib, funktsiyaning qiymat qaytarmasligini bildiradi. *cout << "Salom \n"* ma'lumotlarni chop etish operatori bo'lib, Salom so'zini chop etadi. *\n* belgisi yangi qatarga o'tishni bildiradi.

C ++ tilida yozilgan dastur natijasini olish uchun uni kompilyatsiya qilish kerak. Dastur matni o'qiladi va tahlil qilinadi, agar biron bir xato aniqlanmasa, dasturda ishlatilgan nomlar va amalar tekshiriladi (bizning holatlarimizda bu *cout* va *<<*). Iloji bo'lsa, dastur kutubxonadagi yetishmayotgan ta'riflarni to'ldiriladi. Bizning holatlarimizda *cout* va *<<* oqimlari *stream.h* da tasvirlangan, ya'ni ularning turlari ko'rsatilgan, ammo ularning amalga oshirilishiga oid tafsilotlar berilmagan. Standart kutubxonada *cout* va *<<* uchun joyni tavsiflash va initsializatsiya kodi mavjud. Aslida, ushbu kutubxonada boshqa ko'p amallar ro'xati mavjud, ularning ba'zilari *stream.h* da tavsiflangan.

Quyidagi dastur sizni uzunlikni dyuymlarda kiritishni so'raydi. Buni amalga oshirgandan so'ng, u dyuymni santimetrda ifodalab chop etadi.

```
#include< iostream>
using namespace std;
int main()
{
    int inch = 0;      // inch - дюйм
    cout << "dyum";
    cin >> inch;
```

```

    cout << inch;
    cout << " in = ";
    cout << inch*2.54;
    cout << " sm\n";
    return 0;
}

```

*main()* funksiyasining birinchi satri *inch* butun o'zgaruvchini aniqlaydi. Uning qiymati *cin* standart ma'lumotlarni kiritish oqimining >> amali yordamida o'qiladi. Dastur ishga tushgandan keyin natija quyidagicha ko'rinishi mumkin:

```

dyum
12
in = 30,48 sm

```

Oxirgi to'rtta chop etish buyrug'ini bitta operator bilan yozilishi mumkin:

```

cout << dyuym << "in =" << dyuym * 2.54 << "sm \ n";

```

**Sharxlar.** Dasturda faqt dasturni isharxlash uchun ishlatiladigan, kompilyator xisoblamayigan matnlar sharx deyiladi. C ++ da sharx buni ikkita usulda yoziladi. Birinchi usulda sharx / \* belgilari bilan boshlanib, \* / belgilari bilan tugaydi. Sharxning by turi ko'p satrli sharhlar uchun ishlatiladi. // belgilar bitta satrli sharxlar uchun ishlatiladi.

Har bir nom va har bir ifoda ularda bajarilishi mumkin bo'lgan operatsiyalarni belgilaydigan turga ega. Masalan,

```

int dyuym;

```

Bu yerda *dyuym* turi *int* ekanligini, ya'ni *dyuym* butun qiymatli o'zgaruvchi ekanligini aniqlaydi. Tavsif - bu nomni dasturga kirituvchi operatoridir. Tavsif ushbu nomning turini belgilaydi. Tur nom yoki ifodaning to'g'ri ishlatilishini belgilaydi. Butun sonlar uchun +, -, \* va / kabi amallar aniqlangan. Asosiy turlari quyidagilardir:

***byte , short, int, long, float, double, boolean, char***

Dastlabki to'rtta tur butun sonlarni ifodalash uchun ishlatiladi, keyingii ikkitasi suzuvchi nuqtali haqiqiy sonlarini ifodalash uchun ishlatiladi. Char tipi bitta belgini ifodalash uchun ishlatiladi, boolean esa mantiqiy turni aniqlaydi. Tur bilan ifodalanishi mumkin bo'lgan butun sonlarning diapazoni uning o'lchamiga bog'liq. Asosiy turlarning o'zaro bog'liqligi quyidagicha yozilishi mumkin:

```

1 = sizeof(char) <= sizeof(short) <= sizeof(int) <= sizeof(long)

```

*sizeof(float) <= sizeof(double)*

Apostrof ichida joylashgan belgi belgili kattalikdir.

Arifmetik amallar ushbu turlarning har qanday kombinatsiyasiga qo'llanilishi mumkin:

*+ (qo'shish, unar va binar)*

*- (ayirish, unar va binar)*

*\* (ko'paytirish)*

*/ (bo'lish)*

Shuningdek taqqoslash amallari:

*== (teng)*

*!= (teng emas)*

*< (kichik)*

*> (katta)*

*<= (kichik yoki teng)*

*>= (katta yoki teng)*

Butun sonli bo'linma butun natijani beradi:  $7/2 = 3$ . Butun sonlar ustida qoldiqni xisoblash % amalini bajarish mumkin:  $7\% 2 = 1$ .

C ++ da juda ko'p amallar mavjud. Masalan:

*~ (qo'shimcha)*

*& (Va)*

*^ (eksklyuziv yoki)*

*| (yoki)*

*<< (chapga siljish)*

*>> (o'ngga siljish)*

Ushbu amalar butun sonlarga nisbatan qo'llaniladi.

Amalning natijasi operandlar soniga bog'liq; & unar amali- manzlni qabul qilish operatsiyasi; & binar amali- mantiqiy qo'shishni bildiradi. Amalning natijasi operandlar turiga ham bog'liq:  $a + b$  ifodasida + amali, agar operandlar turi float bo'lsa, amal natijasi haqiqiy son bo'ladi, agar ular butun turga tegishli bo'lsa, amal natijasi butun son bo'ladi. C ++ da o'zlashtirish operatori mavjud =. Masalan,  $a = b = c$  ifodada c ning qiymatini b o'zlashtiradi, so'ng a o'zlashtiradi.  $x * = 4$  amali  $x = x * 4$  amali bilan teng kuchli.

Ko'pincha C ++ dasturi ko'rsatgichlardan keng foydalanadi. \* unar amali bo'lib, \* p - p bilan ko'rsatilgan ob'ektni bildiradi. Ushbu amal bilvosita adreslash deb ham

ataladi. Masalan, char \* p bo'lsa, u holda p- p bilan ko'rsatilgan belgidir. Ko'pincha ko'rsatkichlar bilan ishlashda ++ va -- amallari foydalidir.

C ++ da ba'zi bir keng tarqalgan matematik funktsiyalar <cmath> faylida aniqlanadi. Masalan, ildizni topish, darajaga ko'tarish, sin (), cos () va boshqalar.

1-jadval - C ++ tilidagi matematik funktsiyalar

Функция	Tavsifi	Misol
<b>abs( a )</b>	<b>a</b> ning moduli yoki absolyut qiymati	abs(-3.0)=3.0 abs(5.0)= 5.0
<b>sqrt(a)</b>	nomanfiy <b>a</b> kattalikdan olingan kvadrat ildiz	sqrt(9.0)=3.0
<b>pow(a, b)</b>	<b>a</b> ni <b>b</b> darajaga oshirish	pow(2,3)=8
<b>ceil( a )</b>	<b>a</b> ni <b>a</b> dan kichik bo'lmagan eng kichik butun songacha yaxlitlash	ceil(2.3)=3.0 ceil(-2.3)=-2.0
<b>floor(a)</b>	<b>a</b> ni <b>a</b> dan katta bo'lmagan eng katta butun songacha yaxlitlash	floor(12.4)=12 floor(-2.9)=-3
<b>fmod(a, b)</b>	a/b ifoda qoldig'ini xisoblash	fmod(4.4, 7.5) = 4.4 fmod( 7.5, 4.4) = 3.1
<b>exp(a)</b>	$e^a$ exponentani xisoblash	exp(0)=1
<b>sin(a)</b>	<b>a</b> ning sinusini radianlarda xisoblash	
<b>cos(a)</b>	<b>a</b> ning kosinusini radianlarda xisoblash	
<b>log(a)</b>	<b>a</b> ning natural logarifmini xisoblash	log(1.0)=0.0
<b>log10(a)</b>	<b>a</b> ning o'nli logarifmini xisoblash	Log10(10)=1
<b>asin(a)</b>	<b>a</b> ning arksinusini xisoblash, $-1.0 < a < 1.0$	asin(1)=1.5708
<b>acos(a)</b>	<b>a</b> ning arkkosinusini xisoblash, $-1.0 < a < 1.0$	
<b>atan(a)</b>	<b>a</b> ning arktangensini xisoblash	
<b>sinh(a)</b>	<b>a</b> ning giperboliksinusini xisoblash	
<b>cosh(a)</b>	<b>a</b> ning giperbolikkosinusini xisoblash	
<b>tan(a)</b>	<b>a</b> ning tangensini xisoblash	
<b>tanh(a)</b>	<b>a</b> ning giperboliktangensini xisoblash	

Ushbu funktsiyalarning operandalari har doim haqiqiy qiymatni qabul qilishi kerak.

Chiziqli algoritmni dasturlashga misol keltiramiz.

1-misol.  $Z = \frac{\cos^2(3x + a)}{\operatorname{tg}(bx^2 + a)}$  ifodaning qiymatini hisoblang, bu yerda  $a = -3,15$ ;  $b = 4,33$ ;  $x$  - ixtiyoriy son.

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{ double a,b,x,z;
  cout<< "a,b,x larning qiymatlarini kiriting:\n";
  cin>>a>>b>>x;
  z=pow(cos(3*x+a),2)/tan(b*x*x+a);
  cout <<"z="<<z;
  return 0;
}
```