

**O‘ZBEKISTON RESPUBLIKASI OLIY VA O‘RTA MAXSUS  
TA‘LIM VAZIRLIGI**

**SAMARQAND DAVLAT UNIVERSITETI**

**S.Eshtemirov, F.M.Nazarov**

# **ALGORITMLASH VA DASTURLASH ASOSLARI**



SAMARQAND - 2018

UDK: 681.14(075)  
BBK 32.97

***Algoritmlash va dasturlash asoslari. O‘quv qo‘llanma*** –SamDU, 2018  
–yil, **252**- bet.

*Mazkur qo‘llanma algoritmlar nazariyasi hamda dasturlash asoslari fanini C++ dasturlash tili misolida qarab o‘tilgan. Algoritm nazariyasi, algoritm xossalari, berilish usullari, algoritm turlari keng yoritib o‘tilgan. C++ dasturlash tili tarkibidagi barcha turdagi operatorlar, kalit so‘zlar va qoidalar batafsil keltirib o‘tilgan. Operatorlar, kalit so‘zlar va buyruqlarning umumiy ko‘rinishlari va yozilishlari bir nechta misollar yordamida tushuntirib o‘tilgan. Qo‘llanmada C++ tilining barcha imkoniyatlari bo‘yicha nazariy tushunchalar hamda bu tushunchalarni o‘zlashtirish uchun masalalar yechimlari keltirib o‘tilgan. Har bir mavzu bo‘yicha mavzuni mustahkamlash uchun nazariy savollar hamda mustaqil ishlash uchun topshiriqlar ham keltirib o‘tilgan. Qo‘llanma asosan talabalar va mustaqil o‘rganuvchilar uchun qulay vosita hisoblanadi, chunki har bir mavzu uchun nazariy tushunchalar hamda misollar yechimi keng ravishda yoritib o‘tilgan. Qo‘llanmada algoritmlar uning hossalari, berilish usullari, turlari, C++ dasturlash tili tarkibida boshlang‘ich tushunchalar, standart funksiyalar, chiziqli, tarmoqlanuvchi, takrorlanuvchi jarayonlar, massivlar, satrlar, funksiya proseduralar, fayllar, strukturalar hamda grafik imkoniyatlar keltirib o‘tilgan. Mazkur qo‘llanma oliy o‘quv yurtlari talabalari va magistrantlari, litsey kasb hunar kollej o‘quvchilari hamda mustaqil o‘rganuvchilar uchun qulay vosita hisoblanadi.*

**Mualliflar:**

**Eshtemirov Sodiq**

*Samarqand Davlat Universiteti  
“Axborotlashtirish texnologiyalari “  
kafedrasi dotsenti.*

**Nazarov Fayzullo Maxmadiyarovich**

*Samarqand Davlat Universiteti  
“Axborotlashtirish texnologiyalari “  
kafedrasi o‘qituvchisi.*

**Ma’sul muharrir:**

*Samarqand Davlat Universiteti* **t.f.d. A.Axatov**

**Taqrizchilar:**

TATU Samarqand filiali Dasturiy injiniring  
kafedrasi mudiri **dots.A.Qarshiyev**

SamDU Axborotlashtirish texnologiyalari  
kafedrasi dotsenti **t.f.n. A.Abdullayev**

ISBN 978-9943-

©Samarqand Davlat Universiteti, 2018

## MUNDARIJA

Kirish.....	7
<b>I-BOB. Algoritm, berilish usullari, hossalari va ularni turlari.....</b>	<b>10</b>
<b>1.1 Algoritm, algoritmning berilish usullari va xossalari.....</b>	<b>10</b>
Algoritm.....	10
Algoritmning tasvirlash usullari .....	10
Algoritmning hossalari.....	12
<b>1.2 Algoritm va uning turlari.....</b>	<b>14</b>
Chiziqli algoritmlar.....	14
Tarmoqlanuvchi algoritmlar.....	18
Takrorlanuvchi algoritmlar.....	21
Ichma ich sikllar.....	
<b>II-BOB. C++ dasturlash tili, C++ dasturlash tilida turlar va operatorlar.....</b>	<b>26</b>
<b>2.1 C++ dasturlash tili tarixi va uning tarkibi.....</b>	<b>26</b>
C++ tili tarixi.....	26
C++ tilining leksemmasi.....	28
C++ dasturlash tilidagi dasturning umumiy ko‘rinishi.....	29
C++ dasturlash tilida izohlardan foydalanish.....	31
<b>2.2 C++ dasturlash tili tarkibidagi o‘zgaruvchi turlari va o‘zgarmlar.....</b>	<b>33</b>
Identifikatorlar va kalit so‘zlar.....	33
C++ tilida o‘zgaruvchi va uning turlari.....	34
C++ tilida o‘zgaruvchi va o‘zgarmlarni e’lon qilish.....	36
Turni boshqa turga keltirish	37
<b>2.3 C++ dasturlash tili tarkibidagi arifmetik amallar va mantiqiy amallar.....</b>	<b>40</b>
Arifmetik amallari.....	40
Ta’minlash operatori.....	42
Mantiqiy amallar.....	42
<b>2.4 C++ dasturlash tili tarkibidagi standart funksiyalar va ifodalar...</b>	<b>47</b>
C++ tilida ifodalar.....	47
C++ dasturlash tilida standart funksiyalar.....	48
<b>III- BOB. C++ tilida chiziqli, tarmoqlanuvchi va takrorlanuvchi jarayonlarni dasturlash.....</b>	<b>53</b>
<b>3.1 C++ tilida qiymat berish, kiritish va chiqarish operatorlari.....</b>	<b>53</b>
C++ tilida qiymat berish operatori.....	53

Qiymat berish operatoriningni kengaytirilgan holatlari.....	54
Kiritish operatori.....	56
Chiqarish operatori.....	57
<b>3.2 C++ dasturlash tilida chiziqli jarayonlarni dasturlash.....</b>	<b>63</b>
C++ tilida chiziqli dasturlar.....	63
C++ tilida chiziqli dasturlarga oid misollar.....	64
<b>3.3 C++ dasturlash tilida tarmoqlanuvchi jarayonlarni dasturlash...</b>	<b>69</b>
Qisqa shartli operator .....	70
To‘liq shartli operator .....	71
?: operatori .....	74
<b>3.4 C++ dasturlash tilida tanlash jarayonlari va ularni dasturlash...</b>	<b>78</b>
Break operatori .....	78
Continue operatori .....	79
Tanlash operatori .....	81
<b>3.5 C++ dasturlash tilida takrorlanuvchi jarayonlar va parametr bo‘yicha dasturlash.....</b>	<b>87</b>
Takrorlanuvchi jarayonlarni dasturlash.....	87
for(sikl) operatori .....	89
Shartsiz o‘tish operatori .....	92
<b>3.6 C++ dasturlash tilida shartli takrorlanuvchi jarayonlar dasturlash</b>	<b>98</b>
While operatori .....	98
do while operatori .....	102
Ichma-ich joylashgan sikllar.....	104
<b>IV-BOB. Murakkab turlar va ularning C++ dasturlash tilida foydalanish.....</b>	<b>112</b>
<b>4.1 C++ tilida massivlar va ulardan foydalanish.....</b>	<b>112</b>
Bir o‘lchovli massivlar .....	113
Rand funksiyasi .....	118
Ikki o‘lchovli massivlar .....	119
<b>4.2 C++ dasturlash tilida massiv elementlarini saralash.....</b>	<b>126</b>
Bir o‘lchovli massiv elementlarini tartiblash.....	127
Ikki o‘lchovli massiv elementlarini tartiblash.....	131
<b>4.3 C++ dasturlash tilida massiv elementlari ichidan eng kattasi va kichigini aniqlash.....</b>	<b>137</b>
Bir o‘lchovli massiv elementlari ichidan eng kattasi va kichigini aniqlash.....	137
Ikki o‘lchovli massiv elementlari ichidan eng kattasi va kichigini	

aniqlash.....	138
<b>4.4 C++ dasturlash tilida funksiyalarni yaratish va ulardan foydalanish.....</b>	<b>142</b>
Qism dasturlar.....	142
Funksiya va ularni e’lon qilish.....	145
Funksiya tanasini tasvirlash.....	146
Funksiyani e’lon qilmasdan undan foydalanish.....	150
Global va lokal o’zgaruvchilar.....	152
Global o’zgaruvchilar qiymatini saqlab qolish.....	154
Funksiya qiymatini qaytarish haqida.....	155
<b>4.5 C++ dasturlash tilida bir nechta qiymat qaytaruvchi qism dasturlarni yaratish va ulardan foydalanish.....</b>	<b>163</b>
Prosedura va ularni e’lon qilish.....	164
Prosedura asosiy tanasini tasvirlash va ulardan foydalanish.....	165
<b>4.6 C++ dasturlash tilida ko’rsatkichlar va adres oluvchi o’zgaruvchilar.....</b>	<b>170</b>
Funksiyaga ko’rsatkich.....	171
Obyektga o’zgaruvchiga ko’rsatkich.....	172
Ko’rsatkichlarga boshlang’ich qiymat berish.....	172
Ko’rsatkichlar ustida bajariladigan amallar.....	173
Adresdagi qiymatlar ustida amallarni bajarish.....	175
<b>V-BOB. C++ dasturlash tilida satr, fayl va struktura turidagi ma’lumotlar va ulardan foydalanish.....</b>	<b>180</b>
<b>5.1 C++ dasturlash tilida satr turi va ulardan foydalanish usullari....</b>	<b>180</b>
Belgi va satrlar.....	181
Satr uzunligini aniqlash.....	182
Satrlarni nusxalash.....	185
Satrlarni ulash.....	187
Satrlarni solishtirish (taqqoslash) funksiyalari.....	189
Satrdagi harflar registrini almashtirish funksiyalari.....	191
Satrlarni teskari tartiblash.....	192
Satrdan belgini izlash.....	192
Satr ichidan satr qismini izlash.....	193
Satr turini o’zgartiruvchi funksiyalar.....	194
<b>5.2 C++ dasturlash tilida string turiga mansub bo’lgan satrli o’zgaruvchilar.....</b>	<b>199</b>
String turidagi o’zgaruvchilarni e’lon qilish.....	200

String turdagi satrlar ustida amallar bajarish.....	201
String turidagi satrlarni nusxalash.....	202
Satrning biror qismidan nusxa olish funksiyasi.....	202
Satrning oxiriga boshqa satrni o'rnatish.....	203
Satr tarkibiga boshqa satrni joylashtirish.....	204
Satr qismini o'chirish.....	205
Matn qismini boshqa matn qismi bilan almashtirish funksiyasi.....	206
Satr tarkibidan satr qismini izlash.....	206
Satrlarni solishtirish funksiyalari.....	207
Satr xossalarini aniqlash funksiyalari.....	210
<b>5.3 C++ dasturlash tilida fayllar va ulardan foydalanish.....</b>	<b>213</b>
Fayllarni e'lon qilish.....	213
Fayllarga murojat qilish.....	215
Fayllar ustida amallar bajarish.....	216
Fayldan ma'lumot o'qish.....	218
Fayl tarkibidagi ma'lumotlarni o'chirish.....	220
<b>5.4 C++ dasturlash tilida struktura va ulardan foydalanish.....</b>	<b>224</b>
Struktura va ularni e'lon qilish.....	225
Strukturalarga murojat qilish va ulardan foydalanish.....	227
Struktura massivi.....	229
<b>VI-BOB. C++ dasturlash tilining grafik imkoniyatlari.....</b>	<b>234</b>
<b>6.1 C++ dasturlash tili tarkibida grafik muhitni yaratish va undan foydalanish.....</b>	<b>234</b>
Grafik muhitini yaratish.....	235
Tekislikda chizma va shakllar chizish.....	237
Tozalash va kutib turish funksiyalari.....	242
Grafik rejimning maksimal koordinatalari va izohli ma'lumotlar hosil qilish funksiyalari.....	243
Chiziq shaklini o'zgartirish funksiyasi.....	244
Tekislikda ichi bo'yalgan to'rtburchaklar hosil qilish funksiyasi..	244
<b>6.2 C++tilining grafik muhitida fazoviy jismlarni hosil qilish.....</b>	<b>247</b>
Fazodagi shakllarni hosil qilish.....	247
<b>Foydalanilgan adabiyotlar ro'yxati.....</b>	<b>252</b>

## Kirish

Jamiyatda tobora o‘sib borayotgan axborot oqimi, axborot texnologiyalarining turli tumanligi, kompyuterda yechiladigan masalalarning murakkablashuvi ushbu texnologiyalardan foydalanuvchilarning oldiga bir qator vazifalarni qo‘ydi. Bugungi kunda ma’lumotlar oqimining ko‘pligi tufayli ularni qisqa vaqt ichida qayta ishlash muommosi ham ortib bormoqda. Hozirgi vaqtda axborot-kommunikasiya vositalari barcha turdagi tashkilot va muassasalarga shiddat bilan kirib kelmoqda. Axborotlarning haddan tashqari ko‘pligi bu axborotlarni saqlashda, qayta ishlashda, hamda har xil turdagi tizimlarni yaratish, ulardan keng foydalanishni va axborot tizimlari yaratishni talab qiladi. O‘zbekiston Respublikasi Prezidentining 2017 yil 7 fevraldagi PF-4947-son Farmoni bilan tasdiqlangan 2017-2021 yillarda O‘zbekiston Respublikasini rivojlantirishning beshta ustuvor yo‘nalishi bo‘yicha harakatlar strategiyasi mamlakatning davlat va jamiyat rivojlanishi istiqbolini strategik rejalashtirish tizimiga sifat jihatdan yangi yondashuvlarni boshlab berdi[1]. Unda belgilangan vazifalar sirasida ta’lim va fan sohasini rivojlantirish ham alohida ko‘zda tutilgan. O‘zbekiston Respublikasi birinchi Prezidentining 2012 yil 21 martdagi “Zamonaviy axborot-kommunikasiya texnologiyalarini yanada joriy etish va rivojlantirish chora-tadbirlari to‘g‘risida”gi PQ-1730 Qarori hamda “O‘zbekiston Respublikasida “Elektron ta’lim milliy tarmog‘ini yaratish” investision loyihasini amalga oshirish chora-tadbirlari to‘g‘risida” gi PQ-1740 Qarori va me‘yoriy hujjatlar asosida algoritmik ta‘minot ishlab chiqish va joriy etish keng ko‘lamli hisoblanadi. Barcha tashkilot va muassasalarda avtomatlashtirilgan tizimlar yaratish ulardan keng ko‘lamda foydalanish uchun algoritmlash tillarini o‘rni katta hisoblanadi.

Axborot tizimlari axborotni to‘plash, saqlash va qayta ishlash uchun, keng imkoniyatli maqsadlarda samarali foydalanish uchun xizmat qiladi. Zamonaviy axborotlashtirish tizimi, ma’lumotlar integratsiyasi konsepsiyasiga asoslangan katta hajmdagi ma’lumotlarni saqlash bilan tavsiflanadi va ko‘p sondagi foydalanuvchilarning turli xildagi talablariga javob berishi kerak bo‘ladi.

Axborot tizimi va axborot texnologiyalarining avtomatlashtirilgan elementlarini qo‘llash va avtomatlashtirish asosida yangi axborot texnologiyasini yaratish avtomatlashtirish tizimlarini loyihalashtiruvchilarning asosiy vazifalaridan biri hisoblanadi. Avtomatlashtirilgan tizimlarni yaratish uchun albatta birinchi navbatda muommo obektini infologik yoki diskretli modelni qurish dolzarb hisoblanadi. Infologik yoki diskretli modelni muommo obektiga qarab algoritmlash tillarini qaysi biri asosida yaratish kerakligini tanlab olinish kerak. Elektron hisoblash mashinalarini birinchi avlodlari yaratilishi bilan algoritmlash

tillarini rivojlanishi ham boshlandi. Avval algoritm tuzuvchi mutaxassislar eng sodda mashina tilini o'zida ifodalovchi kompyuter komandalari bilan ishlaganlar.

Bu komandalar nol va birlardan iborat uzun qatorlardan tashkil topgan edi. Keyinchalik insonlar uchun tushunarli bo'lgan mashina komandalarini o'zida saqlovchi assembler tili yaratildi. Shu vaqtlarda BASIC va COBOL singari yuqori sathli tillar ham paydo bo'ldiki, bu tillar yordamida so'z va gaplarning mantiqiy konstruksiyasidan foydalanib algoritmlash imkoniyati yaratildi. Ular komandalarni mashina tiliga interpretatorlar va kompilyatorlar yordamida o'tkazar edi.

Algoritmlash tillari yaratilishi bo'yicha uchta turga ajratiladi:

-quyi darajadagi;

-o'rta darajadagi;

-yo'qori darajadagi.

Bizga ma'lumki, ma'lum bir masalani yechish uchun buyruqlar ketma- ketligi ya'ni algoritmni algoritmlash tilida yozilishida kamroq komandalardan foydalanilsa, bunday tillar darajasi yuqoriroq hisoblanadi.

Quyi darajadagi algoritmlash tillari bevosita kompyuter qurilmalari bilan bog'liq bo'lib buyruqlar ularning kodlari bilan yoziladi. Bu kabi buyruqlardan tashkil topgan algoritmlar katta hajmli bo'lib, ularni taxrirlash mushkul hisoblanadi. Dastlabki kompyuterlar(*ENIAC*, *MESM* va boshqalar) ana shunday tillarda ishlardi.

O'rta darajadagi algoritmlash tillari buyruqlarida faqat raqamlar emas, balki insonlar tushunadigan bazi so'zlar ishlatila boshlandi(*Assemblaer*).

Yuqori darajadagi algoritmlash tillari quyidagicha bosqichlarga bo'linadi:

*Algoritmik*(Basic, Pascal, C va b.)

*Mantiqiy*(Prolog, Lisp va b.)

*Obe'ktga mo'ljallangan*(Object Pascal, C++, Java va b.)

Algoritmlash tillarida yaratilgan algoritmlar mashina tiliga *Translyatorlar* yordamida o'tkaziladi.

Translyator(translator-tarjimon) biror bir algoritmlash tilida yozilgan algoritmni mashina tiliga tarjima qiladi.

Translyatorlar ikki turda bo'ladi:

-Kompilyatorlar(*compiler*-yig'uvchi) biror bir algoritmlash tilida yozilgan algoritmni mashina tiliga to'liq o'qib olib tarjima qiladi.

-Interpretatorlar(*interpreter* —izohlovchi, og'zaki tarjimon) biror bir algoritmlash tilida yozilgan algoritmni mashina tiliga satrma satr tarjima qiladi.



Translyatorlarni bu ikkala turi bir biridan farq qiladi. Komplyatsiya qilingan algoritmlar bir muncha kam vaqt talab etadi, ya'ni tezroq ishlaydi, lekin interpretatsiya qilingan algoritmlarni o'zgartirish osonroq hisoblanadi.

C++ dasturlash tili tarkibida bir nechta imkoniyatlar mavjud, ya'ni consol rejimi, forma ob'yekt rejimi, grafik muhiti va ma'lumotlar bazasi bilan ishlash imkoniyatlari keng joriy etilgan. Ushbu qo'llanmada keltirilgan misol va masalalarning yechimi dasturining intmain funksiyasi tarkibini C++ dasturlash tilining ixtiyoriy versiyalarida ishlatib ko'rish mumkin. Qo'llanma oliy o'quv yurtlari talabalari va magistrantlari, litsey kasb hunar kollej o'quvchilari hamda mustaqil o'rganuvchilar uchun qulay vosita hisoblanadi.

## **I-BOB. ALGORITM, ALGORITMNING BERILISH USULLARI, HOSSALARI VA ULARNI TURLARI**

### **1.1 ALGORITM, ALGORITMNING BERILISH USULLARI VA HOSSALARI**

#### ***Reja:***

- 1.Algoritm;*
- 2. Algoritmning tasvirlash usullari ;*
- 3.Algoritmning hossalari;*
- 4. Qiymat berish buyrug'i.*

Biror bir masalani elektron hisoblash mashinasida hisoblash uchun, albatta, berilgan masalani matematik modeli, algoritmi va biror bir algoritmlash tili asosida algoritmini tuzish kerak bo'ladi. Masalani matematik modelini tuzish deganda berilgan masalani qandaydir matematik tushunchalar bilan ifodalanishiga aytiladi. Algoritm so'zi va tushunchasi IX asrda yashab ijod etgan buyuk vatandoshimiz alloma Muhammad al-Xorazmiy nomi bilan uzviy bog'liq. Algoritm so'zi Al-Xorazmiy nomini Yevropa olimlari tomonidan o'zgacha talaffuz qilinishidan yuzaga kelgan. Al-Xorazmiy birinchi bo'lib o'nlik sanoq sistemasining tamoyillarini va undagi to'rtta amallarni bajarish qoidalarini asoslab bergan.

Masalani yechish uchun yuqorida keltirib o'tilgandek uni algoritmini tuzish kerak bo'ladi. Algoritm tuzish matematik model va algoritm tuzishdan bir muncha murakkab hisoblanadi.

Algoritm aslida biror masalani yechish yoki biror ishni bajarish rejasi bo'lib, rejadan farqi u aniq tugallangan qadamlarga bo'linadi va buyruq ohangida yoziladi.

***Misol 1:*** Kitobdan kerakli mavzuni topish algoritmini tuzing

- 1.kitobning mundarijasi yozilgan sahifasi ochilsin;
- 2.mundarijadan izlanayotgan mavzu topilsin;
- 3.mavzu yozilgan sahifa aniqlansin;
- 4.aniqlangan sahifa ochilsin.

***Misol 2:*** Uchburchak chizish algoritmini tuzing.

- 1.qog'ozda uchta bir to'g'ri chiziqda yotmagan uchta nuqta belgilansin;
- 2.belgilangan nuqtalar chizg'ich yordamida bir birlari bilan tutashtirilsin;
- 3.hosil bo'lgan figura uchburchak deb qabul qilinsin.

***Misol 3:*** Choy damlash algoritmini tuzing.

- 1.suv qaynatiladigan choynakga kerakli miqdorda suv solinsin;
- 2.gaz o'chog'ida olov yoqilsin;
- 3.suv solingan choynak olovga qo'yilsin;

- 4.suv qaynaguncha kutilsin;
- 5.chinni choynakga bir choy qoshiq quruq choy solinsin;
- 6.chinni choynak qaynagan suv bilan to'ldirilsin.

Ko'rinib turibdiki, algoritm aniq tugallangan qadamlarga bo'lingan bo'lib, bu qadamlarning har biri algoritmning buyruqlari deyiladi.

Biz kelajakda algoritmlarni ifodalashning(yozishning) uch xil usuli bilan tanishamiz:

- 1.matn usuli;
- 2.blok-sxema usuli;
- 3.dastur ko'rinishida ifodalash usuli.

Yuqorida yozilgan barcha algoritmlar matn usulida yozildi. Demak, algoritmlarni matn usulida yozish uchun uning har bir buyrug'i nomerlanadi va nomeridan keyin qavs qo'yiladi hamda buyruqlar kichik harflar bilan yoziladi, buyruq oxirida (;) qo'yiladi. Faqat oxirgi buyruq tugagach (.) qo'yiladi.

Matematik ifodalarning qiymatlarini hisoblash algoritmlarini ham matn usulida ifodalash mumkin. Misol 4.  $y=3*x^2 -7*x+28$  funksiya qiymatini hisoblash algoritmini tuzing.

- 1)x ning qiymati kiritilsin;
- 2)3 ni  $x^2$  ga kopaytirib, natija  $a_1$  bilan belgilansin;
- 3)-7 ni x ga kopaytirib, natija  $a_2$  bilan belgilansin;
- 4) $a_2$  ga 28 qo'shib, natija  $a_3$  bilan belgilansin;
- 5) $a_1$  va  $a_3$  larning yig'indisi funksiyaning qiymati deb qabul qilinsin.

Ushbu algoritmnı quyidagi matn ko'rinishida ham ifodalash mumkin:

x ning qiymati kiritilsin;

- 1) $a_1:=3*x^2$ ;
- 2) $a_2:=-7*x$ ;
- 3) $a_3:=a_2+28$ ;
- 4) $y:=a_1+a_3$ .

Algoritmning har bir qadami uning buyrug'i deyiladi . Bu algoritmda 5 ta buyruq bor. Algoritmlarni ifodalashning(yozishning) bu usulini ba'zi adabiyotlarda algoritmlarni ifodalashning formula usuli ham deyishadi.

Algoritm tuzuvchi shu algoritmning **muallifi**, uni ijro etuvchi esa uning **ijrochisi** deyiladi. Xususiyl holda muallifning o'zi ham algoritm ijrochisi bo'lishi mumkin. Algoritm ijrochisi "Algoritm ijrochisi algoritm buyruqlari va ko'rsatmalarini bajara oladigan abstrakt va real (texnik yoki biologik) sistema" deb ta'riflanadi. Algoritm shunday tuzilgan bo'lishi kerakki ijrochi uni hech qanday

fikrlamasdan, ko'r-ko'rona ijro etganda maqsadga erishsin. Algoritmning bu hossasi formal ijro etish hossasi deyiladi.

**Tarif:** *Algoritm- bu biror bir masala yoki muommoni hal etish uchun ijrochi tomonidan beriladigan chekli sondagi buyruqlar ketma- ketligi.*

Masala algoritmini tuzish tartibini quyidagi masala orqali ko'rib chiqamiz.

**Misol:**  $(x_1; y_1)$  va  $(x_2; y_2)$  nuqtalar orasidagi masofani hisoblash algoritmi tuzilsin

1.  $(x_1; y_1)$  va  $(x_2; y_2)$  nuqtalar kiritilsin;
2.  $T = ((x_2 - x_1)^2 + (y_2 - y_1)^2)$  hisoblansin;
3.  $\sqrt{T}$  chiqarilsin;

### Algoritmning tasvirlash usullari

Algoritm tuzish vaqtida masala qanday sohaga tegishliligiga qarab algoritm bir necha xil ko'rinishda tasvirlanishi mumkin. Algoritmni bir necha tasvirlash usullari mavjud. Iqtisodiy masalalar algoritmi jadvallar ko'rinishida, matematik masalalar formula yoki grafik usullarda va sanoatda esa so'zlar yordamida tasvirlanish mumkin.

**1. Algoritmning so'zlar orqali ifodalanishi.** Bu usulda ijrochi uchun beriladigan har bir ko'rsatma jumlar, so'zlar orqali buyruq shaklida beriladi.

**Misol:** Chiziqli tenglamani hisoblash algoritmi tuzilsin.

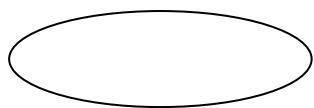
1. boshlanish
2.  $a, b$  lar kiritilsin;
3.  $x = (b/a)$  hisoblansin;
4.  $x$  chiqarilsin;
5. Tamom.

**Misol:** To'g'ri turtburchak ikki tomoni berilganda yuzi va perimetri hisoblansin.

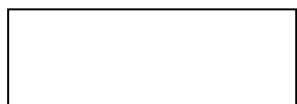
1. boshlanish
2.  $a, b$  lar kiritilsin;
3.  $s = ab$ , va  $p = 2(a + b)$  hisoblansin;
4.  $S$  va  $P$  chiqarilsin;
5. Tamom.

**2. Algoritmning grafik shaklida tasvirlanishida** algoritm maxsus geometrik figuralar yordamida tasvirlanadi va bu grafik ko'rinishi blok-sxema deyiladi. Algoritmning blok-sxema ko'rinishida tasvirlanishi beriladigan buyruqlar qandaydir shakllar orqali ifodalanadi.

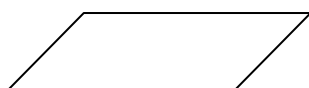
Algoritmning blok-sxemalar yordamida tuzishda foydalaniladigan asosiy sodda geometrik figuralar quyidagilardan iborat.



Ellips u algoritmning boshlanishi yoki tugallashini belgilaydi.



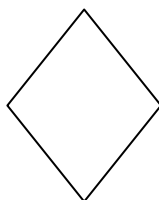
To'g'ri burchakli to'rtburchak, qiymat berish yoki tegishli ko'rsatmalarni bajarish jarayonini belgilaydi.



Parallelogramm, ma'lumotlarni kiritish yoki chiqarishni belgilaydi.



Yordamchi algoritmgga murojatni belgilaydi.



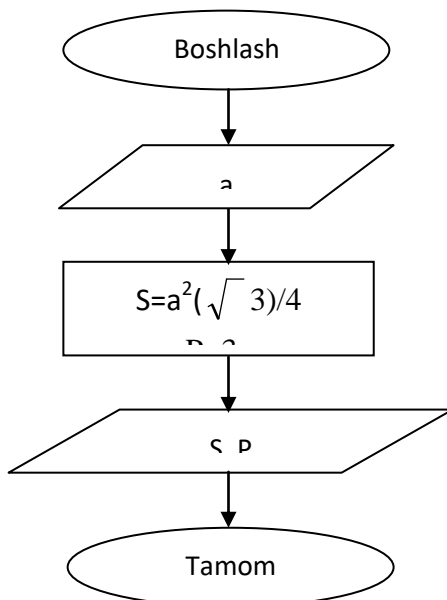
Romb, shart tekshirishni belgilaydi va shart bajarilsa "ha", tarmoq bo'yicha, aks holda "yo'q"-tarmog'i bo'yicha amallar bajarilishini ta'minlaydi.



- Strelka - amallar ketma-ketligining bajarilish yo'nalishini ko'rsatadi.

Yuqorida ko'rilgan algoritmning tasvirlash usullarining asosiy maqsadi, qo'yilgan masalani yechish uchun zarur bo'lgan amallar ketma-ketligining eng qulay holatini aniqlash va shu bilan odam tomonidan programma yozishni yanada osonlashtirishdan iborat. Aslida programma ham algoritmning boshqa bir ko'rinishi bo'lib, u insonning kompyuter bilan muloqotini qulayroq amalga oshirish uchun mo'ljallangan.

**Misol:** Teng tomonli uchburchak tomoni berilganda uning yuzi va perimetrini topish algoritm blok-sxemasini yozing.



## Algoritmning hossalari

Masala yoki mummoni hal etish jarayoni uchun keltirilgan algoritmlar ma'lum bir hususiyatlarga bo'ysinish kerak. Bu hususiyatlarni e'tiborga olib algoritmlar quyidagi xossalarga ega.

1. **Diskretlilik (Cheklilik).** Bu xossaning mazmuni algoritmlarni doimo chekli qadamlardan iborat qilib bo'laklash imkoniyati mavjudligida. Ya'ni uni chekli sondagi oddiy ko'rsatmalar ketma-ketligi shaklida ifodalash mumkin. Agar kuzatilayotgan jarayonni chekli qadamlardan iborat qilib qo'llay olmasak, uni algoritm deb bo'lmaydi.

2. **Tushunarlilik.** Biz kundalik hayotimizda berilgan algoritmlar bilan ishlayotgan elektron soatlar, mashinalar, dastgohlar, kompyuterlar, turli avtomatik va mexanik qurilmalarni kuzatamiz.

Ijrochiga tavsiya etilayotgan ko'rsatmalar, uning uchun tushunarli mazmunda bo'lishi shart, aks holda ijrochi oddiygina amalni ham bajara olmaydi. Undan tashqari, ijrochi har qanday amalni bajara olmasligi ham mumkin.

Har bir ijrochining bajarishi mumkin bo'lgan ko'rsatmalar yoki buyruqlar majmuasi mavjud, u ijrochining ko'rsatmalar tizimi deyiladi. Demak, ijrochi uchun berilayotgan har bir ko'rsatma ijrochining ko'rsatmalar tizimiga mansub bo'lishi lozim. Ko'rsatmalarni ijrochining ko'rsatmalar tizimiga tegishli bo'ladigan qilib ifodalay bilishimiz muhim ahamiyatga ega. Masalan, quyi sinfning a'lochi o'quvchisi "son kvadratga oshirilsin" degan ko'rsatmani tushunmasligi natijasida bajara olmaydi, lekin "son o'zini o'ziga ko'paytirilsin" shaklidagi ko'rsatmani bemalol bajaradi, chunki u ko'rsatma mazmunidan ko'paytirish amalini bajarish kerakligini anglaydi.

3. **Aniqlik.** Ijrochiga berilayotgan ko'rsatmalar aniq mazmunda bo'lishi zarur. Chunki ko'rsatmadagi noaniqliklar mo'ljaldagi maqsadga erishishga olib kelmaydi. Odam uchun tushunarli bo'lgan "3-4 marta silkitilsin", "5-10 daqiqa qizdirilsin", "1-2 qoshiq solinsin", "tenglamalardan biri yechilsin" kabi noaniq ko'rsatmalar robot yoki kompyuterni qiyin ahvolga solib qo'yadi.

Bundan tashqari, ko'rsatmalarning qaysi ketma-ketlikda bajarilishi ham muhim ahamiyatga ega. Demak, ko'rsatmalar aniq berilishi va faqat algoritmda ko'rsatilgan tartibda bajarilishi shart ekan.

4. **Ommaviylik.** Har bir algoritm mazmuniga ko'ra bir turdagi masalalarning barchasi uchun ham o'rinli bo'lishi kerak. Ya'ni masaladagi boshlang'ich ma'lumotlar qanday bo'lishidan qat'iy nazar algoritm shu xildagi har qanday masalani yechishga yaroqli bo'lishi kerak. Masalan, ikki oddiy kasrning umumiy mahrajini topish algoritmi, kasrlarni turlicha o'zgartirib bersangiz ham

ularning umumiy mahrajlarini aniqlab beraveradi. Yoki uchburchakning yuzini topish algoritmi, uchburchakning qanday bo'lishidan qat'iy nazar, uning yuzini hisoblab beraveradi.

**5. Natijaviylik.** Har bir algoritm chekli sondagi qadamlardan so'ng albatta natija berishi shart. Bajariladigan amallar ko'p bo'lsa ham baribir natijaga olib kelishi kerak. Chekli qadamdan so'ng qo'yilgan masala yechimga ega emasligini aniqlash ham natija hisoblanadi. Agar ko'rilayotgan jarayon cheksiz davom etib natija bermasa, uni algoritm deb atay olmaymiz.

**Qiymat berish buyrug'i (taminlash operatori yoki o'zlashtirish operatori)**  
Matematikada  $x=5$  yozuvini  $x$  5 ga teng,  $x=x+1$  yozuvini  $x$   $x+1$  ga teng deb o'qiyamiz. Lekin informatikada o'zgaruvchiga beriladigan qiymat ( $:=$ ) belgi orqali beriladi. Masalan:  $x:=5$  yozuvi  $x$  ga 5 qiymat berilsin deb o'qiladi,  $x:=x+5$  yozuvi esa  $x$  ga  $x+5$  qiymat berilsin yoki  $x$   $x+5$  qiymat bilan taminlansin deb o'qiladi. Informatikada bitta o'zgaruvchi ketma-ket bir necha qiymatlar olsa eng oxirgi olgan qiymati qolib qolganlari o'chib ketadi. Buni quydagicha tasavvur qilamiz. Har bir o'zgaruvchi uchun daftarning bitta katakchasi ajratilgan va o'zgaruvchining har bir qiymati katakcha kattaligidagi qog'ozga yozib o'sha katakchaga kileylab qo'yiladi deb tushuning. Natijada oldingilari ko'rinmay qoladi. Masalan, quydagi buyruqlar bajarilgach  $a$  va  $b$  lar qanday qiymatlar oladi:

$a:=4$ ,  $b:=27$ ;  $a:=\sqrt[3]{b}$ ;  $b:=a*b$ ;  $a:=b/3$ ;  $b:=b+a$ ;  $a:=b/a$ ;

Yechish:

a	b
4	27
3	81
27	108
4	

Javob:  $a=4$ ;  $b=108$

### **Nazariy savollar**

1. Algoritm so'zi tarixini tushuntirib bering?
2. Algoritm muallifi va ijrochisi?
3. Algoritm ta'rifi?
4. Algoritmning tasvirlash usullarini tushuntirib bering?
5. Algoritmning hossalari va ularning vazifalari?
6. Algoritmning ommaviylik hossasini tushuntiring.
7. Algoritmning natijaviylik hossasini tushuntiring.
8. Algoritmning dickretlik hossasini tushuntiring.

9. Algoritmning aniqlik hossasini tushuntiring.
10. Algoritmning tushunarlik hossasini tushuntiring.
11. Algoritm so'zi kimning nomi bilan bog'liq? Nima uchun?
12. Qiymat berish buyrug'i qanday yoziladi?
13. Algoritmni formal ijro etishning mohiyatini tushuntiring.
14. Algoritm qachon to'g'ri tuzilgan hisoblanadi?
- 15  $p := 2p + 3$  yozuvi qanday o'qiladi?
16. Algoritm matn usulida qanday yoziladi?

### ***Mustaqil ishlash uchun topshiriqlar***

Quyidagi masalalarning algoritmlarini tuzing:

1. Aylana chizish;
2. Lug'atdan so'z tarjimasini topish;
3. Kofe tayyorlash;
4. Biror taomni tayyorlash;
5. Telefondan foydalanish;
6. Dars tayyorlash;
7. Misol va masalalarni yechish algoritmlarini tuzing.
8.  $(12a+b-8)/(4x^3-23/(x+1)+c)+5x-28a$  ifodaning qiymatini hisoblash algoritmlarini matn usulida tuzing va har bir algoritmda 5 ta xossa qanday namoyon bolayotganligini va ulardan hohlagan biri buzulsa algoritm qanday ko'rinishda bo'lishini aytib bera olishga tayyorlaning.

## **1.2 ALGORITM VA UNING TURLARI**

*Reja:*

1. Chiziqli algoritmlar;
2. Mantiqiy ifoda va mantiqiy amallar
3. Tarmoqlanuvchi algoritmlar;
4. Takrorlanuvchi algoritmlar;
5. Ichma ich sikllar.

Har qanday masala yoki muommoni hal etishda ma'lum bir algoritm bo'yicha harakatlanamiz. Masalalar va ularning algoritmlari ham ma'lum bir turlarga bo'linadi. Har qanday murakkab masalani ham uchta asosiy strukturaga keltirish mumkin. Algoritmni umumlashtirgan holda quyidagi turlarga ajratamiz.

- Chiziqli algoritmlar;
- Tarmoqlanuvchi algoritmlar;
- Takrorlanuvchi algoritmlar.



## Chiziqli algoritmlar

Masalani hal etish uchun tuzilgan algoritm tarkibidagi buyruqlar ketma ketligi uzluksiz bo'lishi mumkin yoki qandaydir holatlarda shartlar asosida uzluksizlik tarqatilishi mumkin. Chiziqli algoritmlarda esa buyruqlar ketma-ketligi doim uzluksiz bo'ladi.

Algoritmni ijro etishda uning buyruqlari qanday tartibda berilgan bo'lsa, o'sha tartibda bajarilsa, bunday algoritmlarni bajarish " buyruqlar tabiiy tartibiga bo'ysunadi " deyiladi. Agar algoritmlarning buyruqlarini bajarish tabiiy tartibga bo'ysunsa, bunday algoritmlar **chiziqli algoritmlar** deyiladi. Biz to hozirga qadar o'rgangan barcha algoritmlar chiziqli algoritmlardir .

*Tarif: Algoritm bajarilish vaqtida hech qanday to'siqqa uchramasdan buyruqlar bajarilish ketma-ketligi tabiiy tartibgaga bo'ysunsa bunday algoritmlar chiziqli algoritmlar deyiladi.*

Demak algoritm bajarilishida hech qanday shart bo'lmaslik va uzluksizlik yo'qolmaslik kerak. Algoritm tuzish vaqtida uning turini aniqlash uchun masala tarkibida hech qanday shart yoki takrorlanish bo'lmaslik kerak. Har qanday masala algoritmini ham uchta chiziqli, shartli va takrorlanuvchi algoritmlar yordamida tasvirlash mumkin.

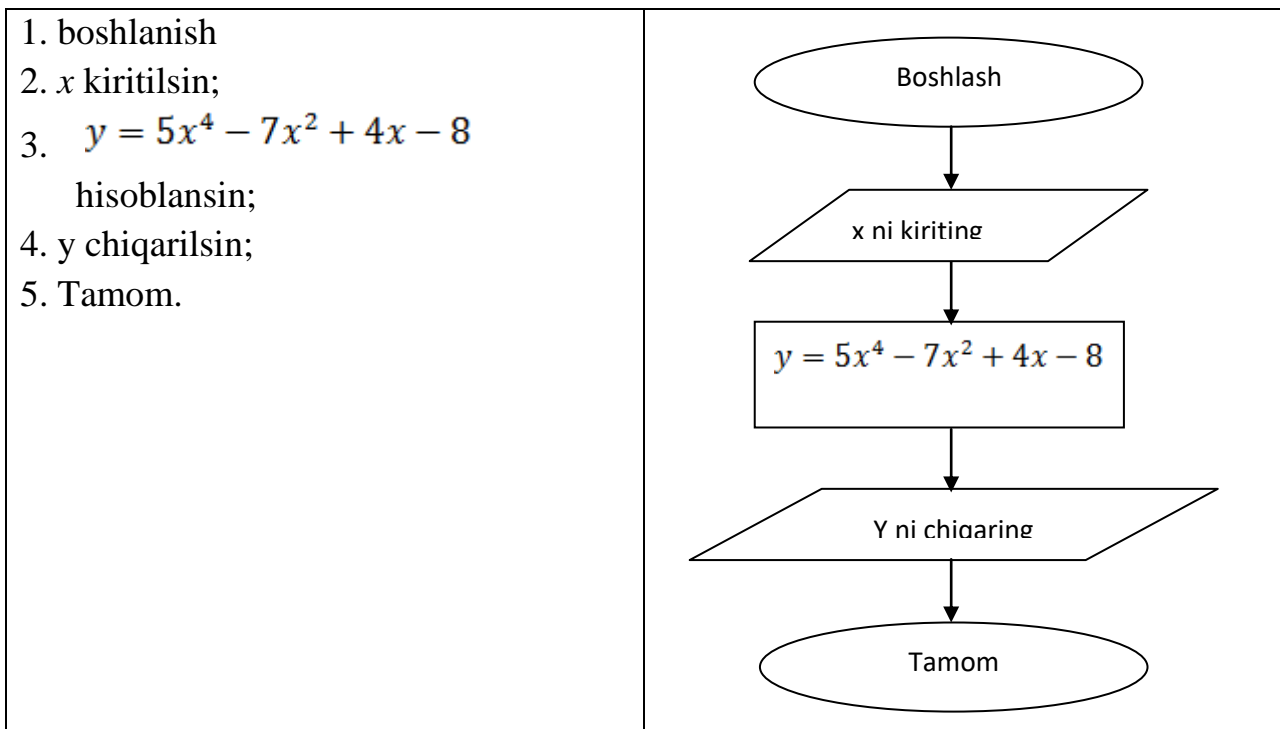
Chiziqli algoritmlar bajarilish vaqtida buyruqlar ketma-ketligi buzilmasdan davom etadi. Masala tarkibida hech qanday shartlar va takrorlanishlar ishtirok etmasa, bunday masalalarni chiziqli algoritmlar asosida hal etish mumkin.

**Misol:** *x ni qiymati berilganda quyidagi funksiyani hisoblash algoritmini keltiring.*

$$y = 5x^4 - 7x^2 + 4x - 8$$

Bu masala algoritmini tuzish jarayoni x nomalumning qiymati berilganda y funksiyaning natijasi hisoblanish kerak. Demak faqat x ning qiymati kiritilib y funksiyaning natijasi hisoblanish kerak bo'ladi. Berilgan masala uchun algoritmning quyidagicha ya'ni so'zlar va blok-sxema ko'rinishida tasvirlanishiga e'tibor bering.

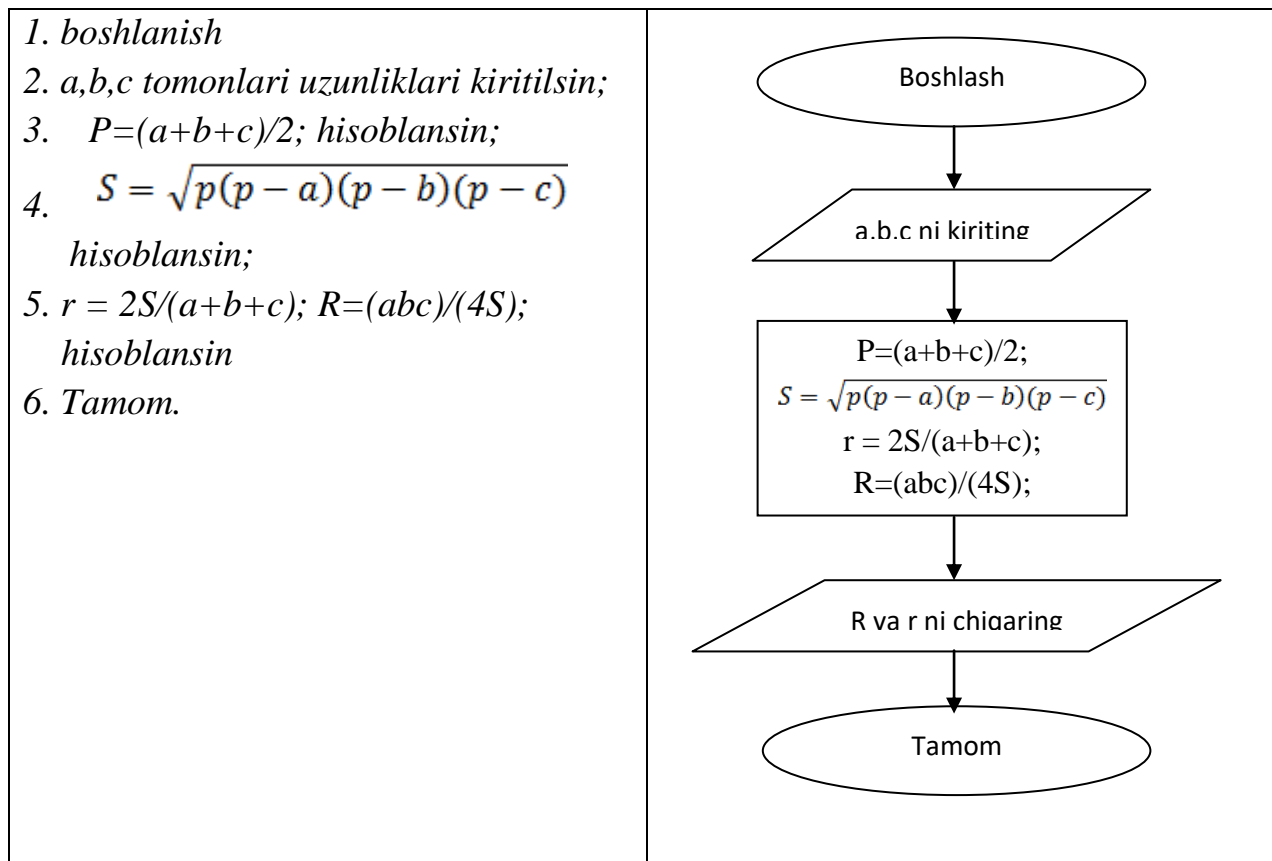
Algoritmni so'zlar yordamida tasvirlanishi	Algoritmni blok-sxema yordamida tasvirlanishi
--	---



Blok-sxemalar bilan ishlashni yaxshilab o‘zlashtirib olish zarur, chunki bu usul algoritmlarni ifodalashning qulay vositalaridan biri bo‘lib, programma tuzishni osonlashtiradi, programmalash qobiliyatini mustahkamlaydi. Algoritmik tillarda blok - sxemaning asosiy strukturalariga maxsus operatorlar mos keladi. Shuni aytish kerakki, blok-sxemalardagi yozuvlar odatdagi yozuvlardan katta farq qilmaydi. Faqat ketma-ket bajariladigan amallardan tashkil topgan algoritmlarga-chiziqli algoritmlar deyiladi. Bunday algoritmni ifodalash uchun ketma-ketlik strukturasi ishlatiladi. Strukturada bajariladigan amal mos keluvchi shakl bilan ko‘rsatiladi. Aslida programma ham algoritmning boshqa bir ko‘rinishi bo‘lib, u insonning kompyuter bilan muloqotini qulayroq amalga oshirish uchun mo‘ljallangan.

**Misol:** Uchburchak tomonlari berilganda unga ichki va tashqi chizilgan aylana radiuslarini hisoblash algoritmi tuzilsin.

Algoritmni so‘zlar yordamida tasvirlanishi	Algoritmni blok-sxema yordamida tasvirlanishi
--	---



### ***Mantiqiy ifoda va mantiqiy amallar***

Mantiqiy ifoda tushunchasi *matematik mantiq(logika) fani* tushinchlariga tayanib tushuntiriladi. Matematik mantiq deganda *fikrlashning shakl va qonunlari tushuniladi*. Matematik mantiq tushunchaga ingliz olimi Jorj Bull tomonidan asos solingan.

Har qanday mantiqiy ifoda yoki mantiqiy fikr faqat ikki xil qiymat qabul qilishi mumkin. Bular rost(true) yoki yolg'on (false). Masalan Sobir qiz bola, po'lat suyuq modda kabi fikrlarning qiymati yolg'on yoki suv  $0^0$  da muzlaydi, ruchka yozish quroli kabi fikrlar rost. Matematik mantiqiy ifodalarni yozishda quyidagi munosabat belgilaridan foydalanadi:

$$=, \neq, <, >, \geq, \leq; \quad (1)$$

Agar ikkita ifoda (1) munosabat belgilari bilan birlashtirilgan bo'lsa bunday yozuv mantiqiy ifoda deyiladi. Masalan:

- 1)  $32-4=20+5$ ; 2)  $x+5 \neq 2x-1$ ; 3)  $26/2*3 \neq 39$ ; 4)  $49/7+3=2*5$ ; 5)  $a+b \geq b+c$ ;

Bu yozuvlarning har biri mantiqiy ifodalar bo'lib, ulardan 1 va 3-ifodalarining qiymati yolg'on, 4-ifodaning qiymati rost, 2 va 5-ifodalarning qiymati o'zgaruvchilarning qiymatiga qarab rost yoki yolg'on bo'ladi. Yuqorida qayd qilganimizdek, faqat ikki xil qiymatlar qabul qiladi, ya'ni ***rost*** va ***yolg'on***.

Agar ikkita ifoda *bitta munosabat belgisi orqali* birlashtirilgan bo'lsa, bunday mantiqiy ifodalar *oddiy mantiqiy ifodalar* deyiladi.

Agar mantiqiy ifoda tarkibida *mantiqiy amallardan* ham foydalanilgan bo'lsa, bunday mantiqiy ifodalar *murakkab mantiqiy ifodalar* deyiladi. Murakkab mantiqiy ifodalar tarkibida quyidagicha mantiqiy amallardan foydalaniladi. ( $\neg$   $\vee$   $\wedge$ )

1. **Mantiqiy qo'shish amali.** Amalning umumiy ko'rinishi quyidagicha  
 **$a \vee b$**

Bunda **a** va **b** lar mantiqiy o'zgaruvchilar yoki mantiqiy ifodalar bo'lib, bu amal natijasi **a** va **b** larning har ikkalasi yolg'on bo'lgandagina yolg'on, qolgan barcha hollarda rost bo'ladi. Demak,  $\vee$  belgisi *yoki* so'zinini almashtiradi yoki matematik mantiq elementlaridagi *dizyunksiya amaliga* mos keladi.

Masalan :  $(3 > 7) \vee (2 * 18 = 37)$  – ifodaning qiymati yolg'on(false)

$(25 - 14 \neq 11) \vee (45 / 9 = 5)$  – ifodaning qiymati rost(true)

$true \vee (29 - 19 = 10)$  – ifodaning qiymati rost(true)

Byndan tashqari mantiqiy ifodaning qiymatini mantiqiy o'zgaruvchiga qiymat sifatida berish ham mumkin. Masalan,

$a := 13 - 5 > 12$ ;  $b := (56 / 8 - 7 < 2) \vee true$ ;  $c := false \vee a$ ; bo'lsa,  $a \vee b \vee c \vee (64 / 8 > 10)$  mantiqiy ifodaning qiymati rost bo'ladi.

2. **Mantiqiy ko'paytirish amali.** Amalning umumiy ko'rinishi quyidagicha

**$a \wedge b$**

Bunda **a** va **b** lar mantiqiy o'zgaruvchilar yoki mantiqiy ifodalar bo'lib, bu amal natijasi **a** va **b** larning har ikkalasi chin bo'lgandagina chin, qolgan barcha hollarda yolg'on bo'ladi. Demak,  $\wedge$  belgisi *va* so'zinini almashtiradi yoki matematik mantiq elementlaridagi *konyuksiya amaliga* mos keladi. Masalan:

$(3 > 7) \wedge (2 * 18 = 37)$  ifodaning qiymati yolg'on

$(25 - 14 \neq 11) \wedge (45 / 9 = 5)$  – ifodaning qiymati rost

$(13 > -12) \wedge (2 * 18 = 37)$  ifodaning qiymati yolg'on

3. **Mantiqiy inkor amali.** Amalning umumiy ko'rinishi quyidagicha

**$\neg a$**

Bunda **a** mantiqiy o'zgaruvchi yoki mantiqiy ifoda bo'lib, bu amal natijasi **a** ning qiymati yolg'on bo'lganda rost, rost bo'lganda yolg'on bo'ladi, ya'ni bu amal **a** mulohazani inkor qiladi. Demak,  $\neg$  belgisi *emas* so'zinini almashtiradi. . Masalan:

- 1)  $(3 > 7) \wedge \neg (2 * 18 = 37)$  ifodaning qiymati yolg'on

- 2)  $\neg (25-14 \neq 11) \wedge (45/9=5)$  ifodaning qiymati yolg'on
- 3)  $(13 > -12) \wedge \neg (2 \cdot 18 = 37)$  ifodaning qiymati rost
- 4)  $\neg \text{true} \vee \neg \text{false} \wedge \neg \text{true} \wedge \neg \text{false}$  ifodaning qiymati yolg'on

**Eslatma:** Biz bilgan amallar ifodada aralash holda kelsa, ularning birinchi bajarilishi nuqtai nazaridan ustunlik darajasi quyidagicha:

$\neg, (*, /), \wedge, (+, -), \vee, (=, \neq), (>, <, \geq, \leq)$

Bunda qavsga olingan amallar bir bosqich amallari hisoblanadi. Tabiiyki, matematikada bo'lgani singari ifodadagi qavslar amallar bajarilish tartibina buzadi. Masalan,

14            1 2 3            11            4 5            13 10            6 7            12  
8 9

$\neg ((13 \cdot 5 - 25 = 40) \wedge (13 - 3 = 10) \vee \neg (5 + 7 = 12) \wedge (2 \cdot 9 = 20));$

### Mustaqil ishlash uchun topshiriqlar

1. Kvadratning tomoni  $a$  berilgan. Uning perimetri aniqlansin.  $P=4 \cdot a$ .
2. Kvadratning tomoni  $a$  berilgan. Uning yuzasi aniqlansin.  $S=a^2$ .
3. To'g'ri to'rtburchakning tomonlari  $a$  va  $b$  berilgan. Uning yuzasi  $S=a \cdot b$ ; va  $P=2 \cdot (a+b)$  perimetri aniqlansin.
4. Aylananing diametri  $d$  berilgan. Uning uzunligi aniqlansin  $L=\pi \cdot d$ .
5. Kubning yon tomoni  $a$  berilgan. Uning hajmini  $V=a^3$  va to'la sirti  $S=6 \cdot a^2$  aniqlansin.
6. Paralelepipedning tomonlari  $a, b, c$  berilgan. Uning hajmi  $V=a \cdot b \cdot c$  va to'la sirti  $S=2 \cdot (a \cdot b + b \cdot c + a \cdot c)$  aniqlansin.
7. Doiraning radiusi  $R$  berilgan. Uning uzunligi  $L$  va yuzasi  $S$  aniqlansin.  
 $L=2 \cdot \pi \cdot R,$   $S=\pi \cdot R^2$ .
8. Ikkita son  $a$  va  $b$  berilgan. Ularning o'rta arifmetigi aniqlansin.  $(a+b)/2$ .
9. Ikkita manfiy bo'lmagan son  $a$  va  $b$  berilgan. Ularning o'rta geometrigi aniqlansin.  $\sqrt{a \cdot b}$ .
10. Nolga teng bo'lmagan ikkita son berilgan. Ularning yig'indisi, ko'paytmasi va har birining kvadrati aniqlansin.
11. Nolga teng bo'lmagan ikkita son berilgan. Ularning yig'indisi, ko'paytmasi va har birining moduli aniqlansin.
12. To'g'ri uchburchakning katetlari  $a$  va  $b$  berilgan. Uning gipotenuzasi  $c$  va perimetri  $P$  aniqlansin.  $c = \sqrt{a^2 + b^2},$   $P = a + b + c$ .
13. Umumiy markazga bo'lgan ikkita aylana radiusi berilgan.  $R_1$  va  $R_2$  ( $R_1 > R_2$ ). Ularning yuzalari  $S_1$  va  $S_2$ , ularning ayirmasi  $S_3$  aniqlansin.  $S_1=\pi \cdot (R_1)^2,$   $S_2=\pi \cdot (R_2)^2,$   $S_3=S_1-S_2$ .

14. Aylananing uzunligi  $L$  berilgan. Uning radiusi  $R$  va yuzasi  $S$  aniqlansin.  
 $L=2\cdot\pi\cdot R$ ,  
 $S=\pi\cdot R^2$ .
15. Aylananing yuzasi  $S$  berilgan. Uning diametri  $D$  va uzunligi  $L$  aniqlansin.  
 $L=2\cdot\pi\cdot R$ ,  $S=\pi\cdot R^2$ .
16. Sonlar o'qida ikkita nuqta orasidagi masofa aniqlansin.  $|x_2-x_1|$ .
17. Sonlar o'qida  $A$ ,  $B$ ,  $C$  nuqtalar berilgan.  $AC$  va  $BC$  kesmalarning uzunligini va kesmalar uzunligining yig'indisini topuvchi algoritm tuzilsin.
18. Sonlar o'qida  $A$ ,  $B$ ,  $C$  nuqtalar berilgan.  $C$  nuqta  $A$  va  $B$  nuqtalar orasida joylashgan  $AC$  va  $BC$  kesmalar uzunligining ko'paytmasini toping.
19. To'g'ri to'rtburchakning qarama-qarshi uchlari koordinatlari berilgan. Uning tomonlari koordinata o'qiga parallel. To'g'ri to'rtburchakning perimetri va yuzasi aniqlansin.
20. Tekislikda berilgan ikki nuqta  $(x_1, y_1)$  va  $(x_2, y_2)$  orasidagi masofa topilsin.  
 $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
21. Uchburchakning uchta tomoni uchlari koordinatlari berilgan  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ . Ikki nuqta orasidagi masofa topilsin.
- 22 da berilgan. Uchburchakning yuzasi va perimetrini toping.  
 $S = \sqrt{p\cdot(p - a)\cdot(p - b)\cdot(p - c)}$ ,  
 $p=(a+b+c)/2$
23. Berilgan  $A$  va  $B$  sonlarining qiymatlarini almashtiring.  $A$  va  $B$  ning yangi qiymati ekranga chiqarilsin.
24.  $A$ ,  $B$  va  $C$  sonlari berilgan.  $A$  ni qiymati  $B$  ga,  $B$  ni qiymati  $C$  ga,  $C$  ni qiymati  $A$  ga almashtirilsin.  $A$ ,  $B$  va  $C$  ning yangi qiymatlari ekranga chiqarilsin.
25.  $A$ ,  $B$  va  $C$  sonlari berilgan.  $A$  ni qiymati  $C$  ga,  $C$  ni qiymati  $B$  ga,  $B$  ni qiymati  $A$  ga almashtirilsin.  $A$ ,  $B$  va  $C$  ning yangi qiymatlari ekranga chiqarilsin.
26.  $x$  ning qiymati berilganda  $y=3x^6-6x^2-7$  funksiya qiymati aniqlansin.
27.  $x$  ning qiymati berilganda  $y=4(x-3)^6-7(x-3)^3+2$  funksiya qiymati aniqlansin.
28.  $A$  son berilgan.  $A$  ning  $A^2$ ,  $A^4$ ,  $A^8$  darajalarini aniqlansin.
29.  $A$  son berilgan.  $A$  ning  $A^2$ ,  $A^3$ ,  $A^5$ ,  $A^{10}$ ,  $A^{15}$  darajalarini aniqlansin.
30.  $\alpha$  burchak gradusda berilgan ( $0 < \alpha < 360$ ). Berilgan burchakning qiymatini radianga o'tkazuvchi algoritm tuzilsin.

### **Tarmoqlanuvchi algoritmlar**

Agar masala yoki muommoni hal etish jarayonida qandaydir shartlarga duch kelinsa, masala yechimi shu shart asosida qandaydir usullar yordamida aniqlanishi mumkin. Masala yechimi aniqlanish jarayonida tuzilayotgan algoritmlar shartlar

asosida tarmoqlanishi mumkin, ya'ni shart bajarilish asosida chin qiymat qabul qilganda algoritmning bir qismi yolg'on qiymat qabul qilganda esa algoritmning boshqa qismi bajariladi.

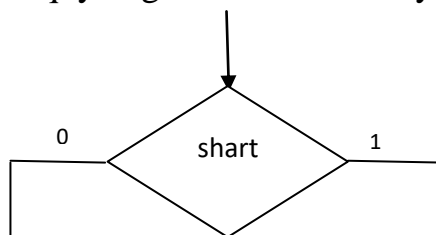
Algoritm bajarilish davomida har doim ham buyruqlar ketma-ketligi bajarilavermaydi, shunday holatlar ham mavjudki, algoritm tarkibida shartlar asosida buyruqlar ketma-ketligi tarmoqlanib ketadi.

*Ta'rif:* Algoritm bajarilish vaqtida buyruqlar ketma-ketligi shartlar asosida u yoki bu qismga tarmoqlanishiga **tarmoqlanuvchi algoritmlar** deyiladi.

Tarmoqlanuvchi algoritmlar quyidagicha so'zlar yordamida tasvirlanadi.

**Agar** (shartli ifoda) **u holda** (hisoblansin) **aks holda** (hisoblansin)

Tarmoqlanuvchi algoritmlar quyidagicha blok-sxema yordamida tasvirlanadi.



Shart chin bo'lganda algoritm 1 tomonga yolg'on bo'lganda esa 0 tomonga harakatlanadi. 1 yoki 0 o'rniga rost yoki yolg'on, + yoki -, xa yoki yo'q belgilarni yozish mumkin. Algoritmni ifodalovchi blok-sxema ko'rinishiga e'tibor qaratsangiz romb belgisi tarkibiga algoritmning shartli ifodasi keltiriladi. Agar algoritm sharti natijasi chin qiymat qabul qilsa, algoritm + tarafdagi buyruqlarga o'tadi aks holda – tarafdagi buyruqlar ketma-ketligiga o'tadi.

**Misol:** Ikkita sondan kattasini aniqlash jarayonini ifodalovchi algoritmni blok-sxema va so'zlar yordamida tuzilsin.

Algoritmni so'zlar yordamida tasvirlanishi	Algoritmni blok-sxema yordamida tasvirlanishi
--	---

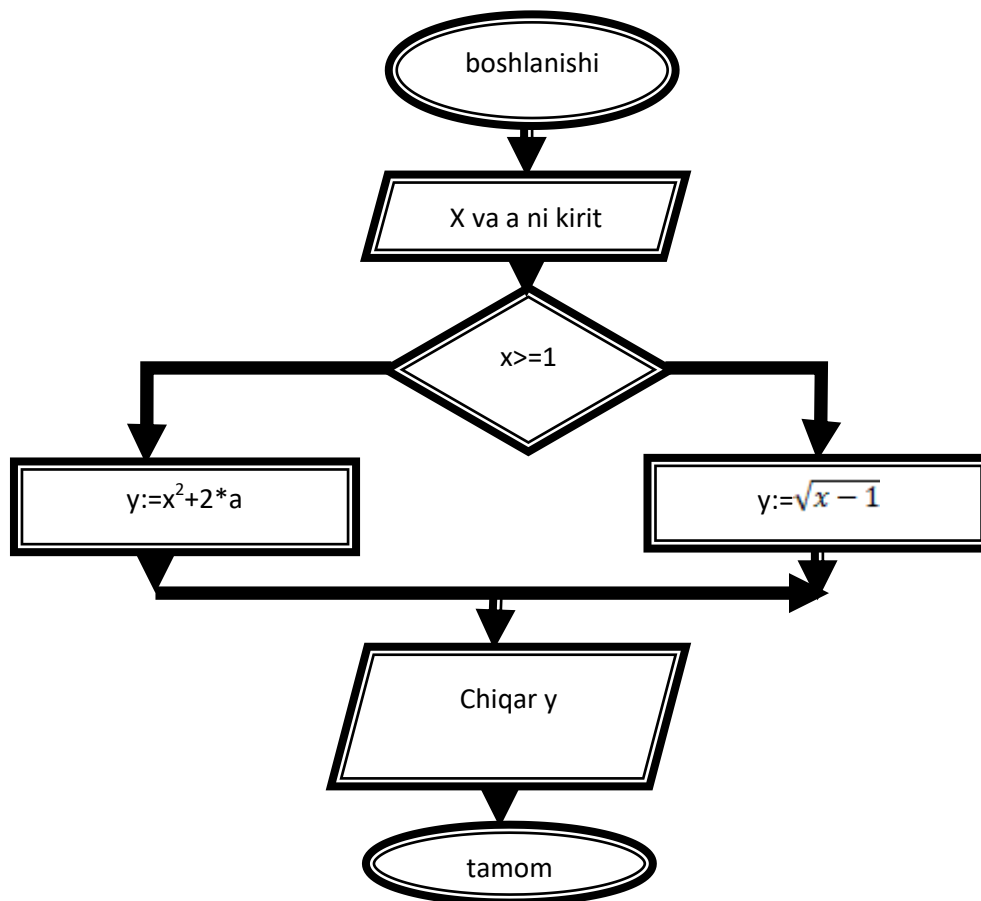
<p>1.boshlanish  2.a va b sonni kiritilsin;  3.Agar <math>a &gt; b</math> uholda <math>max=a</math>  aks holda <math>max=b</math>;  4. max chiqarilsin;  5.tamom.</p>	<pre> graph TD     Start([Boshlanish]) --&gt; Input[/a va b kiritilsin/]     Input --&gt; Decision{a &gt; b}     Decision -- 1 --&gt; Process1[<math>max=a</math>]     Decision -- 0 --&gt; Process2[<math>max=b</math>]     Process1 --&gt; Output[/max chiqarilsin/]     Process2 --&gt; Output     Output --&gt; End([Tamom]) </pre>
---	---

**Misol:** Kvadrat tenglamani hisoblovchi algoritm tuzing.

Algoritmni so‘zlar yordamida tasvirlanishi	Algoritmni blok-sxema yordamida tasvirlanishi
<p>1.boshlanish  2.a,b,c sonni kiritilsin;  3.<math>D=b^2-4ac</math> hisoblansin;  4.Agar <math>D &gt; 0</math> uholda  <math>x_1=(-b+\sqrt{d})/(2a)</math>  <math>x_2=(-b-\sqrt{d})/(2a)</math>  <math>x_1</math> va <math>x_2</math> chiqarilsin  aks holda  agar <math>d &lt; 0</math> uholda (yechim yuq)  aks holda  <math>x=-b/(2a)</math>  <math>x</math> chiqarilsin;  5.Tamom.</p>	<pre> graph TD     Start([Boshlanish]) --&gt; Input[/a,b,c kiritilsin/]     Input --&gt; Process[D=b^2-4ac]     Process --&gt; Decision1{D &gt; 0}     Decision1 -- 1 --&gt; Process1["<math>x_1=(-b+\sqrt{d})/(2a)</math> <math>x_2=(-b-\sqrt{d})/(2a)</math>"]     Decision1 -- 0 --&gt; Decision2{D &lt; 0}     Decision2 -- 1 --&gt; Process2[Yechim yuq]     Decision2 -- 0 --&gt; Process3[x=-b/(2a)]     Process1 --&gt; Output1[/x1 va x2/]     Process2 --&gt; Output2[/Yechim yuq/]     Process3 --&gt; Output3[/x/]     Output1 --&gt; End([Tamom])     Output2 --&gt; End     Output3 --&gt; End </pre>

**Misol:**  $y = \begin{cases} \sqrt{x-1} & \text{agar } x \geq 1 \\ x^2 + 2a & \text{agar } x < 1 \end{cases}$  funksiyani qiymatini hisoblash algoritmini tuzing.





### Nazariy savollar

1. Qanday masalaalrning algoritmlarini tuzishda tarmoqlanish buyrug'dan foydalaniladi?
2. Qanday algoritmlar tarmoqlanish algoritmlari deyiladi?
3. Tarmoqlanish buyrug'i qanday bajariladi?
4. Blok-sxemada strelkali siniq chiziqlardan qanday maqsadda foydalaniladi?
5. Blok-sxemaning har bir blokinini chizib, uuning vazifasini tushuntiring.
6. Blok-sxemada chiqarish blokidan foydalanilmasa qanday xatolik ro'y beradi?
7. Blok-sxemada kiritish blokidan foydalanilmasa qanday xatolik ro'y beradi?

### Mustaqil ishlash uchun topshiriqlar.

Quyidagi masalalalar uchun algoritmlar tuzing.

1.  $A$  butun soni berilgan. Jumlani rostlikka tekshiring " $A$  soni musbat".
2.  $A$  butun soni berilgan. Jumlani rostlikka tekshiring " $A$  soni toq son".
3.  $A$  butun soni berilgan. Jumlani rostlikka tekshiring " $A$  soni juft son".
4. Ikki butun  $A$  va  $B$  sonlari berilgan. Jumlani rostlikka tekshiring:  $A > 2$  va  $B \leq 3$ .
5. Ikki butun  $A$  va  $B$  sonlari berilgan. Jumlani rostlikka tekshiring:  $A \geq 0$  yoki  $B < -2$ .
6. Uchta  $A, B, C$  butun sonlari berilgan. Jumlani rostlikka tekshiring:  $A < B < C$

7. Uchta  $A, B, C$  butun sonlar berilgan. Jumlani rostlikka tekshiring: “ $B$  soni  $A$  va  $C$  sonlari orasida yotadi”.
8. Ikkita butun  $A$  va  $B$  sonlari berilgan. Jumlani rostlikka tekshiring. “ $A$  va  $B$  sonlari toq sonlar”.
9. Ikkita butun  $A$  va  $B$  sonlari berilgan. Jumlani rostlikka tekshiring. “ $A$  va  $B$  sonlarining hech bo‘lmaganda bittasi toq son”.
10. Ikkita butun  $A$  va  $B$  sonlari berilgan. Jumlani rostlikka tekshiring. “ $A$  va  $B$  sonlarining faqat bittasi toq son”.
11. Ikkita butun  $A$  va  $B$  sonlari berilgan. Jumlani rostlikka tekshiring. “ $A$  va  $B$  sonlarining har ikkalasi ham yoki toq son yoki juft son”.
12. Uchta  $A, B, C$  butun sonlar berilgan. Jumlani rostlikka tekshiring: “ $A, B, C$  sonlarining hech bo‘lmaganda bittasi musbat”.
13. Uchta  $A, B, C$  butun sonlar berilgan. Jumlani rostlikka tekshiring: “ $A, B, C$  sonlarining faqat bittasi musbat son”.
14. Uchta  $A, B, C$  butun sonlar berilgan. Jumlani rostlikka tekshiring: “ $A, B, C$  sonlardan ikkitasi musbat son”.
15. Musbat butun son berilgan. Jumlani rostlikka tekshiring: “Berilgan son ikki xonali juft son”.
16. Musbat butun son berilgan. Jumlani rostlikka tekshiring: “Berilgan son uch xonali toq son”.
17. Jumlani rostlikka tekshiring: “Berilgan uchta butun sonlarning hech bo‘lmaganda 2 tasi bir biriga teng”.
18. Jumlani rostlikka tekshiring: “Berilgan uchta butun sonlarning hech bo‘lmaganda bir jufti o‘zaro qarama-qarshi”.
19. Uch xonali son berilgan. Jumlani rostlikka tekshiring: “Ushbu sonning barcha raqamlari xar xil”.

### **Takrorlanuvchi algoritmlar.**

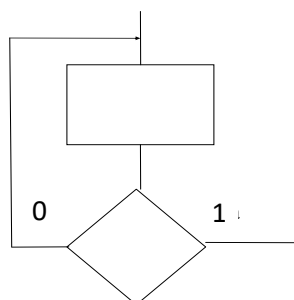
Masala yoki muommoni hal etishda ba’zi jarayonlar bir necha marta takrorlanish mumkin. Algoritm tarkibida biror parametr qandaydir shartga bog‘liq ravishda oshishi yoki kamiyish hisobiga takrorlanish jarayonlari vujudga kelishi mumkin.

*Ta’rif:* Algoritmning ma’lum bir qismi qandaydir shartlar asosida ikki va undan ortiq bajarilishiga **takrorlanuvchi algoritmlar** deyiladi.

Takrorlanuvchi algoritmlar bajarilish vaqtida uning qandaydir qismi bir necha marta takrorlanadi. Har qanday ketma-ketliklarni yig‘indisini formula

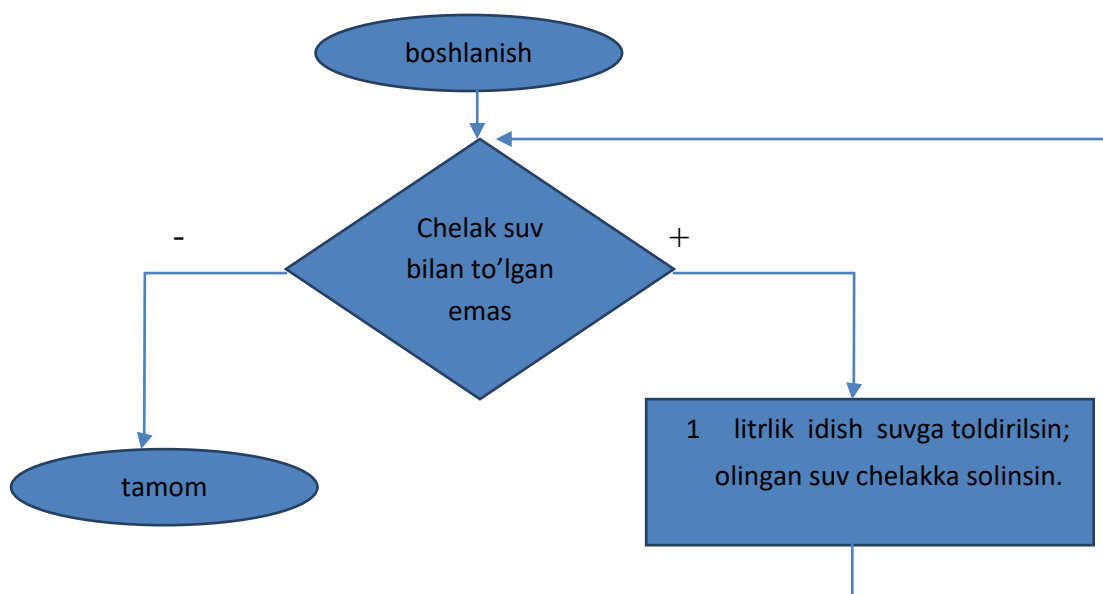
yordamida hisoblash muommo bunday holatlarda EHM da hisoblash bir muncha qulay hisoblanadi. Algoritm bajarilish davomida har doim ham buyruqlar ketma-ketligi bajarilavermaydi, shunday holatlar ham mavjudki, algoritm tarkibida shartlar asosida buyruqlar ketma-ketligi bir necha marta takrorlanish ham mumkin. Algoritm tarkibidagi shartning o'zgarishiga qarab takrorlanish jarayoni bajariladi. Masalan 1 dan n gacha sonlarning kvadratlar yig'indisi yoki yig'indisi x ga teng sinuslarning kvadratlar yig'indisi kabi masalalar algoritmi shartlar asosida takrorlanishlar yordamida tuziladi.

Takrorlanuvchi jarayonlarni quyidagi blok-sxema yordamida tasvirlash mumkin.

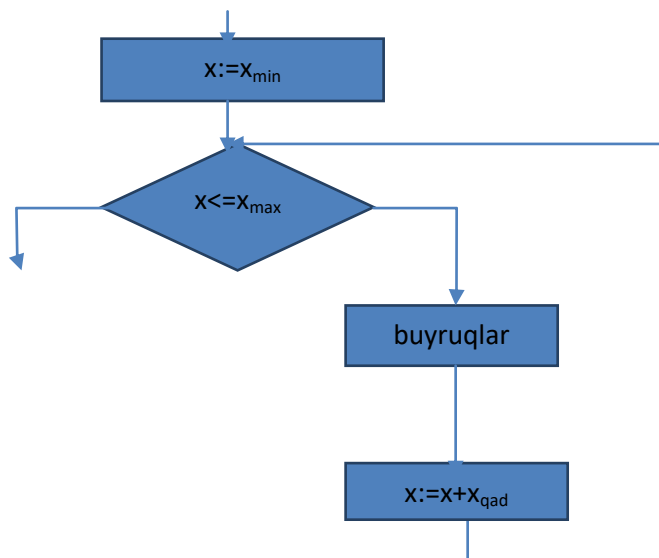


Shunday masalalar ham uchraydiki , uni hal qilish uchun chekli sondagi buyruqlar ketma- ketligini takror-takror bajarish orqali maqsadga erishamiz. Bunday davriy takrorlanish tsiklik jaryon yoki qisqacha tsikl deyiladi.

**Misol:** 1 litrlik idish bilan chelakni suvga to'ldirish algoritmi tuzing.



Topshiriq: 4-5 ta hayotiy masalalar tanlab algoritm tuzing. Biz amalda ko'proq matematik masalalarni yechishda sikl buyrug'idan foydalanamiz. Buyruqning umumiy ko'rinishi quyidagicha:

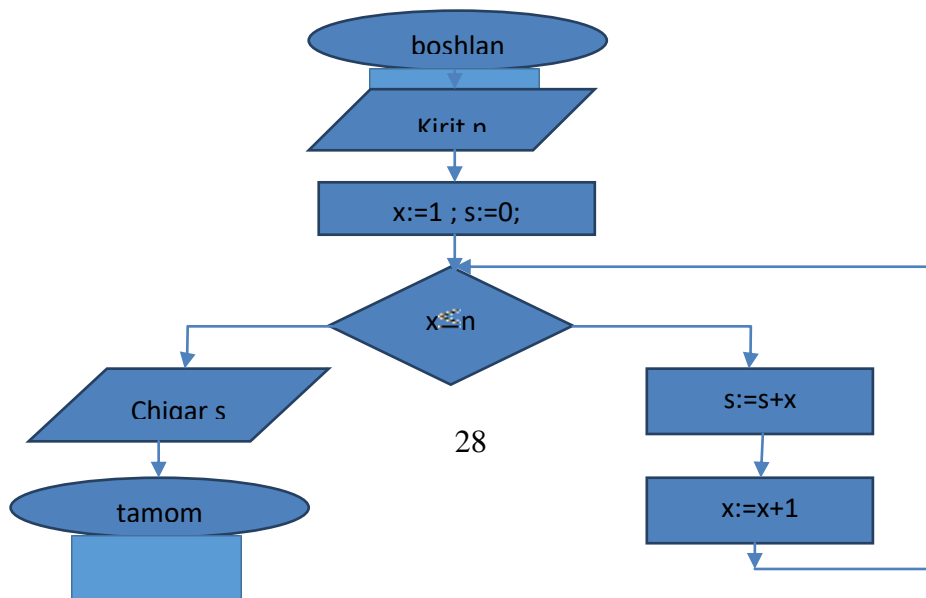


Bunda x-sikl parametri,  $x_{min}$  –sikl parametrining dastlabki qiymati,  $x_{max}$ -sikl parametrining oxirgi qiymati,  $x_{qad}$  –har takrorlanishda sikl parametri o'zgaradigan qadam.

Buyruq quyidagi tartibda bajariladi:

Tsikl parametri o'zining dastlabki  $x_{min}$  qiymatini oladi va oxirgi  $x_{max}$  qiymati bilan solishtiriladi. Agar  $x \leq x_{max}$  mantiqiy ifodaning qiymati rost bo'lsa, buyruqlar bajarilib, sikl parametrining qiymati bir qadamga ortadi va yana  $x \leq x_{max}$  mantiqiy ifodaning qiymati tekshiriladi, agar uning qiymati rost bo'lsa, buyruqlar yana bir bor bajarilib, sikl parametri bir qadamga ortadi va  $x \leq x_{max}$  mantiqiy ifodaning qiymati tekshiriladi va hokazo bu jarayon  $x \leq x_{max}$  mantiqiy ifodaning qiymati yolg'on, ya'ni  $x > x_{max}$  bo'lguncha davom etadi va shundan so'ng sikl buyrug'idan keyingi buyruqlarga o'tiladi.

**Misol:** 1 dan n gacha bo'lgan natural sonlar yig'indisini topish algoritmini tuzing.



**Misol:**  $N!$  ni hisoblash jarayonini ifodalovchi algoritmni blok-sxema va so'zlar yordamida tuzilsin.

Algoritmni so'zlar yordamida tasvirlanishi	Algoritmni blok-sxema yordamida tasvirlanishi
<p>1. boshlanish                  2. <math>n</math> soni kiritilsin;                  3. <math>P=1, i=0</math>;                  4. <math>i=i+1; p=p*i</math>;                  5. Agar <math>i &lt; n</math> uholda 4-ga qaytilsin aks holda <math>P</math> chiqarilsin;                  6. tamom.</p>	<pre>                 graph TD                     Start([Boshlanish]) --&gt; Input[/n kiritilsin/]                     Input --&gt; Init[i=0; p=1;]                     Init --&gt; Loop[i=i+1; p=p*i]                     Loop --&gt; Decision{i &lt; n}                     Decision -- 1 --&gt; Loop                     Decision -- 0 --&gt; Output[/P chiqarilsin/]                     Output --&gt; End([Tamom])                 </pre>

Takrorlanuvchi algoritmlar asosan ikki xil ko'rinishda bo'ladi.

- Takrorlanishlar soni oldindan aniq;
- Takrorlanishlar soni qandaydir shartlarga bog'liq.

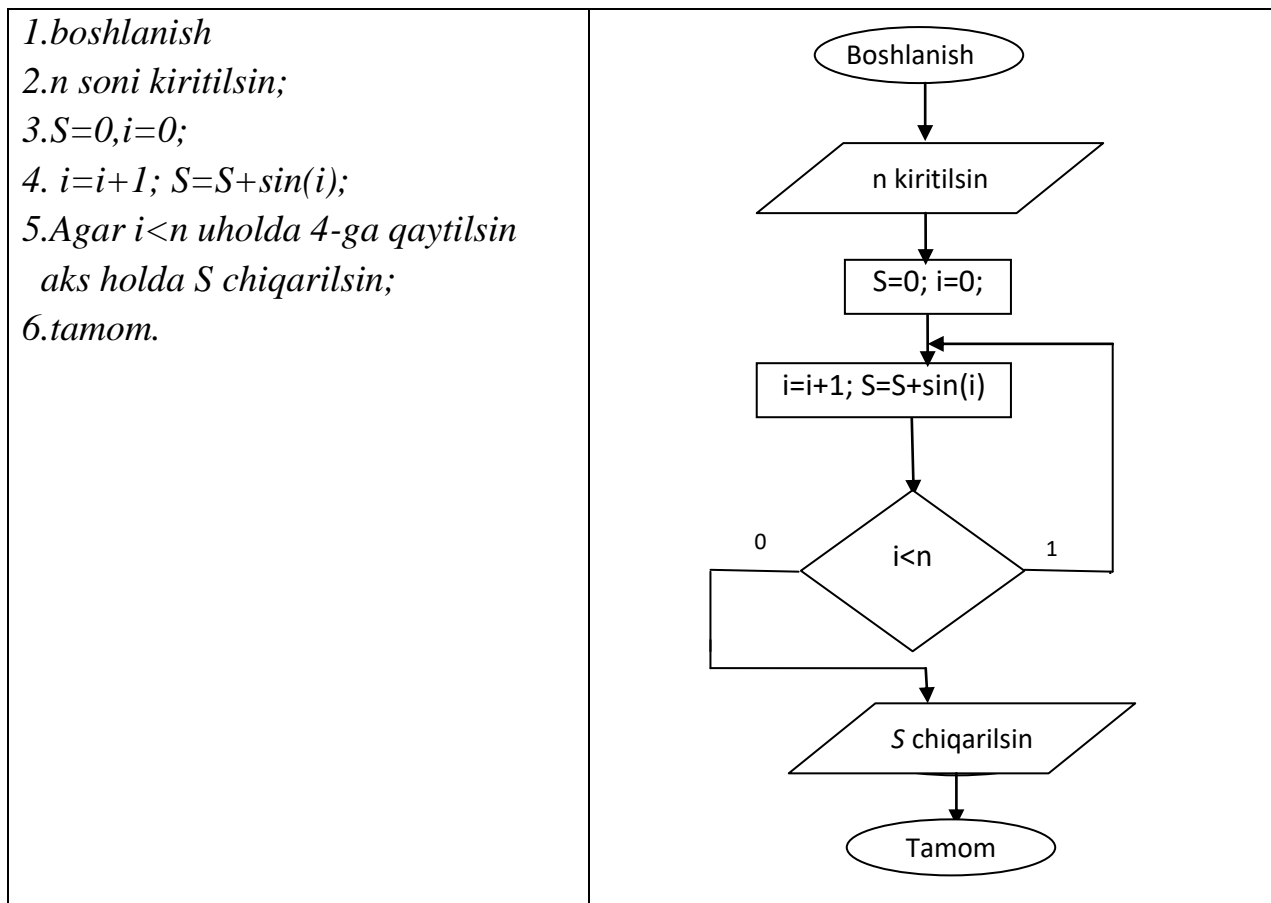
Takrorlanishlar soni oldindan aniq bo'lgan masalalarda algoritm bajarilishini bitta parameter soni bilan bog'lanadi. Takrorlanishlar soni qandaydir shartlarga bog'liq bo'lgan holatlarda takrorlanishni shartli jarayon bilan ifodalanadi.

**Misol:** Quyidagi yig'indini hisoblang.

$$S = \sin(1) + \sin(2) + \dots + \sin(n)$$

Berilgan masalani yechish algoritmi tarkibida takrorlanish soni oldindan ma'lum chunki bu  $n$  ga bog'liqdir.

Algoritmni so'zlar yordamida tasvirlanishi	Algoritmni blok-sxema yordamida tasvirlanishi



### Nazariy savollar

- 1) Sikl buyrug'ining umumiy ko'rinishini yozing va bajarilish tartibini tushuntiring
- 2) Sikl buyrug'ining parametri, boshlang'ich va oxirgi qiymatlari, qadami deyilganda nimani tush-z?
- 3) Yuqoridagi algoritmda  $s:=0$  buyrug'i nega berilgan?
- 4) Yuqoridagi algoritmda qanday o'zgarish qilsak u juft sonlar yig'indisini hisoblaydi?
- 5) Yuqoridagi algoritmda qanday o'zgarish qilsak u toq sonlar yig'indisini hisoblaydi?
- 6) Yuqoridagi algoritmda qanday o'zgarish qilsak u juft sonlar ko'paytmasini hisoblaydi?
- 7) Yuqoridagi algoritmda qanday o'zgarish qilsak u 7 ga karrali sonlar ko'paytmasini hisoblaydi?
- 8) Yuqoridagi algoritmda qanday o'zgarish qilsak, cheksiz takrorlanish sodir bo'ladi?
- 9) Yuqoridagi algoritmda  $s:=0$  buyrug'i olib tashlansa natija qanday bo'ladi?
- 10) Yuqoridagi algoritmda  $x$  ning oxirgi qiymatini toping.

### Quyidagi masalalarning algoritmlarini tuzing

- 1) 1 dan  $n$  gacha bo'lgan juft sonlar yig'indisini topish.
  - 2) 1 dan  $n$  gacha bo'lgan 6 karrali sonlar yig'indisini topish.
  - 3)  $n$  faktorialni hisoblash.
  - 4)  $n!!$  ni hisoblash
  - 5)  $a = (-1)^n(2n-8)$ ,  $n \in [-4;6]$ ,  $\Delta n=2$  bo'lsa, nechta manfiy had hosil bo'lishini aniqlang.
  - 6) Ixtiyoriy natural sonning bo'luvchilari sonini topish algoritmini tuzing?
  - 7) Raqamlar yig'indisi 9 ga teng bo'lgan uch xonali sonlarni va ularning yig'indisini toping.
  - 8)  $K$  ta raqamdan iborat butun son raqamlarning  $k$  darajalari yig'indisi shu son o'ziga teng bo'lsa, bu son Armstrong soni deyiladi. Masalan,  $153=1^3+5^3+3^3$ . 4 xonali Armstrong sonlarini topish al-g tuzing.
  - 9) Chapdan va o'ngdan bir xil o'qiladigan sonlar Palindrom sonlari deyiladi. Masalan, 121, 434,...
- Uch xonali Palindrom sonlarni topish algoritmini tuzing.
- 10) Sonning raqamlar yig'indisini topish algoritminiv tuzing.
  - 11) Agar natural son o'zidan boshqa bo'luvchilar yig'indisiga teng bo'lsa bunday son mukammal son deyiladi.  
Masalan,  $28=1+2+4+7+14$ . Uch xonali mukammal sonlarni toping.

### **Mustaqil ishlash uchun topshiriqlar.**

Quyidagi masalalar uchun algoritmlar tuzing.

1.  $K$  va  $N$  ( $N > 0$ ) butun sonlari berilgan.  $K$  sonini  $N$  marta chiqaruvchi algoritm tuzilsin.
2.  $A$  va  $B$  butun sonlari berilgan ( $A < B$ ).  $A$  va  $B$  sonlar orasidagi barcha butun sonlarni ( $A$  va  $B$  ni ham) chiqaruvchi va chiqarilgan sonlar sonini chiqaruvchi algoritm tuzilsin. ( $A$  va  $B$  ham chiqarilsin).
3.  $A$  va  $B$  butun sonlari berilgan ( $A < B$ ).  $A$  va  $B$  sonlar orasidagi barcha butun sonlarni ( $A$  va  $B$  kirmaydi) kamayish tartibida chiqaruvchi va chiqarilgan sonlar sonini chiqaruvchi algoritm tuzilsin.
4. Bir kg konfet narxi – haqiqiy son berilgan. 1, 2, ..., 10 kg konfet narxini chiqaruvchi algoritm tuzilsin.
5. Bir kg konfet narxi – haqiqiy son berilgan. 0.1, 0.2, ..., 0.9, 1 kg konfet narxini chiqaruvchi algoritm tuzilsin.
6. Bir kg konfet narxi – haqiqiy son berilgan. 1.2, 1.4, ..., 2 kg konfet narxini chiqaruvchi algoritm tuzilsin.
7.  $A$  va  $B$  butun sonlari berilgan ( $A < B$ ).  $A$  dan  $B$  gacha bo'lgan barcha butun sonlar ( $A$  va  $B$  ham kiradi) yig'indisini chiqaruvchi algoritm tuzilsin.

8.  $A$  va  $B$  butun sonlari berilgan ( $A < B$ ).  $A$  dan  $B$  gacha bo'lgan barcha butun sonlar ( $A$  va  $B$  ham kiradi) ko'paytmasini chiqaruvchi algoritmi tuzilsin.

9.  $A$  va  $B$  butun sonlari berilgan ( $A < B$ ).  $A$  dan  $B$  gacha bo'lgan barcha butun sonlar ( $A$  va  $B$  ham kiradi) kvadratlarining yig'indisini chiqaruvchi algoritmi tuzilsin.

10.  $N$  ( $N > 0$ ) butun son berilgan. Quyidagi yig'indini hisoblovchi algoritmi tuzilsin:  
 $S = 1 + 1/2 + 1/3 + \dots + 1/N$ .

11.  $N$  ( $N > 0$ ) butun son berilgan. Quyidagi yig'indini hisoblovchi algoritmi tuzilsin:  
 $S = N^2 + (N+1)^2 + (N+2)^2 + \dots + (2N)^2$ .

12.  $N$  ( $N > 0$ ) butun son berilgan. Quyidagi ko'paytmani hisoblovchi algoritmi tuzilsin:

$P = 1.1 * 1.2 * 1.3 * \dots$  ( $N$  ta ko'paytuvchi).

13.  $N$  ( $N > 0$ ) butun son berilgan. Quyidagi ifodani hisoblovchi algoritmi tuzilsin:  
 $S = 1.1 - 1.2 + 1.3 - \dots$

( $N$  ta qo'shiluvchi ishoralar almashib keladi. Shartli operatoridan foydalanmang).

14.  $N$  ( $N > 0$ ) butun son berilgan. Shu sonning kvadratini quyidagi mula asosida hisoblovchi algoritmi tuzilsin:

$N^2 = 1 + 3 + 5 + \dots + (2N-1)$ .

har bir qo'shiluvchidan keyin natijani ekranga chiqarib boring. Natijada ekranda 1 dan  $N$  gacha bo'lgan sonlarning kvadratlari chiqariladi.

15.  $A$  xaqiqiy va  $N$  ( $N > 0$ ) butun sonlari berilgan.  $A$  ning  $N$ -darajasini aniqlovchi algoritmi tuzilsin:  $A^N = A * A * \dots * A$

16.  $A$  xaqiqiy va  $N$  ( $N > 0$ ) butun sonlari berilgan. Bitta ssikldan foydalanib  $A$  ning 1 dan  $N$  gacha bo'lgan barcha darajasini chiqaruvchi algoritmi tuzilsin.

17.  $A$  xaqiqiy va  $N$  ( $N > 0$ ) butun sonlari berilgan. Bitta ssikldan foydalanib quyidagi  $A$  ning 1 dan  $N$  gacha bo'lgan barcha darajalarini chiqaruvchi va yig'indini hisoblash algoritmi tuzilsin:

$S = 1 + A + A^2 + \dots + A^N$ .

18.  $X$  xaqiqiy va  $N$  ( $N > 0$ ) butun sonlari berilgan. Bitta ssikldan foydalanib quyidagi  $X$  ning 1 dan  $N$  gacha bo'lgan barcha darajalarini chiqaruvchi va yig'indini hisoblovchi algoritmi tuzilsin:

$S = 1 - X + X^2 - X^3 + \dots + (-1)^N X^N$ .

Shartli operatoridan foydalanmang.

19.  $N$  ( $N > 0$ ) butun soni berilgan. Birdan  $N$  gacha bo'lgan natural sonlari ko'paytmasini chiqaruvchi algoritmi tuzilsin:  $N! = 1 * 2 * \dots * N$ .

1 dan  $N$  gacha bo'lgan natural sonlari ko'paytmasi  $N$  faktorial deyiladi.

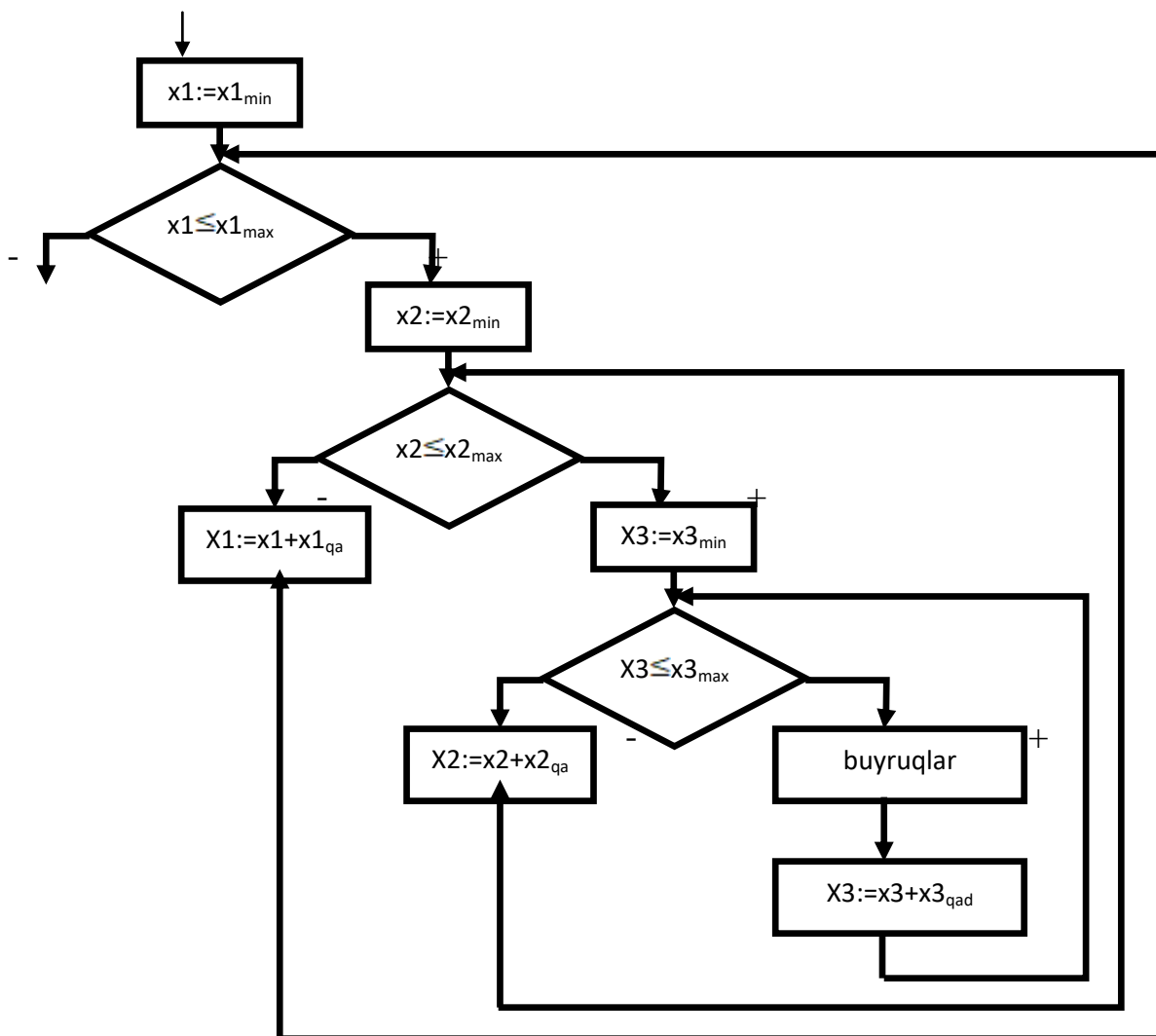


20.  $N$  ( $N > 0$ ) butun sonlari berilgan. Bitta ssikldan foydalanib quyidagi yig'indini hisoblash algoritmi tuzilsin:

$$S = 1! + 2! + 3! + \dots + N!$$

( $N!$  ifoda -  $N$  faktorial - 1 dan  $N$  gacha bo'lgan butun sonlari ko'paytmasini bildiradi:  $N! = 1 * 2 * \dots * N$ ).

### Ichma-ich joylashgan sikllar

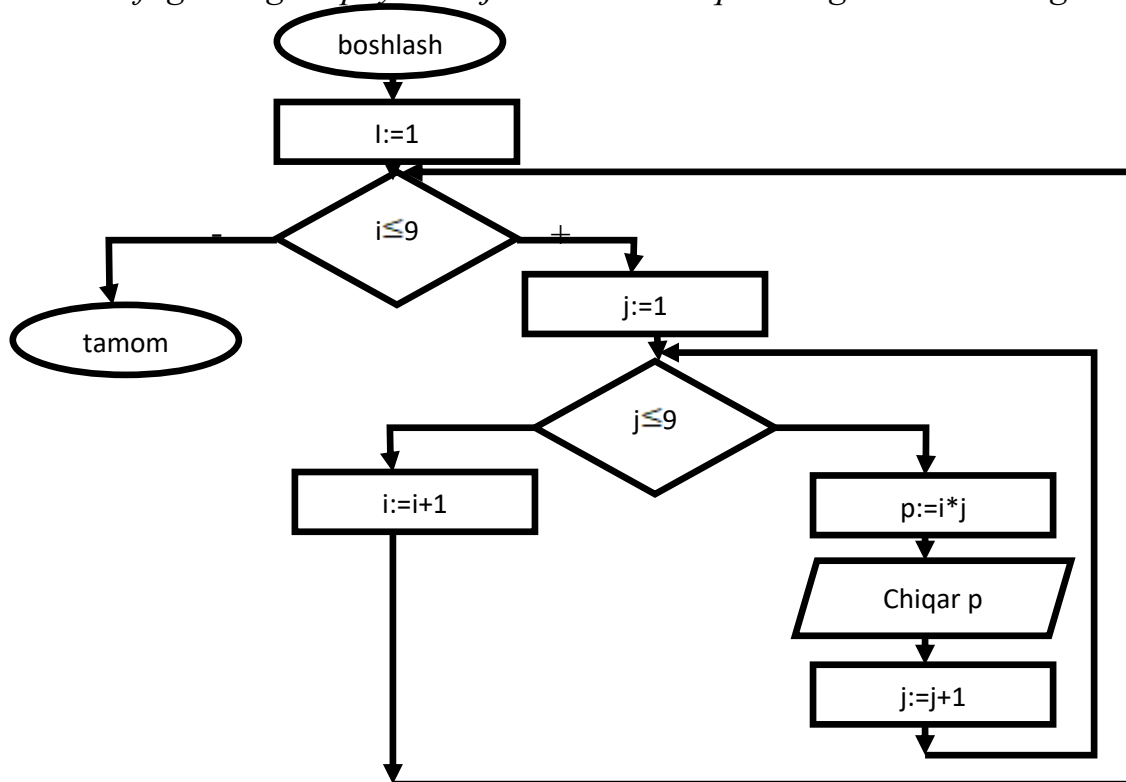


Buyruq quyidagi tartibda bajariladi:

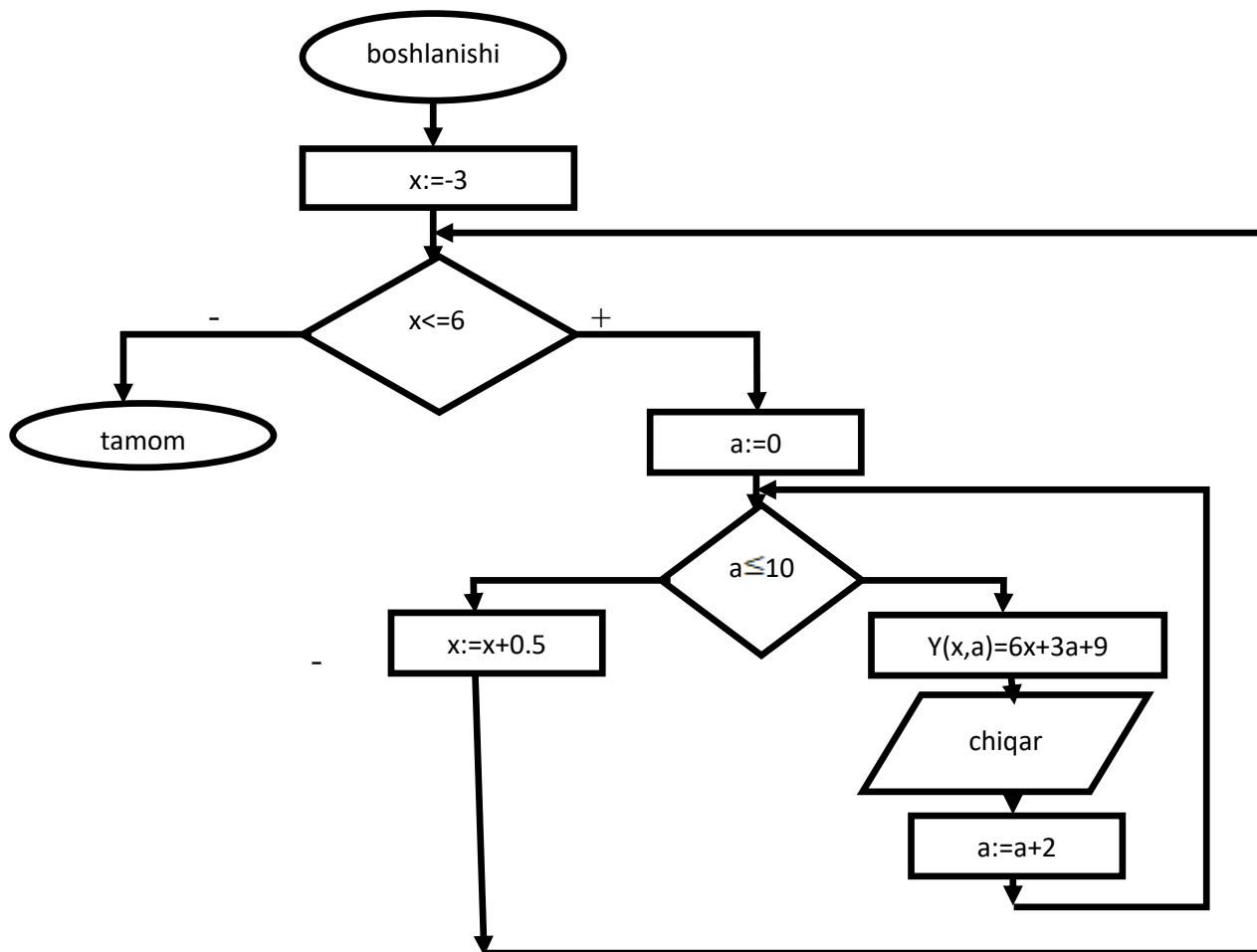
Tashqi sikl parametri o'zining dastlabki qiymatni oladi va oxirgi qiymatini bilan solishtiradi. Agar  $x1 \leq x1_{max}$  mantiqiy ifodaning qiymati rost bo'lsa, o'rta tsikl parametri o'zining dastlabki qiymatini oladi va oxirgi qiymati bilan solishtiradi. Agar  $x2 \leq x2_{max}$  mantiqiy ifodaning qiymati rost bo'lsa, ichki tsikl parametri o'zining dastlabki qiymatini oladi va oxirgi qiymati bilan solishtiradi. Agar  $x3 \leq x3_{max}$  mantiqiy ifodaning qiymati rost bo'lsa, buyruqlar bir marta

bajarilib, ishki tsikl parametri bir qadamga ortadi va  $x_3 \leq x_{3_{\max}}$  mantiqiy ifodaning qiymati yana tekshiradi. Agar u rost bo'lsa buyruqlar yana bir bor bajarilib, ichki tsikl parametri yana bir qadamga ortadi va hokazo ichki tsikldagi takrorlanishlar  $x_3 > x_{3_{\max}}$  bo'lguncha davom etadi. Shundan so'ng o'rta tsikl parametri bir qadamga ortadi va  $x_2 \leq x_{2_{\max}}$  mantiqiy ifodaning qiymati tekshiriladi. Agar u rost bo'lsa ichki tsikldagi takrorlanishlar to'liq qaytariladi (o'rta tsikl parametrining yangi qiymati uchun). Shundan so'ng o'rta tsikl parametri yana bir qadamga ortadi va hokazo o'rta tsikldagi takrorlanishlar  $x_2 > x_{2_{\max}}$  bo'lguncha davom etadi. Shundan so'ng tashqi tsikl parametri bir qadamga ortadi va  $x_1 \leq x_{1_{\max}}$  mantiqiy ifodaning qiymati tekshiriladi, agar u rost bo'lsa o'rta va ichki tsikllardagi takrorlanishlar to'liq qaytariladi (tashqi tsikl parametrining yangi qiymati uchun) va hokazo bu jarayon  $x_1 > x_{1_{\max}}$  bo'lguncha davom etadi. Shundan so'ng ichma – ich joylashgan tsikllardan keyingi buyruqlarga o'tiladi.

**Misol:** Pifagorning ko'paytirish jadvalini hosil qilish algoritmini tuzing.

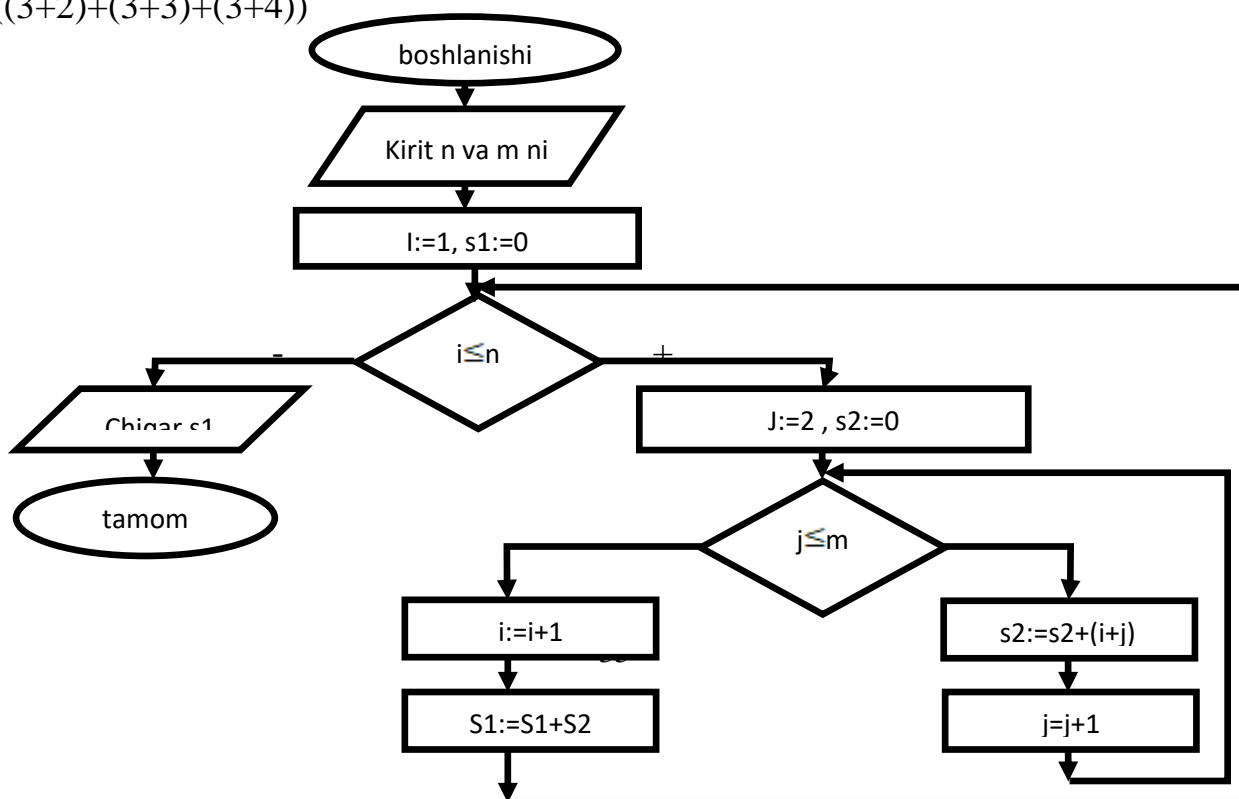


**Misol:**  $y=6x+3a+9$   $x \in [-3;6]$ ,  $\Delta x=0.5$ ,  $a \in [0;10]$ ,  $\Delta a=2$  funksiyaning qiymatini hisoblash algoritmini tuzing.  $\Delta$ -qadam



**Misol:**  $\sum_{i=1}^n \sum_{j=2}^m (i + j)$  ifodaning qiymatini hisoblash algoritmini tuzing . Bu yozuvni tushuntiramiz. Demak  $\sum$  belgi yig'indi (summa) belgisidir. Masalan,  $n=3$  va  $m=4$  bo'lsin.

$$\sum_{i=1}^3 \sum_{j=2}^4 (i + j) = ((1+2)+(1+3)+(1+4)) + ((2+2)+(2+3)+(2+4)) + ((3+2)+(3+3)+(3+4))$$



## Nazariy savollar

1. Ichma-ich joylashgan tsikllarning umumiy ko'rishini yozib, bajarilish tartibini ayting.

2.  $\prod_{i=1}^n \prod_{j=2}^m (i+j)$  ifodaning yoyilmasini yozing.

3.  $\prod_{i=1}^n \sum_{j=2}^m (i+j)$  ifodaning yoyilmasini yozing.

4. Uchta ichma-ich joylashgan tsikllarning bajarilish tartibi nimaga o'xshaydi?

5. To'rtta ichma-ich joylashgan tsikllarning tashqi ikkitasida 12 martadan takrorlanish, ichki ikkitasida 5 martadan takrorlanish ro'y bersa, buyruqlar nacha marta bajariladi? (j: 3600)

6. To'rtta ichma-ich joylashgan tsikllarning tashqi ikkitasida 12 martadan takrorlanish, ichki ikkitasida 5 martadan takrorlanish ro'y bersa va oxirida tsikllar parametrlari, ya'ni  $x+y+z+t$  ifodaning qiymati chiqarilsa, algoritm natijasini aniqlang. Barcha tsikl parametrlarining dastlabki qiymatlari va qadamlari 1 deb hisoblansin (j: 38)

## Mustaqil ishlash uchun topshiriqlar

Bunda  $\prod_{j=2}^m j$  ko'paytma belgisi.

1)  $\prod_{i=1}^n \prod_{j=2}^m (i+j)$  2)  $\prod_{i=1}^n \sum_{j=2}^m (i+j)$  3)  $\sum_{i=1}^n \prod_{j=2}^m (i+j)$

2) Bir va ikki xonali sonlar ichida o'z kvadratining oxirgi raqamlariga teng bo'lgan sonlarni aniqlang. Masalan,  $52=25$ ;  $252=625$ .

3) Berilgan sonning tub bo'luvchilarini aniqlang.

4) Dastlabki ikkita raqamlari yig'indisi uchinchi raqamiga qoldiqsiz bo'linadigan uch xonali sonlarni toping.

5) Nuqta koordinatalari  $x=(-1)^i(2i-j)$ ,  $1 \leq x \leq 8$ ,  $\Delta x=0.8$ ;  $y=(i-(-1)^j)$ ,  $-2 \leq y \leq 4$ ,  $\Delta y=0.6$ ; qonuniyat bilan o'zgaradi. Qaysi koordinata choraklariga nechtadan nuqta tushishini aniqlang.

## II-BOB. C++ DASTURLASH TILI, C++ DASTURLASH TILIDA TURLAR VA OPERATORLAR

### 2.1 C++ DASTURLASH TILI TARIXI VA UNING TARKIBI

*Reja:*

- 1.C++ tili tarixi;*
- 2. C++ tilining leksemmasi;*
- 3. C++ dasturlash tilidagi dasturning umumiy ko‘rinishi.*
- 4.C++ dasturlash tilida izohlardan foydalanish*

Butun dunyoda informatika va axborot texnologiyalarining rivojlanishi bilan bir qatorda dasturlash asoslari ham keng rivojlanib bormoqda. Dasturlash asoslari fanining asosi bo‘lmish dasturlash tillari ham keng ommalashib bormoqda. Birinchi EHM yaratilgan davrdan boshlab dasturlash ham rivojlanib bormoqda, dastlabki kompyuterlar AQSH da harbiy maqsadlar yo‘lida yaratilgan. Dastlab dasturchilar eng sodda mashina tilini o‘zida ifodalovchi kompyuter komandalari bilan ishlaganlar. Bu komandalar uzun qatorli nol va birlardan iborat bo‘lgan. Keyinchalik foydalanuvchi uchun tushunarli bo‘lgan assembler tili yaratiladi. Keyinchalik sekin asta yuqori bosqichli tillar yaratila boshlandi, yuqori bosqichli algoritmik tillarda dasturni foydalanuvchi tushunadigan ma’lum bir standartlar yordamida yaratiladi. Yuqori bosqichli algoritmik tillarda tuzilgan dasturni kompyuter tiliga maxsus interpretator va komplyatorlar yordamida o‘tkazilardi. Elektron hisoblash mashinalari avlodlari rivojlanib borishi bilan bir vaqtda dasturlash va dasturlash tillari ham keskin rivojlandi. Dasturlash tillarida tuzilgan dastur tarkibidagi bo‘yruqlarning soddaligi va kompyuter xotirasidan kam joy egallashi bilan katta farq qiladi. Yuqoridagi jarayonlarni hisobga olib dasturlashga ham talablar o‘zgarib bordi, nafaqat dasturlashni balki dastur yozish texnologiyasini ham o‘zgarishiga olib keldi. Dasturlash asoslari fani bu elektron hisoblash mashinalarining boshqaruvchi fan deb ham yuritish mumkin. Chunki elektron hisoblash mashinalarining asosi bo‘lmish dasturiy ta’minotlar ham insoniyat tomonidan ma’lum bir qoida va buyruqlar asosida yaratiladi Ayni vaqtda bir nechta dasturlash tillari mavjud ulardan keng qo‘llaniladigan va barcha imkoniyatlari keng bo‘ganlari **C++**, **Java**, **PHP** va hokozalar sanaladi. So‘ngi yillarda Java va C++ dasturlash tillari juda takomillashib, tobora ommalashib bormoqda. Mazkur tillardagi vositalar zamonaviy axborot texnologiyasining hamma talablarini o‘z ichiga olgan va unda dastur tuzuvchi uchun ko‘pgina qulayliklar yaratilgan.

Mavjud tillar katta 2 ta guruhga bo‘linadi:

1) Tabiiy tillar. Bu odamlar o`zaro muloqotda bo`ladigan tillar (o`zbek, qozoq, ingliz va hokazo).

2) Sun`iy tillar. Sun`iy tillar yaxshigina kodlangan bo`ladi. Shuning uchun masala dasturini masala kodi ham deyishadi.

Kodlash 2 xil maqsadda amalga oshiriladi:

a) katta hajmdagi ma`lumotlarni qisqacha ifodalash maqsadida.

Masalan, Samarqand davlat universitetining kodi - SamDU, Davlat test markazning kodi - DTM, oziq-ovqat mahsulotlari, yengil sanoat, og`ir sanoat, ta`lim dargohlarining nomlari ham ularning kodlardir. Odamlarning ismlari, hatto laqablari, masalan, xo`jayin, sarimsoq piyoz, baqaloq, lampa shisha, novcha va hokazolarning barchasi kodlangan ma`lumotlardir.

b) maxfiylikni saqlash maqsadida. Bunday usulda kodlangan ma`lumotlarni hamma ham tushunavermaydi. Faqat tor doiradagi odamlargina tushunadi. Maxfiy holda kodlashning ham har xil usullaridan foydalaniladi. Masalan, raqamlar orqali, hayvonlar nomlari orqali, qisqacha so`zlar orqali kodlanadi va hakoza. Bunday usulda kodlashdan josuslar, maxfiy xizmat vakillari ko`p foydalanishadi. Masalan, 16-obyekt yo`qoldi, chiroq yonmadi, mashina portladi, 2 ni olib tashlang va hokazo.

Demak, kodlash biror hajmga ega bo`lgan axborotning boshqa tarzda qisqacha ifodalanishidir.

Barcha dasturlash tillari ham sun`iy tillar jumlasiga kiradi va ularning ijrochisi kompyuterlardir. Ularda ham har bir buyruq qandaydir fikrning kodi hisoblanadi. Masalan,  $a \% 2 = 0$  yozuvini kompyuter "a juft son" deb,  $\text{abs}(x)$  yozuvini esa  $|x|$  deb tushunadi va hokazo. Shuning uchun dasturlash tillarini yaratish uchun qilingan dastlabki harakatlardayoq keng omma bu tillarni tezroq o`zlashtirib olishi uchun biror tabiiy tilning so`zlaridan mumkin qadar kam foydalanishga harakat qilingan.

Biz yuqori darajali dasturlash tillaridan biri bo`lgan C++ dasturlash tili bilan tanishamiz. C++ dasturlash tili 1980 yillar boshida Byarne Straustrup tomonidan C tiliga asoslangan tarzda tuzildi. C dasturlash tili 1972-yil Denis Pitch va Brayan Kornegilar tomonidan yaratilgandi. C++ juda ko`p qo`shimchalarni o`z ichiga olgan, lekin eng asosiysi u obyektlar bilan dasturlashga imkon beradi. Dasturlarni tez va sifatli yozish hozirgi kunda katta ahamiyatga ega. Grafik interfeysga ega dasturlarni yaratish uchun ob`yektga mo`ljallangan dasturlash tayanch baza hisoblanadi. Ob`yektga mo`ljallangan dasturlashning asosiy maqsadi berilganlar va ular ustida amal bajaruvchi proseduralarni yagona ob`yekt sifatida qarashidir.

## C++ tilining leksemmasi

C++ tiliga boshqa dasturlash tillariga nisbatan ko'pgina yangiliklar kiritilgan bo'lib, tilning imkoniyati yanada kengaytirilgan. C++ dasturlash tili ham boshqa dasturlash tillari kabi o'z alfavitiga va belgilariga ega.

Tilning mavjud alfavit va belgilariga quyidagilar kiradi:

1. Katta va kichik lotin alfaviti harflari;
2. Raqamlar - 0,1,2,3,4,5,6,7,8,9;
3. Maxsus belgilar: " {} | [] () + - / % \ ; ' : ? <=> \_ ! & ~ # ^ . \*

Tilning alfavit belgilaridan tilning leksemalari shakllantiriladi:

1. Identifikatorlar;
2. Kalit (xizmatchi yoki zahiralangan) so'zlar;
3. O'zgarmaslar;
4. Amallar;
5. Ajratuvchilar.

C++ dasturlash tilida tuzilgan dasturlarda izohlar istalgan joyda berilishi mumkin. Izohlar ikki turda bo'ladi ular *satriy* va *blokli* ko'rinishlarda bo'ladi. Satriy izohlar uchun "//" belgi ishlatiladi, blokli izohlar uchun "/\*", "\*\*/" belgilari ishlatiladi.

**Dastur-** dasturlash tilida ma'lum bir masala yoki muommoni hal etish uchun yozilgan chekli sondagi bo'yruqlar ketma- ketligi.

**Identifikator-** dastur tarkibidagi o'zgaruvchi, o'zgarmas, funksiya, prosedura va ob'yektlarni belgilashda ishlatiladigan nomlardir.

**Kalit so'zlar-** dastur tarkibidagi malum bir vazifani bajaruvchi so'zlarga aytiladi.

**O'zgarmas-** dastur bajarilish vaqtida o'z qiymatini o'zgartirmaydigan identifikatorlarga aytiladi.

**Amal-** dastur tarkibida yoziladigan matematik yoki mantiqiy amallarga aytiladi.

**Ajratuvchi-** dastur tarkibidagi buyruq va ifodalarni bir biridan ajratilishiga aytiladi.

C++dasturlash tilida ishlatiladigan belgili o'zgarmaslar jadvali quyidagicha.

Belgining yozilishi	Belgining nomlanishi va unga mos amal
\\	Teskari yon chiziqni chop etish
\'	Apostrofni chop etish
\"	Qo'shtirnoqni chop etish
\?	So'roq belgisi
\a	Tovush signalini berish
\b	Kursorni 1 belgi o'rniga orqaga qaytarish

\f	Sahifani o'tkazish
\n	Qatorni o'tkazish
\r	Kursorni ayni qatorning boshiga qaytarish
\t	Kursorni navbatdagi tabulyatsiya joyiga o'tkazish
\v	Vertikal tabulyatsiya (pastga)

C++ tilida yaratilgan dasturni komplyatsiya qilgandan so'ng .cpp kengaytmali bajaruvchi fayl yaratiladi, agar ob'yektli fayl qo'shilsa, u holda .obj kengaytmali fayl yaratiladi.

### C++ dasturlash tilidagi dasturning umumiy ko'rinishi quyidagicha:

#include <iostream.h>	// sarlavha faylni qo'shish
int main ()	// bosh funksiya tavsifi
{	// blok boshlanishi
return 0;	// funksiya qaytaradigan qiymat
}	// blok tugashi

C++ dasturlash tilida dasturning umumiy tuzilishini quyidagi oddiy dastur yordamida tushuntiramiz:

1. #include <iostream.h>
2. using namespace std;
3. int main( );
4. {int a, b, c, s;
5. a=5; b=12; s=a+b;
6. cout << "a=" <<a<< " b=" <<b<< " s=" <<s;
7. system ("pause");
8. return 0;
9. }

C++ dasturlash tilida har qanday dastur 1-satr bilan boshlanadi. Bunda **#include** (o'z ichiga olmoq) so'zi ma'lumotlarni kiritish va chiqarish (o'qish va yozish) funksiyalari va ularning o'zativchilari e'loni joylashgan **<iostream.h>** sarlavha faylini dasturga qo'shadi. **<iostream.h>** yozuvi quyidagicha ma'noni bildiradi: i (input – kirishi (o'qish)), o (output – chiqarish (yozish)), stream – oqim (ya'ni ma'lumotlar oqimi), h (head – sarlavha).

Juda ko'pchilik yangi komplyatorlar **#include <iostream.h>** o'rniga **#include <iostream>** sarlavha faylidan foydalanish imkonini beradi. Faqat undan keyin **using namespace std;** satrini qo'shish kerak.

2-satrdan **main( )** funksiyasi berilgan bo'lib, bu funksiya ham (main – bosh, asosiy demakdir), albatta, bo'lishi shart va dastur shu funktsiyani bajarish bilan o'z ishini boshlaydi. Uning oldidagi int so'zi dastur ijro etilgach, natija butun son sifatida qaytarilishini bildiradi.



C++ dasturlash tilining ba'zi kompyatorlari int main( ) funksiya prototipi e'lon qilingan bo'lsa ham bu funksiya prototipi e'lon qilinmaganligi haqida xabar beradi. Bunday holda int main( ) dan oldin int main( ); funksiya e'lonini ham qo'shish kerak:

```
#include <iostream.h>;
int main( );
int main
{ ... }
```

Dasturdagi boshqa funksiyalarga dasturning qayeridadir murojaat buyrug'i bo'lsagina ular ishga tushadi. Lekin main( ) asosiy funksiyasiga esa dastur ishga tushgan zahoti murojaat bo'ladi.

4 va 9 - satrlarda { va } qavslari keltirilgan bo'lib, ular blok belgisidir. C++ dasturlash tilida, tarmoqlanish, takrorlanish kabi murakkab buyruqlarga taalluqli buyruqlar ham shunday blok belgilari ichiga olib yoziladi. Bu yerda esa bu belgilar dastur boshlanishi va tugashini bildiradi desak ham bo'ladi yoki dastur tanasining chegaralari yoki main( ) funksiyasi bloke (tanasi) deyish ham mumkin. Ammo bu belgilersiz dastur ishlamaydi.

4-satrdagi int a, b, c, s; yozuvi a, b, c, s o'zgaruvchilar e'loni deyiladi. Bunda, albatta, ularning turi ko'rsatilgan bo'lishi shart va ular, albatta, bir-birlaridan (,) belgisi bilan ajratib yoziladi.

5-satrdagi qiymat berish buyrug'i (=) yordamida a, b, s o'zgaruvchilarga dastlabki qiymatlar berilgan.

a,b,s larga ularni e'lon qilishda ham int a=5; b=12; s=a+b; kabi dastlabki qiymatlarini berish (initsializatsiya qilish) mumkin edi.

Ko'rinib turibdiki, C++ dasturlash tilidagi buyruqlar bir-birlaridan (;) belgisi bilan ajratib yoziladi. Faqat sarlavha faylini e'lon qilingandan keyin va bosh funksiyamain( ) dan keyin (;) qo'yilmaydi.

6-satrga <iostream.h> sarlavha faylida aniqlangan cout operatoridan foydalanilgan bo'lib, undan dasturning xotiradagi natijasini ekranga chiqarish maqsadida foydalaniladi. Bu operatoridan keyin qiymati ekranga chiqarilayotgan barcha o'zgaruvchilar hech qanday tinish belgilarisiz "<<" belgilari bilan ajratib yoziladi. Agar matn(satr) ekranga chiqarilishi kerak bo'lsa, u qo'shtirnoq ichiga olib yoziladi va C++ tomonidan qayta ishlanmasdan, qanday yozilgan bo'lsa, shundayligicha chop etiladi. Bizning misolimizda "a=", "b=", "s=" yozuvlari matnlardir.

8-satrdagi `return 0;` yozuvi `main( )` funksiyasi o'z ishini tugatganligini bildiruvchi (`return` – qaytish) operatoridir. Odatda dastur o'z ishini muvaffaqiyatli yakunlasa, funksiyaga nol qiymati qaytariladi.

Dev C++ variantida esa `return 0;` buyrug'idan oldin `System ("pause");` buyrug'i qo'shilmasa, dastur natijasini ko'rib ulgurmasdan o'chib ketadi. Ushbu kutish buyrug'i `Esc` tugmasi bosilguncha natijani ekranda ushlab turadi. C++ dasturlash tilida tuzilgan dastur yangi fayl sifatida saqlab qo'yiladi va bu faylning kengaytmasi **.cpp** bo'ladi.

Shuni qayd etish kerakki, C++ dasturida albatta `main()` funksiyasi bo'lishi shart va dastur shu funksiyani bajarish bilan o'z ishini boshlaydi. Dastur tanasidagi barcha amal va bo'yruqlar `{ }` bloklar orasiga olinib yoziladi. Dastur `"}`" belgisi bilan yakunlanishi shart. Agar dasturda qism dasturlardan foydalanilayotgan bo'lsa, ularning nomlari va haqiqiy parametrlari keltiriladi. So'ngra dasturning asosiy buyruqlari yoziladi. Agar buyruqlar murakkab bo'lsa, ular alohida `"{ }`" belgilari orasiga olingan bo'lishi kerak. C++ tilida dasturning asosi bo'lmish buyruqlar kichik harflar bilan yoziladi.

Buyruqlar nuqta-vergul bilan `(;)` yakunlanadi. Buyruqlar bir qator qilib yozilishi ham mumkin. C++ dasturlash tilida dastur funksiya va funksiyalardan tashkil topadi. Agar dastur bir nechta funksiyalardan tashkil topgan bo'lsa, bir funksiyaning nomi `main` deb nomlanishi shart. Dastur aynan `main` funksiyasining birinchi operatoridan boshlab bajariladi.

### **C++ dasturlash tilida izohlardan foydalanish**

Kimdir tomonidan tuzilgan dasturni boshqa bir dasturchi tomonidan, ayniqsa boshlovchilar tomonidan tushunib olish qiyin bo'ladi. Agar dastur yetarlicha murakkab tuzilishga ega bo'lsa, ma'lum vaqtdan keyin shu dasturning muallifi ham tushunishga qiynalishi mumkin. Bunday vaziyatlar uchun C++ dasturlash tilida dasturning ayrim satrlarini tushuntirish matnlari bilan ta'minlash imkoniyati mavjud. Tushuntirish matnlari esa izohlar tarzida yoki chop etish operatorida qo'shtirnoq (`"`) ichiga olib yoziladi.

C++ dasturlash tilida izohlarning ikki xil ko'rinishidan foydalaniladi:

ikkita bo'lish belgisidan foydalanilsa (`//`), bitta fizik satr shu belgilardan to satr oxirigacha izohga aylanadi;

dasturning ixtiyoriy sondag'i satrlari `/*` va `*/` belgilari ichiga olinsa, o'sha satrlari izohga aylanadi.

```
#include <iostream.h>
using namespace std;
int main( )
```

```

{
int a, b, c; // a, b, c lar butun turli o'zgaruvchilar
cout<<" a va b larning qiymatlarini kiriting";
cin>>a>>b; // a va b larning qiymatlari kiritilishi shart
c=a+b; // c o'zgaruvchiga a va b larning yig'indisining qiymati beriladi
cout << "c=" << c ; // c ning qiymati chop etiladi
// a ning va b ning qiymatlarini kiritishni unutmadingizmi?
/* cout <<" a ning va b ning qiymatlari chop etilmaydi";
cout << "\n Chunki bunga hojat yo'q";*/
return 0; }

```

Bunda to'rt marta satrli izohlardan, bir marta umumiy izohdan foydalanildi, ikki marta chop etish operatoridan foydalanib, tushuntirish matnlari berildi.

Izohlar C++ dasturlash tili tomonidan ijro etilmaydi, unda xohlaganicha tushuntirish matnlarini yozish mumkin. Ular faqat dastur tarkibida dasturchiga ko'rinib turadi. Izohlarning ekrandagi ko'rinishi ham dasturlarning asosiy qismidan rangsizligi bilan ajralib turadi. Ammo chop etish operatoridan foydalanib tushuntirish matnlari berilsa chop etish operatori(cout) ijro etiladi. Faqat izohlar ijro etilmaydi.

*Izohlardan nafaqat tushuntirish matnlari sifatida, balki dasturning ma'lum sonidagi buyruq yoki satrlarini ijro etmaslik maqsadida ham ko'p foydalaniladi.*

## Nazariy savollar

1. Dasturlash tarixi?
2. C++ dasturlash tarixi?
3. C++ dasturlash tilining alfaveti va belgilari?
4. C++ dasturlash tilining leksimmalari va ularning ta'riflari?
5. C++ dasturlash tilining o'zgaruvchilarning belgilar jadvalini yozing?
6. C++ dasturlash tilidagi dasturning umumiy ko'rinishi?
7. Komplyatsiya qilingan fayl kengaytmalari?
8. C++ dasturlash tilida necha xil ko'rinishda izohlardan foydalanish mumkin?
9. Nima uchun dastur oxirida **return 0;** buyrug'i yoziladi?
10. Dasturda har bir buyruq qanday yakunlanadi?
11. `<iostream.h>` yozuvi qanday ma'noni bildiradi?
12. `int main()` funksiyasi qanday funksiya. C++ tilida qanday dastur tarkibida bu funksiyadan foydalanmaslik mumkin?
13. C++ dasturlash tilida `{...}` belgilari qanday belgilar va ular oralig'ida qanday ma'lumotlar yoziladi?

14. C++ dasturlash tilida yozilgan dasturning qayerlarida vergul(,) va nuqtali vergullar(;)dan foydalaniladi?

15. Dastur tarkibida return 0; va system("pause"); buyruqlaridan qanday maqsadlarda foydalaniladi?

16. Dastur tarkibida izohlardan qanday maqsadlarda, qaerlarda va qanday usullarda foydalanish mumkin?

17. Dasturning bir yoki bir nechta buyruqlarini ijro ettirmaslik mumkinmi?

## 2.2 C++ DASTURLASH TILI TARKIBIDAGI O'ZGARUVCHI TURLARI VA O'ZGARMASLAR

### Reja:

1. Identifikatorlar va kalit so'zlar;
2. C++ tilida o'zgaruvchi va uning turlari;
3. C++ tilida o'zgaruvchi va o'zgarmaslarni e'lon qilish;
4. Turni boshqa turga keltirish.

### Identifikatorlar va kalit so'zlar

Matematikada bo'lgani kabi C++ dasturlash tilida ham, dastur tarkibida foydalaniladigan o'zgaruvchilar va o'zgarmaslar o'z nomiga ega bo'lishi kerak. Bu nomlar matematikada lotin alifbesining bir, ikkita harflaridan tashkil topadi. C++ dasturlash tilida esa o'zgaruvchi va o'zgarmaslarning nomlari ketma-ket 255 tagacha bo'lan belgilardan tashkil topadi va ular identifikatorlar deyiladi. Identifikator tarkibida foydalaniladigan belgilar deyilganda faqat lotin alifbesining katta va kichik harflari, raqamlar va tag chiziqcha( \_ ) tushuniladi.

Shuni unutmaslik kerakki, identifikatorlarni yozishga katta va kichik harflar farqlanadi.

Masalan: **romb** va **Romb**; **n12** va **N12**; **aylana\_raduisi** va **Aylana\_Radiusi** kabi identifikatorlar boshqa-boshqa identifikatorlardir.

Bundan tashqari C++ dasturlash tilining kalit so'zlari, funksiya nomlari obyektlar nomlari ham identifikatorlar hisoblanadi.

Agar identifikator ikki yoki undan ortiq so'zlardan tashkil topgan bo'lsa ham ular bir-birlaridan probel bilan ajratilmaydi. Ular qo'shib yoziladi. Agar ajratib yozish kerak bo'lsa tag chiziqcha ( \_ )dan foydalaniladi.

Masalan: `tenglama_yechimi` yozuvi identifikatoridir. C++ dasturlash tilining kalit soʻzlari deyilganda buyruqlarni tashkil qilishda foydalaniladigan ingliz tilida yozilgan operatorlar (yordamchi soʻzlar) tushuniladi.

Masalan, **if, else, return, while**, ..... kabi soʻzlar kalit soʻzlardir.

C++ dasturlash tilining operatorlaridan identifikator sifatida foydalanib boʻlmaydi.

**x, a1, b819, doira, maktab, ptk\_meb, Rels89** larning har biri identifikator boʻla oladi. Lekin,

<b>Лобар</b>	<b>3ka</b>
<b>dars boʻldi</b>	<b>qovun, tarvuz</b>
<b>while</b>	<b>48</b>
<b>if for</b>	<b>a/bc</b>

lar identifikator boʻla olmaydi.

### **C++ tilida oʻzgaruvchi va uning turlari**

C++ dasturlash tili tarkibidagi barcha oʻzgaruvchilari qandaydir turlarga mansub boʻladi, chunki oʻzgaruvchi eʼlon qilingandan soʻng kompyuter xotirasidan oʻzgaruvchi uchun joy ajratiladi. Dasturlash tillarida dastur bajarilishi paytida qandaydir berilganlarni saqlab turish uchun oʻzgaruvchilar va oʻzgaruvchilardan foydalaniladi. Oʻzgaruvchi-dastur obyekt boʻlib, xotiradagi bir nechta yacheykalarni egallaydi va berilganlarni saqlash uchun xizmat qiladi. Dastur ishlashi mobaynida qiymatlari oʻzgarishi mumkin boʻlgan identifikatorga oʻzgaruvchilar deyiladi. Oʻzgaruvchi oʻz nomiga, oʻlchamiga va boshqa atributlarga, xususiyatlarga ega boʻladi. Oʻzgaruvchilarni ishlatish uchun ular albatta eʼlon qilinishi kerak. Eʼlon natijasida oʻzgaruvchi uchun xotiradan qandaydir soha zahiralanadi, soha oʻlchami esa oʻzgaruvchining aniq turiga bogʻliq boʻladi. C++ tilida oʻzgaruvchi eʼloni uning turini aniqlovchi kalit soʻzi bilan boshlanadi va '=' belgisi orqali boshlangʻich qiymat beriladi (shart emas). Bitta kalit soʻz bilan bir nechta oʻzgaruvchilarni eʼlon qilish mumkin. Buning uchun oʻzgaruvchilar bir-biridan ',' belgisi bilan ajratiladi. Eʼlonlar ';' belgisi bilan tugaydi. Oʻzgaruvchi nomi 255 belgidan oshmasligi kerak. Oʻzgaruvchilarni eʼlon qilish dastur matnining istalgan joyida amalga oshirilishi mumkin. Dastur tarkibida qandaydir oʻzgaruvchilardan foydalanish uchun, ular, albatta, eʼlon qilinishi kerak. *Eʼlon qilishdan maqsad ushbu oʻzgaruvchi uchun EHM xotirasidan joy ajratishdir.*

Oʻzgaruvchini eʼlon qilishda, albatta, uning turi, nomi, dastlabki qiymati koʻrsatiladi (yaʼni initsializatsiyalanadi. Ammo initsializatsiyalanmasdan ham eʼlon qilinishi mumkin). Yaʼni initsializatsiyalashni oʻzgaruvchi eʼlonida emas, balki dastur ichida ham amalga oshirish mumkin.

Masalan: int a=5, b=10, c=43; initsializatsiyalangan holda e`lon qilish;  
int a, b, c; initsializatsiyalamasdan e`lon qilish;

Bunda int – tur nomi bo`lib, integer (butun) so`zining qisqartirilgan shakli. Demak, a, b, c lar faqat butun qiymatlarnigina qabul qilishi mumkin. Boshqa turli qiymatlarni qabul qila olmaydi. Masalan, haqiqiy qiymat berilsa, haqiqiy sonning kasr qismi tashlab yuboriladi.

Har xil tur uchun xotirada har xil joy ajratiladi. Hatto bitta tur uchun har xil razryadli mashinalarda har xil hajmdagi joylar ajratilishi mumkin.

Yuqoridagi misollarda ko`rsatilganidek, bitta turdan foydalanib, bir nechta o`zgaruvchilarni e`lon qilish mumkin va ular bir-birlaridan (,) belgisi bilan ajratib yoziladi hamda (;) belgisi bilan yakunlanadi.

**Butun son turlari.** Butun son qiymatlarni qabul qiladigan o`zgaruvchilar. Butun tur int, shorint, longint, long kalit so`zlari bilan aniqlanadi.

**Belgi turi.** Belgi turidagi o`zgaruvchilar char kalit so`zi bilan beriladi va ular o`zida belgining ASCII kodini saqlaydi. Belgi turidagi qiymatlar nisbatan murakkab bo`lgan tuzilmalar – satrlar, belgilar massivlari va hokazolarni hosil qilishda ishlatiladi.

**Haqiqiy son turi.** Haqiqiy son qiymatlarni qabul qiladigan o`zgaruvchilar. Bu turdagi o`zgaruvchi uchun xotiradan 4 bayt joy ajratiladi va ular float, double so`zlari bilan aniqlanadi.

**Mantiqiy tur.** Bu turdagi o`zgaruvchi bool kalit so`zi bilan e`lon qilinib, xotiradan 1 bayt joy egallaydi va 0 (false, yolg'on) yoki (true, rost) qiymat qabul qiladi. Mantiqiy tur o`zgaruvchilar qiymatlar o`rtasidagi munosabatlarni ifodalaydigan mulohazalarni rost (true) yoki yolg'on (false) ekanligi tavsifida qo`llaniladi va ular qabul qiladigan qiymatlar matematik mantiq qonuniyatlariga asoslanadi. Mantiqiy mulohazalar ustida uchta amal aniqlangan:

C++ dasturlash tilida o`zgaruvchi turlari va qiymatlari jadvali quyidagicha:

Tur nomi	Baytlardagi o`lchami	Qiymat chegarasi
<b>Butun tur</b>		
<b>Bool</b>	<b>1</b>	
<b>Unsigned short int</b>	<b>2</b>	<b>0..65535</b>
<b>Short int</b>	<b>2</b>	<b>-32768..32767</b>
<b>Unsigned long int</b>	<b>4</b>	<b>0..42949667295</b>
<b>Long int</b>	<b>4</b>	<b>-2147483648..2147483647</b>
<b>int(16 razryadli)</b>	<b>2</b>	<b>-32768.. 32767</b>

<b>Int (32 razryadli)</b>	<b>4</b>	<b>-2147483648..2147483647</b>
<b>Unsigned int (16 razryadli)</b>	<b>2</b>	<b>0..65535</b>
<b>Unsigned int (32 razryadli)</b>	<b>4</b>	<b>0..42949667295</b>
<b>Belgili tur</b>		
<b>Unsigned char</b>	<b>1</b>	<b>0..255</b>
<b>Char</b>	<b>1</b>	<b>-128.. 127</b>
<b>Haqiqiy tur</b>		
<b>Float</b>	<b>4</b>	<b>1.2E-38..3.4E38</b>
<b>Double</b>	<b>8</b>	<b>2.2E-308..1.8E308</b>
<b>Long double(32 razryadli)</b>	<b>10</b>	<b>3.4e-4932..-3.4e4932</b>
<b>Void</b>	<b>2 ёки 4</b>	<b>-</b>

### C++ tilida o'zgaruvchi va o'zgarmaslarni e'lon qilish

C++ dasturlash tilida o'zgaruvchilarni tasvirlash quyidagicha.

**<tur> <o'zgaruvchi>;**

**Misol:** Butun sonlarni tasvirlash.

```
#include <iostream.h>
int main ()
{   int a,b;           // a, b butun sonni e'lon qilish
    int m=12;         //m butun sonini boshlang'ich qiymatini berish
    return 0;
}
```

**Misol:** Haqiqiy sonlarni tasvirlash.

```
#include <iostream.h>
int main ()
{   double a,b;       // a, b haqiqiy sonni e'lon qilish
    double m=12;      //m haqiqiy sonini boshlang'ich qiymatini berish
    return 0;
}
```

Dasturlash tillarida dastur bajarilishi vaqtida qiymati o'zgarmaydigan identifikatorlar **o'zgarmaslar** deyiladi. C++ tilida o'zgarmas (cons) – bu fiksirlangan sonni, satrni va belgini ifodalovchi leksema hisoblanadi. Kompilyator

o'zgarasni leksema sifatida aniqlaydi, unga xotiradan joy ajratadi, ko'inishi va qiymatiga (turiga) qarab mos guruhlariga bo'ladi.

C++ dasturlash tilida o'zgaraslarni tasvirlash quyidagicha.

**const <o'zgaras>=<qiymat>;**

Dasturni ijro etish jarayonida o'z qiymatini o'zgartirmaydigan kattaliklar (identifikatorlar) o'zgaraslar deyiladi. O'zgaruvchilarni initsiyalizatsiya qilmasdan (ya'ni boshlang'ich qiymatini bermasdan) e'lon qilish mumkin. Lekin o'zgaraslar, albatta, initsiyalizatsiyalanishi shart va bu qiymatni, dasturni bajarish davomida o'zgartirib bo'lmaydi.

O'zgaraslar ham o'zgaruvchilar kabi oldindan e'lon qilinadi. Ularni e'lon qilishdan maqsad ham bu o'zgaraslar uchun xotiradan joy ajratishdir.

O'zgaraslarni e'lon qilish uchun **const** kalit so'zidan foydalaniladi, undan keyin o'zgaras turi va nomi yozilib, yuqorida qayd qilganimizdek, dastlabki qiymati ko'rsatiladi va bu qiymat fiksirlangan qiymat hisoblanadi. Sonlardan tashqari belgili o'zgaraslar tutuq belgisi (apostrof) ichiga olib yoziladi. Misol tariqasida quyidagi dasturni tahlil qiling:

```
#include <iostream.h>
using namespace std;
int main( )
{
int n,m,k=5; int const x=3,y=12;
n=k*2;m=x+n;k=n+y;
cout <<"n="<<n<<" m="<<m<<" k="<<k<<"\n';
system ("pause");
return 0;
}
```

Dastur natijasi quyidagicha bo'ladi:

n=10 m=13 k=22

Manfiy o'zgaraslarni ham aslida ishorasiz o'zgaras deb qaraymiz, faqat unga ishorani o'zgartirish amali qo'llangan deb tushunamiz. Masalan, -3 o'zgarasni ishorasiz 3 soniga minus ishora qo'yilgan yoki 3 ishorasiz soni -1ga ko'paytirilgan deb tushunish kerak.

**Misol:** O'zgaraslarni tasvirlash.

```
#include <iostream.h>
int main ( );
```



```
{   const a=45;           // a o'zgarmasni e'lon qilish
   return 0;
}
```

### Turni boshqa turga keltirish

Bunda dastur tanasida o'zgaruvchining turini boshqa turga o'zgartirish haqida gap boradi va bunday imkoniyat C++ dasturlash tilida mavjud.

O'zgaruvchi turini boshqa turga keltirishning oshkor va oshkormas usullari mavjud.

O'zgaruvchi turni boshqa turga oshkor keltirish uchun yangi tur nomi oshkor yozib ko'rsatiladi. Masalan, quyidagi dasturda bu hol namoyish qilingan:

```
#include <iostream.h>
using namespace std;
int main( )
{
  int a=19, b;
  float x=36.035, y=7.41;
  a=(int)x; // oshkor holda haqiqiy turli x o'zgaruvchining butun turiga
o'tkazilmoqda;
  b=y; // nooshkor holda y o'zgaruvchi haqiqiy turdan butun turga
o'tkazilmoqda;
  cout <<"x="<<x<<" a="<<a<<"\n";
  cout <<"y="<<y<<" b="<<b<<endl;
  system ("pause >> void");
  return 0;
}
```

Dastur natijasi quyidagicha bo'ldi:

```
x = 36.035    a = 36
y = 7.41     b = 7
```

Ikkinchi cout operatorida endl; kalit so'zi qo'shildi. Bu kalit so'z, albatta, shart bo'lmagan operator bo'lib, satrni chop etish tugaganligini bildiradi. Ya'ni endl – end line – satr tugashi demakdir. Turni o'zgartirishni quyidagicha ham amalga oshirish mumkin.

```
#include <iostream.h>
using namespace std;
int main( )
{
  int a = 19, b;
```

```

float x = 36.035, y = 7.41;
cout << "a=" << (int) x << "\n";
cout << "b=" << (int)y;
system ("pause");
return 0;
}

```

Dastur natijasi quyidagicha bo'ldi:

```

a = 36
b = 7

```

Demak, turni boshqa turga keltirish uchun dastlab o'zgaruvchi initsializatsiya qilinadi, so'ngra o'tkazilayotgan turli o'zgaruvchi tomonidan o'zlashtiriladi. Yoki ikkinchi dasturda ko'rsatilganidek bu ishni chop etish jarayonida ham amalga oshirish mumkin. Faqat turni o'zgartirishda bitta muammo paydo bo'lishi mumkin. Agar initsializatsiya qilingan sonning xotirada egallagan joyi o'zlashtirilayotgan o'zgaruvchi uchun ajratilgan joydan katta bo'lsa, noto'g'ri natijaga ega bo'lish mumkin.

### Nazaorat savollari

1. C++ dasturlash tilida o'zgaruvchi va uni e'lon qilish?
2. C++ dasturlash tilida o'zgaruvchi va uni e'lon qilish?
3. C++ dasturlash tilida o'zgaruvchi turlari va ularning ta'riflari?
4. C++ dasturlash tilida o'zgaruvchini e'lon qilishga xizmat qiluvchi kalit so'zlar?
5. O'zgaruvchilarni qayerda va qanday e'lon qilish tartibini ayting.
6. `int x,y,z,n,d,f; int r;g;y;u;I; int(k,j,l,o,iw,der,sx,);` yozuvlarida yo'l qo'yilgan xatolarni aniqlang.
7. Kalit so'zlar qanday so'zlar?
8. O'zgaruvchi va o'zgaruvchilar qanday miqdorlar, ular qayerda va qanday e'lon qilinadi?
9. O'zgaruvchi va o'zgaruvchilarni e'lon qilishdan maqsad nima?
10. C++ dasturlash tilida o'zgaruvchilar va o'zgaruvchilar qanday farq qiladi?
11. O'zgaruvchilar qanday kalit so'z yordamida qanday e'lon qilinadi?
12. O'zgaruvchilarning qiymatlari faqat son ko'rinishida bo'ladimi?
13. Identifikator nima va u qanday yoziladi?
14. Identifikatorlarni yozishda qanday va nechta belgilardan foydalanish mumkin?
15. `for; 25calom; olma; uraa!; yosh bola; 423;` larning qaysilari identifikator bo'la oladi?
26. Turlarni o'zgartirishni tushuntirib bering?

## Mustaqil ishlash uchun topshiriqlar

1. Tomonlari  $a$ ,  $b$  va  $c$  bo'lgan uchburchakning yuzi, perimetri va yarim perimetrini to'g'ri burchakli uchburchakning uchlarida chop etish dasturini tuzing.
2. 1 dan 16 gacha bo'lgan sonlarni ekranning o'rtasidan to'rtta satr va to'rtta ustunda kvadrat matritsa shaklida chop etish dasturini tuzing.
3.  $z=2*x-3/y-a/b$ ,  $a=5$  funksiya qiymatini hisoblash dasturini tuzing.
4. 34 sonini trapetsiyaning uchlarida chop etish dasturini tuzing.
5. Besh xonali son kiritiladi. Shu sonning raqamlarini to'g'ri to'rtburchakning uchlarida va o'rtasida chop etish dasturini tuzing.
6. Olti xonali son kiritiladi. Shu sonning raqamlari o'rta arifmetigini topish dasturini tuzing.

## 2.3 C++ DASTURLASH TILI TARKIBIDAGI ARIFMETIK AMALLAR VA MANTIQUIY AMALLAR

*Reja:*

1. Arifmetik amallari;
2. Ta'minlash operatori;
3. Mantiqiy amallar.

C++ dasturlash tili tarkibida dastur tuziladigan vaqtda albatta matematik ifodalar, amal ishoralar va mantiqiy amallar ishtirok etishi mumkin. Dastur dasturlash tillarida amallar matematikadan yozilishi bilan farq qiladi. Dasturlash asoslarida amallarni ikki turga ajratamiz:

- arifmetik amallar;
- mantiqiy amallar;

### Arifmetik amallar

Berilganlarni qayta ishlash uchun dasturlash tillarida amallarning juda keng majmuasi aniqlangan. Amal - bu qandaydir harakat bo'lib, u bitta (unar) yoki ikkita (binar) operandlar ustida bajariladi, hisob natijasi uning qaytaruvchi qiymati hisoblanadi.

Tayanch arifmetik amallar dasturlash tilida quyidagicha yoziladi.

Matematik ifodasi	C++ tilida ifodasi	Izoh
+	+	qo'shish
-	-	Ayirish
.	*	ko'paytirish
:	/	bo'lish
Qoldiq	%	qoldikli bo'lish

### Arifmetik amallarning bajarilish tartibi

Matematikada bo'lgani kabi C++ dasturlash tilida ham 4 ta arifmetik amal(+; -; \*/)lardan foydalaniladi. Ularning bosqichlarga bo'linishi, bajarilish tartibi, qavslar ishtirokidagi bajarilish tartibi matematikadagi tartibidan farq qilmaydi. Masalan:

$$\begin{array}{cccc} 1 & 2 & 3 & 4 \\ 26-3+18-76+53; \end{array}$$

$$\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ 2) 38/2*x/12*45/y \end{array}$$

$$\begin{array}{cccccc} 7 & 1 & 8 & 2 & 9 & 3 & 4 & 10 & 5 & 6 \\ 3) 14-155/4+3*x+24/8*5-98*4/16; \end{array}$$

$$\begin{array}{cccccccccc} 14 & 11 & 12 & 13 & 5 & 9 & 6 & 7 & 10 & 8 & 1 & 2 & 4 & 3 \\ 4) 24+(96/24*6-(18/5+2*c/7+6*(216/3/4+3*2))). \end{array}$$

Eslatma: matematikada ba'zi hollarda, ayniqsa, harfiy va qavsli ifodalarda ko'paytirish amali oshkor yozilmaydi. C++ dasturlash tilida barcha amallar oshkor yozilishi shart. Masalan,

$$2x+5y-2*x+5*y$$

$$3(7a-5)-3*(7*a-5)$$

kabi yozuvlar xato hisoblanadi. Ularning to'g'ri yozuvi quyidagicha

$$1) 2*x+5*y-2*x+5*y$$

$$2) 3*(7*a-5)-3*(7*a-5)$$

Bundan tashqari C++ dasturlash tilida quyidagi amallardan ham foydalaniladi:11

1. Darajaga ko'tarish amali pow(x,n) funksiyasi yordamida bajariladi:

$$\text{pow}(x,n)=x^n; \text{pow}(3,4)=81;$$

2. n%m – amali n butun sonini m butun soniga bo'lganda qoldiqni hisoblaydi va qoldiqqa bo'linuvchi ishorasi beriladi.

$$37\%32=5; \quad -43\%8=-3; \quad 64\%(-12)=4;$$

3. Bir butun sonni ikkinchi butun songa bo'lganda to'liqsiz bo'linmani hisoblash uchun, natijani qiymat sifatida oluvchi o'zgaruvchi turi int bo'lishi yetarli. Masalan:

$$\text{int } a, b, c, d, e, f;$$

$$a=47; \quad b=23; \quad c=6; \quad d=a/c; \quad e=a/d;$$

$$\text{cout} \ll "d=" \ll d \ll "e=" \ll e;$$

Dastur natijasi d=7e=6 kabi bo'ladi.

4. pow10(intp) funksiyasi 10p ni hisoblaydi. Bunda p—butun son bo'lishi shart.

pow10(3)=1000; pow10(1)=10

Dasturlash asoslarida arifmetik amallar matematikadagi amallarni yozilishi bir oz farq qiluvchi holatlari ham mavjud. Bunday amallar quyidagicha.

C++ dasturlash tilida biror bir o'zgaruvchini qiymatini birga oshirish **inkrement** amali deyiladi va uning umumiy ko'rinishi:

**<o'zgaruvchi>++;**

C++ dasturlash tilida biror bir o'zgaruvchini qiymatini birga kamaytirish **dekrement** amali deyiladi va uning umumiy ko'rinishi:

**<o'zgaruvchi>- -;**

Inkrement va dekrement amallarini qo'llanilishi.

```
#include <iostream.h>
int main ()
{   int a=45; int b=12; // a va b o'zgaruvchisini e'lon qilish
    a++;                // a=46 ga teng bo'ldi
    b--;                // b=11 ga teng bo'ldi
    return 0;
}
```

### Ta'minlash operatori

Ma'lum bir ifodaning natijasi biror o'zgaruvchiga ta'minlash uchun C++ tilida "==" ishlatiladi va ta'minlash operatori quyidagicha:

**<o'zgaruvchi>=<ifoda>;**

C++ dasturlash tilida ta'minlash operatori amallar yordamida ham ishlatiladi. Qo'shish qiymat berish bilan (+=); ayirish qiymat berish bilan (-=); ko'paytirish, qiymat berish bilan (\*=); bo'lish qiymat berish bilan (/=); bo'lish qoldig'ini olish qiymat berish bilan (%=) va boshqalar. Bu holatlarning umumiy ko'rinishi:

**<o'zgaruvchi><amal>=<ifoda>;**

### Mantiqiy amallar

Bu turdagi o'zgaruvchi **bool** kalit so'zi bilan e'lon qilinib, xotiradan 1 bayt joy egallaydi va 0 (false, yolg'on) yoki (true, rost) qiymat qabul qiladi. Mantiqiy tur o'zgaruvchilar qiymatlar o'rtasidagi munosabatlarni ifodalaydigan mulohazalarni rost (true) yoki yolg'on (false) ekanligi tavsifida qo'llaniladi va ular qabul qiladigan qiymatlar matematik mantiq qonuniyatlariga asoslanadi. Mantiqiy mulohazalar ustida uchta amal aniqlangan:

- inkor;

- konyunksiya;

- dizyunksiya;

1) inkor – A mulohazani inkori deganda A rost bo‘lganda yolg‘on yoki yolg‘on bo‘lganda rost qiymat qabul qiluvchi mulohazaga aytiladi. C++ tilida inkor – ‘!’ belgisi bilan beriladi. Masalan, A mulohaza inkori «!A» ko‘rinishida yoziladi;

2) konyunksiya- ikkita A va B mulohazalar konyunksiyasi yoki mantiqiy ko‘paytmasi «A && B» ko‘rinishga ega. Bu mulohaza faqat A va B mulohazalar rost bo‘lgandagina rost bo‘ladi, aks holda yolg‘on bo‘ladi (odatda «&&» amali «va» deb o‘qiladi).

3) dizyunksiya – ikkita A va B mulohazalar dizyunksiyasi yoki mantiqiy yig‘indisi «A || B» ko‘rinishda yoziladi. Bu mulohaza rost bo‘lishi uchun A yoki B mulohazalardan biri rost bo‘lishi yetarli. Odatda «||» amali «yoki» deb o‘qiladi.

Mantiqiy amallarni bajarilish jadvali quyidagicha.

A	B	!A	!B	A&&B	A  B
0	0	1	1	0	0
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	1	1

Matematikadagi taqqoslash amali C++ dasturlash tilida yozilishi.

Nomi	C++ da ifodalanishi	Misol	Natija
Tenglik	==	12==50 5==5	False True
Teng emas	!=	100!=50 50!=50	True False
Katta	>	100>50 50>50	True False
Katta yoki teng	>=	100>=50 50>=50	True True
Kichik	<	100<50 50<50	True False
Kichik yoki teng	<=	100<=50 50<=50	False True

**Mantiqiy ko‘paytirish amali**

Amalning umumiy ko‘rinishi quyidagicha: a&&b.

Bunda a va b lar mantiqiy ifodalar yoki mantiqiy o'zgaruvchilar bo'lib, amal natijasi faqata va b larning har ikkalasi rost bo'lgandagina rost, qolgan barcha hollarda yolg'on bo'ladi. Bu amal matematikadagi konyunksiya mantiqiy amali bilan teng kuchli, ya'ni ikkita ampersant (&&) belgisi "va" so'zini almashtiradi.

Misollar:

$(37-21) != 13 \&\& 49/7 == 4+3$  ifodaning qiymati rost.

$x=15; x>20 \&\& x!=16$  ifoda qiymati yolg'ondir.

### **Mantiqiy qo'shish amali**

Amalning umumiy ko'rinishi quyidagicha:  $a \parallel b$

Bunda a va b lar mantiqiy ifodalar yoki mantiqiy o'zgaruvchilar. Amal natijasi a va b larning har ikkilasi yolg'on bo'lgandagina yolg'on, boshqa barcha hollarda rost bo'ladi. Bu amal matematikadagi dizyunksiya amaliga teng kuchli, ya'ni  $\parallel$  belgilari "yoki" so'ziga mos keladi.

Masalan,

$a=-16; b=32; 2*a-b<0 \parallel b+a<1; (52+a>5+b) \parallel true$  ifodalarning qiymatlari rost  
 $0 \parallel a-b>0$  ifodaning qiymati esa yolg'ondir.

### **Mantiqiy inkor amali**

Amalning umumiy ko'rinishi qo'yidagicha:  $!a$

Bunda a mantiqiy ifoda yoki mantiqiy o'zgaruvchi bo'lib, agar a ning qiymati true bo'lsa, amal natijasi false, amal natijasi false bo'lsa, true bo'ladi.

Masalan,

$!true; !(48/4+6==3*6); !3$  ifodalar qiymatlari false

$!(9*8==64); !(3*8==20 \&\& 100\%4==2); 0!$  ifodalarning qiymatlari esa true bo'ladi.

C++ dasturlash tili mantiqiy ifoda o'rnida 0 soni yoki qiymati nolga teng bo'lgan ifoda bo'sa, uni false deb, 0 dan farqli ixtiyoriy sonni yoki qiymati noldan farqli ifodani true deb qabul qiladi.

Mantiqiy ifodani mantiqiy o'zgaruvchiga qiymat sifatida berish ham mumkin.

Mantiqiy o'zgaruvchilar bool kalit so'z bilan e'lon qilinadi. Bu kalit so'z matematik mantiq ilmiga asos solgan ingliz matematigi Jorj Bul sharafiga shunday tanlangan.

Misol tariqasida quyidagi dasturni tahlil qiling.

```
#include <iostream.h>
using namespace std;
int main( )
{
```

```

int x, y, z, x1, y1, z1;
float a, b, c;
bool n, m, k, p, q;
x=16; y=30; z=21; a=19.25; b=0.75; c=40.16;
x1=3*z-(y+2); y1=c; n=a-b>b-c; m=(x+y)/2!=z+x1; k=y1-x==z+13;
p=k||39-z&&n||x&&!m;
q=!p&&!0||c-a>=0||!false&&!(n||m);
cout <<"n="<<n<<" m="<<m<<" k="<<k;
cout <<"\np="<<p<<" q="<<q<<endl;
system ("pause");
return 0;
}

```

Bu dasturda n, m, k, p, q o‘zgaruvchilarning turi mantiqiy tur bo‘lib, ular mantiqiy ifodalarni qiymat sifatida olmoqda.

Dastur javobi quyidagicha bo‘ladi.

```

n=1      m=1      k=0
p=1      q=1

```

Murakkab mantiqiy ifodalarni yozishda amalning ikki tarafidan joy qoldirmasa ham bo‘ladi. Shuningdek, amalning ikki tarafidagi mantiqiy ifodalarni qavsga olish yoki olmaslikning ham ahamiyati yo‘q. Lekin dastur tushunarli bo‘lishi uchun mantiqiy amallarning har ikkala tarafidan bittadan (agar kerak bo‘lsa xohlagancha) bo‘sh joy qoldirish, zarurat bo‘lganda oddiy mantiqiy ifodalarni qavsga olish maqsadga muvofiq.

Masalan, quyidagi to‘rtta mantiqiy ifodani ham kompilyator xato demaydi.

```

p=k+3>0||x+y-1<=16 && 1;
p=k+3>0 || x+y-1 <= 16&& 1;
p=(k+3>0)||(x+y-1<=16) && 1;
p=(k+3>0) || (x+y-1<=16) && 1;

```

Eslatma: hatto arifmetik amallarning ikkala tarafidan bittadan bo‘sh joy (probel) qoldirilsa, dasturni tushunish yengil bo‘ladi. Lekin bu aytilganlar qat’iy talab emas.

Masalan,  $x+3*y+11/3$  ni  $x + 3 * y + 11/3$  kabi yozish ma‘qulroq.

Biz hozirgacha tanishgan amallarning bir-birlariga nisbatan ustunlik darajasi va bosqichlarga bo‘linishi quyida keltirilgan.

```

(i++,i--), (++i,--i), !, (*,/,%),(+,-), (<,<=,>,>=), (==,! =), &&, ||, ?:,
(=,*=,/=,+=,-=,% =)

```



Bunda qavsga olingan amallar bir bosqich amallari hisoblanadi. Bir bosqich amallari o`rtasida ustunlik daraja bo`lmaydi. Demak, bu amallar ifodada aralash holda kelsa, eng birinchi  $i++$ ,  $i--$  amallar, eng oxirida  $=$ ,  $*=$ ,  $/=$ ,  $+=$ ,  $-=$ ,  $\%=$  amallar bajariladi.

Misollar: Quyidagi shartlar bajarilsa true, aks holda false qiymat oladigan mantiqiy ifodalarni tuzing.

a-juft, b esa toq son.

a,b,c sonlarning hech bo`lmaganda bittasi musbat.

a,b,c mantiqiy o`zgaruvchilardan hech bo`lmaganda bittasi chin.

Javoblar:

$a\%2==0\&\& b\%2==1$ ;

2.  $a>0 \parallel b>0 \parallel c>0$

3.  $a \parallel b \parallel c = true$

### ASCII belgilari (kodi) tushunchasi

ASCII belgilari deyilganda 0 dan 255 gacha kodlangan klaviaturada mavjud bo`lgan barcha belgilar va ASCII ning qo`shimcha belgilari tushuniladi. Bu belgilar klaviaturadagi harflar, raqamlar, qavslar, tinish belgilari va boshqa belgilardir.

**ASCII – bu American Standart Code for Information Interchange (Axborot almashishga mo`ljallangan Amerika standart kodi – Американский стандартный код для обмена информацией).**

### Nazorat savollari

1. Amal?

2. Arifmetik amallarni C++da yozilishi?

3. Inkrement vadekrement amallari va ularni C++ da yozilishi?

4. Mantiqiy amal?

5. Inkor amali va uning C++ da yozilishi;

6. Konyunksiya amali va uning C++ da yozilishi?

7. Dizyunksiya amali va uning C++ da yozilishi?

8. Taqqoslash amallari va ularni C++ da yozilishi?

9. C++dasturlash tilida arifmetik amallardan tashqari yana qanday amallardan foydalaniladi?

10. a)  $3/4$ ; b)  $3./4$ ; c)  $3/4.$  ; amallar natijasini aniqlang.

11.  $int a, b, c, d, e; a=238564159; b=a/1000\%100; c=a/10000/1000\%10; c=a/100; d=c\%1000/100*100;$

$cout<<"a="<<a<<"\n b="<<b<<"\n c="<<c<<" d="<<d<<"\n';$

buyruqlari natijasini aniqlang.

12.C++ dasturlash tilida arifmetik amallarning bajarilish tartibini tushuntiring(misollar yordamida).

### Mustaqil ishlash uchun topshiriqlar

a=rost(1) va b=yolg'on(0) bo'lganda quyidagi mantiqiy ifodalarning qiymatini aniqlang.

1. (a&&b)  !a;	11.(a  b)&&(a&&b);
2. (a  b)&&b;	12.(!a  b)&&(a&&!b);
3. (a  b)&&!b;	13.(a  b)&&(!a&&!b);
4. !(a  b)&&b;	14.(!a  !b)&&(a&&b);
5. (!a  b)&&b;	15.!(a  b)&&!(a&&b);
6. (!a  !b)&&b;	16.!(!a  !b)&&!(a&&b);
7. (!a  !b)&&!b;	17.!(!a  b)&&!(a&&b);
8. !((a  b)&&b);	18.!(a&&b)  !(a&&b);
9. (a&&b)  !b;	19.!(!a  !b)&&!(a  b);
10. (!a&&!b)  !b;	20.!(!a&&b)&&!(a  b);

## 2.4 C++ DASTURLASH TILI TARKIBIDAGI STANDART FUNKSIYALAR VA IFODALAR

*Reja:*

1. C++ tilida ifodalar;
2. Ifodalarni C++ tilida ifodalash tartibi
3. C++ dasturlash tilida standart funksiyalar.

### C++ tilida ifodalar

Barcha dasturlash tillari kabi C++ dasturlash tilida ham matematik ifodalar ma'lum bir standart bo'yicha yoziladi. Ifodalar tarkibidagi matematik funksiyalar C++ tilida standart fuksiyalar yordamida yoziladi, agar ifoda tarkibidagi funksiya standart funksiya tarkibida bo'lmasa, alohida funksiya yaratib olish kerak.

**Ifoda**-sonlar, harflarni arifmetik amallar va qavslar bilan birlashtirilgan yozuvga aytiladi.

C++ dasturlash tilidagi ifodalar tarkibidagi amallarni bajarilishi matematikadagi amallarni bajarilish tartibiga mos keladi. C++ tilida arifmetik amallarni yozilishi yuqoridagi mavzuga asosan yoziladi. Ifodalar tarkibidagi nomalumni faqatgina lotin alifbosida yozilishi kerak. Ifoda tarkibida kasr sonning surati yoki maxrajida ikki va undan ortiq hadlar bo'lsa C++ tilida ular albatta

qavsga olinishi kerak. Ifodalarni C++ dasturlash tilida yozishda quyidagi qoidalarga amal qilish lozim:

- ifodani kasr ko‘rinishida yozib bo‘lmaydi

$$\frac{2 * x + 23}{19 - 7 * a} \longrightarrow (2 * x + 23) / (19 - 7 * a) ;$$

- ifadada grek alifbosi harflari bo‘lsa ularni o‘ziga o‘xshash lotin alifbosi harflari bilan almashtiriladi yoki ularning nomlari to‘liq yoki qisqacha yoziladi

$$(2\alpha + 3\alpha) - 8\beta \sin\gamma + 4\delta - \mu \rightarrow (2 * a + 3 * \text{alfa}) - 8 * \text{beta} * \sin(\text{gamma}) + 4 * b - m;$$

- ikkita amal ketma-ket yozilmaydi

$$\frac{58}{-9} \rightarrow 58 / -9 \text{ bu noto‘g‘ri, to‘g‘ri varianti } 58 / (-9) \text{ kabi bo‘ladi};$$

- agar satr biror amal belgisi bilan tugasa, yangi satr matematikadagidek o‘sha amal bilan boshlanmaydi. Chunki ekranda biz ko‘rib turgan satrlar fizik satrlar bo‘lib, mantiqiy satr bir nechta satrlardan tashkil topadi. Natijada yana ikkita amal ketma-ket yozilgan bo‘lib qoladi;

- ochilgan qavslar soni yopilgan qavslar soniga teng bo‘lishi kerak ;

- dastur tarkibida nol raqami (0) va o harfini farqlashga odatlanish lozim.

### Quyidagi ifodani C++ dasturlash tilida yozing

$$\frac{x^{y+1} + e^{k-1}}{1 + x / y - \tan z} \left( \text{Ctg} \frac{x}{y} - \sqrt{z} \right) \left| \frac{x-1}{y+1} \right|$$

(pow(x, y+1) + exp(k - 1)) / (1 + x/y - tan(z)) \* (1/tan(x/y) - sqrt(z)) \* fabs(fabs((x-1)/(y+1)))

Eslatma: Qavs ichidagi amallar birinchi bajariladi, keyin funksiyalarning qiymatlari hisoblanadi. Undan keyin bizga hozirgacha ma’lum bolgan amallarning birinchi bajarilish nuqtayi nazaridan bajariish tartibi quyidagicha(Qavs ichiga olingan amallar bir bosqich amallari hisoblanadi):

(\* , / , %), (+ , -)

Grek alifbosidagi ifodalarni C++ tilida ifodalanishi

Grek alifbosida	C++ tilida
$y = 3\alpha - \frac{4\beta - \alpha}{\beta} ;$	y=3*alfa-(4*beta-alfa)/beta; <b>yoki</b> y=3*a-(4*b-a)/b;

## C++ dasturlash tilida standart funksiyalar

C++ dasturlash tili tarkibida mavjud bo'lgan matematik funksiyalar **standart funksiyalar** deb ataladi.

Ifodalar tarkibidagi funksiyalarni C++ dasturlash tilida ifodalash uchun standart funksiyalardan foydalaniladi. Funksiyalarni C++ dasturlash tilida ifodalash uchun ularni argumentlarini albatta qavsga olib yozish kerak.

C++ dasturlash tilida standart funksiyalardan foydalanish uchun albatta C++ tili tarkibidagi matematik funksiyalar kutubxonasiga murojat qilish kerak. Matematik funksiyalar kutubxonasiga murojat qilish quyidagicha.

**#include<math.h>**

C++ dasturlash tili tarkibidagi standart funksiyalar ro'yxati quyidagicha.

No	Matematik funksiya	C++ tilida ifodalanishi
1	$ a $ -butun son	$abs(x)$
2	$ a $ -haqiqiy son	$fabs(x)$
3	$\sqrt{x}$	$sqrt(x)$
4	$x^2$	$pow(x,2)$
5	$x^n$	$pow(x,n)$
6	$10^x$	$Pow(10,x)$
7	$e^x$	$exp(x)$
8	$ln(x)$	$log(x)$
9	$Sinx$	$sin(x)$
10	$Cosx$	$cos(x)$
11	$Tgx$	$tan(x)$
12	$Ctgx$	$cos(x)/sin(x)$
13	$Arcsinx$	$asin(x)$
14	$Arccosx$	$acos(x)$
15	$Arctgx$	$atan(x)$

**Misol1:** Quyidagi ifodalarni C++ tilida ifodalash.

Matematik ifodasi

$$y = (x + \sin x)^3 - \cos^2 x + \frac{1 + \log_a x}{\sqrt{t - x^2}}$$

C++ tilida ifodalanishi

$y = pow((x + \sin(x)), 3) + pow((\cos(x), 2) + (1 + \log(x)/\log(a)) / (\sqrt{t - \text{sqr}(x)}));$

**Misol2:** Quyidagi ifodalarni C++ tilida ifodalash.

Matematik ifodasi

$$y = |x - 2| + \sin x - \left| \frac{4}{\sqrt{t - x^2}} \right|$$

C++ tilida ifodalanishi

```
y=abs(x-2)+sin(x)+abs(1/sqrt(t-pow(x,2)));
```

C++ dasturlash tilida matematik funksiyalardan tashqari ba'zi bajariladigan amallarni keltiramiz. Ya'ni sonning butun qismi, qoldiq ni topish yoki yaxlitlash va hakoza amallar quyidagicha tasvirlanadi.

**fmod(x,y)** – x sonini y ga bo'lgandagi qoldiqni hisoblaydi. Bu amalni % belgisi orqali ham ifodalash mumkin.

**Misol:** qoldiqni hisoblash.

**fmod(12,5)=2, fmod(121,100)=21, fmod(1523,1000)=523**

**ceil(x)**- x haqiqiy sonni uzidan katta bo'lgan eng yaqin son bilan almashtiradi.

**Misol:** butun qismini hisoblash.

**ceil(12.5698)=13, ceil(1.5698)=2**

**floor(x)**- x haqiqiy sonni o'zidan kichik bo'lgan eng yaqin son bilan almashtiradi.

**Misol:** butun qismini hisoblash.

**floor(45.6598)=45, floor(789.15246)=789**

**hypot(x, y)**- x va y haqiqiy sonlar uchburchakning katetlari bo'lsa, gepatenuzani hisoblash.

**Eslatma:** Standart funksiyalar kutubxonasidan foydalanish uchun `int main()` funksiyasidan oldin **#include <math.h>** sarlavha faylini ham yozish kerak. Quyida keltirilgan dastur va uning natijalarini tahlil qiling:

```
#include <iostream.h>
#include <math.h>
using namespace std;
int main( )
{ double n, m, c, s, g, d, k,x,y,z,x1,y1,z1;
s= log(9); cout << "s=" << s;
c= log10(1000); cout << "    c=" << c << '\n';
n= pow(2.2,2.2); cout << "n=" <<n;
m = fmod(45,7); cout << "    m="<<m<< '\n';
g = tan(3.14/4); cout << "tg(3.14/4) = "<<g<< '\n';
k = 1/tan(3.14/4); cout << "ctg(3.14/4)=" <<k<< '\n';
x=ceil(14.235601); y=floor(7.235601); cout<<"x="<<x<<"    y="<<y<<'\n';
```

<pre> x1=ceil(-14.235601);      y1=floor(-7.235601);      cout&lt;&lt;"x1="&lt;&lt;x1&lt;&lt;" y1="&lt;&lt;y1&lt;&lt;"\n'; z=fmod(-23,7);cout&lt;&lt;"z="&lt;&lt;z; z1=pow(3,3); cout&lt;&lt;"  z1="&lt;&lt;z1&lt;&lt;"\n'; system ("pause"); return 0; } </pre>
<p>Dastur natijasi quyidagicha bo'ladi:</p> <pre> s=2.19722   c=3 n=5.6667   m=3 tg(3.14/4) = 0.999204 ctg(3.14/4)=1.0008 x=15      y=7 x1=-14    y1=-8 z=-2      z1=27 </pre>

Ro'yxatda bo'lmagan standart funksiyalar matematik formulalardan foydalanib, mavjud funksiyalar orqali ifodalanadi. Masalan,

$\text{Ctg}x \rightarrow \cos(x)/\sin(x)$  yoki  $1/\tan(x)$

$\text{Log}_2 x \rightarrow \log(x) / \log(2)$

**Quyidagi ifoda C++ dasturlash tilida yozilgan. Oddiy matematik ko'rinishini tiklang**

- 1)  $\text{sqrt}(\text{pow}(x,2)+\text{pow}(y,2)*\log(z)+\log_{10}(y+z))*(\text{sqrt}(\text{pow}(x,2) - \text{pow}(y,2)) + 1)$ ;
- 2)  $\exp(\text{abs}(x-y)*\text{abs}(\text{pow}(x-y, x+y) / (a*\sin(x))+a*\cos(x)))+\text{pow}((\text{pow}(x,6)+\text{pow}(\log(y), 2)), 1/3)$ ;
- 3)  $(\text{atan}(1)/4 * \text{pow}(\cos(y), 3) - \text{pow}(a, 3 * \sin(3*x)))/\text{atan}(2+(x/y))$ ;
- 4)  $\exp(\text{pow}(\sin(x), 3) + \text{pow}(x - 1, 1/3) - \log(\text{sqrt}(x+\text{pow}(y,2))))$ ;

Quyidagi ifodalar C++ dasturlash tilida yozilgan. Xatolar sonini aniqlang

- 1)  $j/((1-\text{sqrt}(z)/(t)) - 1+\text{pow}(x+ 3,5) - \text{sprt}(2 * x + 3 * a)$ ;
- 2)  $\text{atn}(1) * \sin x/\cos x + \ln(x) * \text{pow}(\text{sqrt}(x), 1/3) - 85,4$ ;
- 3)  $(28,5 * 64,0 - 0,19 + 5 e 4)/(\log_{10}(1001 + x) - \text{sqrt}(x))$ ;
- 4)  $x + \exp(\text{ciel}(x))+\text{floor}(4 \sin(x))+m \%(\text{nn}) - \text{pow}(\sin x, \cos x)$ .

**Nazorat savollari**

1. *Ifoda deb nimaga aytiladi?*
2. *Ifodalarni C++da yozilish tartibi?*
3. *Standart funksiya deb nimaga aytiladi?*
4. *Standart funksiyalarni C++da yozilish tartibi?*

5. *Ma'lumotlarni kodlash deyilganda nimani tushunasiz?*
6. *Ma'lumotlar nima uchun kodlanadi? Misollar keltiring.*
7. *Dasturlashning kodlashga qanday aloqasi bor?*
8. *Sonlarning oddiy matematik yozuvi bilan C++ dasturlash tilida yozilgan yozuvda qanday farqlari bor?*
9.  *$10^m$  soni C++ da qanday yoziladi?*
10. *Darajaga ko'tarish amali qanday bajariladi?*
11. *C++ da natural sonlarni bo'lishda to'liqsiz bo'linma va qoldiq qanday hisoblanadi?*
12. *Standart funksiyalarni C++ dasturlash tilida yozib chiqing.*
13. *a)  $\text{abs}(x)$  va  $\text{fabs}(x)$ ; b)  $\log(x)$  va  $\log_{10}(x)$ ; c)  $\text{pow}(x,y)$  va  $\text{pow}_{10}(x)$  funksiyalarning hamda  $\text{fmod}(x,y)$  funksiyasi bilan (%) amalining farqini tushuntiring(misollar keltirish bilan).*
14. *C++ dasturlash tilida ifodalarni yozish uchun mustaqil 3 turli masalalar tanlab, bajaring.*
15. *Standart funksiyalar kutubxonasidan foydalanish uchun qaysi sarlavha faylidan foydalaniladi?*
16. *C++ dasturlash tilida arifmetik ifodalarni yozishda grek alifbesi harflari, oddiy kasrlar qanday yoziladi?*
17. *Matematik ifodalarni C++ da yozishda qanday cheklovlar qo'yiladi? Nol raqami va O harfini qanday farqlaysiz?*

**Mustaqil ishlash uchun topshiriqlar.**

Quyidagi ifodalarni dasturlash tilida yozing

$$y = \frac{1 + x^4}{\sin x + |x|} - \operatorname{tg} x$$

$$y = \frac{ax^2 - bx + c}{|x|} - \left| \frac{x - t}{\sqrt{x + a}} \right|$$

$$y = \frac{2}{|ax^2 - bx + c|} - \left| \frac{x - t}{\sqrt{x + a}} \right|$$

$$y = \frac{\sqrt{x + a}}{\sin x + |x|} - 1 + x^4$$

$$y = \frac{\sqrt{x + a}}{\sin x + |x|} - ax^2 - bx + c$$

$$y = ax^2 - bx + c - \left| \frac{\log_a x}{\sqrt{x + a}} \right|$$

$$y = \frac{\sqrt{x + a}}{\sin x + |x|} - \left| \frac{\log_a x}{\sqrt{x + a}} \right|$$

$$y = \pi - 3x - \frac{\sqrt{x + a}}{\sin x + |x|}$$

$$y = \pi - 3x - \left| \frac{\log_a x}{\sqrt{x + a}} \right|$$

$$y = \pi - ax^2 - bx + c - \frac{\sqrt{x + a}}{\sin x + |x|}$$



### III- BOB. C++ TILIDA CHIZIQLI, TARMOQLANUVCHI VA TAKRORLANUVCHI JARAYONLARNI DASTURLASH.

#### 3.1 C++ TILIDA QIYMAT BERISH, KIRITISH VA CHIQUARISH OPERATORLARI

*Reja:*

1. C++ tilida qiymat berish operatori;
2. Qiymat berish operatoriningni kengaytirilgan holatlari;
3. Kiritish operatori;
4. Chiqarish operatori.

##### Qiymat berish operatori

Berilgan ifodaning natijasi hisoblangandan so‘ng ifodaning natijasi biror bir o‘zgaruvchiga ta‘minlanadi. Natijani saqlaydigan o‘zgaruvchi kompyuter xotirasidan joy egallaydi.

*Tarif: Ifodaning natijasi biror bir o‘zgaruvchiga ta‘minlansa yoki o‘zlashtirilishiga **qiymat berish** jarayoni deyiladi.*

Qiymat berish amallari guruhi, xususan, qiymat berish operatorlari ifoda operatorlari hisoblanadi. Qiymat berish operatori har doim ham ifoda natijasini o‘zida saqlash emas ba‘zi hollarda funksiya parametrlari va sanagich qiymatlarini hisoblash uchun ham ishlatiladi.

Qiymat berish operatorining umumiy ko‘rinishi quyidagicha.

**<o‘zgaruvchi>=<ifoda>;**

Qiymat berish operatorining C++dasturlash tilida quyidagicha ifodalanadi.

```
#include <iostream.h>
int main ()
{   int a=10; int b=6;    // a va b larni e‘lon qilish
    a=a+b;               // a ga a+b ni qiymati ta‘minlandi
    b=a-b;               // b ga a-b ni qiymati ta‘minlandi
    return 0;
}
```

**Misol:** a sonini qiymatini 2 ga ko‘paytirib oldingi holatiga qo‘shing.

```
#include <iostream.h>
int main ()
{   int a=10;           // a o‘zgarmasni e‘lon qilish
    a=a+2*a;           // a ga 2a ni qiymati qushildi
}
```

```

return 0;
}

```

Qiymat berish operatori biror o‘zgaruvchiga qiymat ta’minlangandan so‘ng uning boshlang‘ich qiymati o‘chirilib yangi qiymati o‘zlashtiriladi.

### Qiymat berish operatoriningni kengaytirilgan holatlari

Qiymat berish jarayoni bajarilish holatida ba’zi amallarni avtomatik bajarish holatlari ham mavjud, ular quyidagicha.

Amallar	Izoh
<b>y=x++</b>	<b>y ga x ni qiymati ta’minlanib keyin x ni qiymati 1 ga oshadi</b>
<b>y=++x</b>	<b>x ni qiymati 1 ga oshirilib keyin y ga ta’minladi</b>
<b>y=x--</b>	<b>y ga x ni qiymati ta’minlanib keyin x ni qiymati 1 ga kamayadi</b>
<b>y--x;</b>	<b>x ni qiymati 1 ga kamaytirilib keyin y ga ta’minladi</b>
<b>y+=a;</b>	<b>y ga a ni qiymati qushilib natija y ga yoziladi</b>
<b>y-=a;</b>	<b>y dan a ni ayirib natija y ga yoziladi</b>
<b>y*=a;</b>	<b>y ga a ni qiymati ko‘paytirilib natija y ga yoziladi</b>
<b>y/=a;</b>	<b>y ga a ni qiymati bo‘linib natija y ga yoziladi</b>
<b>y%=a;</b>	<b>y ga a ni qiymati bo‘linib qoldiq y ga yoziladi</b>

Oldingi mavzudagi dasturda **x++** buyrug‘i o‘rnida **++x** buyrug‘idan, **y--** buyrug‘i o‘rnida **--y** buyrug‘idan foydalanish mumkin. Chunki bu buyruqlar o‘zgaruvchining qiymatini 1ga oshirish yoki kamaytirish ma’nosida teng kuchli.

Ammo qiymat berish buyrug‘ida inkrement va dekrement amallaridan foydalanishga to‘g‘ri kelsa, “++”, “--“belgilarning o‘zgaruvchi oldidan yoki orqasidan qo‘yishning farqi bor. Quyida qiymat berish buyrug‘ining quyidagi qo‘shimcha korinishlari bilan tanishamiz:

**y=x++;** buyrug‘i bo‘yicha x ning qiymati y ga berilib, so‘ngra x ning qiymati 1ga ortadi. Bu buyruq **y=x; x=x+1;** buyruqlariga teng kuchli;

**y=++x;** buyrug‘i bo‘yicha oldin x ning qiymati 1 ga ortirilib, hosil bo‘lgan yangi qiymati yga beriladi. Bu buyruq **x=x+1; y=x;** buyruqlariga teng kuchli;

**y=a --;** buyrug‘i bo‘yicha oldin y ga a ning qiymati berilib, so‘ngra a ning qiymati 1 ga kamaytiriladi. Bu buyruq **y=a; a=a-1** buyruqlariga teng kuchli;

$y = --a$ ; buyrug‘i bo‘yicha oldin  $a$  ning qiymati  $1$  ga kamaytirilib, o‘sha kamaytirilgan qiymati  $y$  ga beriladi. Bu buyruq  $a = a - 1$ ;  $y = a$  buyruqlari teng kuchli;

$y += a$ ; buyrug‘i bo‘yicha  $y$  ga  $a$  ni qo‘shib, natija  $y$  ga beriladi. Ya‘ni bu buyruq  $y = y + a$ ; buyrug‘iga teng kuchli;

$y -= a$ ; buyrug‘i bo‘yicha  $y$  dan  $a$  ni ayirib, natija  $y$  ga beriladi. Ya‘ni bu buyruq  $y = y - a$ ; buyrug‘iga teng kuchli;

$y * = a$ ; buyrug‘i bo‘yicha  $y$  ni  $a$  ga ko‘paytirib, natija  $y$  ga beriladi. Ya‘ni bu buyruq  $y = y * a$ ; buyrug‘iga teng kuchli;

$y /= a$ ; buyrug‘i bo‘yicha  $y$  ni  $a$  ga bo‘lib, natija  $y$  ga beriladi. Ya‘ni bu buyruq  $y = y / a$ ; buyrug‘iga teng kuchli;

$y \% = a$ ; buyrug‘i bo‘yicha  $y$  ni  $a$  ga bo‘lganda qoldiqni hisoblab, natija  $y$  ga beriladi. Ya‘ni bu buyruq  $y = y \% a$  buyrug‘iga teng kuchli;

$y * = a + 3$ ; buyrug‘i bo‘yicha  $y$  ni  $a + 3$  ga ko‘paytirib, natija  $y$  ga beriladi. Ya‘ni bu buyruq  $y = y * (a + 3)$ ; buyrug‘iga teng kuchli;

$y + = ++a$ ; buyrug‘i bo‘yicha  $a$  ni  $1$  ga orttirib, natija  $y$  ga qo‘shiladi va bu yig‘indining qiymati  $y$  ga beriladi. Ya‘ni bu buyruq  $a = a + 1$ ;  $y = y + a$ ; buyruqlariga teng kuchli;

$y \% = a ++$ ; buyrug‘i bo‘yicha  $y$  ni  $a$  ga bo‘lganda qoldiq hisoblab, natija  $y$  ga beriladi va  $a$   $1$  ga orttiriladi. Ya‘ni bu buyruq  $y = y \% a$ ;  $a = a + 1$ ; buyruqlariga teng kuchli;

$y * = --a$ ; buyrug‘i bo‘yicha  $a$   $1$  ga kamaytiriladi va ayirma  $y$  ga ko‘paytirilib, natija  $y$  ga beriladi. Ya‘ni bu buyruq  $a = a - 1$ ;  $y = y * a$ ; buyruqlariga teng kuchli;

$y -= a--$ ; buyrug‘i bo‘yicha  $y$  dan  $a$  ni ayirib, natija  $y$  ga beriladi va  $a$  ning qiymati  $1$  ga kamaytiriladi. Ya‘ni bu buyruq  $y = y - a$ ;  $a = a - 1$ ; buyruqlariga teng kuchli;

$y += --y$ ; buyrug‘i bo‘yicha  $y$   $1$  ga kamaytirilib, natija  $y$  ga qo‘shiladi va yig‘indining qiymati  $y$  ga beriladi. Ya‘ni bu buyruq  $y = y - 1$ ;  $y = y + y$ ; buyruqlariga teng kuchli.

Qiymat berish buyrug‘ining  $y = ++a$ ; va  $y = --a$ ; ko‘rinishlari uning **prefiks** ko‘rinishi,  $y = a++$ ; va  $y = a--$ ; ko‘rinishlari **postfiks** ko‘rinishlari deyiladi.

**Misol** tariqasida quyidagi dasturni va uning natijalarini tahlil qiling:

```
#include <iostream.h>
int main( )
{
int a, b, c, d, x, y, z, n, m;
a = 5; b = 6; c = 7; d = 8;
```

```

x=++a; y=++a; x+=x++; cout << "x=" << x << " y=" <<y<< " a =" <<a;
x-=b--; y+--y; y%=d--; c*=-a; cout << "\nx =" << x << "\ny =" << y <<
" c =" << c;
n=--d; n=-n; c =-x; cout << '\n'<< "n =" << n << "\nb = " <<b<< "\tc
="<<c<<\n';
system("pause");
return 0;
}

```

Dastur natijasi quyidagicha chop etiladi:

x = 13    y = 7    a = 7

x = 7

y = 4    c = 42

n = 0

b = 5    c = 6

**Eslatma:** Qavs ichidagi amallar birinchi bajariladi, keyin funksiyalarning qiymatlari hisoblanadi. Undan keyin bizga hozirgacha ma'lum bolgan amallarning birinchi bajarilish nuqtayi nazaridan bajariish tartibi quyidagicha(Qavs ichiga olingan amallar bir bosqich amallari hisoblanadi):

(i++, i--), (++i, --i), !, (\*, /, %), (+, -), (<, >, <=, >=), (==, !=), &&, ||, ?:, (=, \*=, /=, +=, -=, %=).

### Kiritish operatori

C++ dasturlash tilida dastur tuzishda o'zgaruvchilar qiymati kompyuter xotirasiga joylashtirish jarayoni ma'lumotlarni kiritish deyiladi. O'zgaruvchilar oqimi va ma'lumotlar jamlanmasi o'zgaruvchilar turiga qarab kompyuter xotirasidan joy egallaydi. Dastur tuzish jarayonida o'zgaruvchilar qiymatini kompyuter xotirasiga kiritish buyrug'i yozilishidan avval o'zgaruvchilarni, albatta, turiga mos ravishda e'lon qilishi kerak. Ba'zi masalalarda parametrlarning qiymatlari oldindan ma'lum bo'lmaydi. Ularning qiymatlarini dasturni bajarish jarayonida kiritib ketishga to'g'ri keladi. Bu hol tuzilgan algoritmnining ommaviyligini ta'minlaydi. Masalan, kvadrat tenglamani yechish dasturini tuzishda a, b, c koeffitsentlarning qiymatlarini har ijro etganda o'zgartirib yurmasdan, dasturni ijro etish jarayonida kiritib ketish qulay.

C++ dasturlash tilida ma'lumotlarni kiritish operatori **cin** kalit so'zi yordamida tasvirlanadi. o'zgaruvchilar ikki va undan ortiq bo'lsa, u holda >> belgi orqali o'zgaruvchilar bir biridan ajratilib yoziladi.

C++ dasturlash tilida ma'lumotlarni kiritish operatorining umumiy ko'rinishi quyidagicha:

**cin>>o'zgaruvchi1>>o'zgaruvchi2;**

Qiymati kiritiladigan o'zgaruvchilar bu operatoridan keyin bir-birlaridan ">>" belgilari bilan ajratilgan holda yoziladi. Masalan: **cin >> a >>b>>c;** buyrug'i dastur ijrosi davomida a, b va c o'zgaruvchi parametrlarning qiymatlari kiritilishi kerakligini bildiradi va kompyuter ushbu buyruqni uchratgach dastur bajarilishini to'xtatib, o'zgaruvchilar qiymatlari kiritilishini kutadi. Shundan so'ng o'zgaruvchilarning qiymatlari bir-birlaridan probel bilan ajratilgan holda 5 -4 12 kabi klaviaturadan terilib, ENTER tugmachasi bosilsa, nooshkor holda a=5; b=-4; c=12 kabi qiymat berish buyruqlari bajariladi. Bunda quyidagilarga alohida e'tibor berish kerak:

- o'zgaruvchilar kiritayotgan qiymatlarni kiritish tartibidagi nomeri bo'yicha qabul qiladi, ya'ni cin operatoridagi **n**- o'zgaruvchi kiritilgan **n**- qiymatni oladi;

- kiritilayotgan qiymatlar o'zgaruvchi uchun e'lon qilingan turga mos kelishi shart, aks holda xatolik sodir bo'ladi;

- cin operatori dasturning istalgan joyida bo'lishi mumkin. Ammo bu operator tarkibidagi o'zgaruvchilardanfoydalanilgan buyruqlardan oldin uchrashi kerak, aks holda dastur ijro etilayotganda buyruqlardagi o'zgaruvchilarning qiymatlari ma'lum bo'lmay qoladi.

**Misol:** C++ dasturlash tilida bir va ikkitadan ortiq o'zgaruvchilarni kiritishni tasvirlash.

```
#include <iostream.h>
int main ()
{   int a,b,c;           // a,b va c o'zgaruvchilarni e'lon qilish
    cin>>a;              // a ni kiritish
    cin>>b>>c;           // b va c larni kiritish
    return 0;
}
```

### Chiqarish operatori

C++ dasturlash tilida o'zgaruvchilar qiymati qayta ishlanib natija kompyuter ekraniga yoki ma'lum bir fayllarga joylashtirish jarayoni ma'lumotlarni chiqarish deyiladi. O'zgaruvchilar va ma'lumotlar oqimi chiqarilish vaqtida, albatta, aniq joyi ko'rsatiladi, agar kompyuter ekraniga ma'lumot chiqarilsa, u holda aynan joyi ko'rsatilishi shart emas.

C++ dasturlash tilida ma'lumotlarni chiqarish operatori **cout** kalit so'zi yordamida tasvirlanadi. O'zgaruvchilar ikki va undan ortiq bo'lsa, u holda << belgi orqali o'zgaruvchilar bir-biridan ajratilib yoziladi.

C++ dasturlash tilida ma'lumotlarni chiqarish operatorining umumiy ko'rinishi quyidagicha:

**cout<<o'zgaruvchi1<<o'zgaruvchi2;**

Qiymati chiqarilayotgan o'zgaruvchilar cout operatoridan keyin bir-birlaridan "<<" belglar bilan ajratilgan holda **cout <<a<<b<<c;** kabi sanab ko'rsatiladi. Bunda quyidagi holatlarga e'tibor berish lozim:

- yuqorida qayd etilganidek cout operatorida ("") ichiga olib yozilgan matn (satr) qayta ishlanmasdan qanday yozilgan bo'lsa, shundayligicha chop etiladi. Bu holat natijalarini chiqarishda tushuntirish matnlaridan foydalanish imkoniyatini beradi;

- '\t' – tabulyatsiya belgilari bo'lib, o'zgaruvchilar va o'zgarmaslarning qiymatlari oralig'ida 7 ta bo'sh joy qoldirib chop etish maqsadida foydalaniladi;

- '\n' – belgilaridan foydalanilganda undan keyin chop etilayotgan ma'lumotlar alohida satrdan chop etiladi. Lekin qo'shtirnoq ichida bu belgilardan foydalanishda apostrof (tutuq belgisi) qo'yilmaydi;

- **cout** operatorida qiymatlari aniqlangan o'zgaruvchili ifodalar yozilgan bo'lsa, ularning qiymatlari hisoblanib, natijasi chop etiladi. Shuningdek, son ham o'zgarishsiz chop etiladi;

- **probel** (bo'sh joy) yoki tabulyatsiyalardan foydalanilmasa, bir nechta sonlar ketma-ket yozilib bitta son kabi chop etiladi. Yuqoridagi holatlarni quyidagi dastur misolida namoyish qilamiz.

**Misol:** C++ dasturlash tilida a ni qiymatini 2 ga oshirib natijani chiqaring.

```
#include <iostream.h>
int main ()
{   int a=14;           // a o'zgaruvchini e'lon qilish
    a+=2;              // a ni qiymati ikkiga oshdi
    cout<<a;          // a ni ekranga chiqarish
    return 0;
}
```

**Misol:** C++ dasturlash tilida ikki va undan ortiq o'zgaruvchilarni ekranga chiqaring.

```
#include <iostream.h>
int main ()
```

```

{   int a=4; int b=3;           // a va b o'zgaruvchilarni e'lon qilish
  a=a+2; b*=a;
  cout<<a<<b;                 // a va b ni ekranga chiqarish
  return 0;
}

```

**Dastur natijasi**

**6 18**

C++ dasturlash tilida izohli ma'lumotlarni ekranga chiqarish uchun qo'shtirnoq ichiga olinib yozilsa, albatta, ma'lumotni izoh sifatida qaraladi.

**Misol:** C++ dasturlash tilida izohli ma'lumotlarni(a ni qiymati) ekranga chiqaring.

```

#include <iostream.h>
int main ()
{   int a=-8;                   // a o'zgaruvchini e'lon qilish
  a=a+2;
  cout<<"a ni qiymati="<<a;    // a ni ekranga chiqarish
  return 0;
}

```

**Dastur natijasi**

a ni qiymati=-6

C++ dasturlash tilida ma'lumotlarni yangi qatordan ekranga chiqarish uchun "\n" yozishi kerak.

**Misol:** C++ dasturlash tilida o'zgaruvchilarni ekranga chiqaring.

```

#include <iostream.h>
int main ()
{   int a,b;
  cout<<"a =";
  cin>>a;                          // a ga 1 ni kiritsangiz
  a=a*8; b=a+6;
  cout<<a<<"\n"<<b;
  return 0;
}

```

**Dastur natijasi**

a =1  
8

**Misol:**  $y = 4ax + bc$  funksiya qiymatini hisoblash dasturini tuzing:

```
#include <iostream.h>
using namespace std;
int main( )
{
int a, b, c, x, y; cin >> a >> b >> c >> x;
y = 4*a*x + b*c; cout << c;
cout << "\n javoblarni tahlil qiling";
cout << " " << a << "\n " << b << " " << '\n' << y << '\n';
cout << a << b << c;
cout << '\n' << (2*a - b)*4 << " " << 200 << '\n';
system("pause");
return 0;}

```

Agar `cin >> a >> b >> c >> x;` buyrug‘i bo‘yicha 2 3 4 5 qiymatlari kiritilsa, ekranda quyidagicha natijalar chop etiladi.

4

Javoblarni tahlil qiling

2

3

52

234

4 200

### Nazariy savollar

1. Qiymat berish operatorining umumiy ko‘rinishi?
2. Qiymat berish operatoriningni kengaytirilgan holatlari?
3. Kiritish operatori va uning umumiy ko‘rinishi?
4. Ikki va undan ortiq o‘zgaruvchilarni kiritish jarayonini tushuntiring?
5. Chiqarish operatori va uning umumiy ko‘rinishi?
6. Ikki va undan ortiq operatorlarni o‘zgaruvchilarni chiqarish?
7. Izohli ma’lumotlarni ekranga chiqarish?
8. Dasturda foydalaniladigan operator deyilganda nimani tushunasiz?



9. Qiymat berish(o'zlashtirish, ta'minlash) operatorini tushuntiring, u qanday yoziladi(misollar keltiring).
10. C++ dasturlash tilida  $y=y-5$ ;  $y=-x-3$ ; yozuvlari qanday o'qiladi?
11. Inkrement va dekrement amallari qanday amallar?
12.  $a++$  va  $++a$  yoki  $-b$  va  $b-$  amallarining qanday farqlari bor?
13. Quyidagi ikkita buyruqni bitta buyruq qilib yozing: a)  $x=x+3$ ;  $y=y+x$ ; b)  $x=x-1$ ;  $y=y\%x$ ; c)  $n=n+c$ ;  $c++$ ;
14. Prefiks va postfiks amallarining mohiyatini tushuntiring.
15. Nol raqami va o harfini qanday farqlaysiz? 0 va o belgilardan qaysi biri nol?
16. Ma'lumotlarni kiritish operatoridan qanday maqsadda foydalaniladi?
17. Dastur tarkibida qiymati kiritilayotgan o'zgaruvchilar qayerda va qanday yoziladi?
18. Ma'lumotlarni kiritish operatoridan dasturning qayerida, qanday va necha marta foydalanish mumkin?
19.  $cin >> a >> b >> c$ ; buyrug'i qanday ijro etilishini tushuntiring.
20.  $cin >> x >> y >> z >> h$ ; buyrug'iga binoan -23 6 9 0.3 sonlari bir-birlaridan probel bilan ajratib yozildi va Enter tugmachasi bosildi. Nooshkor holda qanday buyruqlar bajariladi?
21. O'zgaruvchilarga kiritilayotgan qiymatlar qanday shartlarga bo'ysunishi kerak?
22. '\n' va '\t' buyruqlaridan dastur tarkibida qanday maqsadda foydalaniladi?
23. Ma'lumotlarni chiqarish operatoridan qanday maqsadda foydalaniladi?
24. Ma'lumotlarni chiqarish operatorining qanday ko'rinishlari mavjud?
25. Ekranda 7 ta bo'sh satr chop etish buyrug'i qanday?
26. Matnlar qanday chop etiladi?  
 $cout << 2 << "[ << 4 << "; << 5 << "]" = << 18 << "\n"$  buyrug'i qanday ma'lumotni chop etadi?
27.  $cout << "salom" << '\n' << 4 << '\t' << "ta kitob\n"$ ; buyruq natijasini yozing.
28. Natijalarni bir-birlaridan ajratib, ya'ni probel(bo'sh o'rinlar) blan ajraib chop etish usullarini bayon qiling.
29. Qiymat berish buyrug'i belgisi, tenglik belgisi, teng emaslik belgisi C++ dasturlash tilida qanday yoziladi?
30. Hozirgacha bizga ma'lum bo'lgan amallarning bajarilish tartibini ayting.

### Mustaqil ishlash uchun topshiriqlar

1. Kvadratning tomoni  $a$  berilgan. Uning perimetri aniqlansin.  $P=4\cdot a$ .
2. Kvadratning tomoni  $a$  berilgan. Uning yuzasi aniqlansin.  $S=a^2$ .
3. To'g'ri to'rtburchakning tomonlari  $a$  va  $b$  berilgan. Uning yuzasi  $S=a\cdot b$ ; va  $P=2\cdot(a+b)$  perimetri aniqlansin.
4. Aylananing diametri  $d$  berilgan. Uning uzunligi aniqlansin  $L=\pi\cdot d$ .
5. Kubning yon tomoni  $a$  berilgan. Uning hajmini  $V=a^3$  va to'la sirti  $S=6\cdot a^2$  aniqlansin.
6. Paralelepidning tomonlari  $a, b, c$  berilgan. Uning hajmi  $V=a\cdot b\cdot c$  va to'la sirti  $S=2\cdot(a\cdot b+b\cdot c+a\cdot c)$  aniqlansin.
7. Doiraning radiusi  $R$  berilgan. Uning uzunligi  $L$  va yuzasi  $S$  aniqlansin.  
 $L=2\cdot\pi\cdot R$ ,  $S=\pi\cdot R^2$ .
8. Ikkita son  $a$  va  $b$  berilgan. Ularning o'rta arifmetigi aniqlansin.  $(a+b)/2$ .
9. Ikkita manfiy bo'lmagan son  $a$  va  $b$  berilgan. Ularning o'rta geometrigi aniqlansin.  $\sqrt{a\cdot b}$ .
10. Nolga teng bo'lmagan ikkita son berilgan. Ularning yig'indisi, ko'paytmasi va har birining kvadrati aniqlansin.
11. Nolga teng bo'lmagan ikkita son berilgan. Ularning yig'indisi, ko'paytmasi va har birining moduli aniqlansin.
12. Uzunlik  $L$  santimetrda berilgan. Undagi to'liq metrlar sonini aniqlovchi dastur tuzilsin ( $1 \text{ metr} = 100 \text{ sm}$ ).
13. Og'irlik  $M$  kilogramda berilgan. Undagi to'liq tonnalar sonini aniqlovchi dastur tuzilsin ( $1 \text{ tonna} = 1000 \text{ kg}$ ).
14. Faylning hajmi baytlarda berilgan. Bo'lib butunni olish operatsiyasidan foydalanib fayl hajmining to'liq kilobaytlarda ifodalovchi dastur tuzilsin ( $1 \text{ Kilobayt} = 1024 \text{ bayt}$ ).
15.  $A$  va  $B$  ( $A > B$ ) musbat sonlar berilgan.  $A$  kesmada,  $B$  kesmani necha marta joylashtirish mumkinligini aniqlovchi dastur tuzilsin.
16.  $A$  va  $B$  ( $A > B$ ) musbat sonlar berilgan.  $A$  kesmada,  $B$  kesmani necha marta joylashtirish mumkin.  $A$  kesmada  $B$  kesmaning joylashmagan qismini aniqlovchi dastur tuzilsin.
17. Ikki xonali son berilgan. Oldin uning o'nliklar xonasidagi raqamni, So'ng birlar xonasidagi raqamni chiqaruvchi dastur tuzilsin.
18. Ikki xonali son berilgan. Uning raqamlar yig'indisi va ko'paytmasini aniqlovchi dastur tuzilsin.
19. Ikki xonali son berilgan. Uning raqamlari o'rnini almashtirishdan hosil bo'lgan sonni aniqlovchi dastur tuzilsin.

20. Uch xonali son berilgan. Uning yuzlar xonasidagi raqamini aniqlovchi dastur tuzilsin.

21. Quyidagi buyruqlar natijasini aniqlang

$a=12; b=4; c=(a+b)/8; a=a+c; c=a+c; b=b-c; b=4*b/c;$

$cout<<a<<b<<c<<\n'; \quad j: 14-316$

22.  $x=2; y=5; y=2*x+y; y=2*x+y; x=2*x+y; y=2*x-3; x=x-y;$

$cout<<"x="<<x<<" ; y="<<y<<\n'; \quad j: x=-14; y=31$

23.  $q=4; q=q+2; q=q+2; q=q+2; q=q+2; q=q+2; q=q+2; q=q+2;$

$cout<<"q:=="<<q<<\n'; \quad j: q:==16$

24.  $s=4;d=24;c=-5; d=-c;c*=s-10; s+=c++;c--;d--;$

$cout<<"s="<<s<<" ; d="<<d<<" ; c="<<c<<\n';$

$J: s=34; d=29;c= 30$

25.  $n=15;m=20; k=12; k/=6;n%=k+2;m+=m++;m-=m--;n+=m;$

$cout<<m<<k<<n<<\n'; \quad J:-122.$

26.  $x=5;y=8; x+=++y;x-=x++;y=y+x;y%=--8;cout<<2*x+3*y<<\n'; \quad J: 8.$

27.  $a=12;b=12;a-=++a;b-=b++; \quad cout<<b<<a<<\n'; \quad J: 10.$

### 3.2 C++ DASTURLASH TILIDA CHIZIQLI JARAYONLARNI DASTURLASH

*Reja:*

1. C++ tilida chiziqli dasturlar;

2. C++ tilida chiziqli dasturlarga oid misollar;

#### C++ tilida chiziqli dasturlar

C++ dasturlash tilida tuzilgan dasturlar albatta uchta jarayonga asoslanib tuziladi. Dasturlash tili operatorlari yechilayotgan masala algoritmini amalga oshirish uchun ishlatiladi. Operatorlar chiziqli va boshqaruv operatorlariga bo'linadi. Aksariyat holatlarda operatorlar nuqtali vergul (;) belgisi bilan tugallanadi va u kompilyator tomonidan alohida operator deb qabul qilinadi.

Kompilyator dasturni ishga tushirish vaqtida dasturni kodini mashina tiliga tarjima qiladi.

Dastur tuzish vaqtida buyruqlar ketma-ketligi uzluksiz bajarilib boshqa shartlar talab etilmasa, dastur chiziqli hisoblanadi.

*Tarif: Chiziqli algoritmlarga asoslanib ixtiyoriy dasturlash tilida tuzilgan dasturlar **chiziqli dasturlar** deyiladi.*

Chiziqli dasturlar tarkibiy qismi bo'lgan operator va buyruqlarda hech qanday shart yoki takrorlanish bajarilmaydi. Chiziqli dasturlar tarkibidagi bo'yruqlar, albatta, bir marta bajariladi.

## C++ tilida chiziqli dasturlarga oid misollar

**Misol:** Quyidagi funksiyani hisoblang  $((a+x)>0)$ .

$$y = \frac{ax^3 - \sin x}{1 + \ln x} - \sqrt{a+x}$$

```
#include <iostream.h>
#include <math.h>
int main ()
{   int a,x; double y;
    cin>>a>>x;
    y=(a*pow(x,3)-sin(x))/(1+log(x))-sqrt(a+x);
    cout<<y;
    return 0;    }
```

Yuqoridagi masalaga e'tibor bersak  $a$  va  $x$  o'zgaruvchilar qiymati berilganda  $y$  funksiyani natijasi hisoblandi, algoritm dastur tarkibidagi operatorlar ham bir marta bajarilyapti.

**Misol:** Asosining radiusi  $R$  va yasovchisi  $l$  bo'lgan konus hajmi va to'la sirtini toping.

Bu masala yechimini aniqlash uchun konus to'la sirti va hajm formulalarini aniqlash kerak. Berilgan  $R$  va  $L$  yordamida to'la sirtini aniqlash mumkin, lekin hajmni topish uchun konus balandligi ham kerak.

```
#include <iostream.h>
#include <math.h>
int main ()
{   int r,l;
    cin>>r>>l;
    h=sqrt(l*l- r*r); // konus balandligi
    s=M_PI*r*r+ M_PI *r*l; // konus to'la sirti
    hajm=(1/3)* M_PI *r*r*h; // konus hajmi
    cout<<"s="<<s<<"hajm="<<hajm;
    return 0;
}
```

**Misol:**  $x = 5$ ;  $y = 4,5$ ;  $z = 0,5$  bo'lsa,  $\sqrt{x^2 + y^2} \ln z + \frac{\lg(y+z)}{\sqrt{x^2 - y^2 + 1}}$  ifodaning qiymatini hisoblash dasturini tuzing:

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

```

#include <math.h>
using namespace std;
int main( )
{ float x=5, y=4.5, z=0.5, p, q, m;
p=sqrt(pow(x,2)+pow(y,2));
q=sqrt(pow(x,2)-pow(y,2));
m=p*log(z)+log10(y+z)/q;
cout << "m=" << m << endl;
return 0;
}

```

Ushbu dastur chiziqli dastur hisoblanadi. Chunki unung buyruqlari berilgan tartibda birin-ketin bajariladi va dastur natijasi  $m = -4.34196$  kabi bo'ladi.

Eslatma: Butun sonlar ustida bo'lish amalini bajarayotganda amal natijasi qiymat sifatida berilayotgan o'zgaruvchining turi haqiqiy bo'lsa ham o'zgaruvchi to'liqsiz bo'linmani oladi. Bo'linmaning qiymatini haqiqiy son shaklida to'liq olish uchun bo'linuvchi yoki bo'luvchini haqiqiy son sifatida yozish kerak.

Masalan,  $8/3=2$ ;  $8./3=2.66667$  kabi bajariladi, garchi float  $a=8/3$  yoki double  $a=8/3$  ko'rinishda aniqlangan bo'lsa ham, natija 2 bo'ladi. Lekin funksiyalarning qiymatlarini bo'lishda to'g'ri bajariladi. Chunki funksiyalarning o'zlarini emas, ularning qiymatlarini bo'ladi. Bu fikrlarga quyidagi dasturni tahlil qilish yo'li bilan ishonch hosil qilish mumkin.

```

#include <iostream.h>
#include <math.h>
using namespace std;
int main( )
{ float x, y, z, p, q, m;
x=38/3;y=38./3;z=38/.3;cout<<"x="<<x<<" y="<<y<<" z="<<z<<"\n";
p=cos(atan(1)*4)/3;
q=8.24/3.12; cout<<"p="<<p<<" q="<<q<<"\n";
system("pause");
return 0;
}

```

Dastrur natijasi quyidagicha bo'ladi

```

x=12 y=12.6667 z=126.667
p=-0.333333 q=2.64103

```

Endi son va uning raqamlarini qayta ishlash bilan bog‘liq ikkita masala dasturini tuzamiz:

**Masala.** *Ixtiyoriy to‘rt xonali sonning raqamlari yig‘indisini topish dasturini tuzing.*

```
#include <iostream.h>
using namespace std;
int main( )
{
int n,a1,a10,a100,a1000,s;
cout <<"to`rt xonali sonni kiriting \n";
cin>>n;
a1 = n% 10; a10 = n/10% 10;
a100 = n/100% 10; a1000=n/1000;
s = a1 + a10 + a100 + a1000;
cout <<s<<"\t sonning raqamlari yig‘indisi = "<<s<<endl;
system ("pause");
return 0;
}
```

**Masala.** *Besh xonali sonning birinchi va oxirgi raqamlari almashtirilganda uning qiymati qanday o‘zgarishini aniqlash dasturini tuzing:*

```
#include <iostream.h>
using namespace std;
int main( )
{ int n, a1, a5, a234, at, f;
//cout << "besh xonali sonni kiriting\n";
cin >>n;
a1=n%10;
a5=n/10000;
a234=n/10-a5*1000;
at=a1*10000+a5+a234*10;
if (n>at) {f=n-at; cout <<"\n berilgan "<<n<<" conl "<< f <<" ga
kamaydi\n";}
else {f=at-n; cout << "\n berilgan "<<n<<" sonl "<<f<<" ga ortdi
\n";}
system ("pause");
```

```
return 0;  
}
```

C++ dasturlash tilida chiziqli dasturlar tuzilganda uning tarkibida matematik funksiyalar ishtirok etsa, albatta, matematik funksiyalar paketini chaqirish kerak.

Ifodalarni ketma-ket ijro etish strukturasi C++ tomonidan ta'minlanadi. Normal sharoitda C++ ifodalari dasturdagi navbatiga ko'ra bajariladi.

### Nazorat savollari

1. Chiziqli algoritm deb nimaga aytiladi?
2. Chiziqli dasturlar?

### Mustaqil ishlash uchun topshiriqlar

1. Uch xonali son berilgan. Uning raqamlar yig'indisini va raqamlar ko'paytmasini aniqlovchi programma tuzilsin.
2. Uch xonali son berilgan. Uning raqamlarini teskari tartibda yozishdan hosil bo'lgan sonni aniqlovchi programma tuzilsin.
3. Uch xonali son berilgan. Uning chapdan birinchi raqamini o'chirib o'ng tarafiga yozishdan hosil bo'lgan sonni aniqlovchi programma tuzilsin.
4. Uch xonali son berilgan. Uning o'ngdan birinchi raqamini o'chirib chap tarafiga yozishdan hosil bo'lgan sonni aniqlovchi programma tuzilsin.
5. Uch xonali son berilgan. Uning o'nliklar xonasidagi raqam bilan yuzliklar xonasidagi raqamni almashtirishdan hosil bo'lgan sonni aniqlovchi programma tuzilsin. (Kirish=123; Natija=213).
6. Uch xonali son berilgan. Uning o'nliklar xonasidagi raqam bilan birliklar xonasidagi raqamni almashtirishdan hosil bo'lgan sonni aniqlovchi programma tuzilsin. (Kirish=123; Natija=132).
7. 999 dan katta bo'lgan son berilgan. Bir marta bo'lib butunni va bo'lib qoldiqni olish operatsiyasidan foydalanib berilgan sonni yuzliklar xonasidagi sonni aniqlovchi programma tuzilsin.
8. 999 dan katta bo'lgan son berilgan. Bir marta bo'lib butunni va bo'lib qoldiqni olish operatsiyasidan foydalanib berilgan sonni mingliklar xonasidagi sonni aniqlovchi programma tuzilsin.
9. Kun boshidan boshlab N sekund vaqt o'tdi. Kun boshidan boshlab qancha minut to'la o'tganligini aniqlovchi programma tuzilsin.
10. Kun boshidan boshlab N sekund vaqt o'tdi. Kun boshidan boshlab qancha to'la soat o'tganligini aniqlovchi programma tuzilsin.

11. Kun boshidan boshlab  $N$  sekund vaqt o'tdi. Kun boshidan boshlab qancha minut va sekund o'tganligini aniqlovchi programma tuzilsin.
12. Kun boshidan boshlab  $N$  sekund vaqt o'tdi. Kun boshidan boshlab qancha soat va sekund o'tganligini aniqlovchi programma tuzilsin.
13. Kun boshidan boshlab  $N$  sekund vaqt o'tdi. Kun boshidan boshlab qancha soat, minut va sekund o'tganligini aniqlovchi programma tuzilsin.
14. Hafta kunlari quyidagi tartibda berilgan: 0 – yakshanba, 1 – dushanba, 2 – seshanba, ... , 6 – shanba. 1-365 oraliqda yotuvchi  $K$  soni berilgan. Agar 1-yanvar dushanba bo'lsa, kiritilgan  $K$  – kun haftaning qaysi kuniga to'g'ri kelishini aniqlovchi programma tuzilsin.
15. Hafta kunlari quyidagi tartibda berilgan: 0 – yakshanba, 1 – dushanba, 2 – seshanba, ... , 6 – shanba. 1-365 oraliqda yotuvchi  $K$  soni berilgan. Agar 1-yanvar payshanba bo'lsa, kiritilgan  $K$  – kun haftaning qaysi kuniga to'g'ri kelishini aniqlovchi programma tuzilsin.
16. Hafta kunlari quyidagi tartibda berilgan: 1 – dushanba, 2 – seshanba, ... , 6 – shanba, 7 – yakshanba, 1-365 oraliqda yotuvchi  $K$  soni berilgan. Agar 1-yanvar seshanba bo'lsa, kiritilgan  $K$  – kun haftaning qaysi kuniga to'g'ri kelishini aniqlovchi programma tuzilsin.
17. Hafta kunlari quyidagi tartibda berilgan: 1 – dushanba, 2 – seshanba, ... , 6 – shanba, 7 – yakshanba, 1-365 oraliqda yotuvchi  $K$  soni berilgan. Agar 1-yanvar shanba bo'lsa, kiritilgan  $K$  – kun haftaning qaysi kuniga to'g'ri kelishini aniqlovchi programma tuzilsin.
18. Hafta kunlari quyidagi tartibda berilgan: 1 – dushanba, 2 – seshanba, ... , 6 – shanba, 7 – yakshanba ( $N$  1-7gacha bo'lgan hafta kunlari soni). 1-365 oraliqda yotuvchi  $K$  soni berilgan. Agar 1-yanvar  $N$  chi kunga to'g'ri kelsa, kiritilgan  $K$  – kun haftaning qaysi kuniga to'g'ri kelishini aniqlovchi programma tuzilsin.
19.  $A$ ,  $B$ ,  $C$  butun sonlar berilgan. Tomonlari  $A \times B$  bo'lgan to'g'ri to'rtburchakka tomoni  $C$  bo'lgan kvadrat eng ko'p joylashtirilsin. To'g'ri to'rt burchakka eng ko'p joylashgan kvadratlar soni va joylashmay qolgan qismi yuzasini aniqlovchi programma tuzilsin.
20. Qaysidir yil berilgan. Berilgan yilning qaysi yuz yillikka kirishini aniqlovchi programma tuzilsin. (Masalan: 20 – yuz yillikning boshi 1901 yil).
21. a)  $b \cdot \tan(x) + |a + 2 \cdot x - e^a + b|$ ,  $a=2.1$ ,  $b=34$ ,  $x=9.7$ ;  
 b)  $2,45 \cdot \text{ctg}(x + 2\pi/7) - f/g \cdot a^2x + x^3 + 2f$ ;



$$\frac{\cos^3(x) + \sqrt{a^{2x}} - e^{x+1}}{x + \frac{y}{z + \frac{y}{x}} + (3a^2 + 5b - 1)^4}$$

c)

$$d) \frac{19,32 \cdot 3\pi - \sqrt[3]{x} \cdot \ln y}{b \cdot \sqrt{c} - (e^{-x} - e^y)} \operatorname{Ctg} \frac{\pi}{13}$$

e)

$$\sqrt[3]{x \lg 3} - \sqrt{y} \cdot e^{5x} - \sqrt[5]{\frac{|x|}{y^3}}$$

*ifodalarning qiymatlarini hisoblash dasturlarini tuzing*

22. *Berigan sonning butun va kasr qismlarini aniqlash dasturini tuzing.*
23. *Besh xonali natural sonning raqamlari yig'indisini topish dasturini tuzing.*
24. *Sakkiz xonali natural sonning o'ngdan birinchi, uchinchi, oltinchi, ettinchi raqamlari ko'paytmasini topish dasturini tuzing.*
25. *Yetti xonali natural sonning chapdan juft o'rindagi raqamlari yig'indisini, toq o'rindagi raqamlari o'rta geometrigini nopish dasturini tuzing.*
26. *Ayirmasi 3 ga, birinchi hadi 4 ga teng bo'lgan arifmetik progressiyaning 9-hadi bilan 32-hadlari ko'paytmasini topish dasturini tuzing.*
27. *Ikki xonali natural sonning raqamlari o'rni almashtirilganda uning qiymati qanday o'zgarishini aniqlash dasturini tuzing.*
28. *Radiusi R bo'lgan doiraning yuzi, n gradusli sektori, m gradusli segmenti yuzlarini aniqlash dasturini tuzing.*
29. *Ikkita tomoni a va b, ularning birining qarshisidagi burchagi n gradus bo'lgan uchburchakning perimetri, yizi, qolgan ikkita burchagini topish dasturini tuzing.*
30. *Tomonlari a, b, c bolgan uchburchakga ichki va tashqi chizilgan aylanalar yuzlarini aniqlash dasturini tuzing.*

### **3.3 C++ DASTURLASH TILIDA TARMOQLANUVCHI JARAYONLARNI DASTURLASH**

*Reja:*

1. *Qisqa shartli operator va uning umumiy ko'rinishi;*
2. *To'liq shartli operator va uning umumiy ko'rinishi;*
3. *?: operatori va uning umumiy ko'rinishi.*

Chiziqli jarayonlarni dasturlashda buyruqlar ketma-ket bajariladi, tarmoqlanuvchi jarayonlarni dasturlashda esa, buyruqlar ma'lum bir shartlar asosida tarmoqlanish bo'yicha bajariladi. C++ dasturlash tilida tarmoqlanuvchi (shartli) jarayonlarni bir necha turlarga bo'lingan holda dasturlar

tuziladi. Tarmoqlanuvchi jarayonlarni amalga oshiruvchi operatorlarni qisqacha qilib shartli operatorlar deb yuritimiz. Shartli operatorlar ham qisqa, to'liq va sodda shartli operator ko'rinishlarda tarmoqlanuvchi jarayonlarni amalga oshiradi.

### **if operatori**

Tarmoqlanuvchi jarayonlarni C++ dasturlash tilida amalga oshirish uchun, albatta, tarmoqlanuvchi algoritmlar asosida bajariladi. Algoritm bajarilish vaqtida ma'lum bir shartlar asosida algoritmning u yoki bu qismi bajarilishini ta'minlash maqsadida shartli operatorlardan foydalaniladi.

*Tarif: Algoritm tarkibidagi shart asosida algoritmning tarmoqlarga bo'linishiga xizmat qiluvchi operatorlar **shartli operatorlar** deyiladi.*

Hayotdagi asosiy ko'p masalalarni dasturlash vaqtida, albatta, tarmoqlanuvchi algoritmlar asosida bajariladi. Tarmoqlanuvchi algoritmlar ham ikki xil holatni o'z ichiga oladi, ya'ni shart rost bo'lganda ma'lum bir vazifani yolg'on bo'lganda hech qanday vazifani bajarmaydi va shart rost bo'lganda ma'lum bir vazifani yolg'on bo'lganda boshqa vazifani bajaradi. Demak, yuqoridagi ikki holatni e'tiborga olib, shartli operatorlar ham qisqa va to'liq ko'rinishga ega.

Qisqa shartli operatorning umumiy ko'rinishi quyidagicha

**if <(shart)> {operator1; operator2; ... operatorn;}**

if operatori tarkibida shartlar ikki va undan ortiq bo'lsa oldingi bobdagi mantiqiy ifodalar asosida birlashtiriladi. if operatori tarkibidagi shart faqat chin bo'lgandagina {} ichidagi operatorlar bajariladi.

***Misol:** Uchta natural son berilgan bo'lsa, agar oxirgi ikkitasini yig'indisi birinчисiga teng bo'lsa, u holda birinчисini oxirgi ikkitasini ayirmasi bilan almashtiring aks holda o'zgarishsiz qoldirilsin.*

```
#include <iostream.h>
#include <math.h>
int main ()
{
    int a,b,c;
    cin>>a>>b>>c;
    if(a=(b+c)) { a=b-c; }
    cout<<a<<b<<c;
    return 0;
}
```

Berilgan masala yechimida faqat bitta shart tekshirildi, ya'ni birinchisi oxirgi ikkitasini yig'indisi bilan tekshirildi, chin bo'lganda  $a=b-c$  bajarildi yolg'on bo'lganda o'zgarishsiz qoldirildi.

Blok ichida o'zgaruvchi va o'zgarmaslarni ham e'lon qilish mumkin. Lekin bu o'zgaruvchi va o'zgarmaslardan faqat blok ichidagina foydalanish mumkin, ammo blokdan tashqarida foydalanib bo'lmaydi. Blok ichidagi har bir buyruq (;) bilan yakunlanishi shart, ammo blokdan keyin (;) qo'yilmasligi ham mumkin.

**Misol:** Tanlangan to'rt xonali son musbat bo'lsa, uning raqamlari yig'indisini, manfiy bo'lsa, raqamlari ko'paytmasini topish dasturini tuzing.

```
#include <iostream.h>
#include <math.h>
using namespace std;
int main ()
{
    int x, a; int const d=36;
    cout << "\n Tanlangan sonni kiriting"<<endl;
    cin >> x;
    if (x > 0){ int a, b, c, d; a=x/1000; b=x%1000/100;
    c=x/10%10; d=x%10;
    cout << "\n Bu son musbat bo'lganligi uchun raqamlari yig'indisi="
    <<a+b+c+d<<endl;}
    if (x<0) { int x1, y, z, t; x1=x/1000; y=x%1000/100;
    z=x/10%10; t = x %10;
    cout << "\n Bu son manfiy bo'lganligi uchun raqamlari ko'paytmasi =" <<
    x1*y*z*t<<endl;}
    system ("pause");
    return 0;
}
```

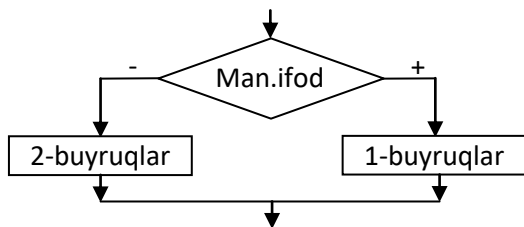
Agar 2345 soni kiritilsa, dastur natijasi quyidagicha bo'ladi:  
Bu son musbat bo'lganligi uchun raqamlari yig'indisi = 14  
kabi bo'ladi.

### *if else operatori*

C++ dasturlash tilida tarmoqlanuvchi jarayonlarni to'liq shartli ko'rinishini ifodalash uchun *if else* operatoridan foydalaniladi. Algoritm tarkibidagi shartlar chin qiymat qabul qilganda ma'lum bir operatorlar, yolg'on bo'lganda boshqa operatorlar bajariladi. Berilgan shart ( mantiqiy

ifoda)ning qiymati **true (rost)** yoki **false (yolg'on)**ligiga bog'liq holda u yoki bu buyruqlar ketma-ketligini bajarish zarur bo'lgan holda tarmoqlanish buyrug'ining if-else ko'rinishidan ifodalaniadi.

Tarmoqlanish algoritmining blok sxema ko'rinishi va bajarilish tartibi bizga algoritmlashtirish asoslari bobidan ma'lum.



Tarmoqlanuvchi jarayonlarni C++ dasturlash tilida to'liq shartli operatorning umumiy ko'rinishi

quyidagicha

```
if <(shart)> {operator1; operator2; ... operatorn;}
else {operator_x1; operator_x2; ... operator_xn;}
```

*if else* operatori tarkibidagi shartlarning chin qiymat qabul qilganda {operator1; operator2; ... operatorn;} bajariladi aks holda { operator\_x1; operator\_x2; ... operator\_xn;} bajariladi.

*Misol:* quyidagi masalani C++ tilida dasturiy ta'minotini yaratish.

$$f(x) = \begin{cases} 2 \cdot \sin(x), & \text{agar } x > 0; \\ x - 6, & \text{agar } x \leq 0; \end{cases}$$

```
#include <iostream.h>
#include <math.h>
int main ()
{
    int x; double y;
    cin>>x;
    if(x>0) { y=2*sin(x);}
        else { y=x-6; }
    cout<<y;
    return 0;
}
```

**Misol2:** Berilgan musbat sonni kabisa yili ekanligini aniqlang. 4 ga karrali yillar va 100 ga karralilar ichida faqat 400 ga karralilari Kabisa yili hisoblanadi.

Masalan: 16,24,2016,2020, 2400 lar kabisa va 5,100,200, 2017 lar kabisa yili emas.

```

#include <iostream.h>
int main ()
{   int x;
    cin>>x;
    if(x%100==0)
        { if(x%400==0) { cout<<"kabisa yili";}
          else {cout<<"kabisa yili emas"}
        }
    else
        { if(x%4==0) {cout<<"kabisa yili";}
          else {cout<<"kabisa yili emas"}
        }
    return 0;
}

```

**Dastur natijasi:**

x=16 bo'lsa kabisa yili

x=100 kabisa emas

x=800 kabisa yili

x=2017 kabisa emas

Shartli o'tish operatorida o'zgaruvchilarni e'lon qilib bo'lmaydi. Ammo bloklarning ichida e'lon qilish mumkin, lekin oldin qayd qilganimizdek, bunday o'zgaruvchilardan blokdan tashqarida foydalanib bo'lmaydi. Bu holat bilan quyidagi dastur orqali tanishib, natijalarni tahlil qiling.

```

#include <iostream.h>
using namespace std;
int main ()
{
    int x=2,y,i=2,j=4;
    if (x>=3) { int i, j; i = 5; j = 8; y = i + j;};
    y= i+j;
    cout << "\n y =" << y << "\n';
    system ("pause");
    return 0;
}

```

Dastur natijasi  $y=6$  kabi bo'ladi. Bir qarashda dasturni tahlil qilib,  $x \geq 3$  mantiqiy ifodaning qiymati yolg'on bo'lganligi uchun natija shunday chiqqandek tuyuladi. Ammo 5 – satrni  $\text{int } x=5, y, i=2, j=4;$  kabi o'zgartirsak ham natija o'zgarmaydi. Chunki  $i, j$  larning blok ichidagi qiymatlari blokdan tashqarida unutiladi.

### ?: operatori

Tarmoqlanuvchi jarayonlarni algoritmlash vaqtida, albatta, shartga e'tibor berish talab etiladi, chunki algoritm shart bo'yicha tarmoqlanib qismlarga ajralib ketadi. Masalan, dastur bajarilishining birorta qadamida qandaydir shartni tekshirish natijasiga ko'ra boshqaruvni dasturning u yoki bu bo'lagiga uzatish mumkin.

Tarmoqlanuvchi jarayonlarni algoritmlash vaqtida shartning bajarilishiga qarab buyruqlar ketma ketligi aniqlanadi. Masalan, dastur bajarilishining birorta qadamida qandaydir shartni tekshirish natijasiga ko'ra boshqaruvni dasturning u yoki bu bo'lagiga uzatish mumkin.

**?: shart amali:** C++,Java dasturlash tilida “?” amali ham aniqlangan bo'lib tekshirilayotgan shart nisbatan sodda bo'lsa, shart amalining  $\langle\langle ? \rangle\rangle$  ko'rinishini ishlatish mumkin. **?: shart amalining** umumiy ko'rinishi quyidagicha.

**$\langle\text{mantiqiy ifoda}\rangle ? \langle\text{ifoda1}\rangle : \langle\text{ifoda2}\rangle;$**

Shart amali **if shart** operatoriga o'xshash holda ishlaydi: agar  $\langle\text{mantiqiy ifoda}\rangle$  0 qiymatidan farqli yoki true bo'lsa,  $\langle\text{ifoda1}\rangle$ , aks holda  $\langle\text{ifoda2}\rangle$  bajariladi. Odatda ifodalar qiymatlari birorta o'zgaruvchiga o'zlashtiriladi.

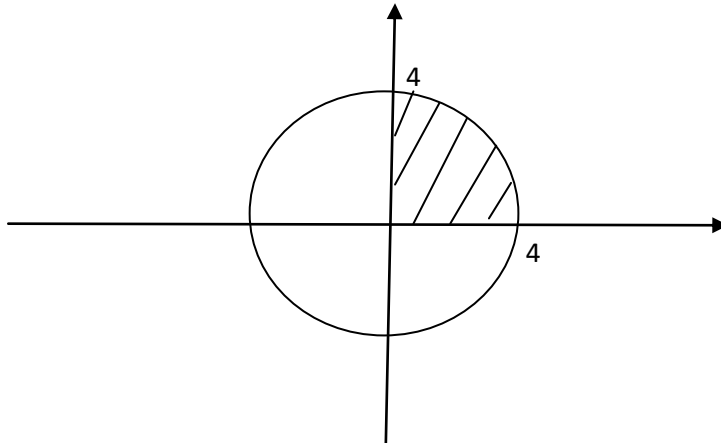
*Misol: Quyidagi masalani C++ tilida dasturiy ta'minotini yarating.*

$$y = \begin{cases} x^2 - 4x, & x > 0 \\ x - \sin x, & x \leq 0 \end{cases}$$

```
#include <iostream.h>
#include <math.h>
int main ()
{   int x; double y;
    cin>>x;
    (x>0)? y=x*x-4*x:
        y=x-sin(x);
    cout<<y;
    return 0;
```

}

**Misol:** Berilgan  $(x,y)$  nuqta quyidagi sohaga tegishli yoki tegishli emasligini aniqlang.



Masala yechimini aniqlash jarayonida berilgan nuqta  $x \geq 0$ ,  $y \geq 0$  va chorak aylana ichida yotish kerak. Masala yechimi dasturi quyidagicha.

```
#include <iostream.h>
int main ()
{   int x,y;
    cin>>x>>y;
    (((x>0)&&(y>0))&&(sqr(x)+sqr(y)<=16))? cout<<"teginshli":
        cout<<"teginshli emas";
    cout<<y;
    return 0;
}
```

C++ tilida shart operatorida umumiy bo'lgan o'zgaruvchilarni e'lon qilish man etiladi, lekin undagi bloklarda o'zgaruvchilarni e'lon qilish mumkin va bu o'zgaruvchilar faqat blok ichida amal qiladi.

Tarmoqlanuvchi jarayonlarini algoritmlarida uchraydigan shartlarga asosan yuqorida keltirib o'tilgan uch xil shartli operatorlarning qulay birortasidan foydalaniladi.

### Nazariy savollar.

- 1 Tarmoqlanuvchi algoritmlar?
- 2 Tarmoqlanuvchi jarayonlarni dasturlashning necha xil usuli mavjud?
- 3 To'liq shartli operatorining ta'rifi va uning umumiy ko'rinishi?
- 4 Qisqa shartli operatorining ta'rifi va uning umumiy ko'rinishi?

5 Sodda shartli operatorining ta'rifi va uning umumiy ko'rinishi?

6. Tarmoqlanish buyrug'ida qachon qavs operatorlaridan foydalaniladi?

7. ?: operatorini va uning bajarilish tartibini tushuntiring.

8. Tarmoqlanish buyruqlaridan if va if-else ko'rinishlarining bajarilish tartibidagi farqlarni tushuntiring.

9. Ichma-ich joylashgan Tarmoqlanish buyruqlarining umumiy ko'rinishini yozib, bajarilish tartibini tushuntiring.

### **Mustaqil ishlash uchun topshiriqlar**

1. Ikkita  $A$  va  $B$  butun tipli o'zgaruvchilar berilgan. Agar ularning qiymatlari o'zaro teng bo'lmasa, har bir o'zgaruvchiga bu qiymatlar yig'indisi ta'minlansin, agar o'zaro teng bo'lsa, o'zgaruvchilarga  $0$  ta'minlansin.  $A$  va  $B$  ning yangi qiymatlari ekranga chiqarilsin.

2. Uchta son berilgan. Ularning kichigini aniqlovchi dastur tuzilsin.

3. Uchta son berilgan. Ularning o'rtasidagi (ya'ni kattasi va kichigi orasida joylashgan) sonni aniqlovchi dastur tuzilsin.

4. Uchta son berilgan. Ularning avval kichigini keyin kattasini ekranga chiqaruvchi dastur tuzilsin.

5. Uchta son berilgan. Shu sonlarning yig'indisi eng katta bo'ladigan ikkitasini ekranga chiqaradigan dastur tuzilsin.

6. Uchta  $A$ ,  $B$  va  $C$  haqiqiy tipli o'zgaruvchilar berilgan. Agar ularning qiymatlari o'sish tartibida berilgan bo'lsa, ularni ikkilantiring, aks holda har bir o'zgaruvchining qiymati qarama-qarshisiga o'zgartirilsin.  $A$ ,  $B$  va  $C$  larning yangi qiymatlari ekranga chiqarilsin.

7. Uchta  $A$ ,  $B$  va  $C$  haqiqiy tipli o'zgaruvchilar berilgan. Agar ularning qiymatlari o'sish yoki kamayish tartibida berilgan bo'lsa, ularni ikkilantiring, aks holda har bir o'zgaruvchining qiymati qarama-qarshisiga o'zgartirilsin.  $A$ ,  $B$  va  $C$  larning yangi qiymatlari ekranga chiqarilsin.

8. Uchta butun son berilgan. Shu sonlarning ikkitasi o'zaro teng, qolgan bittasini tartib raqami aniqlansin.

9. To'rtta butun son berilgan. Shu sonlarning uchasi o'zaro teng, qolgan bittasini tartib raqami aniqlansin.

10. Sonlar o'qida uchta  $A$ ,  $B$ ,  $C$  nuqtalar berilgan.  $A$  nuqtaga eng yaqin nuqta va ular orasidagi masofa topilsin.

11. Kordinatalar tekisligida butun son berilgan. Agar nuqta koordinata boshida yotsa,  $0$  chiqarsin. Agar nuqta  $OX$  yoki  $OY$  o'qlarida joylashsa mos holda  $1$  va  $2$  chiqarilsin. Agar nuqta koordinata o'qida joylashmasa  $3$  chiqarilsin.



12.  $OX$  va  $OY$  koordinata o'qlarida yotmaydigan nuqta berilgan. Nuqta joylashgan koordinata choragi aniqlansin.

13. Koordinata o'qlariga parallel ravishda to'g'ri to'rtburchakning uchta uchi berilgan, to'rtinchi uchi koordinatasini aniqlansin.

14.  $X$  haqiqiy soni berilgan. Quyidagi funksiya hisoblansin.

$$f(x) = \begin{cases} 2 \cdot \sin(x), & \text{agar } x > 0; \\ x - 6, & \text{agar } x \leq 0; \end{cases}$$

15.  $X$  haqiqiy soni berilgan. Quyidagi funksiya hisoblansin.

$$f(x) = \begin{cases} 2 \cdot x, & \text{agar } x < -2 \text{ yoki } x > 2; \\ -3 \cdot x, & \text{aks holda;} \end{cases}$$

16.  $X$  haqiqiy soni berilgan. Quyidagi funksiya hisoblansin.

$$f(x) = \begin{cases} -x, & \text{agar } x \leq 0; \\ x^2, & \text{agar } 0 < x < 2; \\ 4, & \text{agar } x \geq 2; \end{cases}$$

17.  $X$  haqiqiy soni berilgan. Quyidagi funksiya hisoblansin.

$$f(x) = \begin{cases} 0, & \text{agar } x < 0; \\ 1, & \text{agar } x \in [0,1), [2,3), \dots; \\ -1, & \text{agar } x \in [1,2), [3,4), \dots; \end{cases}$$

18. Yil berilgan (musbat butun son). Berilgan yilda nechta kun borligini aniqlovchi dastur tuzilsin. Kabisa yilida 366 kun bor, kabisa bo'lmagan yilda 365 kun bor. Kabisa yil deb 4 ga karralilariga aytiladi. Lekin 100 ga karrali yillar ichida faqat 400 ga karrali bo'lganlari kabisa yil hisoblanadi. Masalan 300, 1300 va 1900 kabisa yili emas. 1200 va 2000 kabisa yili.

19. Butun son berilgan. Berilgan sonni "musbat toq son", "manfiy juft son", "son nolga teng" va h.k ekranga yozadigan dastur tuzilsin.

20. 1-999 oralig'idagi sonlar berilgan. Berilgan sonni "ikki xonali juft son", "uch xonali toq son" va h.k ekranga yozadigan dastur tuzilsin.

21.  $x, y, z, t$  sonlarining eng kattasi va eng kichigini aniqlang.

22.  $M(x; y)$  nuqta koordinata tekisligining qayerida joylashganligini aniqlang.

23. Berilgan oddiy kasrning turini aniqlang.

24.  $x_1, x_2, x_3$  sonlari berilgan va ularning faqat ikkitasi teng. Farq qiluvchi sonni toping.

25.  $y=kx+b$  funksiya grafigining qaysi koordinata choragida joylashganligini aniqlang.

26. *Ikkita chiziqli funksiya berilgan. Ularning grafiklari o'zaro qanday joylashganligini aniqlang.*
27. *Teledasturning bir qismini dasturlashtiring.*
28. *Har qanday sonning kvadrati qanday raqam bilan tugashini aniqlang.*
29.  *$y=\sin x$ ,  $y=\cos x$ ,  $y=\operatorname{tg} x$ ,  $y=\operatorname{ctg} x$  funksiyalarning koordinata choraklaridagi ishoralarini aniqlang.*
30. *Yil oylarining qaysi faslga tegishlilikini aniqlang.*
31. *Ixtiyoriy kiritilgan ko'p xonali sonning 2, 3, 4, 5, 6, 10 sonlaridan qaysi biriga karrali ekanligini aniqlang.*

### **3.4 C++ DASTURLASH TILIDA TANLASH JARAYONLARI VA ULARNI DASTURLASH**

*Reja:*

1. *Break operatori va uning umumiy ko'rinishi;*
2. *Continue operatori va uning umumiy ko'rinishi;*
3. *Tanlash operatori va uning umumiy ko'rinishi.*

C++ dasturlash tilida shartli algoritmning yana bir turi mavjud, bu tanlash asosida algoritmning bajarilishidir. Algoritm tarkibida shartli operatorning natijasi ikki va undan ortiq ham bo'lishi mumkin. Agar shartning natijasi ikki va undan ortiq bo'lsa, bunday holatlarda tanlash jarayonlaridan foydalaniladi.

#### **Break operatori**

Ba'zan dastur yoki qaysidir buyruqlar to'liq ijro etilmasdan, qo'qqisidan ishni to'xtatib (avariya holati) dastur yoki buyruqdan chiqib ketishga to'g'ri keladi. Ayniqsa, bunday holat takrorlanish va tanlash buyruqlarini tashkil qilishda ko'p uchraydi. Bunday vaziyatlarda **break (uzish)** operatoridan foydalaniladi. C++ break operatorini uchratgach, dasturni bajarishni to'xtatadi va break operatori joylashgan blokdan chiqib, undan keyingi buyruqlarni bajarishga o'tadi. Demak, unutmazlik kerakki, break operatori faqat o'zi joylashgan blokdan chiqib ketish imkonini beradi. Lekin tanlash, sikl buyruqlari ichida break operatoridan foydalanilsa, o'sha buyruqning o'zini blok deb tushunadi. Bu operatoridan zarurat bo'lganda dasturning ixtiyoriy joyida foydalanish mumkin. Ammo chuqur o'ylamasdan foydalanmaslik tavsiya etiladi. Masalan, `int main( )` funksiyasi blokida foydalanilsa, `break;` operatoridan keyingi buyruqlar bajarilmaydi. Bu holda break operatori `int main( )` funksiyasi blokidan chiqishni ta'minlaydi va natijada dasturni bajarish to'xtatiladi. Keyingi mavzularda aniq misollar yordamida bu operator bilan yanada yaqinroq tanishamiz.

C++ dasturlash tilida asosan takrorlanish jarayonida takrorlanishlar soni uning tarkibidagi ifodaga bog'liq bo'lib qoladi shunday vaziyatlarda takrorlanishni to'xtatish uchun break operatoridan foydalanish maqsadga muvofiq. Agar **break** operatori dastur asosiy blokida bo'lsa, break operatoridan keyingi operatorlar bajarilmasdan qoladi.

*Misol: Berilgan dastur natijasini aniqlang.*

```
#include <iostream.h>
int main ()
{
    int x=3;
    for(int i=1;i<=x;i++){ cout<<i; break;}
    return 0;
}
```

**Dastur natijasi**

**1**

Yuqoridagi dastur bajarilishi davrida break operatorigacha bo'lgan operatorlar bajariladi qolganlari esa bajarilmasdan dastur shu blokdan chiqib ketadi. Chunki break operatori takrorlanishi bloki ichida joylashgan, shuning uchun dastur natijasi 1 ga teng bo'ladi.

*Misol: Berilgan dastur natijasini aniqlang.*

```
#include <iostream.h>
int main ()
{
    int x=3;
    for(int i=1;i<=x;i++)
        {if(i==2) break; cout<<i; }

    return 0;
}
```

**Dastur natijasi**

**1**

Yuqoridagi dastur bajarilish davrida takrorlanish operatori faqat bir marta bajariladi, chunki break operatori bajarilgandan so'ng takrorlanish blokidan chiqib ketadi. *break* operatorini faqat tarmoqlanuvchi jarayon ichida ishlatish mumkin emas. Yuqoridagi misolda tarmoqlanish jarayoni takrorlanish ichida joylashganligi uchun bajarilyapti.

## Continue operatori

Dasturlash tillarida ma'lum bir holatlarda dastur tarkibidagi buyruqlar faqatgina bir qadamga tashlab keyingisi bajarilish kerak bo'lgan holatlar ham mavjud. Bunday holatlarda break operatoridan foydalana olmaymiz, chunki break o'zi turgan blokdan chiqib ketadi. Dastur tanasida bitta buyruq bajarilmasdan keyingisiga o'tish uchun **continue** operatoridan foydalaniladi.

**Continue** operatori ham break operatoriga o'xshaydi, lekin bunda faqat bitta bo'yruqni cheklab o'tadi. Asosan takrorlanish jarayonlarida ma'lum bir holatlarda takrorlanishning ba'zi bir holatlari hisoblanmaslik kerak bo'lganda continue operatoridan foydalanish maqsadga muvofiq bo'ladi.

Takrorlanish jarayonida takrorlanishning biror bir qadamini tashlab ikkinchisiga o'tish uchun continue bo'yrug'idan foydalanish qulay hisoblanadi.

*Misol: Berilgan dastur natijasini aniqlang.*

```
#include <iostream.h>
int main ()
{   int x=4;
    for(int i=1;i<=x;i++)
    {
        if((i==2)||(i==4)) {continue;}
        cout<<i;
    }
    return 0;
}
```

**Dastur natijasi**

**1 3**

Yuqoridag dasturda for operatori tarkibidagi continue operatori faqatgina i=2 va i=4 bo'lganda keyingi qadamga o'tib ketadi.

**Misol.** 1 dan 10 gacha bo'lgan sonlar ichida 9 va 5 sonlaridan boshqa sonlar yig'indisini hisoblash dasturini tuzing.

```

#include <iostream.h>
using namespace std;
int main()
{
    int x,y,s=0;
    for(x=1;x<=10;x++) {if((x==9)||(x==5)) continue;s+=x;};
    cout<<"s="<<s<<"\n";
    system("pause");
    return 0;
}

```

Dastur natijasi **s=41** kabi bo‘ladi. Chunki sikl parametrining **9** va **5** ga teng qiymatlarini **continue** operatori yordamida o‘tkazib yuboradi va 1 dan 10 gacha bo‘lgan boshqasonlar yig‘indisini hisoblaydi.

Agar ushbu dasturda **{if(x==9) continue;s+=x;};** blokni **{if(x>5) break;s+=x;};** kabi o‘zgartirsak, natija **s=15** kabi bo‘ladi. Chunki **x=6** bo‘lganda **break** operatori sikl tanasining bajarilish jarayonini to‘xtatadi.

**break** va **continue** operatorlaridan tarmoqlanish buyrug‘i bloklarida foydalanib b‘lmaydi. Masalan,

**if(x>=6) {x=x+a; beak;}; else {x=x-a; continue};** kabi buyruqlarni kompilyator xato deb hisoblaydi. Ammo yuqorida keltirilganidek, **if(x>=6) beak; else continue;** kabi buyruqlarga ruxsat etiladi.

### Tanlash operatori

Dasturlash tillarida algoritmlarni dasturini tuzish vaqtida shartlarning qiymatlari ikki va undan ortiq bo‘lgan holatlarda dastur tuzuvchilarga bir muncha qiyinchiliklar tug‘diradi. Agar shartlar qiymati ikkitadan ortiq holatlarni e‘tiborga olish kerak bo‘lgan holatlarda, har bir shart uchun alohida kod yozishga to‘g‘ri keladi.

Agar shartlar soni ko‘p bo‘lgan holatlarda, albatta, tanlash operatorlaridan foydalanish maqsadga muvofiqdir. C++ dasturlash tilida tanlash jarayonlarini if qisqa shartli operator ko‘rinishda ham ifodalash mumkin. Bunday holatda, albatta, har bir tanlov uchun shartli kod yoziladi.

***Misol:** Ranglarni ajratadigan robot faqat ranglarni tartib raqamiga qarab ajratadi. Siz robotga ranglarni ajratishga yordam bering(1-oq, 2-qora, 3-qizil, 4-sariq, 5-yashil, 6-kO‘k).*

```

#include <iostream.h>

```

```

int main ()
{
    int i;
    cout<<"ranglarni tartib raqamini kiriting";
    cin>>i;
    if(i==1) cout<<"oq";
    if(i==2) cout<<"qora";
    if(i==3) cout<<"qizil";
    if(i==4) cout<<"sariq";
    if(i==5) cout<<"yashil";
    if(i==6) cout<<"kO'k";
    return 0;
}

```

### Dastur natijasi

i=2 bo'lsa qora  
i=6 bo'lsa kO'k

Tanlash jarayonlarini dasturiy ta'minotini yaratishda doimo qisqa shartli operatorlaridan foydalanilsa, dasturning ishlash holat tezligi susayganligi seziladi, bunday holatlarda ko'p hollarda tanlash jarayonini **switch** operatoridan foydalanish maqsadga muvofiq.

C++ dasturlash tilida tanlash jarayonini tasvirlash uchun switch operatorining umumiy ko'rinishi quyidagicha.

```

switch(<ifoda>)
{
    case <qiymat1>: <operatorlar1>; break;
    case <qiymat2>: <operatorlar2>; break;
    -----
    case <qiymatn>: <operatorlarn>; break;
    default: <operatorlarx>;
}

```

Tanlash operatorining ishlash jarayoni switch operatori tarkibidagi ifoda qiymati *qiymat1* ga teng bo'lsa *operatorlar1*, *qiymat2* ga teng bo'lsa *operatorlar2* va hakoza *qiymatn* ga teng bo'lsa *operatorlarn* bajariladi aks holda **default** kalit so'zidan keyingi *operatorlarx* bajariladi.

**Misol:** Hafta kunlarini tartib raqami berilganda uning qaysi kun ekanligini aniqlash dasturi.

```

#include <iostream.h>
int main ()

```

```

{   int i;
    cout<<"hafta kunlari tartib raqamini kiriting";
    cin>>i;
    switch(i)
    { 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;
      case 7: cout<<"yakshanba"; break;
      default: cout<<"hafta kuni emas";
    }
    return 0;
}

```

Tanlash operatorini tanlash sharti natijasining bir nechta qiymati uchun yagona qiymat tanlansa, bunday holatlarni ham e'tiborga olish mumkin. Tanlash operatorini tanlash sharti natijasining bir nechta qiymati uchun yagona qiymat tanlanishini quyidagi masala orqali ko'rib chiqamiz.

**Misol:** Berilgan oy tartib soni qaysi faslga tegishli ekanligini aniqlash dasturi.

```

#include <iostream.h>
int main ()
{   int i;
    cout<<"oy tartib raqamini kiriting";
    cin>>i;
    switch(i)
    { case 1:
      case 2:
      case 12: cout<<"qish"; break;
      case 3:
      case 4:
      case 5: cout<<"bahor"; break;
      case 6:
      case 7:
      case 8: cout<<"yoz"; break;
      case 9:

```

```

    case 10:
    case 11: cout<<"ko'z"; break;
    default: cout<<"oy tartib raqami emas";
    }
    return 0;
}

```

### **Dastur natijasi**

i=2 bo'lsa qish

i=7 bo'lsa yoz

Yuqoridagi masala shartiga e'tibor bersak uchta qiymat uchun yagona tanlash bajarilyapti.

Demak tanlash jarayonlarini yuqoridagi holatlarni e'tiborga olib C++ dasturlash tilida tasvirlashimiz mumkin.

### **Dasturlash asoslarini o'rganishda quyidagilarga e'tibor bering**

Dasturlashda "qo'pol xato" yoki "arzimas xato" degan xatolar yo'q ekanligini unutmang. Faqat xato bor, xolos. Faraz qilaylik, tarmoqlanish yoki tanlash buyrug'i noto'g'ri tashkil qilingan bo'lsa, bitta operatorning biror harfi tushib qolishi yoki boshqa harf bilan almashtirilishi (masalan, **include** yoki **includ**) yoki bitta tinish belgisining o'z o'rnida qo'yilmasligi xatolar dasturning ijro etilmasligiga olib kelsa, bu xatolarni ahamiyati jihatidan yuqori yoki quyi darajadagi xatolarga bo'lish mumkinmi?

Ko'pincha dasturlashni boshlab o'rganayotgan boshlovchilar dasturda bir nechta xatolarga yo'l qo'yib, ularni topmasdan soatlab vaqt sarflashadi. Yanada aniqroq aytadigan bo'lsak, biror buyruq yoki mavzuni o'rganishga ketadigan vaqtdan unga oid dasturni kompyuterga kiritib, javobini olishga ketadigan vaqt bir necha marta ko'p bo'ladi.

Bunday kamchiliklarni bartaraf etish uchun quyidagilarga e'tibor qiling:

- har bir kalit so'zni aniq aytilish va yozishda xatoga yo'l qo'ymaguncha mashq qiling;
- tinish belgilarining qayerdan va qanday qo'yilishiga e'tibor qiling;
- figurali (sistemali), oddiy qavslarni, apostrof va qo'shtirnoqni ochganda darhol yopib, kursorni bitta chapga qaytarib, keyin ma'lumotlarni tersangiz u yopilmasdan qolmaydi;
- xatolik haqida C++ dasturlash tilining ma'lumotlari bo'yicha to'g'ri harakat qilishni tezroq o'rganing;



- dastlab biror masalaga mustaqil dastur tuzmasdan tuzilgan dasturni o‘rganib chiqib, qaytadan shu dasturni mustaqil tuzish yo‘li bilan o‘z tajribangizni oshirib boring;

- tajribali hamkasblaringiz hamda o‘rtoqlaringizdan oddiy narsalarni ham so‘rashdan uyalmang.

Bularning hammasi vaqtingizning sezilarli darajada tejaliishiga olib keladi.

**Masalan**, tanlash buyrug‘ini o‘rganishda nimalarga e‘tibor berish kerak?

- **switch, case, default** operatorlarini to‘g‘ri yozish;

- **switch ( )** operatoridan keyin tinish belgisi qo‘yilmasligiga;

- **switch** operatori o‘z blokiga egaligiga;

- **case** so‘zidan keyin yoziladigan o‘zgarmas bilan buyruqlar oralig‘ida va **default** bilan buyruqlar oralig‘ida (**:**) borligiga;

- (**:**) dan keyin buyruqlar ko‘p bo‘lsa ham ular blok belgi orasida yozilishi shart emasligiga;

- (**:**) dan keying buyruqlar, albatta, **break**; operatori bilan tugashiga (**default** bundan mustasno).

### Nazariy savollar

1 Tanlash jarayonlari?

2 break operatori va uning ishlash jarayoni?

3 continue operatori va uning ishlash jarayoni?

4 Tanlash jarayonini qisqa shartli operator ko‘rinishida qanday tasvirlanadi?

5 Tanlash jarayonini switch operatori ko‘rinishida qanday tasvirlanadi?

6 switch operatorining umumiy ko‘rinishi va uning ishlash jarayoni?

7. Tanlash buyruqlaridan qachon foydalaniladi?

8. break va continue operatorlari qanday ijro etilishini tushuntiring.

9. Tanlash buyrug‘ining qanday ko‘rinishlari mavjud?

### Mustaqil ishlash uchun topshiriqlar

1. 1–7 gacha bo‘lgan butun sonlar berilgan. Kiritilgan songa mos ravishda hafta kunlarini so‘zda ifodalovchi dastur tuzilsin. (1-Dushanba, 2-Seshanba va h.).

2. K butun soni berilgan. Baho natijalarini chiqaruvchi dastur tuzing. (1-yomon, 2-qoniqarsiz, 3-qoniqarli, 4-yaxshi, 5-a‘lo). Agar k soni 1-5 oraliqqa tegishli bo‘lmasa “xato” deb chiqarilsin.

3. Oy raqami berilgan. Kiritilgan oy qaysi faslga tegishli ekanligini chiqaruvchi dastur tuzilsin. (Masalan: 2 chi oy, “qish”).

4. Oyning raqami berilgan. Shu oyda nechta kun borligini aniqlovchi dastur tuzilsin.

5.  $A$ ,  $B$  haqiqiy va amal butun soni berilgan.  $A$  va  $B$  sonlari ustida arifmetik amallar bajaruvchi dastur tuzilsin. Amal quyidagi qiymatlarni qabul qiladi: 1-qo'shish, 2-ayirish, 3-ko'paytirish, 4-bo'lish.

6. Uzunlik birliklari quyidagi tartibda berilgan. 1-desimetr, 2-kilometr, 3-metr, 4-millimetr, 5-santimetr. Uzunlik birligini bildiruvchi son berilgan. (1-5 oraliqda) va shu birlikdagi kesma uzunligi berilgan (haqiqiy son). Kesmaning uzunligini metrlarda ifodalovchi dastur tuzilsin.

7. Og'irlik birliklari quyidagi tartibda berilgan. 1-kilogramm, 2-milligramm, 3-gramm, 4-tonna, 5-sentner. Og'irlik birligini bildiruvchi soni berilgan va shu birlikdagi og'irlik qiymati berilgan. Og'irlikni kilogrammda ifodalovchi dastur tuzilsin.

8. Sanani bildiruvchi ikkita butun son berilgan:  $D$ (kun) va  $M$ (oy), kabisa bo'lmagan yil sanasi kiritiladi. Berilgan sanani ifodalovchi dastur tuzilsin. Kabisa yilida 366 kun, kabisa bo'lmagan yilda 365 kun bor bo'ladi.

9. Ikkita butun son berilgan:  $D$  (kun) va  $M$  (oy), kabisa bo'lmagan yil sanasi kiritiladi. Berilgan sanadan keyingi sanani ifodalovchi dastur tuzilsin.

10. Robot faqat to'rtta tomonga ko'cha oladi ("s"-shimol, "j"-janub, "q"-sharq, "g"-g'arb) va uchta raqamli kamanda: 0-harakatni davom ettir, 1-chapga buril, 2-O'ngga buril.  $Y$  – robot yo'nalishi va  $K$  – kamanda berilgan. Berilgan kamanda bajarilgandan keying robot holatini aniqlovchi dastur tuzilsin.

11. Lokatr dunyoning bir tomoniga qaratilgan ("s"-shimol, "j"-janub, "q"-sharq, "g"-g'arb) va uchta raqamli kamanda: 0-o'ngga buril, 1-chapga buril, 2-burilish  $180^\circ$ .  $C$  – lokatrning boshlang'ich holati va  $K1$ ,  $K2$  – kamandalar berilgan. Berilgan kamanda bajarilgandan keying lokatr holatini aniqlovchi dastur tuzilsin.

12. Doiraning elementlari quyidagi tartibda nomerlangan: 1-radius  $R$ , 2 — diametr  $D = 2 \cdot R$ , 3 — uzunligi  $L = 2 \cdot \pi \cdot R$ , 4 — doiraing yuzi  $S = \pi \cdot R^2$ . Shu elementlardan bittasi berilgan qolganlarini topuvchi dastur tuzilsin.  $\pi = 3.14$ .

13. Tengyonli uchburchakning elementlari quyidagi tartibda nomerlangan: 1-katet  $a$ , 2-gipotenuza  $c = a \cdot \sqrt{2}$ , 3 — gipotenuzaga tushirilgan balandlik  $h$  ( $h = c/2$ ), 4 — yuzasi  $S = (c \cdot h)/2$ . Shu elementlardan bittasi berilganda qolganlarini topuvchi dastur tuzilsin.

14. Tengtomonli uchburchakning elementlari quyidagi tartibda nomerlangan: 1-tomoni  $a$ , 2-radiusi  $R1$  ( $R1 = a \cdot \sqrt{3}/6$ ), 3 — radius  $R2$  ( $R2 = 2 \cdot R1$ ), 4 — yuzasi

$S = a^2 \cdot \sqrt{3} / 4$ . Shu elementlardan bittasi berilganda qolganlarini topuvchi dastur tuzilsin.

15. o'yin kartasi turlari berilgan: 1-g'isht, 2-olma, 3-chillak, 4-qarg'a. 10 lik kartadan kata kartalar quyidagi qiymatlarni o'zlashtirsin: 11-valet, 12-dama, 13-qirol, 14-tuz. Ikkita butun son berilgan  $N$ -karta qiymati ( $6 \leq N \leq 14$ ) va  $M$  — karta turi ( $1 \leq M \leq 4$ ) kiritilganda karta nomlarini (masalan, "olti qarg'a") chiqarib beruvchi dastur tuzilsin.

16. Yoshi yillarda aniqlovchi 20-69 gacha butun son berilgan. Son kiritilganda unga mos so'zlarda ifodalovchi dastur tuzilsin. ("yigirma yosh", "qirq uch yosh" va h.k.).

17. o'quv masalalarini aniqlovchi 10-40 gacha butun son berilgan. Son kiritilganda unga mos so'zlarda ifodalovchi dastur tuzilsin. ("yigirmata masala", "O'n uchta masala" va h.k.).

18. 100–999 gacha oraliqdagi sonlarni so'zlarda ifodalovchi dastur tuzilsin. (masalan: 123- "bir yuz yigirma uch").

19. Sharq kalendarida 60 yillik davr qabul qilingan. Yil muchali 5 ta rang (yashil, qizil, sariq, oq va qora) va 12 ta hayvon (sichqon, sigir, yo'lbars, quyon, ajdar, ilon, ot, qo'y, maymun, tovuq, it va to'ngizlardan) nomlarning kombinatsiyasidan kelib chiqadi. Yilning raqamiga qarab uning muchalini aniqlovchi dastur tuzilsin. 1984-davr boshi: "Yashil sichqon yili".

20. Ikkita burj vaqtlarini aniqlovchi butun son berilgan:  $D$  (kun) va  $M$  (oy). Berilgan sana qaysi burjga kirishini aniqlovchi dastur tuzilsin. "Qovg'a (20.1-18.2)", "Baliq (19.2-20.3)", "Qo'y (21.3-19.4)", "Buzoq (20.4-20.5)", "Egizklar (21.5-21.6)", "Qisqichbaqa (22.6-22.7)", "Arslon (23.7-22.8)", "Parizod (23.8-22.9)", "Tarozi (23.9-22.10)", "Chayon (23.10-22.11)", "O'qotar (23.11.21.12)", "Echki (22.12-19.1)".

### **3.5 C++ DASTURLASH TILIDA TAKRORLANUVCHI JARAYONLAR VA PARAMETR BO'YICHA DASTURLASH**

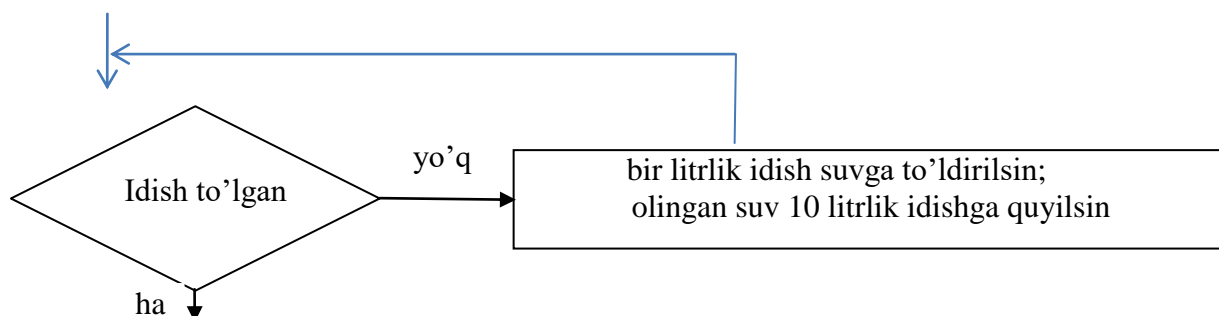
Reja:

1. Takrorlanuvchi jarayonlarni dasturlash;
2. for(sikl) operatori va uning umumiy ko'rinishi;
3. shartsiz o'tish operatori va uning umumiy ko'rinishi.

Amalda shunday masalalar ham uchraydiki, ularni hal qilish dasturlarini hozirgacha tanishgan buyruqlar yordamida tuzib bo'lmaydi. Ular shunday

masalalarki, natijaga erishish uchun qo'yilgan shartga bog'liq holda bir yoki bir nechta amal yoki buyruqlar ketma-ketligini takror-takror bajarishga to'g'ri keladi.

Masalan, bir litrlik idish bilan, o'n litrlik idishni to'ldirish algoritmi quyidagicha bo'ladi:



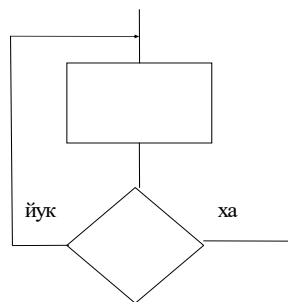
Ko'rinib turibdiki, arifmetik blokdagi ikkita amal(buyruq) 10 marta takror-takror bajarilish oqibatida maqsadga erishiladi. Yoki g'isht teruvchi bir nechta amallar ketma-ketligini talab etilgan balandlikdagi devor hosil bo'lguncha marta takrorlaydi. Takrorlanish buyruqlaridan, ayniqsa, aniq fanlarga doir masalalarni hal qilishda ko'p foydalaniladi. Bunga keyingi mavzular bilan tanishish jarayonida ishonch hosil qilish mumkin. Quyida takrorlanish buyruqlarining uch xil ko'rinishi bilan tanishamiz.

Dasturlash jarayonida ba'zi bir masalalarni algoritmlari tarkibidagi buyruqlar bir necha marta bajarilishiga to'g'ri keladi. Agar algoritm tarkibidagi bir necha marta takrorlanishi kerak bo'lgan buyruqlarni takrorlanuvchi jarayonlar asosida dasturlash tillarida tasvirlanmasa, bu buyruqlarni barchasini bajarish murakkablashadi. Elektron hisoblash mashinalarini insoniyatdan farqi shundaki, insoniyatda bir nechta buyruqlarni bajarish davomida toliqish holatlari bo'lishi mumkin elektron mashinalarga takrorlanishni qanchaligini ma'lum bir buyruqlar asosida berilsa, ular barchasini charchamasdan bajaradi. Tarmoqlanuvchi jarayonlarni masalan matematikada ixtiyoriy ketma ketliklarni yig'indisini hisoblash oddiy usullar bilan hal etilmaydigan holatlarda qo'llash mumkin.

*Tarif: Algoritmning qandaydir qismidagi buyruqlar ikki va undan ortiq bajarilishiga **takrorlanuvchi jarayonlar** deyiladi.*

Yuqoridagi ta'rifga etibor qaratsak, demak algoritmning qandaydir qismi ikki va undan ortiq bajarilishi mumkin bo'lgan holatlar ham mavjud. Bunda dasturchiga shunday vazifa qo'yiladiki takrorlanish holatini bir yaxlit buyruq asosida kompyuterga qulay usulda berish kerak.

Takrorlanuvchi jarayonlarni quyidagi blok sxema ko'rinishda C++ dasturlash tilida tasvirlash mumkin.



Yuqoridagi blok sxema shaklida shart toki chin bo'lgunga qadar takrorlanish bajarilaveradi.

C++ dasturlash tillarida takrorlanuvchi jarayonlarni quyidagi usullar yordamida tasvirlash mumkin.

- Parametr bo'yicha takrorlash(for);
- shartsiz o'tish bo'yicha takrorlash(goto);
- Shart bo'yicha takrorlash(while, do while).

### **for(sikl) operatori**

Takrorlanuvchi jarayonlarni takrorlanish soni aniq bo'lgan holatlardagina parametr bo'yicha takrorlash usulidan foydalaniladi. Takrorlanuvchi jarayonlarni parametr bo'yicha C++ dasturlash tilida tasvirlash uchun, albatta, takrorlanish soniga e'tibor berish kerak. Parametr bo'yicha takrorlanuvchi jarayonlarga, masalan, birdan  $n$  gacha sonlarning kvadratlarini yig'indisini topish bunda takrorlanish soni aniq, ya'ni birdan dan  $n$  gacha deb berilyapti.

Parametr bo'yicha takrorlanuvchi jarayonlar takrorlanish oshishi yoki kamayishiga qarab ikki turga bo'linadi:

- qadam +1ga teng bo'lgan takrorlanish;
- qadam -1ga teng bo'lgan takrorlanish.

Takrorlanuvchi jarayonlarni dasturlash vaqtida takrorlanish qadami birga oshib borilsa, birinchi turdan foydalaniladi. Takrorlanish qadami birga oshib boruvchi parametr bo'yicha sikl operatorining umumiy ko'rinishi quyidagicha.

**for(<o'zgaruvchi>=<boshlang'ich qiymat>;<o'zgaruvchi> <= <oxirgi qiymat>; <o'zgaruvchi>++)**

for operatori takrorlanish sonining boshlang'ich va oxirgi qiymatlari aniq bo'lgandagina ishlatiladi. For operatorining ishlash prinsipi **<o'zgaruvchi> = <boshlang'ich qiymat>** dan takrorlanish boshlanib toki **<o'zgaruvchi> <= <oxirgi qiymat>** gacha davom etadi, **<o'zgaruvchi>++** bu takrorlanishning bir qadamga oshishni ta'minlaydi.

**Misol:** 1 dan  $n$  gacha sonlarning kvadratlar yig'indisini hisoblash dasturini tuzing.

```
#include <iostream.h>
#include <math.h>
int main()
{   int n,s=0;
    cout<<"n=";
    cin>>n;
    for(int i=1;i<=n;i++)
        s=s+sqr(i);
    cout<<" s="<<s;
    return 0;
}
```

Dastur natijasi:

n=3  
s=14

**Misol:** Quyidagi yig'indini hisoblash dasturini tuzing.

$$s = \frac{\sin 1}{1} + \frac{\sin^2 2}{2} + \frac{\sin^3 3}{6} + \dots + \frac{\sin^n n}{n!}$$

Bu masalani hisoblash jarayonida sikl operatori tarkibida yigindini suratini hisoblash uchun alohida funksiya, maxrajini hisoblash uchun alohida funksiya va yig'indi uchun alohida funksiya yaratish kerak.

```
#include <iostream.h>
#include <math.h>
int main()
{   int n,t,p; double s=0;
    t=1; p=1;
    cout<<"n=";
    cin>>n;
    for(int i=1;i<=n;i++)
    {   t=t*sin(1); // suratni hisoblash
        p=p*i; // maxrajni hisoblash
        s=s+t/p; // yig'indini hisoblash
    }
}
```

```

    cout<<" s="<<s;
return 0;
}

```

Takrorlanuvchi jarayonlarni dasturlash vaqtida takrorlanish qadami birga kamayib borilsa, ikkinchi turdan foydalaniladi. Takrorlanish qadami birga kamayib boruvchi parametr bo'yicha sikl operatorining umumiy ko'rinishi quyidagicha.

**for(<o'zgaruvchi>=<oxirgi qiymat>;<o'zgaruvchi> >= < boshlang'ich qiymat>; <o'zgaruvchi>- -)**

for operatorining qadam -1 bo'lgan holati takrorlanish soni kamayish tartibida bajariladi. For operatorining ishlash prinsipi <o'zgaruvchi> = < oxirgi qiymat> dan takrorlanish boshlanib toki <o'zgaruvchi>>=< boshlang'ich qiymat> gacha davom etadi, <o'zgaruvchi>-- bu takrorlanishning bir qadamga oshishkamayishini ta'minlaydi.

*Misol: Sonning natural bo'luvchilar soni va natural bo'luvchilar yigindisini hisoblash dasturi tuzilsin.*

Dastur algoritmi berilgan sonni o'zini yarmigacha natural sonlarga bo'lib chiqiladi qaysi songa qoldiqsiz bo'linsa usha son olinadi va oxirida sonning o'zi ham hisobga olinadi.

```

#include <iostream.h>
#include <math.h>
int main()
{   int n,p,s=0,k=0; double t;
    cout<<"n=";
    cin>>n;
    t=n/2; p=floor(t);    // sonni yarmini aniqlab olish
    for(int i=p;i>=1;i--)
        if(n%i==0)
            { s=s+i;    //bo'luvchilar yigindisi
              k=k+1;    //b o'luvchilar
            }
    s=s+n; k=k+1;    // sonni uzini etiborga olish
    cout<<" soni="<<k<<"ta yigindisi="<<s;
return 0;
}

```

Takrorlanish soni aniq bo'lganda yuqoridagi ikki tur bo'yicha C++ dasturlash tilida tasvirlashimiz mumkin. Takrorlanish jarayonini takrorlanish soni aniq bo'lmagan va shartsiz holatlarda ham dasturlash imkoniyati mavjud.

for sikl buyrug'i sikl parametrining manfiy yoki kasr qiymatlarida ham bajariladi. Masalan, yuqoridagi dasturda

```
float i, s; s=0;
```

```
for (i = 0; i >= -10; i = i - 0.5) s+=i;
```

kabi o'zgarish qilinsa, natija  $s = -105$  kabi bo'ladi.

for sikl buyrug'idagi  $i = 1; i \leq n; i ++$ ; kabi ifodalar ikki yoki undan ortiq ifodalardan tashkil topishi mumkin. Bu holda bir-birlaridan vergul (,) bilan ajratiladi. Masalan, yuqoridagi dasturda:

```
for (i = 1, j = 1; i <= 10, j <= 10; i ++, j ++) s+= i+j;
```

kabi o'zgarish qilinsa, natija  $s=110$  kabi bo'ladi.

Sikldagi  $i=1$  ifoda boshida faqat bir marta bajariladi. Sikl ichida o'zgaruvchi e'loni ham bo'lishi mumkin. Ammo bunday o'zgaruvchilardan sikl tanasidan tashqarida foydalanib bo'lmaydi.

Masalan, `for (int i=1; i<=n; i:=i+3)` kabi holatga ruxsat etiladi.

for sikl buyrug'ining parametrlar qismi bo'sh bo'lishi ham mumkin, lekin ikkita (;) bo'lishi shart.

Masalan, `for(; ;) s = s + i;` buyrug'i bajariladi, ammo cheksiz takrorlanish ro'y beradi.

Eslatma: Cheksiz takrorlanishni bekor qilish uchun `выполнить → сброспрограммы` algoritmi bo'yicha harakatlaning yoki `Alt+F2` tugmachalar kombinatsiyasidan foydalaning.

Shuningdek, tanasiz sikldan ham foydalanish mumkin.

**Masalan,**

```
for (i = 1; i <= 10; s += ++ i);
```

```
cout << "s =" <<s<< "\n";
```

**Misol.**  $y = 5x - 4a$ ,  $x \in [-3; 3]$ ,  $\Delta x = 2$ ; funksiya qiymatini hisoblash dasturini tuzing:



```

#include <iostream.h>
using namespace std;
int main( )
{ float x, y, a, z; cin >> a;
for (x = -3; x <= 3; x = x + 2) { y=5*x-4*a;cout << “y(“<<x<<”)=” << y <<
`n`;}
system(“pause”);
return 0;
}

```

Agar a ga 5 qiymat kiritilsa, natija quyidagi ko‘rinishda chop etiladi:

y(-3)=-35

y(-1)=-25

y(1)=-15

y(3)=-5

### Shartsiz o‘tish(Goto) operatori

Algoritm bajarilish vaqtida ma’lum bir buyruqlar sakrab o‘tish yoki ba’zi buyruqlar bir necha marta takrorlanish holatlari ham mavjud. Bunday holatlarda shartsiz o‘tish operatori yordamida dasturlashimiz mumkin.

*Tarif: Algoritm bajarilish davrida uning qandaydir qismi tashlanib yoki sakrab o‘tish jarayoniga **shartsiz o‘tish jarayoni** deyiladi.*

Shartsiz o‘tish jarayonini takrorlanuvchi va tarmoqlanuvchi jarayonlarda ham ishlatish mumkin.

Shartsiz o‘tish operatorining umumiy ko‘rinishi quyidagicha:

**Goto <belgi>; <belgi>:**

Goto operatorining ishlash jarayoni **goto** kalit so‘zidan keyingi belgini dastur tanasidan izlaydi va topilgan **<belgi>**: dan keying operatorlar bajariladi.

Bunda **goto** kalit so‘zining ma’nosi **“ga o‘tish”** demakdir. Belgi – ixtiyoriy identifikator. Belgi o‘rnida son va raqamlardan foydalanib bo‘lmaydi.

C++ tili goto operatorini uchratgach, undan keyingi belgini dasturning qaysi satrida ko‘rsatilib, undan keyin (:) qo‘yilgan bo‘lsa, dastur bajarilishini o‘sha satrdan davom ettiradi. Goto operatori yordamida dasturning istalgan satridan oldingi yoki keyingi satrlardan biriga sakrab o‘tib, dastur ijrosini davom ettirish mumkin. Bu barcha dasturlash tillarida bo‘lgan kuchli imkoniyat bo‘lishi bilan birgalikda quyidagicha muammolarni ham keltirib chiqaradi:

- dastur tarkibida goto operatoridan ko‘p foydalanilganda dasturni yoki dastur ijrosini tushunish qiyinlashadi. Shuningdek, tuzilgan dasturning qanday

masalani hal qilishini ma'lum vaqt o'tgach dastur muallifining o'zi ham aniqlashi uchun ma'lum vaqt talab etiladi;

- tarmoqlanish, takrorlanish buyruqlarining ichidan tashqariga sakrab chiqish va aksincha tashqaridan buyruq ichiga kirish kutilmagan xatoliklarga olib kelishi mumkin. Shuning uchun dasturchilar bu operatorlardan mumkin qadar kam foydalanishga harakat qilishadi.

**Misol:** Berilgan son juft yoki toqligini aniqlang.

```
#include <iostream.h>
int main()
{
    int n;
    cout<<"n=";
    cin>>n;
    if(n%2==0) { cout<<"juft"; goto a;}
    if(n%2==1) cout<<"toq";
    a:
    return 0;
}
```

Takrorlanuvchi jarayonlarni shartsiz o'tish operatori yordamida quyidagicha C++ dasturlash tilida tasvirlanadi.

**Misol:**  $N!$  ni hisoblash dasturi tuzilsin.

```
#include <iostream.h>
int main()
{
    int n, p=1, i=0;
    cout<<"n=";
    cin>>n;
    a1: i++;
    if(i<=n) {p=p*i; goto a1;}
    else goto a2;
    a2: cout<<p;
    return 0;
}
```

## Nazariy savollar

1 Takrorlanuvchi jarayonlar?

2 Takrorlanuvchi jarayonlarni blok sxema ko'rinishida tasvirlanishi?

- 3 Takrorlanuvchi jarayonlarni dasturlash tilida tasvirlanish usullari?
- 4 For sikl operatori va uning vazifasi?
- 5 For sikl operatorining turlari?
- 6 For sikl operatorining umumiy ko‘rinishi?
- 7 Shartsiz o‘tish jarayoni?
- 8 Shartsiz o‘tish operatori vazifasi va uning umumiy ko‘rinishi?
9. Qanday masalalarning dasturlarini tuzishda sikl buyruqlaridan foydalaniladi?
10. C++ dasturlash tilida sikl buyruqlarining necha xil ko‘rinishidan foydalaniladi? Ularni sanab ko‘rsating.
11. Parametrli sikl buyrug‘ining umumiy ko‘rinishini yozing va bajarilish tartibini tushuntiring.
12. 1 dan  $n$  gacha bo‘lgan 4 ga karrali sonlar yig‘indisini toppish dasturini parametrli sikl buyrug‘idan foydalanib tuzing.
13. Parametrli sikl buyrug‘ida sikl parametrining qadami qayerda ko‘rsatiladi?
14. Parametrli sikl buyrug‘ini bajarish tugaganda sikl parametrining qadami nimaga teng bo‘ladi?
15. 9 dan 41 gacha bolgan sonlar ko‘paytmasini topish dasturini do-while sikl buyrug‘idan foydalanib tuzing.

### **Mustaqil ishlash uchun topshiriqlar**

1.  $K$  va  $N$  ( $N > 0$ ) butun sonlari berilgan.  $K$  sonini  $N$  marta chiqaruvchi dastur tuzilsin.
2.  $A$  va  $B$  butun sonlari berilgan ( $A < B$ ).  $A$  va  $B$  sonlar orasidagi barcha butun sonlarni ( $A$  va  $B$  ni ham) chiqaruvchi va chiqarilgan sonlar sonini chiqaruvchi dastur tuzilsin. ( $A$  va  $B$  ham chiqarilsin).
3.  $A$  va  $B$  butun sonlari berilgan ( $A < B$ ).  $A$  va  $B$  sonlar orasidagi barcha butun sonlarni ( $A$  va  $B$  kirmaydi) kamayish tartibida chiqaruvchi va chiqarilgan sonlar sonini chiqaruvchi dastur tuzilsin.
4. Bir kg konfet narxi – haqiqiy son berilgan. 1, 2, ..., 10 kg konfet narxini chiqaruvchi dastur tuzilsin.
5. Bir kg konfet narxi – haqiqiy son berilgan. 0.1, 0.2, ..., 0.9, 1 kg konfet narxini chiqaruvchi dastur tuzilsin.
6. Bir kg konfet narxi – haqiqiy son berilgan. 1.2, 1.4, ..., 2 kg konfet narxini chiqaruvchi dastur tuzilsin.
7.  $A$  va  $B$  butun sonlari berilgan ( $A < B$ ).  $A$  dan  $B$  gacha bo‘lgan barcha butun sonlar ( $A$  va  $B$  ham kiradi) yig‘indisini chiqaruvchi dastur tuzilsin.
8.  $A$  va  $B$  butun sonlari berilgan ( $A < B$ ).  $A$  dan  $B$  gacha bo‘lgan barcha butun sonlar ( $A$  va  $B$  ham kiradi) ko‘paytmasini chiqaruvchi dastur tuzilsin.

**9.** *A va B butun sonlari berilgan ( $A < B$ ). A dan B gacha bo'lgan barcha butun sonlar (A va B ham kiradi) kvadratlarining yig'indisini chiqaruvchi dastur tuzilsin.*

**10.** *N ( $N > 0$ ) butun son berilgan. Quyidagi yig'indini hisoblovchi dastur tuzilsin:*

$$S = 1 + 1/2 + 1/3 + \dots + 1/N.$$

**11.** *N ( $N > 0$ ) butun son berilgan. Quyidagi yig'indini hisoblovchi dastur tuzilsin:*

$$S = N^2 + (N+1)^2 + (N+2)^2 + \dots + (2N)^2.$$

**12.** *N ( $N > 0$ ) butun son berilgan. Quyidagi ko'paytmani hisoblovchi dastur tuzilsin:*

$$P = 1.1 * 1.2 * 1.3 * \dots (N \text{ ta ko'paytuvchi}).$$

**13.** *N ( $N > 0$ ) butun son berilgan. Quyidagi ifodani hisoblovchi dastur tuzilsin:*

$$S = 1.1 - 1.2 + 1.3 - \dots$$

*(N ta qo'shiluvchi ishoralar almashib keladi. Shartli operatoridan foydalanmang).*

**14.** *N ( $N > 0$ ) butun son berilgan. Shu sonning kvadratini quyidagi mula asosida hisoblovchi dastur tuzilsin:*

$$N^2 = 1 + 3 + 5 + \dots + (2N-1).$$

*har bir qo'shiluvchidan keyin natijani ekranga chiqarib boring. Natijada ekranda 1 dan N gacha bo'lgan sonlarning kvadratlari chiqariladi.*

**15.** *A xaqiqiy va N ( $N > 0$ ) butun sonlari berilgan. A ning N- darajasini aniqlovchi dastur tuzilsin:  $A^N = A * A * \dots * A$*

**16.** *A xaqiqiy va N ( $N > 0$ ) butun sonlari berilgan. Bitta ssikldan foydalanib A ning 1 dan N gacha bo'lgan barcha darajasini chiqaruvchi dastur tuzilsin.*

**17.** *A xaqiqiy va N ( $N > 0$ ) butun sonlari berilgan. Bitta ssikldan foydalanib quyidagi A ning 1 dan N gacha bo'lgan barcha darajalarini chiqaruvchi va yig'indini hisoblash dasturi tuzilsin:*

$$S = 1 + A + A^2 + \dots + A^N.$$

**18.** *X xaqiqiy va N ( $N > 0$ ) butun sonlari berilgan. Bitta ssikldan foydalanib quyidagi X ning 1 dan N gacha bo'lgan barcha darajalarini chiqaruvchi va yig'indini hisoblovchi dastur tuzilsin:*

$$S = 1 - X + X^2 - X^3 + \dots + (-1)^N X^N.$$

*Shartli operatoridan foydalanmang.*

**19.** *N ( $N > 0$ ) butun soni berilgan. Birdan N gacha bo'lgan natural sonlari ko'paytmasini chiqaruvchi dastur tuzilsin:  $N! = 1 * 2 * \dots * N$ .*

*1 dan N gacha bo'lgan natural sonlari ko'paytmasi N faktorial deyiladi.*

**20.** *N ( $N > 0$ ) butun sonlari berilgan. Bitta ssikldan foydalanib quyidagi yig'indini hisoblash dasturi tuzilsin:*

$$S = 1! + 2! + 3! + \dots + N!$$

(  $N!$  ifoda -  $N$  faktorial - 1 dan  $N$  gacha bo'lgan butun sonlari ko'paytmasini bildiradi:  $N!=1*2*... *N$ ).

**21.**  $N$  ( $N>0$ ) butun sonlari berilgan. Bitta ssikldan foydalanib quyidagi yig'indini hisoblash dasturi tuzilsin:

$$S=1+1/1!+ 1/2!+ 1/3!+ ... +1/N!$$

(  $N!$  ifoda -  $N$  faktorial - 1 dan  $N$  gacha bo'lgan butun sonlari ko'paytmasini bildiradi:  $N!=1*2*... *N$ ). Olingan natija  $e=\exp(1)$  o'zgarmasning taqribiy qiymatiga teng bo'ladi.

**22.**  $X$  xaqiqiy va  $N$  ( $N>0$ ) butun sonlari berilgan. Quyidagi ifoda qiymatini hisoblash dasturi tuzilsin:

$$1+X/1!+ X^2/2!+ X^3/3!+ ... + X^N/N! \quad (N!=1*2*... *N).$$

Olingan natija  $e^X=\exp(X)$  ga taqribiy qiymatiga teng bo'ladi.

**23.**  $X$  xaqiqiy va  $N$  ( $N>0$ ) butun sonlari berilgan. Quyidagi ifoda qiymatini hisoblash dasturi tuzilsin:

$$X- X^3/3!+ X^5/5!- ... +(-1)^N X^{2N+1}/((2N+1)!)$$

( $N!=1*2*... *N$ ). Olingan natija  $\sin(X)$  ga taqribiy qiymatiga teng bo'ladi.

**24.**  $X$  xaqiqiy va  $N$  ( $N>0$ ) butun sonlari berilgan. Quyidagi ifoda qiymatini hisoblash dasturi tuzilsin:

$$1- X^2/2!+ X^4/4!- ... +(-1)^N X^{2N}/(2N)!$$

( $N!=1*2*... *N$ ). Olingan natija  $\cos(X)$  ga taqribiy qiymatiga teng bo'ladi.

**25.**  $X$  ( $|X|<1$ ) xaqiqiy va  $N$  ( $N>0$ ) butun sonlari berilgan. Quyidagi ifoda qiymatini hisoblash dasturi tuzilsin:

$$X- X^2/2+ X^3/3- ... +(-1)^{N-1} X^N/N.$$

Olingan natija  $\ln(1+X)$  ga taqribiy qiymatiga teng bo'ladi.

**26.**  $N$  butun soni va sonlar o'qida 2 ta  $A, B$  nuqta berilgan. ( $A$  va  $B$  haqiqiy son).  $[A,B]$  kesmani teng  $N$  ta kesmaga bo'ling.  $[A,B]$  kesmada ajratilgan barcha nuqtalar uchun  $F(X) = 1 - \sin(X)$  funksiya qiymatini hisoblang.

**27.**  $X$  ( $|X|<1$ ) xaqiqiy va  $N$  ( $N>0$ ) butun sonlari berilgan. Quyidagi ifoda qiymatini hisoblash dasturi tuzilsin:

$$X+1*X^3/(2*3)+1*3*X^5/(2*4*5)+...+1*3*... *(2N-1)X^{2n+1}/(2*4*... *(2N)*(2N+1)).$$

**28.**  $N$  butun soni va sonlar o'qida 2 ta  $A, B$  nuqta berilgan. ( $A$  va  $B$  haqiqiy son).  $[A,B]$  kesmani teng  $N$  ta kesmaga bo'ling.  $[A,B]$  kesmada ajratilgan barcha nuqtalarni chiqaring.

### 3.6 C++ DASTURLASH TILIDA SHARTLI TAKRORLANUVCHI JARAYONLAR DASTURLASH

*Reja:*

1. *while operatori va uning umumiy ko‘rinishi;*
2. *do while operatori va uning umumiy ko‘rinishi.*
3. *Ichma-ich joylashgan sikllar*

Bazi bir masalalarni yechish algoritmlari tarkibida takrorlanishlar qandaydir shartlarga asosan bajariladi. Har bir takrorlanish jarayoni bajarilish qadamida shart tekshirilib o‘tib borilaveradi, qachonki shart yolg‘on bo‘lgandagina takrorlanish jarayoni to‘xtatiladi. Masalan yig‘indisi  $n$  ga teng bo‘lgan natural sonlar sonini topish yoki umumiy hadi  $n$  dan kichik bo‘lgan cheksiz kamayuvchi geometrik progressiyani hadini topish kabi masalalarda shartli takrorlanuvchi operatorlardan foydalaniladi. Agar algoritm tarkibidagi bir necha marta takrorlanishi kerak bo‘lgan buyruqlarni takrorlanuvchi jarayonlar asosida dasturlash tillarida tasvirlanmasa, bu buyruqlarni barchasini bajarish murakkablashadi.

Shartli takrorlanuvchi algoritmlarni shartli takrorlanuvchi jarayonlar ham deb ataymiz.

*Tarif: Agar takrorlanishlar soni ma’lum bir shartlar asosida aniqlansa, bunday jarayonlar shartli takrorlanuvchi jarayonlar deyiladi.*

Shartsiz o‘tish operatori va tarmoqlanuvchi operatorlar yordamida ham shartli takrorlanuvchi jarayonlarni dasturlash imkoniyati mavjud. Lekin bunday holatlarda bitta amalni bajarish uchun bir nechta operatorlarni ishlatish kerak bo‘ladi. Shartli takrorlanuvchi operatorlar bajarilish holatlariga qarab turlarga ajratiladi. Shartli takrorlanuvchi operatorlar quyidagi turlari mavjud:

- Shart oldi takrorlanuvchi operatorlar;
- Shart keyin takrorlanuvchi operatorlar.

Yuqorida keltirilgan operatorlar takrorlanish holatiga qarab bajariladi. Shart oldi va shart keyin takrorlanuvchi algoritmlar C++ dasturlash tilida quyidagi operatorlar yordamida dasturlanadi.

#### **While operatori**

Shart oldi takrorlanuvchi jarayonlar bajarilish holati har bir takrorlanish oldidan shart tekshirilib keyin takrorlanish tanasidagi operatorlar bajariladi. Agar takrorlanish holati boshidan shart yolg‘on qiymat qabul qilsa, takrorlanish bir marta ham bajarilmaydi.

Shart oldi takrorlanuvchi operatorlarning C++ dasturlash tilida ifodalash uchun while operatori yordamida tasvirlanadi.

*Ta'rif: Agar shartli takrorlanuvchi jarayonlar tarkibidagi shart takrorlanishdan oldin tekshirilsa, shart oldi takrorlanuvchi jarayonlar deyiladi.*

Takrorlanuvchi operator tarkibiga beriladigan shart tahlil qilinib yozilish kerak, chunki shart hech qachon yolg'on qiymat qabul qilmasa, dastur cheksiz ishlashga to'g'ri keladi. Takrorlanish hech qachon cheksiz bo'lishi mumkin emas, aks holda algoritmnining diskretlik hossasi buziladi.

Shart oldi takrorlanish operatori yani while operatorining umumiy ko'rinishi quyidagicha.

**while(<shart>) { operatorlar; }**

Agar shart chin qiymat qabul qilib tursa {} ichidagi operatorlar bajarilaveradi, qachonki shart yolg'on bo'lgandagina takrorlanish o'z ish faoliyatini to'xtatadi.

Takrorlash strukturasi bir ifoda yoki blokni ma'lum bir shart to'g'ri (true) bo'lishi davomida qaytarish imkonini beradi. Qaytarilayotgan ifoda shartga ta'sir ko'rsatishi kerak. Ma'lum bir vaqt o'tgandan keyin shart false ga o'zgartilishi kerak. Bo'lmasa while (davomida) tugatilmaydi. while faqat o'zidan keyin kelgan ifodaga ta'sir qiladi. Agar biz bir guruh amallarni qaytarmoqchi bo'lsak, ushbu blokni {} qavslar ichiga olishimiz kerak. Shart takrorlanuvchi blokning boshida tekshirilgani sababli, agar shart noto'g'ri bo'lib chiqsa, blokni hech ijro ko'rmasligi ham mumkin.

While operatori tarkibidagi shart yolg'on qiymat qabul qilganda operatorlar bajarilmasdan qoladi, shart chin qiymat qabul qilgandagina operatorlar bajariladi. Ba'zi hollarda shart takrorlanish boshidan yolg'on qiymat qabul qiladi, bunda takrorlanish bir marta ham bajarilmaydi. Shart chin qiymat qabul qilib, lekin takrorlanish tanasida shart tarkibi o'zgartirilmasa, takrorlanish cheksiz bo'lib qoladi.

**Misol:** *Salom so'zini ekranga n mart chiqarilsin.*

Bu masalani for sikl operatori yordamida ham ifodalash mumkin, lekin while operatorining mohiyatini o'rganish uchun oddiy masala yordamida qaraymiz.

```
#include <iostream.h>
int main()
{
    int n,i=1;
    cout<<"n=";
    cin>>n;
    while(i<=n)
```

<pre> { cout&lt;&lt;"Salom \n";   i++; } return 0; } </pre>	
Kirituvchi ma'lumotlar n=3	Dastur natijasi <i>Salom</i> <i>Salom</i> <i>Salom</i>

While operatorini dasturlash tarkibida ishlatish vaqtida doimo takrorlanish tarkibidagi shart bilan tekshiriladigan bitta o'zgaruvchi olish kerak. Bu masalada shart bilan tekshirish uchun  $i$  o'zgaruvchisi tanlandi. Takrorlanish sonini  $n$  orqali  $i$  bilan solishtirish natijasida aniqlanadi.

While operatorining ishlash jarayoni yuqoridagi misolda quyidagicha.

*Boshlang'ich holatda  $i=1$  shart  $i \leq n (1 \leq 3)$  chin*

*Qadam1: salom so'zi ekranda chiqariladi  $i=2$ ;  
shart  $i \leq n (2 \leq 3)$*

*Qadam2: salom so'zi ekranda chiqariladi  $i=3$ ;  
shart  $i \leq n (3 \leq 3)$*

*Qadam2: salom so'zi ekranda chiqariladi  $i=4$ ;  
shart  $i \leq n (4 \leq 3)$  yolg'on takrorlanish to'xtatiladi.*

**Misol:**  $n$  berilganda  $k! < n$  shartni qanoatlantiruvchi eng katta  $k$  ni aniqlang.

Bu masalani C++ dasturlash tilida ifodalash uchun  $k$  faktorialni takrorlanish tanasiga joylashtirish kerak, takrorlanish shartini esa  $k! < n$  ko'rinishida tasvirlanadi.

```

#include <iostream.h>
int main()
{
  int n,k=0,p=1;
  cout<<"n=";
  cin>>n;
  while(p<n)
  {
    k++;
    p=p*k;
  }
  cout<<k-1;
  return 0;
}

```



}	
Kirituvchi ma'lumotlar n=7	Dastur natijasi k=3

Bu masalani C++ dasturlash tilidagi ko'rinishiga e'tibor bersak oxirida `cout<<k-1;` operatori yozilgan, buni mohiyati shart chin qiymatida bitta qadam ortiq bajariladi, shuning uchun `k-1` holat bo'yicha chiqariladi. Bu jarayonni `n= 7` qiymat berib, qo'lda test qilib ko'rsangiz tushunish oson bo'ladi.

While operatori tarkibidagi shart ba'zi hollarda o'zgarmas qiymat ko'rinishda ham beriladi, bunda shart 0 bilan solishtiriladi, agar qiymat ortib borsa, dastur cheksiz takrorlanish mumkin, bunday holatlarda o'zgarmas qiymatni kamaytirish kerak.

```
#include <iostream.h>
int main()
{   int k=0,p=1;
    while(p)
    { k++;
      p=p*k;
      cout<<k-1;
    }
    return 0;
}
```

Yuqoridagi dastur tarkibidagi shart `p` faqat 0 bilan solishtiriladi 0 dan farqli bo'lsa, takrorlanish bajarilaveradi, demak takrorlanish cheksiz bo'lib qoladi. Takrorlanish cheksiz bo'lmasligi holatini quyidagi misol orqali tekshiramiz.

**Misol:** *“Misol” so'zini ekranga n marta chiqaring.*

```
#include <iostream.h>
int main()
{   int n,i=1;
    cout<<"n=";
    cin>>n;
    while(n)
    { cout<<"Misol \n";
      n--;
    }
    return 0;
}
```

}	
Kirituvchi ma'lumotlar n=3	Dastur natijasi <i>Misol</i> <i>Misol</i> <i>Misol</i>

**while** so'zidan keyin qavs ichida qiymat berish buyruqlari ham bo'lishi mumkin va bu buyruqlar mantiqiy ifodadan vergul (,) belgisi bilan ajratiladi hamda ushbu qiymat berish buyug'i bajarilib, keyin mantiqiy ifodaning qiymati tekshiriladi. Qavs ichidagi oxirgi ifoda mantiqiy ifoda bo'lishi kerak.

Sikl buyrug'i quyidagi ko'rinishda bo'lsa ham ishlaydi. While (s+= i++, i<=10);

Qiymat berish buyrug'i ikki yoki undan ortiq bo'lishi ham mumkin.

Masalan, quyidagicha o'zgarish qilinsa juft sonlar yig'indisi hisoblanadi.

while (i+=1, s+=i++, i<=10);

Agar sikl buyrug'ida buyruqlar yoki buyruqlar bloki bo'lmasa ham (;) bo'lishi shart, aks holda keyingi buyruqni sikl tanasi deb tushunadi. Masalan, oxirgi buyruqda (;)bo'lmasa, ya'ni while (i+=1; s+=i++, i<=10) kabi bo'lsa dastur natijasi quyidagicha bo'ladi:

s = 2

s = 6

s = 12

s = 10

Demak, while sikl buyrug'ida qavsdan keyin (;) qo'yish buyruqlar blokining bo'shligini bildiradi.

**Misol.** Tanlangan natural sonning raqamlari ko'paytmasini topish dasturini tuzing.

```
#include <iostream.h>
using namespace std;
int main()
{ int a, a1, p; cout << "a sonini kiriting " << '\n';
cin >>a;
p=1; while(a1=a%10, a=a/10, p=p*a1, a);
cout << "sonning raqamlari ko'paytmasining " <<p<< '\n';
system ("pause");
return 0;
}
```

Ushbu misolda takrorlanish buyrug'ini quyidagicha ham tashkil qilish mumkin edi.

```
p=1; while (a) {a1 = a%10, a=a/10; p=p*a1};
```

bunda takrorlanish buyrug'ining bajarilish tartibi bo'yicha buyruqlar har bajarilganda tanlangan sonning o'ng tomonidan bitta raqami qirqib olinib, p ga ko'paytirilib borilaveradi, toki a=0 bo'lguncha.

Demak while tarkibidagi shart son qiymat bilan berilsa, bunda kamayish tartibida berilish kerak va berilgan son 0 ga tenglashganda takrorlanish to'xtatiladi.

### **do-while takrorlanish operatori**

shartli takrorlanuvchi jarayonlar tarkibida takrorlanish oxirida shart tekshirilish holatlarida shart keyin takrorlanuvchi algoritmlar qo'llaniladi. Shart keyin takrorlanuvchi jarayonlar bajarilish holati har bir takrorlanish bir qadam bajarilgandan keyin shart tekshiriladi. Agar takrorlanish holati boshidan shart yolg'on qiymat qabul qilsa takrorlanish bir marta bajarilib qoladi.

*Ta'rif: Agar shartli takrorlanuvchi jarayonlar tarkibidagi shart takrorlanish oxirida tekshirilsa, bunday jarayonlar shart keyin takrorlanuvchi jarayonlar deyiladi.*

Shart keyin takrorlanuvchi jarayonlarni takrorlanish oldindan shart tekshirilish iloji bo'lmagan vaqtlarda foydalanish mumkin. Shart keyin takrorlanuvchi jarayonlarda takrorlanish tanasi tarkibidagi operatorlar bir marta bajariladi va keyin shart tekshiriladi. Agar shart yolg'on bo'lsa, takrorlanish to'xtatiladi.

Shart oldi takrorlanuvchi operatorlarning C++ dasturlash tilida ifodalash uchun do while operatori yordamida tasvirlanadi.

```
do { operatorlar; } while(<shart>);
```

do kalit so'zidan keying operatorlar takrorlanish tanasi hisoblanadi, while kalit so'zidan keying shart tekshiriladi. Agar chin bo'lsa, yana do kalit so'zidan keying operatorlar bajariladi, aks holda takrorlanish tugatiladi.

Shart keyin takrorlanuvchi jarayonlar, takrorlanuvchi jarayonlar ichida eng murakkabi hisoblanadi, chunki shart tekshirilmasdan takrorlanish boshlanadi.

Takrorlanish oldindan shart tekshirilish iloji bo'lmagan holatlar uchun qo'llanilishi ta'kidlandi. Bunday holatlarni quyidagi masala orqali ko'rib o'tamiz.

*Misol: yig'indisi n ga teng bo'lgan natural sonlarning eng kattasini anilang.*

Bu masalada albatta do-while operatoridan foydalanishimiz mumkin, chunki hech bo'lmaganda bitta natural sonni yigindisini hisoblash kerak bo'ladi.

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

```

int main()
{
    int n,i=0;
    cout<<"n=";
    cin>>n;
    do {
        s=s+i;
        i++;

        } while(s<n);
    cout<<i-1;
    return 0;
}

```

Kirituvchi ma'lumotlar  
n=3

Dastur natijasi  
2

Yuqoridagi masalada oldin yig'indi hisoblanib keyin shart tekshiriladi.

**Misol:** 1- hadi  $b$  va maxraji  $q$  berilganda yig'indisi  $n$  dan kichik bo'lgan geometrik progressiyani eng katta hadini toping.

```

#include <iostream.h>
int main()
{
    int b,q,s=0,n;
    cin>>n>>b>>q;
    do {
        s=s+b;        // yig'indini hisoblash
        b=b*q;        // umumiy hadini hisoblash

        } while(s<n);
    cout<<b/q;
    return 0;
}

```

do/while ifodasi while strukturasi o'xshashdir. Bitta farqi shundaki while da shart boshiga tekshiriladi. do/while da esa takrorlanish tanasi eng kamida bir marta ijro ko'radi va shart strukturaning so'ngida test qilinadi. Shart true bo'lsa blok yana takrorlanadi. Shart false bo'lsa do/while ifodasidan chiqiladi. Agar do/while ichida qaytarilishi kerak bo'lgan ifoda bir dona bo'lsa {} qavslarning keragi yo'qdir.

Shart asosida takrorlanuvchi operatorlar yuqoridagi ikki holatlar shart oldin yoki shart keyin takrorlanishlar bo'yicha C++ dasturlash tilida tasvirlanadi.

### **Ichma-ich joylashgan sikllar**

Barcha dasturlash tillarida ichma-ich joylashgan sikl buyruqlaridan foydalanish imkoniyati yechilishi mumkin bo'lgan masalalar ko'lamini sezilarli darajada kengaytiradi.

for operatori yordamida tashkil qilinadigan ichma-ich joylashgan sikl buyruqlarining umumiy ko'rinishi quyidagicha:

**for (i = imin; i <= imax; i = i + i qad)**

**for (j = jmin; j <= jmax; j = j + j qad)**

**for (k = kmin; k <= kmax; k = k + k qad)** buyruqlar;

Uchta ichma-ich joylashgan holni tanladik. Aslida ularning soniga chegara qo'yilmagan. Buyruqning bajarilish tartibi quyidagicha:

Tashqi sikl parametri o'zining boshlang'ich **imin** qiymatini qabul qiladi va oxirgi **imax** qiymati bilan solishtiradi. Agar **i<=imax** mantiqiy ifodaning qiymati chin bo'lsa, o'rta sikl parametri o'zining boshlang'ich **jmin** qiymatini oladi va oxirgijmax qiymati bilan solishtiradi. Agar **j<=jmax** mantiqiy ifodaning qiymati chin bo'lsa, ichki sikl parametri o'zining boshlang'ich **kmin** qiymatini oladi va oxirgikmax qiymati bilan solishtiradi. Agar **k<=kmax** mantiqiy ifodaning qiymati rost bo'lsa, buyruqlar bir marta bajarilib, ichki sikl parametri bir qadamga ortadi (**k=k+kqad**) va yana **k<=kmax** mantiqiy ifodaning qiymati tekshiriladi. Agar u rost bo'lsa, buyruq yana bir marta bajariladi va hokazo bu jarayon **k>kmax** bo'lguncha davom etadi. So'ngra o'rta sikl parametri bir qadamga ortadi (**j=j+jqad**) va **j<=jmax** shart tekshiriladi. Agar u rost bo'lsa, ichki sikldagi takrorlanishlar o'rta sikl parametrining yangi qiymati uchun to'liq qaytadan amalga oshiriladi va hokazo bu jarayon **j>jmax** bo'lguncha davom etadi. Shundan so'ng tashqi sikl parametri bir qadamga ortadi (**i=i+iqad**) va **i<=imax** mantiqiy ifodaning qiymati tekshiriladi. Agar u rost bo'lsa, tashqi sikl parametrining yangi qiymati uchun o'rta va ichki sikllardagi takrorlanishlar to'liq qaytadan amalga oshiriladi va hokazo bu jarayon **i>imax** bo'lguncha davom etadi.

Demak, ichma-ich joylashgan uchta siklning bajarilish tartibi soatning sekund, minut va soat millari harakatiga juda o'xshaydi. Haqiqatdan, minut mili 1 minutga siljiguncha sekund mili 60 ta harakat sodir qiladi, soat mili bir soatga siljishi uchun esa minut mili 60 minutga siljiydi. Ya'ni tashqi sikl parametri soat miliga, o'rta sikl parametri minut miliga, ichki sikl parametri esa sekund miliga mos keladi. Demak, tashqi siklda 5 marta, o'rta siklda 8 marta, ichki siklda 9 marta takrorlanishi sodir bo'lsa, buyruqlar jami  $5 * 8 * 9 = 360$  marta bajariladi.

Ichma-ich joylashgan sikllarning umumiy ko‘rinishida blok belgilari ({ }) yo‘q. Agar bir for operatori bilan ikkinchisi oralig‘ida kamida bitta buyruq bo‘lsa ham blok belgisi ichiga olinadi.

**Misol.** Natural sonning tub bo‘luvchilarini va ularning yig‘indisini topish dasturini tuzing.

```
#include <iostream.h>
using namespace std;
int main ()
{
int i, j, n, s, m; s=0; cin>>n;
for (i=2; i<=n; i++)
{ m=i/2; for (j=2; j<=m; j++)
if (i%j) continue; else goto tam;
if (!(n%i)) { cout <<n<< " sonning tub bo‘luvchisi=" <<i<< "\n"; s+=i; }
tam: continue;};
cout << "tub bo‘luvchilari yig‘indisi=" <<s<< "\n";
system ("pause");
return 0;
}
```

Masalan, n ning qiymati sifatida 45 soni kiritilsa, dastur javobi quyidagicha chop etiladi:

```
45 sonining tub bo‘luvchisi = 3
45 sonining tub bo‘luvchisi = 5
Tub bo‘luvchilari yig‘indisi = 8
```

Ichma-ich joylashgan sikllarning while va do-while shakllarini har xil ko‘rinishlarda ifodalash mumkin. Old shartli ichma-ich joylashgan sikllarning umumiy ko‘rinishi quyidagicha:

```
i = i min; while (i <= i max)
{ j = j min; while (j <= j max)
{k = k min; while (k <= k max)
{ buyruqlar; k = k + k qad;}
j = j + j qad}
i = i + i qad}
```

Buyruqning bajarilish tartibi for ko‘rinishi bilan bir xil.

**Misol.** Quyidagi ifodaning qiymatini hisoblash dasturini tuzing.

$$\sum_{i=0}^n \sum_{j=2}^m \sum_{k=0}^l (2i + 3j - 4k)$$

```

#include <iostream.h>
using namespace std;
int main( )
{ int i, j, k, n, m, l, s1, s2, s3;
cin >>n>>m>>l; s1=0;
i=1; while (i<=n) {s2=0;
j=2; while (j<=m) {s3=0;
k=0; while (k<=l)
{ s3+=2*i+3 * j-4 * k; k++;}
s2+=s3; j++;}
s1+= s2; i++;}
cout << "s1=" <<s1<< '\n'; system("pause");
return 0;
}

```

do-while ko‘rinishida ichma-ich joylashgan sikllarning umumiy ko‘rinishi quyidagicha:

```

i = i min;
do
{j = j min;
do
{k = k min;
do
{ buyruqlar;
k = k + k qad;}
while (k<=k max);
j = j + j qad;}
while (j <= j max);
i = i + i max;}
while (i <= i max)

```

Buyruqning bajarilish tartibini mustaqil o‘rganing.

**Misol.** Quyidagi ifodaning qiymatini hisoblash dasturini tuzing.

$$\prod_{i=1}^n \prod_{j=2}^m \prod_{k=1}^l (i + j + k)$$

```

#include <iostream.h>
using namespace std;
int main( )
{ int i, j, k, n, m, l, p1, p2, p3; cin >>n>>m>>l;
p1 = 1; i=1; do {
p2 = 1; j=2; do {
p3 = 1; k=1; do {
p3 *= i+j+k; k++} while (k<= l);
p2 *=p3; j++} while (j<=m);
p1 *= p2; i++} while (i<=n);
cout << " p1 =" <<p1<< "\n";system ("pause");
return 0;
}

```

**Misol.**  $y=3x-7a+9,3$ ;  $x \in [-3;2]$ ;  $\Delta x=0,5$ ;  $a \in [1;5]$ ;  $\Delta a=0,8$ ; funksiya qiymatini hisoblash dasturini tuzing.

```

#include <iostream.h>
#include <math.h>
using namespace std;
int main ( )
{
float x,y,a;
x=-3; do {
a=1; do { y=3*x-7*a+9.3;
cout <<"y ("<<x<<","<<a<<")="<<y<<"\n';
a=a+0.8;} while (a<=5); x=x+0.5;}
while (x<=2);system ("pause");
return 0;
}

```

### Nazariy savollar

- 1 Shart asosida takrorlanuvchi jarayonlar deb nimaga aytiladi?
- 2 Shart asosida takrorlanuvchi jarayonlarni turlari?
- 3 Shart oldin takrorlanuvchi jarayon deb nimaga aytiladi?
- 4 While operatorining umumiy ko'rinishi va vazifasi?
- 5 Shart keyin takrorlanuvchi jarayon deb nimaga aytiladi?
- 6 do while operatorining umumiy ko'rinishi va vazifasi?



*Oldshartli sikl buyrug‘ining umumiy ko‘rinishini yozing va bajarilish tartibini tushuntiring.*

*7. Sikl buyruqlarida qachon blok belgilaridan foydalaniladi?*

*8. So‘ngshartli sikl buyrug‘ining umumiy ko‘rinishini, blok-sxemasini yozing va bajarilish tartibini tushuntiring*

*9. Sikl buyrug‘ining do-while ko‘rinishida nima uchun blok belgilaridan foydalanilmaydi?*

*10. Nima uchun sikl buyruqlari old shartli va so‘ng shartli deb nomlanadi.*

### **Mustaqil ishlash uchun topshiriqlar**

*1. A va B butun musbat sonlari berilgan ( $A > B$ ). A uzunlikdagi kesmada maksimal darajada B kesma joylashtirilgan. A kesmaning bo‘sh qismini aniqlovchi programma tuzilsin. Ko‘paytirish va bo‘lish amallarini ishlatmang.*

*2. A va B butun musbat sonlari berilgan ( $A > B$ ). A uzunlikdagi kesmada B kesmadan nechta joylashtirish mumkinligini aniqlovchi programma tuzilsin. Ko‘paytirish va bo‘lish amallarini ishlatmang.*

*3. N va K butun musbat sonlari berilgan. Faqat ayirish va qo‘shish amallarini ishlatib N sonini K soniga bo‘lgandagi qoldiq va butun qismini aniqlovchi programma tuzilsin.*

*4. N butun soni berilgan ( $N > 0$ ). Agar n soni 3 ning darajasi bo‘lsa “3 – ning darajasi”, aks holda “3 – ning darajasi emas” degan natija chiqaruvchi programma tuzilsin. Qoldikli bo‘lish va bo‘lish amallarini ishlatmang.*

*5. 2 sonining qandaydir darajasini bildiruvchi n butun soni berilgan ( $n > 0$ ):  $N = 2^k$ . K ni aniqlovchi programma tuzilsin.*

*6. N natural soni berilgan ( $N > 0$ ). Quyidagi ifodani hisoblovchi programma tuzilsin.*

$$N!! = N * (N-2) * (N-4) * \dots$$

*Agar N juft son bo‘lsa oxirgi ko‘paytuvchi 2, toq bo‘lsa 1 bo‘ladi.*

*7. N natural soni berilgan ( $N > 0$ ). Kvadrati N dan katta bo‘ladigan eng kichik butun K sonini ( $K^2 > N$ ) aniqlovchi programma tuzilsin. Ildizdan chiqaruvchi funksiyadan foydalanmang.*

*8. N natural soni berilgan ( $N > 0$ ). Kvadrati N dan katta bo‘lmagan eng katta butun K sonini ( $K^2 \leq N$ ) aniqlovchi programma tuzilsin. Ildizdan chiqaruvchi funksiyadan foydalanmang.*

*9. N natural soni berilgan ( $N > 1$ ).  $3^k > N$  shartni qanoatlantiruvchi eng kichik butun K sonini aniqlovchi programma tuzilsin.*

*10. N natural soni berilgan ( $N > 1$ ).  $3^k \leq N$  shartni qanoatlantiruvchi eng katta butun K sonini aniqlovchi programma tuzilsin.*

**11.**  $N$  natural soni berilgan ( $N > 1$ ).  $(1 + 2 + \dots + K) \geq N$  shart bajariladigan eng kichik  $K$  sonini aniqlovchi programma tuzilsin. 1 dan  $K$  gacha bo'lgan yig'indi ham ekranga chiqarilsin.

**12.**  $N$  natural soni berilgan ( $N > 1$ ).  $(1 + 2 + 3 + \dots + K) \leq N$  shart bajariladigan eng katta  $K$  sonini aniqlovchi programma tuzilsin. 1 dan  $K$  gacha bo'lgan yig'indi ham ekranga chiqarilsin.

**13.**  $A$  soni berilgan ( $A > 1$ ).  $(1 + 1/2 + 1/3 + \dots + 1/K) \geq A$  shart bajariladigan eng kichik  $K$  sonini aniqlovchi programma tuzilsin.

**14.**  $A$  soni berilgan ( $A > 1$ ).  $(1 + 1/2 + 1/3 + \dots + 1/K) \leq A$  shart bajariladigan eng katta  $K$  sonini aniqlovchi programma tuzilsin.

**15.** Bankka boshlang'ich  $S$  so'm qo'yildi. Har oyda bor bo'lgan summa  $P$  foizga oshadi ( $0 < P < 25$ ). Necha oydan keyin boshlang'ich qiymat 2 martadan ko'p bo'lishini hisoblovchi programma tuzilsin. Necha oy  $K$  – butun son. Bankda hosil bo'lgan summa haqiqiy son ekranga chiqarilsin.

**16.** Sportsmen birinchi kuni 10 km yugurib boshladi. Keyingi kunlari bir oldingi kunga nisbatan  $P$  foiz ko'p yugurdi ( $0 < P < 50$ ). Sportsmenning necha kundan keyin jami yugurgan masofasi 200 km dan oshadi? Jami kunlar soni va masofani (butun son) chiqaruvchi programma tuzilsin.

**17.**  $N$  va  $M$  butun musbat sonlari berilgan ( $N > M$ ).  $N$  sonini  $M$  soniga bo'lib butun va qoldiq qismlarini bo'lish va qoldiqni olish amallarini ishlatmasdan topuvchi dastur tuzilsin.

**18.**  $N$  butun soni berilgan ( $N > 0$ ). Bo'lib butun va qoldiq qismlarini aniqlash orqali, berilgan son raqamlarini teskari tartibda chiqaruvchi dastur tuzilsin.

**19.**  $N$  butun soni berilgan ( $N > 0$ ). Bo'lib butun va qoldiq qismlarini aniqlash orqali, berilgan son raqamlarini yig'indisi va raqamlari sonini chiqaruvchi dastur tuzilsin.

**20.**  $N$  butun soni berilgan ( $N > 0$ ). Bo'lib butun va qoldiq qismlarini aniqlash orqali, berilgan son raqamlarining orasida 2 raqami bor – yo'qligini aniqlovchi dastur tuzilsin.

**21.**  $N$  butun soni berilgan ( $N > 0$ ). Bo'lib butun va qoldiq qismlarini aniqlash orqali, berilgan son raqamlarining orasida toq raqamlar bor – yo'qligini aniqlovchi dastur tuzilsin.

**22.**  $N$  butun soni berilgan ( $N > 0$ ).  $N$  sonini tub yoki tub emasligini aniqlovchi dastur tuzilsin.

**23.**  $A$  va  $B$  butun musbat sonlari berilgan. Berilgan sonlarning eng kata umumiy bo'luvchisini aniqlovchi dastur tuzilsin.

**24.**  $N$  butun soni berilgan ( $N > 1$ ).  $N$  sonini Fibonachchi sonlari orasida bor – yo‘qligini aniqlovchi dastur tuzilsin. (Fibonachchi sonlari while24 masalada berilgan).

**25.**  $N$  butun soni berilgan ( $N > 1$ ).  $N$  sonidan kata bo‘lgan birinchi Fibonachchi sonini aniqlovchi dastur tuzilsin. Fibonachchi sonlari quyidagi qonuniyatlar asosida topiladi.

$$F_1 = 1; F_2 = 1; F_K = F_{K-1} + F_{K-2}; K = 3, 4, \dots$$

**26.** Fibonachchi soni bo‘lgan  $N$  butun soni berilgan ( $N > 1$ ). (Fibonachchi sonlari while24 masalada berilgan).  $N$  sonidan bir oldingi va bir keying Fibonachchi sonlarini chiqaruvchi dastur tuzilsin.

**27.** Fibonachchi soni bo‘lgan  $N$  butun soni berilgan ( $N > 1$ ). (Fibonachchi sonlari while24 masalada berilgan).  $N$  soni Fibonachchi ketma – ketligining nechanchi xadi ekanini chiqaruvchi dastur tuzilsin.

**28.**  $e$  haqiqiy musbat soni berilgan. Ketma-ketlik xadlari quyidagicha aniqlanadi:

$$a_1 = 2; a_k = 2 + 1/a_{k-1}; k = 2, 3, \dots$$

$|a_k - a_{k-1}| < e$  shartni qanoatlantiruvchi eng kichik  $k$  sonini aniqlovchi dastur tuzilsin.  $a_k$  va  $a_{k-1}$  ham ekranga chiqarilsin.

**29.**  $e$  haqiqiy musbat soni berilgan. Ketma-ketlik xadlari quyidagicha aniqlanadi:

$$a_1 = 1; a_2 = 2; a_k = (a_{k-2} + 2 * a_{k-1}) / 3; k = 3, 4, \dots$$

$|a_k - a_{k-1}| < e$  shartni qanoatlantiruvchi eng kichik  $k$  sonini aniqlovchi dastur tuzilsin.  $a_k$  va  $a_{k-1}$  ham ekranga chiqarilsin.

**30.**  $A, B, C$  musbat butun sonlari berilgan.  $A \times B$  to‘rtburchak ichida tomoni  $C$  bo‘lgan kvadratdan nechtasi sig‘ishini aniqlovchi dastur tuzilsin. Ko‘paytirish va bo‘lish amallarini ishlatmang.

**31.** Berilgan sonning bo‘luvchilari sonini aniqlang.

**32.** Berilgan sonni tub yoki murakkabligini aniqlang.

**33.** O‘z‘idan oldingi natural sonlar yig‘indisidan 5 marta kichik bo‘lgan ikki xonali sonlarni aniqlang.

**34.** Shunday uch xonali sonlarni topingki, ularning dastlabki ikkita raqamidan tuzilgan qismi 3 raqamiga qoldiqsiz bo‘linsin.

**35.** Raqamlar yig‘indisi 27 ga teng bo‘lgan 4 xonali sonlarni toping.

**36.** Ketma-ket  $n$  marta uchburchak tomonlari kiritiladi. Qaysi urinishda kiritilgan tomonlar bo‘yicha uchburchak yasash mumkinligini aniqlang.

**37.**  $n$  xonali son kiritiladi. Unda nechta 0 va 1 raqamlari borligini aniqlang.

**38.**  $n$  butun sonning necha xonali ekanligini aniqlang.

## **IV-BOB. MURAKKAB TURLAR VA ULARNING C++ DASTURLASH TILIDA FOYDALANISH**

C++ dasturlash tilida oddiy turdan tashqari boshqa turdagi o'zgaruvchilar ham ishlatiladi. Ba'zi hollarda bir o'zgaruvchining bir nechta xil ko'rinishlari mavjud bo'ladi, matematika kabi ma'lum bir vazifani bajaruvchi o'zgaruvchi turga mansub bo'lgan funksiyalardan foydalanishga to'g'ri keladi. Kompyuter xotirasiga ma'lumotlarni oddiy bo'lmagan usullar yordamida joylashishiga ma'lumotlarning murakkab turlari yordam beradi. Murakkab turlarga massivlar, satrlar, funksiya va to'plamlarni misol tariqasida olish mumkin.

Murakkab turlardan foydalanish, dasturlashda bir nechta o'zgaruvchilardan foydalanishni yuqotadi. Massiv, satr va to'plam ko'rinishdagi ma'lumot turlari bir o'zgaruvchi yordamida bir nechta qiymatlarni kompyuter xotirasiga joylashtirish imkoniyatini beradi. Murakkab turlardan dasturlash tilida foydalanishda bir qancha qulayliklar mavjud, bunda o'zgaruvchini ixtiyoriy elementiga uning indeksleri orqali murojat qilish mumkin.

### **4.1 C++ TILIDA MASSIVLAR VA ULARDAN FOYDALANISH**

*Reja:*

- 1. Bir o'lchovli massivlar va ularning umumiy ko'rinishi;*
- 2. Ikki o'lchovli massivlar va ularning umumiy ko'rinishi;*
- 3. Rand funksiyasi va uning umumiy ko'rinishi;*

C++ dasturlash tilida kompyuter xotirasiga bir o'zgaruvchi yordamida bir nechta qiymatlarda foydalanishga to'g'ri keladi. Bir o'zgaruvchi bilan bir nechta qiymat ustida amallar bajarish uchun berilgan ma'lumotlar bir turga mansub bo'lishi kerak. C++ dasturlash tilida bir o'zgaruvchi yordamida bir nechta qiymatlardan foydalanish uchun massiv degan turdan foydalaniladi. Dasturlash tillarida ro'yxat yoki jadval ko'rinishidagi ma'lumotlarni massiv deb atashadi. Massiv so'zining ma'nosi o'lcham, o'lchov demakdir. Jurnalidagi o'quvchilar ro'yxati, paradda to'g'ri to'rtburchak hosil qilib turgan soldatlar, bog'da tartib bilan ekilgan daraxtlar, talabalarning reytingini aniqlovchi qaydnoma, determinantlar, matritsalar, Pifagorning ko'paytirish jadvali massivga misol bo'la oladi. Massiv o'z ichiga qamrab olgan har bir ob'ekt, ya'ni har bir o'quvchi, soldat, daraxt yoki determinant, matritsani hosil qiluvchi har bir son o'sha massivning elementi yoki massiv komponentasi deyiladi. Keltirilgan misollardan ko'rinib turibdiki, massivning barcha elementlari bitta turga mansub bo'lib, ular

bitta nom bilan nomlanadi va bir-birlaridan nomlari (indekslari) bilan farq qiladi.

Bu qismda dasturdagi ma'lumot strukturalari bilan tanishishni boshlaymiz. Dasturda ikki asosiy tur ma'lumot strukturalari mavjuddir. Birinchisi statik, ikkinchisi dinamikdir. Statik deganimizda xotirada egallagan joyi o'zgarimas, dastur boshida beriladigan strukturalarni nazarda tutamiz. Dinamik ma'lumot tiplari dastur davomida o'z hajmini, egallagan xotirasini o'zgartirishi mumkin. Agar struktura bir xil kattalikdagi tiplardan tuzilgan bo'lsa, uning nomi massiv (array) deyiladi. Massivlar dasturlashda eng ko'p qo'laniladigan ma'lumot tiplaridir. Massivlar hotirada ketma-ket joylashgan, bir tipdagi o'zgaruvchilar guruhidir. Alohida bir o'zgaruvchini ko'rsatish uchun massiv nomi va kerakli o'zgaruvchi indeksini yoziladi.

**Ta'rif:** *Bir turga mansub bo'lgan yagona nom bilan saqlanuvchi tartiblangan ma'lumotlar majmuasi massiv deyiladi.*

Massivlar yagona o'zgaruvchi bilan kompyuter xotirasiga saqlanadi, uning elementlari ma'lum bir indekslar bilan tartiblab joylashtiriladi.

C++ dasturlash tilida massivlardan foydalanishda masalan do'kundagi mahsulotlarning narxini olish mumkin. Mahsulot narxlarini massiv sifatida qaralganda  $narx_1, narx_2, narx_3, \dots, narx_n$  ko'rinishda bir nechta mahsulot narxlarini kompyuter xotirasiga saqlab undan foydalanish mumkin. Massivlar yagona nom bilan bir nechta qiymatni o'zida mujassamlashtiradi, bularga matematikadagi vektorlarni misol keltirish mumkin. Vektor ham yagona nom bilan saqlanib uning tarkibida bir nechta qiymatni o'zida mujassamlashtiradi. Vektorning ham elementlari bir turga mansub va tartiblangan bo'ladi.

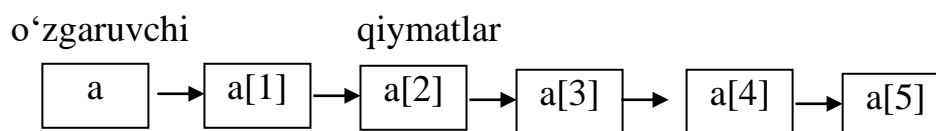
Massivlar holatiga ko'ra ikki turga bo'linadi.

- Bir o'lchovli massivlar;
- Ikki o'lchovli massivlar;

Bir o'lchovli massivlar ma'lumotlarni bir satrli ko'rinishda saqlansa, ikki o'lchovli massivlar esa ma'lumotlarni satrlar satri ko'rinishida saqlaydi.

### **Bir o'lchovli massivlar**

Odatda massivlar zarurat, katta hajmdagi tartiblangan, lekin chekli elementlarga oid masalalarni hal etishda yuzaga keladi. Dastur ishlatilishi davomida massivlar aniq nomga ega bo'lishi va uning elementlari ma'lum bir turda bo'lishi kerak. Bir o'lchovli massivlar kompyuter xotirasiga quyidagi shaklda saqlanadi.



Yuqoridagi holat bo'yicha massivlar kompyuter xotirasiga saqlanadi, bunda massivning ixtiyoriy elementiga murojat qilish uchun uning indeks nomeri bo'yicha murojat qilinadi. Bir o'lchovli massivning har bir elementi bitta nomerga ega bo'ladi. Bunday massivlar asosan ob'ektlar ro'yxati yoki bir o'zgaruvchili funksiya grafigini chizish uchun tuzilgan jadval kabi ma'lumotlardan tashkil topgan bo'ladi.

Bir o'lchovli massivlarni C++ dasturlash tilida bir nechta usullarda e'lon qilish mumkin. Bir o'lchovli massivlarni boshlang'ich qiymatlari berilmasdan C++ dasturlash tilida quyidagicha e'lon qilinadi.

**<tur> <massiv o'zgaruvchisi>[<element soni>;**

Massivni C++ dasturlash tilida e'lon qilish uchun, albatta, elementlar soni yoki massiv elementlarining boshlang'ich qiymatlari berilishi kerak. Masalan, **float a [20];**. Bunda **float** - massiv turi, **a** - massiv nomi. Massiv nomi faqat bitta harfdan iborat bo'lmasdan ixtiyoriy identifikator ham bo'lishi mumkin, **[20]**–massiv o'lchami, ya'ni massiv elementlari soni. Ushbu massivi quyidagicha o'qiladi. «Elementlari haqiqiy sonlardan iborat bo'lgan anomli massiv e'lon qilingan bo'lib, elementlari soni **20 ta** va ular **0** dan **19** gacha nomerlangan».

Massivni bu tarzda e'lon qilishdan maqsad EHM xotirasida uning elementlari uchun joy ajratishdir. Massivni e'lon qilmasdan uning elementlaridan dastur tarkibida foydalanib bo'lmaydi.

C++ dasturlash tilida massiv elementlari noldan boshlab nomerlanadi. Masalan, massiv o'lsami **n** bo'lsa uning elementlari **0** dan **n-1** gacha nomerlanadi.

Massiv elementlari nomerli kvadrat qavsga olib ko'rsatiladi. Demak, yuqorida e'lon qilingan massivning birinchi element **a[0]**, oxirgi elementi esa **a[19]** bo'ladi.

Massiv elementlari nomerli albatta butun son bo'lishi shart. Shuningdek, massiv indeks qiymati butun son bo'ladigan algebrik ifoda bo'lishi ham mumkin. Yoki massiv indeksi butun turli o'zgaruvchi ham bo'lishi mumkin. Shuning uchun agar massiv indeksini sikl parametri kabi aniqlash yo'li bilan massivning barcha elementlari ustida amallar bajarish mumkin.

**float** va **double** kabi haqiqiy turga mansub o'zgaruvchilarning qiymatlari har qanday kichik kesmada cheksiz ko'p bo'lganligi tufayli bunday o'zgaruvchilardan indeks sifatida foydalanib bo'lmaydi, ya'ni bunday o'zgaruvchilar yordamida massiv elementlarini tartiblab sanab bo'lmaydi.

Massiv uzunligi, ya'ni elementlari sonini aniqlash uchun **sizeof(massiv nomi)** funksiyasidan foydalaniladi.

C++ dasturlash tilida bir o'lovli massivni e'lon qilish.

```
#include <iostream.h>
int main()
{   int a[10];    //butun turli 10ta elementli massiv
    double b[10]; //haqiqiy turli 10ta elementli massiv
    return 0;
}
```

Massivni umumiy ko'rinishida birinchi o'zgaruvchi turi, massiv o'zgaruvchisi va massiv elementlari soni yoziladi. Massiv elementlari soni, albatta, butun sondan iborat bo'lishi kerak, chunki elementlar soni, albatta, butun bo'lishi kerak.

Massiv elementlari soni biror bir ifoda yoki yagona o'zgaruvchi bo'lishi mumkin, bitta o'zgaruvchi orqali massivning umumiy indekslarini ifodalash mumkin. Massiv elementlarini tashkil qilish va massiv elementlari ustida amallar bajarishni quyidagi masala orqali qaraymiz. Massiv elementlarini tartib nomeri doimo 0 dan boshlanadi.

*Misol:* 10 ta elementdan tashkil topgan massiv elementlarini hosil qilib, elementlarini ikkiga ko'paytirib ekranga chiqaring.

```
#include <iostream.h>
using namespace std;
int main()
{   int a[10];
    for(int i=0;i<=9;i++)
        cin>>a[i];
    for(int i=0;i<=9;i++)
        cout<<a[i]*2<<" ";
    return 0;
}
```

Kiritish ma'lumotlari

1 2 3 4 5 6 7 8 9 10

Dastur natijasi

2 4 6 8 10 12 14 16 18 20

C++ dasturlash tilida massiv elementlari boshlang'ich qiymatlari berilgan holatlarda e'lon qilish quyidagicha.

**<tur> <massiv o'zgaruvchisi>[<element soni>]={boshlang'ich qiymatlar};**

Bunda massiv elementlari oldindan berilgan holatlarda ishlatilishi mumkin.

```
#include <iostream.h>
int main()
{ int a[5]={2, 3, -4, 5, 6}; //butun turli 5ta elementli massiv
  return 0;
}
```

Boshlang'ich qiymatlari berilganda ham massivni e'lon qilishning bir nechta turlari mavjud. C++dasturlash tilida massiv elementlar soni aniq bo'lganda uning elementlarini to'liq berish yo'li bilan e'lon qilish.

```
int a[5]={2, 3, -4, 5, 6};
```

a massiv 4 ta elementli uning elementlarini boshlang'ich qiymatlari berilgan bo'lsa, bunda  $a[1]=2$ ,  $a[2]=3$ ,  $a[3]=-4$ ,  $a[4]=5$ ,  $a[5]=6$  ko'rinishida yozish mumkin.

Massiv elementlari soni aniq bo'lib, uning elementlarining boshlang'ich qiymatlari to'liq bo'lmagan holatlarda quyidagicha e'lon qilinadi.

```
int a[5]={2, 3, -4};
```

Yuqoridagi holatda massiv 5 ta elementan tashkil topgan, lekin uning uchta qiymatining boshlang'ich qiymatlari berilgan, qolgan ikkitasini boshlang'ich qiymatlari berilmagan. Agar massiv elementlarining boshlang'ich qiymatlari berilmagan bo'lsa, uning boshlang'ich qiymatlari 0 deb qaraladi.

Shuni e'tiborga olish kerakki, massivning boshidagi elementlarini boshlang'ich qiymatlari berilmasdan uning o'rtadagi yoki oxiridagi elementlarini boshlang'ich qiymatlarini berish mumkin emas.

Massiv elementlari soni berilmasdan uning boshlang'ich qiymatlarini berish yo'li bilan e'lon qilish.

```
int a[]={2, 3, -4, 5, 6};
```

Agar massiv elementlari soni berilmasa, albatta, uning boshlang'ich qiymatlari berilishi kerak. Massivning elementlar soni berilmaganda dastur komplyatori massiv elementlari sonini boshlang'ich qiymatlar soni bilan aniqlaydi.

**Misol:** Butun sonlardan iborat  $a[8]$  massiv berilgan uning juft elementlarini 2ga ko'paytirib toq elementlarini 3 ga ko'paytirib ekranga chiqaring.

Bu masala yechimida massiv elementining har birini juft va toqlikka tekshirib chiqiladi, juft bo'lsa, ikkiga toq bo'lsa, uchga ko'paytirildi.

```
#include <iostream.h>
using namespace std;
```



<pre> int main() {     int a[8];     for(int i=0;i&lt;=7;i++)         cin&gt;&gt;a[i];     for(int i=0;i&lt;=7;i++)         if(a[i]%2==0) {cout&lt;&lt;a[i]*2&lt;&lt;" ";}         else cout&lt;&lt;a[i]*3&lt;&lt;" ";     return 0; } </pre>	
Kiritish ma'lumotlari	Dastur natijasi
1 2 3 4 5 6 7 8	3 4 9 8 15 12 21 16

**Misol:**  $n$  natural son va  $n$  ta elementdan tashkil topgan massiv berilgan uning eng katta elementini aniqlang.

Bu masalani yechimini aniqlash uchun birinchi elementni eng katta deb qarab, massivning barcha elementini eng katta deb qaralgan element bilan solishtiriladi, agar solishtirilayotgan sondan kattasi topilsa, u katta bilan almashtiriladi.

<pre> #include &lt;iostream.h&gt; using namespace std; int main() {     int a[10];     for(int i=0;i&lt;=9;i++)         cin&gt;&gt;a[i];     int max=a[0];     for(int i=0;i&lt;=9;i++)         if(a[i]&gt;max) max=a[i];     cout&lt;&lt;max;     return 0; } </pre>	
Kiritish ma'lumotlari	Dastur natijasi
1 2 3 4 5 6 7 8 9 10	10

Massiv elementlarini e'lon qilish va uning elementlari ustida amallar bajarish yuqoridagi holatlar bo'yicha C++ dasturlash tilida e'lon qilib bajariladi.

Alohida bir o'zgaruvchini ko'rsatish uchun massiv nomi va kerakli o'zgaruvchi indeksini yozamiz. C/C++ dagi massivlardagi elementlar indeksi har doim noldan boshlanadi.

C++ dasturlash tilida massivlar va ulardan foydalanish keng imkoniyatlari mavjud, ko'pdan ko'p masalalarning hal etish uchun albatta massivlardan keng qo'llaniladi. Ko'pdan ko'p dasturchilarda birinchi massivlar ustida ishlayotgan vaqtida qo'yiladigan xatolar, bu massivni elementlariga murojat qilishda bo'ladi. Massiv elementlariga murojat birinchi tartibi 0 dan boshlanishini, albatta, e'tibor berish kerak.

Masalan, **float x[15], y[15];** kabi x va y massivlar e'lon qilingan bo'lsa, **x=y** yoki **y=x** kabi buyruqlarga ruxsat etilmaydi. Lekin bir massiv elementi qiymatini ikkinchi massiv elementi olishi mumkin. Masalan, **a[0]=b[10]** yoki **b[5]=a[3]** kabi buyruqlarga ruxsat etiladi.

Massiv o'lchami e'lon qilingandan keyin uning chegarasidan chiqib ketish kutilmagan xatoliklarga olib kelishi mumkin. Masalan, massiv int **b[40]** kabi e'lon qilingan bo'lsa, uning elementlari soni 40 ta bo'lib, ular 0 dan 39 gacha nomerlangan bo'lishi bizga ma'lum. Lekin **b[41]**, **b[46]**, **b[55]** elementlardan foydalanishga yoki ularga qiymat berishga ruxsat etilmaydi.

Yuqoridagi holatlarga, albatta, e'tibor berib C++dasturlash tilida massivlarni e'lon qilib ular ustida amallar bajarilish kerak.

### **Rand funksiyasi**

Massiv elementlarini ixtiyoriy tasodifiy sonlar bilan to'ldirish uchun C++ dasturlash tilida imkoniyat yaratilgan. Agar massiv elementlarini tasodifiy sonlar bilan to'ldirish kerak bo'lsa, tasodifiy sonlar bilan ishlash funksiyasiga murojat qilish kerak. C++ dasturlash tilida tasodifiy sonlarni hosil qilish **rand()** funksiyasining umumiy ko'rinishi quyidagicha bo'ladi.

#### **Rand();**

**rand()** funksiyasining vazifasi biror bir o'zgaruvchiga yoki massiv elementlariga tasodifiy sonni o'zlashtirish uchun xizmat qiladi. Rand() funksiyasi foydalanuvchi tomonidan dastur tarkibiga massiv elementlarini kiritishdan ozod qiladi.

***Misol:** A(n) massiv elementlarini tasodifiy sonlar yordamida hosil qilib uning juft elementlarini ikkiga ko'paytirib ekranga chiqaring.*

```
#include <iostream.h>
using namespace std;
```

```

int main()
{   int a[10];
    for(int i=0;i<=9;i++)
        a[i]=rand();
    for(int i=0;i<=9;i++)
        if(a[i]%2==0)
            cout<<a[i]*2<<" ";
    return 0;
}

```

Dastur natijasi

260 21964 2180 23312 45896 62252 18008

Demak, masala yechimiga e'tibor qaratsak, massivning tasodifiy sonlardan iborat 10 ta elementlari ichidan faqat yettitasi juft son ekan.

### **Ikki o'lchovli massivlar**

C++ dasturlash tilida ba'zi hollarda bir nechta o'lchamlari va turi bir xil bo'lgan bir o'lchovli massivlardan foydalanishga to'g'ri keladi. Bir nechta bir o'lchovli massivlarni birlashtirish natijasida ikki o'lchovli massivlarni hosil qilish mumkin. Ikki o'lchovli massivlarni tarkibida ma'lumotlar satrlarning satri ko'rinishida tasvirlanadi. Ikki o'lchovli massivlarning tarkibi ham bir o'lchovli massivlar kabi tartiblangan bir turga mansub bo'lishi kerak.

Ikki o'lchovli massivlarga matematikadagi matritsalar misol bo'lishi mumkin. Ikki o'lchovli massivlar tarkibidagi elementlar xuddi matritsani elementlari kabi tasvirlanadi.

*Ta'rif:* Bir turga mansub bo'lgan yagona nom bilan saqlangan matritsa ko'rinishdagi tartiblangan ma'lumotlar majmuasi ikki o'lchovli massivlar deyiladi.

Ikki o'lchovli massivning barcha elementlari aniq turga mansub bo'ladi va uning elementlari bir nechta satrlar ko'rinishida bo'ladi. Ikki o'lchovli massivlar quyidagi shaklda bo'ladi.

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

Yuqoridagi shakldan ko'rinib turibdiki ikki o'lchovli massiv bir o'lchovli massivlarning bir nechtasi yoki matritsa ko'rinishida tasvirlanar ekan. Ikki o'lchovli massivlarning kompyuter xotirasiga har bir satr uchun alohida tartib nomer bilan saqlanadi. Ikki o'lchovli massivlarning har bir elementiga o'zining indeksi bo'yicha murojat qilinadi.

Ikki o'lchovli massivlarni C++ dasturlash tilida e'lon qilish va uning umumiy ko'rinishi quyidagicha.

**<tur> <massivO'zgaruvchisi>[<element1 soni>][<element2 soni>;**

Ikki o'lchovli massivlarni C++ dasturlash tilida e'lon qilish uchun massiv turi, massiv nomi, massiv satrdagi elementlar soni va massiv ustunidagi elementlar soni beriladi. Agar a[5,2] massiv berilgan bo'lsa, uni C++ dasturlash tilida a[2][5] ko'rinishda e'lon qilinadi. C++ dasturlash tilida massivlarni yozilishi, matematikadan bir oz farq qilinadi, ya'ni ustun va satrlari joyi almashtiriladi. Ikki o'lchovli massivlarni C++ dasturlash tilida e'lon qilinishi.

```
#include <iostream.h>
int main()
{
    int a[10][10]; //butun turli 100ta elementli massiv
    double b[10][10]; //haqiqiy turli 100ta elementli massiv
    return 0;
}
```

Massivlarni elementlariga murojat qilish uchun uning har bir elementining o'z indeksi bo'yicha murojat qilinadi. Massiv elementlariga murojat qilishni quyidagi shakl orqali bajariladi. Quyidagi ikki o'lchovli massiv berilgan bo'lsin.

$$a = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

a(3,3) massivning elementlariga C++ dasturlash tilida murojat qilish quyidagicha ko'rinishda bo'ladi.

$$a[1][1]=1, a[2][1]=2, a[3][1]=3, a[1][2]=4, a[2][2]=5, a[3][2]=5, a[1][3]=4, \\ a[2][3]=5, a[3][3]=5,$$

C++ dasturlash tilida ikki o'lchovli massivlarning elementlarini boshlang'ich qiymatlarini quyidagi tartibda berilishi mumkin.

```
#include <iostream.h>
int main()
{
    int a[][]={{1,2,3}{4,5,6}{7,8,9}};
    return 0;
}
```

Massiv elementlari soni oldindan berilmasa, kompyator massiv elementlar sonini boshlang'ich qiymatlarga qarab aniqlaydi.

Ikki o'lchovli massiv elementlari boshlang'ich qiymatlari oldindan berilmasa, albatta, uning elementlar sonini oldindan berish kerak bo'ladi. Ikki o'lchovli massiv elementlari soni aniq, lekin uning elementlarini boshlang'ich qiymatlari berilmaganda, massiv elementlarini shakllantirish.

```
#include <iostream.h>
int main()
{
    int a[3][3];
    for(int i=0;i<=2;i++)
        for(int j=0;j<=2;j++)
            cin>>a[i][j];
    return 0;
}
```

**Misol:**  $A(2,3)$  massiv berilgan uning elementlarini ikkiga ko'paytirib ekranga chiqaring.

Bu masala yechimini aniqlashda ham bir o'lchovli massivlar kabi har bir elementiga murojat qilib, har bir elementini ikkiga ko'paytirish kerak.

```
#include <iostream.h>
int main()
{
    int a[3][2];
    for(int i=0;i<=2;i++)
        for(int j=0;j<=1;j++)
            cin>>a[j][i];
    for(int i=0;i<=2;i++)
    {
        for(int j=0;j<=1;j++)
            cout<<2*a[j][i]<<" ";
        cout<<"\n"; }
    return 0;
}
```

Kiritish uchun ma'lumot

1 2 3  
4 5 6

Dastur natijasi

2 4 6  
8 10 12

Ikki o‘lchovli massivlar yordamida C++ dasturlash tilida matematikani balki boshqa sohalarning ham bir necha masalalarini hal etish mumkin.

**Misol:**  $A(n,n)$  matritsa berilgan uning asosiy diogonalidan yuqoridagi elementlarini ikkiga ko‘paytirib ekranga chiqaring.

<pre> #include &lt;iostream.h&gt; int main() {     int a[90][90];     int n,m; cout&lt;&lt;"n=";     cin&gt;&gt;n; m=n;     for(int i=0;i&lt;n;i++)         for(int j=0;j&lt;n;j++)             cin&gt;&gt;a[j][i];     for(int i=0;i&lt;n;i++)     {         m=m-i; j=0;         while(j&lt;=m)         {             cout&lt;&lt;2*a[j][i]&lt;&lt;" ";             j++;         }         cout&lt;&lt;"\n";     }     return 0; } </pre>	
<p>Kiritish uchun ma'lumot</p> <p>n=3</p> <p>1 2 3</p> <p>4 5 6</p> <p>7 8 9</p>	<p>Dastur natijasi</p> <p>2 4 6</p> <p>8 10</p> <p>14</p>

Ikki o‘lchovli massiv elementlarining har birini tekshirish yoki ular ustida amallar bajarish uchun ichma ich sikllardan foydalaniladi. Yuqoridagi masala yechimida ikki o‘lchovli massivning barcha elementlari tekshirilmaydi shuning uchun ichki siklning while yordamida yozish maqsadga muvofiq bo‘ladi. Massiv o‘lchami kattalashgan sayin uning elementlari uchun xotiradan ajratiladigan joy hajmi juda tez o‘sadi. Masalan, 32 razryadli mashinada uch o‘lchamli float b [10] [8] [10] massivi e‘lon qilingan bo‘lsa, u xotirada  $10*10*8*4=3200$  bayt joyni egallaydi.

Agar shu massiv int b [10] [10] [10] [10] kabi to'rt o'lchamli ko'rinishda e'lon qilinsa, xotiradan 40000 baytlik joyni band qilinadi. Demak, massivning o'lchamlari soniga chegara qo'yilmagan, agar xotira muammosiga duch kelinmasa. n o'lchamli massivning xotiradan egallaydigan joyi formulasi quyidagicha. Baytlar soni =1-o'lcham \* 2-o'lcham \*.....\* n-o'lcham \* turning baytlardagi o'lchami. Amalda uch o'lchamli massivlardan ham juda kam foydalaniladi. Uch o'lchamdan yuqori o'lchamli massivlardan esa deyarli foydalanilmaydi.

Massivlarga oid masalalar yuqoridagi holatlar bo'yicha C++ dasturlash tilida tasvirlanadi.

### Nazariy savollar

- 1 Murakkab tur deganda nimani tushunasiz?
- 2 Massiv deb nimaga aytiladi?
- 3 Massiv nechta turga bo'linadi?
- 4 Bir o'lchovli massiv deb nimaga aytiladi?
- 5 Bir o'lchovli massivning umumiy ko'rinishi?
- 6 Bir o'lchovli massivning e'lon qilish usullari?
- 7 Ikki o'lchovli massivning umumiy ko'rinishi?
- 8 Ikki o'lchovli massivning e'lon qilish usullari?
9. Massiv deganda nimani tushunasiz va u qanday ma'noni anglatadi?

Misollar keltiriting.

10. Massivlar qanday va qayerda e'lon qilinadi? Massiv nima maqsadda e'lon qilinadi?
11. Necha xil massivlardan foydalaniladi?
12.  $M(n,m)$  massivda nechta ustun va nechta satr bor?
13. Massiv elementlari boshqa kattaliklar majmuasidan qanday farq qiladi?
14. Bir, ikki, uch o'lchovli massiv elementlari qanday yoziladi? Yozib ko'rsating.
15. Matritsa nima? Kvadrat matritsa qanday matritsa?
16. Kvadrat matritsaning:
  - a) bosh diogonalida;
  - b) bosh diogonalidan yuqorida;
  - c) bosh diogonalidan pastda joylashgan elementlari qanday xususiyatlarga ega?
17. float c[30]; massivi e'lon qilingan bo'lib, uning elementlari  $c_i=2*i-7$  qonuniyat bo'yicha o'zgarsa, uning n-elementi nachaga teng bo'ladi?
18. Elementlari butun turli D nomli ikki o'lchovli massiv elementlari 6 satr va 23 uctunda joylashgan. Shu massivni e'lon qiling.

19 Rand funksiyasi va uning umumiy ko'rinishi?

20 Rand funksiyasi vazifasi?

### **Mustaqil ishlash uchun topshiriqlar**

1. Natural  $n$  va  $m$  hamda haqiqiy sonli  $A(1:n, 1:m)$  massiv berilgan. Uning o'rta arifmetik qiymatini toping.

2. Natural  $n$  va  $m$  hamda haqiqiy sonli  $A(1:n, 1:m)$  massiv berilgan bo'lsin. Uning eng kattasi elementini toping.

3. Natural  $n$  va  $m$  hamda haqiqiy sonli  $A(1:n, 1:m)$  massiv berilgan. Uning eng kichik elementi necha marta uchraydi?

4.  $N$  butun soni va haqiqiy sonli  $B(1:N, 1:N)$  massiv berilgan. Uning diagonal elementlari orasida eng kattasini aniqlang.

5. Butun  $a_1, a_2$ , va  $a_3$  sonlari berilgan. Butun sonli  $B(1:3, 1:3)$  jadval elementlarini  $b_{i,j} = a_i - 3a_j$  formula yordamida aniqlang. Bu jadvalning barcha elementlari ko'paytmasini hisoblang.

6. Butun  $a_1, a_2, a_3$  va  $a_4$  sonlari berilgan bo'lsin. Butun sonli  $B(1:4, 1:4)$  jadvalning elementlari

$$b_{ij} = \frac{2a_i - 3a_j}{i+j}$$

formula yordamida aniqlanadi. Bu jadvalning eng katta va eng kichik elementlarining tartib raqamlarini aniqlang.

7. Natural  $n$  va  $m$  hamda haqiqiy sonli  $A(1:n, 1:m)$  massiv berilgan bo'lsin. Shumassivning har bir satridagi eng katta elementlar ichida eng kichigini toping.

8.  $n$  butun soni va haqiqiy sonli  $B(1:n, 1:n)$  massiv berilgan. Uning bosh va qarama-qarshi diagonallaridagi elementlar yig'indisini hisoblang.

9. Natural  $n$  va  $m$  hamda haqiqiy sonli  $A(1:n, 1:m)$  massiv berilgan bo'lsin. Shu massivning har bir ustunidagi eng kichik elementlar ichida eng kattasini toping.

10. Natural  $n$  va  $m$  hamda haqiqiy sonli  $A(1:n, 1:m)$  massiv berilgan bo'lsin. Shu massivning birinchi elementlari musbat bo'lgan ustunlaridagi elementlarning yig'indisini hisoblansin.

11. Natural  $n$  va  $m$  hamda haqiqiy sonli  $A(1:n, 1:m)$  massiv berilgan bo'lsin. Shumassivning birinchi elementlari musbat bo'lgan satrlardagi elementlarning ko'paytmasini toping

12. Butun sonli  $A(1:10, 1:20)$  massivda necha xil elementlar uchrashini aniqlang.



13. Butun sonli  $A(1:10, 1:10)$  massivda bir xil elementlar mavjud yoki mavjud emasligini aniqlang.

14.  $n$  butun soni va haqiqiy sonli  $B(1:n,1:n)$  massiv berilgan. Hisoblang:  $x_1y_1 + \dots + x_ny_n$ . Bu yerda  $x_i$ - $B$  ning  $i$ -satridagi eng katta element,  $y_j$ - $B$  ning  $j$  ustunidagi eng kichik element.

15.  $n$  natural soni berilgan bo'lsin.  $A(1:n, 1:n)$  haqiqiy sonli jadval elementlarini

$$a_{ij} = \begin{cases} \sin(i+j) & , \text{agar } i < j \\ 1 & , \text{agar } i = j \\ \text{ctg}\left(i + \frac{j}{2i} + j\right) & , \text{agar } i > j \end{cases}$$

formula yordamida aniqlang.

16. Natural  $n$  va  $m$  hamda haqiqiy sonli  $A(1:n,1:m)$  massiv berilgan. Shu massivning eng katta va eng kichik elementlari joylashgan satrlari o'rinlarini almashtiring.

17. Natural  $n$  va  $m$  hamda haqiqiy sonli  $A(1:n, 1:m)$  massiv berilgan. Shu massivning eng katta va eng kichik elementlari joylashgan ustunlari o'rinlarini almashtiring.

18.  $n$  tartibli  $B$  kvadrat matrisa berilgan bo'lsin. Unda hamma elementlari juft sonlardan iborat bo'lgan ustun mavjudmi ?

19.  $n$  tartibli  $B$  kvadrat matrisa berilgan bo'lsin. Unda hamma elementlari bir xil bo'lgan satrlar qancha ?

20.  $m$  tartibli  $A$  kvadrat matrisa berilgan. Unda ikki qo'shni elementlarning o'rta geometrik quymatiga teng bo'lgan elementlar joylashgan satrlarning tartib raqamlarini aniqlang.

21.  $m$  tartibli  $A$  kvadrat matrisa berilgan. Unda elementlarining yig'indisi ikki qo'shni ustunlardagi elementlar yig'indisining yarmiga teng bo'lgan ustun mavjudmi ?

22.  $n$  tartibli  $B$  kvadrat matrisa berilgan. Unda elementlari simmetrik usulda joylashgan ustunlar mavjudmi ?

23.  $n$  natural,  $x$  haqiqiy sonlar hamda  $A(1:n,1:n)$  haqiqiy sonli jadval berilgan bo'lsin.  $B(n)$  - bir o'lchovli jadval elementlarini aniqlang. Bu yerda  $b_i=1$ , agar  $A$  ning  $i$ -satrida  $x$  dan katta bo'lgan elementlar mavjud bo'lmasa, aks holda  $b_i=0$ .

24.  $m$  tartibli  $A$  kvadrat matrisa berilgan. Unda elementlari o'sish tartibida joylashgan satr mavjudmi ?

25.  $m$  tartibli  $A$  kvadrat matrisa berilgan. Unda elementlari kamayish tartibida joylashgan biror ustun mavjudmi?

26. Natural  $n$  va  $m$  sonlar hamda  $A(1:n, 1:m)$  haqiqiy sonlar jadvali berilgan. Shu jadvalning har bir satridagi elementlarni o'sish tartibida tartiblang.

27. Natural  $n$  va  $m$  sonlar hamda  $A(1:n, 1:m)$  haqiqiy sonlar jadvali berilgan. Shu jadvalning har bir ustunidagi elementlarni kamayish tartibida qayta joylashtiring.

28. Natural  $n$  va  $m$  sonlari hamda  $A(1:n, 1:m)$  haqiqiy sonlar jadvali berilgan. Bu jadvalning diagonal elementlarini o'sish tartibida tartiblang.

29. Natural  $n$  soni hamda  $A(1:n, 1:n)$  haqiqiy sonlar jadvali berilgan. Shu jadvalning qarama-qarshi diagonal elementlarini kamayish tartibida tartiblang.

30. Mamlakat futbol chempionati (20x20) jadval orqali berilgan bo'lsin. Uning bosh diagonal elementlari 4 lardan iborat bo'lib, qolgan elementlari 0, 1 yoki 3 ga teng. Shu jadvalni to'ldiring. Bunda bosh diagonaldan yuqoridagi elementlarning qiymatlari kiritiladi. Bosh diagonalning quyi qismidagi elementlarni

$$a_{ij} = \begin{cases} 0, & \text{agar } a_{ji} = 3 \\ 1, & \text{agar } a_{ji} = 1 \\ 3, & \text{agar } a_{ji} = 0 \end{cases}$$

formula bilan aniqlang.

## 4.2 C++ DASTURLASH TILIDA MASSIV ELEMENTLARINI SARALASH

*Reja:*

1. Bir o'lchovli massiv elementlarini tartiblash;

2. Ikki o'lchovli massiv elementlarini tartiblash;

Massiv elementlarini o'sish yoki kamayish tartibida joylashtirish masalasi massivlarga oid masalalarning asosiy vazifasi hisoblanadi. Dasturlash asoslari fanining massivlarga oid masalalari tarkibining aksariyat hollarida massiv elementlarini o'sish yoki kamayish tartibida bo'lishi ta'lab etiladi. Lekin massiv elementlarini hosil qilishda yoki kompyuter xotirasiga kiritishda massiv elementlari tartiblanmagan bo'ladi. Massiv elementlarini tartiblash uchun bir nechta usullar mavjud, lekin biz bu massiv elementlarini tartiblash uchun bir nechta tushunish oson bo'lgan usullarni qarab o'tamiz.

## Bir o'lovli massiv elementlarini tartiblash

Massiv elementlarini tartiblash jarayoni ikki xil bo'ladi, ya'ni massiv elementlarini o'sish yoki kamayish bo'yicha tartiblash mumkin. C++ dasturlash tilida massiv elementlarini o'sish va kamayish bo'yicha tartiblashda bir biridan katta farq qilmaydi, ularning algoritmi bir xil bo'ladi faqat o'sish va kamayish bo'yicha tartiblashda bitta  $>$  yoki  $<$  shartlar bilan farq qiladi. Shuning uchun tartiblash masalalarini bittasini o'sish yoki kamayishni tushuntirib o'tamiz siz uning tarkibidagi shartni teskarisiga almashtirsangiz o'sish bo'lsa kamayish, kamayish bo'lsa o'sishga almashadi. Ketma-ketliklar va ular ustida amallar orqali hal etiladigan masalalar tarkibida ketma-ketliklarning o'sish yoki kamayish tartibda saralash muommolari yotadi.

Dastlab saralash algoritmlarinig eng oddiy turini ko'rib o'tamiz. Bunda berilgan ketmaketlikning har bir elementi o'zidan keying elementlar bilan solishtiriladi agar kichik bo'lsa almashtiriladi aks holda keyingisi bilan solishtiriladi, shu tariqa massivning kichik elementlari ro'yxat boshiga kattalari esa ro'yxat oxiriga borib joylashadi.

Algoritmdagi jarayonni aniqlash uchun quyidagi masalaga e'tibor bering.

$A=[3; 5; 1; 0; 6]$  ketma ketlikni oddiy usul bilan saralash algoritmi bo'yicha quyidagi almashtirishlar hosil bo'ladi.

$[3; 5; 1; 0; 6]$ ,  $[1; 5; 3; 0; 6]$ ,  $[0; 5; 3; 1; 6]$ ,  $[0; 3; 5; 1; 6]$ ,  $[0; 1; 5; 3; 6]$ ,  $[0; 1; 3; 5; 6]$ .

Demak yuqoridagi misolga e'tibor bersak eng kichik elementlar ro'yxat boshiga birin keyin harakatlanib o'tmoqda natijada berilgan ketma ketlik o'sish tartibida saralanib qoladi. Bu usulni C++ dasturlash tilida quyidagicha tasvirlashimiz mumkin.

```
#include <iostream.h>
using namespace std;
int main()
{   int a[90];
    int n,t; cout<<"n=";
    cin>>n;
    for(int i=0;i<n;i++)
        cin>>a[i];
    for(int i=0;i<n-1;i++)
        for(int j=i+1;j<n;j++)
            if(a[i]>a[j]) { t=a[i];
```

<pre>                 a[i]=a[j];                 a[j]=t;             }     for(int j=0;j&lt;n;j++)         cout&lt;&lt;a[j]&lt;&lt;" ";     cout&lt;&lt;"\n";     return 0; } </pre>	
Kiritish uchun ma'lumot n=5 3 5 1 0 6	Dastur natijasi 0 1 3 5 6

Yuqorida keltirib o'tilgan algoritm bilan massiv kamayish tartibida ham saralash mumkin, faqatgina sikl ichidagi shartni almashtirish kerak. Kamayish bo'yicha tartiblash uchun C++ dasturlash tilida quyidagi dasturga e'tibor bering.

<pre> #include &lt;iostream.h&gt; using namespace std; int main() {     int a[90];     int n,t; cout&lt;&lt;"n=";     cin&gt;&gt;n;     for(int i=0;i&lt;n;i++)         cin&gt;&gt;a[i];     for(int i=0;i&lt;n-1;i++)         for(int j=i+1;j&lt;n;j++)             if(a[i]&lt;a[j]) { t=a[i];                           a[i]=a[j];                           a[j]=t;                         }     for(int j=0;j&lt;n;j++)         cout&lt;&lt;a[j]&lt;&lt;" ";     return 0; } </pre>	
Kiritish uchun ma'lumot n=5 3 5 1 0 6	Dastur natijasi 6 5 3 1 0

Saralash uchun bir nechta usullar mavjud, lekin ko‘p hollarda biz ketma-ketliklarni saralashda vaqt chegarasidan yutqazib qo‘yamiz.

Bunday holatlarni oldini olish uchun ba’zi mavjud usullarni keltiramiz.

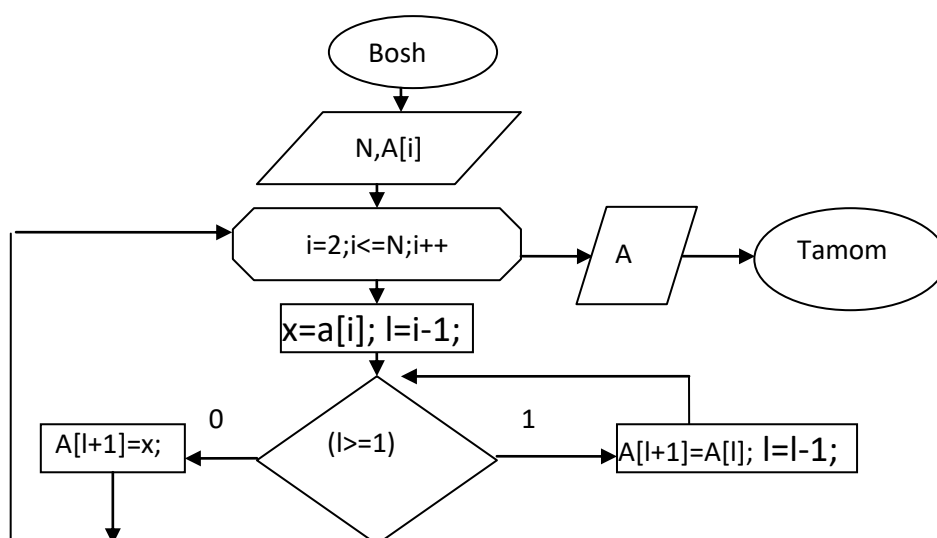
*O‘rniga qo‘yish usuli bilan saralash algoritmi:* Bunda ikkinchi elementdan boshlab har bir element tanlab olinib o‘zidan oldingi elementlar bo‘yicha solishtiriladi. Natijada tanlangan element o‘zidan oldingi elementlar ichida solishtirish natijasida o‘z joyiga borib tushadi. Bu holat toki oxirgi elementgacha bajarilib boradi.

Algoritmdagi jarayonni aniqlash uchun quyidagi masalaga e‘tibor bering.

A=[3; 5; 1; 0; 6] ketma ketlikni o‘rniga qo‘yish usuli bilan saralash algoritmi bo‘yicha quyidagi almashtirishlar hosil bo‘ladi.

[3; 5; 5; 0; 6], [3; 3; 5; 0; 6] , [1; 3; 5; 0; 6] , [1; 3; 5; 5; 6], [1; 3; 3; 5; 6], [1; 1; 3; 5; 6], [0; 1; 3; 5; 6]

Ushbu jarayonni amalga oshiruvchi algoritm quyidagicha:



Algoritm bo‘yicha C++ tilida quyidagi dastur asosida ko‘p elementdan iborat massivlarni ham saralash mumkin.

```

#include <iostream.h>
#include <vcl.h>
#pragma hdrstop
using namespace std;
#pragma argsused
int main(int argc, char* argv[])
{ int l,i,x,n; int a [100]; cout<<"n="; cin>>n;
  
```

```

for(int i=1;i<=n;i++) cin>>a[i];
for(int i=2;i<=n;i++)
{ x=a[i]; l=i-1; while((l>=1)&(a[l]>x)) { a[l+1]=a[l]; l=l-1; }
a[l+1]=x;
}
for(int j=1;j<=n;j++) cout<<a[j]<<" ";
system("pause");
return 0; }

```

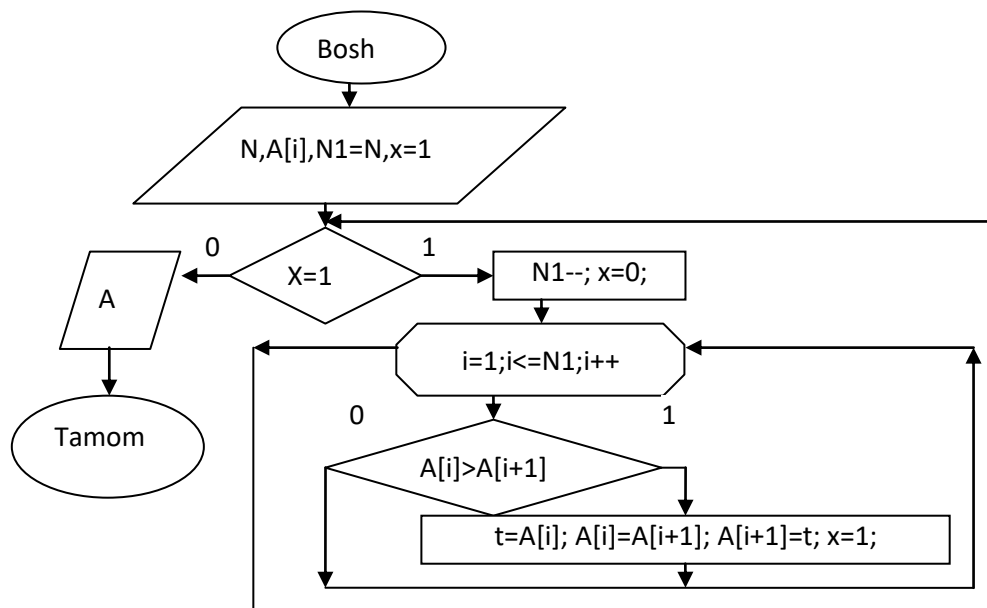
*O‘rin almashtirish usuli bilan saralash algoritmi:* Usulning asosiy prinsipi katta elementlarning ro‘yxat uchiga otilib chiqadi va bu vaqtda kichik qiymatlar pastga tushadi. Usulda ichma ich sikldan foydalaniladi birinchi n-1 marta, ikkinchi n-2 marta va oxiri bir marta har bir element o‘zidan keyingi element bilan solishtiriladi.

Bunda kichik elementlar ro‘yxat oxiriga katta elementlar ro‘yxat boshiga tushadi. Kuzatish mumkinki, har bir o‘tishda bir qancha elementlar siljiydi va bitta elementgina o‘zining o‘rnini qat’iy egallaydi.

Algoritmdagi jarayonni aniqlash uchun quyidagi masalaga e’tibor bering. A=[3; 5; 1; 0; 6] ketma ketlikni o‘rin almashtirish usuli bilan saralash algoritmi bo‘yicha quyidagi almashtirishlar hosil bo‘ladi.

[3; 1; 5; 0; 6], [3; 1; 0; 5; 6], [1; 3; 0; 5; 6], [1; 0; 3; 5; 6], [0; 1; 3; 5; 6]

O‘rin almashtirish usuli bilan saralash algoritmining ko‘rinishi quyidagicha:



O‘rin almashtirish usuli C++ tilida quyidagi ko‘rinishda bo‘ladi.

```

#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{ int l,i,x,n,t; int a [100]; cout<<"n="; cin>>n; int N1=n; x=1;
  for(int i=1;i<=n;i++) cin>>a[i];
  while(x==1)
{ N1--; x=0;
  for(int i=1;i<=N1;i++)
  if (a[i]>a[i+1])
  { t=a[i]; a[i]=a[i+1]; a[i+1]=t; x=1;}
} for(int j=1;j<=n;j++) cout<<a[j]<<" ";
  system("PAUSE"); return 0; return EXIT_SUCCESS; }

```

Yuqoridagi ikki saralash usuli yordamida ko‘p elementli massivlarni qisqa vaqt ichida saralash imkoniyatiga ega bo‘lamiz.

### **Ikki o‘lchovli massiv elementlarini tartiblash**

Massiv elementlarini saralash usullari yordamida ikki o‘lchovli massivlar elementlarini saralash mumkin. Ikki o‘lchovli massiv elementlarini saralash uchun alohida usul ishlab chiqish shart emas, chunki bir o‘lchovli massivlarga ishlatadigan usullarni bir sikl ortig‘i bilan ikki o‘lchovli massivlarga tadbiiq etish mumkin. Ikki o‘lchovli massiv elementlarini saralash uchun uning har bir satr elementlari uchun bir o‘lchovli massivlar uchun o‘rinli bo‘lgan uchta saralash algoritmlari qo‘llaniladi. Ikki o‘lchovli massiv elementlarini saralash uchun quyidagi masala algoritmiga e‘tibor bering.

**Misol:**  $A(n,n)$  massiv berilgan uning har bir satrini o‘shish tartibida saralang.

Bu masalani yechimini aniqlash uchun bir o‘lchovli massivlarni saralashning birinchi usulini ikki o‘lchovli massivni har bir satriga tadbiiq etiladi.

```

#include <iostream.h>
using namespace std;
int main()
{ int a[10][10];
  int n,t; cout<<"n=";
  cin>>n;

```

<pre> for(int i=0;i&lt;n;i++) for(int j=0;j&lt;n;j++)     cin&gt;&gt;a[j][i]; for(int i=0;i&lt;n;i++) for(int k=0;k&lt;n-1;k++) for(int j=k+1;j&lt;n;j++)     if(a[k][i]&gt;a[j][i]) { t=a[j][i];                         a[j][i]=a[k][i];                         a[k][i]=t;                     } for(int i=0;i&lt;n;i++) {     for(int j=0;j&lt;n;j++)         cout&lt;&lt;a[j][i]&lt;&lt;" ";     cout&lt;&lt;"\n"; } return 0; } </pre>	
<p>Kiritish uchun ma'lumot</p> <pre> n=3 3 5 1 0 6 4 8 2 9 </pre>	<p>Dastur natijasi</p> <pre> 1 3 5 0 4 6 2 8 9 </pre>

Yuqorida keltirib o‘tilgan algoritm bilan ikki o‘lchovli massiv elementlarini kamayish tartibida ham saralash mumkin faqatgina sikl ichidagi shartni almashtirish kerak. Kamayish bo‘yicha tartiblash uchun C++ dasturlash tilida quyidagi dasturga e’tibor bering.

```

#include <iostream.h>
using namespace std;
int main()
{
    int a[10][10];
    int n,t; cout<<"n=";
    cin>>n;
    for(int i=0;i<n;i++)
    for(int j=0;j<n;j++)

```



<pre> cin&gt;&gt;a[j][i]; for(int i=0;i&lt;n;i++) for(int k=0;k&lt;n-1;k++) for(int j=k+1;j&lt;n;j++) if(a[k][i]&lt;a[j][i]) { t=a[j][i];                     a[j][i]=a[k][i];                     a[k][i]=t;                     } for(int i=0;i&lt;n;i++) { for(int j=0;j&lt;n;j++) cout&lt;&lt;a[j][i]&lt;&lt;" "; cout&lt;&lt;"\n"; } return 0; } </pre>	
Kiritish uchun ma'lumot	Dastur natijasi
n=3	5 3 1
3 5 1	6 4 0
0 6 4	9 8 2
8 2 9	

Ikki o'lchovli massivlarni samarali usullar bilan saralash uchun bir o'lchovli massivlar uchun o'rinli bo'lgan o'rniga qo'yish va o'rin almashtirish usullarini tadbiq etish yo'li bilan saralanadi.

### Nazariy savollar

- 1 Bir o'lchovli massivlarni saralash usullarini ayting?
- 2 Bir o'lchovli massivlarni oddiy, o'rniga quyish va o'rin almashtirish usullarini tushuntirib bering?
- 3 Bir o'lchovli massivlarni saralash usullarini ayting?
- 4 Bir o'lchovli massivlarni oddiy, o'rniga qo'yish va o'rin almashtirish usullarini tushuntirib bering?

### Mustaqil ishlash uchun topshiriqlar

1.  $n$  natural soni berilgan. Dastlabki  $n$  ta toq sondan tashkil topgan massivni hosil qiling massiv elementlarini o'sish tartibida saralang va elementlarini chiqaring.

2.  $n$  natural soni berilgan. 2 sonining dastlabki  $n$  ta darajasidan tashkil topgan massivni hosil qiling, massiv elementlarini o'sish tartibida saralang va elementlarini chiqaring. (1, 2, 4, 8, .....)

3.  $n$  natural soni va arifmetik progressiyaning dastlabki hadi  $A$  va ayirmasi  $D$  berilgan. Arifmetik progressiyaning dastlabki  $n$  ta hadidan tashkil topgan massivni hosil qiling va massiv elementlarini o'sish tartibida saralang, elementlarini chiqaring.

$$A_i = A_{i-1} + D$$

4.  $n$  natural soni va geometrik progressiyaning dastlabki hadi  $A$  va maxraji  $D$  berilgan. Geometrik progressiyaning dastlabki  $n$  ta hadidan tashkil topgan massivni hosil qiling, massiv elementlarini o'sish tartibida saralang va elementlarini chiqaring.

$$A_i = A_{i-1} * D$$

5.  $n$  natural soni berilgan. Dastlabki  $n$  ta Fibonachchi sonlaridan tashkil topgan massivni hosil qiling, massiv elementlarini o'sish tartibida saralang va elementlarini chiqaring.

$$F_0 = 1; F_1 = 1; F[k] = F[k-1] + F[k-2]; k=2, 3, 4, \dots$$

6.  $n$  natural soni va  $A, B$  butun sonlari berilgan ( $n > 2$ ).  $a[0] = A; a[1] = B$ ; boshqa elementlari o'zidan oldingi barcha elementlari yig'indisiga teng bo'lgan massivni hosil qiling, massiv elementlarini o'sish tartibida saralang va elementlarini chiqaring.

7.  $n$  ta elementdan tashkil topgan massiv berilgan. Uning elementlari teskari tartibda chiqaruvchi dastur tuzilsin.

8.  $n$  ta elementdan tashkil topgan massiv berilgan. Uning elementlari orasidan toqlarini indeksleri o'sish tartibida chiqaruvchi va ularning sonini chiqaruvchi dastur tuzilsin.

Massiv elementlar: 4 5 7 8 6 9

Natija: 5 7 9 toqlar soni = 3

**9.**  $n$  ta elementdan tashkil topgan massiv berilgan. Uning elementlari orasidan juftlarini indeksleri kamayish tartibida chiqaruvchi va ularning sonini chiqaruvchi dastur tuzilsin.

Massiv elementlar: 4 5 7 8 6 9

Natija: 6 8 4 juftlar soni = 3

**10.**  $n$  ta elementdan tashkil topgan massiv berilgan. Dastlab massiv elementlari orasidan juftlarini indeksleri o'sish tartibida chiqaruvchi, keyin massiv elementlari orasidan toqlarini indeksleri kamayish tartibida chiqaruvchi dastur tuzilsin.

Massiv elementlar: 4 5 7 8 6 9

Natija: 4 6 8 9 7 5

**11.**  $n$  ta elementdan tashkil topgan massiv va  $K$  butun soni berilgan ( $1 \leq K < n$ ). Massiv elementlari orasidan indksi  $K$  ga karralilarini kamayish tartibida chiqaruvchi dastur tuzilsin.  $A_k, A_{2k}, A_{3k}, \dots$ . Shart operatori ishlatilmasin.

**12.**  $n$  ta elementdan tashkil topgan massiv berilgan ( $n$  juft son). Massiv elementlari orasidan quyidagilarni kamayish tartibida chiqaruvchi dastur tuzilsin.  $A[0], A[2], A[4], \dots$ . Shart operatori ishlatilmasin.

**13.**  $n$  ta elementdan tashkil topgan massiv berilgan ( $n$  toq son). Massiv elementlari orasidan quyidagilarni o'sish tartibida chiqaruvchi dastur tuzilsin.  $A[n-1], A[n-3], \dots, A[1]$ . Shart operatori ishlatilmasin.

**14.**  $n$  ta elementdan tashkil topgan massiv berilgan. Dastlab massiv elementlari orasidan juft indekslerni keyin toq indekslerni kamayish tartibida chiqaruvchi dastur tuzilsin.  $A[0], A[2], A[4], \dots, A[1], A[3], A[5], \dots$ . Shart operatori ishlatilmasin.

**15.**  $n$  ta elementdan tashkil topgan massiv berilgan ( $n$  juft son). Dastlab massiv elementlari orasidan toq indekslilarini o'sish tartibida keyin juft indekslilarini kamayish tartibida chiqaruvchi dastur tuzilsin.  $A[1], A[3], A[5], \dots, A[6], A[4], A[2], A[0]$ . Shart operatori ishlatilmasin.

**16.**  $n$  ta elementdan tashkil topgan massiv berilgan. Massiv elementlari quyidagicha chiqaruvchi dastur tuzilsin.

$A[0], A[n-1], A[1], A[n-2], A[2], A[n-3], \dots$

*17.  $n$  ta elementdan tashkil topgan massiv berilgan. Massiv elementlari quyidagicha chiqaruvchi dastur tuzilsin.  $A[0], A[1], A[n-1], A[n-2], A[3], A[4], A[n-3], A[n-4], \dots$*

*18.  $n$  ta elementdan tashkil topgan massiv berilgan. Massiv elementlari ichidan musbatlarini alohida manfiylarini alohida kamayish tartibda chiqaring.*

*19.  $A(n)$  massiv elementlarini o'sish tartibida saralang va uning toq elementlarini alohida juft elementlarini alohida chop eting.*

*20.  $A(n)$  massiv elementlarini o'sish tartibida saralang va kamayish tartibda chop etish dasturini tuzing.*

#### **4.3 C++ DASTURLASH TILIDA MASSIV ELEMENTLARI ICHIDAN ENG KATTASI VA KICHIGINI ANIQLASH**

*Reja:*

- 1. Bir o'lchovli massiv elementlari ichidan eng kattasi va kichigini aniqlash;*
- 2. Ikki o'lchovli massiv elementlari ichidan eng kattasi va kichigini aniqlash.*

Massiv elementlari ichidan eng kattasi va eng kichigini aniqlash ko'pgina masalalar yechimini aniqlashga qo'l keladi. Masalan, biror bir ob'ekt haqida ma'lumotlar saqlanganda uning eng yuqorisi yoki eng quyi sohasidagi qiymatlarini aniqlash, yuk solingan idishlarning eng og'iri va eng yengilini aniqlash. Bularning barchasi massiv sifatida qaralsa, massiv elementlari ichidan eng kattasi yoki kichigini aniqlash masalasiga keladi.

Massiv elementlari ichidan eng katta va eng kichigini aniqlash uchun bir o'lchovli va ikki o'lchovli massivlar uchun katta farq qilmaydi, lekin alohida qilib tushuntiramiz.

##### **Bir o'lchovli massiv elementlari ichidan eng kattasi va kichigini aniqlash**

Bir o'lchovli massiv elementlari ichidan eng kattasi va kichigini aniqlash uchun ichma-ich sikldan foydalanib aniqlaymiz. Bir o'lchovli massiv elementlari ichidan eng kattasi va kichigini aniqlash uchun quyidagi algoritmni tavsiya etamiz.

*Bir o'lchovli massiv elementlari ichidan eng kattasi va kichigini aniqlash uchun massivning birinchi elementini eng katta(max) va eng kichik(min) sifatida*

qaraymiz. Tanlangan eng katta(max) va eng kichik(min) elementlar qolganlari bilan solishtiriladi, agar ulardan mos ravishda katta va kichik elementlar topilsa, ular eng katta(max) va eng kichik(min) element sifatida qabul qilinadi.

Yuqoridagi algoritm bo'yicha bir o'lchovli massiv elementlari ichidan eng kattasi va kichigini aniqlash uchun C++ dasturlash tilida quyidagi algoritmni tavsiya etamiz.

<pre> #include &lt;iostream.h&gt; using namespace std; int main() {   int a[20];     int n,t; cout&lt;&lt;"n=";     cin&gt;&gt;n;     for(int i=0;i&lt;n;i++)         cin&gt;&gt;a[i];     max=a[1];     for(int i=1;i&lt;n;i++)         if(max&lt;a[i]) max=a[i];     cout&lt;&lt;"max="&lt;&lt;max;     min=a[1];     for(int i=1;i&lt;n;i++)         if(min&gt;a[i]) min=a[i];     cout&lt;&lt;"min="&lt;&lt;min;     return 0; } </pre>	
Kiritish uchun ma'lumot n=5 3 5 1 0 6	Dastur natijasi max=6 min=0

Demak massiv elementlari ichidan eng kattasi va kichigini aniqlash uchun yuqorida keltirilgan algoritm bo'yicha aniqlanadi. Massivga keladigan boshqa masalalar uchun ham yuqoridagi algoritm asosida hisoblanadi.

### **Ikki o'lchovli massiv elementlari ichidan eng kattasi va kichigini aniqlash**

Ikki o'lchovli massiv elementlari ichidan eng kattasi va kichigini aniqlash uchun ichma-ich ikkita sikldan foydalanib aniqlaymiz. Ikki o'lchovli massiv

elementlari ichidan eng kattasi va kichigini aniqlash uchun xuddi bir o‘lchovli massiv elementlari ichidan eng kattasi va kichigini aniqlash algoritmidan foydalanamiz.

*Ikki o‘lchovli massiv elementlari ichidan eng kattasi va kichigini aniqlash uchun massivning birinchi elementini eng katta(max) va eng kichik(min) sifatida qaraymiz. Tanlangan eng katta(max) va eng kichik(min) elementlar qolganlari bilan solishtiriladi agar ulardan mos ravishda katta va kichik elementlar topilsa ular eng katta(max) va eng kichik(min) element sifatida qabul qilinadi.*

Yuqoridagi algoritm bo‘yicha ikki o‘lchovli massiv elementlari ichidan eng kattasi va kichigini aniqlash uchun C++ dasturlash tilida quyidagi algoritmni tavsiya etamiz.

```
#include <iostream.h>
using namespace std;
int main()
{   int a[20];
    int n,t; cout<<"n=";
    cin>>n;
    for(int i=0;i<n;i++)
    for(int j=0;j<n;j++)
        cin>>a[j][i];
    max=a[1][1];
    for(int i=1;i<n;i++)
    for(int j=0;j<n;j++)
        if(max<a[j][i]) max=a[j][i];
    cout<<"max="<<max;
    min=a[1];
    for(int i=1;i<n;i++)
    for(int j=0;j<n;j++)
        if(min>a[j][i]) min=a[j][i];
    cout<<"min="<<min;
    return 0;
}
```

Kiritish uchun ma'lumot

n=3

3 5 1

0 6 4

7 9 2

Dastur natijasi

max=9

min=0

Demak, massiv elementlari ichidan eng kattasi va kichigini aniqlash uchun yuqorida keltirilgan algoritm bo'yicha aniqlanadi. Ikki o'lchovli massivning har bir satri yoki ustuni uchun eng katta va eng kichik elementlarini aniqlash mumkin. Agar ikki o'lchovli massivning har bir satri yoki ustuni uchun eng katta va eng kichik elementlarini aniqlash uchun faqat sikllarni to'g'ri ifodalab yuqoridagi algoritm bo'yicha aniqlash mumkin.

### **Nazariy savollar**

*1 Bir o'lchovli massiv elementlari ichidan eng kattasi va kichigini aniqlash algoritmini tushuntirib bering?*

*2 Ikki o'lchovli massiv elementlari ichidan eng kattasi va kichigini aniqlash algoritmini tushuntirib bering?*

### **Mustaqil ishlash uchun topshiriqlar**

*1.  $N$  natural soni va  $n$  ta sonlar to'plami berilgan. Kiritilgan to'plamdagi eng katta va eng kichik sonni topuvchi dastur tuzilsin.*

*2.  $N$  natural soni va  $n$  ta to'g'ri burchakli to'rtburchak tomonlarining  $(a, b)$  to'plami berilgan. Kiritilgan to'plamdagi eng kichik yuzali to'rtburchakni topuvchi dastur tuzilsin.*

*3.  $N$  natural soni va  $n$  ta to'g'ri burchakli to'rtburchak tomonlarining  $(a, b)$  to'plami berilgan. Kiritilgan to'plamdagi eng katta perimetrli to'rtburchakni topuvchi dastur tuzilsin.*

*4.  $N$  natural soni va  $n$  ta sonlar to'plami berilgan. Kiritilgan to'plamdagi eng kichik element o'rnini aniqlovchi dastur tuzilsin.*

*5.  $N$  natural soni va  $n$  ta  $(m, v)$  sonlar juftligi to'plami berilgan ( $m$  – og'irlik,  $v$  - hajm). Kiritilgan to'plamdagi eng kata zichlikni aniqlovchi dastur tuzilsin. Zichlik – og'irlikni hajmga nisbatiga teng.*

*6.  $N$  natural soni va  $N$  ta butun sondan iborat to'plam berilgan. Birinchi uchragan eng kichik va oxirgi uchragan eng katta element tartib raqamini aniqlovchi dastur tuzilsin.*

*7.  $N$  natural soni va  $N$  ta butun sondan iborat to'plam berilgan. Birinchi uchragan eng katta va oxirgi uchragan eng kichik element tartib raqamini aniqlovchi dastur tuzilsin.*

*8.  $N$  natural soni va  $N$  ta butun sondan iborat to'plam berilgan. Birinchi va oxirgi uchragan eng kichik element tartib raqamini aniqlovchi dastur tuzilsin.*

*9.  $N$  natural soni va  $N$  ta butun sondan iborat to'plam berilgan. Birinchi va oxirgi uchragan eng katta element tartib raqamini aniqlovchi dastur tuzilsin.*

**10.**  $N$  natural soni va  $N$  ta butun sondan iborat to‘plam berilgan. Birinchi uchragan ekstremal element tartib raqamini aniqlovchi dastur tuzilsin. Ekstremal element deb katta yoki kichik elementga aytiladi.

**11.**  $N$  natural soni va  $N$  ta butun sondan iborat to‘plam berilgan. Oxirgi uchragan ekstremal element tartib raqamini aniqlovchi dastur tuzilsin. Ekstremal element deb katta yoki kichik elementga aytiladi.

**12.**  $N$  natural soni va  $N$  ta butun sondan iborat to‘plam berilgan. Eng kichik musbat sonni aniqlovchi dastur tuzilsin. Agar musbat son bo‘lmasa nol chiqarsin.

**13.**  $N$  natural soni va  $N$  ta butun sondan iborat to‘plam berilgan. Birinchi uchragan eng katta toq element tartib raqamini aniqlovchi dastur tuzilsin. Agar musbat son bo‘lmasa nol chiqarsin.

**14.**  $B$  soni va  $10$  ta butun sondan tashkil topgan musbat sonlar to‘plami berilgan. Shu to‘plamda  $B$  sonidan katta bo‘lgan, eng kichik elementni tartib raqamini chiqaruvchi dastur tuzilsin. Agar berilgan to‘plamda  $B$  sonidan katta son topilmasa, ikkita  $0$  chiqarilsin.

**15.**  $B, C$  sonlari va  $10$  ta butun sondan tashkil topgan to‘plam berilgan ( $0 < B < C$ ). Shu to‘plamda  $(B, C)$  oraliqdagi eng katta elementni tartib raqamini chiqaruvchi dastur tuzilsin. Agar berilgan to‘plamda  $(B, C)$  oraliqda son topilmasa, ikkita  $0$  chiqarilsin.

**16.**  $N$  natural soni va  $N$  ta butun sondan iborat to‘plam berilgan. Birinchi uchragan eng kichik elementgacha bo‘lgan elementlar sonini aniqlovchi dastur tuzilsin.

**17.**  $N$  natural soni va  $N$  ta butun sondan iborat to‘plam berilgan. Oxirgi uchragan eng katta elementdan keyin turgan elementlar sonini aniqlovchi dastur tuzilsin.

**18.**  $N$  natural soni va  $N$  ta butun sondan iborat to‘plam berilgan. Birinchi va oxirgi uchragan eng katta element orasida turgan elementlar sonini aniqlovchi dastur tuzilsin. Agar to‘plamda faqat bitta eng katta element bo‘lsa, nol chiqarsin.

**19.**  $N$  natural soni va  $N$  ta butun sondan iborat to‘plam berilgan. To‘plamdagi eng kichik elementlar sonini aniqlovchi dastur tuzilsin.

**20.**  $N$  natural soni va  $N$  ta butun sondan iborat to‘plam berilgan. To‘plamdagi ekstremal elementlar sonini aniqlovchi dastur tuzilsin. Ekstremal element deb eng katta yoki eng kichik elementga aytiladi.

**21.**  $N$  natural soni va  $N$  ta butun sondan iborat to‘plam berilgan ( $N > 2$ ). To‘plamning o‘rtacha qiymatini aniqlovchi dastur tuzilsin. o‘rtacha qiymatni hisoblashda eng katta va eng kichik qiymatlar hisobga olinmasin.



22.  $N$  natural soni va  $N$  ta butun sondan iborat to'plam berilgan ( $N > 2$ ). To'plamdagi eng kichik 2 ta qiymatni aniqlovchi dastur tuzilsin.

Masalan:  $N = 5$ ; 1 2 3 4 5

Natija: 1 2

23.  $N$  natural soni va  $N$  ta butun sondan iborat to'plam berilgan ( $N > 3$ ). To'plamdagi eng katta 3 ta qiymatni aniqlovchi dastur tuzilsin.

Masalan:  $N = 5$ ; 1 2 3 4 5

Natija: 5 4 3

24.  $N$  natural soni va  $N$  ta butun sondan iborat to'plam berilgan ( $N > 1$ ). Ikkita qo'shni son yig'indisining eng katta qiymatni aniqlovchi dastur tuzilsin.

25.  $N$  natural soni va  $N$  ta butun sondan iborat to'plam berilgan ( $N > 1$ ). Ko'paytmasi eng kichik bo'ladigan ikkita qo'shni element indekslarini aniqlovchi dastur tuzilsin.

#### 4.4 C++ DASTURLASH TILIDA FUNKSIYALARNI YARATISH VA ULARDAN FOYDALANISH

Reja:

1. Qism dasturlar;
2. Funksiya va ularni e'lon qilish;
3. Funksiya tanasini tasvirlash;
4. Funksiyani e'lon qilmasdan undan foydalanish;
5. Global va lokal o'zgaruvchilar;
6. Global o'zgaruvchilar qiymatini saqlab qolish.

Masalaning dasturi yetarlicha katta hajmga ega bo'lsa, uni bir butunligicha tasavvur qilib, uning mayda qismlari o'rtasidagi bog'lanishni sezish va ular o'rtasidagi bog'lanishni tashkil qilish jarayoni ko'p mehnat, aqliy zo'riqish va vaqt talab qiladi. Tajribali dasturchilar yetarlicha murakkab masalaning dasturlarini ham aqliy zo'riqishsiz tuzish texnologiyalari va usullarini ishlab chiqishgan.

Shunday usullardan biri dasturlashga tizimli (sistemali) yondashish usulidir. Dasturlashga tizimli yondashish quyidagicha ikki holdan boshlanadi.

**Birinchidan**, masalani mumkin bo'lgan mayda qismlarga, agar mumkin bo'lsa mayda qismlarni ham yana mayda qismlarga bo'lib ketaveramiz, toki o'sha qismlar hamma tushunadigan oddiy amallardan iborat bo'lib qolsin. Shundan so'ng o'sha qismlarning dasturlarini alohida-alohida tuzib, asosiy (bosh) dasturga birlashtiriladi va yaxlit masalani yechish dasturi paydo bo'ladi.

Masalan, tanlangan funksiyani tekshirish dasturini tuzish talab etilgan bo'lsin. Bunda quyidagi amallarni bajaruvchi dasturlar tuzib olinadi:

- funksiyaning aniqlanish sohasini topish;
- funksiyaning qiymatlar sohasini topish;
- funksiyaning nollarini aniqlash;
- funksiyaning OY o'qini kesish nuqtalarini aniqlash;
- funksiyaning juft-toqligini aniqlash;
- funksiyaning davrini aniqlash;
- funksiyaning musbat oraliqlarini aniqlash;
- funksiyaning manfiy oraliqlarini aniqlash;
- funksiyaning o'sish oraliqlarini aniqlash;
- funksiyaning kamayish oraliqlarini aniqlash;
- funksiyaning ekstremum nuqtalarini aniqlash;
- funksiyaning ekstremum qiymatlarini aniqlash;
- funksiyaning grafigini chizish;

amallarini bajaruvchi dasturlar tuzib olinadi. Agar zarurat bo'lsa, bu dasturlarning har biri yana mayda dasturlarga bo'linib, har birining dasturlari tuziladi va «Funksiyaning tekshirish» nomli asosiy dasturga birlashtiriladi.

**Ikkinchidan**, bitta masalaning dasturini tuzishda bitta masalaning dasturidan bir necha marta foydalanishga to'g'ri keladi. Masalan, binomial koeffitsientlarni aniqlashda, ya'ni,

$$C_n^m = \frac{n!}{m!(n-m)!}$$

ifodaning qiymatini hisoblashda faktorialni hisoblash dasturidan uch marta foydalanish kerak va har safar bu dasturni qayta-qayta tuzib yurmasdan uni bir marta tuzib, kerak bo'lganda uch marta murojaat tashkil qilish kerak. Natijada dasturning hajmi kichiklashadi, uni tushunish ham oson bo'ladi. Bunday vaziyatlarda o'sha mayda dasturlarni funksiya ko'rinishida yozib, kerak bo'lganda unga xohlagancha marta murojaat qilish yo'li bilan natijalarni olish mumkin.

Dasturlash tillari tarkibida bir necha marta bajarilish kerak bo'lgan jarayonlarni bir marta tasvirlab keyin shu dasturga murojat qilish imkoniyati mavjud. Dasturlash tillari tarkibida bir necha marta bajarilishi mumkin bo'lgan holatlarni qism dastur sifatida e'lon qilish va kerakli joyga shu qism dasturga murojat qilish mumkin.

*Ta'rif: Dasturlash tilida yaratilgan dastur tarkibida ma'lum bir vazifani bajaruvchi kichik dasturlar **qism dasturlar** deyiladi.*

Qism dasturlarning mohiyati berilgan masala tarkibi ma'lum bir vazifani bajarish kerak bo'ladi, lekin masala tarkibida ham kichik bir vazifani bajarish kerak bo'ladi shunday vazifalarni qism dastur yordamida hal etish mumkin. Bunday masalalar asosan matematik masalalar tarkibida ko'p bo'lishi mumkin,

yoki boshqa sohalarda ham uchrab turadi. Masalan biror bir tashkilotning ma'lumotlar bazasi berilganda uning tarkibidagi xodimlarning oylik maoshini hisoblash jarayonini qism dastur yordamida hisoblash maqsadga muvofiq. Agar tashkilot ma'lumotlar bazasi tarkibidagi xodimlarning oylik maoshini qism dastur yordamida hisoblanmasa, har bir xodim uchun oylik maoshini hisoblash jarayonini keltirish kerak, qism dasturdan foydalansa har bir xodim uchun qism dasturga murojat qilib qo'yiladi. Dastur tuzish vaqtida qism dasturlardan qachon foydalanamiz, agar siz hal etadigan masala yoki muammo tarkibida ma'lum bir jarayonlar ikki va undan ortiq sodir bo'lsa, o'sha jarayonni qism dastur sifatida e'lon qilish kerak va kerakli joyda qism dasturga murojat qilish kerak. Masalan, quyidagi masalaga e'tibor bering.

$$y = \frac{a^{1+2+\dots+n}}{(1+2+\dots+n)^x}$$

Bu masala tarkibiga e'tibor qaratsak, birdan n gacha bo'lgan sonlar yig'indisi bir necha marta bajarilyapti, bu masalani hal etish uchun tuziladigan dastur tarkibida birdan n gacha bo'lgan sonlar yig'indisini hisoblash jarayonini qism dastur qilib e'lon qilish kerak. Agar qism dastur sifatida e'lon qilinsa, masalani hal etishda qism dasturga ikki marta murojat qilish asosida berilgan masalani hal etish mumkin.

Yuqorida keltirilgan masalalarga o'xshash masalalarni hal etish usulini samarasini oshirish uchun qism dasturlardan foydalaniladi. Qism dasturlar C++ dasturlash tilida qisqacha qilib funksiyalar deb ataladi. Demak, funksiyalar ma'lum bir vazifani bajaruvchi dastur tarkibidagi qism dasturlar ekan.

Funksiyaning o'zi nima degan savol tug'ilishi tabiiy. Funksiya – bu biror aniq masalani yechishga mo'ljallangan, to'liq tugallangan, mustaqil dastur, yanaham aniqroq aytadigan bo'lsak, qism dastur yoki yordamchi dastur (podprogramma).

***Ta'rif:*** Dasturlash tilida tuzilgan dastur tarkibidagi ma'lum bir vazifalarni bajaruvchi qism dasturlar funksiyalar deyiladi.

Dastur ishlash jarayoni samarasi asosan ikki tur bo'yicha oshiriladi, ya'ni xotira hajmi va ishlash tezligi bo'yicha. Dasturning ishlash tezligini oshirish uchun, albatta, dastur tarkibi sodda va ixcham bo'lishi ta'lab etiladi. Dastur tarkibi sodda va ixcham bo'lishi uchun imkoniyat boricha qism dasturlardan foydalanish kerak bo'ladi. Funksiyalar C++ dasturlash tilida ma'lum bir vazifani bajaruvchi dastur tarkibidagi kichik dasturlar, demak, funksiyani hosil qilish uchun datur tarkibida ma'lum bir vazifalar bir necha marta bajarilish kerak bo'ladi. C++ dasturlash tilida har qanday funksiya hech bo'lmaganda bitta bosh funksiyani

**main()** funksiyasini o‘z tarkibida saqlaydi. Dastur tarkibida **main()** funksiyasidan tashqari boshqa funksiyalar ham bo‘lishi mumkin.

Dastur tarkibidagi funksiyaga uning nomi bilan murojat qilinadi, dastur bajarilish vaqtida funksiya nomi uchrasa, komplyator shu funksiyaning tanasiga murojat qilib natijani funksiya nomiga qaytaradi va dastur keyingi qadamlarni bajaradi. C++ dasturlash tilida funksiyalar ikki guruhga ajratiladi, ya’ni standart va standart bo‘lmagan funksiyalar. Standart funksiyalar C++ dasturlash tili tarkibida biror bir kutubxonada joylashtirilgan bo‘ladi. Standart funksiyalar haqida mazkur qo‘llanmaning II-bobida batafsil yoritilgan. Standart bo‘lmagan funksiyalar dastur tarkibida yaratiladi va unga nomi bilan murojat qilinadi. C++ da dasturlashning asosiy bloklaridan biri funksiyalardir. Funksiyalarning foydasi shundaki, katta masala bir necha kichik bo‘laklarga bo‘linib, har biriga alohida funksiya yozilganda, masala yechish algoritmi ancha soddalashadi. Bunda dasturchi yozgan funksiyalar C++ ning standart kutubxonasi va boshqa firmalar yozgan kutubxonalar ichidagi funksiyalar bilan birlashtiriladi. Bu esa ishni bir muncha osonlashtiradi. Ko‘p holda dasturda takroran bajariladigan amallarni funksiya sifatida yozish va kerakli joyda ushbu funksiyani chaqirish mumkin. Funksiyani programma tanasida ishlatish uchun u chaqiriladi, yani uning ismi yoziladi va unga kerakli argumentlar beriladi. () qavslar ushbu funksiya chaqirig'ini ifodalaydi. Masalan: `foo()`; `k = square(1)`; Demak, agar funksiya argumentlar olsa, ular () qavs ichida yoziladi. Argumentsiz funksiyadan keyin esa () qavslarning o‘zi qo‘yiladi.

### **Funksiyalarni e‘lon qilish**

C++ dasturlash tilida funksiyalar yaratish ulardan dasturning ixtiyoriy joyida foydalanish uchun, albatta, funksiyalarni dastur tanasida e‘lon qilish kerak. C++ dasturlash tilida funksiyalarni e‘lon qilishning bir necha usullari mavjud. C++ dasturlash tilida `main()` asosiy funksiya tarkibida boshqa funksiyalardan foydalanish uchun, albatta, funksiyalar `main()` asosiy funksiyadan oldin e‘lon qilinishi kerak. Funksiyalarni e‘lon qilish uning prototipi yoki signaturasi ham deyiladi.

*Ta‘rif: Dastur tarkibidagi funksiyalarni `main()` funksiyasidan oldin dasturlash tilida yozilishi **funksiyalarni e‘lon qilish** deyiladi.*

C++ dasturlash tilida funksiyalarni e‘lon qilishning umumiy ko‘rinishi quyidagicha bo‘ladi:

**tur <funksiya nomi>(tur <argument>);**

Funksiyani e‘lon qilish jarayonida birinchi funksiya natijasining, ya’ni funksiya qaytaradigan turi keltiriladi, keyin funksiya nomi qavs ichida funksiya

o'zgaruvchilari turi va o'zgaruvchilari keltiriladi. Funksiya qaytaradigan turi argument bajarilgandan so'ng hosil bo'ladigan natija turi bilan ustma-ust tushishi kerak aks holda kompyator xato keltiradi. Funksiya nomi funksiya bajaradigan vazifadan kelib chiqqan holda qo'yiladi, chunki dastur tuzuvchi mutaxassislar uchun umumiy holda tushunish oson bo'lishi kerak. Masalan, ketma-ketliklarni yig'indisini hisoblash funksiyasiga **sum()**, to'rtinchi darajali ildizni hisoblash uchun **sqrt4()**, sonning ekubini hisoblash uchun **ekub()**, sonning ekukini hisoblash uchun **ekuk()** va faktorialni hisoblash uchun **fak()** deb nomlash maqsadga muvofiq bo'ladi. Funksiya tarkibiga kiritilishi kerak bo'lgan o'zgaruvchi argumentlar, albatta, avval turi bilan qavs ichida vergul bilan ajratilib yozilishi kerak. Funksiya qaytaradigan turi, funksiya qabul qiladigan argumentlar turi va funksiya tarkibidagi buyruqlar asosida aniqlanadi. C++ dasturlash tilida *birdan n gacha bo'lgan butun sonlarning yig'indisini* hisoblovchi va ikkita butun sonning yig'indisini hisoblovchi funksiyalarni quyidagicha e'lon qilish kerak:

```
#include <iostream.h>
int sum(int n);
int yig(int a, int b);
using namespace std;
int main()
{
}
```

Birdan n gacha bo'lgan butun sonlarning yig'indisini hisoblash uchun **sum** nomi bilan **int sum(int n)** funksiya yaratiladi. Funksiyani natijasi butun tur bo'ladi, chunki butun turdagi sonlarning yig'indisi, albatta, butun bo'ladi. **sum(int n)** funksiyasining argument sifatida bitta n soni qabul qilinadi, chunki birdan n gacha sonlar yig'indisini hisoblash jarayoni faqat n ga bog'liq bo'ladi. Ikkita sonning yig'indisini hisoblash uchun **int yig(int a, int b)** funksiyasi yaratiladi. Funksiya qaytaradigan natija turi butun bo'ladi, chunki butun turdagi sonlarning yig'indisi, albatta, butun bo'ladi. Yig'indini hisoblash funksiya argumenti ikkita *a* va *b* sonlari qavs ichida turi bilan keltiriladi, chunki ikki sonning yig'indisi ikkita butun songa bog'liq bo'ladi. Har doim ham funksiya natijasining turi argument turi bilan bir xil bo'lmaydi. Ikki sonning nisbatini aniqlovchi funksiyani quyidagicha e'lon qilinadi.

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

```
double nisbat(int a, int b);
using namespace std;
int main()
{
}
```

Ikki sonning nisbatini aniqlash uchun `double nisbat(int a, int b)` funksiyasi yaratiladi, nisbat funksiyasi natijasining turi haqiqiy bo‘ladi, chunki ikkita butun sonning nisbati doimo butun bo‘lavermaydi.

### Funksiya tanasini tasvirlash

C++ dasturlash tilida funksiyalardan foydalanish uchun, albatta, funksiyalarni dastur tarkibida tasvirlash kerak. Funksiyalar dasturchi ishini juda yengillashtiradi. Funksiyalar yordamida programma modullashadi, qismlarga bo‘limadi. Bu esa keyinchalik dasturni rivojlantirishni osonlashtiradi. Dastur yozilish davrida hatolarni topishni yengillashtiradi. C++ dasturlash tilida funksiyalar tanasini tasvirlash `main()` funksiyasidan tashqarisida yozilishi, yani funksiya bajaradigan vazifalarni `{ }` qavs ichida buyruqlar ketma- ketligi shaklida yozilishi.

Funksiya tanasini tasvirlash jarayoni ikki qismdan iborat bo‘ladi, ya’ni funksiya sarlavhasi va funksiya tanasidan iborat bo‘ladi. Funksiya tanasini tasvirlash jarayonida funksiya sarlavhasidan keyin nuqtali vergul qo‘yilmaydi, `{` qavs ochilib buyruqlar yoziladi va natija **return** xizmatchi so‘zidan keyin probel bilan yoziladi keyin `}` qavs yopiladi. Funksiya tanasi tarkibi yozilishida xuddi boshqa dastur tuzilishi kabi unda ishlatiladigan o‘zgaruvchilar e‘lon qilinadi, buyruqlar nuqtali vergul yordamida ajratilib yoziladi. Funksiyalarni C++ dasturlash tilida tasvirlash jarayonining umumiy ko‘rinishi quyidagicha bo‘ladi:

```
<tur> <funksiya nomi>(<tur> <argument>)
{ <tur> <o‘zgaruvchilar>;
  <operatorlar>;
  return natija;
}
```

Funksiyalar tansini tasvirlashda funksiya qaytaradigan qiymat yoki ifoda `return` so‘zidan keyin probel bilan yozilishi kerak va nuqtali vergul qo‘yiladi.

*Misol:* C++ dasturlash tilida ikki sonning yig‘indisini hisoblash uchun funksiya yarating va unga murojat qilishni tasvirlang.

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

```

int yig(int a, int b);
using namespace std;
int main()
    { int x,y,z;
      cin>>x>>y;
      z=yig(x,y);
      cout<<z;
      return 0;
    }
int yig(int a, int b)
    { int t;
      t=a+b;
      return t;
    }

```

*Kiritishga ma'lumot*

2 3

*Dastur natijasi*

5

Yuqoridagi masalani hal etish uchun  $yig(a,b)$  funksiyasi yaratildi, funksiya tanasini main() funksiyasidan keyin tasvirlandi. Dastur bajarilish vaqtida kompyator dastur tarkibida  $yig(a,b)$  funksiyasini uchratganda bajarilish qadami  $yig(a,b)$  funksiya tanasiga o'tib natijani hisoblab qaytib keladi va bajarilish qadami buyruqlar ketma-ketligi bo'yicha bajariladi. Yuqoridagi masalani ikkinchi ko'rinishda ham bajarish mumkin.

```

#include <iostream.h>
int yig(int a, int b);
using namespace std;
int main()
    { int x,y,z;
      cin>>x>>y;
      z=yig(x,y);
      cout<<z;
      return 0;
    }
int yig(int a, int b)
    {
      return a+b;
    }

```

}	
<i>Kiritishga ma'lumot</i> 2 3	<i>Dastur natijasi</i> 5

Ikki sonning yi'gindisini hisoblash uchun keltirilgan dasturning ikkinchi ko'rinishi faqat  $yig(a,b)$  funksiyasining asosiy tanasida o'zgartirilgan. Return xizmatchi so'zidan keyin funksiya qaytaradigan qiymat natijasini ifodalovchi ifodani ham yozish mumkin.

**Misol.** Binomial koeffitsentni aniqlash dasturini tuzing. Ma'lumki, binomial koeffitsent

$$C_n^m = \frac{n!}{m!(n-m)!} \quad \text{formula bilan topiladi.}$$

```
#include <iostream.h>
using namespace std;
int fak(int n);
int main()
{
    int n,m,nf,mf,nm; float c;
    cin>>n>>m;
    nf=fak(n); mf=fak(m); nm=fak(n-m);
    c=nf/(mf*nm);
    cout<<"\nc="<<c<<"\n";
    system ("pause");
    return 0;
}
int fak(int n)
{ int f=1, i=1;
while(i<=n) {f=f*i; i++;}
return f;
}
```

Dasturning 3-satrida butun turli **fak** nomli funksiya **prototipi** e'lon qilindi va u bitta butun turli  $n$  parametrga ega.

**int main ()** funksiyasi tanasi tugagandan keyin fak nomli funksiya **14-satrdan 18-satrgacha** aniqlandi. Demak, funksiya biror funksiya tanasidan tashqarida aniqlanishi keak. Lekin funksiya tanasidan boshqa funksiya tanasiga murojaat qilish mumkin. **14-satr fak** nomli funksiya(yordamchi dastur)ning sarlavhasi deyiladi. U funksiya prototipi bilan bir xil. Faqat sarlavha oxirida, ya'ni



qavsdan keyin (;) belgisi qo'yilmagan. **15 va 18** satrlardagi figurali qavslar fak funksiyasining blok belgilari. Dasturning **15- satridan 18-satrigacha** bo'lgan qismi fak nomli funksiyaning tanasi deyiladi.

Bundan buyon tarkibida boshqa funksiyaga murojaat byurug'i bo'lgan dasturni asosiy dastur deb, murojaat bo'layotgan funksiyani esa yordamchi dastur(podprogramma) deb ham ataymiz. Demak, dasturning **4-satridan 13-satrigacha** bo'lgan qismi asosiy dastur, **14-satridan 18-satrigacha** bo'lgan qismi esa yordamchi dastur hisoblanadi.

Asosiy dasturning 8-satrida uchta chaqiriq byuruqlari (yoki funksiyaga murojaat byurulari) berilgan. Ulardan biri, masalan,  $nf=fak(n)$ ; byurug'ini tahlil qilaylik. C++ bu buyruqni uchratgach, dastur bajarilishini 15-satrdan davom ettiradi, ya'ni fak nomli funksiyaning tanasini bajaradi. Bunda 15 va 16-satrlarda  $n!$  hisoblanadi.  $n!$  ning natijaviy qiymatini butun turli f o'zgaruvchi oladi.

17-satrdagi  $return f$ ; byurug'i f o'zgaruvchining qiymatini murojaat bo'lgan joyga qaytarishni ta'minlaydi. Natijada nooshkor holda  $nf=f$  qiymat berish byurug'i bajarilib, keyingi  $mf=fak(m)$  byuruqni bajarishga o'tiladi.

Xulosa: demak, qism dastur tanasida bajarilgan hisob natijalarini return operatoridan keyin yozilgan o'zgaruvchi yoki ifoda yordamilda asosiy dasturga olib chiqiladi.

Funksiya prototipi, funksiya sarlavhasi va chaqiriq buyruqlaridagi parametrlar nomlarining har xil bo'lishi dastur bajarilishiga hech qanday ta'sir qilmaydi. Chunki funksiya prototipi va aniqlanishidagi argumentlar (bizning dasturimizda n parametr) formal (nomiga turgan) parametrlar bo'lib, ularning qiymati shu funksiyaga murojaat bo'lguncha aniq bo'lmaydi. Demak, yuqorida keltirilgan dastur quyidagi ko'rinishda ham bexato ishlaydi.

```
#include <iostream.h>
using namespace std; int fak(int s);
int main()
{ int n, m, nf, mf, nm; float c;
  cin>>n>>m; nf=fak(n); mf=fak(m); nm=fak(n-m); c=nf/(mf*nm);
  cout<<"\nc="<<c<<"\n"; system("pause"); return 0;
}
int fak(int y)
{int f=1, i=1; while(i<=y) {f=f*i; i++;}; return f;
}
```

Lekin zaruriyat bo'lmaganda bu usuldan foydalanmaslik tavsiya etiladi, chunki dasturni tushunish qiyinlashadi.

## Funksiyani e'lon qilmasdan undan foydalanish

C++ dasturlash tilida funksiyalardan foydalanishning bir qancha usullari mavjud. Bu usullardan biri C++ dasturlash tilida funksiyalarni e'lon qilmasdan funksiyalardan foydalanish usulidir. Agar funksiya prototipidan foydalanmasa, albatta, main() funksiyasi ichida shu funksiyaga murojatdan oldin funksiyaning e'lon qilish jarayonini keltirish kerak. Funksiya prototipidan foydalanmasdan funksiyaga murojat qilishni quyidagi masala orqali ko'rib o'tamiz.

**Misol:**  $n!$  faktorialni hisoblash jarayonini funksiya yordamida tasvirlash.

<pre>#include &lt;iostream.h&gt; using namespace std; int main()     { int x;       cin&gt;&gt;x;       int fak(int n);       cout&lt;&lt;fak(x);       return 0;     } int fak(int n)     { int p=1;       for(int i=1;i&lt;=n;i++)         p=p*i;       return p;     }</pre>	
<i>Kiritishga ma'lumot</i>	<i>Dastur natijasi</i>
5	120

Demak, funksiya prototipidan foydalanmasa ham, lekin main() funksiyasi tarkibiga funksiya nomini keltirish kerak ekan.

**Misol:** Sonning natural bo'luvchilar sonini aniqlash uchun bo'luvchi(a,b) funksiyasini yarating.

<pre>#include &lt;iostream.h&gt; using namespace std; int main()     { int x;       cin&gt;&gt;x;</pre>
---

<pre> int buluvchi(int x); cout&lt;&lt;buluvchi(x); return 0; } int buluvchi(int n) { int p; for(int i=1;i&lt;=n;i++) if(n%i==0) p++; return p; } </pre>	
<i>Kiritishga ma'lumot</i>	<i>Dastur natijasi</i>
6	4

Yuqoridagi masalalarga e'tibor qaratsak masala tarkibidagi funksiyalardan dastur tarkibida ixtiyoriy joyida ixtiyoriy marta foydalanish mumkin.

### Global va lokal o'zgaruvchilar

C++ dasturlash tilida tuziladigan dasturlar tarkibida bir nechta o'zgaruvchilardan foydalaniladi. C++ dasturlash tilida funksiyalar mavzusidan keyin o'zgaruvchilar ikki turga ajratiladi, ya'ni global va lokal o'zgaruvchilar.

*Ta'rif: Dastur tarkibining ixtiyoriy joyida foydalanish mumkin bo'lgan o'zgaruvchilar **global o'zgaruvchilar** deyiladi.*

Global o'zgaruvchilar dasturning ixtiyoriy qismida o'z qiymatini saqlaydi, hattoki, main() funksiyasining ichidagi o'zgaruvchilar ham global hisoblanmaydi, chunki main() funksiyasidan tashqarida main() funksiyasi ichidagi o'zgaruvchilar o'z qiymatini saqlamaydi. Global o'zgaruvchilar main() funksiyasidan oldin e'lon qilinadi.

C++ dasturlash tilidagi funksiyalar tarkibidagi o'zgaruvchilar global hisoblanmaydi. Global bo'lmagan o'zgaruvchilar faqatgini o'z funksiya blok ichiga tegishli bo'ladi.

*Ta'rif: C++ dasturlash tilidagi funksiyalar tarkibidagi o'zgaruvchilar **lokal o'zgaruvchilar** deyiladi.*

Dastur tarkibidagi funksiyalar tarkibidagi barcha o'zgaruvchilar lokal hisoblanadi, funksiya tarkibidagi o'zgaruvchilar faqatgina funksiyaning o'z bloki ichida o'rinli bo'ladi. Global va lokal o'zgaruvchilarni aniqlash uchun quyidagi keltirilgan sonning bo'luvchilar yig'indisini hisoblovchi dastur orqali aniqlaymiz.

```

#include <iostream.h>
int buluvchi(int x);
float m;
int x=16;
using namespace std;
int main()
    {
        cout<<buluvchi(x);
        return 0;
    }
int buluvchi(int n)
    { int p;
    for(int i=1;i<=n;i++)
        if(n%i==0) p++;
    return p;
    }

```

*Dastur natijasi*

5

Yuqoridagi dastur tarkibiga e'tibor qaratsak, main() funksiyasidan oldin e'lon qilingan  $m$  va  $k$  o'zgaruvchilar global hisoblanadi, main() va bo'luvchi(n) funksiyalari tarkibidagi o'zgaruvchilar lokal hisoblanadi. Global o'zgaruvchilarni dasturning ixtiyoriy joyida foydalanish imkoniyati mavjud, lokal o'zgaruvchilarni esa faqatgina o'z funksiya bloki ichida foydalanish mumkin. Global va lokal o'zgaruvchilar haqida aytilganda global va lokal o'zgaruvchilarni **yashash davri** haqida tushuncha beriladi. Global o'zgaruvchilarni yashash davri dastur ishga tushirilgandan toki tugaguncha hisoblanadi. Lokal o'zgaruvchilarni yashash davri esa faqatgina o'zgaruvchi keltirilgan funksiya ishga tushirilgan vaqti hisoblanadi. Agar dasturdagi global o'zgaruvchilar funksiya tarkibida alohida boshlang'ich qiymati berilsa, uning global qiymati unutiladi va funksiya tarkibidagi qiymat hisobga olinadi. Global va lokal o'zgaruvchilarni yashash davrini quyidagi masala orqali aniqlaymiz:

```

#include <iostream.h>
int buluvchi(int x);

```

```

int a=20, b=10;
using namespace std;
int main()
    {
        cout<<yig(a,b);
        return 0;
    }
int yig(int a, int b)
    { a=4; b=5;
    return a+b;
    }

```

*Dastur natijasi*

9

Dastur tarkibida a=20, b=10 global sifatida aniqlangan edi, lekin yig(a,b) funksiyasini tarkibida a=4, b=5 bo‘lganligi uchun a va b larning boshlang‘ich qiymatlari unitiladi va natija 9 ga teng bo‘ladi. Global o‘zgaruvchilarning qiymatlarini saqlab qolish uchun a o‘zgaruvchini boshqacha qilib e‘lon qilish kerak. Global o‘zgaruvchilar qiymatlarini saqlab qolish uchun quyidagi rejaga asosan aniqlanadi.

### **Global o‘zgaruvchilar qiymatini saqlab qolish**

C++ dasturlash tilida global o‘zgaruvchilarni lokal sifatida ishlatilsa global o‘zgaruvchining boshlang‘ich qiymatini saqlab qolish uchun **a** o‘zgaruvchini **::a** sifatida ishlatish kerak. Global o‘zgaruvchilarni lokal sifatida ishlatilsa, global o‘zgaruvchining boshlang‘ich qiymatini saqlab qolishni quyidagi dastur orqali aniqlaymiz.

```

#include <iostream.h>
int buluvchi(int x);
int a=20, b=10;
using namespace std;
int main()
    {
        cout<<yig(a,b);
        return 0;
    }
int yig(int a, int b)

```

```
{ a=4; b=5;
  return a+::b;
}
```

*Dastur natijasi*

14

Dastur tarkibida  $a=20$ ,  $b=10$  global sifatida aniqlangan edi, lekin  $yig(a,b)$  funksiyasini tarkibida  $a=4$ ,  $b=5$  lokal qiymatlar berildi. Dastur natijasi 14 ga teng, chunki funksiya tarkibidagi  $a$  ni qiymati lokal sifatida qabul qilinib,  $b$  ni qiymati esa global sifatida qoldi, chunki  $b$  o'zgaruvchini  $::b$  sifatida keltirildi.

### Funksiya qiymatini qaytarish haqida

Barcha dasturlarda **int main()** asosiy funksiyasidan foydalandik va bu funksiyadan xohlagancha boshqa funksiyalarga murojaat tashkil qilish mumkin. **void** turida bo'lmagan har qanday funksiya **return** so'zi bilan yakunlanadi. Jumladan, **main()** funksiyasi ham shu operator bilan yakunlanadi. **return** operatoridan foydalanishda quyidagilarga e'tibor berish kerak:

a) faqat **void** turli funksiyalar parametrsiz **return** operatori bilan yakunlanishi mumkin. Chunki qiymat chaqiriq byurug'iga qaytarilmaydi. Shuning uchun bu turli funksiyalar **return operatorisiz** ham yakunlanishi, ya'ni blok belgisi bilan yakunlanishi mumkin;

b) **void** turidan boshqa turli barcha funksiyalar **return** operatori bilan yakunlanadi, ya'ni **return parametr** ko'rinishida bo'lishi kerak. Agar bu operatoridan parametrsiz foydalanilsa, kompilyator qaytariladigan qiymat yo'q deb xabar beradi;

c) **void** turidan boshqa turli funksiya **returnsiz** tugasa ham, blok belgisi (**{}**) ni uchratgach, kompilyator funksiya tanasini bajarish tugaganligini tushunadi va xatolik haqida xabar bermaydi. Ammo funksiya tanasidagi hisob natijalari qaytarilmaganligi tufayli, chaqiriq byurug'i qandaydir o'zgaruvchiga qiymat sifatida berilgan bo'lsa, o'sha o'zgaruvchi qiymat sifatida tasodifiy sonni oladi va tabiiyki, keyingi buyruqlarda mantiqiy xatoliklar sodir bo'ladi;

d) **return** so'zidan keyin parametr sifatida biror o'zgaruvchi nomidan, masalan, **return p** kabi foydalanilgan bo'lib,  $p$  e'lon qilingan, ammo qiymati aniqlanmagan bo'lsa ham qiymat sifatida tasodifiy son qaytariladi;

e) chaqiriq byurug'idan foydalanilgandan keyin qandaydir qiymat qaytariladi. Lekin o'sha qiymatni zarurat bo'lmasa biror o'zgaruvchiga berib, chaqiriq

byurug'idan keyingi buyruqlarda foydalanish shart emas. Bunday holda funksiya tanasidagi amallar natijasi o'z-o'zidan yo'qolib ketadi;

f) funksiya tanasidagi hisob natijalari qanday bo'lishidan qat'iy nazar **return** da ko'rsatilgan son yoki parametrning qiymati qaytariladi. Masalan, funksiya **return 20** kabi yakunlangan bo'lsa, funksiya tanasidagi hisob natijalari turi va qiymatidan qat'iy nazar, chaqiriq byurug'i natijasi 20 ga teng bo'ladi;

d) yuqorida funksiya turi qanday bo'lsa unga qaytariladigan qiymat turi ham shunday bo'lmasa, xatolik sodir bo'lishi haqida aytilgan edi. Bu xatolik ham mantiqiy xatolik bo'lishi mumkin. Ya'ni kompilyator xato haqida xabar bermasligi mumkin. Masalan, funksiya turi **int** bo'lsin. Lekin unga haqiqiy qiymat qaytarilsa, qaytarilgan qiymatning kasr qismi tashlab yuboriladi. Xatto, butun turli funksiyaga **return true** kabi **bool** turli qiymat qaytarilsa, bu qiymat 1 deb, **false** qiymat qaytarilsa, u nol deb qabul qilinadi;

h) **return** operatori orqali faqat massivning qiymatini qaytarib bo'lmaydi. Lekin massiv elementi qiymatini qaytarish mumkin;

i) **main ()** bosh funksiyasida qiymat qaytarish uchun **return 0**; byurug'idan foydalaniladi. Bunda nol butun son bo'lganligi uchun foydalanilgan. Ammo nol o'rnida ixtiyoriy manfiy va musbat, butun va haqiqiy sonlardan, qiymati aniq bo'lgan arifmetik va mantiqiy ifodalardan ham foydalanish mumkin. Shuningdek, return operatori bo'lmasa ham dastur ijro etiladi. Ya'ni bosh funksiyaning blok belgisini uchratgach C++ dastur yakunlanganligini tushunadi.

Qiymat qaytarish bo'yicha aytilgan fikrlarni quyidagi dastur misolida tushuntiramiz.

```
# include <iostream.h> using namespace std;
int fun1 (int x, int y);
int main()
{int i,j,k,m,r; i=5; j=2;
m=fun1(i,j); k=10+m;
cout<<"\n k="<<k<<"\n";
system ("pause");
return -56;
}
int fun1 (int x,int y)
{int q1,p; q1=x+y;
cout<<"\n q1="<<q1<<"\n";
return q1; system("pause"); }
```

Dastur natijasi

q1=7

k=14 kabi bo'ladi

j) agar dasturning 13-satrdagi **return q1** byurug'i olib tashlansa, **m** ning qiymati tasodifiy tanlanadi va dastur natijasi

q1=7

k=4469706

kabi bo'ladi. Shuningdek, 8-satrdagi **return -56** byurug'i ham olib tashlansa, natija o'zgar olmaydi. Ammo 13-satrdagi **return q1** byurug'ini qoldirib, 8-satrdagi **return -56** byurug'ini olib tashlasak ham, dastur to'g'ri bajariladi va dastur natijasi

q1=7

k=17

kabi bo'ladi. Chunki bosh funksiyadan **return** byurug'i olib tashlangan bo'lsa ham C ++ blok belgisi (}) ni uchratgach, dasturni ijro etish tugaganligini tushunadi.

k) agar 13-satrn **return p** kabi o'zgartirsak, yana qaytariladigan qiymatni tasodifiy tanlaydi va dastur natijasi

q1=7

k=4246714

kabi bo'ladi. Shuningdek, 13-satrdagi buyruqni o'zgarishsiz qoldirib, 8-satrdagi buyruqni **return r** kabi o'zgartirsak, dastur to'g'ri bajariladi. **main()** funksiyasiga qaytariladigan qiymat uning tarkibidagi buyruqlarga ta'sir qilmaydi, chunki ular ijro etilgan bo'ladi.

l) 5-satrdagi buyruqlar **m=fin1(i,j); k=i-j+10;** kabi o'zgartirilsa, dastur natijasi

q1=7

k=13

kabi bo'ladi. Ya'ni qaytarilgan qiymat biror o'zgaruvchiga qiymat sifatida berilmaydi va undan umuman foydalanilmaydi. Natijada qaytarilgan qiymat unutiladi.

m) 13-satrdagi buyruq **return-19** kabi o'zgartirilsa, dastur natijasi

q1=7

k=-9

kabi bo'ladi. Chunki **fun1()** funksiyasi tanasida hisoblar natijasi qanday bo'lishidan qat'iy nazar chaqiriq byurug'iga – 19 qaytariladi.



n) dasturning 4-satri **float i,j,k,m,r; i=5.54, j=2.5412;** kabi o'zgartirilsa, dastur natijasi o'zgarmaydi, ya'ni

q1=7

k=17

kabi bo'ladi. Dasturning 13-satri **return true** kabi o'zgartirilsa dastur natijasi

q1=7

k=11

kabi bo'ladi.

Umuman, **return** operatori parametri qandaydir son bo'lmasdan qiymati aniq bo'lgan sonli yoki harfiy ifoda, hatto, mantiqiy ifoda ham bo'lishi mumkin. Masalan, agar **x** va **y** larning qiymatlari oldindan aniqlangan bo'lsa, quyidagicha buyruqlarga ruxsat etiladi:

**return (2\*x-3)\*y;**

**return (37/7+8\*3)/2;**

**return x+2>=y-5;**

**return fun2(4,5)**

Masalan, dasturning 13-satrida **return 3\*x<=y-1** kabi o'zgarish qilinsa, natija

q1=7

k=10

kabi bo'ladi.

Agar dasturning 2-satriga **int fun2(int x, int y);** yozuvini qo'shib, 5-satrini **m=fun2(i,j); k=10+m** kabi o'zgartirib, 14-18 satrlariga quyidagicha buyruqlar qo'shilsa,

```
int fun2 (int x, int+y)
```

```
{ int q1,p; q1=x-y;
```

```
cout<<"\n q1="<<q1<<"\n";
```

```
return fun1 (4,7);
```

```
}
```

dastur natijasi

q1=3

q1=11

k=21

kabi bo'ladi.

### Nazariy savollar

1 Qism dastur deganda nimani tushunasiz?

2 main() funksiyasi va uning vazifasi?

3 Funksiyalarni e'lon qilish usullarini ayting?

- 4 *Funksiyalarni e'lon qilishning umumiy ko'rinishi?*
- 5 *Funksiya tanasini tasvirlash?*
- 6 *Funksiyani e'lon qilmasdan undan foydalanish usullari?*
- 7 *Global va lokal o'zgaruvchilar deganda nimani tushunasiz?*
- 8 *Global o'zgaruvchilar qiymatini qanday usul bilan saqlab qolish mumkin?*
- 9 *Dasturlashda funksiya tushunchasining kiritilishi qanday imkoniyatlar yaratadi?*
- 10 *Dasturini tuzishda funksiya tushunchasidan foydalanish katta yengilliklar paydo qiladigan bir necha misollar keltiring?*
- 11 *Funksiyaga murojaat qilish qanday amallarning (buyruqlarning) bajarilishini ta'minlaydi?*
- 12 *Funksiya tanasidagi barcha buyruqlar bajarilgandan keyin qaysi buyruqlar bajariladi?*
- 13 *C++ dasturlash tilining asosiy funksiyasi qaysi?*
- 14 *Funksiyani chaqiriq buyrug`i dasturning qayerida va qanday yoziladi?*
- 15 *Standart va nostandart funksiyalar qanday funksiyalar?*
- 16 *Funksiya prototipi, signaturasi tushunchalarini tushntiring.*
- 17 *Funksiya qayerda va qanday aniqlanadi?*
- 18 *Funksiya prototipi bilan uni aniqlashning qanday farqi bor?*
- 19 *Funksiya portotipining umumiy ko'rinishini yozing.*
- 20 *Chiziqli tenglamani yechishga mo'ljallangan funksiyaning signaturasini yozing.*
- 21 *Funksiyaga qaytariladigan qiymat deyilganda qanday qiymat tushuniladi va uning turi qanday bo'lishi kerak?*
- 22 *Funksiya turi bilan argumentlari turida qanday bog'lanish bo'lishi kerak?*
- 23 *Funksiyaning formal parametrlari qanday parametrlar? Nima uchun?*
- 24 *Funksiyaga qiymat qaysi operator orqali, qanday qaytariladi?*
- 25 *return k; buyrug`i nimani bildiradi?*
- 26 *Funksiya prototipini, aniqlanishi va chaqiriq buyrug`idagi argumantlar har xil nomlar bilan nomlanishi mumkinmi? Nima uchun?*
- 27 *Funksiya prototipi yozmasdan ham funksiya yaratish mumkinmi? Agar mumkin bo'lsa u qanday tashkil qilinadi?*
- 28 *Qanday o'zgaruvchilardan funksiya tashqarisida foydalanib bo'lmaydi?*
- 29 *O'zgaruvchining amal qilishi sohasi qaysi soha?*
- 30 *Global o'zgaruvchilar qayerda e'lon qilinadi?*
- 31 *main() funksiyasida aniqlangan o'zgaruvchilar mahalliy o'zgaruvchilarmi yoki umumiy o'zgaruvchilarmi?*
- 32 *Global va lokal o'zgaruvchilar bir xil nomlangan bo'lsin. Qanday yo'l bilan funksiya ichida global o'zgaruvchining qiymatidan foydalanish mumkin?*

33 Bitta funksiyadan ko'pi bilan nechta funksiyaga murojaat tashkil qilish mumkin?

34 Funksiyaning o'zidan o'ziga murojaat qilish mumkinmi va bunday murojaat qanday nomlanadi?

35 Har qanday funksiya tanasi qanday yakunlanadi?

36 *a* e'lon qilingan, ammo qiymati aniqlanmagan o'zgaruvchi bo'lib, `return a;` buyrug'i berilsa qanday xatolik ro'y beradi?

37 Bosh funksiyadan boshqa biror funksiyada `return s;` buyrug'i o'rniga `return 30;` buyrug'idan foydalanilsa qanday xatolik sodir bo'ladi va nima uchun?

38 Bosh funksiyaga qiymat qaytarishda `return 0;` buyrug'idagi nol raqami o'rnida yana qanday kattaliklardan foydalanish mumkin?

### Mustaqil ishlash uchun topshiriqlar

1. Ixtiyoriy sonning 3-darajasini hisoblovchi *PowerA3* nomli funksiya hosil qiling. *PowerA3* funksiyasi orqali *A*, *B*, *C* haqiqiy sonlarining va *D*, *E* butun sonlarining 3-darajasini hisoblovchi dastur tuzilsin.

2. Ixtiyoriy sonning 2, 3, 4 – darajasini hisoblovchi *PowerA234* nomli funksiya hosil qiling. *PowerA234* funksiyasi orqali *A*, *B*, *C* haqiqiy sonlarining 2, 3, 4 – darajasini hisoblovchi dastur tuzilsin. Funksiya prototipi quyidagicha bo'lishi mumkin:

`void PowerA234(float , float*, float*, float*);`

3. 2 ta sonning o'rta arifmetigi va geometrigini hisoblovchi *MEAN* nomli funksiya hosil qiling. *MEAN* funksiyasi orqali *A*, *B*, *C*, *D* sonlaridan (*A*,*B*), (*A*,*C*), (*A*,*D*) juftliklarining o'rta arifmetigi va geometrigini hisoblovchi dastur tuzilsin. Funksiya prototipi quyidagicha bo'lishi mumkin:

`void MEAN(float , float , float*, float*);`

4. Teng tomonli uchburchakning yuzasi va perimetrini hisoblovchi *Triangle* nomli funksiya hosil qiling. *Triangle* funksiyasi orqali uchta teng tomonli uchburchakning perimetri va yuzini hisoblovchi dastur tuzilsin. Funksiya prototipi quyidagicha bo'lishi mumkin:

`void Triangle(float , float*, float*);`

5. To'g'ri to'rtburchakning yuzini va perimetrini uning qarama – qarshi uchlari koordinatasi orqali hisoblovchi *RectPS* nomli funksiya hosil qiling. (*x1*, *y1*, *x2*, *y2*) to'g'ri to'rtburchakning qarama – qarshi uchlari *RectPS* funksiya orqali 2 ta to'rtburchak yuzi va perimetrini hisoblang. To'rtburchak tomonlari koordinatalar o'qiga parallel. Funksiya prototipi quyidagicha bo'lishi mumkin:

`void RectPS(int , int , int*, int*);`

6. Natural sonning raqamlari soni va raqamlari yig'indisini hisoblovchi *DigitCountSum* nomli funksiya hosil qiling. Bu funksiya orqali *a*, *b*, *c* sonlarining raqamlari soni va yig'indisini hisoblovchi dastur tuzilsin.

*void DigitCountSum (int, int\*, int\*);*

7. Butun musbat sonning raqamlarini teskari tartibda chiqaruvchi *InvertDigit* nomli funksiya hosil qiling. Bu funksiya orqali *a*, *b*, *c* sonlarining raqamlari teskari tartibda chiqaruvchi dastur tuzilsin.

*int InvertDigit (int);*

8. Kiritilgan *K* butun musbat sonning o'ng tarafiga (oxiriga) *R* raqamini ( $1 \leq R \leq 9$ ) qo'shuvchi *AddRightDigit* nomli funksiya hosil qiling. Funksiya prototipi quyidagicha bo'lishi mumkin:

*int AddRightDigit (int son, int raqam);*

9. Kiritilgan *K* butun musbat sonning chap tarafiga (boshiga) *R* raqamini ( $1 \leq R \leq 9$ ) qo'shuvchi *AddLeftDigit* nomli funksiya hosil qiling. Funksiya prototipi quyidagicha bo'lishi mumkin:

*int AddLeftDigit (int\*son, int raqam);*

10. Ikkita sonning qiymatini almashtiruvchi *Swap* nomli funksiya hosil qiling. *Swap* funksiya orqali *A*, *B*, *C*, *D* sonlaridan (*A*, *B*), (*D*, *C*) juftliklarining qiymatlarini almashtiruvchi dastur tuzilsin.

Funksiya prototipi quyidagicha bo'lishi mumkin:

*void Swap (int\*, int\*);*

11. *X* va *Y* sonlaridan kichigini *X* ga va kattasini *Y* ga yozuvchi *Minmax(X,Y)* funksiyasini hosil qiling. *Minmax* funksiyasini 4 marta chaqirish orqali *a*, *b*, *c*, *d* butun sonlaridan kattasini va kichigini aniqlovchi dastur tuzilsin.

12. *A*, *B*, *C* sonlarini o'sish tartibida joylashtiruvchi *SortInc3(A, B, C)* funksiyasini hosil qiling. Ya'ni *A*, *B*, *C* sonlari qiymatlarini shunday almashtiringki, natijada *A* ning qiymati eng kichik va *C* ning qiymati eng katta bo'lsin. Bu funksiya orqali (*A1*, *B1*, *C1*) va (*A2*, *B2*, *C2*) sonlarini tartiblang.

13. *A*, *B*, *C* sonlarini kamayish tartibida joylashtiruvchi *SortDec3(A, B, C)* funksiyasini hosil qiling. Ya'ni *A*, *B*, *C* sonlari qiymatlarini shunday almashtiringki, natijada *A* ning qiymati eng katta va *C* ning qiymati eng kichik bo'lsin. Bu funksiya orqali (*A1*, *B1*, *C1*) va (*A2*, *B2*, *C2*) sonlarini tartiblang.

14. o'ngga siklik siljishni amalga oshiruvchi *ShiftRight3(A, B, C)* funksiyasini hosil qiling. Ya'ni *A* ning qiymati *B* ga, *B* ning qiymati *C* ga, *C* ning qiymati *A* ga o'tib qolsin. Bu funksiya orqali (*A1*, *B1*, *C1*) va (*A2*, *B2*, *C2*) sonlarini siljiting.

15. Chapga siklik siljishni amalga oshiruvchi  $\text{ShiftLeft3}(A, B, C)$  funksiyasini hosil qiling. Ya'ni  $C$  ning qiymati  $B$  ga,  $B$  ning qiymati  $A$  ga,  $A$  ning qiymati  $C$  ga o'tib qolsin. Bu funksiya orqali  $(A1, B1, C1)$  va  $(A2, B2, C2)$  sonlarini siljiting.

16. Haqiqiy sonning ishorasini aniqlovchi ishora nomli funksiya hosil qiling. Funksiya argumenti noldan kichik bo'lsa  $-1$ ; noldan katta bo'lsa  $1$ ; nolga teng bo'lsa  $0$  qiymat qaytarsin Haqiqiy  $a$  va  $b$  sonlari uchun  $\text{ishora}(a) + \text{ishora}(b)$  ifodasi hisoblansin.

17. Kvadrat tenglamaning ildizlar sonini aniqlovchi funksiya hosil qiling.  $A * x^2 + B * x + C = 0$  ko'rinishidagi tenglama kvadrat tenglama deyiladi. ( $A$  noldan farqli son)

18. Doiraning yuzini hisoblovchi funksiya hosil qiling. Bu funksiya yordamida 3 ta doira yuzini hisoblang. Doiraning yuzi  $S = \pi R^2$  orqali hisoblanadi.  $\text{Pi} = 3.1415$  ni o'zgarmas deb qabul qiling.

19. Markazi bir nuqtada bo'lgan,  $R1$  va  $R2$  radiusga ega 2 ta aylananing ustma–ust tushmaydigan (kesishmaydigan) qismining yuzasini topuvchi  $\text{RingS}$  nomli funksiya hosil qiling. Doiraning yuzini hisoblash formulasidan foydalaning,  $S = \pi R^2$ .  $\text{Pi} = 3.1415$  ni o'zgarmas deb qabul qiling.

20. To'g'ri burchakli uchburchakning katetlari  $A$  va  $B$  berilganda, uning perimetrini hisoblovchi  $\text{TriangleP}$  nomli funksiya hosil qiling.

21.  $A$  va  $B$  sonlari orasidagi sonlar yig'indisini hisoblovchi  $\text{SumRangle}(A, B)$  nomli funksiya hosil qiling. Agar  $A > B$  bo'lsa, funksiya  $0$  qiymat qaytaradi. Bu funksiya orqali  $A$  dan  $B$  gacha va  $B$  dan  $C$  gacha bo'lgan sonlar yig'indisini hisoblang.  $A, B, C$  butun sonlar.

22. Arifmetik amallarni bajaruvchi  $\text{Calc}(A, B, Op)$  funksiyasini hosil qiling.  $A$  va  $B$  haqiqiy sonlar.  $Op$  o'zgaruvchisi orqali bajariladigan arifmetik amal aniqlanadi. 1-ayirish, 2-ko'paytirish, 3-bo'lish, boshqalari qo'shish. Shu funksiya orqali  $A$  va  $B$  sonlari uchun  $N1, N2, N3, N4$  amallari bajarilsin. ( $N1-N4$  butun sonlar)

23.  $X$  va  $Y$  butun sonlari berilgan ( $X$  va  $Y$  noldan farqli).  $(X, Y)$  nuqta qaysi chorakda ekanini aniqlovchi  $\text{Quarter}$  nomli funksiya hosil qiling. Bu funksiya orqali 4 ta nuqtaning choragini aniqlang.

24. Butun sonning juft – toqligini aniqlovchi  $\text{Even}(K)$  funksiyasini hosil qiling. Funksiya  $K$  juft son bo'lsa –  $\text{true}$ , aks xolda  $\text{false}$  qiymat qaytarsin. Bu funksiya orqali 3 ta sonning juft yoki toqligi aniqlansin.

25. *IsSquare(K)* mantiqiy funksiyasini hosil qiling. ( $K > 0$ ). Agar  $K$  biror butun sonning kvadrati bo'lsa – true, aks xolda false qiymat qaytarsin. Shu funksiya orqali 3 ta sonni tekshiring.

26. *IsPower5(K)* mantiqiy funksiyasini hosil qiling. ( $K > 0$ ). Agar  $K$  soni 5 ning biror darajasi bo'lsa – true, aks xolda false qiymat qaytarsin. Shu funksiya orqali 5 ta sondan nechtasi 5 ning darajasi ekanini aniqlovchi dastur tuzilsin.

27. *IsPowerN(K)* mantiqiy funksiyasini hosil qiling. ( $K > 0$ ). Agar  $K$  soni  $N$  ning biror darajasi bo'lsa – true, aks xolda false qiymat qaytarsin. Shu funksiya orqali 5 ta sondan nechtasi  $N$  ning darajasi ekanini aniqlovchi dastur tuzilsin.

28. *IsPrime(N)* mantiqiy funksiyasini hosil qiling. ( $N > 0$ ). Agar  $N$  soni tub bo'lsa – true, aks xolda false qiymat qaytarsin. Shu funksiya orqali kiritilgan  $k$  ta sondan nechtasi tub ekanini aniqlovchi dastur tuzilsin.

29. Butun qiymat qaytaruvchi *DigitCount(K)* funksiyasini hosil qiling. ( $K > 0$ ). Funksiya  $K$  ning raqamlari sonini qaytarsin. Shu funksiya orqali 5 ta sonning raqamlari soni aniqlansin.

30. Butun qiymat qaytaruvchi *DigitN(K, N)* funksiyasini hosil qiling. ( $K > 0$ ). Funksiya  $K$  sonining  $N$  – raqamini qaytarsin. Agar  $K$  soni raqamlari  $N$  dan kichik bo'lsa, minus bir qaytarsin. Shu funksiya orqali  $K1, K2, K3$  sonlarining  $N$  – raqami aniqlansin.

#### **4.5 C++ DASTURLASH TILIDA BIR NECHTA QIYMAT QAYTARUVCHI QISM DASTURLARNI YARATISH VA ULARDAN FOYDALANISH**

*Reja:*

1. *Prosedura va ularni e'lon qilish;*
2. *Prosedura asosiy tanasini tasvirlash va ulardan foydalanish;*

C++ dasturlash tilida funksiyalar dastur tarkibidagi kichik dasturlar hisoblanadi, ular dastur bajarilish natijasida bitta qiymat qaytaradi. Masalan, sonning faktorialini, sonlarning ekubini, sonlarning ekukini va hakoza shunga o'xshash natijalarni qaytaradi. Lekin dastur tarkibida ikki va undan ortiq natija qaytaradigan kichik muommolar ham mavjud. Masalan, kvadrat funksiyaning ildizlarini aniqlash, unda dastur ko'pi bilan ikkita qiymat qaytarish kerak, massivlarni o'sish yoki kamayish tartibida tartiblash va hakoza shunga o'xshash masalalar ko'p o'chraydi. Bu turdagi masalalarni yechish uchun oddiy funksiyalardan foydalanish maqsadga muvofiq bo'lmaydi. C++ dasturlash tilida ikki va undan ortiq qiymat qaytaradigan funksiyalarni qisqacha qilib prosedura deb

nomlaymiz. Yuklangan funksiyalar chaqirilganda, qaysi funksiyani chaqirish kirish parametrlarining soniga, ularning tipiga va navbatiga bog'liqdir. Yani ism yuklanishida funksiyaning imzosi rol o'ynadi. Agar kirish parametrlari va ismlari ayni funksiyalarning farqi faqat ularning qaytish qiymatlarida bo'lsa, bu yuklanish bo'lmaydi, kompilyator buni xato deb e'lon qiladi. Funksiya yuklanishi asosan ayni ishni yoki amalni farqli usul bilan farqli ma'lumot tiplari ustida bajarish uchun qo'llaniladi. Masalan bir fazoviy jismning hajmini hisoblash kerak bo'lsin. Har bir jismning hajmi farqli formula yordamida, yani farqli usulda topiladi, bir jismda radius tushunchasi bor bo'lsa, boshqasida asos yoki tomon tushunchasi bor bo'ladi, bu esa farqli ma'lumot tiplariga kiradi. Lekin amal ayni hajmni hisoblash. Demak, biz funksiya yuklanishi mexanizmini qo'llasak bo'ladi. Bir hil amalni bajaruvchi funksiyalarni ayni nom bilan atashimiz esa, dasturni o'qib tushunishni osonlashtiradi. Kompilayator biz bergan funksiya imzosidan (imzoga funksiya ismi va kirish parametrlari kiradi, funksiyaning qaytish qiymati esa imzoga kirmaydi) yagona ism tuzadi, dastur ijrosi davruda esa funksiya chaqirig'idagi argumentlarga qarab, kerakli funksiyani chaqiradi. Demak, funksiyani chaqirish uning nomiga bog'liq ekan. Proseduralar esa, albatta, uning imzosida prosedura nomi kirish va chiqish parametrlari, albatta, keltirilishi kerak, chunki prosedura tarkibida bir nechta qaytariladigan qiymatlar, albatta, biror bir parametrlarga bog'langan bo'ladi.

### **Proseduralar va ularni e'lon qilish**

Proseduralar funksiyalardan farqi shundaki, prosedura faqat bitta qiymat qaytarmaydi, balki bir nechta qiymat qaytarishga mo'ljallangandir. Yagona nom bilan saqlangan prosedura yordamida ikki sonning yig'indisini, ko'paytmasini, nisbatini va ayirmasini hisoblovchi prosedura yaratish mumkin.

*Ta'rif: C++ dasturlash tilining dastur tarkibida ikki va undan ortiq qiymat qaytaradigan qism dasturlar **proseduralar** deyiladi.*

Proseduralarni e'lon qilishda, albatta, uning kiritish va qaytariladigan qiymatlar parametrlari keltirilishi kerak. Proseduralarni e'lon qilish usullari to'liq funksiyalarni e'lon qilish usullari bilan bir xil bo'ladi, ya'ni prosedurani main() funksiyasidan oldin e'lon qilish kerak yoki murojat qilishdan avval main() funksiyasining ichida e'lon qilish kerak.

Proseduralarni C++ dasturlash tilida e'lon qilinishining umumiy ko'rinishi quyidagicha bo'ladi:

**tur <prosedura nomi>(tur <parametr1>, tur & <parametr2>);**

Proseduralarni e'lon qilishda qavs ichida birinchi **parametr1** sifatida kiritish parametrlari funksiyani kabi tasvirlanadi keyin **&** bilan parameter2 sifatida qiymat qaytaruvchi parametrlar tasvirlanadi. Proseduralarni tasvirlash uchun ikki sonning yig'indisi va ko'paytnasini hisoblovchi *kop* nomi bilan yaratilgan prosedura dasturiga e'tibor bering.

```
#include <iostream.h>
int kop(int m,int n,int & k,int & t);
using namespace std;
int main()
{
}
```

*kop(m,n,k,t)* prosedurasi *m* va *n* sonlarining yig'indisini *k* o'zgaruvchiga ko'paytnasini esa *t* o'zgaruvchiga saqlaydi. Prosedura qiymat qaytaruvchi parametrlar, albatta, **&** belgisi bilan yoziladi.

### **Prosedura asosiy tanasini tasvirlash va ulardan foydalanish**

Proseduralarning asosiy tanasini C++ dasturlash tilida tasvirlash uchun *main()* funksiyasidan oldin yoki *main()* funksiyasidan keyin xuddi funksiyalar kabi tasvirlanadi. Proseduralarni C++ dasturlash tilida tasvirlash jarayonining umumiy ko'rinishi quyidagicha bo'ladi:

```
<tur> <prosedura nomi>(tur <parametr1>, tur & <parametr2>)
{ <tur> <lokal o'zgaruvchilar>;
  <operatorlar>;
  parametr2= natija;
}
```

Proseduralar tasvirlanish jarayonida proseduraning natijalari, albatta, qaytaradigan qiymatlar parametrlariga ta'minlanishi kerak.

*Misol:* Ikki sonni yig'indisi va ko'paytnasini hisoblovchi *kop(m,n,k,t)* prosedurasini yarating.

```
#include <iostream.h>
int kop(int m,int n,int & k,int & t);
using namespace std;
int main()
{ int a,b,p,q;
```



<pre> cin&gt;&gt;a&gt;&gt;b; kop(a,b,p,q); cout&lt;&lt;"a+b="&lt;&lt;p&lt;&lt;" ab="&lt;&lt;q; } int kop(int m,int n,int &amp; k,int &amp; t) { k=m+n; t=m*n; return 0; } </pre>	
<i>Kiritish ma'lumotlari</i>	<i>Dastur natijasi</i>
4 5	$a+b=9$ $ab=20$

C++ dasturlash tilida proseduralarga murojat qilish jarayonida hech qachon ifoda tarkibida murojat qilinmaydi. Chunki prosedura natijani nomiga emas, balki o'zgaruvchilarga ta'minlaydi, shuning uchun oldin prosedura ishlatiladi va keyin uning prosedura tarkibidagi qiymat qaytaruvchi o'zgaruvchi ifoda ichida ishlatilishi mumkin.

### Nazariy savollar

- 1 Prosedura deganda nimani tushunasiz?
- 2 Proseduralarni e'lon qilish usullarini ayting?
- 3 Proseduralarni e'lon qilishning umumiy ko'rinishi?
- 4 Prosedura tanasini tasvirlashning umumiy ko'rinishi?
- 5 Proseduralarga qanday usullarda murojat qilinadi.

### Mustaqil ishlash uchun topshiriqlar

1. *IsPalindrom(N)* mantiqiy funksiya hosil qiling. ( $N > 0$ ). Agar  $N$  soni palindrom bo'lsa – true, aks xolda false qiymat qaytarilsin. Shu funksiya orqali 5 ta sonidan nechtasi palindrom ekanini aniqlovchi dastur tuzilsin. *IsPalindrom* funksiyasida *DigitCount* va *DigitN* funksiyalaridan foydalanish mumkin. (Qarang 29, 30)
2. Haqiqiy qiymat qaytaruvchi *DegToRad(D)* funksiyasini hosil qiling. ( $360 > D > 0$ ). Funksiya  $D$  gradusning radian qiymatini qaytarsin. Shu funksiya orqali 3 ta berilgan burchakning radianga qiymati aniqlansin.
3. Haqiqiy qiymat qaytaruvchi *RadToDeg(D)* funksiyasini hosil qiling. Funksiya  $R$  radianning gradus qiymatini qaytarsin. Shu funksiya orqali radianda berilgan 3 ta burchakning gradusdagi qiymati aniqlansin.

4. Butun qiymat qaytaruvchi  $Fact(N)$  funksiyasini hosil qiling. ( $N > 0$ ). Funksiya  $N!$  qiymatini qaytarsin. Shu funksiya orqali 3 ta sonlarining faktoriali aniqlansin.

5. Butun qiymat qaytaruvchi  $Fact2(N)$  funksiyasini hosil qiling. ( $N > 0$ ). Funksiya  $N!!$  ikkilangan factorial qiymatini qaytarsin. Shu funksiya orqali 3 ta sonlarining ikkilangan faktoriali aniqlansin. (while6 ga qarang)

6. Butun qiymat qaytaruvchi  $Fib(N)$  funksiyasini hosil qiling. ( $N > 0$ ). Funksiya Fibonachchi sonlarining  $N$  – elementini qaytarsin.

7. Haqiqiy qiymat qaytaruvchi  $Power1(A, B)$  funksiyasini hosil qiling. Funksiya  $A^B$  darajasi qiymatini qaytarsin. Shu funksiya orqali berilgan  $A1, A2, A3$  sonlarining  $B$  darajasi topilsin.  $A, A1, A2, A3, B$  haqiqiy sonlar.

8. Haqiqiy qiymat qaytaruvchi  $Power2(A, N)$  funksiyasini hosil qiling. Funksiya  $A^N$  darajasi qiymatini quyidagi formula asosida topilsin. Shu funksiya orqali berilgan  $A$  sonining  $N, M, K$  darajalari topilsin.  $A^0 = 1$ ; Agar  $N > 0$  bo'lsa,  $A^N = A * A * \dots * A$  ( $N$  ta ko'paytuvchi). Agar  $A < 0$  bo'lsa,  $A^N = 1 / (A * A * \dots * A)$  ( $|N|$  ta ko'paytuvchi).  $A$  haqiqiy,  $N, M, K$  butun sonlar.

9. Haqiqiy qiymat qaytaruvchi  $Power3(A, N)$  funksiyasini hosil qiling. Funksiya  $A^N$  darajasi qiymatini quyidagi topilsin. Agar  $N$  sonining kasr qismi 0 dan farqli bo'lsa,  $Power2(A, N)$  funksiyasi chaqirilsin. Aks holda  $Power1(A, B)$  funksiyasi chaqirilsin (37, 38 masalalarga qarang). Shu funksiya orqali berilgan  $A$  sonining  $N, M, K$  darajalari topilsin.  $A, N, M, K$  haqiqiy sonlar. Eslatma:  $Power2(A, N)$  – funksiyasini chaqirishda  $N$  – sonining butun qismi jO'natiladi.

10. Haqiqiy qiymat qaytaruvchi  $Exp(x, \varepsilon)$  funksiyasini hosil qiling. ( $x$  – haqiqiy son,  $\varepsilon > 0$ ) Funksiya quyidagi yig'indi qiymatini qaytarsin.

$$1 + x + x^2/(2!) + x^3/(3!) + \dots + x^n/(n!) + \dots$$

Yig'indida  $\varepsilon$  dan katta bo'lgan barcha xadlar qatnashsin. Berilgan  $x, \varepsilon1, \varepsilon2, \varepsilon3$  uchun funksiya hisoblansin.

11. Haqiqiy qiymat qaytaruvchi  $sin1(x, \varepsilon)$  funksiyasini hosil qiling. ( $x$  – haqiqiy son,  $\varepsilon > 0$ ) Funksiya quyidagi yig'indi qiymatini qaytarsin.

$$\sin(x) = x - x^3/(3!) + x^5/(5!) - \dots + (-1)^n x^{2n+1}/((2*n+1)!) + \dots$$

Yig'indida  $\varepsilon$  dan moduli katta bo'lgan barcha xadlar qatnashsin. Berilgan  $x, \varepsilon1, \varepsilon2, \varepsilon3$  uchun funksiya hisoblansin.

12. Haqiqiy qiymat qaytaruvchi  $cos1(x, \varepsilon)$  funksiyasini hosil qiling. ( $x$  – haqiqiy son,  $\varepsilon > 0$ ) Funksiya quyidagi yig'indi qiymatini qaytarsin.

$$\cos(x) = 1 - x^2/(2!) + x^4/(4!) - \dots + (-1)^n x^{2n}/((2*n)!) + \dots$$

Yig'indida  $\varepsilon$  dan moduli katta bo'lgan barcha xadlar qatnashsin. Berilgan  $x, \varepsilon1, \varepsilon2, \varepsilon3$  uchun funksiya hisoblansin.

**13.** Haqiqiy qiymat qaytaruvchi  $\text{Ln}1(x, \varepsilon)$  funksiyasini hosil qiling. ( $|x| < 1, \varepsilon > 0$ ) Funksiya quyidagi yig'indi qiymatini qaytarsin.

$$\ln(1+x) = x - x^2/2 + x^3/3 - \dots + (-1)^n x^{n+1}/(n+1) + \dots$$

Yig'indida  $\varepsilon$  dan moduli katta bo'lgan barcha xadlar qatnashsin. Berilgan  $x, \varepsilon_1, \varepsilon_2, \varepsilon_3$  uchun funksiya hisoblansin.

**14.** Haqiqiy qiymat qaytaruvchi  $\text{Arctg}1(x, \varepsilon)$  funksiyasini hosil qiling. ( $|x| < 1, \varepsilon > 0$ ) Funksiya quyidagi yig'indi qiymatini qaytarsin.

$$\arctg(x) = x - x^3/3 + x^5/5 - \dots + (-1)^n x^{2n+1}/(2n+1) + \dots$$

Yig'indida  $\varepsilon$  dan moduli katta bo'lgan barcha xadlar qatnashsin. Berilgan  $x, \varepsilon_1, \varepsilon_2, \varepsilon_3$  uchun funksiya hisoblansin.

**15.** Haqiqiy qiymat qaytaruvchi  $\text{Power}4(x, a, \varepsilon)$  funksiyasini hosil qiling. ( $|x| < 1, a, \varepsilon > 0$ ) Funksiya quyidagi yig'indi qiymatini qaytarsin.

$$(1+x)^a = 1 + a*x + a*(a-1)*x^2/(2!) + \dots + a*(a-1)*\dots*(a-n+1)*x^n/(n!) + \dots$$

Yig'indida  $\varepsilon$  dan moduli katta bo'lgan barcha xadlar qatnashsin. Berilgan  $x, a, \varepsilon_1, \varepsilon_2, \varepsilon_3$  uchun funksiya hisoblansin.

**16.** Butun qiymat qaytaruvchi  $\text{EKUB}(A, B)$  funksiyasini hosil qiling. Funksiya  $A$  va  $B$  sonlarining eng katta umumiy bo'luvchisini qaytarsin.  $A, B, C, D$  sonlari berilgan.  $(A, B), (A, C), (A, D)$  juftliklari uchun  $\text{EKUB}$  hisoblansin.

**17.**  $\text{EKUB}$  funksiyasidan (46) foydalangan holda  $\text{Frac}1(a, b, p, q)$  funksiyasini hosil qiling. Funksiya  $(a/b)$  kasrini qisqartmaydigan  $(p/q)$  ko'rinishini qaytarsin.  $a, b, c, d$  sonlari berilgan.  $(a/b), (a/c), (a/d)$  kasrlarini qisqartiring.

**18.**  $\text{EKUB}$  funksiyasidan foydalangan holda butun qiymat qaytaruvchi  $\text{EKUK}(A, B)$  funksiyasini hosil qiling. Funksiya  $A$  va  $B$  sonlarining eng kichik umumiy karralisini qaytarsin.  $A, B, C, D$  sonlari berilgan.  $(A, B), (A, C), (A, D)$  juftliklari uchun  $\text{EKUK}$  hisoblansin.  $\text{EKUK} = A*B/\text{EKUB}(A, B)$ ;

**19.**  $\text{EKUB}$  funksiyasidan foydalangan holda (46 masalasiga qarang) butun qiymat qaytaruvchi  $\text{EKUB}3(A, B, C)$  funksiyasini hosil qiling. Funksiya  $A, B, C$  sonlarining eng kichik umumiy karralisini qaytarsin.  $A, B, C, D$  sonlari berilgan.  $(A, B, C), (A, C, D), (A, B, D)$  sonlari uchun  $\text{EKUB}$  hisoblansin.

**20.**  $\text{TimeToHMS}(T, H, M, S)$  funksiyasi hosil qiling. Funksiya berilgan  $T$  sekundni  $H$  – soat,  $M$  – minut,  $S$  – sekundlarga ajratsin. Funksiya orqali berilgan  $T_1, T_2, T_3$  sekundlar soat, minut va sekundlarga aylantirilsin.

Masalan:  $T=400$

Natija: 00:06:40

**21.**  $\text{IncTime}(H, M, S, T)$  funksiyasini hosil qiling. Funksiya berilgan  $H$  – soat,  $M$  – minut,  $S$  – sekundni  $T$  sekundga oshirsin. Funksiya orqali berilgan soat, minut va sekundni  $T$  sekundga oshiruvchi dastur tuzilsin.

22. Mantiqiy qiymat qaytaruvchi  $IsLeapYear(Y)$  funksiyasini hosil qiling. Funksiya berilgan  $Y$  – yil kabisa yili bo'lsa true, aks holda false qiymat qaytarsin. Berilgan 5 ta yilning kabisaligi tekshirilsin. (Kabisalik shartini bilish uchun IF28 masalaga qarang).

23.  $IsLeapYear(Y)$  funksiyasidan foydalangan holda, butun qiymat qaytaruvchi  $MonthDays(M, Y)$  funksiyasini hosil qiling. Funksiya berilgan  $Y$  – yilning  $M$  – oyi kunlar sonini qaytarsin. Berilgan yilning  $M1, M2, M3$  oylarining kunlar soni topilsin. (52 masalaga qarang).

24.  $MonthDays$  funksiyasidan foydalangan holda,  $PrevDate(D, M, Y)$  funksiyasini hosil qiling. Funksiya berilgan sanadan oldingi sanani aniqlasin,  $D$  – kun,  $Y$  – yil,  $M$  – oyini qaytarsin. 3 ta berilgan sanadan oldingi sana aniqlansin. (53 masalaga qarang).

25.  $MonthDays$  funksiyasidan foydalangan holda,  $NextDate(D, M, Y)$  funksiyasini hosil qiling. Funksiya berilgan sanadan keyingi sanani aniqlasin,  $D$  – kun,  $Y$  – yil,  $M$  – oyini qaytarsin. 3 ta berilgan sanadan keyingi sana aniqlansin. (53 masalaga qarang).

26. Haqiqiy qiymat qaytaruvchi  $Leng(X1, Y1, X2, Y2)$  funksiyasini hosil qiling. Funksiya berilgan  $(X1, Y1)$  va  $(X2, Y2)$  nuqtalar orasidagi masofani qaytarsin.  $A, B, C, D$  nuqtalar koordinatalari berilgan.  $A$  nuqtadan  $B, C, D$  nuqtalargacha bo'lgan masofalar hisoblansin.  $|AB|, |AC|, |AD|$  - ?

27.  $Leng$  funksiyasidan foydalangan holda, haqiqiy qiymat qaytaruvchi  $Perim(X_A, Y_A, X_B, Y_B, X_C, Y_C)$  funksiyasini hosil qiling. Funksiya berilgan koordinatalar asosida uchburchak perimetrini qaytarsin.  $A, B, C, D$  nuqtalar koordinatalari berilgan.  $ABC, ABD, ACD$  uchburchaklar perimetrlari hisoblansin. (56 masalaga qarang).

28.  $Leng$  va  $Perim$  funksiyalaridan foydalangan holda, haqiqiy qiymat qaytaruvchi  $Area(X_A, Y_A, X_B, Y_B, X_C, Y_C)$  funksiyasini hosil qiling. Funksiya berilgan koordinatalar asosida uchburchak yuzini qaytarsin.  $A, B, C, D$  nuqtalar koordinatalari berilgan.  $ABC, ABD, ACD$  uchburchaklar yuzalari hisoblansin. (57 masalaga qarang).

29.  $Leng$  va  $Area$  funksiyalaridan foydalangan holda, haqiqiy qiymat qaytaruvchi  $Dist(X_A, Y_A, X_B, Y_B, X_P, Y_P)$  funksiyasini hosil qiling. Funksiya berilgan koordinatalar asosida  $P$  nuqtadan  $AB$  kesmaga tushirilgan balandlikni hisoblab qaytarsin. (Formula quyidagicha:  $D(P, AB) = 2 * S_{PAB}/|AB|$ )  $P, A, B, C$ , nuqtalar koordinatalari berilgan.  $P$  nuqtadan  $AB, BC, AC$  kesmaga tushirilgan balandlik hisoblansin. (56 va 58 masalalarga qarang).

30. Dist funksiyasidan foydalangan holda, Heights( $X_A, Y_A, X_B, Y_B, X_C, Y_C, h_A, h_B, h_C$ ) funksiyani hosil qiling. Funksiya berilgan koordinatalar asosida uchburchak tomonlariga tushurilgan balandliklarni hisoblab qaytarsin. A, B, C nuqtalar koordinatalari berilgan. Uchburchak tomonlariga tushuril

#### 4.6 C++ DASTURLASH TILIDA KO'RSATKICHLAR VA ADRES OLVUCHI O'ZGARUVCHILAR

Reja:

1. Funksiyaga ko'rsatkich;
2. Obyektga o'zgaruvchiga ko'rsatkich;
3. Ko'rsatkichlarga boshlang'ich qiymat berish;
4. Ko'rsatkichlar ustida bajariladigan amallar.
5. Adresdagi qiymatlar ustida amallarni bajarish

Ko'rsatkichlar C++ dasturlash tilining kuchli qurollaridan biridir. Tushunchaning nomi nega ko'rsatkich, u nimani ko'rsatadi degan savollar tug'ilishi tabiiy. Ko'rsatkich ham o'zgaruvchi bo'lib, u ham e'lon qilinadi va uning oddiy o'zgaruvchilar e'lonidan farqi shundaki, ularni e'lon qilishga oldidan ko'paytirish amali belgisi, ya'ni (\*) yulduzcha qo'yiladi.

Ko'rsatkich o'zgaruvchi boshqa o'zgaruvchining EHM xotirasidagi adresiga teng qiymatni oladi, ya'ni o'sha o'zgaruvchining adresini ko'rsatadi. C++ dasturlash tilida tuzilgan dastur tarkibidagi o'zgaruvchilar dastur komplyatsiya qilingandan keyin kompyuter xotirasiga borib joylashadi. Komplyator dastur matnini kompyuter xotirasiga joylashtirgandan so'ng uning tarkibidagi o'zgaruvchilar adresini obyekt kod sifatida qabul qilib oladi.

**Masalan, `int *x`** ; yozuvi butun turli x nomli ko'rsatkich o'zgaruvchi, **`float *a`**; yozuvi esa **a** nomli haqiqiy turli ko'rsatkich o'zgaruvchi e'lon qilinganligini bildiradi. Yoki boshqacha qilib aytganda o'zgaruvchi e'lon qilinishi chog'ida uning oldiga (\*) belgisi qo'yilsa, uni ko'rsatkichga aylantiradi. Bundan buyon «**ko'rsatkich o'zgaruvchi**» o'rniga «**ko'rsatkich**» so'zidan foydalanamiz.

Unutmaslik kerakki, ko'rsatkich qaysi turga mansub bo'lsa, faqat shu turga taalluqli o'zgaruvchilarning adreslarini aniqlay oladi.

Ko'rsatkichlar bilan ikkita, ya'ni «\*» va «&» operatorlari qo'llaniladi. Bunda «&» operatori o'zgaruvchining EHM xotirasidagi adresini aniqlaydi.

«\*» operatori esa «&» aniqlangan o'zgaruvchining o'sha adresga joylashgan qiymatini aniqlaydi. Bu fikrlarni quyidagi dastur yordamida tushunib olish mumkin.

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

```

using namespace std;
int main( )
{
int n, m, *x; n=123;
x=&n; cout <<x<<'\n';
m=*x; cout <<m<<'\n';
system("pause");
return 0;
}

```

Dastur natijasi quyidagicha bo‘ladi:

0x28ff44

123

Dasturda quyidagicha o‘zgarishlar qilinsa ham natijasida hech qanday o‘zgarish bo‘lmaydi.

```

int n=123;
int *x=&n; cout<<x<<'\n': m=*x;
yoki
int x=123;
int*x (&n); cout <<x<<'\n'; m=*x;

```

Har ikkala holda ham x ga boshlang‘ich qiymat sifatida n o‘zgaruvchining xotiradagi adresi, m o‘zgaruvchiga esa n o‘zgaruvchi ko‘rsatilgan adresdagi qiymati boshlang‘ich qiymat sifatida berilmoqda.

C++ dasturlash tilida tuzilgan dastur tarkibidagi o‘zgaruvchilar, o‘zgaruvchilar va funksiyalar adreslarini kompyuter xotirasiga alohida saqlash va ular ustida amallar bajarish mumkin.

*Ta’rif: Qiymatlari adres bo‘lgan o‘zgaruvchilar ko‘rsatkich o‘zgaruvchilar deb ataladi.*

Ko‘rsatkichlar uch xil turda bo‘ladi:

- Funksiyaga ko‘rsatkich;
- Obyektga o‘zgaruvchiga ko‘rsatkich;
- Void ko‘rsatkich.

Ko‘rsatkichlar, albatta, birorta turga bog‘langan bo‘ladi, ya’ni u ko‘rsatgan adresda ma’lum bir qiymat joylashishi mumkin va bu qiymat kompyuter xotirasidan qancha joy egallashi oldindan ma’lum bo‘lishi kerak.

## Funksiyaga ko‘rsatkich

Funksiyaga ko‘rsatkich programma joylashgan xotiradagi funksiya kodini boshlang‘ich adresini ko‘rsatadi, ya‘ni funksiya chaqirilganda boshqaruv ayni shu adresga uzatiladi. Bunda funksiya nomi bo‘yicha emas, balki funksiya ko‘rsatuvchi o‘zgaruvchi orqali chaqiriladi. Funksiyani boshqa funksiya argument sifatida uzatish ham funksiya ko‘rsatkichi orqali bajariladi. C++ dasturlash tilida funksiya ko‘rsatkichining yozilishi quyidagicha.

**<tur> (\*<nom>)(<parametrlar ro‘yxati>);**

Bu yerda \*<nom> funksiya ko‘rsatkich o‘zgaruvchisi nomi.

### **Obyektga o‘zgaruvchiga ko‘rsatkich**

C++ dasturlash tilida o‘zgaruvchiga ko‘rsatkich e‘lon qilish, bunda kompyuter xotirasiga o‘zgaruvchining adresi yoziladi. C++ tilida o‘zgaruvchi va o‘zgaruvchi qiymatlarga ko‘rsatkich e‘lon qilish bir xil bo‘ladi. C++ dasturlash tilida o‘zgaruvchi ko‘rsatkichining yozilishi quyidagicha:

**<tur> \*<nom>;**

C++ dasturlash tilida o‘zgaruvchiga ko‘rsatkich e‘lon qilishni quyidagi dastur orqali aniqlaymiz:

```
#include <iostream.h>
using namespace std;
int main()
{ int *a, *b;
  float *i, *y;
return 0;
}
```

Yuqoridagi dasturda a va b butun ko‘rsatkichlar, i va y haqiqiy ko‘rsatkichlar keltirilgan.

### **void ko‘rsatkichlar**

void ko‘rsatkichlar obyekt turi oldindan aniq bo‘lmagan holatlarda ishlatiladi. Void ko‘rsatkichini afzalliklari unga har qanday turdagi ko‘rsatkich qiymatini yuklash mumkin. C++ dasturlash tilida void ko‘rsatkichining yozilishi quyidagicha:

**void \*<nom>;**

void ko‘rsatkichi sifatida e‘lon qilingan ko‘rsatkichga boshqa turdagi o‘zgaruvchilarni taminlash imkoniyati mavjud bo‘ladi.

### **Ko‘rsatkichlarga boshlang‘ich qiymat berish**

C++ dasturlash tili tarkibidagi ko'rsatkichlarga boshlang'ich qiymatlarni berish uchun, albatta, biror o'zgaruvchi orqali qabul qilish maqsadga muvofiq bo'ladi. Ko'rsatkichlarga boshlang'ich qiymatlarni = belgisi yoki qavs ichiga olib berilish mumkin. Ko'rsatkichlarni boshlang'ich qiymatlarini berish uchun quyidagi dasturga e'tibor bering:

```
#include <iostream.h>
using namespace std;
int main()
{ int x=10, k=9;
  int *y=&x;
  int *t(&k);
return 0;
}
```

Yuqoridagi dasturga e'tibor qaratsak, x va k butun sonlar qiymatlari bilan aniqlangan. \*y va \*t ko'rsatkichlarga boshlang'ich qiymat sifatida x va k ni qiymatlari berilgan.

### **Ko'rsatkichlar ustida bajariladigan amallar**

C++ dasturlash tilida ko'rsatkichlar ustida bajariladigan amallar quyidagilarni tashkil etadi:

- obyektga vositali murojat qilish amali;
- qiymat berish amali;
- ko'rsatkichga o'zgarmas qiymatni qo'shish amali;
- ayirish amali;
- inkrement va decrement amali;
- solishtirish amali.

C++ dasturlash tilida ko'rsatkichlar ustida faqatgina yuqorida keltirilgan amallardan foydalanish maqsadga muvofiq bo'ladi.

**Obyektga vositali murojat qilish amali** ko'rsatkichdagi adres bo'yicha joylashgan qiymatni olish tushuniladi. Obyektga vositali murojat qilish amalini quyidagi dastur orqali qarab o'tamiz.

```
#include <iostream.h>
using namespace std;
int main()
{ int x=10;
```



```
int *y=&x;
cout<<*y;
return 0;
}
```

*Dastur natijasi*

10

Obyektga vositali murojat yordamida  $x=10$  butun sonning qiymatini  $*y$  ko'rsatgichga qabul qilindi.

**Qiymat berish amali** yordamida ko'rsatgichdagi qiymatni oddiy butun yoki haqiqiy qiymatga qabul qilinishi mumkin. Qiymat berish amalini quyidagi dastur orqali qarab o'tamiz.

```
#include <iostream.h>
using namespace std;
int main()
{ int x=12;
  int *a(&x);
  int y=*a;
  cout<<y;
return 0;
}
```

*Dastur natijasi*

12

Qiymat berish amali yordamida  $*a$  ko'rsatkichni qiymatini  $y$  butun o'zgaruvchiga qabul qilindi.

**Ko'rsatgichga o'zgarmas qiymatni qo'shish, ayirish, inkrement va dekrement amallarini** C++ dasturlash tilidagi ifodasini quyidagi dastur orqali qarab o'tamiz:

```
#include <iostream.h>
using namespace std;
int main()
{ int x=12;
  int *a(&x);
  a+=4; //O'zgarmas qaiymatni qo'shish
  a=a-2; //ayirish
  a++; //inkrement amali
```

```
a--; //dekrement amali
return 0;
}
```

C++ dasturlash tilida ko'rsatkichlar bo'yicha amal va vazifalar yuqoridagi holatlar bo'yicha amalga oshiriladi.

### Adres olish amali

C++ dasturlash tilida biror bir o'zgaruvchini uning sinonimi bilan almashtirib unga murojat qilish imkoniyati mavjud. Biror bir o'zgaruvchining boshqa o'zgaruvchiga adresini olish mumkin va unga murojat qilish uchun keyingi adres bo'yicha murojat qilinadi. C++ dasturlash tilida adres olish amalining umumiy ko'rinishi quyidagicha bo'ladi:

**<tur> &<nom>;**

Adres olish amali asosan ko'p hollarda funksiya parametrlari sifatida olinadi, adres olish amalini C++ dasturlash tilida quyidagi dastur orqali tushuntirib o'tamiz:

```
#include <iostream.h>
using namespace std;
int main()
{ int x=12;
  int &a=x;
  a=(a+2)/2;
  cout<<"a="<<a;
  return 0;
}
```

*Dastur natijasi*

*a=7*

Adres oluvchi o'zgaruvchining ko'rsatkichdan farqi shundaki, u xotiradan alohida joy egallamaydi va faqat o'z qiymati bo'lgan o'zgaruvchining boshqa nom sifatida ishlatiladi.

### Adresdagi qiymatlar ustida amallarni bajarish

Ba'zan o'zgaruvchilarning xotirada egallangan joyini hisobga olishga ham to'g'ri keladi. 32 razryadli mashinalarda:

1 ta belgi 1 bayt, butun turli (**int**) va haqiqiy turli (**float**) o'zgaruvchilar to'rt bayt, suzuvchi vergulli (**double**) turli o'zgaruvchilar sakkiz bayt joyni egallaydi.

Ko'rsatkichdan qiymat berish buyrug'ining chap tarafidan foydalanish mumkin. Bu holda u qandaydir qiymatni oladi va shu qiymatni o'sha adresga yozib qo'yadi. Ushbu holatni quyidagi dasturni tahlil qilish bilan tushunish mumkin.

```
#include <iostream.h>
using namespace std;
int main( )
{
int *p, x=100; cout <<"x ning eski qiymati= "<<x<<"\n";
p=&x; cout<<"x o`zgaruvchining adresi= "<<p<<"\n";
*p=213;
cout<<"x ning yangi qiymati= "<<x<<"\n";
system("pause");
return 0;
}
```

Dastur natijasi quyidagicha bo'ladi:

x ning eski qiymati= 100

x o'zgaruvchining adresi= 0x28ff44

x ning yangi qiymati= 213

Shuningdek, ko'rsatkich yordamida u ko'rsatgan adresdagi qiymatning ustida inkrement va dekrement amallarini bajarish mumkin. Bunda ko'rsatkich qavs ichida yozilishi shart. Chunki inkrement va dekrement amallari ko'rsatkichga nisbatan oldin bajariladi. Buni quyidagi dastur misolida tushunish mumkin.

```
#include <iostream.h>
using namespace std;
int main( )
{
float x=54.25, *p1; cout <<"x="<<x<<"\n";
p1=&x; cout <<"x ning adresi="<<p1<<"\n";
(*p1)++; cout <<"x="<<x<<"\n";
(*p1)++; cout <<"x="<<x<<"\n";
(*p1)--; cout <<"x="<<x<<"\n";
*p1=13; cout <<"x="<<x<<"\n";
(*p1)++; cout <<"x="<<x<<"\n";
(*p1)++; cout <<"x="<<x<<"\n";
(*p1)--; cout <<"x="<<x<<"\n";
}
```

```
system("pause"); return 0;}
```

Dastur natijasi quyidagicha bo'ladi:

x=54.25

x ning adresi= x28ff44

x=55.25

x=56.25

x=55.25

x=13

x=14

x=15

x=14

Endi ko'rsatkichlar ustida arifmetik amallarni bajarish bilan tanishamiz:

Bizga ma'lumki, **float \*a, \*b;** buyrug'i bo'yicha **a** va **b** nomli ko'rsatkichlarni aniqlaymiz. Agar **a=&n;** buyrug'i bo'yicha **a** ko'rsatkichning qiymati **n** ning xotiradagi adresiga teng bo'ladi. Shundan so'ng (**\*a**) ko'rsatkich **n** ning EHM xotirasidagi qiymatiga teng qiymatni oladi. Lekin, **a=&n;** buyrug'i bo'lmasa (**\*a**) ham hech qanday qiymatni olmaydi. Chunki adressiz ko'rsatkich qaysi qiymatni olishi mumkin? Agar **a=&n;** buyrug'idan keyin (**\*a**)++ buyrug'i o'rniga **a++** buyrug'idan foydalansak, bu inkrement amali **n** ning EHM xotirasidagi adresiga birni qo'shmaydi, balki shu adresdan keyingi adres qiymatiga teng qiymatni oladi. Masalan, **n**ning xotiradagi adresi **0x28ff3c** bo'lsa, **a++** ning qiymati **0x28ff40** bo'ladi. Bunda **0x28ff3c** ning ustiga **4** ni qo'shdi. Chunki yuqorida qayd qilganimizdek, **float** turli o'zgaruvchilar xotirada **4** bayt joyini egallaydi.

Tabiiyki, **a=&n; cout<<- - a <<`\n`;** buyrug'i natijasi **0x288ff38** bo'ladi. Bunda **a** ning dastlabki qiymati hisobga olinyapdi, ya'ni **0x288ff3c** dan **4** ayrilyapdi (16lik sanoq sistemasida).

Demak, biror o'zgaruvchining xotiradagi adresini **1**ga oshirmoqchi bo'lsak, uning qiymati **1** ga emas, balki o'sha o'zgaruvchining xotirada egallaydigan joyining hajmi qadar oshadi.

Xuddi shuningdek, ko'rsatkichga qandaydir butun sonni qo'shsak yoki ayirsak o'sha ko'rsatkich turi xotirada egallaydigan joyi hajmini berilgan butun songa ko'paytirilgani qadar sonni qo'shish yoki ayirish natijasiga teng bo'ladi. Aytilganlar bo'yicha quyidagi dasturni va uning natijalarini tahlil qiling.

```
#include <iostream.h>
using namespace std;
int main( )
```

```

{
int *y,a,n=-3;
y=&n; cout <<"*y="<<*y<<"\n";
    cout <<"y="<<y<<"\n";
y=y+5; cout <<"y+5="<<y<<"\n";
y=y-8; cout <<"y-8="<<y<<"\n";
y=&n-5; cout <<"&n-5="<<y<<"\n";
system ("pause");
return 0;}

```

Dastur natijasi quyidagicha bo'ladi.

```

*y=-3
y=0x28ff3c
y+5=0x28ff50
y-8=0x28gg30
&n-5=0x28ff28

```

Bunda amallar 16 lik sanoq sistemasida bajarilmoqda. Ko'rsatkichlar ustida qo'shish amalini bajarib bo'lmaydi. Lekin, ikkita ko'rsatkich ustida ayirish amalini bajarish mumkin. Bu holni quyidagi dastur misolida namoyish qilamiz.

```

#include <iostream.h>
#include <math.h>
using namespace std;
int main( )
{
int *y, *z,a, n=-3; a=14;
y=&n; cout <<"y="<<y<<"\n";
z=&a; cout <<"z="<<z<<"\n";
cout <<"z-y="<<z-y<<"\n";
cout <<"*z-*y="<<*z-*y<<"\n";
cout <<"*z+*y="<<*z+*y<<"\n";
system ("pause");
return 0;}

```

Dastur natijasi quyidagicha bo'ladi.

```

y=0x28ff34
z=0x28ff38
z-y=1

```

$$*z-*y=17$$

$$*z+*y=11$$

Dasturni va uning natijalarini mustaqil tahlil qiling.

### Nazariy savollar

- 1 Ko'rsatkich o'zgaruvchi deb nimaga aytiladi?
- 2 Ko'rsatkichlar necha xil turda bo'ladi?
- 3 Funksiya ko'rsatkichi va uning umumiy ko'rinishi?
- 4 o'zgaruvchiga ko'rsatkich va uning umumiy ko'rinishi?
- 5 Void ko'rsatkichi va uning umumiy ko'rinishi?
- 6 Ko'rsatkichlarga boshlang'ich qiymat berish qanday bajariladi?
- 7 Ko'rsatkichlar ustida bajariladigan amallar va ularning tasnifi?
- 8 Adres olish amali va uning umumiy ko'rinishi?
- 9 Ko'rsatkich tushunchasining mohiyatini tushuntiring.
- 10 Ko'rsatkich bilan o'zgaruvchining farqini tushuntiring.
- 11 float \*z,\*g; yozuvini qanday tushunasiz?
- 12 c=\*f; va d=&j; yozuvlarini qanday tushunasiz?
- 13 Ko'rsatkichlar bilan ishlashda adresdagi qiymatlar ustida inkrement va dekrement amallari qanday bajariladi?
- 14 Ko'rsatkichlar ustida arifmetik amallar qanday bajariladi?

## **V-BOB. C++ DASTURLASH TILIDA SATR, Fayl va STRUKTURA TURIDAGI MA'LUMOTLAR VA ULARDAN FOYDALANISH.**

C++ dasturlash tili tarkibida satr, to'plam, fayl va struktura turiga mansub bo'lgan ma'lumotlar ustida amallar bajarish imkoniyati ham mavjud. Satr turi bu dasturlashning asosiy turlaridan hisoblanadi, bunda char turiga mansub barcha ma'lumotlar shu tur asosida qayta ishlanadi. Dasturlash asoslarida satr turi bilan ishlash bu fanning asosiy maqsadi hisoblanadi. Tashkilot va muassasalarning axborotlarini qayta ishlash vaqtida uning tarkibida asosan satr turlari bilan ishlashga to'g'ri keladi. Bu bobda satrlar ustida satrning uzunligi, satrni nusxalash, satrni kerakli joyga o'rnatish kabi bir qator vazifalar ko'rib o'tiladi. C++ dasturlash tilida to'plam va fayl turlari ham asosiy turlardan hisoblanadi. Matematik masalalarni hal etishda asosan to'plam turi katta ahamiyat kasb etadi. C++ dasturlash tilida to'plam, to'plam elementlari, element to'plamga tegishli ekanligini aniqlash masalalarini hal etish imkoniyati mavjud. Dasturlash asoslari fanida struktura turiga mansub bo'lgan ma'lumotlar katta ahamiyat kasb etadi. Struktura turidagi ma'lumotlar tarkibida bir nechta turlar birgalikda qaraladi, masalan, tashkilot xodimlari haqida ma'lumotlar struktura turiga tegishli bo'ladi. Struktura turiga e'tibor qaratsak, bu tur dasturlashning asosiy turlari hisoblanadi, bu tur asosan ma'lumotlar bazasini boshqarish tizimlarining asosi hisoblanadi. Yuqorida ko'rilgan turlar va ulardan foydalanish imkoniyati C++ dasturlash tili tarkibida mavjud. Mazkur qullanmaning joriy bobida yuqoridagi turlar va ulardan foydalanish usullari qarab o'tiladi.

### **5.1 C++ DASTURLASH TILIDA SATR TURI VA ULARDAN FOYDALANISH USULLARI**

*Reja:*

- 1. Belgi va satrlar;*
- 2. Satr uzunligini aniqlash;*
- 3. Satrlarni nusxalash;*
- 4. Satrlarni ulash;*
- 5. Satrlarni solishtirish (taqqoslash) funksiyalari*
- 6. Satrdagi harflar registrini almashtirish funksiyalari*
- 7. Satrlarni teskari tartiblash*
- 8. Satrdan belgini izlash*
- 9. Satr ichidan satr qismini izlash*
- 10. Satr turini o'zgartiruvchi funksiyalar*

C++ dasturlash tilida satr turi va ulardan foydalanish usullari va tamoyillarini qarab o'tamiz. Satr turi bu belgilardan tashkil topgan ma'lumotlar majmuasi hisoblanadi. Satrlarning uzunligini aniqlash, satrlarni nusxalash, satrlarni kerakli joyga o'rnatish kabi amallarni bajarish uchun C++ dasturlash tilida keng imkoniyati mavjud. C++ dasturlash tilida satrlar ham boshqa turlar kabi oldin dastur tarkibida e'lon qilinishi kerak. Satrlar e'lon qilingandan keyin ular ustida amallar bajarish mumkin.

### Belgi va satrlar

C++ dasturlash tilida bir nechta turlardan foydalanish va ular ustida amallar bajarish mumkin, dasturlash muhitida boshqa turlar kabi belgili turlar ham mavjud. C++ dasturlash tilida belgili turlar char turiga mansub bo'lgan yagona elementli tur hisoblanadi. C++ dasturlash tilida satrlar uchun alohida maxsus turlar ishlab chiqilmagan satrlarni belgilar massivi sifatida ishlatish imkoniyati mavjud.

*Ta'rif: Alohida nom bilan saqlanuvchi bir nechta belgilardan tashkil topgan belgilar majmuasi satr deyiladi.*

Satrlarni belgilar massivi sifatida qarab ular ustida amallar bajarish mumkin. C++ dasturlash tilida satrlar nol('\0') terminatori bilan tugaydi. Nol terminatori bilan tugaydigan satrlar ASCIIZ –satrlari deyiladi.

Satlardan foydalanish va ular ustida amallar bajarish uchun albatta oldin ularni e'lon qilish kerak. Satr turiga mansub bo'lgan o'zgaruvchilarni char xizmatchi so'zi orqali e'lon qilinadi. Satrlarni e'lon qilish uchun quyidagi dasturlarga etibor bering.

```
#include <iostream.h>
#include <string.h>
using namespace std;
int main()
{ char s[10]; // s satrli o'zgaruvchini e'lon qilish
  return 0;
}
```

Yuqoridagi dastur tarkibida s[10] satr o'zgaruvchisi 9 ta elementga va bitta nol terminatoriga mo'ljallangan. Satrlarni boshlang'ich qiymatlarini berish jarayonida uning elementlar sonini berish shart emas. Agar satrlarni elementlar soni berilmasa dastur avtomatik ravishda uning elementlar soniga boshlang'ich qiymatdagi elementlar sonidan bitta ortiq qilib ta'minlaydi, chunki oxiriga bitta nol



terminatorini hisobga oladi. Doim satrlarni kiritishda uning elementlar soni e'lon qilinganidan oshib ketmasligi kerak aks holda faqat e'lon qilingan elementlar soniga teng elementlarni saqlab qolinib qolganlari olinmaydi.

**Satrlarni kiritish** jarayonida >> oqim bo'yicha kiritishdan foydalanmasdan **getline()** funksiyasidan foydalanish maqsadga muvofiq bo'ladi. Agar oqim bo'yicha kiritishdan foydalanilsa unda probellar inkor qilinib satrni ikkinchi qismlari qabul qilinmaydi. **getline(s,n)** funksiyasi ikkita parametрни o'z ichiga oladi, birinchisi s satr o'zgaruvchisi ikkinchisi n satr elementlar soni. Getline funksiyasi satr elementlaridan ortiq qiymatlarni ham kiritish imkoniyatini yaratadi, natijada satr elementlari ortadi. Kiritish jarayonini amalga oshirish uchun quyidagi dasturlarga etibor bering.

*Misol: Berilgan satrni ekranga chiqaring.*

<pre>#include &lt;iostream.h&gt; #include &lt;string.h&gt; using namespace std; int main() { char s[10]; // s satrli o'zgaruvchini e'lon qilish   cin.getline(s,10);   cout&lt;&lt;s;   return 0; }</pre>	
<i>Kiritish ma'lumotlari</i>	<i>Dastur natijasi</i>
Absdefghl	Absdefghl

Yuqoridagi dastur tarkibidagi s satrli o'zgaruvchi faqat 9 ta elementni qbul qiladi, agar 9 tadan ortiq bo'lsa natija qaytarmaydi.

### Satr uzunligini aniqlash

Satrlar ustida amallar bajarish vaqtida albatta satrlarning uzunligi kerak bo'ladi. Satr uzunligini aniqlash uchun **string.h** kutubhonasiga murojat qilish kerak aksariyat satr uchun ishlatiladigan funksiyalarni string.h kutubhonasiga murojat qilib ishlatishimiz mumkin. Satr uzunligini aniqlashni ikki hil usuli mavjud.

- sotr tarkibidagi mavjud elementlar sonini nol terminatorisiz aniqlash;
- sotr tarkibidagi elementlarga ajratilgan sonini nol terminatori bilan aniqlash;

Satr tarkibidagi mavjud elementlar sonini nol terminatorisiz aniqlash funksiyasining umumiy ko‘rinishi quyidagicha bo‘ladi.

**strlen(<satrli o‘zgaruvchi>);**

Strlen funksiyasi o‘z tarkibiga satrli o‘zgaruvchi yoki satr uzini qabul qilishi mumkin. Masalan strlen (“absd”) bo‘lsa uning natijasi 4 ga teng bo‘ladi.

*Misol: Berilgan satr tarkibidagi elementlar sonini aniqlang..*

<pre>#include &lt;iostream.h&gt; #include &lt;string.h&gt; using namespace std; int main() { char s[10];   cin.getline(s,10);   x=strlen(s);   cout&lt;&lt;x;   return 0; }</pre>	
<i>Kiritish ma'lumotlari</i>	<i>Dastur natijasi</i>
Dastur	5

Demak strlen(s) funksiyasi s satrni tarkibidagi mavjud elementlari sonini aniqlaydi.

Satr tarkibidagi elementlarga ajratilgan sonini nol terminatori bilan aniqlash uchun **sizeof(s)** funksiyasidan foydalaniladi. Sizeof(s) funksiyasi s satr uchun ajratilgan elementlar sonini aniqlaydi, s satrni 10 ta elementga moslab e‘lon qilib lekin 5 ta element kiritilsa ham sizeof funksiyasi 10 qiymatni qaytaradi. Satr tarkibidagi elementlarga ajratilgan sonini nol terminatori bilan aniqlash funksiyasining umumiy ko‘rinishi quyidagicha bo‘ladi.

**sizeof(<satrli o‘zgaruvchi>);**

Sizeof funksiyasi o‘z tarkibiga satrli o‘zgaruvchi yoki satr uzini qabul qilishi mumkin. Masalan strlen (“absd”) bo‘lsa uning natijasi 5 ga teng bo‘ladi.

*Misol: Berilgan satr tarkibidagi elementlar sonini aniqlang.*

<pre>#include &lt;iostream.h&gt; #include &lt;string.h&gt; using namespace std; int main() { char s[10];</pre>
--

<pre>cin.getline(s,10);   x=sizeof(s);   cout&lt;&lt;x;   return 0; }</pre>	
<i>Kiritish ma'lumotlari</i>	<i>Dastur natijasi</i>
Salom	10

Demak sizeof(s) funksiyasi s satrni tarkibidagi mavjud elementlari sonini emas balki unga ajratilgan qiymatni aniqlaydi. Agar sizeof funksiyasi tarkibida o'zgaruvchi emas satr uzini kiritrsa unda sizeof funksiyasi tarkibidagi elementlardan bitta ko'p qiymatni qaytaradi chunki nol terminatorini ham qo'shadi. Quyidagi dasturga e'tibor bering.

<pre>#include &lt;iostream.h&gt; #include &lt;string.h&gt; using namespace std; int main() { cout&lt;&lt;sizeof("dastur");   return 0; }</pre>
<i>Dastur natijasi</i>
6

**Misol:** Berilgan s satr tarkibidan p element necha marta ishtirok etganligini aniqlang.

<pre>#include &lt;iostream.h&gt; #include &lt;string.h&gt; using namespace std; int main() { int x,n=0;   char s[100];   cin.getline(s,10); //s satrni kiritish   char p;   cin&gt;&gt;p;          //p belgini kiritish   x=strlen(s);   for(int i=0;i&lt;=x-1;i++)   if(s[i]==p) n++;   cout&lt;&lt;n;</pre>
---

<pre> return 0; } </pre>	
<i>Kiritish ma'lumotlari</i>	<i>Dastur natijasi</i>
s=assalom p=a	2

Satrlarni ixtiyoriy elementiga murojat qilish uchun doimo bitta kam qilib buyruq berish kerak chunki satrlar belgili massiv bo'lganligi uchun boshlang'ich element nolinchisi o'rindan boshlanadi.

### Satrlarni nusxalash

C++ dasturlash tilida satrlarni bir biriga nusxalashni bir qancha usullari mavjud. Satrlarni nusxalash uchun **strcpy()** funksiyasidan foydalanish mumkin, strcpy() funksiyasining umumiy ko'rinishi quyidagicha bo'ladi.

#### **strcpy(satr1, satr2);**

strcpy(satr1, satr2) funksiyasi satr2 dagi elementlarni to'laligicha satr1 ga nusxalaydi. Satr1 ni elementlar soni doimo satr2 elementlar sonidan katta bo'lishi kerak.

*Misol: Berilgan s satr elementlarini p satrga nusxalang.*

<pre> #include &lt;iostream.h&gt; #include &lt;string.h&gt; using namespace std; int main() { char s[100]; cin.getline(s,10); char p[10]; strcpy(p,s); cout&lt;&lt;p; return 0; } </pre>	
<i>Kiritish ma'lumotlari</i>	<i>Dastur natijasi</i>
s=dastur	Dastur

Satrlarni nusxalash uchun **strncpy()** funksiyasidan foydalanish mumkin, strncpy() funksiyasining umumiy ko'rinishi quyidagicha bo'ladi.

#### **strncpy(satr1, satr2,n);**

strncpy(satr1, satr2,n) funksiyasi satr2 dagi elementlarni n tasini satr1 ni boshiga nusxalaydi. Strncpy() funksiyasini strcpy() funksiyasidan farqi satr2 ni nta elementini satr1 ni boshiga nusxalaydi.

**Misol:** Berilgan s satrni 3 ta elementini p satrga nusxalang.

<pre>#include &lt;iostream.h&gt; #include &lt;string.h&gt; using namespace std; int main() { char s[100]; cin.getline(s,10); char p[10]; cin.getline(p,10); strncpy(p,s,3); cout&lt;&lt;p; return 0; }</pre>	
<p><i>Kiritish ma'lumotlari</i></p> <p>s=algoritm p=dastur</p>	<p><i>Dastur natijasi</i></p> <p>Algtur</p>

**strncpy (st1, st2, n)** funksiyasi. Bunda funksiya **st1** satrining dastlabki **n** ta belgisi ustiga **st2** ning **n** dastlabki **n** ta belgisini nusxalaydi.

**strdup (satr)** funksiyasi. Funksiya xotiradan **satr** uchun joy ajratib, o'sha joyga **satr** ni nusxalaydi.

Agar zarurat bo'lsa ko'rsatkichlardan foydalanib, satrning nechanchidir belgisidan keyingilarini ham nusxalash mumkin. Quyidagi dastur bu funksiyalar mohiyatini ochishga yordam beradi.

```
1 #include <iostream.h>
2 #include <string.h>
3 using namespace std;
4 int main()
5 { int i,j,n,m;
6 char*star1="Satr nusxasini olish";
7 char*satr2;
8 char str3[30]="Assalom", st1[20]="123456789 soni",
9 st2[26];
10 char*korsat=st1; cout<<korsat<<'\n';
```

```

11 korsat+=4; cout<<korsat<<'\n';
12 strepy(st2, "nusxalashga misol"); cout<<st2<<'\n';
13 strncpy(st2, "soniga qarab nusxalash", strlen (str3)); cout<<st2<<'\n';
14 strncpy(st1, "dastlabki 4 belgini nusxalash",4); cout<<st1<<'\n';
15 satr2=strdup(satr1); cout<<satr2<<'\n';
16 syste4m("pause");
17 return 0;
18 }

```

Dastur natijasi quyidagicha bo‘ladi:

```

123456789 soni
56789 soni
nusxalashga misol
soniga shga misol
dast 56789 soni
satr nusxasini olish

```

10-satrdagi «**ko‘rsat**» nomli ko‘rsatkichga **st1** massivi qiymati initsializatsiyalangan. 11-satrdagi **korsar+=4;** buyrug‘i bilan ko‘rsatkich o‘zgaruvchining qiymati 4 bayt o‘ngga siljildi. 12-satrdagi «**nusxalashga misol**» matni **strepy( )** funksiyasi yordamida **st2** massiviga nusxalandi. 13-satrdagi buyruq bo‘yicha «**soniga qarab nusxalash**» matnining dastlabki 7 ta belgisi **st2** massivining dastlabki 7 ta belgisi o‘rniga nusxalandi. Chunki **strlen(str3)** funksiyaning qiymati 7 ga tengligini tushunish qiyin emas. 15-satrdagi buyruq bo‘yicha **satr1** ko‘rsatkichning qiymati **satr2** ko‘rsatkichga nusxalandi.

### Satrlarni ulash

Bir nechta satrlarni ulash natijasida yangi satrlarni hosil qilish mumkin. C++ dasturlash tili tarkibida satrlarni bir biriga ulashni bir nechta usullari mavjud, satrlarni bir biriga ulab yangi satrlar hosil qilinadi.

C++ dasturlash tilida **strcat()** funksiyasi yordamida satrlarni bir biriga ulash imkoniyati mavjud, **strcat()** funksiyasining umumiy ko‘rinishi quyidagicha ko‘rinishda bo‘ladi.

**strcat(satr1, satr2);**

C++ dasturlash tilida **strcat(satr1, satr2)** funksiyasi **satr2** ni **satr1**ni oxiriga ulaydi.

*Misol: Berilgan s satrni p satrga birlashtiring.*

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

```

#include <string.h>
using namespace std;
int main()
{ char s[100];
cin.getline(s,10);
char p[10];
cin.getline(p,10);
strcat(p,s,3);
cout<<p;
return 0;
}

```

*Kiritish ma'lumotlari*

s=ali  
p=vali

*Dastur natijasi*

Valiali

C++ dasturlash tilida strcat() funksiyasi yordamida satrlarni bir biriga faqat n ta elementini ulash imkoniyati mavjud, strcat() funksiyasining umumiy ko'rinishi quyidagicha ko'rinishda bo'ladi.

**strncat(satr1, satr2,n);**

C++ dasturlash tilida strncat(satr1, satr2, n) funksiyasi satr2 ni n ta boshlang'ich elementlarini satr1ni oxiriga ulaydi.

*Misol:* Berilgan s satrni boshidagi 3 ta elementini p satr oxiriga birlashtiring.

```

#include <iostream.h>
#include <string.h>
using namespace std;
int main()
{ char s[100];
cin.getline(s,10);
char p[10];
cin.getline(p,10);
strncat(p,s,3);
cout<<p;
return 0;
}

```

*Kiritish ma'lumotlari*

*Dastur natijasi*

s=algoritm p=dastur	Dasturalg
------------------------	-----------

### Satrlarni solishtirish (taqqoslash) funksiyalari

C++ da satrlarni solishtirish uchun quyidagi funksiyalardan foydalaniladi.

1. **Strcmp (s1, s2)** funksiyasi. Bu funksiya **s1** satrining har bir belgisi **s2** satrining mos nomerlaridagi belgilarining EXM xotirasidagi kodlarni solishtiradi.

**Eslatma:** Lotin alifbesining kichik harflari kodi katta harflari kodidan **32** ga ortiq qilib kodlangan va solishtirish natijasi qandaydir **n** butun soniga qiymat sifatida berilishi kerak. Masalan, **n=strcmp (s1.s2)**

Agar

**s1>s2 bo'lsa n>0**

**s1=s2 bo'lsa n=0**

**s1<s2 bo'lsa n<0 bo'ladi**

**n** ning qiymati birinchi farq qiluvchi belgilar kodlarini solishtirish bilan aniqlanadi.

```
#include <iostream.h>
#include <string.h>
using namespace std;
int main()
{
int j,i,s,n,m; char k1[20]="",k2[20]="";
char s1[60]="Programmash ";char *kor1=s1;
char s2[60]="prograMMalash ";char *kor2=s2;
m=strcmp(kor1,kor2); cout<<" m="<<m;
n=strcmp(kor2,kor1); cout<<" n="<<n<<"\n";
system("pause");
return 0;
}
```

**s1 va s2** larning birinchi belgilari **P** va **p** larning kodlari farq qilayapti. Shuning uchun dastur natijasi **m=-1; n=1** kabi bo'lmoqda.

2. Agar satrlarni solishtirishda harflarning registrlarini (ya'ni katta yoki kichikligini) farqlash kerak bo'lmasa, u holda **strcmpi (s1, s2)** funksiyasidan foydalaniladi. Bu hol quyidagi dasturda o'z aksini topgan.

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



```

#include <string.h>
using namespace std;
int main()
{
    int j,i,s,n,m;
    char satr1[60]="Programmash ";char *kor1=satr1;
    char satr2[60]="prograMMalash ";char *kor2=satr2;
    char satr3[60]="pragraMMalash ";char *kor3=satr3;
    m=strcmpi(kor1,kor2); cout<<" m="<<m;
    n=strcmpi(kor2,kor3); cout<<" n="<<n<<"\n";
    system("pause");
    return 0;
}

```

Dastur natijasi **m=0, n=1** kabi bo‘lishini tushunish qiyin emas.

3. Agar ikkita satrning faqat dastlabki **n** ta belgilarini solishtirish talab etilgan bo‘lsa, **strncmp(satr1, satr2,n)** funksiyasidan foydalaniladi. Quyidagi dasturni tahlil qilib bunga ishonch hosil qilish mumkin.

```

#include <iostream.h>
#include <string.h>
using namespace std;
int main()
{
    int j,i,s,n,m;
    char satr1[60]="programmash ";char *kor1=satr1;
    char satr2[60]="prograMMalash ";char *kor2=satr2;
    m=strncmp(kor1,kor2,6); cout<<" m="<<m;
    n=strncmp(kor1,kor2,7); cout<<" n="<<n<<"\n";
    system("pause");
    return 0;
}

```

Dastur natijasi **m=0, n=32** kabi bo‘ladi.

**Misol.** Ikkita satr berilgan. Ularning farq qiluvchi belgilarini aniqlash dasturini tuzing.

```

#include <iostream.h>
#include <string.h>

```

```

using namespace std;
int main()
{
    int j, i, s, n, m;   char k1[20]="", k2[20]="";
    char satr1[60]="progoamalash tillarini oRcanamiZ"; char
*kor1=satr1;
    char satr2[60]="Programflash tillarini organamiz"; char
*kor2=satr2;
    for(i=1; i<=32; i++) {
        strncpy(k1, kor1, 1);strncpy(k2, kor2, 1); kor1+=1; kor2+=1;
n=strcmp(k1, k2);
        if(n!=0) cout<<i<<"-chi belgisi har xil, ular: "<<k1<< " va "
<<k2<<"\n';
    }
    system("pause");
    return 0;
}

```

Dastur natijasi quyida kltirilgan. Mustaqil tahlil qiling.

1-chi belgisi har xil, ular: p va P

5-chi belgisi har xil, ular: o va r

8-chi belgisi har xil, ular: a va f

25-chi belgisi har xil, ular: R va r

26-chi belgisi har xil, ular: c va g

32-chi belgisi har xil, ular: Z va z

### **Satrdagi harflar registrini almashtirish funksiyalari**

Agar satrdagi belgilari ichida harflar bo'lsa, ularni bosh harflardan kichik harflarga almashtirish uchun **strlwr(satr)** funksiyasidan, aksincha kichik harflarni bosh harflarga almashtirish uchun esa **strupr(satr)** funksiyasidan foydalaniladi.

Masalan

```

#include <iostream.h>
#include <string.h>
using namespace std;
int main()
{
    char satr2[30]="maktabda o'qiyman";
   strupr(satr2);

```

```

cout<<satr2<<"\n";
char satr3[30]="MEN TALABAMAN";
strlwr(satr3);
cout<<satr3<<"\n";
system("pause");
return 0;
}

```

Dastur natijasi  
MAKTABDA O‘QIYMAN  
men talabaman  
kabi bo‘ladi

### Satrlarni teskari tartiblash

Satrlarni teskari tartiblash uchun C++ dasturlash tilidan tashqari boshqa dasturlash tillarida takrorlanish jarayonlari orqali bajariladi. C++ dasturlash tilida satrlarni teskari tartiblash uchun `strrev()` funksiyasi yordamida bajarish mumkin.

`Strrev()` funksiyasining umumiy ko‘rinishi quyidagicha bo‘ladi.

**`strrev(<satrO‘zgaruvchisi>);`**

`Strrev(s)` funksiyasi `s` satrni teskari tartibda jaoylashtiradi.

*Misol:* Berilgan `s` satrni teskari tartiblang.

```

#include <iostream.h>
#include <string.h>
using namespace std;
int main()
{ char s[100];
cin.getline(s,10);
cout<<strrev(s);
return 0;
}

```

*Kiritish ma’lumotlari*

`s=algoritm`

*Dastur natijasi*

Mtirogl

### Satrdan belgini izlash

Satr tarkibidan birorta belgini qidirish uchun C++ dasturlash tilining `string.h` kutubxonasi ichidan `strchr()` funksiyasidan foydalanish kerak. `Strchr()` funksiyasining umumiy ko‘rinishi quyidagicha bo‘ladi.

**`strchr(<s,p>);`**

Strchr(s) funksiyasi s satr ichida p belgini izlaydi agar topilsa shu elementdan keyingi qismni aks holda nullni qaytaradi.

**Misol:** Berilgan s satrdan p belgini qidiring.

<pre>#include &lt;iostream.h&gt; #include &lt;string.h&gt; using namespace std; int main() { char s[100]; cin.getline(s,10); char p; cin&gt;&gt;p;     cout&lt;&lt;strchr(s,p);     return 0; }</pre>	
<i>Kiritish ma'lumotlari</i> s=algoritm p=t	<i>Dastur natijasi</i> Tm

### Satr ichidan satr qismini izlash

Satr tarkibidan birorta qism satrni qidirish uchun C++ dasturlash tilining string.h kutubxonasi ichidan strstr() funksiyasidan foydalanish kerak. Strstr() funksiyasining umumiy ko'rinishi quyidagicha bo'ladi.

**strstr(<s,p>);**

Strstr(s) funksiyasi s satr ichida p qism satrni izlaydi agar topilsa shu satrdan keyingi qismni aks holda nullni qaytaradi.

**Misol:** Berilgan s satrdan p satrni qidiring.

<pre>#include &lt;iostream.h&gt; #include &lt;string.h&gt; using namespace std; int main() { char s[100]; cin.getline(s,100); char p[100]; cin.getline(p,100);     cout&lt;&lt;strstr(s,p);     return 0; }</pre>
---

<i>Kiritish ma'lumotlari</i> s=aliakbar p=ak	<i>Dastur natijasi</i> Akbar
--	---------------------------------

### Satr turini o'zgartiruvchi funksiyalar

Ko'pincha sonlarni qayta ishlashni talab qiluvchi masalalarda sonni matematik almashtirishlardan foydalanib qayta ishlagandan ko'ra uni satr ko'rinishda qayta ishlab, keyin satrni songa aylantirib qo'yish ancha qulay bo'ladi. Buning uchun C++ dagi **<stdlib.h>** funksiyalar kutubxonasidagi quyidagi funksiyalardan foydalaniladi.

1. **atoi(satr)** funksiyasi. Bu funksiya turi butun (**int**) bo'lib, satr ko'rinishida berilgan sonni butun turga aylantiradi. Satrni to'liq songa aylantirish uchun satrning belgilari to'liq raqamlar bo'lishi kerak. Agar satrning boshidagi bir nechta belgilarigina raqamlardan iborat bo'lsa, **atoi()** funksiyasi faqat shu qismini songa aylantiradi va qolgan qismini tashlab yuboradi. Agar satr raqam bilan boshlanmagan bo'lsa, funksiya 0(nol) ga teng qiymatni oladi. Yuqorida keltirilgan fikrlarni quyidagi dastur misolida tahlil qiling.

```
#include <iostream.h>
#include <stdlib.h>
using namespace std;
int main()
{
    int i,j,k,n,m;
    char satr1[30]="215625", satr2[30]="25";
    char satr3[30]="2154254254698",
satr4[30]="7458TALA215AN",satr5[]="assalom";
    k=atoi(satr1); n=atoi(satr2); i=atoi(satr3); j=atoi(satr4);
m=atoi(satr5);
    cout<<"i="<<i<<"\nj="<<j<<"\n";
cout<<"k="<<k<<"\nn="<<n<<"\n";
    cout<<"m="<<m<<"\n";
    system("pause");
    return 0;
}
```

Dastur natijasi quyidagicha bo'ladi  
i=-1819327894

```
j=7458
k=215625
n=25
m=0
```

Satrnı songa aylantirishda son tur chegarasidan chiqib ketsa C++ kutilmagan natijani chop etadi. Masalan, dastur **i** parametri uchun shunday qiymat tanlangan.

2. **atof(satr)** funksiyasi. Bu funksiya ham **atoi()** funksiyasi kabi ishlaydi. Faqat bu funksiyaning qiymati **long** turiga mansub bo‘ladi. Bundan tashqari funksiyaga **float** yoki **double** turi berilsa, satr ko‘rinishidagi sonni **double** ko‘rinishidagi, ya’ni sonning mantissasi va tartibini ko‘rsatish bilan yoki o‘nli kasr ko‘rinishiga o‘tkazadi. Agar songa aylantirilayotgan satr chegaradan chiqib ketsa, funksiya kutilmagan qiymatni oladi.

```
#include <iostream.h>
#include <stdlib.h>
using namespace std;
int main()
{
    int k,n,m; double i,j ;
    char satr1[]="3.14D15E+2";
    char satr3[30]="21.34254254E15", satr4[30]="745.8215";
    n=atoi(satr1); i=atof(satr3); j=atof(satr4);
    cout<<"i="<<i<<"\nj="<<j<<<<"\nn="<<n<<"\n";
    system("pause");
    return 0;
}
```

Dastur natijasi quyidagicha bo‘ladi

```
i=2.13425e+016
j=745.822
n=3
```

3. **itoa (n,satr,m)** funksiyasi. Bu funksiya **int** turiga mansub bo‘lgan **n** sonini **m** lik sanoq sistemasiga o‘tkazib, satr turiga o‘tkazadi va hosil bo‘lgan qiymatni **satrli** massivga beradi.

**Itoa (n,satr,m)** funksiyasi vazifasi ham **itoa()** funksiyasi vazifasi kabi. Faqat **Itoa()** funksiyasi **long** turiga mansub sonlarni ham satrga aylantiradi. Quyidagi dasturni tahlil qilib, bunga ishonch hosil qilish mumkin.

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

```

#include <stdlib.h>
#include <string.h>
using namespace std;
int main()
{
    int n=3245,m; long int i,j=568324557; long double k=5.6e4;
    char satr1[20],satr2[20],
satr3[20],satr4[20],satr5[20],satr6[20],*satr7;
    itoa(n,satr1,2); itoa(n,satr2,8);itoa(n,satr3,16);itoa(n,satr4,10);
    cout<<"satr1="<<satr1<<"\nsatr2="<<satr2<<"\nsatr3="<<satr3<<"\nsa
tr4="<<satr4<<"\n';
    ltoa(j,satr1,2); ltoa(k,satr2,8);ltoa(j,satr3,16);ltoa(j,satr4,10);
    cout<<"satr1="<<satr1<<"\nsatr2="<<satr2<<"\nsatr3="<<satr3<<"\nsatr4
="<<satr4<<"\n';
    itoa(n,satr5,10);ltoa(j,satr6,10);satr7=strstr(satr6,satr5);cout<<"
satr7="<<satr7<<"\n';
    system("pause");
    return 0;
}

```

Dastur natijasi quyidagicha bo'ladi.

```

satr1=110010101101
satr2=6255
satr3=cad
satr4=3245
satr1=100001110111111111000111001101
satr2=155300
satr3=21dff1cd
satr4=568324557
satr7=324557

```

4. **gcvt (son,raqamlar soni, satr)** funksiyasi. Bu funksiya **float** yoki **doudle** turidan sonni satrga aylantiradi. Bunda son **float** yoki **double** turiga mansub son bo'lilshi kerak,**raqamlar soni** satrga aylantirilgan sonning raqamlar soni. Agar sonning raqamlari soni ko'rsatilgan sondan ortiq bo'lsa, quyidagi ikki hol bo'lishi mumkin.

- 1) Butun son berilgan bo'lsa, **unEm** ko'rinishida ifodalanadi;
- 2) O'nli kasar ko'rinishida bo'lsa, ortiqcha raqamlari yaxlitlanadi.

Quyidagi dasturda bu imkoniyatlar o'z aksini topgan

```

#include <iostream.h>
#include <stdlib.h>
using namespace std;
int main()
{
    int n=4; double i=4562, m=1234567,k=12.455476;
    char satr[20];
    gcvt(i,n,satr);cout<<"i ning satr korinishi="<<satr<<"\n";
    gcvt(m,n,satr);cout<<"m ning satr korinishi="<<satr<<"\n";
    gcvt(k,n,satr);cout<<"k ning satr korinishi="<<satr<<"\n";
    system("pause");
    return 0;
}

```

Dastur natijasi quyidagicha bo'ladi.

i ning satr korinishi=4562

m ning satr korinishi=1.235e+006

k ning satr korinishi=12.46

### Nazariy savollar

1. Belgi va satrlar deganda nimani tushunasiz?
2. Satrlarni kiritish operatori va uning vazifasi?
3. Satr uzunligini aniqlash funksiyalari va ularning umumiy ko'rinishi?
4. Satrlarni nusxalash funksiyasi va uning umumiy ko'rinishi?
5. Satrlarni ulash funksiyasi va uning umumiy ko'rinishi?
6. Satrlarni teskari tartiblash funksiyasi va uning umumiy ko'rinishi?
7. Satr tarkibidan belgini izlash funksiyasi va uning umumiy ko'rinishi?
8. Satr ichidan satr qismini izlash funksiyasi va uning umumiy ko'rinishi?
9. Dasturlashda satr nima, belgi nima? Ularning turlari qanday bo'ladi?
10. Satrli massivlar qanday va qayerda e'lon qilinadi?
11. `char sum[15][30];` satrli massiv elementlari qanday joylashadi?
12. `char m[15][30];` massiv e'lonida m harfi, 15 va 30 sonlari nimani bildiradi?
13. Satrli massiv elementlari qiymatlarini kiritish operatori qaysi?
14. `getline(satr,k);` funksiyasi vazifasini tushuntiring.
15. `216+44` va `'216'+ '44'` ifodalar natijasi qanday bo'ladi va nima uchun?



16. Satrlarni taqqoslashda foydalaniladigan barcha funksiyalarni yozing va tushuntiring?

17. Satrdagi harflar registrini almashtirish funksiyalarini yozing va tushuntiring?

### **Mustaqil ishlash uchun topshiriqlar**

1. Nuqta bilan tugaydigan satr berilgan. Satrda nechta soʻz borligini hisoblab chiqing.

2. Ingliz matnidan iborat satr berilgan. *b* harfi bilan boshlanuvchi soʻzlar sonini toping.

3. Satr berilgan. Unda *r*, *k*, *t* harflari qanchaligini hisoblab chiqing.

4. Satr berilgan. Unda \*(yulduzcha), ;(nuqta vergul), :(ikki nuqta) belgilari sonini hisoblab chiqing.

5. Matndan iborat satr berilgan. Eng qisqa va eng uzun soʻz uzunliklarini toping.

6. Orasida ikki nuqtasi boʻlgan belgili satr berilgan. Ungacha boʻlgan belgilar qanchaligini aniqlang.

7. Nuqta bilan tugaydigan, matndan tashkil topgan satr berilgan. Uchta harfdan iborat soʻzni ekranga chiqaring.

8. Berilgan satrdagi barcha na qisman satrni nad qisman satr bilan almashtirng.

9. Satr berilgan. Unda *abc* qisman satr necha marta uchrashini aniqlang.

10. Satr berilgan. Uning oxirgi soʻzidagi *k* harfi migʻdorini hisoblab chiqing.

11. Satr berilgan. Unda har xil belgilar necha marta uchrashini hisoblab chiqing. Ularni ekranga chiqaring.

12. Satr berilgan. Berilgan ikkita gapdagi oʻchraydigan bir xil soʻzni chop eting.

13. Orasida bitta ochilgan qavs va bitta yopilgan qavs mavjud boʻlgan belgili satr berilgan. Bu qavslar orasidagi barcha belgilarni ekranga chiqaring.

14. Lotin harflari va raqamlardan tashkil topgan satr mavjud. ...

15. Ikki nuqta bilan tugovchi va nuqta vergul bilan ajarilgan soʻzlar toʻplami berilgan. *a* harfi bilan tugovchi soʻzlar qanchaligini aniqlang.

16. Satr berilgan. Tarkibida kamida bitta *k* harfi boʻlgan soʻzni koʻrsating.

17. Satr berilgan. Boshlanishi va tugashi bir xil harfdan iborat soʻzni toping.

18. Satrdagi barcha ikki nuqtalarni nuqta vergul bilan almashtiring. Almashishlar migʻdorini hisoblang.

19. Satrdagi ikki nuqta belgilarini o'chiring va o'chirilgan belgilar mig'dorini hisoblang.

20. Satrdagi so'zlar orasidagi bO'sh joy o'rniga vergul va bO'sh joyni qo'ying.

21. Qavsga olingan belgili satr qismini o'chiring (qavslar bilan birgalikda).

22. Berilgan so'z satrda necha marta uchrashini aniqlang.

23. Satrda bitta nuqtali vergul mavjud. Nuqtali vergulgacha bo'lgan va undan keyingi belgilar mig'dorini hisoblang.

24. Satr berilgan. Birinchi  $n/2$  belgilar orasida uchraydigan barcha ikki nuqtalarni nuqta bilan almashtiring va  $n/2$  dan keyingi belgilar orasida uchraydigan barcha undov belgilarni nuqta almashtirib uni o'zgartiring.

25. Satr bitta so'zdan tashkil topgan. Uni chapdan o'ngga va o'ngdan chapga qarab o'qiganda bir xil bo'lishini (ya'ni u polindrom hisoblanishini) tekshiring.

26. Qo'lyozmadagi so'zlarning har biri teskarisiga yozilib shifrlangan. Xabarni oching (rasshifrovka qiling).

27. Berilgan satrda ochilgan qavslar soni yopilgan qavslar soni bilan bir hilligini tekshiring.

28. 200 belgidan oshib ketmaydigan ixtiyoriy ruscha matndan tashkil topgan satr berilgan. Bu matnda qaysi belgi necha marta uchrashini yozing. Javob grammatik jihatdan to'g'ri bo'lishi kerak, masalan «a – 25 marta», «k – 3 marta» va h.k.

29. Ingliz so'zlardan iborat berilgan massivni alfavit bo'yicha tartiblang.

30. Ikkita A va B satrlar berilgan. A satrdagi harflardan B satrni tuzish mumkinligini (harflarni o'rnini almashtirib ishlatish mumkin, lekin ularni bir martadan ortiq ishlatish mumkin emas) tekshiradigan dastur tuzing. Masalan, A:INTEGRAL, B:AGENT – tuzish mumkin; B:AGENT – tuzish mumkin emas.

## 5.2 C++ DASTURLASH TILIDA STRING TURIGA MANSUB BO'LGAN SATRLI O'ZGARUVCHILAR

*Reja:*

1. *String turidagi o'zgaruvchilarni e'lon qilish ;*
2. *String turidagi satrlar ustida amallar bajarish;*
3. *String turidagi satrlarni nusxalash;*
4. *Satrning biror qismidan nusxa olish funksiyasi*
5. *Satr tarkibiga boshqa satrni joylashtirish*
6. *Satr qismini o'chirish*
7. *Matn qismini boshqa matn qismi bilan almashtirish funksiyasi*
8. *Satr tarkibidan satr qismini izlash*
9. *Satrlarni solishtirish funksiyalari*
10. *Satr xossalari aniqlash funksiyalari*

C++ dasturlash tilida satrlar uchun standart turdan tashqari string turi ham qushimcha sifatida qaralgan. String turiga mansub bo'lgan satrlar nol terminatori bilan tugallanmasdan u oddiy belgilar massivi sifatida qaraladi. String turini afzalliklari satrlar ustida amallar bajarilgandan So'ng dinamik ravishda o'zgartirishlar mumkin.

### **String turidagi o'zgaruvchilarni e'lon qilish**

C++ dasturlash tilida string turidagi o'zgaruvchilarni e'lon qilish uchun string xizmatchi so'zi orqali amalga oshiriladi.

**string s,s1;**

String turiga mansub bo'lgan satrlar ustida amallar bajarish imkoniyati C++ dasturlash tilida keng imkoniyatga ega. C++ dasturlash tilida string turiga mansub bo'lgan satrlarni boshlang'ich qiymatlarini bir nechta usullarda berish mumkin. String turidagi satrlarni boshlang'ich qiymatlarini berish jarayonini quyidagi dastur orqali qaraymiz:

```
#include <iostream.h>
#include <string.h>
using namespace std;
int main()
{ string s,s1; // s va s1 larni e'lon qilish
  s="dastur"; // s ni boshlang'ich qiymatini berish
```

```
s1=s; // s1 ni boshlang'ich qiymatini berish
return 0;
}
```

String turidagi satrlar ustida amallar bajarish uchun uning elementlari bilan ishlashga to'g'ri keladi. String turidagi satrlarni har bir elementlariga murojat qilish uchun at() funksiyasi yoki oddiy [] elementga murojat qilish orqali murojat qilinadi. Masalan s.at(3) s satrning uchinchi elementiga murojat qilinadi. s.at(3) va s[3] holatlari teng kuchli hisoblanadi.

Satrlarni ixtiyoriy elementiga murojat qilish uchun doimo bitta kam qilib buyruq berish kerak, chunki satrlar belgili massiv bo'lganligi uchun boshlang'ich element nolinchisi o'rindan boshlanadi.

**Misol:** Berilgan satrni ixtiyoriy elementini ekranga chiqaring.

<pre>#include &lt;iostream.h&gt; #include &lt;string.h&gt; using namespace std; int main() { string s; int n; cin&gt;&gt;s; cin&gt;&gt;n; cout&lt;&lt;s.at(n-1); return 0; }</pre>	
<p><i>Kiritish ma'lumotlari</i></p> <p>s=dastur n=3</p>	<p><i>Dastur natijasi</i></p> <p>S</p>

Demak, satrlarni yuqoridagi qoidalarga asoslanib e'lon qilib uning ixtiyoriy elementlariga ta'sir qilish mumkin.

### String turdagi satrlar ustida amallar bajarish

C++ dasturlash tilida string turdagi satrlar ustida bir nechta amallar bajarish mumkin. String satrlarini char tipidagi satrlardan afzalliklari bir qancha amallarni bajarish inkoniyati kengdir. C++ dasturlash tilida string satrlari ustida quyidagi amallarni bajarish mumkin:

Amal	Mazmuni	Misol
=,+ =	Qiymat berish amali	S="dastur"; s+="absd"; s1=s;

+	Satrlarni ulash	s=s+s1
==,!=,<,>,<=,>=	Satrlarni solishtirish amallari	s==s1 s>=s1 s==s1 && s<s1
[]	Indeks berish	s[2]
<<	Oqimga chiqarish	cout<<s;
>>	Oqimdan o'qish	cin>>s1;

### String turidagi satrlarni nusxalash

String turidagi satrlarni nusxalash uchun assign() funksiyasidan foydalaniladi. Assign() funksiyasining umumiy ko'rinishi quyidagicha bo'ladi:

**<s1>.assign(<s2>);**

Assign funksiyasi s2 satrni s1 satrga nusxalash amalini bajaradi.

**Misol:** s2 satrni s1 satrga nusxalash amalini bajaring.

<pre>#include &lt;iostream.h&gt; #include &lt;string.h&gt; using namespace std; int main() { string s1,s2; cin&gt;&gt;s2; s1.assign(s2); // s1=s2 amaliga teng kuchli cout&lt;&lt;s1; return 0; }</pre>	
<i>Kiritish ma'lumotlari</i> S2=dastur	<i>Dastur natijasi</i> Dastur

Bu funksiyaning quyidagi ko'rinishlaridan foydalaniladi.

a) **satr2.assign(satr1);** ko'rinishi. Bu holda funksiya satr1ni to'liq satr2 ga nusxalaydi. Agar satr2 initsializatsiyalangan bo'lsa ham uning oldingi qiymati unutiladi.

b) **satr2.assign(satr1,n,m);** ko'rinishi. Ushbu variantida funksiya satr1ning n- belgisidan keyingi m ta belgisini satr2 ga nusxalaydi.n-belgisi kirmaydi.

c) **satr2.assign(satr1,n);** ko'rinishi. Bu holda satr1 char turli satrli massiv bo'lishi kerak. Funksiya char turli massivni string turiga aylantirib, uning dastlabki

n ta belgisini satr2 ga nusxalaydi. Barcha variantlaridan foydalanish usullari quyidagi dasturda o‘z aksini topgan.

### Satrning biror qismidan nusxa olish funksiyasi

Bu amalni bajarish uchun **satr1.substr(n,m)** funksiyasidan foydalaniladi. Funksiyaning quyidagicha ko‘rinishlari mavjud.

a) **satr1.substr(n,m)**. Bunda **satr1** qismidan nusxa olinayotgan satr bo‘lib, funksiya shu satrning **n**-belgisidan keyingi **m** ta belgisidan nusxa oladi;

b) **satr1.substr(n)** ko‘rinishida foydalanilsa **satr1** satrning **n**-belgisidan oxirigacha nusxa olinadi;

c) **satr1.substr()** ko‘rinishidan foydalanilganda esa **satr1** dan to‘liq nusxa olinadi. Bu holda funksiyaning qiymati qandaydir **string** turli o‘zgaruvchiga berilishi kerak.

### Satrning oxiriga boshqa satrni o‘rnatish

String turidagi satrni qismiga boshqa satrni o‘rnatish uchun **append()** funksiyasidan foydalaniladi. **append()** funksiyasining umumiy ko‘rinishi quyidagicha bo‘ladi.

#### **s1.append(s2,n,m);**

Append funksiyasi **s2** satrni **n**-elementidan **m** ta elementni **s1** satr oxiriga o‘rnatish amalini bajaradi.

*Misol: s2 satrni ikkinchi elementidan uchtasini s1 satrga o‘rnating amalini bajaring.*

<pre>#include &lt;iostream.h&gt; #include &lt;string.h&gt; using namespace std; int main() { string s1,s2; cin&gt;&gt;s1; cin&gt;&gt;s2; s1.append(s2,2,3); cout&lt;&lt;s1; return 0; }</pre>	
<p><i>Kiritish ma'lumotlari</i></p> <p>S1=dastur</p> <p>S2=amal</p>	<p><i>Dastur natijasi</i></p> <p>Dastural</p>

S2 satrning ikkinchi elementida a joylashgan, chunki satr nolinchidan boshlangan.

Bu funksiyani quyidagi ko‘rinishlari mavjud:

a) **satr2.append(satr1)**; ko‘rinishidan foydalanilganda satr2 ning oxiridan satr1 ulanadi. Agar satr2 initsializatsiya qilinmagan bo‘lsa, satr2 o‘zgaruvchi satr1 o‘zgaruvchining qiymatini oladi;

b) **satr2.append(satr1,n,m)**; ko‘rinishidan foydalanilganda satr1ning n – belgisidan keyingi m ta belgisi satr2 ning oxiridan ulanadi;

c) **satr2.append(satr1,n)**; ko‘rinishidan foydalanilganda satr1 char turli massiv bo‘lishi kerak. Funksiya char turli satr1ni string turga aylantirib, uning dastlabki n ta belgisini satr2 oxiridan ulaydi.

Funksiyaning vazifasini o‘zlashtirish uchun quyidagi dastur va uning natijasini tahlil qiling.

```
#include <iostream.h>
#include <string.h>
using namespace std;
int main()
{ int j,i,s,n,m;
  string s1="matnni      ulash",s3,s4="bo‘lakni",s5,s7="bizning
",s2="matnga ";
  char s6[]="assalom"; s2.append(s1);cout<<"s2="<<s2<<"\n";
  s3.append(s1);cout<<"s3="<<s3<<"\n";
s4.append(s1,6,6);cout<<"s4="<<s4<<"\n" ;
  s5.append(s6,4);cout<<"s5="<<s5<<"\n";
  s7.append("maktabdosh",6);cout<<"s7="<<s7<<"\n";
  system("pause");
  return 0;
}
```

Dastur natijasi quyidagicha bo‘ladi

s2=matnga matnni ulash

s3=matnni ulash

s4=bo‘lakni ulash

s5=assa

s7=bizning maktab

### Satr tarkibiga boshqa satrni joylashtirish

String turidagi satrni qismiga boshqa satrni o‘rnatish uchun insert() funksiyasidan foydalaniladi. insert() funksiyasining umumiy ko‘rinishi quyidagicha bo‘ladi.

**satr2.insert(n,satr1,m,k)**

Funksiyaning quyidagi ko‘rinishlaridan foydalaniladi:

a) **satr2.insert (n,satr1)**; ko‘rinishidan foydalanilganda funksiya **satr2** ning **n**-o‘rnidan boshlab **satr1** ni joylashtiradi.

b) **satr2.insert (n,satr1,m,k)**; ko‘rinishidan foydalanilganda funksiya **satr2** ning **n**- belgisidan keyin **satr1** ning **m**-belgisidan keyingi **k**-ta belgisini joylashtiradi.**m** - belgisi kirmaydi.

s) **satr2.insert (n,satr1,m)**; ko‘rinishidan foydalanilganda funksiya **satr2** ning **n**-belgisidan keyin **satr1** ning dastlabki **m** ta belgisini joylashtiradi. Maslan,

```
#include <iostream.h>
#include <string.h>
using namespace std;
int main()
{ int j,i,s,n,m;
  string s1=" matnni joylash",s3,s4="bo‘lakni",s5="tezroq ",s7="bizning
",s2="matnga ";
  char s6[]="bo‘ladi"; s2.insert(6,s1);cout<<"s2="<<s2<<'\n';
s4.insert(6,s1,10,4);cout<<"s4="<<s4<<'\n' ;
s5.insert(3,s6,4);cout<<"s5="<<s5<<'\n';
s7.insert(3," maktabdosh",7);cout<<"s7="<<s7<<'\n';
  system("pause");
  return 0;
}
```

Dastur natijasi quyidagicha bo‘ladi  
s2=matnga matnni joylash  
s4=bo‘lakylasni  
s5=tezbo‘lroq  
s7=biz maktabning

### Satr qismini o‘chirish

String turidagi satrni qismini o‘chirish uchun `erase()` funksiyasidan foydalaniladi. `erase ()` funksiyasining umumiy ko‘rinishi quyidagicha bo‘ladi:

**s1.erase(n,m);**

Erase funksiyasi `s1` satrni `n`-elementidan `m` ta elementni o‘chirish amalini bajaradi.

*Misol: s1 satrni ikkinchi elementidan uchtasini o‘chirish amalini bajaring.*

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



```
using namespace std;
int main()
{ string s1;
cin>>s1;
s1.erase(2,3);
cout<<s1;
return 0;
}
```

*Kiritish ma'lumotlari*  
S1=dastur

*Dastur natijasi*  
Dar

### Matn qismini boshqa matn qismi bilan almashtirish funksiyasi

Bu amalni bajarish uchun **satr1.replace(n,m,satr2,i,j)** funksiyasidan foydalaniladi. Funksiyaning quyidagicha ko'rinishlari mavjud.

a) **satr1.replace(n,m,satr2);** ko'rinishidan foydalanilganda **satr1** ning **n**-belgisidan keyingi **m** ta belgisi **satr2** bilan almashtiriladi.

b) **satr1.replace(n,m,satr2,i,j);** ko'rinishidan foydalanilganda **satr1** ning **n**-belgisidan keyingi **m** ta belgisi **satr2** ning **i**-belgisidan keyingi **j** ta belgisi bilan almashtiriladi.

s) **satr1.replace(n,m,satr2,k);** ko'rinishidan foydalanilganda **satr1**ning **n**-belgisidan keyingi **m** ta belgisi **satr2char** turli massivning dastlabki **k** ta belgisidan tashkil topgan qismini **string** turiga aylantirib almashtiriladi. Quyidagi dasturni tahlil qilish bilan bu fikrlarga ishonch hosil qilish mumkin.

```
#include <iostream.h>
#include <string.h>
using namespace std;
int main()
{ char s4[]="bo'lmaydi";
string s1="matn bo'lagini almashtirish",s2="bugun dars bo'ladi";
string s3="nima";
s1.replace(5,9,"qismini");cout<<"s1="<<s1<<"\n";
s1.replace(4,8,s3,0,2);cout<<"s1="<<s1<<"\n";
s2.replace(11,7,s4,9);cout<<"s2="<<s2<<"\n";
system("pause");
return 0;}
```

Dastur natijasi quyidagicha bo'ladi

```
s1=matn qismini almashtirish
s1=matnni almashtirish
s2=bugun dars bo'lmaydi
```

### Satr tarkibidan satr qismini izlash

String turidagi satrni tarkibidan satr qismini izlash uchun find() funksiyasidan foydalaniladi. find () funksiyasining umumiy ko'inishi quyidagicha bo'ladi:

**s1.find(s2);**

Find funksiyasi s1 satr ichidan s2 satrni izlash amalini bajaradi, agar topilsa, s2 ning boshlanish indeksini saqlaydi aks holda -1 natija qaytaradi.

*Misol: s1 satr tarkibidan s2 satrni izlash amalini bajaring.*

```
#include <iostream.h>
#include <string.h>
using namespace std;
int main()
{ int i;
string s1,s2;
cin>>s1; cin>>s2;
i=s1.find(s2);
cout<<i;
return 0;
}
```

*Kiritish ma'lumotlari*

S1=dastur

S2=tur

*Dastur natijasi*

3

Agar satr tarkibida izlanadigan qism bir necha marta bajarilsa, oxirgi qismning indeksini aniqlash uchun **rfind()** funksiyasidan foydalaniladi. Rfind() funksiyasini ishlash jarayoni find() funksiyasini ishlash jarayoni bilan bir xil.

### Satrlarni solishtirish funksiyalari

Satrlarni solishtirish funksiyalari satrdagi belgilarning kodlari bo'yicha solishtiradi. Shuning uchun dastlab klavitudagi ko'p foydalaniladigan belgilarning kodlarini aniqlash dasturini va uning natijasini keltiramiz.

```
#include <iostream.h>
#include <string.h>
using namespace std;
```

```

int main()
{ int x,y,z,n,m,t,i;
  char
a[40]="ABCDEFGHJKLMNOPQRSTUVWXYZ",b[30]="abcdefghijklmnopq
rstuvwxyz";
  char c[40]="0123456789 !@#$%&*()+-=:;?^<>\/,.;:";
  for(i=0;i<=32;i++) {x=(int)a[i];y=(int)b[i];z=(int)c[i];
  if(i<=25){cout<<" "<<a[i]<<" ning kodi="<<x<<" "<<b[i]<<" ning
kodi="<<y;
  cout<<" "<<c[i]<<" ning kodi="<<z<<"\n';} else cout<<" "<<c[i]<<"
ning kodi="<<z<<"\n';}
  system("pause");
  return 0;
}

```

Dastur natijasi quyidagicha bo'ladi

<b>A</b> ning kodi=65	<b>a</b> ning kodi=97	<b>0</b> ning kodi=48	<b>^</b> ning kodi=94
<b>B</b> ning kodi=66	<b>b</b> ning kodi=98	<b>1</b> ning kodi=49	<b>&lt;</b> ning kodi=60
<b>C</b> ning kodi=67	<b>c</b> ning kodi=99	<b>2</b> ning kodi=50	<b>&gt;</b> ning kodi=62
<b>D</b> ning kodi=68	<b>d</b> ning kodi=100	<b>3</b> ning kodi=51	<b>/</b> ning kodi=47
<b>E</b> ning kodi=69	<b>e</b> ning kodi=101	<b>4</b> ning kodi=52	<b>,</b> ning kodi=44
<b>F</b> ning kodi=70	<b>f</b> ning kodi=102	<b>5</b> ning kodi=53	<b>.</b> ning kodi=46
<b>G</b> ning kodi=71	<b>g</b> ning kodi=103	<b>6</b> ning kodi=54	<b>:</b> ning kodi=58
<b>H</b> ning kodi=72	<b>h</b> ning kodi=104	<b>7</b> ning kodi=55	
<b>I</b> ning kodi=73	<b>i</b> ning kodi=105	<b>8</b> ning kodi=56	
<b>J</b> ning kodi=74	<b>j</b> ning kodi=106	<b>9</b> ning kodi=57	
<b>K</b> ning kodi=75	<b>k</b> ning kodi=107	<b>probel</b> ning kodi=32	
<b>L</b> ning kodi=76	<b>l</b> ning kodi=108	<b>!</b> ning kodi=33	
<b>M</b> ning kodi=77	<b>m</b> ning kodi=109	<b>@</b> ning kodi=64	
<b>N</b> ning kodi=78	<b>n</b> ning kodi=110	<b>#</b> ning kodi=35	
<b>O</b> ning kodi=79	<b>o</b> ning kodi=111	<b>\$</b> ning kodi=36	
<b>P</b> ning kodi=80	<b>p</b> ning kodi=112	<b>%</b> ning kodi=37	
<b>Q</b> ning kodi=81	<b>q</b> ning kodi=113	<b>&amp;</b> ning kodi=38	
<b>R</b> ning kodi=82	<b>r</b> ning kodi=114	<b>*</b> ning kodi=42	
<b>S</b> ning kodi=83	<b>s</b> ning kodi=115	<b>(</b> ning kodi=40	
<b>T</b> ning kodi=84	<b>t</b> ning kodi=116	<b>)</b> ning kodi=41	
<b>U</b> ning kodi=85	<b>u</b> ning kodi=117	<b>+</b> ning kodi=43	
<b>V</b> ning kodi=86	<b>v</b> ning kodi=118	<b>-</b> ning kodi=45	

<b>W</b> ning kodi=87	<b>w</b> ning kodi=119	<b>=</b> ning kodi=61
<b>X</b> ning kodi=88	<b>x</b> ning kodi=120	<b>;</b> ning kodi=59
<b>Y</b> ning kodi=89	<b>y</b> ning kodi=121	<b>,</b> ning kodi=44
<b>Z</b> ning kodi=90	<b>z</b> ning kodi=122	<b>?</b> ning kodi=63

Satrlarni solishtirishda **compare** funksiyasining quyidagi bir nechta ko‘rinishlaridan foydalaniladi.

**a) s=satr1.compare (satr2);**

Bunda **satr1** va **satr2** lar solishtiriladi. Agar

**satr1 > satr2** bo‘lsa **s > 0**

**satr1 = satr2** bo‘lsa **s = 0**

**satr1 < satr2** bo‘lsa **s < 0**

bo‘ladi. Satrlarni solishtirishda, ularning mos nomerlaridagi belgilarining **ASCII** kodlari solishtirilib ketilaveradi. Qaysidir nomerli belgilari bir-biridan farq qilsa va ulardan qaysi birining kodi katta bo‘lsa, o‘sha satr katta hisoblanadi.

**b) s=satr1.compare (n,m,satr2);** ko‘rinishidan foydalanilganda **satr1** ning **n**-nomeridan keyingi **m** ta belgilari ketma-ketligi **satr2** bilan solishtiriladi.

**c) s=satr1.compare (n,m,satr2,i,j);** ko‘rinishidan foydalanilganda **satr1** ning **n**-belgisidan keyingi **m** ta belgilari ketma-ketligi **satr2** ning **i**-belgisidan keyingi **j** ta belgilari ketma-ketligi bilan solishtiriladi.

Solishtirish funksiyasining imkoniyatlarini quyidagi dastur misolida namoyish qilamiz.

```
#include <iostream.h>
#include <string.h>
using namespace std;
int main()
{ int x,y,z,n,m,t,s5,s6,s7,s8;
  string s1="daraja ko'rsatkichida",s2="ABCDEF",s3="ABCD";
  string s4="daraja ko'rsat",s9="ab",s10="aa3-";
  s5=s2.compare(s3); cout<<"s5="<<s5;
  s6=s2.compare(s1); cout<<" s6="<<s6;
  s8=s9.compare(s10); cout<<" s8="<<s8;
  s7=s1.compare(s4); cout<<" s7="<<s7<<"\n";
  x=s1.compare(0,14,s4); cout<<" x="<<x;
  y=s2.compare(0,5,s3); cout<<" y="<<y;
  z=s1.compare(1,4,s4,1,4); cout<<" z="<<z;
```

```

n=s1.compare(3,4,s4,2,4); cout<<" n="<<n;
m=s1.compare(3,5,s4,3,4); cout<<" m="<<m<<"\n";
system("pause");
return 0;
}

```

Dastur natijasi quyidagicha bo'ladi

```

s5=2   s6=-1   s8=1   s7=7
x=0   y=1   z=0   n=-1   m=1

```

### Satr xossalarini aniqlash funksiyalari

Satrning xossalarini aniqlash uchun quyidagi funksiyalardan foydalaniladi.

**x=satr.size();** funksiyasi **satr** ning o'lchamini aniqlaydi.

**x=satr.length();** funksiyasi **satr** ning belgilari sonini aniqlaydi.

**x=satr.max\_size();** funksiyasi **satr** ning maksimal o'lchamini aniqlaydi.

**x=satr.capacity();** funksiyasi **satr** ning xotirada egallagan joyi hajmini aniqlaydi.

**x=satr.empty();** funksiyasi qiymati **bool** turli bo'lib, agar satr bo'sh bo'lsa, funksiya qiymati **true**, aks holda **false** bo'ladi. Quyidagi dasturni va uning natijalarini tahlil qiling.

```

#include <iostream.h>
#include <string.h>
using namespace std;
int main()
{ int z,n,m,t,s6,s7,s8;bool x,y;float s5;
string s1="daraja ko'rsatkichida",s2="ABCDEF",s3="ABCD";
string s4="daraja ko'rsat",s9="ab87,/+"|",s10="";
s5=s9.size(); cout<<"s5="<<s5;
s6=s9.length(); cout<<" s6="<<s6;
s8=s10.max_size(); cout<<" s8="<<s8;
s7=s9.capacity(); cout<<" s7="<<s7<<"\n";
x=s9.empty(); cout<<" x="<<x;
y=s10.empty(); cout<<" y="<<y<<"\n";
system("pause");
return 0;
}

```

Dastur natijasi quyidagicha bo'ladi

```

s5=8   s6=8   s8=1073741820   s7=8
x=0   y=1

```

## Nazariy savollar

1. String turi deganda nimani tushunasiz?
2. String turidagi o'zgaruvchilarni e'lon qilish
3. String turidagi o'zgaruvchilarni elementlariga murojat qilishni tushuntirib bering?
4. String turdagi satrlar ustida qanday amallar bajariladi va ularning yozilishi?
5. String turdagi satrlarni nusxalash funksiyasi va uning umumiy ko'rinishi?
6. Satrning qismiga boshqa satrni o'rnatish funksiyasi va uning umumiy ko'rinishi?
7. Satr qismini o'chirish funksiyasi va uning umumiy ko'rinishi?
8. Satr tarkibidan satr qismini izlash funksiyasi va uning umumiy ko'rinishi?
9. string turli o'zgaruvchilar bilan satrli massivlar o'rtasida nima farq bor?
10. string turli matnlarni qayta ishlashga mo'ljallangan qanday funksiyalarini bilasiz? Ularni tushuntiring.
11. 1) "gdf3546^.,;'", 2) "9876543 %\$#@!" satrlar uzunligini aniqlang.
12. Satrning o'zini yoki uning biror qismini o'chirish funksiyasi qaysi va u qanday bajariladi?
13. Satrni songa va sonni satrga aylantirish funksiyalarini yozib, tushuntiring.

## Mustaqil ishlash uchun topshiriqlar

1. Nuqta bilan tugaydigan satr berilgan. Satrda nechta so'z borligini hisoblab chiqing.
2. Ingliz matnidan iborat satr berilgan. b harfi bilan boshlanuvchi so'z nechtaligini toping.
3. Satr berilgan. Unda r, k, t harflari qanchaligini hisoblab chiqing.
4. Satr berilgan. Unda \*(yulduzcha), ;(nuqta vergul), :(ikki nuqta) belgilari qanchaligini hisoblab chiqing.
5. Matndan iborat satr berilgan. Eng qisqa va eng uzun so'z uzunliklarini toping.
6. Orasida ikki nuqtasi bo'lgan belgili satr berilgan. Ungacha bo'lgan belgilar qanchaligini aniqlang.
7. Nuqta bilan tugaydigan, matndan tashkil topgan satr berilgan. Uchta harfdan iborat so'zni ekranga chiqaring.
8. Berilgan satrdagi barcha na qisman satrni nad qisman satr bilan almashtirng.
9. Satr berilgan. Unda abc qisman satr necha marta uchrashini aniqlang.

10. *Satr berilgan. Uning oxirgi soʻzidagi k harfi migʻdorini hisoblab chiqing.*
11. *Satr berilgan. Unda har xil belgilar necha marta uchrashini hisoblab chiqing. Ularni ekranga chiqaring.*
12. *Satr berilgan. Berilgan ikkita gapdagi oʻchraydigan bir xil soʻzni chop eting.*
13. *Orasida bitta ochilgan qavs va bitta yopilgan qavs mavjud boʻlgan belgili satr berilgan. Bu qavslar orasidagi barcha belgilarni ekranga chiqaring.*
14. *Lotin harflari va raqamlardan tashkil topgan satr mavjud. ...*
15. *Ikki nuqta bilan tugovchi va nuqta vergul bilan ajarilgan soʻzlar toʻplami berilgan. a harfi bilan tugovchi soʻzlar qanchaligini aniqlang.*
16. *Satr berilgan. Tarkibida kamida bitta k harfi boʻlgan soʻzni koʻrsating.*
17. *Satr berilgan. Boshlanishi va tugashi bir xil harfdan iborat soʻzni toping.*
18. *Satrdagi barcha ikki nuqtalarni nuqta vergul bilan almashtiring. Almashishlar migʻdorini hisoblang.*
19. *Satrdagi ikki nuqta belgilarini oʻchiring va oʻchirilgan belgilar migʻdorini hisoblang.*
20. *Satrdagi soʻzlar orasidagi bOʻsh joy oʻrniga vergul va bOʻsh joyni qoʻying.*
21. *Qavsga olingan belgili satr qismini oʻchiring (qavslar bilan birgalikda).*
22. *Berilgan soʻz satrda necha marta uchrashini aniqlang.*
23. *Satrdagi bitta nuqtali vergul mavjud. Nuqtali vergulgacha boʻlgan va undan keyingi belgilar migʻdorini hisoblang.*
24. *Satr berilgan. Birinchi  $n/2$  belgilar orasida uchraydigan barcha ikki nuqtalarni nuqta bilan almashtiring va  $n/2$  dan keyingi belgilar orasida uchraydigan barcha undov belgilarni nuqta almashtirib uni oʻzgartiring.*
25. *Satr bitta soʻzdan tashkil topgan. Uni chapdan oʻngga va oʻngdan chapga qarab oʻqiganda bir xil boʻlishini (yaʼni u polindrom hisoblanishini) tekshiring.*
26. *Qoʻlyozmadagi soʻzlarning har biri teskarisiga yozilib shifrlangan. Xabarni oching (rasshifrovka qiling).*
27. *Berilgan satrda ochilgan qavslar soni yopilgan qavslar soni bilan bir hilligini tekshiring.*
28. *200 belgidan oshib ketmaydigan ixtiyoriy ruscha matndan tashkil topgan satr berilgan. Bu matnda qaysi belgi necha marta uchrashini yozing. Javob*

grammatik jihatdan to'g'ri bo'lishi kerak, masalan «a – 25 marta», «k – 3 marta» va h.k.

29. Ingliz so'zlardan iborat berilgan massivni alfavit bo'yicha tartiblang.

30. Ikkita A va B satrlar berilgan. A satrdagi harflardan B satrni tuzish mumkinligini (harflarni o'rnini almashtirib ishlatish mumkin, lekin ularni bir martadan ortiq ishlatish mumkin emas) tekshiradigan dastur tuzing. Masalan, A:INTEGRAL, B:AGENT – tuzish mumkin; B:AGENT – tuzish mumkin emas.

### 5.3 C++ DASTURLASH TILIDA FAYLLAR VA ULARDAN FOYDALANISH

*Reja:*

1. Fayllarni e'lon qilish;
2. Fayllarga murojat qilish;
3. Fayllar ustida amallar bajarish.
4. Fayldan ma'lumot o'qish
5. Fayl tarkibidagi ma'lumotlarni o'chirish

C++ dasturlash tili tarkibida fayllar bilan ishlash uchun barcha turdagi imkoniyatlar e'tiborga olingan. Dasturlash tillarida fayllar dastur tarkibidagi o'zgaruvchilar qabul qiladigan qiymatlarni saqlash uchun ishlatiladi. Dasturlash asoslari fani rivojlanib borishi bilan dasturlash tillari tarkibida fayllar ham rivojlanib bormoqda. Dasturlash asoslari tarkibida murakkab masalalarni hal etishda o'zgaruvchilarni qiymatlari soni ko'p bo'lsa, bunday holatlarda qo'lda kiritish qulay hisoblanmaydi. o'zgaruvchilarning qiymatlari soni ko'p bo'lgan holatlarda qiymatlar fayllarda saqlansa dasturning o'zi qiymatlarni tezkor holda fayllardan qabul qilib oladi. Kompyuterning ma'lum bir joyida qiymatlar fayllar asosida saqlanadi.

*Ta'rif:* Kompyuterda alohida nom bilan saqlanadigan ma'lum bir turga mansub bo'lgan ma'lumotlar majmuasi fayl deb nomlanadi.

C++ dasturlash tili tarkibida fayllarga murojat qilish uchun fayl nomiga to'g'ridan to'g'ri murojat qilib bo'lmaydi, fayllarga murojat qilish uchun fayllarni dastur bilan bog'lash uchun alohida o'zgaruvchi qabul qilinadi.

#### Fayllarni e'lon qilish

C++ dasturlash tilida fayllar bilan ishlashda **.txt** kengaytmali fayllardan foydalanish maqsadga muvofiq bo'ladi. Dastur tarkibida fayllarga murojat qilish uchun albatta oldin **.txt** kengaytmali faylni yaratib alohida joyga saqlab quyish



maqsadga muvofiq bo'ladi. Dasturlash tilida ishlatiladigan fayllar nomi ikki turda bo'ladi.

- 1) Fizik nomi;
- 2) Mantiqiy nomi.

Faylning fizik nomi kompyuterda .txt kengaytmali nom bilan saqlangan nomi hisoblanadi.

Faylning mantiqiy nomi esa dastur tarkibida faylning fizik nomi bilan bog'lashga xizmat qiladigan nomi hisoblanadi. C++ dasturlash tilida fayllarni e'lon qilishdan oldin albatta **#include <fstream>** kutubxonasini chaqirish kerak.

C++ dasturlash tilida fayllarning asosiy nomiga to'g'ridan to'g'ri murojat qilib bo'lmaydi. Fayllarga murojat qilish uchun faylning fizik nomiga mantiqiy nomini bog'lab murojat qilish mumkin. Fayllar mantiqiy nomini fizik nomiga bog'lagandan So'ng dastur tarkibida faylga uning nomi bilan murojat qilinadi. Dastur tarkibining ixtiyoriy joyida faylga murojat qilish uchun albatta uning mantiqiy nomi bilan murojat qilinadi.

C++ dasturlash tilida fayllarni e'lon qilish va ularni mantiqiy nomi bilan bog'lash uchun ixtiyoriy o'zgaruvchini mantiqiy nom sifatida qabul qilinadi.

C++ dasturlash tilida fayllar bir nechta holatlar bo'yicha e'lon qilinadi.

- 1)Fayllarga ma'lumot yozish uchun ochish bo'yicha e'lon qilish;
- 2)Fayllardan ma'lumot o'qish uchun ochish bo'yicha e'lon qilish;

C++ dasturlash tilida fayllarga ma'lumot yozish uchun ochish bo'yicha e'lon qilishning umumiy ko'rinishi quyidagicha bo'ladi:

**ofstream <faylni mantiqiy nomi>;**

Ofstream kalit so'zi orqali fayl e'lon qilinganda, e'lon qilingan fayl ustida faqatgina ma'lumot yozish amali bajariladi. Ya'ni dastur natijasi biror bir faylga saqlash uchun faylni yozish bo'yicha e'lon qilinadi.

C++ dasturlash tilida fayllardan ma'lumot o'qish uchun ochish bo'yicha e'lon qilishning umumiy ko'rinishi quyidagicha bo'ladi:

**ifstream <faylni mantiqiy nomi>;**

Ifstream kalit so'zi orqali fayl e'lon qilinganda, e'lon qilingan fayl ustida faqatgina ma'lumot o'qish amali bajariladi. Ya'ni dastur tarkibiga qabul qilinadigan qiymatlar biror bir fayldan o'qish jarayonini amalga oshirish uchun fayl o'qish bo'yicha e'lon qilinadi:

Agar faylni o'zidan ma'lumot o'qib shu faylni o'ziga ma'lumot yozish talab etilsa, albatta, birinchi fayl o'qish uchun ochilib ma'lumot o'qiladi va fayl yopilib keyin fayl yozish uchun ochilib ma'lumot yozilishi kerak.

Fayllarga murojat qilib bo'lgandan so'ng, albatta, fayllar yopish kerak. C++ dasturlash tilida fayllarni yozishning umumiy ko'rinishi quyidagicha bo'ladi:

**<faylni mantiqiy nomi>.close();**

Fayllarni yopish uchun, albatta, mantiqiy nomi dan so'ng .close() kalit so'zi yoziladi.

Ikkita f va g o'zgaruvchilar fayllarning mantiqiy nomi sifatida qabul qilinib, ularni mos ravishda yozish va o'qish bo'yicha quyidagi ko'rinishda e'lon qilinadi:

```
#include <iostream.h>
#include <fstream>
using namespace std;
int main()
{
    ofstream f; // faylni yozish uchun ochish
    ifstream g; // faylni o'qish uchun ochish
    f.close(); // f fayl yopildi
    g.close(); // g fayl yopildi
    return 0;
}
```

C++ dasturlash tilida fayllarni o'qish yoki yozish uchun alohida e'lon qilinadi. Yuqoridagi C++ dasturlash tilida yozilgan dasturga e'tibor bersak, f fayl yozish uchun, g fayl o'qish uchun e'lon qilindi va f , g fayllarni yopish jarayonlari qarab o'tildi.

### **Fayllarga murojat qilish**

C++ dasturlash tilida fayllar bilan ishlashda birinchi o'rinda dasturingiz va foydalaniladigan fayllarni, albatta, bitta papkaga saqalab oling. Dasturlash tillarida fayllarga murojat qilish uchun oldin uning mantiqiy va fizik nomlarini bog'lash kerak. C++ dasturlash tilida fayllarning mantiqiy nomi bog'langandan so'ng faylga uning mantiqiy nomi bilan murojat qilinadi. C++ dasturlash tilida fayllarni mantiqiy nomini fizik nomiga bog'lashning umumiy ko'rinishi quyidagicha bo'ladi:

**<mantiqiy nomi>.open("<fizik nomi adresi bilan>");**

Faylning mantiqiy nomi fizik nomiga bog'lanish vaqtida fizik nomi adresi bilan to'liq yozilish kerak aks holda fayl bog'lanmaydi. Agar yaratilgan dastur turgan papkada fayl yaratilgan bo'lsa, adresi ko'rsatilmadan uning nomi ko'rsatilishi kifoya.

**Misol:** *output.txt* va *input.txt* fayllarini yarating va ularga mos ravishda *f* va *g* mantiqiy nomlari bilan bog'lang.

```
#include <iostream.h>
#include <fstream>
using namespace std;
int main()
{
    ofstream f;
    f.open("output.txt");
    ifstream g;
    g.open("input.txt");
return 0;
}
```

Yuqoridagi dasturga e'tibor qaratsak, *output.txt* va *input.txt* fayllar *f* va *g* fayllar bilan bog'landi. *f* va *g* fayllar mos ravishda yozish va o'qish uchun tayyor holatga keltirildi. C++ dasturlash tilida fayllarga murojat qilish uchun dasturning ixtiyoriy joyida bir qadam oldinda mantiqiy nomi bilan bog'lash kerak.

### **Fayllar ustida amallar bajarish**

C++ dasturlash tillari tarkibida fayllar ustida amallar bajarish uchun yuqoridagi holatlar bo'yicha oldin e'lon qilinishi va mantiqiy nomi bilan bog'lashi shart. Fayllar ustida quyidagi amallarni bajarishimiz mumkin:

- 1)Faylga ma'lumot yozish;
- 2)Fayldan ma'lumot o'qish;
- 3)Fayl tarkibidagi ma'lumotlarni o'chirish.

C++ dasturlash tilida fayllarga ma'lumot yozishning ikki turi mavjud, birinchi turi bo'yicha ma'lumot yozilganda eski qiymatlar uchirilib ustiga yangi ma'lumot yoziladi. Ikkinchi tur bo'yicha faylga ma'lumot yozilsa, unda ma'lumot faylning oxiriga borib yozishni bajaradi. Ikki tur bo'yicha ham faylga ma'lumot o'zgaruvchi yordamida yoziladi.

C++ dasturlash tilida birinchi tur bo'yicha faylga ma'lumot yozish uchun `ofstream <mantiqiy nom>; <mantiqiy nom>.open("fizik nom");` buyruqlari oldin yozilishi

shart undan so‘ng uning tarkibiga ma’lumot yozish mumkin. C++ dasturlash tilida faylga birinchi tur bo‘yicha ma’lumot yozishning umumiy ko‘rinishi quyidagicha bo‘ladi:

**<faylni mantiqiy nomi> << <o‘zgaruvchi>;**

Fayl tarkibiga ma’lumot yozilganda uning oldingi ma’lumotlari o‘chirilib, yangi ma’lumotlar yoziladi. Faylga birinchi tur bo‘yicha ma’lumot yozish tartibi xuddi cout funksiyasi kabi bajariladi.

**Misol:** *n ta elementli massiv berilgan massiv elementlari ichidan toqlarini f faylga juftlarini g faylga yozing.*

```
#include <iostream.h>
#include <fstream>
using namespace std;
int main()
{ int n; int a[90];
  cout<<"n="; cin>>n;
  cout<<"massiv elementlarini kiriting \n";
  for(int i=0;i<n;i++) cin>>a[i];
  ofstream f;
  f.open("toq.txt");
  ofstream g;
  g.open("juft.txt");
  for(int i=0;i<n;i++)
  if(a[i]%2==1) f<<a[i];
  else g<<a[i];
  f.close();
  g.close();
return 0;
}
```

*Kiritishga ma’lumot*

*n=5*

*massiv elementlarini kiriting*

*5 6 9 8 1*

*Dastur natijasi*

*f faylda*

*5 9 1*

*g faylda*

*6 8*

Yuqoridagi dasturda massiv elementlarining toqlarini f fayl orqali toq.txt faylga, juft elementlarini esa g o'zgaruvchi orqali juft.txt faylga joylashtirdi.

C++ dasturlash tilida ikkinchi tur bo'yicha faylga ma'lumot yozish uchun ofstream <mantiqiy nom>; <mantiqiy nom> .open ("fizik nom", ios::app); buyruqlari oldin yozilishi shart, undan so'ng uning tarkibiga ma'lumot yozish mumkin. Ikkinchi tur bo'yicha ma'lumot yozish degani fayl oxiriga ma'lumot yozish tushuniladi. Faylning oxiriga ma'lumot yozish uchun e'lon qilish birinchi turdek e'lon qilinadi faqatgina ios::app xossasi qo'shimcha sifatida qo'shiladi. C++ dasturlash tilida faylga ikkinchi tur bo'yicha ma'lumot yozishning umumiy ko'rinishi quyidagicha bo'ladi:

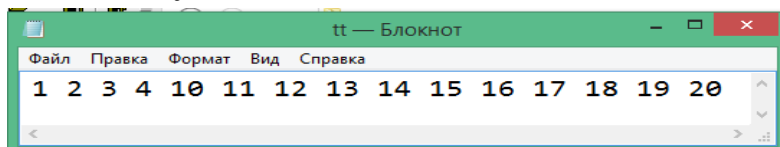
**<faylni mantiqiy nomi> << <o'zgaruvchi>;**

Fayl tarkibiga ikkinchi tur bo'yicha ma'lumot yozish uchun <mantiqiy nom> .open ("fizik nom", ios::app); buyrug'i orqali e'lon qilinadi. Ikkinchi tur bo'yicha yozish uchun e'lon qilinganda kursor fayl oxiriga boradi va yangi yozilgan ma'lumot fayl oxiriga borib yoziladi.

**Misol:** To'rtta 1 2 3 4 elementlari mavjud bo'lgan f fayl oxiriga 10 dan 20 gacha bo'lgan sonlarni yozing.

```
#include <iostream.h>
#include <fstream>
using namespace std;
int main()
{ ofstream f;
  f.open("tt.txt", ios::app);
  for(int i=10;i<=20;i++)
  f<<" "<<i;
  f.close();
return 0;
}
```

*Dastur natijasi*



Yuqoridagi dasturda tt.txt faylida 1 2 3 4 sonlar mavjud edi, 10 dan 20 gacha bo'lgan sonlar 1 2 3 4 sonlaridan keyin tt.txt fayliga joylashtirildi.

**Fayldan ma'lumot o'qish**

**Fayldan ma'lumot o'qish** jarayonini C++ dasturlash tilida amalga oshirish uchun fayl o'qish uchun ochish bo'yicha e'lon qilish kerak. Mavzuni fayllarga murojat qilish qismi bo'yicha fayllar o'qish uchun ochish bo'yicha e'lon qilinadi va mantiqiy nomi fizik nomi bilan bog'lanib undagi ma'lumotlar o'qiladi. Fayldan ma'lumotlar faqat o'zgaruvchilarga o'qiladi. Fayldan ma'lumotlarni o'qish jarayoni xuddi cin funksiyasi kabi bajariladi.

C++ dasturlash tilida fayldan ma'lumot o'qish uchun ifstream <mantiqiy nom>; <mantiqiy nom>.open("fizik nom"); buyruqlari oldin yozilishi shart undan so'ng fayl tarkibidan ma'lumot o'qish mumkin. C++ dasturlash tilida fayldan ma'lumot o'qishning umumiy ko'rinishi quyidagicha bo'ladi:

**<faylni mantiqiy nomi> >> <o'zgaruvchi>;**

Fayl o'qish uchun ochilgandan so'ng uning elementlarini biror bir o'zgaruvchiga o'qib olish imkoniyati mavjud.

**Misol: 1 dan 10 gacha sonlar yozilgan tt.txt faylidan uning elementlarini ko'chirib olib ikkiga ko'paytirib ekranga chiqaring.**

```
#include <iostream.h>
#include <fstream>
using namespace std;
int main()
{ ifstream f; int a;
  f.open("tt.txt");
  for(int i=1;i<=10;i++)
  { f>>a;
    cout<<2*a<<" ";
  }
  f.close();
return 0;
}
```

*Dastur natijasi*

2 4 6 8 10 12 14 16 18 20

**Misol2:** 1 dan 20 gacha sonlar yozilgan f fayli elementlari orasidan juftlarini ikkiga ko'paytirib g faylga yozing.

Masalani yechish uchun f faylni tt.txt, g faylni esa th.txt faylga bog'laymiz.

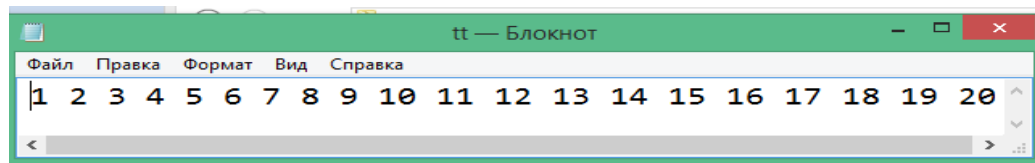
```
#include <iostream.h>
#include <fstream>
using namespace std;
```

```

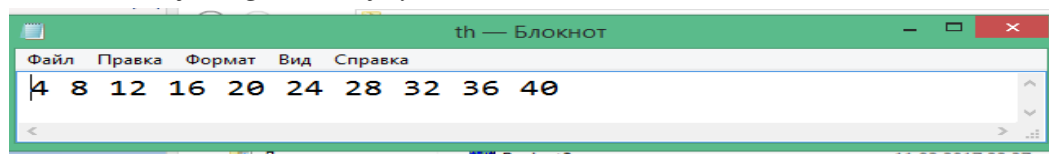
int main()
{ ifstream f; int a;
  f.open("tt.txt");
  ofstream g;
  g.open("th.txt");
  for(int i=1;i<=20;i++)
  { f>>a;
    if(a%2==0) g<<2*a<<" ";
  }
  f.close();
  g.close();
return 0;
}

```

*Kiritishga ma'lumot f(tt.txt) fayl*



*Dastur natijasi g(th.txt) fayl*



C++ dasturlash tilida fayldan ma'lumotlar yuqoridagi holatlar bo'yicha o'qiladi va qayta ishlanadi.

### **Fayl tarkibidagi ma'lumotlarni o'chirish**

**Fayl tarkibidagi ma'lumotlarni o'chirish** uchun C++ dasturlash tili tarkibida imkoniyati mavjud. Ma'lumot yozilgan faylni tozalash yoki uning tarkibidagi elementlarni o'chirish uchun C++ dasturlash tilida fayllarni uchirish buyrug'i mavjud. C++ dasturlash tilida tuzilgan dastur tarkibida foydalaniladigan fayllarni tozalash uchun fayl tozalash sifatida e'lon qilinishi shart. Fayl elementlarini uchirish sifatida e'lon qilishning umumiy ko'rinishi quyidagicha bo'ladi:

**<mantiqiy nom>.open("fizik nom",ios::trunc);**

C++ dasturlash tilida fayl elementlarini o'chirish uchun ofstream <mantiqiy nom>; <mantiqiy nom>.open("fizik nom", ios::trunc); buyruqlari oldin yozilishi shart, undan So'ng fayl tarkibidagi elementlarni o'chirish mumkin. Faylni

o‘chirish uchun e‘lon qilish xuddi yozish uchun e‘lon qilishdek bo‘ladi faqatgina ios::trunc xossasini qo‘shib yozish kerak. C++ dasturlash tilida fayl elementlarini o‘chirishning umumiy ko‘rinishi quyidagicha bo‘ladi:

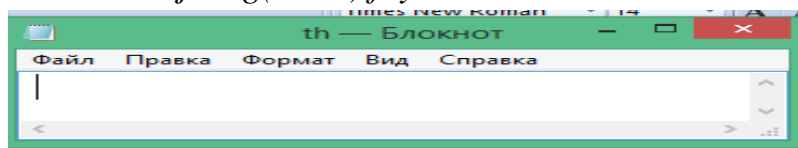
**<faylni mantiqiy nomi>.clear();**

Fayl tarkibidagi ma‘lumotlarni uchirish uchun faylni yuqoridagi buyruq asosida mantiqiy nomi bilan bog‘lab, shu mantiqiy nom yordamida fayl elementlari o‘chirish jarayoni bajariladi.

*Misol: 1 dan 10 gacha sonlar yozilgan th.txt fayl elementlarini o‘chiring.*

```
#include <iostream.h>
#include <fstream>
using namespace std;
int main()
{ ofstream f;
  f.open("th.txt",ios::trunc);
  f.clear();
  f.close();
return 0;
}
```

*Dastur natijasi g(th.txt) fayli*



Yuqoridagi dastur asosida th.txt fayli tozalandi.

C++ dasturlash tili tarkibida fayllardan foydalanish jarayonlari yuqoridagi holatlar bo‘yicha amalga oshiriladi. Fayldan ma‘lumot o‘qish, faylga ma‘lumot yozish va fayldan ma‘lumot o‘chirish jarayonlari asosida fayllarga tegishli masalalarni hal etish imkoniyati mavjud.

### **Nazariy savollar**

1. Fayl deb nimaga aytiladi.
2. Dasturlashda qanday kengaytmali fayllardan foydalaniladi.
3. Dasturlashda fayl nomi qanday qismlardan tashkil topgan.
4. Fayllarni e‘lon qilishda qaysi kutubxonaga murojat qilish kerak.
5. Fayllarni e‘lon qilishning qanday turlari mavjud.
6. Fayllarni yozish uchun e‘lon qilishning qanday turlari mavjud?
7. Fayllarni yozish uchun e‘lon qilishning umumiy ko‘rinishini ayting?
8. Fayllarni o‘qish uchun e‘lon qilishning umumiy ko‘rinishini ayting?



9. Fayllar elementlarini o'chirish uchun e'lon qilishning umumiy ko'rinishini ayting?
10. Fayllarni yopish komandasining umumiy ko'rinishini ayting?
11. Fayllarni fizik va mantiqiy nomlarini bog'lashning umumiy ko'rinishi?
12. Faylga ma'lumot yozishning ikki turdagi komandasining umumiy ko'rinishi?
13. Fayldan ma'lumot o'qish komandasining umumiy ko'rinishi?
14. Fayldan ma'lumot o'chirish komandasining umumiy ko'rinishi?

### **Mustaqil ishlash uchun topshiriqlar.**

1. Elementlari alohida so'z hisoblanuvchi turlashgan fayl berilgan. "t" harfi bilan boshlanuvchi hamma so'zlarni chop qiling. Bunda ikki variantni qarang:

- a) ma'lumki, mavjud faylga 30 ta so'z yozilgan;
- b) mavjud fayl o'lchami ma'lum emas.

2. Elementlari sonli bo'lgan mavjud turlashgan faylni berilgan son bilan almashtiring (yangi qiymat klaviaturadan kiritiladi):

- a) birinchi elementni;
- b) beshinchi elementni;
- v) k-nchi elementni;
- g) oxirgi elementni.

3. Elementlari alohida so'zlar bo'lgan mavjud turlashgan faylni berilgan qiymat bilan almashtiring (yangi so'z klaviaturadan kiritiladi):

- a) birinchi elementni;
- b) uchinchi elementni;
- v) s-nchi elementni;
- g) oxirgi elementni.

4. Turlashgan fayl elementlari alohida so'z hisoblanadi. Uning barcha elementlarini boshqa so'z bilan almashtiring. Bunda ikkita variantni qarang:

- a) ma'lumki, mavjud faylga 12 ta so'z yozilgan;
- b) mavjud fayl hajmi ma'lum emas.

5. Sonli turlashgan fayl mavjud. Uning tartib nomeri 3 ga karrali bo'lgan hamma elementlarini almashtiring. Yangi qiymat klaviaturadan kiritiladi. Bunda ikkita variantni qarang:

- a) ma'lumki, mavjud faylga 20 ta son yozilgan;
- b) mavjud fayl hajmi ma'lum emas.

6. Butun sonli turlashgan fayl mavjud. Uning hamma juft elementlarini nol bilan almashtiring. Bunda ikkita variantni qarang:

- a) ma'lumki, mavjud faylga 13 ta son yozilgan;
- b) mavjud fayl hajmi ma'lum emas.

7. Elementlari alohida soʻzlardan iborat turlashgan fayl mavjud. Agar soʻz “k” harfi bilan boshlansa, uni “K” harfi bilan almashtiring.

8. Sonli turlashgan fayl mavjud. Toping (hamma holatlarda fayl hajmi maʼlum emas):

a) faylning birinchi va ikkinchi sonlar yigʻindisini;

b) faylning k–nchi va q–nchi sonlari yigʻindisi;

v) faylning birinchi va oxirgi sonlari koʻpaytmasini;

g) faylning hamma sonlarining yigʻindisini;

d) fayldagi sonlar miqdorini;

e) a sonidan oshib ketmaydigan fayldagi sonlar miqdorini;

j) faylning musbat sonlarining oʻrta arifmetigini;

i) mavjud fayldagi maksimal sonni;

k) fayldagi minimal sonning tartib nomeri (agar bunday fayllar bir nechta boʻlsa, ularning birinchisini nomerini toping).

9. Elementlari alohida soʻzlardan iborat turlashgan fayl mavjud. Toping (hamma holatlarda fayl hajmi maʼlum emas):

a) “m” harfi bilan boshlanuvchi soʻzlar sonini;

b) “l” harfi bilan boshlanuvchi barcha soʻzlar va ularning birinchisini tartib nomerini chop qiling;

v) tartib nomeri juft boʻlgan soʻzlar va ulardan soʻz tuzing;

g) eng uzun soʻzni;

d) eng qisqa soʻzni.

10. Turlashgan faylga Shaxar nomi va uning aholisi soni yozilgan. Har bir Shaxarning aholisini 5% ga oshiring (aholi miqdori – har doim butun son).

11. Butun sonli turlashgan fayl mavjud. Undan birinchi noldan keyin yozilgan sonni oʻchiring (faylda nol mavjud deb qabul qilinsin). Natija boshqa faylga yozilsin.

12. Elementlari alohida belgilardan iborat turlashgan fayl mavjud. Undan birinchi “o” harfini oʻchiring (“o” harfi faylda mavjud deb qabul qilinsin). Natija boshqa faylga yozilsin.

13. Butun sonli turlashgan fayl mavjud. Birinchi 100 sonidan keyin 100 sonini qoʻying. Natija boshqa faylga yozilsin.

14. oʻttizta sonli turlashgan fayl mavjud. Boshqa faylga mavjud faylni teskari tartibda yozilsin.

15. Elementlari 20 ta alohida belgidan iborat turlashgan fayl mavjud. Boshqa faylga mavjud fayl belgilarini teskari tartibda yozilsin.

16. Elementlari son bo'lgan, o'lchamlari bir xil turlashgan ikkita fayl mavjud. ...

17. Elementlari alohida belgilardan iborat bo'lgan, o'lchamlari bir xil turlashgan ikkita fayl mavjud....

18. Elementlari sondan iborat, o'lchamlari bir xil turlashgan ikkita fayl mavjud. Dastlab birinchi faylning, keyin ikkinchisini sonlarini yozib uchinchi faylni hosil qiling. Sonlar ketma-ketlik tartibi saqlansin.

19. Elementlari alohida belgilardan iborat turlashgan fayl mavjud. Bu faylning barcha raqamlarini ikkinchi faylga, qolgan belgilarni esa uchinchi faylga yozing. Sonlar ketma-ketlik tartibi saqlansin.

20. Elementlari alohida belgilardan iborat bo'lgan, o'lchamlari bir xil turlashgan ikkita fayl mavjud. Elementlar tartibi saqlangan holda birinchi fayl elementlarini ikkinchisiga, ikkinchisini birinchisiga yozing. Yordamchi fayldan foydalaning.

21. Elementlari alohida belgilardan iborat bo'lgan, o'lchamlari bir xil turlashgan ikkita fayl mavjud. Ularning elementlarini mos kelishini aniqlang. Agar mos kelmasa, u holda bu fayllarning bir-biridan farq qiluvchi birinchi komponenta nomerini oling.

22. Elementlari butun sonlardan iborat bo'lgan, turlashgan ikkita tartiblashgan fayl mavjud. Berilgan ikkita faylni tartiblashgan holda birlashtirib, Yangi turlashgan faylni oling. Birlashtirish algoritmidan berilgan fayllar juda katta yozuvlardan iborat bo'lish mumkinligi ham qaralishi kerak.

23. Tashqi matnli fayl mavjud. Uning satrlaridan birinchi eng qisqasini chop qiling.

25.  $T$  matnli faylga bo'sh bo'lmagan haqiqiy sonlar ketma-ketligi bo'sh joy bilan ajratilib yozilgan. Bu sonlarning eng kattasini topish uchun  $\text{Max}(T)$  funksiyasini tavsiflang.

26. Matnli fayl mavjud. Undan elementlarni o'chiring:

## 5.4 C++ DASTURLASH TILIDA STRUKTURA VA ULARDAN FOYDALANISH

*Reja:*

1. *Struktura va ularni e'lon qilish;*
2. *Strukturalarga murojat qilish va ulardan foydalanish;*
3. *Struktura massivi.*

Dasturlashtirish masalalarida biron bir tashkilot yoki muassasalarning ma'lumotlar bazasini hosil qilish va ma'lumotlar bazasi tarkibidagi ma'lumotlarni qayta ishlashga to'g'ri keladi. Ma'lumotlar bazasini hosil qilishda bitta ob'yekt uchun bir nechta parametrlar saqlanadi, bunday holatlar uchun bitta ob'yekt uchun bir nechta parametrlarni birlashtirish kerak bo'ladi. Ma'lumotlar bazasini hosil qilish uchun dasturlash tillaridan qulay bo'lgan dasturiy ta'minotlar ham ko'p, lekin ba'zi hollarda dasturlash tillari ham keng qo'llaniladi. Ma'lumotlar bazasini yaratish jarayoni dasturlash tillari tarkibida keng imkoniyatlari mavjud. C++ dasturlash tilida bir nechta turlarni birlashtirib yangi tur hosil qilish jarayoni strukturalar yaratish deb ataladi. C++ dasturlash tilida ma'lumotlar bazasini hosil qilish uchun strukturalardan foydalaniladi. Struktura tarkibida bitta turga bir nechta turlar birlashadi. Masalan talaba haqida ma'lumotlar bazasi hosil qilishda talaba deb nomlangan struktura tarkibida talaba familyasi, ismi, reytingi, tug'ilgan vaqti kabi parametrlari saqlanish kerak. Demak, har bir talaba uchun yuqoridagi barcha parametrlar inobatga olinishi kerak bo'ladi. Struktura tarkibida bitta o'zgaruvchini bir nechta qiymatlari bo'lishi mumkin. Bir nechta talabalar haqida ma'lumotlarni qayta ishlash davrida bitta talaba uchun bir nechta parametrlar mavjud, bunday holatlarni hal etish uchun strukturalardan foydalaniladi.

### **Struktura va ularni e'lon qilish**

C++ dasturlash tili tarkibida bir nechta turlarni birlashtirib bitta tur asosida ma'lumotlar bazasini yaratish va ularni qayta ishlash imkoniyati keng yaratilgan. Bir nechta turlarni birlashtirish natijasida strukturalar hosil qilib C++ dasturlash tili tarkibida tashkilotlarga tegishli bo'lgan bir nechta masalalarni hal etish imkoniyati mavjud.

*Ta'rif: C++ dasturlash tilida bir yoki bir nechta turlarni jamlanmasini birlashtirish struktura deb nomlanadi.*

Strukturalarni ba'zi hollarda yozuvlar ham deb ataladi, strukturalar tarkibidagi turlarning har biri maydon deb nomlanadi.

*Ta'rif: Ob'yektning bitta xususiyatini uzida saqlaydigan parameter maydon deb nomlanadi.*

Talaba haqida ma'lumotlar strukturasi hosil qilinishda quyidagi maydonlar bo'lishi mumkin: Talabani familiyasi, ismi, otasining ismi, tug'ilgan vaqti, telefoni, manzili va hakoza. Strukturalar maydonlardan tashkil topgan ekan, struktura tarkibidagi har bir maydon strukturani bitta parametrini o'zida saqlaydi. Struktura maydoni uchun xotira hajmi, maydon uzunligiga teng bo'ladi. Struktura maydoni qiymatlari ifodalarda ishlatilishi mumkin. C++ dasturlash tilida strukturalarni e'lon qilishda *struct* xizmatchi so'zidan foydalaniladi. Struktura e'lon qilingandan so'ng dastur tarkibida o'zgaruvchilarni struktura nomi bilan e'lon qilinadi.

C++ dasturlash tilida strukturalarni tasvirlashning umumiy ko'rinishi quyidagicha ko'rinishda bo'ladi:

```
struct <struktura nomi>
{ <tur1> <maydon 1>;
  <tur2> <maydon 2>;
  - - - - -
  <tur n> <maydon n>;
};
```

Strukturalarni tasvirlashda *struct* xizmatchi so'zidan keyin struktura nomi va fequrali qavsda uning maydonlari kiritilishi shart. Struktura nomi masala mohiyatiga qarab tanlansa, maqsadga muvofiq. Masalan, talabalar haqida ma'lumotlar bazasida struktura nomini talaba yoki supermarket mahsulotlari narxlari bazasida esa narx deb nomlash mumkin. Strukturalar tarkibidagi maydonlarni e'lon qilish xuddi oddiy o'zgaruvchilarni e'lon qilishdek bajariladi va oxiri blok yopilishidan so'ng, albatta, nuqtali vergul qo'yilishi shart.

Strukturalar yartish jarayonini tushunish uchun C++ dasturlash tilida talabalar haqida ma'lumotlar bazasini struktura yordamida quyidagicha hosil qilinadi:

*Misol: Talabalar(familyasi, ismi, telefoni, reytingi, tug'ilgan yili) haqida ma'lumotlarga asosan c++ dasturlash tilida struktura hosil qiling.*

```
#include <iostream.h>
using namespace std;
struct talaba
{
    char[20] fam; //familyani saqlovchi maydon
    char[20] ism; //ismni saqlovchi maydon
```

```

char[20] tel; //telefonni saqlovchi maydon
int reyting; //reyting balini saqlovchi maydon
int tugy; //tug'ilgan vaqtini saqlovchi maydon

};
int main()
{
    return 0;
}

```

Yuqoridagi masalada talaba haqida ma'lumotlar strukturasi hosil qilishda struktura nomiga talaba deb nomlandi va uning maydonlari familiyasi(fam), ismi(ism), telefoni(tel), reytingi(reyting), tug'ilgan yili(tugy) ko'rinishlarida tasvirlandi. Maydonlarni e'lon qilishda dastur tarkibida qanday maqsadlarda foydalanishi e'tiborga olinishi kerak. Agar dastur tarkibida maydonlar ustida hisob ishlari olib borilmasa, uning turini char yoki string turi ko'rinishida e'lon qilinsa, dastur bir muncha yengil ishlaydi.

### **Strukturalarga murojat qilish va ulardan foydalanish**

C++ dasturlash tili tarkibida yaratilgan strukturalardan foydalanish mumkin, faqat qayta ishlanadigan ma'lumotlar bazasi uchun olingan struktura o'zgaruvchisini yaratilgan struktura nomi bilan e'lon qilish kerak. Talaba haqida ma'lumotlarni qayta ishlash davrida bitta talabaning bir nechta parametrlariga murojat qilish mumkin. Dasturlashda struktura o'zgaruvchisini e'lon qilishda tur sifatida struktura nomi beriladi. Talaba haqida ma'lumotlar bazasini hosil qilishda talaba deb nomlangan struktura yaratiladi, talabalar ma'lumotlar bazasini esa A o'zgaruvchiga saqlanadi, A o'zgaruvchini turini talaba deb nomlanadi. Struktura o'zgaruvchisini e'lon qilishning umumiy ko'rinishi quyidagicha bo'ladi:

**<struktura nomi> <struktura o'zgaruvchisi>**

Talaba ma'lumotlar bazasini qayta ishlash uchun talaba nomi bilan struktura yaratib, ma'lumotlarni esa A o'zgaruvchiga saqlanish holatini e'lon qilish jarayonini quyidagi dastur yordamida ko'rib chiqamiz:

```

#include <iostream.h>
using namespace std;
struct talaba
{

```

```

char[20] fam; //familyani saqlovchi maydon
char[20] ism; //ismni saqlovchi maydon
char[20] tel; //telefonni saqlovchi maydon
int reyting; //reyting balini saqlovchi maydon
int tugy; //tug'ilgan vaqtini saqlovchi maydon

};
int main()
{ talaba A; // A o'zgaruvchini talaba turi bilan e'lon qilish
  return 0;
}

```

Demak ma'lumotlar bazasini yaratishda uning uchun ajratilgan o'zgaruvchini struktura nomi bilan e'lon qilish kerak.

C++ dasturlash tilida struktura maydoniga murojat qilish uchun struktura nomi va maydon nomi nuqta bilan birlashtirilgan holda amalga oshiriladi. Struktura ma'lumotlarini dastur tarkibiga kiritish vaqtida har bir maydon elementlarini kiritish kerak bo'ladi. C++ dasturlash tilida strukturalarga murojat qilishning umumiy ko'rinishi quyidagicha bo'ladi:

**<operator funksiya> <struktura o'zgaruvchisi>.<maydon nomi>**

Strukturalarga murojatni amalga oshirish uchun quyidagi masalaga e'tibor qarating:

***Misol:** Supermarketdagi ichimlik narxi, saqlash yili haqida ma'lumot berilgan 2015 yildan hozirgacha chiqarilgan ichimliklarni narxini chiqaring aks holda muddati o'tganligini aniqlang.*

Bu masalani hal etish uchun ichimlik nomli struktura yaratiladi, strukturani ikkita narxi va saqlash muddati uchun maydonlari mavjud bo'ladi. Ichimlik haqidagi ma'lumotni A o'zgaruvchiga saqlaymiz. A o'zgaruvchini saqlash muddati maydonini 2015 bilan solishtiriladi, agar kichik bo'lsa, muddati o'tgan aks holda narxi chiqariladi.

```

#include <iostream.h>
using namespace std;
struct ichimlik
{
string nomi; //ichimlik nomini saqlovchi maydon
int narx; //ichimlik narxini saqlovchi maydon
int muddat; //ichimlik saqlash muddatini saqlovchi maydon
}

```

<pre>}; int main() {  ichimlik A;   cin&gt;&gt;A.nomi;   cin&gt;&gt;A.narx;   cin&gt;&gt;A.muddat;   if(A.muddat&lt;2015) cout&lt;&lt;"muddat utgan";   else cout&lt;&lt;A.narx&lt;&lt;"sum";   return 0; }</pre>	
<i>Kiritishga ma'lumot</i> Pepsi 4500 2014  Fanta 5000 2016	<i>Dastur natijasi</i> muddat utgan  5000 sum

C++ dasturlash tilida strukturalar yaratib ularga murojat qilish oddiy o'zgaruvchilar kabi bajariladi faqat struktura o'zgaruvchisi nuqta bilan maydon nomi yoziladi. Biror bir tashkilot yoki mussasa ma'lumotlar bazasi yaratish uchun struktura o'zgaruvchisini ko'p qiymatidan foydalanishga to'g'ri keladi. Struktura o'zgaruvchisini ikki va undan ortiq qiymatlariga murojat qilish uchun struktura o'zgaruvchisini massiv sifatida qarash mumkin.

### Struktura massivi

Dasturlash tarkibida strukturalardan yakka tartibda foydalanish maqsadga muvofiq bo'lmaydi. Strukturalar biror bir tashkilot ma'lumotlar bazasi berilganda ularning qayta ishlash uchun qulay vosita hisoblanadi. Agar strukturalar bir nechta qiymatlar uchun ishlatilsa, struktura o'zgaruvchisini struktura massivi sifatida qarash kerak.

*Ta'rif: struktura o'zgaruvchisi bir nechta qiymatlar uchun ishlatilsa, struktura o'zgaruvchisi struktura massivi deb ataladi.*

Struktura massivini e'lon qilish, xuddi oddiy massiv kabi e'lon qilinadi, faqatgina massiv turiga struktura nomi ko'rsatiladi. C++ dasturlash tilida struktura massivlarini e'lon qilishning umimiy ko'inishi quyidagicha bo'ladi:

**<struktura nomi> <struktura massivi>[<qiymatlar soni>]**

Struktura massiviga murojat qilish uchun massiv elementlariga murojat qilishdek bajariladi faqatgina massiv elementlari maydon nomi bilan birga murojat qilinadi. Talabalar haqida ma'lumotlar yaratish va ularni qayta ishlash uchun



struktura massividan foydalanish kerak, chunki bitta talaba uchun bir nechta parametr ishlatiladi va ma'lumotlar bazasida n ta talaba haqida ma'lumot saqlanadi.

**Misol:** N ta talaba haqida ismi, familyasi, tug'ilgan yili haqida ma'lumot berilgan 20 yoshda katta bo'lgan talabalarning ismi va familyasini ekranga chiqaruvchi dastur tuzing.

```
#include <iostream.h>
using namespace std;
struct talaba
{
string fam;
string ism;
int tyili;
};
int main(int argc, char* argv[])
{ talaba A[50]; int n=3;
for(int i=0;i<n;i++)
{
cin>>A[i].fam;
cin>>A[i].ism;
cin>>A[i].tyili;
}
for(int i=0;i<n;i++)
if((2017-A[i].tyili)>20)
{cout<<A[i].fam<<" "; cout<<A[i].ism<<"\n"; }
return 0;
}
```

<p><i>Kiritishga ma'lumot</i>  Allayorov Jasur 1995  Abduvaliv Umar 1998  Saydiyev Sardor 1996</p>	<p><i>Dastur natijasi</i>  Allayorov Jasur  Saydiyev Sardor</p>
--	---

**Misol:** o'qituvchilarning ismi, familyasi, ilmiy darajasi haqida ma'lumot berilgan siz professorlar, dotsentlar, assistentlar sonini aniqlovchi dastur tuzing.

```
#include <iostream.h>
using namespace std;
struct uqituvchi
```

```

{
string fam;
string ism;
string idaraja;
};
int main(int argc, char* argv[])
{ uqituvchi A[50]; int n=3;
for(int i=0;i<n;i++)
{
cin>>A[i].fam;
cin>>A[i].ism;
cin>>A[i].idaraja;
}
int p=0,d=0,s=0;
for(int i=0;i<n;i++){
if(A[i].idaraja=="professor") p++;
if(A[i].idaraja=="dotsent") d++;
if(A[i].idaraja=="assistent") s++;
}
cout<<"professorlar soni "<<p<<"\n";
cout<<"dotsentlar soni "<<d<<"\n";
cout<<"assistentlar soni "<<s;
return 0;
}

```

*Kiritishga ma'lumot*

Nazarov Fayzullo assistent

Urunbayev Erkin dotsent

Israilov Ismail professor

*Dastur natijasi*

professorlar soni 1

dotsentlar soni 1

assistentlar soni 1

C++ dasturlash tilida ma'lumotlar bazasini ma'lum bir qismlarini yaratish va ularni qayta ishlash imkoniyatini strukturalar yordamida amalga oshirish mumkin. C++ dasturash tili tarkibida strukturalar va ulardan foydalanish yuqoridagi holatlar bo'yicha amalga oshiriladi.

### **Nazariy savollar**

1. *Struktura deb nimaga aytiladi?*

2. *Maydon deb nimaga aytiladi?*

3. *Strukturalardan qanday maqsadlarda foydalaniladi?*
4. *Strukturalarning umumiy ko'rinishi?*
5. *Struktura o'zgaruvchisini e'lon qilish jarayoni?*
6. *Strukturalarga murojat qilish jarayoni?*
7. *Struktura massivi deb nimaga aytiladi?*
8. *Strukturalar massividan qanday holatlarda foydalaniladi?*
9. *Struktura massivini e'lon qilish jarayoni?*
10. *Strukturalar massiviga murojat qilish jarayoni?*

### **Mustaqil ishlash uchun topshiriqlar**

1. *Quyidagi tushunchalar uchun aralash tiplarni e'lon qiling.*
  - a) *Mahsulotlarning bahosi uchun so'm va tiyinlar;*
  - b) *Vaqt uchun soat, minut va soniyalar ;*
  - c) *Manzil uchun Shaxar, ko'cha va uy nomeri;*
  - d) *Telefon uchun kod, telefon raqamlari, abonent familiyasi va ismi;*
  - e) *Mashg'ulotlar uchun kun, fan, o'qituvchi va dars soati;*
  - f) *Imtihon qaydnomasi uchun fan, guruch raqami, talabaning familiyasi va ismi, talaba daftarchasining raqami va bahosi;*
  - g) *Shaxmat taxtasi uchun maydonlar;*
  - h) *Kompleks son uchun uning haqiqiy va mavhum qismi.*
2. *Mashina egasining familiyasi, rangi, rusumi va davlat nomeriga ega bo'lgan struktura yarating va faqat davlat nomeri A harfi bilan boshlangan mashinalar sonini ekranga chiqaring.*
3. *Mashina egasining familiyasi, rangi, rusumi va davlat nomeriga ega bo'lgan struktura yarating va faqat oq rangli mashinalar egalarining familiyasini ekranga chiqaring.*
4. *Mashina egasining familiyasi, rangi, rusumi va davlat nomeriga ega bo'lgan struktura yarating va faqat davlat raqami tarkibida 15 bo'lgan mashinalar egalarining familiyasini ekranga chiqaring.*
5. *Tog' cho'qqilarining nomi, balandligi, joylashgan davlat nomi haqida struktura yarating. Eng baland cho'qqi joylashgan davlat nomi, cho'qqi nomi va balandligini aniqlang.*
6. *Tog' cho'qqilarining nomi, balandligi, joylashgan davlat nomi haqida struktura yarating. Eng past cho'qqi joylashgan davlat nomi, cho'qqi nomi va balandligini aniqlang.*
7. *Mehmonxonalar nomi, narxi, joylashgan ko'cha nomi haqida struktura yarating. Eng arzon mehmonxona nomi va ko'cha nomini aniqlang.*

8. *Mehmonxonalar nomi, narxi, joylashgan kucha nomi haqida structura yarating. Eng qimmat mehmonxona nomi va ko'cha nomini aniqlang.*
9. *Talabalar familyasi, ismi, kursi, reyting bali haqida structura yarating. X kursdagi eng a'lochi talabani familya va ismini aniqlang.*
10. *Talabalar familyasi, ismi, kursi, reyting bali haqida structura yarating. X kursdagi eng past ko'rsatkichli talabani familya va ismini aniqlang.*
11. *Talabalar familyasi, ismi, kursi, reyting bali haqida structura yarating. X kursdagi 5 bahoga o'qiydigan talabalar familya va ismini aniqlang.*
12. *Talabalar familyasi, ismi, kursi, reyting bali haqida structura yarating. X kursdagi 4 bahoga o'qiydigan talabalar familya va ismini aniqlang.*
13. *Talabalar familyasi, ismi, kursi, reyting bali haqida structura yarating. X kursdagi 3 bahoga o'qiydigan talabalar familya va ismini aniqlang.*
14. *Shaxar poliklinikasining aholisi haqida familyasi, ismi, kasallik haqida struktura yarating. Shaxar aholisi ichida X kasallik mavjud bo'lgan odamlarning familyasi, ismi va sonini aniqlang.*
15. *Shaxar poliklinikasining aholisi haqida familyasi, ismi, kasallik haqida struktura yarating. Shaxar aholisi ichida kasallik mavjud bo'lmagan odamlarning familyasi, ismi va sonini aniqlang.*
16. *Telefon kitobida ishchilarning familyasi, ismi, lavozimi, tel raqami haqida struktura yarating. X familyali insonning tel raqami va lavozimini aniqlang.*
17. *Telefon kitobida ishchilarning familyasi, ismi, lavozimi, tel raqami haqida struktura yarating. X tel raqamli insonning familyasi va lavozimini aniqlang.*
18. *O'qituvchilarning familyasi, ismi, lavozimi haqida struktura yarating. X lavozimli o'qituvchilarning familyasi va ismini aniqlang.*
19. *O'qituvchilarning familyasi, ismi, lavozimi, tug'ilgan yili haqida struktura yarating. X yoshga kirgan o'qituvchilarning familyasi va ismini aniqlang.*
20. *Maktab o'quvchilarining familyasi, ismi, sinfi, yashaydigan ko'chasi nomi haqida struktura yarating. X ko'chada yashaydigan o'quvchilar familyasi sinfini aniqlang.*

## VI-BOB. C++ DASTURLASH TILINING GRAFIK IMKONIYATLARI

C++ dasturlash tili tarkibida boshqa dasturlash tillari kabi grafik imkoniyatlari mavjud. Dasturlashda grafik rejim bilan ishlashda, albatta, grafik rejimga o'tish komandalari berilishi shart. Dasturlashning grafik rejimida geometriya fanining ba'zi bir elementlari nuqtalar, to'g'ri chiziqlar, aylana, ellips va yarim aylana kabi elementlarni tasvirlash imkoniyatlari mavjud. C++ dasturlash tilida yuqorida ta'kidlangan elementlardan tashqari ma'lum bir sohalar, shakllar, shakllar atrofiga izohli ma'lumotlar hamda rangli soha va rangli shakllar hosil qilish imkoniyatlari mavjud.

### 6.1 C++ DASTURLASH TILI TARKIBIDA GRAFIK MUHITNI YARATISH VA UNDA FOYDALANISH

*Reja:*

1. *Grafik muhitini yaratish;*
2. *Tekislikda chizma va shakllar chizish;*
3. *Tozalash va kutib turish funksiyalari;*
4. *Grafik rejimning maksimal koordinatalari va izohli ma'lumotlar hosil qilish funksiyalari;*
5. *Chiziq shaklini o'zgartirish funksiyasi;*
6. *Tekislikda ichi bo'yalgan to'rtburchaklar hosil qilish funksiyasi.*

C++ dasturlash tilida ma'lum bir shakllar va chizmalarni hosil qilish uchun avval, albatta, grafik rejimni hosil qilish kerak. Dastur tarkibida kerakli shakl va chizmalarni yaratish buyrug'i berilgandan so'ng grafik rejimni yopib qo'yish kerak bo'ladi. C++ dasturlash tilida grafik rejimi hosil qilingandan so'ng uning tarkibiga kerakli chizma va shakllarni hosil qilish buyruqlarini yozish mumkin. Barcha dasturlash tillarida bo'lgani singari C++ dasturlash tili ham juda boy grafik imkoniyatlarga ega. Odatda qo'shimcha buyruqlar berilmasa, kompyuter ekrani matn muhitida bo'ladi. Grafik muhitni joriy qilish uchun dasturga **#include<graphics.h>** kutubxonasi ham qo'shilishi lozim. Bu kutubxona yuklangandan so'ng main() funksiyasi blokida

```
int gd=DETECT, gm; initgraph(&gd,&gm, "");
```

funksiyalar yozilgandan boshlab grafik muhit ishlay boshlaydi. Grafik muhiti ishlashini to'xtatish uchun **closegraph()**; funksiyasidan foydalaniladi. Demak grafik muhitni yuklash uchun **intigraph()**, yopish uchun esa **closegraph()** funksiyalaridan foydalaniladi. Grafik muhitida mashina ekranini dekart koordinatalari tekisligining birinchi choragi deb qarash kerak. Bunda koordinata

boshini ekranning chap yuqori burchagida, OX o‘qi gorizontaal yo‘nalishda o‘nga, OY o‘qi esa vertikal yo‘nalishda pastga yo‘nalgan deb qaraladi. Natijada, ekranni o‘zaro tig‘iz joylashgan nuqtalardan tashkil topgan deb tushunish mumkin va har bir nuqta o‘z koordinatasiga ega bo‘ladi. Shuningdek, nuqtalar ketma-ketligiga ekran rangidan boshqa rang berish yo‘li bilan tasvir hosil qilamiz. Masalan, ekran rangida biror shakl chizsak, u ko‘rinmaydi. C++ dasturlash tilining grafik muhitida har xil shakllarni chizishga mo‘ljallangan ko‘plab funksiyalar mavjud.

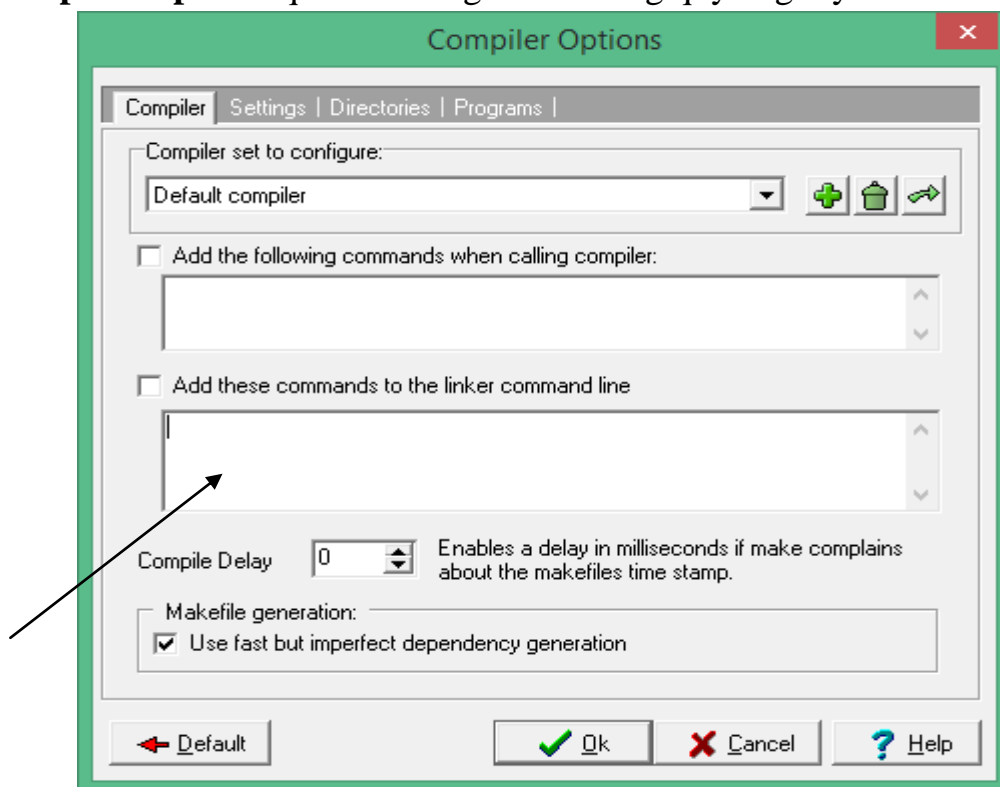
### Grafik muhitini yaratish

C++ dasturlash tili tarkibida boshqa dasturlash tillari kabi grafik rejimi va uning imkoniyatlari mavjud. C++ dasturlash tili tarkibida grafik rejimida ma’lum bir chizma, shakl va sohalar yaratish uchun, albatta, grafik muhitini hosil qilish kerak bo‘ladi. Chizmalar va sohalarni hosil qilish uchun Dev C++ dasturining grafik muhitiga sozlab olish kerak. Dev C++ dasturini grafik muhitiga sozlash uchun Dev C++ ning o‘rnatilmagan papkasi ichidan **graphics.h** va **libbgi.a** fayllarini kompyuteringizning quyidagi adresiga joylashtirish kerak.

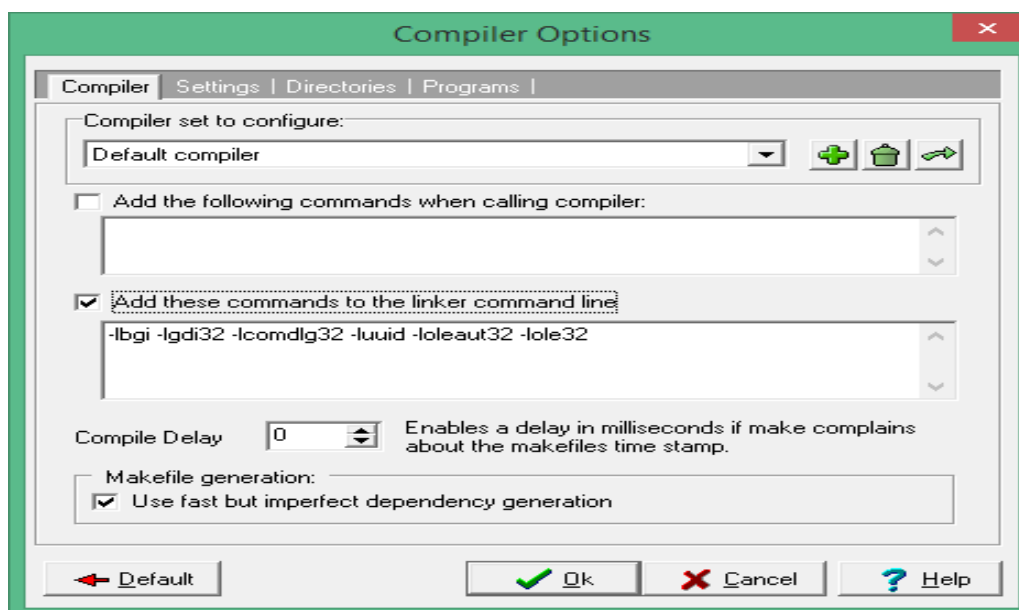
**C:\Dev-Cpp\include\graphics.h**

**C:\Dev-Cpp\lib\libbgi.h**

Yuqoridagi adreslarga fayllarni joylashtirilgandab so‘ng Dev C++ dasturi yuklanadi va dasturning **Tools** bo‘limidan **Compiler Options** qismi tanlanadi. **Compiler Options** qismi tanlangandan so‘ng quyidagi oyna hosil bo‘ladi:



Yuqoridagi oynaning ikkinchi tanlagichini tanlab strelka ko'rsatilgan oynaga **-lbg -lgdi32 -lcomdlg32 -luuid -loleaut32 -lole32** kodni kiritib **ok** tugmasini bosish kerak natijada dastur grafik rejimiga o'tkaziladi va quyidagi oyna hosil bo'ladi:



Grafik muhitdan foydalanib bo'lgandan so'ng dasturdan oddiy muhitga qaytish uchun, albatta, tanlangan bayroqchani olib tashlash kerak aks holda dastur xatolik beradi.

C++ dasturlash tilining grafik imkoniyatlaridan foydalanish uchun kutubxonalarga murojat qilish bo'limida graphics.h kutubxonasiga **#include <graphics.h>** ko'rinishida murojat qilish kerak bo'ladi. Kutubxonaga murojat qilingandan keyin dasturning asosiy **main()** funksiyasi tarkibiga **initwindow(x,y)** Prosedurasi yozilishi kerak. **initwindow(x,y)** prosedurasini vazifasi grafik rejimni (x,y) kattalik bo'yicha hosil qilib beradi. Grafik rejimi tarkibida kerakli vazifalar bajarilgandan so'ng **closegraph();** prosedurasini yordamida grafik rejimi yopiladi.

Initwindow(x,y) va closegraph() proseduralarni ishlatilish jarayoniga e'tibor qarating.

```
#include <iostream.h>
#include <graphics.h>
using namespace std;
int main(int argc, char* argv[])
{ initwindow(500,500);
system("PAUSE");
closegraph();
}
```

Grafik rejimi hosil qilingandan so‘ng kompyuter ekranini koordinatalar sistemasini I-choraki deb qarash kerak. Bunda kompyuter ekraniga chiziladigan shakl va chizmalar xuddi koordinatalar sistemasining I-chorakida chiziladigandek buruqlar beriladi. Kompyuterning ekрани bir nechta nuqtalar matritsasi bilan tashkil topgan. Dasturchi tomonidan chizilgan shakl va chizmalar ekran rangi bilan bir xil rangda bo‘lsa, chizilgan shakl va chizmalar ko‘rinmasdan qoladi, shuning uchun chiziladigan shakl, chizma va nuqtalar uchun alohida ranglar ham berilish mumkin.

### **Tekislikda chizma va shakllar chizish**

C++ dasturlash tili tarkibida shakl va chizmalar chizish uchun bir nechta graphic kutubxonasi tarkibidagi funksiyalar mavjud. Har qanday chizma va shakllar nuqtalar ketma-ketligidan hosil bo‘ladi. C++ dasturlash tilida nuqta, shakl va chizmalarni rangi ikki xil usulda beriladi, ya’ni nomi yoki kodi orqali beriladi. C++ dasturlash tili tarkibida grafik muhit imkoniyatlari quyidagi funksiyalar orqali aniqlanadi:

**SetColor(<rang nomi yoki kodi>)**-funksiyasi chiziladigan shakl va chizmalar rangini beradi. Ranglar ingliz tilidagi nomlari yoki unga mos kodlari bilan aniqlanadi.

**Setbkcolor(<rang nomi yoki kodi>)**- funksiyasi kompyuter ekranining orqa foniga rang berish uchun xizmat qiladi.

Chiziq rangi va orqa fon rangi bir xil ko‘rinishda bo‘lsa, biz hosil qilgan chiziq ko‘rinmasdan qoladi.

Ranglar nomi va kodi quyidagi jadval asosida aniqlanadi:

Rang nomi(O‘zbek tilida)	Rang nomi(ingiliz tilida)	Rang kodi
Qora	Black	0
KO‘k	Blue	1
Yashil	Green	2
Och KO‘k	Cyan	3
Qizil	Red	4
Binafsha	Magenta	5
Malla	Brown	6
KO‘lrang	Lightgray	7
Och qora	Darkgray	8
Och KO‘k	Lightblue	9
Och yashil	Lightgreen	10
TO‘q KO‘k	Lightcyan	11
Och qizil	Lightred	12



Och binafsha	Lightmagenta	13
Sariq	yellow	14
Oq	white	15

SetColor funksiyasini yashash davri qachonki, chizma va shakllarga boshqa rang berguncha ta'sir qiladi.

C++ tilida ranglarga oid barcha xossalar tarkibida ranglar nomi yoki kodi yuqoridagi jadval asosida beriladi.

**Putpixel(x,y,<rang nomi yoki kodi>-)** funksiyasi rang nomi yoki kodiga mos koordinatasi  $(x,y)$  bo'lgan nuqta hosil qiladi.

**Getpixel(x,y)-** funksiyasi koordinatasi  $(x,y)$  bo'lgan nuqta qanday rangda hosil qilinganini aniqlaydi.

**Line(x1,y1,x2,y2)-**funksiyasi koordinatalari  $(x1,y1)$  va  $(x2,y2)$  bo'lgan nuqtalar orasidagi kesma hosil qiladi.

**Circle(x,y,r)-**funksiyasi markazi  $(x,y)$  radiusi  $r$  gat eng bo'lgan aylana hosil qiladi.

**Rectangle(x1,y1,x2,y2)-**funksiyasi yuqori chap koordinatasi  $(x1,y1)$  pastki o'ng koordinatasi  $(x2,y2)$  bo'lgan to'g'ri to'rtburchak hosil qiladi.

**Lineto(x,y)-**funksiyasi joriy nuqtadan  $(x,y)$  nuqtagacha kesma hosil qiladi.

**Linerel(x,y)-**funksiyasi  $(x,y)$  nuqtadan joriy nuqtagacha kesma hosil qiladi.

Lineto va linerel funksiyalari dastur tarkibida bo'yruqlar asosida kursor kelib qolgan nuqtani joriy nuqta sifatida qabul qiladi.

**Drawpoly(n,p)-**funksiyasi ko'pburchak hosil qilish uchun xizmat qiladi. Bu yerda  $n$  burchaklar soni,  $p$  burchaklar koordinatalari hisoblanadi.

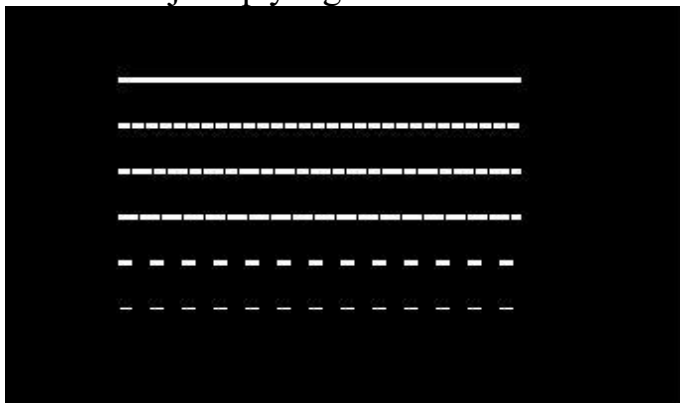
Endi quyidagi dasturni tahlil qiling

```

#include <graphics.h>
int main()
{ int gd = DETECT, gm, c , x = 100, y = 50;
  initgraph(&gd, &gm, "C:\\TC\\BGI");
  for ( c = 0 ; c <=5 ; c++ ) { setlinestyle(c,100, 3); line(x, y, x+200, y);
    y = y + 25; } getch(); closegraph();
return 0;
}

```

Dastur natijasi quydagicha ko‘rinishda bo‘ladi:



**Misol:** Qora rangli ekranga oq rangli aylana chizish va uning markazini oq rang bilan belgilash dasturini hosil qiling.

Bu masala yechimida ekran kattaligi (500,500), ekran fon rangi qora, chiziladigan chiziq rangi oq deb qaraladi. Chiziladigan aylananing markazi (200,200) radiusi 150 ga teng. Aylana markazidagi (200,200) nuqtani oq rang bilan chiziladi.

```

#include <graphics.h>
#include <cstdlib>
int main()
{ initwindow(500,500);
  setbkcolor(0);
  setcolor(15);
  circle(200,200,150);
  putpixel(200,200,15);
  system("PAUSE");
  closegraph();
  return EXIT_SUCCESS;
}

```



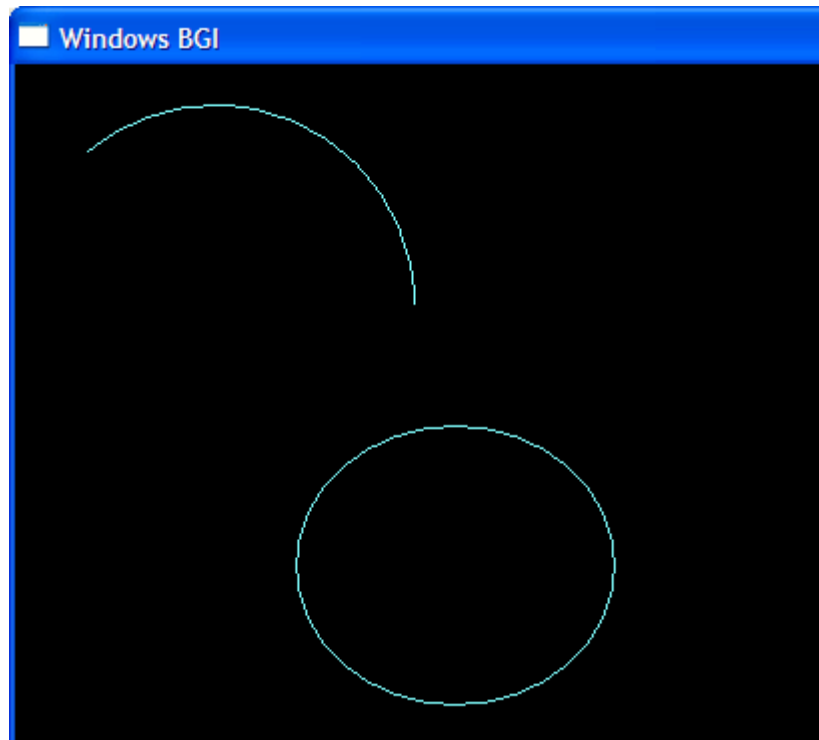
Yuqoridagi dastur natijasiga ko'ra masalani berilish shartidagi shakl va chizmalar hosil bo'ladi.

***Arc(x,y,a,b,r)***-funksiyasi markazi  $(x,y)$  radiusi  $r$  ga teng bo'lgan  $a$  gradusdan  $b$  gradusgacha bo'lgan aylananing qismini hosil qiladi. Bu funksiya aylananing qandaydir qismini chizadi, ya'ni  $a$  gradusdan  $b$  gradusgacha qismini chizadi.

***Ellipse(x,y,a,b,n,m)***- funksiya ellips va uning yoyini hosil qilish uchun xizmat qiladi. Ellips funksiyasi markazi  $(x,y)$  bo'lgan,  $a$  boshlang'ich burchak gradusi,  $b$  tugallanish burchak gradusi,  $n$  OX o'qi bo'yicha,  $m$  OY o'qi bo'yicha raduslari hisoblanadi.

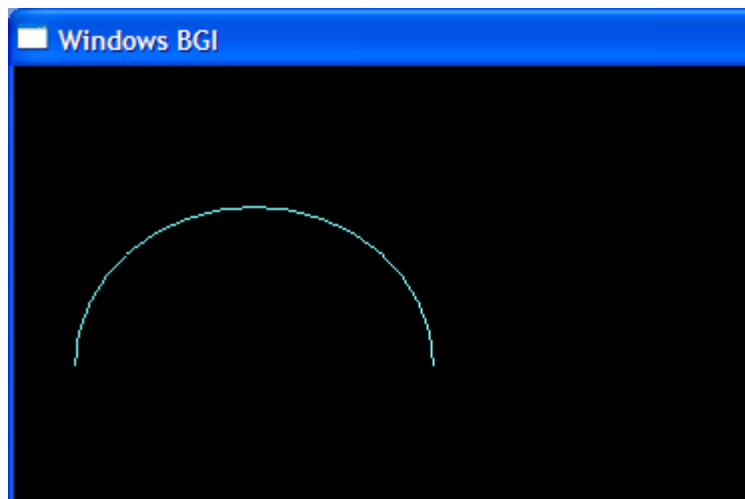
***Misol:*** 130 gradusli yoy va to'liq ellips hosil qilish dasturini tuzing.

```
#include <iostream.h>
#include <graphics.h>
using namespace std;
int main(int argc, char* argv[])
{ initwindow(500,500);
  setbkcolor(0);
  setcolor(15);
  arc(100,120,0,130,100);
  ellipse(220,250,0,360,80,70);
  system("PAUSE");
  closegraph();
  return EXIT_SUCCESS;
}
```



*Misol: 180 gradusli ellips hosil qilish dasturini tuzing.*

```
#include <iostream.h>
#include <graphics.h>
using namespace std;
int main(int argc, char* argv[])
{ initwindow(500,500);
  setbkcolor(0);
  setcolor(15);
  ellipse(120,150,0,180,90,80);
  system("PAUSE");
  closegraph();
  return EXIT_SUCCESS;
}
```



Yuqoridagi ikki misol tarkibidagi parametrlar aniq sonlar ko‘rinishida berilgan. Dasturchi funksiya parametrlarini o‘zgaruvchi orqali ham berish mumkin.

### **Tozalash va kutib turish funksiyalari**

C++ dasturlash tilining grafik imkoniyatlari tarkibida chizmalar va shakllar chizilgan sohani tozalash, hamda buyruqlar orasidagi vaqtni belgilash funksiyalari ham mavjud.

***Cleardevice(x1,y1,x2,y2)***-funksiya yuqori chap koordinatasi  $(x1,y1)$  pastki o‘ng koordinatasi  $(x2,y2)$  bo‘lgan sohani tozalash uchun xizmat qiladi.

***Delay(n)***-funksiya buyruqlar orasidagi vaqtni  $n$  millisekund ushlab turish uchun xizmat qiladi.

***Misol:*** Ekrandagi shakllarni o‘chirib aylana va ellipslarni 2000 millisekund oralig‘ida hosil qilish dasturini tuzing.

```
#include <iostream.h>
#include <graphics.h>
using namespace std;
int main(int argc, char* argv[])
{ initwindow(500,500);
  setbkcolor(0);
  setcolor(15);
  cleardevice(1,1,300,300);
  circle(280,270,50);
  delay(2000);
  ellipse(120,150,0,360,40,50);
  system("PAUSE");
  closegraph();
  return EXIT_SUCCESS; }
```



Delay funksiyasini qulayligi bir nechta chizmalarga berilgan buyruqlar orasidagi vaqtlarni aniqlaydi.

### **Grafik rejimning maksimal koordinatalari va izohli ma'lumotlar hosil qilish funksiyalari**

C++ dasturlash tilini grafik rejimining sohasi uchun ekrandan qancha joy egallanishini aniqlash va hosil qilingan chizma va shakllar uchun izohli ma'lumotlar yozish funksiyalari ham mavjud.

**Getmaxx-** funksiyasi joriy rejim uchun ajratilgan gorizantal nuqtalarning maksimal sonini aniqlaydi.

**Getmaxy-** funksiyasi joriy rejim uchun ajratilgan vertikal nuqtalarning maksimal sonini aniqlaydi.

**Outtextxy(x,y,"matn")-**funksiyasi joriy rejimning (x,y) nuqtasidan boshlab matnni ekranga chop etish uchun xizmat qiladi.

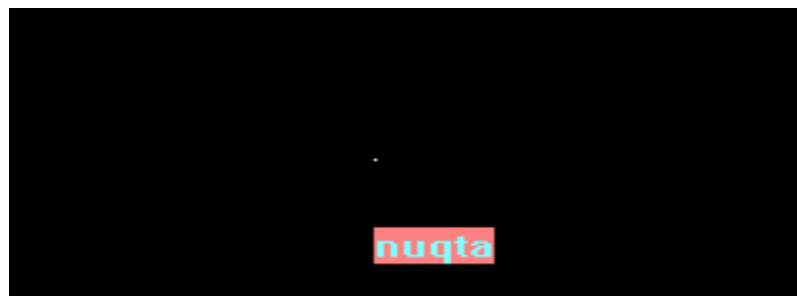
**Misol:**Joriy rejimning o'rtasiga ko'k rangli nuqta hosil qilib, nuqta ostiga nuqta yozuvini hosil qiluvchi dastur hosil qiling.

```
#include <iostream.h>
#include <graphics.h>
using namespace std;
int main(int argc, char* argv[])
{ initwindow(500,500);
  setbkcolor(0);
  setcolor(15);
  putpixel(250,250,15);
```

```

outtextxy(250,280,"nuqta");
system("PAUSE");
closegraph();
return EXIT_SUCCESS;
}

```

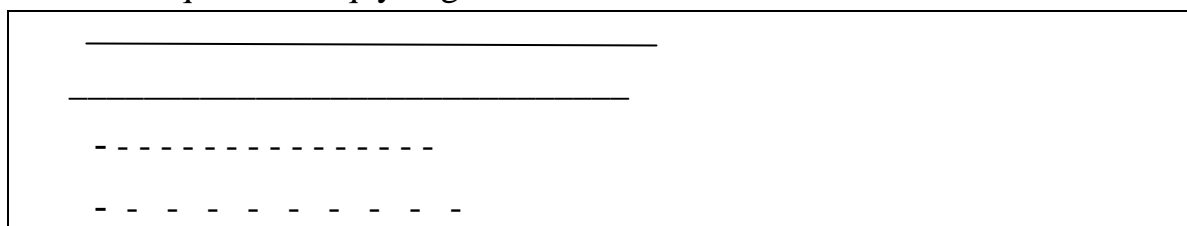


### Chiziq shaklini o‘zgartirish funksiyasi

*Setlinestyle(x,y,z)*- funksiyasi grafik muhitida chiziladigan shakl va chizmalar chiziqining har xil ko‘rinishlarini aniqlaydi. Bu funksiya parametrda  $x$  usul nomi,  $y$  butun son,  $z$  chiziq qalinligini ifodalaydi. Chiziladigan chiziq shakllarining nomi va kodi quyidagi jadval asosida aniqlanadi:

Usul nomi(x qiymati uchun)	Usul kodi
Solid_line	0
Dotted_line	1
Center_line	2
Dashed_line	3
Userbit_line	4

Chiziq shakllari quyidagi ko‘rinishlarda bo‘lishi mumkin.



### Tekslıkda ichi bo‘yalgan to‘rtburchaklar hosil qilish funksiyasi

C++ dasturlash tilida ma’lum bir bo‘yalgan sohalarni hosil qilish funksiyalari ham mavjud.

*Setfillstyle(<usul>,<rang>)*- funksiyasi tekslıkda ichi har xil bo‘yalgan to‘rtburchaklar hosil qilish uchun xizmat qiladi. To‘rtburchaklarni bo‘yash usullari 0 dan 9 gacha kodlar orqali beriladi, ranglari esa rang nomi yoki kodi orqali aniqlanadi.

Yuqoridagi funksiyalar yordamida har qanday shakl va chizmalarni hosil qilish imkoniyati mavjud. Jumladan uchburchaklar, ko'pburchaklar va soatlar hosil qilish ham mumkin.

### **Nazariy savollar**

1. *Grafik muhitini yaratish ketma- ketligini tushuntirib bering?*
2. *graphics.h kutubxonasi vazifasini tushuntirib bering?*
3. *Chiziq rangini aniqlash funksiyasi tushuntirib bering?*
4. *Orqa fon rangini aniqlash funksiyasi tushuntirib bering?*
5. *Nuqta chizish funksiyasi tushuntirib bering?*
6. *Nuqta rangini aniqlash funksiyasi tushuntirib bering?*
7. *Ikki nuqta orasidagi kesma hosil qilish funksiyasi tushuntirib bering?*
8. *Aylana hosil qilish funksiyasi tushuntirib bering?*
9. *To'g'ri to'rtburchak hosil qilish funksiyasi tushuntirib bering?*
10. *Ko'pburchak hosil qilish funksiyasi?*
11. *linere va lineto funksiyalarini vazifasi va umumiy ko'rinishi?*
12. *Qism aylana hosil qilish funksiyasi?*
13. *Ellips hosil qilish funksiyasi?*
14. *Oynani tozalash funksiyasi?*
15. *Vaqtini aniqlash funksiyasi?*
16. *Grafik rejimni maksimal koordinatalarini aniqlash funksiyalari?*
17. *Matnlarni chop etish funksiyasi?*
18. *Chiziq shaklini o'zgartirish funksiyasi?*
19. *Ichki sohasi bo'yalga tortburchaklar hosil qilish funksiyasi?*
20. *putpixel funksiyasi qanday shaklni chizish uchun ishlatiladi?*
21. *Grafik muhitda tayin nuqtaning rangi qanday aniqlanadi?*
22. *Tekislikda chiziladigan shakllar rangi qaysi funksiya yordamida tayinlanadi?*
23. *circle() funksiyasidan foydalanmasdan ham aylana chizishning necha xil usulini bilasiz?*
24. *Grafik muhitni yuklovchi funksiya qaysi?*
25. *rectangle() funksiyasi qanday shakllarni chizish uchun ishlatiladi?*
26. *line() funksiyasi qanday shaklni chizish uchun ishlatiladi?*
27. *drawpoly(n,m) funksiyasida n va m lar qanday parametrlar va ularning qiymatlari qanday kiritiladi?*
28. *getch() funksiyasi qanday vazifani bajaradi?*
29. *Qachon chizilgan shakl ko'rinmaydi?*
30. *Aylana chizish funksiyasining umumiy ko'rinishini yozing va parametrlarining vazifasini tushuntiring.*



31. Dasturning javob oynasini tozalovchi funksiya qaysi?
32. Grafik muhitda javob darchasidan dastur darchasiga qanday o'tiladi?
33. Uchlari (2;1), (3;2) va (4;-5) nuqtalarda bo'lgan uchburchakni qanday usullarda chiza olasiz?
34. ellipse(a,b,c,d,e,f) funksiyasida a, c, e parametrlar vazifasini tushuntiring.
35. delay(50); buyrug'i vazifasini tushuntiring.
36. Ekraning tayin joyida tasvir hosil qilish nima qilish kerak?
37. arc() funksiyasi yordamida aylana chizish mumkinmi? Mumkin bo'lsa qanday qilib?
38. Chiziqlarni har xil usullarda chizish uchun qaysi funksiyadan foydalaniladi?
39. C++ grafik muhitida necha xil rangdan foydalaniladi?
40. C++ grafik muhitidan matn muhitiga qanday o'tiladi?
41. closegraph() funksiyasining vazifasi nima?
42. putpixel() funksiyasi yordamida uzunligi 2 ga teng kesma chizish mumkinmi, agar mumkin bo'lsa qanday chiziladi?

### **Mustaqil ishlash uchun topshiriqlar**

**Quyidagi har bir masalalarni yechishda, albatta, har xil ranglardan foydalanishingiz kerak.**

1. Uchburchak uchlari koordinatalari berilganda qora rangli ekranda qizil rangli uchburchak hosil qiling va uchburchak ostiga "uchburchak" yozuvini hosil qiling.
2. Ikki nuqta koordinatalari berilganda qora rangli ekranda oq rangli kesma hosil qiling va kesma o'rtasidagi nuqtaga "kesma o'rtasi" yozuvini hosil qiling.
3. Koordinatalar sistemasi hosil qiling va markazi koordinata boshida bo'lgan aylana hosil qiling.
4. Soat dasturini yarating.
5. Kamaz chizmasini yarating.
6. Kamaz chizmasini yaratib kamazni harakatlantiring.
7. Aylanani 50 millisekund oralig'ida ekran bo'ylab harakatlantiring.
8. 30 gradusli yo'ylar yordamida 40 millisekund oralig'i bilan aylana hosil qiling.
9. 15 gradusli yo'ylar yordamida 60 millisekund oralig'i bilan n ta aylanalar hosil qiling.
10. Ellips chizmasi yordamida bir biri bilan kesishadigan n ta ellips yarating.
11. Ellips chizmasi yordamida bir biri bilan kesishmaydigan n ta ellips yarating.
12. Aylana yordamida harakatlanuvchi halqa yarating. Halqa bilan birga halqa markazida "halqa" yozuvi birga harakatlansin.

13. Ikki nuqta orasidagi kesma yordamida harakatlanuvchi beshburchak yarating. beshburchak bilan birga halqa markazida “besh” yozuvi birga harakatlansin.
14. To'rtburchakli soat dasturini yarating.
15. Kesma yordamida 60 millisekund oralig'i bilan uchburchak hosil qiling.
16. Kesma yordamida  $n$  millisekund oralig'i bilan harakatlanuvchi to'rtburchak hosil qiling.
17. Kesma yordamida  $n$  millisekund oralig'i bilan harakatlanuvchi ko'pburchak hosil qiling.
18. Yarim aylana yordamida  $n$  millisekund oralig'ida harakatlanuvchi aylana yarating va aylana bilan birga “harakat” so'zi harakatlansin.
19. Kesma yordamida harakatlanuvchi laylak chizmasini yarating.
20. Kesma aylana yordamida harakatlanuvchi odam shaklini hosil qiling.

## 6.2 C++ TILINING GRAFIK MUHITIDA FAZOVIIY JISMLARNI HOSIL QILISH

*Reja:*

1. Fazodagi shakllarni hosil qilish;
2. Fazodagi shakllarni hosil qilishga oid misollar.

C++ dasturlash tili tarkibining grafik muhit imkoniyatlarida nafaqat tekislikdagi balki fazodagi jismlarni ham shakllarini hosil qilish imkoniyati mavjud. Geometriya fanining ba'zi turdagi chizmalarini C++ dasturlash tili yordamida ham chizish imkoniyatlari mavjud.

### Fazodagi shakllarni hosil qilish

Fazodagi barcha fequralarga rang berish uchun *setfillstyle()* funksiyasi orqali bajariladi.

**Bar3d(x1,y1,x2,y2,r,a)**- funksiyasi to'g'ri burchakli paralelopepid hosil qilish uchun xizmat qiladi. Bu funksiya tarkibida  $(x1,y1)$  paralelopepidning old yoqining yuqoridagi chap tomonini koordinatasi,  $(x2,y2)$  paralelopepidning pastdagi o'ng tomonini koordinatasi,  $r$  yon yog'ining kengligi,  $a$  ustki yoqining bor yoki yo'qligini aniqlaydi.

**Fillellipse(x,y,a,b)**- bu funksiya yordamida yassi ellipslar hosil qilish mumkin. Funksiya tarkibida  $(x,y)$  ellips markazi koordinatasi,  $a$  va  $b$  lar ellipsning radiuslari hisoblanadi. Agar  $a$  va  $b$  ga bir xil qiymatlarni kiritsangiz, u holda funksiya doira hosil qiladi.

**Fillpoly(x,p)**- funksiyasi ichi bo‘yalgan yassi ko‘pburchak hosil qiladi. Bu funksiya xuddi **drawpoly** funksiyasiga o‘xshash holda ishlaydi.

**Floodfill(x,y,<rang>)**-funksiyasi yopiq soha ichini bo‘yash uchun xizmat qiladi. Funksiya tarkibidagi (x,y) nuqta koordinatalari yopiq sohaga tegishli bo‘lsa, soha ichini, aks holda soha tashqarisini bo‘yaydi. Soha chegarasidagi rangi va soha ichidagi rang bir xil bo‘lmasa, dastur xatolik berishi mumkin. Soha ichiga **setfillstyle()** funksiyasi orqali rang berish mumkin, agar **setfillstyle()** funksiyasi ishlatilmasa, jimlik qoidasi bo‘yicha oq rang tanlanadi.

**Sector(x,y,a,b,n,m)**- funksiyasi sector hosil qilish uchun xizmat qiladi. Funksiya tarkibidagi (x,y) sector uchi koordinatasi, a chizish boshlanish burchaki, b chizish tugallanish burchaki, n va m lar yarim o‘qlardir.

C++ dasturlash tilining grafik muhiti tarkibida yuqoridagi funksiyalar yordamida fazoviy jismlarni ham tasvirlash imkoniyati mavjud.

Har bir fazoviy jismlarni hosil qilgandan so‘ng, albatta, close bo‘yruq‘i bilan grafik muhiti yopiladi.

**Misol:** *Doira va yassi ellips hosil qiluvchi dastur yarating.*

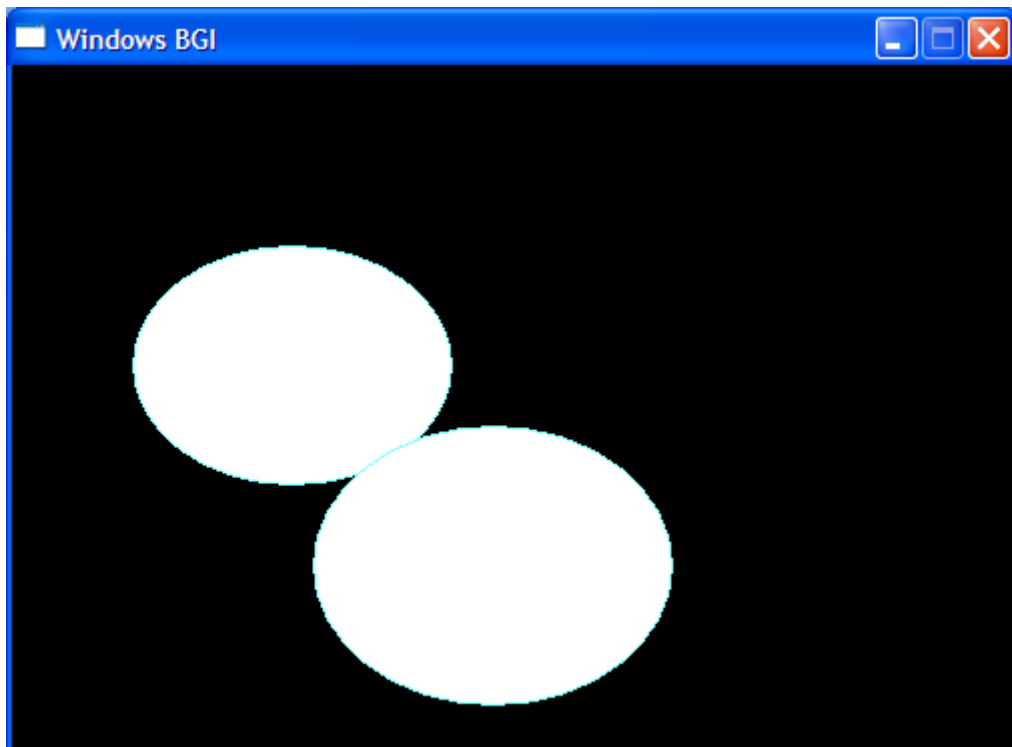
```
#include <iostream.h>
#include <graphics.h>
using namespace std;
int main(int argc, char* argv[])
{
    initwindow(500,500);

    setbkcolor(0);
    setcolor(15);

    fillellipse(140,150,80,60);
    fillellipse(240,250,90,70);

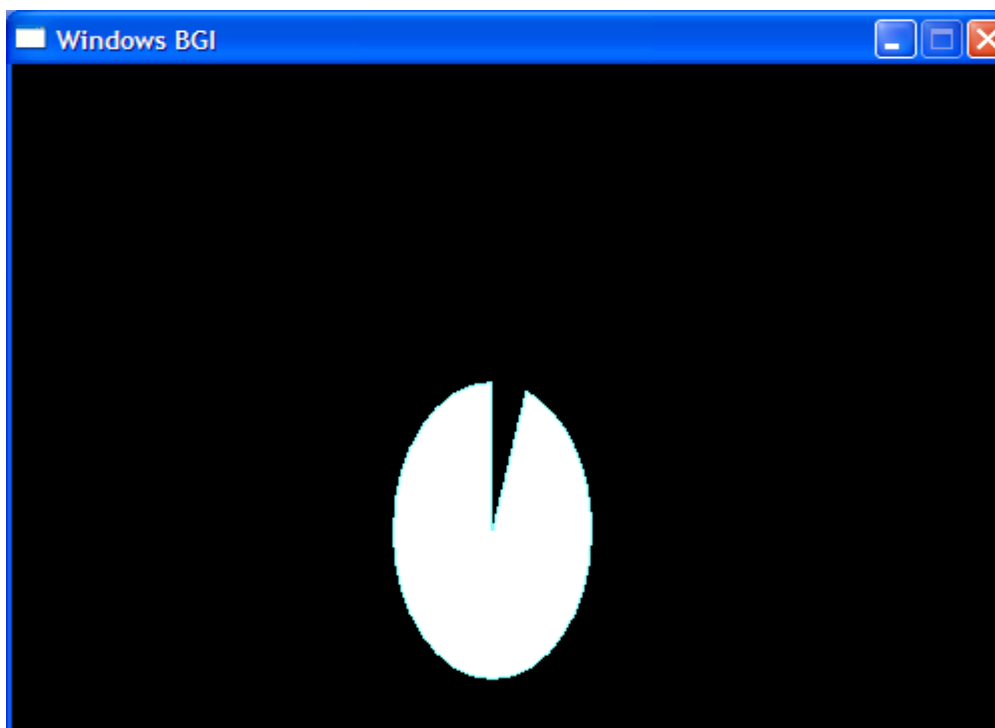
    system("PAUSE");
    closegraph();

    return EXIT_SUCCESS;
}
```



*Misol: Sektor hosil qiluvchi dastur yarating.*

```
#include <iostream.h>
#include <graphics.h>
using namespace std;
int main(int argc, char* argv[])
{ initwindow(500,500);
  setbkcolor(0);
  setcolor(15);
  sector(240,250,90,70,50,80);
  system("PAUSE");
  closegraph();
  return EXIT_SUCCESS;
}
```



Yuqoridagi funksiyalar yordamida uch o'lchovli shakllarni ham chizish mumkin.

### **Nazariy savollar**

1. *To'g'ri burchakli paralelopid hosil qilish funksiyasi?*
2. *Yassi ellipslar hosil qilish funksiyasi?*
3. *Ichi bo'yalgan yassi ko'pburchak hosil funksiyasi?*
4. *Yopiq soha ichini bo'yash uchun xizmat funksiyasi?*
5. *Sector hosil qilish funksiyasi?*

### **Mustaqil ishlash uchun topshiriqlar**

***Quyidagi har bir masalalarni yechishda, albatta, har xil ranglardan foydalanishingiz kerak.***

1. *Paralelopid uchlari koordinatalari berilganda qora rangli ekranda qizil rangli paralelopid hosil qiling va paralelopid ostiga "paralelopid" yozuvini hosil qiling.*
2. *Kesma va paralelopid yordamida 50 millisekund oralig'ida uy yasash dasturini yarating.*
3. *Kesma va paralelopid yordamida harakatlanuvchi uy dasturini yarating.*
4. *O'zbekiston bayrog'ini yarating.*
5. *AQSh bayrog'ini yarating.*
6. *To'rtburchakli kamaz ustida bayroq hosil qiling*
7. *Harakatlanuvchi to'rtburchakli kamaz ustida bayroq hosil qiling.*
8. *N gradusli sector hosil qiling.*

9. Ekranda faqat gorizantal yo'nalishda harakatlanuvchi sektor yarating.
10. Ekranda faqat vertikal yo'nalishda harakatlanuvchi sektor yarating..
11. Sektor yordamida doira va doira ostiga "doira"sozini yozuvchi dastur yarating.
12. Sektor yordamida harakatlanuvchi doira va doira ostiga "doira"sozini yozuvchi dastur yarating.
13. Sektor yordamida keshishmaydigan harakatlanuvchi doiralar yarating.
14. Sektor yordamida keshishadigan harakatlanuvchi doiralar yarating.
15. Sektor yordamida 60 millisekund oralig'i bilan  $n$  ta sektor hosil qiling.
16. Sektor yordamida  $n$  millisekund oralig'i bilan  $n$  ta doira hosil qiling.
17. Birinchisini markazidan o'tuvchi ikkita yassi ellips hosil qiling.
18. O'zidan oldingisini markazidan o'tuvchi yassi ellipslarni  $n$  millisekund oralig'i bilan gorizantal yo'nalish bo'yicha ekranni to'ldirib hosil qiling.
19. O'zidan oldingisini markazidan o'tuvchi yassi ellipslarni  $n$  millisekund oralig'i bilan vertikal yo'nalish bo'yicha ekranni to'ldirib hosil qiling.
20. Sektor yordamida 60 millisekund oralig'i bilan vertikal yo'nalish bo'yicha ekranni to'ldirib sektorlar hosil qiling.

## FOYDALANILGAN ADABIYOTLAR RO'YXATI

1. Стефан Р. Девис. С++ для чайников. Диалектика. Москва -2003. 337 с.
2. Страуструп Б. Язык программирования С++. Специальное издание. – Москва 1997. -1055 с.
3. Фролов А.Б. Фролов Г.Б. Язык С# Самоучитель. Москва 2003.-559 с.
4. Герберт Ш. С++ Базовый курс. Москва Издательский дом “Вильямс”.2010.-621 с.
5. Стенли Липпман. Язык программирование С++. Базовой курс. Вильямс - М.: 2014.
6. Никита Культин. Microsoft Visual С++ в задачах и примерах. БХВ-Петербург - Петербург.:2010.
7. Б. Страуструп. Язык программирования С++. Специальное издание.-М.:ООО «Бином-Пресс», 2006.-1104 с.
8. Павловская Т.А. С++. Программирование на языке высокого уровня –СПб.: Питер. 2005.- 461 с.
9. Подбельский В.В. Язык С++.- М.; Финансы и статистика- 2003 562с.
10. Павловская Т.С. Щупак Ю.С. С/С++. Структурное программирование. Практикум.-СПб.: Питер,2002-240с
11. Павловская Т.С. Щупак Ю.С. С++. Объектно- ориентированное программ-ирование. Практикум.-СПб.: Питер,2005-265с
12. С++: Учебный курс.- Харьков: Фолио; М.: ООО «Издательство АСТ», 2001.-500с.
13. Ш.Ф. Мадрахимов, С. М. Гайназаров С++ тилида программалаш асослари. Т. 2009.
14. [www.Intuit.ru](http://www.Intuit.ru). Интернет-Университет информационных технологий.Москва.
15. Пильшиков В.Н. Упражнения по языку Паскаль-М.: МГУ, 1986.
16. Абрамов С.А., Гнезделова Капустина Е.Н. и др. Задачи по программ-ированию. - М.: Наука, 1988.
17. Вирт Н. Алгоритмы + структуры данных = программа.- М.:Мир,1985.-405с.
18. Информатика. Базовой курс. Учебник для Вузов., Санк-Петербург,2001. под редакцией С.В.Симоновича.
19. Eshtemirov S. Nomozov F. С++ dasturlash tili. Uslubiy qo'llanma. Samarqand 2016. -146 b.
20. Nazarov F. С++ tilida dasturlash asoslari. Uslubiy qo'llanma. Samarqand 2017. -160 b.

**Eshtemirov S., Nazarov F.M.**

# **C++ TILIDA DASTURLASH ASOSLARI**

Muharrir: Gulbahor Rahimova  
Musahhih: Mardon Ro‘ziboyev  
Tex. muxarrir: Bahodir Egamberdiyev

ISBN 978-9943-

2008-yil 19-iyun 68-buyruq.

2017-yil 27-martda noshirlik bo‘limiga qabul qilindi.

2017-yil 8-aprelda original-maketdan bosishga ruxsat etildi.

Bichimi 60x84/ 1,16. «Times New Roman» garniturasida.

Ofset qog‘ozi. Shartli bosma tabog‘i 10,0.

Nashriyot hisob tabog‘i 9,0. Adadi 25 nusxa. 77-buyurtma.

---

*SamDU bosmaxonasida chop etildi.*

*140104, Samarqand sh., Universitet xiyoboni, 15.*