

**O'zbekiston Respublikasi Oliy va o'rta maxsus
ta'lim vazirligi**

Samarqand davlat universiteti

Aminov I.B., Suyarov A.M.

**DASTURLASH TILLARI ASOSIDA
MATEMATIK MASALALARNI YECHISH**

O'quv qo'llanma

Samarqand - 2019

UDK 618.3

UDK 32.973

**Samarqand davlat universiteti kengasi
tomonidan 5130100-matematika va 5140300-mexanika
ta'lim yo'nalishi talabalari uchun o'quv qo'llanma si-
fatida tavsiya etilgan**

Taqrizchilar:

texnika fanlari nomzodi, dotsent A.A.Ganiyev (TATU)

*texnika fanlari nomzodi, dotsent A.Qarshiyev
(TATU Samarqand filiali).*

**Aminov I.B., Suyarov A.M. “Dasturlash tillari asosida ma-
tematik masalalarni yechish” O'quv qo'llanma: O'zbekiston
respublikasi oliy va o'rta maxsus ta'lim vazirligi – SamDU,
Samarqand: 2019. – 180 bet**

Matematika va mexanika ta'lim yo'nalishi bo'yicha bilim olayotgan talabalarga “Dasturlash asoslari” fanini o'qitishda fanlarning o'zaro integratsiyasiga hamda o'qitilayotgan fanning kasbga yo'nalgan sohaga tatbiqlariga jiddiy e'tibor berish lozim bo'ladi. Shu bilan bir qatorda bakalavriyatning 5130100-matematika va 5140300-mexanika ta'lim yo'nalishlarida ta'lim olayotgan talabalarga algoritmlash va dasturlash asoslarini bir xil o'qitilishi, ularga kelgusi kasbiy faoliyatida qay darajada kerak bo'lishi to'g'risidagi tushunchaga va uning o'rnini sezishga imkon bermaydi. Shuning uchun ham algoritmlash va dasturlash asoslaridan amaliy mashg'ulotlardagi tushunchalarni kasbga yo'naltirilgan holda tashkil etishni maqsadga muvofiq deb hisoblaymiz.

ISBN – 978-9943-537-74-3

© Aminov I.B., Suyarov A.M.

Kirish

Bugungi kunda mamlakatimizda ta'lim tizimini isloh qilish, uning mazmunini yangilash, o'qituvchining kasbiy bilim va tayyorgarligini oshirish, o'quvchilarning shaxsiy tajribalarini boyitish, ularni ijtimoiy hayotga tayyorlash sohasida keng ko'lamli izlanishlar amalga oshirilmoqda. Ushbu sohadagi kamchiliklarni bartaraf etish yo'llari izlanmoqda. Bularning barchasi zamonaviy maktab ta'limining umumiy yo'nalishini belgilash, dunyo miqyosida uning rivojlanishiga ta'sir ko'rsatadigan omillarga keng yo'l ochish imkonini beradi. Buning uchun, avvalo, uzliksiz ta'lim tizimining barcha shakldagi ta'lim muassalarida ta'lim jarayoni sifatida ta'minlovchi ilg'or ilmiy-metodik jihatdan asoslangan zamonaviy o'qitish texnologiyalarni amalda oshirish lozim.

Insoniyat tarixining ko'p asrlik tajribasi ezgu go'yalardan va sog'lom mafkura hamda zamonaviy bilimlardan maxrum har qanday jamiyat uzoqqa bora olmasligini ko'rsatdi. Shuning uchun, mustaqillikka erishgan mamlakatimiz o'z oldiga ozod va obod vatan, demokratik jamiyat barpo qilish, erkin va farovon hayot qurish, rivojlangan mamlakatlar qatoridan o'rin olish kabi muhim vazifalarni qo'ydi. Bu vazifalarni hal qilish asosan yosh avlod zimmasiga tushadi.

Demak, yoshlardan zamonaviy komputerlar bilan ishlashni o'rganish, xalq xo'jaliginining turli masalalarini yechishga mo'ljallangan dasturiy ta'minot bilan tanishish hamda dasturlash vositalari yordamida hali komputerda yechilmagan masalalar uchun yangi dasturlar yaratishni talab qilinadi.

Darvoqi, har bir jamiyatning bugungi kuni va kelajagi uning ajralmas qismi va hayotiy zarurati bo'lgan ta'lim tizimining qay darajada rivojlanganligi bilan belgilanadi. Bugungi kunda mustaqil taraqqiyot yo'lidan borayotgan mamlakatimizning uzluksiz ta'lim tizimini isloh qilish va takomillashtirish, yangi sifat bosqichiga ko'tarish, unga ilg'or pedagogik va axborot texnologiyalarini joriy qilish hamda samaradorligini oshirish davlat siyosati darajasiga ko'tarildi. "Ta'lim to'g'risida"gi qonun va "Kadrlar tayyorlash milliy dasturi"ning qabul qilinishi bilan uzluksiz ta'lim tizimi orqali zamonaviy kadrlar tayyorlashning asosi yaratildi.

Oliy o'quv yurtlarida talabalarni mustaqil bilim olish faoliyatini tashkil etishga, bilimlarni mustaqil egallashga va ularni amaliyotda

qo'llash malakalarini shakllantirishga qaratish o'qitishning metodlari, texnologiyalarini shunday tanlash lozimligini ko'rsatdiki, ular talabalarga nafaqat tayyor bilimlarni o'zlashtirishlarida, balki bilimlarni turli manbalardan mustaqil egallashlari, o'zlarida shaxsiy nuqtai nazarning shakllanishi, uni asoslashi va erishilgan bilimlarni yangi bilimlar olishlarida foydalanish malakalariga ega bo'lishlarini taqoza etadi. Bunday ulkan vazifani muvaffaqiyatli amalga oshirishda o'qituvchi va talabalarga davlat tilida kerakli o'quv adabiyotlarini yetkazish alohida ahamiyatga ega.

Ushbu qo'llanma "Dasturlash tillari asosida matematik masalalarni yechish" fani bo'yicha matematika va mexanika ta'lim yo'nalishlarida bilim olayotgan talabalar uchun mo'ljallangan bo'lib, uning asosiy maqsadi algoritmash va dasturlash tillari asosida matematik masalalarni yechish imkoniyatlaridan xabardor qilib, kompyuter bilan muloqot o'rnatish usullarini o'rgatish va unda turli masalalarni yecha olish bilim va malakalarini shakllantirishdan iboratdir.

I-bob. Algoritm va algoritmlash asoslari

1§. Algoritmlar

1.1. Masalalarni kompyuterda yechish bosqichlari.

Har qanday masalani kompyuterga tayyorlash va uni o'tkazish quyidagi bosqichlarda amalga oshiriladi.

1. *Masalani qo'yilishini aniqlash va matematik modelini ishlab chiqish.*
2. *Masalani yechishning sonli usulini tanlash.*
3. *Masalani yechish algoritmini ishlab chiqish.*
4. *Kompyuter uchun dastur tuzish.*
5. *Dasturni kompyuter xotirasiga kiritish, rostlash va tekshirish.*
6. *Hisoblash natijalarini qayta ishlash va tahlil qilish.*

1. Masalaning qo'yilishini aniqlash va matematik modelini ishlab chiqish. Masalani yechishdan oldin uning qo'yilishi oydinlashtiriladi, ya'ni bunda uning maqsadi va yechilish shartlari aniqlanadi, boshlang'ich ma'lumotlar va natijalarning tarkibi asoslanadi. Bu ma'lumotlar asosida u matematik formulalar ko'rinishida ifoda qilinadi.

2. Masalani yechishning sonli usulini tanlash. Qo'yilgan matematik masalalar uchun ularning sonli yechish usulini tanlash kerak bo'ladi. Sonli usullar turli-tuman bo'lganligidan ularning eng samarali va qulayini tanlash kerak. Bu masala bilan matematikaning sonli usullar bo'limi shug'ullanadi. Yechish usulini tanlash masalaga qo'yilgan barcha talablarni va uni konkret kompyuterlarda hal qilish imkoniyatlarini hisobga olish kerak.

3. Masalani yechish algoritmini ishlab chiqish. Masalani yechish uchun tanlangan sonli usulning algoritmi ishlab chiqiladi, ya'ni masalani yechish uchun bajariladigan arifmetik va mantiqiy amallar ketma-ketligi yoritiladi. Masalani yechish algoritmlari ko'rgazmalibroq bo'lishi uchun, ular ko'p hollarda blok- sxema ko'rinishida ifodalanadi.

4. Kompyuter uchun dastur tuzish. Kompyuter uchun dastur masalaning umumiy yechimidir. U algoritmning mashina buyruqlari ketma-ketligi shaklidagi yozuvdir. Buning uchun dasturlash tillari (Basic, Fortran, C++ , C++ va boshqalar) dan biri tanlanadi va unga mos dastur tuziladi. Tuzilgan dasturni sifatli bo'lishi va uni mashina xotirasidan kam joyni egallashi muhim ahamiyatga ega.

5. Dasturni kompyuter xotirasiga kiritish, rostlash va tekshirish. Tuzilgan dastur kompyuter klaviaturasi orqali uning xotirasiga

kiritiladi. Kiritilgan dasturni rostlash va tekshirish amalga oshiriladi, ya'ni yo'l qo'yilgan xatoliklar tuzatiladi.

6. Hisoblash natijalarini qayta ishlash va tahlil qilish. Bu bosqichdan tuzilgan dastur bo'yicha hisoblash bajariladi va hosil bo'lgan natija kompyuterning displey ekraniga chiqariladi yoki chop etish qurilmasi orqali qog'ozga chop etiladi. Natijalarni jadvallar, grafiklar yoki diagrammalar ko'rinishida hosil qilish mumkin. Hosil bo'lgan natija esa foydalanuvchi tomonidan tahlil qilinadi.

Qo'yilgan masalani u yoki bu turini yechishning algoritmlarini shakllantirish va ishlab chiqish eng ma'suliyatli hamda muhim bosqichlardan hisoblanadi, chunki bu bosqichda keyinchalik shaxsiy kompyuterda bajarilishi kerak bo'lgan amallarning ketma-ketligi oldindan belgilab olinadi. Algoritmida yo'lga qo'yilgan xatoliklar hisoblash jarayonini noto'g'ri bajarilishiga olib keladi, ya'ni noto'g'ri natijalarni beradi.

1.2.Algoritm tushunchasi.

Algoritm – bu masalani yechish usullarini izohlashdir, yoki boshqacha qilib aytganda, kutilayotgan natijalarni shaxsiy kompyuter tomonidan olish uchun bajarilayotgan hisoblash jarayolarining ketma-ketliklaridir.

Algoritm - bu biror masalani yechish uchun bajarilishi zarur bo'lgan buyruqlarning tartiblangan ketma-ketligidir. Har bir algoritm aniq va tugallangan qadamlarga bo'lingan bo'lishi kerak.

Algoritm deb, masalani yechish uchun bajarilishi lozim bo'lgan amallar ketma-ketligini aniq tavsiflaydigan qoidalar tizimiga aytiladi. Boshqacha aytganda, **algoritm** –boshlang'ich va oraliq malumotlarni masalani yechish natijasiga aylantiradigan jarayonni bir qiymatli qilib, aniqlab beradigan qoidalarining biror bir chekli ketma-ketligidir.

Buning mohiyati shundan iboratki, agar algoritm ishlab chiqilgan bo'lsa, uni yechilayotgan masala bilan tanish bo'lmagan biron bir ijrochiga, shu jumladan kompyuterga ham bajarish uchun topshirsa bo'ladi va u algoritmning qoidalariga aniq rioya qilib masalani yechadi.

Algoritm atamasi o'rta asrlarda yashab ijod etgan buyuk o'zbek matematigi Al-Xorazmiy nomidan kelib chiqqan. **Algoritm** so'zi al-Xorazmiyning arifmetikaga bag'ishlangan asarining dastlabki sahifadagi **“Dixit Algoritmi”** (**“dediki al-Xorazmiy”** ning lotincha ifodasi) degan jumladan kelib chiqqan. U o'zi kashf etgan o'nli sanoq tizimida IX asrning 825 yilidayoq to'rt arifmetika amallarini bajarish qoidalarini

bergan. Arifmetika amallarini bajarish jarayoni esa al-Xorazm deb atalgan. Bu atama 1747 yildan boshlab algorismus, 1950 yilga kelib algoritm deb ham ataldi. Bu yerda al-Xorazmiyning sanoq sistemasini takomillashtirishga qo'shgan hissasi, uning asarlari algoritm tushunchasining kiritilishiga sabab bo'lganligi o'quvchilarga ta'kidlab o'tiladi.

Qadimgi Gresiyalik matematik Evklid 2 ta natural A va B sonlarning eng katta umumiy bo'luvchisini topish algoritmini taklif etdi. Uning ma'nosi quyidagicha:

Katta sondan kichigini ayirish, natijani katta son o'rniga qo'yish va ikkala son tenglashguncha bu amalni takrorlash. Ushbu teng sonlar izlangan natijadir.

Evklid algoritmida A va B sonlarning eng katta umumiy bo'luvchisi ushbu sonlar ayirmasining eng katta bo'luvchisi hamda ikkala A,B sonlarning ham umumiy eng katta bo'luvchisi bo'lishligidan foydalanilgan.

Evklid algoritmining bu ifodasiga aniqlik yetishmaydi, shuning uchun uning konkretlashtirish zarur bo'ladi.

Haqiqiy Evklid algoritmi quyidagicha:

1. A sonni birinchi son deb, B sonni ikkinchi son deb qaralsin. 2-qadamga o'tilsin.
2. Birinchi va ikkinchi sonlarni taqqoslang. Agar ular teng bo'lsa, 5-qadamgao'tilsin, aks holda 3-qadamgao'tilsin.
3. Agar birinchi son ikkinchi sondan kichik bo'lsa, ularning o'zni almashtirilsin. 4-qadamgao'tilsin.
4. Birinchi sondan ikkinchi son ayirilsin va ayirma birinchi son deb hisoblansin. 2-qadamga o'tilsin.
5. Birinchi sonni natija sifatida qabul qilinsin. Tamom.

Bu qoidalar ketma-ketligi algoritmning tashkil etadi, chunki ularni bajargan ixtiyoriy ayirishni biladigan kishi ixtiyoriy sonlar jufti uchun eng katta umumiy bo'luvchini topa oladi.

Matematiklar uzoq vaqtlar davomida algoritmlarning bunday ifodalaridan keng foydalanib turli hisoblash algoritmlarini ishlab chiqdilar.

Masalan, kvadrat va kubik tenglamalar ildizlarini topish algoritmlari topildi. Asta-sekin olimlar qiyinroq masalalar ustida bosh qotirib, masalan, ixtiyoriy darajali algebraik tenglamalar ildizlarini topish algoritmlarini qidiradilar. Hatto, XVII asrda Leybnis ixtiyoriy

matematik masalani yechishning umumiy algoritmini topishga urinib ko'rgan. Ammo bunga o'xshash algoritmlarni ko'rishning iloji bo'lmagan va asta-sekin buning butunlay imkoni yo'q degan xulosaga kelingan.

Shunday bo'lishiga qaramay, algoritm tushunchasining aniq tavsifi berilmagunga qadar, masalaning algoritmik yechimsizligini isbotlash mumkin emas edi.

Algoritm tushunchasi juda qadim zamonlardan shakllanib kelgan. Shunga qaramay, asrimizning yarmiga qadar matematiklar bu ob'ekt haqida ma'lum bir qarashlarga qanoatlanib kelganlar. Algoritm atamasi matematiklar tomonidan faqat konkret masalalarni yechish bilan bog'liq holda olinar edi.

XX asr boshida matematika asoslarida vujudga kelgan qarama-qarshiliklar va muammolar ularni hal etishga qaratilgan turli konsepsiyalar va oqimlarning vujudga kelishiga olib keldi. 20- yillarga kelib, effektiv hisoblash masalalari ko'ndalang bo'ldi.

Algoritm tushunchasining o'zi matematik tadqiqotlar ob'ekti bo'lib qolganligi uchun aniq va qat'iy ta'rifga muhtoj edi. Bundan tashqari kompyuter asrini yaqinlashtiruvchi fizika va texnikaning rivojlanishi ham shuni taqozo etar edi.

Algoritmni mukammalroq tushunarli bo'lishi uchun o'quvchilarga turli hayotdan, fandan algoritmlarga misollar keltirish va bunga o'zlari tuzishga harakat qilishlarini taklif etish mumkin. Masalan, taom tayyorlash, turli qurilmalarni ishlatish, sport musobaqasi yoki yo'l harakati qoidalari algoritmlarini keltirish mumkin, yoki matematik formula bo'yicha qiymat hisoblash algoritmi yoki kompyuterni ishlatish bo'yicha algoritm kabi misollar keltirilishi mumkin.

1.3.Algoritmning asosiy xossalari.

Algoritmning asosiy xossalari haqida quyidagilarni ta'kidlash mumkin:

1-xossa. Diskretlilik, ya'ni algoritmni chekli sondagi oddiy ko'rsatmalar ketma-ketligi shaklida ifodalash mumkin. Tugallangan amallar ketma-ketligi qadam deyiladi. Demak, algoritm chekli qadamlardan iborat bo'lishi kerak.

2-xossa. Tushunarlilik, ya'ni ijrochiga tavsiya etilayotgan ko'rsatmalar uning uchun tushunarli bo'lishi shart, aks holda ijrochi oddiy amalni ham bajara olmay qolishi mumkin. Har bir ijrochining bajara olishi mumkin bo'lgan ko'rsatmalar tizimi mavjud.

3-xossa. Aniqlik, *ya'ni ijrochiga berilayotgan ko'rsatmalar aniq mazmunda bo'lishi lozim hamda faqat algoritmda ko'rsatilgan tartibda bajarilishi shart.* Algoritmning har bir qoidasi aniq va bir qiymatli bo'lishi zarurki, bunda vaqtning biror daqiqasida olingan miqdorlar qiymati vaqtning shundan oldingi daqiqasida olingan miqdorlar qiymati bilan bir qiymatli aniqlangan bo'ladi.

4-xossa. Ommaviylik, *ya'ni har bir algoritm mazmuniga ko'ra bir turdagi masalalarning barchasi uchun yaroqli bo'lishi lozim.* Algoritm bitta masalani yechish uchun emas balki shunga o'xshash turdosh masalalar sinfini hal etish uchun xizmat qiladi

Masalan, ikki oddiy kasr umumiy maxrajini topish algoritmi har qanday kasrlar umumiy maxrajini topish uchun ishlatiladi.

5-xossa. Natijaviylik, *ya'ni har bir algoritm chekli sondagi qadamlardan so'ng albatta natija berishi lozim.* Algoritm masalaning yechimiga chekli sondagi qadamlar ichida olib kelishi yoki masalani "yechib bo'lmaydi" degan xabar bilan tugashi kerak.

Bu xossalar mohiyatini o'rganish va konkret algoritmlar uchun qarab chiqish talabalarning xossalar mazmunini bilib olishlariga yordam beradi.

1.4.Algoritmning tasvirlash usullari.

Algoritmning tasvirlash usullari haqida gapirganda algoritmning berilish usullari xilma-xilligi va ular orasida eng ko'p uchraydiganlari quyidagilar ekanligini ko'rsatib o'tish joiz:

1. Algoritmning so'zlar orqali ifodalanishi. Algoritmnlarni ifodalashning eng keng tarqalgan shakli – oddiy tilda so'zlar bilan bayon qilishdir. Bu nafaqat hisoblash algoritmlarida, balki hayotiy, turmushdagi "algoritm"larga ham tegishlidir.

Masalan, biror bir taom yoki qandolat mahsulotini tayyorlashning retsipti ham oddiy tilda tavsiflangan algoritmdir. Shaharlararo telefon - avtomat orqali aloqa o'rnatishning o'ziga xos algoritmidan foydalanasiz. Do'kondan yangi kir yuvish mashinasi yoki magnitafon sotib olinsa, ishni foydalanishning algoritmi bilan tanishishdan boshlaymiz.

Masalani kompyuterda yechishda ham, ko'pincha matematika tilini ham o'z ichiga olgan tabiiy tildan foydalanish mumkin. Algoritmning bunday tildagi yozuvi izlanayotgan natijaga olib keladigan amallar ketma-ketligi ko'rinishida bo'lib, odam tomonidan bir ma'noli idrok etilishi kerak. So'zlar bilan ifodalan gan har bir amal “algoritmning qadami” deb ataladi. Qadamlar tartib nomeriga ega bo'ladi.

Algoritm ketma-ket, qadam-ba qadam bajarilishi kerak. Agar algoritm matnida "N sonly qadamga o'tilsin" deb yozilgan bo'lsa, bu algoritmning bajarilishi ko'rsatilgan N-qadamdan davom etishini bildiradi.

2. Algoritmning formulalar yordamida berilishi. Algoritm matematik formulalar yordamida tavsiflanganda har bir qadam aniq formulalar yordamida yoziladi. Misol tariqasida

$$ax^2 + bx + c = 0 \quad (a \neq 0)$$

Kvadrat tenglama yechimlari bo'lmish x_1, x_2 ni aniqlash algoritmini ko'rib chiqaylik.

1. a, b, c koeffitsiyentlar qiymatlari berilsin.
2. $D = b^2 - 4ac$ diskriminant hisoblansin.
3. $D < 0$ bo'lsa, tenglamaning haqiqiy yechimlari yo'q. Faqat haqiqiy ildizlar izlanayotgan bo'lsa, masala hal bo'ldi.
4. $D = 0$ bo'lsa, tenglama ikkita bir-biriga teng, ya'ni karrali yechimga ega bo'ladi va ular formulalar bilan hisoblanadi. Masala hal bo'ldi.

5. $D > 0$ bo'lsa, tenglama ikkita haqiqiy yechimga ega, ular

$$x_1 = (b + \sqrt{D}) / 2a \quad \text{va} \quad x_2 = (-b - \sqrt{D}) / 2a$$

formulalar bilan hisoblanadi. Ya'ni masala hal bo'ldi.

3. Algoritmning jadval ko'rinishida berilishi. Algoritmning bu tarzda tasvirlanishdan ham ko'p foydalanamiz. Masalan, maktabda qo'llanib kelinayotgan to'rt xonali matematik jadvallar yoki turli xil lotereyalar jadvallari. Funksiyalarning grafiklarini chizishda ham algoritmning qiymatlari jadvali ko'rinishlaridan foydalanamiz. Bu kabi jadvallardan foydalanish algoritmлари sodda bo'lgan tufayli ularni o'zlashtirib olish oson.

4. Algoritmning dastur shaklida ifodalanishi, ya'ni *algoritm kompyuter ijrochisiga tushunarli bo'lgan dastur shaklida beriladi.*

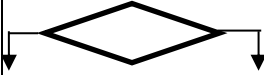
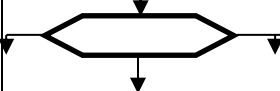
5. Algoritmning algoritmik tilda tasvirlanishi, ya'ni algoritm bir xil va aniq ifodalash, bajarish uchun qo'llanadigan belgilash va qoidalar majmui algoritmik til orqali ifodalashdir. Algoritmik tillar - algoritmni bir ma'noli tavsiflash imkonini beradigan belgilar va qoidalar majmuidir. Har qanday tillardagidek ular ham o'z alifbosi, sintaksisi va semantikasi bilan aniqlanadi

Bizga o'rta maktabdan ma'lum bo'lgan kompyutrlarsiz algoritmishga mo'ljallangan algoritmik tizim algoritmik tilning namunasidir. Algoritmik tilga misol sifatida yana algoritmлarni belgili operatorlar tizimi shaklida tavsiflashni ham ko'rsatish mumkin. Bu tillar odatdagi tilga

o'xshash bo'lib, kompyuterda bevosita bajarishga mo'ljallanmagan. Ulardan maqsad algoritmni bir xil shaklda va tushunarli qilib, tahlil qilishga oson qilib yozishdir.

6. Algoritmning grafik shaklda tasvirlanishi. Masalan, grafiklar, sxemalar ya'ni blok - sxema bunga misol bo'la oladi. Blok - sxemaning asosiy elementlari quyidagilar: **oval (ellips shakli)**-algoritm boshlanishi va tugallanishi, **to'g'ri burchakli to'rtburchak**-qiymat berish yoki tegishli ko'rsatmalarni bajarish. **Romb** - **shart** tekshirishni belgilaydi. Uning yo'naltiruvchilari tarmoqlar bo'yicha biri ha ikkinchisi yo'q yo'nalishlarni beradi, **parallelogramm**- ma'lumotlarni kiritish yoki chiqarish, **yordamchi algoritmgacha murojaat** - parallelogramm ikki tomoni chiziq, **yo'naltiruvchi chiziq** - blok-sxemadagi harakat boshqaruvi, **nuqta-to'g'ri chiziq (ikkita parallel)** - qiymat berish.

Algoritmida bajarilishi tugallangan amallar ketma-ketligi **algoritm qadami** deb yuritiladi. Har bir alohida qadamni ijro etish uchun bajarilishi kerak bo'lgan amallar haqidagi ko'rsatma buyruq deb aytiladi. Algoritmni ko'rgazmaliroq qilib tasvirlash uchun blok-sxema, ya'ni geometrik usul ko'proq qo'llaniladi. Algoritmning blok-sxemasi algoritmning asosiy tuzilishining yaqqol geometrik tasviri: algoritm bloklari, ya'ni geometrik shakllar ko'rinishida, bloklar orasidagi aloqa esa yo'naltirilgan chiziqlar bilan ko'rsatiladi. Chiziqlarning yo'nalishi bir blokdan so'ng qaysi blok bajarilishini bildiradi. Algoritmni ushbu usulda ifodalashda vazifasi, tutgan o'rniga qarab quyidagi **geometrik shakl (blok)** lardan foydalaniladi.

Blokning atalishi	Belgilanishi	Tushunilishi
Hisoblashlar bloki (to'g'ri-to'rtburchak)		Hisoblash amali yoki hisob-lash amallari ketma-ketligi
shartli blok (romb)		Shartlarni tekshirish
siklik jarayon (oltiburchak)		Siklning boshlanishi
qism dastur		qism dastur bo'yicha hisoblash, standart qism dasturi
birlashtirish (aylana)		Yo'nalish chizig'ini o'zgartirish

Ma'lumotlarni kiritish va chiqarish parallelogramm)		Ma'lumotlarni kiritish va natijalarni chiqarish
Algoritmning boshi va oxiri (oval)		Boshlash, tamom, to'xtash
Chiqarish bloki		Ma'lumotlarni qog'ozga chiqarish

Blok-chizmalarning asosiy afzalligini, algoritmni tasvirlashning yaqqolligi bilan izohlash hisoblanadi.

Masalani yechishning algoritmlarini blok-chizmalar asosida ifoda etib ularning yaqqolligini ta'minlash

- bloklarni gorizontaal va vertikal holatda joylashtirish kerak;
- bloklar orasidagi masofani shunday olish kerakki, ularni ulab turuvchi chizig'lar yetarli darajada kalta bo'lsin;
- bloklarning geometrik o'lchami hisoblash jarayonini tavsiflashning hajmiga to'g'ri kelishi kerak. Shu bilan birgalikda blokning uzunligi uning balandligiga nisbatan $v=1,5 \cdot a$ ko'rinishda bo'lishi kerak;

d) blok-chizmalarda bloklarning joylashishi va guruhlariga ajratilishi shunday bo'lishi kerakki, ularni ko'rib axborotlarni yaxshi qabul qilish ta'minlanishi lozim

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

Nazorat uchun savollar

- Algoritm tushunchasi qanday paydo bo'lgan?
- Algoritm deb nimaga aytiladi?
- Algoritm nima uchun ishlatiladi?
- Kompyuter bilan algoritmning bog'liqligi nimada?
- Algoritmni aniqlashtirish bilan kimlar shug'ullanganlar?
- Algoritmlarga misollar keltiring.
- Yevklid algoritmi nima uchun ishlatiladi?
- Masalalarning umumiy algoritmini topish bilan kim shug'ullangan?

9. Algoritim qanday xossalarga ega?
 10. Diskretlik xossasini izohlang.
 11. Natijaviylik nima?
 12. Ommaviylik xossasiga misol ayting.
 13. Tushunarlik va anqlik xossalarini izohlang.
 14. Algoritm ijrochisi kim?
 15. Algoritm xossalariga misollar keltiring va ularni izohlang.
 16. Kibernetika va informatika sohalari bo'yicha o'zbek olimlaridan kimlarni bilasiz?
1. Algoritm qanday ko'rinishlarda beriladi?
 2. Algoritmni formula orqali bilishni tushuntiring?
 3. Algoritmni jadval ko'rinishiga misollar keltiring.
 4. Blok-sxema nima?
 5. Algoritmik til nima?
 6. Algoritmilar dasturlash tillari orqali qanday beriladi?
 7. Algoritmning qaysi ko'rinishi samaraliroq bo'ladi?
 8. Algoritmilar kim uchun tuziladi?
 9. Blok-sxemaning tarkibiy qismlarini tushuntiring.
 10. Algoritmni tasvirlar usullarining maqsadi nima?

Mustaqil topshiriqlar

1. Choy damlash jarayonini matn usulida yozing.
2. Telefon orqali so'zlashuv jarayonini matn usulida ifodalang.
3. $y=2x$ funksiyani qiymatlar jadvalini tuzing va izohlang.
4. Taom tayyorlash jarayonini yozing.
5. Yo'l harakati qoidasiga rioya qilish jarayonini tasvirlang?
6. $y=x^2$ funksiyani grafigini chizing va uni izohlang.

2§. Algoritm turlari. Algoritmash jarayoni.

“Algoritmash” deganda masalani biri ketidan boshqasini bajariladigan hamda oldingisining natijalari keyingilarining bajarilishida ishlatiladigan bosqichlar ketma-ketligiga keltirish tushuniladi. Ayni paytda bu bosqichlardagi amallarni kompyuter bajara olishi ko'zda tutilishi kerak.

Kengroq ma'noda qaraydigan bo'lsak algoritmash, o'zidan oldingi bosqich - masalani yechish usulini tanlash bosqichi ham, o'zidan keyingi bosqich - kompyuterning xususiyatlarini hisobga olgan holda

boshlang'ich, oraliq va natijaviy axborotlarni tuzilishining ifoda shakllarini tanlashni xam o'z ichiga oladi.

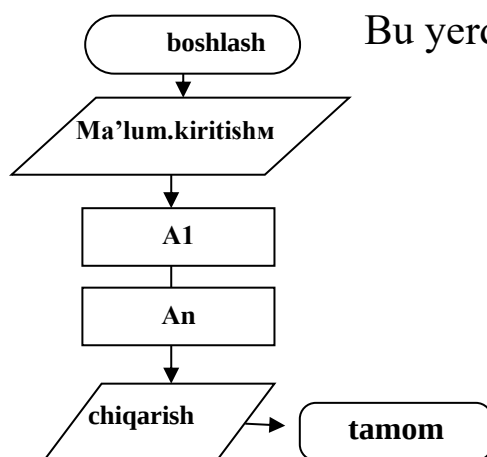
Algoritmning bosqichining natijasi masalani yechish algoritmi bo'ladi, ya'ni bu bosqichda masalani yechish algoritmi ishlab chiqiladi. Bunda masalani matematik qo'yilishi va tanlangan usul qidirilayotgan natijani olishga olib keladigan harakatlar ketma-ketligini aniqlash uchun asos bo'lib xizmat qiladi.

Algoritmning turlari bilan tanishtirganda, avvalo hech qanday shart tekshirilmaydigan va tartib bilan faqat ketma-ket bajariladigan jarayonlarni ifodalaydigan **chiziqli algoritmlar** aytilib o'tiladi.

2.1. Chiziqli jarayonlarni algoritmash

Chiziqli algoritmlar algoritmning eng sodda va oddiy ko'rinishi hisoblanadi. Unda bajariladigan amallar ham buyruqlar ham buyruqlar ham qanday tartibda berilgan bo'lsa shunday tartibda ketma-ket bajariladi. Chiziqli tuzilishga ega bo'lgan algoritmlarda ko'rsatmalar yozilish tartibida bajariladi. Ularning blok-sxemalari ishga tushirish, to'xtatish, kiritish, chiqarish jarayoni blok hamda avvaldan ma'lum jarayon bloklari yordamida tuzilib, bir chiziq bo'ylab ketma-ket joylashgan bo'ladi.

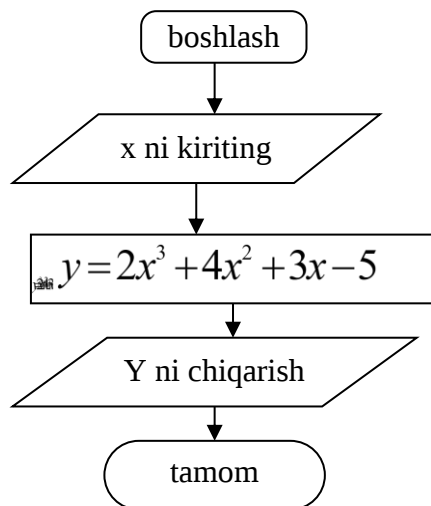
Chiziqli tuzilishdagi algoritmni tuzish masalani yechish uchun kerak bo'ladigan boshlang'ich ma'lumotlarni tashkil qiluvchi o'zgaruvchilar nomi, ularning turi va o'zgarish ko'lamini aniqlashdan boshlanadi. Keyin oraliq va yakuniy natijalar o'zgaruvchilarining nomlari, turlari va mumkin bo'lsa o'zgarish ko'lamini aniqlash kerak. Endi algoritm mana shu boshlang'ich ma'lumotlarni qanday qayta ishlab oraliq va yakuniy natijalarni olish kerakligini aniqlashdan iborat bo'ladi. Chiziqli algoritmlarni quyidagi ko'rinishda ifodalash mumkin.



Bu yerda $A_1, \dots, A_N$ lar chiziqli algoritmlarda bajarilishi kerak bo'lgan buyruqlar ketma-ketligidir.

1- misol. $y = 2x^3 + 4x^2 + 3x - 5$ funksiyani x ning ixtiyoriy qiymatlarida hisoblash algoritmini tuzing.

Yechish. Algoritmning blok sxema Algoritmning matn ko'rinishi:

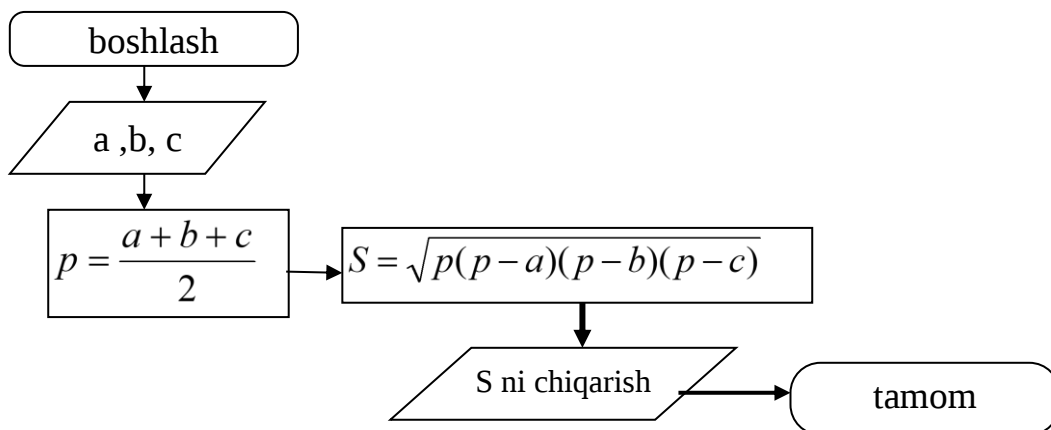


1. Boshlash.
2. x ni kiritish.
3. $y = 2x^3 + 4x^2 + 3x - 5$ ni qiymatini hisoblash.
4. y ni qiymatini chiqarish.
5. Tamom

2-misol. $S = \sqrt{p(p-a)(p-b)(p-c)}$ Geron formulasidan foydalanib uchburchak yuzini hisoblash algoritmini tuzing.

Yechish. Bu yerda boshlang'ich ma'lumotlar: a, b, c uchburchak tomonlari. Shuning uchun $a, b, c \in R$ va $a > 0, b > 0, c > 0$, ya'ni a, b, c — o'zgaruvchilar nomi; ular haqiqiy son qiymatlar qabul qiladi. Yana, bu uch son uchburchak tomonlarini ifoda qilishi uchun ularning istalgan biri qolgan ikkitasi yig'indisidan katta bo'lmasligi, ya'ni $a < b + c, b < a + c, c < a + b$ shartlar bajarilishi kerak. Shunday qilib, o'zgarish ko'lami ana shu munosabatlar bilan aniqlanar ekan.

Geron formulasi bilan uchburchak yuzini hisoblash uchun uning yarim perimetrining qiymati kerak. Demak, p o'zgaruvchining qiymatioraliq ma'lumot bo'ladi. Yuqoridagi shartlarda $p > b$. Yakuniy natija: S -uchburchak yuzi $S > 0$ qiymatlar qabul qiladi.



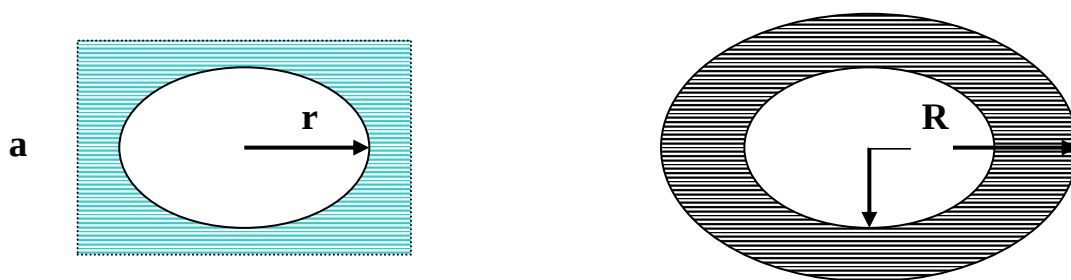
Nazorat uchun savollar

1. Algoritm tushunchasi qanday paydo bo'lgan?

2. Algoritmga qo'yiladigan talablarni sanab o'ring.
3. Algoritmning qanday xossalari bilasiz?
4. Masala yechimi algoritmini tuzishda qanday qoidalarga amal qilish lozim?
5. Chiziqli algoritm deb nimaga aytiladi?
6. Chiziqli algoritmni bajarilish tartibini ayting.

Mustaqil bajarish uchun mashqlar

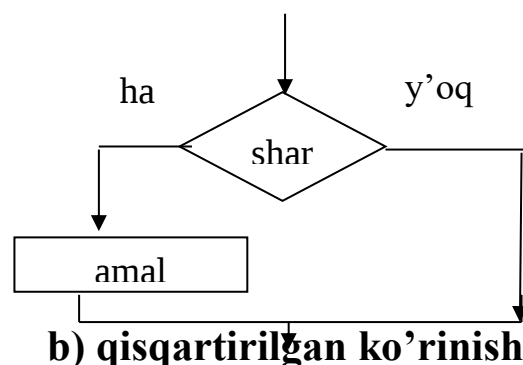
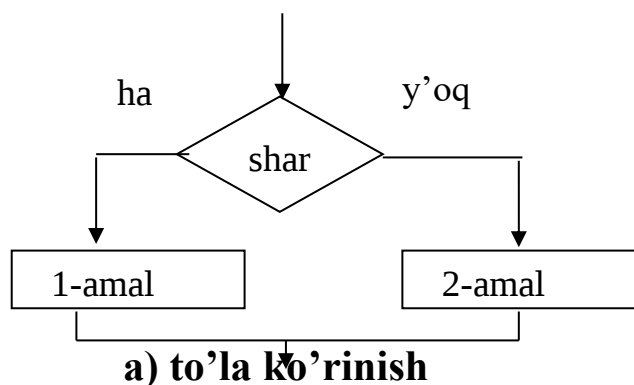
1. Asosi uzunligi va balandligi bilan berilgan uchburchak yuzasini hisoblash algoritmini tuzing.
2. To'g'ri to'rtburchakning perimetri va yuzasini hisoblash algoritmini tuzing.
3. Jismning potensial energiyasini hisoblash algoritmini tuzing.
4. Tomonlaridan biri va balandligi bilan berilgan parallelogramning yuzasini hisoblash algoritmini tuzing.
5. Kvadratning perimetri va yuzasini hisoblash algoritmini tuzing.
6. Jismning kinetik energiyasini hisoblash algoritmini tuzing.
7. Asoslarining yig'indisi va balandligi berilgan trapetsiyaning yuzasini hisoblash algoritmini tuzing.
8. Teng tomonli uchburchakning perimetri, medianasi va yuzasini hisoblash algoritmini tuzing.
9. Tomonlari va ular orasidagi burchgi berilgan parallelogramning diagonallarini hisoblash algoritmini tuzing.
10. Uch tomoni bo'yicha berilgan uchburchakning yuzasini hisoblash algoritmini tuzing.
11. To'g'riburchakli parallelopipedning hajmi, to'la sirtining yuzi, qirralar yig'indisini hisoblash algoritmini tuzing.
12. Radiusu berilgan aylanaga ichki chizilgan to'g'ri oltiburchakning perimetri va yuzasini hisoblash algoritmini tuzing.
13. a, b, c sonlarining o'rta arifmetigi va o'rta geometrigini hisoblash algoritmini tuzing.
14. Geometrik progressiyaning birinchi hadi va maxraji berilgan bo'lsa, uning n –hadi va n ta hadlar yig'indisini hisoblash algoritmini tuzing.
15. Quyida berilgan shakllarning shtrixlangan sohalarini yuzalarini hisoblash algoritmini tuzing.



2.2. Tarmoqlanuvchi jarayonlarni algoritmlash.

Shunday hisoblash jarayonlari mavjud bo'ladiki, bunda qo'yilgan ayrim mantiqiy shartlarning bajarilishiga qarab, bu jarayonlar bir nechta tarmoqqa bo'linadi. Hisoblash jarayonlarining shundayiga tarmoqlangan deb ataladiki, unda u birlamchi yoki oraliq ma'lumotlar xususiyatidan kelib chiqqan holda bir yoki bir necha yo'nalish bo'yicha bajarilishi mumkin bo'ladi. Bunda har bir yo'nalish hisoblash jarayonining tarmog'i hisoblanadi. U yoki bu tarmoqning tanlanishi mantiqiy shartlarning bajarilishini tekshirish asosida ta'minlanadi. Aniq bir holda jarayon faqat tarmoqlarning bittasi bo'yicha bajariladi. Boshqa tarmoqlanish-larning bajarilishi mumkin emas. Tarmoqlanuvchi struktura odatda qandaydir mantiqiy shartni tekshirish blokini o'z ichiga oladi. Tekshirish natijasiga ko'ra, tarmoq deb ataluvchi u yoki bu amallar ketma-ketligi bajariladi va shu tarmoqlardan hyech bo'lmaganda bittasi bajariladi. Shartni tekshirish natijasi faqat ikki hil bo'lganda: bajarilgan hoi uchun «Ha» (yoki «+»), bajarilmagan hoi uchun «Yo'q» (yoki «—») belgilari qo'yiladi.

Tarmoqlanish matematik ifoda qiymatining ishorasi bo'yicha bo'lganda (arifmetik shart): «>» — musbat, «<» — manfiy va «=» — nolga teng belgilar qo'yiladi. Ana shunday jarayonlar uchun algoritmlar tuzishda tarmoqlanuvchi algoritmlardan foydalaniladi. Tarmoqlanuvchi algoritm to'la va qisqartirilgan ko'rinishda berilishi mumkin. Ular quyidagicha sxema orqali ifodalanadi:



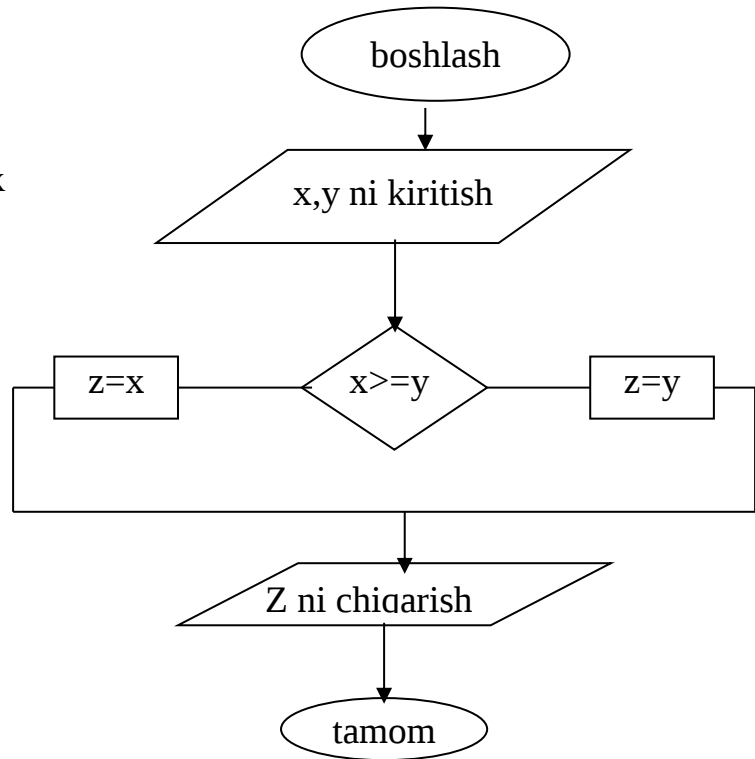
Tarmoqlanuvchi jarayon uchun algoritm tuzishga doir misollar keltiramiz.

1-misol. Ixtiyoriy berilgan x va y sonlaridan eng kattasini topish algoritmini tuzing.

Yechish. Buning uchun berilga ikkita sonni taqqoslaymiz: agar $x \geq y$ bo'lsa u holda x sonini katta son $z=x$ deb, aks holda esa y sonini katta son $z=y$ deb qabul qilamiz.

Algoritmning matn va blok sxema ko'rinishlari:

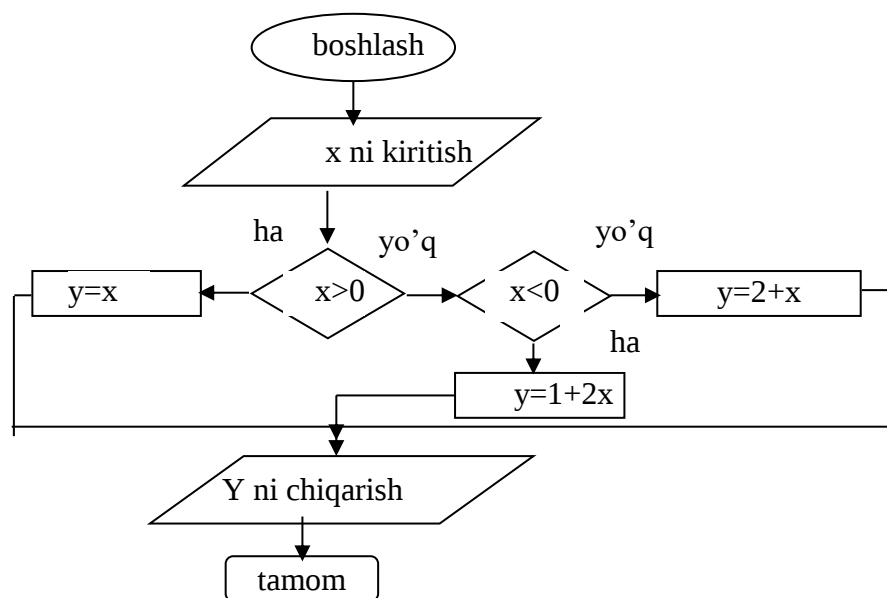
1. Boshlash.
2. x va y qiymatlarni kiritish.
3. agar $x \geq y$ bo'lsa, natija x deb olinib 5 ga o'tilsin.
4. natija y deb olinsin.
5. Tamom.



2- misol.
$$\begin{cases} 1+2x, & \text{agar } x < 0 \\ 2+x, & \text{agar } x = 0 \\ x, & \text{agar } x > 0 \end{cases}$$
 ifodaning qiymatini hisoblash algoritmini

tuzing.

Yechish. Bu yerda x ni qiymati kiritilgandan so'ng u 0 soni bilan taqqoslanadi. Agar $x < 0$ bo'lsa $y=1+2x$, $x=0$ bo'lsa $y=x+1$, $x > 0$ bo'lsa $y=2+x$ ifodani qiymati hisoblanishi kerakligi ko'rsatiladi.



Nazorat uchun savollar

1. Tarmoqlanuvchi jarayon nima ?
2. Tarmoqlanuvchi jarayon qanday bajariladi ?
3. Tarmoqlanuvchi algoritmni ta'riflang va unga misollar keltiring.
4. Taqmoqlanuvchi jarayon qanday ko'rinishlarda beriladi?
5. To'la tarmoqlanuvchi jarayonni umumiy strukturasini tushuntirng.
6. Qisqartirilgan tarmoqlanuvchi jarayonni umumiy strukturasini ko'rsating.
7. Tarmoqlanuvchi jarayon uchun blok-sxemaning qaysi elementlaridan foydalaniladi ?
8. Algoritmning qadami nima ? Uning tarmoqdan farqi qanday ?

Mustaqil bajarish uchun mashqlar

1. Kvadrat tenglama ildizlarini topish algoritmini tuzing .
2. Berilgan a, b, c, d sonlaridan kattasini topish algoritmini tuzing.
3. a, b, c sonlarini o'sish tartibida joylashtirish algoritmini tuzing.
4. a, b, c kesmalar uzunliklari bilan berilgan. Ular orqali uchburchak yasash mumkin yoki mumkin emasligini aniqlash algoritmini tuzing.
5. Yuzasi S_1 ga teng bo'lgan doira, yuzasi S_2 ga teng bo'lgan kvadtaga joylashishini aniqlash algoritmini tuzing .
6. Koordinatalari bilan berilgan $a(x, y)$ nuqtani koordinatalar tekisligida joylashish o'rnini aniqlash algoritmini tuzing.
7. a, b, c sonlari berilgan bo'lsa, ularning manfiylarining kubi, musbatlarining kvadratidan hosil bo'lgan sonlarni chop etish algoritmini tuzing.
8. Berilgan ikkita a va b sonlarining kichigini ularning yig'indisi, kattasini esa ularning ko'paytmasi bilan almashtirish algoritmini tuzing.

9. Tomonlari bilan berilgan uchburchakning teng tomonli bo'lishini aniqlash algoritmini tuzing.

10. $S = \max(x, y, z) + \min(a, b, c)$ ni hisoblash algoritmini tuzing.

11. Funksiya qiymatini hisoblash algoritmini ruzing:

$$z = \begin{cases} (2x + y + 3)/3, & \text{agar } x + y > 0 \text{ va } x > 0 \\ \sqrt{x^2} + 2y^{2+3}, & \text{agar } x + y > 0 \text{ va } y < 0 \\ (x^2 + y^2)/x, & \text{qolgan hollarda} \end{cases}$$

12. Tomonlari a,b,c larga teng bo'lgan uchburchakning turini aniqlash algoritmini tuzing.

13. a,b,c,d sonlarini kamayish tartibida joylashtirish algoritmini tuzing.

14. a(x,y) nuqtani uchlari koordinatalari a(x1,y1) va b(x2,y2) ga teng bo'lgan kesmaga tegishli bo'lishligini aniqlash algoritmini tuzing.

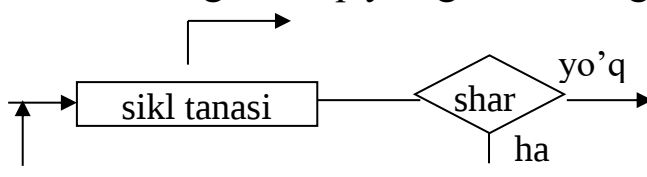
15. A(x,y) nuqtani radiusu R ga, markazi O(0,0) nuqtada joylashgan aylanaga tegishli bo'lishligini aniqlash algoritmini tuzing.

2.3. Takrorlanuvchi jarayonlarni algoritmlash.

Shunday hisoblash jarayonlari mavjud bo'ladiki, bunda uning ayrim bo'laklarini bir necha marta takroran hisoblashga to'g'ri keladi. Bunday jarayonlar takrorlanuvchi yoki siklik jarayonlar deyiladi va ular uchun algoritmlar tuzishda takrorlanuvchi algoritmlardan foydalaniladi. Siklik jarayonlar sikl parametri va sikl tanasidan iborat bo'ladi. Hisoblash jarayonining ko'p marta takrorlanadigan qismi ichki **sikl tanasi** deb yuritiladi. Sikl tanasi bir necha marta takroran bajariluvchi amallar ketma-ketligidan iborat bo'ladi. Siklli jarayonlarni algoritmlashda bitta yoki bir nechta parametrlar qatnashadi. Ularning qiymatlarini bir vaqtda o'zgarishida sikl tanasi ichidagi amallar ketma-ketligi ko'p marotaba takrorlanadi. Masalaning qo'yilishiga qarab siklik jarayonlartakrorlanish sonima'lum bo'lgan va takrorlanish soni noma'lum bo'lgan sikllarga bo'linadi.

Takrorlanuvchi algoritmlar 2 xil ko'rinishda ifodalanishi mumkin.

Sikl-gacha takrorlanuvchi algoritm quyidagi ko'rinishga ega:

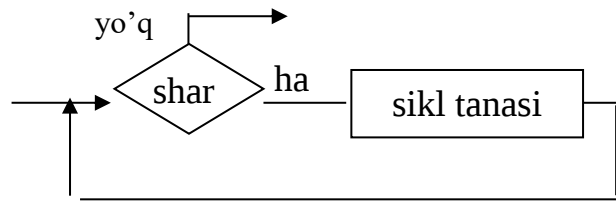


a) sikl - gacha

Bu ko'rinishdagi algoritmda avval sikl tanasi bajarilib, so'ngra parametr-ning qiymati, ya'ni sikldan chiqish sharti tekshiriladi

Bu yerda sikl tanasi qo'yilgan shart bajarilib turguncha takrorlanaveradi.

Sikl - hozircha takrorlanuvchi algoritm quyidagi ko'rinishga ega:



b) sikl - hozircha.

Bu ko'rinishdagi algoritmlarda avval shart tekshiriladi, so'ngra agar shart qanoatlantirsa, sikl tanasi bajariladi, aks holda hisoblash to'xtatiladi.

Aniq berilgan son asosida sikllarni tashkil qilishda sikl parametrining boshlang'ich va oxirigi qiymatlari, uning har bir takrorlanishidagi sikl parametrining o'zgarish qonunlari, sikllarning takrorlanish sonlari ko'rsatilishi kerak bo'ladi.

Sikl tanasidagi birlamchi ma'lumotlar doimiy kattalik, oddiy o'zgaruvchan, indeksli o'zgaruvchan ko'rinishlarida bo'lishi mumkin.

1-misol. $S = \sum_{i=1}^{10} \frac{1}{i^2} = 1 + \frac{1}{4} + \frac{1}{9} + \dots + \frac{1}{100}$ yig'indini hisoblash

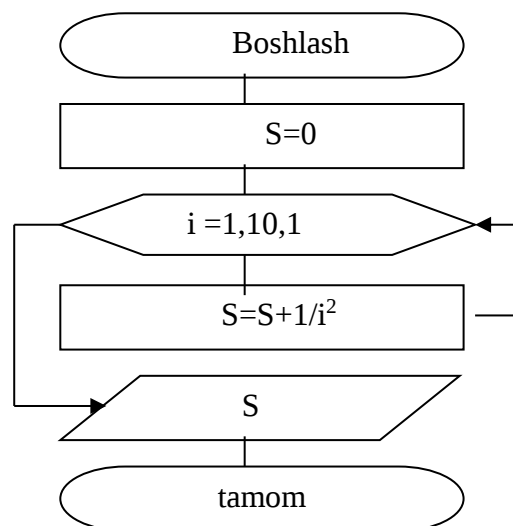
algoritmini tuzing.

Yechish: Bu yerda sikl parametrlarning barcha qiymatlari ma'lum. Yig'indini takroriy qiymatini hosil qilish uchun uning boshlang'ich qiymati sifatida $S=0$ olinadi. Sikl parametri i bo'lib u 1 dan 10 gacha qiymatlarni qabul qiladi. Uning har bir qiymatida $1/i^2$ ifoda takroran $S=S+1/i^2$ hisoblanadi.

Matin ko'rinish:

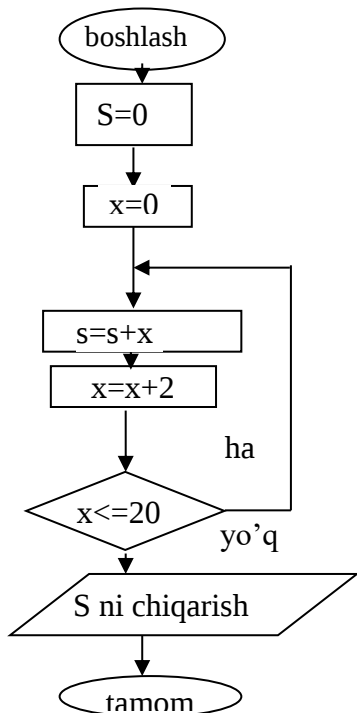
1. Boshlash.
2. $S=0$
3. $i=1$
4. $S=S+1/i^2$
5. Agar $i < 100$ bo'lsa, u holda $i=i+1$ ba 4ga o'tish. Aks holda 6 ga.
6. S ni qiymati chiqarilsin.
7. Tamom.

Blok -sxema ko'rinishi:

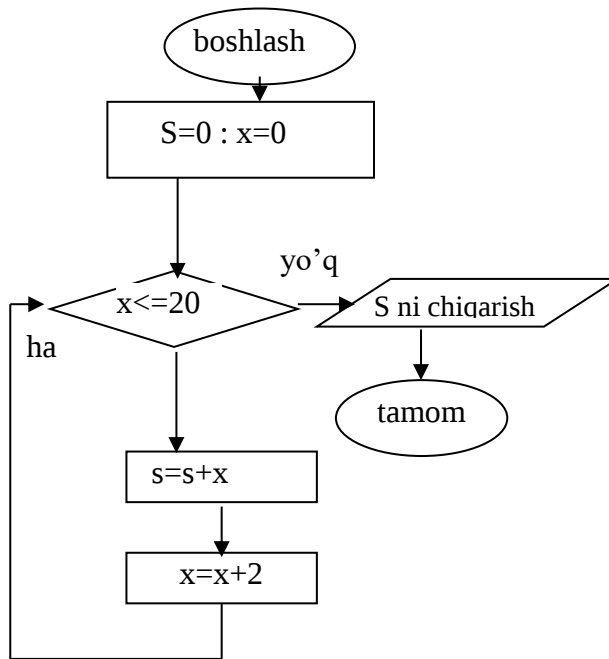


2 - misol. 0 dan 20 gacha juft sonlar yig'indisini hisoblash algoritmini tuzing.

Yechish: a) sikl-gacha



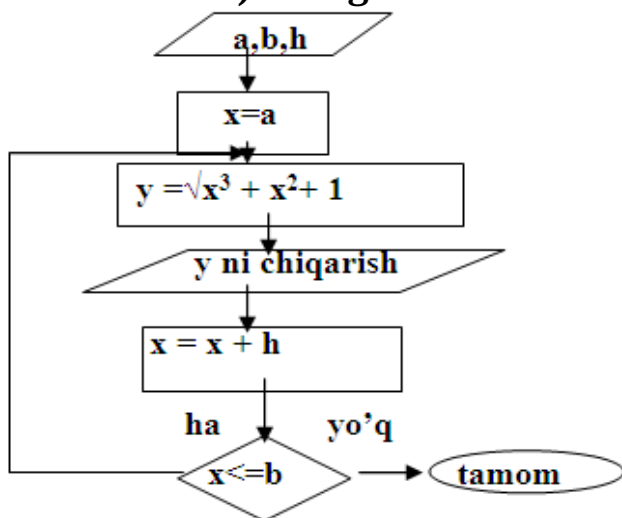
b) sikl – hozircha



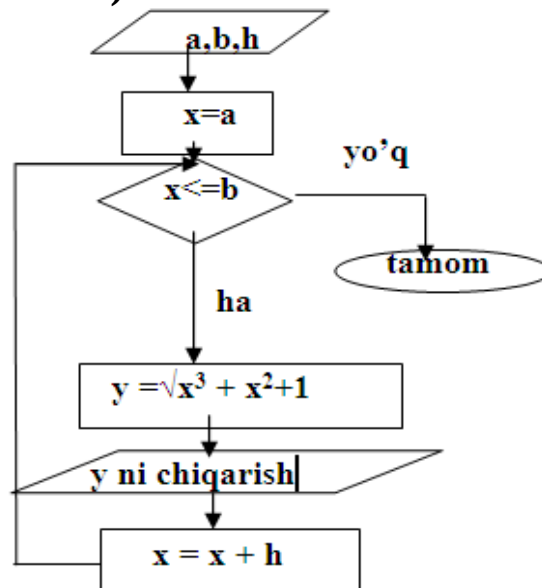
2-misol. $y = \sqrt{x^3 + x^2 + x}$ funksiyani x ning $[a,b]$ oraliqda $h=0.1$ qadam bilan o'zgarish qiymatini hisoblash algoritmini tuzing.

Yechish. Bu yerda algoritmlarning gacha va hozircha ko'rinishlari keltirilgan.

1) Sikl-gacha



2) Sikl – hozircha



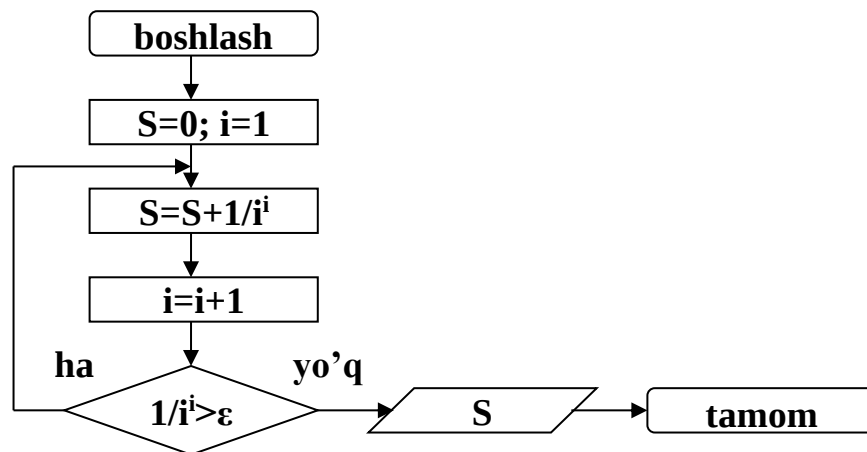
Takrorlanishlar soni aniq bo'lmagan sikllar. Bunday sikllarda takrorlanishlar sonini oldindan belgilab bo'lmaydi. Unda sikl ma'lum bir shart bajarilishigacha davom etadi.

Takrorlanishlar soni aniq bo'lmagan sikllarga interision sikllarni misol qilish mumkin bo'lib, unda sikldan chiqib ketish aniq bir natijaga erishilgandan keyin amalga oshiriladi.

Hisoblashlar erishish kerak bo'lgan natijaga bosqichma-bosqich yaqinlashib borish orqali bajariladi.

3-misol. $S = \sum_{i=1}^{\infty} \frac{1}{i^i}$ ifodani ε aniqlik bilan hisoblash algoritmini tuzing.

Yechish. Bu yerda $1/i^i$ ifoda sikl parametrlarning qiymati oshib bor-gan sari nolga yaqinlashib boradi. Hisoblash jarayoni $1/i^i \leq \varepsilon$ shart ba-jarilguncha davom etadi.

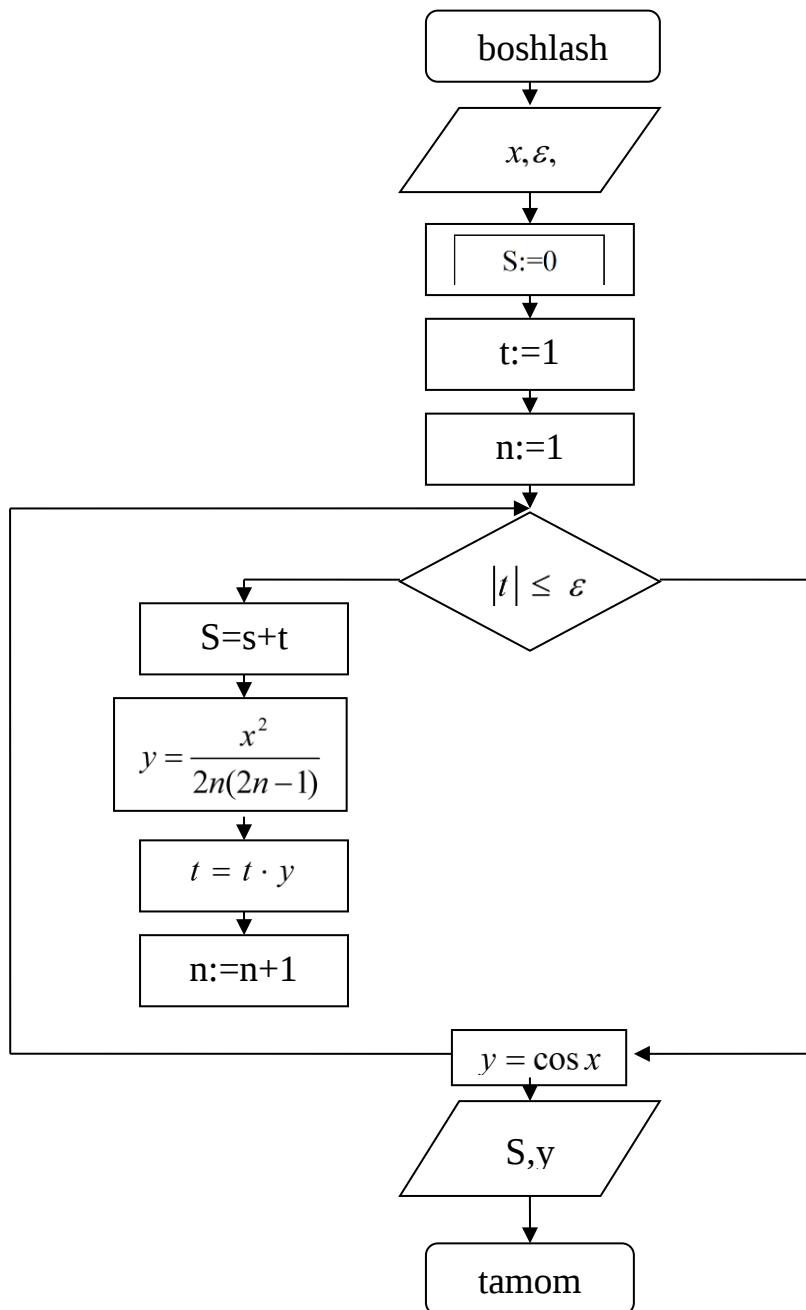


4-misol. Kosinusni qatarga yoyishdan foydalanib $S = \cos x$ funksiya qiymatini $\varepsilon = 10^{-4}$ xatolik bilan hisoblash algoritmini tuzing:

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots = \sum_{n=0}^{\infty} t_n(x), \text{ bu yerda } t_n(x) = (-1)^n \frac{x^{2n}}{(2n)!}$$

Yechish. Bu yig'indini hisoblash uchun, birinchidan, t_n hadning keyingi qiymatini aniqlash, ikkinchidan, iterasion formula $S_n = S_{n-1} + t_n$ bo'yicha yig'indini hisoblash.

Ushbu misolda faktorialni mavjudligini hisobga olganda, t_n ni aniqlash uchun to'g'ridan – to'g'ri umumiy formula bo'yicha hisoblash emas, balki recurrent munosabat $t_n = t_{n-1} \varphi_n$ dan foydalaniladi



Nazorat uchun savollar

1. Takrorlanuvchi jarayon nima ?
2. Takrorlanuvchi jarayon qanday bajariladi ?
3. Takrorlanuvchi algoritmni ta'riflang va unga misollar keltiring.
4. Takrorlanuvchi jarayon qanday ko'rinishlarda beriladi?
5. Hozircha tarmoqlanuvchi jarayonning umumiy strukturasini tushuntiring.
6. Gacha tarmoqlanuvchi jarayonning umumiy strukturasini ko'rsating.

7. Takrorlanuvchi jarayon uchun blok-sxemaning qaysi elementlaridan foydalaniladi ?
8. Algoritmning ko'rsatmasi nima ?
9. Takrorlanuvchi algoritmgaga misol keltiring.
10. Takrorlanishlar soni noma'lum jarayon qanday bajariladi ?
11. Takrorlanishlar soni noma'lum takrorlanuvchi jarayonga misol keltiring.

Mustaqil bajarish uchun mashqlar

1. 1 dan n gacha bo'lgan natural sonlar kvadratlarining yig'indisini hisoblash algoritmini tuzing.
2. 1 dan n gacha bo'lgan natural sonlarning o'rta arifmetigi va o'rta geometrigini hisoblash algoritmini tuzing.
3. 1 dan n gacha bo'lgan juft sonlar yig'indisi va toq sonlari ko'paytmasini hisoblash algoritmini tuzing.
4. Barcha juft ikki xonali sonlarni hosil qilish algoritmini tuzing.
5. Barcha 4 ga karrali bo'lgan uch xonali sonlar yig'indisini hisoblash algoritmini tuzing.
6. $y = 5x^2 - 2x + 1$ funksiyaning $[-5; 5]$ oraliqda $h = 2$ qadam bilan qiymatlarini hisoblash algoritmini tuzing.
7. Agar arifmetik progressiyada $a_1 = -200$; $d = 0,2$ bo'lsa uning birinchi N hadi yig'indisini hisoblash algoritmini tuzing.
8. N sonining barcha tub bo'luvchilarini hosil qilish algoritmini tuzing.
9. $[a, b]$ oraliqdagi barcha tub sonlarini hosil qilish algoritmini tuzing.
10. Berilgan N sonining raqamlari sonini hisoblash algoritmini tuzing.
11. Berilgan N sonining raqamlari yig'indisi va ko'paytmasini hisoblash algoritmini tuzing.
12. Berilgan N sonini tub ko'paytuvchilarga ajratish algoritmini tuzing.
13. Barcha toq ikki xonali sonlarni hosil qilish algoritmini tuzing.
14. Barcha 3 va 7 ga karrali bo'lgan uch xonali sonlar yig'indisini hisoblash algoritmini tuzing.
15. $S = \sum_{n=1}^{10} 2^n + \prod_{k=1}^5 3^k + \sum_{m=1}^{10} 4^m$ ni hisoblash algoritmini tuzing.

2.4. Murakkab hisoblash jarayonlarini algoritmash

Bundan tashqari murakkab ko'rinishdagi takrorlanuvchi jarayonlar uchun algoritmlar tuzishda ichma-ich joylashgan takrorlanuvchi –siklik

algoritmalaridan foydalaniladi. Takrorlanuvchi jarayon sikl parametri va sikl tanasidan iborat bo'ladi.

Agar sikl tanasi ham yana sikldan iborat bo'lsa, u holda bunday sikllar ***ichma-ich joylashgan yoki murakkab sikllar*** deb ataladi. Ichida boshqa sikl joylashgan sikl ***tashqi sikl*** deyiladi. Boshqa sikl ichida joylashgan sikl esa ***ichki sikl*** deyiladi. Tashqi va ichki sikllarni tashkil etish qoidasi oddiy sikllarnikiga o'xshab ketadi. Faqat ichma-ich sikllarni tuzishda qo'yidagi qo'shimcha shartga rioya qilish kerak: ***ichki sikl tanasining amallari to'liq tashqi sikl tanasi ichida joylashgan bo'lishi kerak.***

Murakkab sikllar shartli ravishda ichma-ichlik darajasiga bo'linadi, ya'ni eng tashqi sikl 1- *darajali*, keyingi sikl 2- *darajali* va hokozalar kabi bo'ladi. Har xil darajali sikllarning parametrl qiymatlari bir vaqtda o'zgarmaydi.

Murakkab siklda avval hamma sikl parametrlari o'zlarining boshlang'ich qiymatiga ega bo'ladi. So'ngra eng ichki sikl to'liq bajariladi. Shundan keyin bunga nisbatan tashqi sikl parametri o'z qiymatini bittaga o'zgartiradi, so'ngra ichki sikl esa yana to'liq bajariladi va hokazo bu jarayon hamma darajadagi sikllar parametrlari talab etilgan barcha qiymatlarini qabul qilib bo'lguncha davom etadi. Ichma-ich joylashgan sikllar tanasining takrorlanishlarining umumiy soni, barcha sikl parametrlarining qabul qiladigan oxirgi qiymatlarining ko'paytmasiga teng: $N=N_1 \cdot N_2 \cdot \dots \cdot N_k$.

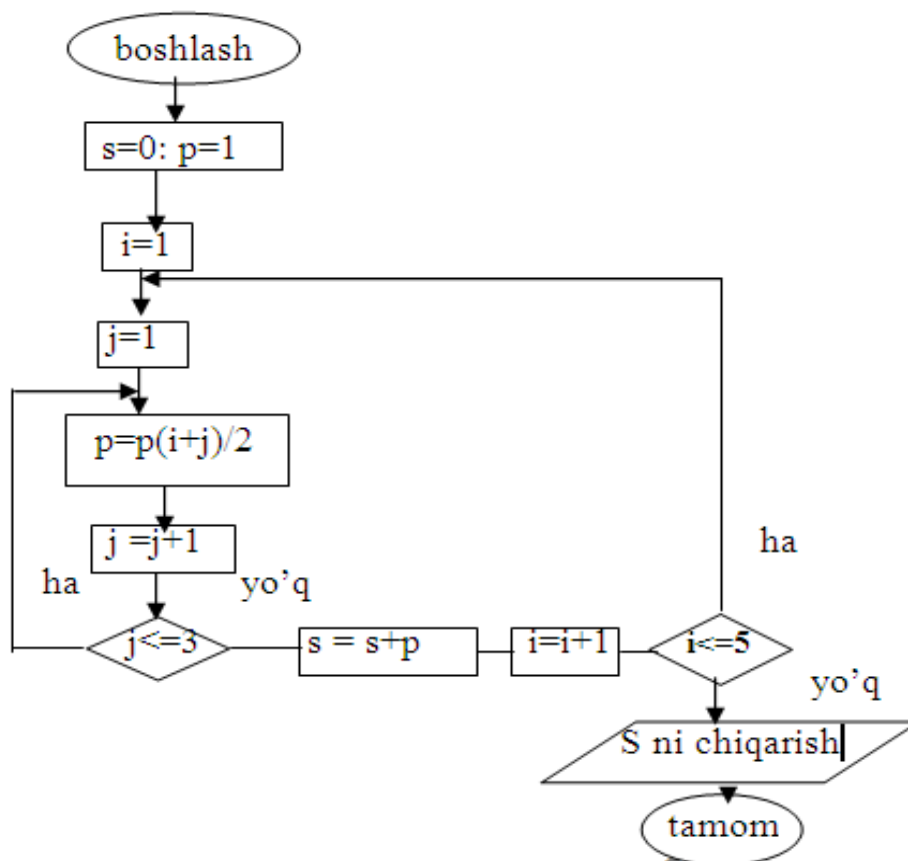
Ichma-ich joylashgan siklik jarayonlarni bajarilish tartibini quyidagi misol orqali berish mumkin:

$$S = \sum_{i=1}^2 \prod_{j=1}^3 (i+j)^2 = (1+1)^2 \cdot (1+2)^2 \cdot (1+3)^2 + (2+1)^2 \cdot (2+2)^2 \cdot (2+3)^2 = \\ = 4 \cdot 9 \cdot 16 + 9 \cdot 16 \cdot 25 = 576 + 3600 = 4176$$

Ichma-ich joylashgan siklik jarayonlar algoritmalar tuzishga doir misollar qaraymiz .

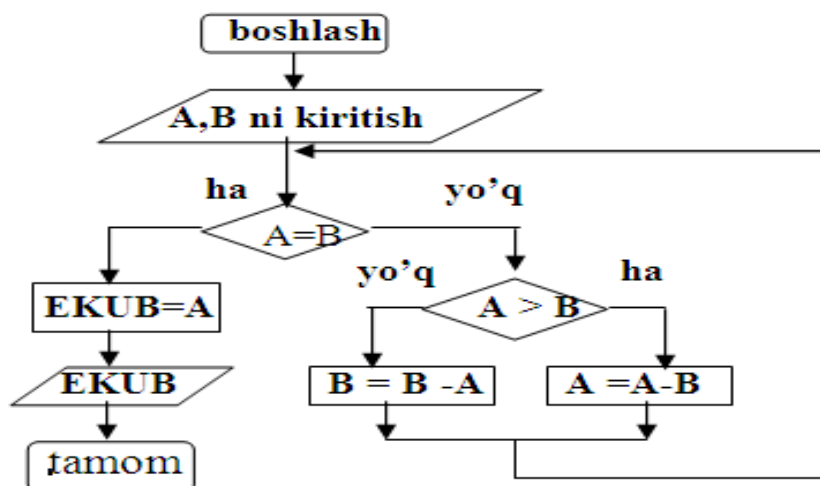
1-misol. $s = \sum_{i=1}^5 \prod_{j=1}^3 \frac{i+j}{2}$ ifodaning qiymatini hisoblash algoritmini

tuzing. *Yechish.* Bu yerda i bo'yicha takrorlanuvchi algoritmi tashqi, j bo'yicha takrorlanuvchi algoritmi ichki sikllar deb yuritiladi.



2-misol. Ikki a va b natural sonlari uchun eng katta umumiy bo'luvchi topilsin (blok-sxema ko'rinishi):

Yechish. Bu masalaning yechimi Evklid algoritmidan iborat bo'ladi. (Evklid algoritmining tavsifi yuqorida berilgan).



Nazorat uchun savollar

1. Sikl nima? Qanday sikllarni bilasiz?
2. Murakkab sikl nima va u qanday tuzilgan?
3. Ichma-ich va iteratsion sikllar qanday bajariladi?
4. Tashqi siklni tushuntiring.

5. Ichki sikl qanday bajariladi?
6. Iteratsion sikllarga misollar keltiring.
7. Ichma-ich sikllar tanasi nima?
8. Ichma-ich sikllar nimadalardan tarkib tiogan bo'ladi?
9. Ichma-ich sikllarning takrorlanishlar soni nimaga bog'liq?
10. Ichma-ich sikllarning soni nimaga bog'liq?

Mustaqil mashqlar

1. $S = \sum_{n=1}^{10} \prod_{k=1}^5 (a^k + b^n)$ ni hisoblash algoritmini tuzing.
2. Ko'paytirish jadvali elementlarini hosil qilish algoritmini tuzing.
3. $S = \sum_{i=1}^3 \frac{(\sum_{j=1}^{10} j + \prod_{k=1}^4 \frac{1}{k})}{i!}$ ni hisoblash algoritmini tuzing.
4. $S = \prod_{k=1}^4 \sum_{j=1}^{10} \frac{(k! + j!)}{a + b}$ ni hisoblash algoritmini tuzing.
5. $S = \sum_{k=1}^3 (\prod_{n=1}^4 kn^2 + \sum_{i=1}^5 ki^3)$ ni hisoblash algoritmini tuzing.
6. $S = \sum_{k=1}^3 (\prod_{n=1}^4 kn^2 + \sum_{i=1}^5 ki^3)$ ni hisoblash algoritmini tuzing.
7. $S = \prod_{k=1}^4 \sum_{j=1}^{10} (k! + j!)$ ni hisoblash algoritmini tuzing.
8. $S = \sum_{n=1}^3 \prod_{k=1}^5 \sum_{j=1}^{10} (k^2 + j^2 + n)$ ni hisoblash algoritmini tuzing.
9. $S = \prod_{k=1}^5 \sum_{j=1}^{10} (k^2 + j^4)$ ni hisoblash algoritmini tuzing.
10. $S = \sum_{n=1}^5 \prod_{i=1}^8 (n^2 + i^3)$ ni hisoblash algoritmini tuzing.

II-BOB. C++ TILIDA DASTURLASH ASOSLARI

1§. C++ tilining asosiy elementlari

1.1. C++ dasturlash tili to'g'risida qichqacha ma'lumotlar.

“Dasturlash asoslari” fani ham bugungi kunda eng ommabop bo'lgan C++ dasturlash tilini o'rgatishni o'z ichiga oladi. Ma'lumki C tili Denni Richchi tomonidan, C++ tili esa B.Straustrup tomonidan yaratilgan.

Ko'p dasturlash tillaridan farqli o'laroq to 1989 yilgacha C tili standarti mavjud emas edi. Bu davrda ishchi qo'llanma sifatida 1978 yilda bosilib chiqqan B.Kernigan va D.Ritchi kitobidan foydalanilar edi. Odatda bu kitobga ilovalar K&R maxsus qisqartmasi bilan belgilanadi. Bu kitobning ikkinchi bosmasi ANSI (American National Standards Institute) ishlab chiqqan va ANSI C deb ataluvchi til standartiga moslashtirilgandir.

Bundan tashqari ISOS (International Standard Organization S) standarti ham mavjud. Bu standartlar orasida farq ancha kam. Dastur asosan ANSIS standartiga asoslangan.

Bern Straustrup (Bjarne Stroustrup), AT&T Bell Laboratories xodimi, 1980 yili C++ tili ustida ish boshladi. Hozirgi C tiliga yaqinligini bildiruvchi C++ nomi rasmiy ravishda 1983-yili berildi.

Til birinchi tijorat versiyasi 1985-oktabrida taqdim etilib, xuddi shu yili B.Straustrupning «C++ dasturlash tili» (The C++ Programming Language) kitobi bosilib chiqdi.

Til standarti ustida 1990-yili ANSI (American National Standards Institute) ish boshladi. Bu standart oxirgi varianti 1997-yil noyabrida e'lon qilindi. Hozirgi davrda C++ tili eng ommaviy dasturlash tiliga aylandi.

1.2. C++ dasturlash muhiti va unda ishlash asoslari

Dasturlash muhiti deganda dasturchilar dastur tuzish va yozish uchun mo'ljallangan dastur tushiniladi. Dasturlash muhiti alohida bitta dasturlash tili yoki bir nechta dasturlash tillari uchun mo'ljallangan bo'lib, bitta guruhga kiruvchi dasturlash tillarini qayta ishlash uchun yaratiladi.

Dasturlash muhitlari dasturlarni qayta ishlash uchun bir nechta qismlarni o'z ichiga oladi:

- *muharir (редактор)*- dastur matnini yozish uchun ishlatiladi;
- *kompilyator (компилятор)* – dasturlash tilida yozilgan dasturni mashina tiliga tarjima qilish uchun mo'ljallangan;

– *dasturni rostlash, ishga tushirish (отладка)* – *dasturda mavjud xatoliklarni aniqlash va uni qayta ishlash uchun ishlatiladi.*

Dasturlash muhitida dasturlarni yozish va qayta ishlash bir nechta boshqichda amalga oshiriladi:

1. Dasturni muharrirlash.
2. Dastlabki qayda ishlash.
3. Tuzatish (компоновка).
4. Yuklash.
5. Bajarish.

C++ tilida tuzilgan dasturni kompilyasiya qilish, ishga tushirish va natijani olish uchun bir nechta zamonaviy kompilyatorlardan foydalaniladi. Bular Turbo C, Borland C++, Code Blocks, Dev C++ va hakoza.

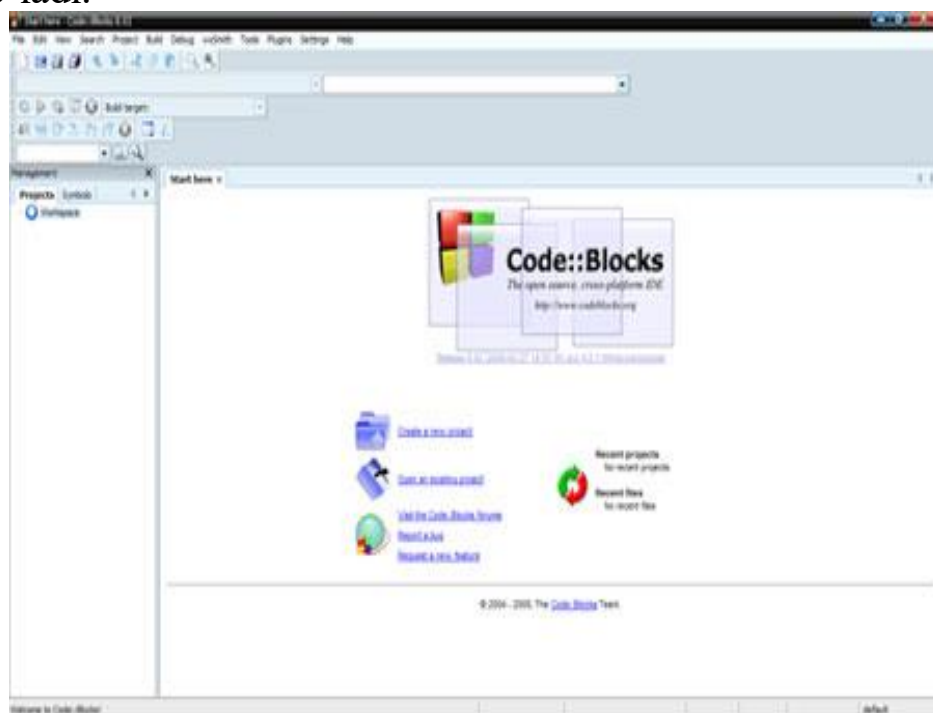
C++ dasturlash asoslari o'rganishni boshlaganlar uchun Code Blocks muhitida foydalanish yengilroqdir.

Code Blocks muhitini kompyuter xotirasiga yuklash ham qiyinchilik hosil qilmaydi. C++ tilida tuzilgan dasturni yozish uchun Code Blocks muhitini ishga tushiramiz:

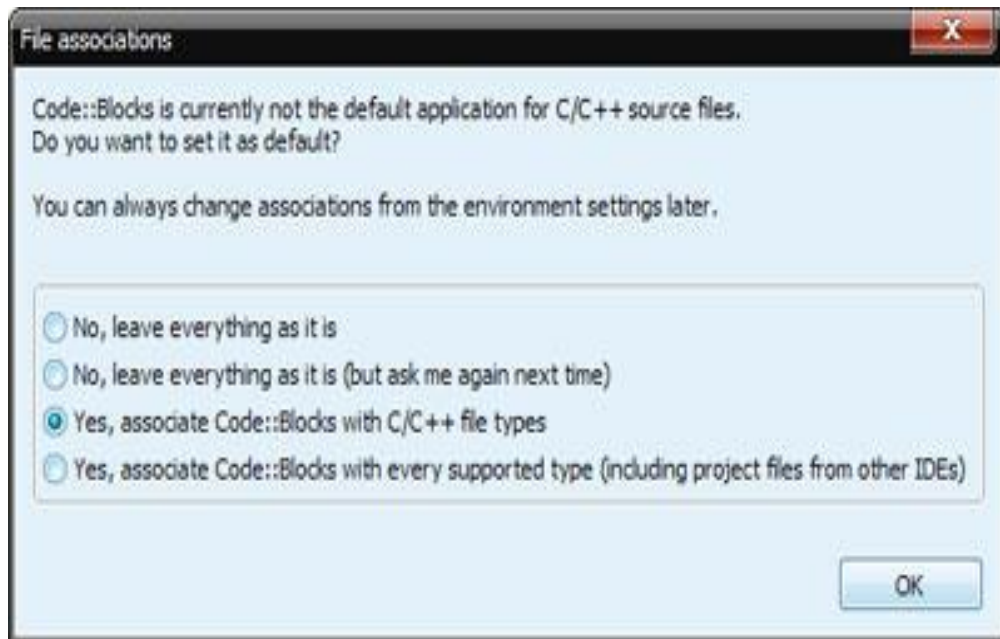
1. **Пуск ->Все программы ->CodeBlocks ->CodeBlocks .**

Ishchi stolda joylashgan mos yorliqqa  sichqoncha ko'rsatkichini ketma-ket ikki marta bosish orqali.

Bu buyruqlar bajarilganida ekranda **CodeBlocks** ning bosh oynasi paydo bo'ladi.



Dastur muhitini birinchi bor ishga tushirishda quyidagi xabar paydo bo'ladi. Bu yerda uchunchi variant tanlanib **Ok** tugmachasi bosiladi.

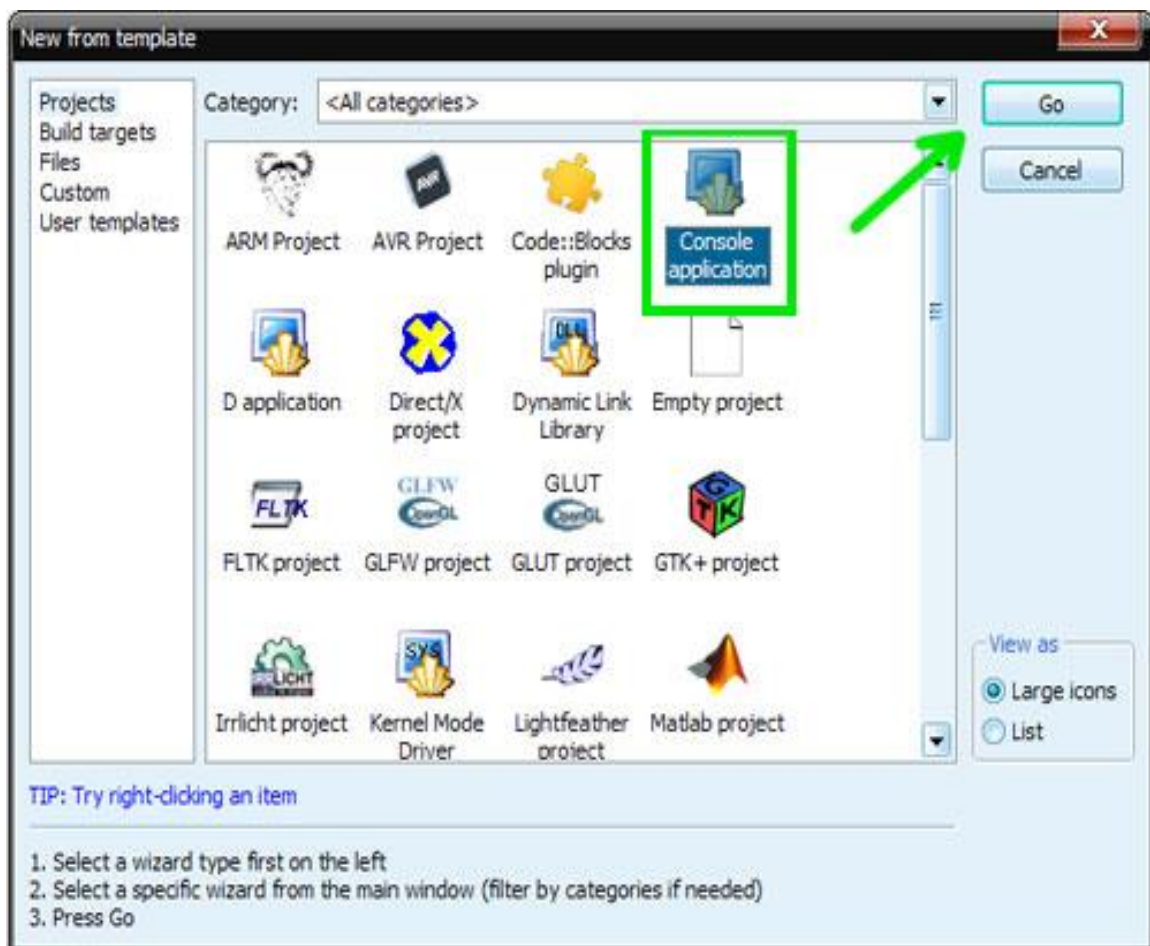


Jimlik bo'yicha bu xabardan tashqari quyidagi oyna ham payda bo'lishi mumkin:

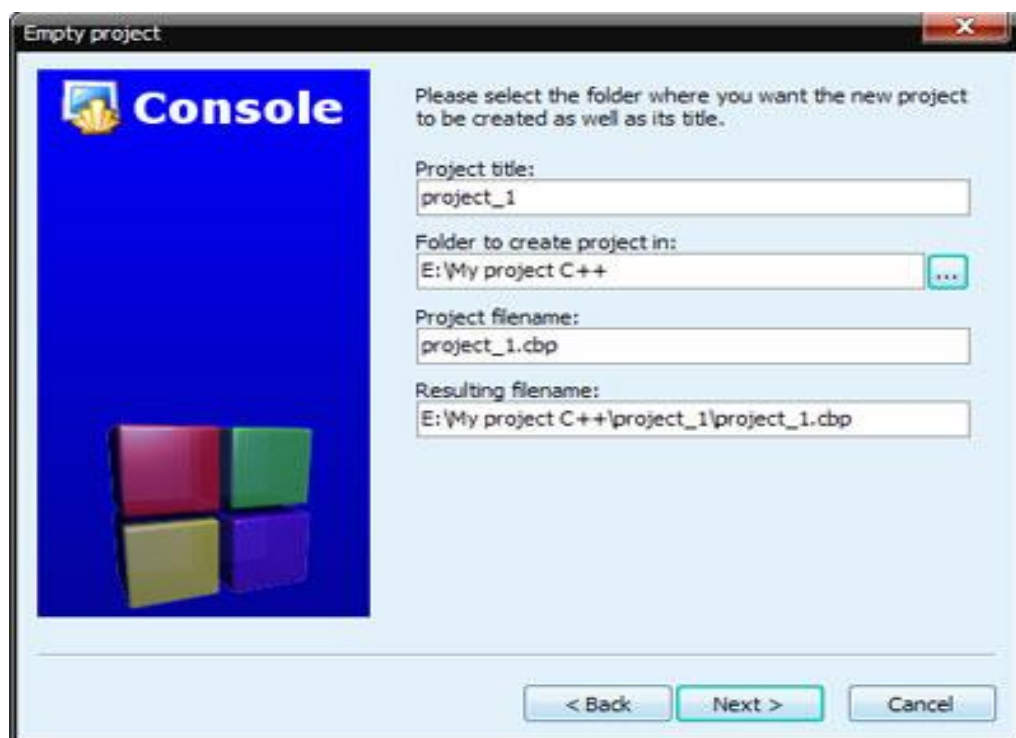


Tuzilgan dasturni kiritish uchun yangi loyiha, ya'ni dastur hujjatini yaratish kerak bo'ladi. Buni 2 xil usulda amalga oshirish mumkin.

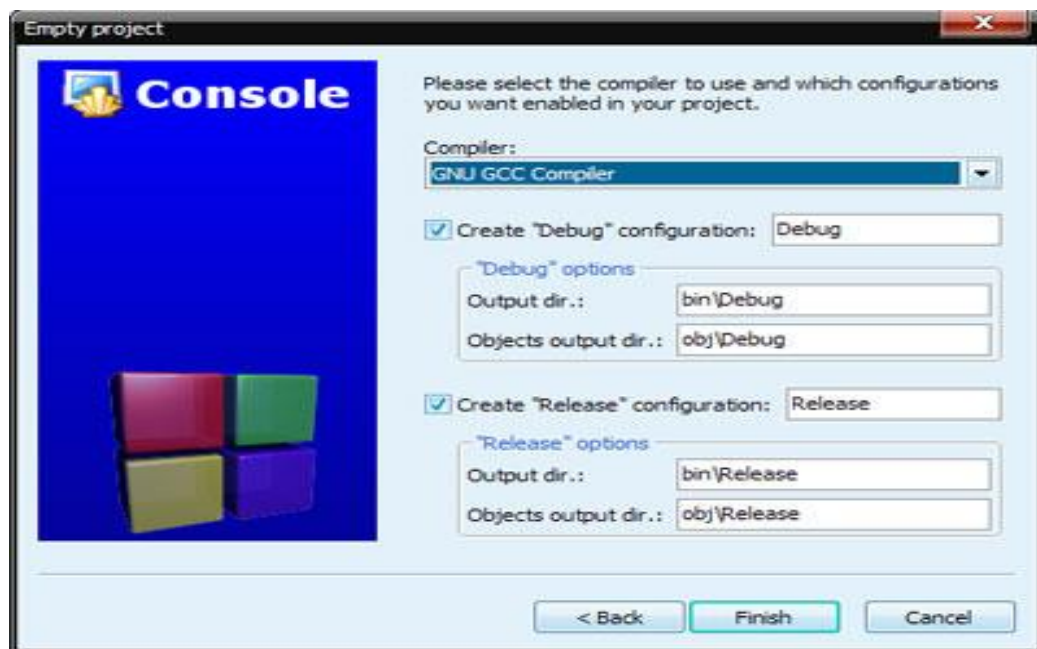
1. Hosil bo'lgan quyidagi oynadan **"Create a new project"** (Yangi loyiha yaratish) tanlanadi.



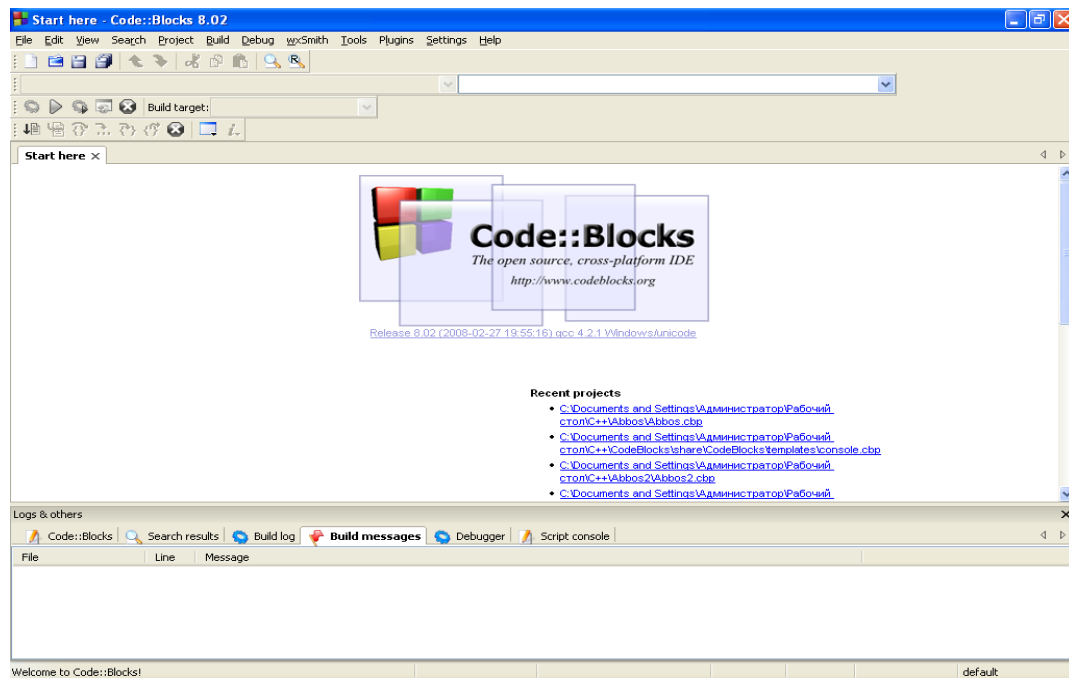
Quyida paydo bo'lgan keying oynada **"Next"** tugmachasini bosamiz.



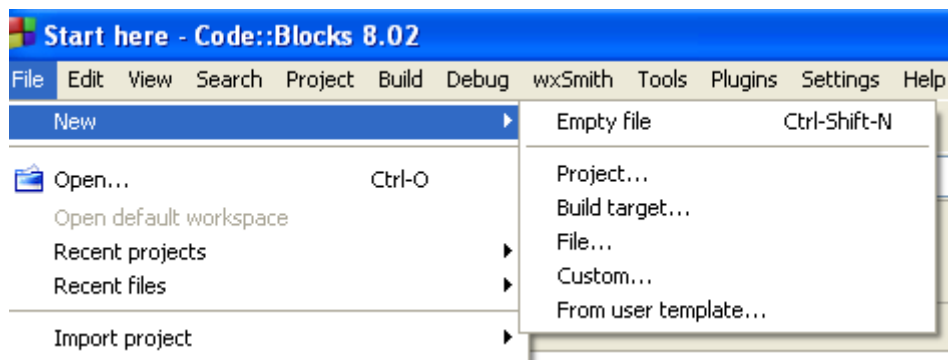
Keyingi hosil bo'lgan oynada **Finish** tugmasini bosamiz:



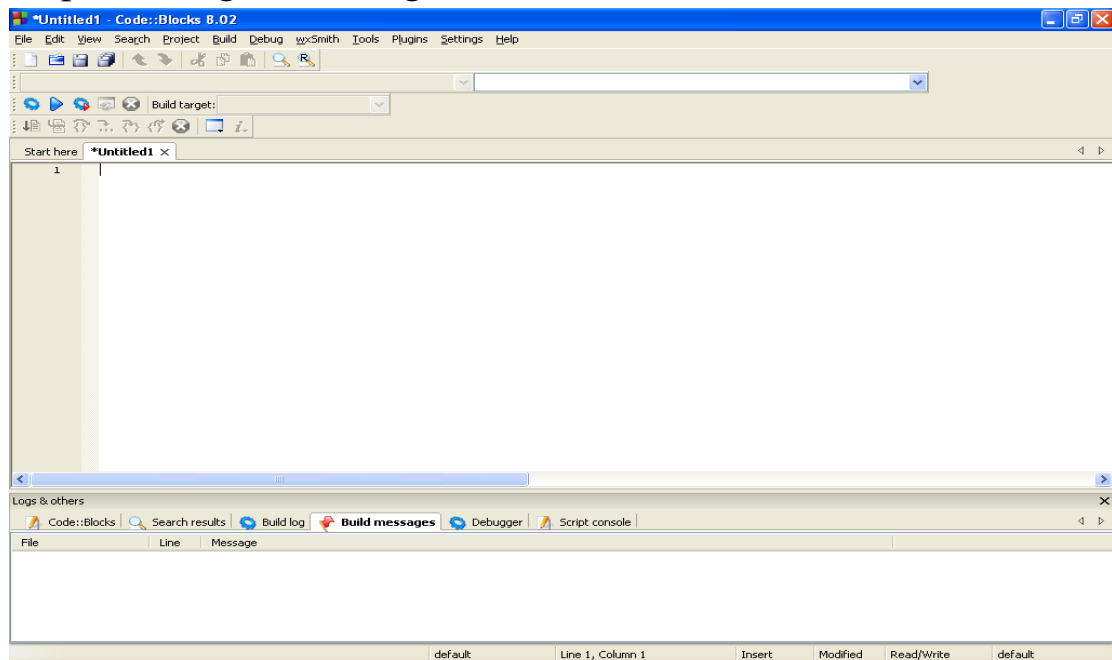
- Yangi loyiha yaratildi.
2. Quyidagi oyna paydo bo'lgandan keyin yangi loyiha yaratishni boshqacha ham amalga oshirish mumkin:



Bu oynada quyidagi buyruq amalga oshiriladi:



Bu buyruqdan so'ng muhitning ishchi sohasi hosil bo'ladi:



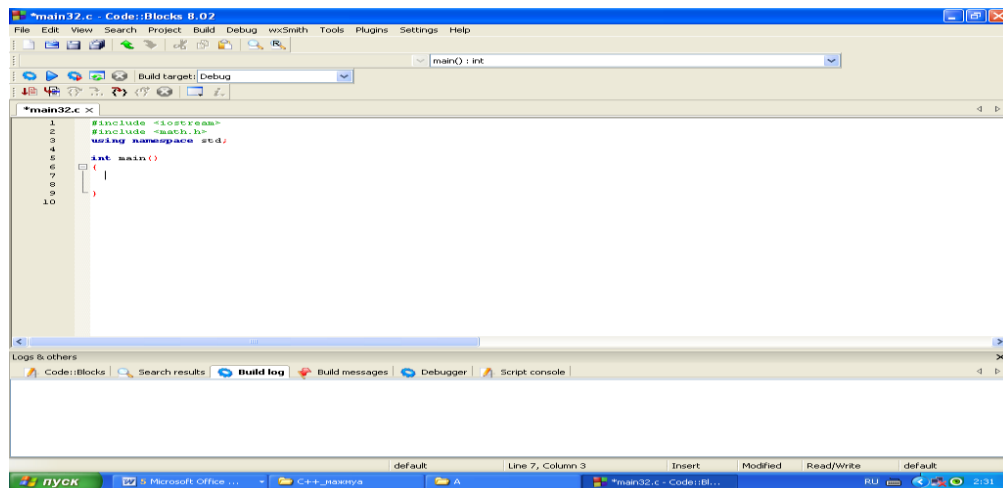
Endi bu oynaga bemaolol dasturni kiritish va uni kompyuter xotirasiga yangi ixtiyoriy nom bilan saqlash kerak bo'ladi.

1.2.1. Code Blocks muhitida ishlash asoslari

C++ muhiti ishga tushirilgandan keyin kompyuter ekranida uning oynasi paydo bo'ladi. Oyna sarlavxalar satri, bosh menyu va ishchi soxa (matn muxarriri) va buyruqlarsatridan iborat.

Sarlavhalar satri matn muharirida yozilgan dastur nomidan iborat bo'ladi.

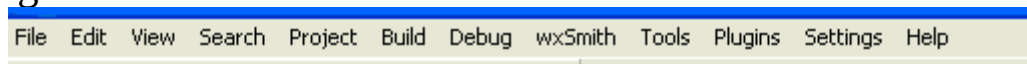
C++ tilida tuzilgan dastur biln ishlash uchun asosan bosh menyudan foydalaniladi. Bosh menyuga chiqish uchun **F10** tugmacha bosiladi



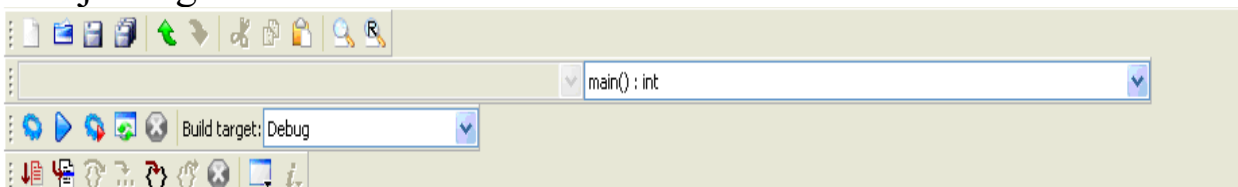
Sarlavhalar satri dastur nomi va oynani boshqarish tugmachalaridan iborat bo'ladi:



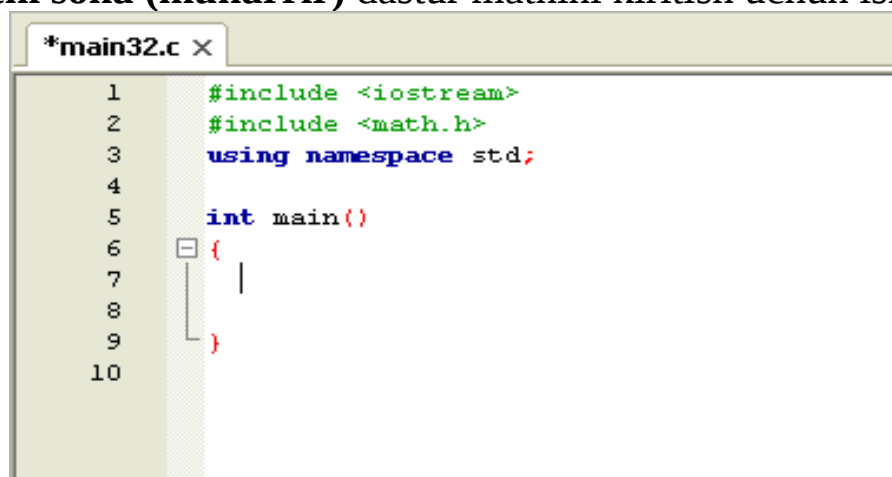
Menular satri 11 ta bo'limdan iborat bo'lib, ko'rinishi quyidagicha:



Uskunalar paneli ma'lum bir amallarni bajarish uchun mo'ljallangan vositalar iborat bo'ladi:



Ishchi soha (muharrir) dastur matnini kiritish uchun ishlatiladi:



1.2.3. C++ muhitining bosh "menyu" si.

Bosh "menyu" quyidagi asosiy bo'limlardan iborat:

1. File – fayli amallar ma’nosi bo’lib, fayllar bilan ishlash amallarini bajarishga mo’ljallangan. **File** bilan ishlash amallari:

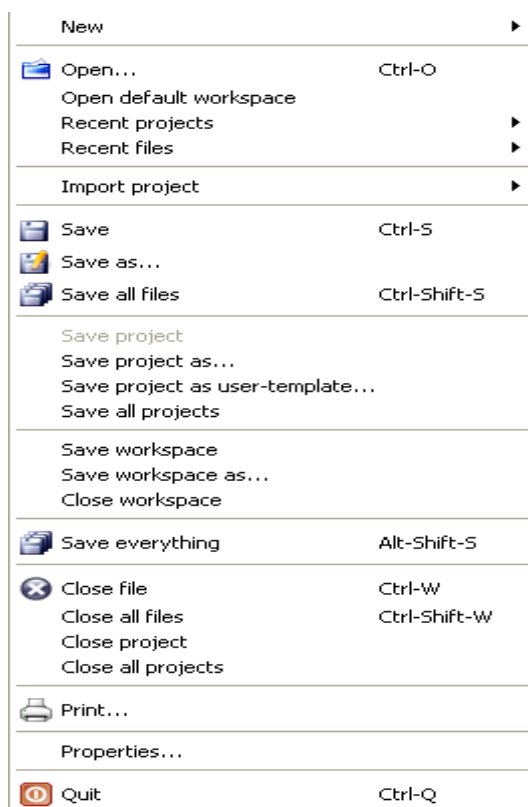
Open– mavjud faylni muharrirlash uchun yuklash. Bu bo’lim tanlangandan keyin ekranda muloqat oynasi hosil bo’ladi. Oynada yuklash lozim bo’lgan fayl nomi ko’rsatilib, **Enter** tugmachasi bosiladi. Fayl yuklangandan keyin u muharrirlash va uni bajarish uchun tayyor bo’ladi.

Save– faylni saqlash. Bu bo’lim ishga tushirilganda muharrir xotirasidagi fayl diskga yoziladi. Agar muharrirlanayotgan fayl standart **Noname.pas** nomiga ega bo’lsa, u holda saqlashda fayl nomini o’zgartirish imkoniyati mavjud bo’ladi.

Save as – muharrirlanayotgan faylni boshqa nom bilan saqlash. Ushbu buyruq amalga oshirilgandan keyin, muharrirlanayotgan faylga boshqa nom beriladi va u kompyuter xotirasiga yangi nom bilan saqlanadi.

New– yangi faylni yaratish. Bu buyruq berilgan integrallashgan muhit muharriri yangi fayl yaratish tartibiga o’tadi. Yaratilyotgan faylga **Noname.pas** nom beriladi. Faylni saqlayotgan unga ixtiyoriy nom berish mumkin bo’ladi.

Quit – dasturlash muhitida ishni tugallash. Bu buyruq berilgandan keyin integrallashgan muhitda ish tugallaniladi va undan chiqiladi.



2. **Edit** – fayllarni muharrirlashni yu qo’shimcha amallari “**menyu**”si bo’lib, matnni muharrirlashning asosiy funksiyalarini bajaradi. Bu amallar quyidagilardan iborat:

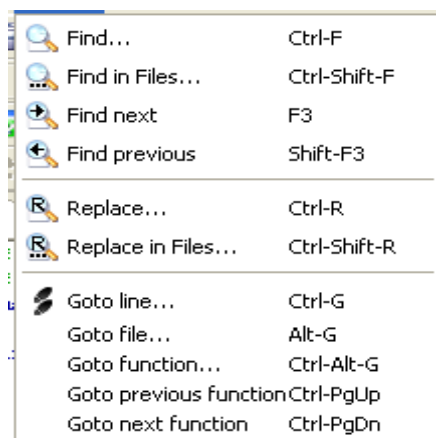
- *matn bo’laklarini nusxalash;*
- *matn bo’laklarini o’rnatish;*
- *matn bo’laklarini o’chirish;*
- *muxarrirlanayotgan faylning boshlang’ich topshiriqini tiklash ;*

	Undo	Ctrl-Z
	Redo	Ctrl-Shift-Z
	Cut	Ctrl-X
	Copy	Ctrl-C
	Paste	Ctrl-V
Swap header/source F11		
	Highlight mode	▶
	Bookmarks	▶
	Folding	▶
	End-of-line mode	▶
	File encoding	▶
	Special commands	▶
	Select all	Ctrl-A
	Comment	Shift-Ctrl-C
	Uncomment	Shift-Ctrl-X
	Toggle comment	
	Stream-Comment	
	Box-Comment	
	Auto-complete	Ctrl-J
	Goto matching brace	Ctrl-Shift-B
	Complete code	Ctrl-Space
	Show call tip	Ctrl-Shift-Space

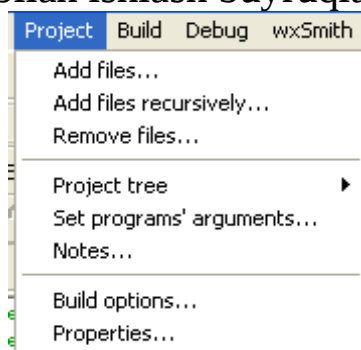
1. **View** –ko’rinishni belgilaydi

	Layouts	▶
	Toolbars	▶
	Manager	Shift-F2
	✓ Logs	F2
	✓ Script console	
	✓ Status bar	
	Symbols browser	
	Open files list	
	To-Do list	
	FullScreen	
	Focus editor	

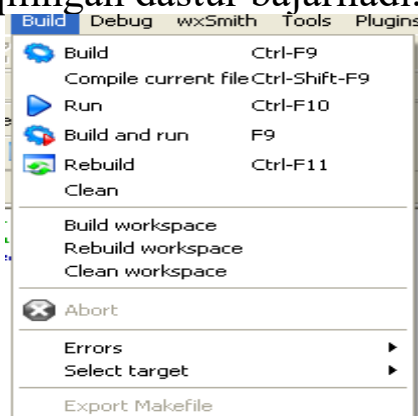
4. **Search** – matn bo’laklarini izlash va almashtirish “**menyu**”si bo’lib, matn qismlarini izlashni, lozim bo’lganda ularni yangisi bilan almash-tirishni amalga oshiradi



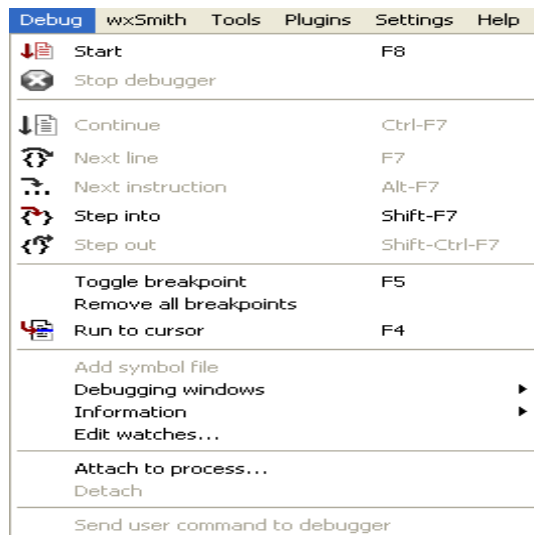
5. Project – loyihalarlar bilan ishlash buyruqlaridan iborat



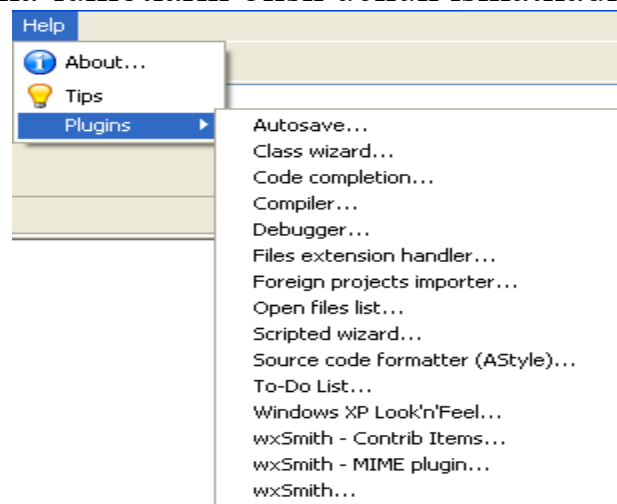
6. Build - kompilyasiya “menyu”si bo’lib, ishchi sohada joylashgan dasturni kompilyasiya qilish uchun ishlatiladi. Bu buyruq berilganda dastur kompilyasiya qilinadi. Agar kompilyasiya vaqtida dasturda xatolik mavjud bo’lsa, u holda bu haqda kerakli xabar kompyuter ekraniga chiqariladi. Xatoliklar tuzatilgandan keyin dastur yana kompilyasiya qilinadi. Shuning uchun dasturda mavjud xatoliklarni aniqlash maqsadida, uni vaqti-vaqti bilan kompilyasiya qilib turish kerak bo’ladi. Kompilyasiya qilingan dastur bajariladi.



7. Debug –dasturni roslash (otladka qilish) amallari “menyu”si bo’lib, dasturda mavjud xatoliklarni topishni yengillashtiruvchi buyruqlardan iborat



8. **Help** – yordam olish “menyu”si bo’lib, muhitda ishlash bo’yicha kerakli ma’lumotlarni olish uchun ishlatiladi.



Yuqorida ta’kidlaganimizdek, **bosh “menyu”ga** qarashli bo’lim – **ostki “menyu”** deb yuritiladi. Ostki “menyuga” buyrug’ini tanlash uchun, uning nomi ustiga kursor ko’rsatkichi joylashtirilib, **Enter** tugmachasi bosiladi. Ostki “menyu” dan chiqish **Esc** tugmachasini bosish orqali amalga oshiriladi.

1.3. C++ tili alfaviti va asosiy elementlari.

Alfavit. C++ alfavitiga quyidagi belgilar kiradi:

- katta va kichik lotin alfaviti harflari (A,B,...,Z,a,b,...,z)
 - raqamlar: 0,1,2,3,4,5,6,7,8,9
 - maxsus belgilar: “ , { } | [] () + - / % \ ; ‘ . : ? < = > _ ! & * # ~ ^
 - ko’rinmaydigan belgilar (“umumlashgan bo’shliq belgilari”).
- Leksemalarni o’zaro ajratish uchun ishlatiladigan belgilar (misol uchun bo’shliq, tabulyatsiya, yangi qatorga o’tish belgilari).

Izohlarda, satrlarda va belgili o’zgarmaslarda boshqa literalalar, masalan kiril alifbosi harflarini ishlatilishi mumkin.

Identifikator. Identifikatorlar lotin harflari, ostki chiziq belgisi va sonlar ketma- ketligidan iborat bo'ladi. Identifikator lotin harfidan yoki ostki chiziq belgisidan boshlanishi lozim. **Misol uchun:** *A1, _MAX, adress_01, RIM, rim*

Katta va kichik harflar farqlanadi, shuning uchun oxirgi ikki identifikator bir biridan farq qiladi.

Xizmatchi so'zlar. Tilda ishlatiluvchi ya'ni dasturchi tomonidan o'zgaruvchilar nomlari sifatida ishlatish mumkin bo'lmagan identifikatorlar xizmatchi so'zlar deyiladi.

C++ tilida quyidagi xizmatchi so'zlar mavjud:

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

1.4. C++ tilida miqdorlar.

O'zgaruvchilar va o'zgarmaslar. C++ tilida dasturlash jarayonida miqdorlar va ularning turlaridan foydalaniladi. Miqdorlar ikki xil bo'ladi: **o'zgaruvchilar va o'zgarmaslar.**

Dasturni bajarilish jarayonida o'z qiymatini o'zgartiradigan miqdorlar **o'zgaruvchilar** deyiladi. C++ tilining asosiy tushunchalaridan biri nomlangan xotira qismi – ob'ekt tushunchasidir. Ob'ektning xususiy holi bu o'zgaruvchidir. O'zgaruvchiga qiymat berilganda unga ajratilgan xotira qismiga shu qiymat kodi yoziladi. O'zgaruvchi qiymatiga nomi orqali murojaat qilish mumkin, xotira qismiga esa faqat adresi orqali murojaat qilinadi. O'zgaruvchi nomi bu erkin kiritiladigan identifikatordir. O'zgaruvchilardan foydalanish uchun ularga nom beriladi. O'zgaruvchi nomi lotin harflari, ostki chiziq belgisi va sonlar ketma ketligidan iborat bo'lishi mumkin.

Masalan: *YUZA, _yuza23, tenglama, Ildiz, _Alfa.*

O'zgaruvchi nomi sifatida xizmatchi so'zlarni ishlatish mumkin emas.

O'zgaruvchi nomida katta va kichik harflar farqlanadi, **masalan**, infor , INFOR . Bu ikki o'zgaruvchi bir biridan farq qiladi. O'zgaruvchi nomi har doim harfdan, belgidan yoki (_) chiziqdan boshlanishi lozim.

Dasturni bajarilish jarayonida o'z qiymatini o'zgartirmaydigan miqdorlar **o'zgarmas** deyiladi. C++ tilida **butun, haqiqiy va belgili** turdagi o'zgarmaslar ishlatiladi.

1. **Butun turdagi o'zgarmaslar** va butun sonlardan iborat bo'ladi. **Masalan:** 23, -987, 567 va hakoza.

2. **Haqiqiy turdagi o'zgarmaslar** haqiqiy sonlardan, ya'ni butun va kasr qismlardan iborat bo'ladi. Haqiqiy o'zgarmaslar ikki xil ko'rinishda ifodalanadi:

1) **Qo'zg'almas vergulli:** **Masalan:** 9.78, -98.67, -1.87.

2) **Qo'zg'aluvchan vergulli.** Juda katta va juda kichiksonlarni ifodalash uchun ishlatiladi. Bu turdagi sonlar darajali (eksponensial) ko'rinishda yoziladi. **Masalan:** a) $23,567 \cdot 10^4$ soni 23.567E4 ko'rinishda yoziladi.

b) $4,67 \cdot 10^{-12}$ soni 4.67E-12 ko'rinishda yoziladi.

c) 9000000 ya'ni $9 \cdot 10^6$ soni 9E6 yoki 90E5 yoki 0.9E7 va hakoza ko'rinishlarida yozilishi mumkin.

d) 0,000000023 ya'ni $23 \cdot 10^{-10}$ soni 23 E-10 yoki 0.23 E-7 yoki 2.3 E-9 va hakoza ko'rinishlarida yozilishi mumkin.

3. **Belgili o'zgarmaslar.** Belgili o'zgarmaslar qo'shtirnoq ichiga olingan bitta belgidan yoki bittadan ko'p bo'lgan belgi (satr)dan iborat bo'lishi mumkin. **Masalan,** "3", "a" , "g", "Salom", "tenglama", "Kompyuter".

1.5.O'zgaruvchilarning turlari

Dastur tuzish jarayonida o'zgaruvchilardan foydalanish uchun ular qabul qiladigan qiymat yoki ma'lumotlarning turlarini ko'rsatish ya'ni e'lon qilish kerak bo'ladi. C++ tilida o'zgaruvchilarning quyidagi turlaridan foydalaniladi:

1. **Butun sonli tur.**

tur	Qiymat chegarasi
int	32768 dan 32767 gacha
char	-128 dan 127 gacha
long	-2 147 483 648 dan 2 147 483 647 gacha

Butun sonlar ta'riflanganda ko'rilgan turlar oldiga unsigned (ishorasiz) ta'rif o'shilishi mumkin. Bu ta'rif qo'shilgan butun sonlar ustida amallar

mod 2 n arifmetikasiga asoslangandir . Bu yerdan soni int turi xotirada egallovchi razryadlar sonidir. Agar ishorasiz k soni uzunligi int soni razryadlar sonidan uzun bo'lsa, bu son qiymati k mod 2 n ga teng bo'ladi. Ishorasiz k son uchun $-k$ amali $2n - k$ formula asosida hisoblanadi. Ishorali ya'ni signed turidagi sonlarning eng katta razryadi son ishorasini ko'rsatish uchun ishlatilsa unsigned (ishorasiz) turdagi sonlarda bu razryad sonni tasvirlash uchun ishlatiladi.

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

Int a, b1, ac; eki

Int a;

int b1;

int ac;

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

Int I=0;

Char c='k';

Typedef ta'riflovchisi yangi turlarni kiritishga imkon beradi.

Misol uchun yangi COD turini kiritish:

Typedef unsigned char COD;

COD simbol;

2. Haqiqiy tur. Haqiqiy turli o'zgaruvchilarni e'lon qilish uchun **float** kalit so'zidan foydalaniladi. Agar o'zgaruvchilarning qiymati juda katta va juda kichik, ya'ni $5.18 \cdot 10^{18}$ yoki $9,15 \cdot 10^{-12}$ ko'rinishlarda bo'lsa **double** yoki **long double** kalit so'zlaridan foydalanib e'lon qilinadi. $\alpha \cdot 10^{\beta}$ ko'rinishdagi sonlarda α -sonning tartibi deyiladi.

tur	Qiymat chegarasi
float	$3.4 \cdot E-38 - 3.4 \cdot E38$
double	$1.7 \cdot E-308 - 1.7 \cdot E308$
long double	Juda ko'p

3. Mantiqiy tur. Mantiqiy turlar bilan ishlash jarayoni matematik mantiq fanning qonuniyatlariga asoslanayapti. Matematik mantiq fikrlashning shakli va qonuniyatlarini o'rganuvchi fandır. Matematik mantiq o'zi mulohazalarga asoslanadi. Mulohaza esa ixtiyoriy fikr bo'lib, uning qiymati **rost (true)** yoki **yolg'on (false)** ekanligi ta'minlanishi shart.

1.6. C++ tilida amallar

Arifmetik amallar. C++ dasturlash tilida ham 4 ta arifmetik amal (+ - qo'shish; - - ayirish; * - ko'paytirish; / - bo'lish, % - modul olish ya'ni qolqikli bo'lish) laridan foydalaniladi. Ularning bosqichlarga bo'linishi, bajarilish tartibi, qavslar ishtirokida bajarilish tartibi matematikadagi tartibidan farq qilmaydi. Amallar odatda unar ya'ni bitta operandga qo'llaniladigan amallarga va binar ya'ni ikki operandga qo'llaniladigan amallarga ajratiladi.

<i>C++ tilida arifmetik amallar</i>			
<i>amallar</i>	<i>belgilash</i>	<i>amallar</i>	<i>belgilash</i>
<i>ko'paytirish</i>	$a*b$	<i>ayirish</i>	$a-b$
<i>bo'lish</i>	a/b	<i>bo'linmaqoldig'i</i>	$a\%b$
<i>qo'shish</i>	$a+b$	<i>Darajaga ko'tarish</i> x^n	$\text{pow}(x,n)$

Binar amallar additive ya'ni + qo'shuv va - ayirish amallariga , hamda multiplikativ ya'ni * ko'paytirish, / bo'lish va % modul olish amallariga ajratiladi.

Additiv amallarining ustivorligi multiplikativ amallarining ustivorligidan pastroqdir. Amallar quyidagicha bajariladi:

- 1) qavs ichidagi amallar, keyin...
- 2) funksiyalarga murojat, keyin, ...
- 3) chapdan o'ngga qarab ko'paytirish, bo'lish va bo'lish qoldig'i, keyin,

...

- 4) chapdan o'ngga qarab qo'shish va ayirish.

Masalan: 2 1 5 4 3 8 6 7

$$x = (a + 5 * b) * \text{fabs}(c + d) - (3 * b - c);$$

Butun sonni butun songa bo'lganda natija butun songacha yaxlitlanadi. Misol uchun $18/5 = 3$; $(-18)/5 = -3$; $18/(-5) = -3$.

Butun sonni butun songa bo'lishda qoldiq qismi tashlab yuboriladi, ya'ni $5/2$ ning natijasi 2 ga teng bo'ladi. Shuning uchun bo'luvchi yoki bo'linuvchilardan bittasi haqiqiy son sifatida ifodalash kerak bo'ladi. Natija ham haqiqiy sondan iborat bo'ladi. $5/2$ yoki $5./2$ ning natijasi 2,5 ga teng bo'ladi. **Masalan:**

inti, n;

floatx;

i = 7;

$x = i / 4$; // $x=1$, butun son butun songa bo'linadi

$x = i / 4.$; // $x=1.75$, butun son kasr songa bo'linadi;

$x = (\text{float}) i / 4$; // $x=1.75$, kasr son butun songa bo'linadi;

$n = 7. / 4.$; // $n=1$, natija butun o'zgaruvchiga beriladi

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

Masalan:

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

Darajaga ko'tarish amali **pow (x,n)** funksiyasi yordamida bajariladi: $\text{pow}(x,n)=x^n$. **Masalan:** $\text{pow}(3,4)=81$; $\text{pow}(7, 3)= 343$;

Pow10 (intp) funksiyasi 10^p ni hisoblaydi. Bunda p—butun son bo'lishi shart. **Masalan:** $\text{Pow10}(2)=100$, $\text{Pow10}(5)=10000$

Inkrement va dekrement amallar. Dasturlarda o'zgaruvchiga 1 ni qo'shish va ayirish amallari juda ko'p hollarda uchraydi. C++ tilida qiymatni 1 ga oshirish inkrement, 1 ga kamaytirish esa dekrement deyiladi. Bu amallar uchun maxsus operatorlar mavjuddir.

Inkrement operatori (++) o'zgaruvchi qiymatini 1 ga oshiradi, dekrement operatori (-) esa o'zgaruvchi qiymatini 1 ga kamaytiradi. Masalan, s o'zgaruvchisiga 1 qiymatni qo'shmoqchi bo'lsak quyidagi ifodani yozishimiz lozim: $s++$ //s o'zgaruvchi qiymatini 1 ga oshirdik.

Bu ifodani quyidagicha yozishimiz mumkin edi.

$$s=s+1;$$

Bu ifoda o'z navbatida quyidagi ifodaga teng kuchli:

$$s+=1;$$

Bundan tashqari qiymat berish buyrug'ini qisqartirilgan va to'liq ko'rinishlarda berish mumkin. **Masalan:**

Qisqartirilgan yozuv	To'liq yozuv
$x += a;$	$x = x + a;$
$x -= a;$	$x = x - a;$
$x *= a;$	$x = x * a;$
$x /= a;$	$x = x / a;$
$x \% = a;$	$x = x \% a;$
$y -= a$	$y = y - a$
$y \% = a ++$	$y = y \% a; a = a + 1$
$y* = a + 3$	$y = y * (a + 3)$
$y* = -- a$	$y = y - a; a = a - 1$
$y + = -- y$	$y=y-1; y=y+y$
$i++$	$i=i+1$
$i--$	$i=i-1$

Prefiks va postfiks. Inkrement operatori ham, dekrement operatori ham ikki variantda ishlaydi: prefiksli va postfiksli. Prefiksli variantda ular o'zgaruvchidan oldin (++Age), postfiksli variantda esa o'zgaruvchidan keyin (Age++) yoziladi.

Oddiy ifodalarda bu variantlarni qo'llanishida farq katta emas, lekin bir o'zgaruvchiga boshqa o'zgaruvchining qiymatini o'zlashtirishda ularning qo'llanilishi boshqacha xarakterga ega. Prefeksli operator qiymat o'zlashtirilguncha, postfiksli operator esa qiymat o'zlashtirilgandan keyin bajariladi. 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.

Munosabat amallari. Munosabat amallari qiymatlari 1 ga teng agar munosabat (true) bajarilsa va aksincha (false) bo'lsa 0 ga tengdir. Munosabat amallari arifmetik turdagi operandlarga yoki ko'rsatkichlarga qo'llaniladi.

Nomi	Operator	Misol	Hosil bo'ladigan qiymat
Tenglik	=	100==50 50==50	0 (false) 1 (true)
Teng emas	!=	100!=50 50!=50	1 (true) 0 (false)
Katta	>	100>50 50>50	1 (true) 0 (false)
Katta yoki teng	>=	100>=50 50>=50	1 (true) 1 (true)
Kichik	<	100<50 50<50	1 (true) 0 (false)
Kichik yoki teng	<=	100<=50 50<=50	0 (false) 1 (true)

Misollar:

1!=0 qiymati 1 ga teng;
 1==0 qiymati 0 ga teng;
 3>=3 qiymati 1 ga teng;
 3>3 qiymati 0 ga teng;
 2<=2 qiymati 1 ga teng;
 2<2 qiymati 0 ga teng;

Katta $>$, kichik $<$, katta yoki teng $>=$, kichik yoki teng $<=$ amallarining ustivorligi bir xildir. Teng $==$ va teng emas $!=$ amallarining ustivorligi o'zaro teng va qolgan amallardan pastdir.

Mantiqiy amallar. Mantiqiy ma'lumotlar uchun quyidagi mantiqiy amallardan foydalanish mumkin: $\|$ - qo'shish (**va**); $\&\&$ - ko'paytirish (**yoki**); $!$ - inkor (**yo'q**). **Or** va **and** mantiqiy amallar ikkita kattalik bilan, **not** amali esa bitta kattalik bilan bajariladi.

amal	belgilash	misol
qo'shish (or);	$\ $	$a\ a$
ko'paytirish (and);	$\&\&$	$a \&\& b$
inkor (not).	$!$	$!a$

Mantiqiy qiymatlar ustida amallar natijasi quyidagicha:

a	b	$!$	$\ $	$\&\&$
<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>
<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>
<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>false</i>
<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>

Misol. Agar $a = 25$ va $b = 20$ bo'lsa:

- a) $(a > 10) \| (b < 12)$ (*true*) b) $(a > 10) \&\& (b < 3)$ (*false*)
c) $(a > 10) \&\& (b > 12)$ (*true*) d) $(a > 10) \| (b > 12)$ (*true*)
e) $!((a > 10) \&\& (b > 12))$ (*false*)

Nazorat uchun savollar

1. C++ tilining tarkibiy asosiy qismlari qanday?
2. Ma'lumot nima va uning qanday turlari bor?
3. Ma'lumotlar qanday tavsiflanadi?
4. Butun, haqiqiy, satrli va mantiqiy turlarni tushintiring.
5. C++ tilida qanday amallar ishlatiladi?
6. Qanday arifmetik amallarni bilasiz?
7. Munosabat amallarni bajarilish tartibini tushuntiring?
8. C++ tilida o'zgarmas nima? Qanday o'zgarmaslarni bilasiz?
9. Qanday mantiqiy amallarni bilasiz?
10. Taqqoslash amallari nima uchun ishlatiladi?
11. Sanab o'tiladigan turlarni tushuntiring?
12. Nima uchun literalli o'zgarmasga nisbatan belgili o'zgarmasni ishlatish yaxshiroq?
13. Butun sonli va haqiqiy tiplarni qanday farqi bor?
14. `unsigned short int` va `long int` tiplarining o'zaro farqi nimada?

15. const kalitli so'zini **#define** direktivasi o'rniga qo'llashni afzalligi nimada?

Mustaqil topshiriqlar

1. Quyidagilar qaysi biri butun turga tegishli:

a) -3 b) 12a +5.2 c) -9999999999999999 d) ++60 e) ++-+-+3957

2. Quyidagilar qaysi biri haqiqiy turga tegishli:

a) 0.3 b) -2.56 c) 3,5 d) 6.5487E3 e) +.3 f) -7. j) 0.0E6

3. Quyidagilar qaysi biri belgili turga tegishli:

a) 'abc' b) 'ы' c) '3,5' d) "начало" e) '///// ' f) 'Алексj) '0"0's) ""ко-нец""

4. Quyidagilar qaysi biri mantiqiy turga tegishli:

a) true b) False c) t d) 'false' e) истина f) trunc j) TRUE

5. Quyidagilardan qaysi biri identifikator bo'lishi mumkin:

a) Alphab) beginc) a125 d) 2be) end3 f) Жyкj) end_for

6. Agar $a = 11$ va $b = 22$ bo'lsa natijani aniqlang:

a) $(a > 10) \&\& (b < 12)$ b) $(a > 10) \parallel (b < 3)$ c) $(a > 10) \&\& (b > 12)$

d) $(a > 10) \parallel (b > 12)$ e) $!(a > 10) \&\& (b > 12)$

7. Agar $x = 9$ bo'lsa natijani aniqlang:

a) $x/3$ b) $x/3.$ c) $(\text{float } x/3)$

8. Natijani aniqlang:

a) $12.37\% 4$ b) $-9 \% 8$ c) $66 \% (-12)$ d) $\text{pow}(4,6)$ e) $\text{pow}10(6)$

28. C++ tilida standart funksiya va ifodalar

2.1. C++ tilida standart funksiyalar.

C++ tilida **math.h** matematik funksiyalar kutubxonasiga tegishli quyidagi funksiyalar mavjud:

funksiya	ifodalanishi	funksiya	ifodalanishi
$\sin x$	$\sin(x)$	$\arctan x$	$\text{atan}(x)$
$\text{tg } x$	$\tan(x)$	$\arccos x$	$\text{acos}(x)$
e^x	$\exp(x)$	$\sqrt[3]{x^2}$	$\text{pow}(x, 2/3.)$
$\ln x$	$\log(x)$	$\log_2 x$	$\log(x)/\log(2)$
$\lg x$	$\log_{10}(x)$	x dan kichik bo'lmagan butun son	$\text{ceil}(x)$
x^a	$\text{pow}(x, a)$	x dan katta bo'lmagan butun son	$\text{floor}(x)$
$\sqrt{x}$	$\text{sqrt}(x);$ $\text{pow}(x, 1/2.)$		
$ x $	$\text{abs}(x)$ yoki $\text{fabs}(x)$		

Masalan : $\text{ceil}(4.8) = 5$, $\text{floor}(4.8) = 4$, $\text{pow}(2,5) = 32$, $\text{fmod}(5,3) = 2$.

Trigonometrik funksiyalarning qiymatlari radian o'lchov birligida hisoblanadi.

Masalan: $\sin(1.2) = 0.932039$, $\cos(1.2) = 0.362358$, $\tan(1.2) = 2.57215$.

Agar trigonometrik funksiyalarni qiymatlarini graduslarda hisoblash kerak bo'lsa, ular radianga aylantiriladi yoki π orqali ifodalanadi.

Masalan: $\sin 60^\circ = \sin(\pi/3) = \sin(\pi * 60/180) = 0.86576$.

2.2. C++ tilida ifodalar.

C++ tilida ifodalar o'zgarmaslardan, o'zgaruvchilardan, funksiyalardan, amallar ishorasidan va qavslardan tuzilgan bo'lib, ostki va uctki indeksiz bir satrga yoziladi.

C++ tilida dasturlash tilida arifmetik, mantiqiy va satrli ifodalar ishlatiladi.

Arifmetik ifoda. C++ tilida arifmetik ifodalar o'zgarmaslardan, o'zgaruvchilardan, funksiyalardan, amallar ishorasidan va qavslardan tuzilgan bo'ladi.

Ifodalarda qatnashayotgan o'zgarmaslar, o'zgaruvchilar va funksiyalar dasturda e'lon qilingan bo'lishi yoki standart nomiga ega bo'lishi kerak. Sonli qiymatlarni hisoblash uchun arifmetik ifoda qo'llaniladi.

Arifmetik ifodada bo'lish va ko'paytirish amallari qo'shish va ayirish amallaridan oldin bajariladi.

Masalan: $3 + 2 * 5 - 3$ ifodaning bajarilishidan 10 hosil bo'ladi.

Bir xil o'rinli amallar ishtirok etgan ifodalarda ular chapdan o'ngga qarab bajariladi.

Masalan. $4 * 6 / 3$ ifoda natijasi 8 ga teng.

Qavs ichiga olingan ifodalar birinchi navbatda bajariladi.

Masalan. $(7-2) * (5+4)$ ifoda natijasi 45 ga teng

Arifmetik ifodada haqiqiy va butun turdagi ma'lumotlar qatnashishi mumkin, natija ham mos ravishda haqiqiy yoki butun bo'ladi.

Oddiy yozuvda	C++ tilida
$\frac{a+12,686}{c_2-1,87 \cdot 10^3} + \gamma$	$(a+12.686)/(C2-1.87E3)+gamma$
$\frac{ 2\sin 4x + x }{3x}$	$fabs(2*\sin(4*x)+x)/(3*x)$
$\frac{2^2 + \ln 1,13}{tg 2x}$	$(2*2+\log(1.13))/\tan(2*x)$
$\frac{a^2 + 2^{1,17}}{\sqrt{x+y}}$	$(pow(a,2)+pow(2,1.17))/sqrt(x+y)$
$\frac{ x - y }{1+ xy }$	$(fabs(x) - fabs(y)) / (float)(1 + fabs(x * y))$
$\ln \left \left(y - \sqrt{ x } \right) \left(x - \frac{y}{z + x^2 / 4} \right) \right $	$\log(fabs((y - sqrt(fabs(x))) * (x - y / (z + pow(x,2) / 4.0))))$

Mantiqiy ifoda. Mantiqiy ifodalar mantiqiy qiymatlar, mantiqiy va munosabat amallari, arifmetik hamda belgili qiymatlar qatnashib, natijasi **true** va **false** qiymat bo'lishi mumkin. Unda amallar quyidagi tartibda bajariladi:

1. *not* (yo'q);
2. *** (ko'paytirish), */* (bo'lish), *div*, *mod*, *and* (va)
3. *+* (qo'shish), *-* (ayirish), *or* (yoki);
4. *<* (kichik), *<=* (kichik yoki teng), *=* (teng), *<>* (teng emas), *>=* (katta yoki teng), *>* (katta);

Bitta satrda berilgan amallar bir xil bajarilish o'rniga ega. Mantiqiy ifodalarda faqat oddiy qavslar yoziladi.

Masalan. 1) $(A>3)$ and $(B=A+6)$ or not $(c=4)$

2) $(A<=x)$ and $(x<=B)$ or $(c<=x)$ and $(x<=d)$

2.3. C++ tilida dasturning umumiy tuzilishi

C++ tilidagi dastur quyidagi tarkibdan tashkil topadi:

1. **Direktivalar** – `#include <file.h>` direktiva – instruksiya degan ma'noni beradi. C++ tilida dasturning tuzilishiga, ya'ni ehtiyojiga qarab, kerakli direktivalar ishlatiladi. Ular `<>` belgisi orasida keltiriladi. Umuman olganda quyidagi direktivalar mavjud (jami 32 ta):

- `#include <stdio.h>` - C++ da oddiy kiritish/chiqarish dasturi uchun. Bu yerda `std` - standart, `i` – input, `o` - output degani.
- `#include <iostream.h>` - C++ da kiritish/chiqarish uchun, oddiy amallar bajarilsa.
- `#include <math.h>` - standart funksiyalarni ishlatish uchun.

- `#include <conio.h>` - dasturning tashqi ko'rinishini shakllantirish uchun.
- `#include <string.h>` - satr toifasidagi o'zgaruvchilar ustida amallar bajarish uchun.
- `#include <stdlib.h>` - standart kutubxona fayllarini chaqirish uchun.
- `#include <time.h>` - kompyuter ichidagi soat qiymatlaridan foydalanish uchun.
- `#include <graphics.h>` - C++ tilining grafik imkoniyatlaridan foydalanish uchun.

Bu fayllar maxsus kutubxona e'lon fayllari hisoblanadilar va ular alohida INCLUDE deb nomlanadigan papkada saqlanadilar. Hozirda C++ kutubxonasini yangilandi va undagi fayllarning nomlaridan .h (head – bosh ma'nosida) kengaytmasi olib tashlandi va oldiga c harfi qo'shildi (C dan qolgan 18 tasiga). Bu fayllarda funksiya prototoifalari, toifalari, o'zgaruvchilar, o'zgarmlar ta'riflari yozilgan bo'ladi.

Direktivalar dasturni uni kompilyasiya qilinishidan oldin tekshirib chiqadi.

2. Makroslar - # define makro qiymati. Masalan:

```
#define y sin(x+25) – u = sin(x+25) qiymati berildi;
#define pi 3.1415 - pi = 3.1415
#define s(x) x*x - s(x) = x*x (; belgisi qo'yilmaydi)
```

Global o'zgaruvchilarni e'lon qilish. Asosiy funksiya ichida e'lon qilingan o'zgaruvchilar lokal, funksiya tashqarida e'lon qilinganlari esa global o'zgaruvchilar deyiladi. Global o'zgaruvchilar dastur davomida ishlaydi va xotiradan ma'lum joyni egallaydi. O'zgaruvchini bevosita ishlatishdan oldin e'lon qilsa ham bo'ladi, u holda o'z lokal bo'ladi. Global o'zgaruvchilar nomi lokal o'zgaruvchilar nomi bilan bir xil bo'lishi ham mumkin. Bunday holatda lokal o'zgaruvchining qiymati joriy funksiya ichidagini qiymatini o'zgartiradi, funksiya chiqishi bilan global o'zgaruvchilar ishlaydi.

Asosiy funksiya - **main ()** hisoblanadi. Bu funksiya dasturda bo'lishi shart. Umuman olganda C++ dagi dastur funksiyalardan iborat deb qaraladi. `main ( )` funksiyasi { boshlanadi va dastur oxirida berkitilishi shart } . `main` – asosiy degan ma'noni beradi. Bu funksiya oldida uning toifasi ko'rsatiladi. Agar `main ( )` funksiyasi beradigan (qaytaradigan) javob oddiy so'z yoki gaplardan iborat bo'lsa, hech qanday natija qaytarmasa, `void` so'zi keltiriladi. `main ( )` funksiyasi

dastur tomonidan emas, balki OS tomonidan chaqiriladi. OSga qiymat qaytarish shart emas, chunki u bu qiymatdan foydalanmaydi. Shuning uchun `main ( )` funksiyasining turini void deb ko'rsatganimiz ma'qul. Har bir funksiyaning o'z argumenti bo'ladi, shuning uchun `main` funksiya () lari ichiga uning parametri keltiriladi. Ba'zan u bo'sh bo'lishi ham mumkin. Bu funksiyadan chiqish uchun odatda return operatori ishlatiladi. 0 (nol) qiymatining qaytarilishi operasion tizimga ushbu dastur normal bajarilib turganini bildiradi. return orqali qaytadigan qiymat toifasi funksiya e'lonidagi qaytish toifasi bilan bir xil bo'lishi kerak. Masalan `int main ( )` va 0 (nol) qiymat butun toifalidir. Bu funksiyadan so'ng lokal o'zgaruvchilar, qism dasturlar, ularning haqiqiy parametrlar e'lon qilinadi. So'ngra dasturning asosiy operatorlari (kiritish/chiqarish, hisoblash va h.k.) yoziladi. Agar bu operatorlar murakkab toifali bo'lsalar, ularni alohida { } qavslarga olinadi. C++ tilida dastur kichik harflarda yoziladi. Ba'zi operatorlar katta harflar bilan kelishi mumkin, bunday xollarda ular alohida aytib o'tiladi. Operatorlar oxiriga ; belgisi qo'yiladi. Operatorlar bir qatorga ketma-ket yozilishi mumkin. Dasturda izohlar xam kelishi mumkin, ular /**/ belgisi orasiga olinadi. Agar izoh bir qatorda tugasa, uni // belgisidan keyin yoziladi. Masalan:

main () // C++ tilining asosiy funksiyasi

Izohlar /*..... */ichida yoziladi va ularni kompilyator tarjima qilmaydi.

Agar dasturda **clrscr();** prosedurasi ishlatilsa, ekranda avvalgi natijalari tozalanib, yangisi hosil bo'ladi.

Nazorat uchun savollar

1. Qanday arifmetik amallarni bilasiz?
2. Qiymat berish amali qanday?
3. Ifoda qanday tuzilgan?
4. Qanday turdagi ifodalardan foydalaniladi?
5. Inkriment va dekriment amallarini izoxlab bering?
6. Prefiks va postfiks amallarini tushintiring.
7. Qanday standart funksiyalar ishlatiladi?
8. Sizeof amali nima?
9. C++ tilida qanday qismlardan iborat?
10. Direktiva nima va qanday direktivalar ishlatiladi?
11. O'zgaruvchilar qanday tavsiflanadi?

Mustaqil bajarish uchun mashqlar

1. Arifmetik ifoda qiymatini hisoblang:

- 1) $\sin(\sqrt{x} - 1) + 2 * \text{abs}(y) / \cos(2 + y)$ bu yerda $x = 1, y = -2$
- 2) $\text{ceil}(6.9) - \text{floor}(6.2)$
- 3) $30 \% 6 - 30 \% 5 + \text{floor}(2.7)$
- 4) $4 * \arctan(1) - 1 + \text{fabs}(-1)$
- 5) $1 + 2.5$

2. Arifmetik ifoda turini aniqlang:

- 1) $1 + \sqrt{4} + \sin(0) + 0.1$
- 2) $\sqrt{2.0} + \sqrt{121} - \text{ceil}(-11.7)$
- 3) $\text{ceil}(32.7) + 10 \% 3$
- 4) $\text{floor}(\sin(0)) + 1$

3. Quyidagilardan qaysi biri qiymat berish operatori

- a) $a := b$ b) $a = c + 1$ c) $z :- 0$ d) $y := y$ f) $w := w - 1$ e) $-y := y$
f) $b := \text{true}$ j) $x := 'a' + 1$

4. Agar $\text{int } i, j, k, m; \text{ char } c, d; i = 1; j = 2; k = -7; m = 0; c = 'w';$ bo'lsa quyidagi ifodalardan qaysi biri to'g'ri yozilgan:

- a) $d = 'a' + 1 < c$ b) $m = -i - 5 * j >= k + 1$ c) $i + j++ + k == -2 * j$
d) $m = 3 < j < 5$ e) $m = 3 == j < 5$ f) $m == c = 'w'$
j) $m = c != 87$ s) $m = c = !87$ l) $m = !c = 87$
i) $m = !c + 87$ o) $i - k == '0' + 9 < 10$ p) $m != c + 87$

5. Ifodani C++ tilida yozing:

a) $z = (1 + u) \frac{x + \frac{y}{u}}{a - \frac{1}{1 - x^2}};$

b) $a = \frac{(x + y)^3 + \sqrt[3]{|x + y|}}{3}$

c) $S = 2\pi R^3 + \sqrt{R + vt}$

d) $y = a * e^{\lg x} - \sqrt{ax + b}$

e) $C = \sqrt{\frac{x^2 + y^2}{3}} + \ln|x + y|;$ f) $S = \ln \frac{x + \sqrt{1 + x^2}}{x} + \text{tg}^3 \frac{1 - e^x}{1 + e^x};$

3§.C++ tilida oddiy operatorlar

3.1. Qiymat berish buyrug'i .

Qiymat berish buyrug'i deyilganda qandaydir o'zgaruvchilarga biror qiymatni berish haqidagi buyruq tushuniladi.

C++ dasturlash tilida qiymat berish buyrug'i matematikadagi tenglik belgisi ("=") yordamida beriladi. Masalan: $X=6$; yozuvi matematikada "x oltiga teng" deb o'qilsa, C++ dasturlash tilida "x ga olti qiymat berilsin" deb o'qiladi.

O'zgaruvchilarga qiymat berish uchun o'zlashtirish operatori qo'llaniladi. Masalan, **Width** o'zgaruvchisiga 5 qiymatni berish uchun quyidagilarni yozish lozim:

unsigned short Width;

Width = 5;

Bu ikkala satrni Width o'zgaruvchisini aniqlash jarayonida birgalikda yozish mumkin.

unsigned short Width = 5;

Bir necha o'zgaruvchilarni aniqlash vaqtida ham ularga qiymat berish mumkin:

Long width = 5, length = 7;

Ma'lumki, dasturlash tillarida har bir e'lon qilingan o'zgaruvchi qiymatni yozib qo'yish uchun xotirada joylashtiriladi. Siz qiymat berish buyrug'ining mohiyatini to'g'ri tushunish uchun kompyuter xotirasini daftarning bir varog'i deb, varoqning har bir katakchasini xotira yachaykasi deb tushuning. Natijada, o'zgaruvchiga berilgan har bir qiymatni go'yo o'sha katakcha o'lchamidagi qog'ozga yozib, o'sha katakchaga kleylab qo'yiladi deb tushunasiz. Demak, bitta o'zgaruvchi bir nechta qiymatlar olsa, oxirgisi ko'rinib, oldingilari ko'rinmasdan qoladi.

Misol: $X = 5$; $X = X + 3$; $Y = 10$; $X = X - Y$; $X = X + Y$; $Y = X + Y$. Buyruqlar bajarilgach, X va Y lar qanday qiymatlar oladi.

Javob: qiymat berish buyrug'i ba'zi adabiyotlarda "o'zlashtirish operatori" yoki "o'zlashtirish amali" yoki "ta'minlash operatori" deb ham yuritiladi.

3.2.Ma'lumotlarni kiritish operatori.

Ma'lumotlar **cin** operatori yordamida kiritiladi. Qiymati kiritiladigan o'zgaruvchilar bu operatoridan keyin bir-birlaridan ">>" belgi bilan ajratilgan holda yoziladi.Uning umumiy ko'rinishi quyidagicha:

cin>>a1>>a2>>...>>an;

Bu yerda $a_1, a_2, \dots, a_n$ qiymatlari kiritilishi talab etilgan o'zgaruvchilarning nomi. **cin** operatori bajarilganda dastur bajarilishi to'xtaydi va kerakli ma'lumotlarni kiritilishini kutadi.

3.3. Ma'lumotlarni chiqarish operatori.

Dasturda hosil bo'lgan natijalarni chiqarish uchun chiqarish operatori ishlatiladi. Dastur natijalari **cout** operatori yordamida ekranga chiqariladi.

O'zgaruvchilar **cout** operatoridan keyin bir-birlaridan "<<" belgi bilan ajratilgan holda ko'rsatiladi. Uning umumiy ko'rinishi quyidagicha:

cout<<b1<<b2<<.....<<bn; bu yerda **b1,b2,...,bn** chiqarilayotgan natijalar.

3.4. Sizeof amali

O'zgaruvchilar, o'zgarmlar va turlarning xotiradan qancha joy egallashini aniqlash maqsadida **sizeof** amalidan foydalaniladi. Buni quyidagi dastur misolida ko'ramiz:

```
int main ( )
{ int a, b, c; float x, y, z;
cout << "\n butun =" << sizeof (int);
cout << "\n haqiqiy =" << sizeof (float);
cout << "\n bul turi =" << sizeof (bool);
cout << "\n a =" << sizeof (a);
cout << "\n x =" << sizeof (x);
return 0;
}
```

3.5. C++ tilida dasturda izohlar berish

C++ tilida dasturda operatorlarning bajarilishi hamda unda qatnashayotgan o'zgaruvchi va o'zgarmlarga izohlar berish imkoniyati mavjud. Izohlar matn ko'rinishida beriladi. Izohlar kompilyatorda qayta ishlanmaydi va operatorning bajarilishiga ta'sir qilmaydi.

C++ tilida izohlarni ikki xil ko'rinishda joylashtirish mumkin:

- ikkita ketma-ket keluvchi qiyshiq chiziq (//) orqali. Uni dasturni ixtiyoriy satriga joylashtirish mumkin. Bundan qisqa izohlar berish uchun foydalaniladi.
- Izohning boshiga / * belgini va oxiriga */ belgisini qo'yish orqali. / * va */ belgilari orasidagi matn izohni bildiradi. Bu usulda uzun izohlarni berish uchun foydalaniladi.

Masalan:

```
{
int i, n, k; // i, n, k lar butun turli o'zgaruvchilar
float x,y,z; / * x,y,z o'zgaruvchilar haqiqiy turga tegishli ekanligi
ko'rsatilgan va ular qabul qiladigan qiymat haqiqiy ko'rinishda bo'ladi
*/
.....
}
```

Nazorat uchun savollar

1. Operator nima?
2. C++ tilida dastur qanday tuzilgan?
3. Dasturning har bir qismini izohlab bering.
4. Ma'lumotlarni kiritish uchun qanday operatorlar ishlatiladi?
5. Kiritish operatoridan foydalanish usullarini tushintiring.
6. Ma'lumotlarni chiqarish uchun qanday operatorlar ishlatiladi?
7. Chiqarish operatoridan foydalanishni tushintiring.
8. O'zgaruvchilar qiymati va o'zgarishlar qanday chiqariladi?
9. C++ tilida dasturi izohlar qanday beriladi?
10. Chiziqli dastur nima?

Mustaqil bajarish uchun mashqlar

1. Ifodani qiymatini hisoblash dasturini tuzing:
$$a = \sqrt[3]{x + \sqrt[4]{y}}; \quad b = \sqrt{y} l^{-(y+4/2)},$$
2. Tomonlari bilan berilgan uchburchakning perimetri va yuzasini hisoblash dasturini tuzing.
3. Uchlarining koordinatalari bilan berilgan uchburchakning perimetri va yuzasini topish dasturini tuzing.
4. Asosi uzunligi va balandligi bilan berilgan uchburchak yuzasini hisoblash dasturini tuzing.
5. To'g'ri to'rtburchakning perimetri va yuzasini hisoblash dasturini tuzing.
6. Tomonlaridan biri va balandligi bilan berilgan parallelogramning yuzasini hisoblash dasturini tuzing.
7. Kvadratning perimetri va yuzasini hisoblash dasturini tuzing.
8. Radisi berilgan aylanaga ichki chizilgan to'g'ri olti burchakning perimetri va yuzasini hisoblash dasturini tuzing.
9. a,b,c sonlarining o'rta arifmetigi va o'rta geometrigini hisoblash dasturini tuzing.
10. Geometrik progressiyaning birinchi hadi va maxraji berilgan bo'lsa, uning n –hadi va n ta hadlar yig'indisini hisoblash algoritmi va dasturini tuzing.

4§.C++ tilida murakkab operatorlar

4.1. Shartsiz o'tish operatori.

Bu operator hech imkon bo'lmay qolgan holda dasturning tabiiy bajarilishini buzib, boshqaruvni dasturning ixtiyoriy joyiga uzatish uchun xizmat qiladi.

O'tish operatori quyidagicha beriladi:

goto nishon(belgi);

bu yerda **goto(..ga o'tish)** - operator nomi, **nishon** – bu belgi bo'lib, u harf, harflar yoki harf va sonlar bo'lishi mumkin. Bu operatoridan foydalanish uchun boshqaruv shartsiz ravishda uzatiladigan operator belgilanadi. Operatorni belgilash uchun uning oldiga **nishon** va : qo'yiladi: **nishon : operator;**

Masalan: **goto a;**

.....

a: y=x*x+c;

Bu yerda **goto a;** operatori bajarilgandan keyin boshqaruv to'g'ridan-tog'ri **a:y=x*x+c;** operatorga uzatiladi.

4. 2. Shartli o'tish operatori.

C++ tilida shartli o'tish operatorining qisqa va to'liq ko'rinishlaridan foydalaniladi.

Qisqa shartli o'tish operatori. Shartli operatorning qisqartirilgan ko'rinishi quyidagicha:

if (ifoda) {s1;s2;...;sn};

Bu yerda **if** kalit so'zi bo'lib, **agar** degan ma'noni bildiradi hamda har doim lotin alifbosining kichik harflari bilan yoziladi. Qavs ichida yozilgan **ifoda** arifmetik, mantiqiy, taqqoslash va boshqa amallardan tashkil topgan mantiqiy ifoda bo'lib, u **true** yoki **false** qiymatlarini qabul qiladi. **s1,s2,..sn** operatorlar ketma-ketligi.

To'liq shartli o'tish operatori. Shartli operatorning to'liq ko'rinishi quyidagicha: **if (ifoda) {s1;s2;...;sn;} else {p1;p2; ...;pm;}**

Bu yerda **if** va **else** kalit so'zlar bo'lib, mos ravishda **agar** va **aks holda** degan ma'noni bildiradi hamda ular har doim lotin alifbosining kichik harflari bilan yoziladi. Qavs ichida yozilgan **ifoda** arifmetik, mantiqiy, taqqoslash va boshqa amallardan tashkil topgan mantiqiy ifoda bo'lib, u **true** yoki **false** qiymatlarini qabul qiladi. **s1;s2;...;sn;** va **p1;p2; ...;pm** lar operatorlar ketma-ketligi.

? shart operatori.

? shart operatorining umumiy ko'rinishi quyidagicha:

X=B1 ? S1: S2;

bu yerda X-operatorni bajarilish natijasini qabul qiluvchi o'zgaruvchi; B1-mantiqiy ifoda bo'lib chin (true) yoki yolg'on (false) qiymatni qabul

qiladi; S1-mantiqiy ifodaning qiymati true bo'lgan bajariladigan operator; S2-mantiqiy ifodaning qiymati false bo'lgan bajariladigan operator.

4.3. Tanlash operatori.

Tanlash operatori C++ tilida hosil qilinadigan operatorlardan hisoblanib, shartli operatorning umumlashmasidan iboratdir.

Tanlash operatori quyidagicha umumiy ko'rinishga ega :

switch (ifoda)

```
{ case 1-o'zgarmas ifoda: 1-operatorlar ketma-ketligi; break;  
  case 2-o'zgarmas ifoda: 2-operatorlar ketma-ketligi; break;
```

.....

```
case n-o'zgarmas ifoda: n-operatorlar ketma-ketligi; break;
```

```
  default;
```

```
    operatorlar ketma-ketligi;
```

```
}
```

Tanlash operatorida ifodaning qiymatiga qarab, dasturda berilgan operatorlar ketma-ketligidan birini tanlaydi.

break operatori dastur bajarilishini **switch** operatoriga taluqli blokdan chiqarib yuboradi.

Agar ifodaning qiymati ro'yxatda berilgan ifodalarning qiymati bilan bir xil bo'lmasa **default** operatoridan keyingi operatorlar bajariladi va tanlash operatorini bajarilishi tugaydi.

Switch ($x \leq 0$)

```
{ case true:  $y := \sqrt{\sqrt{x} - x + 2}$ ;  $\text{cout} << y$ ; break;
```

```
  case false:  $y := \ln(x + 3)$ ;  $\text{cout} << y$ ; break;
```

```
default:
```

```
}
```

4.4. Sikl operatorlari. Parametrli sikl operatori.

Agar sikllik jarayonlarda takrorlanish soni oldindan ma'lum bo'lsa, ko'p hollarda parametrli sikl operatoridan foydalaniladi.

Parametrli sikl operatorining umumiy ko'rinishi quyidagicha:

for ($x = x_0$; $x \leq x_n$; $x = x + x_{\text{qad}}$)

```
{ 1-operator; 2-operator; ...; k-operator; }
```

Bu yerda **for (uchun)** - xizmatchi so'z; **x**- *sikl parametri* deb ataluvchi o'zgaruvchi; x_0, x_n - sikl parametrining mos ravishda *boshlangich va oxirgi qiymatlari*; x_{qad} - *sikl parametrining o'zgarish qadami*; { 1-operator; 2-operator; ...; k-operator; }- *sikl parametrning qabul qiladigan har bir qiymatida takroran bajariladigan operatorlar*.

Agar bu operatorlar bittadan ko'p bo'lsa { } belgi ichiga olinadi, aks holda shart emas.

4.5. Sharti oldin qo'yilgan sikl (while) operatori.

Sharti oldin qo'yilgan sikl (**while**) operatorining umumiy ko'rinishi quyidagicha: **while B {s1;s2;...;sn}** - bu yerda **while** (toki) xizmatchi so'z, **B**- mantiqiy ifoda, {s1;s2;...;sn}-operatorlar.

Bu operatorlarda {s1;s2;...;sn} operatorlar ketma-ketligi bajarilmasligi yoki bir martadan ortiq bajarilishi mumkin, chunki {s1;s2;...;sn} operatorlarning har bir bajarilishidan oldin B mantiqiy ifodaning qiymati tekshiriladi. Agar ifoda chin (true) qiymat qabul qilsa {s1;s2;...;sn} operatorlar bajariladi, aks holda, ya'ni B yolg'on (false) qiymat qabul qilsa, sikl operatorining bajarilishi tugaydi.

Agar **B** mantiqiy ifoda birinchi hisoblanishidayoq *yolg'on qiymatni* qabul qilsa, {s1;s2;...;sn} **operatorlar** bir marta ham bajarilmaydi. Ko'rinib turibdiki, bu yerda siklik jarayonning tugallanish sharti {s1;s2;...;sn} **operatorlar** bajarilishidan oldin tekshiriladi. Shuning uchun bunday ko'rinishdagi sikl operatori - *sharti oldin qo'yilgan sikl operatori* deb yuritiladi. Bu yerda ham takrorlanadigan operatorlar bittadan ortiq bo'lsa ular { } belgi ichiga olinadi, aks holda shart emas.

Masalan: 1) $y:=1; k:=0; m:=5; \text{while } y \leq m \{ y:=y+3; k:=k+1; \}$

2) $x:=3; y:=1; v:=1; \text{eps}:=0,0001;$

$\text{while } \text{abs}(v) > \text{eps} \{ V:=(x/y-y); y:=y+v; \}$

4.6 .Sharti keyin qo'yilgan (do..while) sikl operatori.

Bu operator takrorlanuvchi hisoblash jarayonlarida takrorlanishlar soni ma'lum, noma'lum bo'lgan holatlarda ishlatiladi. Operatorning umumiy ko'rinishi quyidagicha Sharti oldin qo'yilgan sikl (**while**) operatorining umumiy ko'rinishi quyidagicha: **do {s1;s2;...;sn} ; while B** bu yerda **while** (toki), **do** (bajar) xizmatchi so'zlar, **B**-mantiqiy ifoda, {s1;s2;...;sn ;} -operatorlar ketma-ketligi..

Bu operatorlarda {s1;s2;...;sn ;} operatorlar ketma-ketligi hech bo'lmaganda bir marta va undan ortiq bajarilishi mumkin, chunki {s1;s2;...;sn ;} operatorlar bajarilgandan keyin **B** mantiqiy ifodaning qiymati tekshiriladi. Agar ifoda chin (true) qiymat qabul qilsa {s1;s2;...;sn;} operatorlar bajariladi, aks holda, ya'ni **B** yolg'on (false) qiymat qabul qilsa, sikl operatorining bajarilishi tugaydi.

Bu operatorlarda siklik jarayonning tugallanish sharti {s1;s2;...;sn;} operatorlar bajarilgan dan keyin tekshiriladi. Shuning uchun bunday

ko'rinishdagi sikl operatori - *sharti keyin qo'yilgan sikl operatori ham* deb yuritiladi. Bu yerda ham takrorlanadigan operatorlar bittadan ortiq bo'lsa ular { } belgi ichiga olinadi Operatorning bajarilishida **do...while** operatorlari orasida joylashgan operatorlar ketma- ketligi bir marta va undan ortiq bajarilishi mumkin. Bu jarayon qo'yilgan mantiqiy ifoda **yolg'on qiymat** qabul qilgunga qadar davom etadi.

do..while operatorining **while** operatoridan farqi shundaki, bunda eng avval operatorning qiymati hisoblanib, so'ngra qo'yilgan shart tekshiriladi. Bu esa bunda berilgan operatorning hech bo'lmaganda bir marta bajarilishiga imkon yaratadi.

Masalan:

```
y:=1; do
v:=(2*x/y-3*y)/2; y:=y+2*v
while( abs(v)<eps);
```

4.7. Continue va break operatorlari

continue operatori sikl tanasining qolgan qismini o'tkazib yuborish uchun ishlatiladi, ya'ni u takrorlanish tanasi bajarilishini to'xtatib, keyingi qadamga o'tishni ta'minlaydi. **While** va **do ... while** operatorlarida u boshqaruvni sikl shartini qayta tekshirish uchun uzatadi, ya'ni **while** keyin qavs ichida yozilgan ifoda hisoblanadi va tahlil qilinadi, so'ngra agar u rost bo'lsa siklni ishlashi davom etadi.

for operatorida qiymatni oshirish qismi bajariladi, so'ngra shart tekshiriladi, agar u rost bo'lsa sikl bajariladi.

Nazorat uchun savollar

1. Tarmoqlanuvchi jarayonni tushuntiring.
2. Tarmoqlanuvchi jarayonlarni dasturlash uchun qanday operatorlar ishlatiladi?
3. Shartli o'tish operatori qanday tuzilmalarga ega?
4. To'liq shartli o'tish operatorini bajarilishini tushuntiring.
5. Murakkab shartli o'tish operatorlari qanday bo'ladi?
6. Shartsiz va shartli o'tish operatorlarini vazifasini tushuntiring.
7. Tanlash operatorining vazifasi nimadan iborat va uning umumiy ko'rinishi?
8. Tanlash operatorining shartli operatoridan farqi qanday?
9. ? shart operatorini bajarilishini tushuntiring.
10. Nima uchun gotooperatori ko'p ishlatilmaydi?
11. Takrorlanuvchi jarayonni tushuntiring.
12. Takrorlanuvchi jarayonlarni dasturlash qanday amalga oshiriladi?

13. Takrorlanuvchi jarayonlar uchun shartli o'tish operatorlari orqali dastur qanday tuziladi?
14. Qanday takrorlash operatorini bilasiz?
15. while, do-while operatorlari qanday ishlaydi?
16. while va do .. while operatorlarini qanday farqlari bor?
17. For operatoridan foydalanish tartibini tushuntiring.
18. Boshqaruv operatorlari qayerda ishlatiladi?
19. Qaysi sikl operatorlarining imkoniyatlari keng?
20. Dasturlash jarayonini tushuntiring.
21. Sikl operatorlarining farqi nimada?
22. for siklida bir nechta schyotchik ishlatish mumkinmi?
23. Tanasida biror amal yozilmagan for operatori yordamida sikl tashkil etish mumkinmi?
24. for sikli ichida while siklini tashkil etish mumkinmi?
25. Hech qachon tugallanmaydigan sikl tashkil etish mumkinmi?

Mustaqil bajarish uchun topshiriqlar

1. a,b,c sonlarini o'sish tartibida joylashtirish dasturini tuzing.
2. a,b,c kesmalar uzunliklari bilan berilgan. Ular orqali uchburchak yasash mumkin yoki mumkin emasligini aniqlash dasturini tuzing.
3. Yuzasi S_1 ga teng bo'lgan doira, yuzasi S_2 ga teng bo'lgan kvadratga joylashishini aniqlash dasturini tuzing.
4. Koordinatalari bilan berilgan $a(x,y)$ nuqtani koordinatalar tekisligida joylashish o'rnini aniqlash dasturini tuzing.
5. Berilgan ikkita a va b sonlarining kichigini ularning yig'indisi, kattasini esa ularning ko'paytmasi bilan almashtirish dasturini tuzing.
6. a,b,c,d sonlarini kamayish tartibida joylashtirish dasturini tuzing.
7. $a(x,y)$ nuqtani uchlari koordinatalari $a(x_1,y_1)$ va $b(x_2,y_2)$ ga teng bo'lgan kesmaga tegishli bo'lishligini aniqlash dasturini tuzing.
8. $A(x,y)$ nuqtani radiusi R ga, markazi $O(0,0)$ nuqtada joylashgan aylanaga tegishli bo'lishligini aniqlash dasturini tuzing.
9. 1 dan n gacha bo'lgan natural sonlarning o'rta arifmetigi va o'rta geometrigini hisoblash dasturini tuzung.
10. Barcha juft ikki xonali sonlarni hosil qilish dasturini tuzung.
11. Barcha 4 ga karrali bo'lgan uch xonali sonlar yig'indisini hisoblash dasturini tuzung.
12. $y = 5x^2 - 2x + 1$ funksiyani $[-5; 5]$ oraliqda $h = 2$ qadam bilan qiymatlarini hisoblash dasturini tuzing.
13. $[a,b]$ oraliqdagi barcha tub sonlarini hosil qilish dasturini tuzing.

14. Berilgan N sonining raqamlari yig'indisi va ko'paytmasini hisoblash dasturini tuzing.
15. Berilgan N sonini tub ko'paytuvchilarga ajratish dasturini tuzing.
16. Barcha toq ikki xonali sonlarni hosil qilish dasturini tuzung.
17. Barcha 3 va 7 ga karrali bo'lgan uch xonali sonlar yig'indisini hisoblash dasturini tuzung.
18. Oylarning tartib raqamiga mos ravishda bahor, yoz, kuz va qish so'zlarini hosil qilish dasturini tuzing.
19. Oylarning tartib raqamiga mos ravishda kvartallarni hosil qilish dasturini tuzing.
20. Talabalarining reyting natijalari asosida a'lo, yaxshi, qoniqarli va yomon so'zlarini hosil qilish dasturini tuzing.
21. $S = \sum_{n=1}^{10} 2^n + \prod_{k=1}^5 3^k + \sum_{m=1}^{10} 4^m$ ni hisoblash algoritmi va dasturini tuzing.
22. $S = \prod_{a=1}^{10} \prod_{i=2}^4 \frac{k^3}{k+a}$ ni hisoblash dasturini tuzing.
23. $S = \sum_{n=1}^3 \prod_{k=1}^5 \sum_{j=1}^{10} (k^2 + j^2 + n)$ ni hisoblash dasturini tuzing.
24. $S = \prod_{k=1}^3 (a \prod_{n=1}^4 kn^2 + b \sum_{i=1}^5 ki^3)$ ni hisoblash dasturini tuzing.
25. EKUK(a,b,c) ni topish dasturini tuzing.

5§. C++ tilida massiv elementlari bilan ishlash

5.1. Bir o'lchovli massiv elementlari bilan ishlash

C++ tilida massiv elementlarining indeksleri 0 dan boshlanadi va uning eng katta qiymati massiv e'lonidagi uzunlikdan bitta kam bo'ladi

Bir o'lchovli massivlar quyidagicha tavsiflanadi:

tur nom [<uzunlik>];

bu yerda **tur**-massiv elementlarining turi bo'lib, u **int**, **float**, **char** kabi ma'lumotlar turidan iborat bo'lishi mumkin; **nom** – massiv nomi bo'lib foydalanuvchi tomonida ixtiyoriy aniqlanadi va identifikatorlar qoidasi bo'yicha yoziladi; **uzunlik** – butun sondan iborat bo'lib, massivni tashkil etgan elementlar soni, yoki elementlar joyshadigan yachaykalar sonini aniqlaydi.

Masalan: 1) **float A[10];**

2) **n=10; float A[n];**

3) **int a[10] = {9,8,-8,-6,-5,5,4,3,12,14}**

Massiv elementlarini kiritish

Massiv elementlarini quyidagi ko'rinishlarda berish mumkin:

1) Massiv elementlarini tavsiflash vaqtida:

a) **int a[10] = {9,8,-8,-6,-5,5,4,3,12,14}**

b) **int n=20; a[n] = {9,8,-8,-6,-5,5,4,3,12,14}**

c) **char T[11]={‘м’,’а’,’т’,’е’,’м’,’а’,’т’,’и’,’к’,’а’};**

d) **char T[11]=“математика”;**

Bunda agar massiv elementlari soni ko'rsatilgan chegaradan kam bo'lsa, uning qolgan elementlari 0 qiymatni qabul qiladi. Agar massiv elementlari soni ko'rsatilgan chegaradan ko'p bo'lsa, u kompilyatorga uchun tushunarsiz bo'ladi.

2) Massiv elementlarini klaviaturadan kiritish. Buning uchun sikl poeratorlaridan foydalaniladi. Masalan,

```
for (i=0; i<=9; i++)  
    cin>>a[i];
```

Bu yerda **a[i]** massivning **a[0]** dan **a[9]** gacha elementlariga qiymat beriladi.

```
for (i=1; i<=10; i++)  
    cin>>a[i];
```

Bu holda **a[i]** massivning **a[1]** dan **a[10]** gacha elementlariga qiymat beriladi. **a[0]** element 0 qiymatga ega bo'ladi

3) Ayrim hollarda massiv elementlarini tasodifiy sonlar datchgi orqali ham berish mumkin:

```
const n=5; int a[n]; randomize();  
for (int i=0; i<n; i++)  
    a[i]=random(100);
```

Massiv elementlarini chiqarish

Uncha katta bo'lmagan massivlar elementlarini quyidagicha chop etish mumkin:

```
cout<< “\n Массив: \n “;  
for (int i=0; i<n; i++)  
    cout<<A[i]<<" ";  
cout<<endl;
```

Katta hajmdagi massivlar elementlarini quyidagicha chop etish mumkin:

```
int K=10;  
for (int i=0; i< n; i++)  
{    cout<<A[i]<<" ";  
    if ((i+1) %K == 0)    cout<<endl;  
}
```

5.2. Ikki o'lchovli massiv elementlari bilan ishlash

Ikki o'lchovli massivlar matematikada matrisa yoki jadval tushunchasiga mos keladi. Jadvallarning inisializasiya qilish qoidasi, ikki o'lchovli massivning elementlari massivlardan iborat bo'lgan bir o'lchovli massiv ta'rifiga asoslangandir. Aytaylik $a(n,m)$ o'lchovli matrisa berilgan:

$$A = \begin{vmatrix} a_{00} & a_{01} & \dots & a_{0n} \\ a_{10} & a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots & \dots \\ a_{m0} & a_{m1} & \dots & a_{mn} \end{vmatrix}$$

Matrisa elementlarini belgilash uchun ikki indeksli o'zgaruvchi ishlatiladi, $a[n][m]$, bu yerda n -satrlar soni, m -ustunlar soni.

Matrisa quyidagicha e'lon qilingan bo'lishi mumkin:

const n=5, m=5; int A[n][m],

bu yerda n - satrlar soni(birinci, chap o'lchov), m – ustunlar soni yoki satrdagi elementlar soni (ikkinchi o'ng o'lchov). Matrisa tezkor xotirada satr bo'yicha joylashadi va uzluksiz joyni egallaydi, hamda uning hajmi $n*m*sizeof(int)$ ga teng bo'ladi, bu yerda int – matrisa elementlarining turi.

Matrisani tavsiflashning u yoki bu usulu konkret masalaning turiga qarab belgilanishi ham mumkin. Matrisa satr va ustunlarni nomerlash 0 dan boshlanadi.

Matrisani inisializiyalash uni tavsiflash vaqtida amalga oshirilishi mumkin, masalan:

$$a = \begin{vmatrix} 1 & -2 & 3 \\ 10 & 20 & 33 \\ -11 & 22 & 36 \end{vmatrix}$$

int A[3][3]= {{1, -2, 3}, { 10, 20, 33}, {-11, 22, 36}};

Agar satrda talab qilinganidan kam sondagi element berilgan bo'lsa, u holda qolganlari 0 bilan inisialaziya qilinadi:

int A[5][3]= {{1, -2, 3, -4, 5}, { 10, 20, 33, -40}, {-11, 22, 300, 400, 500}};

Bu misolda 2-satrning oxiri elementi 0 ga teng bo'ladi.

Misol uchun ikki qator va uch ustundan iborat bo'lgan haqiqiy turiga tegishli d massiv boshlang'ich qiymatlari quyidagicha ko'rsatilishi mumkin:

float a[2][3]={(1,-2.5,10),(-5.3,2,14)};

Bu yozuv quyidagi qiymat berish operatorlariga mosdir:

a[0][0]=1;a[0][1]=-2.5;a[0][2]=10;a[1][0]=-5.3;a[1][1]=2;a[1][2]=14;

Bu qiymatlarni bitta ro'yxat bilan hosil qilish mumkin:

float a[2][3]={1,-2.5,10,-5.3,2,14};

Inisializasiya yordamida boshlang'ich qiymatlar aniqlanganda massivning hamma elementlariga qiymat berish shart emas. Misol uchun: **int x[3][3]={(1,-2,3),(1,2),(-4)}.**Bu yozuv quyidagi qiymat berish operatorlariga mosdir: **x[0][0]=1;x[0][1]=-2;x[0][2]=3;x[1][0]=-1;x[1][1]=2;x[2][0]=-4;**

Inisializasiya yordamida boshlang'ich qiymatlar aniqlanganda massivning birinchi indeksi chegarasi ko'rsatilishi shart emas, lekin qolgan indekslar chegaralari ko'rsatilishi shart. Misol uchun:

Double x[] [2]={(1.1,1.5),(-1.6,2.5),(3,-4)}

Bu misolda avtomatik ravishda qatorlar soni uchga teng deb olinadi.

Ikki o'lchovli massiv elementlarini kiritish

Oddiy ko'rinishda ikki o'lchovli massiv elementlarini kiritish quyidagicha bajariladi:

for (i=0; i<n; i++)

for (j=0; j<m; j++)

cin>>A[i][j];

Matrisa elementlarini tasodifiy sonlar orqali aniqlash

Matrisa elementlarini kiritish uchun ketadigan vaqtni tejash maqsadida ba'zi masalalarni yechish uchun massiv elementlarini tasodifiy sonlar generatori orqali aniqlash mumkin:

for (i=0; i<n; i++)

for (j=0; j<m; j++)

A[i][j]=rand();

Agar masalaga qo'yilgan shart bo'yicha, matrisa elementlari [a,b) oralikka tegishli bo'lishi kerak bo'lsa, bu yerda $a > b$, u holda sikl operatorida $A[i][j] = \text{random}(b-a)+a$ yozish kerak. Bu usuldan matrisa elementlari uchun cheklov qo'yilmagan paytda foydalanish mumkin

Massiv elementlariga son qiymat berishda kompyuter xotirasidagi tasodifiy butun sonlardan foydalanish xam mumkin. Buning uchun standart kutubxonaning **rand ()** funksiyasini ishga tushirish kerak. **rand**

() funksiyasi yordamida $0 \wedge 32767$ oraliqdagi ixtiyoriy sonlarni olish mumkin. Bu qiymatlar umuman tasodifiydir. (**psevdo** - tasodifiy degani). Agar dastur qayta-qayta ishlatilsa, ayti tasodifiy qiymatlar takrorlanaveradi. Ularni yangi tasodifiy qiymatlar qilish uchun **srand** () funksiyasini dasturda bir marta e'lon qilish kerak. Dastur ishlashi jarayonida ixtiyojga qarab **rand** () funksiyasi chaqirilaveradi. Tasodifiy qiymatlar bilan ishlash uchun **<stdlib.h>** fayilni e'lon qilish zarur.

Ikki o'lchovli massiv elementlarini chiqarish

Oddiy ko'rinishda ikki o'lchovli massiv elementlarini kiritish quyidagicha bajariladi:

```
for ( i=0; i<n; i++)  
    for ( j=0; j<m; j++)  
cout<<a[i][j];
```

Nazorat uchun savollar

1. Massiv deb nimaga aytiladi?
2. C++ tilida qanday turdagi massivlar ishlatiladi?
3. C++ tilida massivlarni qanday tavsivlanadi?
4. Bir o'lchovli massiv elementlarini kiritish va chiqarish usullari qanday?
5. Massiv elementlariga son qiymat berishda tasodifiy qiymatlardan foydalanish qanday bajariladi?
6. Massiv elementlarini natija sifatida chiqarish usullari qanday?
7. Bir o'lchovli massiv elementlari uchun qanday o'zgaruvchilar ishlatiladi?
8. Ikki o'lchovli massivlarni ta'riflang.
9. Ikki o'lchovli massiv elementlari qanday tavsiflanadi?
10. Ikki o'lchovli massiv elementlari qanday kiritiladi?
11. Ikki o'lchovli massiv elementlari qanday chiqariladi?
12. Ikki o'lchovli massiv elementlari uchun qanday o'zgaruvchilar ishlatiladi?
13. Ikki o'lchovli massiv elementlari qanday anqilanadi?

Mustaqil bajarish uchun topshiriqlar

1. A va B ikkita vektorning elementlari butun sonlarda iborat. A vektorning toq elementlardan tashkil topgan, lekin B vektorning elementi bo'lmagan C vektorni hosil qiling.
2. Qo'shimcha, massivdan foydalanmasdan turib, massiv elementlarini teskari tartibda joylashtiring.
3. X(N) massivning eng kichik va eng katta elementlarini

ro'yxatdan chiqarib, massivning o'rta arifmetigini hisoblang.

4. $B(N)$ massiv elementlarini, B massiv elementlari yig'indisiga bo'lib, yangi elementlardan tashkil topgan B massivni hosil qiling.

5. $X(M)$ vektorning eng katta elementigacha bo'lgan barcha elementlarini nol bilan almashtiring.

6. $R(K)$ massivning musbat elementlari orasidan eng kichigini va manfiy elementlari orasidan eng kattasini toping.

7. $Y(N)$ vektorning eng kichik elementidan keyingi barcha elementlarini nol bilan almashtiring.

8. X butun son va $A(N)$ butun tipdagi massiv berilgan. Massiv elementlarini shunday o'zgartiring, birinchi X sondan kichiklar, keyin X ga eng bo'lganlari va undan so'ng X dan kattalari joylashsin, faqat ularning massivda turgan tartibi buzilmasin.

9. Tekislikda R va R_1 ichki va tashqi radiusga ega bo'lgan doiraviy halqaning markazidan $s[1], s[2], \dots, s[10]$ uzoqlikda nuqtalar joylashgan. Xalqa ichida joylashgan nuqtalar sonini aniqlang.

10. X vektor berilgan. Vektor elementlarining tartibini buzmasdan, yangi massiv kiritmasdan, faqat nolga teng bo'lgan elementlarni massivning oxiriga joylashtiring.

11. Berilgan N musbat butun son uchun $A(N, M)$ matritsa hosil qiling, uning diagonal elementlari birdan, qolganlari noldan iborat bo'lsin.

12. Berilgan N musbat butun son uchun $A(N, M)$ matritsa hosil qiling, uning diagonal elementlari satrlarining tartib raqamiga, qolgan elementlari noldan iborat bo'lsin.

13. $U(K)$ massivi berilgan. $[a, b]$, intervaliga tegishli bo'lgan elementlar sonini aniqlang va ekranga chop eting.

14. $R(N)$ massivi berilgan. D massivga uning maksimal va minimal elementidan tashqari bo'lgan elementlarni yozing

15. $X(N)$ massivining o'rta arifmetik va o'rta geometrik qiymatlarini aniqlang. X massivining o'rta arifmetik qiymatidan katta va o'rta geometrik qiymatdan kichik bo'lgan elementlar sonini aniqlang va chop eting.

16. $U(K)$ massivining musbat, toq indeksli elementlarining yig'indisini toping.

17. $Z(K)$ massivining manfiy, juft indeksli elementlarining ko'paytmasini toping.

18. $H(N)$ massivining o'rta arifmetik qiymatidan kichik bo'lgan elementlarining ko'paytmasini toping.

19. $H(N)$ massivining eng katta va eng kichik elementlarini toping va massivda ulardan qaysi biri avval joylashgani ni aniqlang.
20. Berilgan ikkita $X=(X_1, X_2, \dots, X_n)$ va $Y=(Y_1, Y_2, \dots, Y_n)$ vektorning skalyar ko'paytmasini toping.
21. $X(N)$ massiv berilgan. X massiv elementlarini kamayib borish tartibida joylang.
22. $X(N)$ massivining musbat elementlaridan Y massivini tashkil eting, so'ngra Y massiv elementlarini o'sib borish tartibida yozing
23. $A(N, N)$ matritsaning p va q tartib raqamli satrlari o'rnini almashritiring.
24. $A(N, M)$ matritsaning oxirgi satridan oldinda joylashgan satrini o'chirib yangi matritsa hosil qiling.
25. $X(K, L)$ matritsaning eng katta va eng kichik elementlari o'rnini almashtiring.
26. $A(N, N)$ matritsa elementlarini quyidagi tartibda tanlang: diagonal elementlari birdan, diagonaldan yuqoridagi elementlar noldan, diagonaldan pastdagi elementlar mos indekslarning yig'indisiga teng bo'lsin.
27. $B(3, 5)$ matritsa berilgan. B matritsaning eng kichik elementi joylashgan satr va ustunni yo'qotish yo'l bilan V matritsani hosil qiling.
28. $A(M, N)$ matritsa berilgan. Matritsani $(M+1)$ satr va $(N+1)$ ustun bilan to'ldiring, bu satr va ustunning elementlari dastlabki matritsaning mos satr va ustunlarining yig'indisidan tashkil topsin.
29. $X(N, M)$ matritsani transponerlarng.
30. $A(3, 4)$ matritsaning satr elementlari ko'paytmasidan B massivni hosil qiling.

6§.C++ tilida funksiyalardan foydalanish

6.1. Funksiyalarni tavsiflash

Funksiya ta'rifida funksiya nomi, tipi va formal parametrlar ro'yxati ko'rsatiladi. Formal parametrlar nomlaridan tashqari tiplari ham ko'rsatilishi shart. Formal parametrlar ro'yxati funksiya signaturasi deb ham ataladi. Funksiya ta'rifi umumiy ko'rinishi quyidagichadir:

Funksiya turi funksiya nomi(formal_parametrlar_ta'rifi)

Formal parametrlarga ta'rif berilganda ularning boshlang'ich qiymatlari ham ko'rsatilishi mumkin.

Funksiya qaytaruvchi ifoda qiymati funksiya tanasida **return**<ifoda> ; operatori orqali ko'rsatiladi.

Misol: Float min(float, float b) { if (a<b) return a; return b; }

Funksiyaga murojaat qilish quyidagicha amalga oshiriladi:

Funksiya nomi (haqiqiy parametrlar ro'yxati)

Haqiqiy parametr ifoda ham bo'lishi mumkin. Haqiqiy parametrlar qiymati hisoblanib mos formal parametrlar o'rnida ishlatiladi.

Misol uchun yuqoridagi funksiya quyidagicha murojaat qilish mumkin:

```
Int x=5,y=6,z; z=min(x,y) yeki int z=Min(5,6) yeki int x=5; int  
z=min(x,6)
```

```
Masalan: d:=sin(0.7); f:=fak(45);
```

```
S1:=uchburchak(2,6,7);
```

6.2. Funksiyalarga murojaat qilish.

Funksiyaga murojaat standart funksiylarga o'xshash amalga oshiriladi (**sin, cos, tan va hakoza**) va ifodadagi operandalarni turli xilligi bilan procedurani chaqirishdan farq qiladi. Qayerda ifodani yozish kerak bo'lsa, o'sha joyda funksiya nomi yoziladi, undan keyin qavs ichida faktik parametrlar beriladi. Agar parametrsiz funksiya chaqirilsa, u holda faqatgina funksiyaning nomi ko'rsatiladi.

Funksiyaga murojaat qilinganda biror haqiqiy parametr ko'rsatilmasa, uning o'rniga mos formal parametr ta'rifida ko'rsatilgan boshlangich qiymat ishlatiladi.

```
Misol uchun: float min(float a=0.0, float b) { if (a<b) return a;  
return b; }
```

Bu funksiya yuqorida ko'rsatilgan murojaat usullaridan tashqari quyidagicha murojaat qilish mumkin: **Int y=6,z; z=min(,y) yeki int z=Min(,6);**

Agar funksiya hech qanday qiymat qaytarmasa uning tipi **void** deb ko'rsatiladi.

```
Misol uchun: Void print; { Cout<<("\n Salom!"); }
```

6.3. Rekursiya.

Rekusiv funksiyalar. Rekursiv funksiya deb o'ziga o'zi murojlat qiluvchi funksiya aytiladi . Misol uchun faktorialni hisoblash funksiyasini keltiramiz:

```
Long fact(int k) {if (k<0) return 0; if (k==0) return 1; return  
k*fact(k-1);}
```

Manfiy argument uchun funksiya 0 qiymat qaytaradi. Parametr 0 ga teng bo'lsa funksiya 1 qiymat qaytaradi. Aks holda parametr qiymat birga kamaytirilgan holda funksiyaning o'zi chaqiriladi va uzatilgan par-

ametr ga ko'paytiriladi. Funktsiyaning o'z-o'zini chaqirish formal parametr qiymati 0 ga teng bo'lganda to'xtatiladi.

Keyingi misolimizda ixtiyoriy haqiqiy sonning butun darajasini hisoblash rekursiv funksiyasini keltiramiz. **Double expo(double a, int n)**

{ if (n==0) return 1; if (a==0.0) return 0; if (n>0) return a*expo(a,n-1); if(n<0) return expo(a,n+1)/a; }

Nazorat uchun savollar

1. Funktsiya deb qanday strukturaga aytiladi?
2. Funktsiya nimalardan tashkil topadi?
3. Funktsiyadan qanday chiqiladi?
4. Funktsiyaning soxta parametrlari qanday aniqlanadi?
5. Funktsiyaga murojaat qilish, uning haqiqiy parametrlarini aniqlash.
6. Funktsiyalarni e'lon qilish qoidalari.
7. Funktsiya ichida parametrlar qiymatlarini o'zgartirish.
8. Funktsiya - proseduralar hosil qilish va ularga murojaat qilish.
9. Rekursiya holati. Rekursiyani to'xtatish.

Mustaqil topshiriqlar

Functiondan foydalanib hisoblash dasturini tuzing

1. $U = f(a,b) + f(a^2, b^2) + f(a^2 - 1, b)$, buyerda

$$f(u,t) = \begin{cases} u^2 + t^2, & \text{agar } u > 0, t > 0 \\ u + t^2, & \text{agar } u \leq 0, t \leq 0 \\ u^2 + t, & \text{qolg an hollarda} \end{cases}$$

2. $U = \frac{EKUB(m+n, n, k+m)}{EKUB(m, n+k, m+n)} + EKUB(m+n, n+k, m+k)$

3. $S = f(a,b) + f(a^2, b^2) + f(a^2 - 1, b)$, buyerda

$$f(u,t) = \begin{cases} u^2 + t^2, & \text{agar } u > 0, t > 0 \\ u + t^2, & \text{agar } u \leq 0, t \leq 0 \\ u^2 + t, & \text{qolg an xollarda} \end{cases}$$

4. N ta uchburchak o'z tomonlari bilan berilgan. Bu uchburchaklar yuzlarini hisoblang va ulardan kattasini aniqlang.

5. $Z = f(\sqrt{x}, y) + f(a, b) + f(\sqrt{x} + 1, y) + f(x + y, x)$, bu yerda

$$f(u,t) = \begin{cases} u + 2t, & \text{agar } u > 0 \\ u + t, & \text{agar } u \leq -1 \\ u^2 - 2t + 1, & \text{qolg an xollarda} \end{cases}$$

6. n ta uchburchakning uchlari koordinatalari bilan berilgan. Bu uchburchaklarning yuzlarini hisoblang va ulardan kichigini toping.

$$7. P = \frac{EKUB(n, m, n+n) + EKUB(n+m, n+k, m+k)}{EKUB(m^2, n^2, m+n)}, \quad m, n, k \in N$$

8. n ta uchburchak o'z tomonlari bilan berilgan. Bu uchburchaklarga ichki chizilgan aylana radiuslarini hisoblang va ularning kattasini aniqlang.

$$9. B = \begin{cases} \frac{T(3x+z, y+x) + T(x^2-y, z^2)}{T(x-y^2, z) + T(2x+y, z+y)} & \text{bu yerda, } T(a,b) = \begin{cases} \sqrt{a^2+b^2}, & \text{agar } a > b \\ \sqrt[3]{2a^2-b}, & \text{agar } a \leq b \end{cases} \end{cases}$$

10. n ta muntazam uchburchak tomonlari bilan berilgan, bu uchburchakka tashqi chizilgan aylana radiuslarini hisoblang va ularni kamayib borish tartibida joylashtiring.

Testlar

1. C++ dasturlash tilida funksiyayordamida faollashtiriladi.

A) funksiya murojaat B) funksiyaning tafsirlash C) funksiyaning aniqlash

D) funksiyaning yopish E) funksiyaning o'zgarishi

2. Funksiya ichida ma'lum va unda aniqlangan o'zgaruvchi – bu

A) lokal o'zgaruvchi B) global o'zgaruvchi C) haqiqiy turdagi o'zgaruvchi

D) bazaviy o'zgaruvchi E) rekursiv o'zgaruvchi

3. Ixtiyoriy blok yoki funksiya tashqarida e'lon qilingan o'zgaruvchi-bu...

A) lokal o'zgaruvchi B) global o'zgaruvchi C) haqiqiy turdagi o'zgaruvchi

D) bazaviy o'zgaruvchi E) rekursiv o'zgaruvchi

4. Qaysi funksiya tasodifiy sonlarni hosil qilish uchun ishlatiladi?

A) rand B) srand C) return D) atan2 E) floor

5. O'z-o'ziga murojaat qiliuvchi funksiya-bu..

A) rekursiv funksiya B) bazaviy funksiya C) bosh funksiya

D) prototip funksiya E) global funksiya

III-bob. Matematik masalalar uchun dasturlash tillarida dastur tuzish.

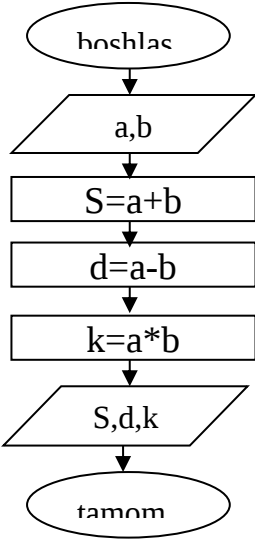
3.1. Matematik masalalar uchun chiziqli dastur tuzish

Chiziqli dastur tuzish masalani yechish uchun kerak bo'ladigan boshlang'ich ma'lumotlarni tashkil qiluvchi o'zgaruvchilar nomi, ularning turi va o'zgarish ko'lamini aniqlashdan boshlanadi. Keyin oraliq va yakuniy natijalar o'zgaruvchilarining nomlari, turlari va mumkin bo'lsa o'zgarish ko'lamini aniqlash kerak.

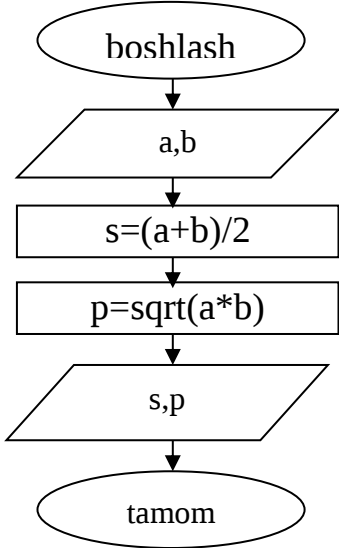
Chiziqli dastur deganda masalaning algoritmgiga mos keluvchi buyruqlar ketma-ketligi dasturda uchrash tartibiga mos ravishda bajariladigan dasturlar tushuniladi. Dasturning bitta satrida bir nechta buyruqlar kelishi mumkin. Bunda ular bir-biridan «;» belgisi bilan ajratiladi. C++ tilidagi dastur qo'yilgan masalaning yechish algoritmidagi buyruqlarni kompilyatorga «tushunarli» bo'lgan ko'rinishda yuqoridagi umumiy sxemaga muvofiq ifodalash natijasida hosil bo'ladi.

1-masala. A va B ikkita haqiqiy sonlar berilgan. Ularning yig'indisi, ayirmasi va ko'paytmasini hisoblang.

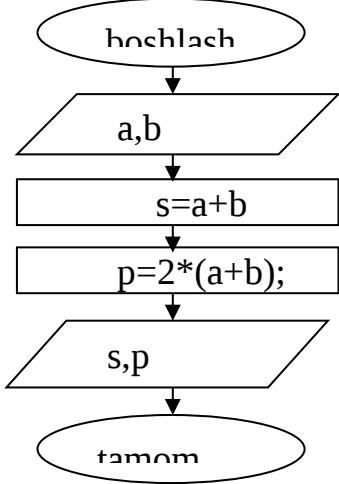
Yechish. A va B sonlar yig'indisini S, ayirmasini D, ko'paytmasini K bilan belgilasak, $S=a+b$, $d=a-b$, $k=a*b$ formulalar o'rinli bo'ladi.

Algoritm	C++ da dasturi
 <pre> graph TD Start([boshlac]) --> Read[/a,b/] Read --> S[S=a+b] S --> d[d=a-b] d --> k[k=a*b] k --> Print[/S,d,k/] Print --> End([tamom]) </pre>	<pre> #include<iostream.h> int main() { float a,b,s,p,k; cout<<"a=";<<cin>>a; cout<<"b=";<<cin>>b; s=a+b; k=a-b; p=a*b; cout <<"s=" <<s; cout<<"p=" <<p; cout<<"k=" <<k; system("PAUSE"); return 0; } </pre>

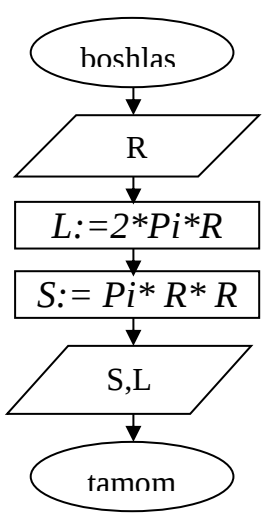
2-masala. Ikkita musbat son berilgan, bu sonlarning o'rta arifmetik va o'rta geometrik qiymatlarini aniqlang.

Algoritm	C++ da dasturi:
 <pre> graph TD Start([boshlash]) --> Input[/a,b/] Input --> Process1[s=(a+b)/2] Process1 --> Process2[p=sqrt(a*b)] Process2 --> Output[/s,p/] Output --> End([tamom]) </pre>	<pre> #include <math.h> #include<iostream.h> Int main() { float a,b,s,p; cout<<"a=";>>a; cout<<"b=";>>b; s=(a+b)/2; p=sqrt(a*b); cout <<"s=" <<s; cout<<"p=" <<p; system("PAUSE"); return 0; } </pre>

3-masala. Tomonlari A va B ga teng to'g'ri to'rtburchakning yuzi va perimetri hisoblansin.

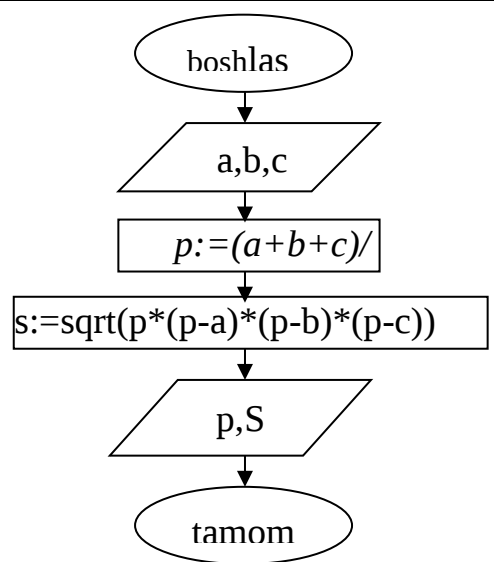
Algoritm	C++ da dasturi
 <pre> graph TD Start([boshlash]) --> Input[/a,b/] Input --> Process1[s=a+b] Process1 --> Process2[p=2*(a+b)] Process2 --> Output[/s,p/] Output --> End([tamom]) </pre>	<pre> #include<iostream.h> int main() { float a,b,s,p; cout<<"a=";>>a; cout<<"b=";>>b; s=a+b; p=2*(a+b); cout <<"s=" <<s; cout<<"p=" <<p; system("PAUSE"); return 0; } </pre>

4-masala. Berilgan radius bo'yicha aylana uzunligi va u bilan chegaralangan doira yuzini toping.

Algoritm	C++ da dasturi
 <pre> graph TD Start([hoshlas]) --> Input[/R/] Input --> L["L:=2*Pi*R"] L --> S["S:=Pi*R*R"] S --> Output[/S,L/] Output --> End([tamom]) </pre>	<pre> #include <iostream> #include <math.h> using namespace std; int main() { float r,Pi, s, l; cout << "Aylana radiusini ki- riting= "; cin >> r; l=2*Pi*r; s=Pi*r*r; cout << "aylana uzunligi= " << l<<endl; cout<<" doira yuzasi= "<<s; return 0; } </pre>

5-masala. a,b,c tomonlari bilan berilgan uchburchakni yuzasini hisoblash algoritmi va dasturini tuzing.

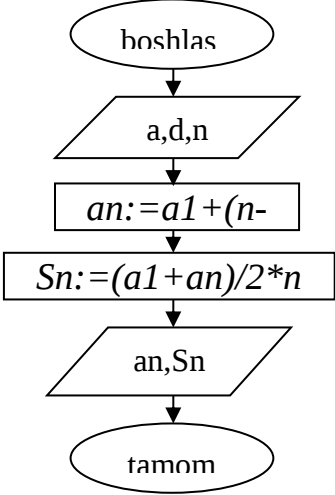
Bu masalani yechish uchun Geron formulasidan foydalanamiz. Uning dasturi quyidagicha bo'ladi:

Algoritm	C++ da dasturi
 <pre> graph TD Start([hoshlas]) --> Input[/a,b,c/] Input --> p["p:=(a+b+c)/2"] p --> s["s:=sqrt(p*(p-a)*(p-b)*(p-c))"] s --> Output[/p,S/] Output --> End([tamom]) </pre>	<pre> #include <iostream> #include <math.h> using namespace std; int main() { float a,b,c,p,s; cout << "Uchburchak tomonlarini kiriting (a+b)>c,(c+a)>b,(c+b)>a bo'lsin= "; cin >> a>>b>>c; p=(a+b+c)/2; s=sqrt(p*(p-a)*(p-b)*(p-c)); cout << "Uchburchakning peri- metri= " << p<<endl; cout<<" Ucburchakning yuzasi= "<<s<<endl; system("pause"); return 0; } </pre>

6-masala. Arifmetik progressiyaning birinchi hadi va ayirmasi berilgan bo'lsa uning n -hadi va n ta hadlari yig'indisini hisoblash algoritmi va dasturini tuzing.

Bu yerda arifmetik progressiyaning birinchi hadi a_1 , ayirmasi d va hadlar soni n berilgan bo'lsin.

U holda uni yechish algoritmi va dasturi quyidagicha bo'ladi:

Algoritm	C++ da dasturi
 <pre> graph TD Start([hoshlas]) --> Input[/a,d,n/] Input --> Process1[an:=a1+(n-1)*d] Process1 --> Process2[Sn:=(a1+an)/2*n] Process2 --> Output[/an,Sn/] Output --> End([tamom]) </pre>	<pre> #include <iostream> #include <math.h> using namespace std; int main() { float a1,d,xn,sn; int n; cout<<"arifmetik progressi- yaning 1-hadini kiritng= "; cin>>a1; cout<<"arifmetik progressiyaning ortirmasini kiriting= "; cin>>d; cout<<"arifmetik progressiyaning hadlar sonini kiriting= "; cin>>n; xn= a1+(n-1)*d; sn=(a1+xn)/2*n; cout<<"arifmetik progressiyaning "<<n<<" hadi= "<<xn; cout<<"\narifmetik progressiyaning "<<n<<" ta hadi yig'indisi= "<<sn; system("pause"); return 0; } </pre>

7-masala. Koordinatalari X_1, Y_1 va X_2, Y_2 ga teng bo'lgan nuqtalari orasidagi masofani hisoblang.

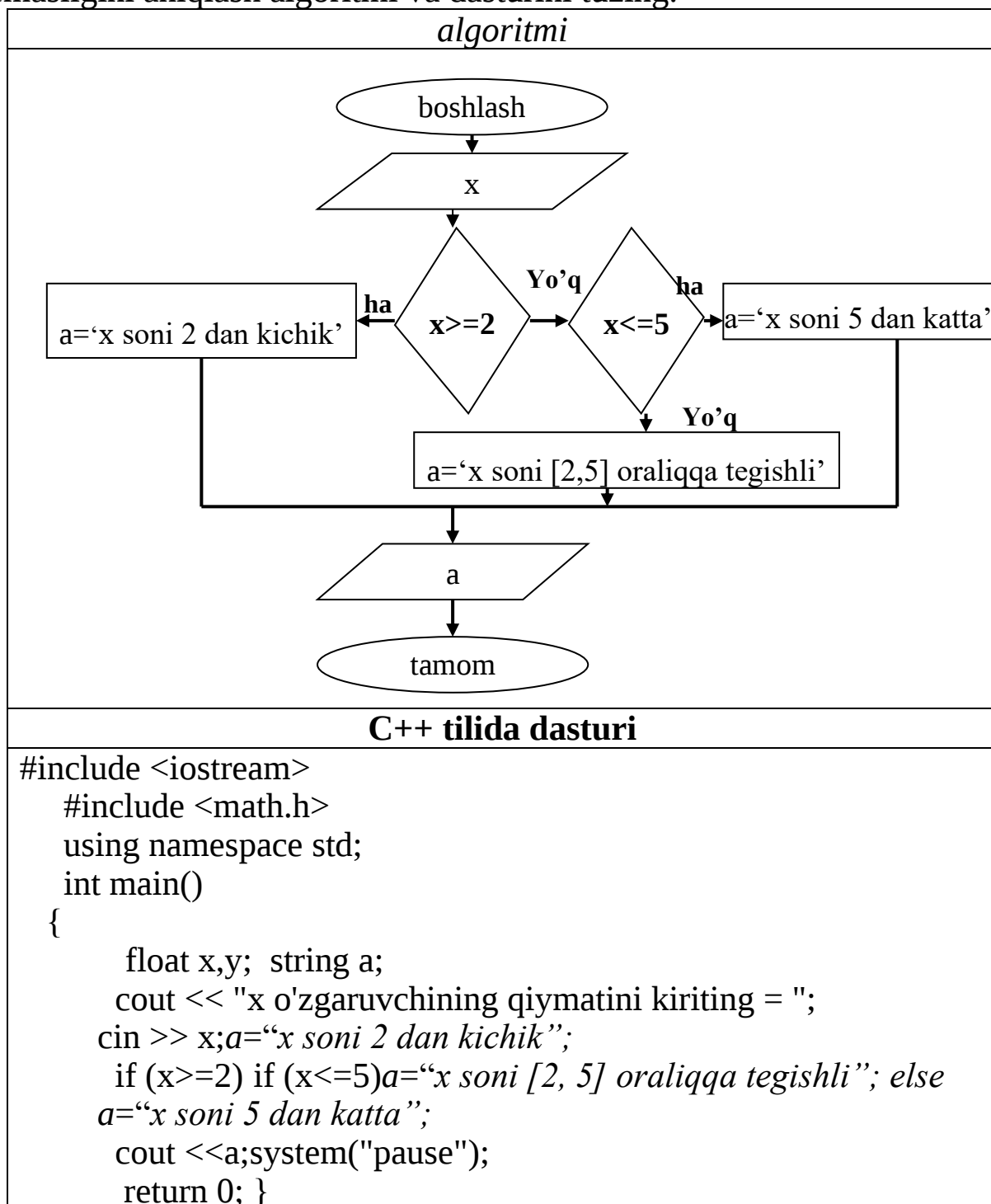
Algoritm	C++ da dasturi
<pre> graph TD A([boshlash]) --> B[/x1,x2,y1,y2/] B --> C[d:=sqrt(sqr(x2-x1)+sqr(y2-y1))] C --> D[/an,Sn/] D --> E([tamom]) </pre>	<pre> #include <math.h> #include <iostream> int main() { float x1,x2,y1,y2,d; cout<<"x1="; cin>>x1; cout<<"x2="; cin>>x2; cout<<"y1="; cin>>y1; cout<<"y2="; cin>>y2; d=sqrt(sqr(x2-x1)+sqr(y2-y1)); cout <<"d=" <<d; system("PAUSE"); return 0; } </pre>

8-masala. Birinchi hadi B, maxraji Q va hadlari soni N ga teng geometrik progressiyaning hadlarining yig'indisini hisoblang.

Algoritm	C++ da dasturi
<pre> graph TD A([boshlash]) --> B[/b,q,n/] B --> C[s:=(b*(1-exp(n*log(q))))/(1-q);] C --> D[/S/] D --> E([tamom]) </pre>	<pre> #include <math.h> #include <iostream> int main() { float B,q,n,S; cout<<"B=";cin>>B; cout<<"q=";cin>>q; cout<<"n=";cin>>n; S=(B*(1-exp(N*log(q))))/(1-q); cout <<"S=" <<S; system("PAUSE"); return 0; } </pre>

3.2. Matematik masalalarga tarmoqlanuvchi algoritmi va dastur tuzish

1-masala. Berilgan x sonini $[2,5]$ oraliqqa tegishli yoki tegishli emasligini aniqlash algoritmi va dasturini tuzing.



Bu dastur uchun kompyuter quydagi natijalarni beradi:

x ning qiymatini kiriting 4

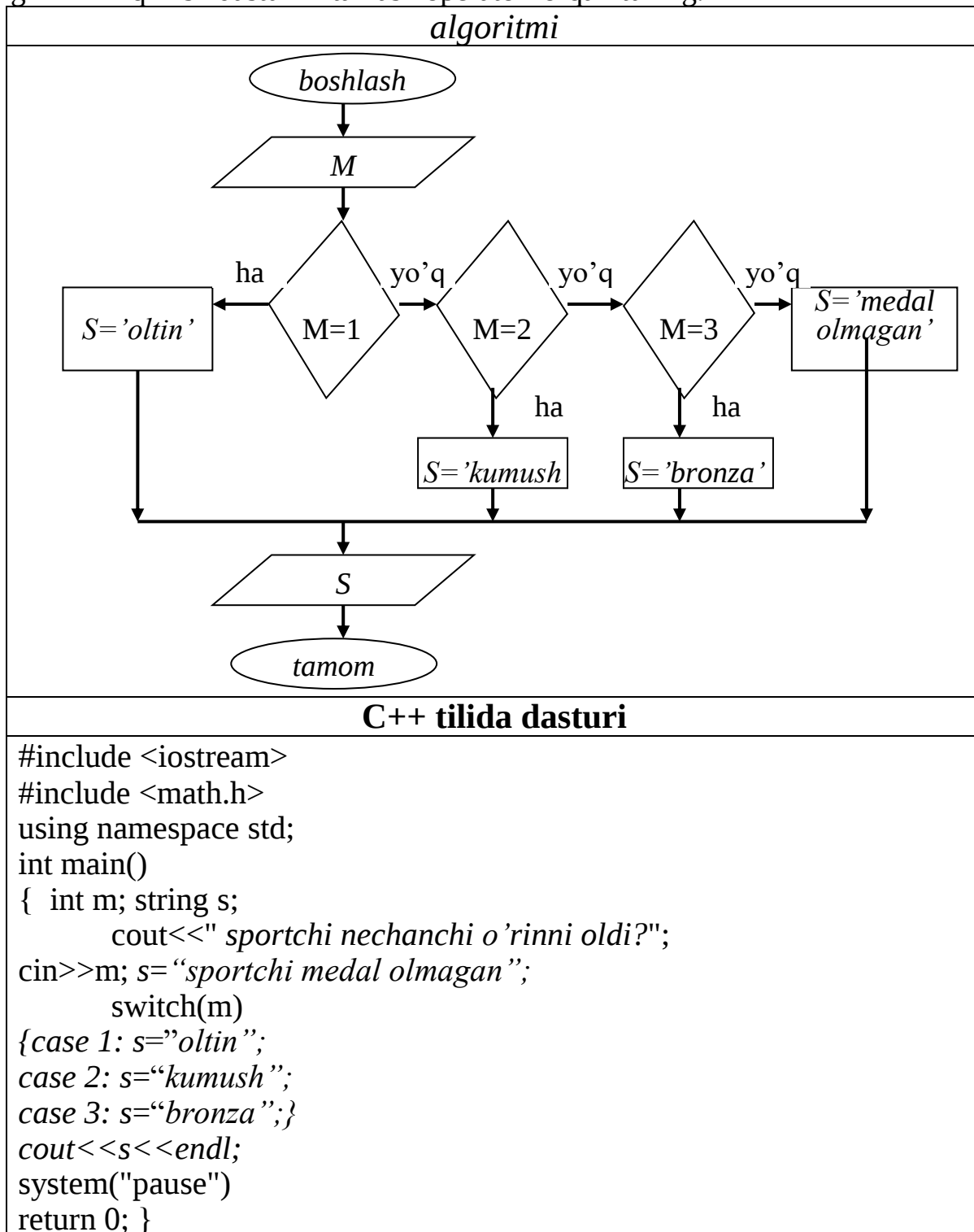
x soni [2, 5] oraliqqa tegishli

x ning qiymatini kiriting 7

x soni 5 dan katta

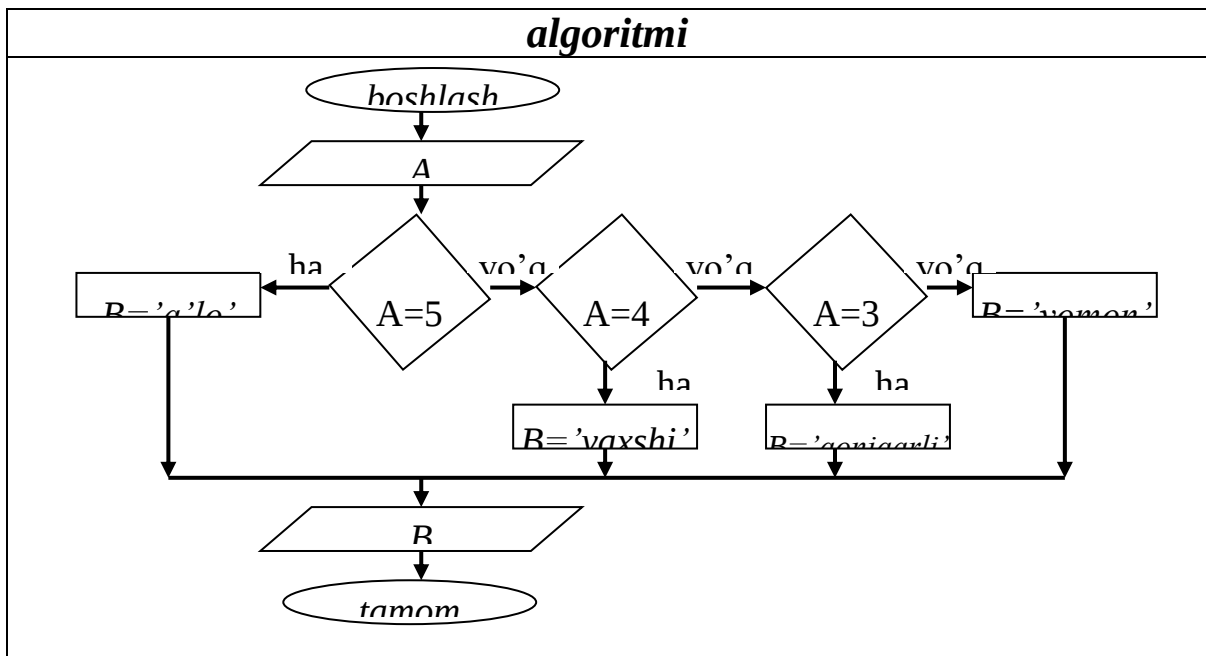
*x ning qiyatini kiriting 1
x soni 2 dan kichik*

2-masala. Sportchi musobaqada M – o‘rinni egalladi. U qanday medal olganini aniqlash dasturini tanlash operatori orqali tuzing.



Komputer bu dastur uchun quyidagi natiani beradi:
sportchi nechanchi urinn oldi? 5 sportchi medal olmagan
sportchi nechanchi urinn oldi?1 oltin

3-masala. O‘quvchilarni olgan baholarini ifodalash dasturini tuzing.



C++ tilida dastur

```

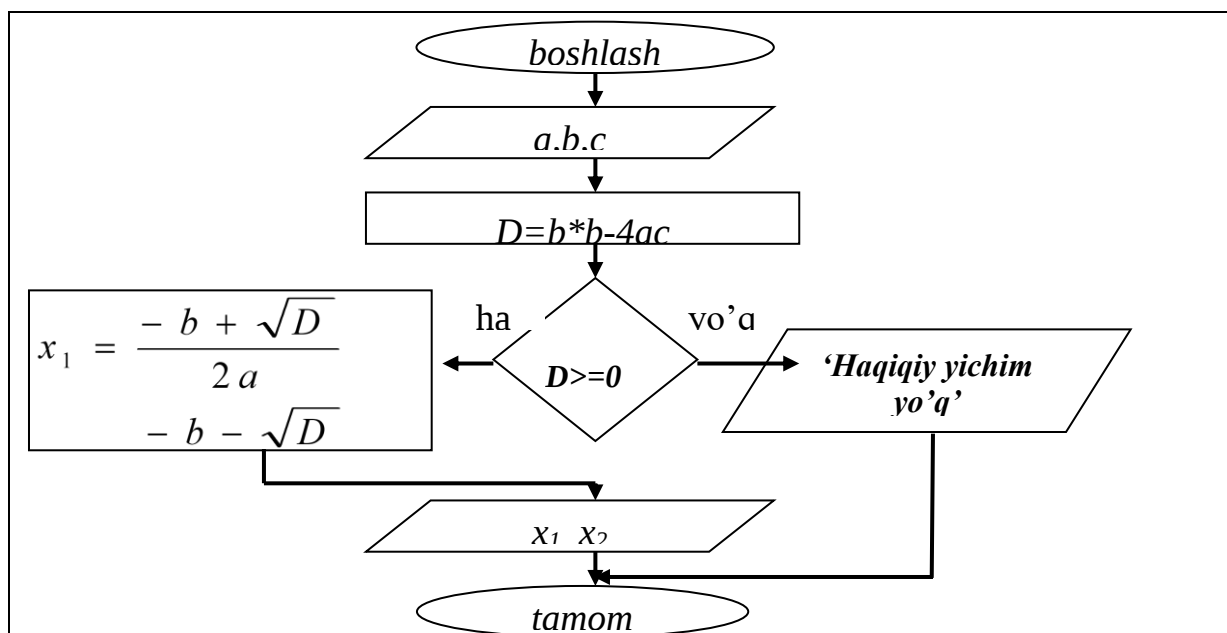
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    int a;
    cout<<" bahoni kiritng ";
    cin>>a;
    switch(a)
    {
        case 1:cout<<"juda yomon"<<endl;break;
        case 2: cout<<"yomon"<<endl; break;
        case 3:cout<<"qoniqarli"<<endl;break;
        case 4: cout<<"yaxshi"<<endl; break;
        case 5:cout<<"a'lo"<<endl;break;
        default: cout<<"bunday baho yo'q";
    } system("pause");
    return 0;
}
  
```

Komputer bu dastur uchun quyidagi natijalarni beradi:

Bahoni kiritng 4 yaxshi

Bahoni kiritng 1 yomon

4-masala. $ax^2 + bx + c = 0$ ($a \neq 0, b \neq 0, c \neq 0$) kvadrat tenglamani yechish algoritmi va dasturini tuzing.

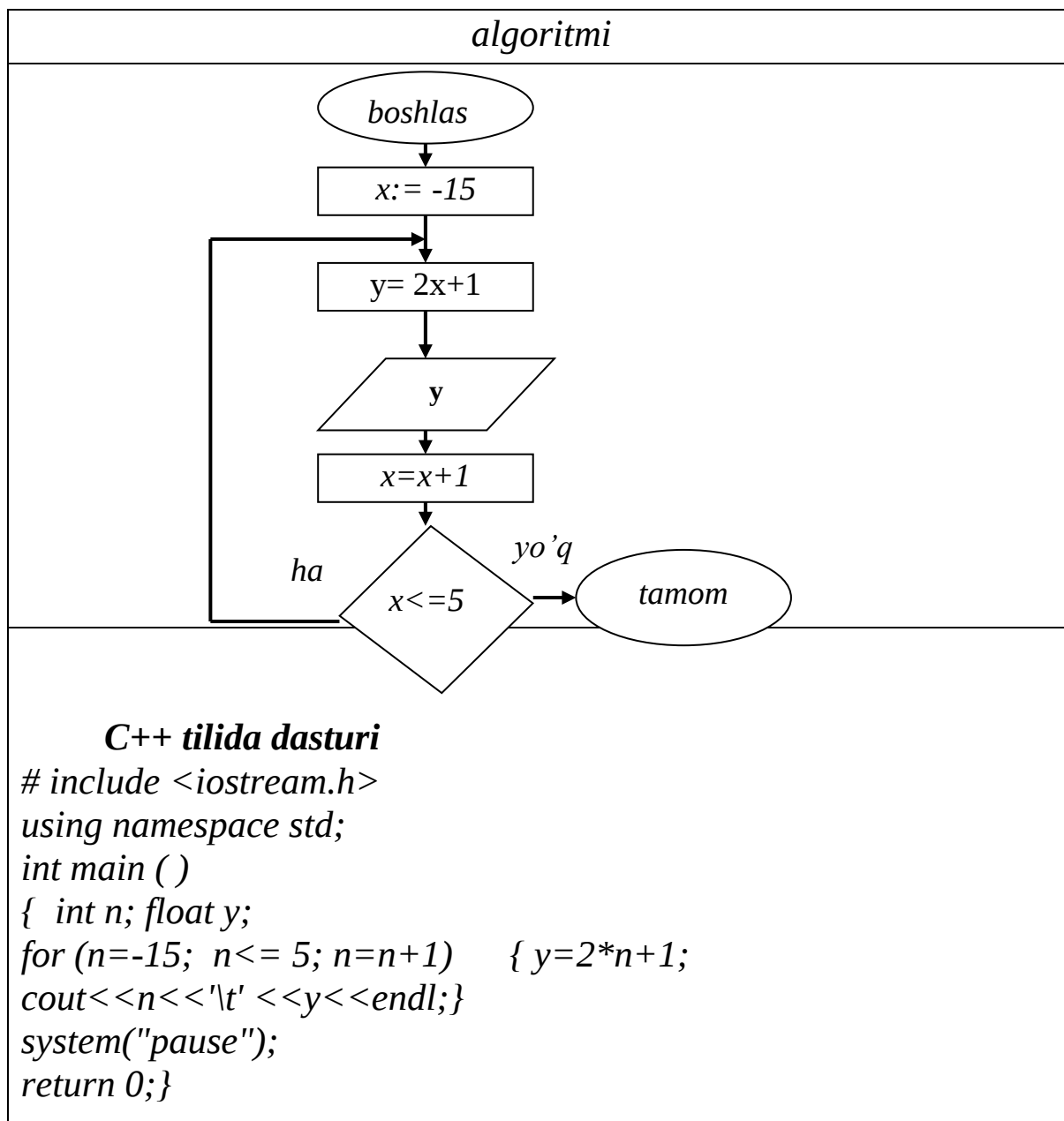


C++ tilida dastur

```

#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    float a,b,c,d,x1,x2;
    b:cout << "Kvadrat tenglama koefitsientlarini kiritng
a<>0,b<>0,c<>0 bo'lsin= ";
    cin >> a>>b>>c;
    if ((a==0) or (b==0) or (c==0)) goto b;
    d=b*b-4*a*c;
    if (d<0)
    {
        cout<<"Haqiqiy yechim mavjud emas"<<endl;
        return 0;
    }
    if (d>0) {x1=(-b+sqrt(d))/(2*a); x2=(-b-sqrt(d))/(2*a);
    cout << "Ikkita yechim mavjud: x1= " << x1<<" x2=
"<<x2<<endl; } else
    cout<< "Karrali yechim mavjud: x1=x2="<<x1<<endl;
    system ("pause");
    return 0;
}
  
```

2-masala. $y=2x+1$ funksiyaning qiymatini x ning $[-15,5]$ oraliqdagi butun qiymatlarida hisoblash algoritmi va dasturni tuzing.



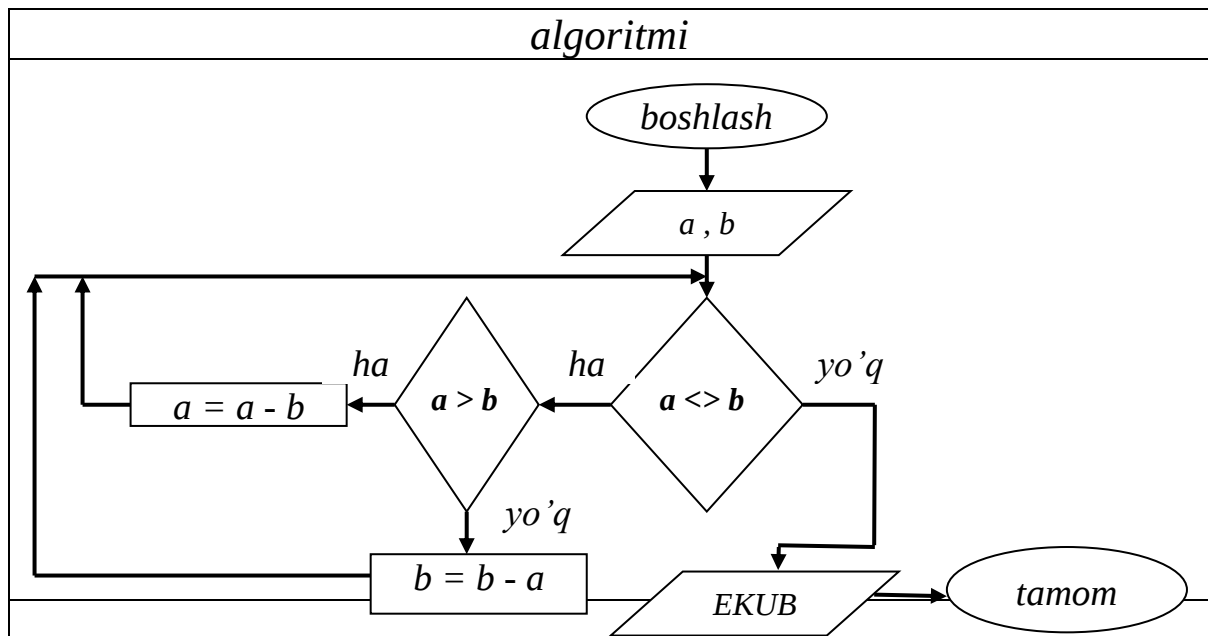
Shart bo'yicha takrorlash operatorlari yordamida.

1-masala. Shart bo'yicha sikl operatoridan foydalanib quyidagi ifodaning qiymatini ϵ s aniqlik bilan hisoblash algoritmini va dasturini tuzing: $P = \prod_{i=1}^{\infty} \frac{1}{i^2 + 1}$, bu yerda ϵ s –kichik haqiqiy son.

Bunda hisoblash jarayoni ko'paytma qiymati berilgan haqiqiy son ϵ s dan kichik yoki unga teng bo'lguncha davom etadi, yani $\epsilon s \leq 1/(i*i+1)$ yoki $\epsilon s \geq 1/(i*i+1)$ shartlari tekshiriladi.

<i>algoritm</i>	<i>Dastur (while yordamida)</i>
<pre> graph TD A[boshlash] --> B[/EPS/] B --> C[P:=1; i:=1] C --> D[P:=P*(1/(i^2+1))] D --> E[i:=i+1] E --> F{1/(i^2+1)>=eps} F -- ha --> D F -- yo'q --> G[/P ni chiqarish/] G --> H[tamom] </pre>	<pre> #include <iostream.h> using namespace std; int main () { int n,i; float s,eps; s=0; i=1; cout<<"Ixtiyoriy juda kichik son kiritng"; cin>>eps; while (1./(i*i+1)>eps) { s=s+1./(i*i+1);i=i+1;} cout << "yig'indi =" <<s<<endl; system("pause"); return 0; } </pre>
	<i>Dastur (do while yordamida)</i> <pre> #include <iostream.h> using namespace std; int main () { int n,i; float s,eps; s=0; i=1; cout<<"Ixtiyoriy juda kichik son kiriting"; cin>>eps; do { s=s+1./(i*i+1);i=i+1;} while (1./(i*i+1)>eps); cout << "yig'indi =" <<s<<endl; system("pause"); return 0; } </pre>

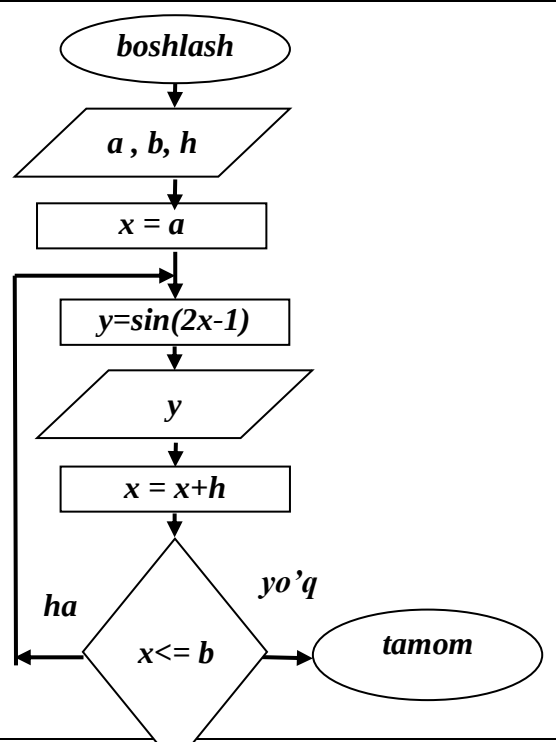
2-masala. A va B natural sonlar berilgan bo'lsin. Ularning eng katta umumiy bo'luvchisini toping. Bu masalani yechish "Yevklid algoritmi" asosida amalga oshiriladi. Bu algoritmgga ko'ra toki A va B sonlari bir-biriga teng bo'lib qolmaguncha, kattasidan kichigi ayirilaveradi. Har gal ayirma shu sonlardan kattasini ifodalab turgan o'zgaruvchi nomi bilan belgilab qo'yiladi. Bu jarayon $A = B$ bo'lganda to'xtaydi. Yechim sifatida A yoki B ni olish mumkin.



C++ tilida dastur

Dastur (while yordamida)	Dastur (do while yordamida)
<pre> # include <iostream.h> using namespace std; int main() { int a,b; cout<<"Ikkita butun son ki- ritng"; cin>>a>>b; while (a!=b) {if (a>b)a=a-b; else b=b-a;} cout <<a<<endl; system("pause"); return 0; } </pre>	<pre> # include <iostream.h> using namespace std; int main() { int a,b; cout<<"Ikkita butun son kiritng"; cin>>a>>b; do {if (a>b)a=a-b; else b=b-a;} while(a!=b); cout <<a<<endl; system("pause"); return 0; } </pre>

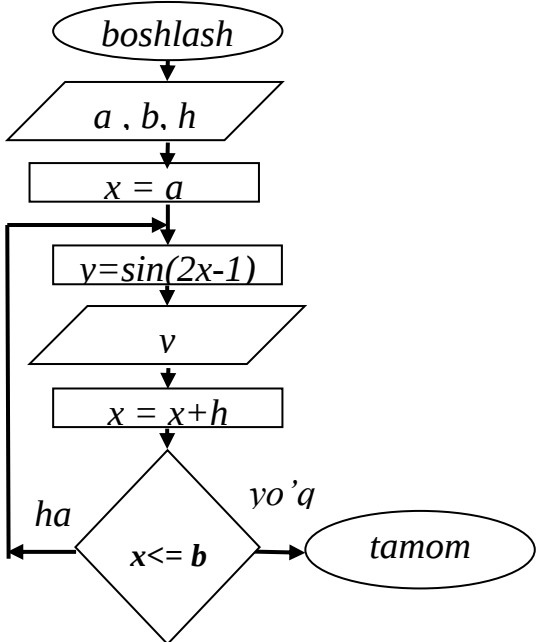
3-masala. $y = \sin(2x-1)$ funksiyani $[a,b]$ oraliqda h qadam bilan qiymatlarini hisoblash algoritmi va dasturini tuzing.

algoritmi	
 <pre> graph TD Start([boshlash]) --> Input[/a, b, h/] Input --> AssignX[x = a] AssignX --> CalcY[y = sin(2x-1)] CalcY --> OutputY[/y/] OutputY --> AssignXplusH[x = x + h] AssignXplusH --> Decision{x <= b} Decision -- ha --> CalcY Decision -- yo'q --> End([tamom]) </pre>	
C++ tilida dastur	
Dastur (while yordamida)	Dastur (do while yordamida)
<pre> #include <iostream.h> #include <math.h> using namespace std; int main() { float a,b,h,y,x; cout<<" oraliq qiymatni kiritng "; cin>>a>>b; cout<<" qadamni kiritng "; cin>>h; x=a; while (x<=b) { y=sin(2*x-1); cout<<x<<y; x=x+h; } system("pause"); return 0; } </pre>	<pre> #include <iostream.h> #include <math.h> using namespace std; int main() { float a,b,h,y,x; cout<<" oraliq qiymatni kiritng "; cin>>a>>b; cout<<" qadamni kiritng "; cin>>h; x=a; do { y=sin(2*x-1; cout<<x<<y; x=x+h; } while (x<=b); system("pause"); return 0; } </pre>

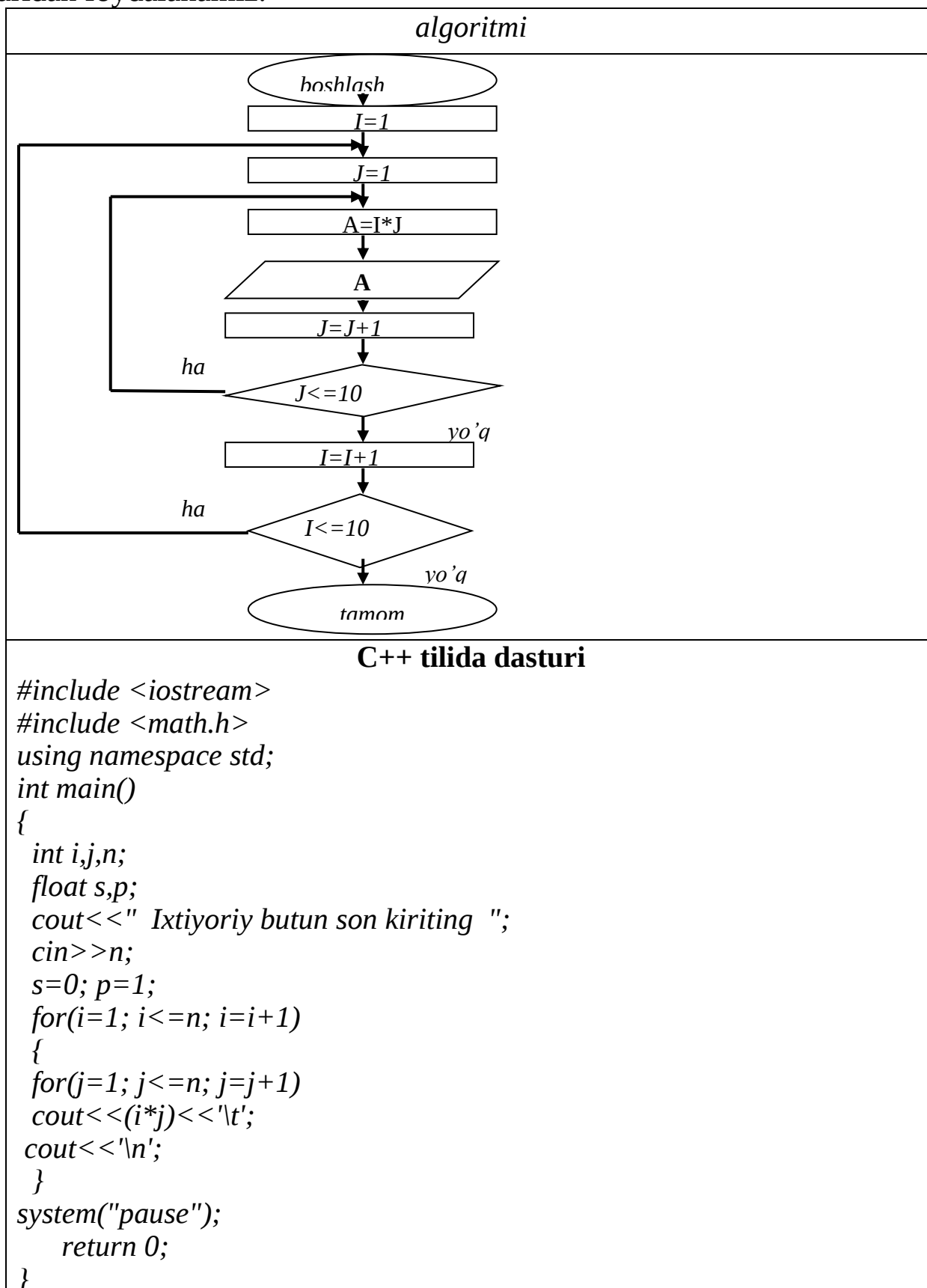
Ichma-ich joylashgan sikllar

1-masala. $S = \sum_{k=1}^{15} \sum_{n=1}^{10} (2a^k + x^n)$ ifodaning qiymatini hisoblash dasturini

tuzing.

algoritmi	
 <pre> graph TD Start([boshlash]) --> Input[/a, b, h/] Input --> Init[x = a] Init --> LoopStart(()) LoopStart --> Calc[y = sin(2x-1)] Calc --> Output[/v/] Output --> Inc[x = x + h] Inc --> Cond{x <= b} Cond -- ha --> LoopStart Cond -- yo'q --> End([tamom]) </pre>	
C++ tilida dastur	
Dastur (for yordamida)	Dastur (while yordamida)
<pre> #include <iostream.h> #include <math.h> using namespace std; int main () { int n,k; float a,x, s,s1; cout<<"a va x ga ixtiyoriy son kiritng"; cin>>a>>x; s=0; for (k=1; k<= 15; k=k+1) { s1=0; for (n=1; n<= 15; n=n+1) s=s+ 2*pow(a,k)+pow(x,n); s1=s1+s;} cout << "yig'indi =" <<s1; system("pause"); return 0; } </pre>	<pre> #include <iostream> #include <math.h> using namespace std; int main() { int n,k; float s,p,s1,a,x; cout<<" a va x ga ixtiyoriy qiymat bering"; cin>>a>>x; s=0; k=0; while(k=k+1, k<=15) { n=0;s1=0; while(n=n+1, n<=15) s1=s1+ 2*pow(a,k)+pow(x,n); s=s+s1; } cout<<" yig'indi="<<s; system("pause"); return 0; } </pre>

2-masala. Ko'paytirish jadvali elementlarini hosil qilish algoritmi va dasturini tuzing. Buning uchun ichma-ich joylashgan sikl operatorlaridan foydalanamiz.



3-masala. $S = \sum_{k=1}^{15} \sum_{n=1}^{10} (i! + j!)$ ifodani hisoblash dasturini tuzing.

C++ tilida dastur

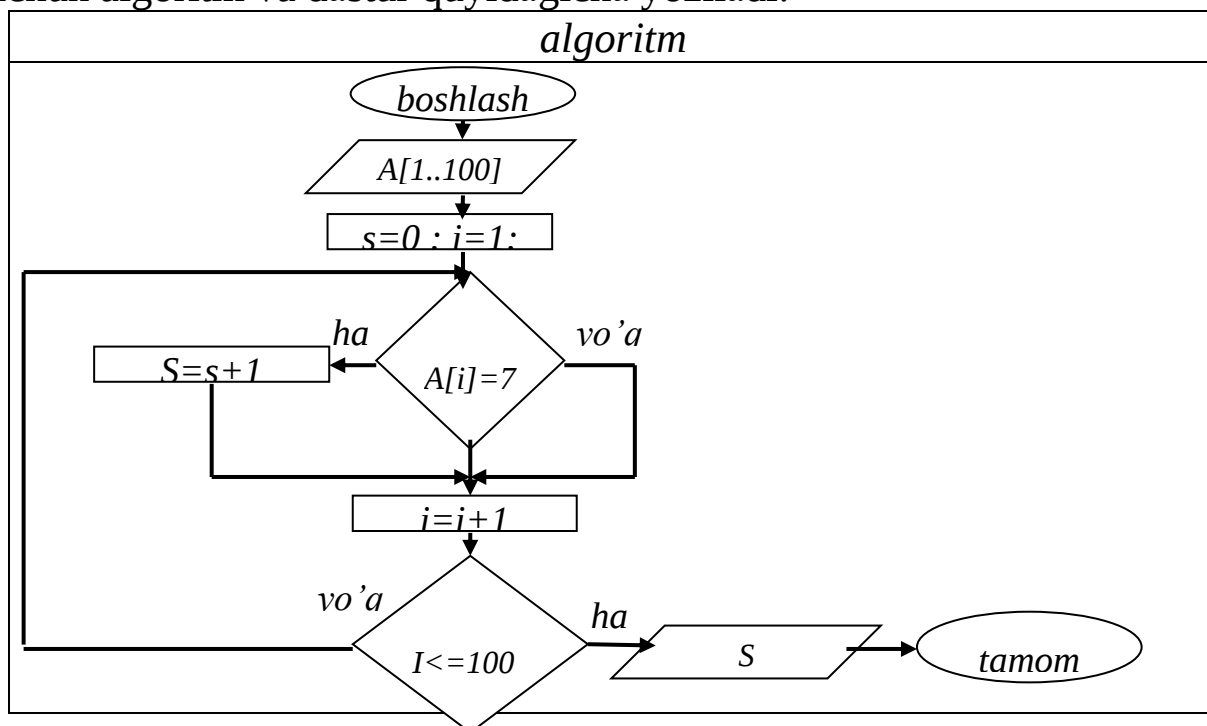
<pre>#include <iostream.h> #include <math.h> using namespace std; int main () { int n,k; float a,x, s,s1,f,f1; cout<<"a va x ga ixtiyoriy son kiritng"; cin>>a>>x;</pre>	<pre>s=0;f=1; for (k=1; k<= 15; k=k+1) { s1=0;f=f*k;f1=1; for (n=1; n<= 10; n=n+1) {f1=f1*n; s=s+ f+f1;} s1=s1+s;} cout << "yig'indi =" <<s1; system("pause"); return 0;}</pre>
---	---

3.3. Massiv elementlari bilan ishlash algoritmi va dasturini tuzish

Bir o'lchovli massiv elementlari bilan ishlash.

1-masala. $A[1 \dots 100]$ butun sonli massivda qiymati 7 ga teng elementlar sonini aniqlash algoritmi va dasturini tuzing.

Dastlab 7 ga teng bo'lgan elementlarni sanash uchun $s := 0$ o'zgaruvchi tanlanadi. So'ngra, massiv elementlarini bittadan kiritiladi. Har bir kiritilgan element 7 soni bilan taqqoslanadi. Agar $a[i]$ element 7 ga teng bo'lsa, s ning qiymati 1 ga oshiriladi. Keyin navbatdagi elementga o'tiladi. Bu jarayon 100 marta takrorlanadi. Qo'yilgan masala uchun algoritm va dastur quyidagicha yoziladi:



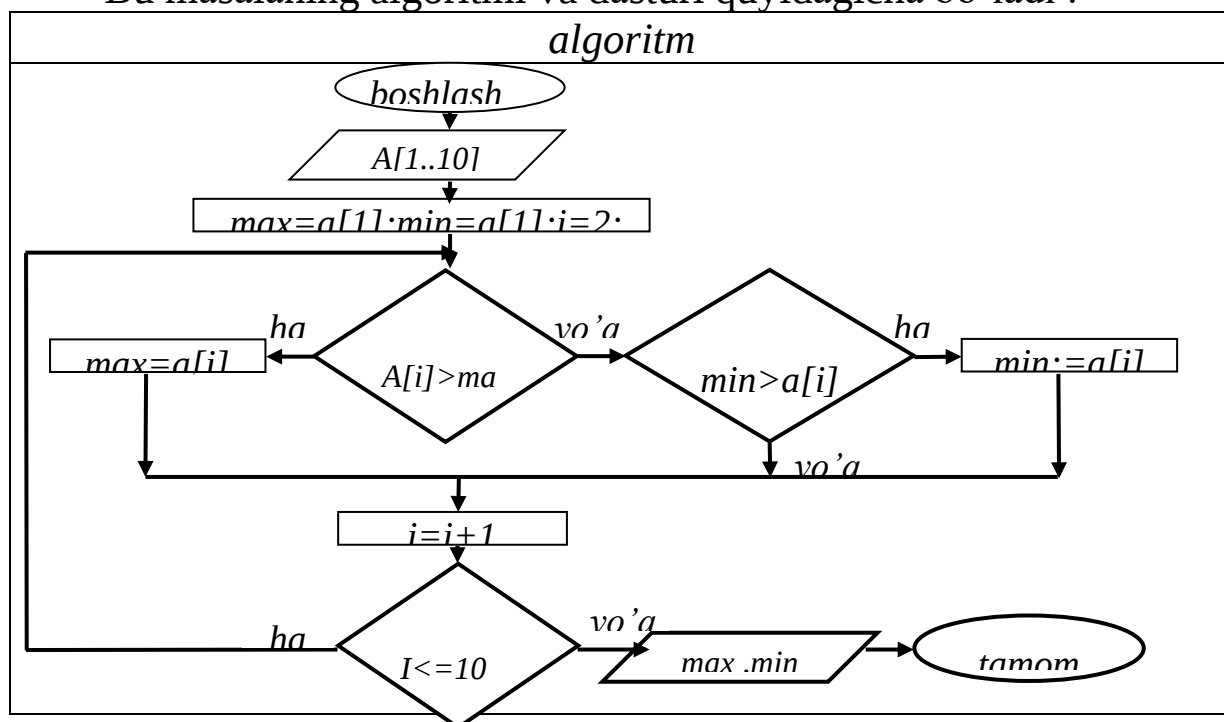
C++ tilida dastur

```
#include <iostream.h>
#include <conio.h>
int main()
{int a[100]={9,0,7,7,5,7,4,5,7,3,2,6,7,8,}, n, i;
  float s,p;
  cout<<"massiv elementlari soni-";
  cin>>n; s=0;
  for (i=0;i<=n-1;i++)
  if (a[i]==7) s=s+1;
  cout << "7 ga teng elementlar soni = "<<s;
  getch();
  system("pause");
  return 0;
}
```

2-masala. 10 ta haqiqiy elementli massivdagi eng katta va eng kichik elementlarni topish algoritmi va dasturini kiritng.

Bu masalani yechish uchun avval massivning barcha elementlari aniqlab olinadi. So'ngra, birinchi elementni izlangan element deb faraz qilinadi va qolgan elementlar uchun qilingan farazni to'g'ri, noto'g'riligi tekshiriladi. Agap eng katta (eng kichik) degan elementdan ham kattaroq (kichikroq) element topilib qolsa, izlangan element sifatida u qabul qilinadi va tekshirishni kelgan yeridan ana shu element uchun davom ettiriladi.

Bu masalaning algoritmi va dasturi quyidagicha bo'ladi :



C++ tilida dastur

```
#include <iostream.h>
#include <conio.h>
int main()
{int a[10], i; int max,min;float s,p;
for (i=0;i<=9; i=i+1)
cin>>a[i];  max=a[0]; min=a[0];
    for (i=0;i<=9;i++)
    { if (a[i]>max) max=a[i];
if (a[i]<min) min=a[i];
    }
cout << "  max="<<max<<"\t"<< "min="<<min;
    getch();
return 0;
}
```

Bu dastur uchun avval kompyuterga massiv elementlari kiritiladi.

- 1 – element 36
- 2 - element 22
- 3 - element 27
- 4 - element 25
- 5 - element 29
- 6 - element 21
- 7 - element 32
- 8 - element 11
- 9 - element 24
- 10 - element 34

Bu ma'lumotlar uchun kompyuter bergan natija qo'yidagicha bo'ladi:

kiritilgan elementlar ichida

eng katta element 36

eng kichik element 11

3-masala. Biror bir massiv kiritilsin va shu massivning minimal va maksimal qiymatlari orasidagi elementlar ekranga chiqarilsin.

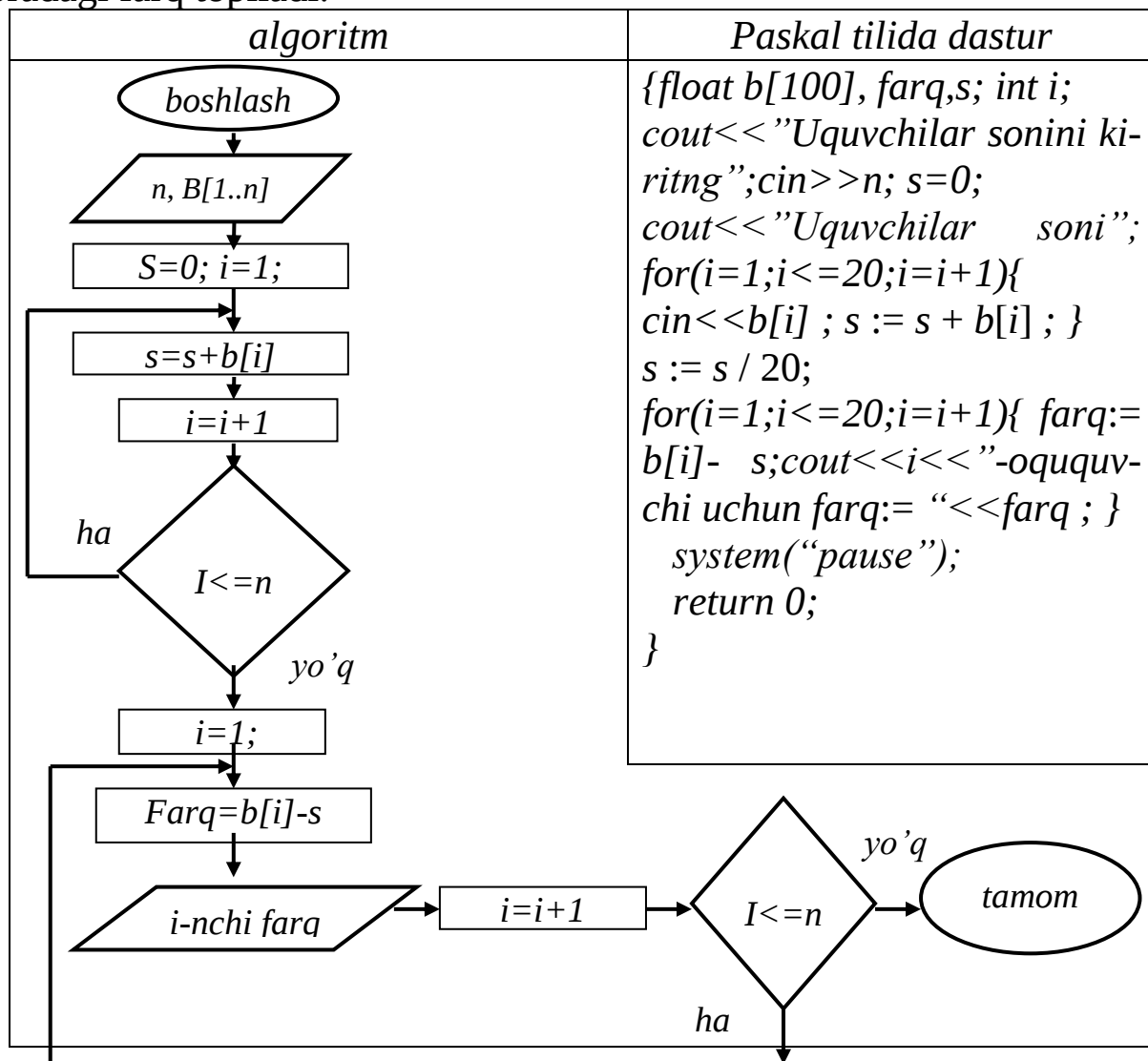
Masalan: 2,3,1,3,4,5,6,7,4,3,5, Javob: 1,3,4,5,6,7 || -1,2,3,7,2,1, -3,-4, 2,1 , Javob: 7,2,1,-3,-4

C++ tilida dastur

```
#include<iostream.h>
#include<conio.h>
using namespace std;
int main()
{
    double a[50],max,min; int i,n,b,t;
    cout<<"Massiv o`lchami kiritilsin:\nn=";   cin>>n;
    for (i=1;i<=n;i++)
    {
        cout<<"a["<<i<<"]=";   cin>>a[i];
    }
    min=a[1]; max=a[1];
    for (i=1;i<=n;i++)
    {
        if(min>a[i])
        {
            min=a[i];  b=i;
        }
        if(max<=a[i])
        {
            t=i;  max=a[i];
        }
    }
    if (t<b)
    for(i=t;i<=b;i++)
    cout<<a[i]<<",";
    else
    for(i=b;i<=t;i++)
    cout<<a[i]<<",";
    getch();  }
```


5-masala. Sinfdagi N o'quvchining informatika fani bo'yicha imtihondan olgan baholari berilgan bo'lsin. Har bir o'quvchining bahosi o'rtacha o'zlashtirish bahosidan qanchaga farq qilishini aniqlash algoritmi va dasturini tuzing.

Bu masalani yechish uchun har bir o'quvchi olgan baholarni kiritib, umumiy baholar yig'indisi topiladi. Umumiy yig'indini o'quvchilar soni N ga bo'lib, o'rtacha o'zlashtirish aniqlanadi. Keyin har bir o'quvchining bahosidan o'rtacha o'zlashtirish bahosini ayirib, oradagi farq topiladi.



6-masala. $A[1..N]$ massiv elementlarini o'sish tartibida joylashtirish algoritmi va dasturini tuzing.

Massivlar haqidagi masalalar ichida eng ko'p uchraydigani bu uning elementlarini o'sish yoki kamayish tartibida tartiblash masalasidir. Uni yechish usullari ko'p bo'lib, ulardan birortasini boshqasidan ustun

qo'yib bo'lmaydi. Har bir usul elementlarning joylashishiga ko'ra boshqasidan yaxshi bo'lishi mumkin.

1-usul. Berilgan massiv elementlari ichidan eng kichigini topib, uning o'rnini 1- element bilan almashtiriladi. Demak, 1-element tartiblandi. Endi 2-element tartiblanadi. Buning uchun qolgan elementlar ichidan eng kichigi topilib, uning o'rnini 2-element bilan almashtiriladi va hokazo.

Bu jarayon $N-1$ marta takrorlanganda massiv elementlarini tartiblash tugaydi. Bu usul uchun dastur quyidagicha yoziladi:

```
program tartiblash ;
```

```
const n = 100 ;
```

```
var c, min : real ; i, j, k : integer;
```

```
a : array[1..n] of real;
```

```
begin writeln(' massiv elementlarini kiritng');
```

```
for i:=1 to n do readln (a[i]);
```

```
  for i:=1 to n do begin min:= a[i] ; k:= i;
```

```
    for j := i +1 to n do
```

```
      if min>a[j] then begin (*eng kichik element va uning tartib raqami aniqlanmoqda*) min:= a[j]; k:= j; end s:= a[i];
```

```
(*eng kichik element bilan tartiblanayotgan element o'rinlari almashtirilib, ekranga chiqirilmoqda.*) a[i]:= a[k]; a[k]:= s ;
```

```
writeln(a[i]:8:3) end;
```

```
end.
```

Bu usulni eng kichik elementni chiqarish usuli deyiladi.

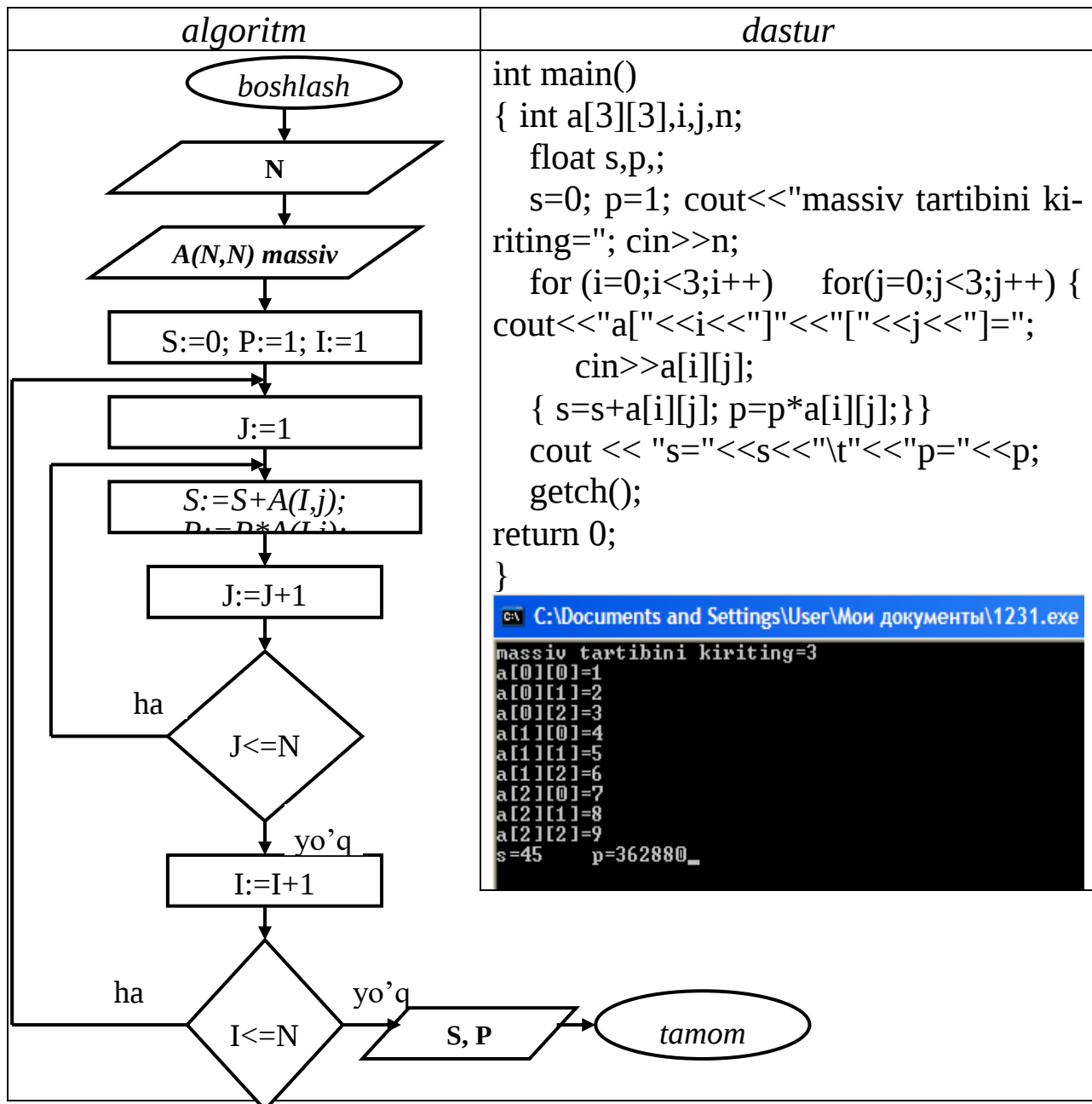
Ikki o'lchovli massiv elementlari bilan ishlash.

1-masala. Haqiqiy sonlardan iborat $A(n,n)$ o'lchovli massiv berilgan. Uning barcha elementlari yigindisi va ko'paytmasini hisoblash algoritmi va dasturini tuzing.

Bu masalani yechish uchun avval N soni kiritiladi. Keyin boshlang'ich qiymati 0 ga teng bo'lgan S va 1 ga teng bo'lgan P o'zgaruvchi e'lon qilinadi.

So'ngra ichma-ich joylashgan sikllar yordamida massivning navbatdagi elementi kiritiladi va uning qiymatini S ga qo'shib, P ga ko'paytiramiz.

Sikllar tugagandan so'ng S va P ning qiymatini chiqaramiz.



2-masala. $A(n,n)$ massivning ustun elementlari yig'indisi va ko'paytmasini hosil qiling va ularni chop eting.

C++ tilida dastur

```

int main()
{ int a[3][3]={8,-9,6,-5,-3,12,-13,16,17},i,j;
  float s,p;;
  s=0; p=1;
  for ( i=0; i<3; i++)
  { for ( j=0; j<3; j++)
  cout<<"\t"<<a[i][j]<<" ";cout<<"\n";}
  for (j=0;j<3;j++)
  { for (i=0;i<3;i++)
          
```

```

{ s=s+a[i][j]; p=p*a[i][j];}
    cout<<"\n" <<"\n" <<"\t" <<j<<"-ustun elementlari yig'indisi va
ko'paytmasi" <<"\n" <<"\n";
cout << "\t" <<"s=" <<s<<"\t" <<"p=" <<p;}
    getch();
return 0;
}

```

```

8      -9      6
-5      -3      12
-13     16     17

0-ustun elementlari yig'indisi va ko'paytmasi
s=-10    p=520
1-ustun elementlari yig'indisi va ko'paytmasi
s=-6     p=224640
2-ustun elementlari yig'indisi va ko'paytmasi
s=29     p=2.74959e+08_

```

3-masala. $n \times n$ o'lchovli massiv shaxmat doskasi ko'rinishida bo'lsin, har bir katak 4 ta element o'rini egallagan bo'lishi kerak; qora kataklar bu 1 dan tashkil topgan elementlar, oq kataklar 0 (nol)dan tashkil topgan elementlar.

C++ tilida dastur
<pre> #include<iostream.h> #include<conio.h> int main() { int a[100][100], n,i,j,b,c; cout<<"Massiv ulchamlari:"; cout<<"n="; cin>>n; b=1;c=n; for(i=1; i<=n; i++) for(j=1; j<=n; j++) { a[i][j]=0; } for(i=1; i<=n; i+=4) </pre>

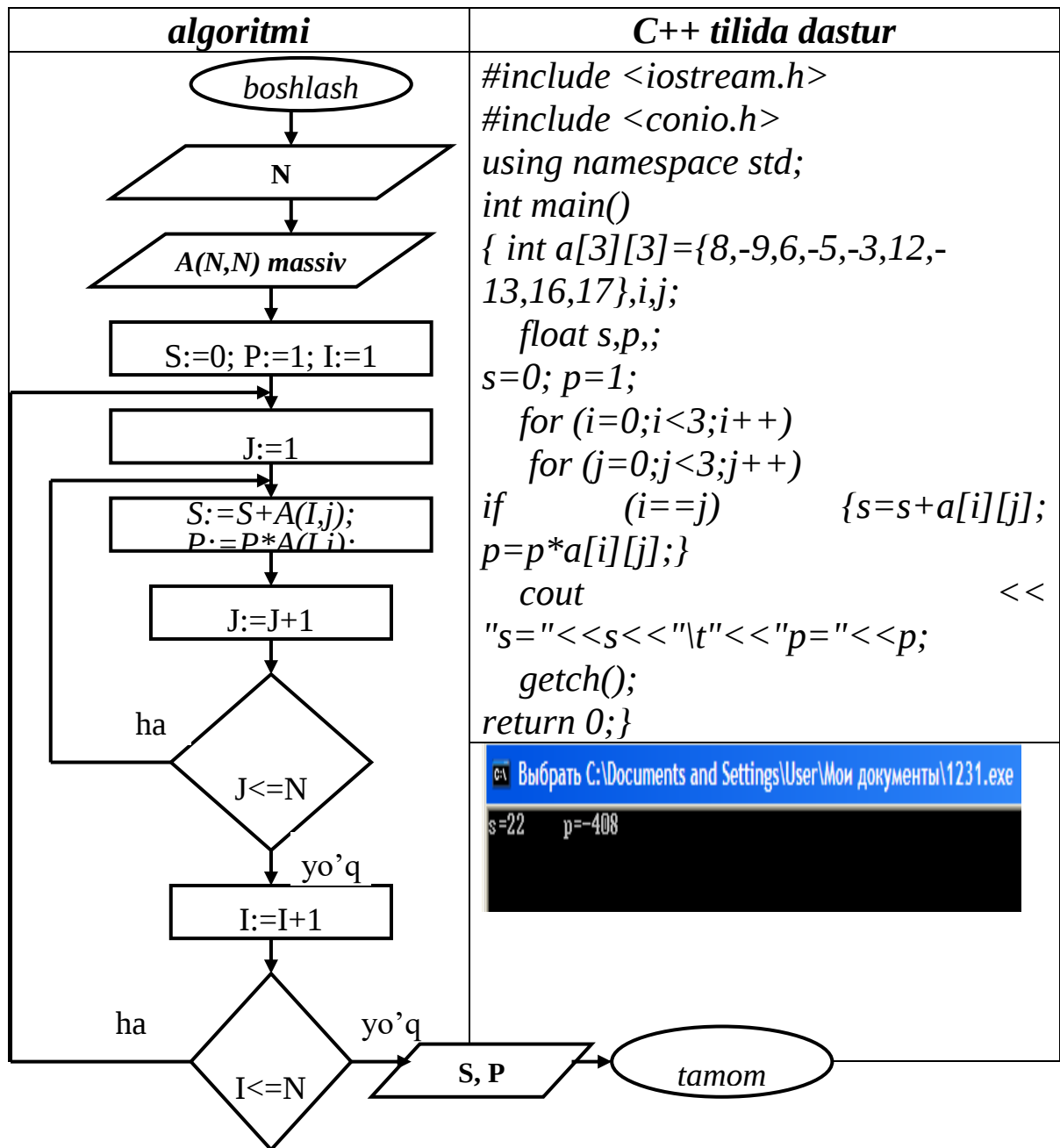
```

for(j=1; j<=n; j+=4)
{
    a[i][j]=1;
    a[i][j+1]=1;
    a[i+1][j]=1;
    a[i+1][j+1]=1;
}
for(i=3; i<=n; i+=4)
for(j=3; j<=n; j+=4)
{
    a[i][j]=1;
    a[i][j+1]=1;
    a[i+1][j]=1;
    a[i+1][j+1]=1;
}
for(i=1; i<=n; i++)
{
    for(j=1; j<=n; j++)
        cout<<a[i][j]<<" "; cout<<endl;    }
getch();    }

```

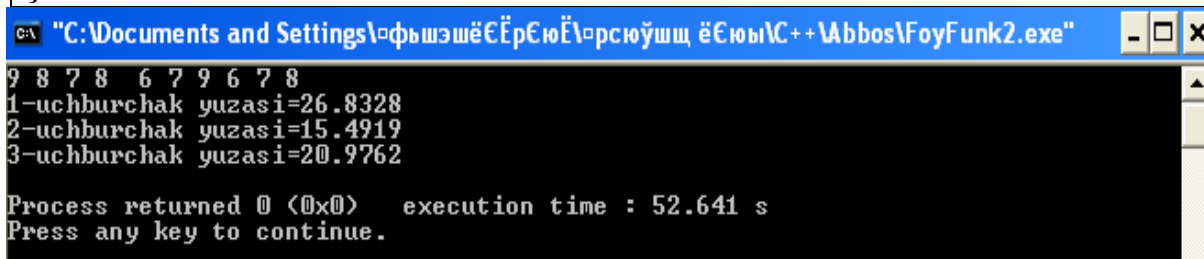
The image shows two screenshots of a C++ program's output. The left screenshot is titled "C:\Documents and Settings\User\Мои доку..." and shows the output for "Massiv ulchamlari: n=9". It displays a 9x9 matrix of 0s and 1s. The right screenshot is titled "C:\Documents and Settings\User\Мои доку..." and shows the output for "Massiv ulchamlari: n=16". It displays a 16x16 matrix of 0s and 1s. Both outputs show a pattern of 1s along the main diagonal and the anti-diagonal, with 0s elsewhere.

4-masala. Haqiqiy sonlardan iborat $A(n,n)$ o'lchovli massiv berilgan. Uning bosh diognali elementlari yig'indisi va ko'paytmasini hisoblash algoritmi va dasturini tuzing.

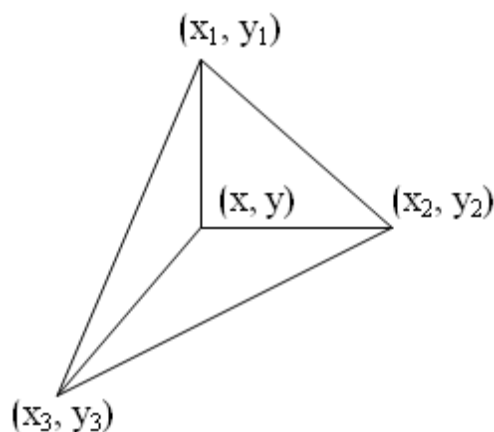


2-masala. 3 ta uchburchak tomonlarining uzunliklari bilab berilgan. Ularning yuzalarini hisoblash dasturini tuzing.

C++ tilida dasturi
<pre>#include <iostream.h> #include <conio.h> #include <math.h> using namespace std; float yuza(int a,int b,int c) { float s,p; p=(a+b+c)/2; s=sqrt(p*(p-a)*(p-b)*(p-c)); return s;} int main() { float a1,b1,c1,a2,b2,c2,a3,b3,c3,s1,s2,s3; cin>>a1>>b1>>c1>>a2>>b2>>c2>>a3>>b3>>c3; s1=yuza(a1,b1,c1);s2=yuza(a2,b2,c2);s3=yuza(a3,b3,c3); cout<<"1-uchburchak yuzasi="<<s1<<"\n"; cout<<"2-uchburchak yuzasi="<<s2<<"\n"; cout<<"3-uchburchak yuzasi="<<s3<<"\n"; system("pause"); return 0; }</pre>



4-masala. Uchburchak uchlarining koordinatalari (x_1, y_1) , (x_2, y_2) , (x_3, y_3) berilgan bo'lsin. (x, y) koordinatali nuqta shu uchburchakka tegishli bo'la olish yoki olmasligini aniqlovchi dastur tuzing.



Berilgan uchburchak yuzasi S bo'lsin. Uchburchak uchlarini (x, y) koordinatali nuqta bilan tutashtirib, 3 ta uchburchak hosil qilinadi. Ularning yuzalari mos ravishda S_1, S_2 va S_3 bo'lsin. Agar (x, y) nuqta berilgan uchburchakning ichida yotsa, $S = S_1 + S_2 + S_3$ bo'lishi kerak. Aks holda nuqta uchburchak ichida yotmaydi. Ko'rinib turibdiki, bu yerda uchlarining koordinatalari berilgan to'rtta uchburchak yuzalarini hisoblashga to'g'ri kelmoqda. Uchburchak uchlarining koordinatalari uchun formal o'zgaruvchilarni $(a_1, b_1), (a_2, b_2)$ va (a_3, b_3) tarzida tanlanadi. Bu uchburchakning tomonlari va yarim perimetrini belgilash uchun mos ravishda A, V, S va R o'zgaruvchilari olinadi. Ular masala shartida ko'rsatilmagani uchun lokal o'zgaruvchilar hisoblanadi. So'ngra ketma-ket to'rt marta $(x_1, x_2, x_3, y_1, y_2, y_3), (x, x_1, x_2, y, y_1, y_2), (x, x_2, x_3, y, y_2, y_3)$ va $(x, x_1, x_3, y, y_1, y_3)$ joriy o'zgaruvchilar uchun protseduraga murojaat qilinadi. Ya'ni uchlarining koordinatalari ana shu nuqtalarda yotgan uchburchaklarning yuzalari S_1, S_2, S_3 va S larni hisoblab topiladi. So'ng masalani yechish g'oyasiga ko'ra $S = S_1 + S_2 + S_3$ munosabatini o'rinli bo'lishi yoki bo'lmasligiga qarab xulosa chiqariladi.

C++ tilida dasturi

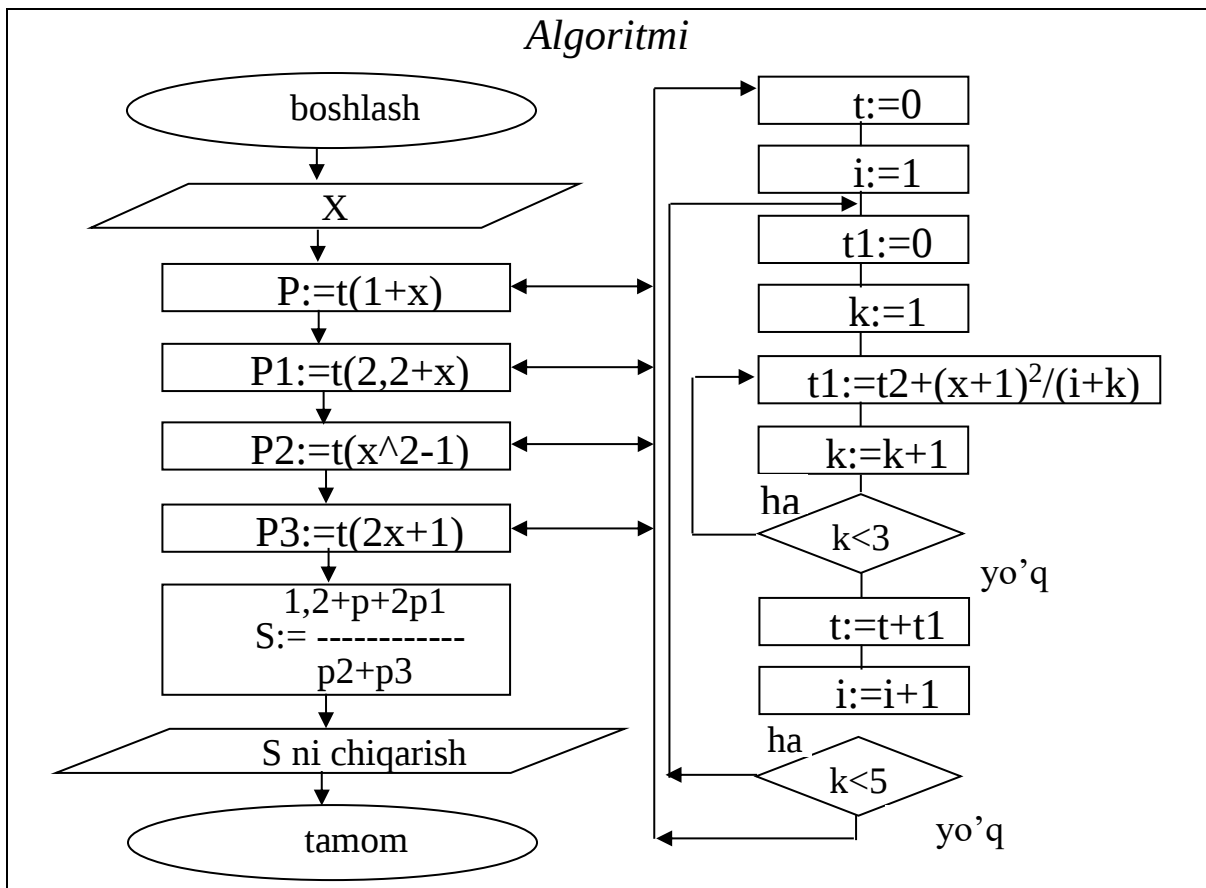
```
float uza (float a1, float a2, float a3, float b1, float b2, float b3)
{float a,b,c,p;
a= sqrt(pow(a2-a1,2) + pow(b2-b1,2)) ;
b= sqrt(pow(a3-a2,2) + pow(b3-b2,2)) ;
c= sqrt(pow(a3-a1,2) + pow(b3-b1,2)) ;
p= (a + b + c) / 2 ; return sqrt(p*(p-a) *(p-b) *(p-c)) ;}
int main()
{float x,x1,x2,x3,y,y1,y2,y3,s,s1;
cout<<"uchburchakning birinchi uchi"; cin>>x1>>y1;
cout<<"uchburchakning ikkinchi uchi"; cin>>x2>>y2;
cout<<"uchburchakning uchinchi uchi"; cin>>x3>>y3;
cout<<"nuqtaning koordinatasi"; cin>>x>>y;
s=uza(x1,x2,x3,y1,y2,y3)+ uza(x,x1,x2,y,y1,y2)+uza(x,x2,x3,y,y2,y3) ;
s1=uza (x,x1, x3,y,y1,y3) ;
if (s==s1) cout<<" nuqta uchburchakka tegishli"; else cout<<"
nuqta uchburchakka tegishli emas";
system("pause");
return 0;
}
```

uchburchakning birinchi uchi 2 3
 uchburchakning ikkinchi uchi 1 2
 uchburchakning uchinchi uchi 0 —3
 nuqtaning koordinatasi 0 0
 berilgan nuqta uchburchakka tegishli emas
 uchburchakning birinchi uchi 1 2
 uchburchakning ikkinchi uchi 2 3
 uchburchakning uchinchi uchi 3 4
 nuqtaning koordinatasi 2 2.5
 berilgan nuqta uchburchakka tegishli

4-masala. $S = \frac{1,2 + t(1+x) + 2t(2,2+x)}{t(x^2-1) + (2x+1)}$ ifodani qiymatini

hisoblash algoritmi va dasturini tuzing. Bu yerda $t(x) = \sum_{i=1}^5 \sum_{k=1}^3 \frac{(x+1)^2}{i+k}$ funksiya bo'lib, uning qiymati x argumentga bog'liq ravishda o'zgaradi.

Bu masalani yechish algoritmini tuzish uchun, birinchi navbatda noma'lum x o'zgaruvchiga qiymat beramiz. $t(x) = \sum_{i=1}^5 \sum_{k=1}^3 \frac{(x+1)^2}{i+k}$ funksiyani qiymatini hisoblash jarayonini qism dastur (function) shaklida ifodalaymiz va x ning har bir qiymatida unga murojoatni amalga oshirib boramiz, hamda murojoat qilishdan hosil bo'lgan qiymatlarni mos ravishda P, P1,P2,P3 orqali belgilab boramiz. Hosil bo'lgan qiymatlarni formulaga qo'yib, S ning qiymatini hisoblaymiz va natijani chop etamiz.



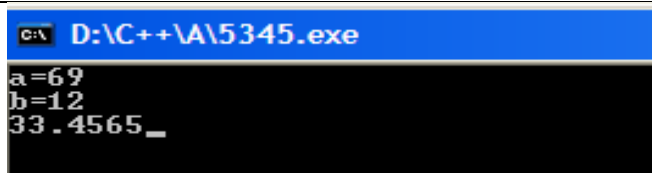
C++ tilida dasturi

```

#include <iostream.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>
float sum(float x)
{int i,j; float s,s1; s=0;
for (i = 1; i<=3; i=i+1)
{s1=0; for (j = 1; j<=3; j=j+1)
s=s+pow(2,1+x)/(i+j);s1=s1+s;} return s1;}
int main()
{float a,y;
cout<<"a ga qiymat bering = "; cin>>a;
y=(1.2+sum(1+a)+2* sum(2.2+a))/(sum(a*a-1)+sum(2*a+1));
cout<<"y ning qiymati= "<<y;
system("pause");
return 0;
}
  
```

5-masala. Ikkita a, b o'zgaruvchi e'lon qilinsin va bu o'zgaruvchilardan foydalangan holda $qush()$ va $ayir()$ funksiyalari tuzilib, dasturni asosiy qismida $(a+b)*(a-b)/((a+b)+(a-b))$ formula hisoblansin.

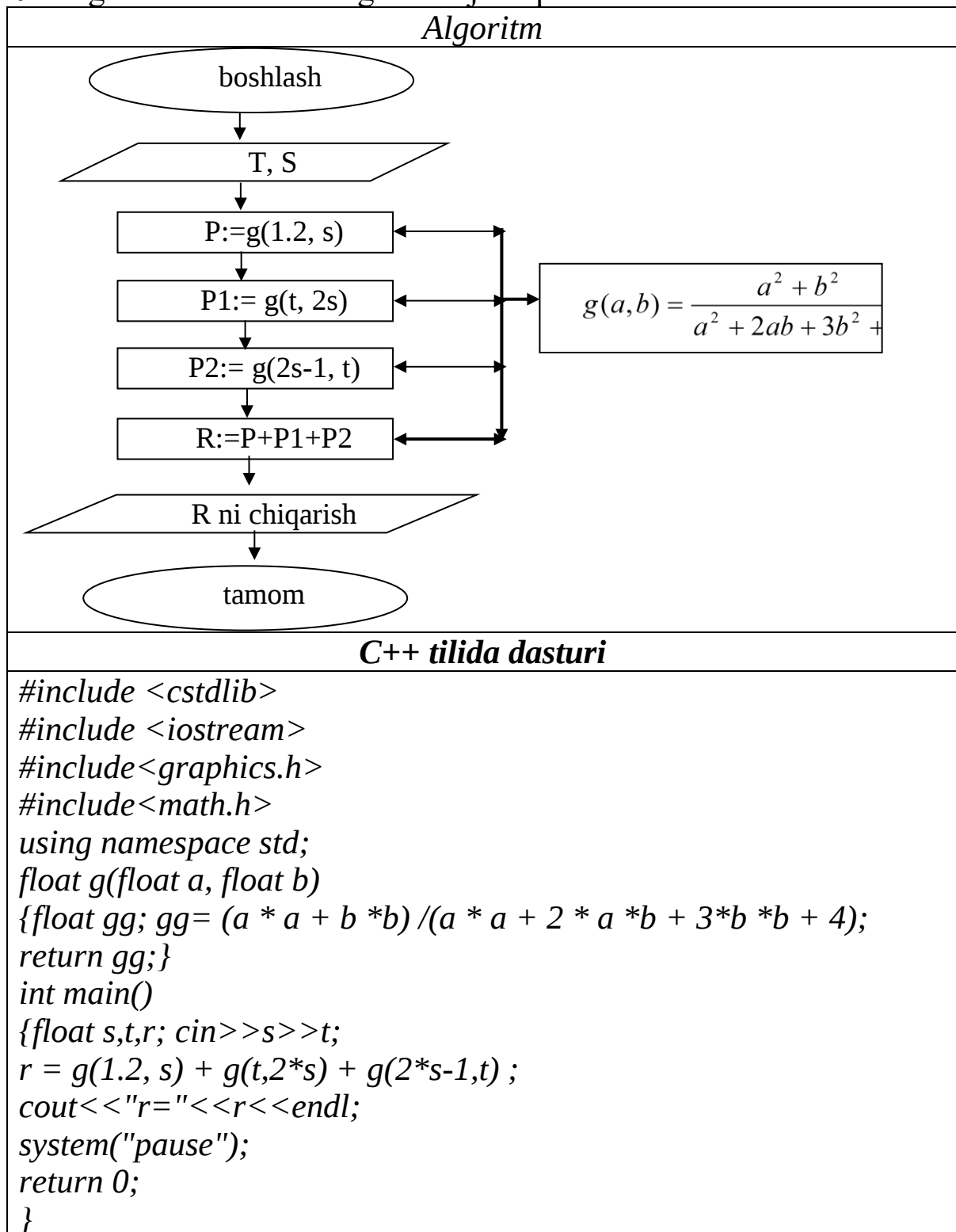
<i>C++ tilida dastur</i>
<pre> #include<iostream.h> #include<conio.h> #include<cmath> using namespace std; double qush(double x,double y) { double z; z=x+y; return z; } double ayir(double x,double y) { int z; z=x-y; return z; } int main() { double a,b; cout<<"a="; cin>>a; cout<<"b="; cin>>b; cout<<qush(a,b)*ayir(a,b)/(qush(a,b)+ayir(a,b)); getch(); } </pre>



6-masala. S va T haqiqiy sonlar berilgan bo'lsin. Quyidagi ifodaning qiymatini toping: $r = g(1.2, s) + g(t, 2s) + g(2s - 1, t)$. Bu erda $g(a, b) = \frac{a^2 + b^2}{a^2 + 2ab + 3b^2 + 4}$ funksiya a va b argumentlarga bog'liq funksiya.

Bu masalaning qo'yilishidan ko'rinib turibdiki, $g(a, b)$ funksiyasining qiymatini 3 marta hisoblashga to'g'ri kelmoqda. Shuning

uchun $g(a,b)$ funksiyani qiyamatini hisoblash maqsadida funksiya tashkil qilinadi. So'ngra, birinchi marta $a = 1.2$, $b = s$ bo'lgan hol uchun, ikkinchi marta $a = t$, $b = 2s$ uchun va uchinchi marta $a = 2s - 1$, $b = t$ bo'lgan hollar uchun unga murojaat qilinadi.



Bu dastur uchun kompyuter quyidagi natijani berdi:
 S va T ni kiriting 12 -5
 $R := 2.7373$

7-masala. Ikkita tub son «egizak» deyiladi, agar bir-biridan farqi 2 teng bo'lsa (masalan, 41 va 43 sonlari). Berilgan natural n uchun $[n..2n]$ kesmadagi barcha «egizak» sonlar juftliklari chop etilsin. Masalani yechish uchun berilgan k sonini tub son yoki yo'qligi aniqlovchi mantiqiy qiymat qaytaruvchi funksiya tuzish zarur bo'ladi. Funksiyada k soni $2..k/2$ gacha sonlarga bo'linadi, agar k bu sonlarning birortasiga ham bo'linmasa, u tub son hisoblanadi va funksiya true qiymatini qaytaradi. Bosh funksiya, berilgan n uchun $[n..2n]$ oraliqdagi $(n, n+2), (n+1, n+3), \dots, (2n-2, 2n)$ son juftliklarini tub sonlar ekanligi tekshiriladi va shart qanoatlantirgan juftliklar chop etiladi.

Dastur kodi:

```
bool TubSon (unsigned long k);
int main()
{
    unsigned long n,i;
    unsigned char egizak=0;
    cout<<"n->";
    cin>>n;
    cout<<'['<<n<<".."<<2*n<<']';
    for(i=n; i<=2*n-2; i++)
    if(TubSon (i) && TubSon(i+2))
    {
        if(!egizak)
            cout<<"\n";K"oralig'idagi egizak tub sonlar  else cout<<"; ";
        egizak=1;
        cout<<'{'<<i<<','<<i+2<<'}';
    };
    if(!egizak)
        cout<<"oralig'idagi egizak tub sonlar mavjud emas."; else cout<<'.';
    return 0;
}
bool Tub Son(unsigned long k);
{
    unsigned long m;
    for (m=2; m<=k/2; m++)
    if (k%m==0) return false;
    return true;}
Natural n soni uchun 100 kiritilsa, programma quyidagi sonlar juftliklarini chop qiladi: [100..200] orag'ida egizak tub sonlar:
{101,103}; {107,109}; {137,139}; {149,151}; {179,181}; {191,193};
{197,199}.
```

IV-bob. Aralash va murakkab darajadagi masalalarni yechish dasturlari.

4.1. Aralash masalalarni C++ tilida yechish dasturlari

1-masala. Ixtiyoriy kiritiladigan n sonining necha xonaligini aniqlovchi dastur tuzilsin.

C++ tilida dastur kodi

```
#include<conio.h>
#include<iostream.h>
int main()
{
    int i,n,j,k;
    cout<<"n=";
    cin>>n;
    j=0;
    k=n;
    while(n>0)
    {
        n=(n-n%10)/10;
        j++;
    }
    cout<<k<<" soni "<<j<<" xonali";
    getch();
}
```

2-masala. m va n sonlari kiritilsin. 1 dan m gacha bo'lgan oraliqda n ga qoldiqsiz bo'linadigan sonlar topilsin. Shu topilgan chegara sifatida qaralib har bir chagara oraligidagi sonlarning ko'paytmasi topilsin. Misol: $m=10$; $n=3$; Bunda 1 dan 10 gacha bo'lgan sonlar orasida 3 ga bo'linadiganlari: 3, 6, 9; har bir oraliq ko'paytmalari quyidagicha:

- a) $1 \dots 3 = 1 * 2 * 3 = 6$
- b) $3 \dots 6 = 3 * 4 * 5 * 6 = 360$
- c) $6 \dots 9 = 6 * 7 * 8 * 9 = 3024$;

C++ tilida dastur kodi

```
#include<conio.h>
#include<iostream.h>
int main()
{
    int n,m,j,i,a,s;
    cout<<"n=";
```

```

cin>>n;
cout<<"m=";
cin>>m;
a=1;
s=1;
for(i=1; i<=n;i++)
{
    if(i%m==0)
    {
        for (j=a;j<=i;j++)
            s=s*j;
        cout<<s<<" ";a=i;
    }
    s=1;
}

```

getch();

3-masala. Bozordan sigir, qo`y va tovuq sotib olish kerak. Xaridorda faqatgina 500 so`m bor. U shu puliga hammasi bo`lib 100 ta jonivor sotib olishi kerak. Xaridor shunda nechta sigir, qo`y va tovuq sotib oladi.

C++ tilida dastur kodi

```

#include<iostream.h>
#include<conio.h>
int main()
{
    int i,j,k;
    for(i=1;i<=100;i++)
    for(j=1;j<=100;j++)
    for(k=1;k<=100;k++)
    {
        if(i+j+k==100 && i*50+j*10+k*1==500)
            cout<<"Sigir:"<<i<<" ", Qo`y:"<<j<<" ", Tovuq:"<<k<<endl;
    }
    getch();
}

```

4-masala. Biror bir ixtiyoriy n va m sonlari kiritilsin. n sonini oxirgi raqami va n sonining qolgan qism oralig`iga m ta nol joylashtirilsin; Misol uchun n=123, m=4; Natija: 1200003;

C++ tilida dastur kodi

```
#include<iostream.h>
#include<conio.h>
int main()
{
    int n,i,m,a;
    cout<<"n=";
    cin>>n;
    cout<<"m=";
    cin>>m;
    a=n%10;
    n=n-a;
    for (i=1;i<=m;i++)
        n=n*10;
    n=n+a;
    cout<<n;
    getch();
}
```

5-masala. Berilgan ixtiyoriy n soni tashkil etuvchi raqamlarining o`rta arifmetigi topilsin. Misol uchun N=123, Javob: 2; N=1234, Javob:2,5

C++ tilida dastur kodi

```
#include<iostream.h>
#include<conio.h>
int main()
{
    int n; double s,a;
    cout<<"n=";
    cin>>n;
    a=0;
    s=0;
    while(n>0)
    {
        s=s+n%10;
        n=(n-n%10)/10;
    }
}
```

```

        a++;
    }
    cout<<"Natija:"<<s/a;
    getch();
}

```

6-masala. Fibonachchi sonlarini hisoblovchi dasturni yozing. Bunda n soni kiritilsin va n dastlabki n ta Fibonachchi soni topilsin.

C++ tilida dastur kodi

```

#include<iostream.h>
#include<conio.h>
#include<math>
int main()
{
    int f[100],i,n;
    cout<<"n=";
    cin>>n;
    f[0]=f[1]=1;
    for(i=2;i<=n;i++)
    {
        f[i]=f[i-1]+f[i-2];
        cout<<"F("<<i<<"")="<<f[i]<<endl;
    }
    getch();
}

```

7-masala. X (n) massivda har bir element 0, 1 yoki 2 ga teng. Massi v elementlari o'rnini shunday almashtirinki, natijada avval hamma nollar, keyin birliklar, nihoyat, hamma ikkiliklar joylashsin (qo'shimcha massiv ishlatish mumkin emas).

Masalani yechish algoritmi. Birinchi navbatda massiv elementlari soni so'raladi va kiritilib olinadi. So'ng dasturdan X nomli N ta elementdan iborat bo'lgan massiv e'lon qilinadi. Keyin eng kichik element deb birinchi element tanlanadi va qolgan o'zidan keyingi n-1 ta element bilan solishtiriladi. Agar undan kichik bo'lgan son bo'lsa u bilan o'rin almashtiriladi. Takrorlanish shu tarzda o'zidan keyingi elementlarga solishtirish yo'li bilan amalga oshiriladi.

Masalan: N=5;
1, 0,2,1,0 bo'lsa

0, 0, 1, 1, 2 natija olishimiz zarur.

Agar N=6;

0, 0, 2, 2, 1, 1 bo'lsa

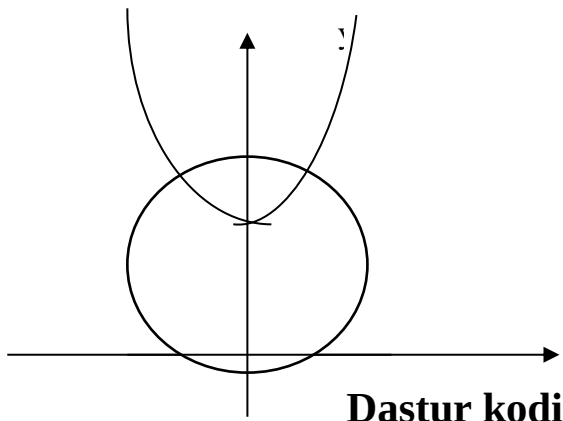
0, 0, 1, 1, 2, 2 natija olishimiz kerak.

Dasturlashda masalaning matematik tahlilidan kelib chiqib uning blok sxemasi chizib olinadi.

Dastur kodi

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    int n,m,t,k=0; //o'zgaruvchilarni e'lon qilish
    cout<<" N ni kiriting"<<endl;
    cin>>n;// n ni o'qitib olish
    int x[n];// x nomli n ta elementli massiv e'lon qilindi
    for (int i=0;i<n;i++){
        cout<<"X["<<i<<"]= ";
        cin>>x[i];//har bir element o'qib olinmoqda
    }
    //asosiy jarayonlar
    for(int i=0;i<n;i++){
        m=x[i];
        for(int j=i;j<n;j++){
            if (m>x[j]){ m=x[j]; t=j;k=k+1; }
        }
        if(k!=0){
            x[t]=x[i];
            x[i]=m;}
        k=0;
    } // Natijalarni chiqarish
    cout<<" Natijalar quyidagicha"<<endl;
    for (int i=0;i<n;i++){
        cout<<"x["<<i<<"]= "<<x[i]<<endl;
    }
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

8-masala. Ixtiyoriy berilgan $M(x,u)$ nuqta $u=x^2$ va $x^2+u^2=4$ aylana bilan kesishgan sohaga yoki shu aylananing 4-choragi tashqarisiga tushishini tekshiring.



Демак:

$y > x^2$ and $x^2 + y^2 \leq 4$
 or $x > 0$ and $y < 0$ and $x^2 + y^2 \geq 4$
 $x = 1, y = 1 \rightarrow \text{false}$
 $x = 1, y = 0 \rightarrow \text{true}$
 $x = -2, y = 0.5 \rightarrow \text{false}$
 $x = 2, y = -2 \rightarrow \text{true}$

Dastur kodi

```
#include <iostream.h>
void main ( )
{ float x, y; int n;
cout << "nuqtaning koordinatalarini kiriting:";
cin >> x>>y;
if ((y>=x*x && x*x+y*y<=4) || (x>0 && y<0 && x*x+y*y>=4))
n=1;
else n=0;
cout << "x="<<x<<endl;
cout << "y="<<y<< endl;
cout << "n="<<n<< endl;
system("PAUSE");
return EXIT_SUCCESS;
}
```

9-masala. Berilgan butun $n=100$ va a, b - haqiqiy sonlar uchun $f_1(x) = 5\sin(3x) + x$, $f_2(x) = \cos(x)$ va $f_3(x) = x^2 + 1$ funksiyalar uchun $\int_a^b f(x)dx$ integralini to'g'ri to'rtburchaklar formulasi bilan taqriban hisoblansin:

$\int_a^b f(x)dx \approx h[f(x_1) + f(x_2) + \dots + f(x_n)]$, bu erda $h = \frac{b-a}{n}$, $x_i = a + ih - h/2, i = 1..n$

.

Dastur kodi

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

```

#include <math.h>
const int n=100;
double f1(double x){return 5*sin(3*x)+x;}
double f3(double x){return x*x+1;}
double Integral(double(*f)(double),double a,double b)
{
    double x,s=0;
    double h=(b-a)/n;
    x=a-h/2;
    for(int i=1;i<=n; i++)s+=f(x+=h);
    s*=h;
    return s;
}
int main()
{
    double a,b;
    int menu;
    while(1)
    {
        cout<<"\nIsh regimini tanlang:\n";
        cout<<"1:f1(x)=5*sin(3*x)+x integralini\
hisoblash\n";
        cout<<"2:f2(x)=cos(x) integralini hisoblash\n";
        cout<<"3:f3(x)=x^2+1 integralini hisoblash\n";
        cout<<"0:Programmadan chiqish\n";
        do
        {
            cout<<" Ish regimi-> ";
            cin>>menu;
        }
        while (menu<0 || menu>3);
        if(!menu)break;
        cout<<"Integral oralig'ining quyi chegarasi a=";
        cin>>a;
        cout<<"Integral oralig'ining yuqori chegarasi b=";
        cin>>b;
        cout<<"Funksiya integrali S=";
        switch (menu)

```

```

{
case 1 : cout<<Integral(f1,a,b)<<endl; break;
case 2 : cout<<Integral(cos,a,b)<<endl; break;
case 3 : cout<<Integral(f3,a,b)<<endl;
}
}
return 0;
}

```

10-masala. Rekursiv funksiyadan foydalangan holda ikkita sondan raqamlari yig'indisi katta bo'lgan sonni topuvchi dastur tuzing.

```

int sum, sum_1, sum_2 ;
int raqam(int son)
{
    sum += son % 10;
    son = son / 10;
    if (son == 0) return sum;
    raqam (son);
}
int main()
{
    int sum_1 = 0, sum_2 = 0;
    int son_1, son_2;
    cin>>son_1>>son_2;
    sum_1 = raqam(son_1);
    sum_2 = raqam(son_2);
    if (sum_1 > sum_2) cout << son_1; else cout<<son_2;
    getch();
    return 0;
}

```

11-masala. n o'lchamli butun sonlardan iborat massiv berilgan . Bu massivning toq elementlarini indekslarini o'sib borish tartibida chop etish va toq elementlar sonini hisoblash dasturi tuzilsin.

Dastur kodi

```

#include<iostream>
int main ()
{
    int k[100]; int i,n,s;
    cout<<" sonini kiriting=";  cin>>n;

```

```

for ( i=1; i<=n; i++) cin>>k[i]; s=0;
for (i=1; i<=n; i+=2)
{   cout<<"k["<<i<<"]="<<k[i]<<endl;
    s++; }
cout<<"soni"<<" "<<"->"<<" "<<s<<" "<<"ta"<<endl;
system("pause");
return 0;
}

```

12-masala. 2 ta matrisa berilgan. Ularni o'zaro ko'paytirib yangi matrisa hosil qiling . Bu yerda 1-matrisaning ustunlar soni 2- matrisaning satrlar soniga teng bo'lishi kerak.

Dastur kodi

```

#include <iostream.h>
#include <conio.h>
#include <stdlib.h>
#include <time.h>
void main ( )
{
{ srand ( time (0));
int a[3][3], b[3][3],c[3][3], i, j, k;
for (i=0; i<3; i++)
for (j=0; j<3; j++) a[i][j] = rand ( );
for (i=0; i<3; i++)
for (j=0; j<3; j++) b[i][j] = rand ( );
for (i=0; i<3; i++)
{ for (j=0; j<3; j++)
{ c[i][j] = 0; for (k=0; k<3; k++)
c[i][j] = c[i][j] + a[i][k]*b[k][j];
cout <<"c="<<c[i][j]<<"\t"; }
cout << endl; }
getch ( )

```

13-masala. Keltirilgan dastur quyidagi mantiqiy o'yinni ifodalaydi

Dastur kodi

```

#include <iostream.h>
int main()
{
unsigned short kichik;

```

```

unsigned long katta;
const unsigned short MaxKichik=65535;
cout << "Kichik sonni kiriting:"; cin >> kichik;
cout << "Katta sonni kiriting:"; cin >> katta;
cout << "kichik son:" << kichik << "...";
//Har bir iteratsiyada uchta shart tekshiriladi.
while (kichik<katta && katta>0 &&
        kichik< MaxKichik )
{
    if(kichik%5000==0) //Har 5000 satrdan
        //keyin nuqta chiqariladi
        cout<<"." ; kichik++; katta-=2 ;
}
cout<<"\n kichik son:"<<kichik<<" katta son:"
<<katta << endl ;
return 0 ;
}

```

NATIJA:

Kichik sonni kirit : 2

Katta sonni kirit : 100000

Kichik son : 2

Kichik son :33335 katta son : 33334

Keltirilgan dastur quyidagi mantiqiy o'yinni ifodalaydi. Oldin ikkita son – kichik va katta kiritiladi. Undan so'ng toki ular bir-biriga teng bo'lmaguncha, ya'ni «uchrashmaguncha» kichik son birga oshiriladi, kattasi esa ikkiga kamaytiriladi. O'yinni maqsadi qiymatlar «uchrashadigan» sonni topishdir.

9 – 12 – satrlarda qiymatlar kiritiladi. 15 – satrda esa siklni davom ettirishning quyidagi uchta sharti tekshiriladi:

1. kichik o'zgaruvchisi qiymati katta o'zgaruvchisi qiymatidan oshmasligi.
2. katta o'zgaruvchisi qiymati manfiy va nolga teng emasligi
3. kichik o'zgaruvchisi qiymati Max Kichik qiymatidan oshib ketmasligi

17 – satrda esa kichik soni 5000 ga bo'lingandagi qoldiq hisoblanadi. Agarda kichik 5000 ga qoldiqsiz bo'linsa bu operatsiyaning bajarilishi natijasi 0 ga teng bo'ladi. Bu holatda hisoblash jarayonini vizual ifodasi sifatida ekranga nuqta chiqariladi. Keyin esa kichik qiymati bittaga

oshiriladi, katta qiymati esa 2 taga kamaytiriladi. Sikl agarda tekshirish sharti tarkibidagi birorta shart bajarilmasa to'xtatiladi va boshqaruv 24 – satrga o'tadi.

14-masala. Mantiqiy o'yinni murakkablashgan ko'rinishi.

Dastur kodi

```
int main()
{ unsigned short  kichik ;
unsigned long  katta;
unsigned long  qadam;
unsigned long  maqsad ;
const unsigned short  MaxKichik = 65535;
cout<< "Kichik nomerni kiriting:"; cin >>kichik ;
cout<< "Katta nomerni kiriting :"; cin >>katta ;
cout<<"Qadam qiymatini kiriting:"; cin >>qadam ;
cout<<"Maqsadli kattalik qiymatini kiriting;; cin >> maqsad ;
cout << "\n";
while(kichik<katta && katta>0 &&
kichik<MaxKichik)
{  kichik++ ;  if(kichik%qadam==0)
{    cout << "qadam:" << kichik << endl ;  continue ;
  }
if(katta==maqsad) //maqsadli nuqtaga
// tengligini tekshirish
{
cout << "Maqsadga erishildi !; break;
}
katta -= 2;
}
cout<< "\n Kichik son:" << kichik <<
<<" katta son:"<< katta << endl ;
return 0;
}
```

NATIJA:

Kichik sonni kiriting: 2

Katta sonni kiriting: 20

Qadam qiymatini kiriting: 4

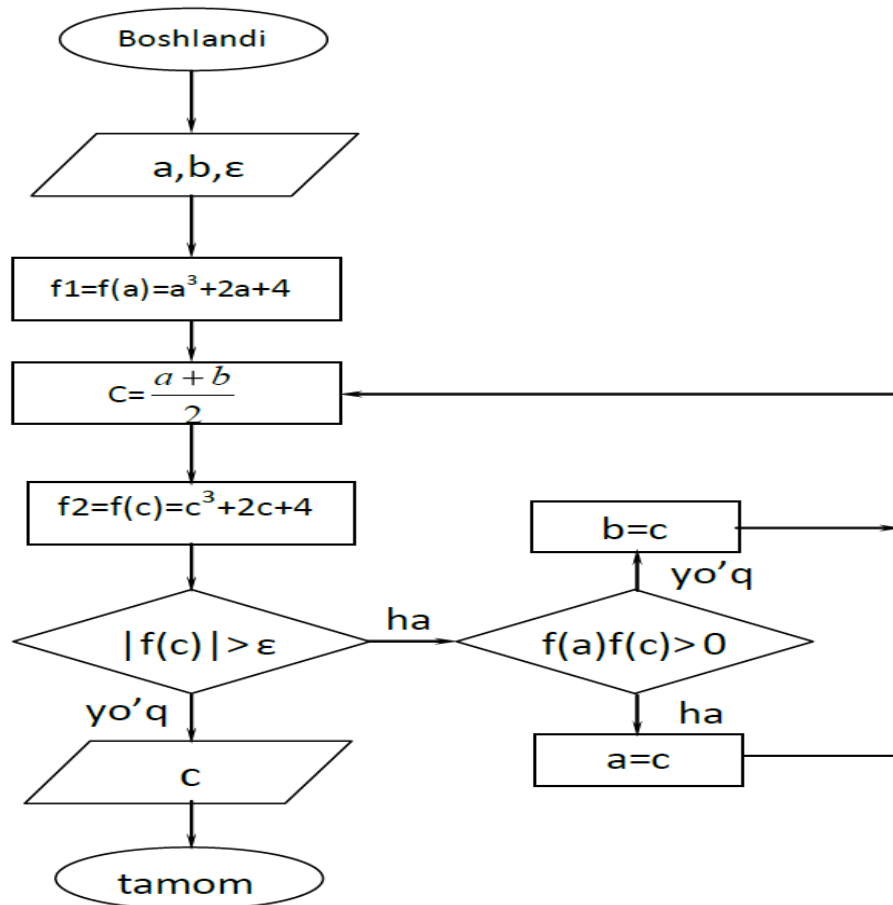
Maqsadli kattalik qiymatini kiriting: 6

Qadam :4

Qadam: 8

Kichik son : 10 Katta son:8

14-masala. $x^3 + 2x + 4 = 0$ algebraik tenglamani kesmani teng ikkiga bo'lish usuli orqali yechish algoritmi va dasturini tuzing.



C++ tilida dastur kodi

```
#include <cstdlib>
#include <iostream>
#include<graphics.h>
#include<math.h>
using namespace std;
int main()
{
float t=0.000000000001;
float a,b,f1,f2,c;
cout<<"a va b ni kiriting: ";
cin>>a>>b;
f1=pow(a,3)+2*a+4;
do
```

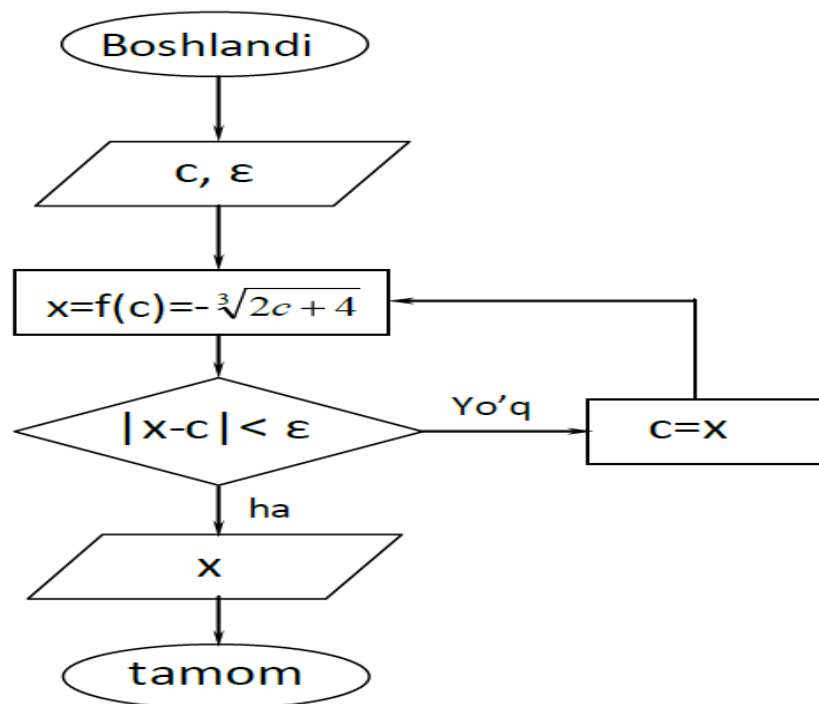
```

{
c=(a+b)/2; f2=pow(c,3)+2*c+4;
if (f1*f2>0) a=c; else b=c;
}
while (fabs(f2)<t);
cout<<"tenglamaning yechimi c="<<c;
system("PAUSE");
return EXIT_SUCCESS;
}

```

15-masala. $x^3 + 2x + 4 = 0$ algebraik tenglamani iteratsiya usuli orqali yechish algoritmi va dasturini tuzing.

Algoritmi



Dastur kodi

```

#include <cstdlib>
#include <iostream>
#include<graphics.h>
#include<math.h>
using namespace std;
int main()
{
float t=0.000000000001; float c,x;
cin>>c;
do
{

```

```

c=x;
x=exp(1/3*log(2*c+4));
}
while (fabs(x-c)<t);
cout<<"x="<<x;
getch();
system("PAUSE");
return EXIT_SUCCESS;
}

```

16-masala. Berilgan son raqamlari yig'indisini hisoblash

Dastur kodi

```

#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{int n; double s,a,t;
    cout<<"n="; cin>>n;
    a=0; s=0;
    while(n>0)
    {
        t=n%10;cout<<t<<"\t"; s=s+t;
        n=(n-n%10)/10; a=a+1;
    }
    cout<<endl<<"raqamlar
soni="<<a<<"\t"<<"Natija:"<<s/a<<endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}

```

17-masala. Kompyuter o'ylagan sonni topish

Dastur kodi

```

#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{ int x, y= 0, u = 0;
    srand(time(NULL));
    x = rand() % 1000 + 1;
    cout << "Kompyuter o'ylagan sonni toping" << endl;
}

```

```

while (x != y)
{
    cin >> y;
    u++;
    if (x > y) cout << "Kompyuter o'ylagan son katta" << endl;
    else if (x < y) cout << "Kompyuter o'ylagan son kichik" << endl;
}
cout << "Qoyil topdingiz!!!" << endl;
cout << "Urinishlar soni=" << u << endl;
system("PAUSE");
return EXIT_SUCCESS;

```

18-masala. 1 dan m gacha tub sonlar yig'indisini hisoblash

Dastur kodi

```

#include <cstdlib>
#include <iostream>
#include<graphics.h>
#include<math.h>
using namespace std;

int main(int argc, char *argv[])
{
    int i,k,m,a, n=0;float s;
    cout<<("Ixtiyoriy butun son kiriting: ");
    cin>>m; s=0;
    for (a=1;a<=m;a=a+1){n=0;
    for ( i=2; i<a; i++) {if (a%i == 0) n++;}
    if (n == 0 ) s=s+a;}
    cout<<s<<endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}

```

19-masala. Kiritilgan sonni tub yoki tub soni emasligini aniqlash .

Dastur kodi

```

#include <cstdlib>
#include <iostream>
#include<graphics.h>
#include<math.h>
using namespace std;
int main(int argc, char *argv[])

```

```

{
    int i, a, n=0;
    cout<<("Ixtiyoriy butun son kiriting: ");
    cin>>a;
    for ( i=2; i<a; i++) { if (a%i == 0) n++; }
    if (n == 0 ) cout<< '\t'<<a<<" soni tub\n";
    else cout<< '\t'<<a<<" tub son emas\n";
    system("PAUSE");
    return EXIT_SUCCESS;
}

```

20-masala. $ax+b=0$ tenglamani yechish .

Dastur kodi

```

#include <cstdlib>
#include <iostream>
#include <math.h>
using namespace std;
main()
{ float a,b,x,y;
  cout<<"a=";cin>>a; cout<<endl<<"b="; cin>>b;
  if (a!=0) {x=-b/a; cout<<"yechim="<<x<<endl;}
  if((a==0)&&(b==0)) cout<<"cheksiz yechim"<<endl;else
  if((a==0)&&(b!=0)) cout<<"bo'sh yechim"<<endl;
  system("PAUSE");
  return EXIT_SUCCESS;
}

```

21-masala. a va b sonlarining eng katta umumiy bo'luvchisini topish.

Dastur kodi

```

#include <iostream>
#include <conio.h>
#include <math.h>
using namespace std;
void EKUB(int a, int b)
{
    if (a > b) a= a - b; else b = b - a; // Kattasidan kichigini ayiramiz
    if (b == 0)
    {

```

```

        cout << a; return; // Funksiya ishini tugashi
    }
    EKUB(a, b); // Funksiya o'zini yangi
                // parametrlar bilan chaqiradi
}
int main()
{
    int K, L,D;
    cin >> K >> L;
    EKUB (K,L); EKUB(L,D);
    cout<<endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}

```

22-masala. Eng kichik umumiy karralini topish

Dastur kodi

```

#include <iostream>
#include <graphics.h>
#include <conio.h>
#include <math.h>
int nod(int m, int n)
{
    if (n == 0) return m;
    else
    return nod(n, m % n);
}
int main() {
    int n = 0;
    int m = 0;
    std::cout << "Birinchi sonni kiriting:\n";
    std::cin >> n;
    std::cout << "Ikkinchi sonni kiriting:\n";
    std::cin >> m;
    std::cout << "EKUK= "<< n*m/nod(m, n) << "\n";
    getch();
    return 0;
}

```

4.2. Murakkab darajadagi masalalarni yechish algoritmi va dasturlari.

1-masala. «Kublar yig'indisi». Berilgan N natural sonni 2ta natural sonlar kublarining yig'indisi ($N=i^3+j^3$) ko'rinishida necha xil usul bilan yozish mumkin? Qo'shiluvchilarning o'rin almashtirilishi yangi usulni bermaydi. $1/3$ -darajaga ko'tarish amalidan foydalanish mumkin emas.

Algoritm

$M=0$ ni olamiz $i-1, j$ esa birinchi son, uning uchun $j^3+1 \geq N$. $K=i^3+j^3$ deb olamiz va « K » ni « N » bilan taqqoslaymiz. Uch hol bo'lishi mumkin:

1. agar $k < N$ bo'lsa, $i=i+1$ ni hosil qilamiz.
2. agar $K > N$ bo'lsa, $j=j-1$ ni hosil qilamiz.
3. agar $K=N$ bo'lsa, $M=M+1; j=j-1; i=i+1$ ni hosil qilamiz.

Bu hisoblashlar $i \leq j$ bo'lguncha davom ettiriladi. Shuningdek, (3) holda $i=i+1$ o'rniga darrov $i=i+2$ yozish qulayligini bildiramiz. Boshida $i \leq j \leq \sqrt[3]{N} + 1$ bo'lgani va har bir qadamda i o'sgani yoki j kamaygani uchun qadamlar $\sqrt[3]{N}$ dan katta bo'lmaydi. Algoritm to'g'riligini isbotlash uchun $P(i,j)=\{(x,y)\}$ orqali $i \leq x \leq y \leq j$ va $x^3+y^3=N$ shartlarni qanoatlantiruvchi shunday x va y natural son juftlari to'plamini belgilaymiz. Faraz qilamiz, $k=i^3+j^3$ bo'lsin, agar $k < N$ bo'lsa, $P(i,j)$ to'plamdagi ixtiyoriy (x,u) juftlik uchun $i^3+j^3 < x^3+y^3$ bo'ladi, lekin $u \leq j$, bundan $i < x$ va, demak, agar $k < N$ bo'lsa, $P(i,j)=P(i+1,j)$.

Xuddi shunga o'xshash, agar $k > N$ bo'lsa, $P(i,j)=P(i,j-1)$, nihoyat, $k=N$ da, (i,j) juftlik $P(i,j)$ dan olib tashlanganidan keyin, agar $k=N$ bo'lsa, $P(i,j) \setminus (i,j)=P(i+1, j-1)$ bo'ladi.

Dastur kodi

```
#include <cstdlib>
#include <iostream>
#include <graphics.h>
#include <math.h>
using namespace std;
int main(int argc, char *argv[])
{int i,j,m,k,n; cout<<"N="; cin>>n;
m=0; j=1; i=1;
while ((j*j*j+1)<n) j=j+1;
```

```

do {
k=i*i*i+j*j*j;
if (k==n) m=m+1;
if (k<=n) i=i+1;
if (k>=n) j=j-1;} while(i<j);
cout<<m;
system("PAUSE");
return EXIT_SUCCESS; }

```

2-masala. «Oddiy bo'luvchilar». Natural N son berilgan . Uning barcha oddiy bo'luvchilarini toping.

Algoritm

$i=2,3,\dots$ sonlarini i soni N sonining bo'luvchisi bo'lguncha almashtiramiz. N ni i ga bo'lamiz $N = N / I$ va bo'lishni N, i soni i ga bo'linguncha takrorlaymiz. Shundan keyin, $N > 1$ bo'lsa, i ning navbatdagi qiymatiga o'tamiz va hokazo. Shunday tarzda boshlang'ich N sonining hamma oddiy bo'luvchilarini va faqat ularni topish mumkinligini sezish qiyin emas.

Boshda 0 ga, keyinchalik, oxirgi bo'luvchiga teng bo'lgan, yana bitta j o'zgaruvchini kiritamiz. Yangi bo'luvchi birinchi marta j dan farq qiladi va chop etiladi. Dastur ishini tezlashtirish uchun, $i = 2$ holini alohida ko'rish va keyinchalik esa i ning toq qiymatlari bo'yicha siljish mumkin.

Dastur

```

#include <cstdlib>
#include <iostream>
#include<graphics.h>
#include<math.h>
using namespace std;
int main(int argc, char *argv[])
{int i,j,n; cout<<"N="; cin>>n;
j=0; i=2;
while (n>1)
{ while ((n % i)!=0) i=i+1;
if (i!=j)
{cout<<i;j=i;
}
n=n / i ;
}
system("PAUSE");
}

```

```

    return EXIT_SUCCESS;
}

```

3-masala.«Tub sonlar». Berilgan M sonidan katta bo'lmagan barcha tub sonlarni bosmadan chiqaring.

Algoritm

Topilgan tub sonlar uchun hisoblashni tezlashtirish maqsadida jadval hosil qilish foydali. Navbatdagi nomzodni faqat shu jadvaldagi sonlarga bo'linishini tekshirish kerak. Juft sonlar, tabiiy, qaralmaydi. Jadval kamida $\sqrt{M}/2$ sonlar uchun ishlatilishi mumkin. Ya'ni 1000 ta sonli jadval 4000000 gacha bo'lgan sonlarni chop etish uchun yetarli. Algoritm va dasturni, 3 ga bo'linadigan sonlarni e'tibordan chetda qoldirib yaxshilash mumkin.

Dastur kodi

```

#include <cstdlib>
#include <iostream>
#include<graphics.h>
#include<math.h>
using namespace std;
int main(int argc, char *argv[])
{int n=200, m,i,j,k,g;
float p[n];
cout<< "M=" ; cin>>m;
cout<< "N=" <<n;
if (m>=2) cout<<2;
if (m>=3) cout<<3;
k=1; p[k]=3; i=5;
while (i<=m)
{
for ( j=1;j<=k; j=j+1)
{ g=p[j];
if (g*g>i) goto n1;
if ((i % g)==0) goto n2;
}
if (k==n) i=m-1;
else
n1: {
cout<<i;
if (k<=n-1) {
k=k+1;
p[k]=i; }

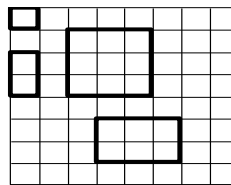
```

```

}
n2: i=i+2;
}
system("PAUSE");
return EXIT_SUCCESS;
}

```

4-masala. «To'rtburchaklar». 100x100 katakchali o'lchamdagi qog'oz varoqqa bir nechta to'rtburchaklar chizilgan. Har bir to'rtburchak butun kataklardan iborat, har xil to'rtburchaklar bir-birining ustiga yotmaydi va bir-biriga urilmaydi. (100x100) o'lchovli massiv berilgan, unda agar $[i,j]$ katak qandaydir bir to'rtburchakka tegishli bo'lsa, $A[i,j]=1$ va aks holda $A[i,j]=0$. To'rtburchaklar sonini hisoblab, chop etuvchi dasturni yozing.



Algoritm

Bu masala «bir fikrlash» da yechiladi: to'rtburchaklarning shimoliy-g'arbiy burchaklari (boshqacha aytganda-yuqori chap) qancha bo'lsa, to'rtburchaklar shuncha. Faqat burchak chegarada bo'lgan holda chalkashilmasa bo'lgani. Bu qiyinchilik dasturlarda har xil hal qilinishi mumkin. Faqat dasturda $i>1$ and $a[i-1,j] = 0$ turdagi ifodani ishlatmaslik kerakligini ogohlantiramiz. Chunki, agar indeksasiya birdan boshlansa, $i=1$ da bunday ifoda sintaktik xato bo'lib hisoblanadi.

Dastur

```

#include <cstdlib>
#include <iostream>
#include<graphics.h>
#include<math.h>
using namespace std;
int main(int argc, char *argv[])
{int nn=100, mm=100,m, i,n,s,j;
float a [mm][nn];
cin>>m>>n;
for (i=0;i<=m; i=i+1)
for (j=0;j<=n; j=j+1)
if ((i==0) || (j==0)) a[i][j]=0; else

```

```

{
cout<<"a["<<i<<"<<j<<"]="<<endl;
cin >>a[i][j];
}
s=0;
for (i=0;i<=m; i=i+1)
for (j=0;j<=n; j=j+1)
if ((a[i][j]==1) && (a[i-1][j]+a[i][j-1]==0)) s=s+1;
cout<<s;
system("PAUSE");
return EXIT_SUCCESS;
}

```

5-masala. «Qo'shiluvchilarga ajratish». Berilgan N natural sonni qo'shiluvchilarga ajratilgan hamma ko'rinishlarini hosil qiling. Qo'shiluvchilarning o'rin almashtirilishi yangi ko'rinish deb hisoblanmaydi.

Algoritm

Qo'shiluvchilar o'rnini almashtirishda ajratishlarning takrorlanmasligiga erishish uchun « n » sonining shunday natural qo'shiluvchilar-ga ajralishini ko'rib o'tamizki, $n=m_1+m_2+\dots$, bunda, $m_1 \geq m_2 \geq \dots$

Ajralishni tartibga solamiz:

$$n=m'_1+m'_2+\dots \quad (m')$$

$$n=m''_1+m''_2+\dots \quad (m'')$$

- ikkita ajratish. Agar birinchi tengmas $m'_i \neq m''_i$ juftdayoq, $m'_i > m''_i$, ya'ni hamma $j < i$ va $m'_j > m''_j$ da $m'_j \neq m''_j$ bo'lsa (m') ajratish (m'') ajratishdan oldin keladi, deb hisoblaymiz. Shu qoidaga ko'ra, $n=5$ sonini qo'shiluvchilarga ajratish quyidagi tartibda bo'ladi:

$$5=5,$$

$$5=4+1,$$

$$5=3+2,$$

$$5=3+1+1,$$

$$5=2+2+1,$$

$$5=2+1+1+1,$$

$$5=1+1+1+1+1$$

Berilgan N sonining qo'shiluvchilarga ajratishning hamma ko'rinishlarini hosil qilish uchun ishni $n=n$ ajratishdan boshlaymiz. Agar navbatdagi ajratish

$$n=m_1+m_2+\dots+m_i \quad (*)$$

hosil qilingan bo'lsa, uni chop etamiz va bevosita undan keyin keluvchisini tuzishga o'tamiz. Buning uchun massiv bo'yicha o'ngdan chapga tomon siljib borib, $m_i, m_{i-1}, \dots, m_k$ ni, « m_k » ning birdan katta qo'shiluvchisigacha, ko'rib boramiz. Agar bunaqasi uchramasa, hosil bo'lgan (*) ajratish oxirgisi va ish tugagan, $m_k > 1$ qo'shiluvchi topilgan bo'ladi. m_k ni birga kamaytirib va undan keyin keluvchi birlik qo'shiluvchilarni olib tashlab, butun yig'indini $S=1+i-k$ miqdorda kamaytiramiz. Endi $j=k+1, k+2, \dots$, uchun m_j ning yangi ketma-ket qo'shiluvchilarini aniqlaymiz. Agar $S > m_k$ bo'lsa, $m_j = m_k$, deb olamiz va S ni m_k miqdorga kamaytiramiz. Agar $S \leq m_k$ bo'lsa, $m_j = S$ deb olamiz va shu bilan n ning qo'shiluvchilarga ajratishning yangi ko'rinishini tugatamiz.

Dastur

```
#include <iostream>
#include<graphics.h>
#include<math.h>
using namespace std;
int main(int argc, char *argv[])
{int nn=100, n,i,k,t,s;
int m[nn];
cout<<"n="; cin>>n;
cout<<n<<endl;
m [1]=n; k=1; i=1;
ra: t=m[k]-1 ; s=t+i-k+1;
for (i=k;i<=n; i=i+1)
if (s>t) {
m[i]=t; s=s-t;
}
else
{
m[i]=s;
goto br;
}
br: for (k=1;k<=i;k=k+1) cout<<m[k]<<" ";
cout<<endl;
for (k=i;k<=1;k=k-1)
if (m[k]>1) goto ra;
system("PAUSE");
return EXIT_SUCCESS;
}
```

6-masala.«Ryukzak». Berilgan n predmetdan shundaylarini tanlab olish kerakki, ularning jami og'irligi 30 kg dan kichik, qiymatlari esa eng katta bo'lsin. Tanlangan predmetlarning jami qiymati chop etilsin.

Aniqrog'i – 2 ta musbat sonli $A(n)$ va $B(n)$ massivlar berilgan. Shunday har xil juftli $i_1, i_2, \dots, i_k$ sonlarni tanlash kerakki, natijada

$$a_{i_1} + a_{i_2} + \dots + a_{i_k} < 30$$

$$b_{i_1} + b_{i_2} + \dots + b_{i_k} = \max$$

bo'lsin, faqat max miqdorini chop eting.

Dastur kodi

```
{#include <cstdlib>
#include <iostream>
#include<graphics.h>
#include<math.h>
using namespace std;
int main(int argc, char *argv[])
{
float a,b; /*tomonlar uzunligi*/
float f; /*graduslarda ifodalovchi burchak kattaligi*/
float s; /*uchburchak yuzi*/
cout<<"Uchburchak yuzini toppish"<<endl;
cout<<" Uchburchak tomonlari uzunligi (sm):";
cout<<" - >";
cin>>a>> b;
cout<<"Uchburchak ikki tomoni o'rtasidagi burchak kattaligi (gradus):
"<<endl;
cout<<" - > ";
cin>>f;
/*s = a * h/2, h (uchburchak balandligi) h=b*sin(f) formula bilan hisob-
lanishi mumkin. Lekin, sin funksiyaning argumenti radianlarda ifoda-
lanishi kerak (1 rad. = 180/3.1415925, bu yerda 3.1415926 - "PI" conl)*/
s=a*b*sin (f*3.1415926/180) /2;
cout<<endl;
cout<<"uchburchak yuzi:"<<s<< "sm.kv."<<endl;
system("PAUSE");
return EXIT_SUCCESS;
}
```

7-masala. Har qanday 7 dan katta butun sondagi pul miqdoriga ega kupyuralarni 3 va 5 so'mlik kupyuralarda berish mumkinligi isbotlansin. Qo'yilgan masala $p=3n+5m$ tenglamasini qanotlantiruvchi m, n sonlar juftliklarini topish masalasidir (p -pul miqdori). Bu shartning ba-

jarilishini m va n o'zgaruvchilarining mumkin bo'lgan qiymatlarining barcha kombinatsiyalarida tekshirish zarur bo'ladi.

Dastur kodi

```
#include <iostream.h>
int main()
{ unsigned int pul; //pul- kiritiladigan pul miqdori
  unsigned int n3,m5; //n-3 so'mliklar , m-5 so'mliklar soni
  bool xato=false; //pul qiymatini kiritilgandagi xatolik
  do { if (xato) cout<<"kiritilgan pul qiymati 7 dan kichik !";
    xato=true ; //keyingi takrorlash xato hisoblanadi
    cout<<"\npul qiymatini kiriting (>7):";
    cin>>pul; }
  while (pul<=7); // toki 7 sonidan katta son kiritulguncha
  n3=0 ; //birorta ham 3 so'mlik yo'q
  do { m5=0; // birorta ham 5 so'mlik yo'q
    do
    {
      if (3*n3+5*m5==pul)
        cout<<n3<<" ta 3 so'mlik + "<<m5<<" ta 5 so'mlik\n";
      m5++ //5so'mliklar bittaga oshiriladi
    }
    while(3*n3+5*m5<=pul);
    n3++; //3 so'mliklar bittaga oshiriladi
  }
  while(3*n3<=pul);
  system("PAUSE");
  return EXIT_SUCCESS;
}
```

8-masala. $y=\sin 2x$, $y=\cos 2x$ va $y=x^2$ funksiyalarning grafigini chizing.

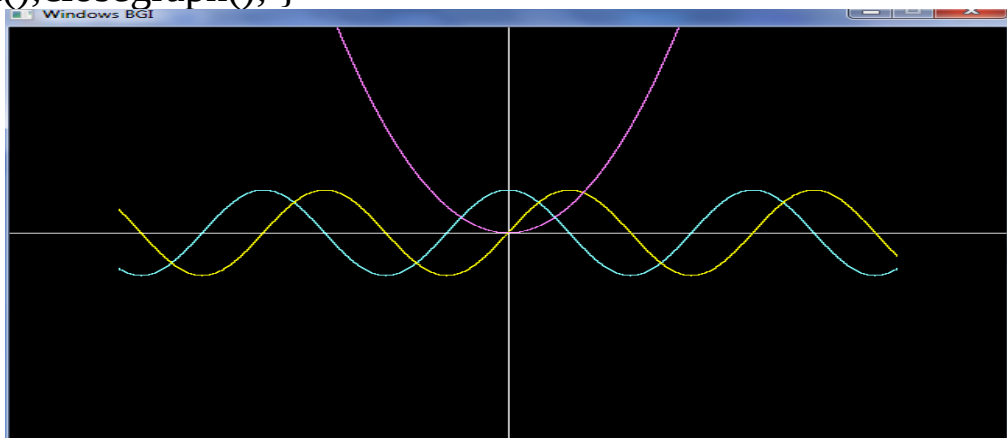
Dastur kodi

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
#include <graphics.h>
int main() { float y,x; float z,w;
  initwindow(640,480); //Oyna o'lchami
  moveto(320,0); //koordinata gorizontal o'qi boshi
  lineto(320,480); //koordinata gorizontal o'qi oxiri
```

```

moveto(0,240); //koordinata vertikal o'qi boshi
lineto(640,240); //koordinata vertikal o'qi oxiri
z=320; w=240;
for(x=-5;x<=5;x+=0.0001)
{ y=cos(2*x); moveto(z,w);
putpixel(320+x*50,240-y*50,11); z=320+x*50; w=240-y*50; }
z=320; w=240;
for(x=-5;x<=5;x+=0.0001)
{ y=sin(2*x); moveto(z,w);
putpixel(320+x*50,240-y*50,14); z=320+x*50; w=240-y*50; }
z=320; w=240;
for(x=-5;x<=5;x+=0.0001)
{ y=x*x; moveto(z,w);
putpixel(320+x*50,240-y*50,13); z=320+x*50; w=240-y*50;}
getch();closegraph(); }

```



9-masala. Biror bir massiv kiritilsin va shu massivning minimal va maksimal qiymatlari orasidagi elementlar ekranga chiqarilsin. Masalan:
 2,3,1,3,4,5,6,7,4,3,5, Javob: 1,3,4,5,6,7 ||
 -1,2,3,7,2,1, -3,-4, 2,1 , Javob: 7,2,1,-3,-4

```

#include<iostream.h>
#include<conio.h>
using namespace std;
int main()
{   double a[50],max,min; int i,n,b,t;
    cout<<"Massiv o`lchami kiritilsin:\nn=";
    cin>>n;
    for (i=1;i<=n;i++)
    {
        cout<<"a["<<i<<"]="";
        cin>>a[i];
    }
}

```

```

}
min=a[1];
max=a[1];
for (i=1;i<=n;i++)
{
    if(min>a[i])
        { min=a[i]; b=i; }
    if(max<=a[i])
        {
            t=i;
            max=a[i];
        }
    }    if (t<b)
for(i=t;i<=b;i++)
cout<<a[i]<<",";
else
for(i=b;i<=t;i++)
cout<<a[i]<<",";
    getch(); }

```

V-bob. Mustaqil algoritm va dastur tuzishga doir masalalar.

5.1. Matematik ifodalarni dasturlash tilida yozing.

$$1) a = u^{(x+y)/2} - \sqrt{\frac{x-1}{y+1}};$$

$$2) b = \frac{e^{3xy} + \ln|3x^2 - y|}{\sqrt{x^2 + y^2}};$$

$$3) a = \frac{(x+y)^3 + \sqrt[3]{|x+y|}}{3}$$

$$4) t = \frac{e^{a+b} + \sqrt{x^2 + 3}}{a}$$

$$5) y = \frac{\sqrt[3]{x+2} + \operatorname{tg}^3 \sin|x|}{4 \lg \sin x^2}, \quad f) z = \frac{e^{|x+1|} + 2}{\operatorname{ctg}^2 3x - 1},$$

$$6) Z = \sin^2 x^3 + (a+b)^{2 \sin^2 x} + b \cos^2 y^3$$

$$7) Z = \sqrt[3]{\frac{c^2}{b^2}} - \frac{2ab + c^2}{\sqrt[3]{a+b} - 2ab^2}$$

$$8) Z = \frac{a}{b^2} + \frac{a^2 + 6a^2 b^3 + 2c^2}{\sqrt[3]{a^3 + b^3} - 2ab}$$

$$9) y = \frac{y^5 \sqrt{7,2631} + x^3 \sqrt{71,8672}}{\sqrt[3]{a} + \sqrt[5]{b}}$$

$$10) y = \frac{2a^2 + bc^3 + 2d}{\sqrt[3]{a^3 - b^3} 3abc}$$

$$11) a = \frac{(x+y)^3 + \sqrt[3]{|x+y|}}{3}$$

$$12) b = \frac{\ln \left| \frac{x}{y} \right| + \lg \left| \frac{y}{x} \right|}{e^{xy}}$$

$$13) z = (1+u) \frac{x + \frac{y}{u}}{a - \frac{1}{1-x^2}};$$

$$14) t = e^{|x|} \cdot \operatorname{tg} \frac{a+b}{a-b}, \quad b = \frac{\cos x + \sin x^2}{e^x + a},$$

$$15) a = \frac{\sqrt[3]{8-x-y^2+1}}{x^2 + y^2 + 1}; \quad b = e^{x-y} (\operatorname{tg}^2 z + 1)$$

$$16) a = \sqrt{10(\sqrt[3]{x} + x^{y+2})}; \quad b = (\arcsin z)^2 + x + y$$

$$17) a = \lg(\sqrt{e^{x-y}} + x^y + z) \quad b = x - \frac{x^3}{3!} + \frac{x^5}{5!}$$

$$18) y = a * e^{\lg x} - \sqrt{ax + b}$$

$$19) Z = t^2 * \lg(|a + b|) + |x^3 - 2x|$$

$$20) c = \sin^2 x^2 + \frac{x - a}{x + b}$$

5.2. Chiziqli jarayonlarga algoritm va dastur tuzishga doir masalalar

1. O'lchami x bo'lgan qubik berilgan. Uning hajmini toping.
2. Radyuslari r1, r2, r3 bo'lgan 3 ta doira berilgan. Doiralarni yuzini hisoblang.
3. Yuzasi s va balandligi h bo'lgan uch burchag berilgan. Uni asosini toping.
4. Radyusi r bo'lgan sharing yuzini toping.
5. a,b va c tamonli uch burchag berilgan. Uch burchagning pirimetiri topilsin.
6. Asoslari avab, balandligi h bo'lgan o'layuzini toping.
7. Qo'nisni balandligi h va radyusi r bo'lsa uni hajmi nimaga teng bo'ladi.
8. h balandlikqan erkin tushayotgan jism qancha vaqtdan keyin erga uriladi.
9. Jo'mrakdan 1 s da 1 milli litr suv tomsa x yilda necha litr suv to-madi.
10. 1 dan n gacha sonlar berilgan. Berilgan sonlarni yig'indisini top-ing.

$$11. c1 = \frac{x + y}{y^2 + \left| \frac{y^2 + 2}{x + \frac{x^3}{5}} \right|} + e^{y+2} \quad x - \text{butun}, y - \text{haqiqiy}.$$

$$12. f1 = \frac{2tg(x + \frac{\pi}{6})}{\frac{1}{3} + \cos^2(y + x^2)} + \log_2^{(x^2+2)} \quad x, y - \text{haqiqiy}$$

$$13. f2 = \frac{1/(x + 2/x^2 + 3/x^3) + e^{x^2+3x}}{\arctg(x + y) + |5 + x|^2} - \cos^2(y^2 + x^2/2);$$

$$14. z = \ln \left| (x + y)^2 + \sqrt{|y| + 2} - (x - \frac{xy}{x^2/2 - 5}) \right| + \frac{\cos^2(x + y)}{(x + y)^{1/3}}; \quad x, y - \text{butun}$$

15. $T_{11} = \frac{x^2 + 1}{x^2 + \frac{xy + y^2}{y^2 + \frac{y + xy}{|xy| + 5}}} + \frac{1}{1 + \cos x + \frac{1}{\sin|x|}}$ x, y- haqiqiy
16. $P = \sqrt[3]{\frac{x(x^2 + 1)}{(x^2)}} + \lg(x - \cos y); x = 2,52; y = \frac{\pi}{6};$
17. $P = e^{\sqrt[3]{\lg 2x^2}} (2,5 + \arctg \sqrt{1 - x^2})^{-2}; x = 0,029$
18. $S = \sqrt{x^3 + y^3} (\sin^2 x + \cos e^3 y); x = 0,64; y = 0,32 \cdot 10^{-2}$
19. $a = \frac{y^{x+1}}{\sqrt[3]{y-2} + 3} + \frac{x + y/2}{2x + y}; b = (x-1)^{-1/\sin^2 z}; x = 1,625; y = 15,4; z = 0,255$
20. $a = \frac{\sqrt[3]{8 - x - y^2 + 1}}{x^2 + y^2 + 1}; b = e^{x-y} (tg^2 z + 1); x = 4,5; y = 0,007; z = 0,845$

5.3. Tarmoqlanuvchi jarayonlarga algoritm va dastur tuzishga doir masalalar

1. x, y(x va y teng emas) haqiqiy son berilgan. Ularning kichigini ularning yarim yig'indisi bilan, kattasini ularning ikkilangan ko'paytmasi bilan almashtiring.
2. Uchta haqiqiy son berilgan. Ularning musbatini kvadrati bilan almashtiring.
3. Agar o'zaro farqli x, y, z haqiqiy sonlar birdan kichik bo'lsa, u holda bu uchta sonidan eng kichigini boshka ikkitasining yarim yig'indisi bilan almashtiring. Aks holda o'zgarishsiz qoldiring.
4. a, b, c va d haqiqiy sonlar berilgan. Agar $a \leq b \leq c \leq d$ tengsizlik bajarilsa, u holda ularning har birini ularning kattasi bilan almashtiring.
5. x, y haqiqiy sonlar berilgan. Agar x va y manfiy bo'lsa, ularning har birini modullari bilan almashtiring; agar faqat bittasi manfiy bo'lsa ikkala sonning har birini 0.5 ga oshiring.
6. x, y, z haqiqiy musbat sonlar berilgan. x, y, z uzunlikka ega tomonli uchburchak mavjudmi?

7. $K_2 = \begin{cases} \sqrt{15a^2 - 21b^2}, & \text{agar } a > b \\ \sqrt{15b^2 - 21a^2}, & \text{agar } a \leq b \end{cases}$

8. $z = \begin{cases} (2x + y + 3)/3, & \text{agar } x + y > 0 \text{ va } x > 0 \\ \sqrt{x^2} + 2y^{2+3}, & \text{agar } x + y > 0 \text{ va } y < 0 \\ (x^2 + y^2)/x, & \text{qolgan hollarda} \end{cases}$

9. $Q = \begin{cases} \frac{a + b^2 + 2c^3}{a \cdot \sin^2 x^2}, & \text{agar } a > |b| \\ \frac{a^2 + 2ab + c^2}{a + 2c^2 + xy}, & \text{agar } xy > a \end{cases}$

$$10. \quad Z = \begin{cases} \sqrt[3]{cy^2 + a^3} - \ln x, & \text{agar } x > 0 \\ \frac{a^2 + bx + c^2}{\sqrt{a^3 - b^3}}, & \text{agar } a^3 > b^3 \end{cases}$$

$$11. \quad Z = \begin{cases} \log_3 x^2 + \sqrt[3]{x^3}, & \text{agar } x > 2 \\ \sqrt{x^2 + 16} - e^{2x}, & \text{agar } x = 2 \\ a^x + |x^3|, & \text{agar } x < 2 \end{cases}$$

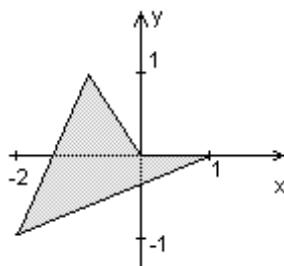
$$12. \quad S = \max(x, y, z) + \min(a, b, c)$$

$$13. \quad Z = \begin{cases} \sin \ln|x| + \sqrt{|x|^3 + 5}, & \text{agar } x < -3 \\ x^2 + e^{|x|} + \operatorname{tg} x, & \text{agar } x = -3 \\ 0,8 \cdot \sin x + 10^{-5,6}, & \text{agar } x > -3 \end{cases}$$

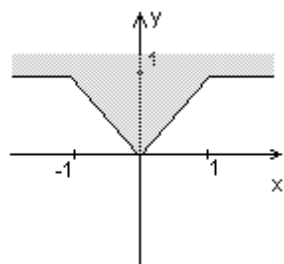
$$14. \quad Z = \begin{cases} \sqrt{a + \sin 2x} + ax^3 + b \log|2x|, & \text{agar } \sqrt{a-b} \leq x \\ \frac{\operatorname{arctg} 5x}{b \cdot \cos x + \ln a \cdot x}, & \text{agar } \sqrt{a-b} > x \end{cases}$$

$$15. \quad z = \begin{cases} \frac{1}{x} + \frac{1}{y} + \frac{1}{(x+y)}, & \text{agar } x \neq 0 \text{ va } y \neq 0 \\ \frac{1}{x} + \frac{1}{(x+1)}, & \text{agar } x \neq 0 \text{ va } y = 0 \\ \frac{1}{y} + \frac{1}{(y+1)}, & \text{agar } x = 0 \text{ va } y \neq 0 \end{cases}$$

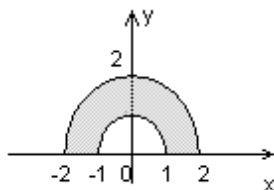
16. x, y haqiqiy sonlar berilgan. Koordinatalari (x, y) bo'lgan nuqta quyda keltirilgan rasmdagi tekislikning shtrixlangan qismiga tegishlimi? (ha/yo`q)



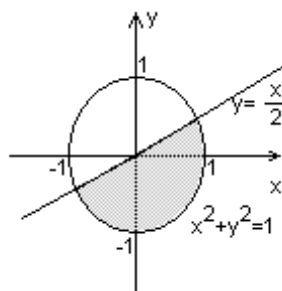
17. x, y haqiqiy sonlar berilgan. Koordinatalari (x, y) bo'lgan nuqta quyda keltirilgan rasmdagi tekislikning shtrixlangan qismiga tegishlimi? (ha/yo`q)



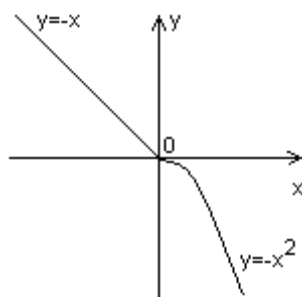
18. x, y haqiqiy sonlar berilgan. Koordinatalari (x, y) bo'lgan nuqta quyda keltirilgan rasmdagi tekislikning shtrixlangan qismiga tegishlimi? (ha/yo`q)



19. x, y haqiqiy sonlar berilgan. Koordinatalari (x, y) bo'lgan nuqta quyda keltirilgan rasmdagi tekislikning shtrixlangan qismiga tegishlimi? (ha/yo`q)



20. a haqiqiy son berilgan. Quydagi rasmlarda tasvirlangan $y(x)$ funksiya uchun $y(a)$ ni hisoblang.



5.4. Takrorlanuvchi jarayonlarga doir masalalar.

$$1. S = \frac{x-1}{1} - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \frac{(x-1)^4}{4} + \dots + \frac{(-1)^{n-1}(x-1)^n}{n}$$

$$2. P = 1 - \frac{x^2}{1} + \frac{x^4}{2} - \frac{x^6}{3} + \dots + \frac{(-1)^n x^{2n}}{n}$$

$$3. S = 1 - x^2 + x^4 - x^6 + \dots + (-1)^n \cdot x^{2n}$$

$$4. SS = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots + \frac{(-1)^{n-1} x^{2n-1}}{n}$$

5. $S = x + \frac{x^2}{2} + \frac{x^3}{3} + \dots + \frac{x^{2n-1}}{n}$ 9.
6. $y = \sqrt[3]{\frac{\sin ax + b^{2c}}{b^2 + \cos^2 x}} - \frac{\sin x^2}{ab}$, $c \leq x \leq d, n = 25$
7. $y = \sqrt[3]{\frac{ax + b}{b^2 + \cos^2 x}} - \frac{\sin x^2}{ab}$, $a \leq x \leq c, h = 0.3$
8. $y = \sqrt[3]{a^a + x^2 \cos ax}$, $-\frac{\pi}{2} \leq x \leq \pi; h = \frac{\pi}{19}, a = 3.26$
9. [a,b] oraliqdagi barcha tub sonlarini hosil qilish algoritmi va dasturini tuzing.
10. Berilgan N sonining raqamlari sonini hisoblash algoritmi va dasturini tuzing.
11. Berilgan N sonining raqamlari yig'indisi va ko'paytmasini hisoblash algoritmi va dasturini tuzing.
12. Berilgan N sonini tub ko'paytuvchilarga ajratish algoritmi va dasturini tuzing.
13. Barcha toq ikki xonali sonlarni hosil qilish algoritmi va dasturini tuzung.
14. Barcha 3 va 7 ga karrali bo'lgan uch xonali sonlar yig'indisini hisoblash algoritmi va dasturini tuzung.
15. $S = \sum_{n=1}^5 n^2 + \sum_{i=1}^{12} i^3$ ni hisoblang.
16. 1 dan n gacha toq sonlar kvadratlari yig'indisini hisoblang.
17. $P = \prod_{i=1}^{10} \frac{i^2 + ai}{i^2 + a^2} + \prod_{i=2}^4 \frac{k^3}{k + a}$ hisoblang.
18. $S = \sum_{n=1}^{10} 2^n + \prod_{k=1}^5 3^k + \sum_{m=1}^{10} 4^m$ ni hisoblash algoritmi va dasturini tuzing.
19. $S = 1 + \frac{4}{3} + \frac{9}{5} + \dots + \frac{n^2}{2n-1}$ ni hisoblang.
20. $y = n!!$ ($n > 0$) ni hisoblang.

5.5. Ichma-ich joylashgan sikllar bilan ishlashga doir masalalar.

1. $S = \sum_{n=1}^{10} \prod_{k=1}^5 (a^k + b^n)$ ni hisoblash algoritmini tuzing.
2. Ko'paytirish jadvali elementlarini hosil qilish algoritmini tuzing.
3. $s = \sum_{i=1}^3 \frac{(\sum_{j=1}^{10} j + \prod_{k=1}^4 \frac{1}{k})}{i!}$ ni hisoblash algoritmini tuzing.
4. $s = \prod_{k=1}^4 \sum_{j=1}^{10} \frac{(k! + j!)}{a + b}$ ni hisoblash algoritmini tuzing.

5. $S = \sum_{k=1}^3 \left(\prod_{n=1}^4 kn^2 + \sum_{i=1}^5 ki^3 \right)$ ni hisoblash algoritmini tuzing.
6. $S = \sum_{k=1}^3 \left(\prod_{n=1}^4 kn^2 + \sum_{i=1}^5 ki^3 \right)$ ni hisoblash algoritmini tuzing.
7. $S = \prod_{k=1}^4 \sum_{j=1}^{10} (k! + j!)$ ni hisoblash algoritmini tuzing.
8. $S = \sum_{n=1}^3 \prod_{k=1}^5 \sum_{j=1}^{10} (k^2 + j^2 + n)$ ni hisoblash algoritmini tuzing.
9. $S = \prod_{k=1}^5 \sum_{j=1}^{10} (k^2 + j^4)$ ni hisoblash algoritmini tuzing.
10. $S = \sum_{n=1}^5 \prod_{i=1}^8 (n^2 + i^3)$ ni hisoblash algoritmini tuzing.

5.6. Massiv elementlari bilan ishlashga doir masalalar.

1. A(n) massiv berilgan. Uning elementlarining eng kattasi va eng kichigini topish algoritmi va dasturini tuzing
2. A(n) massiv berilgan. Massiv elementlarini o'sib borish tartibida joylashtirish algoritmi va dasturini tuzing.
3. A(n) massiv berilgan. Uning elementlarining eng kattasini 1-soni bilan, eng kichigini esa -1 soni bilan almashtirish algoritmi va dasturini tuzing.
4. A(n) massiv berilgan. Uning elementlarining eng kattasi va eng kichigini hamda ularning o'rnini aniqlash algoritmi va dasturini tuzing.
5. A(n) massiv berilgan. Uning manfiy, musbat va nol elementlari sonini aniqlash algoritmi va dasturini tuzing.
6. $x_1, x_2, \dots, x_{55}$ massiv berilgan: $x_1(x_2+x_3) \cdot (x_4+x_5+x_6) \dots (x_{46}+x_{47}+\dots, x_{55})$ ni hisoblash algoritmi va dasturini tuzing.
7. $x_1, x_2, \dots, x_n$ massiv berilgan. Uning toq elementlaridan U, juft elementlaridan Z massiv hosil qilish algoritmi va dasturini tuzing.
8. A(n) massiv berilgan. Uning birinchi va oxirgi elementlari o'rnini almashtirish algoritmi va dasturini tuzing.
9. A(n) massiv berilgan. Uning manfiy elementlarining eng kattasini aniqlash algoritmi va dasturini tuzing.
10. A(n) massiv berilgan. Uning juft nomerdagi elementlari yig'indisi va toq nomerdagi elementlari ko'paytmasini hisoblash algoritmi va dasturini tuzing.

11. $A(N,N)$ massivning barcha musbat elementlari yig'indisi va manfiy elementlari ko'paytmasini hisoblash algoritmi va dasturini tuzing.

12. $A(N,N)$ massivning barcha musbat, manfiy va nol elementlari sonini aniqlash algoritmi va dasturini tuzing.

13. $A(N,N)$ massivning bosh diagonal elementlari yig'indisi va manfiy elementlari ko'paytmasini hisoblash algoritmi va dasturini tuzing.

14. $A(N, N)$ massivning bosh diagonalining eng katta elementi va u turgan ustunni aniqlash algoritmi va dasturini tuzing.

15. $A(N,M)$ massiv har bir ustuni elementlari yig'indisini, so'ngra har bir satri elementlari ko'paytmasini topish algoritmi va dasturini tuzing.

16. $A(N,M)$ massivning har bir satrining eng katta elementini toping va ularning indeks tartib raqamini aniqlash algoritmi va dasturini tuzing.

17. $A(N,M)$ massiv har bir satr elementlari yig'indisini hisoblash va ulardan eng kattasi va eng kichigini topish algoritmi va dasturini tuzing.

18. $A(3,3)$ massiv berilgan. Uning birinchi va uchinchi ustun elementlarini 3 ga ko'paytirib yangi massiv hosil qilish algoritmi va dasturini tuzing.

19. $A(N,N)$ massivning eng kichik elementi joylashgan satr va ustun elementlarini o'chirish algoritmi va dasturini tuzing.

20. $A(3,3)$ massiv berilgan. Uning satr va ustun elementlarini kvadratga kutarib yangi massiv hosil qilish algoritmi va dasturini tuzing.

21. Butun $N \times M$ ($2 \leq N, M < 100$) matrisa berilgan. Uning o'ng va chap diagonalini o'ng va chap qismini 1 bilan qolgan elementlarini esa 0 bilan to'ldiring.

Kiruvchi ma'lumotlar: Birinchi satrda N va M soni probel bilan berilgan.

Chiquvchi ma'lumotlar: $N \times M$ o'lchamli matrisani masala sharti bo'yicha chiqaring.

Kiritishga misol	Chiqarishga misol
3 3	000 101 0 0 0

5 5	000	0	0
	100 0 1		
	110 1 1		
	100 0 1		
	0	00 0 0	

22. Butun $N \times N$ ($3 \leq N < 100$) matrisa berilgan. Matrisaning ustun va satrini o'rtalarini tutashtirishdan hosil bo'lgan kvadratni 1 bilan qolgan qismini esa 0 bilan to'ldiring.

Kiruvchi ma'lumotlar: Birinchi satrda N soni berilgan.

Chiquvchi ma'lumotlar: $N \times N$ o'lchamli matrisani masala sharti bo'yicha chiqaring.

Kiritishga misol				Chiqarishga misol			
3				010			
				111			
				0 1 0			
5				001	0	0	
				011 1 0			
				111 1 1			
				011 1 0			
				0	01 0 0		

23. Butun $N \times N$ ($3 \leq N < 100$) matrisa berilgan. Matrisaning bosh diagonalidan yuqori elementlarini 1 bilan qolgan qismini esa 0 bilan to'ldiring.

Kiruvchi ma'lumotlar: Birinchi satrda N soni berilgan.

Chiquvchi ma'lumotlar: $N \times N$ o'lchamli matrisani masala sharti bo'yicha chiqaring.

Kiritishga misol	Chiqarishga misol
3	111 011 0 0 1
5	111 1 1 011 1 1 001 1 1 000 1 1 0 00 0 1

24. Butun $N \times N$ ($3 \leq N < 100$) matrisa berilgan. Matrisaning bosh diagonalidan pastgi elementlarini 1 bilan qolgan qismini esa 0 bilan to'ldiring.

Kiruvchi ma'lumotlar: Birinchi satrda N soni berilgan.

Chiquvchi ma'lumotlar: $N \times N$ o'lchamli matrisani masala sharti bo'yicha chiqaring.

Kiritishga misol	Chiqarishga misol
3	100 110 1 1 1
5	100 0 0 110 0 0 111 0 0 111 1 0 1 11 1 1

25. Butun $N \times N$ ($3 \leq N < 100$) matrisa berilgan. Matrisaning chap diagonalidan yuqori elementlarini 1 bilan qolgan qismini esa 0 bilan to'ldiring.

Kiruvchi ma'lumotlar: Birinchi satrda N soni berilgan.

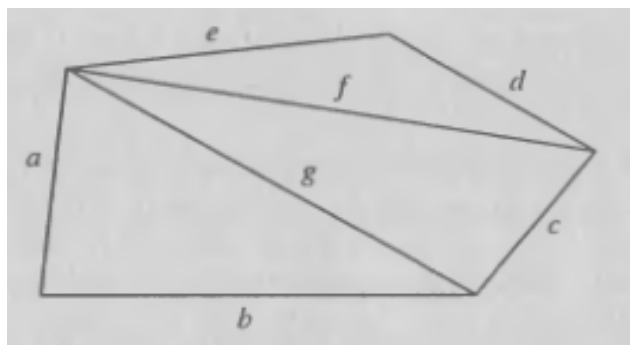
Chiquvchi ma'lumotlar: $N \times N$ o'lchamli matrisani masala sharti bo'yicha chiqaring.

Kiritishga misol	Chiqarishga misol
3	111 110 1 0 0
5	111 1 1 111 1 1 111 0 0 110 0 0 100 0 0

5.7.Funksiyadan foydalanib dastur tuzishga doir masalalar.

1. Sonni o'nlik sanoq sistemasidan o'n oltilik sanoq sistemasiga (hamda teskarisiga) o'tkazuvchi funksiya tuzing.
2. Sonni o'nlik sanoq sistemasidan sakkizlik oltilik sanoq sistemasiga o'tkazuvchi funksiya tuzing.

3. Sonni o'nlik sanoq sistemasidan ikkilik sanoq sistemasiga o'tkazuvchi funksiya tuzing.
4. Sonni o'n oltilik sanoq sistemasidan ikkilik sanoq sistemasiga o'tkazuvchi funksiya tuzing.
5. Sonni o'n oltilik sanoq sistemasidan sakkizlik sanoq sistemasiga o'tkazuvchi funksiya tuzing.
6. Sonni sakkizlik sanoq sistemasidan ikkilik sanoq sistemasiga o'tkazuvchi funksiya tuzing.
7. Q sonini P darajasini topuvchi funksiya tuzing.
8. P sonini oxiridan L sonini qo'shuvchi funksiya tuzing.
9. Berilgan to'g'ri burchakli uchburchakning katetlari yordamida gipotenuzasini topuvchi funksiya tuzing.
10. Berilgan ikki nuqtaning koordinatasi asosida ular orasidagi masofani topuvchi funksiya tuzing.
11. Berilgan sondagi qo'shni raqamlarining raqamlarining yig'indisiga teng bo'lgan raqamni o'chiruvchi funksiya tuzing.
12. Berilgan N soni ikkita tub sonni yig'indisi bo'lishini tekshiruvchi funksiya tuzing.
13. Berilgan son 11 ga bo'linsa 1 aks holda o'chiqaruvchi funksiya tuzing.
14. Berilgan sonni K o'rindagi raqamini N o'rindagi raqami bilan almashtiruvchi funksiya tuzing.
15. Sonni raqamlarini o'sish (kamayish) tartibida saralovchi funksiya tuzing.
16. Uchburchakni uchta uchining koordinatalari berilgan. Uning yuzasini topuvchi funksiya tuzing.
17. So'z berilgan. Shu so'zni palindromlikka tekshiruvchi funksiya tuzing. Masalan, kiyik. So'z palindrom bo'lsa "Palindrom" aks holda "Palindrom emas" so'zini chiqaring.
18. . Besh burchakning uchlarining koordinatalari $x_1, y_1, x_2, y_2, \dots, x_5, y_5$ berilgan. Beshburchakdagi uchburchakni uchta uchining koordinatalari berilgan. Uning yuzasini topuvchi funksiya tuzing.



19. Sonni tublikka tekshiruvchi funksiya tuzing va barcha uch xonali tub sonlarni chiqaring.
20. Barcha tub sonlar ichidan “egizak”larini topuvchi funksiya tuzing. Tub sonlar “egizak” deyiladi, agarda ular 3 ga farq qilsa. Masalan, 41 va 43. Uch xonali sonlar ichidan barcha “egizak”larni chiqaring.
21. Ikkita son berilgan. Ularni raqamlari yig‘indisi kattasini toping. Sonni raqamlari yig‘indisini topuvchi funksiya tuzing.
22. Ikkita son berilgan. Ularning raqamlaridan soni ko‘pini toping. Sonni raqamlari sonini topuvchi funksiya tuzing.
23. 6 xonali barcha baxtli sonni toping. Agarda 6 xonali sonni dastlabki uchta raqamini yig‘indisi oxirgi uchta raqamini yig‘indisiga teng bo‘lsa baxtli son deyiladi. 6 xonali sonni raqamlarini uchta lab yig‘indisini topuvchi funksiya tuzing.
24. Berilgan sonni palindromlikka tekshiruvchi funksiya tuzing. Masalan, 1221. Son palindrom bo‘lsa “Palindrom”, aks holda “Palindrom emas” so‘zini chiqaring.
25. Berilgan a va b sonlarini EKUBini topuvchi funksiya tuzing.
26. Berilgan a va b sonlarini EKUKini topuvchi funksiya tuzing.
27. Berilgan a, b va c sonlarini EKUBini topuvchi funksiya tuzing.
28. Berilgan y (yil), o (oy) va k (kun) sonlari berilgan. Shu sonlarni kiritgan holda qaysi kunga to‘g‘ri kelishini aniqlovchi dastur tuzing. Masalan, 2019 8 2 uchun “Juma” chiqishi lozim.
29. 0110100110010110 tartibda berilgan ketma–ketligining n – o‘rnida necha soni turganligini topuvchi dastur tuzing. Bu ketma–ketlik birinchi elementi 0 ga teng. Keyingi elementlari esa berilgan satrni 0 ni 1 ga 1 ni esa 0 ga o‘g‘irgan holatga ko‘chirilgan ya’ni 0 1 10 1001 10010110
30. 2^{2^n} ning 10^9 ga bo‘lgandagi qoldiqni hisoblovchi rekursiv funksiya tuzing. n soni $[1; 1000000]$ oraliqda bo‘lishi mumkin.

5.8. Ko'rsatgich funksiyalardan foydalanib dastur tuzish

1. $a[n][m]$ massiv elementlarini $[0; 10000]$ oraliqdagi sonlar bilan to'ldiring. Bu massivning har bir satr elementlarini o'sish tartibida chiqaring.
2. $a[n][m]$ massiv elementlarini $[0; 10000]$ oraliqdagi sonlar bilan to'ldiring. Uning eng katta va eng kichik elementini toping. Agarda bunday elementlar ko'p bo'lsa, ularning sonini chiqaring.
3. $a[n][m]$ massiv elementlarini $[-1000; 1000]$ oraliqdagi tasodifiy sonlar bilan to'ldiring. Uning eng katta va eng kichik elementlarini nomer(indeks)larini toping.
4. $a[n][m]$ massiv elementlarini $[-10000; 10000]$ oraliqdagi tasodifiy sonlar bilan to'ldiring. Uning eng katta va eng kichik elementlarining ayirmasini toping.
5. $a[n][m]$ massiv elementlarini $[-500; 500]$ oraliqdagi tasodifiy sonlar bilan to'ldiring. Bu massivni ustun elementlarini o'sish tartibida chiqaring.
6. $a[n][m]$ massiv elementlarini $[-900; 900]$ oraliqdagi tasodifiy sonlar bilan to'ldiring. Bu massivni bosh diagonal elementlarini o'sish tartibida chiqaring. Albatta mos ravishda satrlari ham o'zgarishi lozim.
7. $a[n][m]$ massiv elementlarini $[0; 100000]$ oraliqdagi tasodifiy sonlar bilan to'ldiring. Elementini qiymati 200 dan kattalari sonini va ularning o'rta arifmetigini toping.
8. $a[n][m]$ massiv elementlarini $[-1000; 1000]$ oraliqdagi tasodifiy sonlar bilan to'ldiring. Bu massivning bosh diagonal elementlarini kamayish tartibida chiqaring.
9. $a[n][m]$ massiv elementlarini $[-100000; 100000]$ oraliqdagi tasodifiy sonlar bilan to'ldiring. Elementining qiymatini bir vaqtda 5 ga va 3 ga qoldiqsiz bo'linadiganlarining soni va ularni yig'indisini toping.
10. $a[n][m]$ massiv elementlarini $[-10000; 10000]$ oraliqdagi tasodifiy sonlar bilan to'ldiring. Massivni chap diagonal elementlarini kamayish tartibida chiqaring.
11. $a[n][m]$ massiv elementlarini $[-10000; 10000]$ oraliqdagi tasodifiy sonlar bilan to'ldiring. Massiv elementlari tublari sonini va ularning yig'indisini chiqaring.
12. $a[n][m]$ massiv elementlarini $[-10000; 10000]$ oraliqdagi tasodifiy sonlar bilan to'ldiring. Massivning eng katta va eng kichik elementlari sonini chiqaring.

13. $a[n][m]$ massiv elementlarini $[-1000; 1000]$ oraliqdagi tasodifiy sonlar bilan to'ldiring. Massivning eng oxirgi manfiy elementining nomerini aniqlang va chiqaring.
14. $a[n][m]$ massiv elementlarini $[-1000; 1000]$ oraliqdagi tasodifiy sonlar bilan to'ldiring. Massivning eng oxirgi musbat elementining nomerini aniqlang va chiqaring.
15. $a[n][m]$ massiv elementlarini $[-1000; 1000]$ oraliqdagi tasodifiy sonlar bilan to'ldiring. Massivning eng oxirgi manfiy elementining nomerini aniqlang va chiqaring.
16. $a[n][m]$ massiv elementlarini $[-1000000; 100000]$ oraliqdagi tasodifiy sonlar bilan to'ldiring. Massivning musbat va manfiy elementlarining yig'indisini toping.
17. $a[n][m]$ massiv elementlarini $[-1000; 1000]$ oraliqdagi tasodifiy sonlar bilan to'ldiring. Massivning eng oxirgi manfiy elementining nomerini aniqlang va chiqaring.
18. $a[n][m]$ massiv elementlarini $[-10000; 10000]$ oraliqdagi tasodifiy sonlar bilan to'ldiring. Massivning eng oxirgi manfiy elementining nomerini aniqlang va chiqaring.
19. $a[n][m]$ massiv elementlarini $[-1000; 1000]$ oraliqdagi tasodifiy sonlar bilan to'ldiring. Berilgan massiv elementlarining a va b oraliqdagilari sonini toping.
20. $a[n][m]$ massiv elementlarini $[-1000; 1000]$ oraliqdagi tasodifiy sonlar bilan to'ldiring. Massivni manfiy elementlarini ularning absolyut qiymati bilan almashtiring va almashtirishlar sonini chiqaring.

5.9. Murakkab darajadagi masalalarga algoritm va dastur tuzing

1. **«Ko'ndalang diagonal»**. $A(m,n)$ massivning, indekslari ayirmasi berilgan K soniga (K -manfiy son ham bo'lishi mumkin) teng bo'lgan ($i-j=k$), elementlari yig'indisini toping.
2. **«Kvadratchalar»**. Har bir elementi 0,1,5 yoki 11 ga teng $A(m, m)$ massiv berilgan. Har birida elementlari har xil bo'lgan to'rtliklar $A(i,j)$, $A(i+1,j)$, $A(i,j+1)$, $A(i+1,j+1)$ miqdorini toping.
3. **«Sanoq sistemalari»**. $M(9)$ massivda qandaydir natural sonning I – sanoq sistemasida raqamlari yozilgan. ($M(1)$ -birlar xonasi va h.k.). Birlar xonasidan boshlab turib, J -sanoq sistemasida bu son raqamlarini chop eting. I, J sonlar 10 dan oshmaydi.

4. «**Kalendar**». Kun, oy, yilni bildiruvchi uchta a, b, c sonlar berilgan. Shu kunning yil boshidan hisoblangandagi n -tartib raqamini toping. Ko'rsatma: tartib raqami 400 ga bo'linadigan, shuningdek tartib raqami 4 ga bo'linib, 100 ga bo'linmaydigan yillar kabi yili hisoblanadi.
5. «**So'zning matnga kiritilishi**». Butun sonli $X(n)$ va $Y(k)$ 2 ta massivlar berilgan. $x_{i+1}=u_1, x_{i+2}=u_2, \dots, x_{i+k}=u_k$ shartlarni qanoatlantiruvchi va $x_{i+1}, x_{i+2}, \dots, x_{i+k}$ ketma-ket keluvchi « k » ta elementlarni birinchi massivdan tanlab olish mumkinmi? Bu masalani yechuvchi va Ha yoki Yo'q javoblarni beruvchi dasturni yozing.
6. «**Nollar seriyasi**». $A(n)$ butun qiymatli massiv berilgan. Massivning elementlari nolga teng va ketma-ket keluvchi eng uzun ketma-ketligi uzunligini toping.
7. «**Egar nuqta**». $A(m, n)$ sonli massiv berilgan. O'z satrida eng kichik bo'lgan element o'z ustunida eng katta bo'lsa, u egar nuqta deyiladi. Agar massivda egar nuqta bo'lsa, u yotgan satr va ustun raqamini, agar unday nuqta bo'lmasa, nol sonini chop eting.
8. «**Kasrni qisqartirish**». Natural « m » va « n » sonlar berilgan. Umumiy bo'luvchiga ega bo'lmagan shunday natural m_1 va n_1 sonlarni topish kerakki, $m_1/n_1 = m/n$ bo'lsin.
9. «**Massivlarning qo'shilishi**». « M » va « N » sonlar va ikkita tartibga solingan massivlar: $a_1 \leq a_2 \leq \dots \leq a_m$ hamda $v_1 \leq v_2 \leq \dots \leq v_n$ berilgan. Bu elementlardan tartibga solingan uchinchi massivni hosil qiling: $s_1 \leq s_2 \leq \dots \leq s_{m+n}$ Ko'rsatma: « M » va « N » lar katta sonlar bo'lganda dasturdagi amallar miqdoriga e'tibor bering.
10. «**Moda**». Butun sonli $A(n)$ massivda eng ko'p uchraydigan sonni toping. Agar bunday sonlar bir nechta bo'lsa, ulardan bittasini aniqlang.
11. «**Markaziy qishloq**». « K » ta qishloq mavjud. Agar i -qishloqda tez yordam punkti joylashtirilsa, chaqiruv bo'yicha j -qishloqqa borish $a_{ii} + a_{ij}$ ($1 \leq i, j \leq k, i \neq j$) vaqtni oladi.. Shunday i -qishloq tartib raqamini topingki, undan eng uzoq qishloqqa borish uchun kam vaqt sarflansin. $A(k, k)$ massiv berilgan. Unda hamma a_{ij} elementlar noldan katta va a_{ij} - element a_{ji} elementga teng bo'lmasligi mumkin.
12. «**Tartib indeksleri**». $A(n)$ sonli massiv berilgan. $1, 2, \dots, n$ sonlarning shunday, $i_1, i_2, \dots, i_n$ o'rin almashtirishni topingki, natijada $a_{i_1} \leq a_{i_2} \leq \dots \leq a_{i_n}$ bo'lsin.
13. «**Nollashtirish**». Berilgan ikki o'lchovli $A(m, n)$ massivda noli bo'lgan satr va ustun elementlarini nollar bilan almashtiring.

Shart. Yordamchi bir o'lchovli massivdan foydalanish mumkin, lekin yordamchi ikki o'lchovli massiv ishlatish mumkin emas .

14. **«Ulgurji xarid»**. Paypoq jufti 105 so'm, bog'lami (12 juft) 1025 so'm, qutisi (12 bog'lam) 11400 so'm turadi.

Xaridor sotib olmoqchi bo'lgan paypoqlarning n juft soniga ko'ra, xaridor sotib olishi kerak bo'lgan n_1, n_2, n_3 quti, bog'lam, paypoqlar juftini hisoblab beruvchi dastur tuzing.

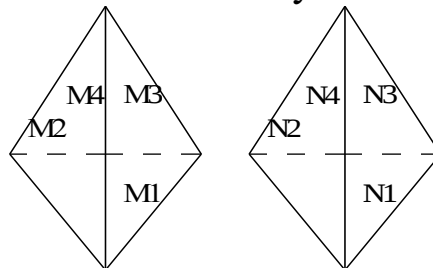
Tushuntirish. 11 juft paypoq o'rniga bir bog'lamni xarid qilish arzonga tushadi.

15. **«To'ntarilgan sonlar»**. $A(N)$ sonli massiv berilgan. Massivning maksimal uzunlikdagi kesmasini topish kerak. Unda birinchi son oxirgisiga, ikkinchi son oxirgisidan bitta oldingisiga va h.k. teng bo'lsin . Bu kesma uzunligini chop eting.

16. **«Ikki marta monoton»**. Sonlarning $A(m,n)$ massivi satrlar va ustunlar bo'yicha kamayib borish tarzida tartiblangan, ya'ni hamma $i=1,...,m$ lar uchun, $a_{i1} \leq a_{i2} \leq ...$ hamma $j=1,..., n$ lar uchun, $a_{1j} \leq a_{2j} \leq ...$. Massiv elementlari ichidan berilgan «x» soniga teng bo'lganini toping. Agar bunday element bo'lmasa, «Yo'q » deb chop eting. Majburiy shart. Yechimda amallar soni $m \cdot n$ atrofida emas, $m+n$ atrofida bo'lsin.

17. **«Tetraedrlar»**. Ikkita teng to'g'ri M va N tetraedrlar qirrasiga M_1, M_2, M_3, M_4 va N_1, N_2, N_3, N_4 sonlar rasmda ko'rsatilgan tartibda yozilgan.

Tetraedrlarni bir xil sonlar yozilgan qirralari bilan bir-biriga mos tushirib, joylashtirish mumkinmi? «Ha» yoki «Yo'q» javob berilsin .



18. **«Kasr davri»**. « M » va « N » natural sonlar berilgan. O'nlik M/N kasr davrini chop eting. Masalan, $1/7$ kasr uchun davr 142857 ga teng, kasr chekli bo'lsa, uning davri bitta 0 raqamiga teng bo'ladi.

19. **«Arra»**. $X(m)$ massiv berilgan. Eng uzun «arra» shaklidagi (tishlari yuqoriga qaragan) ketma-ket keluvchi sonlar qatori uzunligini toping.
 $X[p+1] < X[p+2] > X[p+3] < ... > X[p+k]$

20. **«Nollar oxiriga»**. Bir o'lchovli massiv berilgan. Uning nolga teng bo'lmagan hamma elementlarini, tartibni saqlagan holda, massiv boshi-

lan berilgan. Unda agar (k,m) katakdan o'tish mumkin bo'lsa, $a[k,m]=0$, (k,m) katakdan o'tish mumkin bo'lmasa, $a[k,m]=1$. Yo'lovchining boshlang'ich nuqtasi, o'tish mumkin bo'lgan, $[i,j]$ katakda. Yo'lovchi, agar kataklar bitta umumiy tomonga ega bo'lsa, bir katakdan ikkinchisiga o'ta oladi. Yo'lovchi chegara katakka (ya'ni (k,m) , bu yerda «k» va «m» 1 yoki 40 ga teng) yetsagina labirintdan chiqadi.

30. «**Mukammal sonlar**». Agar natural son o'zining barcha bo'luvchilari (1 ham hisoblanadi) yig'indisiga teng bo'lsa, u mukammal son bo'ladi. Berilgan M sonidan kichik barcha mukammal sonlarni topish dasturini tuzish.

31. «**Hosil qilinmaydigan son**». $R(n)$ natural sonlar massivi berilgan. Shunday eng kichik natural sonni topish kerakki, uni R massivning hech bir elementlari yig'indisi ko'rinishida ifodalab bo'lmasin. Yig'indi bitta qo'shiluvchidan ham iborat bo'lishi mumkin, lekin massivning har bir elementi unga faqat bir marta kira oladi.

32. «**Do'mbira**». Aylana bo'yicha 12 ta son yozilgan : $a_1, a_2, \dots, a_{12},$. Agar k-tartib raqamidan boshlab, ularni qo'shib yozsak, X_k vektor hosil bo'ladi: $X_k = (a_k, a_{k+1}, a_{k+11})$, bu yerda a_{13} ni a_1 , a_{14} ni a_2 va h.k. bildiradi. Agar birinchi tengmas juftdayoq $a_{k+1} < a_{p+i}$ ($i=0,1,\dots$) bo'lsa, X_k vektor X_r vektordan kichik deb hisoblanadi. Shunday «k» ni topish kerakki, vektor X_k eng kichik bo'lsin.

33. «**Kublar yig'indisi**». Berilgan N natural sonni 2ta natural sonlar kublarining yig'indisi ($N=i^3+j^3$) ko'rinishida necha xil usul bilan yozish mumkin? Qo'shiluvchilarning o'rin almashtirilishi yangi usulni bermaydi. 1/3-darajaga ko'tarish amalidan foydalanish mumkin emas.

34. «**Raqamlarning berilgan yig'indisi**». Raqamlari yig'indisi berilgan natural songa teng bo'lgan barcha uch xonalik o'nlik sanoq sistemasidagi sonlarni aniqlang.

35. «**Poker**». Beshta sondan iborat massiv berilgan. Ularning ichidan beshtasi bir xil bo'lsa-1; to'rttasi bir xil bo'lsa-2; uchta va ikkitasi bir xil bo'lsa-3; uchta bir xil bo'lsa-4; ikkita ikkitalik bir xil bo'lsa-5; ikkitasi bir xil bo'lsa-2; aks holda-7 sonini chop eting.

36. «**Kvadratlar yig'indisi**». Berilgan M natural sonni ikkita natural sonlar kvadratlari yig'indisi ko'rinishida yozish mumkinmi?

VI-bob. Nazariy va amaliy bilimlarni mustahkamkashga doir test savollari.

6.1. Algoritm va algoritmlash jarayoniga doir test savollari

1. Algoritmning xossalarini toping

- a) chiziqli,tarmoqlanuvchi,takrorlanuvchi
- b) aniqlanganlik, tushunarlilik, diskretlik, ommaviylik,natijaviylik
- c) aniqlanganlik, tushunarlilik, diskretlik, tarmoqlanuvchi,takrorlanuvchi
- d) chiziqli,tarmoqlanuvchi, diskretlik, ommaviylik,natijaviylik

2. Algoritm bu:

- a) amallar ketma-ketligi
- b) chekli amallar ketma-ketligi
- c) buyruqlar va amallar ketma-ketligi
- d) a),b),c)

3. Algoritmning berilish usullari qanday

- a) matn orqali
- b) formulalar orqali
- c) jadval ko`rinishida
- d) matn, formula, jadval ,dastur, algoritmik til,blok-sxema

4. Algoritm turlarini ko`rsating

- a) chiziqli, tarmoqlanuvchi,takrorlanuvchi
- b) chiziqsiz, tarmoqlanuvchi,takrorlanuvchi
- c) tarmoqlanuvchi
- d) takrorlanuvchi

5. blok qanday ko'rsatmani bildiradi?

- a) Shart tekshiradi
- b) Hisoblash bajarishni
- c) Boshlanishni
- d) Qiymat kiritishni

6. Quyidagilardan qaysi biri algoritm ko'satmasi bo'la olmaydi?

- a) "Sakrash mumkin"
- b) "2 karra 3 necha bo'ladi?"
- c) "Tugallansin"
- d) a va b

7. blok qanday ko'rsatmani bildiradi?

- a) Shart tekshiradi
- b) Oddiy harakatni
- c) Boshlanishni

d) Qiymat kiritishni

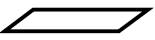
8.  blok qanday ko'rsatmani bildiradi?

a) Shart tekshiradi

b) Oddiy harakatni

c) Boshlanishni

d) Qiymat kiritishni

9.  blok qanday ko'rsatmani bildiradi?

a) Shart tekshiradi

b) Oddiy harakatni

c) Boshlanishni

d) To'g'ri javob yo'q

10. Algoritm so'zi qaysi olim nomi bilan bog'liq

a) Abu Rayxon Beruniy

b) Al-Xorazmiy

c) Ibn Sino

d) Mirzo Ulug'bek

11. Algoritm javobini aniqlang :

1. Boshlash.

2. $x:=3$; $y:=2$; $x:=\text{sqrt}(y)*2$ hisoblansin.

3. $x:=\text{sqr}(x)+y$ hisoblansin

4. x va y chiqarilsin.

5. Tamom.

a) 8 va 2 b) 2 va 8 c) 10 va 2 d) 2 va 10

12. Algoritm javobini aniqlang :

1. Boshlash.

2. $a:=1$; $b:=2$; $a:=3$ bajarilsin

3. a , b va a ni qiymati chiqarilsin

4. Tamom.

a) 1;2;3 b) 1;1;3 c) 2;1;3 d) 3;2;3

13. Algoritm javobini aniqlang

1. Boshlash.

2. $x:=1$; $y:=1$; $z:=0$ hisoblansin

3. agar $x>1$ va $y>0$ bo'lsa $z:=1$ va 5 ga o'tilsin

4. aks holda $z:=2$

5. z ni qiymati chiqarilsin

6. Tamom.

a) 1 b) 2 c) 0 d) -1

14. f ning qiymatini aniqlang

1. Boshlash.

2. $i:=2$ bajarilsin
3. $i:=i+1$ hisoblansin.
4. $f:=i*i$ hisoblansin
5. agar $i>3$ bo'lsa $f:=f^2$ hisoblansin va 7 ga o'tilsin.
6. aks holda $f:=f^{1/2}$
7. f ni qiymati chiqarilsin.

a) 2 b) 3 c) 36 d) 81

15. Algoritm natijasini aniqlang?

1. Boshlash.
2. $A:=10$; $B:=8$; $S:=6$ berilsin.
3. agar $A>B$ va $S>B$ bo'lsa $y:=A+B-S$ hisoblansin va 5 ga o'tilsin.
4. Aks holda $y:=A-B+S$ hisoblansin.
5. y ning qiymati chiqarilsin.
6. Tamom.

a) 12 b) 8 c) 10 d) 14

16. Algoritm natijasini aniqlang?

1. Boshlash.
2. $a:=12$; $b:=14$; $s:=10$ deb olinsin.;
3. agar $a>b$ yoki $b>s$ bo'lsay $:=a+b-s$ hisoblansin va 5 ga o'tilsin.
4. $y:=a-b+s$ hisoblansin,
5. Y ning qiymati chiqarilsin.
6. Tamom.

a) 16 b) 14 c) 8 d) 6

17. Algoritm natijasini aniqlang.

1. Boshlash.
2. $A:=22$; $B:=18$; $D:=20$ deb hisoblansin.
3. agar $A>18$ bo'lsa $D:=A-B$ hisoblansin.
4. D ning qiymati chop etilsin.
5. Tamom.

a) 5 b) 20 c) 8 d) 4

18. Algoritm natijasini aniqlang:

1. Boshlash.
2. $X:=2$; $p:=1$ deb olinsin.
3. $P:=P*(2*x-2)$ hisoblansin
4. $X:=X+3$ bajarilsin
5. agar $X\leq 6$ bo'lsa 3 ga o'tilsin;
6. Aks holda P ning qiymati chiqsin.
7. Tamom

a) 20 b) 16 c) 2 d) 24

19. Algoritm natijasi nimaga teng?

1. Boshlash.
 2. $X:=-2; y:=4; A:=-2$ deb olinsin.
 3. agar $A*A-y>x*x$ bo'lsa $z:=y+A$ hisoblansin va 5 ga o'tilsin.
 4. Aks holda $z:=X+A$ bajarilsin.
 5. A ni qiymati chiqarilsin.
 6. Tamom.
- a) -5 b) 1 c) 7 d) -4

20. Algoritm natijasi nimaga teng?

1. Boshlash.
 2. $k:=0; x:=1$ deb olinsin.
 3. $k:=k+1$ hisoblansin
 4. $x:=x*k$ bajarilsin.
 5. agar $k<5$ bo'lsa 3 ga o'tilsin;
 6. k va x ning qiymati chiqarilsin.
 7. Tamom.
- a) 4;24 b) 4;120 c) 4;12 d) 5; 120

6.2. C++ dasturlash tiliga doir test savollari

1. Interpretator qanday ishlaydi

- a) Interpretator dasturni o'qish jarayonida uning komandalarini ketma - ket mashina tiliga o'tkazadi.
- b) Interpretator yaxlit programma kodini biror bir oraliq forma obekt fayliga o'tkazadi.
- c) Faqatgina kompilyasiya qiladi.
- d) To'g'ri javob berilmagan.

2. Kompilyator qanday ishlaydi

- a) Kompilyator dasturni o'qish jarayonida uning komandalarini ketma-ket mashina tiliga o'tkazadi.
- b) Kompilyator yaxlit programma kodini biror bir oraliq forma - obyekt fayliga o'tkazadi.
- c) Faqatgina kompilyasiya qiladi.
- d) To'g'ri javob berilmagan.

3. Strukturaviy dasturlashning asosiy g'oyasi ...

- a) Bo'laklavahukmronlikqil.
- b) Dasturning yaxlitligi
- c) Dasturning mustahkamligi
- d) To'g'ri javob berilmagan.

4. Obyektga mo'ljallangan dasturlashning asosiy maqsadi ...

- a) Dasturni bir qancha bo'laklarga bo'lish.

b) Berilganlar va ular ustida amal bajaruvchi proseduralarni yagona obyekt deb qarash

c) Berilganlar alohida, ular ustida amal bajaruvchi prosedura va funksiyalar alohida faoliyat ko'rsatadi.

d) To'g'ri javob berilmagan.

5. Obyektga mo'ljallangan dasturlashning uch asosiy prinsiplari ...

a) Inkapsulyasiya, Dekapsulyasiya, Polimorfizm

b) Inkapsulyasiya, Merosxo'rlik, Polimorfizm

c) Induksiya, Deduksiya, Merosxo'rlik

d) To'g'ri javob keltirilmagan.

6. Inkapsulyasiya bu - ...

a) Tashqi konstruksiyada qo'llaniladigan yashirinlik yoki obyektning avtonomligi xossasi inkapsulyasiya deyiladi.

b) Yangi berilganlar tipi (sinf) oldindan mavjud bo'lgan sinfni kengaytirishdan hosil bo'lishidir.

c) shaklning ko'p xilligidir.

d) To'g'ri javob keltirilmagan.

7. Dekrement nima

a) Qiymatni 1 ga oshirish ($a+=1$) b) Qiymatni 1 ga kamytirish ($a-=1$)

c) Qiymat berish ($a=1$) d) to'g'ri javob keltirilmagan.

8. Prefiks operatori qachon bajariladi

a) qiymat o'zlashtirilguncha ($++Age$, $--Age$)

b) qiymat o'zlashtirilgandan keyin ($Age++$, $Age--$) c) Farqi yo'q.

d) To'g'ri javob keltirilmagan.

9. postfiks operatori qachon bajariladi

a) qiymat o'zlashtirilguncha ($++Age$, $--Age$)

b) qiymat o'zlashtirilgandan keyin ($Age++$, $Age--$)

c) Farqi yo'q.

d) To'g'ri javob keltirilmagan.

10. Butun sonli arifmetika ishlatiladigan amalni ko'rsating:

A) +

B) %

C) &&

D) **

E) !!

11. Ishorasiz butun turdagi ma'lumotni tavsiflash uchun ma'lumot turini ko'rsating:

A) long

B) short

C) char

D) unsigned int

E)

double

12. Ishorasiz ma'lumotlarni tavsiflashning ma'lumotlar turi:

A) double

B) float

C) unsigned

D) char

E) int

13. O'zgarmaslarni aniqlash uchun preprosessorning qaysi direktivasidan foydalaniladi?

A) #include B) #define C) typedef D) union E) extern

14. Belgili o'zgarmas to'g'ri tavsiflanganini aniqlang

A) define S='A' B) #define S "A" C) #define S A D) #define S 'A'
E) define S="A"

15. Satrli o'zgarmas to'g'ri tavsiflanganini aniqlang

A) #define S 'stroka' B) define S="stroka" C) #define S='stroka'
D) define S stroke E) #define S "stroka"

16. C++ tilida ishorali miqdorlarni ko'rsatishda qaysi kalit so'zidan foydalaniladi

a) short b) long c) Unsigned d) To'g'ri javob keltirilmagan.

17. C++ tilida necha turdagi o'zgarmaslar aniqlangan

a) Ikki turdagi - literal va belgili
b) Uch turdagi – short, long va Unsigned
c) Ikki turdagi - oddiy va murakkab
d) To'g'ri javob keltirilmagan.

18. C++ tilida belgili o'zgarmasni aniqlashning necha usuli mavjud

a) # define direktivasi yordamida va const kalitli so'zi orqali
b) # definedirektivasi yordamida
c) const kalitli so'zi orqali
d) to'g'ri javob keltirilmagan.

19. C++ tilida bajariluvchi faylni hosil qilish uchun dastlab qanday amalni bajarish lozim

a) cpp kengaytirmali dastur kodi hosil qilinadi;
b) Dastur kodini kompilyasiya qilish orqali .obj kengaytmali obyektli fayl tuziladi;
c) Bajariluvchi faylni hosil qilish maqsadida .obj kengaytmali fayli zaruriy bibliotekalar orqali komponovka qilinadi;
d) To'g'ri javob keltirilmagan.

20. Inkrement nima

a) Qiymatni 1 ga oshirish (a+=1)
b) Qiymatni 1 ga kamytirish (a-=1)
c) Qiymat berish (a=1)
d) to'g'ri javob keltirilmagan.

21. Ifoda to'g'ri yozilgan satrni ko'rsating: $b = \frac{1 + \cos(y - 2)}{\frac{x^4}{2} + \sin^2 z}$

A) $b = 1 + \cos(y - 2) / (pow(x,4) / 2 + pow(\sin(z),2));$
B) $b = (1 + \cos(y - 2)) / pow(x,4) / 2 + pow(\sin(z),2);$
C) $b = (1 + \cos(y - 2)) / (pow(x,4) / 2 + pow(\sin(z),2));$

D) $b = (1 + \cos(y - 2)) / (\text{pow}(x, 4) / 2 + \sin(\text{pow}(z, 2)))$;

E) $b = (1 + \cos(y - 2)) / (\text{pow}(4, x) / 2 + \text{pow}(\sin(z), 2))$;

22. Ifoda to'g'ri yozilgan satrni ko'rsating: $y = \ln|5x| + \text{tg}x^3$

A) $y = \log(\text{abs}(5 * x)) + \tan(\text{pow}(x, 3))$;

B) $y = \ln(\text{abs}(5x) + \text{pow}(\text{tg}(x), 3))$;

C) $y = \log(\text{abs}(5 * x)) + \tan(\text{pow}(3, x))$;

D) $y = \ln(\text{abs}(5 * x)) + \tan(\text{pow}(x, 3))$;

E) $y = \log(\text{abs}(5 * x)) + \text{tg}(\text{pow}(x, 3))$;

23. Ifoda to'g'ri yozilgan satrni ko'rsating: $a = \frac{\sqrt{|x-1|} - \sqrt[3]{|y|}}{1 + \frac{x^2}{2} + \frac{y^2}{4}}$

A) $a = \text{sqr}(\text{abs}(x - 1)) - \text{pow}(\text{abs}(y), 1/3) / (1 + \text{pow}(x, 2) / 2 + \text{pow}(y, 2) / 4)$;

B) $a = (\text{sqr}(\text{abs}(x - 1)) - \text{sqr}(\text{abs}(y))) / (1 + \text{pow}(x, 2) / 2 + \text{pow}(y, 2) / 4)$;

C) $a = (\text{sqr}(\text{abs}(x - 1)) - \text{pow}(\text{abs}(y), 1/3)) / (1 + \text{sqr}(x) / 2 + \text{sqr}(y) / 4)$;

D) $a = \text{sqr}(\text{abs}(x - 1)) - \text{pow}(\text{abs}(y), 1/3) / (1 + \text{pow}(x, 2) / 2 + \text{pow}(y, 2) / 4)$;

E) $a = (\text{sqr}(\text{abs}(x - 1)) - \text{pow}(\text{abs}(y), 1/3)) / (1 + \text{pow}(x, 2) / 2 + \text{pow}(y, 2) / 4)$;

24. Ifoda to'g'ri yozilgan satrni ko'rsating: $y = \cos^2 x + \sqrt[3]{\pi + (2 - x)^5}$

A) $y = \text{sqr}(\cos(x)) + \text{pow}(3.14 + \text{pow}(2 - x, 5), 1/3)$;

B) $y = \text{pow}(\cos(x), 2) + \text{sqr}(3.14 + \text{pow}(2 - x, 5))$;

C) $y = \text{pow}(\cos(x), 2) + \text{pow}(3.14 + \text{pow}(2 - x, 5), 1/3)$;

D) $y = \cos(\text{pow}(x, 2)) + \text{pow}(3.14 + \text{pow}(2 - x, 5), 1/3)$;

E) $y = \text{pow}(\cos(x), 2) + \text{pow}(3.14 + \text{pow}(5, 2 - x), 1/3)$;

25. Ifoda to'g'ri yozilgan satrni ko'rsating: $a = \frac{2 \cos(x - \pi)}{3 + \sin^2 y}, b = 1 + \frac{z^2}{3 + \frac{z^2}{5}}$

A) $a = 2 * \cos(x - 3.14) / (3 + \text{pow}(\sin(y), 2))$;
 $b = 1 + \text{sqr}(z) / (3 + \text{pow}(z, 2) / 5)$;

B) $a = 2 * \cos(x - 3.14) / 3 + \text{pow}(\sin(y), 2)$;
 $b = (1 + \text{pow}(z, 2)) / (3 + \text{pow}(z, 2) / 5)$;

C) $a = 2 * \cos(x - 3.14) / (3 + \text{sqr}(\sin(y)))$;
 $b = 1 + \text{pow}(z, 2) / (3 + \text{pow}(z, 2) / 5)$;

D) $a = 2 * \cos(x - 3.14) / (3 + \sin(\text{pow}(y, 2)))$;
 $b = 1 + \text{pow}(z, 2) / (3 + \text{pow}(z, 2) / 5)$;
 $a = 2 * \cos(x - 3.14) / (3 + \text{pow}(\sin(y), 2))$;

E) $b = 1 + \text{pow}(z, 2) / (3 + \text{pow}(z, 2) / 5)$;

26. Ifoda to'g'ri yozilgan satrni ko'rsating: $y = \sqrt[5]{x^3 + 4} + \sin x^4 + \text{tg}x^3$

A) $y = \text{pow}(\text{pow}(x, 3) + 4, 1/5) + \sin(\text{pow}(x, 4)) + \tan(\text{pow}(x, 3))$;

- B) $y = \text{pow}(1/5, \text{pow}(x, 3) + 4) + \text{pow}(\sin(x, 4)) + \text{tg}(\text{pow}(3, x));$
 C) $y = \text{pow}(1/5, \text{pow}(x, 3) + 4) + \sin(\text{pow}(4, x)) + \tan(\text{pow}(3, x));$
 D) $y = \text{pow}(\text{pow}(x, 3) + 4, 1/5) + \sin(\text{pow}(x, 4)) + \text{pow}(\tan(x, 3));$
 E) $y = \text{pow}(\text{pow}(x, 3) + 4, 1/5) + \text{pow}(\sin(4, x)) + \text{pow}(\text{tg}(x, 3));$

27. Ifoda to'g'ri yozilgan satrni ko'rsating: $y = \sqrt[4]{e^{3x+7}} + \cos x^5 + \ln|3x+7|$

- A) $y = \text{pow}(\exp(3*x) + 7, 1/4) + \cos(\text{pow}(x, 5)) + \ln(\text{abs}(3*x + 7));$
 B) $y = \text{pow}(\exp(3*x) + 7, 1/4) + \text{pow}(\cos(5, x)) + \log(\text{abs}(3*x + 7));$
 C) $y = \text{pow}(\exp(3*x) + 7, 1/4) + \cos(\text{pow}(5, x)) + \log(\text{abc}(3*x + 7));$
 D) $y = \text{pow}(\exp(3*x) + 7, 1/4) + \cos(\text{pow}(x, 5)) + \log(\text{abs}(3*x + 7));$
 E) $y = \text{pow}(\exp(3*x) + 7, 1/4) + \text{pow}(\cos(x, 5)) + \ln(\text{abs}(3*x + 7));$

28. Ifoda to'g'ri yozilgan satrni ko'rsating: $y = \ln 5x + x^5 \sin^3 x + \text{tg} x^4$

- A) $y = \ln(5*x) + \text{pow}(x, 5) \text{pow}(\sin(x), 3) + \tan(\text{pow}(x, 4));$
 B) $y = \log(5*x) + \text{pow}(x, 5) * \text{pow}(\sin(x), 3) + \tan(\text{pow}(x, 4));$
 C) $y = \log(5*x) + \text{pow}(x, 5) * \sin(\text{pow}(x, 3)) + \text{tg}(\text{pow}(x, 4));$
 D) $y = \log(5*x) + \text{pow}(5, x) * \sin(\text{pow}(x, 3)) + \tan(\text{pow}(x, 4));$
 E) $y = \ln(5*x) + \text{pow}(x, 5) \text{pow}(\sin(x), 3) + \text{tg}(\text{pow}(4, x));$

29. Ifoda to'g'ri yozilgan satrni ko'rsating: $y = \text{tg} \sin \pi x + x^7 + \sqrt[3]{e^{2x+3}}$

- A) $y = \text{tg}(\sin(\text{PI}*x)) + \text{pow}(x, 7) + \exp((2*x+3)/3);$
 B) $y = \tan(\sin(\text{PI}*x)) + \text{pow}(7, x) + \exp(2*x+3/3);$
 C) $y = \text{tg}(\sin(\text{PI}*x)) + \text{pow}(7, x) + \exp((2*x+3)/3);$
 D) $y = \text{tg}(\sin(\text{PI}*x)) + \text{pow}(x, 7) + \exp(2*x+3/3);$
 E) $y = \tan(\sin(\text{PI}*x)) + \text{pow}(x, 7) + \exp((2*x+3)/3);$

30. Ifoda to'g'ri yozilgan satrni ko'rsating: $y = \ln|5x+3| + \cos \pi^2 + \text{tg} x^3$

- A) $y = \log(\text{abc}(5*x+3)) + \cos(\text{PI}*x*x) + \tan(\text{pow}(x, 3));$
 B) $y = \ln(\text{abs}(5x+3)) + \cos(\text{PI}*x*x) + \text{tg}(\text{pow}(x, 3));$
 C) $y = \log(\text{abs}(5*x+3)) + \cos(\text{PI}*x*x) + \tan(\text{pow}(x, 3));$
 D) $y = \ln(\text{abc}(5*x+3)) + \cos(\text{PI}*x*x) + \tan(\text{pow}(x, 3));$
 E) $y = \log(\text{abs}(5x+3)) + \cos(\text{PI}*x*x) + \text{tg}(\text{pow}(x, 3));$

31. Preprocessorning qaysi direktivasi grafik funksiyalar bilan ishlashni amalga oshiradi?

- A) `#include <conio.h>`
 B) `#include <graphics.h>`
 C) `#include <math.h>`
 D) `#include <stdio.h>`
 E) `#include <dos.h>`

32. Preprocessorning qaysi direktivasi matematik funksiyalar bilan ishlashni amalga oshiradi?

- A) `#include <math.h>` B) `#include <stdio.h>` C) `#include <conio.h>`

D) #include <string.h> E) #include <grafh.h>

33. Ekranni tozalash amali uchun preprosessorning qaysi direktivasidan foydalaniladi?

A) #include <conio.h> B) #include <grafh.h> C) #include <math.h>

D) #include <stdio.h> E) #include <string.h>

34. Preprosessorning qaysi direktivasi satrlar bilan ishla funksiyalar bilan ishlashni amalga oshiradi?

A) #include <conio.h> B) #include <grafh.h> C) #include <string.h>

D) #include <stdio.h> E) #include <stdlib.h>

35. Dastur natijasini aniqlang:

main()

{int x=4, y=3, w =1, t;

w++;

t=++x-y;

y--;

cout<<"t="<< t<<x<< y<<w;}

A) t=2,4,2,3 B) t=2,5,2,2 C) t=3,4,3,2 D) t=1,2,1,3 E) t=4,4,2,3

36. Dastur natijasini aniqlang:

main()

{int s=7, t=9, z=7, x, y, f;

x=!(s>t)|| (s>z);

y=++x+--z;

f=!(y>=z);

cout<<"x= "<<x<<"y="<<y<<"f="<< f; }

A) x=1, y=7, f=0 B) x=2, y=8, f=1 C) x=2, y=9, f=0 D) x=2, y=8, f=0

E) x=2, y=7, f=0

37. Dastur natijasini aniqlang:

main()

{int a=20,b=3,c=4,x,y,z;

x=a++;

y=!(++b<c);

z=x--+y;

cout<<"x= "<<x<<"y="<<y<<"z="<< z; }

A) x=20, y=1, z=22 B) x=19, y=1, z=21 C) x=19, y=0, z=20

D) x=19, y=1, z=20 E) x=20, y=0, z=20

38. Dastur natijasini aniqlang:

main()

{int a=12,b=13,c=14,x,y,z;

x=++a;

```
y=++b<c&&c--<a;
```

```
z=!(b<a);
```

```
cout<<"x= "<<x<<"\ty="<<y<<"\tz="<<z; }
```

A) x=13, y=0, z=0 B) x=12, y=0, z=0 C) x=12, y=1, z=0 D)

x=13, y=0, z=1

E) x=12, y=0, z=1

39. Dastur natijasini aniqlang:

```
main()
```

```
{int x=4,y=17,z=3,s,d,f;
```

```
  s=(int)y/x+z;
```

```
d=y%x+s%z;
```

```
  f=++x+--y;
```

```
  cout<<"s= "<<s<<",f="<<f<<",d="<<d;}
```

A) s=2, f=21, d=3 B) s=7, f=21, d=2 C) s=7, f=20, d=2

D) s=7, f=21, d=0 E) s=7, f=22, d=3

40. Dastur natijasini aniqlang:

```
main()
```

```
{int a=9,b=25,c=8,x,y,z;
```

```
  x=++a+b%7;
```

```
  y=!(b++<c&&c--<a--);
```

```
  z=(b>++a%3);
```

```
  cout<<"x= "<<x<<",y="<<y<<",z="<<z;}
```

A) x=14, y=0, z=1 B) x=14, y=1, z=0

C) x=13, y=1, z=0

D) x=13, y=1, z=1 E) x=14, y=1, z=1

41. Dastur natijasini aniqlang:

```
main()
```

```
{int a=4,b=6,c=1,s,d,f;
```

```
  s=a++;
```

```
  d=--a+c--;
```

```
  f=a<b&&c<d;
```

```
  cout<<"s= "<<s<<",f="<<f<<",d="<<d;}
```

A) s=4, f=1, d=5 B) s=4, f=0, d=4 C) s=4, f=1, d=6 D) s=4, f=0, d=5

E) s=5, f=0, d=5

42. Dastur natijasini aniqlang:

```
main()
```

```
{int x=10,y=4,z=3,a,b;
```

```
  a=!(x>y);
```

```
  b=a++-z--;
```

```
  cout<<"a="<<a<<", b="<<b;}
```

A) a=0, b=-3 B) a=1, b=-2 C) a=2, b=-2 D) a=1, b=-3 E) a=2, b=-3

43. Dastur natijasini aniqlang

main()

```
{int s=17,t=29,z=47,x,y,f;
```

```
x!=(s>t)&&(s>z);
```

```
y=x++ +--z;
```

```
f=(y>=z);
```

```
cout<<"x= "<<x<<",y="<<y<<",f="<<f;}
```

A) x=1, y=46, f=1 B) x=2, y=47, f=1 C) x=1, y=47, f=1

D) x=1, y=46, f=0 E) x=2, y=46, f=1

44. C++ tilidagi har bir dastur qaysi main() fuksiyasini o'zida saqlaydi

a) main() b) iostream.h c) #include d) To'g'ri javob keltirilmagan.

45. To'g'ri yozilgan operatorni toping: agar $a \leq b$ va $c < 3$ bo'lsa, u holda $a = (c+b)$ bo'lsin.

A) if (a<=b && c<3) a=c+b;

B) if ((a<=b) && c<3) a=c+b;

C) if (a<=b) && (c<3) a=c+b;

D) if (a<=b !! c<3) a=c+b;

E) if (a<=b & c<3) a=c+b;

46. To'g'ri yozilgan operatorni toping: agar x va y lardan bittasi manfiy bo'lsa, u holda son 0,5 ga oshirilsin.

A) if (x<0 & y<0) {x+=0.5; y+=0.5;}

B) if (x<0 || y<0) {x+=0.5; y+=0.5;}

C) if (x<0 && y<0) {x+=0.5; y+=0.5;}

D) if (x<0 ! y<0) {x+=0.5; y+=0.5;}

E) if (x<0 || y<0) x+=0.5; y+=0.5;

47. To'g'ri yozilgan operatorni toping: agar $a \geq b \geq c$ bo'lsa u holda har bir qiymatni uning kvadrati bilab almashtiring.

A) if (a>=b && b>=c) a*=a; b*=b; c*=c;

B) if (a>=b && b>=c) {a*=a; b*=b; c*=c;}

C) if (a>=b || b>=c) {a*=a; b*=b; c*=c;};

D) if (a>=b & b>=c) {a*=a; b*=b; c*=c;}

E) if (a>=b) && (b>=c) {a*=a; b*=b; c*=c;}

48. To'g'ri yozilgan operatorni toping: x va y berilgan, agar ular (0,5; 2) oraliqqa tegishli bo'lmasa, u holda sonlarni 10 marta kamaytirib.

A) if ((x<=0.5 && x>=2) || (y<=0.5 && y>=2)) {x/=10; y/=10;}

B) if ((x<=0.5 || x>=2) && (y<=0.5 || y>=2)) {x/=10; y/=10;}

- C) if (($x \leq 0.5 \parallel x \geq 2$) && ($y \leq 0.5 \parallel y \geq 2$)) $x/=10$; $y/=10$;
 D) if ($x \leq 0.5$) $\parallel$ ($x \geq 2$) && ($y \leq 0.5$) $\parallel$ ($y \geq 2$) { $x/=10$; $y/=10$ };
 E) if (($x \leq 0.5 \parallel x \geq 2$) & ($y \leq 0.5 \parallel y \geq 2$)) $x/=10$; $y/=10$;

49. To'g'ri yozilgan operatorni toping: a va b berilgan, ularning kichigini yig'indining yarmi bilan, kattasini ko'paytmaning ikkilanganligi bilan almashtiring.

- A) if ($a < b$) { $a1=(a+b)/2$; $b1=2*a*b$ };
 B) if ($a < b$) { $a=(a+b)/2$; $b=2*a*b$ };
 C) if $a < b$ $a1=a+b/2$; $b1=2*a*b$;
 D) if $a < b$ { $a=(a+b)/2$; $b=2*a*b$ };
 E) if ($a < b$) $a1=(a+b)/2$; $b1=2*a*b$;

50. To'g'ri yozilgan operatorni toping: agar $a \leq b \leq c \leq d$ bo'lsa, u holda har bir qiymatni ularning eng kattasi bilan, aks holda har bir qiymatni ularning kvadrati bilan almashtiring.

- A) if ($a \leq b$ & $b \leq c$ & $c \leq d$) $a=b=c=d$ else { $a*=a$; $b*=b$; $c*=c$; $d*=d$ };
 B) if ($a \leq b$) && ($b \leq c$)&&($c \leq d$) $a=b=c=d$ else { $a*=a$; $b*=b$; $c*=c$; $d*=d$ };
 C) if ($a \leq b$ && $b \leq c$ && $c \leq d$) $a=b=c=d$; else $a*=a$; $b*=b$; $c*=c$; $d*=d$;
 D) if ($a \leq b$ && $b \leq c$ && $c \leq d$) $a=b=c=d$; else { $a*=a$; $b*=b$; $c*=c$; $d*=d$ };
 E) if ($a \leq b \parallel b \leq c \parallel c \leq d$) $a=b=c=d$; else $a*=a$; $b*=b$; $c*=c$; $d*=d$;

51. To'g'ri yozilgan operatorni toping: agar ikki sonning yig'indisi 1 dan kichik bo'lsa ularning o'rta arifmetigin, aks holda ularning o'rta geometrigini hisoblang.

- A) if ($x+y > 1$) $s=(x+y)/2$; else $s=\text{sqr}(x*y)$;
 B) if ($x+y < 1$) $s=(x+y)/2$; else $s=\text{sqrt}(x*y)$;
 C) if ($x+y > 1$) $s=x+y/2$; else $s=\text{sqr}(x*y)$;
 D) if ($x+y < 1$) $s=(x+y)/2$ else $s=\text{sqrt}(x*y)$;
 E) if ($x+y < 1$) $s=x+y/2$ else $s=\text{sqrt}(x*y)$;

52. Dastur natijasini aniqlang:

```
main()
{ int i=3, p=1;
  for(;i*i>0;i--) p*=i*(i+1); cout<<"i="<< i<<"p="<< p;}
```

- A) $i=1$, $p=144$ B) 0 , 72 C) $i=0$, $p=72$ D) $i=1$, $p=72$ E) $i=0$, $p=144$

53. To'g'ri tuzilgan dasturni aniqlang: x_1, x_2, x_3, x_4 ; sonlari berilgan, $x_1x_4+x_2x_3+x_3x_2+x_4x_1$ ni hisoblang.

- A) $p=x$; $s=0$; for($i=0$; $i < 4$; $i++$) $s+=*(p+i)**(p+3)$;
 B) $p=x$; $s=1$; for($i=0$; $i < 4$; $i++$) $s*=*(p+i)**(p+3-i)$;

- C) $p=x; s=0; \text{ for}(i=1;i \leq 4;i++) \quad s+=*(p+i)**(p+3-i);$
 D) $p=x; s=0; \text{ for}(i=1;i \leq 4;i++) \quad s+=*(p+i)**(p+3+i);$
 E) $p=x; s=0; \text{ for}(i=0;i < 4;i++) \quad s+=*(p+i)**(p+3-i);$

54. To'g'ri tuzilgan dasturni aniqlang: $a_1, \dots, a_6$ butun sonlar berilgan, $(a_1-a_6)(a_3-a_4)(a_5-a_2)$ ni hosil qiling.

- A) $px=a; p=0; \text{ for}(i=0;i < 6;i+=2) \quad p+=*(px+i)-*(px+5-i);$
 B) $px=a; p=1; \text{ for}(i=0;i < 6;i++) \quad p*=*(px+i)-*(px+5);$
 C) $px=a; p=0; \text{ for}(i=0;i < 3;i++) \quad p+=*(px+i)-*(px+5+i);$
 D) $px=a; p=1; \text{ for}(i=0;i < 6;i+=2) \quad p*=*(px+i)-*(px+5-i);$
 E) $px=a; p=1; \text{ for}(i=0;i < 3;i++) \quad p*=*(px+i)-*(px+5-i);$

55. To'g'ri tuzilgan dasturni aniqlang: x_1, x_2, x_3, x_4 lar berilgan, $x_1-x_4, x_2-x_4, x_3-x_4$ ketma-ketlikni hosil qiling.

- A) $p=x; \text{ for}(i=0;i < 3;i++) \quad *(p+i)=*(p+i)-*(p+3);$
 B) $p=*x; \text{ for}(i=0;i < 3;i++) \quad *(p+i)=*(p+i)-*(p+3);$
 C) $p=x; \text{ for}(i=0;i < 3;i+=2) \quad *(p+i)=*(p+i)-*(p+3);$
 D) $p=x; \text{ for}(i=0;i < 3;i++) \quad *p+i=*(p+i)-*(p+3);$
 E) $p=x; \text{ for}(i=0;i < 3;i+=2) \quad *(p+i)=*(p+i)-*(p+3-i);$

56. To'g'ri yozilgan dasturni toping: $\sum_{k=1}^{\infty} \frac{(-1)^k (k+1)x^k}{3^k}$ ifofadani

ϵ aniqlikda hisoblash.

- A) $k=1; \text{ do } \{ ak=pow(-1,k)*(k+1)*pow(x,k)/pow(3,k); k++; s+=ak; \}$
 $\text{ while } (abs(ak) < \epsilon);$
 B) $k=1; \text{ do } \{ ak=pow(-1,k)*(k+1)*pow(x,k)/pow(3,k); k++; s+=ak; \}$
 $\text{ while } (abs(s) > \epsilon);$
 C) $k=1; \text{ do } ak=pow(-1,k)*(k+1)*pow(x,k)/pow(3,k); k++; s+=ak;$
 $\text{ while } (abs(ak) > \epsilon);$
 D) $k=1; \text{ do } \{ ak=pow(-1,k)*(k+1)*pow(x,k)/pow(3,k); k++; s+=ak; \}$
 $\text{ while } (abs(ak) > \epsilon);$
 E) $k=1; \text{ do } \{ ak=pow(-1,k)(k+1)pow(x,k)/pow(3,k); k++; s+=ak; \}$
 $\text{ while } (abs(ak) > \epsilon);$

57. To'g'ri yozilgan dasturni toping: $\prod_{k=1}^n \left(1 + \frac{\sin(kx)}{k!}\right)$ ifodani hisoblash.

- A) $k=p=f=1; \text{ while } (k \leq n) \{ f*=k; p*=(1+\sin(k*x)/f); k--; \}$
 B) $k=p=f=1; \text{ while } k \leq n \{ f*=k; p*=(1+\sin(k*x)/f); k++; \}$
 C) $k=p=f=1; \text{ while } (k \leq n) \{ f*=k; p*=(1+\sin(k*x)/f); k++; \}$
 D) $k=p=f=1; \text{ while } (k \leq n) \quad f*=k; \quad p*=(1+\sin(k*x)/f); \quad k++;$
 E) $k=p=f=1; \text{ while } (k \leq n) \{ f*=k; p*=(1+\sin(k*x)/f*k); \} \quad k++;$

58. To'g'ri yozilgan dasturni toping: $\sum_{k=1}^{\infty} \frac{1}{\sqrt{|x|} + k^2}$ ifodani ε aniqlikda

hisoblash.

- A) k=1; do { ak=1/(sqrt(abs(x))+pow(k,2)); k++; s+=ak; }
while(abs(ak)>ε);
- B) k=1; do { ak=1/(sqrt(abs(x))+pow(k,2)); k++; s+=ak; }
while (abs(ak)<ε);
- C) k=1; do sqrt(abs(x))+pow(k,2)); k++; s+=ak;
while (abs(ak)>ε);
- D) k=1; do { ak=1/(sqrt(abs(x))+pow(k,2)); k++; s+=ak; }
while (abs(s)<ε);
- E) k=1; do { ak=1/sqrt(abs(x))+sqr(k); k++; s+=ak; }
while (abs(s)>ε);

59. To'g'ri yozilgan dasturni toping: $\sum_{i=1}^n ix + 1$ ni hisoblash.

- A) i=1; while (i<=n) { s+=i*x+1; i++; }
- B) i=1; while (i<n) { s+=ix+1; i++; }
- C) i=1; while i<=n s+=i*x+1; i++;
- D) while (i<=n) s+=i*x+1; i++;
- E) i=1; while (i<=n) { s+=i*x+1; i--; }

60. Dastur natijasini aniqlang:

m=0; i=1; do { m+=1/(2*i+1); i--; } while (i>=1);

- A) 1.0; B) 1/3; C) 0.2; D) 0.75; E) 0.15.

61. Dastur natijasini aniqlang:

```
main()
{int i=5,p=1;
  while (--i>1) p*=i;
  cout<<"i="<< i<<"p="<< p;}
```

- A) i=1, p=120 B) i=1, p=24 C) i=0, p=24 D) i=24, p=1 E) i=0, p=120

62. Dastur natijasini aniqlang:

```
main()
{ int k,f; float p;
  k=f=p=1;
  while (++k<=3)
    f*=k;
  p*=1/f;
  cout<< k<< f<< p;}
```

- A) k=4 f=6 p=0.083 B) 4 6 0.167 C) k=5 f=24 p=0.042

D) $k=4$ $f=6$ $p=0.167$ E) $k=5$ $f=6$ $p=0.167$

63. For operatori to'g'ri foydalanilgan qatorni ko'sating:

- A) `main () { int i, j; for(i=6; j=0; j<10; i--, j++); cout<<i<<j; }`
- B) `main () { int i, j; for(i=6, j=0; j<10; i--, j++) cout<<i<<j; }`
- C) `main () { int i, j; for(i=6, j=0, j<10; i--, j++); cout<<i<<j; }`
- D) `main () { int i, j; for(i=6, j=0; j<10; i--, j++) cout<<i<<j; }`
- E) `main () { int i, j; for i=6, j=0; j<10; i--, j++ cout<<i<<j;`

64. Dastur natijasini aniqlang:

`m=0; i=1; do { m+=1/(2*i-1); i--; } while (i>=1);`

- A) 1.0; B) 1/3; C) 0.2; D) 0.75; E) 0.15.

65. To'g'ri tuzilgan dastur bo'lagini ko'rsating: a[10] massivning manfiy elementlarini kvadratga ko'tarish.

- A) `for(i=0; i<10; i++) if (a[i]<0) a[i]=sqr(a[i],2);`
- B) `for(i=0; i<10; i++) if (a[i]<0) a[i]=pow(a[i],2);`
- C) `for(i=0; i<=10; i++) if (a[i]<0) a[i]=pow(a[i],2);`
- D) `for(i=0, i<10, i++) if (a[i]<0) a[i]=pow(a[i],2);`
- E) `for(i=0; i<10; i++) if a[i]<0 a[i]=pow(a[i],2);`

66. To'g'ri tuzilgan dastur bo'lagini ko'rsating: a[30] massivni hosil qiling, bu yerda $a_i=2*i+1$.

- A) `for (i=0; i<30; i++) { a[i]=2*i+1; cout<< a[i]; }`
- B) `for (i=0; i<30; i++) a[i]=2*i+1; cout<< a[i];`
- C) `for (i=0; i<30; i++) { a[i]=2*i+1; } cout<< a[i];`
- D) `for (i=1; i<30; i++) a[i]=2*i+1; cout<< a[i];`
- E) `for (i=1; i<30; i++) { a[i]=2*i+1; cout<< a[i]; }`

67. To'g'ri tuzilgan dastur bo'lagini ko'rsating: a[n] massivning musbat juft elementlari sonini aniqlash.

- A) `for(i=0; i<n; i++); if (a[i]>0 && a[i]%2 == 0) k++;`
- B) `for(i=0; i<n; i++) if (a[i]>0 !! a[i]%2 == 0) k++;`
- C) `for(i=0; i<n; i++) if (a[i]>0 & a[i]%2 == 0) k++;`
- D) `for(i=0; i<n; i++) if (a[i]>0 && a[i]%2 == 0) k++;`
- E) `for(i=0; i<n; i++) if (a[i]>0 && a[i]%2 == 0) k++;`

68. Dastur natijasini aniqlang:

`double a[7]={2,3,1,4,-2,5,6};`

`main()`

`{ int i;`

`for(i=0; i<7; i+=2)`

`a[i]=pow(a[i],3);`

`for(i=0; i<7; i++)`

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

A) 8 3 1 4 -8 5 216 B) 2 27 1 64 -2 125 6 C) 8 27 1 64 -8 125 216
D) 3 31 64 -2 5 216 E) 2 3 1 4 -8 125 216

69. Dastur natijasini aniqlang:

```
int a[10]={1,3,5,2,8,6,12,9,13,20};
```

```
main()
```

```
{ int i, k=0;
```

```
  for(i=0;i<10;i+=3) if (a[i]%2 == 0) k++; cout<<k; }
```

A) 4 B) 2 C) 3 D) 5 E) 6

70. Dastur natijasini aniqlang:

```
int a[10]={2,4,5,7,8,6,3,12,19,9};
```

```
main()
```

```
{ int i, max=2;
```

```
  for(i=0;i<10;i+=2) if (a[i]>max) max=a[i]; cout<<max; }
```

A) 12 B) 19 C) 9 D) 8 E) 6

71. b[10] massivni hosil qilish to'g'ri tuzilgan dasturni aniqlang:

A) for(i=0;i<10;i++); b[i]=5*i-1;

B) for(i=0;i<10;i++) b[i]=5*i-1;

C) for(i=0;i<10;i++) b[i]=5*i-1;

D) for(i=0;i<10;i++) b[i]=5i-1;

E) for(i=0;i<9;i++) b[i+1]=5*i-1;

72. Dastur natijasini aniqlang:

```
int a[8]={1,2,3,4,5,-2,6,16};
```

```
main()
```

```
{ int i, p=1; for(i=0;i<8;i++) if (a[i] % 2 != 0) p*=a[i]; cout<<p; }
```

A) 256 B) 216 C) 16 D) 15 E) 360

73. Dastur natijasini aniqlang: a[15] massivning birinchi 0 elementigacha elementlar yig'indisi.

A) i=0; do {s+=a[i]; i--; } while (a[i] == 0 && i<15);

B) i=0; do s+=a[i]; i++; while (i != 0 && i<15);

C) i=0; do { s+=a[i]; i++; } while (i == 0 && i<15);

D) i=0; do { s+=a[i]; i++; } while (a[i] != 0 & i<15);

E) i=0; do { s+=a[i]; i++; } while (a[i] != 0 && i<15);

74. Dastur natijasini aniqlang:

```
int a[9]={2,3,4,6,9,8,12,12,11};
```

```
main() {int i; s=0;
```

```
  for (i=0; i<9; i+=2) if (a[i] % 2 == 0) s+=a[i]; cout<<s; }
```

A) 26 B) 18 C) 16 D) 25 E) 36

75. A[7,7] massivning toq satrlarida joylashgan musbat elementlar yig'indisi aniqlash to'g'ri yozilgan qatorni ko'rsating:

- A) for(i=0,s=0;i<7;i+=2) for(j=0;j<7;j++); if(a[i][j]>0) s+=a[i][j];
 B) for(i=0,s=0;i<7;i+=2) for(j=0;j<7;j++); if(a[i][j]>0) s+=a[i][j];
 C) for(i=0,s=0;i<7;i+=2); for(j=0;j<7;j++) if(a[i][j]>0) s+=a[i][j];
 D) for(i=0;s=0;i<7;i+=2) for(j=0;j<7;j++) if(a[i,j]>0) s+=a[i,j];
 E) for(i=0;s=0;i<7;i+=2) for(j=0;j<7;j++) if(a[i][j]>0) s+=a[i][j];

76. Dastur natijasini aniqlang

main()

{ static int a[4][4]={ {2,3,4,5},{7,5,9,8}, {1,0,6,7},{-2,4,5,3} };

int i,k=0; for(i=0;i<4;i++) if(a[i][3-i]>0) k++; cout<<k; return 0;

- A) 4 B) 2 C) 1 D) 3 E) 0

77. a[4][4] massivning bosh diagonalining yuqorisida joylashgan nol bo'lmagan elementlar soni va yig'indisini hisoblash uchun to'g'ri yozilgan dastur bo'lagini ko'rsating.

- A) k=0; p=1; for(i=0;i<3;i++) for(j=i+1;j<3;j++) if(a[i][j]<>0) {p*=a[i][j]; k++ }
 B) k=0; p=1; for(i=0;i<3;i++) for(j=i;j<3;j++) { if(a[i][j]<>0){ p*=a[i][j]; k++ } }
 C) for(i=0;i<3;i++) for(j=1;j<3;j++) if(a[i][j]<>0) { p*=a[i][j]; k++ }
 D) for(i=0;i<3;i++) for(j=i+1;j<3;j++) if(a[i][j]!=0) { p*=a[i][j]; k++; }
 E) for(i=0;i<3;i++) for(j=1;j<3;j++) if(a[i][j]!=0) { p*=a[i][j]; k++;}

78. a[n][n] massivning bosh diagonalida joylashgan musbat elementlar yig'indisi va ko'paytmasini hisoblash uchun to'g'ri yozilgan dastur bo'lagini ko'rsating.

- A) s=0; p=1; for(i=0;i<n;i++) if(a[i][i]>0) { s+=a[i][i]; p*=a[i][i]; }
 B) s=0; p=1; for(i=0;i<n;i++) if(a[i][i]>0) s+=a[i][i] p*=a[i][i];
 C) s=0; p=1; for(i=0;i<n;i++) if(a[i][i]>0) { s+=a[i][j] p*=a[i][j] }
 D) s=0; p=1; for(i=0;i<=n;i++) if(a[i][i]>0) { s+=a[i][i] p*=a[i][j] }
 E) s=0; p=1; for(i=0;i<n;i--) if(a[i][i]>0) { s+=a[i][i]; p*=a[i][i]; }

79. a[5,5] massivning yordamchi diagonalidan yuqorida joylashgan manfiy elementlar sonini aniqlash uchun to'g'ri yozilgan dastur bo'lagini ko'rsating.

- A) k=0; for(i=0;i<3;i++) for(j=0;j<3-i;j++) if(a[i][j]<0) k++;
 B) k=0; for(i=0;i<4;i++) for(j=1;j<i;j++) if(a[i][j]<0) k++
 C) k=0 for(i=0;i<3;i++) for(j=1;j<i+1;j++) if(a[i][j]<0) k++;
 D) k=0; for(i=0;i<4;i++) for(j=1;j<4-i;j++) if(a[i][j]<0) k++
 E) k=0; for(i=0;i<3;i++) for(j=0;j<2-i;j++) if(a[i][j]<0) k++

80. A[6,6] massivning toq satrlarida joylashgan musbat elementlar yig'indisi aniqlash to'g'ri yozilgan qatorni ko'rsating:

A) for(i=1;i<6;i+=2) for(j=s=k=0;j<6;j++) if (a[i][j]%2==0) { k++; s+=a[i][j]; }

B) for(i=1,s=k=0;i<6;i+=2) for(j=0;j<6;j++) if (a[i][j]%2!=0) { k++; s+=a[i][j]; }

C) for(i=s=k=0;i<6;i+=2) for(j=0;j<6;j++) if (a[i][j]%2!=0) { k++; s+=a[i][j]; }

D) for(i=s=k=0;i<6;i+=2) for(j=0;j<6;j++) if (a[i][j]%2==0) { k++; s+=a[i][j]; }

E) for(i=s=k=0;i<6;i+=2) for(j=0;j<6;j++) if (a[i][j]%2==0) k++; s+=a[i][j];

81. A(4,4) massivning bosh diagonalining quyi qismida joylashgan elementlarning eng kattasi va uning o'rnini aniqlash uchun to'g'ri yozilgan dastur bo'lagini ko'rsating:

A) max=a[0,0]; l=m=0; for(i=1;i<4;i++) for(j=0;j<i;j++) if (a[i,j]>max) {max=a[i][j]; l=i; m=j; }

B) max=a[0][0]; l=m=0; for(i=1;i<4; i++) for(j=0;j<i+1;j++) if (a[i][j]>max) max=a[i][j]; l=i; m=j;

C) max=a[0][0]; l=m=0; for(i=1;i<4; i++) for(j=0;j<i+1;j++) if (a[i][j]<max){ max=a[i][j]; l=i; m=j;}

D) max=a[0][0]; l=m=0; for(i=1;i<4; i++) for(j=0;j<i+1;j++) if (a[i][j]>max) {max=a[i][j]; l=i; m=j;}

E) max=a[1,1]; l=m=0; for(i=2;i<=4; i++) for(j=1;j<=i;j++) if (a[i][j]>max) { max=a[i][j]; l=i; } m=j;

82. Ikki indeks ishlatiladigan massiv-bu

A) ikki o'lchovli massiv B) vektor C) blok D) jadval E) funksiya massivi

83. A[7,7] massivning toq satrlarida joylashgan musbat elementlar yig'indisi aniqlash to'g'ri yozilgan qatorni ko'rsating:

A) for(i=0, s=0;i<7;i+=2) for(j=0;j<7;j++); if(a[i][j]>0) s+=a[i][j];

B) for(i=0, s= 0;i<7;i+=2) for(j=0;j<7;j++) if (a[i][j]>0) s+=a[i][j];

C) for (i=0, s=0;i<7;i+=2); for(j=0;j<7;j++) if (a[i][j]>0) s+=a[i][j];

D) for(i=0; s=0;i<7;i+2) for(j=0;j<7;j++) if (a[i,j]>0) s+=a[i,j];

E) for(i=0;s=0;i<7,i+=2) for(j=0;j<7;j++) if(a[i][j]>0) s+=a[i][j] ;

84. Dastur natijasini aniqlang

main()

{ static int a[4][4]={ {2,3,4,5},{7,5,9,8}, {1,0,6,7},{-2,4,5,3} };

int i,k=0; for(i=0;i<4;i++) if (a[i][3-i]>0) k++; cout<<k; return 0;

A) 4

B) 2

C) 1

D) 3

E) 0

Foydalanish uchun adabiyotlar

1. Stroustrup Bjarne The C++ Programming Language (Fourth Edition). Amazon Product Page, 2012.
2. Virender Singh Learn C++ Programming Language :Become A Complete C++ Programmer, Amazon Digital Services, Inc. 2015.
3. Madraximov Sh.F., Gaynazarov S.M. C++ tilida dasturlash asoslari.// Uslubiy qo'llanma, O'zMU, 2009 - 196 b.
4. Madraximov Sh.F., Ikramov A.M., Babajonov M.R. C++ tilida dasturlash bo'yicha masalalar to'plami.// . O'quv qo'llanma, O'zMU, "Universitet" nashriyoti, 2014 - 160 b.
5. Павловская Т.С., Шупак Ю.С. C++. Объектно-ориентированное программирование. Практикум.- СПб.: Питер, 2005 – 265 с.
6. Культин Н.Б. C++ Builder в задачах и примерах. – СПб.:БХВ-Петербург, 2005. – 336 с.
7. Абрамов С.А., Гнезделова, Капустина Е.Н., и др. Задачи по программированию. - М.: Наука, 1988.
8. Аленский Н.А. и др. Задачи и методические рекомендации по программированию. – Мн.: БГУ, 1990, – 67 с.
9. Подбельский В.В. Программирование на языке СИ. – М., 2001, – 600 с.
10. Б. Керниган, Д. Ритчи, А. Фьюэр Язык программирования Си. Задачи по языку Си. М., «Финансы и статистика», 1985
11. Н. Джахани. Программирование на языке Си. М., «Радио и связь», 1988
12. С. Баурн. Операционная система UNIX.М., «Мир», 1986
13. <http://www.edu.uz>
14. <http://dastur.uz>
15. <http://www.intuit.ru>
16. <http://www.stroustrup.com/4th.html>,
17. <http://www.cplusplus.com/>

Ilova

1. Matematik funksiyalar C++ dasturlash tilida **<math.h>** kutubxonasida joylashgan bo'lib, ular quyidagicha yoziladi:

№	Matematik ko'rinishi	C++ da	Izoh
Trigonometrik funksiyalar			
1	$\sin x$	$\sin(x)$	x radianga teng bo'lgan burchak sinusini qaytaradi
2	$\cos x$	$\cos(x)$	x radianga teng bo'lgan burchak kosinusini qaytaradi
3	$\operatorname{tg} x$	$\tan(x)$	x radianga teng bo'lgan burchak tangensini qaytaradi
4	$\operatorname{ctg} x$	$1/\tan(x)$	x radianga teng bo'lgan burchak kotangensini qaytaradi
5	$\sec x$	$\sinh(x)$	x radianga teng bo'lgan burchak giperbolik sinusini qaytaradi
6	$\operatorname{cosec} x$	$\cosh(x)$	x radianga teng bo'lgan burchak giperbolik kosinusini qaytaradi
Teskari trigonometrik funksiyalar			
1	$\arcsin x$	$\operatorname{asin}(x)$	$[-1;1]$ chegarada berilgan x ning arksinusini qaytaradi
2	$\arccos x$	$\operatorname{acos}(x)$	$[-1;1]$ chegarada berilgan x ning arkkosinusini qaytaradi
3	$\operatorname{arctg} x$	$\operatorname{atan}(x)$	$[-1;1]$ chegarada berilgan x ning arktangensini qaytaradi
4	$\operatorname{arcctg} x$		$[-1;1]$ chegarada berilgan x ning arkkotangensini qaytaradi
Daraja, ildiz, logarifm, modul, e natural soni			
1	e^x	$\exp(x)$	e^x qiymatini qaytaradi

2	a^x	$pow(a,x)$	a^x qiymatini qaytaradi
3	$ x $	$fabs(x)$	x sonining absolyut qiymatini qaytaradi
4	$\ln x$	$log(x)$	x sonining natural logarifmini qaytaradi
5	$\log_{10} x$	$log10(x)$	x sonining 10 asosli logarifmini qaytaradi
6	$\sqrt{x}$	$sqrt(x)$	x sonining kvadrat ildizini qaytaradi
7	e $= 2.718281828459045$	M_E	e sonining qiymatini qaytaradi.
8	π $= 3.1415926535897932$	M_PI	π sonining qiymatini qaytaradi
Butun sonli funksiyalar			
1	x ga eng yaqin kichin butun son	$floor(x)$	x haqiqiy qiymatni eng yaqin kichik butun songa aylantiradi va uni haqiqiy ko'rinishda qaytaradi
2	x ga eng yaqin katta butun son	$ceil(x)$	x haqiqiy qiymatni eng yaqin katta butun songa aylantiradi va uni haqiqiy ko'rinishda qaytaradi
3	a ni b ga bo'lgandagi qoldiq	$a \% b$	Butun a sonini butun b soniga bo'lgandagi qoldiqni butun ko'rinishda qaytaradi.
4	a/b bo'linmaning butun qismi	a/b	Butun a sonini butun b soniga bo'lgandagi bo'linmaning butun qismini qaytaradi
5	$c = \sqrt{a^2 + b^2}$ gipotenuza	$hypot(a,b)$	To'g'ri burchakli uchburchakning katetlari bo'yicha gipotenuzani hisoblab natijaga qaytaradi

C++ tilida amallar ustivorligi jadvali

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

C++ da son turlari:

Tur	Xotira	Chegarasi
Butun sonlarni saqlovchi o'zgaruvchi tiplari		
int	4 bayt	-2 147 483648 dan 2 147 483647 gacha
unsigned int	4 bayt	0 dan 4294967295 gacha
short int	2 bayt	-32768 dan 32767 gacha
unsigned short int	2 bayt	0 dan 65,535 gacha
long long	8 bayt	-9 223 372 036 854 775 808 dan 9 223 372 036 854 775 807 gacha
unsigned long long	8 bayt	0 dan 18 446 744 073 709 551 615 gacha

Xaqiqiy sonlarni saqlovchi o'zgaruvchi turlari

float	4 bayt	+/- 3.4e +/- 38 (~7 raqam)
double	8 bayt	+/- 1.7e +/- 308 (~15 raqam)
long double	12 bayt	+/- 3.4e +/- 4932 (~15 raqam)

C++ tiliga oid atamalar

Termin	O'zbek tilidagi sharhi	Ingliz tilidagi sharhi
!=	Teng emas operatori; mantiqiy inkor amali qiymati bilan birhil.	The inequality operator; compares values for inequality returning a bool.
#define	Makro deriktivalarni belgilash	a directive that defines a macro.
#include	Bir source fayl ichida boshqa bir faylag murojatni amalga oshirish mexanizmi	a mechanism for textual inclusion of one source file into another.
+=	add-and-assign operatori; masalan $a+=b$ vazifazi jihatdan $a=a+b$ bilan bir xil	add-and-assign operator; $a+=b$ is roughly equivalent to $a=a+b$.
.c file	Dastur jodini o'zida jamlovchi fayl	file containing definitions.
.cpp file	Dastur jodini o'zida jamlovchi fayl	file containing definitions.
.h file	Sarlavha fayli	header file
address	Hotira manzili	a memory location
aggregate	Konstruktorsiz massiv yoki struktura	an array or a struct without a constructor
algorithm	Hisoblashning aniq ketma-ketligi	a precise definition of a computation.
and	$\&\&$ mantiqiy "va" (ko'paytirish) operatori sinonimi	synonym for $\&\&$, the logical and operator.
ANSI	Amerika milliy standart agentligi.	The American national standards organization.

application	Umumiy maqsadga ega dasturlar to'plami	a collection of programs seen as serving a common purpose (usually providing a common interface to their users)
bit	0 yoki 1 qiymatga ega birlik hotira	a unit of memory that can hold 0 or 1.
bool	Mantiqiy tip. Bu tip faqat rost yoki yolg'on qiymat qabul qiladi	the built-in Boolean type. A bool can have the values true and false.
Borland C++ Builder	C++ tilida visual dasturlashga ixtisoslashtirilgan IDE muhiti.	Borland's implementation of C++ together with proprietary libraries for Windows programming in an IDE
bug	Xatolik termini	colloquial term for error.
byte	Xotiradagi bir nechta belgilar yig'indisi	a unit of memory that can hold a character of the C++ representation character set.
C++	Tizimli dasturlashni protsedurali qo'llab quvvatlovchi dasturlash tili.	a general-purpose programming language with a bias towards systems programming that supports procedural programming, data abstraction, object-oriented programming, and generic programming. C++ was designed and originally implemented by Bjarne Stroustrup.

char	Belgili tip. Har bir belgi 8 bit, ya'ni baytga teng.	character type; typically an 8-bit byte.
char*	Char massiviga ko'rsatkich	pointer to a char or an array of char. Typically assumed to point to a C-style string.
cin	Standart kiritish oqimi	standard istream.
class	Foydalanuvchi belgilaydigan tur, sinf. Sinf foydalanuvchi funksiyasi, foydalanuvchi ma'lumotlari va kontentlari bo'lishi mumkin.	a user-defined type. A class can have member functions, member data, member constants, and member types.
compiler	C++ da yozilgan buyruqlarni mashina tiliga o'girib beruvchi vosita.	the part of a C++ implementation that produces object code from a translation unit.
const	Faqatgina bir marta qiymat berish mumkin bo'lgan o'zgaruvchilarni e'lon qilsh.	attribute of a declaration that makes the entity to which it refers readonly.
copy()	Nusxa olish operatori	Copy operator
UML	Unifikasiyaqilinganmodellashstil	Unified Modeling Language
OMG	Obyektlarniboshqarishguruhi	Object Management Group
4GL	To'rtinchiavlodtili	Fourth-Generation Language
ANSI	Amerikamilliystandartlashinstituti	American National Standards Institute
AMPS		Advanced Mobile Phone Service
ERP	Korxonaresurslarinirejalashtirish	Enterprise Resource Planning
CRM	Mijozlarbilano'zaromunosabatlarni boshqarish	Customer Relations Management
SQL	Tuzilmalashganso'rovlartili	Structured Query Language

OLAP	Xaqiqiyvaqtdama'lumotlargaanalitikaishlovberish	On-Line Analytical Processing
OLTP	Xaqiqiyvaqtdatranzaktsiyalargaishlovberish	On-Line Transaction Processing
TCO	Egalikqilishningyalpiqiymati	Total Cost of Ownership
JIT	Aynivaqtda	Just-In-Time
LAN	Lokalhisoblashtarmog'i	Local Area Network
MAN	Maxalliyhisoblashtarmog'i	Metropolitan Area Network
WAN	Xududiyhisoblashtarmog'i	Wide Area Network
ISO	Halqarostandartlashtirish tashkiloti	International Organization for Standardization
API	amaliydasturlashtirish maxsus interfeysi	Application Programming Interface
WWW	Umumjahono'rgamchakto'ri	World Wide Web
ASCII	Axborotalmashishning Amerikastandarti	American Standard Code for Information Interchange
LIFO	«Oxiridakeldi, birinchiketdi» prinsipi	Last In, First Out
FIFO	«Birinchi keldi, birinchi ketdi» prinsipi	First In, First Out
PDA	personal raqamli kotib	Personal Digital Assistant

“Dasturlash asoslari” faniga doir atamalar

O'zbek tilida	Rus tilida	Ingliz tilida
Algoritm	Алгоритм	Algorithm
Avtomatik dasturlash	Втоматическое программирование	Automatic programming
Blokli dasturlash	Блочное программирование	Block programming
Boshlash	Начать	Begin
Dasturlash	Программирование	Programming
Dastur	Программа	Programm
DLL	Динамически подключаемая библиотека	Dynamic link library
Forma	Форма	Form
Funksiya	Функция	Function
-gacha	До	Until
GDI	Интерфейс графического устройства	Graphic device interface
HDC	Ручка контекст устройства	Handle device context
Hodisa	Случай	Case
Hozircha	Пока	While
IDE (Integrallashgan dasturlash muhiti)	Интегрированная среда разработки	Integrated development environment
Import qilish	Импортировать	Import
Jamoatchilik	Общность	Public
Java	Ява	Java
Kod taxirlagichi	Редактор кода	Editor of the code
Kompilyator	Компилятор	Compiler
Kompilyatsiya	Компиляция	Compiling

Komponent	Компонент	Component
Konstanta va o'zgaruvchilar	Константы и переменные	Constants and variables
Konstruktor	Конструктор	Konstruktor
Ma'lumot tizimini yaratish	Создание системы отсчета	Making the reference system
Ma'lumotlar ombori	База данных	The database
Massiv	Массив	Array
Maydon	Поле	Field
Metod	Метод	Method
Obyekt	Объект	Object
Obyektlar inspektori	Инспектор объектов	Object inspector
Oyna deskriptori	Дескриптор окна	The descriptor window
Parallel dasturlash	Параллельное программирование	Concurrent programming
Polimorfizm	Полиморфизм	Polimorfizm
Prosedura	Процедура	Procedure
Rekursiya	Рекурсия	Recursion
Sinf	Класс	Class
Strukturali dasturlash	Структурное программирование	Structured programming
Strukturali kodlash	Структурное кодирование	Structured coding
Takrorlash	Повторять	Repeat
Tamom	Конец	End
Testdan o'tkazish	Тестирование	Testing
Tip	Тип	Type
Tugma	Кнопка	Button
Uchun	Для	For

Vizual komponentalar kutubxonasi	Библиотека визуальных компонентов	Visual components library (VCL)
Vorisiylik	Наследование	Inheritance
Xodisa	Событие	Event
Xossa	Свойства	Property
Этикетка	Этикетка	Label

Mundarija

	bet
Kirish	3
I-bob Algoritm va algoritmlash asoslari	5
1§. Algoritm	5
1.1. Masalalarni komputerdagi yechish bosqichlari	5
1.2. Algoritm tushunchasi.	6
1.3. Algoritmning asosiy xossalari	8
1.4. Algoritmning tasvirlash usullari	9
Nazorat uchun savollar	12
Mustaqil topshiriqlar	13
2§. Algoritm turlari. Algoritmlash jarayoni.	13
2.1. Chiziqli jarayonlarni algoritmlash	14
Nazorat uchun savollar	15
Mustaqil topshiriqlar	16
2.2. Tarmoqlanuvchi jarayonlarni algoritmlash	17
Nazorat uchun savollar	19
Mustaqil topshiriqlar	19
2.3. Takrorlanuvchi jarayonlarni algoritmlash	20
Nazorat uchun savollar	24
Mustaqil topshiriqlar	25
2.4. Murakkab hisoblash jarayonlarini algoritmlash	25
Nazorat uchun savollar	27
Mustaqil topshiriqlar	28
II-bob C++ tilida dasturlash asoslari	29
1§. C++ tilining asosiy elementlari	29
1.1. C++ dasturlash tili to'g'risida qisqacha ma'lumotlar	29
1.2. C++ dasturlash muhiti va unda ishlash qoidalari	29
1.3. C++ tili alfaviti va asosiy elementlari	39
1.4. C++ tilida ifodalar	40
1.5. O'zgaruvchilarning turlari	41
1.6. C++ tilida amallar	43
Nazorat uchun savollar	46
Mustaqil topshiriqlar	47
2§. C++ tilida standart funksiya va ifodalar	47
2.1. C++ tilida standart funksiyalar	47
2.2. C++ tilida ifodalar	48

2.3.	C++ tilida dasturning umumiy tuzulishi	49
	Nazorat uchun savollar	51
	Mustaqil topshiriqlar	52
3§.	C++ tilida oddiy operatorlar	52
3.1.	Qiymat berish buyrug'i	52
3.2	Ma'lumotlarni kiritish operatori	53
3.3	Ma'lumotlarni chiqarish operatori	53
3.4.	Sizeof amali	54
3.5.	C++ tilida dasturda izoh berish	54
	Nazorat uchun savollar	55
	Mustaqil topshiriqlar	55
4§.	C++ tilida murakkab operatorlar	55
4.1.	Shartsiz o'tish operatori.	55
4.2.	Shartli o'tish operatori	56
4.3.	Tanlash operatori	57
4.4.	Sikl operatorlari. Parametrli sikl operatori	57
4.5.	Sharti oldin qo'yilgan sikl operatori	58
4.6.	Sharti keyin qo'yilgan sikl operatori	58
4.7.	Continue va break operatorlari	59
	Nazorat uchun savollar	59
	Mustaqil topshiriqlar	60
5§.	C++ tilida massiv elementlari bilan ishlash	61
5.1.	Bir o'lchovli massiv elementlari bilan ishlash	61
5.2.	Ikki o'lchovli massiv elementlari bilan ishlash	63
	Nazorat uchun savollar	65
	Mustaqil topshiriqlar	65
6§.	C++ tilida funksiyalardan foydalanish	67
6.1.	Funksiyalarni tavsiflash.	67
6.2.	Funksiyalarga murojaat qilish.	68
6.3.	Rekursiya	68
	Nazorat uchun savollar	69
	Mustaqil topshiriqlar	69
III-bob.	Matematik masalalar uchun dasturlash tillarida dastur tuzish.	
3.1.	Matematik masalalar uchun chiziqli dastur tuzish	71
3.2.	Matematik masalalarga tarmoqlanuvchi algoritm va dastur tuzish	76

3.3.	Massiv elementlari bilan ishlash algoritmi va dasturini tuzish	86
IV-bob	Aralash va murakkab darajadagi masalalarga doir dastur tuzish	
4.1.	Aralash masalalarni C++ tilida yechish dasturlari	104
4.2.	Murakkab darajadagi masalalarni yechish algoritmi va dasturlari	121
V- bob.	Mustaqil dastur tuzishga doir masalalar.	
5.1.	Matematik ifodani dasturlash tilida yozish	131
5.2.	Chiziqli jarayonlarni dasturlashga doir masalalar	132
5.3.	Tarmoqlanuvchi jarayonlarni dasturlashga doir masalalar	133
5.4.	Takrorlanuvchi jarayonlarni dasturlashga doir masalalar	135
5.5.	Ichma-ich joylashgan sikllar bilan ishlashga doir masalalar.	136
5.6.	Massiv elementlari bilan ishlashga doir masalalar.	137
5.7.	Funksiyadan foydalanib dastur tuzishga doir masalalar.	140
5.8.	Ko'rsatkich funksiyalardan foydalanib dastur tuzish.	143
5.9.	Murakkab darajadagi masalalarga dastur tuzish.	144
V- bob.	Nazariy va amaliy bilimlarni mustahkamlashga doir test savollari.	
6.1.	Algoritm va algoritmlash jarayoniga doir test savollari	149
6.2.	C++ dasturlash tiliga doir test savollari	152
	Foydalanish uchun adabiyotlar	167
	Ilova	168

Aminov I.B., Suyarov A.M.

**“Dasturlash tillari asosida
matematik masalalarni yechish”**

o’quv qo’llanma

Muharrir
Musahhih
Texnik muharrir

J. Bozorova
L. Xoshimov
N.Isroilov

2019 yil 26 oktabrda tahririy-nashriyot bo’limiga qabul qilindi.
2019 yil 9 noyabrda original-maketdan bosishga ruxsat etildi.
Qog’oz bichimi 60x84_{1/8}. “Times new roman” garniturasini. Offset
qo’g’ozi. Shartli bosma tabog’i – 11,25.
Adadi 50 nusxa. Buyurtma № 11/8.

ISBN – 978-9943-537-74-3

SamDU tahririy-nashriyot bo’limida chop etildi.
140104, Samarqand sh., Universitet xiyoboni, 15.

