

Nurullayev M.M.



AXBOROT TEXNOLOGIYALARI VA JARAYONLARNI MATEMATIK MODELLASHTIRISH

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

BUXORO MUHANDISLIK-TEXNOLOGIYA INSTITUTI

Nurullayev M.M.

**AXBOROT TEXNOLOGIYALARI VA
JARAYONLARNI MATEMATIK
MODELLASHTIRISH**

5310600 – Transport vositalari muhandisligi (avtomobil transporti) bakalavriat
ta`lim yo`nalishida tahsil olayotgan talabalar uchun
o`quv qo'llanma

Toshkent - 2022

Axborot texnologiyalari va jarayonlarni matematik modellashtirish.
Nurullayev M.M.

T a q r i z c h i l a r :

X.F.Djo'rayev – Buxoro muhandislik-texnologiya instituti “Texnologik jarayonlarni boshqaruv tizimlari” fakulteti dekani, texnika fanlari doktori, professor.

M.O'.Xudoyberganov – Mirzo Ulug'bek nomidagi O'zbekiston milliy universiteti “Hisoblash matematikasi va axborot tizimlari” kafedrası mudiri, fizika-matematika fanlari nomzodi, dotsent.

O'quv qo'llanmada axborot texnologiyalari, dasturiy ta'minot haqida tushunchalar, algoritmlash va dasturlash, C++ dasturlash tili haqida ma'lumotlar, jumladan, C++ dasturlash tili va uning asosiy operatorlari, uning kiritish va chiqarish operatorlari, tarmoqlanish va takrorlanish operatorlari, massivlar bilan ishlash kabi ma'lumotlar berilgan. Bundan tashqari, matematik modellarni elektron jadvaldagi tadqiqoti hamda MS Excel da matematik modellashtirishning muayyan elementlari keltirilgan. Shu bilan birga matematikaning sodda, oddiy matematik ifodalarni MathCAD tizimida hisoblash, unda ikki va uch o'lchovli grafiklar qurish, integral va limitlarni hisoblash, matritsalar ustida amallar bajarish, tenglamalar va tenglamalar sistemasini yechish hamda masalalarning analitik yechimlarini qurish kabi masalalarni yechish proseduralari to'g'risida ma'lumotlar va vizual materiallar berilgan.

В учебном пособии дается информация об информационных технологиях, понятиях программного обеспечения, алгоритмизации и программировании, в том числе о языке программирования C++ и его основных операторах, таких как операторы ввода-вывода, операторы ветвления и повторения, работа с массивами. Кроме того, представлена разработка математических моделей в электронной таблице, а также приведены определенные элементы математического моделирования в MS Excel. В то же время даны сведения и наглядные материалы о процедурах решения простейших математических выражений в системе MathCAD, построения двух - и трехмерных графиков, вычисления интегралов и пределов, выполнения операций над матрицами, решения уравнений и систем уравнений, а также построения аналитических решений задач.

The training manual provides information about information technology, concepts about software, algorithm and programming, information about the C++ programming language, including the C++ programming language and its main operators, its input and output operators, conditional expressions and loops, working with arrays. In addition, the study of mathematical models in a spreadsheet, as well as certain elements of mathematical modeling in MS Excel, is presented. At the same time, information and visual materials are given about the procedures for solving the simplest mathematical expressions in the MathCAD system, constructing two-and three-dimensional graphs in it, calculating integrals and limits, performing operations on matrices, solving equations and systems of equations, as well as constructing analytical solutions to problems.

KIRISH

Bugungi kunda ishlab chiqarishning barcha sohalarida axborot texnologiyalaridan keng foydalanilmoqda. Zamonaviy texnik tizimlaridagi barcha muhandislik muammolari yangi yondashuvni talab etadi. Avtomatlashtirilgan loyihalash tizimlari, Geometrik modellash dasturlari, tarmoqdagi ma'lumotlar bazalarini ishlab chiqarish jarayonidagi va texnik tizimlardagi muammolarni yechishda samarali qo'llanilmoqda. Shuningdek, hozirda talabalarning yangi bilimlarni egallashida ta'lim jarayonida va mustaqil ta'limda axborot texnologiyalaridan unumli foydalana olishi va zamonaviy texnologiyalarni tadbiq etishi talab qilinadi.

Axborot kommunikatsiya texnologiyalari tez rivojlanayotgan yo`nalish bo`lib, hozirda u qamrab olmagan biror sohani topish qiyin. Texnik boshqaruv tizimlarida va muhandislik masalalarini yechishda axborot texnologiyalarining o`rni katta. "Axborot texnologiyalari va jarayonlarni matematik modellashtirish" fanini o`qitishdan maqsad zamonaviy axborotli fikrlash va ilmiy dunyoqarashni shakllantirilgan holda talabalarni kompyuter imkoniyatlaridan foydalanish, kompyuter bilan muloqot o`rnatish usullarini o`rgatish va unda turli masalalarni yecha olishga yo`naltirishdan iborat. Jumladan, yo`nalishlar sohalaridagi masalalarni yechishda, soha masalalarini loyihalashtirishda, tizimli tahlilda, tizim va kommunikatsiyani loyihalashda, muhandislik tahlilida, soha muhandislik ishlarini loyihalash jarayonlarini avtomatlashtirishda, texnik tizimlarda ma'lumotlar almashish jarayonida zamonaviy dasturiy vositalardan va texnologiyalardan hamda Internet texnologiyalaridan mukammal foydalanish katta ahamiyatga ega.

O`quv qo'llanmada axborot kommunikatsiya texnologiyalari, algoritmlash va dasturlash, C++ dasturlash tili haqida ma'lumotlar, jumladan, C++ dasturlash tili va uning asosiy operatorlari, uning kiritish va chiqarish operatorlari, tarmoqlanish va takrorlanish operatorlari, massivlar bilan ishlash kabi ma'lumotlar berilgan. Bundan tashqari, matematik modellarni elektron jadvaldagi tadqiqoti, hamda MS Excelda matematik modellashtirishning muayyan elementlari keltirilgan, shu bilan

birga matematikaning eng sodda, oddiy matematik ifodalarini hisoblash, bir va ikki o`lchamli grafiklar qurish, integral va limitlarni hisoblash, matritsalar ustida amallash bajarish, chiziqli va chiziqsiz tenglamalarni yechish, tenglamalar sistemasini yechish, differensial tenglamalarni chegaraviy va boshlang`ich shartlar bilan yechish, tajriba natijalarini tahlil qilishda interpolatsiyalash, regressiya tenglamalarini qurish va chiziqli dasturlash masalalarini yechish, hamda masalalarning analitik yechimlarini qurish kabi masalalarni yechish protseduralari keltirilgan. Ularni yechishda MathCAD dasturidan qanday foydalanish yo`llari haqida ham ma'lumot berilgan.

**AXBOROT TEXNOLOGIYALARI, AXBOROTLI JARAYONLAR VA
AXBOROTLASHGAN JAMIYAT HAQIDA TUSHUNCHA.**

**“Axborot texnologiyalari va jarayonlarni matematik modellash”
fanining mohiyati, maqsadi va vazifalari.**

“Informatika” atamasi XX asrning 60 yillarida birinchi marta Frantsiyada qo'llanilgan. U Information (axborot) hamda Automatique (avtomatika) so'zlarining qo'shilmasidan hosil bo'lgan Informatique (informatika) deb olingan. Bu axborotga avtomatik ravishda ishlov berish fani degan ma'noni anglatadi. G'arbiy Evropaning ko'pchilik mamlakatlarida hamda AQSh va Kanadada informatika termini o'rniga uning sinonimi *Computer science* (kompyuter haqidagi fan) – tushunchasi qo'llaniladi. Informatika kompyuter texnikasining rivojlanishi tufayli yuzaga keldi, unga asoslanadi, usiz mavjud bo'la olmaydi va o'z navbatida uning rivojiga, yangilanishiga o'z hissasini qo'shadi. Uning asosiy resursi – axborotdir. Informatika ko'p qirrali funksiyalarga ega bo'lgani uchun unga har xil ta'rif berilgan.

Informatika – ma'lumotlarni hosil qilish, saqlash, uzatish va qayta ishlash usullarini hisoblash texnikasi vositalari yordamida tizimlashtiruvchi, aloqadorligini o'rnatuvchi va ularni boshqaruvchi inson faoliyatining aniq sohasidir.

Informatika - ilmiy axborotlarning struktura va umumiy xossalarini, shuningdek ularni yaratish, o'zgartirish, uzatish va ulardan inson faoliyatining turli sohalarida foydalanish qonuniyatlarini o'rganadigan fan.

Informatika — bu axborotning nafaqat umumiy xususiyatlari, balki unga avtomatlashtirilgan ishlov berishning uslublari, jarayonlari va texnik vositalarini o'rganuvchi fandir.

Informatika - axborotlar haqidagi fan bo'lib, uning qayta ishlashdagi takomillashgan usul va vositalarini o'rgatadi.

Informatika - axborotlarni kompyuter yordamida tasvirlash, saqlab turish, uzatish va ishlov berish usullarini o'rganadigan fandir.

1973 yildan boshlab EHM tarixining yangi sahifasi, shaxsiy kompyuterlar sahifasi boshlandi va kompyuter so’zi ishlatila boshlandi. (Kompyuter inglizcha hisoblovchi demakdir).

Hozirgi kunda “Informatika” atamasi o’rniga ko’proq “Axborot texnologiyalari” (Information technology) atamasi ishlatilmoqda.

Texnologiya so’zi grekchadan (techne) tarjima qilinganda san’at, mahorat, ko’nikma kabi ma’nolarni anglatadi.

Bu ma’lum bir jarayonga nisbatan qo’llanilgan. Jarayon deganda esa maqsadga erishishga yo’naltirilgan hatti harakatlar majmui tushunilgan. Ushbu jarayon kishi tomonidan tanlangan strategiya bilan belgilanadi va turli xildagi vositalar, usullar yordamida amalga oshiriladi.

Axborot texnologiyalari to’g’risida gap ketganda, qayta ishlashning materialli sifatida ham, mahsulot sifatida ham axborot ishtirok etadi. Biroq bu ob’ekt, jarayon yoki hodisa to’g’risidagi sifat jihatidan yangi ma’lumot bo’ladi. Texnologiya insonning axborot bilan ishlash usuli va uslubi hamda texnik vositalar orqali namoyon bo’ladi.

Axborot texnologiyalari ko’pqirrali funksiyalarga ega bo’lib, u axborotlarga ishlov berishda kompyuterlardan foydalanish, hisoblash texnikalari va aloqa texnikalarini qamrab oladi. Unga ham har xil ta’rif berilgan, ya’ni:

Axborot texnologiyalari - axborotni to’plash, saqlash, izlash, unga ishlov berish va uni tarqatish uchun foydalaniladigan jami uslublar, qurilmalar, usullar va jarayonlardir. (Qonun)¹.

Axborot texnologiyalari - ma’lumotlarni to’plash, ularga ishlov berish, saqlash, uzatish va ulardan foydalanish jarayonida hisoblash texnikasidan foydalanish yo’llari, usullari va uslublaridir.

Axborot texnologiyalari - hujjatlashtirilgan axborot, jumladan dasturli vositalarga ishlov berishning jami usublari, yo’llari, usullari va vositalari hamda ulardan foydalanishning belgilangan tartibidir.

¹ O’zbekiston Respublikasining “Axborotlashtirish to’g’risida” gi Qonuni. Toshkent shahri, 1993 yil, may

Axborot texnologiyalari - inson faoliyatining turli sohalarida axborot mahsulotini ishlab chiqarishda axborot jarayonlarini amalga oshirishning jami usullaridir.

Axborot texnologiyalari - insonlar tomonidan axborotni yig'ish, saqlash, ishlov berish va tarqatish uchun foydalaniladigan jami uslublari, qurilmalar va ishlab chiqarish jarayonlaridir.

Axborot texnologiyalari - ob'ekt, jarayon yoki hodisa (axborot mahsulot) holati haqida yangi sifatdagi ma'lumotlarni olish uchun foydalanadigan ma'lumotlarni (birlamchi) yig'ish, ishlov berish va o'zlash vositalari, hamda usullari majmuasidir.

Axborot texnologiyalari - axborot oqimini yig'ish, saqlab turish, ishlov berish va uzatishni kompyuterlar yordamida amalga oshiradigan usullar majmuasidir.

Axborot texnologiyalari o'z-o'zicha mavjud bo'lmay, balki boshqa sohalaridagi muammolarini hal etish uchun yangi axborot texnika va texnologiyalarini yaratishga qaratilgan kompleks ilmiy – texnik sohadir. U boshqa sohalar, hatto jarayonlar va hodisalar noformallashuvi tufayli miqdoriy uslublarni qo'llash mumkin emas deb hisoblanadigan sohalariga ham tadqiqot uslub va vositalarini taqdim etadi. Axborot texnologiyalarida kompyuter texnikasi sharofati tufayli amaliy ro'yobga chiqish mumkin bo'lgan matematik modellash uslublarining hal qilinishini alohida ajratib ko'rsatish lozim.

Axborot texnologiyalari *amaliy fan sohasi* sifatida quyidagilar bilan shug'ullanadi:

- axborot jarayonlaridagi qonuniyatlarni o'rganish (axborotlarni yig'ish, qayta ishlash, tarqatish);
- inson faoliyatining turli sohalarida kommunikatsion – axborot modellarini yaratish;
- aniq bir sohalarida axborot tizimi va texnologiyalarini ishlab chiqish va ularning hayotiy bosqichini, ularni ishlab chiqarish, ishlashni va hokazolarni loyihalash, ishlab chiqish bosqichlari uchun tavsiyalar tayyorlash.

Matematik ta'minot esa - bu axborot tizimi ishlab turishi uchun zarur bo'lgan matematik usullar, modellardan iborat. Matematik ta'minotga quyidagilar kiradi:

- Boshqaruv jarayonlarini modellashtirish vositalari;
- Tipik masalalarni tayyor yechish usul va vositalari (jumladan, matematik analiz, algebra, hisoblash usullari, differensial va integral tenglamalarga oid tayyor algoritm va usullar);
- Matematik dasturlash, matematik statistika, ommaviy xizmat ko'rsatish va boshqa usullar.

“Axborot texnologiyalari va jarayonlarni matematik modellashtirish” fanini o'qitishdan maqsad talabalarni axborotni to'plash, saqlash, qayta ishlash va uzatishning asosiy qonuniyatlarini nazariy va amaliy masalalarda qo'llash uchun zarur bo'lgan asosiy tushunchalar bilan tanishtirishdan, yangiliklarni mustaqil o'rganib, uning tadbirlarini o'zlashtirishga o'rgatishdan, talabalarning mantiqiy fikrlash qobiliyatlarini o'stirishdan, ijodiy izlanishlarga va muhandislik masalalar yechimini axborot texnologiya tushunchalari yordamida bayon qilishga o'rgatishdan iborat.

Bu fan talabalarda ilmiy tafakkur va dunyoqarashni rivojlantirish asosida matematik modellar va ilmiy farazlarning qo'llanish chegaralari to'g'risida aniq tushunchalar hosil qilgan holda, uning qonuniyatlarini ongli ravishda o'zlashtirish hamda olingan bilimlarni mustaqil ravishda texnologik jarayonlarni boshqarishda va apparatlaridan foydalanishga qo'llashga qaratilgan.

Fanning maqsadi - bo'lajak muhandis-texnologlar kompyuterning yangi dasturiy ta'minotlarini, axborot tizimlarini, zamonaviy axborot kommunikatsiya texnologiyalarini yaxshi bilishi va kelajakda ushbu bilimlardan samarali foydalangan holda turli shakldagi axborotlarni qayta ishlash bo'yicha bilimlar darajasi bilan ta'minlash, hamda bo'lajak mutaxassislarda zamonaviy informatsion fikrlash va ilmiy dunyoqarashni shakllantirish, turli texnik bilimlarga oid

masalalarni kompyuter yordamida hal etish va informatsion modellashtirish ko'nikmalarini shakllantirishdan iborat.

Fanning vazifasi – “Axborot texnologiyalari va jarayonlarni matematik modellashtirish” fanining mohiyati, uning asosiy tamoyillari, vazifalari, yangi dasturiy ta'minotlar, axborot tizimlari va axborot kommunikatsiya texnologiyalarini rivojlantirishning maqsad va vazifalari haqida tushuncha berish. Fanning dasturlash asoslari qismida dasturlash tilining tuzilmasi (C++ algoritmik tili), funksiyalari va asosiy parametrini o'rganish, shuningdek bu algoritmik tilning barcha imkoniyatlarini talabalarga o'rgatish. MathCAD tizimida matematik modellashtirishga doir bilimlarni zamonaviy pedagogik texnologiyalarga asoslangan holda o'qitish, talabani ushbu fan bo'yicha olgan bilimlarini real sharoitlarga qo'llash bo'yicha ko'nikmalar hosil qilishdir.

**Zamonaviy axborot texnologiyalarining hozirgi zamondagi o'rni va
rivojlanish istiqbollari hamda ularning fan-texnika taraqqiyotida, jamiyat
rivojida tutgan o'rni**

Bugungi kunda, biz kuzatayotgan axborot-kommunikatsiya texnologiyalarining jadal rivojlanishi natijasida, shubhasiz quyidagilar kuzatish mumkin:

- yangi axborot va kommunikatsiya texnologiyalari shaxsiy kompyuter va internet ixtiro qilinishi sababli, jamiyat hayotiga juda tez, ya'ni bir inson umri davomida kirib keldi;
- axborotlar bilan ishlash kontsepsiyasining oddiyligi ayniqsa tasvir, ovoz va multimedia bilan ishlashni osonlashtirdi;
- kommunikatsiya usuli o'zgarib, uning imkoniyatlari kengayishi natijasida, yangi texnologiyalar fan, ta'lim, madaniyat va siyosatga sezilarli darajada ta'sir ko'rsatishi kuzatildi.

Respublikada kompyuterlashtirish va axborot kommunikatsiya texnologiyalarini rivojlantirishga doir muayyan vazifalar O'zbekiston Respublikasi Prezidenti tomonidan 2002 yil 30 mayda imzolangan “Kompyuterlashtirishni yanada rivojlantirish va axborot-kommunikatsiya texnologiyalarini joriy etish

Zamonaviy axborot texnologiyalarining hozirgi zamondagi o'rni va rivojlanish istiqbollari hamda ularning fan-texnika taraqqiyotida, jamiyat rivojida tutgan o'rni to'g'risida" gi Farmonida belgilab berilgan bo'lib, O'zbekiston Respublikasi Vazirlar Mahkamasining 2002 yil 6 iyun 200-sonli "Kompyuterlashtirishni yanada rivojlantirish va axborot-kommunikatsiya texnologiyalarini joriy etish chora-tadbirlari to'g'risida"gi Qarorida o'z aksini topgan.

a. Respublikada axborot-kommunikatsiya texnologiyalari sohasidagi Qonunlar.

- Axborotlashtirish to'g'risidagi qonuni (1993 yil, may.)
- EHM va ma'lumotlar bazasi uchun dasturlarni huquqiy muhofazalash haqida (1994 yil, may)
- Elektron raqamli imzo to'g'risidagi qonun (2003 y.)
- Elektron hujjat aylanishi to'g'risidagi qonun (2004 y.)
- Elektron tijorat to'g'risidagi qonun (2004 y.)
- Elektron to'lovlar to'g'risidagi qonun (2005 y.)

b. O'zbekiston Respublikasi Prezidentining farmonlari.

- Telekommunikatsiyalar sohasida boshqaruvni takomillashtirishga doir chora-tadbirlar to'g'risida (2000 y.)
- Kompyuterlashtirishni yanada rivojlantirish va axborot-kommunikatsiya texnologiyalarini joriy etish to'g'risida (2002 y.)
- Axborot-kommunikatsiya texnologiyalarini yanada rivojlantirishga oid qo'shimcha chora-tadbirlar to'g'risida (2005 y.)
- O'zbekiston Respublikasining jamoat ta'lim axborot tarmog'ini tashkil etish to'g'risida (2005 y.)
- Respublika aholisini axborot-kutubxona bilan ta'minlashni tashkil etish to'g'risida (2006 y.)

v. O'zbekiston Respublikasi Prezidenti va O'zbekiston Respublikasi Vazirlar mahkamasining qarorlari va sohaga oid me'yoriy hujjatlari.

- 2001-2005 yillarda kompyuter va axborot texnologiyalarini rivojlantirish, "Internet"ning xalqaro axborot tizimlariga keng kirib borishini ta'minlash dasturini ishlab chiqishni tashkil etish chora-tadbirlari to'g'risida (2001

yil)

- Kompyuterlashtirishni yanada rivojlantirish va axborot-kommunikatsiya texnologiyalarini joriy etish chora-tadbirlari to'g'risida (2002 y.)
- Axborotlashtirish sohasida normativ-huquqiy bazani takomillashtirish to'g'risida (2005 y.)
- Davlat va xo'jalik boshqaruvi, Mahalliy davlat hokimiyati organlarining axborot-kommunikatsiya texnologiyalaridan foydalangan holda yuridik va jismoniy shaxslar bilan o'zaro aloqadorligini yanada takomillashtirish chora-tadbirlari to'g'risida (2007 y.)
- Internet tarmog'ida O'zbekiston Respublikasining Hukumat portalini yanada rivojlantirish chora-tadbirlari to'g'risida (2007 y.)
- Internet tarmog'ida O'zbekiston Respublikasining Hukumat portaliga axborotlarni taqdim etish va joylashtirish tartibi to'g'risida (2009 y.)

AKTni davlat boshqaruviga joriy etish

- o Davlat xodimlarini kompyuter texnikasi bilan ta'minlash - 90 % (2010 yil boshida 88,5%);
- o Markaziy apparatlarda lokal tarmoqlarni yaratish - 98,3% yaratilgan (2010 yil boshida 90,2%);.
- o Xodimlarning kompyuterda ishlash ko'nikmalarini egallash darajasi - 68,3% (2010 yil boshida 65%);
- o Idora ichida qog'ozsiz hujjat almashish ulushi 53%ni tashkil qiladi (2010 yil boshida 52%);
- o ".uz" domeni zonasida ro'yxatga olish narxi 15 AQSh dollaridan milliy valyutaga – so'mga o'tkazilib qiymati 6 ming so'mga (35%ga) tushirildi;
- o 2010 yilda qo'shimcha 1,5 ming veb-sayt ro'yxatdan o'tib jami domen nomlari soni 11 mingtani tashkil etdi (o'sish 16%).

Axborotlashgan jamiyat

Mutaxassislar, guruhlar, yirik ishlab chiqarish korxonalari yoki muassasalarining faoliyati ko'p jihatdan ularning qay darajada zaruriy ma'lumot va

axborotlar bilan to'la ta'minlanganligiga, hamda ushbu ma'lumotlardan qay darajada samarali foydalana olayotganliklariga bog'liq bo'lib qolmoqda. Avvalo bir yechimga kelishdan, qaror qilishdan oldin ushbu masalaga (sohaga) aloqador bo'lgan juda ko'plab ma'lumotlarni to'plash, ularni qayta ishlash va tahlil qilish zarur bo'ladi. Ba'zan bunday ma'lumotlar to'plami shu qadar ko'payib ketadiki, ularni qayta ishlash va tahlil qilishni maxsus texnik tizimlar yordamisiz amalga oshirib bo'lmay qoladi. Bundan tashqari, kundalik hayotda qabul qilish va qayta ishlash zarur bo'lgan axborotlar hajmi nihoyatda ortib borayotganligi sababli, ba'zan ularni qabul qilishga ham ulgurilmayapti. Axborot va ma'lumotlar hajmining bu qadar ko'payib ketishi, oqimining esa tezlashib borishini asosiy sabablaridan biri axborot texnika va texnologiyalarning rivojlanishi zamonaviy eng yangi texnologiyalarini qo'llash, ikkilamchi xomashyolardan oqilona foydalanish, energetik resurslarni tejamkorlik bilan ishlatish, inson mehnatini engillashtirish hisobiga oshirish bosqichiga kirganligidir. Bu esa, o'z navbatida, jamiyatni yuqori darajada axborotlashgan bo'lishini talab etadi.

*Jamiyatni axborotlashtirish – yuridik va jismoniy shaxslarning axborotga bo'lgan ehtiyojlarini qondirish uchun axborot resurslari, axborot texnologiyalari hamda axborot tizimlaridan foydalangan holda sharoit yaratishning tashkiliy, ijtimoiy-iqtisodiy va ilmiy-texnikaviy jarayonidir.*²

Jamiyatni axborotlashtirish jarayoni quyidagi qator muammolarni hal etilishini talab etadi:

1. Hisoblash texnika vositalarini jamiyat faoliyatining barcha tarmoqlariga tadbiq qilish.
2. Jamiyat a'zolarini hisoblash texnikasi vositalaridan samarali foydalanishga o'rgatish.
3. Jamiyat a'zolarining turli xil ehtiyojlarini qondirishda axborot resurslaridan to'la va samarali foydalanishlarini ta'minlash.

Jamiyatning axborotlashuvi atamasi D.Benk va E.Masudolar tomonidan ilk bor qo'llanilgan bo'lib, bugungi kunda axborotlashgan jamiyat nomini oldi.

² O'zbekiston Respublikasining "Axborotlashtirish to'g'risida" gi Qonuni. Toshkent shahri, 1993 yil 1 may

Axborotlashgan jamiyat haqida olimlar turlicha fikr yuritadilar. Masalan yapon olimlarining hisoblashicha, *axborotlashgan jamiyatda kompyuterlashtirish jarayoni odamlarga ishonchli axborot manbaidan foydalanish, ishlab chiqarish va ijtimoiy sohalarda axborotni qayta ishlashni avtomatlashtirishning yuqori darajasini ta'minlashga imkon beradi*. Jamiyatni rivojlantirishda esa harakatlantiruvchi kuch moddiy mahsulot emas, balki axborot ishlab chiqarish bo'lmog'i lozim.

Axborotlashgan jamiyatda nafaqat ishlab chiqarish, balki butun turmush tarzi, qadriyatlar tizimi ham o'zgaradi. Barcha harakatlar tovarlarni ishlab chiqarish va iste'mol etishga yo'naltirilgan sanoat jamiyatiga nisbatan axborotlashgan jamiyatida intellekt, bilimlar ishlab chiqariladi va iste'mol etiladiki, bu hol aqliy mehnat ulushining oshishiga olib keladi. Insondan ijodiyotga qobiliyat talab etiladi, bilimlarga ehtiyoj oshadi.

Axborotlashgan jamiyatining moddiy va texnologik negizini kompyuter texnikasi va kompyuter tarmoqlari, axborot texnologiyalari, telekommunikatsiya aloqalari asosidagi turli xil tizimlar tashkil etadi.

Axborotlashgan jamiyat - jamiyatning ko'pchilik a'zolari axborot, ayniqsa uning oliy shakli bo'lmish bilimlarni ishlab chiqarish, saqlash, qayta ishlash va amalga oshirish bilan band bo'lgan jamiyatdir.

Axborotlashgan jamiyatga o'tishda kompyuter va telekommunikatsiya axborot texnologiyalari negizida yangi axborotni qayta ishlash sanoati yuzaga keladi.

Axborotlashgan jamiyat quyidagi jihatlarda namoyon bo'ladi:

- texnik qurilmalardan foydalanish ko'nikmalariga ega bo'lish;
- o'z faoliyatida kompyuter, axborot texnologiyalaridan foydalanish;
- turli manbalardan axborotni olishni bilish va undan samarali foydalanish;
- axborotni tahliliy qayta ishlash asoslarini egallash;
- o'z faoliyatiga taalluqli axborotni bilish va u bilan ishlashni uddalash.

Hozirgi paytda shu narsa ravshan bo'lib qolmoqdaki, u yoki bu mamlakat XXI asrda munosib o'rin egallashi va boshqa mamlakatlar bilan iqtisodiy

musobaqada teng qatnashishi uchun, o'z iqtisodiy tuzilishi, ustivorliklari, boyliklari, institutlarini qayta qurish va sanoatini axborot tizimlari talablariga moslashtirishi kerak.

Axborotlashgan jamiyatning shakllanishi davlat va jamiyat hayotining barcha sohalarida jiddiy o'zgarishlarga olib kelmoqda. Hozirda axborot odamlar va jamiyat faravonligining eng muhim manbaasiga aylanmoqda, rivojlangan davlatlar iqtisodiyoti tarkibida axborot texnologiyalari industriyasi eng yirik sohani tashkil etadi.

Axborot madaniyati va etikasi

Axborotlashgan jamiyatda inson axborot bilan ishlash bo'yicha ma'lum darajada kompyuter savodxonligi, axborot madaniyati va etikasiga ega bo'lishi zarur. Shu sababli hozirgi kunda axborotlashgan jamiyatda kompyuter savodxonligi, axborot madaniyati va etikasi atamasi ko'p ishlatilmoqda.

Kompyuter savodxonligi – kompyuterda ishlash uchun zarur bilim va ko'nikmalarning eng kam to'plamini egallashdir.

Axborotlashgan jamiyatda barcha fuqarolar, tashkilotlar va davlatning axborotga bo'lgan ehtiyojini qondirish uchun hamma sharoit yaratilgan bo'ladi. Mehnat qiluvchilarning ko'pchiligi yoki axborot ishlab chiqarish, saqlash, qayta ishlash va sotish bilan band bo'ladi yoki bu jarayonlarsiz ishlab chiqarish majburiyatlarini bajara olmaydigan bo'ladi. Bu shuni anglatadiki, bunday jamiyat fuqarolari axborot madaniyatiga ega bo'ladi. Ya'ni axborot bilan ishlashni, uni olish, qayta ishlash va uzatish uchun axborot tizimlari va texnologiyalaridan foydalanishni biladilar. Bu inson faoliyatining barcha jabhalariga oid boy bilimlar, ishonchli axborotlardan to'liq va o'z vaqtida foydalanishni ta'minlashga qaratilgan kompleks chora-tadbirlarni joylarda tatbiq etishni anglatadi.

Axborot madaniyati - jamiyatning o'z ixtiyorida bo'lgan axborot resurslari va axborot kommunikatsiya vositalaridan samarali foydalanish qobiliyatidir.

Axborot madaniyati bu kompyuterlardan umuminsoniy qadriyatlar nuqtai nazariga muvofiq foydalanish etikasidir.

Axborot madaniyatida estetik xususiyat ham muhim hisoblanadi. Masalan, matnlarni ma'lum bir formatda tayyorlash, so'zlar orasida ortiqcha probellarning bo'lmasligi, bir xil abzatslarni ishlatish yoki ekran namoyishlarida ob'ektlarni o'z o'rniga muvofiq ravishda qo'ya olish va ranglardan foydalanish singarilar.

Kompyuter etikasi

Kompyuter etikasi bu - kompyuterlardan foydalanishni tartibga solish uchun bayon qilingan tamoyillar to'plamidir.

Kompyuter etikasida uchta faktor ko'rib chiqiladi:

- *Intellektual mulk huquqi.*
- *Maxfiylik muammosi.*
- *Kompyuterlarning jamiyatga ta'siri.*

Intellektual mulk huquqi – masalan, egasining ruxsatisiz dasturiy ta'minotdan nusxa olish kabilarni qamrab oladi.

Maxfiylik muammosi – masalan, xakerlik, yoki boshqa noqonuniy tarzda insonning shaxsiy ma'lumotlariga murojaat qilish kabilarni qamrab oladi.

Kompyuterlarning jamiyatga ta'siri – ishchi o'rinlarni yo'qotish, ijtimoiy zarbalar va boshqa faktorlarni qamrab oladi.

Internetdan foydalanish – plagiat (egasining ruxsatisiz ko'chirmachilik) ning yanada ko'payib ketishiga olib keldi, ya'ni, boror bir insonning g'oyasi yoki ishini olgan holda o'ziniki qilib olish kuchayib ketdi.

ALGORITMLASH VA DASTURLASH ASOSLARI.**Kompyuterda masalalarni yechish bosqichlari**

Kompyuterda masalalarni yechish dasturlarini yaratish jarayonida nazariy va amaliy nuqtai nazardan algoritmlash, dasturlash va kompyuterlarni qo'llash bilan bog'liq bo'lgan bilimlar kerak bo'ladi. Bu yerda asosiy maqsad masalani qo'yish, masalaning yechish algoritmini tuzish, algoritmnı mashina dasturi ko'rinishida amalga oshirish, dasturning to'g'riligini tekshirish (algoritmnı samaradorligini ko'rsatish) va dasturnı foydalanishga topshirishdan iborat. Bu jarayonlar dasturnı to'liq yaratish tushunchasiga olib keladi va quyidagi bosqichlarnı belgilaydi:

1. Masalaning qo'yilishi.
2. Modelni yaratish.
3. Algoritmnı ishlab chiqish.
4. Algoritmnı tahlil qilish va to'g'riligini tekshirish.
5. Dastur tuzish.
6. Dasturnı ishga tushirish va tekshirish.
7. Dasturnı foydalanishga topshirish.

1. Masalaning qo'yilishi. Masalani yechishdan oldin, uni berilishini aniq shakllantirib olish zarur. Bu jarayon to'g'ri savollarnı aniqlash bo'lib, savollar quyidagicha bo'lishi mumkin:

- 1.1. Dastlabki berilgan masala shartlarida hamma iboralar tushunarlimi?
- 1.2. Nima berilgan?
- 1.3. Nimani topish kerak?
- 1.4. Yechimni qanday ta'riflash kerak?
- 1.5. Qaysi ma'lumotlar yetarli emas va hammasi kerakmi?
- 1.6. Qanaqa shartlar qabul qilingan?

Albatta, bulardan tashqari boshqa savollar ham tug'ilishi mumkin, yoki ayrim savollarnı bir necha bor takror ishlatishga to'g'ri keladi.

Bu bosqichda masalaga tegishli barcha ma'lumotlar yig'iladi, o'rganib chiqiladi va tahlil qilinadi. So'ngra yechishning maqsadi, ya'ni masalani yechish

natijasida ayni nimani yoki nimalarni aniqlash zarurligi ko'rsatiladi hamda boshlang'ich ma'lumotlar yoki argumentlar hamda qiymatlari aniqlanishi kerak bo'lgan kattaliklar, ya'ni natijalar ajratiladi.

2. *Modelni yaratish.* Matematik model tashqi dunyoning hodisalar turkumini matematik belgilar yordamida taxminiy tavsiflashdir. Hodisani tavsiflash uchun uning muhim xususiyatlarini, qonuniyliklarini, ichki aloqalarini, ayrim xossalarning ahamiyatini aniqlash zarur. Eng muhim faktorlari aniqlanganda, ahamiyatlari kamroq bo'lganlarini hisobdan chiqarish mumkin. Umuman, modelni tanlash fandan ko'ra, ko'proq san'at ishi deb hisoblanadi. Yaxshi tuzilgan modellarni o'rganish modellashtirishda tajriba orttirishning eng yaxshi usuli hisoblanadi. Modelni yaratishda quyidagi savollarni aniqlash maqsadga muvofiq:

2.1. Masalani yechish uchun qaysi matematik uslub ko'proq mos keladi?

2.2. O'xshash masalaning yechimi bormi?

2.3. Masalaning barcha muhim ma'lumotlari matematik ob'yektlar orqali tavsiflanadimi?

2.4. Izlanayotgan natija biron bir matematik o'lchamga mos keladimi?

2.5. Modelning ob'yektlari orasidagi bog'lanishlar aniqlanganmi?

2.6. Tuzilgan model bilan ishlash qulaymi?

Bu yerda tekshirilayotgan ob'ektni matematik atamalarda tavsiflash, ya'ni uning matematik modelini qurish kerak. (Misol, $S=ab$ formula to'g'ri to'rtburchakning yuzini hisoblaydi).

3. *Algoritmni ishlab chiqish.* Masalaning matematik modeli yaratilgandan so'ng, yechish usulini izlash boshlanadi. Ayrim hollarda, masalaning qo'yilishidan keyin to'g'ridan-to'g'ri, masalani yechish usuliga ham o'tish kerak bo'ladi. Bunday masala oshkor ko'rinishdagi matematik model bilan ifodalanmasligi mumkin. Bu holda uni hal qilish usullari bayon qilinadi. Bu bosqichda masalani kompyuterdan foydalanib yechishning algoritmi tuziladi. Algoritm tuzish jarayoniga algoritmlashtirish deyiladi. Har bir algoritmni ishlab chiqish bevosita o'ziga xos yondashishni talab qilishiga qaramasdan, bu faoliyatni umumiy uslub va bosqichlari ham mavjud. Ba'zan dasturlarni tezroq tuzish uchun algoritmni

tuzmasdan, to'g'ridan to'g'ri dastur yaratishga kirishuvchilar ham bor. Bu sodda masalalar uchun o'rinli bo'lishi mumkin, ammo murakkab masalalar uchun bu xato uslub hisoblanadi. Chunki, aynan algoritmni ishlab chiqish bosqichida eng optimal va samarali yechim ishlab chiqiladi. Algoritm turli xil ko'rinishda yozilishi mumkin. Algoritm so'z, formula, jadval yoki blok-sxema ko'rinishlarida beriladi.

4. *Algoritmni tahlil qilish va to'g'riligini tekshirish.* Algoritmni tahlil qilishdan maqsad uning to'g'ri tuzilganligini aniqlashdir. To'g'ri deb shunday algoritmgaga aytiladiki, u masalaning qo'yilishida talab qilinadigan natijani har qanday ruxsat etilgan boshlang'ich ma'lumotlar bilan ham tez bera oladi. Bundan tashqari, algoritmning samaradorligi ham tahlil qilinadi, ya'ni u kam qadamlar bilan natijani tez bera olsun.

5. *Dastur tuzish.* Dastur tuzish deganda, kompyuter uchun dasturni yozish tushuniladi. Buning uchun quyidagi savollarga javob berish kerak:

- 5.1. Asosiy o'zgaruvchilarni aniqlash.
- 5.2. O'zgaruvchilarning turlarini aniqlash.
- 5.3. Nechta massiv yoki fayllar va ular qanday kattalikda kerak bo'ladi?
- 5.4. Bog'langan ro'yxatlardan foydalanish ma'nolimi?
- 5.5. Qanday dasturiy qismlar kerak bo'lishi mumkin (tayyor bo'lsa ham)?
- 5.6. Qaysi dasturlash tilini tanlash kerak?

Bu bosqichni - 3 bosqich natijasini kompyuter tiliga "tarjima" qilish deyish mumkin. Bu yerda oldingi bosqichda amalga oshirilgan ishlar tahlil qilinadi va lozim topilganda takomillashtiriladi. Ba'zi bir hollarda dasturlash tili imkoniyatlaridan kelib chiqqan holda jiddiy o'zgartirishlar ham kiritish mumkin.

6. *Dasturni ishga tushirish va tekshirish.* Dastur ko'rinishida yozilgan algoritmni kompyuter yordamida bajarilishi. Dasturda xatolik bo'lmasa, u natija olish bilan tugalanadi. Biz dasturni har bir qismini tekshiradigan kirituvchi ma'lumotlar to'plamini tanlashimiz kerak. Ushbu ma'lumotlarni dasturga kiritib, olingan natijani, qo'lda olingan natija bilan solishtirib ko'rish lozim. Odatda, dastur bergan natijalar ma'lum bo'lgan yoki qo'lda hisoblangan ma'lumotlar bilan taqqoslanadi, va ular to'g'riligi aniqlansa, dastur to'g'ri ishlaydi degan xulosaga

kelish mumkin. Ammo bu usul bilan foydalanuvchini hamma shubhalardan xalos qilib bo'lmaydi, ya'ni dastur ishlamaydigan hamma holatlarni hisobga olish qiyin.

7. *Dasturni foydalanishga topshirish.* Dasturda to'g'ri ishlasa va kutilgan natijani berayotgan bo'lsa, uni kelgusida foydalanishga topshiriladi.

Algoritm tushunchasi va uning xossalari

Har bir inson kundalik hayotida eng sodda yoki murakkab ishlarni bajarishiga to'g'ri keladi. Ko'pchilik hollarda biror ishni qanday bajarish kerakligini tushuntiruvchi ma'lum qoidalar (ko'rsatmalar) berilgan bo'ladi. Inson ba'zi bir ishlarni bajarishda hayotiy tajribasiga suyanadi. Chunki u, o'ziga tanish bo'lgan ishlarni yoki masalalarni bajarish uchun zarur bo'ladigan ko'rsatmalarni qachonlardir, qayerlardandir olgan yoki o'rgangan bo'ladi. Agar undan, bu ishni qanday bajarish lozimligi so'ralsa, uni og'zaki yoki yozma ravishda ketma-ket bajariladigan harakatlar tarzida bayon qilib beradi. Ishni bajarish ko'rsatmasi qanchaliq aniq va tushunarli tavsiflansa, boshqa kishi uni shunchalik tez o'zlashtiradi va undan samarali foydalanadi. Biror maqsadga erishish uchun tavsiya qilingan ko'rsatma (buyruq, qoida, tartib, ketma-ketlik)lar algoritm atamasi bilan ifodalanadi. Demak, inson har kuni bajaradigan ishlarida o'zi bilmagan holda ma'lum bir algoritmlarga rioya qiladi. Masalan, ovqat tayyorlash, choy damlash, qahva tayyorlash, telefondan foydalanish.

Algoritmning aniq ta'rifi yo'q va unga turlicha ta'riflar berilgan. Masalan:

Algoritm - bu chekli qadamlar orqali ko'zlangan natijani olishga qaratilgan ketma-ket harakatdir.

Algoritm - qo'yilgan masalaning yechimini chekli qadamlardan so'ng olish uchun ijrochiga berilgan aniq va tushunarli ko'rsatmalardir.

Algoritm ijrochisi - algoritmda ko'rsatilgan buyruqlarni bajara oladigan shaxs yoki texnikadir.

Har bir ijrochi faqatgina o'zi tushunadigan buyruqlarni (ya'ni, buyruqlar tizimini) bajaradi.

Ijrochi buyruqlarni bajarish jarayonida oddiy harakatlarni bajaradi.

Odatda ijrochiga algoritmning maqsadi ma'lum bo'lmaydi. Shuning uchun ijrochi "nimaga" va "nima uchun" degan savollarni bermaydi.

Kundalik hayotimizda ko'pincha bir necha harakatlarni bajarib, kutilgan natijaga erishamiz. Bu harakatlarni tartib bilan yozsak ko'rsatma hosil bo'ladi.

Misol 1. Choy damlash uchun quyidagi amallarni bajarish zarur:

1. Choy qaynatadigan idishga kerakli miqdorda ichimlik suvi solinsin.
2. Suv qaynatilsin (elektr toki yoki gaz plitasi yordamida).
3. Choy damlanadigan choynakka kerakli miqdorda quruq choy solinsin.
4. Suv qaynagach u choynakka solinsin.

Shunga o'xshash ko'rsatmalar matematik masalalarni yechishda ham tuziladi.

Misol 2. a va b sonlarning o'rta arifmetik qiymatini topish uchun quyidagi ko'rsatmani keltirish mumkin:

1. a va b sonlar qo'shilsin.
2. Hosil bo'lgan yig'indi 2 ga bo'linsin.

Demak har xil hayotiy voqealar ham algoritm tarzida tavsiflanar ekan.

Algoritmni yaratish deganda, qo'yilgan masalani yechish jarayonini ketma-ket qadamlarga (bosqichlarga) bo'lib chiqish tushuniladi. Bunda, oldingi qadamda olingan natijalar keyingi qadamlarda ishlatiladi. Algoritmida har bir qadami (bosqich)ning mazmuni va ularning bajarilish tartibi aniq ko'rsatilishi kerak. Algoritmning alohida qadami (bosqichi), yechilish algoritmi oldin ishlab chiqilgan, boshqa, ancha oddiy masalani bildirishi yoki yetarli darajada oddiy va izohlarsiz tushunarli bo'lishi kerak. Algoritmnlarni tuzishda, insondan maxsus algoritmik fikrlash uslubi malakasiga ega bo'lishlik talab qilinadi. Algoritmik fikrlash uslubining asosini amallarning uzundan-uzoq ketma-ketligini rejalashtira olish, ularni algoritm ko'rinishida yozish, ularning oqibatlarini ko'ra bilish va algoritmni bajarishda kelib chiqishi mumkin bo'lgan barcha shart-sharoitlarni ko'zda tuta bilish tashkil etadi. Algoritmga arifmetik amallarni bajarish qoidalari, eng katta umumiy bo'luvchini topish, kvadrat tenglamaning ildizlarini topish, ko'phadning hosilasini topish qoidalari misol bo'ladi. Yuqorida keltirilgan misollardan ko'rinadi-ki, bir xil tipli masalalar to'plami uchun yagona algoritm tuzilsa, ushbu

algoritm bir xil tipli masalalar bo'yicha umumiy hisoblanadi va bunday masalalarni yechish uchun undan foydalanishimiz mumkin ekan. Masalan, $ax^2+bx+c=0$ ko'rinishdagi kvadrat tenglamalarni yechish masalasi ommaviy muammodir. Chunki biz a , b , c larni o'zgartirib bir xil tipli masalalar sinfini hosil qilamiz.

Algoritm so'zi buyuk o'zbek mutafakkiri Abu Abdulloh Muhammad ibn Muso al-Xorazmiyning (783-855) nomidan kelib chiqqan. 825 yilda u, Hindistonda qo'llanilgan o'nli sanoq tizimini o'z risolasida batafsil bayon qilib berdi. Al-Xorazmiy risolasida yangi sanoq tizimida amallarni bajarish qoidalarini keltirib o'tgan va birinchi marta 0 sonini qo'llagan. (U sonlarning hindcha nomini arab tilida as-sifr yoki oddiygina qilib sifr deb ishlatgan, shu tufayli "stifr" va "shifr" so'zlari vujudga kelgan) Shundan so'ng islom olami olimlari hind sonlarini qo'llay boshlashgan. XII asrning birinchi yarmida al-Xorazmiyning "Kitab al - jabr val – muqabala" ("Qo'shish va ayirishlar haqida kitob") kitobi lotin tiliga tarjima qilingan va u Evropaga tarqaldi. Bu kitob g'arbda juda mashhur bo'lgan va undan "algebra" so'zi vujudga kelgan. Bu asar sababli hisoblash ishlarini ixchamlash va osonlashtirish imkoni yaratildi, chunki bu bilan o'sha davrda qo'llash rasm bo'lgan rim raqamlari va sonlarni so'z orqali yozib bajarishdagi noqulayliklar bartaraf etildi.

Al-Xorazmiyning ilmiy risolalari yevropalik olimlarga juda chuqur ta'surot qoldirib algoritm (algoritmi) so'zining kelib chiqishiga sabab bo'ldi. Chunki, Al-Xorazmiyning lotin tiliga tarjima qilingan kitobi sarlavhasida Abu Abdulloh Muhammad ibn Musa al-Khwarizmi nomi keltirilgan. Bundan tashqari kitobning dastlabki beti "Dixit Algoritmi" ("Dediki Al-Xorazmiy" ning lotincha ifodasi) degan jumlar bilan boshlangan. Keyinchalik al-Xorazmiy so'zi Evropada "algoritm" deb talqin qilingan. Al-Xorazmiy ushbu kitobida matematikada ko'p uchraydigan bir necha amaldan iborat misollarni yechish tartibini birinchi bo'lib qo'llagan (avval qavslar ichidagi va darajaga ko'tarish, keyin ko'paytirish va bo'lish, va nihoyat qo'shish va ayirish amallari bajariladi). U birinchi bo'lib o'nli

sanoq sistemasining tamoyillari va unda turli amallar bajarish qoidalarini asoslab berdi hamda kvadrat tenglamalarni yechish usullarini keltirib o'tgan.

Algoritm xossalari. Biror masalani kompyuterda yechish uchun avval uning matematik modelini, keyin algoritmini va dasturini tuzish kerak bo'ladi. Bu uchlikda, algoritmni tuzish muhim ahamiyatga ega hisoblanadi. Har qanday algoritm quyidagi xususiyatlarga ega bo'lishi kerak:

1. Uzluksizlik (diskretlilik).
2. Aniqlilik (determinantlik).
3. Natijaviylik (yoki cheklanganlik).
4. Umumiylik (ommaviylik).
5. Tushunarlilik.

Har bir xossani alohida ko'rib o'tamiz.

1. *Uzluksizlik (diskretlilik).* Masalani yechish jarayoni algoritmi aniq va oddiy qadamlarning ketma-ket bajarilishi ko'rinishida ifodalanishi shart.

Algoritmning har bir qadami to'la bajarilgandan so'ng navbatdagi qadam bajarilishi lozim. Ketma-ket bajariladigan buyruqlar, amallar algoritmning qadamlari deyiladi. Demak, algoritmlarni doimo chekli qadamlarga bo'laklash orqali uni chekli sondagi oddiy ko'rsatmalar ketma – ketligi shaklida ifodalash mumkin ekan. Bu qadamlar ketma-ket tartibda bajarilishi lozim. Agar jarayonni ketma-ket bajariluvchi chekli qadamlarga bo'laklay olmasak, uni algoritm deb bo'lmaydi.

2. *Aniqlilik (determinantlik).* Algoritmning har bir qadami aniq yozilishi va faqat bitta ma'noga ega bo'lishi shart.

Bu xossa natijasida algoritmning bajarilishi mexanik tavsifga ega bo'ladi va yechilayotgan masala haqida hech qanday qo'shimcha ko'rsatmalar talab qilinmaydi.

Algoritmdagi qiymatlar to'plami o'zidan oldingi qiymatlar orqali to'liq aniqlanadi. Har bir qadam bir xil ma'noda tushunilib, keyingi qadam o'zidan oldingi qadamlar bilan to'laligicha aniqlanishi lozim. Bundan tashqari ijrochiga berilayotgan ko'rsatmalar mazmuni aniq bo'lishi zarur, chunki ko'rsatmadagi

noaniqliqlar mo'ljallangan maqsadga erishishga xalal beradi. Shuningdek, ko'rsatmalarning qaysi ketma – ketlikda bajarilishi ham muhim ahamiyatga ega. Demak, ko'rsatmalar aniq berilishi va faqat algoritmda ko'rsatilgan tartibda bajarilishi shart.

3. *Natijaviylik (yoki cheklanganlik)*. Algoritm chekli miqdordagi qadamlardan so'ng masalaning yechimiga olib kelishi kerak.

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

4. *Umumiylik (ommaviylik)*. Algoritmni, bir-biridan faqat boshlang'ich qiymatlar bilan farq qiluvchi ma'lum sinf masalalarining hammasiga qo'llash mumkin bo'lsin.

Bunda boshlang'ich qiymatlar algoritmning qo'llanilish sohasi deb atalgan sohasidan tanlanishi mumkin. Masalan, $ax^2 + bx + c = 0$ kvadrat tenglamani haqiqiy sonlar to'plamida yechish algoritmini misol sifatida oladigan bo'lsak, $D = b^2 - 4ac \geq 0$, bo'lganda ikkita yechim hosil qilamiz. Bu yechimlar algoritmning natijasiga aylanadi. Agar $D < 0$ bo'lsa, tenglamaning haqiqiy ildizlari yo'q va algoritmning natijasi sifatida “tenglama haqiqiy ildizlarga ega emas” degan jumla olinadi. Demak, har bir algoritm mazmuniga ko'ra bir turdagi masalalarning barchasi uchun ham o'rinli va masaladagi boshlang'ich ma'lumotlar qanday bo'lishidan qat'iy nazar algoritm shu xildagi har qanday masalani yechishga yaroqli bo'lishi lozim. Masalan, ikki oddiy kasrning umumiy maxrajini topish algoritmi kasrlarni turlicha o'zgartirib bersangiz ham ularning umumiy maxrajlarini aniqlab beradi. Yoki uchburchakning yuzini topish algoritmi uchburchakning qanday bo'lishidan qat'i nazar uning yuzini hisoblab beradi.

5. *Tushunarlilik*. Algoritmilar ma'lum bir ijrochiga mo'ljallab tuziladi. Chunki, har bir ijrochi o'ziga xos ko'rgazmaga ega va bu ko'rgazmalar faqat shu ijrochiga tushunarlidir.

Algoritmning ijrochisi hamma vaqt ham inson bo'lavermaydi. Chunki, kundalik hayotimizda ma'lum bir algoritmlar bilan ishlayotgan elektron soatlar, mashinalar, dastgohlar, kompyuterlar, turli avtomatik va mexanik qurilmalarni ko'ramiz. Bundan kelib chiqadiki, ijrochiga tavsiya etilayotgan ko'rsatmalar, uning uchun tushunarli mazmunda bo'lishi shart, aks holda ijrochi oddiygina amalni ham bajara olmaydi. Har bir ijrochining bajara olishi mumkin bo'lgan ko'rsatmalar yoki buyruqlar majmui mavjud bo'lib, u ijrochining ko'rsatmalar tizimi deyiladi. Demak, ijrochi uchun berilayotgan har bir ko'rsatma ijrochining ko'rsatmalar tizimiga mansub bo'lishi lozim. Ko'rsatmalarni ijrochining ko'rsatmalar tizimiga tegishli bo'ladigan qilib ifodalay bilishimiz muhim ahamiyatga ega. Masalan, quyi sinfning a'lochi o'quvchisi «sonni kubga oshirilsin» degan ko'rsatmani tushunmaganligi uchun uni bajara olmaydi. Ammo, bu ko'rsatmani «sonni o'zini o'ziga uch marta ko'paytirilsin» shaklida ifodalasak, u bu ko'rsatmani bemalol bajara oladi. Chunki, u ko'rsatma mazmunidan ko'paytirish amalini bajarishi kerakligini anglaydi.

Algoritmga qo'yiladigan talablar. Algoritm yuqorida bayon qilingan xossalardan tashqari bir qancha talablarga javob berishi lozim bo'ladi.

Soddalik. Har bir tuzilgan algoritm, talab qilingan xossalarni o'zida mujassamlashtirgan holda, tabiatiga ko'ra imkon darajasida, ishlash jarayoni soddaga bo'lgan qadamlardan iborat bo'lishi lozim.

Bundan tashqari, algoritm tuzishda quyidagilarga amal qilinsa, quyilgan masalaning natijasini tez va to'g'ri olish mumkin:

- qo'yilgan masalani to'g'ri o'qish va tushunib olish, masalaning asosiy maqsadini ajrata bilish.
- ishga daxldor qiyinchiliklarni aniq ko'ra olish va ortiqcha, masala yechimiga ta'siri bo'lmagan parametrlarni yo'qota bilish.
- qo'yilgan masalani bir-biriga bog'liq bo'lmagan mustaqil bo'laklarga ajrata olish va ular orasidagi bog'liqlikni to'g'ri tashkil etish.
- qo'yilgan masalaning yechimini olishda har bir bo'lak yechimlar to'plamini bir butun holga keltirish.

- masala yechimini sodda va tushunarli tilda bayon eta olish.

Algoritmni ishlab chiqish uchun qo'yilgan masala maqsadi va mohiyatini aniq tushunish, masala tegishli bo'lgan soha bo'yicha chuqur bilimga ega bo'lish hamda masalani yechish yo'lini yaxshi tasavvur qilish talab qilinadi. Bunda masalani avval qog'ozda, katta hisoblashlarni talab qilmaydigan qiymatlarning qandaydir bir to'plami bilan yechib ko'rish, keyin esa uni rasmiylashtirish, ya'ni aniq qoidalar ketma-ketligi ko'rinishida yozish kerak.

Algoritm turlari

Algoritmni ifodalash shakllari. Algoritmni ifodalaganda, uni ko'rgazmali, aniq va ixcham ko'rinishda keltirish va amaliy qo'llash uchun qulay holda ifodalash talab qilinadi. Amaliyotda ishlab chiqilgan algoritmni bir necha shaklda bayon etish qo'llaniladi. Biz ularning eng ko'p uchraydiganlari bilan tanishamiz:

1. So'z orqali, ya'ni tabiiy inson tilida (matn ko'rinishida) ifodalash.
2. Formulalar orqali ifodalash.
3. Jadvallar ko'rinishida ifodalash.
4. Grafik ko'rinishda, ya'ni blok-sxemalar ko'rinishida ifodalash.
5. Maxsus algoritmik tilda ifodalash.
6. Dastur shaklida ifodalanishi.

1. So'z orqali, ya'ni tabiiy inson tilida (matn ko'rinishida) ifodalash. Bu holda ijrochi uchun beriladigan har bir ko'rsatma jumlar orqali buyruq mazmunida beriladi.

1-misol. A, B, C, D va x sonlari berilgan. Agar $Cx - D \neq 0$ shart berilgan sonlar uchun o'rinli bo'lsa, quyidagi ifoda hisoblansin:

$$Y = (Ax + B) : (Cx - D).$$

Yechish: Ifodani hisoblash algoritmini so'z orqali keltiramiz:

- 1) A ni x ga ko'paytirib, natija R bilan belgilansin.
- 2) R ni B ga qo'shib, natija R bilan belgilansin.
- 3) C ni x ga ko'paytirib, natija R_1 bilan belgilansin.
- 4) R_1 dan D ni ayirib, natija R_1 bilan belgilansin.

5) R ni R_1 ga bo'lib, natija Y ning qiymati deb hisoblansin.

2. *Formulalar orqali ifodalash.* Algoritmni formulalar bilan berish shaklidan matematika, fizika, kimyo kabi aniq fanlarni o'rganishda ko'proq foydalaniladi. Bu usulni ba'zan analitik ifodalash deyiladi.

Masalan:

$$1) A * x = R$$

$$2) R + B = R$$

$$3) C * x = R_1$$

$$4) R_1 - D = R_1$$

$$5) R / R_1 = Y$$

3. *Jadvallar ko'rinishida ifodalash.* Boshlang'ich qiymatlar va natijalar jadval shaklida tasvirlanadi. Masalan, maktabda qo'llanib kelinayotgan to'rt xonali matematik jadvallar yoki turli lotereya, zayomlarning yutuq jadvallari bunga misol bo'la oladi. Funksiyalarning grafiklarini chizishda ham algoritmning qiymatlari jadvali ko'rinishlaridan foydalanamiz. Bu kabi jadvallardan foydalanish algoritmлари sodda bo'lganli sababli ularni o'zlashtirib olish oson.

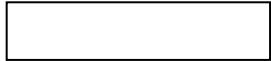
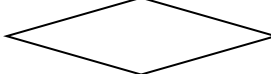
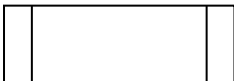
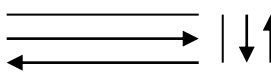
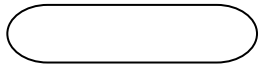
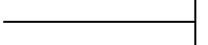
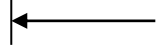
4. *Grafik ko'rinishda, ya'ni blok-sxemalar ko'rinishida ifodalash.* Algoritmning so'zlar orqali bayoni unchalik ko'rgazmali va qulay bo'lmaganligi sababli uni ifodalashda asosan grafik usulda yoki blok-sxema ko'rinishida ifodalash keng tarqalgan.

Matematika kursida chizilgan grafiklarning ko'pchiligi algoritmning grafik usulda berilishiga misol bo'ladi. Bundan tashqari, shahar yoki turar joy mavzellarida joylashgan biror uy hamda inshootlarini izlash va harakatlanish bo'yicha berilgan karta – sxemalar ham shunga misol bo'la oladi.

Algoritmni blok sxemalar ko'rinishida ifodalashda alohida bosqichlar har xil geometrik figuralar (blokлар) yordamida tasvirlanadi, ular o'rtasidagi bog'liqliklar esa bloklarni bog'lovchi yo'nalishlar bilan ko'rsatiladi. Bloklarni ulashda oqimlarning faqat vertikal va gorizontall chiziqlaridan foydalanish kerak. O'ngdan chapga tomon yo'nalishga ega bo'lgan gorizontall oqimlar va pastdan yuqoriga

yo'naltirilgan vertikal oqimlar, albatta, yo'nalishlar bilan belgilanishi kerak. Oqim chiziqlari tashqi ramka yoki sahifa chegaralariga parallel bo'lishi kerak.

Quyida blok-sxemaning asosiy belgilashlarini keltiramiz.

Nomi	Belgilanishi	Bajaradigan vazifasi
Jarayon		Bir yoki bir nechta amallarni bajarilishi natijasida ma'lumotlarning o'zgarishi
Qaror		Biror shartni kiritish va unga bog'liq ravishda algoritmning bajarilish yo'nalishini tanlash
Shakl o'zgartirish, takrorlash		Takrorlash sarlavhasi, takrorlash bloki
Avval aniqlangan jarayon		Oldindan ishlab chiqilgan dastur yoki algoritmdan foydalanish
Kiritish Chiqarish		Axborotlarni qayta ishlash mumkin bo'lgan shaklga o'tkazish yoki olingan natijani tasvirlash
Display		Kompyuterga ulangan displeydan axborotlarni kiritish yoki chiqarish
Hujjat		Axborotlarni qog'ozga chiqarish yoki qog'ozdan kiritish
Axborotlar oqimi chizig'i		Bloklar orasidagi bog'lanishlarni tasvirlash
Bog'lagich		Uzilib qolgan axborot oqimlarini ulash belgisi
Boshlash Tugatish		Algoritmning boshlanishi yoki tugallanishini belgilaydi
Izoh		Bloklarga tegishli turli xildagi tushuntirishlar
Birlashish		Oqim chiziqlarining birlashishi

Algoritm blok-sxemalari uchun Davlat Standarti qabul qilingan va unga rioya qilish talab qilinadi.

Har bir blok ichida bajariladigan amal yoziladi. Bloklar o'zaro chiziqlar orqali birlashtiriladi va ular hisoblash yo'nalishini belgilaydi. Ketma-ketlikni aniq ko'rsatish maqsadida har bir blokning yuqori chap chizig'ida raqam qo'yish mumkin.

Oqimlarning parallel chiziqlari orasidagi masofa 3 mm dan, sxemaning qolgan elementlari orasidagi masofa 5 mm dan kam bo'lmashligi kerak.

Bloklarni joylashtirish uchun sahifa maydonini gorizontal va vertikal (tarmoqlanuvchi sxemalar uchun) sohalarga bo'lish tavsiya etiladi.

Algoritmning har bir bloki qulaylik uchun raqamlanishi mumkin.

Barcha shakllar bir tekisda tasvirlanishi va bloklarning kattaligi bir o'lchamda chizilishi shart. Agar blokning balandligi v ga teng bo'lsa, uning uzunligi $a=1,5 \cdot v$ ga teng bo'lishi kerak.

5. Maxsus algoritmik tilda ifodalash. Bu algoritmlarni bir xil va aniq yozish hamda ularni bajarish uchun ishlatiladigan belgilashlar va qoidalar tizimidir.

Birinchidan algoritmik til oddiy tilga yaqin bo'lib, u oddiy matndek yozilishi va o'qilishi mumkin. Ikkinchidan, algoritmik til matematik simvollar: sonlar, kattaliklar va funkstiyalarning belgilari, amallar ishoralari, qavslar va boshqalarni o'z ichiga oladi. Algoritmik til o'z lug'atiga ega. Bu lug'atning asosini biror algoritm ijrochisining buyruqlari tizimiga kirgan buyruqlarni ifodalashda qo'llaniladigan so'zlar tashkil qiladi. Hozirgi kunda turli tuman algoritmik tillar mavjud. A. P. Yershov boshchiligida yaratilgan algoritmik tilda shaharlararo telefon-avtomatdan foydalanish ko'rsatmasini yozilishini ko'ramiz.

alg shaharlararo telefon-avtomatdan foydalanish

boshl

trubkani oling

telefon jetonini tashlang

uzluksiz signalni kuting

kerakli shahar kodini tering

abonent ovozi kuting

“gaplashish” tugmasini bosing

gaplashing

tam

6. *Dastur shaklida ifodalanishi.* Millionlab kompyuterlarning keng tarqalishi algoritmning dastur tarzidagi tasvirining keng ommalashib ketishiga katta turtki berdi. Chunki, kompyuterlar doimo dasturlar yordamida boshqariladi.

Algoritmning turlari. Har qanday murakkab algoritm amalda uch xil strukturaning kombinatsiyasidan iborat bo'ladi. Ular:

1. *Chiziqli algoritmlar.*
2. *Tarmoqlanuvchi algoritmlar.*
3. *Takrorlanuvchi algoritmlar.*

Algoritmni asosiy turlaridan tashqari shartli ravishda yana quyidagi turlarga ajratish mumkin:

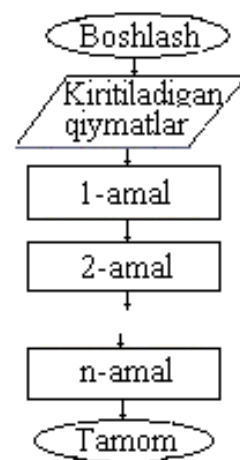
- *ichma-ich joylashgan takrorlanish algoritmlari.*
- *rekurrent algoritmlar.*
- *takrorlanishlar soni oldindan noma'lum algoritmlar.*
- *ketma-ket yaqinlashuvchi algoritmlar.*

Algoritmni shartli ravishda qo'shimcha turlarga ajratgan bo'lsakda, aslida ular yuqorida keltirilgan asosiy turlarning kombinatsiyasidan tashkil topadi.

Ularning eng soddasi chiziqli algoritmlar bo'lib hisoblanadi.

1. *Chiziqli algoritmlar.* Faqat ketma-ket bajariladigan amallardan tashkil topgan algoritmga-chiziqli algoritmlar deyiladi.

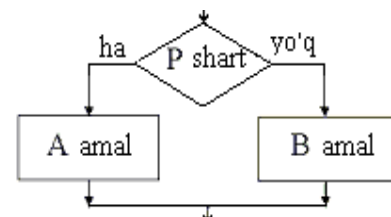
Chiziqli algoritm amallar ketma-ketligini tashkil qilib, hisoblash jarayonida bu amallar kelish tartibi bo'yicha faqat bir martagina bajariladi. Amalda chiziqli algoritmlar alohida juda kam uchraydi. Ular boshqa turdagi algoritmlar tarkibida qo'llaniladi. Bunday algoritmardan, masalan, matematika, fizika, kimyo va boshqa formulalarni yuqori aniqlikda hisoblashda foydalanish mumkin. Ularda hisoblash amallari ketma-ket bajariladi.



Blok sxemadagi bloklarni turli xil holatlarda joylashtirish mumkin. Buning uchun eng asosiysi, ular ijrochiga tushunarli va qaysi qadamdan keyin qaerga borishi kerakligini aniq ko'rsatish kifoya qiladi.

Blok sxemada hisoblash amallari to'g'ri to'rt burchaklar ichida beriladi.

2. *Tarmoqlanuvchi algoritmlar.* Agar hisoblash jarayoni biror bir berilgan shartning bajarilishiga qarab turli tarmoqlar bo'yicha davom ettirilsa va hisoblash jarayonida har bir tarmoq faqat bir marta bajarilsa, bunday hisoblash jarayonlariga tarmoqlanuvchi algoritmlar deyiladi.



Tarmoqlanuvchi algoritmlar o'z tarkibida bir yoki bir nechta shartni tekshiruvchi blokning bo'lishi bilan xarakterlanadi va ularda ayri strukturasi ishlatiladi. Bu algoritmda shartning bajarilishi yoki bajarilmasligining natijasiga qarab u yoki bu amalning bajarishi ta'minlanadi. Shuning uchun shartning bajarilishiga qarab hisoblash jarayoni albatta tarmoqlanadi.

Berilgan shart romb orqali ifodalanadi, P - berilgan shart. Agar shart bajarilsa, "ha" tarmoq bo'yicha A amal, shart bajarilmasa "yo'q" tarmoq bo'yicha B amal bajariladi.

Tarmoqlanuvchi algoritmlarning blok-sxemasi turli xil holatda joylashtirilishi mumkin, eng asosiysi u tushunarli bo'lishi talab qilinadi.

Tarmoqlanuvchi algoritmlarning ikki xil konstruktsiyasi mavjud:

1. To'liq shartli konstruktsiya:
2. To'liq bo'lmagan shartli konstruktsiya:

Ko'pgina masalalarni yechishda, shart asosida tarmoqlanuvchi algoritmlarning ikkita tarmog'idan bittasining, ya'ni yoki "ha" yoki "yo'q" ning bajarilishi yetarli bo'ladi. Bu holat tarmoqlanuvchi algoritmning xususiy holi sifatida aylanish strukturasi deb atash mumkin. Aylanish strukturasi quyidagi ko'rinishlarga ega:

Shart sifatida ijrochiga tushunarli, rioya qilinishi yoki qilinmasligi mumkin bo'lgan har qanday ta'kid ishlatiladi. Ta'kid so'zlar yoki formula orqali ifodalanishi mumkin.

Ba'zi bir masalalarni yechishda shartsiz o'tish ishlatiladi va u orqali boshqaruvni u yoki bu hisoblashga berish mumkin.

3. *Takrorlanuvchi algoritmlar.* Agar biror masalani yechish uchun tuzilgan zarur bo'lgan amallar ketma-ketligining ma'lum bir qismi biror parametrga bog'liq holda ko'p marta qayta bajarilsa, bunday algoritmi takrorlanuvchi yoki tsiklik algoritmlar deyiladi.

Amaliyotda murakkab jarayonlarni dasturlashda ma'lum buyruqlar ketma-ketligini ma'lum shartlar asosida qayta-qayta bajarish zaruriyati tug'iladi. Ma'lum bir o'zgaruvchining turli qiymatlarida ma'lum buyruqlar tizimining biron bir qonuniyatga asosan takrorlanib bajarilishi takrorlanuvchi hisoblash jarayonini (tsikl) hosil qiladi. Takrorlanuvchi hisoblash jarayonining takror-takror hisoblanadigan qismini takrorlanishning tanasi deb ataladi. Takrorlanish ichida qiymatlari o'zgarib boradigan o'zgaruvchini takrorlanish o'zgaruvchisi yoki takrorlanishning boshqaruvchi o'zgaruvchisi (tsikl parametri) deb yuritiladi.

Takrorlanuvchi jarayonning algoritmi umumiy holda quyidagilarni o'z ichiga olishi kerak:

1. Takrorlanishni tayyorlash - takrorlanishni boshlashdan oldin, takrorlanishda qatnashadigan o'zgaruvchilarning boshlang'ich qiymatlari yoki takrorlanish o'zgaruvchisining boshlang'ich qiymati o'rnatiladi, takrorlanish o'zgaruvchisining o'zgarish qadami belgilanadi.

2. Takrorlanish tanasi - takrorlanish o'zgaruvchilarining turli qiymatlari uchun takror bajariladigan amallar ketma-ketligi ko'rsatiladi.

3. Takrorlanish o'zgaruvchisiga yangi qiymat berish - har bir takrorlanishdan avval o'zgaruvchiga o'zgarish qadamiga mos ravishda yangi qiymat beriladi.

4. Takrorlanishni boshqarish-takrorlanishni davom ettirish sharti tekshiriladi, takrorlanishning boshiga o'tish ko'rsatiladi. Algoritmning takrorlash qismini umumiy holda quyidagi sxema orqali ko'rsatish mumkin:

Takrorlash algoritmi takrorlash tanasi tanasi va takrorlashni amalga oshiruvchi shartdan iborat.

Takrorlash algoritmi ham uch turga bo'linadi:

1 - tur “toki” takrorlash deb atalib, bunda berilgan shart tekshiriladi, shu shart qanoatlangunga qadar tsikl tanasi takrorlanaveradi. Shart bajarilmasa, tsikldan chiqib ketiladi.

2 - tur algoritmi “gacha” takrorlashdir. “Gacha” tsiklida dastlab tsikl tanasi bajarilib, so’ngra shart tekshiriladi. Shart qanoatlangunga qadar tsikl tanasi takrorlanaveradi. Shart bajarilishi bilanoq tsikldan chiqiladi.

3 - tur parametr bo’yicha takrorlashdir. Unda boshlang’ich qiymat, oxirgi qiymat va qadam mavjud bo’ladi. Agar qadam ko’rsatilmagan bo’lsa, u holda qadam 1 ga teng deb olinadi. Birinchi qadamda boshlang’ich qiymat bo’yicha takrorlanish tanasi bajariladi va qadam oshirilib yana takrorlanish tanasi bajariladi. Bu jarayon oxirgi qiymat bajarilgancha davom etadi. Oxirgi qiymat bajarilgach takrorlanishdan chiqiladi.

Bu yerda i - ixtiyoriy o’zgaruvchi, i_1 - boshlang’ich qiymat, i_2 - oxirgi qiymat va i_3 - qadam.

Chiziqli hisoblash jarayonlarini algoritmash

Chiziqli jarayonlar algortmlari amallar ketma-ketligini tashkil qilib, hisoblash jarayonida bu amallar kelish tartibi bo’yicha faqat bir martagina bajariladi. Amalda chiziqli algoritmlar juda kam uchraydi. Bunday algoritmlardan, asosan, katta formulalarni yuqori aniqlikda hisoblashda foydalanish mumkin. Chiziqli hisoblash jarayoni algoritmida – bloklar yuqoridan quyiga tomon, ya’ni boshidan oxiriga tomon bir chiziq ostida (shuning uchun chiziqli deyiladi) ketma-ket bajariladi.

Chiziqli algoritmning ko’rgazmali sxemasida quyidagi ishlar bajariladi:

1. Algoritmni ishga tushirish.
2. Foydalanuvchidan ma’lumotlar olish.
3. Ma’lumotlarni qayta ishlash.
4. Natijalarni displeyga yoki printeriga chiqarish.

Chiziqli algoritmgaga misol ko’raylik.

Misol 1: Kompyuter xotirasining A va B yacheykalardagi qiymatlarni almashtirish masalasi algoritmini tuzamiz:

Algoritmining sxemasi 5 blokdan iborat. “Boshlash” bloki algoritmni bajarishga sozlash, algoritmda ishlatiladigan o`zgaruvchilar uchun xotiradan joy ajratish ishini bajaradi.

2-blokda A va B o`zgaruvchilarning qiymatlari xotiraning shu nomli yacheykalariga uzatiladi.

3-blokda A yacheykadagi qiymat C o`zgaruvchi tomonidan o`zlashtiriladi, ya`ni C nomli yacheykaga uzatiladi. A yacheykadagi eski qiymat o`chib, uning o`rnida B o`zgaruvchining qiymati yoziladi. C yacheykadagi qiymat (A yacheykaning 1-qiymati) B yacheykaga uzatiladi. B o`zgaruvchining eski qiymati yangi C o`zgaruvchi qiymati bilan almashtiriladi.

4-blokda A va B yacheykalardagi yangi hosil bo`lgan qiymatlar talab qilingan tarzda chiqariladi.

5-blokda algoritm o`z ishini tugatadi.

Algoritmnlarni ta`riflash usullari

Olimlarning izchil izlanishlariga qaramay, algoritm tushunchasiga bitta aniq ta`rif berishning imkoni bo`lmadi va natijada bir qancha ta`riflar ishlab chiqildi. Ilmiy tadqiqotlarning natijasiga ko`ra formal ta`riflarning o`zaro teng kuchli va ekvivalent ekanligi aniqlandi. Biz quyida ularning ba`zi birlarini keltirib o`tamiz.

Ta`rif 1. Algoritmi – berilgan masalani yechish uchun boshlang`ich ma`lumotlar ustida bajariladigan cheklangan ketma-ketlikdagi oddiy amallarning qaysidir tilda berilishidir.

Ta`rif 2. (Knut). Algoritmi – bu chekli sondagi qoidalar to`plami bo`lib, u aniq masalalar to`plamini yechish amallari ketma-ketligini aniqlaydi va beshta muhim xususiyatlarga ega: cheklanganlik, aniqlik, kiritish, chiqarish va unumdorlik.

Ta`rif 3. (Kolmogorov). Algoritmi – bu qo`yilgan masala natijasiga qandaydir sondagi qadamlardan keyin olib keluvchi ma`lum qoidalar bo`yicha bajariluvchi har qanday hisoblash tizimidir.

Ta`rif 4. (Markov). Algoritmi – bu boshlang`ich ma`lumotlardan izlangan natijaga olib keluvchi hisoblash jarayonini aniqlovchi aniq ko`rsatmalardir.

Ta'rif 5. Algoritm – bu qo'yilgan masalani yechimini chekli qadamlardan so'ng olish uchun ijrochiga berilgan aniq va tushunarli ko'rsatmalardir.

Umuman olganda algoritm tushunchasi ta'riflari bir qator talablarga javob berishi kerak.

- Algoritm chekli sondagi elementlar bajaruvchi ko'rsatmalardan iborat bo'lishi kerak.
- Algoritm chekli sondagi qadamlardan iborat bo'lishi kerak.
- Algoritm barcha boshlang'ich ma'lumotlar uchun umumiy bo'lishi kerak.

Masalalarni algoritmash va dasturlash asoslari

Amaliyotda algoritm ijrochisi kompyuter hisoblanadi. Hozirgi zamon kompyuterlari inson qo'llaydigan rus, o'zbek yoki boshqa shu kabi biror-bir tilda tuzilgan algoritmni tushuna oladigan darajada takomillashgan emas. Shu bois, mashinaga mo'ljallangan buyruqlar kompyuter tushunadigan shaklda yozilishi zarur. Bu maqsadda algoritmik yoki dasturlash tillari deb ataluvchi sun'iy tillar qo'llaniladi. Algoritm ushbu sun'iy tilda yozilishi kerak. Ijrochi (kompyuter) berilgan buyruqlarni aniq va to'g'ri bajarishi uchun unga beriladigan buyruqlar aniq yozilishi va ijrochi tomonidan bir xil tushunilishi lozim. Shu sababli kompyuter tushunadigan til dasturlash tili, bu tilda yozilgan algoritm esa kompyuter dasturi deyiladi.

Dastur – kompyuterga algoritmni beruvchi ko'rsatmalar (buyruq yoki tasnif va operatorlar)ning ketma-ketligi.

Dastur kompyuter tomonidan qaysi tartibda, qaysi ma'lumotlar ustidan va qaysi amallar bajarilishi va natija qaysi shaklda taqdim etilishini ko'rsatadi. Kompyuterning boshqarish qurilmasi mashina buyruqlari ketma-ketligi shaklida berilgan dasturni qabul qiladi. Dasturni mashina tilida yaratish – noqulay va katta mehnat talab qiluvchi jarayon. Shuning uchun dastur inson tomonidan dasturlash tillarining birida yaratilib, keyin u mashina tiliga o'giriladi.

Dasturlash tillari paydo bo'lishidan oldin dasturlar mashina kodlarida (mashina kodi - protsessor tomonidan bajariladigan xotiradagi instruktsiyalar kema-ketligidir) tuzilar edi. Katta - katta dasturlarni tuzishda juda ko'p vaqt talab qilinardi, ularni xatolarini tuzatish juda qiyin, modifikatsiyalash esa ko'p hollarda ilojisiz edi. Shularni hisobga olib, inson uchun tushunarli bo'lgan dasturlash tillarini tashkil qilish muammosi paydo bo'ldi va dasturlash tillari vujudga keldi.

Dasturlash tili – kompyuter uchun dasturlar (ko'rsatmalar yig'masi) yozilgan uni u yoki bu harakatlarni bajarishga majbur qiladigan rasmiy til.

Dasturlash tilining ta'rifi quyidagilarni o'z ichiga oladi:

1. Mumkin bo'lgan belgilar ro'yxati.
2. Zahiralangan so'zlar ro'yxati.
3. Sintaksis (belgilarni va zahiralangan so'zlarni birikmalash usullari).
4. Semantika (dasturlash tilining birikmalari ma'nosi)

Dasturlash tilida yozilgan ko'rsatmalar dastlabki kod deb ataladi.

Dastlabki kod kompyuterda bajarilishidan avval, uni mashina kodiga o'girish lozim.

Mashina kodi tili kompyuter qurilmalari bilan bevosita bog'liq bo'lib, buyruqlar ularning kodlari yordamida yoziladi. Dastlabki kompyuterlar (Eniak, MESM va boshqalar) ana shu tilda ishlar edi.

Dasturlash tillari yaratilishi bo'yicha ikki guruhga bo'linadi:

- Quyi darajadagi tillar yoki assemblerlar.
- Yuqori darajadagi tillar.

Quyi darajadagi dasturlash tillarining buyruqlari faqat raqamlar emas, balki inson nutqiga yaqin so'zlar ham ishlatila boshlandi. Ular ko'pincha assembler tillari deb ham yuritiladi. Mazkur tilda tuzilgan dastur yuqori darajadagi tillarda tuzilgan dasturdan 10-15 marta tezroq ishlaydi. Unda boshqa darajadagi tillarda bo'lmagan amallar mavjud.

Assembler tillarida buyruqlar qisqartirilgan so'zlar yoki so'zlar majmuidan iborat bo'lib, gohida ularni mnemokodlar deb ham yuritiladi.

Yuqori darajadagi dasturlash tillari ancha rivojlangan tuzilishga ega bo'lib, unda ishlatiladigan xizmatchi so'zlar inson nutqiga juda yaqinlashtirilgan. Mazkur tilda foydalanuvchi amallarni ketma-ket yozadi, kompyuter esa uni kerakli ko'rinishga o'tkazib oladi.

Mavjud dasturlash tillari bir necha guruhga bo'linadi:

- Strukturali (tuzilmali)
- Protsedurali
- Mantiqiy
- Ob'ektga yo'naltirilgan

Hozirgi kunda dasturlash tillarini besh avlodga ajratishadi:

- Birinchi avlod: mashina tili.
- Ikkinchi avlod: Assembler tili.
- Uchinchi avlod, yuqori bosqichli tillar: Masalan , C, C++, C#, Java, ...
- To'rtinchi avlod, ma'lumotlar bazalarini boshqarish tizimlari bilan ishlaydigan va ma'lumotlar oqimini boshqaradigan yuqori bosqichli tillar (4GL): Masalan, SQL, Focus, Metafont, PostScript, RPG-II, IDL-PV/WAVE, Gauss, Mathematica, AVS, Iris Explorer, ...
- Beshinchi avlod, sun'iy intellekt va neyron tarmoqlarida masala yechishda foydalanadigan tillar.

Dasturlash tillari inson uchun tushunarli bo'lgan dasturlarni tuzish imkonini beradi. Bunday dasturlarning matnini mashinalarlarda bajarish uchun, ularni mashina kodiga aylantirish zarur. Buning uchun translyator deb ataluvchi maxsus dasturlardan foydalaniladi.

Translyator (ingl. Translator — tarjimon) dasturlash tilida yozilgan matnni mashina kodiga aylantiradi (ya'ni, tarjima qiladi).

Translyator oldindan kompyuter xotirasiga yozilgan bo'ladi. Translyatorlar kompilyator va interpretator ko'rinishida bo'lishi mumkin. Kompilyator va interpretatorlar dasturni bajarishda tubdan farq qiladi.

Kompilyator (ingl. compiler — yig'uvchi) dasturni to'laligicha o'qiydi va uni tarjima qiladi (mashina tiliga o'giradi).

Interpretator (ingl. interpreter — izohlovchi, og'zaki tarjimon) dasturni tarjima qiladi va satrma-satr bajaradi.

Dastur kompilyatsiya qilingandan so'ng dasturlash tilida yozilgan dastur matni va kompilyatorga ehtiyoj qolmaydi. Masalan, info.pas fayli kompilyatsiya qilingandan so'ng info.exe fayliga o'zgaradi va bu fayl kompilyatorsiz ham ishlashi mumkin. Agar info.pas faylini interpretator qayta ishlaganda har doim dastur ishini bajarilishi uchun interpretator ishga tushishi shart. Shu sababli kompilyator va interpretatorlarning ishlash printsipturlichadir.

Kompilyatsiya qilingan dasturlar tezroq ishlaydi, lekin interpretatsiya qilinayotgan dasturlarga o'zgartirish kiritish osonroq.

Har bir dasturlash tili qaysi maqsadda yaratilganligiga qarab kompilyatsiya yoki interpretatsiyaga mo'ljallangan bo'lishi mumkin.

Bayt-kod (bytecode) Hisoblash texnikasining va web texnologiyalarning rivojlanishi, aniq bir mashinaga va operatsion tizimga bog'liq bo'lmagan yuqori darajadagi dasturlash tillarining paydo bo'lishi hamda ularning rivojlanishiga olib keldi. Bu dasturlash tillari dastur matnini bayt-kod deb ataluvchi oraliq shaklga translyatsiya qiladi. Bu bayt-kodli dasturni ixtiyoriy joyda va ixtiyoriy vaqtda ishga tushirish mumkin. Masalan, Java va C# singari dasturlash tillari dastur matnini interpretator orqali bayt-kodga o'giradi. Bayt-kod ixtiyoriy platformada tezda mashina kodiga o'girilib ishga tushiriladi.

Dasturlash tili inson va mashinaning muloqot vositasi hisoblanadi va u quyidagi afzalliklarga ega:

- u jonli tilimizga o'xshash va uni o'rganish oson.
- bu tilda yozilgan dastur mashina tilidagidan qisqaroq bo'ladi.
- dastur yozishga kamroq vaqt sarflanadi va kam xatolikka yo'l qo'yiladi.
- yozilgan dasturni ixtiyoriy dasturchi o'qiy oladi.
- dasturlash tili mashina turiga bog'liq emas.

Demak, dasturlash (yuqori darajadagi) tilida dastur tuzish qulayroq, osonroq va buning uchun aynan bir mashina (quyi darajadagi) tilini bilish shart emas ekan.

Bugungi kunda yuzlab dasturlash tillaridan foydalaniladi. Har bir dasturlash tili ma'lum bir soha yoki yo'nalishga oid masalalarni hal qilish uchun mo'ljallangan.

Nazorat savollari:

1. Masalalarni kompyuterda yechish bosqichlari deganda nimani tushunasiz?
2. Algoritm deyilganda nima tushuniladi?
3. Algoritmning qanday xossalari bilasiz?
4. Algoritmni ifodalashni qanday shakllarini bilasiz?
5. Algoritmni blok sxema shaklida ifodalashni tavsiflang.
6. Algoritmning turlarini tushuntiring.
7. Chiziqli algoritmlar deyilganda nima tushuniladi?
8. Tarmoqlanuvchi algoritmlar deyilganda nima tushuniladi?
9. Tarmoqlanuvchi algoritmning to'liq shartli konstruktsiyani izohlang.
10. Tarmoqlanuvchi algoritmning to'liq bo'lmagan shartli konstruktsiyasi.
11. Takrorlanuvchi algoritmlar deyilganda nima tushuniladi?
12. Takrorlanuvchi algoritmning "toki" takrorlanish turini izohlang.
13. Takrorlanuvchi algoritmning "gacha" takrorlanish turini izohlang.
14. Algoritm qo'yiladigan talablar haqida tushuncha bering.
15. Algoritm tushunchasi ta'riflari qanday talablarga javob berishi kerak.
16. Dastur deganda nimani tushunasiz?
17. Dasturlash tillari deganda nimani tushunasiz?
18. Mashina kodi tili deganda nimani tushunasiz?
19. Dasturlash tillari yaratilishi bo'yicha necha guruhga bo'linadi va ular haqida tushuncha bering.
20. Translyator deganda nimani tushunasiz?
21. Kompilyator deganda nimani tushunasiz?
22. Interpretator deganda nimani tushunasiz?
23. Bayt-kod deganda nimani tushunasiz?

C++ DASTURLASH TILI VA UNING ASOSIY OPERATORLARI.

C++ dasturlash tili.

C++ tili 1980 yilda B'yarn Straustrup tomonidan yaratilgan va u C tilining ancha kengaytirilgan, takomillashtirilgan hamda ob'ektga yo'naltirilgan shaklidir. Straustrup C++ tilini yaratishdan avval o'z oldiga ikki maqsadni qo'ydi. Birinchidan C++ tilida C tili standartining barcha imkoniyatlarini saqlab qolgan holda qo'shimchalar qo'shish, ikkinchidan tilga ob'ektga yo'naltirilgan dasturlashning tuzilmalarini kiritish. C++ tilining ham yuqori tezlikka ega ekanligi, ko'chiruvchanligi va tuzilmaliligi sababli ko'pchilik operatsion tizimlar, kompilyatorlar va kommunikatsiya tizimi dasturlari ushbu tilda tuzilgan hamda tuzilmoqda.

C# tili 1998 yilda MicroSoft kompaniyasi tomonidan yaratilgan va u C++ tilining ancha kengaytirilgan, takomillashtirilgan hamda platformalarga bog'liq bo'lmagan shaklidir.

C++ tilining alfaviti va xizmatchi so'zlari.

C++ dasturlash tilining alifbosi deb, shu tilda ma'lumotlarni ifodalash va dastur ishlab chiqish jarayonida kompilyator tomonidan qabul qilishga ruxsat berilgan belgilar yoki maxsus belgilardan iborat zanjirlar to'plamiga aytiladi. Bu alifbo ASCII (xalqaro belgilar va ularning kodlari) jadvalining hamma belgilarini, ya'ni quyidagilarni o'z ichiga oladi:

1) lotin alifbosining katta va kichik harflari (A, B, ... Z, a, b, ... z).

2) 0 dan 9 gacha arab raqamlari.

3) tagiga chizish belgisi (_).

4) bo'sh joy belgisi.

5) boshqaruvchi belgilar: ASCII jadvalidagi (1-ildovaga qarang) kodlari 0 dan 31 gacha bo'lgan belgilar. Bu belgilar satr va konstantalarni ifodalashda qo'llanishi mumkin.

6) turli ko'rsatmalarni yozish uchun ishlatiladigan maxsus belgilar:

+ - * / = { } () [] . , @ # ' . . \$ % A ! ? & \ < >

7) asosiy bo'lmagan belgilar (ASCII ni kengaytiruvchi, ya'ni kodi 128 dan 255 gacha bo'lgan belgilar, rus alifbosining katta va kichik harflari, psevdografika

elementlari shu sinfga kiradi. Bu belgilar turli konstantalarni hosil qilish, matnlarni yozish, izohlarni tashkil qilishda qo'llanishi mumkin).

8) murakkab belgilar: `< = > = : = . . * *`.

9) xizmatchi so'zlar. Ular C++ dasturlash tilida ma'lum bir ma'no yoki ko'rsatmani anglatuvchi maxsus belgilar zanjiridan iborat bo'lib, bu zanjirni o'zgartirish yoki qisqartirib qo'llash mumkin emas. Shuningdek, ularni o'zgaruvchilar nomlari sifatida ishlatish mumkin emas.

C++ tilida quyidagi xizmatchi so'zlar mavjud:

- Tiplar nomlari: `char, short, int, unsigned, long, float, double`.
- Operatorlar nomlari: `if, else, switch, case, while, do, for, default, break, continue, goto`.
- Xotira turlari: `auto, register, static, extern`.
- Tiplar bilan ishlash: `typedef, sizeof`.
- Struktura: `struct, union`.
- Chiqish: `return, entry`.

C++ tilining xizmatchi so'zlari: `asm, auto, break, case, catch, char, class, const, continue, default, delete, do, double, else, enum, explicit, extern, float, for, friend, goto, if, inline, int, long, mutable, new, operator, private, protected, public, register, return, short, signed, sizeof, static, struct, swith, template, this, throw, try, typedef, typename, union, unsigned, virtual, void, volatile, while`.

10) ko'rinmaydigan belgilar ("umumlashgan bo'shliq belgilari"). Leksemalarni o'zaro ajratish uchun ishlatiladigan belgilar (misol uchun bo'shliq, tabulyatsiya, yangi qatorga o'tish belgilari).

Eslatma. Agar dastur tarkibida yuqorida sanab o'tilgan belgilardan boshqa belgi yoki xizmatchi bo'lmagan so'zlar uchrab qolsa, bu haqidagi maxsus axborot kompyuter ekraniga chiqariladi.

So'z - bo'sh belgiga ega bo'lmagan belgilar ketma-ketligidir. Xizmatchi so'z esa C++ tilida ma'lum bir buyruqni amalga oshirish ko'rsatmasini beruvchi so'zdir.

C++ tilida dasturni tavsiflash uchun izohlar ishlatiladi. Izohlarda, satrlarda va belgili konstantalarda boshqa literallar, masalan rus harflari ishlatilishi mumkin.

C++ tilida olti xil turdagi leksemalar ishlatiladi: erkin tanlanadigan va ishlatiladigan identifikatorlar, xizmatchi soʻzlar, konstantalar (konstanta satrlar), amallar (amallar belgilari) va ajratuvchi belgilar.

Identifikator. Identifikatorlar lotin harflari, ostki chiziq belgisi va sonlar ketma-ketligidan iborat boʻladi. Identifikator lotin harfidan yoki ostki chiziq belgisidan boshlanishi lozim.

Misol uchun:

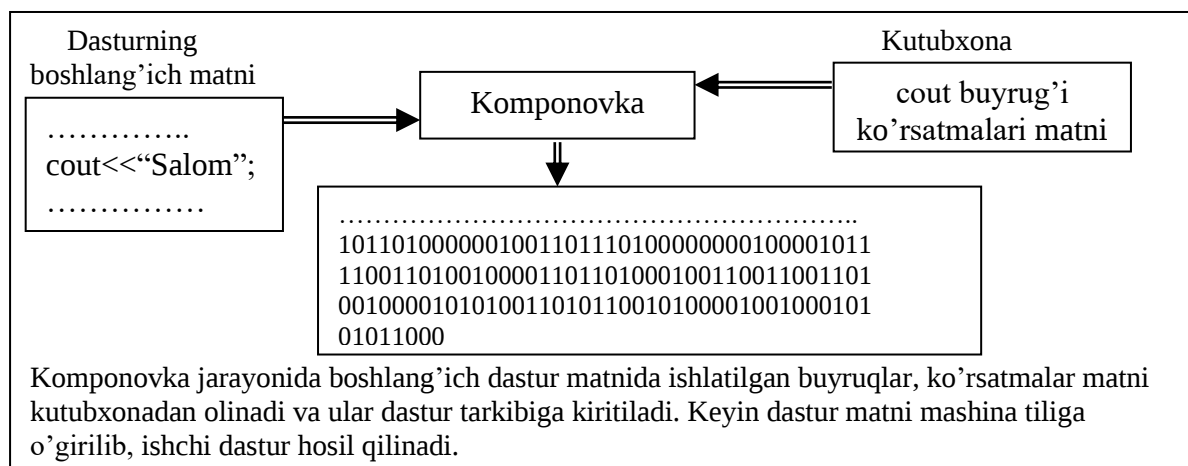
A1, _MAX, adress_01, RIM, rim

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

Borland kompilyatorlaridan foydalanilganda nomning birinchi 32 harfi, baʼzi kompilyatorlarda 8 ta harfni inobatga oladi. Bu holda NUMBER_OF_TEST va NUMBER_OF_ROOM identifikatorlari bir-biridan farq qilmaydi.

Funksiyalar kutubxonasi.

C++ tilining asosiy qudrati uning funksiyalaridadir va ular kompilyator bilan birga funksiyalar kutubxonasi tarzida taqdim qilinadi.

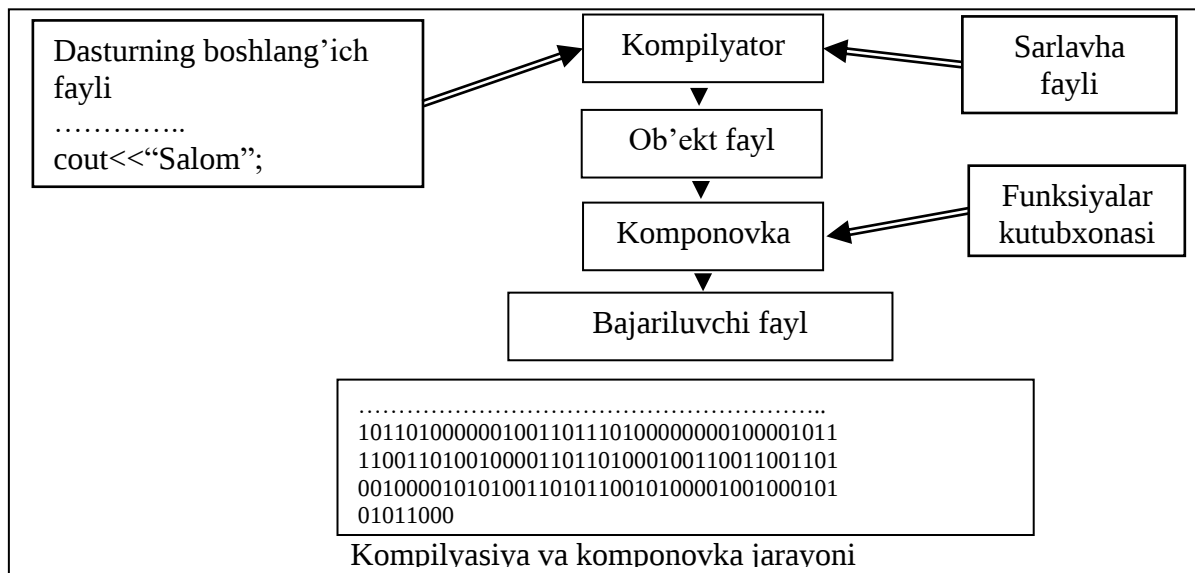


Funksiya maxsus masalalarni hal qilish uchun ishlatiladigan ketma-ket tartibda berilgan koʻrsatmalar toʻplamidir. Kutubxona-kompilyatorga biriktirilgan alohida fayl boʻlib, u turli xil muammolarni hal qiluvchi funksiyalar toʻplamidan tashkil topgan.

Masalan, ma'lumotni ekranga chiqarish uchun bir necha funksiya mavjud bo'lib, ularning muvofiqlig'idan o'z o'rnida foydalanish orqali kerakli ish bajariladi. Har bir funksiya o'z nomiga ega. Biror bir ishni bajarish uchun uni amalga oshiruvchi funksiyaning nomini sintaksisiga muvofiq yozish kifoya qiladi.

Ba'zi bir funksiyalar avvaldan kompilyatsiya qilingan kodga ega. Agar kompilyator dastur matnida bunday funksiyaga duch kelsa, uni ikkilik kodga o'girish bilan shug'ullanmaydi. Chunki, funksiya matni ikkilik kodga o'girilgan bo'ladi. Komponovka jarayonida funksiyaning ikkilik kodi kutubxonadan olinib dastur matniga biriktiriladi.

Funksiya avvaldan kompilyatsiya qilinganligi uchun juda ham unumli kodlashtirilgan hamda u to'liq optimallashtirilgan bo'ladi.



Kompilyator bilan birgalikda juda ko'p ishlatiladigan funksiyalarning boshlang'ich matni ham taqdim qilinadi. Ular *sarlavha fayllari* deb ataluvchi fayllarda saqlanadi va odatda *.h kengaytmaga ega bo'lishadi. Sarlavha fayllari tarkibida kompilyator direktivalari va biror bir aniqlanishni ishlatish qo'llanmalari ham mavjud. Kompilyatsiya jarayonida ishlatilgan sarlavha fayli matni ham sizning xususiy matningiz tarkibiga kiritiladi va birgalikda umumiy ob'ekt fayl hosil qilinadi.

Sarlavha fayllari kutubxona fayllaridan farqli ravishda kompilyatsiya qilinmagan bo'ladi. Ularni ham xuddi boshlang'ich dasturingiz matni singari o'qish, o'zgartirish va chop qilish mumkin. Ammo, kompilyator bilan birga taqdim

qilingan sarlavha fayllarini o'zgartirishdan saqlanishingizni maslahat beramiz, chunki ular barcha imkoniyatlarni hisobga olib tuzilgan va biror xatoga yo'l qo'ysangiz kompilyator hech qachon ular ob'ekt kodini generatsiya qila olmaydi.

Kompilyator bilan birga taqdim qilinayotgan kutubxonaga qo'shimcha, maxsus ma'lumotlar bazasini qurish, grafik, kommunikatsiya singari har xil kutubxonalarni ham olishingiz mumkin. Kutubxona qanchalik boy bo'lsa ishingiz unumli va tez bo'ladi. Bundan tashqari dasturlaringizda juda ko'p ishlatilayotgan ish jarayonlarini umumlashtirib o'z xususiy kutubxonangizni ham tashkil qilishingiz mumkin. Masalan, ko'pchilik dasturlaringizda A va B jarayonni bajarishni, keyin esa C jarayonni bajarish kerakligini ko'p marta takrorlaysiz. Ular ish jarayonini tavsiflovchi funksiyalarni yaratib, ularni xususiy kutubxonangizga joylashtirasiz. Ular oldindan yozilgan va kompilyatsiya qilinganligi uchun har doim ehtiyoj tug'ilganda dasturlaringizda ulardan tez foydalana olasiz.

Til imlo qoidalariga rioya qilgan va kompilyator kutubxonasidagi funksiyalardan foydalangan holda yozilgan dasturlar oson ko'chiriluvchan bo'ladi. Chunki, har bir platforma kompilyatori standart talablarni qanoatlantiradi va yaratilgan ishchi dastur ushbu platformada xatosiz ishlaydi.

Dasturning tuzilishi.

C++ tilida barcha dasturlar `main()` nomli funksiya bilan boshlanishi lozim. `main()` funksiya bo'lgani uchun qiymat qaytarishi lozim. Shu sababli ko'pchilik C++ dasturlash muhitlarida `int main()` tipli sifatida (ya'ni butun sonli qiymat qaytaruvchi) boshlanadi. Bu holda dastur oxirida `return 0` (yoki ixtiyoriy son) buyrug'i bo'lishi lozim. Ba'zi hollarda `void main()` tipsiz sifatida (ya'ni qiymat qaytarmaydigan) boshlanadi. Bu holda dastur oxirida `return 0` (yoki ixtiyoriy son) buyrug'i bo'lmaydi.

Qavslar funksiya nomi qismi bo'lib, aynan u kompilyatorga ingliz tilidagi oddiy `main` emas, balki funksiya ishlatilayotganligini bildiradi. Qavslar har bir funksiyaning ajralmas qismi bo'lib, ko'pchilik hollarda qavs ichida funksiya parametrlari beriladi.

`main()` dan keyin ko`rsatmalar beriladi. Ko`rsatmalar standart buyruq va kutubxona tarkibidagi standart funksiya yoki o`zimiz yozgan xususiy funksiya shaklida beriladi. Birinchi ko`rsatmadan oldin ochuvchi figurali qavs (`{`), oxirgi ko`rsatmadan keyin esa yopuvchi figurali qavs (`}`) qo`yiladi. Ochuvchi va yopuvchi figurali qavslar orasidagi funktsiyani tashkil qiluvchi ko`rsatmalar ketma-ketligi, ko`pincha funksiya tanasi deyiladi. Shunday qilib, C++ tilida yozilgan eng oddiy dasturning tuzilishi quyidagicha:

```
int main()    //dasturning boshlanishini bildiruvchi funksiya. Kirish nuqtasi.
{
    //funksiya shu yerdan boshlanadi.
    .....
    .....    //kompyuter bajarishi kerak bo`lgan ko`rsatmalar
    .....    //shu yerda joylashtiriladi.
    return 0;    //funksiya qiymat qaytaradi.
}    //funksiya shu yerda tugaydi.
```

Ochuvchi va yopuvchi figurali qavslar cheklovchi deb ataladi hamda ular kodning ma`lum bir qismini yagona blok sifatida ajratish uchun xizmat qiladi. Funksiya ochuvchi figurali qavs bilan boshlanib, yopuvchi figurali qavs bilan yakunlanishi lozim. Bundan tashqari, funksiya ichida alohida bloklar xususiy figurali qavslar bilan ajratilishi va ular bir nechta bo`lishi mumkin. Har bir ochuvchi figurali qavsning o`z yopuvchi figurali qavsi mavjud.

```
int main()
{
    .....
    {
        .....
        {
            .....
        }
    }
    .....
}
```

• • • • •

```
return 0;
```

}

Qavslarni ishlatish tartibi matematikada qavslar ishlatilish tartibiga asoslangandir. Qavslarni ishlatishga doir cheklov mavjud emas.

$$\{\dots\{\dots\{\dots\{\dots\{\dots\}\dots\}\dots\}\dots\}\dots\}$$

Dastur ishga tushirilganda kompyuter ishni `main()` funksiyaning birinchi ko`rsatmasini bajarishdan boshlaydi. Masalan, C++ tilida *Salom* so`zini ekranga chiqarish dasturi quyidagi ko`rinishda yoziladi.

1–misol. *Salom.cpp* dasturi misolida C++ tilida tuzilgan dastur qismlarini namoyish qilish

// Salom.cpp dasturi

```
#include <iostream>
```

using namespace std; //Dev C++ muhitida qo'yilishi lozim

```
int main()
```

```
{ cout << "Salom!\n";
```

```
return 0;}
```

NATIJA:

Salom!

TAHLIL

1 – satrda `iostream.h` fayli joriy faylga biriktilyapti. Dasturda birinchi funta (#) belgisi joylashgan. U preprotssessorga signal uzatadi. Kompilyatorning har safar ishga tushirilishida preprotssessor ham ishga tushiriladi. U dasturdagi funta (#) belgisi bilan boshlanuvchi qatorlarni o`qiydi.

include - **preprotsessorning** buyrug'i bo'lib, u quyidagicha tarjima qilinadi:

“Bu buyruqni ortidan fayl nomi keladi. Ushbu nomdagi faylni topish va fayldagi mazmunni dasturning joriy qismiga yozish lozim”.

Burchakli qavs ichidagi faylni mos fayllar joylashtirilgan barcha papkalardan izlash lozimligini ko'rsatadi. Agarda kompilyator to'g'ri sozlangan bo'lsa, burchakli qavslar `iostream.h` faylini sizning kompilyatoringiz uchun mo'ljallangan `.h` kengaytmali fayllarni o'zida saqlovchi papkadan izlashi kerakligini ko'rsatadi. `iostream.h` (input - output stream - kiritish-chiqarish oqimi) faylida ekranga ma'lumotlarni chiqarish jarayonini ta'minlaydigan `cout` ob'ekti aniqlangan. Birinchi qator bajarilgandan so'ng `iostream.h` fayli joriy dasturga xuddi uning mazmunini qo'l bilan yozganimizdek biriktiriladi. Preprotssessor kompilyatordan keyin yuklanadi va funt (#) belgisi bilan boshlanuvchi barcha qatorlarni bajaradi, dastur kodlarini kompilyatsiyaga tayyorlaydi.

```
#include <iostream>
```

`#include` direktivasi kompilyatorga `iostream.h` sarlavha faylidagi ma'lumotdan foydalanish zarurligini ko'rsatadi. Qisqartirilgan `stdio` standart kiritish/chiqarish (standart input/output) so'zini anglatadi. `iostream.h` fayl, kompilyator disk fayllari va printer bilan ishlashi uchun kerakli bo'lgan ko'rsatmalarni saqlaydi.

Sarlavha faylida saqlanadigan ko'rsatmalarni `main()` funksiyadan oldin joylashtirish lozim.

Sarlavha fayli (< >) belgilari orasida berilsa, bu fayl `INCLUDE` katalogida mavjud bo'lishi mumkinligini kompilyatorga bildiradi. Instalyatsiya jarayonida sarlavha fayllari shu katalogga joylashtiriladi. Agar kompilyatsiya jarayonida sarlavha fayllari joriy katalogdan topilmasa, kompilyator uni `INCLUDE` katalogidan izlaydi.

Siz sarlavha faylini qo'shtirnoq ichiga olib ham yozishingiz mumkin.

```
#include "iostream"
```

Bu holda kompilyator sarlavha faylini faqat joriy katalogdan izlaydi, agar uni topa olmasa, xatolik mavjudligi xabarini beradi. Shuning uchun kompilyator hujjati bilan tanishib, qaysi holat o'rinli ekanligini aniqlang.

using namespace std;

ko'rsatmasi Dev C++ muhitida qo'yilishi lozim, Borland C++ muhitida bu ko'rsatma berilmaydi.

Dasturning asosiy kodi `main()` funksiyasini chaqirish bilan boshlanadi. C++ dasturida albatta `main()` funksiyasi bo'lishi shart va dastur shu funksiyani bajarish bilan o'z ishini boshlaydi. Funksiya bu bir yoki bir necha amalni bajaruvchi dastur blokidir. Odatda funksiyalar boshqa funksiyalar orqali chaqiriladi, lekin `main()` funksiyasi alohida xususiyatga ega bo'lib u dastur ishga tushirilishi bilan avtomatik tarzda chaqiriladi.

`main()` funksiyasini boshqa funksiyalar kabi qaytaradigan qiymati tipini e'lon qilish lozim. `SALOM.cpp` dasturida `main()` funksiyasi `int` (`integer` – butun so'zidan olingan) tipli qiymat qaytaradi, ya'ni bu funksiya ishini tugatgandan so'ng operatsion sistemaga butun sonli qiymat qaytaradi. Operatsion sistemaga qiymat qaytarish unchalik muhim emas, umuman sistema bu qiymatdan foydalanmaydi, lekin C++ tili standarti `main()` funksiyasi barcha qoidalarga muvofiq e'lon qilinishini talab qiladi.

Eslatma: Ayrim kompilyatorlar `main()` funksiyasini `void` tipidagi qiymat qaytaradigan qilib e'lon qilish imkonini beradi. C++ da bundan foydalanmaslik kerak, chunki hozirda bunday uslub eskirgan. `main()` funksiyasini `int` tipini qaytaradigan qilib aniqlash lozim va buning hisobiga funksiyaning oxiriga `return 0` ifodasi yoziladi.

Barcha funksiyalar ochiluvchi figurali qavs (`{`) bilan boshlanadi va (`}`) yopiluvchi qavs bilan tugaydi. `main()` funksiyasi figurali qavsida 3 – satrdan 6 – satrgacha joylashtirilgan. Figurali qavslarni ichida joylashgan barcha satrlar funksiya tanasi deb aytiladi.

Bizning oddiy dasturimizning barcha funktsionalligi 4–satrda keltirilgan. `cout` ob'ekti ekranga ma'lumotni chiqarish uchun qo'llaniladi. `cin` va `cout` ob'ektlari mos ravishda ma'lumotlarni kiritish (masalan, klaviatura orqali) va

ularni chiqarish (ekranga chiqarish) uchun `main()` funksiyasi 6 – satr bilan tugallanadi.

Dastur tanasida konsol rejimi (Consol – rejimi bu MS DOS oynasi ko'rinishiga o'xshash oyna bo'lib, unda foydalanuvchi dastur tuzushda faqat dastur kodlari bilan ishlaydi. Graphic interface – rejimida esa faqat tilning kodlari bilangina emas muhitning menyulari, komponentalari bilan ham ishlashi mumkin bo'ladi) da belgilar ketma-ketligini oqimga chiqarish amali qo'llanilgan.

Ekranga ma'lumotni chiqarish uchun `cout` so'zini, undan so'ng chiqarish operatorini (oqimini) (`<<`) kiritish lozim. C++ kompilyatori (`<<`) belgisini bitta operator deb qaraydi.

2 – misol. `cout` ob'ektini qo'llanilishi

```
//cout ob'ektini qo'llanilishi
```

```
#include <iostream>
```

```
using namespace std; //Dev C++ muhitida qo'yilishi lozim
```

```
int main()
```

```
{ cout << "Bu son 5 ga teng:" << 5 << "\n";
```

```
cout << "endl operatori ekranda yangi ";
```

```
cout << " satrga o'tish amalini bajaradi";
```

```
cout << endl;
```

```
cout << "Bu katta son:\t" << 70000 << endl;
```

```
cout << "Bu 5 va 8 sonlarining yig'indisi:
```

```
<< \t" << 8+5 << endl;
```

```
cout << "Bu kasr son:\t\t" << (float) 5\8
```

```
<< endl;
```

```
cout << "Bu esa juda katta son: \t";
```

```
cout << (double) 7000*7000 << endl;
```

```
return 0; }
```

NATIJA:

- Bu son 5 ga teng: 5
- endl operatori ekranda yangi satrga o'tish amalini bajaradi

- Bu katta son: 70000
 - Bu 5 va 8 sonlarining yig`indisi: 13
 - Bu kasr son: 0.625
 - Bu esa juda katta son: 4.9e+07
- Eslatma: endl operatori *end line* (catr oxiri) degan so`zdan olingan bo`lib «end-el» deb o`qiladi.

Izohlar. Siz dastur yozayotgan vaqtingizda nima ish qilmoqchi ekanligingiz doimo aniq bo`ladi. Lekin bir oy yoki bir yildan so`ng bu dasturga qaytish lozim bo`lsa dasturga tegishli detallar va ularning vazifalari nimadan iborat ekanligini bilmasligingiz mumkin.

Dasturni butunlay xotirangizdan o`chirib yubormaslik va boshqalarga ham tushunarli bo`lishi uchun izohlardan foydalanish lozim. Izohlar kompilyator tomonidan tushirib qoldiriladigan dasturning alohida satrida yoki butun bir blokida qo`llaniladi.

```
/* va */ belgilar orasidagi har qanday ifoda izoh hisoblanadi.  
// belgidan keyin qatordagi har qanday ifoda izoh hisoblanadi.  
// C++ tilida yozilgan eng oddiy ko`rinishdagi dastur tuzilmasi  
# include <iostream> /* direktiva orqali chaqirilgan sarlavha fayli*/  
using namespace std; //Dev C++ muhitida qo`yilishi lozim  
int main() /* dasturga kirish nuqtasini ko`rsatuvchi  
funksiya*/  
{ /* blokning boshlanishini qayd qiluvchi figurali qavs*/  
    cout << "Salom"; /* natijani ekranga chiqarish operatori*/  
    /*va ular oxirini ko`rsatuvchi nuqta vergul belgisi*/  
    return 0; //funksiya qiymat qaytaradi.  
} // blokning tugashini qayd qilish
```

Ushbu dastur C++ tili qoidalariga rioya qilingan holda yozilgan.

Dastur matnidagi ishlatilgan nuqta vergul ko`rsatma oxirini qayd qiladi va u ajratgich deb ataladi. Ajratgich kompilyatorga ko`rsatma tugaganligini va undan keyin boshqa ko`rsatma berilishi yoki dastur tugashi kerakligini bildiradi. Har bir

alohida ko`rsatma oxiriga albatta nuqta vergul qo`yilishi lozim. Har bir qator oxiriga emas, balki alohida tugallangan ko`rsatma oxiriga nuqta vergul qo`yiladi. Ba`zan ko`rsatma bir qatorga sig`maydi va ikkinchi qatordan davom ettiriladi. Bu holda ko`rsatma tugagandan keyin bir marta nuqta vergul qo`yiladi.

3– misol.

```
#include <iostream>
using namespace std;
int main()
{   cout << "Salom, Men Muhammedov Mirshodman. Buxoro shahrida
    istiqomat qilaman.";
    cout << "Dastur tuzish yaxshi va bu mening birinch dasturim.";
    return 0;}
```

Dastur ishga tushirilsa ekranga quyidagi ikki qatordan iborat bo`lgan ma`lumot chiqadi.

Salom, Men Muhammedov Mirshodman. Buxoro shahrida istiqomat qilaman.
Dastur tuzish yaxshi va bu mening birinch dasturim.

C++ tili erkin formatli til hisoblanadi. Shuning uchun dastur qatori qayerdan boshlanishi va cheklagichlar qayerga qo`yilishi ahamiyatga ega emas. Yuqoridagi misolni bir necha xil usulda yozsak ham natija bir xilda bo`laveradi.

Dastur yozishda har bir dasturchining o`z uslubi mavjud. Ammo shunday bo`lsada har bir dasturchi ma`lum bir tartibga rioya qilgan holda, dasturning o`qilishini osonlashtirish maqsadida, probellardan unumli foydalanishi lozim. Chunki, kompilyatsiya jarayonida dastur matnidagi probellar olib tashlanadi.

Dastur matni o`qimishli bo`lishi uchun bir qancha tavsiyalarni berib o`tamiz va ularga rioya qilsangiz maqsadga muvofiq bo`lar edi.

- Doimo main() funksiyasini alohida qatorga yozing.
- Doimo ochuvchi va yopuvchi figurali qavslarni alohida qatorga yozing.
- Dastur matnini tabulyatsiyadan foydalangan holda yozing. Agar matn hajmi katta bo`lsa chekinishlar yordamida dastur tuzilmasini tushunarli tarzda

yo'zish va mantiqiy birliklarini ajratish natijasida uni ancha o'qimishli qilishingiz mumkin.

Bu talablarga rioya qilishingiz natijasida sizning dasturingiz matni boshqa dasturchilar uchun ham o'qimishli bo'ladi. Dasturda ortiqcha yoki kam sondagi probellarning bo'lishi kompilyator uchun ahamiyatga ega emas. Ammo qavslar va nuqta vergullarning tushirib qoldirilishi xatolik hisoblanadi. Bu ko'rinishdagi xatoliklar sintaksis xatoliklari deb ataladi. Sintaksis xatoliklar tuzatilgandan so'ng kompilyator ishchi dastur kodini yarata oladi.

C++ tilida direktivalar

Direktivalar – funksiyalar kutubxonasini chaqirish. Ular maxsus *include* katalogida joylashgan va *.h* kengaytmali fayllar bo'ladi. C++ tilida masalaning qo'yilishiga qarab kerakli kutubxonalar chaqiriladi. Bu esa dasturning xotirada egallaydigan joyini minimallashtiradi.

Masalan, ma'lumotlarni kiritish-chiqarish protseduralari uchun:

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

`#include "iostream.h"` joriy katalogdan chaqirish.

`#include` direktivasi kompilyatorga `iostream.h` sarlavha faylidagi ma'lumotdan foydalanish zarurligini ko'rsatadi. Qisqartirilgan `stdio` standart kiritish/chiqarish (standart input/output) so'zini anglatadi. `iostream.h` fayl, kompilyator disk fayllari va printer bilan ishlashi uchun kerakli bo'lgan ko'rsatmalarni saqlaydi.

Sarlavha faylida saqlanadigan ko'rsatmalarni `main()` funksiyadan oldin joylashtirish lozim.

Sarlavha fayli (`< >`) belgilari orasida berilsa, bu fayl `INCLUDE` katalogida mavjud bo'lishi mumkinligini kompilyatorga bildiradi. Instalyatsiya jarayonida sarlavha fayllari shu katalogga joylashtiriladi. Agar kompilyatsiya jarayonida sarlavha fayllari joriy katalogdan topilmasa, kompilyator uni `INCLUDE` katalogidan izlaydi.

Siz sarlavha faylini qo'shtirnoq ichiga olib ham yozishingiz mumkin.

```
#include "iostream.h"
```

Bu holda kompilyator sarlavha faylini faqat joriy katalogdan izlaydi, agar uni topa olmasa, xatolik mavjudligi xabarini beradi. Shuning uchun kompilyator hujjati bilan tanishib, qaysi holat o`rinli ekanligini aniqlang.

C++ dasturlash tili bilan ishlovchi eng sodda dasturlash muhiti Dev C++ va CodeBlocks dasturlari hisoblanadi. Ularning tarkibida 300 dan ortiq kutubxonalar mavjud. Eng ko`p ishlatiladigan kutubxonalar quyidagilar:

```
#include<iostream.h>
#include <math.h>
#include <conio.h>
#include <graphics.h>
#include <memory.h> va boshqalar
```

Makrolar (#define) – dastur bajarilishi davomida o`zgaruvchi ko`rsatilgan qiymatni qabul qilishi uchun ishlatiladi (const). Unda makroning nomi va qiymati ko`rsatiladi. Masalan:

```
#define pi 3.1415
#define x 556
#define s[100]
#define M x*x*x
```

Nazorat savollari:

1. C++ tilida dastur tuzilmasini tavsiflang.
2. Dasturdagi xatolar haqida tushuncha bering.
3. Kompilyator ish jarayonini tavsiflang.
4. Kompilyator va interpretator orasidagi farq nimadan iborat?
5. C++ tilining kompilyatorlari orasida tafovut mavjudmi?
6. Assembler tilining yuqori darajali tillardan farqi nimada?
7. Dastur matni boshlang`ich fayli nima?
8. Kompilyatsiya xatoligining bajarilish xatoligidan farqi nimada?
9. C++ tilining alfaviti va xizmatchi so`zlari haqida tushuncha bering.
10. Identifikator deganda nimani tushunasiz.
11. Funksiyalar kutubxonasi haqida tushuncha bering.
12. C++ tilida dasturning tuzilishi qanday?
13. #include direktivasining vazifasi nimadan iborat?
14. main() funksiyasining vazifasi nimadan iborat?

C++ ALGORITMIK TILINING KIRITISH VA CHIQRISH OPERATORLARI.

C++ algoritmik tilida ma'lumotlarni kiritish va chiqarish.

Qo'yidagi funksiyalar dasturda simvollarni kiritish va chiqarish uchun ishlatiladi. `getch(arg)` – bitta simvol kiritilishini kutish. Kiritilayotgan simvol monitorda aks etmaydi. Bu funksiyani programma oxirida argumentsiz ishlatilsa, monitorda ma'lumotlarni to klavisha bosilguncha o'qish mumkin bo'ladi.

`putch(arg)` – bitta simvolni standart oqimga chiqarish uchun ishlatiladi. Simvol monitorda aks etmaydi.

`getchar(arg)` – bitta simvol kiritilishini kutish. Kiritilayotgan simvol monitorda aks etadi. Bu funksiyani programma oxirida argumentsiz ishlatilsa, monitorda ma'lumotlarni to klavisha bosilguncha o'qish mumkin bo'ladi.

`putchar(arg)` – bitta simvolni standart oqimga chiqarish uchun ishlatiladi. Simvol monitorda aks etadi. Bu funksiyalar *iostream.h* modulida joylashgandir.

Misol:

```
Include <iostream.h>
Void main()
{
    int c;
    c=getchar();
    putchar(c);
    c=getch();
    putchar();
    getch();
}
```

Formatli chiqarish – printf.

`Printf` funksiyasi ko'rsatilgan parametrlarni standart oqimga chiqarish uchun ishlatiladi. Funksiya *iostream.h* modulida joylashgan bo'lib, umumiy ko'rinishi qo'yidagicha:

```
Printf(control, arg1, arg2, ...)
```

Control boshqaruvchi qator deb atalib, ikki turdagi simvollardan iborat bo'ladi: oddiy chiqariluvchi simvollar va navbatdagi parametрни o'zgartirib chiqaruvchi spetsifikatsiyalar. Har bir spetsifikatsiya % simvolidan boshlanib o'zgartirish turini ko'rsatuvchi simvol bilan tugaydi. % belgisi va o'zgartirish simvoli orasiga qo'yidagi simvollarni qo'yish mumkin. Chiqarilayotgan argument chapga tekislash lozimligini ko'rsatuvchi minus belgisi. Maydon minimal uzunligini ko'rsatuvchi raqamlar qatori.

Maydon uzunligini keyingi raqamlar qatoridan ajratuvchi nuqta.

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

Chiqarilayotgan son **long** tipiga tegishli ekanligini ko'rsatuvchi uzunlik markeri l.

O'zgartirish simvollar qo'yidagilardan iborat.

d – parametr unli butun songa aylantiriladi.

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

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

h – parametr ishorasiz o'nlik songa aylantiriladi.

c – parametr bitta simvol deb qaraladi.

s – parametr satr simvollar nolinchi simvol uchramaguncha yoki ko'rsatilgan sondagi simvollar bosiladi

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

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

g - %e yoki %f sifatida ishlatiladi.

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

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

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

```
:%10S:      :HELLO, WORLD:
:%10-S:      :HELLO, WORLD:
:%20S:      :  HELLO, WORLD:
:%-20S:      :HELLO, WORLD  :
:%20.10S:    :  HELLO, WOR:
:%-20.10S:   :HELLO, WOR   :
:%.10S:      :HELLO, WOR:
```

Formatli kiritish – scanf.

Scanf funksiyasi [iostream.h](#) modulida joylashgan bo'lib, umumiy ko'rinishi qo'yidagichadir:

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

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

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

% simvolidan boshlanuvchi spetsifikatsiya simvollarini;

% simvolidan boshlanuvchi qiymat berishni ta'qiqlovchi * simvoli;

% simvolidan boshlanuvchi maydon maksimal uzunligini ko'rsatuvchi son;

Quyidagi spetsifikatsiya simvollarini ishlatish mumkin:

d – o'nli butun son kutilmoqda.

o – 0 bilan boshlangan yoki boshlanmagan sakkizlik son kutilmoqda.

x – 0x belgili eki belgisiz o'n oltilik son kutilmoqda.

h – o'nlik son kutilmoqda.

c – bitta simvol kutilmoqda.

s – satr kutilmoqda.

f - [float](#) tipidagi son kutilmoqda. Kiritilayotgan sonning butun raqamlari va nuqtadan so'ng kasr raqamlari soni va E yoki e belgisidan so'ng mantissa raqamlari soni ko'rsatilishi mumkin.

Fayllar. Oqimli kiritish va chiqarish.

C++ tilining asosiy xususiyatlaridan biri oldindan rejalashtirilgan fayllar strukturasi yo'qligidir. Hamma fayllar, baytlar ketma-ketligi deb ko'riladi. U N I X operatsion sistemasida har bir qurilmaga «Maxsus fayl» mos keladi, shuning uchun C++ bibliotekasidagi funksiyalar fayllar bilan ham, qurilmalar bilan ham ma'lumot almashinishi uchun foydalaniladi. C++ tili bibliotekasida kiritish – chiqarish, quyi darajadagi kiritish, chiqarish va portlar uchun kiritish – chiqarish, oqimli daraja tizim xususiyatlariga bog'liq bo'lishi uchun bu yerda qaralmaydi.

Oqimli chiqarish va kiritishda ma'lumotlar bilan almashish baytma-bat amalga oshiriladi. Lekin tashqi xotira qurilmalari bilan almashish oldidan belgilangan ma'lumotlar bloki orqali amalga oshiriladi. Odatda u blokning minimal hajmi 512 yoki 1024 baytga teng bo'ladi. Diskga o'qilishda ma'lumotlar operatsion qatordagi buferi yoziladi, so'ngra baytma bayt buferga yig'iladi va diskka har bir murojaat qilinganda yagona blok sifatida uzatiladi. Shuning uchun ma'lumot almashishi diskka to'g'ridan to'g'ri murojaat qilishiga ko'ra tezroq amalga oshadi.

Oqim bilan ishlashda qo'yidagi vazifalarni bajarish mumkin.

- Oqimlarni ochish va yopish.
- Simvol, qator satr, formatlangan ma'lumot ixtiyoriy uzunlikdagi ma'lumotlarni kiritish yoki chiqarish va fayl oxiriga etganlik shartini tahlil qilish;
- Buferlash va bufer hajmini boshqarish;
- Ko'rsatkich oqimdagi o'rnini aniqlash yoki yangi o'ringa ko'chirish.

Bu vazifalarni boshqaruvchi funksiyalar teng foydalanish dasturiga [Stdio.h](#) – faylini ulash lozim.

Dastur bajarilishi boshlanganda avtomatik ravishda 5 ta oqim ochilib, bulardan:

- Standart kiritish oqimi stdin;

- Standart chiqarish oqimi stdout;
- Xatolar haqida malumotlar standart oqimi stderr;

Oqimlarni ochish va yopish

Oqim ochilishi uchun, oldindan kiritilgan FILE tipidagi struktura bilan bog'lash lozim. FILE strukturasini ta'riflash iostream.h bibliotekasida joylashgan. Bu strukturada buferga ko'rsatkich, o'qilayotgan pozitsiyaga ko'rsatkich va boshqa ma'lumotlar saqlanadi. Oqim ochilganda dasturda oqimga ko'rsatkich ya'ni FILE strukturali tipdagi ob'ektga ko'rsatkich qaytariladi. Bu ko'rsatkich qo'yidagicha e'lon qilinishi lozim.

FILE * <ko'rsatkich nomi>

Misol uchun FILE * fp

Oqim ochish funksiyasi quyidagi ko'rinishga ega;

<oqimga ko'rsatkich nomi>=fopen(<fayl-nomi>,<ochish rejimi>)

Misol uchun:fp=fopen("t.tnt", "r")

Oqim bilan bog'liq faylni quyidagi rejimlarda ochish mumkin:

“w”- Yangi fayl o'qish uchun ochiladi. Agar fayl mavjud bo'lmasa yangidan yaratiladi.

“r” - Mavjud fayl faqat o'qish uchun ochiladi.

“a” - Fayl davom ettirish uchun ochiladi.

“wt” - Fayl yozish va tahrirlash uchun ochiladi. Faylni ixtiyoriy joyidan o'qish yoki yozish mumkin.

“rt”- faylni ixtiyoriy joyidan o'qish yoki yozish mumkin, lekin fayl oxiriga qo'shish mumkin emas.

“at” - Fayl ixtiyoriy joyidan o'qish va yozish uchun ochiladi “wt” rejimidan farqli ravishda fayl oxiriga ma'lumot qo'shish mumkin.

Oqim ochilganda quyidagi xatolar kelib chiqishi mumkin: ko'rsatilgan fayl mavjud emas (o'qish rejimida); disk to'la yoki yozishdan himoyalangan va hokazo. Yana shuni aytish kerakki fopen() funksiyasi bajarilganda dinamik xotira ishlatiladi. Agar xotirada joy qolmagan bo'lsa “not enough ” - xatosi kelib chiqadi.

Ko'rsatilgan hollarda ko'rsatkich ~ NULL qiymatga ega bo'ladi.

Bu xatolar haqidagi ma'lumotlarni ekranga chiqarish uchun perror() funksiyasi ishlatiladi. Bu funksiya [iostream.h](#) bibliotekasida saqlanuvchi prototipi quyidagi ko'rinishga ega.

```
Void perror(court char * s);
```

Fayllar bilan ishlashning bitli rejimi.

Fayl bilan bitli almashish rejimi [gets\(\)](#) va [puts\(\)](#) funksiyalari yordamida tashkil etiladi. Bu funksiyalarga qo'yidagi shaklda murojat etiladi:

```
C=gets(fp);
```

```
Puts(c, fp);
```

Bu yerda fp - ko'rsatkich

C-int tipidagi o'zgaruvchi

Misol tariqasida klaviaturadan simvol kiritib faylga yozishni ko'ramiz. Matn oxirini '#' belgisi ko'rsatadi. Fayl nomi foydalanuvchidan so'raladi. Agar <enter> klavishasi bosilsa, faylga CR va LF (qiymatlari 13 va 10) konstantalar yoziladi. Keyinchalik fayldan simvollarni o'qishda bu konstantalar satrlarni ajratishga imkon beradi.

```
#include <iostream.h>

int main()
{
    file *fp;
    char c;
    const char CR='\015';
    const char LF='\012';
    char f name [20];
    puts("fayl nomini kiriting:\n");
    gets(f name);
    if((fp=fopen(f name, "w")) ==null)
    {
        perror(f name);
        return 1;
    }
}
```

```
        while ((c=getchar())!='#')
    }
    if (c=='\n')
    { puts(CR,fp);
      puts(LF,fp);
    }
    else puts (c,fp);
}
Fclose (fp);
Return 0;
```

Keyingi programma fayldan simvollarni o'qib ekranga chiqaradi.

```
#include <iostream.h>
int main()
{ file *fp;
  char c;
  char f name [20];
  puts("fayl nomini kiriting:\n");
  if((fp=fopen (f name, "r")) ==null)
  { perror(f name);
    return 1;
  }

  while ((c=getc(fp))!=eof)
  putchar(c);
  fclose (fp);
  return 0;
}
```

Satrlar yordamida fayllar bilan bog'lanish.

Matnli fayllar bilan ishlash uchun [fget](#) va [fputs](#) funksiyalaridan foydalaniladi. Bu funksiyalari prototiplari [iostream.h](#) faylida qo'yidagi ko'rinishga ega:

```
Int fputs (const char *s, FILE *stream);
Char *fgets (char * s, int n, FILE * stream);
```

Fputs funksiyasi '\0' simvoli bilan chegaralangan satrni stream ko'rsatkichi orqali aniqlangan faylga yozadi. '\0' simvoli faylga yozilmaydi.

Fgets() funksiyasi stream ko'rsatkichi orqali aniqlangan fayldan (n-1) simvolni o'qiydi va S ko'rsatgan satrga yozib qo'yadi. Funksiya n-1 simvolni o'qib bo'lsa yoki 1-chi qator simvoli '\n'ni uchratsa ishini to'xtatadi. Har bir satr oxiriga qo'shimcha \0 belgisi qo'shiladi. Xato bo'lganda yoki fayl oxiriga yetganda agar fayldan birorta simvol o'qilmagan bo'lsa **NULL** qiymat qaytariladi. Qo'yidagi dasturda bir fayldagi matnni ikkinchi faylga yozishni ko'rib chiqiladi. Har qanday dastur operatsion sistemada ma'lumotni **argc** va **argv** parametrlar qiymati sifatida oladi. Birinchi programmaga uzatilayotgan satrlar sonini ko'rsatadi. Argv[0] bu faylning nomini saqlovchi satrga ko'rsatkich massivining qolgan elementlari argv[1]...argv[argc-1] komanda qatorida fayl nomidan so'ng bo'shlik tashlab yozilgan parametrlarga ko'rsatkichlar.

Programma nomi copyfile.exe bo'lsin va bu programma yordamida f1.dat faylni f2.dat faylga yoziladi. Komanda qatori qo'yidagi ko'rinishga ega:

```
<copyfile.exe f1.dat f2.txt
```

Programma matni:

```
#include <iostream.h>
main (int argc, char*argv[])
{ char cc[256];
  FILE *f1, *f2;
  If (argc!=3)
{ print ("\n Format bazovix programm:");
  print f ("\n copyfile.exe")
  Cout<< ("\n Fayl priyomnik");
  return 1;
}
if ((f1=fopen(argv[1], "r"))==NULL)
```

```
{perror(argv[1]);  
    return 1;  
}  
if ((f2=fopen(argv[2], "w"))==NULL)  
{perror(argv[2]);  
    return 1;  
}  
  
while (fgets(cc,256,f1)!=NULL)  
    fputs(CC,f2);  
fclose(f1);  
fclose(f2);  
return 0;  
}
```

Bu dastur bajarilishi natijasida int.dat fayliga Cout<< funksiyasi yordamida monitorga qanday chiqsa shunday ko'rinishda ma'lumotlar yozadi.

Nazorat savollari:

1. Chiqarish operatorini tavsiflang.
2. Kiritish va natija.
3. Ifoda nima?
4. $x=5+7$ yozuvi ifoda bo'la oladimi? Uning qiymati nechaga teng?
5. $201/4$ ifodaning qiymati nechaga teng?
6. $201\%4$ ifoda qiymati nechaga teng?
7. Operator deganda nimani tushunasiz?
8. Taqqoslash amallari haqida tushuncha bering.

C++ ALGORITMIK TILINING TARMOQLANISH OPERATORLARI.**Tarmoqlanuvchi dastur tuzilmasi.**

if operatori. Odatda dastur satrma–satr tartib bilan bajariladi. *if* operatori shartni tekshirish (masalan, ikki o`zgaruvchi tengmi) va uning natijasiga bog`liq ravishda dasturni bajarilish tartibini o`zgartirish imkonini beradi. *if* operatorining oddiy shakli quyidagi ko`rinishdadir:

if (shart)

ifoda.

Qavs ichidagi shart ixtiyoriy ifoda bo`lishi mumkin. Agarda bu ifoda *false* qiymatini qaytarsa undan keyingi ifoda yoki blok tushirib qoldiriladi. Agarda shart *true* qiymat qaytarsa navbatdagi ifoda bajariladi.

if operatorida figurali qavs ichiga olingan ifodalar blokini ham ishlatish mumkin.

if (shart)

{

1 - ifoda

2 - ifoda

3 - ifoda

}

Quyida ifodalar blokining qo`llanilishiga oid misol keltirilgan

if(kattaSon>kichikSon)

{ *kattaSon=kichikSon*

cout<<"kattaSon:"<<kattaSon << "/n";

cout<<"kichikSon:"<<kichikSon<< "/n"; }

Bu holda *kattaSon* o`zgaruvchisiga nafaqat *kichikSon* o`zgaruvchisi o`zlashtirilayapti, balki ekranga bu haqida axborot ham chiqarilayapti.

1 – misol. Munosabat operatorining qo`llanilishi orqali tarmoqlanishga misol.

// 1. – misolda munosabat operatori bilan birgalikda *if* instruktsiyasining

// qo`llanilishi namoyish etilgan

```

# include <iostream>
using namespace std;
int main( ) {
    int BuxoroGol, PaxtakorGol;
    cout<<"Buxoro komandasi kiritgan to`plar"
    << "sonini yozing:";
    cin >> BuxoroGol;
    cout<<"Paxtakor komandasi kiritgan"
    << "to`plar sonini yozing:";
    cin >> PaxtakorGol;
    cout << "\n";
    if ( BuxoroGol>PaxtakorGol)
    cout << "Yashasin Buxoro!\n"
    if (BuxoroGol < PaxtakorGol)
    {
        cout << "Yashasin Paxtakor. Gol. \n"
        cout << "Bugun Toshkentda bayram!\n";
    }
    if (BuxoroGol==PaxtakorGol)
    {
        cout << "Durrangmi? Yo-oq? Bo`lishi"<<
        " mumkin emas \n";
        cout <<"Paxtakorning kiritgan to`plari"
        << "haqida ma`lumotni qaytadan
        yozing\n"
        cin >> PaxtakorGol;
        if (BuxoroGol>PaxtakorGol)
        {
            cout<<"Buxoro yutishini oldindan
            bilgan"

```

```

    <<"    edim!    Shuning    uchun    qayta
    so`radim\n";
    cout<< "Yashasin Buxoro!";
    if (BuxoroGol<PaxtakorGol)
    {
        cout<<"Paxtakor            yutishini
        oldindan bilgan"
        <<"    edim!    Shuning    uchun    qayta
        so`radim\n";
        cout<< "Yashasin Paxtakor!";
        cout    <<    "Bugun    Toshkentda
        bayram!\n";
        if (BuxoroGol==PaxtakorGol)
        cout<<"Qoyil!    Haqiqatan    ham
        durang ekan\n";
    }
    cout<<"\n    Ma`lumotingiz    uchun
    rahmat\n";
}

}
return 0;
}

```

NATIJA:

- Buxoro komandasi kiritgan to`plar sonini yozing: 3
- Paxtakor komandasi kiritgan to`plar sonini yozing: 3
- Durrangmi? Yo-oq? Bo`lishi mumkin emas
- Paxtakorning kiritgan to`plari haqida ma`lumotni qaytadan yozing: 2
- Buxoro yutishini oldindan bilgan edim! Shuning uchun qayta so`radim
- Yashasin Buxoro!

else kalit so`zi.

Dasturlarda ko`p hollarda biror bir shartning bajarilishiga (ya`ni bu shart *true* qiymat qaytarsa) bog`liq ravishda bir blok, uning bajarilmasligiga asosan esa (ya`ni bu shart *false* qiymat qaytarsa) boshqa bir blokning bajarilishi talab qilinadi. Misoldagi birinchi tekshirish (BuxoroGol>PaxtakorGol) *true* qiymat qaytarsa ekranda bir xabar, *false* qiymatida esa boshqa bir xabar chiqariladi.

Bunday masalalarni yuqorida ko`rsatilgan usul, ya`ni qator shartlarni tekshirish uchun bir nechta *if* operatorini qo`llash orqali hal qilish mumkin. Dasturning soddaligini ta`minlash uchun *else* kalit so`zidan foydalanish mumkin.

if (shart)

Ifoda

else

Ifoda

2 – misol. *else* kalit so`zining ishlatilishi.

// 2 misol. If va else kalit so`zlarini ishlatilishiga oid misol

```
# include <iostream>
using namespace std;
int main()
{
    int BirinchiSon, IkkinchiSon;
    cout << "Katta sonni kiriting: ";
    cin >> BirinchiSon;
    cout<<"\n Kichik sonni kiriting:";
    cin >> IkkinchiSon;
    if (BirinchiSon > IkkinchiSon)
        cout << "\n Rahmat! \n";
    else
        cout << "\n Ikkinchisi katta son-ku!".
    return 0;
}
```

NATIJA:

Katta sonni kiriting: 10

Kichik sonni kiriting: 12

Ikkinchisi katta son - ku!

if operatori orqali murakkab konstruktsiyalarni hosil qilish. if-else konstruktsiyasida ifodalar blokida ixtiyoriy operatorlarni ishlatishda hech qanday chegara yo`q. Shu jumladan, ifodalar bloki ichida yana if-else operatorlarini ishlatish mumkin. Bu holda bir nechta if operatoridan iborat ichma – ich konstruktsiya hosil bo`ladi.

```

if (1-shart)
{
    if (2-shart)
        1-ifoda
    else
    {
        if (3-shart)
            2-ifoda
        else
            3-ifoda
    }
}
else
    4-ifoda.

```

Ushbu bir nechta *if* operatoridan tashkil topgan konstruktsiya quyidagi tartibda ishlaydi: agarda 1-shart va 2-shart rost bo`lsa 1-ifoda bajariladi. Agarda 1-shart rost va 2-shart yolg`on natija qaytarsa, u holda 3-shart tekshiriladi va agarda bu shart rost bo`lsa 2-ifoda, yolg`on bo`lsa esa 3-ifoda bajariladi. Va eng oxiri, agarda 1-shart yolg`on bo`lsa 4-ifoda bajariladi. Bunday murakkab konstruktsiyaga misol 3 – misolda keltirilgan.

3 – misol. *if* operatori ichki bo`lgan murakkab konstruktsiya.

// 3 – misol. *If* operatori ichki bo`lgan murakkab konstruktsiyaga misol

```
# include < iostream.h>
using namespace std;
int main()    {
//   Ikkita   son   kiritamiz.Ularni   BirinchiSon   va
IkkinchiSon o`zgaruvchilariga beramiz
//Agarda KattaSon qiymati KichikSon qiymatidan katta
bo`lsa, katta son kichigiga
//qoldiqsiz bo`linishini tekshiramiz.   Agarda   u
koldiksiz bo`linsa ular teng yoki
//teng emasligini tekshiramiz.
    int BirinchiSon, IkkinchiSon;
        cout<<"Ikkita son kiriting.\n Birinchisi: ";
        cin >> BirinchiSon;
        cout << "\n Ikkinchisi: ";
        cin >> IkkinchiSon;
        cout << "\n\n";
        if (BirinchiSon>=IkkinchiSon)
            if ( (BirinchiSon%IkkinchiSon)==0)
                {
                    if (BirinchiSon==IkkinchiSon.)
                        cout<< "Ular bir - biriga teng!\n";
                    else
                        cout<< "Birinchi son ikkinchisiga"<<
                            "karrali!\n";
                }
            else
                cout<<"Ikkinchi son katta!\n";
return 0;
}
```

NATIJA

Ikkita son kiriting

Birinchisi: 9

Ikkinchisi: 3

Birinchi son ikkinchisiga karrali!

Mantiqiy operatorlar.

Dasturlashda bir emas balki bir nechta shartli ifodalarni tekshirish zaruriyati juda ko'p uchraydi. Masalan, x o'zgaruvchisi y o'zgaruvchisidan, y esa $o'z$ navbatida z o'zgaruvchisidan kattami sharti bunga misol bo'la oladi. Bizning dasturimiz mos amalni bajarishdan oldin bu ikkala shart rost yoki yolg'onligini tekshirishi lozim.

Quyidagi mantiq asosida yuqori darajada tashkil qilingan signalizatsiya sistemasini tasavvur qiling. Agarda eshikda signalizatsiya o'rnatilgan bo'lsa VA kun vaqti kech soat olti VA bugun bayram YOKI dam olish kuni BO'LMASA politsiya chaqirilsin. Barcha shartlarni tekshirish uchun C++ tilining uchta mantiqiy operatori ishlatiladi.

Mantiqiy operatorlar

Operator	Belgi	Misol
VA	&&	1 ifoda && 2 ifoda
YOKI		1 ifoda 2 ifoda
INKOR	!	!ifoda

Mantiqiy ko'paytirish operatori. Mantiqiy ko'paytirish operatori ikkita ifodani hisoblaydi, agar ikkala ifoda *true* qiymat qaytarsa VA operatori ham *true* qiymat qaytardi. Agarda sizning qorningiz ochligi rost bo'lsa VA sizda pul borligi ham rost bo'lsa siz supermarketga borishingiz va u yerdan o'zingizga tushlik qilish uchun biror bir narsa xarid qilishingiz mumkin. Yoki yana bir misol, masalan,

$if(x==5)\&\&(y==5)$

mantiqiy ifodasi agarda x va y o'zgaruvchilarini ikkalasining ham qiymatlari 5 ga teng bo'lsagina *true* qiymat qaytaradi. Bu ifoda agarda o'zgaruvchilardan

birortasi 5 ga teng bo'lmagan qiymat qabul qilsa *false* qiymatini qaytaradi. Mantiqiy ko'paytirish operatori faqatgina o'zining ikkala ifodasi ham rost bo'lsagina *true* qiymat qaytaradi.

Mantiqiy ko'paytirish operatori && belgi orqali belgilanadi.

Mantiqiy qo'shish operatori. Mantiqiy qo'shish operatori ham ikkita ifoda orqali hisoblanadi. Agarda ulardan birortasi rost bo'lsa mantiqiy qo'shish operatori *true* qiymat qaytaradi. Agarda sizda pul YOKI kredit kartochkasi bo'lsa, siz schyotni to'lay olasiz. Bu holda ikkita shartning birdaniga bajarilishi: pulga ham va kredit kartochkasiga ham ega bo'lishingiz shart emas. Sizga ulardan birini bajarilishi yetarli. Bu operatorga oid yana bir misolni qaraymiz. Masalan,

if(x==5)||(y==5)

ifodasi yoki *x* o'zgaruvchi qiymati, yoki *y* o'zgaruvchi qiymati, yoki ikkala o'zgaruvchining qiymati ham 5 ga teng bo'lsa rost qiymat qaytaradi.

Mantiqiy inkor operatori. Mantiqiy inkor operatori tekshirilayotgan ifoda yolg'on bo'lsa *true* qiymat qaytaradi. Agarda tekshirilayotgan ifoda rost bo'lsa inkor operatori *false* qiymat qaytaradi. Masalan,

(if !(x==5))

ifodasining qiymati, agarda *x* o'zgaruvchisi 5 ga teng bo'lmasa *true* qiymat qaytaradi. Bu ifodani boshqacha ham yozish mumkin:

if(x!=5)

Switch operatori.

Ayrim masalalarni yechishda *if* operatori ichida ko'p sondagi *if* operatorlarini qo'llashga to'g'ri keladi. Bu esa dasturni yozishni ham, uni tushinishni ham murakkablashtirib yuboradi. Bunday muammoni yechish uchun C++ tilida *switch* operatori qo'llaniladi. Bu operatorning *if* operatoridan asosiy farqi shuki, unda bir yo'la bir nechta shart tekshiriladi. Natijada dasturni tarmoqlanishi nisbatan samaraliroq bo'ladi. *switch* operatorining sintaksisi quyidagicha:

switch(ifoda)

{

```

case Birinchi qiymat:  operator;
                        break;
case Ikkinchi qiymat:  operator;
                        break;

.....

case N inchi qiymat:    operator;
                        break;

default:                operator;
}

```

switch operatoridan keyingi qavs ichida tilning konstruktsiyasi nuqtai-nazaridan to'g'ri bo'lgan ixtiyoriy ifodani ishlatish mumkin. Operator identifikatori o'rnida ham ixtiyoriy operator yoki ifoda, hamda operator va ifodalarning ketma-ketligini ishlatish mumkin. Lekin bu yerda mantiqiy operatsiyalar yoki taqqoslash ifodalarini ishlatish mumkin emas.

switch operatorining qo'llanilishi. *switch* operatorining qo'llanilish sintaksisi quyidagicha:

```

switch(ifoda)
{
    case 1-qiymat: ifoda;
    case 2-qiymat: ifoda;
    ...
    case n-qiymat: ifoda;
    default : ifoda;
}

```

switch operatori orqali dasturning tarmoqlanishi bir necha mumkin bo'lgan qiymatlarni qaytaruvchi ifodaning natijasi asosida tashkil etiladi. *switch* operatoridagi qavs ichida berilgan ifodaning qaytargan qiymati *case* operatoridan keyinda ko'rsatilgan qiymat bilan solishtiriladi. Ifodaning qiymati bilan *case* operatoridan keyingi qiymat mos kelsa tanlangan *case* operatoridan keyingi barcha satrlar bajariladi. Bunda amallarni bajarilishi *break* operatorigacha davom etadi.

Agarda *case* operatorlari qiymatidan birortasi ham qaytarilgan qiymatga mos kelmasa *default* operatoridan keyingi dastur satrlari bajariladi. Agarda bu operator mavjud bo`lmasa boshqaruv *switch* bloki tanasidan chiqadi va keyingi dastur satrlariga beriladi.

4-misol.

```
switch(choice)
{
case 0:
    cout<< "zero!"<< endl;
    break;
case 1:
    cout<< "one!"<< endl;
    break;
case 2:
    cout<< "two!"<<endl;
    break;
default:
    cout<< "default!"<<endl;
}
```

5-misol

```
switch (choice)
{
case 0:
case 1:
case 2:
    cout< "Less than 3!"<< endl;
    break;
case 3:
    cout<< Equals 3!" << endl;
    break;
default:
    cout<< Greater than3 ! " << endl;
}
```

Operator yoki ifodalardan keyin *break* operatori qo`llanilmasa joriy *case* operatoridan keyingi *case* blokidagi barcha ifodalar bajariladi. Ko`p hollarda bunday holatda xatolik ro`y beradi. Shuning uchun, *break* operatorini tushirib qoldirsangiz bu amalni tavsiflovchi mos izohni yozishni unutmang.

Switch operatorining qo`llanilishi 6-misolda keltirilgan.

6 – misol. switch operatorining qo`llanilishi.

```

#include <iostream.h>
using namespace std;
int main()
{ unsigned short int  number;
    cout<< "1 dan 5 gacha bo'lgan sonlarning birini
kiriting va Enter klavishini bosing:  "
    cin>> number;
    switch (number)
    {
case 0:    cout << "Salom!";
           break;
case 5:    cout<<  "Sardor \n";
case 4:    cout << "Mirshod \n" ;
           case 3:    cout<< "Mironshoh \n";
           case 2:    cout << "Mirxon \n"
           case 1:    cout << "Muhammad \n";
                break;
default:   cout << "Hisobdan tashqari.\n";
           break;
    }
    cout<< "\n\n";
    return 0;
}

```

TAHLIL. Dastur oldin son kiritishni so'raydi. Keyin esa kiritilgan son *switch* operatori orqali tekshiriladi. Agarda 0 kiritilgan bo'lsa unga muvofiq ravishda ekranga 9 – satrda yozilgan kichik son xabari chiqariladi. Va undan keyin yozilgan *break* operatori *switch* konstruktsiyasini bajarilishini yakunlaydi. Agarda 5 soni kiritilsa, boshqaruv 11 – satrga beriladi va unga muvofiq xabar chiqariladi. Undan keyingi toki *break* komandasigacha barcha satrlar ketma – ket bajariladi.

Nazorat savollari:

1. Ifoda nima?
2. $x=5+7$ yozuvi ifoda bo`la oladimi? Uning qiymati nechaga teng?
3. Tarmoqlanuvchi dastur haqida tushuncha bering.
4. *if* operatori haqida tushuncha bering.
5. *else* kalit so`zi haqida tushuncha bering.
6. Tarmoqlanuvchi dasturda bloklarning vazifasi nimadan iborat?
7. $\text{If}(x = 3)$ va $\text{if}(x = =3)$ ifodalarning farqi nimadan iborat?
8. Ichma-ich joylasgan tarmoqlanuvchi jarayon haqida tushuncha bering.
9. Mantiqiy operatorlar haqida tushuncha bering.
10. Mantiqiy ko`paytirish operatori haqida tushuncha bering.
11. Mantiqiy qo`shish operatori haqida tushuncha bering.
12. Mantiqiy inkor operatori haqida tushuncha bering.
13. *switch* operatori haqida tushuncha bering.
14. *switch* operatorining qo`llanilishini tavsiflang.

C++ ALGORITMIK TILINING TAKRORLASH OPERATORLARI.**Takrorlash operatorlari**

Har qanday dasturning strukturasi tarmoqlanish va takrorlanish to'plamining kombinatsiyasidan iborat bo'ladi. Quyidagi savollar paydo bo'ladi:

1. Takrorlanish nima va ular qanday ishlatiladi?
2. Takrorlanishni tashkil etishning qanday usullari bor?
3. Ko'p tarmoqlanuvchi `if/else` konstruktsiyasi o'rniga boshqa konstruktsiyani ishlatilishi.

Takrorlanishni tashkil etish. Qator masalalarni yechish uchun ko'pincha bitta amalni bir necha marotaba bajarish talab qilinadi. Amaliyotda bu rekursiyalar va iterativ algoritmlar yordamida amalga oshiriladi. Iterativ jarayonlar – bu operatsiyalar ketma-ketligini zaruriy sonda takrorlanishidir.

Takrorlash operatori «Takrorlash sharti» deb nomlanuvchi ifodaning rost qiymatida dasturning ma'lum bir qismidagi operatorlarni (takrorlash tanasini) ko'p marta takror ravishda bajaradi(iterativ jarayon).

Takrorlash o'zining kirish va chiqish nuqtalariga ega, lekin chiqish nuqtasi bo'lmazligi mumkin. Bu holda takrorlashga cheksiz takrorlash deyiladi. Cheksiz takrorlash uchun takrorlashni davom ettirish sharti doimo rost bo'ladi.

Takrorlash shartini tekshirish takrorlash tanasidagi operatorlarni bajarishdan oldin tekshirilishi mumkin (`for`, `while` takrorlashlari) yoki takrorlash tanasidagi operatorlari bir marta bajarilgandan keyin tekshirilishi mumkin (`repeat-until`, `do-while`).

while operatori orqali takrorlanishni tashkil etish.

`while` operatori yordamida takrorlanishni tashkil etishda operatsiyalar ketma-ketligi takrorlashning davom etish sharti “to'g'ri” bo'lsagina uning navbatdagi operatsiyalari amalga oshiriladi.

1 – misol. `while` operatori yordamida takrorlashni tashkil etish.

```
#include <iostream>
using namespace std;
```

```

int main()    {
    int counter=0; //Birlamchi qiymatni o`zlashtirish
    while(counter<5)//Takrorlash shartini tekshirish
    {counter ++;
    cout << "counter :'" << counter << "; \n" ;}
    cout<<"                Takrorlash                tugadi.
    Counter:"<<counter<<".\n";
return  0;    }

```

NATIJA:

```

counter : 1
counter : 2
counter : 3
counter : 4
counter : 5

```

Takrorlash tugadi.Counter: 5.

while operatori orqali murakkab konstruktsiyalarni tuzish.

while operatori shartida murakkab mantiqiy ifodalarni ham qo'llash mumkin. Bunday ifodalarni qo'llashda && (mantiqiy ko'paytirish), || (mantiqiy qo'shish), hamda !(mantiqiy INKOR) kabi operatsiyalardan foydalaniladi. 2 - misolda while operatori konstruktsiyasida murakkabroq shartlarni qo'yilishiga misol keltirilgan .

2 – misol. while konstruktsiyasidagi murakkab shartlar.

```

include <iostream>
using namespace std;
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

TAHLIL

Misolda 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 takrorlashni 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 `MaxKichik` qiymatidan oshib ketmasligi.

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. Takrorlash agarda tekshirish sharti tarkibidagi birorta shart bajarilmasa to`xtatiladi.

break va ***continue*** operatorlari. Ko`pincha takrorlashning navbatdagi iteratsiyasiga takrorlash tanasidagi boshqa operatorlar (navbatdagi operatorlar) bajarilmasdan turib o`tish zaruriyati tug`iladi. Bunday holatlarda `continue` operatori qo`llaniladi. Bundan tashqari, takrorlashni bajarilishi sharti qanoatlantirilganda ham, qator hollarda undan chiqib ketish zaruriyati paydo bo`ladi. Bu holda esa `break` operatori ishlatiladi. Bunday operatorlarni qo`llanilishi 3 – misolda keltirilgan. Bu misol bizga oldingi misoldan tanish bo`lgan o`yinining biroz murakkablashtirilgan variantidir. Bu yerda kichik va katta qiymatlardan tashqari qadam va maqsadli kattalikni kiritish talab etiladi. Agarda kichik son qiymati qadam o`zgaruvchisiga (qadam) karrali bo`lmasa katta qiymati ikkiga kamaytiriladi. Qachonki, kichik o`zgaruvchisi qiymati katta o`zgaruvchisi qiymatidan katta bo`lsa o`yin tugatiladi. Agarda katta o`zgaruvchisi qiymati maqsadli kattalik (maqsad) ning qiymati bilan ustma – ust tushsa o`yin bekor qilinganligi haqida xabar ekranga chiqariladi.

3 – misol. `break` va `continue` operatorlarining qo`llanilishi.

```
include <iostream>
using namespace std;
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

while(true) konstruktsiyasini qo'llanilishi.

Takrorlashning navbatdagi iteratsiyasiga o'tishda shart sifatida C++ tilida sintaksis bo'yicha to'g'ri bo'lgan ixtiyoriy ifoda qatnashishi mumkin. Bunda shart "to'g'ri" bo'lsa takrorlash bajarilaveradi. Cheksiz takrorlanishni tashkil etish uchun shart sifatida `true` mantiqiy o'zgarmas qo'llaniladi. Bu 4 - misolda ko'rsatilgan.

4 – misol. `while` operatorini qo'llashga oid yana bir misol.

```
#include <iostream>  
using namespace std;  
int main()    {  
    int counter = 0 ;  
    while(true)  
    {  
        counter++ ;  
        if(counter>10)  
            break ;  
    }  
    cout<<"counter:"<<counter << "\n " ;  
    return 0 ;    }
```

HATIJA:

Counter: 11

do...while konstruktsiyasi yordamida takrorlashni tashkil etish.

Ayrim hollarda `while` operatori yordamida takrorlanishni tashkil etishda uning tanasidagi amallar umuman bajarilmasligi mumkin. Chunki takrorlashni davom etish sharti har bir iteratsiyadan oldin tekshiriladi. Agarda boshlang'ich berilgan shart to'g'ri bo'lmasa takrorlash tanasining birorta operatori ham bajarilmaydi. Bu 5 – misolda keltirilgan.

5–misol. `while` takrorlash tanasidagi amallarning bajarilmay qolishi

```
# include <iostream>
using namespace std;
int main()    {
    int counter ;
    cout << "How manu hellos ?:";
    cin >> counter ;
    while (counter>0 )
    {    cout << "Hello ! \n";
        counter -- ;    }
    cout<<"Counter is OutPut ." << counter ;
    return  0;    }
```

HATIJA:

How manu hellos ? : 2

Hello !

Hello !

Counter is OutPut : 0

How manu hellos ? : 0

Counter is OutPut : 0

do...while konstruktsiyasining qo'llanilishi.

do...while konstruktsiyasida takrorlash sharti uning tanasidagi operatsiyalar bir marta bajarilgandan so`ng tekshiriladi. Bu takrorlash operatorlarini hech bo`lmaganda bir marta bajarilishini kafolatlaydi.

Quyidagi dasturda oldingi dasturda keltirilgan variantning bir oz o`zgartirilgan shakli, ya`ni *while* operatori o`rniga *do...while* konstruktsiyasi qo`llangan shakli keltirilgan.

6–misol. *do...while* konstruktsiyasining qo`llanilishi.

```
# include <iostream>
using namespace std;
int main()    {
    int counter;
    cout<<"How manu hellos ?" ;
    cin >>counter;
    do
    {
        cout << "hello \h" ;
        counter --;
    }
    while(counter>0)
    cout << "Counter is :" << counter <<endl ;
    return 0 ; }
```

HATIJA :

how manu hellos ? 2

hello

hello

Counter is : 0

How manu hellos ? 0

Hello

Counter is: - 1

for operatori.

while operatori yordamida takrorlanishni tashkil etishda 3 ta zaruriy amallar: takrorlash o`zgaruvchisiga boshlang`ich qiymat berish, har bir iteratsiyada takrorlashni davom etish sharti bajarilishini tekshirish va takrorlash o`zgaruvchisi qiymatini o`zgartirishni bajarishimiz kerak.

7 – misol. *while* operatorining ishlatilishiga yana bir misol

```
# include < iostream>
using namespace std;
int main()      {
    int counter=0;
    while (counter <5)
    {
        counter++ ;
        cout << "Looping!";
    }
    cout << "\n Counter:" << Counter << "\n";
    return 0;      }
```

NATIJA:

- Looping! Looping! Looping! Looping! Looping!

Counter: 5

for operatori takrorlashni ishlashi uchun zarur bo`ladigan uchta operatsiyani o`zida birlashtiradi. Bu operatsiyalarni qisqacha quyidagicha xarakterlash mumkin: boshlang`ich qiymatni o`zlashtirish, shartni tekshirish, takrorlash schyotchigini qiymatini oshirish. *for* operatori ifodasidagi qavsning ichida shu uchchala operatsiyani amalga oshiruvchi ifodalar yoziladi. Qavs ichidagi ifodalar nuqtali vergul orqali ajratiladi.

`for` takrorlashining birinchi ifodasi takrorlash schyotchigiga boshlang'ich qiymatni o'zlashtiradi. Schyotchik – to'g'ridan-to'g'ri `for` takrorlashida e'lon qilinadigan va qiymat o'zlashtiriladigan butun sonli o'zgaruvchidir. C++ da bu o'rinda schyotchikka qiymat beradigan ixtiyoriy ifoda yozilishiga imkon berilgan. `for` takrorlashining ikkinchi parametrida takrorlashni davom etish sharti aniqlanadi. Bu shart `while` konstruktsiyasining sharti bajaradigan vazifani amalga oshiradi. Uchinchi parametrdada esa takrorlash schyotchigi qiymatini o'zgartiruvchi (oshiruvchi yoki kamaytiruvchi) ifoda yoziladi. `for` takrorlashidan foydalanish 8 – misolda keltirilgan.

8 – misol. `for` takrorlashining qo'llanilishiga misol.

```
#include<iostream>
using namespace std;
int main()    {
    int counter;
    for (counter=0 ; counter<5; counter++ )
        cout<< "Looping!";
    cout<< "\n Counter:" << counter<< " .\n";
    return 0;    }
```

NATIJA:

Looping! Looping! Looping! Looping! Looping!

Counter: 5

for operatori uchun murakkab ifodalarni berilishi.

`for` takrorlashi dasturlashning kuchli va qulay instrumentidir. `for` operatorida takrorlashni o'zaro bog'liq bo'lmagan parametrlar (boshlang'ich qiymat o'zlashtirish, bajarilish sharti va qadam) ni qo'llanilishi takrorlash ishini boshqarishda juda yaxshi imkoniyatlarni ochib beradi.

`for` takrorlashi quyidagi ketma-ketlikda ishlaydi.

1. Takrorlash schotchigiga boshlang'ich qiymat o'zlashtiriladi.
2. Takrorlashni davom etish shartidagi ifoda qiymati hisoblanadi.

3. Agarda shart ifodasi *true* qiymat qaytarsa oldin takrorlash tanasi bajariladi, keyin esa takrorlash schyotchigi ustida berilgan amallar bajariladi.

Har bir iteratsiyada 2 – va 3 – qadamlar takrorlanadi.

Takrorlashda bir nechta schyotchikni qo'llanilishi. `for` takrorlashining sintaksisi unda bir nechta o'zgaruvchi - schetchikni qo'llanilishiga, takrorlashni davom etishini murakkab shartlarini tekshirishga va takrorlash schyotchiklari ustida ketma-ket bir nechta operatsiyani bajarilishiga imkon beradi.

Agarda bir nechta schyotchikka qiymat o'zlashtirilsa yoki ular o'rtasida bir nechta operatsiya bajarilsa, bu ifodalar vergul bilan ajratilgan holda ketma – ket yoziladi. 9 – misolda ikkita schyotchikli `for` operatorining qo'llanilishi ko'rib chiqiladi.

9–misol. `for` takrorlashida bir nechta schyotchikni qo'llanilishi.

```
# include< iostream>
using namespace std;
int main() {
    for (int i=0, j=0; i<3; i++, j++)
        cout<< "i:" <<i<< "j:" <<j<< endl;
    return 0;}
```

HATIJA:

i: 0 j: 0

i: 1 j: 1

i: 2 j: 2

for takrorlashida nol parametrlarni ishlatilishi. `for` takrorlashining ixtiyoriy parametri tushirib qoldirilishi mumkin. Nol parametrlar `for` takrorlashini boshqa parametrlaridan nuqtali vergul (;) bilan ajratiladi. Agarda `for` takrorlashini 1 – va 3 – parametrlarini tushirib qoldirsak, u xuddi *while* operatoridek qo'llaniladi.

10 – misol. `for` takrorlashining nol parametrlari.

```
# include < iostream>
```

```

using namespace std;
int main() {
int counter=0;
    for ( ; counter<5 ; )
    {
        counter++;
        cout << "Looping!";
    }
    cout<< "\n Counter:" << counter<< ".\n";
return 0;}

```

NATIJA:

Looping! Looping! Looping! Looping! Looping!

Counter: 5

TAHLIL

Bu takrorlashni bajarilishi 7 – misolda keltirilgan `while` takrorlashini bajarilishiga o'xshash tarzda amalga oshiriladi. 4-satrdagi `counter` o'zgaruvchisiga qiymat o'zlashtirilayapti. `for` takrorlashida esa parametr sifatida faqatgina takrorlashni davom etish shartini tekshirish ifodasi ishlatilgan. Takrorlash o'zgaruvchisi ustida operatsiya ham tushirib qoldirilgan. Bu holda ushbu takrorlashni quyidagicha ifodalash mumkin:

```
while(counter<5)
```

Qaralgan misolimiz yana bir marta C++ tilida bir xil masalani bir necha usul bilan hal qilish imkoniyati borligini ko'rsatadi. Bundan tashqari, `for` takrorlashining 3 ta parametrini ham tushirib qoldirish va takrorlashni `break` va `continue` operatorlarini qo'llash orqali boshqarish mumkin. `for` konstruktsiyasini parametrlarsiz qo'llanilishi 11 – misolda ko'rsatilgan.

11 – misol. `for` operatorlarini parametrlarsiz qo'llanilishi.

```

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

```

```

int counter=0;
int max;
cout<< "How many hellos?";
cin>> max;
for( ; ; )
{
    if (counter <max)
    {
        cout << "Hello! \h";
        counter++;
    }
    else
        break;
}
return 0; }

```

NATIJA:

How many hellos? 3

Hello!

Hello!

Hello!

for takrorlashining tanasi bo`sh bo`lgan holda qo`llanilishi. Takrorlashda *for* operatori orqali uning tanasiga hech qanday operator yozmasdan turib ham biror bir amalni bajarish mumkin. Bunda takrorlash tanasi bo`sh satrdan iborat bo`ladi. Tanasi bo`sh bo`lgan *for* takrorlashi 12 – misolda keltirilgan.

12 – misol. *For* takrorlashining tanasi bo`sh bo`lgan holda qo`llanilishi.

```

# include< iostream>
using namespace std;
int main()
{
    for (int i=0; i<5; cout<< "i" <<i++ <<endl);
}

```

```
    return 0;
}
```

HATIJA:

i:0

i:1

i:2

i:3

i:4

Ichki takrorlanish. Boshqa takrorlashning ichida tashkil etilgan takrorlash ichki takrorlash deb aytiladi. Bu holda ichki takrorlash tashqi takrorlashni har bir iteratsiyasida to'liq bajariladi. 13 – misolda matritsa elementlarini ichki takrorlash orqali to'ldirilishi namoyish qilingan.

13 – misol. Ichki takrorlanish.

```
# include< iostream>
using namespace std;
int main()    {
    int rows, columns;
    char theChar;
    cout << "How many rovs?";
    cin >> rows;
    cout << "How many columns?";
    cin >> columns;
    cout << "What character?";
    cin>>theChar;
    for ( int i=0;  i<rows;  i++)
    {
        for (int j=0; j<columns; j++)
            cout << theChar;
        cout<< "\n";
    }
}
```

```

    }
    return 0;
}

```

HATIJA:

How many rows? 4

How many columns? 12

What character? x

```
x x x x x x x x x x x x
```

```
x x x x x x x x x x x x
```

```
x x x x x x x x x x x x
```

```
x x x x x x x x x x x x
```

*for takrorlashi schyotchigining ko`rinish sohasi.* ANSI ning yangi standarti bo`yicha takrorlashda e`lon qilingan o`zgaruvchining ko`rinish sohasi faqat takrorlash ichidangina iborat. Lekin ko`pgina kompilyatorlar eski standartlarni ham qo`llab – quvvatlaydilar. Quyida keltirilgan dastur kodini kiritib o`zingizning kompilyatoringiz yangi standartga mos kelish- kelmasligini tekshirishingiz mumkin.

14 – misol.

```

#include <iostream>
using namespace std;
int main()      {
    for ( int i = 0; i<5; i++ )
    {
        cout << " i: " << i << endl ;
    }

    i=7; // i ko`rinish sohasi chegarasidan tashqarida
    return 0; }

```

Agarda bu dastur xatoliksiz kompilyatsiya qilinsa demak u ANSI ning yangi standartini qo`llab - quvvatlamaydi. Yangi standartga muvofiq kompilyatorlar i=7

ifoda uchun xatolik haqida xabar berishi kerak. Dastur kodiga ozgina o`zgartirish kiritilganda so`ng dastur barcha kompilyatorlar uchun xatoliksiz ishlaydi.

15 – misol.

```
# include <iostream>
using namespace std;
int main()      {
    int i;
    for ( int i = 0; i<5; i++ )
        {
            cout << " i: " << i << endl ;
        }
    i=7;    //Endi i barcha kompilyatorlar tomonidan
            //xatoliksiz qabul qilinishi mumkin.
    return 0;
}
```

Funksiyalar

Ko'p hollarda dasturning biror–bir qismini turli xil qiymatlarda ko'p marta takroran bajarishga to'g'ri keladi. Bunday hollarda dasturning ushbu qismi alohida bir qism sifatida ajratib qo'yiladi. Qachonki, ushbu qismga tegishli hisoblashlarni bajarishga to'g'ri kelsa, bu qismga kerakli qiymatlarni bergan holda murojaat tashkil qilinadi. Agar bu qism natija qaytarsa, u funksiya, aks holda prosedura deyiladi.

Funksiyaga bir necha qiymat uzatish mumkin, biroq u bitta qiymat qaytariladi. Funksiya ichida dasturlashning barcha qoidalari bajariladi.

Funksiyalar dasturchi ishini juda yengillashtiradi. Funksiyalar yordamida dastur modullashadi, qismlarga bo'linadi. Bu esa keyinchalik dasturni rivojlantirishni osonlashtiradi. Bunda dasturchi yozgan funksiyalar C++ ning standart kutubxonasi va boshqa kutubxonalar ichidagi funksiyalar bilan birlashtiriladi. Bu esa ishni osonlashtiradi. Ko'p holda dasturda takroran

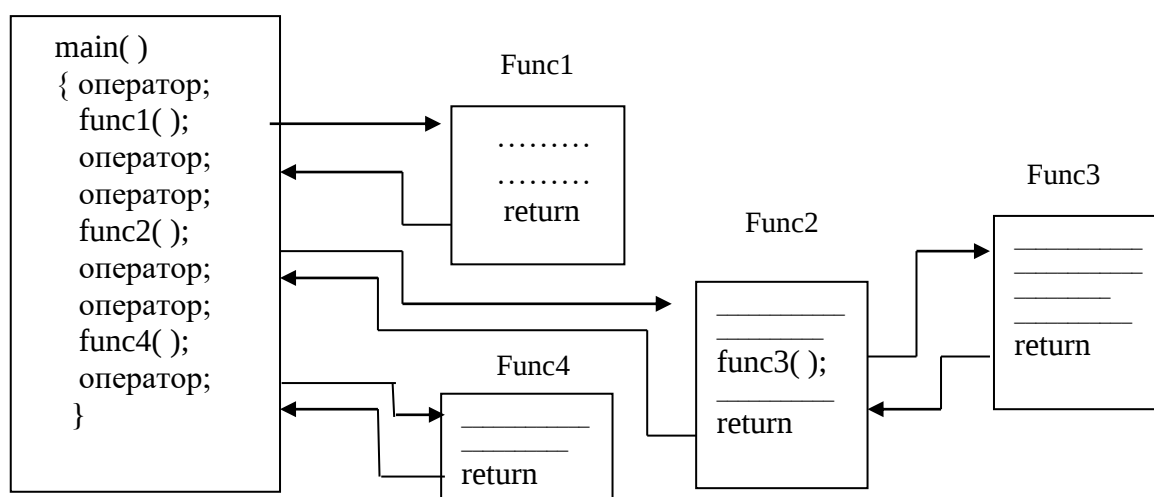
bajariladigan amalni funksiya sifatida yozish va kerakli joyda ushbu funktsiyani chaqirish mumkin. Dastur yozilish davrida xatolarni topishni yengillashtiradi.

C++ tilida asosiy funksiya *main()* hisoblanadi. Dasturda ushbu funksiya albatta bo'lishi shart, dasturning bajarilishi shu funksiya bilan boshlanadi, boshqa barcha funksiyalar ana shu funksiya orqali chaqiriladi. Dasturning qismida qaysi funksiya nomi yozilgan bo'lsa, boshqaruv shu funksiya tanasiga o'tadi. Bu jarayon funktsiyani chaqirilishi (yoki funktsiyaga murojaat qilish) deb aytiladi. Funktsiya ishini tugatgandan so'ng dastur o'z ishini funksiya chaqirilgan qatorning keyingisidan boshlab davom ettiradi, ya'ni boshqarish murojaat qilingan dastur qismiga qaytariladi.

Funktsiya dasturining nom berilgan shunday qismiki, unga dasturning boshqa funktsiyalarning tuzilish usuliga to'xtalib o'tishdan oldin funktsiyaga har bir bo'limdan qancha murojat qilish talab qilinsa, shuncha murojat qilish mumkin.

Funktsiya bu ma'nosiga ko'ra dastur osti bo'lib, u ma'lumotlarni o'zgartirishi va biror bir qiymat qaytarishi mumkin. C++ da har bir dastur hech bo'lmaganda bitta *main()* funktsiyasiga ega bo'ladi. *main()* funktsiyasi dastur ishga tushirilishi bilan operatsion sistema tomonidan avtomatik chaqiriladi. Boshqa funktsiyalar esa u tomonidan chaqirilishi mumkin.

Har bir funksiya o'zining nomiga egadir. Qachonki, dasturda bu nom uchrasa boshqaruv shu funksiya tanasiga o'tadi. Bu jarayon funktsiyani chaqirilishi (yoki funktsiyaga murojaat qilish) deb aytiladi. Funktsiya ishini tugatgandan so'ng dastur o'z ishini funksiya chaqirilgan qatorning keyingisidan boshlab davom ettiradi. Dastur bajarilishining bunday sxemasi yuqoridagi rasmda ko'rsatilgan.



Funksiyaning chaqirilishi tartibi

Standart va foydalanuvchi funksiyalari. C++ tili foydalanuvchilarga juda ko'p funksiyalarni taqdim qiladi. Ular kodi avvaldan yozib qo'yilgan. Foydalanuvchi kerakli funksiya nomini va parametrlari qiymatini berishi kifoya qiladi. Ushbu funksiyalar standart funksiyalar deyiladi. Ular ma'lum bir guruhlar shaklida jamlangan.

Standart matematik funksiyalar. Standart matematik funksiyalari ko'pgina amallarni bajarishga imkon beradi. Masalan, bizning dasturimizda quyidagi satr bor bo'lsin:

```
double = k;  
int m = 123;  
k = sin(m);
```

Kompilyator ushbu satrni ko'rganida, standart kutubxonadan `sin` funksiyasini chaqiradi. Kirish qiymati sifatida `m` ni berdik. Javob, ya'ni funksiya qaytgan qiymat `k` ga berildi. Funksiya argumentlari o'zgarmas sonlar (`const`), o'zgaruvchilar, ifodalar va boshqa mos keluvchi qiymat qaytaradigan funksiyalar bo'lishi mumkin. Masalan:

```
int g = 49, k = 100;  
cout << "4900 ning ildizi -> " << sqrt( g * k );
```

Ekranda:

4900 ning ildizi -> 70.

Matematik funksiyalar aksariyat hollarda `double` tipidagi qiymat qaytarishadi.

Kiruvchi argumentning tipi sifatida esa `double` ga keltirilishi mumkin bo'lgan tip beriladi. Bu funksiyalarni ishlatish uchun `math.h` (yangi ko'rinishda `cmath`) e'lon faylini `include` bilan asosiy dastur tanasiga kiritish kerak.

Quyida matematik funksiyalar kutubxonasining ba'zi birlari keltiriladi. `x` va `y` o'zgaruvchilari `double` tipiga ega.

Matematik funksiyalar

Funksiya	Aniqlanishi	Misol
ceil(x)	x ni x dan katta yoki unga teng bo'lgan eng kichik butun songacha yaxlitlaydi.	ceil(12.6) = 13.0 ceil(-2.4) = -2.0
cos(x)	x ning trigonometrik kosinusi (x radianda).	cos(0.0) = 1.0
exp(x)	e ning x - darajasi (eskponetsial funksiya).	exp(1.0) = 2.71828 exp(2.0) = 7.38906
fabs(x)	x ning absolyut qiymati.	$x > 0 \Rightarrow \text{abs}(x) = x$ $x = 0 \Rightarrow \text{abs}(x) = 0.0$ $x < 0 \Rightarrow \text{abs}(x) = -x$
floor(x)	x ni x dan kichik bo'lgan eng katta butun songacha yaxlitlaydi.	floor(4.8) = 4.0 floor(-15.9) = -16.0
fmod(x,y)	x/y ning qoldig'ini kasr son tipida beradi.	fmod(7.3,1.7) = 0.5
log(x)	x ning natural logarifmi (e asosiga ko'ra).	log(2.718282) = 1.0
log10(x)	x ning 10 asosiga ko'ra logarifmi.	log10(1000.0) = 3.0
pow(x,y)	x ning y chi darajasini beradi.	pow(3,4) = 81.0 pow(16,0.25) = 2
sin(x)	x ning trigonometrik sinusi (x radianda).	sin(0.0) = 0.0
sqrt(x)	x ning kvadrat ildizi.	sqrt(625.0) = 25.0
tan(x)	x ning trigonometrik tangensi (x radianda).	tan(0.0) = 0

Har bir funksiya – funksiyalar shabloni yoki funksiyalar shabloni to'plami yordamida ifodalanadi. Shunday qilib, funksiya har xil tipdagi qiymatlarga ega bo'lgan har xil konteynerlar bilan ishlay oladi. Barcha funksiyalarni argumentlari (begin, end) yarim oraliqlar bo'ladi.

Parametrlarni funksiyalarga uzatish.

Qaytariladigan qiymatlar, parametrlar va argumentlar. Funksiya biror bir qiymat qaytarishi mumkin. Funksiyaga murojaat qilingandan so'ng u qandaydir amallarni bajaradi, keyin esa u o'z ishining natijasi sifatida biror bir qiymat qaytaradi. Bu qaytariladigan qiymat deb ataladi va bu qiymatning tipi oldindan

e`lon qilinishi lozim. Quyidagi yozuvda `myFunction` funksiyasi butun sonli qiymat qaytaradi.

```
int myFunction()
```

Funksiyaga ham o`z navbatida biror bir qiymat uzatish mumkin. Uzatiladigan qiymatlar funksiyaning parametrlari deb aytiladi.

```
int myFunction (int Par, float ParFloat);
```

Bu funksiya nafaqat butun son qaytaradi, balki parametr sifatida butun va haqiqiy sonli qiymatlarni qabul qiladi.

Parametrda funksiya chaqirilganda unga uzatiladigan qiymat tipi aniqlanishi lozim. Funksiyaga uzatiladigan haqiqiy qiymatlar argumentlar deb aytiladi.

```
int theValueReturned=myFunction(5,6,7);
```

Bu yerda `theValueReturned` nomli butun sonli o`zgaruvchiga argument sifatida 5, 6 va 7 qiymatlar berilgan `myFunction` funksiyasining qaytaradigan qiymati o`zlashtirilayapti. Argument tiplari e`lon qilingan parametr tiplari bilan mos kelishi lozim.

**Funksiyani e`lon qilish va aniqlash.** Dasturda funksiyaning qo`llash uchun, oldin uni e`lon qilish, keyin esa aniqlash lozim. Funksiyani e`lon qilishda kompilyatorga uning nomi, qaytaradigan qiymatlari va parametrlari haqida xabar beriladi. Funksiyani aniqlanishidan kompilyator uning qanday ishlashi haqida ma`lumot oladi. Dasturdagi biror funksiyaning oldindan e`lon qilmasdan turib chaqirish mumkin emas. Funksiyani e`lon qilinishi uning prototipini (timsolini) hosil qilish deb ataladi.

**Funksiyani e`lon qilish.** Funksiyani e`lon qilishning uch xil usuli mavjud:

- Funksiya prototipi faylga yoziladi, keyin esa u `#include` ifodasi qo`llanilib kerakli dasturga qo`shib qo`yiladi.
- Funksiya ishlatiladigan faylga uning prototiplari yoziladi.
- Funksiya uni chaqiruvchi ixtiyoriy funksiyadan oldin yoziladi va bu holda funksiya e`lon qilinishi bilan bir vaqtda aniqlanadi.

Funksiyani prototipini tuzmasdan turib ham uni ishlatishdan oldin e`lon qilish mumkin. Lekin, dasturlashning bunday uslubi quyidagi uchta sababga ko`ra yaxshi hisoblanmaydi.

Birinchidan, funksiyani faylda ko`rsatilgan tartibda yozish, uni dastur ishlatilishida o`zgartirish jarayonini murakkablashtiradi.

Ikkinchidan, quyidagi ko`p uchraydigan holatni amalga oshirish imkoniyati mavjud emas.

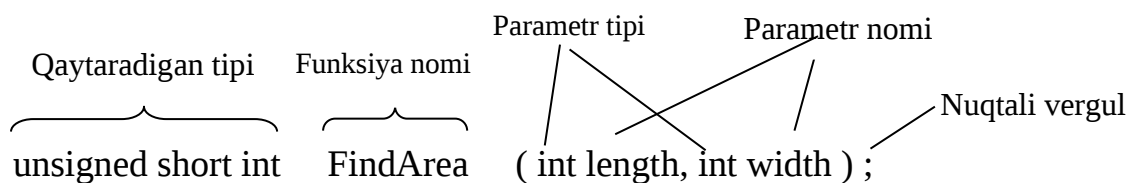
A () funksiya B () funksiyani chaqirsin. Xuddi shuningdek, dasturning biror bir qismida B () funksiya A () funksiyani chaqirsin. U holda A () funksiyani B () funksiya aniqlanmasdan turib ishlatib bo`lmaydi.

Bu holda hech bo`lmaganda bitta funksiya oldindan e`lon qilinishi lozim.

Uchinchidan, funksiyaning prototiplari dasturni tekshirish jarayonida juda yaxshi ishlatiladi. Agarda funksiya prototipi aniqlangan bo`lsa unga muvofiq funksiya aniqlangan parametrini qabul qiladi yoki aniqlangan biror bir qiymat qaytaradi. Dasturda e`lon qilingan prototipga muvofiq bo`lmagan funksiyani ishlatishga urinsak kompilyator bu xatolikni kompilyatsiya jarayonini o`zidayoq aniqlaydi va dastur ishlashida turli xatoliklarni ro`y berishining oldini oladi.

Funksiya prototiplari. Ko`pgina ichki qurilgan funksiyalarning prototiplari dasturga `#include` kalit so`zi yordamida qo`shiladigan fayl-sarlavhasida yoziladi. Foydalanuvchi tomonidan tuziladigan funksiyalar uchun esa ularning mos prototiplarini dasturga qo`shish dasturchi tomonidan bajarilishi lozim.

Funksiyaning prototipi nuqtali vergul orqali tugaydigan funksiyani qaytaradigan qiymati va signaturasidan iborat. Funksiyani signaturasi deb uning nomi va parametrlar ro`yxati tushiniladi.



Funksiya prototipining tarkibiy qismlari

Formal parametrlar ro'yxati barcha parametrlar va ularning tiplarini ifodalaydi. Funksiya prototipining tarkibiy qismlari yuqoridagi rasmda ko'rsatilgan.

Funksiyaning prototipi hamda aniqlanishidagi uning qaytaradigan qiymati tipi va signaturasi mos bo'lishi lozim. Agarda bunday mutanosiblik bo'lmasa kompilyator xatolik haqida xabar beradi. Funksiya prototipida parametr nomlarisiz tiplarni ko'rsatilishi yetarlidir. Masalan, quyida keltirilgan misol to'g'ri:

```
long Area(int, int)
```

Bu prototip ikkita butun sonli parametrni qabul qilib, `long` tipidagi qiymat qaytaradigan `Area()` nomli funktsiyani e'lon qiladi. Prototipning bunday yozilishi unchalik yaxshi variant emas. Prototipga parametrlarning nomlarini qo'shilishi uni tushunarliroq bo'lishini ta'minlaydi.

Har bir funktsiyaning qaytaradigan qiymati tipi aniqlangan bo'ladi. Agarda u ochiq aniqlanmagan bo'lsa avtomatik ravishda `int` tipini qabul qiladi.

16–misol. Funktsiyani e'lon qilinishi, aniqlanishi va ishlatilishi.

// 16 – Misol. Funksiya prototipini qo'llanilishi

```
# include <iostream>
using namespace std;
//funksiya prototipi
int Yuza(int uzunlik, int kenglik);
int main()
{
    int YerUzunligi, int YerKengligi, int YerMaydoni;
    cout<< "\n Yerning uzunligi necha metr?\n";
    cin >> YerUzunligi;
    cout<< "\n Yerning kengligi necha metr?";
    cin >> YerKengligi;
    YerMaydoni=Yuza(YerUzunligi,YerKengligi);
    cout << "\n Yer maydoni yuzasi ";
    cout >>YerMaydoni;
```

```

    cout << "kvadrat metr\ n \ n";
    return 0;
}

int Yuza(int YerUzunligi, int YerKengligi)
{
    return YerUzunligi* YerKengligi
}

```

NATIJA

- Yerning uzunligi necha metr? 200
- Yerning kengligi necha metr? 100
- Yer maydoni yuzasi 20000 kvadrat metr

Funksiyaning aniqlanishi. Funksiyaning aniqlanishi ikki qismdan – uning sarlavhasi va tanasidan iboratdir. Funksiyaning sarlavhasi uning prototipiga o`xshash aniqlanadi, faqatgina bu holda parametrlar nomlangan bo`lishi shart va sarlavha oxirida nuqtali vergul qo`yilmaydi. Funksiya tanasi figurali qavsga olingan ifodalar to`plamidan iborat. Funksiyaning sarlavhasi va tanasi quyida ko`rsatilgan.

```

Yuza(int uzunlik,int kenglik)
{ - ochiluvchi figurali qavs.
// funksiya tanasi
    return (uzunlik*kenglik).
} - yopiluvchi figurali kavs.

```

Funksiyaning bajarilishi. Funksiya chaqirilganda unda ko`rsatilgan amallar ochiluvchi figurali qavsdan ({}) keyingi birinchi ifodadan boshlab bajariladi. Funksiya tanasida `if` shartli operatoridan foydalanib tarmoqlanishni ham amalga oshirish mumkin.

Funksiya o`z tanasida boshqa funksiyalarni va hatto o`z – o`zini ham chaqirishi mumkin.

Lokal, statik, dinamik o'zgaruvchilar.

Lokal o'zgaruvchilar. Funksiyaga qiymatlar uzatish bilan birga uning tanasida o'zgaruvchilarni e'lon qilish ham mumkin. Bu lokal o'zgaruvchilar orqali amalga oshiriladi. Qachonki dasturni bajarilishi funksiyadan asosiy qismga qaytsa, bu funksiyadagi lokal o'zgaruvchilar xotiradan o'chiriladi.

Lokal o'zgaruvchilar xuddi boshqa o'zgaruvchilar kabi aniqlanadi. Funksiyaga beriladigan parametrlarni ham lokal o'zgaruvchilar deb atash mumkin va ularni funksiya tanasida aniqlangan o'zgaruvchilar kabi ishlatish mumkin. Navbatdagi misolda funksiya parametrlari va funksiya ichida aniqlangan lokal o'zgaruvchilarni qo'llashga oid misol keltirilgan.

17 – misol. Funksiya lokal o'zgaruvchilari va parametrlarining qo'llanilishi.

```
# include <iostream>
using namespace std;
float Almashtirish(float);
int main()
{
    float TempFer;
    float TempCel;
    cout << "Farengeyt bo'yicha temperaturani
    << "kiriting:";
    cin >> TempFer;
    TempCel = Almashtirish(TempFer);
    cout << "\n Bu temperatura selziy shkalasi
    << "bo'yicha: ";
    cout << TempCel << endl;
    return 0 ; }
float Almashtirish(float TempFer)
{
    float TempCel;
    tempCel= ((TempFer-32) *5) /9;
```

```
return TempCel;
}
```

NATIJA:

- Farengeyt bo`yicha temperaturani kiriting: 50
- Bu temperatura selziy shkalasi bo`yicha: 10

**Global o`zgaruvchilar.** `main()` funksiyasida aniqlangan o`zgaruvchilar dasturdagi barcha funksiyalar uchun murojaat qilishga imkonli va ko`rinish sohasiga ega hisoblanadi. Bunday o`zgaruvchilar dasturdagi funksiyalar uchun global o`zgaruvchilar deyiladi.

Global o`zgaruvchi nomi bilan funksiya ichida nomlari ustma-ust tushadigan lokal o`zgaruvchilar faqatgina joriy funksiyaning ichidagina global o`zgaruvchining qiymatini o`zgartiradi. Lekin global o`zgaruvchi funksiya o`z ishini tugatgach u chaqirilishidan oldingi qiymatini saqlab qoladi, ya`ni funksiya tanasida e`lon qilingan lokal o`zgaruvchi funksiyaning ichida global o`zgaruvchini yashiradi xolos. Bunda lokal o`zgaruvchi alohida hosil qilinadi va funksiya ishlash vaqtida global va lokal o`zgaruvchilarning nomlari bir xil bo`lsa faqatgina lokal o`zgaruvchi ustida amallar bajariladi. Global o`zgaruvchi esa funksiyaning bajarilishi davomida oldingi qiymatini saqlab turadi. Bu g`oya navbatdagi misolda namoyish qilingan.

18–misol. Global va lokal o`zgaruvchilarning qo`llanishi.

```
# include <iostream>
using namespace std;
void MeningFunksiyam() ; // prototip
int x = 5, y = 7; // global o`zgaruvchilar
int main()      {
    cout << "main()dagi x ning qiymati:"
    <<x<<"\n";
    cout<<"main()dagi y ning qiymati y:"
    <<y<<"\n";
    MeningFunksiyam();
```

```

    cout << "MeningFunksiyam() funksiyasi"<<
    "ishini tugatdi!\ n \ n";
    cout<<"main()dagi x ning qiymati:"
    <<x<<"\n";
    cout<<"main()dagi y ning qiymati:"
    <<y<<"\n";
return 0;      }
void MeningFunksiyam();
{ int y = 10;
    cout<<"MeningFunksiyam()dagi"<<
    << "x:" << x <<"\ n";
    cout<<"MeningFunksiyam()dagi"<<
    << "y:" << y <<"\ n";      }

```

HATIJA:

- main()dagi x ning qiymati: 5
- main()dagi y ning qiymati: 7
- MeningFunksiyam()dagi x: 5
- MeningFunksiyam()dagi y: 10
- MeningFunksiyam()funksiyasi ishini tugatdi!
- main()dagi x ning qiymati: 5
- main()dagi y ning qiymati: 7

Funksiyani ichida aniqlangan o`zgaruvchilar lokal ko`rinish sohasiga ega deyiladi. Yuqorida aytib o`tilganidek, bu o`zgaruvchilarni faqatgina funksiyaning ichidagina qo`llash mumkinligini anglatadi. C++ da o`zgaruvchilarni nafaqat dasturning boshida, balki ixtiyoriy joyda aniqlash mumkin. Agarda o`zgaruvchi funksiya tanasidagi biror bir blok ichida aniqlangan bo`lsa, bu o`zgaruvchi faqatgina shu blok ichidagina ta`sirga ega bo`lib butun funksiyaning ichida ko`rinish sohasiga ega bo`lmaydi.

19 – misol. Lokal o`zgaruvchini ko`rinish sohasi.

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

```

using namespace std;
void MeningFunksiyam();
int main() {
    int x=5;
    cout<<"\n\n main()dagi x ning qiymati:"
    <<x;
    MeningFunksiyam();
    cout<<"\n main()dagi x ning qiymati:"<< x;
    return 0; }
void MeningFunksiyam();
{int x = 8;
cout <<"\n\n MeningFunksiyam()dagi"
<<"lokal x ning qiymati: "<< x << endl;
{
    cout << "\ n\ MeningFunksiyam() "
    <<"funksiyasi blokidagi x ning qiymati"
    << " x:"<<x;
    int x = 9;
    cout<<"\n Blok ichida aniqlangan"
    << x ning qiymati:"<<x;
}
cout<<"\n    MeningFunksiyam()dagi"    <<    "blockdan
tashqarisida x ning qiymati:"
<<x<<endl;
}

```

HATIJA:

main() dagi x ning qiymati: 5

- MeningFunksiyam() dagi lokal x ning qiymati: 8
- MeningFunksiyam() funksiyasi blokidagi x ning qiymati: 8
- Blok ichida aniqlangan x ning qiymati: 9

- MeningFunksiyam() dagi blockdan tashqarisida x ning qiymati: 8
- main()dagi x ning qiymati: 5

Funksiyada ishlatiladigan operatorlar. Funksiyada ishlatiladigan operatorlar soni va tipiga hech qanday chegara yo`q. Funksiya ichida istalgancha boshqa funksiyalarni chaqirish mumkin bo`lsada, lekin funksiyalarni aniqlash mumkin emas.

C++ tilida funksiyaning hajmiga hech qanday talab qo`yilmasa ham, uni kichikroq tarzda tuzgan ma`quldir. Har bir funksiya tushunish oson bo`ladigan bitta masalani bajarishi lozim. Agarda funksiya hajmi kattalashayotganligini sezsangiz, bu funksiyaning bir nechta kichikroq funksiyalarga ajratish haqida o`ylashingiz kerak.

Funksiya argumentlari. Funksiyaning argumentlari turli tipda bo`lishi mumkin. Shuningdek, argument sifatida C++ tilidagi biror bir qiymat qaytaradigan o`zgarmlar, matematik va mantiqiy ifodalar va boshqa ixtiyoriy funksiyalarni berish mumkin.

Misol sifatida bizda biror bir qiymat qaytaruvchi `double()`, `triple()`, `square()` va `cube()` funksiyalari berilgan bo`lsin. Biz quyidagi ifodani yozishimiz mumkin:

```
Answer=(double(triple(square(my Value))))
```

Bu ifodada `myValue` o`zgaruvchisini qabul qilinadi va u `cube()` funksiyasiga argument sifatida uzatiladi.

`cube()` funksiyasi qaytargan qiymat esa `square()` funksiyasiga argument sifatida uzatiladi. `square()` funksiyasi qiymat qaytargandan keyin, buning qiymati o`z navbatida `triple()` funksiyasiga argument sifatida beriladi. `triple()` funksiyasining qiymati esa `double()` funksiyasiga argument qilib beriladi va u `Answer` o`zgaruvchisiga o`zlashtiriladi.

Yangi stildagi e'lon fayllari va nomlar sohasi tushunchasi. C++ tilining standarti `.h` bilan tugaydigan (`stdio.h` ...) standart kutubxona e'lon fayllarini

yangidan nomlab chiqdi. Bunda .h qo'shimchasi olib tashlandi. C tilidan qolgan fayllar nomiga esa c harfi qo'shildi.

Misol uchun:

```
iostream.h -> iostream
string.h -> cstring
stdlib.h -> cstdlib
time.h -> ctime
```

C tilidan meros qolgan kutubxona 18 ta e'lon fayli orqali berilgan. C++ tiliga tegishli standart kutubxonada esa 32 ta e'lon fayli bor. Fayllarni yangidan belgilashdan maqsad kutubxonadagi funksiya va ob'ektlarni `std` deb ataluvchi nomlar sohasiga (`namespace`) kiritish hisoblanadi.

Nomlar sohasining o'zi ham nisbatan yangi tushuncha. Nomlar sohasini alohida dastur qismlari deb faraz qilsak bo'ladi. Boshqa-boshqa sohalarda ayni nomli funksiya, o'zgaruvchi nomlari va ob'ektlar berilishi mumkin. Bunda hech qanday nomlar to'qnashuvi sodir bo'lmaydi. C++ tilida to'liq aniqlangan nom (fully qualified name) degan tushuncha bor. Shunga ko'ra har bir cout ob'ektinig to'liq nomi quyidagicha bo'ladi:

Nomlar sohasi ob'ekt

```
global          ::cout
std              std::cout
fun::obj         fun::obj::cout
```

`::` operatori sohalarni bog'lash uchun qo'llaniladi. `fun::obj` nomli nomlar sohasida `obj` `fun` ichida joylashgan nom sohasidir. Global nomlar sohasida aniqlangan funksiya va boshqa turdagi dastur birliklariga dasturning istalgan yeridan yetishsa bo'ladi. Masalan, global nomlar sohasida e'lon qilingan `int` tipidagi `k` o'zgaruvchi bol'sa, uning ustidan dasturning xohlagan blokida amal bajarish mumkin. Nomlar sohasi mexanizmi dasturchilarga yangi kutubxonalarni yozish ishini ancha osonlashtiradi. Chunki, yangi kutubxonada ayni nomlar qo'llanishiga qaramay, nomlar konflikti yuz bermaydi. Dastur yoki kutubxona yozganda yangi nomlar sohasini belgilash uchun

```
namespace istalgan_nom {  
    ...  
    foo();    int k;    String str;  
    ...  
}
```

deb yozish mumkin. Dasturda ushbu nomlar sohasida aniqlangan o'zgaruvchilarni ishlatish uchun ularning to'liq nomini yozish kerak. Masalan:

```
istalgan_nom::foo() .
```

Ammo bu usul ancha mashaqqatli bo'ladi. Har bir funksiya yoki o'zgaruvchi oldiga uning to'liq aniqlangan nomini yozish ko'p vaqt oladi. Buning o'rniga **using namespace** *istalgan_nom* deb yozish kifoya qiladi. *using* (ishlatish, qo'llash) buyrug'i nomlar sohasini dastur ichiga tanishtiradi. C++ ning standart kutubxonasida aniqlangan ifodalarni qo'llash uchun *using namespace std* yoziladi. Bunda qo'llanilayotgan e'lon fayllari yangi tipda bo'lishi kerak.

20 – misol.

//Yangi tipdagi e'lon fayllari va nomlar sohasini qo'llash.

```
#include <iostream>  
  
using namespace std;  
  
int main()    {  
    std::cout << "Hello!\n";  
    cout << "Salom!";  
    return (0); }  

```

Ekranda:

Hello!

Salom!

`std::cout << "Hello\n";` satrida chiqish oqimi ob'ekti `cout` ning to'liq nomi qo'llanildi. Keyingi satrda esa yana ayni ob'ekt ishlatildi, lekin endi uni to'liq yozilmadi, chunki `std` nomlar sohasini `using` bilan e'lon qilib bo'lingan. Nomlarni global nomlar sohasida e'lon qilish uchun ularni blok va funksiyalar tashqarisida aniqlash kerak. Masalan:

```
# include <stdio>
```

```
int i;
```

```
int main() {
```

```
...      int k;      ...
```

```
return 0;    }
```

Bu yerda `i` global nomlar sohasida joylashgan, `k` esa `main()` funksiyasiga tegishli. O'zgaruvchilarni global nom sohasida aniqlashning boshqa yo'li, ularni nomsiz nomlar sohasida belgilash kerak. Yani:

```
namespace { int j; }
```

j o'zgaruvchisi global boldi. Uni ishlatish uchun:

```
::j = ::j + 7;
```

:: operatorini qo'llash mumkin, yoki oddiygina qilib faqat o'zini:

```
j = j * 9;
```

kabi yozish mumkin.

Nazorat savollari:

1. `for` takrorlashida bir nechta schyotchik ishlatish mumkinmi?
2. Tanasida biror amal yozilmagan `for` operatori yordamida takrorlash tashkil etish mumkinmi?
3. `for` takrorlashi ichida `while` takrorlashini tashkil etish mumkinmi?
4. Hech qachon tugallanmaydigan takrorlash tashkil etish mumkinmi?
5. `while` va `do .. while` operatorlarini qanday farqlari bor?
6. Nima uchun funksiyaga argument sifatida uzatilgan o'zgaruvchilar qiymati funksiya tanasida o'zgartirilsa dasturning asosiy kodidagi shu o'zgaruvchi qiymatiga akslantirilmaydi.
7. Funksiya prototipini e'lon qilish va funksiyaning aniqlash o'rtasida qanday farq bor?
8. Funksiya prototipini ko'rsatishda, aniqlashda va chaqirishda uning parameterlari nomlari ustma – ust tushishi kerakmi?
9. Agarda funksiya hech qanday qiymat qaytarmasa uni qanday e'lon qilish kerak?
10. Lokal o'zgaruvchi nima?
11. Nima uchun barcha o'zgaruvchilarni global deb e'lon qilish maqsadga muvofiq emas.
12. Global o'zgaruvchi nima?

MASSIVLAR BILAN ISHLASH. BIR VA IKKI O'LCHAMLI MASSIVLAR.

Bir o'lchovli massivlar.

Massiv bu bir tipli nomerlangan ma'lumotlar jamlanmasidir. Massiv indeksli o'zgaruvchi tushunchasiga mos keladi. Massiv ta'riflanganda tipi, nomi va indekslar chegarasi ko'rsatiladi. Misol uchun `long int a[5]; char w[200]; double f[4][5][7]; char[7][200]`. Massivda indekslar har doim 0 dan boshlanadi. C++ tili standarti bo'yicha indekslar soni 31 tagacha bo'lishi mumkin, lekin amalda bir o'lchovli va ikki o'lchovli massivlar qo'llaniladi. Bir ulchovli massivlarga matematikada vektor tushunchasi mos keladi. Massivning `int z[3]` shakldagi ta'rifi, `int` tipiga tegishli `z[0]`, `z[1]`, `z[2]` elementlardan iborat massivni aniqlaydi.

Massivlar ta'riflanganda initsializatsiya qilinishi, ya'ni boshlang'ich qiymatlarlari ko'rsatilishi mumkin. Misol uchun:

```
float C[]={1,-1,2,10,-12.5};
```

Bu misolda massiv chegarasi avtomatik aniqlanadi. Agar massiv initsializatsiya qilinganda elementlar chegarasi ko'rsatilgan bo'lsa, ruyxatdagi elementlar soni bu chegaradan kam bo'lishi mumkin, lekin ortiq bo'lishi mumkin emas. Misol uchun `int A[5]={2,-2}`. Bu holda `a[0]` va `a[1]` qiymatlari aniqlangan bo'lib, mos holda 2 va -2 ga teng.

Massivda musbat elementlar soni va summasini hisoblash.

```
#include <iostream.h>;
#include <conio.h>;
Main() {
    Int x[]={-1;2;5;-4;8;9};
    Clrscr();
    For (int s=0,int k=0, int I=0; I<6; I++) {
        If (x[I]<=0) continue;
        k++;s++;
    };
}
```

```

    Cout<<("%d",k);
    Cout<<("%d",k);
    getch();
};

```

Massivning eng katta, eng kichik elementi va o'rta qiymatini aniqlash:

```

#include <iostream.h>
Void main()
{
    Int I,j,n;
    Float a,b,d,x[100];
    While(1)
    {
        Cout<<("\n n="); Cin>>("%i",&n);
        If ( n>0 && n <= 100 ) break;
        Cout<<("\n Xato! 0<n<101 shartga qarang!");
    }
    Cout<<("\n elementlar qiymatlarini kiriting:\n");
    For (i=0;i<n;i++)
        { Cout<<("x[%i]=",i);Cin>>("%f",&x[i]); }
    max=x[0];min=x[0];
    For (s=0,i=0;i<n;i++)
        { s++;
          If (max<x[i]) max=x[i];
          If (min>x[i]) min=x[i];
        };
    s/=n;
    Cout<<("\n max=%f",max);
    Cout<<("\n min=%f",min);
    Cout<<("\n urta qiymat=%f",s);
}

```

Jadvallar.

Ikki o'lchovli massivlar matematikada matritsa yoki jadval tushunchasiga mos keladi. Jadvallarning initsializatsiya qilish qoidasi, ikki o'lchovli massivning elementlari massivlardan iborat bo'lgan bir o'lchovli massiv ta'rifiga asoslangandir. Misol uchun ikki qator va uch ustundan iborat bo'lgan haqiqiy tipga tegishli d massiv boshlang'ich qiymatlari qo'yidagicha ko'rsatilishi mumkin:

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

Bu yozuv quyidagi qiymat berish operatorlariga mosdir:

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

Bu qiymatlarni bitta ro'yxat bilan hosil qilish mumkin:

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

Initsializatsiya 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 qo'yidagi 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;
```

Initsializatsiya 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.

Qo'yidagi ko'radigan misolimizda jadval kiritilib har bir qatorning maksimal elementi aniqlanadi va bu elementlar orasida eng kichigi aniqlanadi:

```
#include <iostream.h>  
  
void main()  
{ double a[4,3]; double s, max=0.0, min=0.0;  
  int i, j;
```

```

for(i=0; i<4; i++)
{
    for(j=0; j<3; j++)
    { Cout<<(" a[%d][%d]=" , i, j); Cin>>("%f", s);
      a[i, j]=s;
      if (max<s) max=s;
    };
    Cout<<("\n");
    if (max<min) min=max;
}
Cout<<("\n min=%f", min);
}

```

Simvolli massivlar.

C++ tilida satrlar simvolli massivlar sifatida ta'riflanadi. Simvolli massivlar qo'yidagicha tasvirlanishi mumkin: `Char pas[10];`

Simvolli massivlar qo'yidagicha initsializatsiya qilinadi:

`Char capital[] = "TASHKENT";` Bu holda avtomatik ravishda massiv elementlari soni aniqlanadi va massiv oxiriga satr ko'chirish '\n' simvoli qo'shiladi.

Yuqoridagi initsializatsiyani qo'yidagicha amalga oshirish mumkin:

```
Char capital[]={ 'T', 'A', 'S', 'H', 'K', 'E', 'N', 'T', '\n' };
```

Bu holda so'z oxirida '\n' simvoli aniq ko'rsatilishi shart.

Misol uchun palindrom masalasini ko'rib chiqamiz. Palindrom deb oldidan ham oxiridan ham bir xil o'qiladigan so'zlarga aytiladi. Misol uchun non. Dasturda kiritilgan so'z palindrom ekanligi aniqlanadi:

```

#include <iostream.h>
void main()
{
    gets(a);

```

```
for( int j=0, a[j]!='\0';j++);  
I=0;  
while(I<j) if (a[I++]!=a[j--]) break;  
if ((j-I)>1) Cout<<("Palindrom emas") else  
Cout<<("Palindrom");
```

Keyingi misolda kiritilgan so'zdan berilgan harf olib tashlash dasturi berilgan:

```
#include <iostream.h>  
void main()  
{  
    char s[];  
    int c;  
    gets(a);  
    int i, j;  
    for ( i = j = 0; s[i] != '\0'; i++)  
        if ( s[i] != c )  
            s[j++] = s[i];  
    s[j] = '\0';  
    puts(s);  
}
```

Har gal 's' dan farqli simvol uchraganda, u j pozitsiyaga yoziladi va faqat shundan so'ng j qiymati 1 ga oshadi. Bu qo'yidagi yozuvga ekvivalent:

```
if ( s[i] != c )  
s[j] = s[i];  
j++;
```

So'zlar massivlari.

C++ tilida so'zlar massivlari ikki o'lchovli simvolli massivlar sifatida ta'riflanadi. Misol uchun:

```
Char Name[4][5].
```

Bu ta'rif yordamida har biri 5 ta harfdan iborat bo'lgan 4 ta so'zli massiv kiritiladi.

So'zlar massivlari qo'yidagicha initsializatsiya qilinishi mumkin:

```
Char Name[3][8]={ "Anvar", "Mirkomil", "Yusuf"}.
```

Bu ta'rifda har bir so'z uchun xotiradan 8 bayt joy ajratiladi va har bir so'z oxiriga '\0' belgisi qo'yiladi.

So'zlar massivlari initsializatsiya qilinganda so'zlar soni ko'rsatilmaligi mumkin.

Bu holda so'zlar soni avtomatik aniqlanadi:

```
Char      comp[][9]={ "komp'yuter",      "printer",
"kartridj"}.
```

Quyidagi dasturda berilgan harf bilan boshlanuvchi so'zlar ruyxati bosib chiqariladi:

```
#include <iostream.h>
void main()
{ char a[10][10];
  char c;
    for (int i=0;i<10;i++)
      gets(a[i]);
    c=getchar();
    for (i=0;i<10;i++) if (a[i][0]==c) puts(a[i]);
}
```

Qo'yidagi dasturda fan nomi, talabalar ruyxati va ularning baholari kiritiladi.

Dastur bajarilganda ikki olgan talabalar ruyxati bosib chiqariladi:

```
#include <iostream.h>
void main()
{ char a[10][10];
  char s[10];
  int k[10];
    gets(s);
    for (int i=0;i<10;i++) gets(a[i]);
```

```
for (i=0;i<10;i++) {Cin>>("%d",k[i]);  
for (int i=0;i<10;i++) if (k[i]==2) puts(a[i]);  
}
```

Ko'rsatkichlar massivlari.

Ko'rsatkichlar massivlari qo'yidagicha ta'riflanadi:

```
<tip> *<nom>[<son>]
```

Misol uchun `int *pt[6]` ta'rif int tipidagi ob'ektlarga olti elementli massivni kiritadi. Ko'rsatkichlar massivlari satrlar massivlarini tasvirlash uchun qulaydir. Misol uchun familiyalar ruyxatini kiritish uchun ikki o'lchovli massivdan foydalanishi kerak.

```
char fam[][20]={ "Olimov", "Rahimov", "Ergashev" }
```

Xotirada 60 elementdan iborat bo'ladi, chunki har bir familiyagacha 0 lar bilan to'ldiriladi. Ko'rsatkichlar massivi yordamida bu massivni qo'yidagicha ta'riflash mumkin:

```
Char *pf[] = { "Olimov", "Rahimov", "Ergashev" }.
```

Bu holda ruyxat xotirada 23 elementdan iborat bo'ladi, chunki har bir familiya oxiriga 0 belgisi qo'yiladi.

Ko'rsatkichlar massivlari murakkab elementlarni sodda usulda tartiblashga imkon beradi. Quyidagi misolda matritsa satrlari birinchi elementlari o'sishi tartibida chiqariladi. Bu misolda yordamchi ko'rsatkichlar massivi yaratilib, shu massiv tartiblanadi va massiv asosida matritsa elementlari chiqariladi.

```
# include <iostream.h>  
  
void main()  
{int n=2;  
int m=3;  
array[][3]={ (1,3,5) , (3,1,4) , (5,7,1) };  
int *pa[n];  
for (I=0;I<n;I++) pa[I]=(int *)&a[I];  
for (I=0;I<n-1;I++)
```

```

    {for (int k=I=1;k<n;k++)
    if a[I][1]>a[k][1]
    { int *pp=pa[I];
    pa[I]=pa[k];pa[k]=pp;};
    for (I=0;I<n;I++)
    {Cout<<("\n%I",I+1);
    for (int j=0;j<magistr;j++)
    Cout<<("%I",pa[I][j]);}
};

```

Ko'rsatkichlar massivlari funksiyalarda matritsalar qiymatlarini o'zgartirish uchun mumkin. Quyidagi misolda matritsani transponirlash funksiyasi ishlatiladi.

```

Void trans(int n,double *p[]);
{ double x;
  for (int I=0;I<n-1;I++)
  for (int j=I+1;j<n;j++)
  {x=p[I][j];
  p[I][j]=p[j][I];
  p[j][I]=x;}
};

void main()
{double a[3,3]={11,12,13,21,22,23,31,32,33};
  double ptr={(double*)&a[0], (double*)&a[1],
  (double*)&a[2]};
  int n=3;
  trans(n,ptr);
  for (int I=0;I<n;I++)
  { Cout<<("\n %i",i+1);
    for (int j=0;j<n;j++)
    Cout<<("\n %f",a[I][j]);};
};

```

Funksiyalar va massivlar.

Funksiyalar va sonli massivlar.

Funksiyalarda bir o'lchovli sonli massivlar argument sifatida ishlatilganda ularning chegarasini ko'rsatish shart emas. Misol tariqasida n o'lchovli vektorlar bilan bog'lik funksiyalarni ta'riflashni ko'rib chiqamiz.

Vektorning uzunligini aniqlash funksiyasi:

```
float mod_vec(int n,float x[])
{
    float a=0;
    for (int I=0;I<n;I++)
        a+=x[I]*x[I];
    return sqrt(double(a));
}
```

Ikki vektorning skalyar ko'paytmasi funksiyasi:

```
float scalar(int n,float x[],float y[])
{
    float a=0;
    for (int I=0;I<n;I++)
        a+=x[I]*y[I];
    return a;
}
```

Quyidagi dasturda ikki vektor orasidagi burchak kosinusini hisoblash funksiyasi kiritiladi va bu funksiya yordamida berilgan vektorlar orasidagi burchak kosinusi hisoblanadi:

```
#include <iostream.h>
#include <math.h>
float cosinus(int n,float x[],float y[])
{
    float a=0,b=0,c=0;
    for (int I=0;I<n;I++)
        {
            a+=x[I]*y[I];
            b+=x[I]*x[I];
            c+=y[I]*y[I];
        }
}
```

```

    }
    return a/sqrt(double(b*c));
}

void main()
{
    float e[]={1,2,3};
    float g[]={-1,7,4};
    Cout<<("\n%i",cosinus(3,e,g));
}

```

Funksiyalarda bir o'lchovli massivlar qaytariluvchi qiymatlar sifatida ham kelishi mumkin. Misol uchun ikki vektor summasini hisoblovchi funksiya protsedurani ko'ramiz:

```

Void sum_vec(int n, float a,float b, float c)
{
    for(int i=0;i<n;i++,c[i]=a[i]+b[i]);
}

```

Bu funksiya quyidagicha murojaat qilish mumkin:

```

Float  a[]={1,-1.5,-2},    b[]={-5.2,1.3,-4},    c[3];
sum_vec(3,a,b,c);

```

Massiv qiymat qaytaruvchi funksiya ta'rifini:

```

Float *sum_vec(int n,float a,float b)
{
    float d[n];
    for(int i=0;i<n;i++,d[i]=a[i]+b[i]);
    return d;
}

```

Bu funksiya qo'yidagicha murojaat qilish mumkin:

```

Float a[]={1,-1.5,-2},b[]={-5.2,1.3,-4};
float c[]=sum_vec(3,a,b);

```

Massivlarni tartiblash.

Keyingi misolimizda massivlarni qiymatlari o'sish tartibida joylashtirish funksiyasi berilgan:

```
Void function sort(int n, double a[])
{ int i,j; double c;
for(i=0;i<n;i++)
for(j=i+1;j<n;j++)
if (a[i]>a[j]) {c=a[i];a[i]=a[j];a[j]=c;}
```

Bu usulda har bir element $a[0]$ dan boshlab, keyingi hamma elementlar bilan solishtiriladi. Biror element ko'rilayotgan $a[i]$ dan kichik bo'lsa, bu elementlar o'rin almashtiriladi. Natijada $a[i]$ o'rniga i dan n gacha elementlar ichida eng kichigi qo'yiladi.

Qo'yidagi funksiya butun sonlar massivini Shell usuli asosida tartiblaydi. Bu usulda oldin bir biridan uzoqda joylashgan elementlar solishtiriladi. Elementlar orasidagi interval birgacha kamayib boradi.

```
shell(v, n) /* sort v[0]...v[n-1]
into increasing order */
int v[], n;
int gap, i, j, temp;
for (gap = n/2; gap > 0; gap /= 2)
for (i = gap; i < n; i++)
for (j=i-gap; j>=0 && v[j]>v[j+gap]; j-=gap)
temp = v[j];
v[j] = v[j+gap];
v[j+gap] = temp;
```

Bu dasturda uchta tsikl ishlatilgan. Eng tashqi tsikl elementlar orasidagi intervalni $N/2$ dan nol'gacha ikki martadan kamaytirib boradi. O'rta tsikl interval bilan ajratilgan elementlarni solishtiradi; eng ichki tsikl tartiblanmagan elementlar o'rnini almashtiradi. Interval oxiri birga teng bo'lgani uchun hamma elementlar to'g'ri tartiblanadi.

Keyingi misolda berilgan x qiymat tartiblangan massivda mavjudmi yoki yo'qligini aniqlovchi funksiyani ko'rib chiqamiz. Bu funksiya ikkiga bo'lish usulidan foydalanadi. Massiv elementlari o'sish tartibida joylashgan bo'lishi kerak.

```
binary(x, v, n) /* find x in v[0]...v[n-1] */
int x, v[], n;
int low, high, mid;
low = 0;
high = n - 1;
while (low <= high)
mid = (low + high) / 2;
if (x < v[mid])
high = mid - 1;
else if (x > v[mid])
low = mid + 1;
else /* found match */ return(mid);
return(-1);
```

MATEMATIK MODELLARNI ELEKTRON JADVALDAGI TADQIQOTI.

MS Excelda Solver imkoniyatlari.

Ishlab chiqarish ish jarayoni faoliyati natijalarini nazorat qilish va hisobga olish bilan bog'liq ko'plab turli masalalarni yechishda elektron jadvallardan keng foydalaniladi. Mutaxassis o'z ish faoliyati davomida ishlab chiqarish jarayonidagi juda ko'plab tahliliy masalalarni hal qilishga to'g'ri keladi va ular tahlili natijasida bir qancha yechimlar vujudga kelishi mumkin. Bu yechimlar orasidan eng yaxshisi va eng samaralisini tanlab olish maqsadga muvofiq hisoblanadi. Odatda, bunday yechim optimal deb ataladi. “Optimal” - lotincha “optimus” so'zidan olingan bo'lib, eng yaxshi (mavjud imkoniyatlar doirasida undan yaxshisi yo'q), eng ma'qul, eng samarali kabi ma'nolarni anglatadi. Optimal yechimni izlash masalasi keng ko'lamli bo'lib, matematikada avvalo ekstremal (maksimal yoki minimal) masalalar deb ataluvchi sohani qamrab oladi. Bunday masalalarning optimal yechimlarini aniqlashda Excel elektron jadvali qo'llanilib, undagi Solver – yechimlarni qidirish dasturlari paketidan foydalaniladi. Bu paket amallarni tadqiq qilish nazariyasining asosiy yo'nalishlaridan bo'lgan matematik dasturlash usullari va algoritmlari asosida yaratilgan.

Tizimni elektron jadvallar yordamida tadqiq qilish jarayonini kundalik amaliy faoliyatda jadvallar orqali bajariladigan hisob-kitoblarning tabiiy davomi deb qarash mumkin. Bu faoliyatga “Nima bo'ladi, agar,?”, “Bu qanday o'zgarishlarga olib keladi?” degan nuqtai nazardan yondoshish mumkin. Bu savollarning javoblari tabiiy resurslardan foydalaninshning optimal taktikasi va strategiyasi, “optimal nuqta”ga erishish va shu kabi masalalarning yechimlariga bog'liq holda topiladi. Javoblarni topish ixtiyoriy tizim tahlilining asosiy maqsadini tashkil etadi. Demak, elektron jadvallarda amalga oshiriladigan oddiy hisob kitoblar bilan bunday jadvallarda ishlab chiqarish jarayonining optimal yechimlarini qidirish tabiiy bog'langandir.

Excel dasturida ma'lumotlarni tahlil qilishning kuchli vositasi – Solver (yechimlarni qidirish) hisoblanadi.

Yechimlarni qidirish (Excel Solver – Поиск решения) dasturi Excel elektron jadvali dasturining qo`shimcha vositasi bo`lib, undan 1991 yildan buyon chiziqli tenglamalar sistemasini optimallashtirishning chiziqli va chiziqli bo`lmagan masalalarini yechishda foydalaniladi.

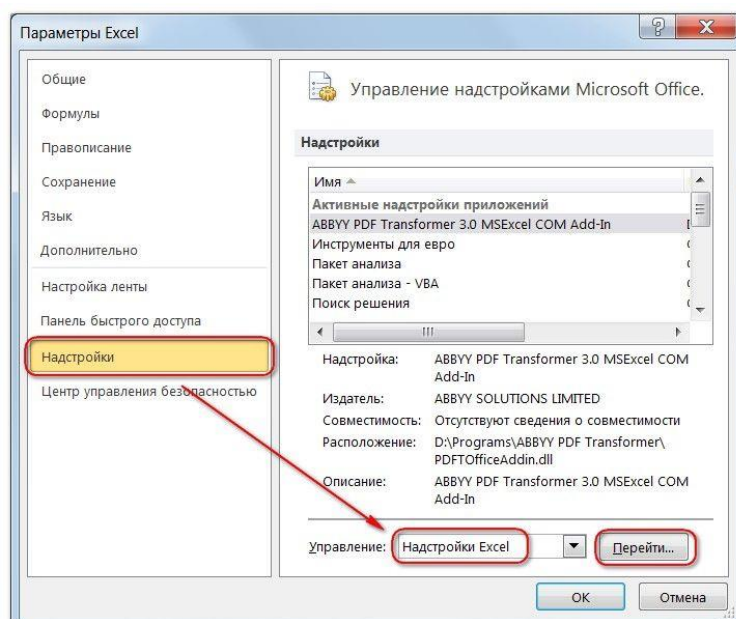
Dasturning bazaviy versiyasida yechiladigan masala o`lchamlariga quyidagi cheklovlar qo`yilgan:

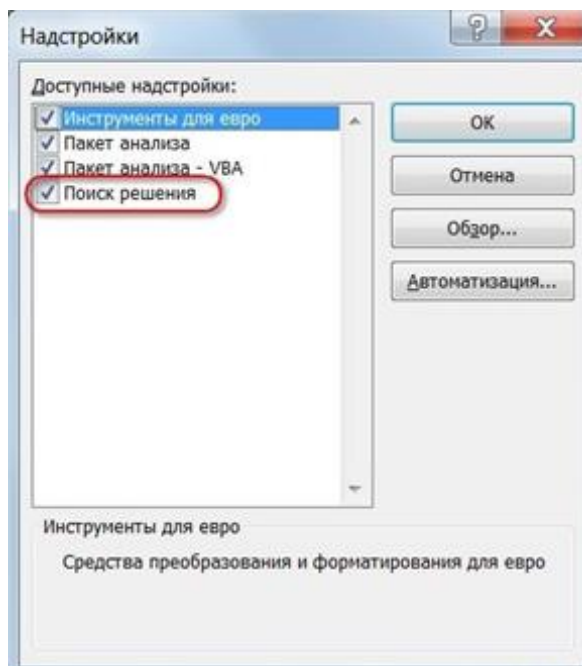
- noma`lumlar soni 200 tagacha.
- noma`lumlarga qo`yilgan cheklov formulalari soni 100 tagacha.
- noma`lumlarga qo`yilgan eng ko`p shartlarning soni 400 tagacha.

Excelga o`rnatilgan Solver dasturi optimallashtirishning kuchli va qulay usullarini yaratishga ixtisoslashgan Frontline System kompaniyasi tomonidan ishlab chiqilgan.

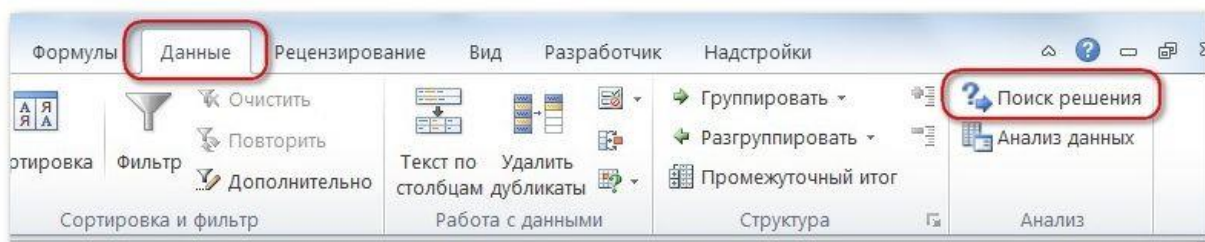
Odatda Excel dasturining “Поиск решения” (yechimni qidirish) vositasi faollashtirilmagan holatda bo`ladi. Uni faollashtirish uchun “Файл-Параметры” buyrug'i ishga tushirilsa, ochilgan oynaning chap qismida joylashgan “Настройка” bosiladi.

Ochilgan sohaning “Управление:” maydonidan “Настройка Excel” tanlanib, “Перейти” bosiladi. Ochilgan oynadan “Поиск решения” tanlangach, unga belgi qo`yiladi va OK bosiladi.



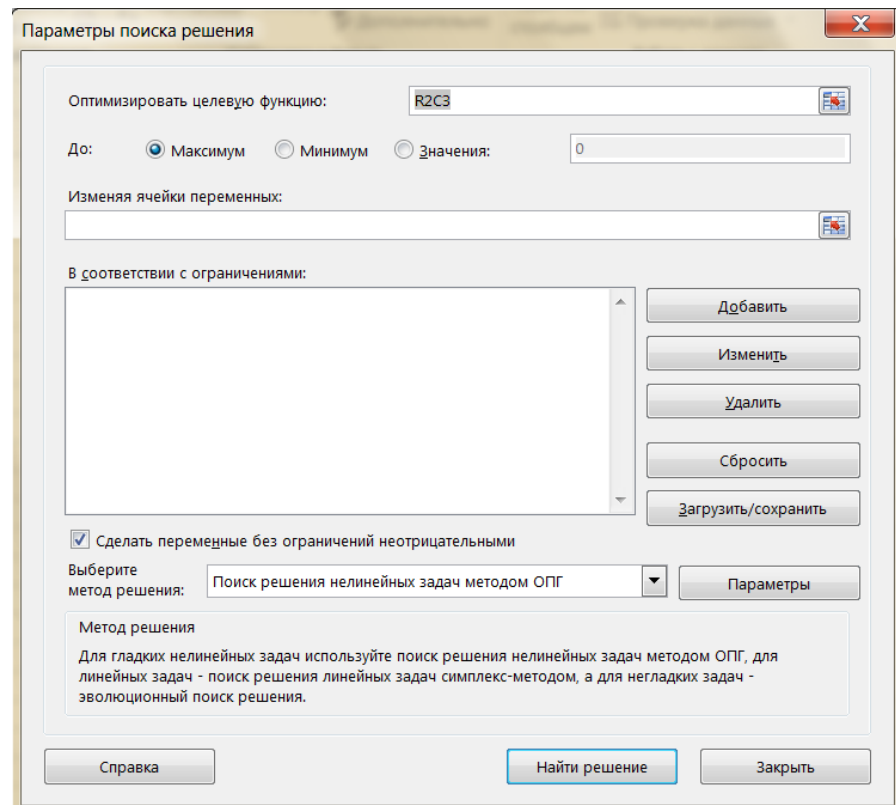


Natijada Excel dasturining “Данные” lentasida “Анализ” guruhi va unda “Поиск решения” buyrug’i paydo bo’ladi.



Endi bevosita yechimlarni qidirish amalini bajarishga qirishishimiz mumkin. Yechimlarni qidirish amalining bajarilish ketma-ketligi quyidagicha:

1. Katakлар orasida bog’lanishlarni o’rnatadigan formulalarni ifodalagan jadvalni hosil qiling.
2. Kerakli qiymatni qabul qilishi lozim bo’lgan maqsad funksiyasi joylashgan katakni ajrating va “Данные” lentaning “Анализ” guruhidan “Поиск решения” buyrug’ini bajaring. Natijada “Параметры поиска решения” nomli muloqot oynasi ochiladi. Undagi “Оптимизировать целевую функцию:” maydonida maqsad funksiyasi joylashgan katakning manzili yoziladi.



3. Maqsad funksiyasi qabul qilishi kerak boʻlgan holatlardan (“Максимум, Минимум, Значение”) biri beriladi. Agar oxirgisi tanlansa, uning oʻng tomonidagi maydonga qaysi qiymatga teng boʻlishi kiritiladi.

4. Optimal natijani qidirishda, dastur qaysi kataklardagi qiymatlarni oʻzgartirishi kerakligi “Изменяя ячейки переменных:” maydonida koʻrsatiladi.

5. Mavjud cheklovlar “В соответствии с ограничениями:” roʻyxatida akslantiriladi. Buning uchun “Добавить” tugmasi bosiladi va hosil boʻlgan “Добавление ограничения” muloqot oynada cheklovlar beriladi.

6. “Выберите метод решения:” roʻyxatidan yechimni (“Поиск решения нелинейных задач методом ОПГ, Поиск решения лин. задач симплекс методом, Эволюционных поиск решения”) usullaridan biri tanlanadi. Agar “Эволюционных поиск решения” usuli tanlansa, “Параметры” ni bosish orqali, ushbu usul kattaliklarini kiritish muloqot oynasi ochiladi va unda kerakli kattaliklar kiritiladi.

7. Yechimni qidirish jarayonini ishga tushirish “Найти решение” ni bosish bilan amalga oshiriladi.

8. Yechimni qidirish jarayoni tugagandan so'ng, “Результаты поиска решения” muloqot oynasi ochiladi. Undagi “Сохранить найденное решение” yoki “Восстановить исходные значения” buyruqlaridan keraklisi tanlanadi va OK bosiladi.

Ekkranda “Параметры поиска решения” kattaliklari haqida ma'lumot beriladi.

Excel 2007 versiyasidan boshlab yechimni qidirish vositasini ishlatish bir xil tamoyilga asoslangan, faqat keying versiyalarda oyna ko'rinishi biroz o'zgartirilgan.

Marshrutlash masalasi.

Qaror qabul qilishning matematik modeli bo'lgan optimallashtirishning eng keng tarqalgan masalasi – marshrutlashni o'rganish hisoblanadi. Kommivoyajer masalasi marshrutlashning bazaviy modeli deb qabul qilingan.

Kommivoyajer masalasi dastlab marketing sohasida qo'llanilgan. Keyinchalik boshqaruv faoliyatining boshqa sohalariga, ayniqsa hududlardagi tarqoq ob'ektlarning boshqaruviga tadbiq etila boshlandi.

Bunday amaliy qo'llanilishga misol sifatida:

- Sanoatning har xil sohalaridagi tarqoq ob'ektlarga xizmat ko'rsatishda;
- Ekspeditsiyalarni uchib aylanib chiqish va mos yuklarni ularga tashlashda;
- Sanoat sohalaridagi texnik ob'ektlarga xizmat ko'rsatishda optimal marshrutlarni aniqlashda.

Turli hududlarda joylashgan sanoat korxonalarida marshrutlash masalasi alohida o'rin tutadi. Masalan, korxona muhandisi xizmat faoliyati chog'ida turli hududlarda joylashgan ob'ektlarga turli zaruriy jihozlarni etkazishi, ob'ektlarga borishiga va ulardagi ish jarayonlarini kuzatishiga to'g'ri keladi. Muhandisning xizmati yo'nalishini optimallashtirish ushbu xizmat samaradorligini oshirib, yo'nalishdagi bir ob'ektdan boshqasiga qisqa vaqt ichida o'tilishini ta'minlaydi.

Masalaning qo'yilishi. Muhandis tomonidan m ta ob'ekt tekshirilishi kerak. Bu ob'ektlarning hududdagi joylashishi ma'lum va $C = |c_{ij}|$, $i = \overline{1, m}$, $j = \overline{1, m}$ matritsa bilan aniqlanadi. (C – masofalar matritsasi, c_{ij} i - ob'ektdan j – ob'ektgacha bo'lgan masofa). Muhandis joylashgan punkt $i = 1$ deb qabul qilinadi. Harakat shu punktdan boshlanadi. Muhandis eng qisqa vaqt orasida har qaysi ob'ektni bir martadan tekshirishi, so'ngra boshlang'ich punktga qaytishi talab etiladi, ya'ni optimal yo'nalishni tuzishi kerak. Ayniqsa, ob'ektlarga turli xil jihozlarni yetkazganda, jihozning turi va boriladigan ob'ektlarning o'rniga muvofiq optimal marshrutlash muhim hisoblanadi.

Umumiy holda, masalaning aniq yechimi m kattalik bilan tavsiflanuvchi masala o'lchamiga hamda qo'llanilayotgan algoritim va kompyuterning samaradorligiga bog'liq.

Matematik model. Ob'ektlarni aylanib o'tish rejasi $|x_{ij}|$ - matritsa orqali berilgan bo'lsin. Matritsa elementlari ikkilik sonlardan iborat, ya'ni $x_i = 1$, agar i - ob'ektdan j – ob'ektga o'tsa va $x_i = 0$, agar i - ob'ektdan j – ob'ektga bormasa.

Maqsad funktsiyasining

$$F = \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij}$$

Minimumini quyidagi cheklovlarda aniqlashni talab etiladi:

$$\sum_{i=1}^m x_{ij} = 1, \quad j = \overline{1, m}$$

faqat bir marta j – ob'ektga kirish.

$$\sum_{j=1}^m x_{ij} = 1, \quad i = \overline{1, m}$$

faqat bir marta i – ob'ektdan chiqish.

Bu turdagi masalalarda yana bir qo'shimcha shart qo'yiladi:

$$u_i - u_j + (m-1)x_{ij} \leq m-2, \quad i \neq j, \quad i, j = 2, \dots, m$$

Bu maxsus shart o'zaro bog'lanmagan bir qancha yo'nalish va takrorlanishlarni bartaraf etadi, ya'ni muhandisning ob'ektlarni aylanib chiqishi yopiq marshrut bo'ylab amalga oshadi.

Demak, muhandisning ob'ektlarni aylanib chiqishining shunday rejasi tuzilishi kerakki, “Umumiy masofa” = “Masofalar matritsasi” * “Aylanib o'tish rejasi” minimal qiymatni quyidagi cheklovlarda qabul qilishi kerak:

- “*Kirish*” = 1 (muhandisning ob`ektga kirishi).
- “*Chiqish*” = 1 (muhandisning ob`ektdan chiqishi).
- o`zaro bog`lanmagan yo`nalishlar mavjudligini istisno qilish.

Excel dasturida masalani yechish: beshta ob`ekt berilgan bo`lib, ular orasidagi c_{ij} masofa (km da) quyidagi jadvalda berilgan.

Ob`ekt raqami	1	2	3	4	5
1	∞	11	10	6	12
2	8	∞	6	7	9
3	7	5	∞	8	4
4	3	9	4	∞	5
5	4	6	7	4	∞

Jadvalning diagonal kataklarida cheksiz belgisi (∞) yozilgan. Ob`ektning o`zidan o`ziga marshrut bo`lmagani uchun bu belgi o`rniga amaldagi jadvaldagi sonlarga nisbatan katta son ko`rsatiladi.

Excel dasturining ish sahifasiga boshlang`ich ma`lumotlarni, cheklovlar va maqsad funksiyasining hisoblash formulalarini kiritamiz.

1) “*O`zgaruvchilar matritsasi*” va “*cheklovlar*” matnini mos ravishda B2 va G2 kataklarga yozamiz.

2) Qulaylik uchun birdan beshgacha bo`lgan raqamlarni B3:F3 va A4:A8 kataklar sohasiga kiritamiz.

3) O`zgaruvchilar matritsasi uchun B4:F8 kataklar sohasini ajratamiz va ularning qiymati sifatida nolni kiritamiz.

4) Cheklovlar formulalarini (ob`ektga faqat bir marta kirish) G4:G9 kataklar sohasiga yozamiz: $CYMM(B4:F4)$. $=CYMM(B5:F5)$. $=CYMM(B6:F6)$. $=CYMM(B7:F7)$. $=CYMM(B8:F8)$.

5) Cheklovlar so`zini A9 katakka kiritamiz.

6) Cheklovlar formulalarini (ob`ektdan faqat bir marta chiqish) mos ravishda B4:F9 kataklarga yozamiz: $=CYMM(B4:B8)$. $=CYMM(C4:C8)$. $=CYMM(D4:D8)$. $=CYMM(E4:E8)$. $=CYMM(F4:F8)$.

	A	B	C	D	E	F	G
1							
2		Ўзгарувчилар матрицаси					чекловлар
3		1	2	3	4	5	
4	1	0	0	0	0	0	=СУММ(B4:F4)
5	2	0	0	0	0	0	=СУММ(B5:F5)
6	3	0	0	0	0	0	=СУММ(B6:F6)
7	4	0	0	0	0	0	=СУММ(B7:F7)
8	5	0	0	0	0	0	=СУММ(B8:F8)
9	чекловлар	=СУММ(B4:B8)	=СУММ(C4:C8)	=СУММ(D4:D8)	=СУММ(E4:E8)	=СУММ(F4:F8)	
10	B10да мақсад функцияси	=СУММПРОИЗВ(B4:F8;B14:F18)					
11	C11:F11 блокта u ўзгарувчилар						

7) “B10 da maqsad funksiyasi” va “C11:F11 sohada u o`zgaruvchilar matnini A10 va A11 kataklarga kiritamiz.

8) B10 katakka – СУММПРОИЗВ(B4:F8.B14:F18) formulani yozamiz.

9) “Masofalar matritsasi” matnini, birdan beshgacha raqamlarni mos ravishda B12, B13:F13, A14:A18 kataklarga kiritamiz.

10) Masofalar matritsasi qiymatini B14:F18 kataklar sohasiga yozamiz.

11) “Qo`shimcha u o`zgaruvchilar bo`yicha cheklovlar uchun formulalar” matnini A19 katakka yozamiz.

12) A21:A24, B20:E20 kataklar sohasiga u2, u3, u4, u5 belgilarni kiritamiz.

13) A21:E24 kataklar sohasiga “o`zaro bog`lanmagan yo`nalishlar mavjudlik holatini istisno qiluvchi” cheklovlarni, ya`ni:

B21:E21 kataklarga:

=C11-C11+4*C5: =C11-D11+4*D5. =C11-E11+4*E5. =C11-F11+4*F5.

B22:E22 kataklarga:

=D11-C11+4*C6: =D11-D11+4*D6. =D11-E11+4*E6. =D11-F11+4*E6.

B23:E23 kataklarga:

=E11-C11+4*C7: =E11-D11+4*D7. =E11-E11+4*E7. =E11-F11+4*F7.

B24:E24 kataklarga:

=F11-C11+4*C8: =F11-D11+4*D8. =F11-E11+4*E5. =F11-F11+4*F8.

formulalarni yozamiz.

	A	B	C	D	E	F
12		Масофалар матрицаси				
13		1	2	3	4	5
14	1	9999	11	10	6	12
15	2	8	9999	6	7	9
16	3	7	5	9999	8	4
17	4	3	9	4	9999	5
18	5	4	6	7	4	9999
19	Қўшимча u ўзгаришчилар бўйича чекловлар учун формулалар					
20		u2	u3	u4	u5	
21	u2	=C11-C11+4*C5	=C11-D11+4*D5	=C11-E11+4*E5	=C11-F11+4*F5	
22	u3	=D11-C11+4*C6	=D11-D11+4*D6	=D11-E11+4*E6	=D11-F11+4*F6	
23	u4	=E11-C11+4*C7	=E11-D11+4*D7	=E11-E11+4*E7	=E11-F11+4*F7	
24	u5	=F11-C11+4*C8	=F11-D11+4*D8	=F11-E11+4*E8	=F11-F11+4*F8	

14) “Поиск решения” elementini ishga tushirib, unga quyidagi kattaliklarni kiritamiz:

Maqsad katagi - \$B\$10.

U kichik qiymatga teng (Равной: * минимальному значению).

Qiymatlari o`zgaruvchi kataklar (Изменяя ячейки): \$B\$4:\$F\$8.
\$B\$11:\$F\$11. Bunda faqat qiymatlari o`zgaruvchi va masalaning yechimi yoziluvchi kataklar (\$B\$4:\$F\$8) bilan bir qatorda  $u_i$  o`zgaruvchini o`zida saqllovchi va qiymatlari o`zgaruvchi \$C\$11:\$F\$11 kataklar ham kiritiladi.

Чекловлар:

\$B\$21:\$E\$24 ≤ 3.

\$B\$4:\$F\$8 = ikkilik son.

\$B\$9:\$F\$9 = 1.

\$G\$4:\$G\$8 = 1.

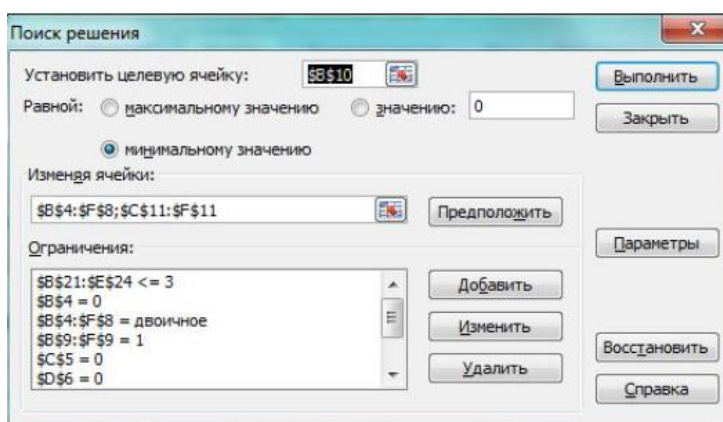
\$B\$4 = 0.

\$C\$5 = 0.

\$D\$6 = 0.

\$E\$7 = 0.

\$F\$8 = 0.



Катталиклар: chiziqli model (линейная модель), manfiy bo`lmagan qiymatlar (неотрицательные значения), to`g`ridan-to`g`ri masshtablash (автоматическое масштабирование).

Параметры поиска решения

Максимальное время: 100 секунд

Предельное число итераций: 100

Относительная погрешность: 0.000001

Допустимое отклонение: 5 %

Сходимость: 0.0001

☒ Линейная модель

☒ Неотрицательные значения

☒ Автоматическое масштабирование

☐ Показывать результаты итераций

Оценки: ☒ линейная ☐ квадратичная

Разности: ☒ прямые ☐ центральные

Метод поиска: ☒ Ньютона ☐ сопряженных градиентов

ОК Отмена Загрузить модель... Сохранить модель... Справка

“Поиск решения” muloqot oynada kursorni “Выполнить” tugmasiga keltirib sichqonchanning chap tugmasi bosilsa, dasturning ish sahifasida masalaning yechimi hosil bo`ladi.

Optimal yechim: maqsad funksiyasining qiymati $F=28$ bo`lib, ob`ektlarni aylanib chiqish 1, 4, 3, 5, 2, 1 ketma-ketlikda amalga oshiriladi.

	A	B	C	D	E	F	G
1							
2		Ўзгарувчилар матричаси					чекловлар
3			1	2	3	4	5
4	1	0	0	0	0	1	0
5	2	1	0	0	0	0	1
6	3	0	0	0	0	1	1
7	4	0	0	1	0	0	1
8	5	0	1	0	0	0	1
9	чекловлар	1	1	1	1	1	
10	B10 да мақсад функцияси	28					
11	C11:F11 блока и ўзгарувчилар			3	1	0	2
12		Масофалар матричаси					
13			1	2	3	4	5
14	1	9999	11	10	6	12	
15	2	8	9999	6	7	9	
16	3	7	5	9999	8	4	
17	4	3	9	4	9999	5	
18	5	4	6	7	4	9999	
19	Қўшимча и ўзгарувчилар бўйича чекловлар учун формулалар						
20	u2	u3	u4	u5			
21	u2	0	2	3	1		
22	u3	-2	0	1	3		
23	u4	-3	3	0	-2		
24	u5	3	1	2	0		

Nazorat savollari:

1. MS Excelda Solver vositasi vazifasi nimadan iborat?
2. Optimal yechim deganda nimani tushunasiz?
3. “Поиск решения” (yechimni qidirish) vositasi qanday faollashtiriladi?
4. Yechimlarni qidirish amalining bajarilish ketma-ketligini tavsiflang.
5. Marshrutlashning bazaviy modeli, kommivoyajer masalasining mohiyati nima?
6. Kommivoyajer masalasining modelini tavsiflang.
7. Kommivoyajer masalasining matematik modeli qanday shakllantiriladi?
8. Kommivoyajer masalasiga qanday qo'shimchs cheklovlar qo'yiladi?
9. “Параметры поиска решения” muloqot oynasidagi “Автоматическое масштабирование” kattaligi nima sababdan faollashtiriladi?
10. Kommivoyajer masalasiga o'xshash masalaga misol keltiring?

MATEMATIK MODELLASHTIRISHNING ELEKTRON JADVAL YORDAMIDA YECHISH.

Optimallashtirish masalasi

Chiziqli programmashtirish inson faoliyatining turli sohalarida keng qo'llaniladigan fanlardan biridir. Uning asosiy yo'nalishi esa optimallashtirish masalasidir. Bu sohasidagi muvaffaqiyatlarga katta texnik tizimlarni loyihalash va tahlil qilish natijasida erishilgan.

Inson oldiga qo'yiladigan barcha masalalarni yechishda yaxshi yoki yomon degan yechimni qabul qilishi mumkin. Yechim qabul qilish jarayoni formallashtirish va formallashtirmagan holda bo'lishi mumkin.

Formallashtirmagan yechim qabul qilish – bu ijdod, san'atdir. Formallashtirmagan yechimni qabul qilishda insondan hech narsa talab etmaydi. Masalan, inson o'tirdi, o'yladi, yechim qabul qildi. Haqiqatdan ham bu holda yechimning to'g'riligiga hech qanday kafolat yo'q. Inson ko'p hollarda hech qanday asossiz o'zi o'ylagan holda, ayrim hollarda esa tajribasiga ishongan holda yechimlarni qabul qiladi.

Formallashtirishgan yechimga kelsak, u aniq tavsiyaga asoslanib qabul qilinadi.

Formallashtirishgan yechim qabul qilish ikkita usulga asoslanadi:

- mantiqiy modellash;
- optimallashtirish

Mantiqiy modellashda yuqori bilimga ega bo'lgan mutaxassislar tomonidan usul tanlanib, uning yordamida biror bir holat yoki boshqa holatda nima qilish kerakligi aniqlanadi. Bunda matematik mantiqiy funktsiyalar xizmat qiladi.

Yechim qabul qilishning optimallashtirish usuli quyidagilarga asoslanadi:

- iqtisodiy matematik modellash;
- kompyuterda masalani yechish;
- boshlang'ich ma'lumotlar;

Iqtisodiy matematik modellashning ikkita imkoniyati mavjud:

- qo'yilgan savolga tez javob berish;
- keng tajriba o'tkazish imkoniyati.

Yechim qabul qilish algoritmi ancha murakkab bo'lib, ayniqsa hozirgi bozor iqtisodiyoti davrida kompyuterni qo'llamasdan turib amaliy jihatdan bajarish mumkin emas. Kompyuterda optimal yechimni izlash algoritmi uchun ishlab chiqilgan dasturiy ta'minot va boshlang'ich ma'lumotsiz izlanayotgan natijaga erishib bo'lmaydi.

Ma'qul bo'lgan yechimlarni qabul qilishni quyidagi etaplarga bo'lish mumkin:

1. Masalani tanlash. Bunda masala qanday talablarni qanoatlantirishi kerakligi aniqlashtiriladi.
2. Masalani qo'yish. Bunda masalaning iqtisodiy matematik modeli va uning elementlari aniqlanadi.
3. Masalaning iqtisodiy matematik modelini tuzish. Bu etapda optimal yechim qabul qilishga asoslangan iqtisodiy matematik modelni tuzish masalasi qaraladi.
4. Masala uchun boshlang'ich ma'lumotlarni yig'ish. Bu etapda tuzilgan iqtisodiy matematik modelga mos boshlang'ich ma'lumotlar yig'iladi.
5. Masalani yechish. Bu etapda kerakli usul tanlanib masala yechiladi.
6. Yechimni tahlil qilish. Bu etapda olingan optimal yechim iqtisodiy matematik model asosida tahlil qilinadi.
7. Optimal yechimni qabul qilish. Optimal yechimni olingan natijalarga asoslanib mutaxassis tomonidan qabul qilinadi.

Simpleks usuli chiziqli dasturlash masalasini yechishning asosiy usullaridan biri bo'lib, ketma-ket yaqinlashish usuli yordamida $x_1, x_2, \dots, x_n$ o'tkazuvchilarning shunday optimal qiymatini topadiki, bu qiymatlar maqsad funksiyasiga maksimal (yoki minimal) qiymat beriladi.

Chiziqli dasturlash masalasini simpleks usul yordamida yechish ikki etapdan iborat:

1. Boshlang'ich tayanch rejani topish.
2. Tayanch planlar ichidan masalaning optimal rejasini topish.
3. Jadvalda simpleks almashtirish bajariladi.

4. Yangi topilgan simpleks jadvalda tayanch plan mavjud bo'lsa ikkinchi bosqichga, ya'ni optimal plani topishga o'tiladi, aks holda yuqoridagi jarayon yangi jadval uchun toki tayanch plan topilguncha qayta takrorlanadi.

$$Z = \sum_{i=1}^m c_i x_j \quad Z = \sum_{j=1}^n c_i x_j \quad \begin{matrix} i = 1, m \\ j = 1, n \end{matrix} \quad \frac{dZ}{dx_j} = 0$$

(13) Masala sharti:

Fermadagi mollarni boqish uchun 2 turdagi yemdan ratsion tanlash mumkin. 1-turdagi 1 kg ozuqa tarkibida: yog` 1 birlik, oqsil 3 birlik, 1 birlik uglevod, 2 birlik nitrat mavjud va narxi 80 pul birligiga teng. 2-turdagi 1 kg ozuqa tarkibida esa: yog` 3 birlik, oqsil 1 birlik, 8 birlik uglevod, 4 birlik nitrat mavjud va narxi 10 pul birligiga teng. Tarkibida kamida 6 birlik yog`, 9 birlik oqsil, 8 birlik uglevod va 16 birlik nitratlar bo'lgan eng arzon ratsionni tuzing.

Masala shartiga muvofiq ma'lumotlarni Excel jadval protsessoridan foydalanib shakllantirish mumkin. Maqsad funksiyasi uchun quyidagi formuladan foydalaniladi:

Variant 13

Fermadagi mollarni boqish uchun 2 turdagi yemdan ratsion tanlash mumkin. 1-turdagi 1kg ozuqa tarkibida: yog` 1 birlik, oqsil 3 birlik, 1 birlik uglevod, 2 birlik nitrat mavjud va narxi 80 pul birligiga teng. 2-turdagi 1kg ozuqa tarkibida esa: yog` 3 birlik, oqsil 1 birlik, 8 birlik uglevod, 4 birlik nitrat mavjud va narxi 10 pul birligiga teng. Tarkibida kamida 6 birlik yog`, 9 birlik oqsil, 8 birlik uglevod va 16 birlik nitratlar bo'lgan eng arzon ratsionni tuzing.

	Ozuqa 1 tarkibi	Ozuqa 2 tarkibi	Har biri uchun formula	Talab tarkibi
Yog	1	3	0	6
Oqsil	3	1	0	9
Uglevod	1	8	0	8
Nitrat	2	4	0	16

Ozuqa 1	Ozuqa 2	M funksiya
0	0	0

Formula bar: $=80 \cdot K7 + 10 \cdot L7$

Ozuqa 1 va **Ozuqa 2** uchun tegishli ma'lumotlarni mos ustunlarga kiritib, ozuqalarning tarkibiy qismlarining har biri uchun mos hisoblashlarni bajaradigan formuladan foydalaniladi:

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad (i = \overline{1, m})$$

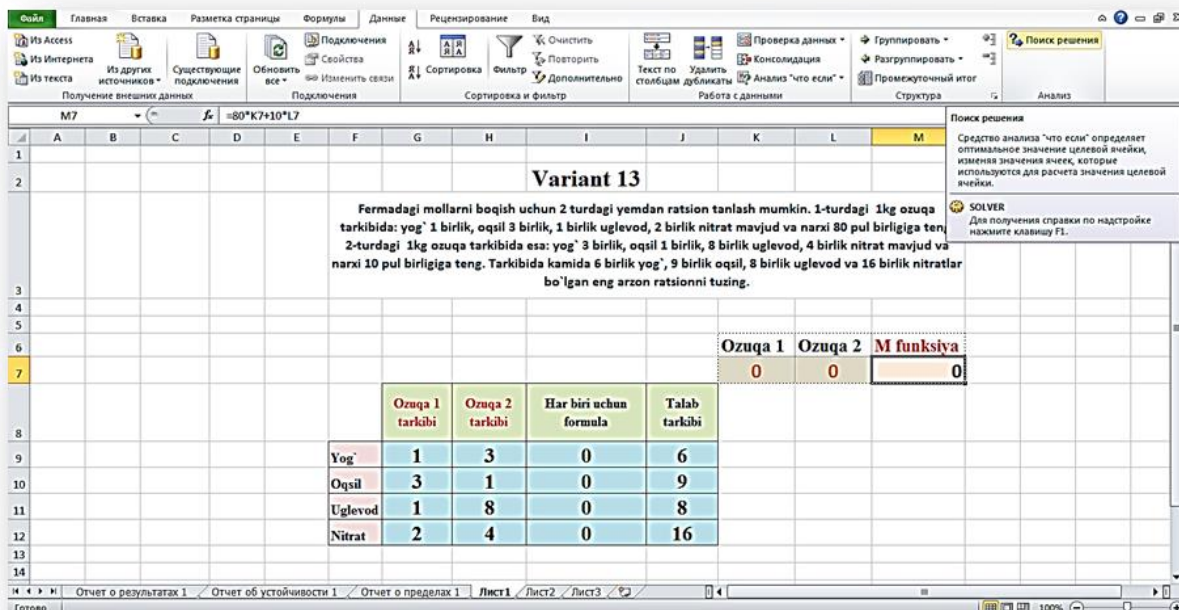
$$x_j \geq 0 \quad (j = \overline{1, n})$$

$$Z = \sum_{j=1}^n c_j x_j \rightarrow \max$$

	Ozuqa 1 tarkibi	Ozuqa 2 tarkibi	Har biri uchun formula	Talab tarkibi
Yog`	1	3	0	6
Oqsil	3	1	0	9
Uglevod	1	8	0	8
Nitrat	2	4	0	16

Hisoblashlarning o`zaro bog`liqligi

	Ozuqa 1	Ozuqa 2	M funksiya
	0	0	0
Har biri uchun formula			
0			
0			
0			
0			



Параметры поиска решения

Оптимизировать целевую функцию:

До: ☐ Максимум ☒ Минимум ☐ Значения:

Изменяя ячейки переменных:

В соответствии с ограничениями:

Добавить
Изменить
Удалить
Сбросить
Загрузить/сохранить

☒ Сделать переменные без ограничений неотрицательными

Выберите метод решения: Параметры

Метод решения

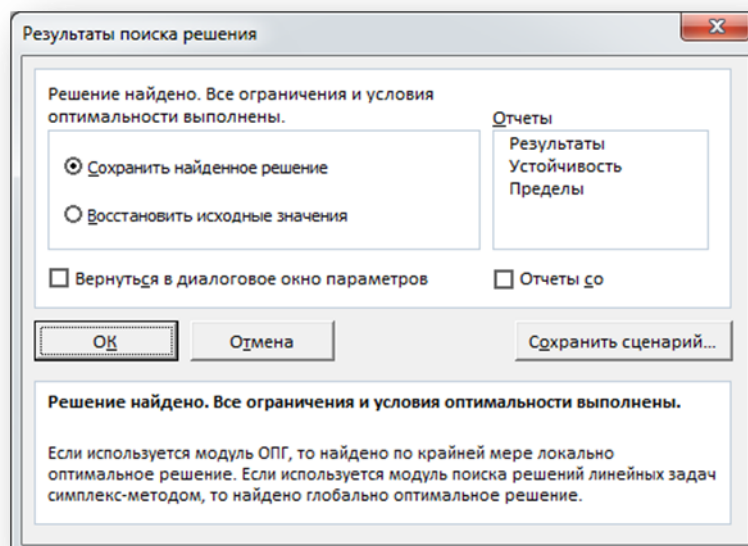
Для гладких нелинейных задач используйте поиск решения нелинейных задач методом ОПГ, для линейных задач - поиск решения линейных задач симплекс-методом, а для негладких задач - эволюционный поиск решения.

Справка Найти решение Закрыть

Qoidaga binoan
Ozuqa 1 va **Ozuqa 2**
ni “>=” katta yoki
teng 0 (nol) shartini
kiritamiz

Mos ravishda **Har**
biri uchun formula
ustunini **Talab**
tarkibi ustuni bilan
katta yoki teng
shartini kiritamiz

Выберите
метод решения: Поиск решения линейных задач симплекс-методом
Метод решения Поиск решения нелинейных задач методом ОПГ
Поиск решения линейных задач симплекс-методом
Эволюционный поиск решения



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
4																	
5																	
6																	
7																	
8																	
9																	
10																	
11																	
12																	
13																	

Ozuqa 1	Ozuqa 2	M funksiya
0	9	90

Yuqorida berilgan ko`rsatkichlar bo`yicha qo`yilgan chiziqli masalani simpleks metodi yordamida optimal yechimi sifatida **Ozuqa 2** dan 9 birlikni 90 pul birligiga olish eng ma`qul hisoblanadi.

2-Masala.

Masala sharti:

Quyidagi shartlarga binoan transpot masalasini yeching:

Quyidagi jadvalda 3 ta yetkazuvchi va 4 ta talabgorlar keltirilgan. Yetkazuvchilarning quvvati va talabgorlarning mahsulotga bo'lgan ehtiyojini hisobga olib, mahsulotlarni talabgorlarga eng arzon yetkazish masalasini hal eting.

Yetkazuvchi	Yetkazuvchlarning quvvati	Talabgorlar va ularning talabi			
		1	2	3	4
		50	50	10	60
1	30	5	4	6	3
2	70	4	5	5	8
3	70	7	3	4	7

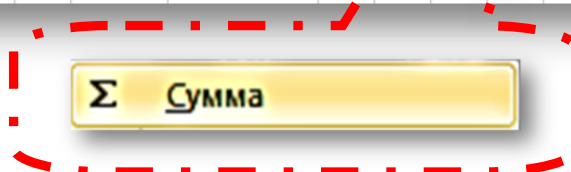
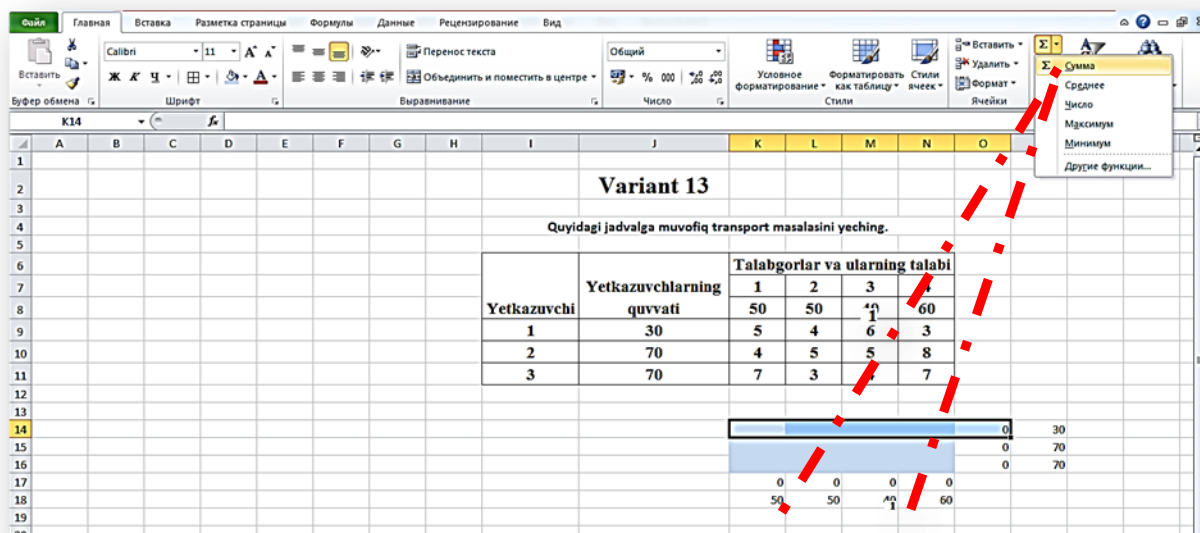
Yechim:

Masalani yechish uchun quyidagicha matematik modelni qurib olamiz.

$$Z = \sum_{i=1}^3 \sum_{j=1}^4 c_{ij} x_{ij} \quad x_{ij} \geq 0, \quad i \in [1,3], \quad j \in [1,4]$$

$$\sum_{i=1}^3 x_{ij} = b_{ij}, \quad j \in [1,4] \quad \sum_{j=1}^4 x_{ij} = a_{ij}, \quad i \in [1,3]$$

Masalaning amaliy qismini Excel jadval protsessoridan foydalanib bajaramiz.



Скриншот Excel-таблицы с заголовком "Variant 13". В центре таблицы находится таблица "Talabgorlar va ularning talabi".

Yetkazuvchi	Yetkazuvchilarning quvvati	Talabgorlar va ularning talabi			
		1	2	3	4
1	30	5	4	6	3
2	70	4	5	5	8
3	70	7	3	4	7

В правой части таблицы "Talabgorlar va ularning talabi" есть дополнительные данные:

	1	2	3	4
0	0	0	0	0
50	50	40	60	60

В правой части таблицы "Talabgorlar va ularning talabi" есть дополнительные данные:

	1	2	3	4
0	0	0	0	0
50	50	40	60	60

Скриншот Excel-таблицы с заголовком "Variant 13". В центре таблицы находится таблица "Talabgorlar va ularning talabi".

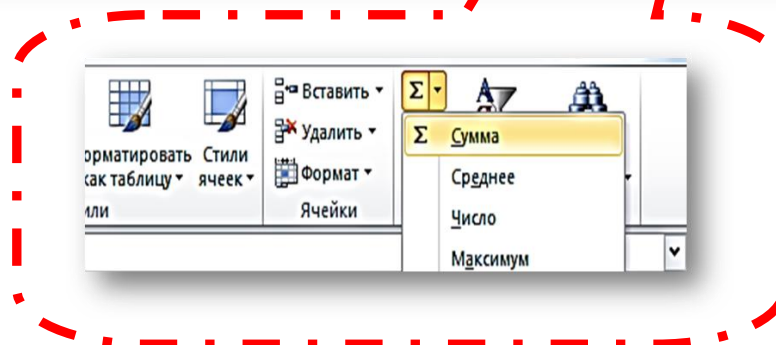
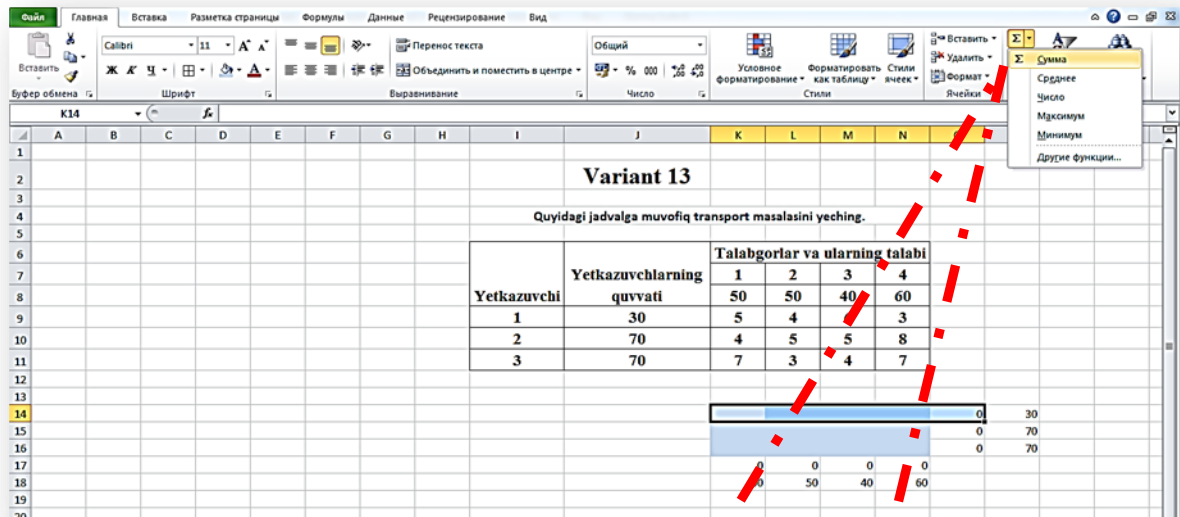
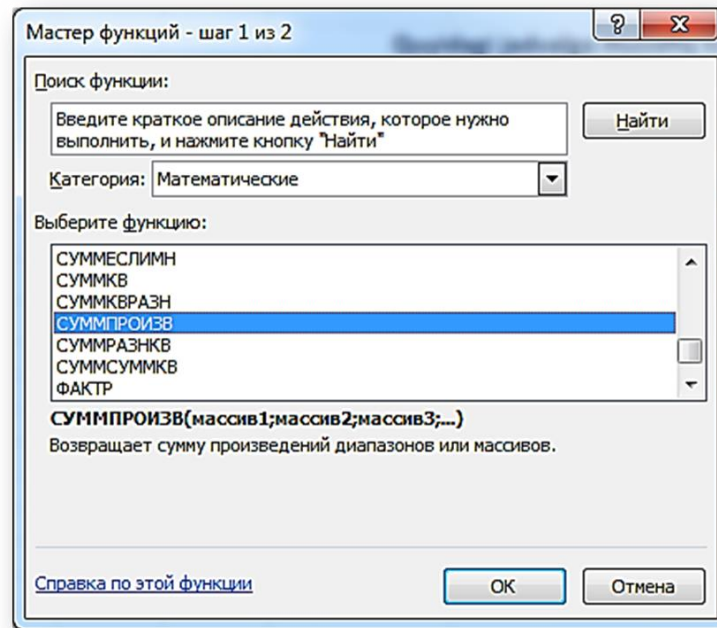
1	2	3	4
50	50	40	60
5	4	6	3
4	5	5	8
7	3	4	7

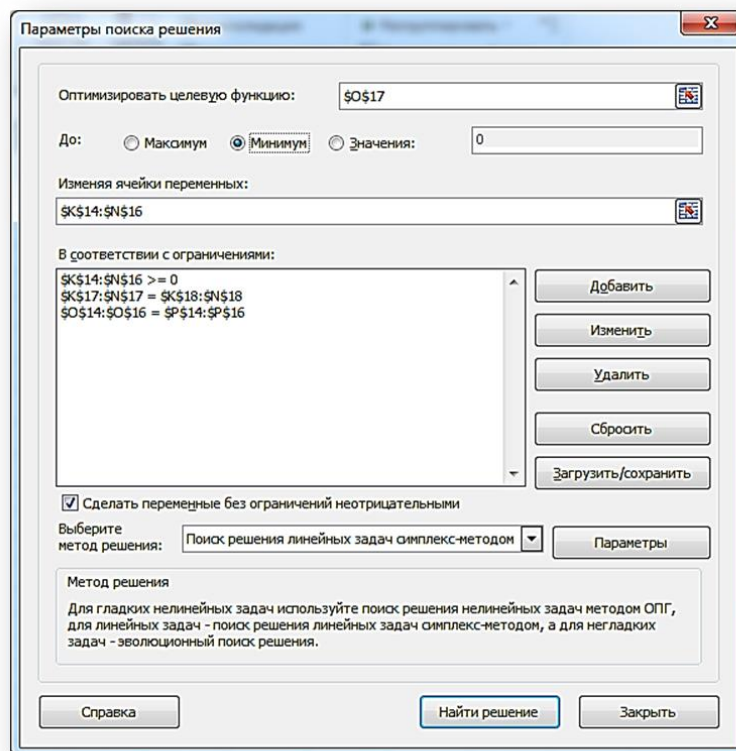
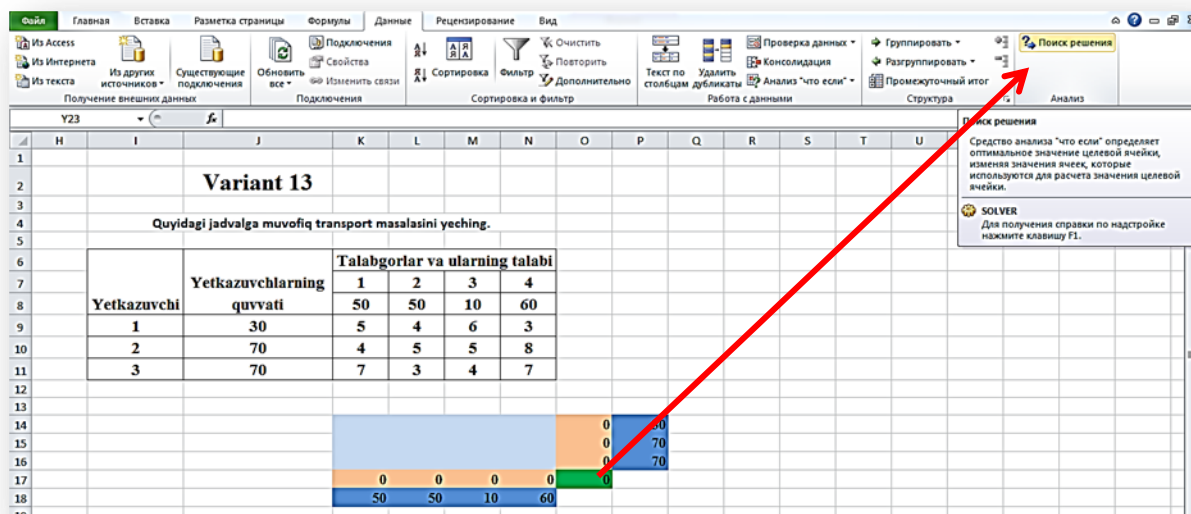
В правой части таблицы "Talabgorlar va ularning talabi" есть дополнительные данные:

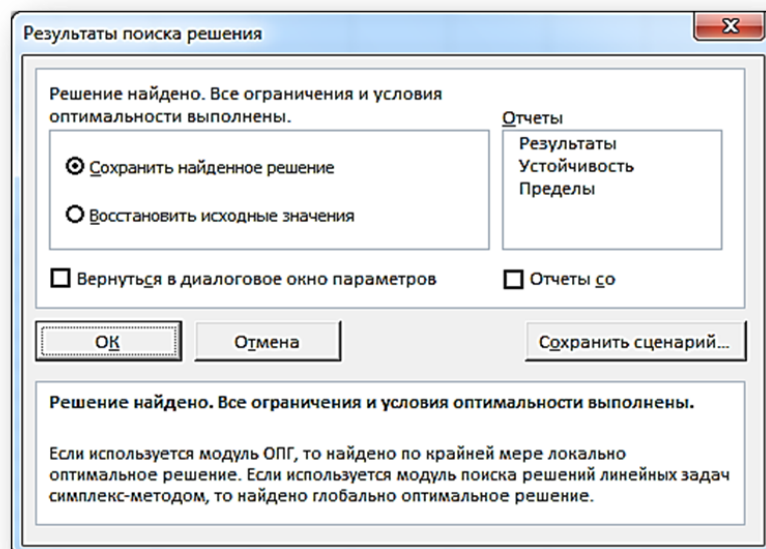
	1	2	3	4
0	0	0	0	0
50	50	40	60	60

В правой части таблицы "Talabgorlar va ularning talabi" есть дополнительные данные:

	1	2	3	4
0	0	0	0	0
50	50	40	60	60







Quyidagi jadvalga muvofiq transport masalasini yeching.

Yetkazuvchi	Yetkazuvchlarning quvvati	Talabgorlar va ularning talabi					
		1	2	3	4		
		50	50	10	60		
1	30	5	4	6	3		
2	70	4	5	5	8		
3	70	7	3	4	7		
		0	0	0	30	30	30
		50	0	0	20	70	70
		0	50	10	10	70	70
		50	50	10	60	710	
		50	50	10	60		

Masala shartiga binoan mahsulotni yetkazuvchilardan talabgorlarga tashishning transport masalasi optimal yechimi quyidagicha amalga oshiriladi va umumiy qiymati 710 pul birligiga teng.

0	0	0	30	30	30
50	0	0	20	70	70
0	50	10	10	70	70
50	50	10	60	710	
50	50	10	60		

MATHCADDA MATEMATIK IFODALAR VA FUNKSIYALAR BILAN ISHLASH.

MathCAD dasturi haqida umumiy ma'lumotlar

MathCAD – bu matematikaning turli sohalaridagi masalalarni yechishga mo'ljallangan ajoyib tizimdir. U o'z ichiga matematikaning rivojlanishi natijasida yig'ilgan tajribalar, qoidalar va matematik hisoblash usullarini olgan.

MathCAD paketi muhandislik hisob ishlarini bajarish uchun dasturiy vosita bo'lib, u turli soha vakillari uchun mo'ljallangan. U yordamida o'zgaruvchi va o'zgarmas parametrli algebraik va differentsial tenglamalarni yechish, funksiyalarni tahlil qilish va ularning ekstremumini izlash, topilgan yechimlarni tahlil qilish uchun jadvallar va grafiklar qurish mumkin. MathCAD murakkab masalalarni yechish uchun o'z dasturlash tiliga ham ega.

MathCAD (Mathematical Computer Aided Design) 1986 yil Massachuset texnologiya institutida, MathSoft Inc. kompaniyasi ta'sischisi Allen Razdov tomonidan yaratildi. Dasturning nomlanishi ikkita so'zdan iborat – MATHeMatica (matematika) va CAD (avtomatik loyihalash tizimi).

MathCAD foydalanuvchiga tenglamalarni kiritish, tahrirlash, yechish, natijalarni vizualizatsiya qilish, ularni hujjat shaklida rasmiylashtirish va natijalarni tahlil qilish maqsadida boshqa dasturlar bilan almashinish imkoniyatini beradi. Hujjat-ilovalarni tayyorlashda WYSIWYG (What You See Is What You Get - "Nimani ko'rsang, shuni olasan") tamoyili qo'llaniladi. Dastur asosan dasturchi bo'lmagan foydalanuvchilarga mo'ljallangan bo'lsada, u murakkab loyihalarda matematik modellashtirish natijalarini vizuallashtirish imkonini beradi.

Dastlab dastur matematik masalalarni sonli yechish maqsadida yaratilgan. 1994 yildan boshlab unga simvolli hisoblashlarni bajarish funksiyalari qo'shildi. Bunda Maple dasturining simvolli hisoblashlaridan foydalanilgan.

MathCAD murakkab hisoblashlarni bajara olish imkoniga ega. MathCAD yuzdan ortiq o'zgaruvchili va konstantali chiziqli va chiziqsiz tenglamalar sistemasini, matritsa va vektorlar ustida amallarni, algebraik hisoblashlarni, Laplas, Fur'e integrallarini, massivlarni, oddiy differentsial tenglamalarni, chegaraviy

shartlarni, xususiyl hosilali differentsial tenglamalarni, polinomlarni tushuna oladi hamda ular ustida hisoblash ishlarini bajara oladi.

MathCAD turli funksiyalarni grafiklarini qura oladi. Unga ixtiyoriyl matnni joylashtirish mumkin, bu esa hisoblash jarayonini izohlash uchun juda zarurdir. Foydalanuvchi MathCADda funksiya va tenglamalariga bema'lol izoh yozishi hamda 2 va 3 o'lchovli grafiklarni chizishi mumkin. Shuningdek, u grafiklarini turli ranglar orqali, turli qalinlikda tasvirlay olish imkoniga ham ega bo'ladi.

Umuman olganda MathCAD muhitida ishlashning yaqqol ajralib turadigan imkoniyatlarini ajratib ko'rsatish mumkin:

- Windows bilan moslik. Masalan, oyna o'lchovlarini o'zgartirish va ularni siljitish, bir necha oyna ochish, sichqonchani qollash, menyu va asboblal paneli bilan ishlash.
- Matematik ifodalar matematikada qabul qilingan tabiiyl ko'rinishda yoziladi. Masalan, daraja yuqorida, indeks pastda, integralning yuqori va quyi chegaralari esa an'anaviyl joyida turadi.
- Matematik ifodalar, grafiklar va ularga izoh sifatida beriladigan matnlarni ekranning ixtiyoriyl qismiga yozish mumkin.
- Tengsizliklar va tenglamalar sistemasini yechish biriktirilgan algoritmgaga ega.
- MathCAD muhitida "dasturni" tuzish va ularni bajarilish jarayoni parallel kechadi. Foydalanuvchi MathCAD hujjatiga yangi ifoda kiritar ekan, uning qiymatini birdaniga hisoblash va ifodani kiritishda yo'l qo'yilgan yashirin xatoliklarni ko'rish imkoniga ega bo'ladi.
- MathCAD paketi yetarli darajada qudratli matematik apparat bilan qurollanganki, ular orqali tashqi protseduralarni chaqirmasdan turib, paydo bo'ladigan muammolarni hal qilishimiz mumkin.
- MathCAD paketi matematik va fizik-kimyoviy formulalar hamda o'zgarmlarga asoslangan yordamchi qo'llanmalar bilan boyitilgan.
- MathCAD paketida turli sohalar bo'yicha elektron darsliklar yaratish mumkin. Masalan: oddiyl differentsial tenglamarni yechish, statistika,

termodinamika, boshqaruv nazariyasi, materiallar qarshiligi va boshqalar bunga misol bo'la oladi.

- MathCAD muhitida foydalanuvchi o'z oldiga qo'yilgan masalani yechish bilan cheklanibgina qolmay, fizik-kimyoviy masalalarni yechishda o'lchamlarni hisobga olish imkoniyatiga ega.

- MathCAD muhiti animatsiya vositasi bilan qurollangan, bunda tuzilgan modellarni nafaqat statik (o'zgarmas), balki dinamik (animatsion kliplar) holda yaratish mumkin.

- MathCAD muhiti belgili matematika elementlari bilan boyitilgan bo'lib, bunda masalani nafaqat sonli yechish, balki analitik usulda ham yechishga imkoniyat yaratilgan.

- MathCAD muhitidan chiqmagan holda boshqa serverdagi hujjatlarni ishlatish va Internet tavsiya qiladigan yuqori axborot texnologiyalari imkoniyatlaridan foydalanish mumkin.

MathCAD muhiti:

- Qatorlar yig'indisi va ko'paytmasini hamda iteratsiyalarni hisoblash;
- Funksiyalar qiymatini hisoblash;
- Hosila, differensial va integrallarni hisoblash;
- Hisoblashlarni simvol rejimida bajarish;
- Dekart, qutb, silindr koordinatalar sistemalarida ikki va uch o'lchovli grafiklarni, gistogrammalarni qurish;
- Trigonometrik, giperbolik, eksponentsial va Bessel funktsiyalar mavjudligi;
- Statistik funktsiyalar, chiziqli regressiya, Eyler gamma funktsiyalari, integral xatoligi, taqsimlangan ehtimollik funktsiyalari bilan ta'minlangan;
- Chiziqli va chiziqli bo'lmagan algebraik tenglama va tenglamalar sistemasini yechish;
- Oddiy differensial tenglama va tenglamalar sistemasini (Koshi masalasi va chegaraviy masala) yechish;
- Xususiy hosilali differensial tenglamalarni yechish;

- Ma'lumotlarga statik qayta ishlov berish (interpolyatsiya, ekstrapolyatsiya, approksimatsiya, animatsiya va ko'pgina boshqa amallar);
- Vektor va matritsalar bilan ishlash (Chiziqli algebra va boshqalar);
- Funksional bog'liqlikning maksimum va minimumini izlash kabi masalalarni oson, ko'rgazmali hal qilishga imkon beradi.

MathCAD dasturida hisoblashlar

MathCAD muhitida sonli hisoblashlar.

Barcha matematik sistemalarda matematik ifoda markaziy tushuncha bo'lib hisoblanadi. U sonlar ko'rinishida (ayrim hollarda simvollar ko'rinishida) nima hisoblanishi kerak ekanligini belgilaydi. Masalan: $12+3$; $2.5-1+2*3-\sin(x)$; $\cos(\pi)$.

Matematik ifodalar - sonlar, konstantalar, o'zgaruvchilar, operatorlar, funksiyalar va har xil maxsus belgilar yordamida tuziladi.

Sonlar – ifodaning miqdor qiymatlarini o'zida aks ettiruvchi eng sodda ob'ektdir. Agar sonning nomi qiymatiga mos kelsa, uni konstanta deb atash mumkin. Sonlar butun, kasr, belgilangan yoki suzuvchi nuqtali bo'lishi mumkin. Ularni mantissasi va tartibi orqali ham ko'rsatish mumkin. Masalan: 0.2; -3; 2.399; 0.00001; -123.465E-21; 345.657E12.

MathCADda sonning mantissasida butun qism kasr qismidan vergul orqali emas, balki nuqta orqali ajratiladi. Sonning mantissasini tartibidan ajratish uchun "E" simvoli ishlatiladi. Musbat sonlarning oldiga ishora qo'yilmaydi, manfiy sonlarning oldiga esa "-" (minus) ishorasi qo'yiladi.

Sonlar kompleks bo'lishi mumkin: $z=\text{Re}(x)+\text{Im}(x)*i$. Bunday sonlar haqiqiy $\text{Re}(x)$ va mavhum $\text{Im}(x)$ qismlardan iborat bo'ladi. Sonlarning mavhum qismi -1 dan kvadrat ildizga teng bo'lgan i yoki j ko'paytiruvchiga ega bo'ladi: $a+bi$.

Konstanta – yagona nomga ega bo'lgan va oldindan aniqlangan sonli yoki simvolli qiymat. Sonlar (masalan: 1.6, 13, -67, ...) nomsiz sonli konstantalar hisoblanadi. Simvolli konstanta – apostroflar orasiga olingan simvollar, masalan, "MathCAD". Agar apostroflar orasiga matematik ifoda olingan bo'lsa, u hisoblanmaydi, balki simvollar ketma-ketligi deb qaraladi.

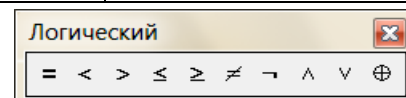
Operatorlar – bu ma'lumotlar (operandlar) ustida bajariladigan ma'lum amallarni ifodalovchi maxsus belgilash.

Mantiqiy amallar:

Not $\neg$	And $\wedge$	Or $\vee$	Xor $\otimes$
$0 \neg = 1$	$0 \wedge 0 = 0$	$0 \vee 0 = 0$	$0 \otimes 0 = 0$
$1 \neg = 0$	$0 \wedge 1 = 0$	$0 \vee 1 = 1$	$0 \otimes 1 = 1$
	$1 \wedge 0 = 0$	$1 \vee 0 = 1$	$1 \otimes 0 = 1$
	$1 \wedge 1 = 1$	$1 \vee 1 = 1$	$1 \otimes 1 = 0$

Munosabat va mantiqiy amallarini

“Логический” panelidan olish mumkin.



Arifmetik ifoda qiymatini hisoblash.

MathCADda oddiy sonli ifodalar ustida hisoblashlarni amalga oshirish mumkin. Hisoblashlarni amalga oshirish uchun matematik ifodalar klaviatura yoki “Калькулятор” panelidan foydalangan holda kiritiladi. Kiritiladigan ifodalar o'rni MathCAD dasturi ishchi oynasiga qizil rangli “+” belgi (kursor) ko'rsatadi. Kursor o'rni belgilash uchun sichqoncha ko'rsatkichi kerakli joyga olib boriladi va chap tugma bir marta bosiladi. Kursorni “↓↑→←” klavishlar yordamida siljitish mumkin.

MathCADda ishlashning xususiyatlari:³

- Ifodalalar kitobda qanday yozilsa, MathCADda ham shunday yoziladi.
- Ifodalar kiritilayotganda kiritish maydonchasi deb nomlangan to'g'ri to'rtburchak ko'rsatiladi.

- MathCADda turli arifmetik amallar qatnashgan ifodani qavslarsiz kiritish mumkin. Chunki, MathCAD arifmetik amallarni bajarish tartibini biladi.

- “=” belgisi yozilgandan keyin MathCAD natijani chiqaradi.

- Kiritilgan ifodalarni ixtiyoriy vaqtda tahrirlash mumkin.

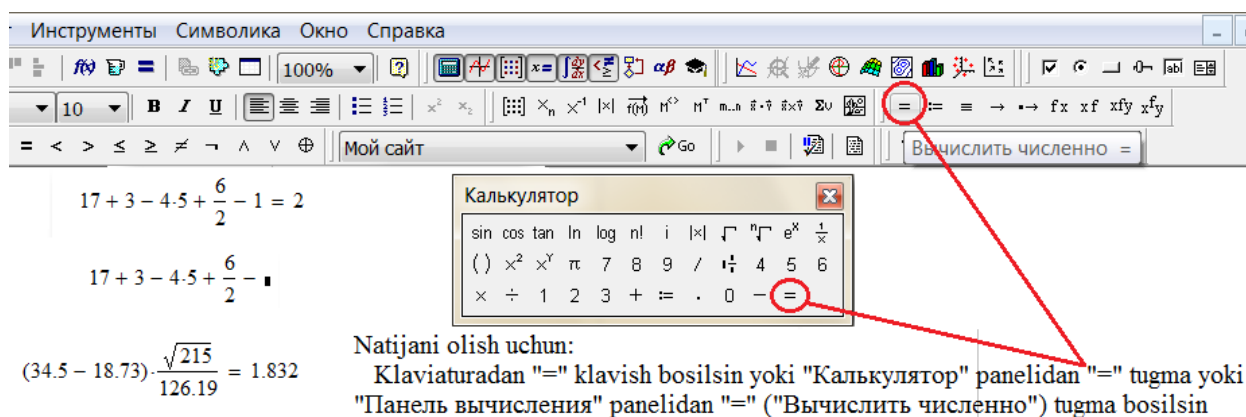
Misol 1. $17+3-4*5+6:2-1$ ifodani MathCADga kiriting va natijani oling.

1. “+” kursorni kerakli joyga qo'ying.

³ MATHCAD User's Guide with Reference Manual. MathSoft Engineering & Education, Inc. Springer-Verlag Berlin Heidelberg, 2001.-513p

2. Klaviaturadan $17+3-4*5+6/2-1$ ifodani kiriting.

3. “=” klavishni bosning.



Natijani olish uchun:
Klaviaturadan “=” klavish bosilsin yoki “Калькулятор” panelidan “=” tugma yoki “Панель вычисления” panelidan “=” (“Вычислить численно”) tugma bosilsin

Ifoda kiritilayotganda, belgi kiritishi kerak bo’gan joyda “■” ko’rinish paydo bo’ladi. O’nli kasrlarni kiritishda “,” emas, balki “.” ishlatiladi.

Matematik ifodani kiritishni boshlaganimizda, birinchi belgini kiritsak to’rtburchakli soha paydo bo’ldi. Kursor esa ko’k rangli, belgi ostida va undan keyin chiziq ko’rinishini  oldi. Soha ichida $17+3-4.5-■$ kursorni o’ng yoki chap tomonga siljitishimiz mumkin. Kursorda chapda turgan belgini “BackSpace”, o’ng tomondagi belgini esa “Delete” klavishasi bilan $17+3-4.5-\frac{6}{2}$ o’chirishimiz mumkin. Oxirgi kiritilgan belgini “Стандартная” panelidagi  “Отменить” orqali bekor qilish yoki  “Вернуть” orqali tiklash mumkin.

Oddiy kasrni kiritgach “Probel” yoki “→” klavishasi bosilsa, ko’k rangli kursor kattalasadi, yani u kasr qiymatini kiritish holatidan chiqadi.

MathCADda sin, cos, tan, log, ! (factorial) singari standart matematik funktsiyalarga qiymat berib, natijani olish mumkin. Ular “Калькулятор” panelida berilgan.

$\sin(1) = 0.841$	$\ln(4) = 1.386$	$e^4 = 54.598$	$\sqrt{65-3+4^2} = 8.832$
$\sin(\pi) = 0$	$\log(4) = 0.602$	$\frac{1}{6} = 0.167$	$\sin(\pi) + \cos(\pi) = -1$
$\cos(1) = 0.54$	$10! = 3.629 \times 10^6$	$4^5 = 1.024 \times 10^3$	
$\cos(\pi) = -1$	$ -4 = 4$		
$\tan(3) = -0.143$	$\sqrt{97} = 9.849$		
$\sqrt[3]{29} = 3.072$	$\pi = 3.142$	$5\frac{8}{17} + \sqrt{77} - 8 + \frac{(1761-83)}{61} = 33.754$	

O'zgaruvchilar.

O'zgaruvchilar – qiymatlari har xil bo'lgan ma'lumotlarni saqlovchi nomga ega bo'lgan ob'ektlardir. Bunday ma'lumotlarga mos holda o'zgaruvchilar sonli yoki simvolli, vektorli yoki matritsali bo'lishi mumkin.

Dasturlash tillaridagidek MathCADda ham o'zgaruvchilarni qo'llash va ularga ma'lum qiymatlarni berish mumkin. Buning uchun “:=” o'zlashtirish amalidan foydalaniladi: O'zgaruvchi_nomi:=ifoda.

O'zgaruvchilarni aniqlash va qo'llashda:

1. O'zgaruvchilar nomi ingliz, grek va rus alfaviti harflar, sonlar va “'”, “_”, “∞” belgilar kombinatsiyasidan tashkil topadi.
2. Nom son bilan boshlanmaydi.
3. MathCAD bir xil katta va kichik harfli nomlarni turli o'zgaruvchi sifatida qaraydi. Masalan, As va as turli nomli o'zgaruvchi sifatida qaraladi.
4. Ifodadan oldin unda ishtirok etuvchi o'zgaruvchilar e'lon qilinishi va ularga boshlang'ich qiymat berilishi shart.
5. MathCAD ifodani chapdan o'ngga qarab o'qiydi. Shu uchun ifodada qatnashuvchi o'zgaruvchilar o'ngda yoki yuqorida joylashishi lozim.
6. ∞ belgi o'zgaruvchi nomining faqat birinchi belgisi bo'lishi mumkin. Masalan, ∞a.
7. Nomlashda quyi indeks ishlatilishi mumkin va indeksdan keyin “.” qo'yilishi lozim. Masalan, A₁₁.

a o'zgaruvchini aniqlash uchun a: kiritish lozim. Natijada a:=■ hosil bo'ladi. Bo'sh maydonchaga o'zgaruvchi qiymati kiritiladi. Shu bilan x o'zgaruvchini aniqlash va uning qiymatini kiritish tugaydi.

Masalan, a:=3, ∞a:=3, as:=5, A₁₁:=4. Bu yerda := belgisi o'zlashtirish operatori vazifasini bajaradi, yani = dan o'ng tomondagi qiymatni = dan chap tomondagi o'zgaruvchiga o'zlashtiradi. O'zgaruvchini kiritgach “.” klavishni bossak, avtomatik ravishda “:=” paydo bo'ladi. O'zgaruvchilarga qiymat bergach, undan o'ng tomonda yoki pastda o'zgaruvchilar ishtirok etgan arifmetik ifodani

kiritib, “=” ni kiritsak, natijani olamiz. Masalan, $a + \infty a - as + A_{11} =$ deb kiritsak, 5 natijani olamiz.

$$a := 3 \quad \infty a := 3 \quad as := 5 \quad A_{11} := 4 \quad a + \infty a - as + A_{11} = 5$$

Shuningdek, o'zgaruvchilarni kiritgach, standart funksiyalarni kiritib, uning argumenti sifatida ushbu o'zgaruvchilarni kiritgach, “=” ni bossak, mos natijani olamiz. Masalan, $x := 5$ va $\ln(x) =$ deb kiritsak, 1.609 natijani olamiz. Bir o'zgaruvchini e'lon qilib, unga boshlang'ich qiymatlarni bergandan so'ng yana shu o'zgaruvchini e'lon qilib, unga boshqa boshlang'ich qiymatlarni bersak, bu holda o'zgaruvchi oxirgi kiritilgan qiymatni qabul qiladi. Masalan, x o'zgaruvchiga $x := 4$ kiritib, keyin $x := 5$ kiritsak x o'zgaruvchi 5 qiymatni oladi.

O'zgaruvchilarning turlari oldindan belgilanmaydi. Ular turi o'zgaruvchi tomonidan o'zlastiriluvchi qiymat yoki ifodaga mos holda aniqlanadi. Agar ifoda vektor yoki matritsa bo'lsa, o'zgaruvchi ham vektor yoki matritsa bo'ladi.

MathCADda o'zgarmas kattaliklar nomi kiritilib “=” bosilsa, avtomatik ravishda ushbu kattalik qiymati chiqariladi. Masalan, $\pi := 3.142$, $e := 2.718$.

Ifodani hisoblash $a := 7 \quad b := 8$ $7 + 8 = 15$ $a + b = 15$ $a + b \frac{8}{4} - \frac{a + b}{3} = 12$	Sonli o'zgarmaslar $e = 2.718 \quad \pi = 3.142$ $\frac{\pi e^2}{4} = 5.803$
Standart funksiyalar	
$\sin(a) = 0.657$ $\tan(b) = -6.8$ $ \tan(b) = 6.8$	$\cos(b) = -0.146$ $\ln(a) = 1.946$ $\sqrt{a} = 2.646$
$\sqrt[3]{b} = 2$ $\frac{1}{a} = 0.143$	$\ln(a) + \frac{e^a + \log(b)}{\sin(a)} = 1.673 \times 10^3$ $e^a = 1.097 \times 10^3$

Diskret o'zgaruvchilar.

Diskret o'zgaruvchilar deganda qiymati ma'lum bir qadam bilan o'suvchi yoki kamayuvchi sonlar ketma-ketligidan iborat bo'lgan o'zgaruvchilar tushuniladi. Masalan: $x = 0, 1, 2, 3, 4, 5$ yoki $x := [0.10]$. Bu shuni bildiradiki, o'zgaruvchi qiymati 0 dan boshlanadi va 1 qadam bilan o'sib boradi, ya'ni $x = 0, 1, 2, 3, 4, 5, \dots$ sonlardir.

Masalan, $x := [0.10]$ diskret o'zgaruvchilarni aniqlash uchun:

1. O'zgaruvchi nomi “x” yoziladi va “:=” belgi kiritiladi. Natijada $x := \blacksquare$ ko'rinish hosil bo'ladi.

2. Diskret o'zgaruvchi intervalining boshlang'ich qiymatini kiriting. Natijada $x:=1$ ko'rinish hosil bo'ladi.

1) “Панель векторов и матриц” paneldan $m..n$ “Область переменной” tugmani yoki “.” klavishni bosish orqali “.” belgi kiritiladi. Natijada $x:=1..$ ko'rinish hosil bo'ladi.

2) Diskret o'zgaruvchi intervalining oxirgi qiymatini kiriting.

Hisoblangan qiymatni chiqarish uchun esa o'zgaruvchi va tenglik belgisini kiritish kifoya. Natijada o'zgaruvchi qiymati ketma-ket shaklda jadvalda chiqadi. Masalan, $x:=0..10$ deb yozib, keyin $x=$ kiritish kerak.

$x := 1..10$	$y := 1,1.5..6$	$y =$	$ds := -11,-9..10$	$ds =$
$x =$				
1		1		-11
2		1.5		-9
3		2	+	-7
4		2.5		-5
5		3		-3
6		3.5		-1
7		4		1
8		4.5		3
9		5		5
10		5.5		7
		6		9

Agar diskret o'zgaruvchini $x:=0..10$ deb yozsak, qadam 1 deb olinadi. Agar diskret o'zgaruvchini $y:=1,1.5..6$ deb yozsak, qadam 0.5 deb olinadi. Bu yerda diskret o'zgaruvchining birinchi kiritilgan soni 1, ikkinchi kiritilgan son 1.5 dan ayiriladi va olingan natija qadam deb hisoblanadi. Ya'ni $1.5-1=0.5$ qadam hisoblanadi.

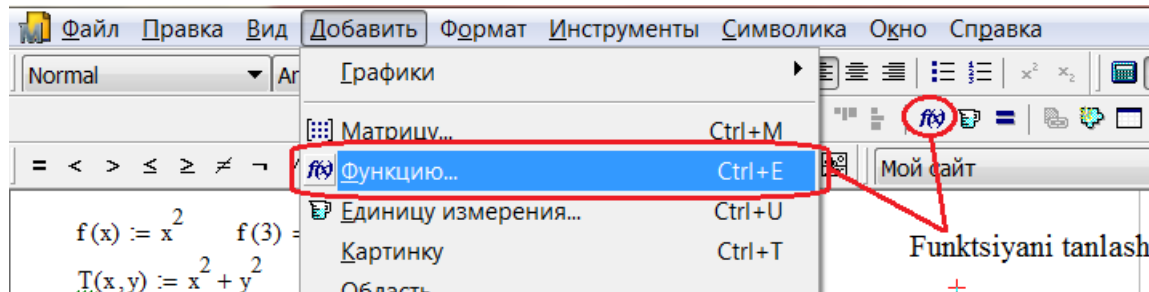
O'zgaruvchining nomi amalda cheklanmagan uzunlikdagi simvollardan iborat bo'lishi mumkin, ammo uni 25 tadan oshirmaslikni tavsiya qilamiz. Shuningdek, har qanday o'zgaruvchining nomi boshqa o'zgaruvchilarning, funksiyalarning nomi bilan bir xil bo'lmasligi kerak.

MathCAD dasturida funksiyalar masteri bilan ishlash

Funksiyalar – bu o'z argumentlarini ma'lum tarzda o'zgartirishni amalga oshiruvchi va bu o'zgartirishlar natijasini qaytaruvchi yagona nomga ega bo'lgan ob'ektlardir. Natijani qaytarish – funksiyaning o'ziga xos xususiyatidir.

Funksiya umumiy holda qavs ichiga olingan argumentlar ro'yxatiga (parametrlarga) ega bo'ladi. Masalan, $f(x)=x+1$. Bu yerda f funksiya bir argumentlidir. Funksiyalar biriktirilgan (ichki) va foydalanuvchi (tashqi) funksiyalari ko'rinishida bo'lishi mumkin. Biriktirilgan funksiyalarga keng tarqalgan elementar funksiyalar ($\sin(x)$, $\cos(x)$, $\tan(x)$ kabilar) misol bo'la oladi. Foydalanuvchi funksiyasi esa foydalanuvchi tomonida kiritilgan ixtiyoriy funksiya hisoblanadi.

MathCAD juda ko'p biriktirilgan funksiyalar to'plamiga ega. MathCADda mavjud bo'lgan funksiyalar ro'yxati menyuning “Добавить-Функцию” yoki “Стандартная” panelidagi  “Вставить функцию” tugmani bosish orqali ochiladi.

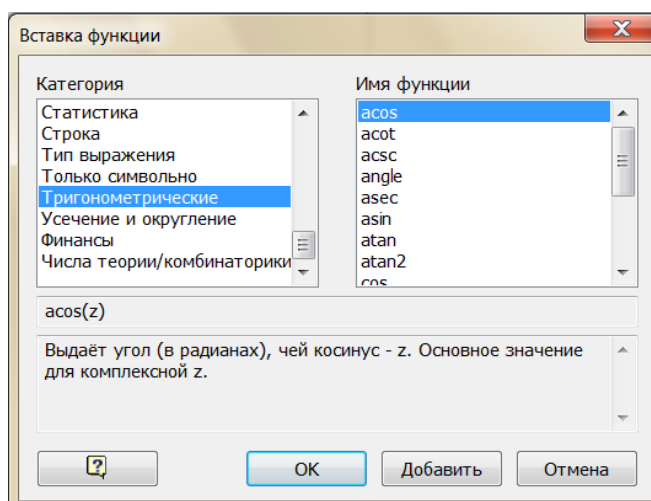


Mavjud funksiyalar guruhlarga ajratilgan va ular “Вставка функции” oynasining “Категория” sohasida ko'rsatiladi. Biror bir guruh tanlansa, ushbu guruhga kiruvchi funksiyalar ro'yxati “Имя функции” sohada ko'rsatiladi. Sohadan kerakli funksiya tanlangach “OK” tugma bosilsa, funksiya ishchi oynaga yoziladi.

MathCADda foydalanuvchi o'z funksiyasini ham qo'llash mumkin. Foydalanuvchi funksiyasini aniqlash 4 qismdan iborat bo'ladi:

1. *Funksiya nomi.* Funksiyani nomlash ham xuddi o'zgaruvchini nomlashdek amalga oshiriladi.

2. *Funksiya argumenti*
(argumentlari). Funksiya argumentini nomlash ham xuddi o'zgaruvchini nomlashdek amalga oshiriladi. Agar funksiya bir necha argumentlardan iborat bo'lsa, ular bir-biridan “,” orqali ajratib ko'rsatiladi.



3. O'zlashtirish belgisi. O'zlashtirish “:=” ni kiritish orqali amalga oshiriladi. Ekranda “:=” paydo bo'ladi.

4. Ifoda. U konstanta, o'zgaruvchi, funktsiyalarni chaqirish (standart va foydalanuvchi funktsiyalari) va matematik amallarni saqlashi mumkin. Shuningdek, o'zlashtirish belgisidan keyin dasturlash matni ham bo'lishi mumkin.

Masalan, $f(x)=x^2$ funktsiyani qanday aniqlashni ko'rib chiqamiz.

1) $f(x)$: ni tering, natijada $f(x):=■$ ko'rinish hosil bo'ladi.

2) x^2 ni tering, natijada $f(x):=x^2$ funksiya hosil bo'ladi.

Bu yerda f funksiya nomi, x esa funksiya argumenti (x funktsiyaning formal parametr deyiladi. Funktsiyani aniqlashda uning qiymatini berish shart emas). Funktsiyaning ixtiyoriy nuqtadagi qiymatini hisoblash mumkin, ya'ni, $f(3)=9$, $f(5)=25$, $f(4)=16$. Bu yerda funktsiyaga berilayotgan 3, 5 va 4 qiymatlar funktsiyaning haqiqiy parametri deyiladi. Xuddi shu tartibda ikki argumentli, uch argumentli va n argumentli funktsiyani aniqlash mumkin. Masalan, ikki argumentli funksiya quyidagicha aniqlanadi: $T(x,y):=x^2+y^2$, $T(2,2)=8$, $T(4,5)=41$.

Misol 4. $f(x)=x^2$, $T(x,y)=x^2+y^2$ va $g(x) = 40x + 5$ funktsiyalarni kiriting va turli haqiqiy parameterlarda ularning natijasini oling.

$$\begin{array}{llll}
 f(x) := x^2 & f(3) = 9 & f(5) = 25 & f(4) = 16 \\
 T(x,y) := x^2 + y^2 & T(2,2) = 8 & T(4,5) = 41 & \\
 g(x) := 40x + 5 & & + & \\
 g(3) = 125 & g(3.5) = 145 & \frac{g(3)}{5} = 25 & g(3) + f(3) = 134
 \end{array}$$

Foydalanuvchi funksiyasida takroriy yoki iteratsion hisoblashlarni amalga oshirish uchun uning argumenti sifatida o'zgaruvchilarni va diskret o'zgaruvchilarni qo'llash mumkin. Masalan, $f(x)=x^2$ funksiyani va $t:=0..5$ diskret o'zgaruvchini yozib, $f(t)=$ ni kiritsak, 1,3,9,16,25 natijani vertikal jadval shaklida olamiz.

$$f(x) := x^2 \quad k := 4 \quad t := 1..5 \quad T(x,y) := x^2 + y^2$$

$f(3) = 9$ $f(k) = 16$ $T(2,2) = 8$ $T(4,5) = 41$

t =	f(t) =
1	1
2	4
3	9
4	16
5	25

MathCADda ixtiyoriy funksiyani e'lon qilishimiz yoki oldindan yaratilgan maxsus standart funksiyalardan foydalanishimiz mumkin. Masalan: $\sin(x)$, $\cos(x)$, $\ln(x)$ va boshqa funksiyalar.

$$x := 1 \quad k(x) := \sin(x) + \ln(x) + x^3 - 1 \quad t := 0.1, 0.3..1$$

$k(x) = 0.841$ $k(x)^3 = 0.596$

t =	k(t) =
0.1	-3.202
0.3	-1.881
0.5	-1.089
0.7	-0.369
0.9	0.407

Xuddi shuningdek, sonlar qatori yig'indisi, ko'paytmasi, integrallar, differentsiallar qiymatini hisoblash mumkin. Bu uchun “Панель вычисление” panelidan foydalaniladi.

Lokal va global o'zgaruvchilar.

Dasturlash tillarida lokal va global o'zgaruvchi tushunchasi mavjud, bu yerda ham bu tushuncha bor. Agar o'zgaruvchi $x:=1$ ko'rinishda aniqlansa, u lokal o'zgaruvchi bo'ladi. Agar o'zgaruvchi $k:=3$ ko'rinishda aniqlansa u global o'zgaruvchi bo'ladi. Lokal o'zgaruvchini aniqlash funksiya qiymatini hisoblashdan o'ngda yoki yuqorida bo'lishi shart. Global o'zgaruvchini esa ixtiyoriy joyda yozish mumkin.

MathCAD ishchi hujjatni tepadan pastga va chapdan o'ngga qarab o'qiydi. Yuqorida keltirilgan misolda, agar ifodani qiymatini hisoblashda o'zgaruvchilar ifodadan pastga e'lon qilingan bo'lsa, ifodani qiymatini hisoblashda xatolik yuz

beradi. Global o'zgaruvchilarda esa ifoda qayerda yozilishidan qat'iy nazar unda ta'sir qiladi.

MathCAD muhitida xatoliklar.

MathCAD dasturida xatoliklar diagnostikasi katta ahamiyatga ega. Kiritilayotgan buyruq va ifodalarni MathCAD tekshiradi va xatolar to'g'risida axborot yoki ogohlantirishlar beriladi. Agar ifodada xatolik ro'y bersa, xatolik holati qizil rangda ko'rsatiladi va xatolik sababi izoh ko'rinishida chiqariladi. Masalan,

$$a := 3 \quad b := 3 \quad \frac{1}{a - b} = \text{■ ■}$$

Деление на ноль.

Bu xabar nolga bo'lish mumkin emasligini anglatadi. Bu holda o'zgaruvchi qiymatini o'zgartirish bilan tuzatiladi.

Nazorat savollari:

1. MathCAD dasturi qachon va nima maqsadda yaratilgan?
2. MathCAD dasturi vazifasi nimadan iborat?
3. MathCAD da oddiy hisoblashlar qanday amalga oshiriladi?
4. MathCAD dasturida formulalar qanday ko'rinishda yoziladi?
5. MathCAD dasturida sonlar qanday beriladi?
6. MathCAD dasturining eng sodda arifmetik operatorlarini izohlang.
7. MathCAD dasturining munosabat operatorlarini izohlang.
8. MathCAD dasturining mantiqiy amallarini izohlang.
9. MathCAD dasturida arifmetik ifoda qiymatini hisoblash qanday amalga oshiriladi?
10. MathCAD dasturida o'zgaruvchilarni qo'llanishi haqida tushuncha bering.
11. MathCAD dasturida diskret o'zgaruvchilarni qo'llanishi haqida tushuncha bering.
12. MathCAD dasturida funktsiyalar masteri bilan ishlash tushuncha bering.
13. MathCAD dasturida o'zgaruvchi va funktsiyalarni aniqlashni tavsiflang.
14. MathCAD dasturining lokal va global o'zgaruvchilari haqida tushuncha bering.
15. MathCAD muhitida xatoliklar qanday ifodalanadi?

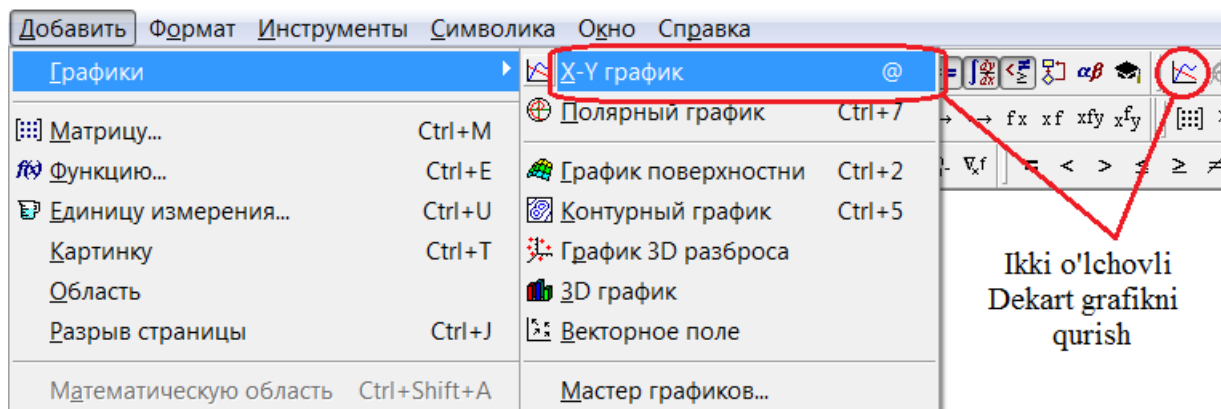
MATHCAD PAKETIDA IKKI VA UCH O'LCHAMLI GRAFIKLARNI QURISH.

Dekart koordinatalar sistemasida ikki o'lchovli grafiklarni qurish

MathCAD dasturi ixtiyoriy funksiyaning yoki diskret o'zgaruvchilarga bog'liq bo'lgan ifodalar grafiklarini chizish imkoniyatiga ega. Bundan tashqari bir nechta funksiyaning grafigini bitta sohada (koordinatada) tasvirlash mumkin. Chizmada har bir grafik diskret o'zgaruvchiga bog'liq bo'ladi. Bu diskret o'zgaruvchi ham absissalar o'qi uchun, ham ordinatalar o'qlari uchun ifodada qatnashishi kerak. MathCAD diskret o'zgaruvchilarning har bir qiymati uchun bitta nuqtani tasvirlaydi.

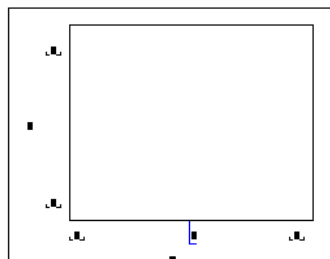
Har qanday bir argumentli funksiyaning grafigini Dekart koordinatalar sistemasida ikki o'lchovli grafigini chizish mumkin. Masalan, $f(x)=2(x-3)^2+x-8$ funksiya grafigini qanday chizish kerakligini ko'rib chiqamiz. Bu uchun quyidagi ishlar ketma-ket bajariladi.

1. $f(x)=2(x-3)^2+x-8$ funksiyaning MathCAD ishchi sohasiga kiriting.
2. Sichqoncha bilan grafik yasash joyini belgilang.
3. Menyudan “Добавить-Графика-Х-У график” buyrug'ini yoki “Панель графиков” panelidan “Х-У график” tugmani yoki “Shift+@” klavishni bosish.



Ekkranda grafik tekisligi sohasi paydo bo'ladi.

$$f(x) := 2(x-3)^2 + x - 8$$

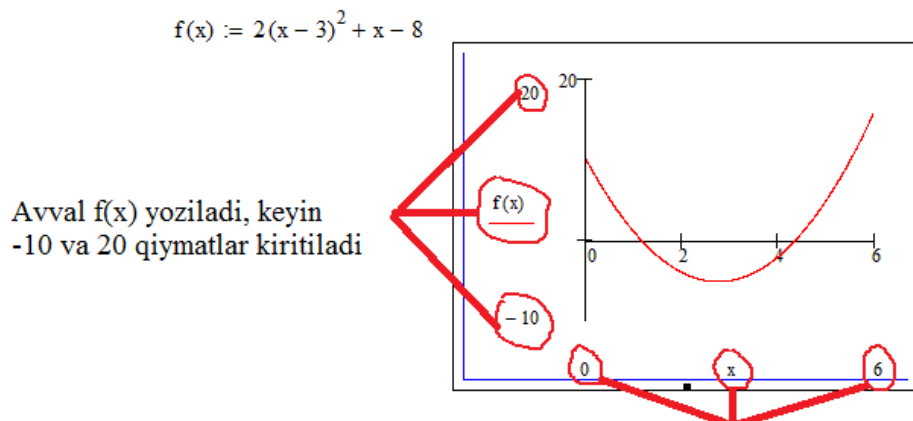


4. To'rtburchak soha ichidagi ■ belgili bo'sh joylarni to'ldiring. Gorizontall absissalar o'qining o'rtasidagi ■ belgili joyga funksiya argumenti x yoziladi va x

argumentning ikkala tomonidagi ■ belgili joylarga absissalar o'qi intervalining birinchi va oxirgi qiymatlari kiritiladi.

5. Vertikal ordinatalar o'qining o'rtasidagi ■ belgili joyga $f(x)$ yoziladi va $f(x)$ ning ikkala tomonidagi ■ belgili joylarga ordinatalar o'qi intervalining birinchi va oxirgi qiymatlari kiritiladi.

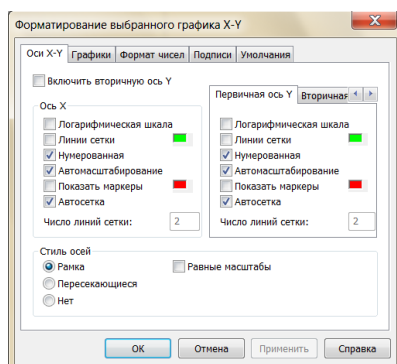
6. Keyin soha tashqarisida sichqoncha chap tugmasi bosilsa grafik paydo bo'ladi.



Grafik sohani boshqa joyga ko'chirish (to'rtburchak soha chegarasiga sichqoncha ko'rsatkichi olib borilgach, qo'l tasviri paydo bo'lganda, sichqoncha chap tugmasini bosib turgan holda siljitish bajariladi), sohani kattalashtirish yoki kichraytirish mumkin. Soha o'lchovini o'zgartirish uchun to'rtburchak soha o'ng yoki quyi chegarasidagi yoki quyi-ong burchakdagi ■ belgi ustiga sichqoncha ko'rsatkichi olib borilganda, ko'rsatkich ko'rinishi “↔” bo'lgach, sichqoncha chap tugmasini bosib turgan holda o'lchovni o'zgartirish bajariladi.

Dekart koordinatalar sistemasida ikki o'lchovli grafik formatlarini o'zgartirish

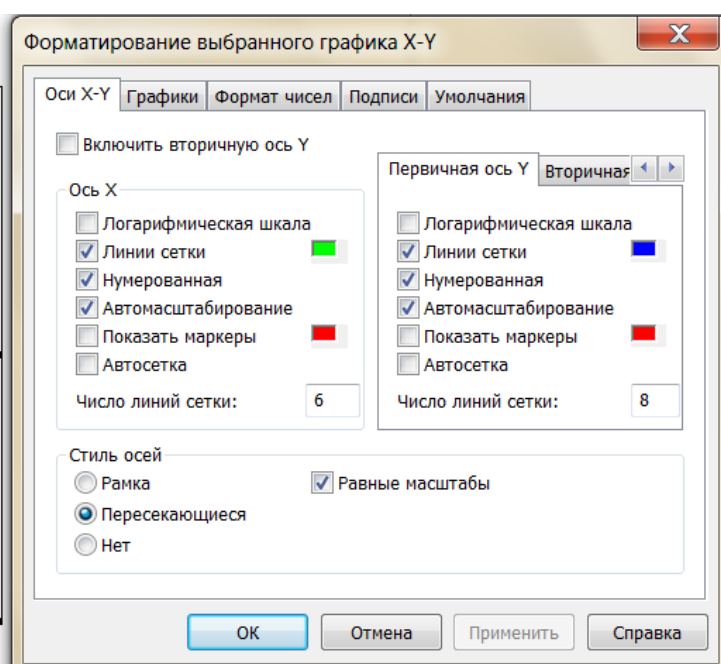
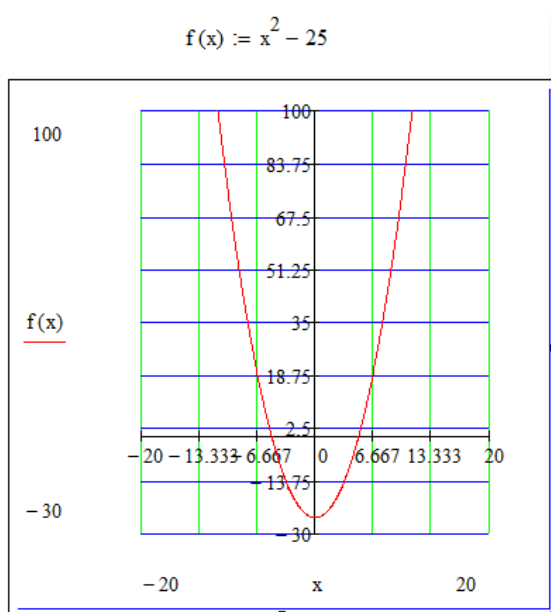
Grafik formatlarini o'zgartirish uchun sichqoncha ko'rsatkichi grafik soha ichiga olib borib, chap tugma ikki marta chiqillatilsa yoki sichqoncha ko'rsatkichi grafik soha ichiga olib borib, o'ng tugma bosilgach, ochilgan konteks menyudan “Формат” yoki menyuning “Формат-Графики-График X-Y” buyruq ishga tushirilsa “Форматирование выбранного графика X-Y” oyna ochiladi.



Оуна 5 та “Оси X-Y”, “Графики”, “Формат чисел”, “Подписи” va “Умолчания” sahifalaridan iborat.

Har bir sahifada parametr o'zgartirilgach, “Применить” tugma bosilsa, o'zgarish grafikka qo'llaniladi.

1. “Оси X-Y” sahifada x va y koordinata o'qlariga doir setka, sonli qiymatlar singari parametrlarni o'rnatish mumkin. Grafikda x ordinata o'qini [-20,20], y abtsissa o'qini esa [-30,100] interval deb o'zgartiramiz.



“Ось X” va “Ось Y” sohalar parametrlari:

- “Логарифмическая шкала” – koordinata o'qida logarifmik masshtabdagi sonli qiymatlarni tasvirlash;
- “Линия сетка” – koordinata o'qlariga masshtabli setkalar qo'yish va uning rangini ko'rsatish;
- “Нумерованная” – koordinata o'qi bo'yicha sonlarni qo'yish;
- “Автомасштабирование” – son qiymatlar chegarasini o'qda avtomatik masshtablash;
- “Показать маркеры” – koordinata o'qlari bo'yicha bo'lish, grafikka belgi kiritish va uning rangini ko'rsatish;

- “Автосетка” – chiziqli setkasi sonini avtomatik tanlash.

“Стиль осей” soha tanlovlari:

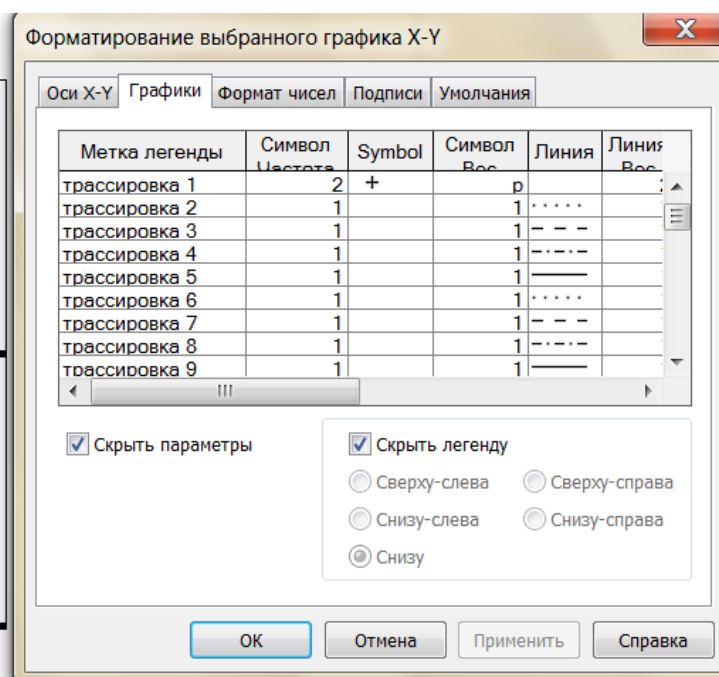
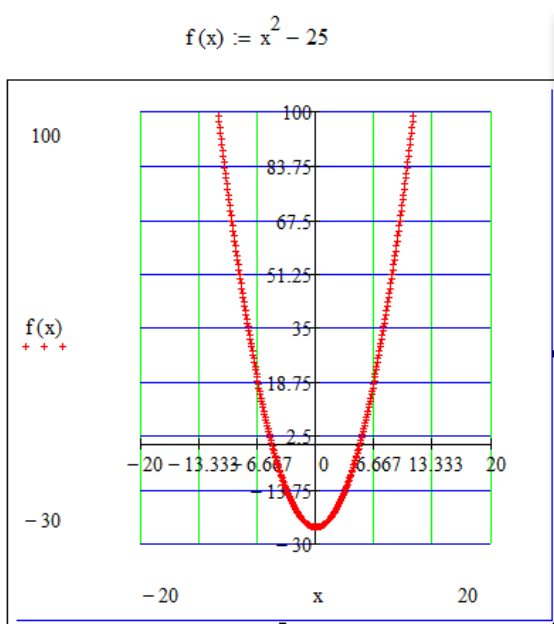
- “Рамка” – Chekli soha;
- “Пересекающиеся” – Dekart koordinata o'qlarini ko'rsatish;
- “Нет” – Chegarasiz;
- “Равные масштабы” – koordinata o'qida barcha sonlarni ko'rsatish.

Sahifada kerakli parametruga qayd qo'ygach, “Применить” tugma bosilsa, natija grafikda tasvirlanadi. “OK” tugma bosilsa, qaydlar tasvirlanib, oyna yopiladi.

2. “Графики” sahifada chizilgan grafika chizig'i qalinligi, turi va rangi singari parametrlarni o'rnatish mumkin.

“Графики” soha parametrlari:

- “Метка легенды” – grafikni tanlash (трассировка 1 – birinchi grafika);
- “Символ частота” – belgilar chastotasi;
- Symbol – grafikni chizishda qo'llaniladigan belgilar (– belgisiz, x, +, aylana, to'g'ri burchak, romb kabilar);
- “Символ все” – chiziqli juftlik soni;
- “Линия” - chiziqli ko'rinishi (uzliksiz, punktir, shtrixli va shu kabilar);
- “Линия все” – chiziqlar soni;

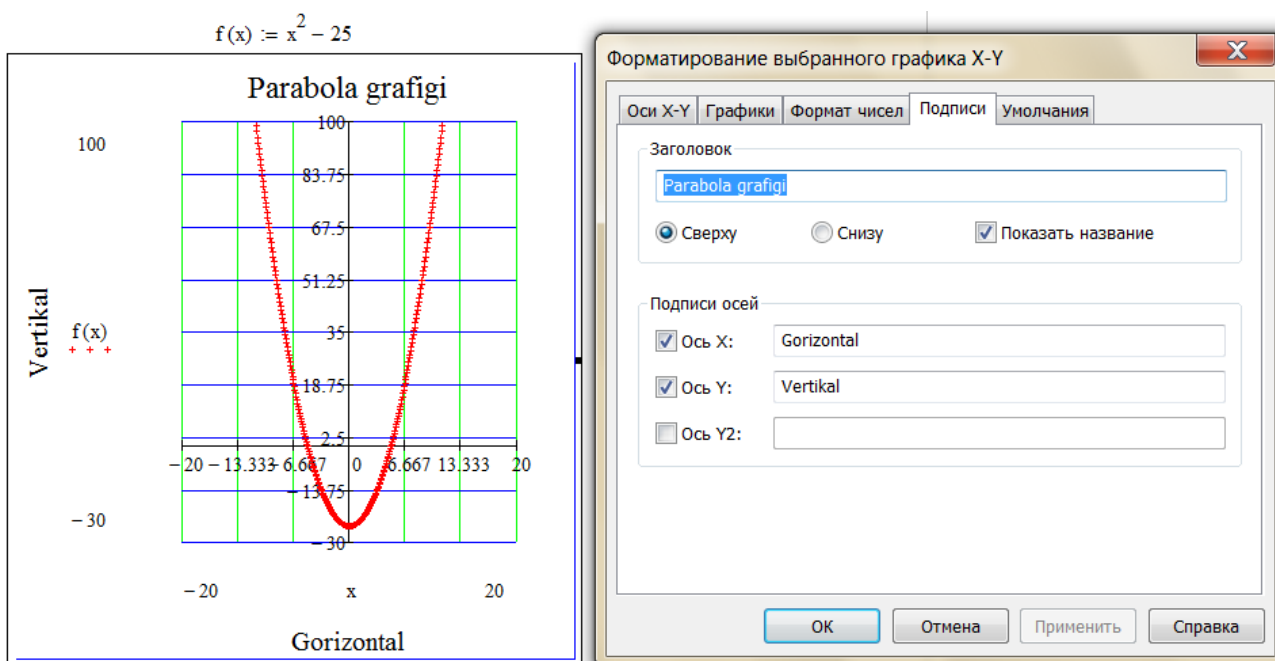


Mathcad paketida ikki va uch o'lchamli grafiklarni qurish.

- “Цвет” - chiziqli rangi (qizil, ko'k va shu kabilar);
- “Тип” - grafik tipi (chiziqli, nuqtali va shu kabilar);
- “Скрыть параметры” – parametrlarni yashirish;
- “Скрыть легенду” – grafika chegara chizig'ini yashirish (“Сверху-слева” – yuqori-chap, “Снизу-слева” – quyi-chap, “Снизу” - quyi, “Сверху-справа” - yuqori-o'ng, “Снизу-справа” – quyi o'ng).

3. “**Формат чисел**” grafika sonlari formatini alohida o'rnatish (ko'pchilik hollarda ishlatilmaydi).

4. “**Подписи**” – grafik nomi va koordinata o'qlariga izoh kiritiladi.

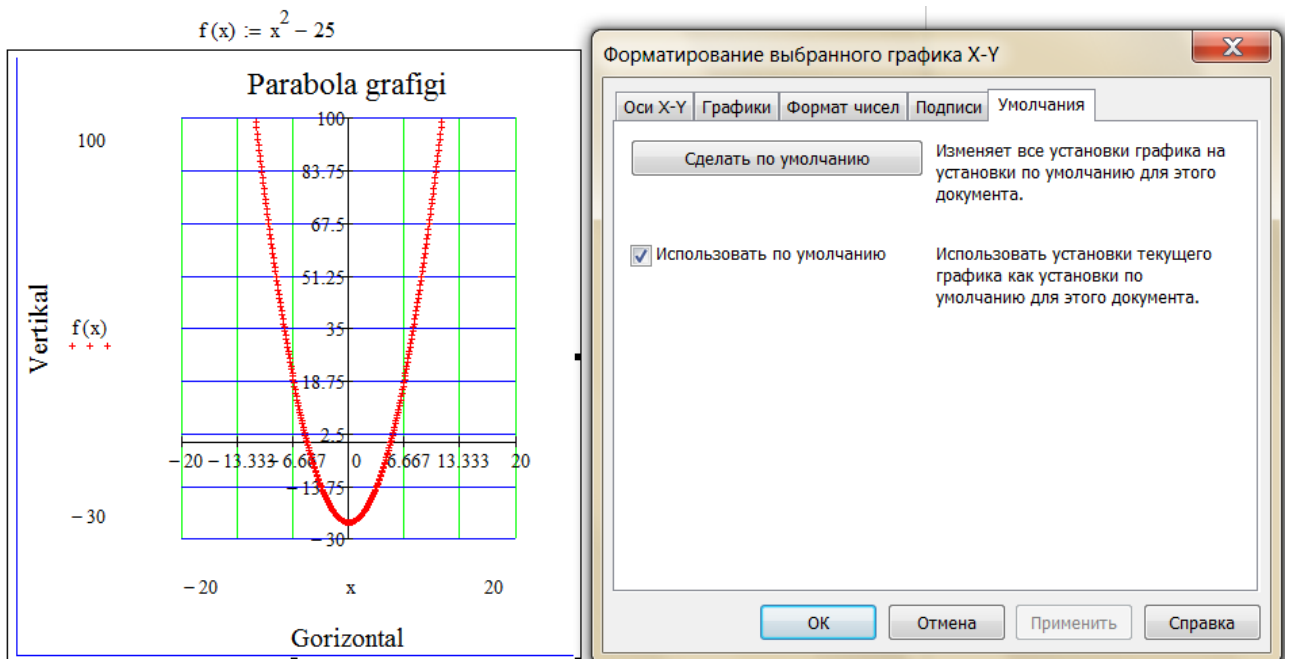


“Подписи” soha parametrlari:

- “Заголовок” maydoniga sarlavha matni kiritiladi. Sarlavha grafikning yuqorisida tasvirlanishi uchun “Сверху”, quyisida “Снизу” belgilanadi.
- “Подписи осей” sohaning “Ось X:” va “Ось Y:” maydonlariga koordinatalar o'qiga mos izoh matni kiritiladi.

Kiritilgan matnlar grafik sohasiga chiqishi uchun “Показать название” qayd qilinishi lozim.

5. “**Умолчания**” – o'zgartirilgan parametrlarni boshlang'ich jimlik holati sifatida qabul qilish yoki qilmaslik belgilanadi.



Ushbu formatlar qo'llanilsa grafik ancha ko'rgazmali bo'ladi.

Bir sohada bir nechta ikki o'lchovli grafiklarni qurish

Bir necha grafikni bitta sohada chizish mumkin. Bu uchun avval funksiyalar yoziladi, keyin esa ordinatalar o'qida funksiya nomi yoziladigan joyda funksiyalar bir-biridan “,” (vergul) bilan ajratib yoziladi.

Misol 2. $r(x)=x^4$, $f(x)=x^2-25$, $q(x)=x$, $g(x)=x^3-10$ va $k(x)=x^3+x^2+10$ funksiyalar grafigini bir sohada yasang.

Bir necha grafikni bitta sohada chizish uchun quyidagi ishlar ketma-ket bajariladi:

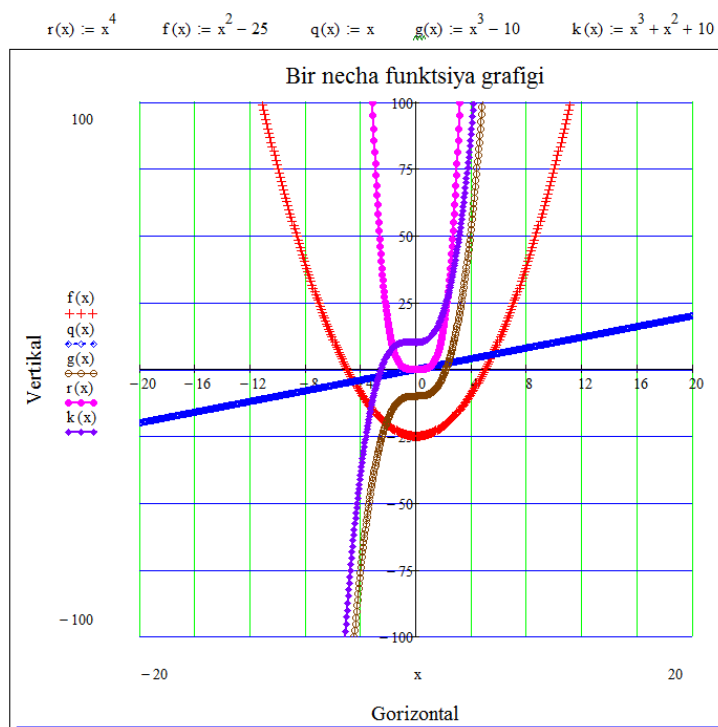
1. $r(x)=x^4$, $f(x)=x^2-25$, $q(x)=x$, $g(x)=x^3-10$ va $k(x)=x^3+x^2+10$ funksiyalarni ketma-ket MathCAD ishchi sohasiga kiriting.
2. Sichqoncha bilan grafik yasash joyini belgilang.
3. “Shift+@” klavishni bosing. Ekranda grafik tekisligi sohasi paydo bo'ladi.
4. To'rtburchak soha ichidagi gorizontalar absissalar o'qining o'rtasidagi ■ belgili joyga funksiya argumenti x yozing va x argumentning ikkala tomonidagi ■ belgili joylarga absissalar o'qi intervalining birinchi va oxirgi qiymatlarini kiriting.
5. Vertikal ordinatalar o'qining ikkala tomonidagi ■ belgili joylarga ordinatalar o'qi intervalining birinchi va oxirgi qiymatlarini kiriting.
6. Vertikal ordinatalar o'qining o'rtasidagi ■ belgili joyga $r(x)$ yozing.

Mathcad paketida ikki va uch o'lchamli grafiklarni qurish.

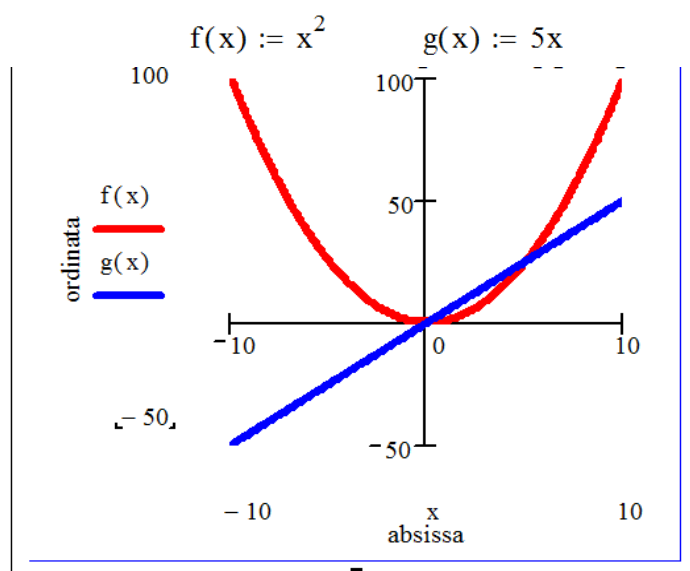
7. “,” ni kiriting va $f(x)$ yozing. Shu tartibda “ $q(x),g(x),k(x)$ ” yozing.

8. Keyin soha tashqarisida sichqoncha chap tugmasi bosilsa grafiklar paydo bo'ladi.

Grafiklar turli rangda chiziladi. Har bir grafikning formatini “Форматирование выбранного графика X-Y” oynani ochib, alohida belgilang.



Bitta sohada ikkita funktsiyani joylashtirish:



Berilgan intervalda ikki o'lchovli grafiklarni qurish

Funksiyani $[a,b]$ oraliqda grafigini chizish.

Biror $f(x)$ funksiya berilgan bo'lsin. Bu funksiyaning grafigini $[a,b]$ oraliqni n ta bo'lakka bo'lib chizish uchun i diskret o'zgaruvchi olib $[a,b]$ kesmani quyidagicha n ta bo'lakka bo'lamiz. h qadam sifatida $\frac{b-a}{n}$ ni olamiz va i diskret o'zgaruvchini quyidagicha aniqlaymiz: $i:=0..n$. x_i ni quyidagicha aniqlaymiz: $x_i:=a+h*i$ va bizga x_i va $f(x_i)$ nuqtalar hosil bo'ladi. Bu nuqtalarga mos funksiyaning grafigini chizish mumkin. Bu holda x va y o'qlarga intervalning boshlang'ich va oxirgi qiymatlari kiritilmaydi.

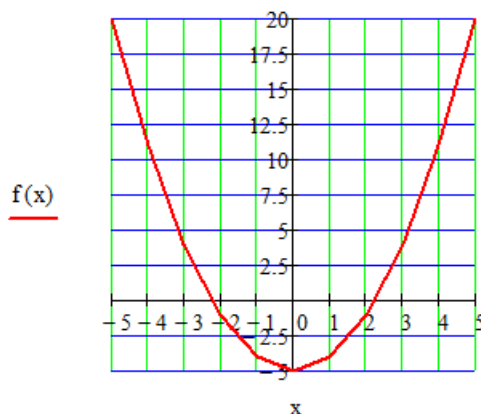
Misol 3. $f(x)=x^2-5$ funksiya grafigini $[a,b]$ intervalda $h=(b-a)/n$ qadamdagi nuqtalarda hisoblang va grafigini yasang. Bu yerda $a=-5$, $b=5$ va $n=10$.

Grafikni chizish uchun quyidagi ishlar ketma-ket bajariladi:

1. $a:=-5$, $b:=5$, $n:=0..n$ o'zgaruvchilarni ketma-ket kiriting.
2. $h:=(b-a)/n$, $x_i:=a+h*i$ ifodalarni va $f(x):=x^2-5$ funksiyaning ketma-ket kiriting.
3. Sichqoncha bilan grafik yasash joyini belgilang.
4. "Shift+@" klavishni bosing. Ekranda grafik tekisligi sohasi paydo bo'ladi.
5. To'rtburchak soha ichidagi gorizontaal abtsissalar o'qining o'rtasidagi ■ belgili joyga funksiya argumenti x yozing. (Intervalning boshlang'ich va oxirgi qiymatlari kiritilmaydi)
6. Vertikal ordinatalar o'qining o'rtasidagi ■ belgili joyga $f(x)$ yozing. (Intervalning boshlang'ich va oxirgi qiymatlari kiritilmaydi)
7. Keyin soha tashqarisida sichqoncha chap tugmasi bosilsa grafiklar paydo bo'ladi.
8. Grafikdan keyin "x=" kiritilsa, diskret nuqtalar qiymati chiqariladi.
9. " $f(x)=$ " kiritilsa, funksiyaning diskret nuqtalardagi qiymati chiqariladi.

$$a := -5 \quad b := 5 \quad n := 10 \quad i := 0..n$$

$$h := \frac{(b-a)}{n} \quad x_i := a + h \cdot i \quad f(x) := x^2 - 5$$



$$x =$$

	0
0	-5
1	-4
2	-3
3	-2
4	-1
5	0
6	1
7	2
8	3
9	4
10	5

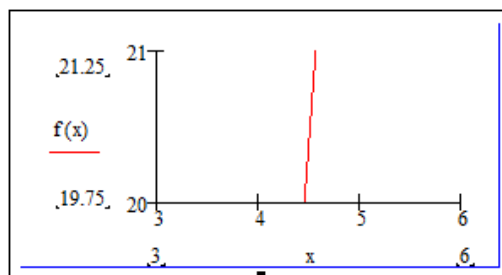
$$f(x) =$$

	0
0	20
1	11
2	4
3	-1
4	-4
5	-5
6	-4
7	-1
8	4
9	11
10	20

Misol 4. [3,6] oraliqda $f(x)=x^2$ funksiya grafigini chizing.

Grafikni chizganda avval funksiyani kiritib, keyin diskret qiymatlarni kiritisa ham bo'ladi. Shu uchun funksiyani $[a,b]$ oraliqda grafigini soddaroq chizish uchun quyidagilar bajariladi.

$$f(x) := x^2 \quad x := 3..6$$



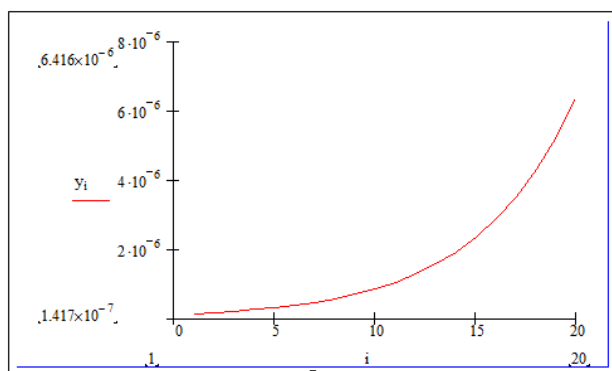
Bu yerda avval funksiyani kiritib, keyin diskret qiymatlarni berdik va ushbu oraliqda funksiya grafigini oldik.

Vektorni grafik shaklda ifodalash.

Vektor elementlarini ham grafik shaklda ifodalash mumkin. Uni yasash uchun quyidagilarni bajaring.

1. Chizilishi kerak bo'lgan element intervalini kiriting, ya'ni $i:=a..b, i \geq 0$.
2. y_i vektorni aniqlang, ya'ni $y_i := c^i \cdot d^{b-i}$.
3. y_i ni vertikal o'qning o'rtasidagi maydoniga, i ni esa gorizontal o'qning o'rtasidagi maydonga joylashtiring.
4. Grafikni ko'rish uchun F9 ni bosing yoki sichqonchani grafikdan tashqariga bosing.

$$i := 1..20 \quad y_i := 0.55^i \cdot 0.45^{20-i}$$



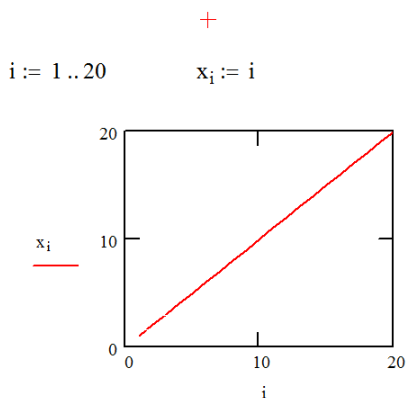
Vektorni grafigini olish:

1. $i := 1..20$ ni kiriting;
2. $y_i := 0.55^i \cdot 0.45^{20-i}$ ni kiriting;
3. "Shift+2" orqali grafik sohani oching;
4. Absissa x o'qining qora belgisi joyiga i ni yozing;
5. Ordinata y o'qining qora belgisi joyiga y_i ni yozing;
6. Grafik soha tashqarisida sichqoncha chap tugmasini bir marta bosib.

Quyidagi misolda funksiya sifatida quyi indeksli o'zgaruvchilar, ya'ni vektor ishlatilmoqda.

	0
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9

x =



MathCAD muhitida uch o'lchovli grafiklar

Dekart koordinatalar sistemasida uch o'lchovli grafiklarni qurish

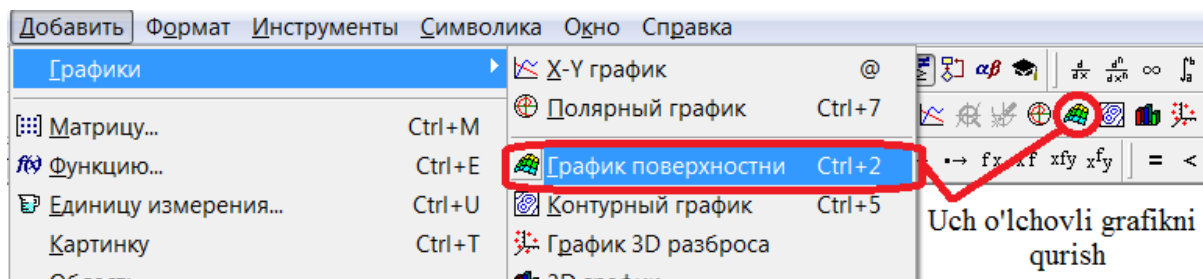
MathCAD dasturida uch o'chovli grafiklarni ham tasvirlash mumkin. Masalan, $f(x,y)=x^2+y^2$ funksiya grafigini qurish uchun quyidagi ishlar ketma-ket bajariladi.

1. $f(x,y)=x^2+y^2$ funksiyaning MathCAD ishchi sohasiga kiritilgan.

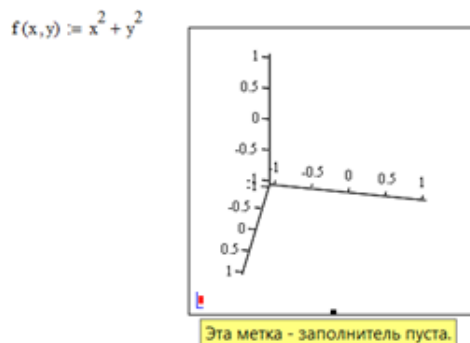
2. Sichqoncha bilan grafik yasash joyini belgilang.

3. Menyudan "Добавить-Графика-График поверхности" buyrug'ini yoki "Панель графиков" panelidan "График плоскости" tugmani yoki "Ctrl+2" klavishni bosib.

Mathcad paketida ikki va uch o'lchamli grafiklarni qurish.

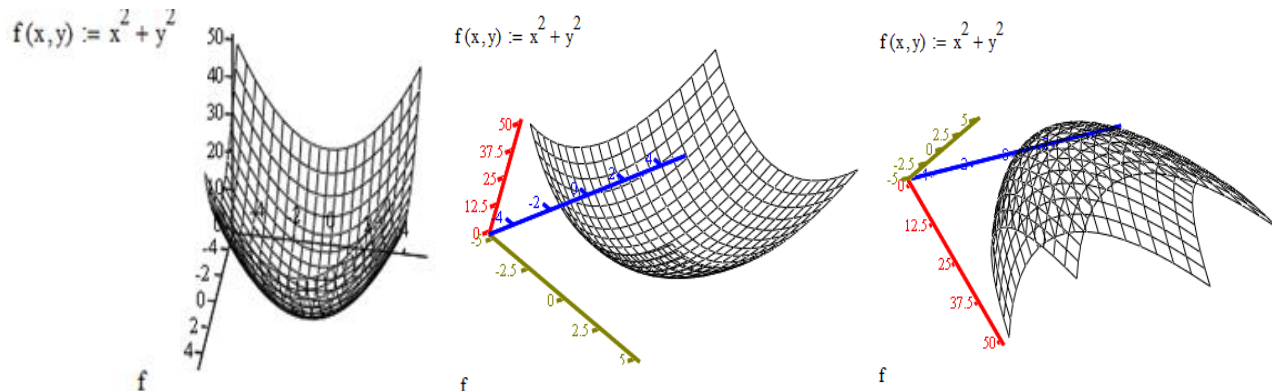


Ekranida grafik tekisligi sohasi paydo bo'ladi.



4. To'rtburchak soha ichidagi quyi-chap burchakdagi ■ belgili bo'sh joyga "f" ni kiriting.

5. Keyin soha tashqarisida sichqoncha chap tugmasi bosilsa grafik paydo



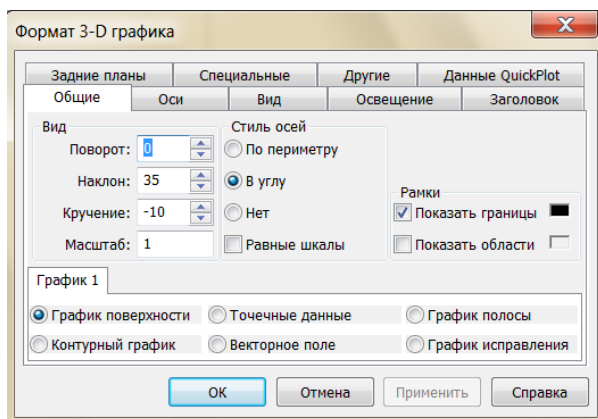
bo'ladi.

Grafik ustiga sichqoncha ko'rsatkichini olib borib, chap tugmani bosib turgan holda grafikni turli holatlarga burish mumkin. Shunigdek, uning o'lchovini o'zgartirish va siljitish mumkin.

Dekart koordinatalar sistemasida uch o'lchovli grafik formatlarini o'zgartirish

$f(x,y)=x^2+y^2$ funksiya grafigi formatlarini o'zgartirishni ko'rib chiqamiz. Grafik formatlarini o'gartirish uchun sichqoncha ko'rsatkichi grafik soha ichiga olib borib, chap tugma ikki marta chiqillatilsa yoki sichqoncha ko'rsatkichi grafik

soha ichiga olib borib, o'ng tugma bosilgach, ochilgan kontekst menyudan “Свойства” yoki menyuning “Формат-Графики-3D график” buyruq ishga tushirilsa “Формат 3-D графика” oyna ochiladi.

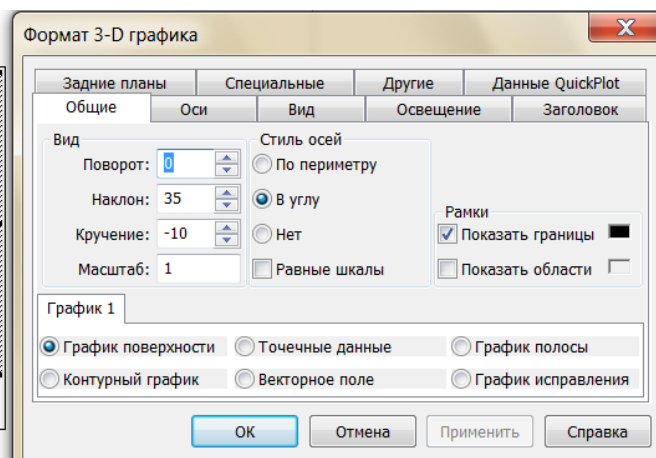
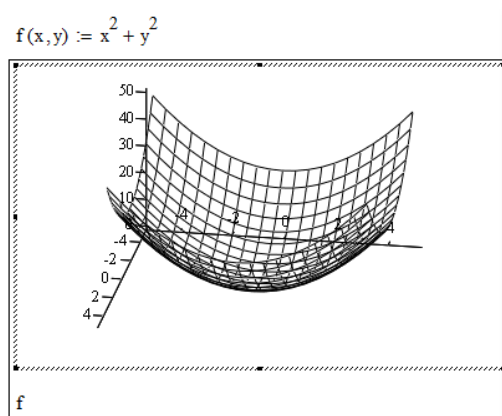


Ойна 9 ta “Общие”, “Оси”, “Вид”, “Освещение”, “Задные планы”, “Специальные”, “Другие”, “Данные QuickPlot” va “Заголовок” sahifalaridan iborat.

Har bir sahifada parametr o'zgartirilgach, “Применить” bosilsa,

o'zgarish grafikka qo'llaniladi.

1. **“Общие”** Bu sahifada grafikni burush burchaklari, tasvirlash turini o'zgartirish, chegaralarini tasvirlash singari formatlar o'rnatiladi.



“Вид” sohada:

- “Поворот” – tasvirni ma'lum bir burchakka gorizontal burish;
- “Наклон” – tasvirni ma'lum bir burchakka vertikal burish;
- “Кручение” - tasvirni ma'lum bir burchakka aylantirish;
- “Масштаб” – tasvirni kattalashtirish yoki kichiklashtirish.

“Стил осей” sohada:

- “По периметру” – perimetrga muvofiq koordinatalar o'qini joylashtirish;
- “В углу” – koordinatalar o'qini burchakka joylashtirish;

Mathcad paketida ikki va uch o'lchamli grafiklarni qurish.

- “Нет” - koordinatalar o'qisiz tasvirni ko'rsatish;
- “Равные шкалы” – tasvirni teng shkalalarda berish.

“Рамки” sohada:

- “Показать границы” – tasvir chegaralarini ko'rsatish;
- “Показать области” – barcha tomondan tashqi chegaralarini ko'rsatish.

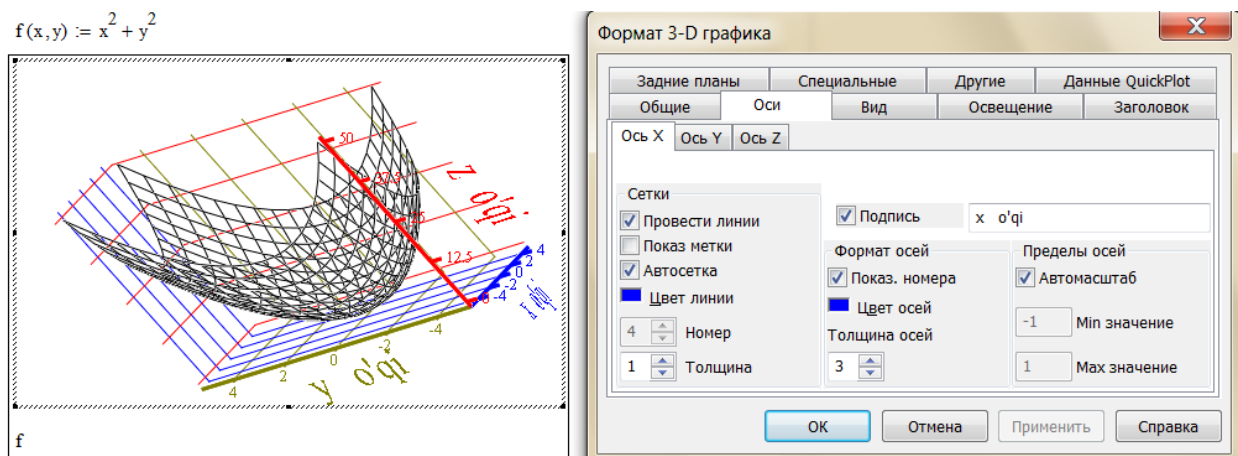
“График1” sohada:

- “График поверхности” – yuza grafik tasvir;
- “Контурный график” – konturli grafik tasvir;
- “Точечные данные” – nuqtali grafik tasvir;
- “Векторное поле” – vektorli grafik tasvir;
- “График полосы” – grafik nuqtalaridan koordinatalar yuzasiga chiziqlar

berilgan tasvir;

- “График исправления” - grafik nuqtalari alohida shaklda berilgan tasvir.

2. **“Оси”** sahifada x, y va z koordinata o'qlariga doir chiziq rangi, qalinligi, setka, sonli qiymatlar singari parametrlarni o'rnatish mumkin. Grafikda o'qlardagi interval avtomatik ravishda qo'yiladi.



“Ось X”, “Ось Y”, “Ось Z” sohalar parametrlari bir xil:

“Сетки” sohada:

- “Провести линии” – to'r chiziqlarini o'tkazish;
- “Показ метки” – metkalarni (qayd) ko'rsatish;
- “Автосетка” – avtoto'r qo'yish;
- “Цвет линии” – to'r chizig'i rangini o'rnatish;
- “Номер” – to'r nomeri

- “Толщина” - to'r chizig'i qalingini o'rnatish.

“Подпись” maydonga o'qga izoh yoziladi.

“Формат осей” sohada:

- “Показ. номера” – o'qlardagi nuqtalar raqamini ko'rsatish;
- “Цвет осей” - o'qlar chizig'i rangini belgilash;
- “Толщина осей” - o'qlar chizig'i qalinligini belgilash.

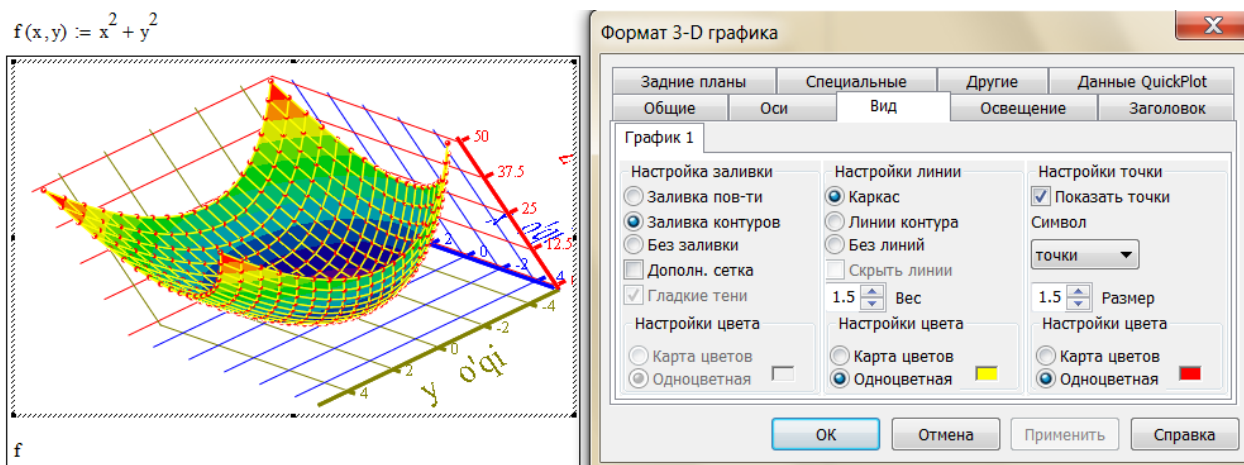
“Пределы осей” sohada:

- “Автомасштаб” – avtomatik interval belgilash;
- “Min значение” – intervalning minimal qiymatini ko'rsatish;
- “Max значение” - intervalning maksimal qiymatini ko'rsatish.

3. **“Вид”** sahifada “График1” sahifadagi grafikni (bir sohada bir necha grafik tasvirlangan holatlarda, har bir grafik alohida) uning chiziqlarini va nuqtalarini ko'rsatish, ranglash, qalinligini belgilash singari parametrlarni o'rnatish mumkin.

“Настройки заливки” sohasida:

- “Заливка пов-ти” – график yuzasini bir xil ko'rsatgan holda rangini tasvirlash;
- “Заливка контуров” – график yuzasini konturli rangini tasvirlash;
- “Без заливки” – график yuzasini bo'yalmagan holda bir xil tasvirlash;
- “Дополн. сетка” – qo'shimcha to'r o'rnatish;
- “Гладкие тени” – silliq soya berish.



“Настройки цвета” sohada:

- “Карта цветов” – графикni turli rangda bo'yash;

- “Одноцветная” - grafikni bir xil rangda bo'yash.

“Настройки линии” sohada:

- “Каркас” – grafik chiziqlarini karkasli ko'rsatish;
- “Линии контура” - grafik rang ajratilgan chiziqlarini ko'rsatish;
- “Без линии” - grafik chiziqlari kesishgan nuqtalarni tasvirlab, chiziqlarni

ko'rsatmaslik;

- “Скрыть линии” – – grafik chiziqlarini yashirish;
- “Вес” – chiziq qalinligi ravshanligini belgilash.

“Настройки цвета” sohada:

- “Карта цветов” – grafik chiziqlarini turli rangda bo'yash;
- “Одноцветная” - grafik chiziqlarini bir xil rangda bo'yash.

“Настройки точки” sohada:

- “Показать точки” – grafik chiziqlari keshishgan nuqtalarni ko'rsatish;
- “Символ” - grafik chiziqlari keshishgan nuqtalarda tasvirlanadigan

belgini ko'rsatish;

- “Размер” - grafik chiziqlari keshishgan nuqtalar o'lchovini o'rnatish.

“Настройки цвета” sohada:

○ “Карта цветов” – grafik chiziqlari keshishgan nuqtalarni turli rangda bo'yash;

○ “Одноцветная” - grafik chiziqlari keshishgan nuqtalarni bir xil rangda bo'yash.

4. **“Освещение”** sahifada grafikni turli xil ranglar hamohangligida yoritish singari parametrlarni o'rnatish mumkin.

“Освещение” sohada:

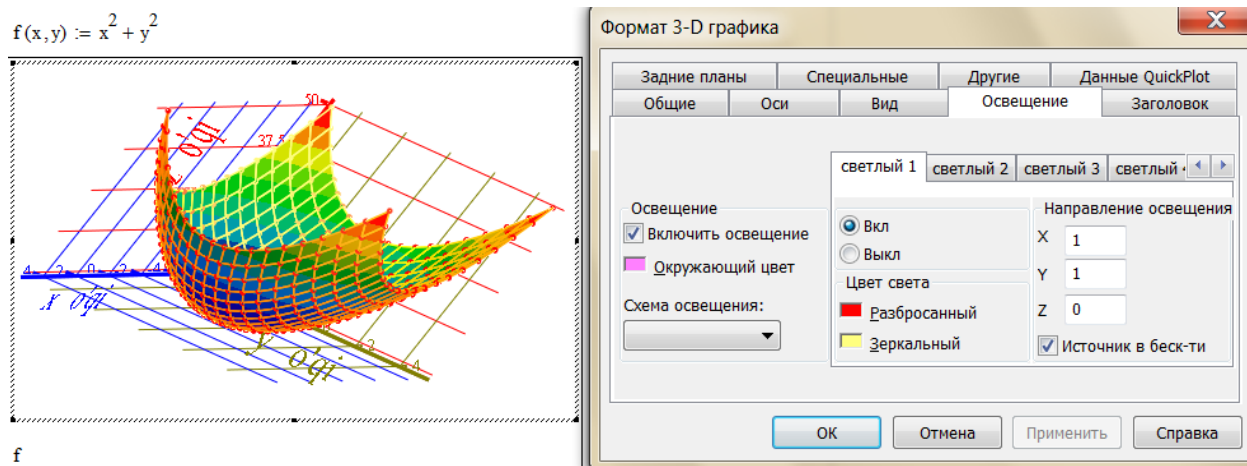
- “Включить освещение” – yoritishni parametrlarini qo'llash;
- “Окружающий цвет” – grafik atrofi rangini tanlash va qo'llash;
- “Схема освещение” – taqdim qilinayotgan parametrlarni tanlash va

qo'llash.

“Светлый 1” sohada:

- “Вкл” – yoritish parametrlarini qo'llash;

- “Выкл” – yoritish parametrlarini qo'llamaslik.



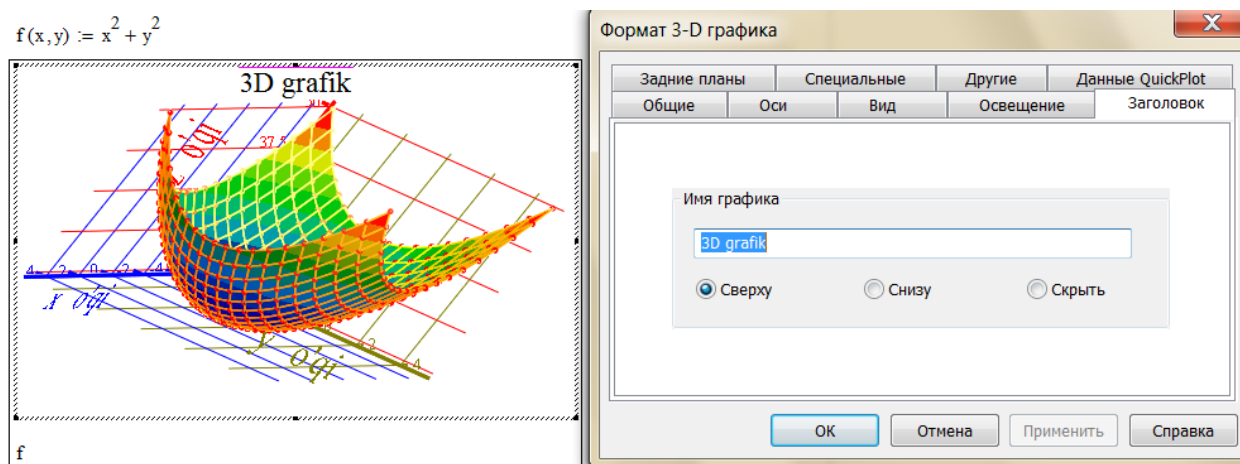
“Цвет света” sohada:

- “Разбросанный” – yoyilgan rangni o'rnatish;
- “Зеркальный” – tasvirli rangni o'rnatish.

“Направление освещения” sohada:

- “X” – X o'qi yo'nalishi parametrini o'rnatish;
- “Y” – Y o'qi yo'nalishi parametrini o'rnatish;
- “Z” – Z o'qi yo'nalishi parametrini o'rnatish;
- “Источник в беск-ти” – cheksizlik manbaini qo'llash.

5. “Заголовок” sahifada grafik nomini kiritish va uni yuqiri yoki quyida ko'rsatish amalga oshiriladi.



6. “Задние планы” sahifada “Оси” sahifasidagi x, y va z koordinata o'qlariga doir chiziq rangi, qalinligi, setka, sonli qiymatlar singari parametrlarga qo'shimcha parametrlarni o'rnatish amalga oshiriladi.

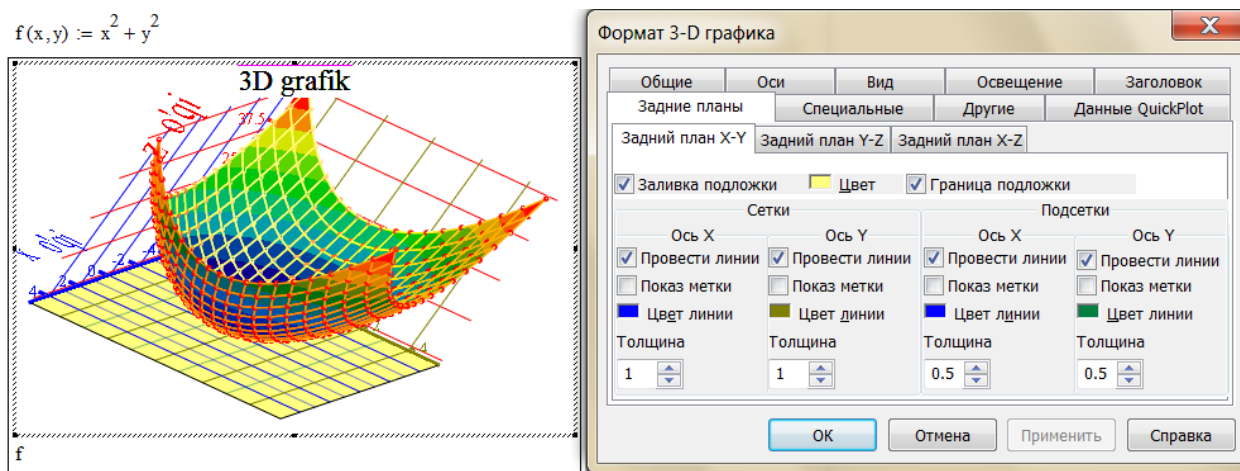
“Задний план X-Y”, “Задний план Y-Z”, “Задний план X-Z” sohalar parametrlari bir xil:

“Задний план X-Y” sohada:

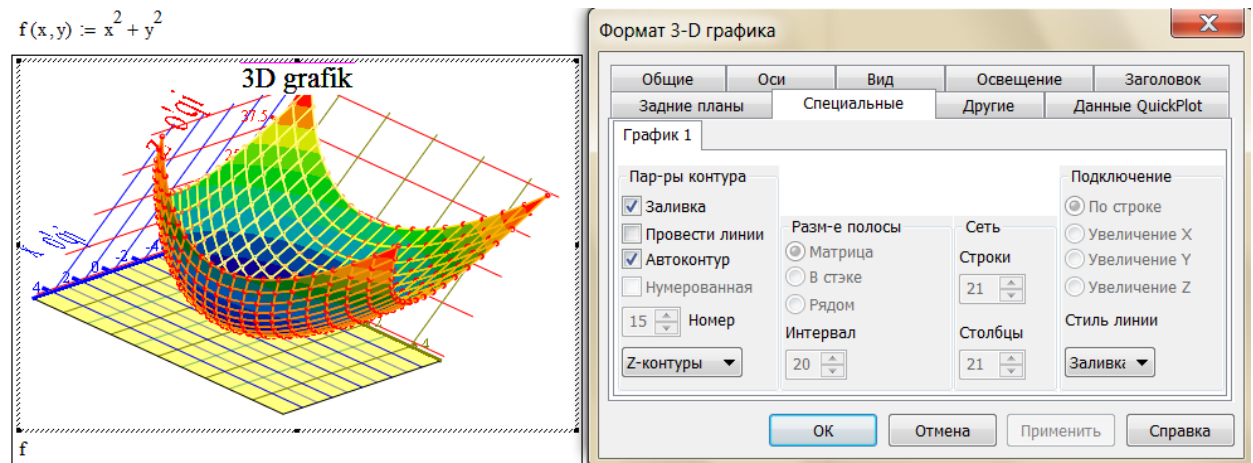
- “Заливка подложки” – X-Y soha yuzasini bo'yashni o'rnatish;
- “Цвет” – X-Y soha yuzasini bo'yash rangini o'rnatish;
- “Граница подложки” – X-Y soha yuzasi chegaralarini qo'yish.

“Сетки” (То'р) va “Подсетки” (То'rosti) sohalarda:

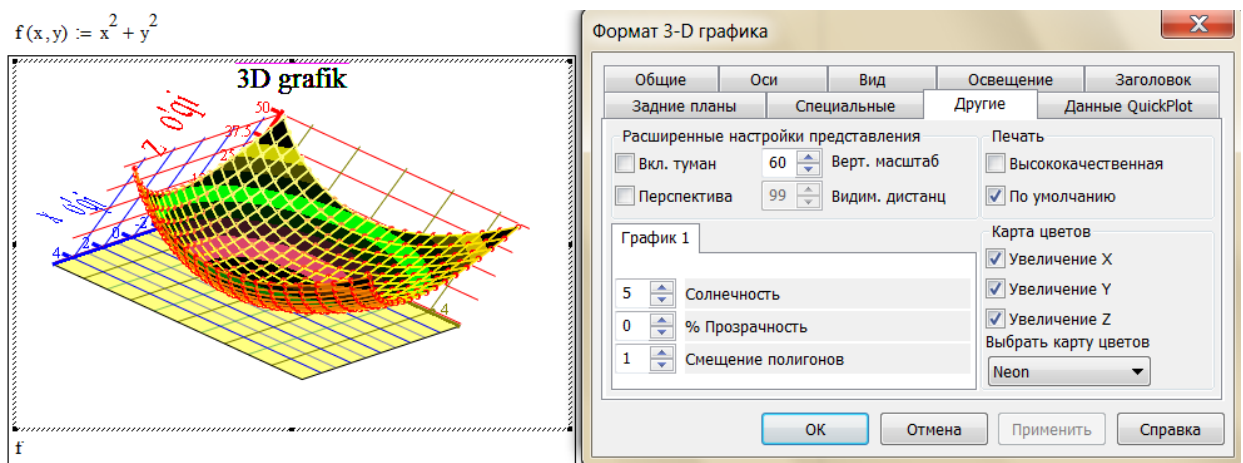
- “Провести линии” – to'r chiziqlarini o'tkazish;
- “Показ метки” – metkalarni (qayd) ko'rsatish;
- “Цвет линии” – to'r chizig'i rangini o'rnatish;
- “Толщина” – to'r chizig'i qalingini o'rnatish.



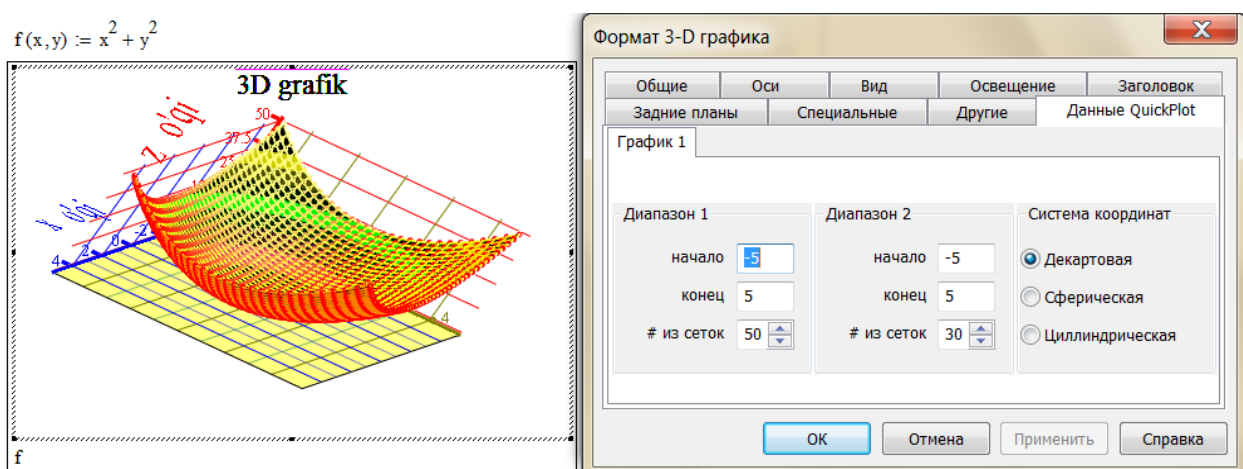
7. “Специальные” sahifada “Вид” sahifasidagi “График1” sahifasida keltirilgan grafikni (bir sohada bir necha grafik tasvirlangan holatlarda, har bir grafik alohida) uning chiziqlarini va nuqtalarini ko'rsatish, rahglash, qalinligini belgilsh singari parametrlarga qo'shimcha parametrlarni o'rnatish amalga oshiriladi.



8. “Другие” sahifada grafikni tumanlilik, vertikal masshtabni va ko'rish yaqinligini belgilash, yorug'lilik, shaffoflik, grafik rangi kartasini tanlash singari parametrlarni o'rnatish amalga oshiriladi.



9. “Данные QuickPlot” sahifada grafik to'ri nuqtalar sonini oshirish va grafikni Dekart yoki sferik yoki tsilindrik koordinatalar tizimida tasvirlash singari parametrlarni o'rnatish amalga oshiriladi.



“Диапозон 1” va “Диапозон 2” sohalarda:

- “начало” – grafik intervali boshlang'ich nuqtasini belgilash;
- “конец” - grafik intervali oxirgi nuqtasini belgilash;
- “# из сеток” - grafik to'ri chiziqlarini oshirish yoki kamaytirish.

“Система координат sohada:

- “Декартовая” – grafikni Dekart koordinatalar tizimida tasvirlash;
- “Сферическая” – grafikni sferik koordinatalar tizimida tasvirlash;
- “Цилиндрическая” - grafikni tsilindrik koordinatalar tizimida tasvirlash.

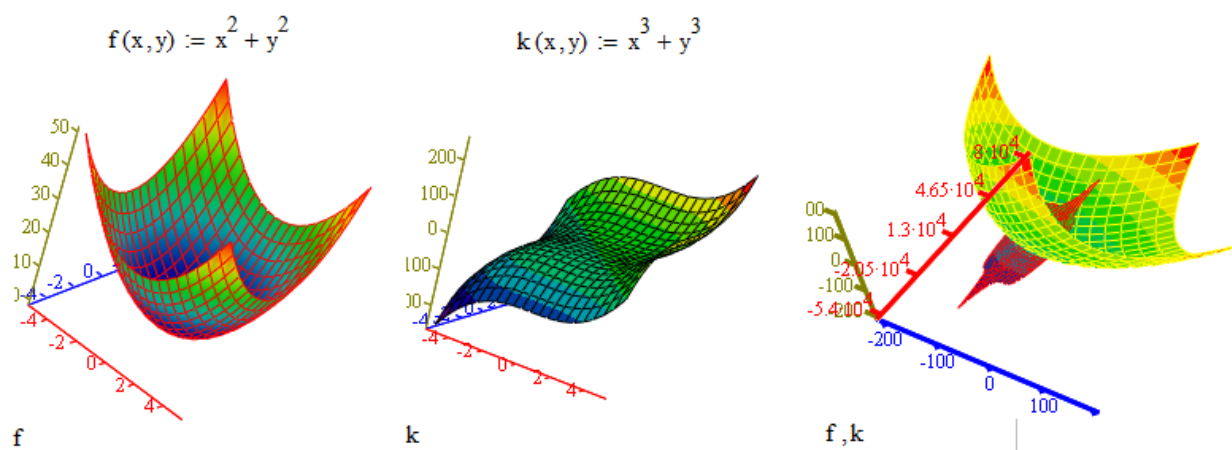
Bir sohada bir nechta uch o'lchovli grafiklarni qurish

Bir necha grafikni bitta sohada chizish mumkin. Bu uchun avval funksiyalar yoziladi, keyin esa grafik sohaning o'ng-quyi qismiga funksiya nomi yoziladigan joyda funksiyalar bir-biridan “,” (vergul) bilan ajratib yoziladi.

Misol. $f(x,y)=x^2+y^2$ va $k(x,y)=x^3+y^3$ funksiyalar grafigini bir sohada yasang.

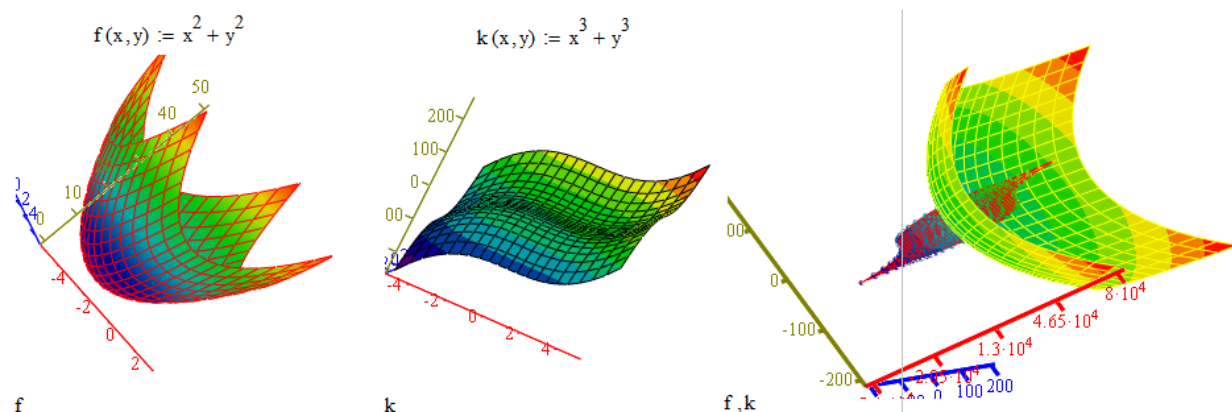
Bir necha grafikni bitta sohada chizish uchun quyidagi ishlar ketma-ket bajariladi:

1. $f(x,y)=x^2+y^2$ va $k(x,y)=x^3+y^3$ funksiyalarni ketma-ket MathCAD ishchi sohasiga kiriting.
2. Sichqoncha bilan grafik yasash joyini belgilang.
3. “Ctrl+2” klavishni bosing. Ekranda grafik tekisligi sohasi paydo bo'ladi.
4. To'rtburchak soha ichidagi quyi-chap burchakdagi ■ belgisi bo'sh joyga “f” ni kiritgach “,” kiritib, “k” ni kiriting.
5. Keyin soha tashqarisida sichqoncha chap tugmasi bosilsa, grafik paydo bo'ladi



Biz bir sohada yasalgan bir nechta uch o'lchovli grafik ko'rgazmali bo'lishi uchun avval har bir grafik tasvirini yaratdik.

Biz bir sohada yasalgan bir nechta uch o'lchovli grafiklarni boshqacha burilgan holati.



Berilgan intervalda uch o'lchovli grafiklarni qurish

Biror $f(x,y)$ funksiya berilgan bo'lsin. Bu funktsiyani grafigini mos ravishda $[a,b]$ oraliqni n ta bo'lakka bo'lib chizish uchun i diskret o'zgaruvchi olib $[a,b]$ kesmani n ta bo'lakka bo'lamiz. h qadam sifatida 1 ni olamiz. Masalan, o'zgaruvchini va funktsiyani quyidagicha aniqlaymiz:

$$n=20, i=[0,n], j=[0,n], x_i=i, y_j=j$$

$$z_{i,j}=\sin(x_i)+ y_j/2$$

Grafikni chizish uchun quyidagi ishlar ketma-ket bajariladi:

1. Kursor joyini belgilang.
2. $n:=20, i:=0..n, j:=0..n, x_i:=i, y_j:=j$ o'zgaruvchilarni ketma-ket kiriting.
3. $z_{i,j}:=\sin(x_i)+ y_j/2$ funktsiyani kiriting.

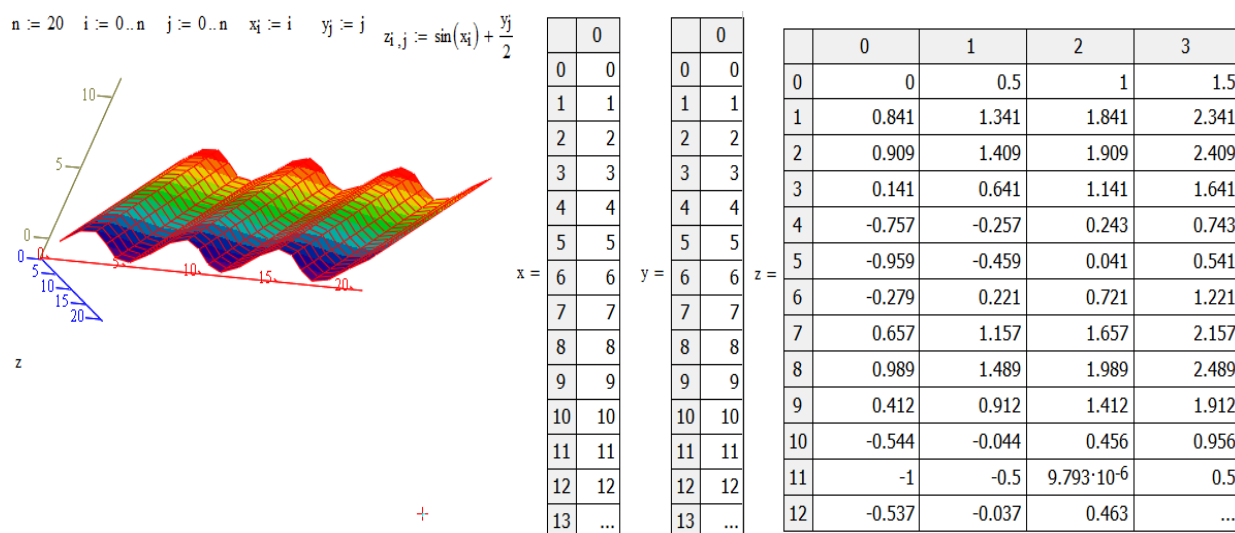
4. Sichqoncha bilan grafik yasash joyini belgilang va “Ctrl+2” klavishni bosing. Ekranda grafik tekisligi sohasi paydo bo'ladi.

4. To'rtburchak soha ichidagi quyi-chap burchakdagi ■ belgili bo'sh joyga “f” ni kiritgach “,” kiritib, “k” ni kiriting.

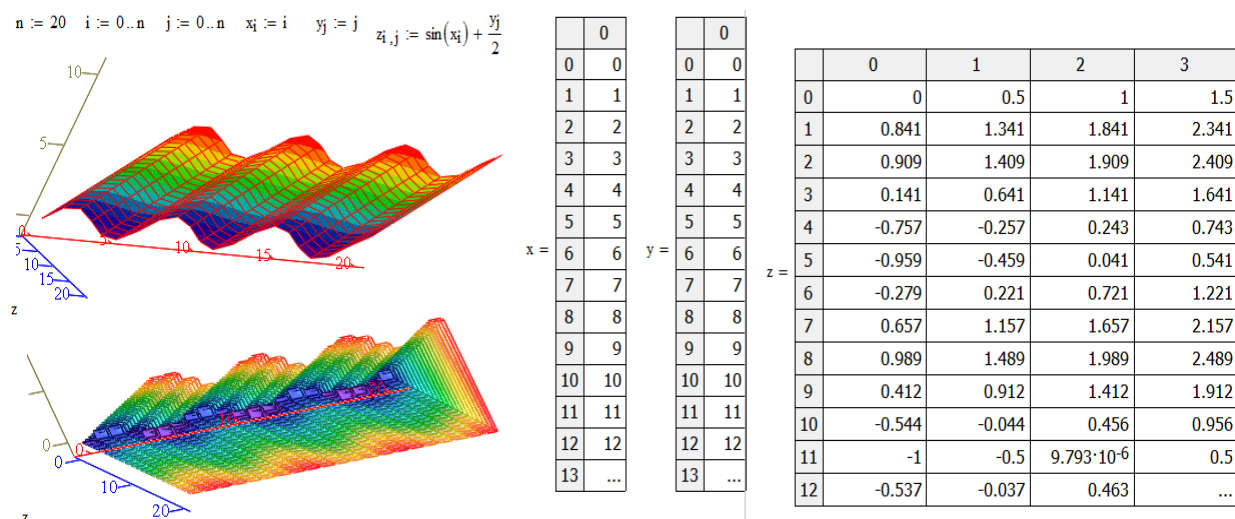
5. Keyin soha tashqarisida sichqoncha chap tugmasi bosilsa, grafik paydo bo'ladi

6. Grafikdan keyin “x=” va “y=” kiritilsa, diskret nuqtalar qiymati chiqariladi.

7. “z=” kiritilsa, funksiyaning diskret nuqtalardagi qiymati chiqariladi.



Yuqorida berilgan funksiyanı “Панель графиков” panelining “График 3D полс” tugmasini ishlatgan holda yasasak, quyidagi tasvir hosil bo'ladi.



MathCAD muhitida turli koordinatalar sistemasida grafiklarni qurish

Qutb koordinatalar tizimida ikki o'lchovli grafiklarni qurish

MathCAD muhitida nafaqat Dekart koordinatalar sistemasida, balki boshqa koordinatalar sistemasida ham grafiklarni qurish mumkin.

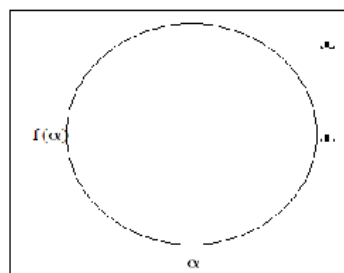
Qutb koordinatalar sistemasida ikki o'lchovli grafiklarni qurish uchun:

1. Funksiyaning burchak argumentlari kiritiladi.
2. Funksiya kiritiladi.
3. Menyudan “Добавить-Графика-Полярный график” buyrug'ini yoki “Панель графиков” panelidan “Полярный график” tugmani yoki “Ctrl+7” klavishni bosing.
4. Ochilgan to'rtburchak sohaga funksiya argumenti va nomini xuddi Dekart koordinatalar tizimi grafigini qurgandek qo'ying.

Masalan, $f(\alpha) = \cos 4\alpha$ funksiya grafigini $\alpha = [0; 2\pi]$ oralig'ida $h=0$ - $\pi/60$ qadamda chizing.

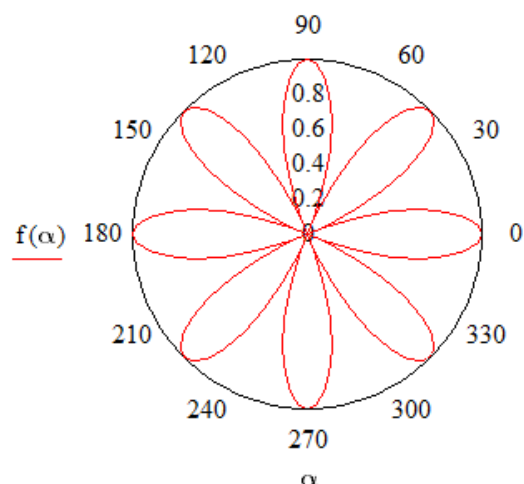
1. $\alpha := 0, \frac{\pi}{60} \dots 2\pi$ va $f(\alpha) := \cos(4\alpha)$ ni kiritamiz.
2. “Ctrl+7” klavishni bosib, grafik sohani ochamiz.
3. Grafik sohaga funksiya argumenti va nomini kiritamiz.

$$\alpha := 0, \frac{\pi}{60} \dots 2\pi \quad f(\alpha) := \cos(4\alpha)$$



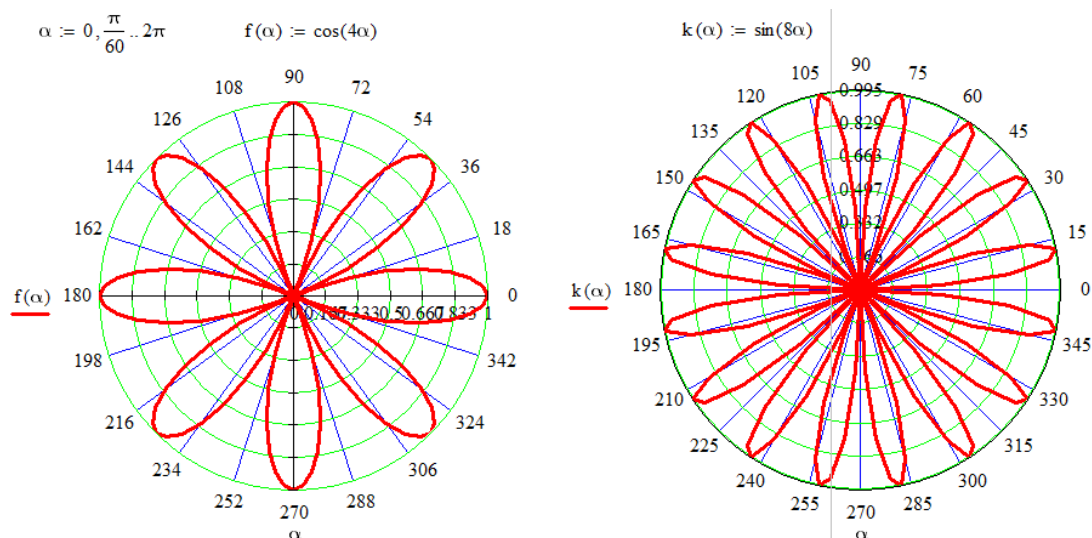
4. Soha tashqarisida sichqoncha tugmasi bosilsa, grafik hosil bo'ladi.

$$\alpha := 0, \frac{\pi}{60} .. 2\pi \quad f(\alpha) := \cos(4\alpha)$$



Ushbu grafik formatlarini o'zgartirish, xuddi Dekart koordinatalar sistemasida grafigi formatni o'zgartirishdek amalga oshiriladi.

Sohada yana $k(\alpha) = \sin(8\alpha)$ funktsiya grafigini $\alpha = [0; 2\pi]$ oralig'ida $h = 0 - \pi/60$ qadamda chizamiz. Ikkala grafik formatlarini o'zgartirsak quyidagicha tasvirni olamiz.



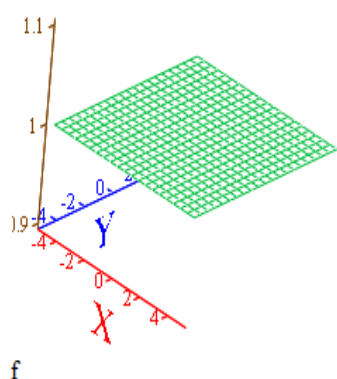
Quyidagi misol qutb koordinatalarini to'g'ri koordinatalarga almashtirishni tasvirlaydi. Bu metod qutb koordinatalarda yoki kompleks tekisliklarda yo'llar grafigini yasashga imkoniyat yaratadi.

Uch o'lchovli grafiklarni turli koordinatalar sistemasida qurish

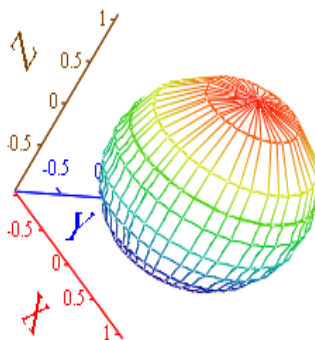
Misol 1. $f(x,y)=1$ funktsiya grafigini turli koordinatalar sistemasida quring.

Eng avvalo $f(x,y)=1$ funktsiya grafigini Dekart koordinatalar sistemasida quramiz.

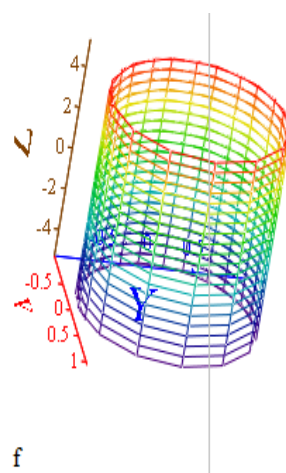
$$f(x,y) := 1$$



f



f



f

Dekart koordinatalar sistemasida Sferik koordinatalar sistemasida

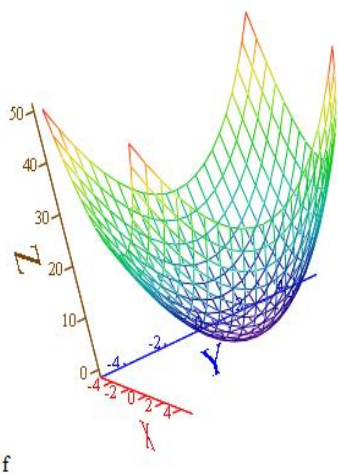
Silindrik koordinatalar sistemasida

Keyin uni “Формат 3-D графика” oynadan formatlarini ozgartirib, “Данные QuickPlot” sahifadagi grafikni sferik yoki tsilindrik koordinatalar tizimida tasvirlash parametrlarni o’rnatish amalga oshiramiz.

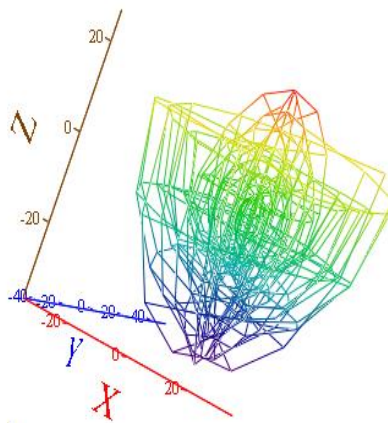
Misol 2. $f(x,y)=x^2+y^2$ funksiya grafigini turli koordinatalar tizimida quring.

Eng avvalo $f(x,y)$ funksiya grafigini Dekart koordinatalar tizimida quramiz.

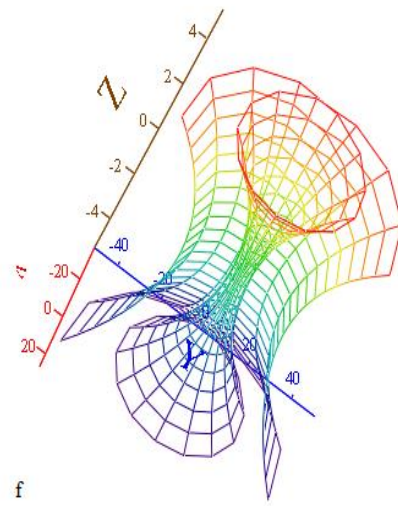
$$f(x,y) := x^2 + y^2$$



f



f



f

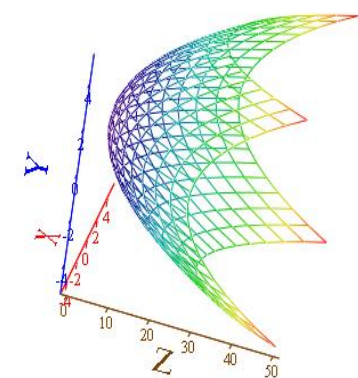
Dekart koordinatalar sistemasida Sferik koordinatalar sistemasida

Silindrik koordinatalar sistemasida

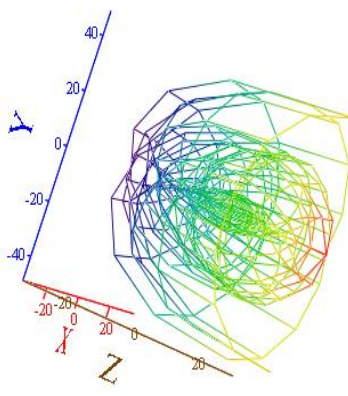
Keyin uni “Формат 3-D графика” oynadan formatlarini ozgartirib, “Данные QuickPlot” sahifadagi grafikni sferik yoki tsilindrik koordinatalar tizimida tasvirlash parametrlarni o’rnatish amalga oshiramiz.

$f(x,y)=x^2+y^2$ funksiya grafigini boshqacha joylashuvi:

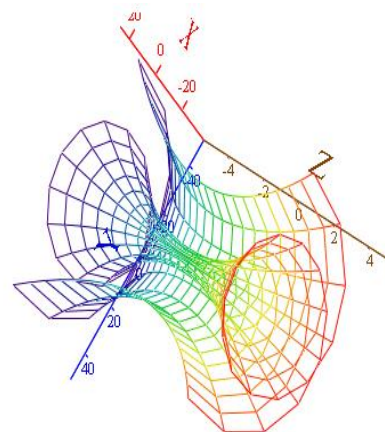
$$f(x,y) := x^2 + y^2$$



f



f



f

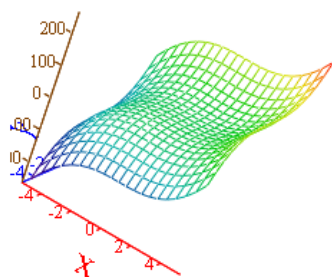
Dekart koordinatalar sistemasida

Sferik koordinatalar sistemasida

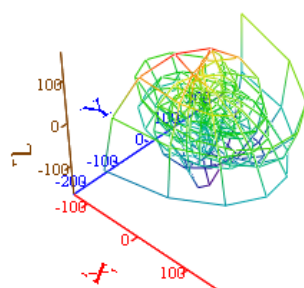
Silindrik koordinatalar sistemasida

Misol 3. $f(x,y)=x^3+y^3$ funksiya grafigini turli koordinatalar tizimida quring.

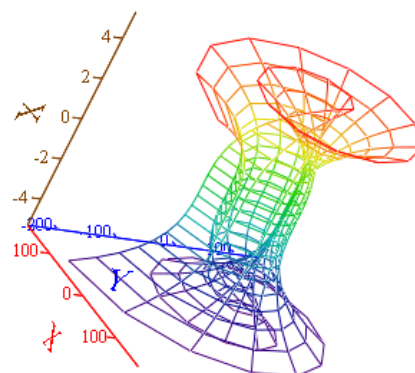
$$f(x,y) := x^3 + y^3$$



f



f



f

Dekart koordinatalar sistemasida

Sferik koordinatalar sistemasida

Silindrik koordinatalar sistemasida

Demak, funksiyaning grafigini MathCAD yordamida chizash va uni turli holatlarini, grafikni burish orqali ko'rish mumkin. Shuningdek, ixtiyoriy funksiyaning grafigini Dekart, Sferik yoki Silindrik koordinatalar tizimida yasash mumkin.

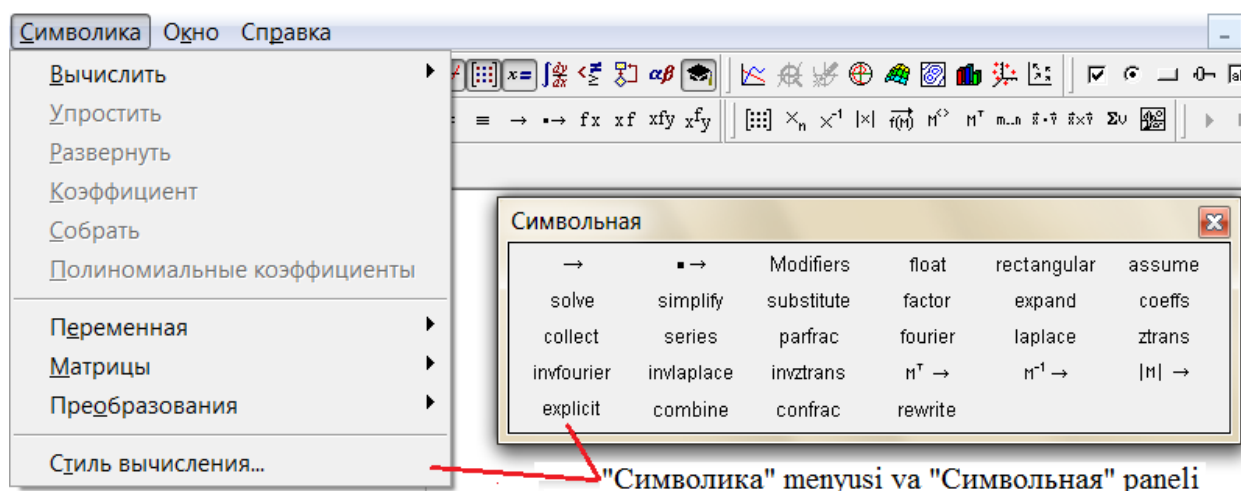
MATHCADDA SIMVOLLI HISOB KITOBLARNI BAJARISH.

Simvolli hisoblashlar

Miqdor jihatdan hisoblashda MathCAD “=” belgisidan so’ng bir yoki bir nechta sonlarni chiqaradi. Bu sonlarni bilish foydali bo’lsa ham, ular orqali argumentlar va ifodalar o’rtasidagi bog’liqlikni tushunish qiyin. Shu sababli, MathCAD sonli hisoblashlardan tashqari belgili (simvolli) hisoblashlarni ham amalga oshiradi. Bu degani, hisoblashlar natijasini analitik ko’rinishda tasvirlash mumkin. Masalan, aniqmas integral, differentsiallashtirish va boshqa shu kabi masalalarni yechishda uning yechimini analitik ko’rinishda tasvirlaydi.

Simvolli hisoblashlarni bajarishda ikkita asosiy vosita mavjud:

- “Символика” menyusi - simvolli hisoblashlar buyruqlari;
- “Символьная” paneli.



"Символика" menyusi va "Символьная" paneli

Bu vositalar ancha murakkab simvolli hisoblashlarda qo'llaniladi. Biz oddiy simvolli hisoblashni bajarishning eng sodda usuli, ya'ni tez-tez ishlatilib turiladigan usullardan biri - simvolli tenglik belgisi ($\rightarrow$) usulini ko'rib chiqamiz. Quyida bu usuldan foydalanishning ketma-ketlik tartibi berilgan:

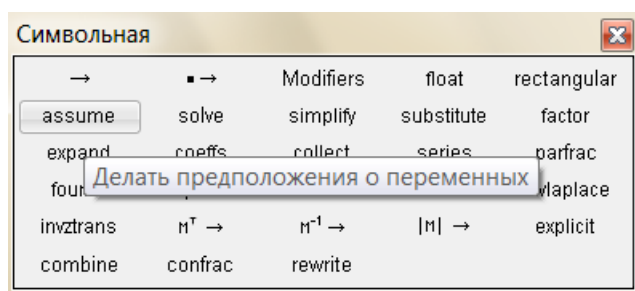
1. Funksiya kiritilsin, masalan $f(x)=\sin(x)$;
2. “Панель вычисления” panelidan matematik tushuncha tanlanadi. Masalan, aniqmas integral qaralayapti, ya'ni $\int$.
3. Kiritish joylari to'ldiriladi, ya'ni funksiya nomi va o'zgaruvchi nomi kiritiladi. Masalan, $\int dx$ ifoda $\int f(x)dx$ ko'rinishda yoziladi.

4. “Панель вычисления” panelidan simvolli belgi tengligi $\rightarrow$ “Вычислить символически” yoki “Ctrl+” klavish bosiladi.

Ushbu ishlar bajarilsa, natija beriladi.

$$\begin{array}{lll} f(x) = \sin(x) & f1(x) = x^2 & f2(x) = x^3 + x^2 + x \\ \int f(x) dx \rightarrow -\cos(x) & \int f1(x) dx \rightarrow \frac{1}{3} \cdot x^3 & \left(\frac{d}{dx} f2(x)\right) \rightarrow 3 \cdot x^2 + 2 \cdot x + 1 \end{array}$$

“Символьная” panelida simvolli hisoblashlarga doir buyruqlar to’plami berilgan. Agar biror buyruq ustiga sichqoncha ko’rsatkichini olib borsak, ushbu buyruq vazifasi haqida izoh paydo bo’ladi.



“Символьная” paneli buyruqlari vazifasi:

Nomi	Vazifasi
$\rightarrow$	Ifodani simvolli hisoblash natijasini olish.
$\blacksquare \rightarrow$	Simvolli kalit so’zni belgilash.
Modifiers	Kalit so’zlar modifikatsiyasi panelini ochish.
float	Suzuvchi vergulli qiymatlarni aniqlash.
rectangular	Kompleks sonlar uchun to’rtburchakli shakl olish.
assume	Aniq qiymat bilan yuborilgan o’zgaruvchini hisoblash.
solve	Tenglama va tenglamalar sistemasini yechish.
simplify	Ifodani soddalashtirish, ixchamlashtirish.
substitute	Ifodadalarini hisoblash.
factor	Agar butun ifodani ko’paytuvchilar ko’paytmasi shaklida ifodalash mumkin bo’lsa, tanlangan ifodani ko’paytuvchilarga ajratadi.
expand	Ifodada yig’indining barcha darajalari va ko’paytmalarini ochib

	chiqadi.
coeffs	Ifodani $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ ko'rinishda soddalashtirib barcha koeffitsiyentlarini aniqlaydi
collect	Oddiy yig'indida tasvirlangan polinom ko'rinishidagi ifodani ixchamlash.
series	Ifodani Teylor qatoriga yoyadi.
parfrac	Oddiy kasrga ifodani yoyish (aylantirish).
fourier	Fur'e qatoriga yoyish (aylantirish).
laplace	Laplas funksiyasiga yoyish (aylantirish).
ztrans	Z - yoyish (aylantirish).
invfourier	Teskari Fur'e qatoriga yoyish (aylantirish).
invlaplace	Teskari Laplas funksiyasiga yoyish (aylantirish).
invztrans	Teskari Z- yoyish (aylantirish).
$M^T \rightarrow$	Simvolli matritsani transponirlash.
$M^{-1} \rightarrow$	Teskari simvolli matritsa.
explicit	O'zgaruvchini belgilamasdan almashtirish.
combine	Kombinatsiyalashgan ifoda.
confrac	Kasrni kengaytirishni davom ettirish.
rewrite	Ifodani qayta yozish.

“Модификатор” (Modifiers) paneli buyruqlari vazifasi:

Nomi	Vazifasi
integer	Butun simvolli qiymatli modifikator.
real	Haqiqiy simvolli qiymatli modifikator.
RealRang e	Haqiqiy simvolli intervaldagi qiymatli modifikator.
complex	Kompleks son shakliga o'tkazish.
filly	To'liq simvolli qiymatli modifikator.

MathCAD simvolli matematikani qo'llaganda quyida ko'rsatilganidek, hisoblash natijasining o'rniga boshqa ifoda paydo bo'ladi. Bunda ifodaning o'zi yoki ko'paytuvchilarga ajratish yoki qatorga yoyish kabilar bo'lishi mumkin.

$$x := 3 \quad f(x) := x^2 + 1 \quad f(x) = 10 \quad f(x) \rightarrow 10 \quad g(x) := \sqrt[3]{x+5} \quad g(x) = 2 \quad g(x) \rightarrow 2$$

$$(a+b)^2 \rightarrow (a+b)^2 \quad (a+b)^2 \text{ expand} \rightarrow a^2 + 2 \cdot a \cdot b + b^2 \quad a^3 + 3 \cdot a^2 \cdot b + 3 \cdot a \cdot b^2 + b^3 \text{ factor} \rightarrow (a+b)^3$$

Ifodadan so'ng simvolli buyruqlardan foydalanib, ifodani turli ko'rinishdagi soddalashtirilgan hollarini olish mumkin.

MathCAD dasturining maxsus funksiyalari imkoniyatlaridan foydalanib masalalar va sodda differensial tenglamalarni yechish

Hosila, integral, limitga doir boshlang'ich funksiyalarni simvolli hisoblash:

Hosilani hisoblashga doir

$$\frac{d}{dx} \cos(x^2) \rightarrow -2 \cdot \sin(x^2) \cdot x \quad , \quad \frac{d^2}{dx^2} (x^2 + 2 \cdot x)^2 \rightarrow 2 \cdot (2 \cdot x + 2)^2 + 4 \cdot x^2 + 8 \cdot x$$

Integrallarni hisoblashga doir

$$\int_1^2 x^2 dx \rightarrow \frac{7}{3} \quad , \quad \int_1^y x^2 dx \rightarrow \frac{1}{3} \cdot y^3 - \frac{1}{3} \quad , \quad \int_0^b \int_1^a \sin(x) dx da \rightarrow -\sin(b) + \cos(1) \cdot b$$

Aniqmas integral yoki boshlang'ich funksiyalarni hisoblashlarga doir

$$\int e^{\frac{-1}{3} \cdot x} dx \rightarrow -3 \cdot \exp\left(\frac{-1}{3} \cdot x\right) \quad , \quad \int (3 \cdot x + 4)^3 dx \rightarrow \frac{1}{12} \cdot (3 \cdot x + 4)^4$$

Limit va yig'indilarni hisoblashga doir

$$\lim_{x \rightarrow -3} \frac{x^2 - 9}{x + 3} \rightarrow -6 \quad , \quad \lim_{x \rightarrow 0} \frac{\sin(x)}{x} \rightarrow 1 \quad , \quad \sum_n n^2 \rightarrow \frac{1}{3} \cdot n^3 - \frac{1}{2} \cdot n^2 + \frac{1}{6} \cdot n$$

$$\sin(2\pi) + \sin(1) \rightarrow \sin(1) \quad \frac{d}{dx} \sin(x) \rightarrow \cos(x) \quad \lim_{x \rightarrow -3} \frac{x^2 - 9}{x + 3} \rightarrow -6$$

$$\lim_{x \rightarrow 0} \frac{\sin(x)}{x} \rightarrow 1 \quad \sum_n n^2 \rightarrow \frac{n \cdot (n + 1) \cdot (2 \cdot n + 1)}{6}$$

Simplify buyrug'i. Simplify buyrug'i ifodalarni soddalashtirishda ishlatiladi, bunda u barcha asosiy ayniyatlardan foydalanib ifodani sodda ko'rinishda keltiradi.

$$\frac{(a+3)^2}{a^2-9} \text{ simplify} \rightarrow \frac{(a+3)}{(a-3)} \quad ,$$

$$\sin(x)^2 + \cos(x)^2 \text{ simplify} \rightarrow 1$$

$$\frac{5}{x+1} + \frac{x}{x-1} \text{ simplify} \rightarrow \frac{(6 \cdot x - 5 + x^2)}{[(x+1) \cdot (x-1)]}$$

Expand buyrug'i. Ifodada yig'indining barcha darajalari va ko'paytmalarini ochib chiqadi va ifodani soddalashtirilgan holda qaytaradi.

$$(a + b)^2 \text{ expand} \rightarrow a^2 + 2 \cdot a \cdot b + b^2$$

$$(a - c)^3 \text{ expand} \rightarrow a^3 - 3 \cdot a^2 \cdot c + 3 \cdot a \cdot c^2 - c^3$$

$$(x - y) \cdot (x + y) \text{ expand} \rightarrow x^2 - y^2$$

$$(a + b) \cdot (a - c) \text{ expand} \rightarrow a^2 - a \cdot c + a \cdot b - b \cdot c$$

$$(a + b + c)^2 \text{ expand} \rightarrow a^2 + 2 \cdot a \cdot b + 2 \cdot a \cdot c + b^2 + 2 \cdot b \cdot c + c^2$$

Factor va complex buyruqlari. Factor buyrug'i asosan ifodalarni ko'paytuvchilarga ajratishda ishlatiladi, agar u ifodani ko'paytuvchilarga ajrata olmasa ifodani o'zini qaytaradi.

$$i \cdot 2 + 2 \text{ complex} \rightarrow 2 + 2 \cdot i$$

$$e^{2i-2} \text{ complex} \rightarrow \exp(-2) \cdot \cos(2) + i \cdot \exp(-2) \cdot \sin(2)$$

$$a^2 + 2 \cdot a \cdot b + b^2 \text{ factor} \rightarrow (a + b)^2$$

$$x^2 - y^2 \text{ factor} \rightarrow (x - y) \cdot (x + y)$$

$$a^2 - a \cdot c + a \cdot b - b \cdot c \text{ factor} \rightarrow (a + b) \cdot (a - c)$$

$$x^3 - 6 \cdot x^2 + 11 \cdot x - 6 \text{ factor} \rightarrow (x - 1) \cdot (x - 2) \cdot (x - 3)$$

$$a^2 + 2 \cdot a \cdot b + 2 \cdot a \cdot c + b^2 + 2 \cdot b \cdot c + c^2 \text{ factor} \rightarrow (a + b + c)^2$$

Coeffs va substitute buyruqlari. Coeffs buyrug'i berilgan ifodani soddalashtirib, polinom koeffitsiyentlarini aniqlaydi. Substitute buyrug'i esa berilgan ifodani o'zgaruvchilarini almashtirib soddalashtiradi.

$$a \cdot x^3 + b \cdot x^2 + c \cdot x + 2 \text{ coeffs}, x \rightarrow \begin{pmatrix} 2 \\ c \\ b \\ a \end{pmatrix}$$

$$(x+2)^2 \text{ coeffs}, x \rightarrow \begin{pmatrix} 4 \\ 4 \\ 1 \end{pmatrix} \quad (a+b) \cdot (a-b) \text{ coeffs}, a \rightarrow \begin{pmatrix} -b^2 \\ 0 \\ 1 \end{pmatrix}$$

$$(a+b)^2 \text{ substitute}, a = 1 \rightarrow (1+b)^2$$

$$(a+b)^2 \text{ substitute}, a = 1, b = 2 \rightarrow 9 \quad +$$

$$(a+b)^2 \text{ substitute}, a = x+b \rightarrow (x+2 \cdot b)^2$$

Solve buyruqlari. Solve buyrug'i yordamida algebraik tenglamalarni yechish mumkin. Chunki bu buyruq ifodani biror o'zgaruvchiga nisbatan nollarini aniqlash imkoniyatiga ega.

$$a \cdot x^2 + b \cdot x + c \text{ solve}, x \rightarrow \begin{bmatrix} \frac{1}{(2 \cdot a)} \cdot \left[-b + (b^2 - 4 \cdot a \cdot c)^{\left(\frac{1}{2}\right)} \right] \\ \frac{1}{(2 \cdot a)} \cdot \left[-b - (b^2 - 4 \cdot a \cdot c)^{\left(\frac{1}{2}\right)} \right] \end{bmatrix}$$

$$x^4 + 9 \cdot x^3 + 31 \cdot x^2 + 59 \cdot x + 60 \text{ solve}, x \rightarrow \begin{pmatrix} -4 \\ -3 \\ -1 + 2 \cdot i \\ -1 - 2 \cdot i \end{pmatrix}$$

$$(x+4) \cdot (x+3) \cdot (x^2 + 5 \cdot x - 6) \text{ solve}, x \rightarrow \begin{pmatrix} -4 \\ -3 \\ -6 \\ 1 \end{pmatrix}$$

Ifodani Teylor qatoriga yoyish. Series buyrug'i yordamida berilgan ifodani biror nuqta atrofida Teylor qatoriga yoyish mumkin.

$$\frac{\sin(x)}{x} \text{ series, } x, 4 \rightarrow 1 - \frac{1}{6} \cdot x^2, \quad \sin(x) \text{ series, } x, 6 \rightarrow x - \frac{1}{6} \cdot x^3 + \frac{1}{120} \cdot x^5$$

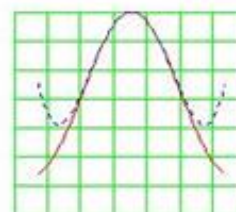
$$\frac{\sin(x)}{x} \text{ series, } x, 6 \rightarrow 1 - \frac{1}{6} \cdot x^2 + \frac{1}{120} \cdot x^4$$

$\frac{\sin(x)}{x}$ va uning teylor qatordagi yoyilmasini bitta grafikda tasvirlaymiz.

$$F(x) := \frac{\sin(x)}{x} \quad F1(x) := 1 - \frac{1}{6} \cdot x^2 + \frac{1}{120} \cdot x^4$$

$$x := -4, -3.9 \dots 4$$

$\frac{F(x)}{F1(x)}$



Nazorat savollari:

1. MathCAD maxsus operatorlari haqida tushuncha bering.
2. MathCAD maxsus operatorlari qanday kiritiladi?
3. MathCAD maxsus operatorlari orqali hisoblashlarni bajarish haqida tushuncha bering.
4. MathCAD muhitida hosila operatorining qo'llanilishi haqida tushuncha bering.
5. MathCAD muhitida integrallash operatorining qo'llanilishi haqida tushuncha bering.
6. MathCAD dasturida o'zgaruvchilarning satrli tipining qo'llanilishi haqida tushuncha bering.
7. MathCAD dasturida simvolli hisoblashlarni qo'llanilishi haqida tushuncha bering.
8. MathCAD dasturi "Символьная" paneli buyruqlari haqida tushuncha bering.
9. MathCAD dasturida hosila, integral, limitga doir boshlang'ich funksiyalarni simvolli hisoblashlarga doir misollar keltiring.
10. MathCAD dasturining maxsus funksiyalari imkoniyatlaridan foydalanib sodda differensial tenglamalarni yechish haqida tushuncha bering.

ADABIYOTLAR RO'YXATI

1. Heikki Topi, Bentley University Waltham, Massachusetts, USA
“COMPUTING HANDBOOK THIRD EDITION” Information Systems
and Information Technology, 2014.
2. Aripov M.M va b. Axborot texnologiyalari, o'quv qo'llanma. Toshkent,
Noshir, 2009 y.
3. Aripov M.M., Muxammadiyev J.U. Informatika, information
texnologiyalar. Darslik. Toshkent. 2007 y.
4. Shodimetov X.M va b. Informatika va axborot texnologiyalari. O'quv
qo'llanma, Toshkent 2012 y.
5. D. Abdukadirova. Axborot texnologiyalari va jarayonlarning matematik
modellashtirish fanidan laboratoriya va amaliyot ishlari uchun o'quv
uslubiy qo'llanma. Toshkent 2018 y.
6. Iqtisodiy-matematik usullar va modellar (O'quv qo'llanma)
X.M.Shodimetov, N.A.Asqarov, B.N. Abduqayumov. Toshkent 2010 y.
7. Kadirov M.M. «Axborot texnologiyalari» 1-qism. O'quv qo'llanma. «Fan
va texnologiya», Toshkent 2018, 316 b.
8. Kadirov M.M. «Texnik tizimlarda axborot texnologiyalari» 2-qism. O'quv
qo'llanma. «Fan va texnologiya», Toshkent 2018, 306 b.
9. V.Qobulov va boshqalar. C va C++. O'quv qo'llanma. – Toshkent:
Aloqachi. 2013.
10. Greg Harvey. MS Excel 2010 all in one. USA, 2010.
11. Mathcad Tutorial By Colorado State University Student: Minh anh
Nguyen Power Electronis III.
12. Sohibov T.F. va boshqalar. Muhandislik ishlarida MathCAD dasturi.
Buxoro: 2019, 200 b.

TAYANCH IBORALAR

- Algoritm, 16, 18, 20, 21, 22, 23, 24, 27, 32, 33, 37
- Axborot texnologiyalari, 1, 3, 4, 5, 6, 7, 13, 187
- Axborotlashgan jamiyat, 9, 11, 13
- Dasturlash tili, 34, 37
- Direktivalar, 51
- Diskret o`zgaruvchilar, 148
- do...while, 81, 82
- Expand, 184
- for operatori, 83, 84, 87, 105
- Funksiyalar, 41, 53, 90, 106, 114, 143, 150
- Global o`zgaruvchilar, 99
- Identifikator, 41, 53
- if operatori, 63, 67, 70, 74
- Informatika, 3, 4, 187
- Interpretator, 36, 38
- Kompilyator, 36, 38, 42, 43, 53, 92
- Lokal o`zgaruvchilar, 98, 106
- Mantiqiy operatorlar, 69, 74
- Marshrutlash masalasi, 122
- massivlar, 1, 106, 108, 109, 111, 114, 115
- Matematik model, 16, 123
- MathCAD paketi, 141
- Optimallashtirish masalasi, 129
- Simplify, 183
- Simvolli hisoblashlar, 180
- Solve, 185
- Switch operatori, 70
- Texnologiya, 4
- Translyator, 36, 38
- while operatori, 75, 76, 81, 82, 83

MUNDARIJA

KIRISH.....	3
Axborot texnologiyalari, axborotli jarayonlar va axborotlashgan jamiyat haqida tushuncha.....	5
“Axborot texnologiyalari va jarayonlarni matematik modellashtirish” fanining mohiyati, maqsadi va vazifalari.	5
Zamonaviy axborot texnologiyalarining hozirgi zamondagi o’rni va rivojlanish istiqbollari hamda ularning fan-texnika taraqqiyotida, jamiyat rivojida tutgan o’rni	9
Axborotlashgan jamiyat	11
Axborot madaniyati va etikasi	14
Algoritmlash va dasturlash asoslari.	16
Kompyuterda masalalarni yechish bosqichlari	16
Algoritm tushunchasi va uning xossalari	19
Algoritm turlari	25
Chiziqli hisoblash jarayonlarini algoritmlash	32
Algoritmlarni ta’riflash usullari	33
Masalalarni algoritmlash va dasturlash asoslari.....	34
C++ dasturlash tili va uning asosiy operatorlari.	39
C++ dasturlash tili.....	39
Dasturning tuzilishi.	43
C++ tilida direktivalar	51
C++ algoritmik tilining kiritish va chiqarish operatorlari.....	53
C++ algoritmik tilida ma'lumotlarni kiritish va chiqarish.	53
Fayllar. Oqimli kiritish va chiqarish.	56
C++ algoritmik tilining tarmoqlanish operatorlari.....	62
Tarmoqlanuvchi dastur tuzilmasi.....	62
Mantiqiy operatorlar.....	68
Switch operatori.	69
C++ algoritmik tilining takrorlash operatorlari.....	74

Takrorlash operatorlari.....	74
for operatori.....	82
Funksiyalar.....	89
Massivlar bilan ishlash. Bir va ikki oʻlchamli massivlar.....	105
Bir oʻlchovli massivlar.....	105
Jadvallar.....	107
Simvolli massivlar.....	108
Soʻzlar massivlari.....	109
Koʻrsatkichlar massivlari.....	111
Matematik modellarni elektron jadvaldagi tadqiqoti.....	117
MS Excelda Solver imkoniyatlari.....	117
Marshrutlash masalasi.....	121
Matematik modellashtirishning elektron jadval yordamida yechish.....	128
Optimallashtirish masalasi.....	128
MathCADda matematik ifodalar va funksiyalar bilan ishlash.....	140
MathCAD dasturi haqida umumiy maʼlumotlar.....	140
MathCAD dasturida hisoblashlar.....	143
MathCAD dasturida funktsiyalar masteri bilan ishlash.....	149
Mathcad paketida ikki va uch oʻlchamli grafiklarni qurish.....	153
Dekart koordinatalar sistemasida ikki oʻlchovli grafiklarni qurish.....	153
Dekart koordinatalar sistemasida ikki oʻlchovli grafik formatlarini oʻzgartirish...	154
Bir sohada bir nechta ikki oʻlchovli grafiklarni qurish.....	158
Berilgan intervalda ikki oʻlchovli grafiklarni qurish.....	160
MathCAD muhitida uch oʻlchovli grafiklar.....	162
Berilgan intervalda uch oʻlchovli grafiklarni qurish.....	172
MathCAD muhitida turli koordinatalar sistemasida grafiklarni qurish.....	174
Mathcadda simvolli hisob kitoblarni bajarish.....	178
Simvolli hisoblashlar.....	178
Adabiyotlar roʻyxati.....	185

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
Понятие об информационных технологиях, информационных процессах и информатизированном обществе.	5
Сущность, цели и задачи дисциплины “Информационные технологии и математическое моделирование процессов”.....	5
Место и перспективы развития современных информационных технологий в современном мире и их роль в научно-техническом развитии, развитии общества	9
Информатизированное общество	11
Информационная культура и этика	14
Основы алгоритмизации и программирования.	16
Этапы решения задач на компьютере	16
Понятие алгоритма и его свойства	19
Типы алгоритмов	25
Алгоритмизация линейных вычислительных процессов	32
Методы описания алгоритмов	33
Алгоритмизация задач и основы программирования	34
Язык программирования C++ и его основные операторы.	39
Язык программирования C++	39
Структура программы.	43
Директивы в C++	51
Операторы ввода и вывода алгоритмического языка C++.	53
Ввод и вывод данных на алгоритмическом языке C++.	53
Файл. Поточковый вход и выход.	56
Разветвляющиеся операторы алгоритмического языка C++.	62
Разветвляющиеся структура приложения.	62
Логические операторы.	68
Оператор switch.	69
Операторы повторения алгоритмического языка C++.	74
Операторы повторения	74

Оператор for.	82
Функции	89
Работа с массивами. Одно-и двумерные массивы.	105
Одномерные массивы.	105
Таблицы.	107
Символьные массивы.	108
Массивы слов.	109
Массивы указателей.	111
Исследование математических моделей в электронных таблицах.	117
Solver решения в MS Excel.	117
Вопрос маршрутизации.	121
Решение математического моделирования с помощью электронной таблицы..	128
Вопрос оптимизации	128
Работа с математическими выражениями и функциями в MathCAD е.	140
Общие сведения о программе MathCAD	140
Расчеты в программе MathCAD	143
Работа с мастером функций в программе MathCAD	149
Просмотр двух-и трехмерной графики в пакете Mathcad.	153
Построение двумерных графиков в декартовой системе координат	153
Преобразование форматов двумерных графиков в декартовой системе координат	154
Построение нескольких двумерных графиков в одной области	158
Построение двумерных графиков на заданном интервале	160
Трехмерная графика в среде MathCAD	162
Построение трехмерных графиков на заданном интервале	172
Построение графиков в различных системах координат в среде MathCAD	174
Выполнение символьных вычислений в маткаде.	178
Символьные вычисления	178
Список литературы	185

CONTENT

INTRODUCTION	3
The concept of information technologies, information processes and an informatized society.	5
The essence, goals and objectives of the discipline "Information technologies and mathematical modeling of processes".	5
The place and prospects of the development of modern information technologies in the modern world and their role in scientific and technological development, the development of society	9
Informatized society	11
Information culture and ethics	14
Fundamentals of algorithmization and programming.	16
Stages of solving problems on a computer	16
The concept of the algorithm and its properties	19
Types of algorithms	25
Algorithmization of linear computing processes	32
Methods for describing algorithms	33
Algorithmization of problems and the basics of programming	34
The C++ programming language and its basic operators.	39
C++ programming language	39
Program structure.	43
Directives in C++	51
C++algorithmic language input and output operators.	53
Data input and output in the C++algorithmic language.	53
File. Streaming input and output.	56
Branching operators of the C++algorithmic language.	62
Branching application structure.	62
Logical operators.	68
Switch statement.	69
C++algorithmic language repetition operators.	74
Repeat operators	74

The for statement.	82
Functions	89
Working with arrays. One-and two-dimensional arrays.	105
One-dimensional arrays.	105
Tables.	107
Character arrays.	108
Arrays of words.	109
Arrays of pointers.	111
Research of mathematical models in spreadsheets.	117
Solver solutions in MS Excel.	117
Routing issue.	121
Solving mathematical modeling using a spreadsheet.	128
Optimization issue	128
Working with mathematical expressions and functions in MathCAD	140
General information about MathCAD	140
Calculations in MathCAD	143
Working with the function wizard in MathCAD	149
View two-and three-dimensional graphics in the Mathcad package.	153
Plotting two-dimensional graphs in a Cartesian coordinate system	153
Converting formats of two-dimensional graphs in the Cartesian coordinate system .	154
Plotting multiple two-dimensional graphs in a single area	158
Plotting two-dimensional graphs on a given interval.	
Three-dimensional graphics in MathCAD	162
Plotting three-dimensional graphs on a given interval	172
Plotting graphs in various coordinate systems in MathCAD	174
Performing symbolic calculations in the MathCAD.	178
Symbolic calculations	178
References	185



Nurullayev Mirxon Muhammadovich.
Buxoro muhandislik-texnologiya instituti "Axborot kommunikatsiya texnologiyalari" kafedrası katta o`qituvchisi.

M.M.Nurullaev muallifligida 3 ta o`quv qo`llanma, 10 ga yaqin dasturlar uchun mualliflik guvohnomalari olingan, 2 ta xalqaro grant ishtirokchisi, 2 ta xorijiy davlatda malaka oshirib kelgan, 20 dan ortiq ilmiy maqolalar nashr ettirgan. Axborot xavfsizligi bo`yicha ilmiy ish olib bormoqda.

