

INFORMATIKA

VA AXBOROT
TEXNOLOGIYALARI

9 - SINIF



20 20

INFORMATIKA

VA AXBOROT TEXNOLOGIYALARI

9-SINF

Umumiy o'rta ta'lim maktablarining 9-sinfi uchun darslik

O'zbekiston Respublikasi Xalq ta'limi
vazirligi tomonidan tavsiya etilgan

«NASHRIYOT UYI TASVIR»

TOSHKENT – 2020

UO'K 004(075.3)

KBK 32.81я72

Ф 20

Fayziyeva M. R.

Informatika va axborot texnologiyalari: umumiy o'rta ta'lim maktablarining 9-sinfi uchun darslik: M. R. Fayziyeva, D. M. Sayfurov, N. S. Xaytullayeva - Toshkent: Tasvir, 2020. – 112 b.

Professor F. M. Zakirovning umumiy tahriri ostida.

Taqrizchilar:

T. Z. Teshabayev – Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti rektori, iqtisod fanlari doktori;

B. B. Mo'minov – Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti, "Informatika asoslari" kafedrasini mudiri, texnika fanlari doktori;

F. R. Tursunova – Toshkent shahar XTXQTMOHM "Aniq va tabiiy fanlar metodikasi" kafedrasini katta o'qituvchisi;

U. B. Mamatqulov – Qashqadaryo viloyati Koson tumani 9-umumiy o'rta ta'lim maktabining "Informatika va axborot texnologiyalari" fani o'qituvchisi.

The graphic contains a yellow box with a QR code and the text "SCAN ME TO GET TO DIGITAL RESOURCES" and "http://...O'ZBEK TILIDA". To the right, a red box with a warning icon says "DIQQAT!". Below that, a grey box with a warning icon says "Darslikni to'ldiruvchi raqamli tashkil etuvchilarga o'tish uchun QR Code yoki web-saytdan foydalaning." At the bottom, a yellow box contains the URL "dr.rtm.uz" with a mouse cursor pointing to it.

Respublika maqsadli kitob jamg'armasi
mablag'lari hisobidan chop etildi

I BOB. KOMPYUTERNING MANTIQUIY ISHLASH PRINSIPI

Mantiq asoslari.....	4
Mantiqiy amallar va ifodalar.....	6
Mantiqiy ifodalarning rostlik jadvalini tuzish.....	10
Mantiqiy sxemalar	13
Amaliy mashg'ulot.....	16

II BOB. KOMPYUTERDA MASALALARNI LOYIHALASHTIRISH VA MODELLASHTIRISH

Masalalarni kompyuterda yechish bosqichlari.....	17
Model va uning turlari	20
Amaliy mashg'ulot.....	26

III BOB. ALGORITMLASH ASOSLARI

Algoritm tushunchasi va uning xossalari	27
Algoritm turlari va tasvirlash usullari.....	30
Amaliy mashg'ulot.....	34
Nazorat ishi.....	35
Chiziqli algoritmlar.....	35
Tarmoqlanuvchi algoritmlar.....	37
Amaliy mashg'ulot.....	39
Takrorlanuvchi algoritmlar	40
Amaliy mashg'ulot.....	43
Aralash (kombinatsiyalashgan) algoritmlar.....	44

IV BOB. DASTURLASH ASOSLARI

Dastur va dasturlash haqida.....	47
Dasturlash tillari	48
Python dasturlash tilini o'rnatish	51
Pythonda o'zgaruvchilarni tavsiflash	54
Pythonda xatoliklar bilan ishlash	57
Pythonda ma'lumot turlari	59
Amaliy mashg'ulot.....	61
Nazorat ishi.....	62

Pythonda arifmetik amallarni bajarish.....	62
Amaliy mashg'ulot.....	64
Pythonda satrlar bilan ishlash.....	65
Amaliy mashg'ulot.....	67
Pythonda operator va ifodalar.....	68
Amaliy mashg'ulot.....	69
Pythonda sodda masalalarni dasturlash	70
Amaliy mashg'ulot.....	72
Pythonda mantiqiy masalalarni dasturlash	73
Amaliy mashg'ulot.....	75
Tarmoqlanuvchi algoritmlarni dasturlash. if...else operatori	76
Amaliy mashg'ulot.....	78
Tarmoqlanuvchi algoritmlarni dasturlash. elif operatori.....	79
Amaliy mashg'ulot.....	81
Takrorlanuvchi algoritmlarni dasturlash. for operatori	82
Amaliy mashg'ulot.....	85
Nazorat ishi.....	86
Takrorlanuvchi algoritmlarni dasturlash. while operatori.....	86
Amaliy mashg'ulot.....	88
Sikllarni boshqarish: continue, break operatorlari.....	89
Qism dasturlar: funksiyalar va protseduralar	91
Amaliy mashg'ulot.....	95
Funksiyalar va o'zgaruvchilar	96
Amaliy mashg'ulot.....	98
Python dasturlash tili kutubxonasi	100
Amaliy mashg'ulot.....	103
Pythonda foydalanuvchi grafik interfeysi bilan ishlash.....	104
Amaliy mashg'ulot.....	107
Nazorat ishi.....	108
Foydalanilgan adabiyotlar va web-saytlar ro'yxati.....	110

1-DARS. MANTIQ ASOSLARI

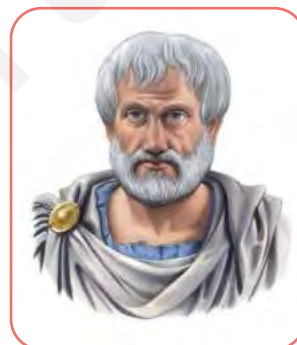


1. Osmon bilan yerning orasida nima bor? ("bilan" so'zi bor)
2. Dengiz tubida qanday tosh bo'lmaydi? (quruq tosh)
3. Qaysi qo'lda choyni aralashtirish qulayroq? (qoshiq bor qo'lda)

Sizningcha, bular qanday savollar toifasiga kiradi? To'g'ri, bular mantiqiy savollar edi, chunki Siz mantiqiy fikrlash qobiliyatingiz tufayligina bu savollarga to'g'ri javob bera oldingiz. Xo'sh, mantiq nima, unga qachon asos solingan?

Mantiq o'zining shakllanish va rivojlanish tarixiga ega. Mantiq masalalari dastlab Parmenid, eleylik Zeno va Geraklit ta'limotida u yoki bu darajada ko'rib chiqilgan. Mantiqqa oid fikrlar, tafakkur shakllari va usullari to'g'risidagi dastlabki ta'limotlar qadimgi Sharq mamlakatlari, xususan, Hindiston va Xitoyda vujudga kelgan bo'lsa-da, qadimda mantiq falsafaning tarkibida bo'lgan, mustaqil fan sifatida shakllanmagan. Qadimgi Yunon mutafakkirlari tomonidan yaratilgan ta'limotlar zamonaviy mantiqning asosi hisoblanadi.

Mantiq ilmining alohida fan sifatida shakllanishi miloddan avvalgi IV asrda yashab o'tgan buyuk yunon olimi Arastu (Aristotel) ismi bilan bog'liq. U birinchi bo'lib mantiq ilmi o'rganadigan masalalar doirasini aniqladi, mantiqqa "ma'lum bilimlardan noma'lum bilimlarni aniqlovchi", "chin fikrni xato fikrdan ajratuvchi" fan sifatida ta'rif berdi. Olim birinchi marta mantiqiy tafakkur shakllarini uning mazmunidan ajratib, mantiq va matematikani uyg'unlashtirishga harakat qildi, dalillar nazariyasiga asos soldi.



Formal mantiq ilmining asoslari hamda mantiq jarayonini turli matematik belgilar bilan ifodalashga intilish ham Arastu asarlarida ko'zga tashlanadi. Markaziy osiyolik alloma Abu Nasr Forobiy Arastuning umumiy formal mantiqqa oid qarashlariga tayangan holda mantiq fanini shakllantira olgan.

Kundalik hayotda mantiq atamasi "fikrlar mantig'i", "gap mantig'i", "xatti-harakat mantig'i", "narsalar mantig'i", "voqealar mantig'i" kabi jumalarda fikrni ifodalash maqsadida qo'llaniladi.

Mantiq tafakkurning namoyon bo'lish shakllari va taraqqiyotini, shu jumladan, fikrlar o'rtasidagi aloqadorlikni ko'rsatadigan qonun-qoidalar yig'indisini o'rganadi.

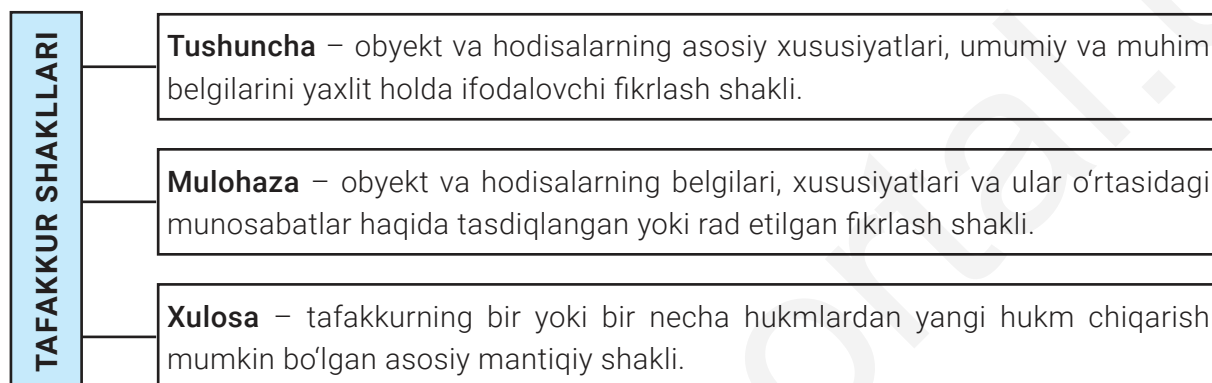
Hozirda mantiqning formal mantiq, dialektik mantiq va matematik mantiq kabi yo'nalishlari mavjud.

Eslab qoling!

Mantiq (logika) atamasi qadimgi yunoncha λογικος – "fikrlash ilmi" atamasiga mos keladi va "so'z", "fikir", "mulohaza", "nutq", "aql" degan ma'nolarni anglatib, bilish jarayoni bilan uzviy bog'liqdir. Mantiqning o'rganish obyektini **tafakkur** tashkil etadi.

Eslab qoling!

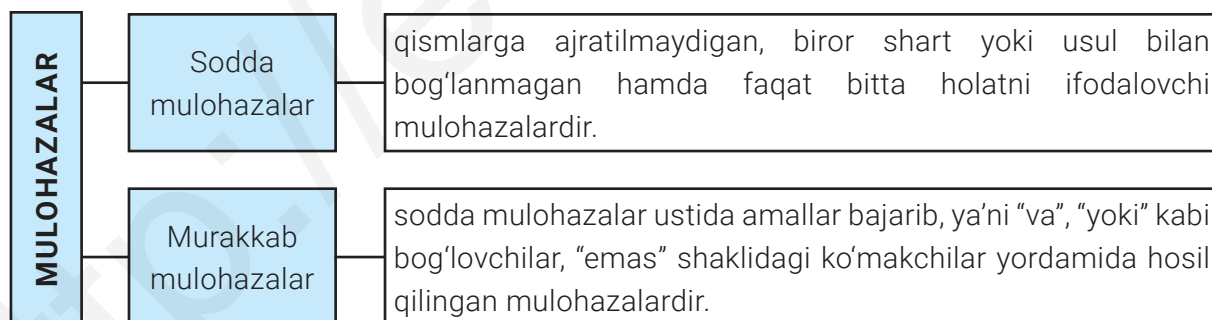
- **Formal mantiq** statik voqelikga oid bo'lib, tafakkur strukturasi fikrning aniq mazmuni va taraqqiyotidan chetlashgan holda, nisbatan mustaqil ravishda o'rganadi. Uning diqqat markazida muhokamani to'g'ri qurish bilan bog'liq qoida va mantiqiy amallar yotadi.
- **Dialektik mantiq** dinamik voqelikga oid bo'lib, tafakkurni uning mazmuni va shakli yaxlitligi hamda rivojlanishi orqali o'rganadi.
- **Matematik mantiq** tafakkurni matematik usullar yordamida tadqiq etadi. U hozirgi zamon matematikasining muhim yo'nalishlaridan biri hisoblanadi.



Masalan. Alining yoshi Valining yoshidan katta. Ra'noning yoshi esa Valining yoshidan kichik. Xulosa Alining yoshi Ra'noning yoshidan katta.

Mantiqning boshlang'ich tushunchalaridan biri mulohaza tushunchasidir. Mulohaza deganda, rost yoki yolg'onligi haqida fikr yuritish mumkin bo'lgan darak gapni tushunamiz.

S'o'roq va undov gapli ta'riflar ham mulohaza bo'la olmaydi. Masalan, "2 ga bo'linuvchi songa juft son deyiladi", degan ta'rif mulohaza bo'la olmaydi. Ammo "agar butun son 2 ga bo'linsa, u holda bu son juft son bo'ladi", degan darak gap mulohaza bo'ladi. Bu mulohaza – rost.



Mulohazalar narsa yoki obyektlarning xususiyatlarini, tushunchalar o'rtasidagi munosabatlarni haqqoniy (to'g'ri) aks ettirsa, rost, aksincha, haqiqatga zid, noto'g'ri bayon qilsa, yolg'on mulohaza bo'ladi. Har qanday mulohaza rost yoki yolg'on bo'ladi. Hech bir mulohaza bir vaqtning o'zida ham rost, ham yolg'on bo'la olmaydi. Masalan, " $7 < 5$ ", "A" – unli harf, "11 – tub son", "kompyuter XVI asrning oxirida ixtiro qilingan", "Covid-19 virusi inson uchun xavfli emas" mulohazalarining birinchisi – yolg'on, ikkinchisi va uchinchisi – rost, to'rtinchi va beshinchisi esa yolg'on mulohazalardir.

Mulohazaning qiymati deganda, biz uning rost yoki yolg'onligini tushunishimiz kerak. Har bir mulohaza faqat bitta: "rost" yoki "yolg'on" mantiqiy qiymatga ega bo'ladi!

Mantiqiy o'zgaruvchi deganda, istalgan mulohazani anglatuvchi, "rost" yoki "yolg'on" mantiqiy qiymatni qabul qilishi mumkin bo'lgan o'zgaruvchilar tushuniladi. Qulaylik uchun "rost" yoki "yolg'on" mantiqiy qiymatlarni belgilash usullarini keltirib o'tamiz:

Rost	R	Ha	True	T	1
Yolg'on	Y	Yo'q	False	F	0

Mulohazalarni, odatda, lotin alifbosining bosh harflari (A, B, C, ..., X, Y, Z) bilan belgilash qabul qilingan. Masalan: A = "Toshkent – go'zal shahar", B = "WWW – butun dunyo o'rgimchak to'ri". Bu misolda A va B mantiqiy o'zgaruvchilar orqali berilgan mulohazalar rost bo'lib, bu mulohazalarning qiymati 1 ga teng.

SAVOL VA TOPSHIRIQLAR



1. Mantiq tushunchasiga qachon asos solingan?
2. Mulohaza deb nimaga aytiladi?
3. Mulohazaning qanday turlari mavjud?
4. Sodda mulohaza nima va u qanday qiymatlarni qabul qiladi?
5. Har qanday o'tgan zamon darak gapi mulohaza bo'la oladimi? Kelasi zamon darak gaplari-chi?
6. Mantiqiy o'zgaruvchi nima? Uning qiymatlarini ayting.

UY VAZIFASI



- 1-mashq. Quyidagi fikrlarning qaysilari mulohaza bo'la oladi? Mulohazalarning rost yoki yolg'onligini aniqlang.
- a) Ushbu tasmaning uzunligi qanday?
 - b) "1234321" – palindrom son.
 - d) Ertalabki badantarbiya mashqlarini bajaring!
 - e) Uchburchak burchaklarining yig'indisi 160° ga teng.
 - f) Axborotlarni kiritish qurilmalarini ayting.
 - g) 5 ga bo'lingan har qanday son 3 ga ham bo'linadi.

2–3-DARSLAR. MANTIQUIY AMALLAR VA IFODALAR

Mantiqiy amallar mulohazalar mazmuni yoki hajmining o'zgarishiga hamda yangi mulohazalarni shakllantirishga olib keladi. Mantiqiy ifodalar A, B, C, ... lotin harflari yordamida belgilangan mulohazalarni konyunksiya, dizyunksiya, inversiya, implikatsiya va ekvivalensiya kabi mantiqiy bog'lovchilar bilan ma'lum bir qoidaga ko'ra birlashtirish natijasida hosil qilinadi. Mantiqiy ifodalar unda ishtirok etgan o'zgaruvchilarning qiymatlariga bog'liq ravishda rost (mantiqiy 1) yoki yolg'on (mantiqiy 0) qiymatlardan birini qabul qilishi mumkin.

Quyida sodda mulohazalar ustida bajarilishi mumkin bo'lgan ba'zi amallar bilan tanishamiz.

Ikkita sodda mulohazaning "va" bog'lovchisi orqali bog'lanishidan hosil bo'lgan yangi mulohazaga **sodda mulohazalar ko'paytmasi** deyiladi.

Eslab qoling!

- 1-ta'rif: Ikkita A va B sodda mulohaza bir paytda rost bo'lgandagina rost bo'ladigan yangi (murakkab) mulohazani hosil qilish amali **konyunksiya** (lot. conjunctio – bog'layman) – **mantiqiy ko'paytirish amali** deb ataladi.
- Mantiqiy ko'paytirishni ifodalaydigan quyidagi jadvalga **rostlik jadvali** deyiladi:

Quyidagi sodda mulohazalar berilgan bo'lsin:

A = "Mantiq algebrasining asoschisi Jorj Bul hisoblanadi";

B = "Klod Shennonning tadqiqotlari mantiq algebrasini hisoblash texnikasida qo'llash imkonini bergan".

Mantiqiy ko'paytmaning natijasi. Mantiq algebrasining asoschisi Jorj Bul hisoblanadi va Klod Shennonning tadqiqotlari mantiq algebrasini hisoblash texnikasida qo'llash imkonini bergan.

Natijaviy mulohazaning qiymati: rost.

Agar ikkala boshlang'ich mulohaza bir vaqtning o'zida rost bo'lsagina, "Mantiq algebrasining asoschisi Jorj Bul hisoblanadi va Klod Shennonning tadqiqotlari mantiq algebrasini hisoblash texnikasida qo'llash imkonini bergan", degan yangi mulohaza ham rost bo'ladi.

Konyunksiya amali faqat ikkita sodda mulohazalar ustida emas, balki bir nechta mulohazalar ustida ham bajarilishi mumkin. Rostlik jadvaliga boshlang'ich mulohazalarning barcha mumkin bo'lgan qiymatlari (A va B ustunlarga) kiritiladi. Odatda, mulohazalarning ikkilik sanoq sistemasidagi qiymatlari jadvalga o'sish (00, 01, 10, 11) tartibida joylashtiriladi. Oxirgi ustun tegishli operandlar (amallar bajariladigan ma'lumotlar elementi) uchun bajarilgan mantiqiy amal natijasini o'z ichiga oladi.

A va B , A and B , $A \wedge B$, $A \cdot B$, $A \cap B$, $A \& B$ ko'rinishlardan biri orqali ikkita A va B mulohazalar konyunksiyasi belgilanadi. 

Ikkita sodda mulohazaning "yoki" bog'lovchisi orqali bog'lanishidan hosil bo'lgan yangi mulohazaga **sodda mulohazalar yig'indisi** deyiladi.

Eslab qoling!

- 2-ta'rif: Ikkita A va B sodda mulohazaning kamida bittasi rost bo'lganda rost, qolgan holatlarda yolg'on bo'ladigan yangi (murakkab) mulohazani hosil qilish amali **dizyunksiya** (lot. disjunctio – farqlayman, ajrataman) – **mantiqiy qo'shish amali** deb ataladi.

Mantiqiy qo'shish amaliga mos **rostlik jadvali** quyidagi ko'rinishda bo'ladi:

Quyidagi sodda mulohazalar berilgan bo'lsin: A = "Mantiqda matematik belgilardan foydalanish g'oyasi Gotfrid Vilgelm Leybnisga tegishli";

B = "Leybnis binar arifmetikaning asoschisidir".

Mantiqiy qo'shishning natijasi. Mantiqda matematik belgilardan foydalanish g'oyasi Gotfrid Vilgelm Leybnisga tegishli yoki Leybnis binar arifmetikaning asoschisidir.

Natijaviy mulohazaning qiymati: yolg'on.

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

A	B	A ∨ B
0	0	0
0	1	1
1	0	1
1	1	1

DIQQAT

! **A yoki B, A or B, A \vee B, A + B, A \cup B** ko'rinishlardan biri orqali ikkita A va B mulohaza dizyunksiyasi belgilanadi.

Berilgan A mulohazaga "emas" shaklidagi to'liqsiz fe'lni qo'shish orqali hosil qilingan yangi mulohazaga **sodda mulohazaning inkori** deyiladi.

Eslab qoling!

- 3-ta'rif: A mulohazani qiymati rost bo'lganda yolg'on, yolg'on bo'lganda rost qiymatga o'zgartira oladigan amalga **inversiya** (lot. *inversio* – to'ntaraman) – **mantiqiy inkor amali** deyiladi.

Inkor amaliga mos **rostlik jadvali** quyidagi ko'rinishda bo'ladi:

Inkor amali har bir mulohazaning asl ma'nosiga qarama-qarshi bo'lgan yangi mulohazani hosil qiladi. Masalan, A = "Bizning uyimiz shahar markazida joylashgan" mulohazaning inkori $\neg A$ = "Bizning uyimiz shahar markazida joylashgan emas" bo'ladi.

A	$\neg A$
0	1
1	0

! **A emas, not A, $\neg A$, \bar{A}** ko'rinishlardan biri orqali A mulohazaning inversiyasi belgilanadi.

Eslab qoling!

- 4-ta'rif: A mulohaza rost, B mulohaza yolg'on bo'lgandagina yolg'on, qolgan holatlarda rost bo'ladigan mulohazaga A hamda B mulohazalarning **implikatsiyasi** deyiladi.

" \Rightarrow " belgi **implikatsiya belgisi** deb ataladi. **$A \Rightarrow B$** mantiqiy ifoda "Agar A bo'lsa, u holda B bo'ladi" yoki "A mulohazadan B mulohaza kelib chiqadi", degan ma'nolarni anglatadi.

Implikatsiya amaliga mos **rostlik jadvali** quyidagi ko'rinishda bo'ladi:

Implikatsiya so'zi mahkam bog'layapman degan ma'noni anglatadi.

Masalan: **A** = "Agar 72 soni 9 ga karrali bo'lsa, u holda bu son 3 ga ham karrali bo'ladi". A mulohazaning implikatsiyasi rost, chunki murakkab mulohaza tarkibidagi sodda mulohazalarning ikkalasi ham rost.

A	B	$A \Rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

B = "Agar $-3 < -1$ bo'lsa, u holda $9 < 8$ bo'ladi". B mulohazaning implikatsiyasi yolg'on, chunki $-3 < -1$ shart – rost, $9 < 8$ esa yolg'on.

! **$A \Rightarrow B$, $A \rightarrow B$** ko'rinishlardan biri orqali A mulohazaning implikatsiyasi belgilanadi.

Eslab qoling!

- 5-ta'rif: A va B mulohazalar bir vaqtda rost yoki bir vaqtda yolg'on bo'lganda rost bo'ladigan mulohazaga A va B mulohazalarning **ekvivalensiyasi** deyiladi.

" \Leftrightarrow " belgi **ekvivalensiya belgisi** deb ataladi. $A \Leftrightarrow B$ yozuv "A mulohazadan B mulohaza va B mulohazadan A mulohaza kelib chiqadi" yoki "A bo'ladi, faqat va faqat shu holdaki, agar B bo'lsa" yoki "A ekvivalent B" deb o'qiladi.

Ekvivalensiya amaliga mos **rostlik jadvali** quyidagi ko'rinishda bo'ladi:

Masalan, **A** = "972 soni 9 ga karrali", **B** = "972 soni raqamlarining yig'indisi 9 ga karrali" mulohazalari berilgan bo'lsin. U holda A va B mulohazalarning ekvivalensiyasi "972 soni 9 ga karrali bo'ladi, faqat va faqat shu holdaki, qachon bu son raqamlarining yig'indisi 9 ga karrali bo'lsa" kabi bo'ladi. Bu ekvivalensiya – rost.

A	B	$A \Leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

$A \Leftrightarrow B$, $A \leftrightarrow B$ ko'rinishlardan biri orqali A mulohazaning ekvivalensiyasi belgilanadi.

DIQQAT

Ixtiyoriy murakkab mulohazani mantiqiy ifoda ko'rinishida ham yozish mumkin. Murakkab mantiqiy ifodalar mantiqiy amallar yordamida bog'langan bir yoki bir necha oddiy (murakkab) mantiqiy ifodalardan tashkil topadi. Bu mantiqiy ifodalar mantiqiy o'zgaruvchilar, munosabatlar, mantiqiy amallar va qavslarni o'z ichiga oladi. Masalan, $(A \vee \neg B) \& (C \Leftrightarrow \neg D)$

Eslab qoling!

- Mantiqiy ifodalarda mantiqiy amallar quyidagi ketma-ketlikda bajariladi: inversiya (\neg); konyunksiya (&); dizyunksiya (\vee); implikasiya (\Rightarrow); ekvivalensiya (\Leftrightarrow).

Teng kuchli yoki bir xil amallar ketma-ketligi bajarilayotganda, amallar chapdan o'ngga qarab tartib bilan bajariladi. Ifodada qavslar ishtirok etganda, dastlab qavslar ichidagi amallar bajariladi. Ichma-ich joylashgan qavslarda eng ichkaridagi qavs ichidagi amallar birinchi bajariladi.

Mantiqiy amallarga misollar keltiramiz.

1-misol. A mulohaza yolg'on qiymat qabul qilsa, "(A EMAS) yoki A" mulohazaning qiymatini aniqlang.

Yechish. A "yolg'on" qiymat qabul qilganligi uchun (A EMAS) "rost" qiymatga ega bo'ladi. U holda "rost" va "yolg'on" qiymatlarning yig'indisidan ("YOKI" amali) "rost" natijaga ega bo'lamiz. Demak, javob "rost" ekan. Javob: rost.

2-misol. $x = 1,6$ va $y = 8,7$, A = "rost" va B = "yolg'on" bo'lganda, $(A \vee \neg B) \& (x > y)$ mantiqiy ifodaning qiymatini hisoblang.

Yechish. B mulohazaning qiymati "yolg'on" bo'lganligidan $\neg B$ mulohazaning qiymati "rost" bo'ladi. A mulohazaning qiymati "rost" va $\neg B$ mulohazaning qiymati "rost" bo'lganligidan, $(A \vee \neg B)$ mulohaza qiymati "rost" bo'ladi. $(1,6 > 8,7)$ munosabat noto'g'ri ekanligidan bu mulohaza "yolg'on" bo'ladi. U holda $(A \vee \neg B) \& (x > y)$ mantiqiy ifoda qiymati "yolg'on" bo'ladi. Javob: yolg'on.



1. Mulohazalar ustida bajariladigan qanday mantiqiy amallar mavjud?
2. Mantiqiy ifodalar qanday hosil qilinadi?
3. Mantiqiy qo'shish amali va uning rostlik jadvali haqida so'zlab bering.
4. Mantiqiy ko'paytirish amali va uning rostlik jadvali haqida so'zlab bering.
5. Mantiqiy ifodaning rostlik jadvalini tuzishda nimalarga e'tibor berish kerak?
6. Quyidagi mulohazalarni mantiqiy ifoda ko'rinishida yozing: "O'quvchilar fizika darsida laboratoriya ishini bajarishdi va tajriba natijalarini o'qituvchiga yetkazishdi".



1. $A = \text{rost}$, $B = \text{yolg'on}$, $C = \text{rost}$ qiymatlar uchun quyidagi amallarni bajaring:
 - a) $A \vee B \& C$; c) $B \vee (C \& A)$;
 - b) $B \vee \neg C$; d) $\neg (A \& B) \vee (B \Rightarrow C \vee \neg A)$.
2. Agar $A = \text{yolg'on}$, $B = \text{"Inversiya mantiqiy inkor amali hisoblanadi"}$, $C = 3, 14$, $D = 7, 9$ bo'lsa, quyidagi amallarni bajaring:
 - a) $(D = C) \& A \& B$; c) $A \vee (C < D) \& A \vee B$;
 - b) $B \Leftrightarrow (C > D) \& A$; d) $\neg (A \& B) \Rightarrow ((C + D) > 16)$.

4-DARS. MANTIQUIY IFODALARNING ROSTLIK JADVALINI TUZISH

Formal, ya'ni formulalarga tayangan tilda mantiqiy bog'lovchilar deb ataluvchi maxsus belgilar ($\&$, \vee , \neg , \Rightarrow , \Leftrightarrow)dan foydalaniladi. Biz bu amallar, ularning yozilishi hamda o'qilishi bilan avvalgi darslarda tanishib chiqqan edik.

Mantiqiy formulalar **rostlik jadvallari** yordamida izohlanadi. Bunday jadvallar mantiqiy bog'lovchi orqali tuzilgan murakkab mulohazaning rost(1) yoki yolg'on(0)ligini tashkil etuvchi mulohazalarning rostligiga qarab aniqlanadi. Mantiqiy amallarning rostlik jadvallaridan foydalanib, murakkabroq mulohazalar uchun rostlik jadvalini tuzish mumkin.

Rostlik jadvalini tuzishda amallarning bajarilish tartibiga rioya qilish shart. Mulohazalar to'plamida avval inkor amali, keyin konyunksiya, dizyunksiya, implikasiya va nihoyat ekvivalensiya amallari bajariladi. Agar inkor amali qavslardan tashqarida bo'lsa, u holda dastlab qavs ichidagi amallar bajariladi. Teng kuchli yoki bir xil amallar ketma-ketligi bajarilayotganda, amallar chapdan o'ngga tomon tartib bilan bajariladi. Ichma-ich joylashgan qavslarda eng ichkaridagi qavs ichidagi amallar birinchi bajariladi.

Eslab qoling!

Mulohazalar to'plami va unda bajariladigan barcha $\&$, \vee , \neg , \Rightarrow , \Leftrightarrow amallar birgalikda mulohazalar algebrasi deb yuritiladi.

A, B, C, \dots mulohazalarni mantiqiy bog'lovchilar bilan ma'lum tartibda birlashtirib hosil qilingan murakkab mulohazaga **mantiqiy formula** deyiladi.

Mantiqiy ifodaning rostlik jadvalini tuzish ketma-ketligini ko'rib chiqamiz:

1. Mantiqiy ifoda tarkibida o'zgaruvchilar soni n ni aniqlash:

$A \vee A \& B$. O'zgaruvchilar lotin alifbosi harflari bilan belgilanishini bilamiz. $n = 2$.

2. Mantiqiy amallar soni k ni aniqlash:

Qatnashgan mantiqiy amallar: \vee , $\&$. $k = 2$.

3. Qavslar va amallarning bajarilish ketma-ketligini hisobga olgan holda mantiqiy amallarning bajarilish tartibini aniqlash:

- 1) $A \& B$;
- 2) $A \vee (A \& B)$.

4. Jadvaldagi ustunlar sonini aniqlash. O'zgaruvchilar soniga amallar soni qo'shiladi: $c = n + k$:

$$2 + 2 = 4.$$

5. O'zgaruvchilar va 3-bandda aniqlangan amallar bajarilish ketma-ketligiga muvofiq jadvalning 1-satri (sarlavha satri) to'ldiriladi:

A	B	A & B	A \vee A & B
---	---	-------	----------------

6. Jadval satrlari soni $r = 2^n$ formula yordamida aniqlanadi (5-banddagi sarlavha satri hisobga olinmaydi):

$$r = 2^2 = 4.$$

Kirish o'zgaruvchilari to'plamini 0 dan $2^2 - 1 = 3$ gacha.
Ikkilik raqamlar: 00, 01, 10, 11

A	B	A & B	A \vee A & B

7. O'zgaruvchilar to'plamining n -razryadli ikkilik raqamlari yoziladi:

A	B	A & B	A \vee A & B
0	0		
0	1		
1	0		
1	1		

8. Kiruvchi ma'lumotlarga mos jadval ustunlarida belgilangan mantiqiy amallar ketma-ketligini bajarish va jadvalni to'ldirish. Ya'ni mantiqiy amallarga qarab, kirish ma'lumotlariga mos mantiqiy amallarni bajarish:

A	B	A & B	A \vee A & B
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

Rostlik jadvalini tuzishga oid misolni ko'ramiz.



1-misol: $((A \vee B) \& (\neg A)) \Rightarrow B$ mulohazaning rostlik jadvalini tuzing:

Yechish:

- Ushbu ifodada o'zgaruvchilar soni $n = 2$ ga teng. A, B o'zgaruvchilar ishtirok etgan.
- Qatnashgan mantiqiy amallar: $\vee, \neg, \&, \Rightarrow$. $k = 4$.
- Amallar ketma-ketligi:
 - $A \vee B$; 2) $\neg A$; 3) $(A \vee B) \& (\neg A)$; 4) $((A \vee B) \& (\neg A)) \Rightarrow B$.
- Jadvalda ustunlar soni $c = 2 + 4 = 6$.
- Jadvalning 1-satri hosil qilinadi:

A	B	$A \vee B$	$\neg A$	$(A \vee B) \& (\neg A)$	$((A \vee B) \& (\neg A)) \Rightarrow B$
---	---	------------	----------	--------------------------	------------------------------------------

6. Jadvaldagi satrlari soni $r = 2^2 = 4$.

7. O'zgaruvchilar to'plamining n-razryadli ikkilik raqamlari yoziladi:

A	B	$A \vee B$	$\neg A$	$(A \vee B) \& (\neg A)$	$((A \vee B) \& (\neg A)) \Rightarrow B$
1	1				
1	0				
0	1				
0	0				

8. Rostlik jadvali to'ldiriladi:

A	B	$A \vee B$	$\neg A$	$(A \vee B) \& (\neg A)$	$((A \vee B) \& (\neg A)) \Rightarrow B$
1	1	1	0	0	1
1	0	1	0	0	1
0	1	1	1	1	1
0	0	0	1	0	1



- Mulohaza nima? Sodda mulohaza bilan murakkab mulohazaning farqini tushuntiring?
- Asosiy mantiqiy amallarni aytib bering.
- Mantiqiy amallarning bajarilish tartibini ayting.
- Rostlik jadvali nima? Unga qanday elementlar kiritiladi?
- Rostlik jadvalini tuzish ketma-ketligini aytib bering.
- Qaysi mantiqiy amallarda kamida ikkita mulohaza ishtirok etishi kerak?



- Quyidagi mantiqiy mulohazalarga mos rostlik jadvalini tuzing:
 - $B \& (A \vee B)$; 2. $A \& (A \vee B \vee C)$; 3. $\neg A \& B \vee \neg C$;
 - $(A \vee B) \& \neg A$; 5. $B \Leftrightarrow (\neg C \vee D) \& A$; 6. $\neg (A \& B \Rightarrow C) \vee (B \Rightarrow C \vee \neg A)$.
- $((C \vee B) \Rightarrow B) \& (A \& B) \Rightarrow B$ mantiqiy ifodaning rostligini aniqlang.

5–6-DARSLAR. MANTIQIY SXEMALAR

Mantiq algebrasi matematikaning bir bo'limi hisoblanib, avtomatik qurilmalarni loyihalashtirishda, axborot va kommunikatsiya texnologiyalarining apparatli va dasturiy vositalarini ishlab chiqishda muhim o'rin tutadi. Ma'lumki, har qanday axborot diskret ko'rinishda, ya'ni alohida qiymatlarning fiksirlangan (belgilangan) to'plami ko'rinishida taqdim etilishi mumkin.

Diskret qayta ishlovchi qurilma ikkilik signallarni qayta ishlaganidan keyin biror mantiqiy amalning qiymatini chiqarsa, u **mantiqiy element** deb ataladi. Bunday qiymatlar (signallar)ni qayta ishlovchi qurilmalarga esa **diskret qurilmalar** deyiladi.

Mantiqiy elementlar kompyuterning tarkibiy qismi bo'lib, ikkilik o'zgaruvchilar ustida muayyan mantiqiy amallarni bajarish uchun mo'ljallangan elementlar hisoblanadi.

Zamonaviy raqamli texnologiyalarning barcha hisoblash qurilmalari (kompyuter, mobil qurilmalar) mantiqiy elementlarga asoslangan. Kompyuterning har qanday mantiqiy amali asosiy mantiqiy elementlar yordamida bajariladi. Har bir mantiqiy element bir yoki bir necha mantiqiy amalning bajarilishini ta'minlaydi.

Quyida eng sodda va keng tarqalgan mantiqiy elementlar bilan tanishamiz.

Elementlarning o'zi oddiy elektr sxemalardan tuziladi. Bunda sxemaning kirish qismiga kelgan signallarga **argument** deyilsa, chiqishdagi signallar esa **argumentning funksiyasi** bo'ladi. Sxemaning ma'lum qismida signalning mavjud bo'lishi bir (1)ni, mavjud emasligi esa nol (0)ni ifodalaydi.

"VA(&)" va "YOKI" mantiqiy funksiyalarining bajarilishi uchun kiruvchi signallar soni kamida ikkita bo'lishi zarur. Ayrim hollarda esa kiruvchi signallar soni ikkitadan ko'p bo'lishi ham mumkin.

Kompyuterning bazaviy mantiqiy elementlari, asosan, uchta mantiqiy amalni bajaradi:

- 1) **konyunktor** ("VA" mantiqiy elementi) mantiqiy ko'payishni amalga oshiradi;
- 2) **dizyunktor** ("YOKI" mantiqiy elementi) mantiqiy qo'shishni amalga oshiradi;
- 3) **invertor** ("EMAS" mantiqiy elementi) rad etishni amalga oshiradi.

Quyidagi jadvalda ikkita kiruvchi A va B elementlar uchun "VA" mantiqiy elementining rostlik jadvali ko'rsatilgan. Ko'rinib turibdiki, ikkala kiruvchi element bir vaqtning o'zida "1" kirish signali bilan ta'minlangandagina chiquvchi signal orqali "1" signali hosil bo'ladi. Boshqa uchta holatda chiqish signali nolga teng bo'ladi.

Konyunktor – mos tushish sxemasida kamida ikkita (A, B) kiruvchi va bitta (A&B) chiquvchi signal mavjud. Raqamli sxemalarda "VA" mantiqiy elementi 1-rasmda ko'rsatilgandek belgilanadi.

Xorij sxemalarida "VA" elementining belgisi boshqacha ko'rinishga ega (2-rasmga qarang). Uni qisqacha **AND** elementi deb atashadi.

Eslab qoling!

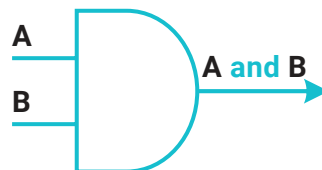
- **Mantiqiy o'zgaruvchi** deb, faqat ikkita: 0 va 1 qiymatni qabul qiluvchi kattalikka aytiladi.
- **Mantiqiy funksiya** deb, argumentlari faqat 0 va 1 qiymatni qabul qiluvchi funksiyaga aytiladi.

Dizyunktor – yig'uvchi sxemada ham kamida ikkita (A, B) kiruvchi va bitta (A **yoki** B) chiquvchi signal mavjud.

Ikkita (A, B) kiruvchi signal uchun "YOKI" mantiqiy elementi biroz boshqacha ishlaydi.



1-rasm. "VA" mantiqiy elementi



2-rasm. "AND" elementi

Kiruvchi A	Kiruvchi B	Chiquvchi (A&B)
0	0	0
1	0	0
0	1	0
1	1	1

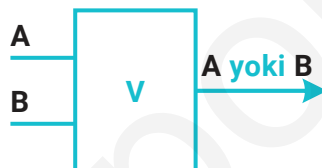
Yig'uvchi sxemada ikkita kiruvchi signalning ixtiyoriy bittasi "1" kirish signali bilan ta'minlansagina, chiquvchi signal orqali "1" signali hosil bo'ladi. Boshqa holatda chiqish signali nol (0)ga teng bo'ladi.

Raqamli sxemalarda

"YOKI" mantiqiy elementi 3-rasmda ko'rsatilgandek belgilanadi.

Xorij sxemalarida

"YOKI" elementining belgisi 4-rasmdagi kabi ko'rinishga ega. Uni qisqacha "OR" elementi deb atashadi.



3-rasm. "YOKI" mantiqiy elementi



4-rasm. "OR" elementi

Kiruvchi A	Kiruvchi B	Chiquvchi (A yoki B)
0	0	0
1	0	1
0	1	1
1	1	1

Invertor sxemasida

faqat bitta (A) kiruvchi va bitta (A emas) chiquvchi signal mavjud. Invertor sxemasi "teskari zanjir" deb ham ataladi.

Invertor sxemasida kiruvchi signalning qiymati qarama-qarshisiga o'zgaradi, Masalan, kiruvchi signal "1" kirish signali bilan ta'minlansa, chiquvchi signal orqali "0" signali hosil bo'ladi va aksincha.

Kiruvchi A	Chiquvchi (A emas)
0	1
1	0

Raqamli sxemalarda "EMAS" mantiqiy elementi 5-rasmda ko'rsatilgandek belgilanadi.

Xorij sxemalarida "EMAS" elementining belgisi 6-rasmdagi kabi ko'rinishga ega. Uni qisqacha "NOT" deb atashadi.



5-rasm. "EMAS" mantiqiy elementi



6-rasm. "NOT" elementi

Mantiqiy elementlardan arifmetik amallarni bajarish va axborotlarni saqlash uchun mo'ljallangan murakkab raqamli sxemalar yaratiladi. Bir necha mantiqiy element va ularning turli birikmasi yordamida berilgan funksiyalarni bajarishga qodir sxemalarni tuzish mumkin.

Eslab qoling!

Mantiqiy sxema – kompyuter qurilmalarining ishlashini tavsiflovchi har qanday mantiqiy funksiyani bajaradigan elektron qurilma.

1-misol. $A \& B \vee \neg(B \vee A)$ mantiqiy ifodaga mos sxemani chizing va kirish signallar qiymatlari $A=1$ va $B=0$ holat uchun chiquvchi signal qiymatini hisoblang.



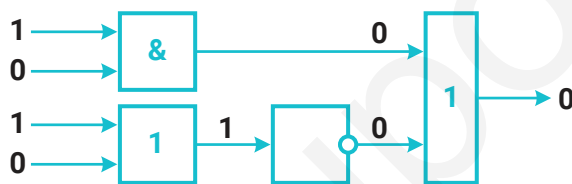
GURUHDA ISHLANG

Yechish:

- 1) Ifodada ikkita o'zgaruvchi mavjud: A va B.
- 2) Ifodada to'rtta mantiqiy amal bor: konyunksiya, ikkita dizyunksiya va 1 ta inkor. Amallarning bajarilish tartibi quyidagicha:

$$\begin{array}{cccc} 3 & 4 & 2 & 1 \\ A \& B \vee \neg & (B \vee A) \end{array}$$

- 3) Sxema mantiqiy amallarni bajarish tartibiga muvofiq chapdan o'ngga qarab chiziladi (7-rasm).



7-rasm. Mantiqiy sxemani chizish

- 4) Chiquvchi signal qiymatini hisoblash: $1 \& 0 \vee \neg(0 \vee 1) = 0$. Mantiqiy sxemaning to'g'riligini <https://www.semestr.online> web-sahifasi yordamida tekshirish mumkin. Avval video ko'rsatmalarni ko'rib chiqing.

Mantiqiy sxemani chizishning ketma-ketligi:

- 1) mantiqiy o'zgaruvchilar sonini aniqlash;
- 2) asosiy mantiqiy amallar soni va ularni bajarish ketma-ketligini aniqlash;
- 3) har bir mantiqiy amal uchun tegishli mantiqiy elementni ko'rsatish;
- 4) mantiqiy amallarning bajarilish tartibida mantiqiy elementlarni o'zaro bir-biri bilan ulash.



DIQQAT

1. Mantiqiy element nima?
2. Asosiy mantiqiy elementlar nomini ayting, ularni sxemalarda tasvirlang.
3. Konyunktor elementi qanday vazifani bajaradi?
4. Dizyunktor elementi qanday vazifani bajaradi?
5. Invertor elementi qanday vazifani bajaradi?
6. Mantiqiy sxemalar nima uchun qurilgan?
7. Mantiqiy sxemalarni yaratish algoritmini aytib bering.



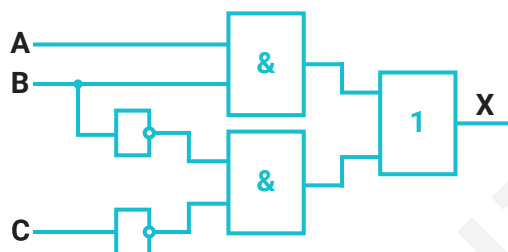
SAVOL VA TOPSHIRIQLAR



1. Quyidagi mantiqiy ifodalarga mos sxemalarni chizing.

- a) $D \vee \neg B \& A \& (B \vee \neg A)$;
- b) $A \& B \vee (\neg C)$;
- d) $A \vee B \& (\neg C \vee E)$;
- e) $A \vee \neg B \& C$;
- f) $\neg (A \& C) \vee \neg B$;
- g) $\neg A \& (\neg B \vee C)$;

2. Quyidagi sxemaga mos mantiqiy ifodani yozing:



7-DARS. AMALIY MASHG'ULOT



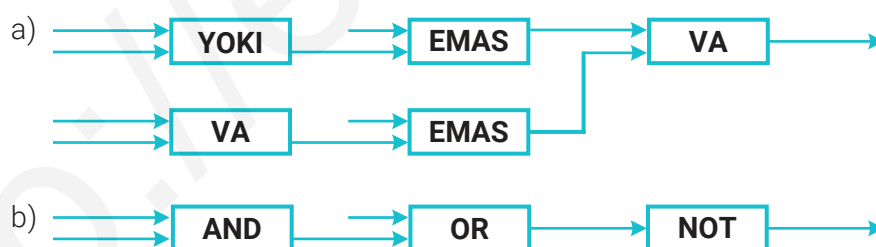
1. Keltirilgan gaplarning qaysilari mulohaza bo'la oladi?

- a) Eng yengil metall litiy hisoblanadi;
- b) Afrika Yevrosiyodan keyin ikkinchi yirik qit'adir;
- d) O'zbekiston Respublikasining birinchi Prezidenti Islom Karimov 1938-yil 30-yanvar kuni Samarqand shahrida tug'ilgan;
- e) Yozda ko'pincha qor yog'adi;
- f) $5 \cdot 2 < 14,6 + 3$;
- g) Rastrli grafika – informatikaning eng qiziqarli mavzularidan biri;
- h) Maktabga ketayotib, eshiklarni qulflang.

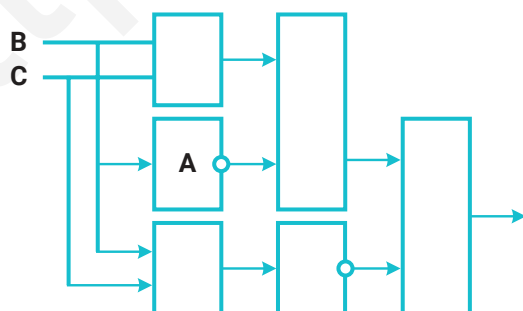
2. Mantiqiy ifodalarning rostlik jadvali va sxemasini tuzing:

- a) $A \& (B \Rightarrow \neg C)$;
- b) $B \& \neg (C \vee \neg B \Rightarrow A)$;
- d) $(A \vee C) \Rightarrow (\neg B \vee A \& D)$;
- e) $B \Leftrightarrow (\neg C \vee D) \& A$;
- f) $(A \Rightarrow \neg D) \& (C \vee \neg B)$;
- g) $(A \vee C) \Leftrightarrow (\neg B \vee A \vee D)$;
- h) $\neg (A \Rightarrow B \vee C) \& (D \Rightarrow \neg B)$;
- i) $\neg (A \& B \Rightarrow C) \vee (B \Rightarrow C \vee \neg A)$.

3. Quyidagi sxemaning chiqishida 0 hosil bo'lishi uchun, kirishida qanday qiymatlar bo'lishi mumkin?



4. Berilgan sxemaga mos mantiqiy ifoda toping:



8–9-DARSLAR. MASALALARNI KOMPYUTERDA YECHISH BOSQICHLARI

Inson amaliy ish jarayonida ko'plab masalalarni hal etishiga to'g'ri keladi. Bu masalalarning ba'zilari osongina, ba'zilari esa murakkab hisob-kitoblar orqali hal etiladi. Ba'zi masalalarni yechishda esa qandaydir amallar guruhini minglab marta bajarishga to'g'ri kelishi mumkin. Shuning uchun beminnat va o'ta tez ishlaydigan yordamchimiz bo'lmish kompyuter bizga yaqindan yordam bera oladimi, agar yordam bera olsa, ularni kompyuterda hal etish qanday tashkil etiladi, degan savollar tug'ilishi tabiiy hol. Aslida esa kompyuter masalalarni tez hal qilish va ma'lumotlarni qayta ishlash uchun yaratilgan.

Kompyuter yordamida har qanday muammoni hal qilish quyidagi bosqichlarni o'z ichiga oladi:

- 1-bosqich: masalaning qo'yilishi;
- 2-bosqich: masalaning matematik modelini tuzish;
- 3-bosqich: algoritmlash;
- 4-bosqich: dasturlash;
- 5-bosqich: dasturni kompyuter xotirasiga kiritish;
- 6-bosqich: natija olish va tahlil etish.

1-bosqichda masalaning to'g'ri qo'yilganligi, maqsadi va mazmuni aniqlanadi. Barcha ko'rsatkichlar, ularning xususiyatlari o'rganib chiqiladi. Qanday natija olinishi kerakligi hamda masalaga mos boshlang'ich hamda natijaviy kattaliklar aniqlanadi. Masalaning aniq, to'liq va tushunarli bo'lishi muammoga to'liq yechim topish imkonini beradi.

2-bosqichda masala ko'rilayotgan sohaning ilmiy yutuqlaridan kelib chiqqan holda matematik munosabatlar orqali ifodalanadi, ya'ni barcha kattaliklar, ularning o'zaro bog'lanishi amalga oshiriladi va matematik model yaratiladi. Qo'yilgan masalani aniq va tushunarli hal qilish uchun kerakli matematik usul tanlanadi. Muammoni hal qilishda matematik usullarning turli xillaridan foydalanish mumkin hamda tanlangan usul, albatta, aniq yechimga olib borishi zarur.

3-bosqichda masalaning modelidan foydalanib, hal etishning algoritmi tuziladi. Muayyan ko'rsatmalarning ketma-ketligi algoritmi tasvirlash usullaridan biri orqali tasvirlanadi. Masalan, blok-sxema yoki so'zlar ko'rinishida ifodalash mumkin. Algoritm tayyor holatga keltirilgandan so'ng navbatdagi bosqichga o'tiladi.

4-bosqichda algoritmdagi ko'rsatmalarning ketma-ketligi tayyor holatga keltirilganidan keyin, uning asosida kompyuter bajara oladigan tilga o'tkaziladi.

5-bosqichda dasturlash tili yordamida tuzilgan dastur kompyuter xotirasiga kiritiladi.

6-bosqichda dastur ishga tushiriladi, natijasi tahlil qilinadi. Dasturning to'g'riligini tekshirish testlar yordamida amalga oshiriladi. Agar dasturda xato va kamchiliklar aniqlansa, ular bartaraf etiladi. Xatolar muayyan dasturlash tilida dastur yozish qoidalarini buzish bilan bog'liq bo'lishi mumkin. Jarayon boshlang'ich ma'lumotlar bilan solishtirilib, to'g'ri natija olingandan keyin to'xtatiladi.

1-masala. Massasi 150 g bo'lgan xokkey shaybasi muz ustida turibdi. Agar xokkeychi unga 100 N kuch bilan zarba bersa, shayba qanday tezlanish oladi?

Masalani tahlil etamiz: fizika kursidan ma'lumki, jismning tezlanishi unga ta'sir etayotgan kuchga to'g'ri proporsional, massasiga esa teskari proporsional. Bu Nyutonning ikkinchi qonuniga asoslanadi.

Berilgan:

$m = 150 \text{ g} = 150:1000 = 0,15 \text{ kg}$.
 $F = 100 \text{ N}$.

Formula:

$$a = \frac{F}{m}$$

Yechish:

$$a = \frac{100}{0,15} \cdot \frac{\text{N}}{\text{kg}} = 666,67 \frac{\text{m}}{\text{s}^2}$$

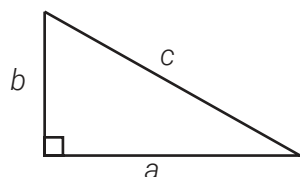
Topish kerak: $a = ?$

Javob: $666,67 \frac{\text{m}}{\text{s}^2}$.

2-masala. Dilnura oq qog'ozga ko'k qalam bilan tomonlari 12 cm va 9 cm bo'lgan to'g'ri burchakli uchburchak chizdi. Shu uchburchak perimetri va yuzasini toping.

Masalani tahlil etamiz: *birinchidan*, masala yechimini topish uchun Dilnura uchburchakni qanday rangli qalamda chizganligining ahamiyati yo'qligini, ya'ni bu biz uchun "keraksiz" axborot, *ikkinchidan*, uchburchakning **to'g'ri burchakli** bo'lishi muhim axborot ekanligini aniqlaymiz. Agar matematika kursidan Pifagor teoremasini esga olsak, u holda masalaning yechilishi quyidagi ko'rinishda ifodalanadi:

Chizmasi:



Berilgan:

$a = 12 \text{ cm}$.
 $b = 9 \text{ cm}$.

Topish kerak:

$P_{\text{uchb.}} - ?$
 $S_{\text{uchb.}} - ?$

Formular:

Gipotenuza: $c = \sqrt{(a^2 + b^2)}$.

Perimetr: $P_{\text{uchb.}} = a + b + c$.

Yuza: $S_{\text{uchb.}} = \frac{1}{2}ab$.

Yechish:

Pifagor teoremasidan: $c = \sqrt{(a^2 + b^2)} = \sqrt{((12 \text{ cm})^2 + (9 \text{ cm})^2)} = \sqrt{(225 \text{ cm}^2)} = 15 \text{ cm}$.

U holda: $P_{\text{uchb.}} = 12 \text{ cm} + 9 \text{ cm} + 15 \text{ cm} = 36 \text{ cm}$. $S_{\text{uchb.}} = 1/2(12 \text{ cm} * 9 \text{ cm}) = 54 \text{ cm}^2$.

Javob: 36 cm va 54 cm².

3-masala. Bunyod Informatikadan har bir sahifasida 40 ta satr, har bir satrida esa 67 ta belgi bo'lgan 46 sahifali referat tayyorladi. Agar Bunyod 51 Mbayt mobil internet paketiga ega bo'lsa, u o'qituvchiga o'zi tayyorlagan hujjatni elektron pochta orqali jo'nata oladimi?

Masalani tahlil qilishga o'tamiz.

Masalaning boshlang'ich qiymatlari:

- hujjat 46 sahifadan iborat;
- har bir satrdagi belgilar soni 67 ga teng;
- 1 ta sahifada 40 ta satr bor.

Masalaning maqsadi:

Eng avvalo, hujjat hajmini hisoblash va undan so'ng hujjatni elektron pochta orqali jo'nata olish mumkin yoki mumkin emasligini aniqlash.

Masala shartlariga mos matematik munosabatlarni hosil qilish:

Masalada 1 ta sahifadagi belgilar sonini x , barcha sahifalardagi belgilar sonini y bilan belgilaymiz. Masala shartiga ko'ra, Bunyod referat tayyorlash davomida har bir sahifaga

x tadan belgi yozgan bo'lsa, u holda referatning barcha sahifasida $y = x \cdot 46$ ta (x belgidan iborat 46 ta sahifa) belgi yozgan bo'ladi. Bu ma'lumot hujjat hajmini aniqlash uchun zarur. Informatikadan ma'lumki, bitta belgi – bir baytga teng. Bunyod ega bo'lgan 51 Mbayt mobil internet paketi orqali o'qituvchiga o'zi tayyorlagan hujjatni elektron pochta orqali jo'nata olishi uchun $y < 51$ tengsizlik o'rinli bo'lishi kerak. Ya'ni masalaning shartlariga mos tengsizlikni hosil qildik.

Bajariladigan amallar ketma-ketligi:

- 1) 1 ta sahifadagi belgilar sonini hisoblab olamiz: $x=67 \cdot 40=2680$ ta belgi;
- 2) hujjatdagi barcha belgilar sonini topamiz: $y=2680 \cdot 46=123280$ ta belgi;
- 3) Bunyod ega bo'lgan mobil internet hajmi Mbaytda bo'lganligi sababli, axborot hajmini Mbaytda ifodalaymiz:

$$123\ 280 : 1024: 1024 \text{ Mbayt} = 0,118 \text{ Mbayt};$$

- 4) $y < 51$ tengsizlikni tekshiramiz: $0,118 \text{ Mbayt} < 51 \text{ Mbayt}$ tengsizlik o'rinli.

Natijaning tahlili:

$0,118 \text{ Mbayt} < 51 \text{ Mbayt}$ tengsizlik o'rinli bo'lganligi sababli, Bunyod tayyorlagan referat faylini o'qituvchiga elektron pochta orqali jo'nata oladi.

Javob: ha, jo'nata oladi.

Yuqorida ko'rib chiqqan masalalar misolida qaysi bosqichlar amalga oshirilganini ko'rib chiqamiz:

1. Har bir masalada avval **masalaning qo'yilishi**, ya'ni masalaning to'g'ri qo'yilganligi, maqsadi va uning mazmuni, masalada berilgan boshlang'ich kattaliklar va natijaviy (topilishi kerak bo'lgan) kattaliklar aniqlanadi.

2. Masalani yechish uchun zarur bo'lgan **formulalar**, boshqacha aytganda **matematik munosabatlar** hosil qilinadi.

3. Masala yechimidagi **amallar** (formulalar, munosabatlar)ning **bajarilish ketma-ketligi** aniqlanadi.

4. **Natija olish** va **tahlil etish**.

Yuqorida keltirilgan masalalarni kompyuterda yechishning "Dasturlash" va "Dasturni kompyuter xotirasiga kiritish" bosqichlariga oid amallarni kompyuter tushunadigan tilga o'girish, kompyuter xotirasiga kiritish ma'lum bilim va malaka talab etganligi sababli keyingi darslarda bajariladi.

1. Kompyuterda masala yechish bosqichlarini sanab bering.
2. Nima uchun har bir masalaga algoritm tuzilishi zarur?
3. Masalaga mos boshlang'ich va natijaviy kattaliklar nima uchun aniqlanadi?
4. Kalkulyatorda hisob-kitob ishlari bajarilganda qanday xatoliklar yuzaga keladi?
5. Kvadrat tenglamani yechishda amallarni bajarish ketma-ketligini aytib bering.
6. Nima uchun olingan natija tahlil etiladi?
7. Masala shartiga mos tenglama tuzishga misol keltiring.

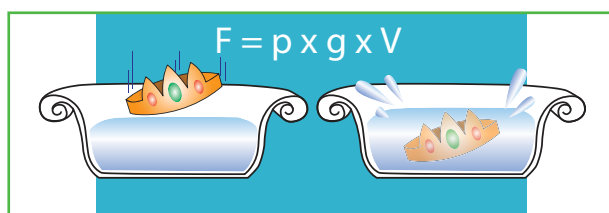




Quyidagi masalalar shartini tahlil qiling va bosqichlarga bo'lib yeching:

1. Elektr lampa qisqichlaridagi kuchlanish 110 V ga teng bo'lganda, lampadagi tok kuchi 0,5 A bo'ladi. Agar lampadan 0,25 A tok o'tsa, lampaga qanday kuchlanish berilgan?
2. Bir tomoni 24 cm, ikkinchi tomoni undan 8 cm qisqa, ular orasidagi burchagi esa 30 gradus bo'lgan uchburchak yuzini toping.
3. Og'irligi 4000 N bo'lgan beton plitani 10 cm ga ko'tarish uchun richag ishlatildi. Richagga 1200 N kuch qo'yildi. Richagning ikkinchi uchi 40 cm masofani o'tdi. Richagning foydali ish koeffitsiyentini toping. ($\eta = A_1/A_2 \cdot 100\%$)

10-11-DARSLAR. MODEL VA UNING TURLARI



Rasmga qarang. Unda siz nimalarni ko'rib turibsiz? Unda nima va qanday jarayon tasvirlangan? Formuladagi harflar nimalarni bildiradi?

Bizni qurshab turgan olam bu – obyektlar va hodisalar olami. Odatda, insonning e'tiborini tortadigan, uni qiziqtiradigan va o'rganilayotgan barcha narsa yoki jarayonga obyekt deyish mumkin. Ularning barchasi: o'simlik, hayvon, daryo, tog', mamlakat, qushlar parvozi va uy qurilishi inson uchun bilish obyektiga aylanadi. Biror narsa yoki hodisa haqida bilish deganda, o'sha narsa yoki hodisa haqidagi ba'zi ma'lumotlarni anglab olish tushuniladi.

Odatda, obyektlar insonning e'tiborini tortadigan jihatlari bilan nomlanadi. Obyekt – bu inson tomonidan idrok etiladigan atrofdagi voqelik (narsa (predmet), jarayon, hodisa)ning biror bir qismi.

Har bir obyektning uni boshqa obyektlardan ajratib turadigan nomi mavjud. Odatda, insonlar **“Bu kim?”** yoki **“Bu nima?”** degan savolga javob berish orqali har bir obyektning boshqa obyektlardan farqlash imkonini beruvchi nomga ega bo'ladilar.

Nomlar *ikkiga* bo'linib, *umumiy nom* – ko'p obyektlarni belgilash uchun va *xususiy nom* – ma'lum bir to'plamdagi ba'zi bir obyektning belgilash uchun ishlatiladi.

Bir turdagi o'rganilayotgan obyektlar o'zining xususiyatlari – **tavsifiga** ega bo'ladi.

Eslab qoling!

Obyekt (lot. *objectum* – narsa, anjom) haqiqatda mavjud bo'lgan, kuzatuv (o'rganish) jarayonining va subyektning (kuzatuvchining) e'tibori qaratilgan narsa (*predmet*), jarayon (voqea), hodisa.



Predmetga oid obyektlar

Har bir alohida olingan obyekt esa boshqasidan shu tavsifga mos **tavsif qiymati** bilan farqlanadi.

Obyekt to'g'risidagi xabarda ismga qo'shimcha ravishda, uning belgilari: xususiyati, xossasi, xatti-harakati, sharoiti, holati batafsil bayon qilinishi mumkin.

Quyidagi jadvalda obyektlar, ularning xususiyatlari, shuningdek, ushbu xususiyatlarga mos keladigan miqdor va qiymatlar ko'rsatilgan.



Jarayonga oid obyektlar



Hodisaga oid obyektlar

Obyekt nomi	Xususiyati	O'lchov birligi	Qiymati
inson	ko'k ko'z	ko'z rangi	ko'k
inson	uzun	bo'yi	180 cm
fayl	eski	yaratilgan vaqti	15.02.1980
fayl	grafik	tipi	.bmp tipli rasm

Inson har doim ham real obyekt bilan amalga oshirib bo'lmaydigan narsa yoki hodisalarni o'rganish orqali dunyoni bilishga harakat qiladi. Shuning uchun ham bu obyekt yoki hodisalarni tasvirlaydigan modellar yaratiladi va o'rganiladi. *Model* haqiqiy obyektning soddalashtirilgan ko'rinishi hisoblanadi.

Odatda, insonlar obyektlarning nafaqat xususiyatlari, balki uning boshqa obyektlar bilan *o'zaro munosabatlari, o'xshash jihatlarini* haqida ham gapirishi mumkin. Masalan: "*Nafisa – Sahobiddin akaning qizi*"; "*21 soni 3 ga karrali*"; "*Samarqand – Buxoro kabi qadimiy shahar*". Bu misollarning har birida ikki obyekt o'rtasidagi bog'liqlikni bildiruvchi munosabatlar nomi ta'kidlangan. Obyektlarning o'zaro munosabatlari nafaqat ikki obyekt o'rtasida, balki ko'plab obyektlar o'rtasida ham mavjud bo'lishi mumkin. Masalan: "**Disk** axborot *tashuvchi vosita hisoblanadi*"; "**Kamchatka** – yarim orol".

Aksariyat hollarda ma'lum bir sohaga oid izlanishlar olib borilayotganda haqiqiy obyekt emas, balki uning qandaydir ma'nodagi nusxasi o'rganiladi. Bunga, bir tomondan, ma'lum bir sabablarga ko'ra (chaqmoqning turg'un emasligi, quyoshning uzoqligi, obyekt bilan ishlash katta mablag' talab etishi yoki inson hayotiga xavf solishi va hokazo) haqiqiy obyektning to'g'ridan-to'g'ri o'rganishning imkoni bo'lmasa, ikkinchi tomondan, izlanishlar uchun obyektning qandaydir ma'nodagi nusxasini o'rganishning o'zi ham yetarli bo'ladi. Albatta, bunday hollarda obyektning nusxasi izlanish olib borilayotgan sohaning talablariga to'liq javob berishi kerak bo'ladi.

Model (lot. modulus – o'lchov, me'yor) – biror haqiqiy *obyekt* yoki *obyektlar tizimining* obrazi yoki nusxasi bo'lib, u izlanish olib borilayotgan sohaning ma'lum talablariga javob berishi zarur.

Modellashtirish – *bilish obyektlari (fizik hodisa va jarayonlar)ni ularning modellari yordamida tadqiq qilish, mavjud predmet va hodisalar modellarini yasash va o'rganishdan iborat jarayon.*

Hayotda obyektlarning modellariga juda ko'p misollar keltirish mumkin. Masalan, yerning modeli – globus yoki xarita.



Insoniyatni farovon hayot shart-sharoitlarini yaratish, tabiiy ofatlarni oldindan aniqlash muammolari qadimdan qiziqtirib kelgan. Shuning uchun ham u tashqi dunyoning turli hodisalarini o'rganishi tabiiy holdir.

Aniq fan sohasi mutaxassisleri u yoki bu jarayonning faqat ularni qiziqtirgan xossalari uchun o'rganadi. Masalan, geologlar yerning rivojlanish tarixini, ya'ni qachon, qayerda va qanday hayvonlar yashaganligi, o'simliklar o'sganligi, iqlim qanday o'zgarishganligini o'rganadi. Bu ularga foydali qazilma konlarini topishlarida yordam beradi.

Atrofimizdagi dunyoni o'rganish natijasida noaniq va to'liq bo'lmagan ma'lumotlar olinishi mumkin. Lekin koinotga uchish, atom yadrosining sirini aniqlash, jamiyatning rivojlanish qonunlarini egallash boshqalarga xalaqit bermaydi. Ular asosida o'rganilayotgan hodisa va jarayonning modeli yaratilishi va bu model ularning xususiyatlarini mumkin qadar to'laroq aks ettirishi zarur.

Haqiqiy obyekt va uning modeli o'tkazilayotgan tajribalarda bir xil natija bersagina, izlanish olib borilayotgan soha talablariga javob beradi. Masalan, samolyot va uning kichik nusxasi bo'lgan model bir xil aerodinamik qonunlarga bo'ysunadi. Model uchun topilgan natijalar haqiqiy samolyot uchun ham o'rinlidir. Loyihalashtirilgan haqiqiy samolyot qurilgach, uni laboratoriyadagi maxsus qurilmalar – samolyotga havo oqimini yuboruvchi stendlarda sinab ko'riladi. Bu holda laboratoriyadagi stendlar atmosferaning modeli bo'lib xizmat qiladi.

Hech qaysi model prototipning barcha xususiyatlarini va xatti-harakatlarini to'liq hisobga olmaydi, shuning uchun model asosida olingan natija haqiqatga yaqin keladi. Yaqinlashish darajasi modelning moslik darajasiga bog'liq. Modelni yaratishda inson, eng avvalo, obyektning eng muhim xususiyatlarini tanlashga intiladi, natijada sezilarli ta'sir ko'rsatmaydigan xususiyatlarni e'tiborsiz qoldiradi.

Endi modellarning ayrim turlari bilan tanishib chiqamiz.

1. Foydalanish sohasi bo'yicha: o'quv, tajriba, o'yin, imitatsion, ilmiy-tadqiqot modellari.

O'quv modellari o'qitish jarayonida ishlatiladi. Bularga ko'rgazmali qo'llanmalar, trenajyorlar, o'qitish dasturlari kiradi.

Tajriba modellari obyektning tadqiq qilish hamda uning bo'lajak tavsiflarini prognozlashtirish (bashorat qilish) uchun qo'llaniladi. Misol uchun, samolyot qanotining modeli uning havo oqimlariga qarshiligini o'rganish uchun aerodinamik quvurda "puflanadi";

Ilmiy-texnik modellar jarayon va hodisalarni tadqiq qilish uchun yaratiladi. Bunday modellarga momaqaldiroq elektr razryadini olish uchun qurilma, Quyosh tizimi sayyoralarining harakatlanish modeli, ichki yonish dvigatelining modellarini kiritish mumkin.

O'yin modellari – bu har xil o'yinlar: rivojlantiruvchi, iqtisodiy, harbiy. Bunday modellar yordamida ziddiyatli vaziyatlarni hal qilish, psixologik yordam ko'rsatish, turli vaziyatlarda obyektning harakat qilishini bashorat qilish mumkin.

Imitatsion modellar nafaqat turli darajadagi aniqlikda voqelikni aks ettiradi, balki uni imitatsiya (taqlid) qiladi. Model bilan tajriba turli boshlang'ich holatlarda ko'p marotaba takrorlanadi yoki turli sharoitlarga qo'yilgan shunga o'xshash obyektlar bilan bir vaqtda o'tkaziladi, ushbu tajriba-sinov ishlar mobaynida real holatda biror-bir harakatlarning oqibatlarini o'rganiladi va baholanadi. Tadqiqot natijalari bo'yicha xulosa qilinadi.

2. Vaqt omili (dinamikasi) bo'yicha: statik va dinamik modellar.

Statik modellar obyektning ma'lum vaqt ichida u bilan sodir bo'layotgan o'zgarishlarni hisobga olmasdan aks ettiradi. Bu modellarda vaqt omili yo'q.

Statik model sifatida hafta davomida o'rtacha havo haroratining o'zgarishi grafigi, vodorod va kislorod atomlaridan tashkil topgan suv molekulasining maketi yoki rasmi xizmat qilishi mumkin.

Dinamik modellar obyektning vaqt mobaynida o'zgarish jarayonini aks ettiradi. Masalan, ma'lum bir hudud xaritasi, poliklinikadagi bitta tekshiruv natijasi.

3. Taqdim etish usuli bo'yicha: axborot (nomoddiy, mavhum) va moddiy modellar.

Moddiy modellar modellashtirish obyektlarining moddiy nusxalaridir. Misol uchun: globus – yer shari shaklining modeli, qo'g'irchoq – inson tashqi ko'rinishining modeli, robot – zararli ishlab chiqarishda inson harakatlarining modeli.

Moddiy modellashtirishda anglashning eksperimental (tajriba) usuli qo'llanilsa, nomoddiy modellashtirishda anglashning nazariy usulidan foydalaniladi.

Axborot modeli – bu obyekt, jarayon yoki hodisa xususiyatlari va holatini tavsiflovchi ma'lumotlar yig'indisi. Axborot modellari, o'z navbatida, *belgili* (maxsus belgilar, ya'ni har qanday formal til vositalari bilan ifodalangan axborot modeli) va *verbal* (og'zaki yoki fikr shaklidagi axborot modeli) modellar va aralash axborot modellarga ajratiladi.

Belgili axborot modellari turli tillar (belgilar tizimlari)dan foydalangan holda quriladi. Belgili axborot modeli tabiiy tilda matn yoki dasturlash tilida dastur, formula (misol uchun, to'g'ri to'rtburchak yuzi $S=ab$) shaklida ifodalinishi mumkin. Belgili axborot modellariga misol tariqasida geografik xarita, grafik, diagramma va boshqalarni ko'rsatish mumkin.

Belgili axborot modellariga doir misollar		
		
Sxema	Turar joy rejasi	Chizma

Verbal modellar og'zaki yoki fikr shaklidagi axborot modeli hisoblanadi. Ular biror-bir axborot tashuvchisida (qog'oz, foto va kinoplyonka va b.) aks ettirilgan obyektlarning vizual shakllaridan iborat. Bu turdagi modellardan ta'lim sohasida (darsliklarda rasmlar, turli fanlar bo'yicha o'quv plakatlari) hamda obyektlarni tashqi belgilar bo'yicha tasniflash kerak bo'lgan fanlarda (botanika, biologiya, paleontologiya va b.) keng qo'llaniladi.

Informatikada kompyuter yordamida yaratish va tadqiq qilish mumkin bo'lgan modellar qo'llaniladi. Bu holda belgili modellar *kompyuter* va *nokompyuter* modellarga bo'linadi.

Kompyuter modeli dasturiy muhit vositalari bilan ifodalangan matematik model hisoblanadi. Hozirgi paytda kompyuter modellarning ikki turi mavjud:

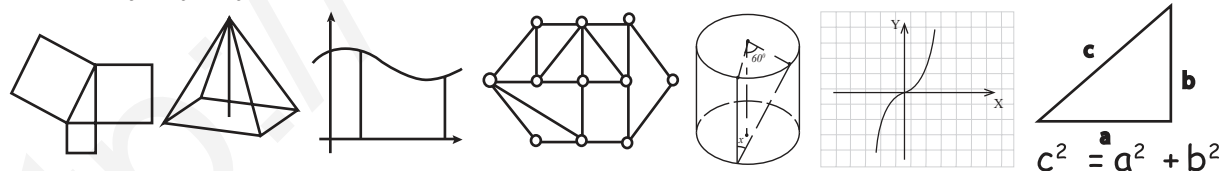
- tuzilmaviy-funksional modellar – kompyuter texnologiyalari yordamida tavsiflangan obyektning shartli ifodasi;
- imitatsion modellar – bu obyektning turli sharoitlarda ishlash jarayonlarini ifodalash imkonini beruvchi dastur yoki dasturlari majmui.



4. Obyektlarni ifodalash vositalari bo'yicha.

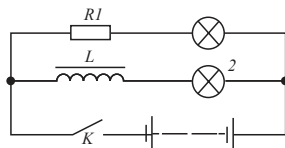
a) **Abstrakt modellar**, o'z navbatida, ikki guruhga bo'linadi: **matematik** va **iqtisodiy matematik** modellar.

Matematik modellar obyektning tuzilishi va o'zaro bog'lanish qonuniyatlarining matematik munosabatlari, formulalari va matematik-mantiqiy tavsifidan iborat. *Matematik modellar* jarayon yoki hodisani tavsiflovchi matematik belgilar tizimi hisoblanadi.



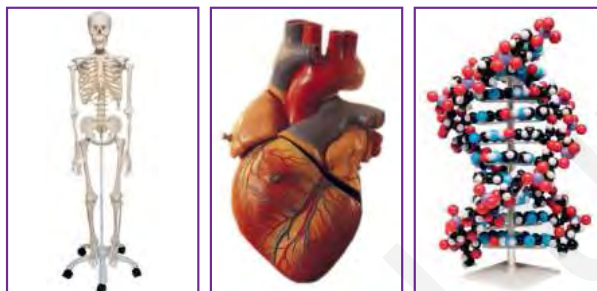
Iqtisodiy matematik modellar yordamida iqtisodiy taraqqiyotning eng umumiy qonuniyatlari tekshiriladi. Turli iqtisodiy ko'rsatkichlar, jumladan, milliy daromad, iste'mol, ish bilan bandlik, jamg'armalar, investitsiya ko'rsatkichlarining o'zgarishi va nisbatini tahlil qilish, uni oldindan aytib berish uchun murakkab iqtisodiy modellar qo'llaniladi. Aniq xo'jalik vaziyatlarini tekshirishda kichik iqtisodiy tizimlardan, murakkab iqtisodiy tizimlarni tekshirishda, asosan, matematik modellardan foydalaniladi.

b) **Fizik modellarda** obyektning tabiati va tuzilishi asl nusxasi kabi bo'ladi, lekin undan miqdor (o'lchami, tezligi va hokazo) jihatidan farq qiladi. Masalan, samolyot, kema, avtomobil, poyezd va boshqalarning modellari bo'lishi mumkin.



$$F = k \frac{q_1 \cdot q_2}{r^2}$$

d) **Biologik model** turli jonli obyektlar va ularning qismlari (hujayra, organizm va hokazo)ga xos biologik tuzilish, funksiya va jarayonlarni modellashtirishda qo'llaniladi. Biologik model odam va hayvonlarda uchraydigan ma'lum bir holat yoki kasalliklarni laboratoriya hayvonlarida sinab ko'rish imkonini beradi. Bunda shu holat yoki kasallikning kelib chiqish mexanizmi, kechishi, oqibati kabilar tajriba asosida o'rganiladi.



Bugungi kunda kompyuter yordamida biologik jarayonlarning modellarini kuzatish, ya'ni inson kasalligini aniqlash uchun **kompyuter tomografiyasidan** keng foydalanilmoqda.

1. Obyekt deb nimaga aytiladi?
2. Umumiy va xususiy nomga ega obyektlarga misollar keltiring.
3. Model deb nimaga aytiladi? Modellar necha turga bo'linadi?
4. Obyekt va unga mos modellarga misollar keltiring.
5. Matematik model nima? Misollar keltiring.
6. Belgili model nima? Unga misollar keltiring.
7. Matematik modelning boshqa modellardan farqini tushuntiring.
8. Axborot modellari nima? Unga misollar keltiring.



SAVOL VA TOPSHIRIQLAR

Mashqlar:

1. Quyidagi obyektlarga mos ularning xossalari va xususiyatlarini yozing:


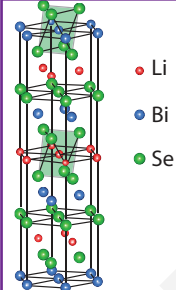


Obyekt	Xossasi	Xususiyati
Inson	Sariq sochli	Baland bo'lyi, ko'ngli ochiq, saxiy
Kitob		
Qattiq disk		
Monitor		
Kompyuter		
Fayl		

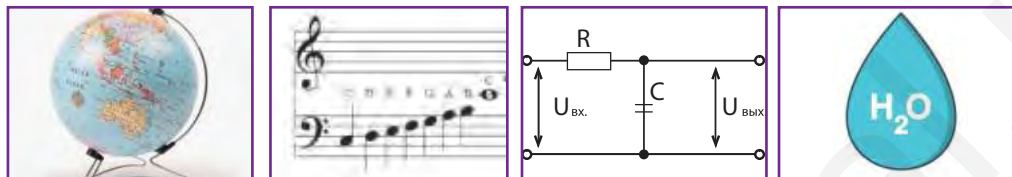
2. Inson tomonidan ko'rsatilgan harakatni bajaradigan ikkitadan obyektning nomini ayting: a) yig'ish; b) to'ldirish; d) ochish; e) bog'lash; f) taxlash; g) o'lchash; h) ushlab.

UY VAZIFASI

3. Chap va o'ng ustundagi so'zlarni o'zaro moslang:

Gazeta	Predmetga oid obyekt			
Kamalak				
Sayr	Jarayonga oid obyekt			
Stadion	Hodisaga oid obyekt			
O'qish				
Ziyofat				

4. Quyidagi rasmlar qanday model turiga kirishini ayting:



12-DARS. AMALIY MASHG'ULOT

Quyida iqtisodiy, fizik va biologik jarayonlarning matematik modellariga oid masalalarni tahlil qilamiz.

Masala. O'ylangan sonni topish bo'yicha masala (matematik fokus) modelini tuzing.

Yechish. Har biringiz ixtiyoriy sonni o'ylang va u bilan quyidagi amallarni bajaring:

- 1) o'ylangan son beshga ko'paytirilsin;
- 2) ko'paytmaga bugungi sanaga mos son (yoki ixtiyoriy boshqa son) qo'shilsin;
- 3) hosil bo'lgan yig'indi ikkilantirilsin;
- 4) natijaga joriy yil soni qo'shilsin.

Endi o'ylagan sonni topish matematik fokusiga mos modelni keltirib o'tamiz. Buning uchun, eng avvalo, masalani rasmiylashtirib olamiz:

X – o'ylangan son; **U** – hisoblash natijasi; **N** – sana; **M** – joriy yil.

Demak, yuqoridagi 1–4-ko'rsatmalarni $U=(X*5+N)*2+M$ formula orqali ifodalash mumkin. Bu formula masala (matematik fokus)ning *matematik modeli* bo'lib xizmat qiladi va **X** o'zgaruvchiga nisbatan chiziqli tenglamani ifodalaydi.

$X = (U - (M + 2N))/10$ tenglama yechilsa, o'ylangan sonni topish mumkin bo'ladi.

$X = (U - (M + 2N))/10$ formula esa o'ylangan sonni topish algoritmi hisoblanadi.



1. O'lchami 2048 x 1366 bo'lgan ekranda faqat 4 xil rangli tasvir aks etadi. Ekrandagi axborot hajmini aniqlang.

2. To'rtburchakning perimetri 54 cm va tomonlari 3 : 4 : 5 : 6 kabi nisbatda bo'lsa, uning tomonlari uzunliklarini toping.

3. Massasi 200 g bo'lgan jism 15 m/s tezlik bilan yuqoriga tik ravishda otildi. 1 s dan keyin jismning kinetik energiyasi va otilgan nuqtaga nisbatan potensial energiyasi qancha bo'ladi? $g=10$ m/s deb olinsin.

4. 9 % li eritma hosil qilish uchun 12 % li 200 mg eritmaga necha mg suv qo'shilishi kerak?

5. Toshkentdan yo'lga chiqqan "Afrosiyob" tezyurar poyezdi Buxoroga yetib kelguncha 550 km masofani bosib o'tdi. Ushbu masofa kartada 11 cm ga teng. Kartaning masshtabini aniqlang.

13–14-DARSLAR. ALGORITM TUSHUNCHASI VA UNING XOSSALARI

Inson hayoti davomida har kuni katta-kichik vazifa yoki masalalarni hal etishni, turli ko'rsatmalarni bajarishni, harakatlar rejasini tuzishni yoki tuzilgan rejaga muvofiq ishlarni amalga oshirishni o'z oldiga maqsad qilib qo'yadi. Masalan, bironta masalani yechish yo'llarini daftarga yozish, qandaydir bir taom yoki qandolat mahsulotini tayyorlashda unga oid ko'rsatma (retsept)-lardan foydalanish, maishiy texnika vositalarini ishlatishda uning yo'riqnomalaridan foydalanish, kimgadir biror manzilga yetib borish yo'lini tushuntirish va hokazo.

1. Biz muayyan harakatlarni amalga oshirishdan avval o'z oldimizga qanday maqsadni qo'yamiz?
2. Algoritm nima, uning kelib chiqish tarixini bilasizmi?
3. Algoritm ijrochisi nima, nimalar algoritm ijrochisi bo'lishi mumkin?



BUNI BILASIZMI?

Algoritm so'zi va tushunchasi IX asrda yashab ijod etgan buyuk alloma Abu Abdulloh Muhammad ibn Muso al-Xorazmiy (783 – 850) nomi bilan uzviy bog'liq. Algoritm so'zi al-Xorazmiy nomini Yevropa olimlari tomonidan buzib talaffuz qilinishidan kelib chiqqan. Al-Xorazmiy arifmetikaga bag'ishlangan "Al-kitob al-muxtasar fi hisob al-jabr va al-muqobala" nomli risolasida birinchi bo'lib o'nlik sanoq sistemasining prinsiplari va undagi to'rt amalni bajarish qoidalarini asoslab bergan. Olimning "al-Xorazmiy" nomi esa "algoritm" shaklida fanda abadiy o'rnashib qolgan.



Algoritmdagi har bir ko'rsatma yoki buyruq biror amalni bajarishni ko'zda tutadi. Algoritmdagi amallarni bajaradigan obyektни *ijrochi* deb atash mumkin. Har qanday algoritm amallarni belgilab beruvchi qoida bo'lib, uning zanjiri natijasida berilgan qiymatlardan izlangan natija kelib chiqadi. Bunday amallar zanjiri *algoritmik jarayon*, algoritm tarkibidagi umumiy jihatga ega har bir amal esa *algoritmning qadami* deb ataladi.

Algoritm deganda, ijrochi uchun qo'yilgan masalani yechishga qaratilgan aniq va tushunarli ko'rsatmalarning chekli ketma-ketligi tushuniladi.

Misol sifatida "Internetdan qiziqarli kitob topish va o'qish algoritmi"ni ko'rib chiqamiz:

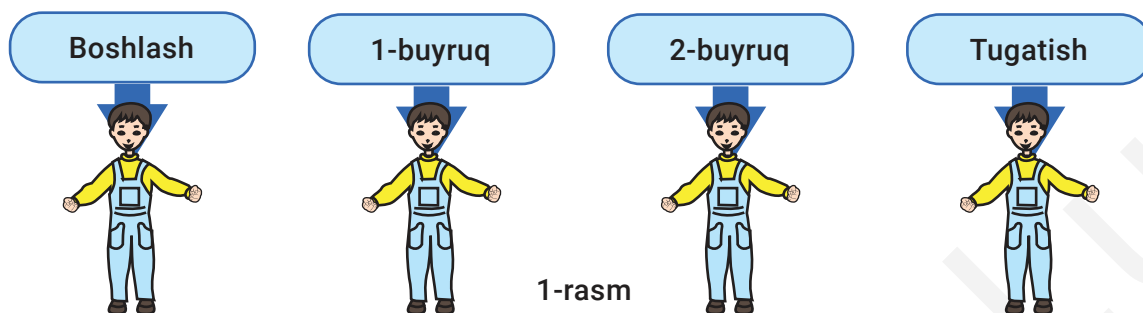
- 1) internet qidiruv tizimiga kirish;
- 2) qiziqarli kitobni qidirib topish;
- 3) topilgan kitobni yuklab olish;

Eslab qoling!

Odatda, inson o'z hayotiy tajribasi yoki o'zlashtirgan bilimlariga asoslanib, ma'lum bir harakatlar, boshqacha aytganda, ko'rsatma yoki buyruqlar ketma-ketligini bajarish orqali ko'zlagan maqsadiga erishadi. Bunday ko'rsatma va harakatlar ketma-ketligi algoritm so'zi bilan ifodalanadi.

- 4) kitobni o'qish;
- 5) o'qilgan kitob bo'yicha xulosa qilish.

Algoritmدا buyruq, ko'rsatma, amal, harakatlar ketma-ketligi bo'lishi mumkin (1-rasm). Algoritm bajarilishi zarur qadamlar ketma-ketligidan iborat bo'ladi. Algoritm to'g'ri tuzilgan bo'lsagina, to'g'ri natijaga erishish mumkin.



Misol sifatida qaralgan "Kitob o'qish algoritmi"da ham inson ko'zlagan maqsadiga erishish uchun ijrochi sifatida algoritmni bajarishi zarur. Undagi amallar, boshqacha aytganda, ko'rsatma yoki buyruqlar ketma-ketligi biror kishi tomonidan bajarilgachgina, ko'zlangan maqsadga erishiladi.

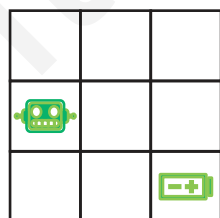
Hayotimizda har kuni va har soatda uchrab turadigan turli qoidalar ichida biror zaruriy natijaga erishishga olib keladigan amallarni ketma-ket bajarishni talab etadigan qoidalar informatikaning asosiy tushunchalaridan biri – algoritm so'zi bilan ifodalanadi. Aksariyat algoritmlar inson hayoti davomida ko'p bajarilganligi uchun ham odatga aylanib qoladi. Masalan, taom tayyorlash, ovqatlanish, tartibli kiyinish, xonadan chiqish, yozish, bir joydan ikkinchi joyga borish va hokazo.

Demak, chekli sondagi ko'rsatma yoki buyruq (amal)lar ketma-ketligi algoritm, bu algoritmlarni bajarayotgan inson esa ijrochi bo'lar ekan.

Algoritm ijrochisi roli tabiat, inson, avtomatlashtirilgan qurilma (kompyuter, texnika, robot) va boshqalar tomonidan bajariladi. Agar biror masalani hal qilish uchun texnik vositadan foydalanilsa, buyruq yoki ko'rsatmalar ketma-ketligi aniq va tushunarli bo'lishi lozim. Buyruq yoki ko'rsatmalar ketma-ketligi qanchalik aniq va tushunarli bo'lsa, natijaga shunchalik tez va aniq erishish mumkin.

Ijrochi bajara olishi mumkin bo'lgan ko'rsatma yoki buyruqlar to'plamiga *ijrochining ko'rsatmalar tizimi* (qisqacha IKT) deyiladi.

Ijrochi ko'rsatmalar tizimini tushunib olish uchun quyidagi masalani ko'rib chiqamiz.



1-misol. Robot katakchalar bo'ylab o'ngga, yuqoriga yoki pastga harakatlana oladi. U bir katakdan bir necha marta o'tishi ham mumkin. Robot turgan katagidan batareyka joylashgan katakka bora oladigan zaruriy ko'rsatmalar ketma-ketligini yozing.

Masalaning shartida ijrochi Robotning ko'rsatmalar tizimi (RKT) $RKT = \{\text{o'ngga; yuqoriga; pastga}\}$ dan iborat. Endi masala yechimi sifatida quyidagi algoritmlardan birini olish mumkin:

Tayanch tushunchalar

Algoritmning ijrochisi – *algoritm*da belgilangan buyruq yoki ko'rsatmalarni bajarishga qodir mavhum (abstrakt) yoki moddiy (texnik, biologik yoki biotexnik) tizim.

Qadamlar soni	1-algoritm	2-algoritm	3-algoritm	4-algoritm	5-algoritm
1	1) o'ngga;	1) pastga;	1) o'ngga;	1) yuqoriga;	1) yuqoriga;
2	2) pastga;	2) o'ngga;	2) o'ngga;	2) o'ngga;	2) o'ngga;
3	3) o'ngga.	3) o'ngga.	3) pastga.	3) o'ngga;	3) pastga;
4				4) pastga;	4) pastga;
5				5) pastga.	5) o'ngga.

Demak, masalaning yechimiga olib boruvchi algoritm yagona bo'lmisligi ham mumkin ekan.

Endi algoritmning asosiy xossalari bilan tanishib chiqamiz.

Diskretlilik. Algoritmni chekli sondagi oddiy ko'rsatmalar ketma-ketligi shaklida ifodalash kerak.

Aniqlilik. Algoritmida ijrochiga berilayotgan ko'rsatmalar yagona ma'noga ega, aniq mazmunli bo'lishi hamda faqat algoritmida ko'rsatilgan tartibda bajarilishi shart. Ko'rsatmadagi noaniqliklar ko'zlangan maqsadga erishishga to'sqinlik qiladi. Masalan, "ozgina o'ngga yurilsin" (ozgina deganda, qanday masofa nazarda tutilgan – 100 metrmi yoki 50?), "keragicha shakar solinsin" (kerak deganda, qancha shakar nazarda tutilgan – 1 choy qoshiqmi yoki 1 osh qoshiq?), "dastur ishga tushirilsin" (qaysi dastur?) kabi ko'rsatmalar har xil (aksariyat hollarda keraksiz) natijalarga olib keladi.

Tushunarlilik. Ijrochiga tavsiya etilayotgan ko'rsatmalar unga tushunarli bo'lishi, ijrochining imkoniyatlariga javob berishi shart. Aks holda, ijrochi oddiy amalni ham bajara olmay qolishi mumkin. Agar ijrochi inson bo'lsa, u holda algoritm inson tushunadigan til, uning bilimi, hayotiy tajribasi, kasbiy malakasi, yoshi, qolaversa, jismoniy imkoniyatlaridan kelib chiqib tuzilishi kerak. Agar ijrochi texnik vosita (masalan, kompyuter, elektron soat, dastgoh) bo'lsa, u holda algoritm shu texnik vositaning imkoniyatlaridan kelib chiqib tuzilishi lozim.

Demak, berilayotgan har qanday ko'rsatma ijrochining ko'rsatmalar tizimidan olinishi, ya'ni ijrochi uni qanday bajarishni bilishi kerak ekan.

Ommaviylik. Har bir algoritm, o'z mazmuniga ko'ra, bitta turdagi masalalarning barchasi uchun yaroqli bo'lishi lozim. Masalan, ikkita oddiy kasrning umumiy maxrajini topish algoritmi har qanday kasrlar umumiy maxrajini topish uchun o'rinli bo'lsa, berilgan ikkita natural sonning eng katta umumiy bo'luvchisi (EKUB)ni topishning *Yevklid* algoritmi ham barcha natural sonlar uchun o'rinlidir.

2-misol. n va m natural sonlarning eng katta umumiy bo'luvchisi topilsin.

- 1) boshlansin;
- 2) agar $n = m$ bo'lsa, n natija deb olinsin va 6-bandga o'tilsin;
- 3) n va m sonlarning kattasi aniqlansin;
- 4) n va m sonlarning kattasi o'zi bilan kichik sonning ayirmasiga teng deb olinsin;
- 5) 1-bandga o'tilsin;
- 6) tugallansin.

Natijaviylik. Har bir algoritm chekli sondagi qadamlardan keyin albatta natija berishi lozim. Chekli qadamlar bajarilganidan keyin qo'yilgan masala yechimga ega emasligini aniqlash ham natija hisoblanadi. Algoritm har doim ham avvaldan ko'zlangan maqsadga

erishishga olib kelmasligi mumkin. Bunga ba'zan algoritmning noto'g'ri tuzilganligi yoki boshqa xatoliklar sabab bo'ladi. Lekin salbiy *natija* ham natija deb qabul qilinadi.

3-misol. Tomonlarining uzunliklari a, b, c bo'lgan uchburchakning yuzi hisoblansin.

Quyida keltirilgan "Uchburchakning mavjud bo'lish shartini tekshirish" algoritmidan foydalanib, avval uchburchakning mavjud yoki mavjud emasligini aniqlaymiz. Agar uchburchak mavjud bo'lsa, uning yuzini hisoblaymiz, aks holda hisoblamaymiz. Bu ham natija sanaladi.

1) a, b, c qiymatlar aniqlansin;

2) agar $a \leq 0$, yoki $b \leq 0$, yoki $c \leq 0$ bo'lsa, uchburchak tomonlarining uzunligi manfiy son bo'la olmaydi, deb olinsin va 5-bosqichga o'tilsin;

3) agar $((a+b \leq c)$ va $(a+c \leq b)$ va $(b+c \leq a)$) bo'lsa, uchburchak mavjud emas, deb olinsin va 5-bosqichga o'tilsin;

4) yarim perimetr $P=(a+b+c)/2$ hisoblansin;

5) yuza: $S=\sqrt{(P*(P-a))*(P-b)*(P-c)}$ hisoblansin;

6) tugallansin.

Yuqoridagi barcha xossalar bajarilgandagina, ko'rsatma yoki buyruqlar ketma-ketligi algoritm bo'la oladi hamda u qandaydir (ijobiy yoki salbiy) natijaga ega bo'ladi.



1. Qanday ko'rsatmalarni ijrochi bajara olmaydi?
2. Algoritmning asosiy xossalarini aytib bering.
3. Tushunarlilik xossasiga oid misol keltiring.
4. Ko'rsatmalar ijrochiga tushunarli bo'lishi uchun qanday tizimdan olinishi kerak?
5. Algoritmning diskretlilik xossasi mohiyatini tushuntirib bering.
6. Algoritmning natijaviylik xossasiga oid misollar keltiring.
7. Natijaviylik xossasi bajarilmaydigan ko'rsatmalar ketma-ketligiga misol keltiring.
8. Algoritmning ommaviylik xossasini misollar asosida tushuntiring.



1. x va y sonlarining o'rta arifmetigini hisoblash algoritmini yozing.
2. Kalkulyator dasturi yordamida ixtiyoriy 3 ta sonning o'rta geometrigini hisoblash algoritmiga oid kichik loyiha ishini tayyorlang (taqdimot ko'rinishida).

15–16-DARSLAR. ALGORITM TURLARI VA TASVIRLASH USULLARI

Avvalgi darsda ko'rilgan misollarda biz masalani yechish algoritmini so'zlar va matematik formulalar orqali ifodalagan edik. Lekin algoritm boshqa ko'rinishlarda ham berilishi mumkin. Quyida biz algoritmlarni tasvirlashning eng keng tarqalgan usullari bilan tanishib chiqamiz.



1. Algoritmni qanday ko'rinishlarda ifodalash mumkin?
2. Formulalardan iborat algoritm bo'lishi mumkinmi?
3. Grafiklardan iborat algoritm bo'lishi mumkinmi?

1. Algoritmning soʻzlar yordamida ifodalanishi. Bu usulda ijrochi uchun beriladigan har bir koʻrsatma tabiiy tildagi jumlar, soʻzlar orqali buyruq shaklida beriladi. Bunda algoritmning har bir buyrugʻi ijrochiga tushunarli soʻzlar orqali ifodalanadi.

Algoritmning soʻzlar yordamida ifodalanishiga oid misol keltirib oʻtamiz.

1-masala. Tomonlariga koʻra toʻgʻri toʻrtburchakning perimetri, diagonal va yuzasini hisoblash:

- 1) boshlansin;
- 2) tomonlar qiymati kiritilsin (a, b);
- 3) perimetr qiymati hisoblansin (P);
- 4) diagonal qiymati hisoblansin (D);
- 5) yuzasi hisoblansin (S);
- 6) perimetr, diagonal va yuzasi qiymatlari chop etilsin;
- 7) tugatilsin.

2. Algoritmning formulalar yordamida ifodalanishi. Bu usulda algoritmning har bir amali matematik formulalar yordamida ifodalanadi. Algoritm amallarini ifodalashda oddiy matematik yozuvlardan foydalanish mumkin. Bu usuldan matematika, fizika, kimyo kabi aniq fanlardagi formulalarni oʻrganishda foydalaniladi. Bu usulga baʼzan *analitik ifodalash* ham deyiladi.

Endi 1-masalaning formulalar yordamida ifodalanishini koʻrib chiqamiz:

- 1) boshlansin;
- 2) toʻrtburchak tomonlari a va b ning qiymatlari aniqlansin;
- 3) $P=2*a+2*b$;
- 4) $D=\sqrt{a^2+b^2}$;
- 5) $S=a*b$;
- 6) P, D va S qiymatlar chop etilsin;
- 7) tugatilsin.

3. Algoritmning jadval yordamida ifodalanishi. Algoritmning bu tarzda tasvirlanishidan ham koʻp foydalanamiz. Masalan, maktabda qoʻllanib kelinayotgan dars jadvali, Pifagor jadvali, kimyoviy elementlar jadvali va h. k. Funksiyalarning grafiklarini chizishda ham algoritmlar qiymatlarining jadvali koʻrinishlaridan foydalanamiz. Bunday jadvallardan foydalanish algoritmlari sodda boʻlganligi tufayli, ularni oʻzlashtirib olish oson.

Biror funksiyaning grafigini chizish uchun ham funksiyaning argument qiymatlariga mos qiymatlar jadvalini hosil qilamiz. Bu ham algoritmning jadval koʻrinishiga misol boʻla oladi. Masalan, $y=x^2+2$ algoritmi asosida harakat qilayotgan ijrochi oʻtadigan nuqtalarning baʼzilari koʻrsatilgan quyidagi jadval bilan matematikadan tanishsiz:

x	-4	-3	-2	-1	0	1	2	3	4
y	18	11	6	3	2	3	6	11	18









4. Algoritmning grafik shaklda ifodalanishi. Algoritmning bu koʻrinishdagi ifodasi sizga avvaldan tanish, chunki matematika kursida chizilgan grafiklarning koʻpchiligi algoritmning grafik usulda berilishiga misol boʻladi. Bundan tashqari, shahar yoki turar joy mavzelerida joylashgan uylar yoki inshootlarning joylashish sxemasi, biror uy hamda inshootlarni izlash

va harakatlanish bo'yicha berilgan karta-sxemalari, avtobuslarning yo'nalish sxemasi ham bunga misol bo'la oladi.

Algoritm lash asoslarini o'rganishning yana bir qulay grafik shakli blok-sxema usulidir.

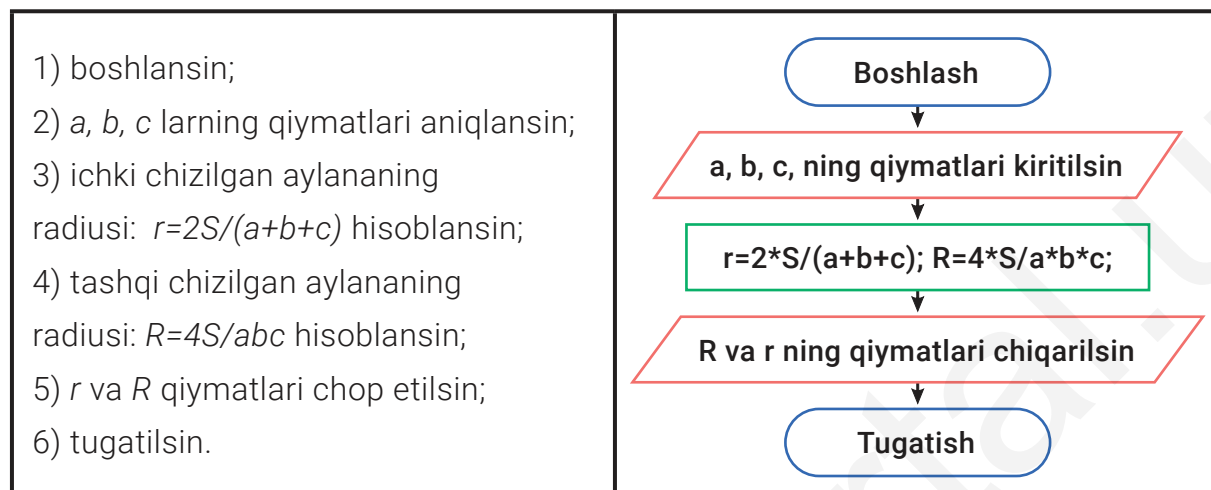
Blok-sxemalar yo'nalish chiziqlari orqali tutashtirilgan ma'lum buyruq yoki ko'rsatmani aks ettiruvchi maxsus geometrik shakl – bloklardan tashkil topadi.

Blok-sxemalarni tuzishda foydalaniladigan asosiy sodda geometrik figuralar quyidagilardan iborat:

Blokning nomi	Blokning ko'rinishi	Blokning vazifasi
Algoritmni boshlash / tugatish bloki		Algoritm blok-sxemasining boshlanishi va tugashida qo'llaniladi.
Kiritish/ chiqarish bloki		Dastlabki ma'lumotlar kiritishni tashkil qilish va olingan ma'lumotlarni chiqarish uchun xizmat qiladi.
Funksional blok (operator bloki)		Qiymat berish yoki tegishli ko'rsatmalarni bajarishga xizmat qiladi. To'rtburchak ichiga bajarilishi kerak bo'lgan buyruq yoziladi. Bir blokda bir nechta buyruqni yozish mumkin.
Muqobil blok		Shart tekshirish orqali algoritmning bajarilish yo'nalishi belgilanadi. Agar romb ichida yozilgan shart o'rinli bo'lsa, u holda boshqaruv "ha" tarmog'i, aks holda, "yo'q" tarmog'i bo'ylab uzatiladi.
Sikl bloki		Parametrli takrorlanish jarayonni tashkil qilishda qo'llaniladi. Siklning takrorlanishlar soni va qadami ma'lum bo'lishi kerak. Blokning ichida sikl parametrining boshlang'ich, yakuniy qiymati va uni o'zgartirish qadami ko'rsatiladi.
Qism dastur bloki		Oldindan yaratilgan yordamchi algoritmgamurojaat qilish uchun ishlatiladi.
Xabarlarni chop etish bloki		Natijalarni bosib chiqarish uchun ishlatiladi.
Yo'nalish bloki		Blok-sxemadagi harakat yo'nalishini ko'rsatadi.
O'zlashtirish bloki	=	Qiymat berish ko'rsatmasi hisoblanadi.

2-masala. Uchburchak tomonlarining uzunligi bilan berilgan. Uchburchakga ichki va tashqi chizilgan aylanalar radiuslari va uzunliklari hisoblansin.

Ichki chizilgan aylana radiusi $r=2S/(a+b+c)$, tashqi chizilgan aylananing radiusi esa $R=4S/abc$ formulalar orqali hisoblanadi. Bu yerda S – uchburchakning yuzi, a, b, c – uchburchak tomonlarining uzunliklari.



Biz blok-sxemalar algoritmlarni ifodalashning qulay vositalaridan biri hamda ularning ko'rgazmalilik imkoniyati ancha katta ekanligini inobatga olgan holda, ulardan dasturlashni o'rganish davomida foydalanib boramiz. Shu sababli blok-sxemalar bilan ishlashni hozirdan boshlab puxta o'zlashtirib borishimiz zarur.

DIQQAT

5. Algoritmning dastur shaklida ifodalanishi.

Hozirgi kunda juda ko'p algoritmik tillar mavjud bo'lib, ularni *dasturlash* tillari deb ataymiz. *Algoritmik til* – algoritmlarni bir xil va aniq yozish uchun ishlatiladigan belgilashlar va qoidalar tizimi. Algoritmik til oddiy tilga yaqin bo'lib, u matematik belgilarni (yuqorida aytilganidek) o'z ichiga oladi. Qo'yilgan masalalarni yechish uchun tuzilgan algoritmlarni to'g'ridan-to'g'ri mashinaga kiritib, yechib bo'lmaydi, shu sababli yozilgan algoritmni birorta algoritmik tilga o'tkazish zarur.

Har qanday algoritmik til o'z qo'llanilish sohasiga ega. Odatda, algoritmning kompyuter tushunadigan tilda yozilishi **dastur** deb ataladi. Kompyuter tushunadigan tilga **dasturlash tili** deyiladi. Jahonda minglab dasturlash tillari mavjud va ularning soni yanada ortib bormoqda. Hozirgi kunda **Pascal, Delphi, C, C++, Java, Phyton** dasturlash tillari keng tarqalgan va o'rganish uchun qulay.

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



1. Algoritmni tasvirlash usullari haqida ma'lumot bering.
2. Algoritmni formulalar orqali ifoda etilishiga fizika fanidan misollar keltiring.
3. Algoritmni grafik shaklda berilishiga misollar keltiring.
4. Algoritmni so'zlar orqali ifoda etilishiga hayotiy misollar keltiring.
5. Hayotda uchraydigan jadval bilan ifodalangan algoritmlarga misollar keltiring.
6. Matematika fanida algoritmlarning jadval shaklidan foydalanilishiga misollar keltiring.
7. Qaysi fanlarda algoritmni formulalar yordamida berish qulay?
8. Blok-sxema nima? Blok-sxema asosiy elementlarining vazifalari bilan tanishtiring.



1. AB kesmani teng ikkiga bo'lish algoritmini so'zlar yordamida ifodalang.
2. Radiusi R ga teng doiraning yuzasini hisoblash algoritmini grafik va blok-sxema yordamida tuzing.
3. MS Excel dasturida $y = x^2 + 3$ funksiya grafigini hosil qilish algoritmini so'zlar yordamida yozing.
4. MS Power Point dasturida "Mening oilam" mavzusida fotoalbom tayyorlash algoritmini qulay usulda tasvirlang.
5. Ixtiyoriy $n > 0$ sonining kvadrat ildizini aniqlash algoritmini blok-sxema yordamida tuzing.

17-DARS. AMALIY MASHG'ULOT



1. Ijrochi sifatida quyidagi ko'rsatmalardan qaysilarini bajara olmaysiz va nima uchun?
 - a) 45 ta quyon 10 ta katakka bir xil bo'lmagan sonda joylashtirilsin;
 - b) Tomonlari 12 cm bo'lgan kvadrat yuzi hisoblansin;
 - d) 5000 m balandlikda uchib ketayotgan samolyotda turib yerdagi binolar soni aniqlansin;
 - e) 4 ta 6 raqami va +, -, * amallari yordamida 66 soni hosil qilinsin.
2. Berilgan ko'rsatmalar tizimiga mos ijrochilarni aniqlang:
 - a) {axborotlarni: to'plash; saqlash; qayta ishlash; uzatish; chop etish};
 - b) {kutubxonaga bor; kitob ol; kitobni oxirigacha o'qi; kitobni kutubxonaga topshir};
 - d) {kompyuterni yoq; Word dasturini ishga tushir; "Mening oilam!" faylini yarat; hujjatni saqla; hujjatni qog'ozga chiqar; dastur va kompyuterni o'chir}.
3. Quyidagi masalalar algoritmini so'zlar yordamida tuzing:
 - a) ijrochining ko'rsatmalar tizimi faqat {5 ni qo'sh; 3 ga ko'paytir; 6 ni ayir; 2 ga bo'l} ko'rsatmalaridan iborat. Bu ijrochi 12 sonidan 20 sonini hosil qilishi uchun algoritm tuzing;
 - b) berilgan a, b, x sonlardan $y = ax + 2b$ funksiyaning qiymatini hisoblash algoritmini tuzing;
 - d) o'z telefoningizdan biror yaqiningizga sms xabar yuborish algoritmini tuzing;
4. Quyidagi masalalar algoritmlarini blok-sxema yordamida tuzing:
 - a) tomonlari a va b bo'lgan to'g'ri burchakli uchburchak yuzasi va perimetrini topish algoritmini tuzing;
 - b) uzunligi L bo'lgan aylana bilan chegaralangan doira yuzini topish algoritmini tuzing;
 - d) agar xarajat 200 000 so'mdan oshsa, 12 % li chegirma bilan qilingan 367 450 so'mlik xaridda qancha pul tejab qolganligini hisoblash algoritmini tuzing.

18-DARS. NAZORAT ISHI

1. A="Klaviatura – axborotni kiritish qurilmasi", B="1010₂ =10₁₀", C="128 bayt=16 bit" mulohazalar qiymati asosida quyidagi mantiqiy ifodalar qiymatini aniqlang:

a) $A \vee \neg C$; b) $A \& \neg (B \& \neg C)$; c) $A \Leftrightarrow (\neg B \Rightarrow \neg C)$.

2. a) X=1; b) X=12; c) X=3 holatlar uchun $((X>3) \vee (X<3)) \Rightarrow (X<4)$ mantiqiy ifoda qiymatini aniqlang.

A	B	C	$C \vee A$	$(C \vee A) \Rightarrow B$
0	0		0	1
0		0	0	1
	0	0	1	0
1	1	1	1	

3. Ushbu rostlik jadvalidagi bo'sh kataklarni to'ldiring:

4. Quyidagi mulohazani mantiqiy ifoda ko'rinishida yozing va rostlik jadvalini tuzing: "Men buvimning uyiga boraman, agar u yerda do'stlarimni uchratsam, u holda vaqtni juda qiziqarli o'tkazaman".

5. $(A \vee B) \& (\neg A \& \neg B) \vee \neg C$ mantiqiy ifodaning rostlik jadvalini tuzing.

6. $X \vee Y \& \neg Z$ ifoda uchun X, Y, Z o'zgaruvchilarning qanday qiymatlarida ifoda rost qiymat qabul qiladi?

7. Misollarda amallarning bajarilish tartibini aniqlang va barcha mulohazalar rost bo'lganda hisoblang:

a) $A \& B \vee (\neg C)$; b) $A \& B \vee C \& E$; d) $\neg (A \& B) \vee (A \& B)$; e) $(A \& B) \vee (\neg B) \vee (\neg D)$.

8. "Yak-40" rusumli samolyot Toshkent – Buxoro shaharlari orasidagi masofani bir soat-u 10 minutda uchib o'tdi. Samolyotning tezligi 600 km/soat bo'lsa, masshtabi 1:2 000 000 bo'lgan kartada bu masofa necha cm ga teng? Mazkur shaharlar orasidagi masofani aniqlang.

19-DARS. CHIZIQLI ALGORITMLAR



XX asrning 70-yillarida golland olimi Edsger Deykstra (1930–2002) har qanday algoritm uning nima maqsadda tuzilganligi va murakkabligidan qat'i nazar, uchta: ketma-ketlik, tarmoqlanish va takrorlanish algoritmik konstruksiyadan foydalanilgan holda yozilishi mumkinligi haqidagi g'oyani ilgari surdi va to'liq asoslab berdi.

Algoritm informatikaning asosiy tushunchalaridan biri hisoblanishi hamda algoritm natijaga erishish uchun zarur harakatlar ketma-ketligi ekanligini yaxshi bilamiz.

Har qanday algoritm mantiqiy tuzilishi, ya'ni bajarilish tartibiga ko'ra uchta asosiy turga bo'linadi: chizikli, tarmoqlanuvchi va takrorlanuvchi.

Chizikli algoritm deb, barcha ko'rsatmalari hech qanday shartsiz, faqat ketma-ket bajariladigan jarayonlarga aytiladi.

Bunga qo'shish yoki ko'paytirish natijalarini hisoblash, bir nechta o'zgaruvchilar qiymatlarini almashtirish, choy damlash, doira yuzini hisoblash kabilarni misol sifatida keltirish mumkin.

Chizikli algoritmning so'zlar yordamida berilishi, tavsiflanishiga oid misolni ko'rib chiqaylik.

Choy damlash maqsad qilib qo'yilgan bo'lsin. U holda choy damlayotgan shaxs biz uchun

odatiy hol bo'lib qolgan quyidagi ishlarni ketma-ketlik tartibida bajarishi lozim:

- 1) choynak qopqog'i ochilsin;
- 2) choynak qaynoq suv bilan chayilsin;
- 3) choynakka bir choy qoshiq miqdorida quruq choy solinsin;
- 4) choynak to'lguncha qaynagan suv quyilsin;
- 5) choynakning qopqog'i yopilsin;
- 6) choynak sochiq bilan yopilib, besh daqiqaga qoldirilsin.

Chiziqli tuzilishga ega bo'lgan algoritmning blok-sxemasi, asosan, *algoritmni boshlash/tugatish bloki, kiritish/chiqarish bloki hamda funksional (operator) bloki* yordamida tuziladi. Chiziqli tuzilish bir chiziq bo'ylab joylashgan, ketma-ket bajariladigan ko'rsatma (buyruq)lar to'plami ko'rinishida bo'ladi va ular algoritmda qanday tartibda yozilgan bo'lsa, aynan shu tartibda bajariladi (1-rasm). Chiziqli algoritmni ifodalash uchun quyidagi tuzilmadan foydalaniladi:

So'zlar orqali	Blok-sxema ko'rinishida	Namuna
1-ko'rsatma 2-ko'rsatma ... n-ko'rsatma		Tomonlari a va b bo'lgan to'g'ri to'rtburchak perimetri va yuzasini toping.

1-rasm. Chiziqli algoritm

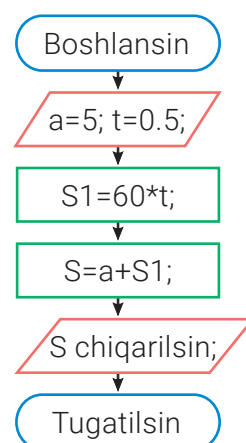
Endi chiziqli algoritmlarga oid misollarni ko'rib chiqamiz.

1-misol. Sayyoh qishloqdan chiqib, shahar tomon jo'nadi. U a kilometr piyoda yurganidan keyin avtobusga o'tirdi va avtobusda t soatda shaharga yetib keldi. Agar avtobus 60 km/soat tezlik bilan harakat qilgan bo'lsa, $a = 5$ va $t = 0,5$ bo'lganda, qishloq bilan shahar orasidagi S masofani hisoblash algoritmini tuzing.

Yechish: Masofani hisoblash formulasini esga olamiz: $S = v \cdot t$. Sayyoh avtobusda t soatda $S_1 = 60t$ kilometr yo'l bosgan. Shuning uchun qishloq bilan shahar orasidagi masofa $S = a + 60t$ formulasi bilan ifodalanadi. $a = 5$ va $t = 0,5$ bo'lganda, $S = 5 + 60 \cdot 0,5 = 35$ km bo'ladi.

Endi S masofani hisoblash algoritmini so'zlar va blok-sxema orqali ifodalaymiz:

- 1) boshlansin;
- 2) a , t larning qiymatlari kiritilsin;
- 3) sayyoh avtobusda t soatda bosib o'tilgan yo'l: $S_1 = 60 \cdot t$; hisoblansin;
- 4) qishloq bilan shahar orasidagi masofa: $S = a + S_1$; hisoblansin;
- 5) S ning qiymati yozilsin;
- 6) tugatilsin.



1. Algoritmning mantiqiy tuzilishiga ko'ra qanday turlari mavjud?
2. Qanday algoritmlarga chiziqli algoritmlar deyiladi?
3. Chiziqli algoritmlarga oid hayotiy misollar keltiring.
4. Chiziqli algoritmlar blok-sxemasi qanday tuzilishga ega?
5. Uydan maktabga kelish algoritmini aytib bering.



1. Algoritm uchun quyidagi jadvalda keltirilgan harakat (buyruq)lar to'g'ri tartibini aniqlang va uni jadvalning 2-ustuniga yozing. Algoritmga nom bering.

1) poyabzalni joyiga qo'yish;	
2) zinadan chiqish;	
3) cho'tka bilan poyabzal yaltiragunga qadar ishqalash;	
4) poyabzal changini mato bilan tozalash;	
5) barchasini xonaga olib kelish;	
6) poyabzal cho'tkasi va kremini joyiga qo'yish;	
7) poyabzal cho'tkasi va kremini olish;	
8) kremni poyabzalga surtish;	
9) poyabzalni olish.	

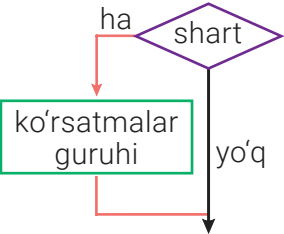
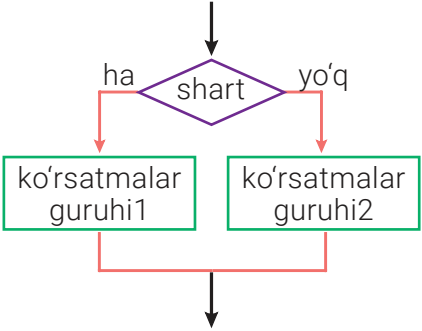
2. Agar uchburchakning ikki tomoni va ular orasidagi burchak ma'lum bo'lsa, uchburchak yuzini topish algoritmini so'zlar va blok-sxema orqali ifodalang.

20-DARS. TARMOQLANUVCHI ALGORITMLAR

Shunday hisoblash jarayonlari ham bo'ladiki, unda qo'yilgan ayrim mantiqiy shartlarning bajarilishiga ko'ra jarayonlar bir necha tarmoqqa bo'linadi hamda ulardan hech bo'lmaganda bittasi bajariladi. Bunday jarayonlar bajarilishi uchun tarmoqlanuvchi algoritmlar tuziladi.

Agar hisoblash jarayoni qandaydir 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 struktura, odatda, qandaydir mantiqiy shartni tekshirish blokini o'z ichiga oladi. Tekshirish natijasiga ko'ra, tarmoq deb ataluvchi u yoki bu amallar ketma-ketligi bajariladi. Tarmoqlanuvchi tuzilish shart tekshirish natijasiga (ha yoki yo'q) qarab ikki yo'ldan birini tanlash imkoniyatini beradi, ya'ni ko'rsatilgan tarmoqdan faqat bittasining bajarilishini ta'minlaydi.

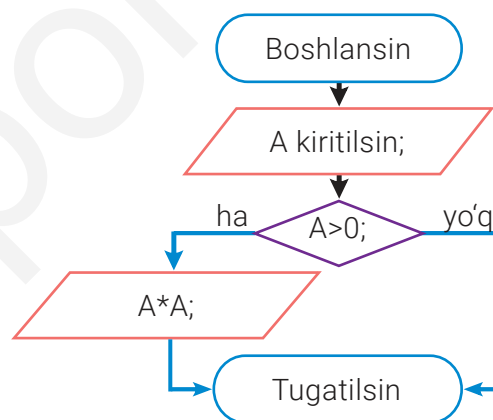
Bu tuzilmalar, asosan, 2 xil – to'liq va qisqartirilgan ko'rinishda berilishi mumkin. Ular quyidagi sxema orqali ifodalanadi:

Turi	So'zlar orqali	Blok-sxema ko'rinishida	Namuna
agar – u holda;	agar shart u holda ko'rsatmalar guruhi oxiri		Agar berilgan sonning raqamlari yig'indisi 3 ga qoldiqsiz bo'linsa, u holda bu son 3 ga karrali bo'ladi.
agar – u holda – aks holda;	agar shart u holda ko'rsatmalar guruhi1 aks holda ko'rsatmalar guruhi2 oxiri		Agar berilgan sonning raqamlari yig'indisi 3 ga qoldiqsiz bo'linsa, u holda bu son 3 ga karrali bo'ladi, aks holda 3 ga karrali bo'lmaydi.

1-misol. Berilgan A son 0 (nol)dan katta musbat son bo'lsa, u holda uning kvadratini hisoblash algoritmini tuzing:

- 1) boshlansin;
- 2) **A** kiritilsin;
- 3) agar **A > 0** bo'lsa, u holda 4-bandga o'tilsin;
- 4) natija **A*A** deb olinsin;
- 5) tugatilsin.

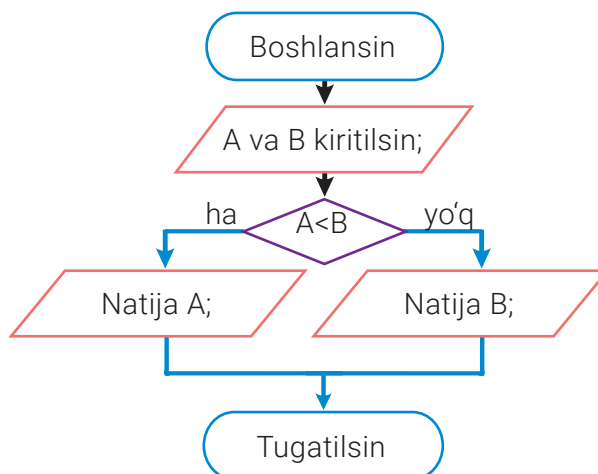
Bu misolda agar $A > 0$ bo'lsa, 4-banddagi ko'rsatma bajariladi, aks holda, ya'ni $A \leq 0$ shart bajarilsa, 3-banddagi ko'rsatma bajarilmaydi.



2-misol. Berilgan ikkita A va B sonlardan kichigini topish algoritmini tuzing:

- 1) boshlansin;
- 2) **A** va **B** kiritilsin;
- 3) agar **A < B** bo'lsa, 4-bandga o'tilsin; aks holda, 5-bandga o'tilsin;
- 4) natija **A** deb olinsin va 6-bandga o'tilsin;
- 5) natija **B** deb olinsin;
- 6) tugatilsin.

Bu misoldan quyidagicha xulosa chiqarish mumkin: agar $A < B$ shart bajarilsa, 5-banddagi ko'rsatma bajarilmaydi, aks holda, ya'ni $A > B$ bo'lsa, 4-banddagi ko'rsatma bajarilmaydi.





1. Qanday algoritmlarga tarmoqlanuvchi algoritmlar deyiladi?
2. Tarmoqlanuvchi algoritmlarga oid hayotiy misollar keltiring.
3. Tarmoqlanuvchi algoritmlarning qanday tuzilishlari mavjud?
4. Tarmoqlanuvchi algoritmlarning to'liq va qisqartirilgan ko'rinishlari farqlarini tushuntiring?
5. Tarmoqlanuvchi algoritmlarning to'liq ko'rinishiga misollar keltiring.



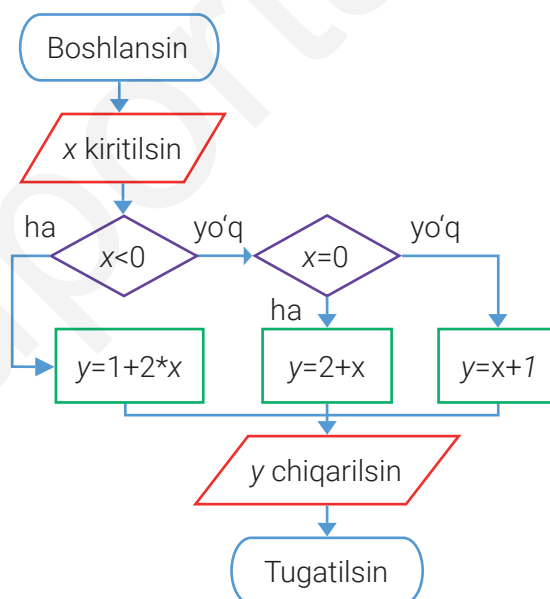
1. Berilgan a son 5 dan katta bo'lsa, uning kvadrat ildizini hisoblash algoritmini tuzing.
2. Berilgan butun sonning absolyut qiymatini aniqlash algoritmini tuzing.
3. $ax^2 + bx + c = 0$ ko'rinishidagi kvadrat tenglama ildizini topish algoritmini tuzing.

21-DARS. AMALIY MASHG'ULOT

Misol. Algoritm formula yordamida berilgan.

$$y = \begin{cases} 1+2x, & \text{agar } x < 0 \\ 2+x, & \text{agar } x = 0 \\ x+1, & \text{agar } x > 0 \end{cases}$$

Funksiyaning qiymatini hisoblashga oid tarmoqlanuvchi algoritmi blok-sxema yordamida tasvirlang:



1. Chiziqli tuzilmaga oid masalalar:

a) gilamni tozalash algoritmini so'zlar yordamida ifodalang;

b) radiusi R ga teng aylananing uzunligi, doira yuzi va shar hajmini hisoblash algoritmini tuzing (yo'llanma: $L=2\pi R$; $S=\pi R^2$; $V=\frac{4}{3}\pi R^3$);

d) o'lchamlari $200 \text{ mm} \times 120 \text{ mm} \times 700 \text{ mm}$ bo'lgan devorni hosil qilish uchun $250 \text{ mm} \times 120 \text{ mm} \times 65 \text{ mm}$ o'lchamdagi g'ishtdan qancha kerakligini aniqlash algoritmini tuzing.

2. Tarmoqlanuvchi tuzilmaga oid masalalar:

a) a , b va c sonlar berilgan. Agar ularning yig'indisi manfiy bo'lsa, ular har birining kvadrat ildizini, aks holda, ularning har biridan 30 ni ayirib chiqaruvchi algoritm tuzing.

b) $a=10$ va $b=1,5$ qiymatlar uchun quyidagi funksiya qiymatini hisoblash algoritmini tuzing.

$$y = \begin{cases} x^2 + |x - 1| + 2, & \text{agar } x > 1,5; \\ \sqrt{3a - 2bx + x^2}, & \text{agar } x = 1,5; \\ \frac{2ax}{5} + b, & \text{agar } x < 1,5. \end{cases}$$



22-DARS. TAKRORLANUVCHI ALGORITMLAR



Rasmda nimalarni ko'rayapsiz?

Agar tabiatda va atrofingizda sodir bo'layotgan voqealarga diqqat bilan qarasangiz, bir qancha siklik (takrorlanuvchi) jarayonlarni ko'rishingiz mumkin. Masalan, yil fasllarining, kun va tunning almashishi, darslarning har hafta takrorlanishi, har kuni tushlik qilish yoki har doim uyqudan uyg'ongach, yuvinish.



Takrorlanuvchi algoritm deb, biron bir shart tekshirilishi yoki qandaydir parametrning har xil qiymatlari asosida algoritmda takrorlanish yuz beradigan jarayonlarga aytiladi.

Shunday jarayonlar ham borki, ularning ayrim bo'laklari bir necha marta takrorlanadi. Masalan, biror fandan test topshira olmagan, ya'ni "qoniqarsiz" baho olgan o'quvchi toki testdan "qoniqarli" baho olgunga qadar fanga oid mavzularni qayta-qayta o'qishiga, testga tayyorlanishiga to'g'ri kelsa, $9! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 \cdot 9$ ifodani hisoblash uchun esa 8 marta ko'paytirish amalini bajarishga to'g'ri keladi.

Bunday jarayonlar uchun algoritmlar tuzishda takrorlanuvchi algoritmlardan foydalaniladi.

Takrorlanuvchi algoritmlar " $i=i+1$ ", " $S=S+i$ " yoki " $P=P*i$ " ko'rinishidagi ko'rsatmalarning ishtiroki bilan ajralib turadi (* – ko'paytirish amali). Bunday ko'rsatmalarning mohiyatini tushunish uchun takrorlanishning bir nechta qadamini ko'rib chiqish lozim.

Odatda, yig'indi uchun boshlang'ich qiymat (inglizchadan SUMM, ya'ni yig'indi ma'noli so'zning bosh harfi) $S=0$ va ko'paytma uchun (inglizchadan PRODUCT, ya'ni ko'paytma ma'noli so'zning bosh harfi) $P=1$ deb olinadi, chunki bu qiymatlar, ya'ni 0 va 1 lar, mos ravishda, yig'indi va ko'paytmaning natijasiga ta'sir etmaydi:

Qadamlar	i	S	P
1-qadam:	$i=1$ bo'lsin, u holda	$S=S+i=0+1=1,$	$P=P*i=1*1=1;$
2-qadam:	$i=i+1=1+1=2,$	$S=S+i=1+2=3,$	$P=P*i=1*2=2;$
3-qadam:	$i=i+1=2+1=3,$	$S=S+i=3+3=6,$	$P=P*i=2*3=6;$
4-qadam:	$i=i+1=3+1=4,$	$S=S+i=6+4=10,$	$P=P*i=6*4=24.$

Hisoblash jarayonining ko'p marta takrorlanadigan qismi ichki sikl tanasi (jismi) deb yuritiladi.

Takrorlanadigan harakat (ko'rsatma)larni amalga oshirish uchun sikl yoki takrorlash buyruqlari deb nomlangan maxsus algoritmik tuzilmalar mavjud. Takrorlanish tuzilmalari bir necha ko'rsatmalar guruhining ko'p marta bajarilishini ta'minlaydi. Bu tuzilmalar 41-betdagi jadval ko'rinishlarida o'rin olgan.

Shart oldin tekshiriluvchi (toki) takrorlanuvchi algoritmlarda avval shart tekshiriladi, so'ngra, agar shart qanoatlantirsa (rost bo'lsa), sikl tanasi bajariladi, aks holda hisoblash to'xtatiladi.

Shart keyin tekshiriluvchi (gacha) takrorlanuvchi algoritmda avval sikl tanasi bajarilib, so'ngra sikldan chiqish sharti tekshiriladi, ya'ni sikl tanasi qo'yilgan shart bajarilib bo'lguncha takrorlanaveradi.

Turi	Soʻzlar orqali	Blok-sxema koʻrinishida	Izoh
Toki – shart oldin tekshiriluvchi	toki (shart) sikl boshi koʻrsatmalar guruhi; sikl oxiri		Shart oʻrinli boʻlmasa (yolgʻon boʻlsa), sikl toʻxtatiladi. Agar shart boshidan oq yolgʻon boʻlsa, u holda koʻrsatmalar guruhi hech qachon bajarilmaydi.
Gacha – shart keyin tekshiriluvchi	... koʻrsatmalar guruhi; gacha (shart)		Shart oʻrinli (rost) boʻlsa, sikl toʻxtatiladi.
Parametr – dan – gacha	Parametr A dan B gacha sikl boshi koʻrsatmalar guruhi; sikl oxiri		Koʻrsatmalar guruhining takrorlanishlar soni sikl parametrining boshlangʻich va oxirgi qiymatlariga bogʻliq. Bu sikl takrorlanishlar soni $(B-A+1)$ aniq boʻlganda ishlatiladi.

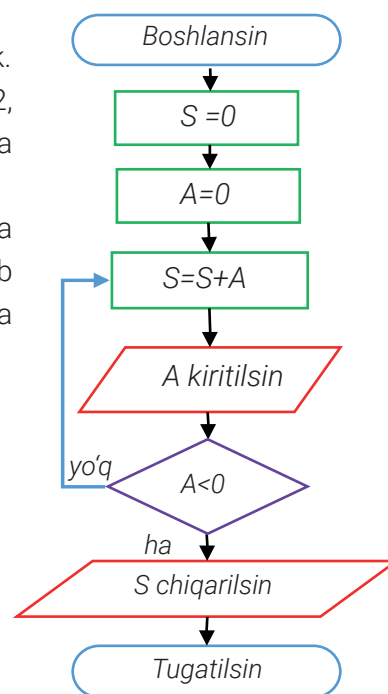
Shart oldin tekshiriluvchi va shart keyin tekshiriluvchi sikllar birgalikda *iteratsion sikllar* hisoblanadi.

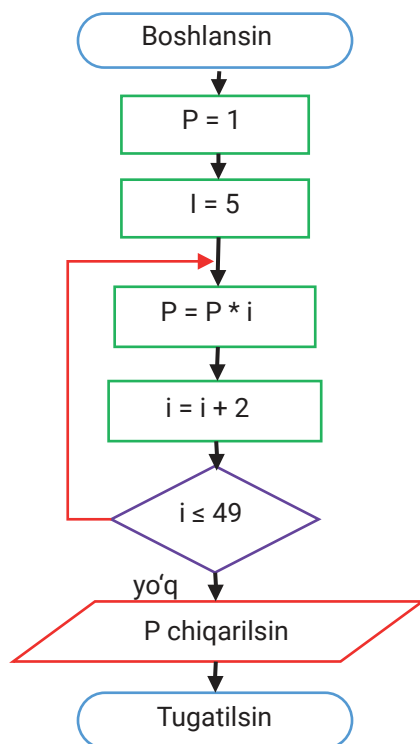
Takrorlanuvchi jarayonlarga oid misollarni koʻrib chiqaylik.

1-misol. Tasavvur qiling, klaviaturadan sonlar (1, 6, 8, 2, -6, 76, 1, -5) kiritilmoqda. Birinchi kiritilgan manfiy son (-6) gacha kiritilgan sonlar (1, 6, 8, 2) yigʻindisini hisoblash algoritmini tuzing.

Yechish. Soʻzlar bilan ifodalangan algoritmda blok-sxema bilan mutanosiblikni koʻrsatish uchun qavslar ichida izohlar berib boramiz. Yigʻindini **S** orqali, klaviaturadan kiritilayotgan sonni esa **A** orqali belgilab olamiz.

- 1) boshlansin;
- 2) **S=0** deb olinsin (yaʼni **S=0**);
- 3) **A=0** deb olinsin (yaʼni **A=0**);
- 4) **S** ga **A** ni qoʻshib, **S** deb olinsin (yaʼni **S= S+A**);
- 5) **A** kiritilsin;
- 6) agar **A<0** boʻlmasa, 4-bandga oʻtilsin;
- 7) natija **S** deb olinsin;
- 8) tugatilsin.





2-misol. 5 dan 49 gacha bo'lgan toq sonlar ko'paytmasini, ya'ni $P=5*7*9*...*49$ ni hisoblash algoritmini tuzing.

- 1) boshlansin;
- 2) $P=1$ deb olinsin;
- 3) i ning qiymati **5** deb olinsin;
- 4) P ni i ga ko'paytirib, P deb olinsin;
- 5) i ga 2 ni qo'shib, i deb olinsin;
- 6) agar $i \leq 49$ bo'lsa, 4-bandga o'tilsin;
- 7) natija P deb olinsin;
- 8) tugatilsin.

Demak, algoritmlarning chiziqli, tarmoqlanuvchi yoki takrorlanuvchi turlari mavjud ekan. Inson hayotida uchraydigan algoritmlar, asosan, shu uch turdagi algoritmlarning uzviy birligi sifatida namoyon bo'ladi.



1. Qanday algoritmlar takrorlanuvchi algoritmlar deb ataladi? Misol keltiring.
2. Takrorlanuvchi tuzilmalari haqida ma'lumot bering. Ularning bir-biridan farqini tushuntiring.
3. Sharti oldin tekshiriluvchi algoritmlarni tushuntirib bering va misol keltiring.
4. Sharti keyin tekshiriluvchi algoritmlarga hayotiy misol keltiring.
5. Uchta sondan eng kichigini aniqlab beruvchi algoritmlar tuzing.

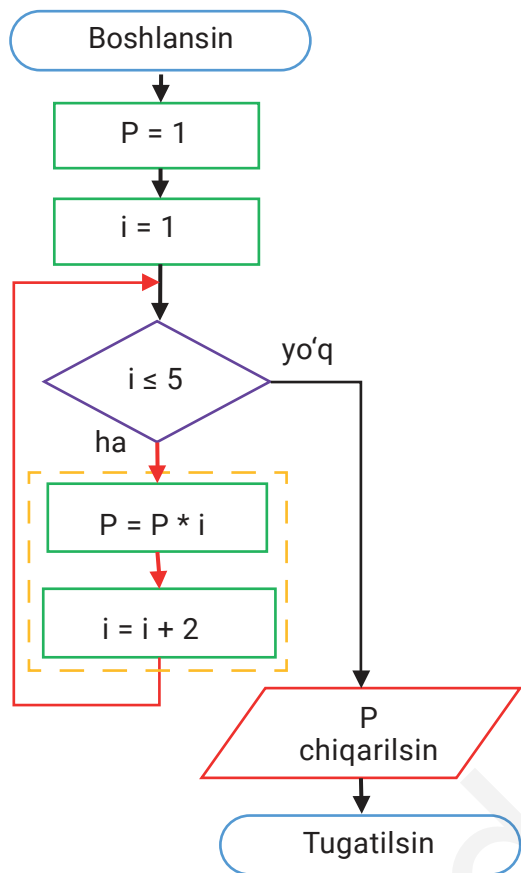


1. 3 xil takrorlanish tuzilmalaridan foydalanib, 1 dan 100 gacha bo'lgan sonlar yig'indisini hisoblash algoritmlarini tuzing.
2. Quyidagi algoritmlarga mos masala sharhini aniqlang va blok-sxema tuzing:
 - 1) boshlansin;
 - 2) $S=0$ deb olinsin;
 - 3) i ning qiymati **1** deb olinsin;
 - 4) P ni i ga ko'paytirib, P deb olinsin;
 - 5) i ga **1** ni qo'shib, i deb olinsin;
 - 6) agar $i \leq 50$ bo'lsa, 4-bandga o'tilsin;
 - 7) natija P deb olinsin;
 - 8) tugatilsin.

23-DARS. AMALIY MASHG'ULOT

Misol. 3 xil takrorlanish tuzilmalaridan foydalanib, 1 dan 5 gacha bo'lgan sonlar ko'paytmasini hisoblash algoritmini tuzing. Matematik modeli: $P = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$

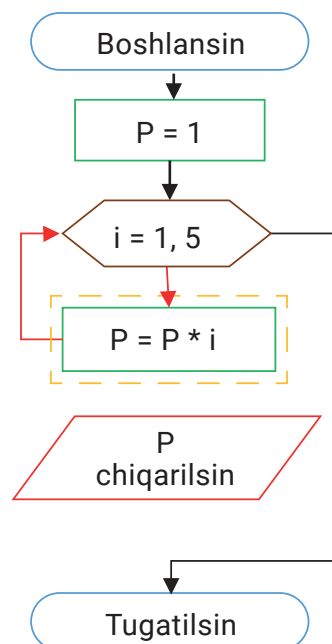
1-usul.



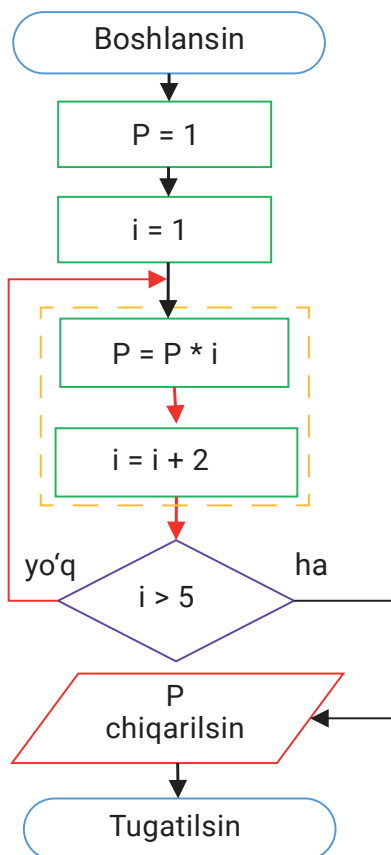
Qadam	Amallar	P	i	Shart tekshirish
1	P=1	1		
2	i=1;	1	1	
3	i<=5 P=P*i; i=i+1;	1	1	1<=5, ha (rost)
4	i<=5 P=P*i; i=i+1;	2	2	2<=5, ha (rost)
5	i<=5 P=P*i; i=i+1;	6	3	3<=5, ha (rost)
6	i<=5 P=P*i; i=i+1;	24	4	4<=5, ha (rost)
7	i<=5 P=P*i; i=i+1;	120	5	5<=5, ha (rost)
8	i<=5 P=P*i; i=i+1;			6<=5, yo'q (yolg'on)

2-usul.

Qadam	Amallar	P	i
1	P=1	1	
2	i=1; P=P*i;	1	1
3	i=2; P=P*i;	2	1
4	i=3; P=P*i;	6	3
5	i=4; P=P*i;	24	4
6	i=5; P=P*i;	120	5



3-usul.



Qa-dam	Amallar	P	i	Shart tekshirish
1	P=1	1		
2	i=1	1	1	
3	P=P*i i=i+1 i>5	1	1	2>5, yo'q (yolg'on)
4	P=P*i i=i+1 i>5	2	2	3>5, yo'q (yolg'on)
5	P=P*i i=i+1 i>5	6	3	4>5, yo'q (yolg'on)
6	P=P*i i=i+1 i>5	24	4	5>5, yo'q (yolg'on)
7	P=P*i i=i+1 i>5	120	5	6>5, ha (rost)



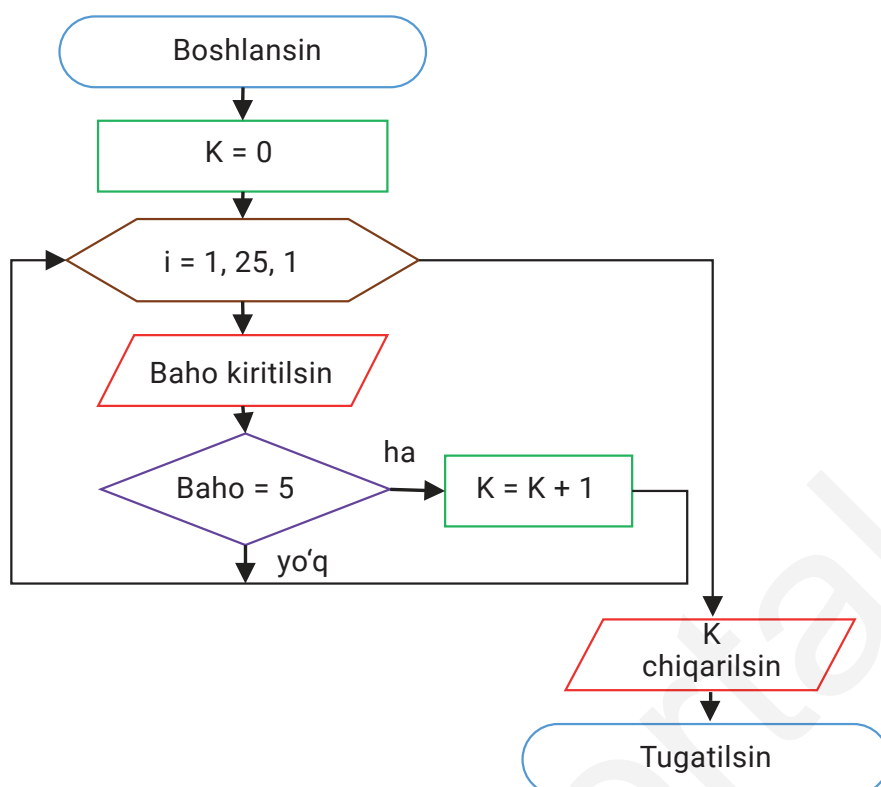
- 1 dan n gacha bo'lgan sonlar ichidan faqat 5 ga karrali sonlar yig'indisini hisoblash algoritmini tuzing.
- $y = \sin 0 + \sin 0,1 + \sin 0,2 + \dots + \sin n$ ifoda qiymatini hisoblash algoritmini tuzing.
- 3 xil takrorlanish tuzilmalaridan foydalanib $P=(2+2)*(2+3)*(2+4)*(2+5)$ ko'paytmani hisoblash algoritmini tuzing.
- $y = 2x^3 + 1$ funksiyasini x ning $[-3; 5]$ oraliqdagi butun qiymatlarida hisoblash algoritmini blok-sxema shaklida tuzing.

24-DARS. ARALASH (KOMBINATSIYALASHGAN) ALGORITMLAR

Tarkibida bir necha turdagi algoritmlar qatnashgan algoritmgaga **aralash** (*kombinatsiyalashgan*) **algoritm** deyiladi.

Aralash algoritmlarga quyidagilar misol bo'ladi.

1-masala. Sinfda 25 nafar o'quvchi bor. Qancha o'quvchi informatikadan "a'lo" baho olganligini aniqlash algoritmi tuzilsin.



Masalani yechishda tarmoqlanuvchi va takrorlanuvchi algoritmlardan foydalanildi.

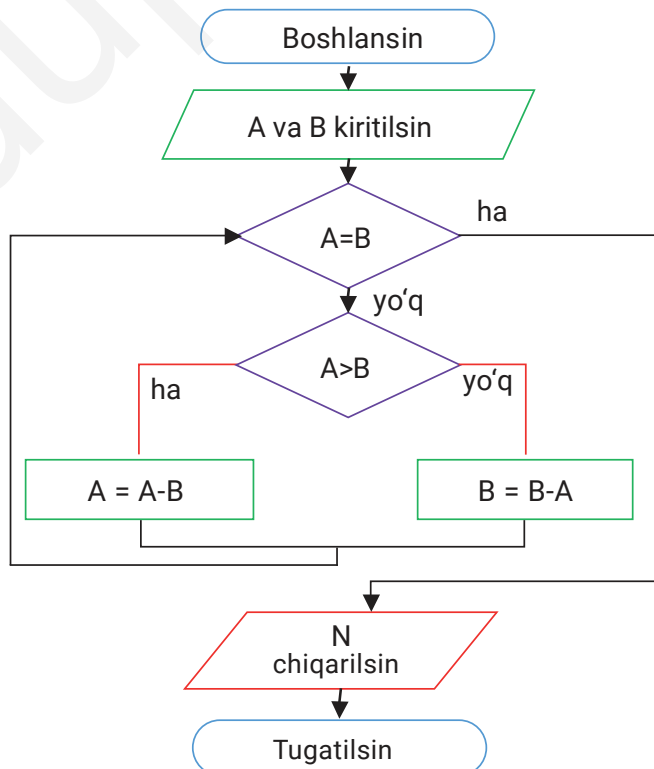
2-masala. Berilgan ikkita natural sonning eng katta umumiy bo'luvchisi (EKUB)ni topish algoritmini so'zlar va blok-sxema yordamida ifodalang.

Bizga avvaldan ma'lumki, Yevklid algoritmi aynan aralash algoritmgga misol bo'la oladi.

Quyida keltirilgan ikkita natural sonning eng katta umumiy bo'luvchisi (EKUB)ni topishning Yevklid algoritmi ham barcha natural sonlar uchun o'rinlidir.

A va B natural sonlarning eng katta umumiy bo'luvchisi topilsin:

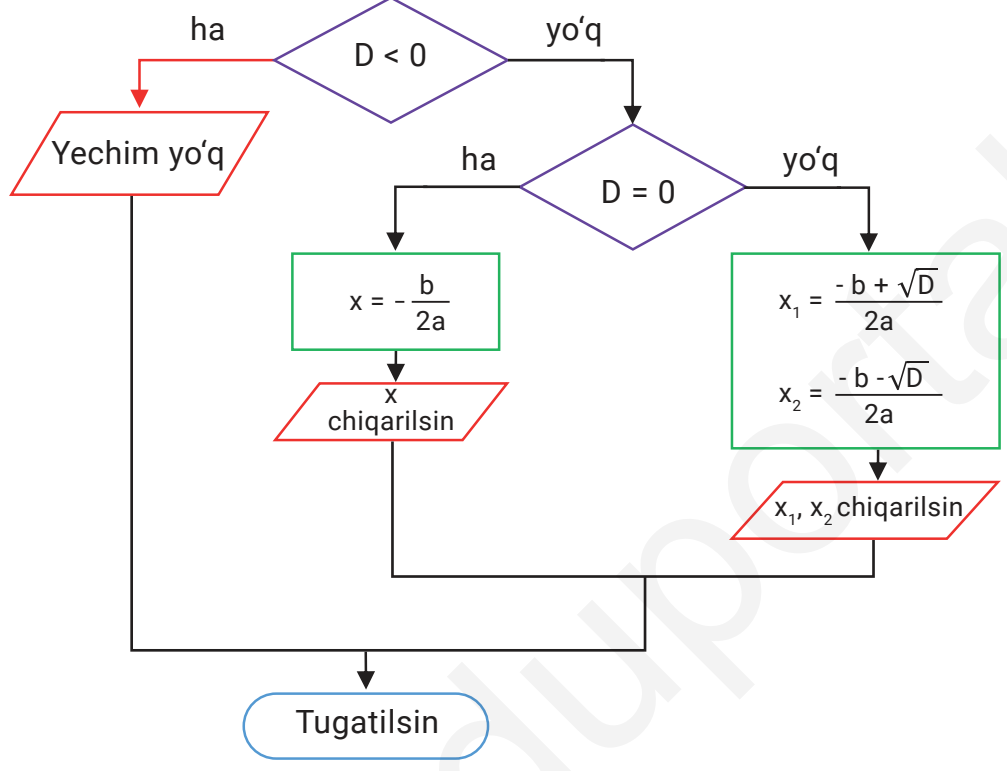
- 1) boshlansin;
- 2) agar $A = B$ bo'lsa, N natija deb olinsin va 6-bandga o'tilsin;
- 3) A va B sonlarning kattasi aniqlansin;
- 4) A va B sonlarning kattasi o'zi bilan kichik sonning ayirmasiga teng deb olinsin;
- 5) 2-bandga o'tilsin;
- 6) tugatilsin.



Boshlansin

a, b, c
kiritilsin

$D = b^2 - 4ac$



3-masala. $ax^2 + bx + c = 0$
ko'rinishidagi kvadrat tenglamaning
yechimlarini topish algoritmiga
mos blok-sxema tuzing.

SAVOL VA
TOPSHIRIQLAR



1. Qanday algoritmlar aralash algoritm deb ataladi?
2. Aralash algoritmlarga oid hayotiy misollar keltiring.
3. Aralash algoritmlardan qayerlarda foydalanish mumkin?
4. Aralash algoritmlar tarkibida qanday algoritmlar bo'lishi mumkin?

UY VAZIFASI



1. 8 ta turli son berilgan. Ular orasidagi musbat sonlar ko'paytmasini hisoblash algoritmini tuzing.
2. Sinfda 18 nafar o'quvchi bor. Qancha o'quvchi matematika fanidan "yaxshi" baho olganligini aniqlash algoritmi tuzilsin.
3. 12 ta son berilgan. Manfiy sonlar va toq sonlar yig'indisini toping.

25-DARS. DASTUR VA DASTURLASH HAQIDA

Bugungi kunda ijtimoiy hayotning qaysi jabhasiga ko'z tashlamaylik, hamma joyda kompyuterlashtirish jarayoni jadal sur'atlarda amalga oshayotganini ko'rish mumkin. Endi telefon nafaqat so'zlashish qurilmasi, u orqali matnli, audio, videoxabarlarini yuborishdan tashqari, ijtimoiy tarmoqlar orqali muloqot qilish ham mumkin. Zamon talabi o'quvchilarimiz oldiga bu qurilma va texnologiyalardan foydalanishni bilishnigina emas, shuningdek, dasturlash yordamida ularni rivojlantirish, raqamlashtirish vazifalarini ham qo'yimoqda.

1. Kompyuter dasturi nima?
2. Dasturlash deb nimaga aytiladi?
3. Kompyuter ilovalari dastur hisoblanadimi?

Ma'lumki, kompyuter – foydalanuvchi uchun turli masalalarni yechishda eng yaqin yordamchi. Xususan, insonga qulaylik yaratish uchun matn, grafik muharrirlar, taqdimot yaratuvchi dasturlar, elektron jadvallar singari ko'plab amaliy dasturlar ishlab chiqilgan. Shuningdek ta'lim, bank ishi, soliq, huquq, tibbiyot sohalari uchun maxsus yozilgan kompyuter dasturlari ham mavjud.

Xo'sh, dastur o'zi nima?

Kompyuterda biror masalani hal qilish uchun, eng avvalo, uning modeli va algoritmi tuziladi, so'ngra bu algoritm ma'lum bir qonun-qoidalar negizida kompyuter tushunadigan ko'rsatma va buyruqlarga aylantiriladi va o'ziga xos alifbo yordamida yoziladi. Yaratilgan matn kompyuter tilida **yozilgan dastur** deb ataladi.

Kompyuter dasturi – har qanday masalani yechish uchun kompyuter bajarishi lozim bo'lgan ko'rsatmalarning izchil tartibi. Kompyuter uchun yozilgan dastur chet tiliga o'xshaydi, istagan odam uni tez vaqtda o'rganib olishi mumkin.

Faraz qiling, atrofingizdagi deyarli barcha uskuna-yu qurilmalar kompyuter dasturlari orqali boshqariladi, ya'ni bu qurilmalar dasturchi tomonidan ketma-ketlikda yozilgan ko'rsatmalarga amal qiladi. Masalan, kompyuter ilovalari – hujjat yaratish, qo'shiq tinglash, video ko'rish, internetga bog'lanish kabilar – aslida dasturchilar tomonidan yozilgan va kompyuter amal qiladigan dasturlardir.

Tayanch tushunchalar

- Kompyuter dasturi** – biror masalani yechish uchun kompyuter tomonidan ijro etilishi lozim bo'lgan buyruqlarning izchil to'plami.
- Dasturlash** – kompyuter uchun dastur tuzish jarayoni.
- Dasturchi** – dastur tuzuvchi shaxs.



Mobil telefonlar.

Qurilma dasturi qo'ngiroq qilish, xabar yuborish imkonini beradi. Shuningdek, kontaktlar ro'yxatidan o'tgan shaxs ismiga ko'ra, dastur uning telefon raqamini o'zi topadi.



Kir yuvish mashinasi.

Qurilma turli rejimlarda kir yuvishga dasturlashtirilgan, shu sababli kod harorat va kir yuvish vaqtini nazoratga oladi.



Avtomobillar.

Zamonaviy kompyuter va uning dasturlarisiz avtomobil harakatini tasavvur qilish qiyin. Kompyuter dasturi tezlik, yoqilg'i, harorat kabilarni kuzatib boradi.



Dasturchi – kompyuterda ko'rish va bajarish mumkin bo'lgan jarayonlarni nazorat qiluvchi hamda dasturlarni yaratuvchi shaxs.

SAVOL VA TOPSHIRIQLAR



- 1 Dastur nima?
2. Qanday dasturlarni bilasiz?
3. Dasturlash deganda nimani tushunasiz?
4. Kir yuvish mashinasi ham dasturlash asosida ishlaydimi?
5. Dastur asosida ishlaydigan qurilmalarni sanab bering.

UY VAZIFASI



Jadvalda berilgan yangi atamalarga mos tushunchalarni toping va ularni chiziq bilan birlashtiring.

Dastur	kompyuter uchun dastur tuzish jarayoni.
Dasturchi	kompyuterda ko'rish va bajarish mumkin bo'lgan jarayonlarni nazorat qiluvchi hamda dasturlarni yaratuvchi shaxs.
Dasturlash	kompyuter tushunadigan "til".
Dasturlash tili	biror masalani yechish uchun kompyuter bajarishi lozim bo'lgan ko'rsatmalarning izchil tartibi.

26-DARS. DASTURLASH TILLARI

Inson kabi kompyuter ham o'ziga xos tilda muloqot qiladi. Bu til faqat kompyuter tushunadigan til bo'lib, u cheklangan lug'at va qat'iy yozish qoidalaridan iboratdir. Kompyuter tushunadigan va muloqot olib boradigan "til" **dasturlash tili** deb ataladi. Istalgan dasturlash tilini bilgan shaxs o'z dasturini bema'lol tuza oladi.

1. Dasturlash tillari deganda nimani tushunasiz?
2. Translyator nima?
3. Kompilyator nima?
4. Interpretator nima?

Protsessor dasturlash tilida yozilgan dasturni to'g'ridan-to'g'ri tushunmaydi. Buning uchun dasturni protsessor tiliga tarjima qiladigan (raqamli ko'rinishga o'tkazib beruvchi) tarjimon tili – **translyator**dan foydalaniladi.

Tarjimon tilining ikkita turi mavjud: kompilyator va interpretator.

Kompilyator dasturlash tilida yozilgan dastur kodlarini to'laligicha o'qib, mashina kodiga tarjima qiladi va tarjima natijalarini bajariladigan yaxlit bitta faylga yig'adi.

Interpretator dasturlash tilida yozilgan kodni bosqichma-bosqich mashina kodiga aylantirib, tahlil qiladi va berilgan buyruqlarni ketma-ketlikda bajaradi. Agar xatolik sodir bo'lsa, o'sha zahoti xabar beradi.

Dasturlash tillari juda ko'p bo'lib, ularning har biridan o'ziga xos masalalarni yechishda foydalanish mumkin. Quyida ularning eng mashhurlari haqidagi ma'lumotlar bilan tanishamiz.

C	kompyuter operatsion tizimlarini yozish uchun mo'ljallangan til.	JAVA	kompyuter, mobil telefon va planshetlar uchun dastur yozuvchi til.
JAVASCRIPT	interfaol web-saytlarni yozish uchun mo'ljallangan til.	PHP	dinamik web-saytlar yaratish uchun mo'ljallangan til.
SCRATCH	dasturlarni o'rganish uchun eng mos vizuallashtirilgan dasturlash tili.	PYTHON	turli masalalarni yechish, sun'iy intellekt tizimlari uchun mo'ljallangan dasturlash tili.

Aksariyat dasturlash tillari, xususan, C++, Pascal, Java, Python va boshqalar integrallashgan dasturlash muhiti (IDE)ga ega.

IDE (Integrated Development Environment – integrallashgan dasturlash muhiti) – dasturiy ta'minot yaratish uchun dasturiy vositalar majmui.

Tizimli va amaliy dasturlarni yaratish uchun dasturlash tillaridan foydalaniladi. Dastur yaratish jarayoni katta hajmga ega bo'lib, dasturlash tillarida dastur yozish shu jarayonning bir qismi sanaladi. Avvalgi mavzularimizda kompyuterda masalani yechish bosqichlarini ko'rib chiqqan edik. *Kompyuter dasturlarini yaratish jarayoni* ham kompyuterda masalani yechish jarayoniga o'xshash bo'lib, bir necha bosqichni o'z ichiga oladi.

Birinchi bosqichda *dastur uchun talab aniqlanadi*. Ushbu jarayonda dasturning qanday maqsadda qo'llanilishi, kirish va chiqish ma'lumotlari aniqlanadi. Dastur va resurslarni ishlab chiqish, uni joriy etishning narxi baholanadi.

Jayanch tushunchalar

Dasturlash tili – inson va kompyuter o'rtasidagi rasmiy aloqa tili. U ma'lum qoidalar asosida ma'lumotlarni qayta ishlash dasturlarini yaratish uchun qo'llaniladi.

IDE (Integrated Development Environment – integrallashgan dasturlash muhiti) – dasturchilar uchun qo'shimcha funksiyalarga ega bo'lgan maxsus matn muharriri.

Ikkinchi bosqichda *dastur loyihasi ishlab chiqiladi*. Dasturchilar uchun texnik topshiriq va vazifalar shakllantiriladi. Ishchi hujjatlar hamda ish rejasining grafigi tuziladi.

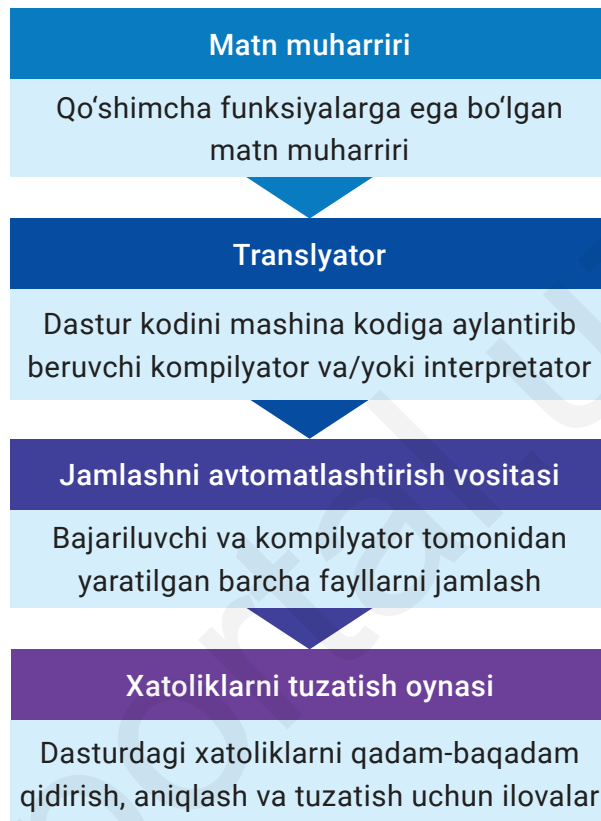
Uchinchi bosqichda *dastur kodi yoziladi*. Bu kodlash (dasturlash) jarayoni bo'lib, tuzilgan algoritm dasturlash tilida yoziladi.

To'rtinchi bosqichda kodlash jarayoni tugab, dasturdagi *xatoliklarni aniqlash va testlash boshlanadi*. Bu jarayonda dasturning to'g'ri tuzilganligi, ishlash samaradorligi va boshqa parametrlariga aniqlik kiritiladi.

Beshinchi bosqichda *dastur amaliyotga joriy etiladi*. Agar dastur aniq buyurtmachi talabiga muvofiq yaratilgan bo'lsa, u holda bu bosqich eng muhim bosqich sanaladi. Bunda avvalgi dasturda foydalanilgan qurilma va ma'lumotlar yangi dasturga moslanadi. Ushbu dastur bilan ishlovchi mutaxassislar o'qitiladi.

Oltinchi so'nggi bosqichda *qo'llab-quvvatlash amalga oshiriladi*. Bu bosqichda foydalanuvchilarga tavsiyalar beriladi. Shuningdek, foydalanuvchi xohishini inobatga olgan holda, ish jarayonida yuzaga kelgan kamchilik va takliflar asosida dasturga o'zgartirishlar kiritiladi.

IDE tarkibi



1. Dasturlash tili deganda nimani tushunasiz?
2. Qanday dasturlash tillari mavjud?
3. Kompyuterda dasturlash jarayoni necha bosqichdan iborat?
4. Interpretatorning vazifasi nimadan iborat?



1. Jadvalda kompyuter dasturlarini yaratish jarayoni berilgan. Undagi bosqichlar ustunchasiga jarayonlarga mos bosqichlarni yozib chiqing:

Jarayonlar	Bosqichlar
dastur kodi yoziladi	
dastur qo'llab-quvvatlanadi	
dastur uchun talab aniqlanadi	

xatoliklar aniqlanadi va dastur testdan o'tkaziladi	
dastur loyihasi ishlab chiqiladi	
amaliyotga joriy etiladi	

2. Jadvalda berilgan yangi atamalarga mos tushunchalarni toping va ularni chiziq bilan birlashtiring:

JAVASCRIPT	kompyuter operatsion tizimlarini yozish uchun mo'ljallangan til.
JAVA	dasturlashni o'rganish uchun eng mos vizuallashtirilgan dasturlash tili.
PYTHON	kompyuter, mobil telefon va planshetlar uchun dastur yozuvchi til.
SCRATCH	dinamik web-saytlar yaratish uchun mo'ljallangan til.
PHP	interfaol web-saytlarni yozish uchun mo'ljallangan til.
C	turli masalalarni yechish, sun'iy intellekt tizimlari uchun mo'ljallangan dasturlash tili.

27-DARS. PYTHON DASTURLASH TILINI O'RNATISH

Endi Python dasturlash tili yordamida dasturlashni o'rganishni boshlaymiz. Python dunyo miqyosida eng ommalashgan, mukammal dasturiy mahsulotlarni yaratishga mo'ljallangan dasturlash tillaridan biri bo'lib, uning yordamida web-sayt, ilova va o'yinlar yaratiladi. Quyidagi imkoniyatlar bu tilni o'rganish uchun yordamchi omillar hisoblanadi.

O'rganish va qo'llashning soddaligi.

Python sodda va qulay dasturlash tili bo'lib, boshqa dasturlash tillariga nisbatan uning yordamida dastur tuzish qiyinchilik tug'dirmaydi.

Mukammal kutubxonaning mavjudligi. Pythonda dastur tuzish jarayonida kutubxonadagi tayyor funksiyalardan foydalanish mumkin. Bu esa murakkab dasturlarni ham qisqa vaqtda tuzish imkonini beradi.

Tayanch tushunchalar

IDLE (*Integrated Development and Learning Environment* – integrallashtirilgan dasturlash va o'rganish muhiti) – Python tilini o'rganish uchun taqdim etilgan IDE.

Interfaol muhit – dastur kodini kiritish, uni fayl sifatida saqlamasdan turib natijasini ko'rish muhiti.

Dasturlash muhiti – dastur kodini kiritish, alohida fayl sifatida saqlash va ishga tushirish muhiti.

Operator – dasturlash tili buyruqlari.

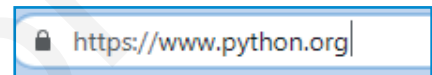


1. Python dasturlash tili qanday o'rnatiladi?
2. Interfaol muhit nima?
3. Dasturlash muhiti nima?
4. IDLE qanday muhit?
5. Xatoliklar qanday aniqlanadi?

Python dasturlash tilini o'rganish uchun uni o'zining rasmiy saytidan yuklab olib, keyin o'rnatish zarur. Python kompyuterga IDLE dasturi bilan birga o'rnatiladi. IDLE dasturlashni endi boshlaganlar uchun mo'ljallangan IDE bo'lib, kod yozish uchun uncha murakkab bo'lmagan matn muharriri hamda dastur natijasi va xatolarni ko'rsatib turuvchi oynaga ega.

Python dasturini o'rnatish

1. Pythonning rasmiy saytiga kiriladi. Brauzer qatoriga <http://www.python.org> manzilini kiritib, **Downloads** bo'limi tanlanadi.

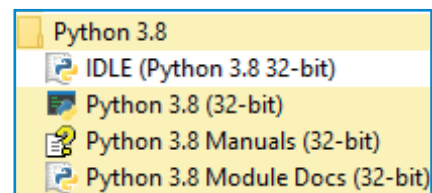


2. Pythonni yuklab olish. Kompyuter operatsion tizimi (masalan, Windows)ga mos o'rnatgichning so'nggi varianti (Python 3.8) yuklab olinadi.

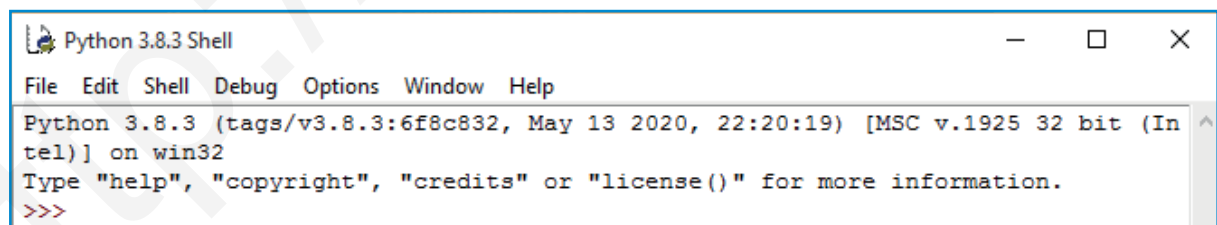


3. O'rnatish. Yuklab olingan o'rnatgich fayl ishga tushiriladi va **"Установить для всех пользователей"** ko'rsatmasi tanlanadi hamda o'rnatish jarayonidagi barcha oyna so'rovlarida **"Next"** tugmachasi bosiladi.

4. IDLEni ishga tushirish. Dastur xatolarsiz o'rnatilganligini tekshirish uchun **"Пуск"** orqali **"Все программы"** ko'rsatmasi tanlanadi, Python papkasi ochiladi va IDLE ishga tushiriladi.



Dastur kodlarini kiritish uchun Python oynasi ochiladi.



IDLE oynasining ikki xil muhiti mavjud bo'lib, ular interfaol muhit va dasturlash muhiti deb nomlanadi.

IDLEning interfaol muhiti (*konsol* deb ham yuritiladi) – dastur kodini kiritib, natijasini darhol ko'rish mumkin bo'lgan oyna. Bu muhit dastur buyruqlari vazifalarini o'rganish, kichik dasturlar tuzish, ularning natijasini tezroq ko'rish uchun ishlatiladi. Interfaol muhitdan kalkulyator sifatida foydalanish ham mumkin.

IDLEning dasturlash muhiti – dastur kodini kiritish, tahrirlash va ishga tushirish mumkin boʻlgan oyna. Dastur natijasi interfaol muhitda chiqariladi. Bu muhit katta dasturlarni tuzish jarayonida ularni saqlash va kerakli vaqtda tahrirlash uchun ishlatiladi. Aks holda, dastur kodini takror-takror yozishga toʻgʻri keladi. Uning yagona kamchiligi – dastur dastlab saqlanadi, undan keyingina ishga tushiriladi.

Python dasturlash muhitida birinchi dasturimizni yaratishga harakat qilamiz. Buning uchun **print()** operatoridan foydalanamiz.

print() operatori yordamida oʻzgaruvchi qiymatini ekranga chiqarish mumkin.

Sintaksisi:

print(chiquvchi maʼlumot)

print – operator (yoki funksiya);

chiquvchi maʼlumot – ekranga chiqarilishi kerak boʻlgan va vergul bilan ajratilgan oʻzgaruvchilar, oʻzgarmaslar yoki ifodalar ketma-ketligi.

Eslab qoling!

IDLE dasturlash muhiti oynasida dastur kodini yozib va saqlab, undan keyin ishga tushirish lozim. Dasturni saqlamasdan turib ishga tushirmang!

IDLEning interfaol muhitida dastur tuzish

1. IDLE ishga tushiriladi. Bu oyna orqali dastur kodini kiritish, natija va xatolarni koʻrishimiz mumkin.

2. `>>>` belgisidan soʻng dastur kodi kiritiladi.

Masalan, `print("Salom!")`;

3. Enter tugmachasini bosib, dastur natijasini koʻrishimiz mumkin.

Yodingizda boʻlsin, keyingi darslarimizda interfaol muhit (konsol oynasi)da dasturlashni quyidagi (moviy) rangda ifodalaymiz:

```
Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("salom");
salom
>>>
```

```
>>> print("salom")
salom
```

IDLEning dasturlash muhitida dastur tuzish

1. IDLE ishga tushiriladi. Bu oynada natija va xatolarni koʻrishimiz mumkin. Dastur kodi dasturlash muhitida kiritiladi.

2. **File** menyusidan **New File** buyrugʻi tanlanadi. (1)

3. Dasturlash muhiti oynasida dastur kodi kiritiladi. (2)

4. **File** menyusidan **Save** yoki **Save as** buyrugʻi tanlanadi. Fayl nomini kiritib, **Save** tugmachasi orqali dastur saqlanadi.

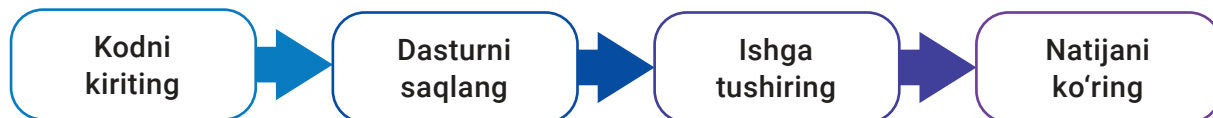
5. Dasturlash muhiti oynasida **Run** menyusida **Run Module** buyrugʻi tanlanadi. (3)

6. Natijani IDLE interfaol oynasida koʻrishimiz mumkin.

Yodingizda boʻlsin, keyingi darslarimizda dasturlash muhitida dasturlashni quyidagi (pushti) rangda ifodalaymiz (2):

```
Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
New File Ctrl+N
Open... Ctrl+O
Open Module... Alt+M
Recent Files
Module Browser Alt+C
```

```
1.py - D:/python-ilova/1.p...
File Edit Format Run Options Window Help
print('salom')
Run Module F5
Run... Customized Shift+F5
Check Module Alt+X
Python Shell
```



Eslab qoling!

Boshqa dasturlash tillari kabi Pythonda ham turli buyruqlarni kiritish uchun ushbu buyruq sintaksisidan foydalaniladi. Shuningdek, Pythonda kodlashning umumiy qabul qilingan uslubini tavsiflovchi hujjat sifatida PEP8 (Python Enhanced Proposal - Python tilini takomillashtirish bo'yicha takliflar) ham ishlab chiqilgan.

<http://dr.rtm.uz/pep8> sahifa orqali PEP8 ning to'liq tavsifi bilan tanishishingiz mumkin.

SAVOL VA TOPSHIRIQLAR



1. Python dasturlash tili nima maqsadda yaratilgan?
2. Python dasturlash tili qanday ishga tushiriladi?
3. Interfaol muhit nima?
4. Dasturlash muhiti nima?
5. IDLEning interfaol va dasturlash muhitlarining bir-biridan farqi nimada?
6. Python dasturlari qaysi muhitda kiritiladi?

UY VAZIFASI



1. IDLEning interfaol muhitida "Mening birinchi dasturim" satrini ekranga chiqaring.
2. IDLEning dasturlash muhitida "Mening birinchi dasturim" satrini ekranga chiqaruvchi dastur yozing, uni saqlang va natijani ko'ring.
3. Kichik loyiha ishini tayyorlang (matn yoki taqdimot muharriri yordamida).

Mavzu: Python dasturlash tili

Reja:

1. Python dasturlash tili tarixi.
2. Python dasturlash tili haqida qisqacha ma'lumot.
3. Python dasturlash tilida yaratilgan dastur va o'yinlar.
4. Python dasturlash tili muhiti menyularining vazifalari.

28-DARS. PYTHONDA O'ZGARUVCHILARNI TAVSIFLASH

Har bir tilning alifbosi bo'lgani kabi dasturlash tilining ham o'z alifbosi mavjud. Python dasturlash tilining alifbosi katta va kichik lotin harflari, arab raqamlari, maxsus belgilar va xizmatchi so'zlardan tarkib topgan.

BUNI BILASIZMI?



1. Python dasturlash tilining alifbosi qaysi harf, raqam va belgilarni o'z ichiga oladi?
2. Identifikator nima?
3. O'zgaruvchilar va doimiylarning farqi nimada?
4. O'zgaruvchilarning qanday turlari mavjud?

Python dasturlash tilining alifbosi

Katta va kichik lotin harflari

A, B, C, ..., X, Y, Z, a, b, c, ..., x, y, z

Arab raqamlari

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Maxsus belgilar

Arifmetik amallar +, -, *, /,
qavslar, tinish belgilari va b.

Xizmatchi so'zlar

if, for, print, input, class va b.

Tayanch tushunchalar

O'zgaruvchilar – o'z qiymati va turiga ega kattalik, o'zida qiymatlarni saqlaydigan kompyuter xotirasidagi yacheyka nomi.

O'zgaruvchining qiymatlari dastur davomida o'zgarib turishi mumkin.

Doimiy (o'zgarmas) – faqat o'qish uchun mo'ljallangan qiymatlarni saqlovchi kompyuter xotirasidagi yacheyka nomi.

Doimiy o'zgaruvchilar kabi o'z qiymati va turiga ega.

Identifikatorlar – o'zgaruvchilar, doimiy, funksiyalar, protseduralar, modullar, dasturlarning umumiy nomi.

Odatda, dasturlar kiritilgan ma'lumotlarni qabul qilish, qayta ishlash, shuningdek, natijani ekranga chiqarish uchun mo'ljallangan bo'ladi. Dasturlarni yozishda o'zida asosiy ma'lumotlarni saqlaydigan o'zgaruvchi yoki doimiydan foydalaniladi. O'zgaruvchilar dastur jarayonida o'zgarishi mumkin bo'lgan ma'lumotlarni belgilaydi, doimiydan esa o'zgarmas ma'lumotlar uchun foydalaniladi. O'zgaruvchilar va doimiy larni belgilash uchun turli nomlar, ya'ni identifikator (identification)lardan foydalaniladi.

Identifikatorlar harf va raqamlar kombinatsiyasidan tarkib topadi. Masalan, a25, b5c88, sonlar to'plami va boshqalar. Pythonda katta va kichik harflar bir-biridan farq qiladi.

Eslab qoling!

O'zgaruvchilarni e'lon qilish qoidalari:

o'zgaruvchi nomida **ixtiyoriy harf yoki raqamdan** foydalanish mumkin;

katta va kichik harflar bir-biridan farqlanadi. belgi5, Belgi5, BELGI5 nomli o'zgaruvchilar har xil o'zgaruvchi nomlaridir, shu sababli kichik harflardan foydalangan ma'qul;

o'zgaruvchi nomlarida **probellar** ishlatish mumkin emas, uning o'rniga so'zlar ('_')

tagchiziq belgisi yordamida bir-biriga bog'lanadi;

o'zgaruvchining nomi **raqamdan boshlanishi** mumkin emas;

-, /, # yoki @ **belgilarni** ishlatish mumkin emas;

maxsus buyruq nomlarini ishlatish mumkin emas;

and, as, assert, break, class, continue, def, del, elif, else, except, finally, for, from, global, if, import, in, is, lambda, nonlocal, not, or, pass, print, raise, return, try, while, with, yield kabi **xizmatchi so'zlarni** o'zgaruvchi nomi sifatida ishlatish mumkin emas.

O'zgaruvchini e'lon qilish.

O'zgaruvchi "=" belgisi yordamida satr yoki sonni o'zlashtirishi mumkin va bu *o'zgaruvchining qiymati* deb ataladi. O'zgaruvchiga sonli qiymat berish uchun "=" belgisidan so'ng son yoziladi. Satrli qiymat berilganda esa "=" belgisidan so'ng bittalik (' '), ikkitalik (" ") qo'shtirnoq ichida satr yoziladi.

O'zgaruvchi qiymatini o'zgartirish.

O'zgaruvchi qiymatini o'zgartirish uchun unga yangi qiymat berish kifoya.

O'zgaruvchilarni o'zaro ishlatish.

Ikkita o'zgaruvchida birining qiymatini ikkinchisiga o'zlashtirish uchun «=» belgisidan foydalaniladi.

```
>>> age = 15
>>> print (age)
15
>>> name = 'Anvar'
>>> print (name)
Anvar
>>> age = 18
>>> print (age)
18
```

```
>>> age = 18
>>> grad_age = age
>>> print (age, grad_age)
```

18 18

Yacheyka nomi	age	grad_age
Yacheyka qiymati	18	18

bitirgan_yoshi o'zgaruvchisi yoshi o'zgaruvchisining qiymatini o'zlashtirib oldi. Natijada ikkala o'zgaruvchining qiymatlari bir xil chiqdi.

```
>>> age = 18
>>> grad_age = age
>>> age = 22
>>> print (age, grad_age)
```

22 18

Yacheyka nomi	age	grad_age
Yacheyka qiymati	22	18

bitirgan_yoshi o'zgaruvchisi yoshi o'zgaruvchisining qiymatini o'zlashtirib oldi, yoshi o'zgaruvchisiga yangi qiymat berildi. Natijada yoshi yangi qiymatni, bitirgan_yoshi avvalroq o'zlashtirgan qiymatni chiqardi.

Misol.

```
>>> a = 6
>>> b = a
>>> a = 10
>>> c = a + b
>>> print (c)
```

16

Doimiylar

Doimiy (konstanta)larni ifodalash uchun faqat bosh harflardan foydalaniladi.

Masalan,
PI = 3.1415



1. Python dasturlash tilining alifbosi qanday tuzilmadan iborat?
2. Identifikator nima?
3. O'zgaruvchan va doimiyning bir-biridan farqli jihati nimada?
4. O'zgaruvchilarni nomlashda qanday belgilardan foydalanish mumkin emas?

1. Darvozaning eni 4 metr, bo'yi esa 3 metr:

- 1) darvoza yuzasi (S)ni hisoblash dasturini tuzing;
- 2) darvoza qirrasining uzunligi (P)ni hisoblash dasturini tuzing.

2. Ikkita qayiq turg'un suvda bir-biriga tomon 4 km/soat va 2 km/soat tezlik bilan suzmoqda. Ular orasidagi masofa 24 km bo'lsa:

- 1) ular qancha vaqtdan keyin uchrashadi?
- 2) qancha vaqtdan keyin ular orasidagi masofa 12 km ni tashkil etadi?

3. Doiraning radiusi 4 metrga teng ($\pi=3.14$):

- 1) doira yuzini hisoblang;
- 2) aylana uzunligini hisoblang.

29-DARS. PYTHONDA XATOLIKLAR BILAN ISHLASH

Har qanday dasturni yozish jarayonida turli xatoliklarga yo'l qo'yilishi mumkin. Agar yozilgan dastur tarkibida xato uchraydigan bo'lsa, u holda dastur ishga tushmaydi va ekranga yo'l qo'yilgan xatolik to'g'risida xabar chiqadi.

1. Dastur tuzish jarayonida qanday xatoliklar yuzaga kelishi mumkin?
2. Dasturda uchragan xatolikni bartaraf etish mumkinmi?
3. Python dasturlash tilida xatolik uchrab qolsa, u qanday bartaraf etiladi?
4. `NameError` qanday xatolikni anglatadi?

Dasturlash muhitidagi xatolik va uni tuzatish

Dasturlash muhitida dastur kodi yozilib, dastur ishga tushirilganda, xatolik haqidagi xabar (`SyntaxError`) oynasi ekraniga chiqadi. Bu esa dasturning kodida uni ishga tushirishga to'sqinlik qiluvchi qandaydir xato borligini anglatadi.

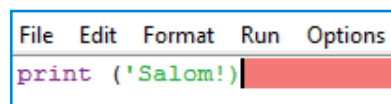
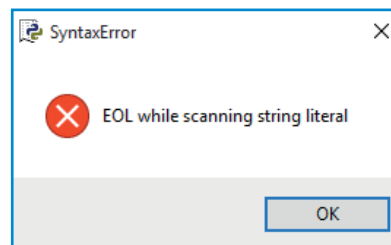
1. Xatolik haqidagi xabar (`SyntaxError`) oynasida ko'rsatilgan xatolik aniqlanadi (masalan, `invalid syntax` – matnni terishda xatolik bor – `unexpected indent` – dasturda noto'g'ri joy ajratilgan va h. k.). Ok tugmachasini bosib, dastur oynasiga qaytiladi.

2. Xatolik mavjud bo'lgan – qizil bilan belgilangan – qator sinchiklab tekshiriladi va xatolik tuzatiladi.

Interfaol muhitida yuz bergan xatolik va uni tuzatish

Ba'zan yo'l qo'yilgan xatolik interfaol muhitda qizil rang bilan namoyon bo'ladi. Bunday xatolik dasturning ishga tushishiga to'sqinlik qiladi.

1. `NameError` xatoligi Python dasturida qaysidir so'z tanilmayotganini anglatadi (masalan, `print` o'rniga `pront` yozilgan) (1).



2. Agar interfaol muhitda shunday xatolik yuz bersa, **File** so'zi bilan boshlanuvchi satr ustida sichqonchani o'ng tugmachasini bosib (2), **Go to file/line** tanlanadi (3).

3. Dasturlash muhitida xatolik ko'rsatilgan satrga o'tib, noto'g'ri ko'rsatilgan buyruq nomi tuzatiladi (4).

Xatolik aniqlangan satr (2)

```
Traceback (most recent call last):
  File "C:/Users/User/AppData/Local/Programs/Python/Python38-32/111.py", line 1,
  in <module>
    pront ('Salom!');
NameError: name 'pront' is not defined
>>>
```

Python tanimayotgan so'z (1)

Cut
Copy
Paste
Go to file/line
Squeeze

Dasturlash muhitiga o'tish (3)

```
print ('Salom!');
```

(4)

SAVOL VA TOPSHIRIQLAR



1. Python dasturlash tili muhitida qanday xatoliklar uchrashi mumkin?
2. Xatoliklar qanday bartaraf etiladi?
3. SyntaxError qanday xatolik?
4. NameError qanday xatolik?

Eslab qoling!

Dasturlashda tez-tez uchrab turuvchi xatoliklar.

Katta yoki kichik harflar. Agar dasturda bir vaqtda **print** va **Print** so'zlari ishlatilsa, Python bu qanday buyruq ekanligini tushunmaydi.

Bittalik va ikkitalik qo'shtirnoqlar. Qo'shtirnoqlarning bu ikkala turini aralashtirib ishlatish mumkin emas. Ochiluvchi qo'shtirnoq sifatida qaysi biri ishlatilsa, yopishda ham mana shu qo'shtirnoq ishlatilishi lozim.

Minus va tagchiziq. Minus (-) va tagchiziq(_) belgilarini adashtirish zinhor mumkin emas.

Turli qavslar. Qavslarning bir necha turi mavjud: (), {} va []. Ochiluvchi va yopiluvchi qavslar bir-biriga mos bo'lishi lozim.

UY VAZIFASI



1. Quyida berilgan dasturiy koddagi xatolikni toping:

a

```
>>> a = 12
>>> b = 18
>>> k = a + b
>>> print (c)
```

b

```
>>> A = 12
>>> B = 18
>>> C = a + b
>>> print (c)
```

d

```
>>> age=15
>>> print ("Ahmadning yoshi:")
>>> print (ag)
```

e

```
>>> age = 15
>>> print("Ahmadning yoshi:")
>>> print(age)
```

f

```
>>> age = 15
>>> grad_year = 2019
>>> print("Ahmadning yoshi:")
>>> print(age)
>>> print("O'qishni tamomlagan yili:")
>>> print(grad_year)
```

30-DARS. PYTHONDA MA'LUMOT TURLARI

Ma'lumki, axborotlar matnli, raqamli, ovozli, grafik va boshqa shakllarda uzatilishi mumkin. Bunday axborotlarni dasturlash tillarida qayta ishlash uchun, ularni turlarga ajratish lozim. Dasturlarda foydalaniladigan ma'lumotlar turlari dasturning maqsadiga bog'liq bo'ladi: oddiy kalkulyator sonlardan foydalanadi, elektron pochta manzillarini tekshirishga mo'ljallangan dastur esa matnlar bilan ish ko'radi. Sonlar natural, butun va haqiqiy sonlarga ajratiladi. Matnli axborotlar esa belgilar yoki satrli ma'lumotlardan iborat bo'lishi mumkin.

1. Ma'lumotlar turi nima?
2. Ma'lumotlarning qanday turlarini bilasiz?
3. Ma'lumotlar turini o'zgartirish mumkinmi?



BUNI BILASIZMI?

Ma'lumotlar turi – bu o'zgaruvchi yoki doimiy qiymatlardagi ma'lumotlar shakli.

Ma'lumotlar turi kompyuter xotirasida yetarlicha joyni zaxiraga olib qo'yish uchun kerak bo'ladi.

Odatda, dasturlash tillarida ma'lumotlar turi o'zgaruvchi yoki doimiy bilan birga e'lon qilinadi. Python dinamik turlarga ajratuvchi dasturlash tili hisoblanadi. Shu sababli, Pythonda o'zgaruvchining turi u foydalanayotgan qiymat bo'yicha belgilanadi, lekin ma'lumot turini boshqa turga o'zgartirish uchun tur ko'rsatilishi shart.

Tayanch tushunchalar

Ma'lumotlar turi – kompyuter xotirasidagi yacheykada saqlanadigan ma'lumotlar shakli.

Ma'lumotlar turi	Ma'lumotlar turi tavsifi	Misol
<code>int()</code>	Butun sonlar , masalan, o'quvchilar sonini ifodalash uchun.	<code>>>> yoshi = 15</code>
<code>float()</code>	Haqiqiy sonlar , masalan, pul miqdorini ifodalash uchun.	<code>>>> narxi = 20.45</code>

<code>str()</code>	Satrlı , masalan, soʻz yoki gaplarni ifodalash uchun.	<pre>>>> name = 'Ahmad'</pre>
<code>bool()</code>	Mantiqiy , rost yoki yolgʻon ekanligini ifodalash uchun	<pre>>>> a= True >>> b= False</pre>

Python dasturlash tilida maʼlumotlarning boshqa turi ham mavjud, ular bilan keyinroq tanishamiz.

Maʼlumotlar turini oʻzgartirish

Oʻzgaruvchi tarkibida ixtiyoriy turdagi maʼlumot saqlanishi mumkin. Maʼlumotlar turini oʻzgartirish uchun mos maʼlumotlar turi buyruqlaridan foydalaniladi. **input() operatori** yordamida kiritilgan maʼlumotlar satrli koʻrinishda boʻladi.

Eslab qoling!

Oʻzgaruvchi qabul qilgan qiymat turini **type()** funksiyasi yordamida aniqlash mumkin.

```
>>> name = 'Ahmad'
>>> type(name)
<class 'str'>
```

Sintaksisi:

input(kiruvchi maʼlumot)

input – operator (yoki funksiya);

kiruvchi maʼlumot – bu oʻzgaruvchi yoki vergul bilan ajratilgan oʻzgaruvchilar ketma-ketligi.

Butun son turidagi maʼlumotlarni kiritish uchun **int(input())** koʻrinishida ishlatish lozim. Yoki kiritilgan maʼlumot turini **int()** funksiyasi yordamida oʻzgartirish kerak.

Misol:

```
>>> age=input('Enter your age:')
Enter your age: 15
>>> print(age + 1)
```

Enter your age: 15

```
>>> print(age + 1)
```

```
Traceback (most recent call last):
  File «<pyshell#27>»,
    line 1, in <module>
      print(age+1)
TypeError: can only concatenate
str (not «int») to str
```

```
>>> age=input('Enter your age:')
Enter your age: 15
>>> print(int(age) + 1)
```

Enter your age: 15

```
>>> print(int(age) + 1)
```

16

Ushbu misolda yoshi oʻzgaruvchisiga 1 sonini qoʻshishda xatolik kelib chiqdi, sababi foydalanuvchi tomonidan kiritilgan 15 ni kompyuter satrli oʻzgaruvchi deb qabul qildi va satrga sonni qoʻsha olmadi.

Ikkinchi holatda esa foydalanuvchi tomonidan kiritilgan 15 satrli oʻzgaruvchini **int()** buyrugʻi yordamida butun songa aylantirib oldi va 15 soniga 1 ni qoʻshib natijani chiqardi.



1. Oʻzgaruvchi qiymati qaysi turdagi maʼlumotlarni qabul qiladi?
2. Identifikator nima?
3. Oʻzgaruvchi va doimiyning bir-biridan farqli jihati nimada?
4. Oʻzgaruvchilarni nomlashda qanday belgilardan foydalanish mumkin emas?

1. Darvozaning eni va bo'yi foydalanuvchi tomonidan kiritiladi:
 - 1) darvoza yuzasini hisoblash dasturini tuzing;
 - 2) darvoza qirrasining uzunligini hisoblash dasturini tuzing.
2. Quyidagi o'zgaruvchilar turini aniqlang:
alfa = 8764; beta = 'Lola'.
3. O'zgaruvchi uchun qiymat kiritilganda, uning turini aniqlovchi dastur tuzing.

31-DARS. AMALIY MASHG'ULOT

Agar o'zgaruvchiga butun son yuklansa – int, haqiqiy son yuklansa – float, bittalik yoki ikkitalik qo'shtirnoq yordamida satr yuklansa – str turiga ega bo'ladi.

Misol.

```
>>> a = 12
>>> print(type(a))
<class 'int'>
>>> b = 12.3e+2
>>> print(type(b))
<class 'float'>
>>> name='Aqljon'
>>> print(type(name))
<class 'str'>
>>> name='Aqljon'
>>> print(type(name))
<class 'str'>
>>> parol='@Asdf123'
>>> print(type(parol))
<class 'str'>
```

1. Quyidagi o'zgaruvchilar turini aniqlang:
alfa = 579413; beta = 'Aqlbek'; d = True; s = 0; resp = 'd'; b = 100; max = False;
fc = 'True34'; t = 102.5; res = '2500'; a = '-50'; b = 45.67.
2. Ikki qayiq turg'un suvda bir-biriga tomon a km/soat va b km/soat tezlik bilan suzmoqda. Ular orasidagi masofa S km bo'lsa, ular qancha vaqtda uchrashadi? a va b foydalanuvchi tomonidan kiritiladi.
3. Xonaning bo'yi (x) va eni (y) metr foydalanuvchi tomonidan kiritiladi. Xonaning yuzasi (S) va perimetri (P)ni topish dasturini tuzing.
4. Uchburchak asosining uzunligi (c) va balandligi (h) foydalanuvchi tomonidan kiritiladi. Uchburchak yuzasi (S)ni hisoblash dasturini tuzing.
5. Avtomobilning o'rtacha tezligi v (km/soat) va bosib o'tgan yo'li s (km) foydalanuvchi tomonidan kiritiladi. Avtomobilning yurgan vaqti t (soat)ni hisoblash dasturini tuzing.
6. Doiraning radiusi r ga teng. Doiraning yuzi (s) va aylana uzunligi (l) ni topish dasturini tuzing. Doiraning radiusi foydalanuvchi tomonidan kiritiladi ($Pi=3.14$).

32-DARS. NAZORAT ISHI

1. Radiusi R ga teng doiraning yuzasini hisoblash algoritmini ixtiyoriy usulda tuzing.
2. MS Excel dasturida $y = x^2 + 3$ funksiya grafigini hosil qilish algoritmini soʻzlar yordamida yozing.
3. Uchburchakni asosi va balandligi berilgan. Uchburchak yuzasini topish algoritmini tuzing.
4. Kiritilgan kitob nomi siz mutolaa qilgan kitob boʻlsa, ekranga "Bu kitobni oʻqiganman", aks holda, "Afsuski, bu kitobni oʻqimaganman", degan yozuvni chiqaruvchi algoritmni soʻzlar yordamida yozing va blok-sxemasini tuzing.
5. Quyidagi funksiyalar qiymatini hisoblovchi algoritm blok-sxemasini tuzing:
$$y = \begin{cases} x + 0.3, & \text{agar } x > -6 \\ \frac{2+x}{5}, & \text{agar } x \leq -6 \end{cases}$$
6. -30 dan 20 gacha boʻlgan sonlar oraligʻidagi juft sonlar koʻpaytmasini hisoblash algoritmini tuzing.
7. 15 ta butun son berilgan. Quyidagilarni aniqlash algoritmini tuzing:
 - a) manfiy sonlar yigʻindisini;
 - b) musbat sonlar sonini;
 - d) 3 ga karrali sonlar sonini.
8. Sonning ishorasini aniqlash algoritmini tuzing.
9. 30 dan katta va 90 dan kichik 6 ga karrali sonlar yigʻindisini hisoblash algoritmini soʻzlar yordamida ifodalang va blok-sxemasini tuzing.
10. Boshlangʻich tezligi v_0 boʻlib, a tezlanish bilan tekis harakat qilayotgan moddiy nuqtaning t vaqt ichida bosib oʻtadigan yoʻlini aniqlash dasturini tuzing (yoʻllanma: $S = v_0 t + at^2 / 2$).

33–34-DARSLAR. PYTHONDA ARIFMETIK AMALLARNI BAJARISH

Python dasturlash tilida sonli maʼlumotlar ustida turli arifmetik amallarni bajarish mumkin. Agar ifodalar toʻgʻri tuzilsa, hisoblash topshiriqlarini bajaruvchi dasturlarni tuzish mumkin boʻladi.



1. Arifmetik amallarni sanab bering.
2. Arifmetik amallardan boshqa amallar ham bormi?
3. Boʻlinmaning qoldigʻi qanday hisoblanadi?

Arifmetik amallar

Amal nomi	Amal belgisi		Misol
Qo'shish	+	$x + y$	<code>print(7+5) # 12</code>
Ayirish	-	$x - y$	<code>print(7-5) # 2</code>
Ko'paytirish	*	$x * y$	<code>print(7*5) # 35</code>
Bo'lish	/	x / y	<code>print(7/5) # 1.4</code>
Bo'linmaning butun qiymatini hisoblash	//	$x // y$	<code>print(7//5) # 1</code>
Bo'linmaning qoldig'ini hisoblash	%	$x \% y$	<code>print(7%5) # 2</code>
Darajaga ko'tarish x^y	**	$x ** y$	<code>print(5**2) # 25</code>

Dastur tuzish jarayonida tez-tez foydalaniladigan amallardan biri bu – o'zgaruvchini belgilangan miqdorga oshirish yoki kamaytirishdir. Bunday amallarni bajarish uchun **(+=)** **increment** (oshirish) va **(-=)** **decrement** (kamaytirish) amallaridan foydalaniladi.

Arifmetik amallarda o'zlashtirish amali (=) yonma-yon qo'llanganda, amal hisoblanib, natija chap tomondagi o'zgaruvchiga tenglanadi.

Dasturda amallarni qisqa holda qo'llash

Amal belgisi	Ifodaning qisqa yozilishi	Ifodaning to'liq yozilishi	Misol x=4
+=	$x+=y$	$x=x+y$	<code>x+=1 # 5</code>
-=	$x-=y$	$x=x-y$	<code>x-=2 # 2</code>
=	$x=y$	$x=x*y$	<code>x*=2 # 8</code>
/=	$x/=y$	$x=x/y$	<code>x/=2 # 2</code>
//=	$x//=y$	$x=x//y$	<code>x//=2 # 2</code>
%=	$x%=y$	$x=x\%y$	<code>x%=2 # 0</code>
=	$x=y$	$x=x**y$	<code>x**=2 # 16</code>

Ifodalar amallarning bajarilish tartibini bildiradi. Ifodalar o'zgaruvchi, doimiy, qavs va amallardan tashkil topadi.

Matematik ifoda	Ifodaning dasturlash tilida yozilishi
$y = \frac{x^2 + x - 3}{x^2 + 5x} + \frac{1}{x}$	<code>y=(x**2+x-3)/(x**2+5*x)+1/x</code>

Misol. To'rt xonali son berilgan. Ushbu son birinchi va oxirgi xonasi raqamining ko'paytmasini topish dasturini tuzing.

<pre>>>> print('Enter a 4-digit number') Enter a 4-digit number >>> x=int(input()) 4568</pre>	
<pre>>>> a= x// 1000 >>> b= x % 10 >>> c= a * b >>> print(c, '-multiply the first and last digit of the number', x)</pre>	<pre>x=4568 a=4568//1000=4 b=4568%10=8 c=4*8=32</pre>
<pre>32-multiply the first and last digit of the number 4568</pre>	



1. Qanday arifmetik amallarni bilasiz?
2. Bo'linmaning butun qiymati qanday hisoblanadi?
3. Bo'linmaning qoldig'i qanday hisoblanadi?
4. Sonni darajaga ko'tarish qanday amalga oshiriladi?



1. Berilgan a va b sonlari yig'indisi, ko'paytmasini toping. Yig'indi va ko'paytma oxirgi raqamlarining ko'paytmasini topish dasturini tuzing.

Berilgan	Ifoda	Natija
$a = 8$	$d = a + b = 8 + 9 = 17$	$S = 7 * 2 = 14$
$b = 9$	$c = a * b = 8 * 9 = 72$	

2. Berilgan ikki xonali son xonalarining yig'indisini hisoblash dasturini tuzing.
3. Quyidagi ifodani Python tilida yozing:
 - 1) $x + 2y + 5^2 * 4 - 58$
 - 2) $256 + (2589 - 1549) * 458 + 456^{14} - 4565/5$

35-DARS. AMALIY MASHG'ULOT

Masala. Arqonning uzunligi millimetrlarda berilgan. Shu uzunlikni kilometr, metr, santimetr va millimetrlarda ifodalash dasturini tuzing.

```
>>> a=int(input())
12325458
>>> b=a//1000000 # necha km
>>> a=a%1000000 # qolgan qismi
>>> c=a//1000 # necha metr
>>> a=a%1000 # qolgan qismi
>>> d=a//10 # necha cm
>>> a=a%10 # qolgan qismi mm da
>>> print(b, 'km' , c, 'm', d, 'cm', a, 'mm')
```

12 km 325 m 45 cm 8 mm

```
a=12325458
b=12325458//1000000=12
a=12325458%1000000=325458
c=325458//1000=325
a=325458%1000=458
d=458//10=45
a=458%10=8
```



1. Hovuzning bo'yi 6 metr, eni 4 metr va chuqurligi 3 metr. Berilganlar asosida quyidagi topshiriqlar uchun dastur tuzing:

- a) Hovuz ichiga kafel qoplash uchun qancha kvadrat metr kafel kerak bo'ladi?
- b) Hovuzni to'ldirish uchun qancha litr suv kerak bo'ladi ($1 l = 1000 \text{ cm}^3$)?

2. Yuqoridagi masalada hovuz bo'yi, eni va chuqurligi kattalıkları foydalanuvchi tomonidan kiritiladigan hol uchun dastur tuzing.

- 10 288 sekundni soat, minut va sekundlarda ifodalang.
- Berilgan ikki xonali son xonalarining ko'paytmasini hisoblash dasturini tuzing.
- Berilgan uch xonali son xonalarining yig'indisi va ko'paytmasini hisoblash dasturini tuzing.

Berilgan	Ifoda	Natija
897	8+9+7	24
	8*9*7	504

6. Qisqartirilgan amallarni qo'llagan holda, quyidagi dastur natijasini chiqaring.

Berilgan	Ifoda	Natija
a=8 b=5 c=9	a=int(input()) b=int(input()) c=int(input()) a=b a*=c a+=(b*c+b) print(a)	

7. Quyidagi ifodani Python tilida yozing: $y = \frac{7}{x^2+x+1} + x^2$

36-DARS. PYTHONDA SATRLAR BILAN ISHLASH

Python matn va uning qismlari bilan ishlash uchun eng qulay dastur hisoblanib, uning yordamida satrlarni bir-biriga bog'lash yoki satr ichidan biror qismini qirqib olish mumkin.

BUNI BILASIZMI?



1. Satr nima?
2. Satr orasidan qism satr qirqib olish mumkinmi?
3. Satr uzunligi qanday aniqlanadi?

Tayanch tushunchalar

Satrlar – qo'shtirnoq ichiga olingan Unicode kodidagi belgilar ketma-ketligi.

Satrlar o'zgaruvchilarni yaratish va satrlar bilan ishlash

Satr – harf, son va belgilar hamda probeldan tarkib topgan belgilar ketma-ketligi. Satrlarni o'zgaruvchilar yordamida kiritish mumkin. Pythonda satrlar bittalik va ikkitalik qo'shtirnoqlar orqali beriladi.

Satrlar ustida keng bajariladigan amallardan biri bu – birlashtirish amali. Satrlarni birlashtirish uchun + amali qo'llaniladi.

Eslab qoling!

Satr uzunligini **len()** funksiyasi yordamida aniqlash mumkin. Python barcha belgi va probellar sonini o'zi hisoblab chiqaradi.

```
>>> a='Bahor!'
>>> len(a)
```

6

```
>>> a = 'Good morning!'
>>> b = 'Welcome.'
>>> c=a + b
>>> print(c)
```

```
Good morning! Welcome.
```

```
>>> c = a+'Dear pupil.' + b
>>> print(c)
```

```
Good morning! Dear pupil. Welcome.
```

Pythonda bitta soʻzni ekranga bir necha marta chiqarish imkoniyati mavjud boʻlib, buning uchun uni bir marta yozishning oʻzi kifoya.

```
>>> a = 'Hello!'
>>> print(a * 10)
```

```
Hello! Hello! Hello! Hello! Hello!
Hello! Hello! Hello! Hello! Hello!
```

Qism satrlarni belgilash

Satrdagi har bir belgining oʻz raqami boʻlib, u belgining turgan oʻrnini anglatadi. Bunday raqam belgining turgan oʻrnini aniqlash yoki uni satrdan qirqib olish uchun kerak boʻladi. Pythonda satrdagi belgilarni raqamlash 0 dan boshlanadi va bu raqam *indeks* deb ataladi.

O	.	Z	B	E	K	I	S	T	O	N
0	1	2	3	4	5	6	7	8	9	10

Satrdan nafaqat belgi, balki qism satrni ham qirqib olish mumkin. Buning uchun quyidagi sintaksisdan foydalanamiz:

`a[index]` – **a** satrdagi **index**da turgan belgini qirqib oladi.

`a[:end]` – **a** satrdan 0 indeksdan boshlab **end** indeksgacha boʻlgan belgilar ketma-ketligini qirqib oladi;

`a[start:end]` – **a** satrdan **start** indeksdan boshlab **end** indeksgacha boʻlgan belgilar ketma-ketligini qirqib oladi;

`a[start:]` – **a** satrdan **start** indeksdan boshlab oxirigacha boʻlgan belgilar ketma-ketligini qirqib oladi;

`a[start:end:step]` – **a** satrdan **step** qadam bilan **start** indeksdan boshlab **end** indeksgacha boʻlgan belgilar ketma-ketligini qirqib oladi.

```
>>> a = "O'ZBEKISTON"
>>> a[4]
```

```
'E'
```

```
>>> a[3:6]
```

```
'BEK'
```

```
>>> a[:6]
```

```
'O'ZBEK'
```

```
>>> a[6:]
```

```
'ISTON'
```

```
>>> a[3:10:3]
```

```
'BIO'
```

4 indeksdagi belgini chiqaradi.

3 indeksdan 6 indeksgacha boʻlgan belgilar ketma-ketligini chiqaradi.

0 indeksdan 6 indeksgacha boʻlgan belgilar ketma-ketligini chiqaradi.

6 indeksdan oxirigacha boʻlgan belgilar ketma-ketligini chiqaradi.

3 indeksdan 10 indeksgacha boʻlgan belgilar ketma-ketligini 3 qadam bilan chiqaradi.



1. Satrlarni birlashtirish qanday amalga oshiriladi?
2. Satr uzunligi qaysi funksiya yordamida aniqlanadi?
3. Satrdan qism satr qirqib olish qanday bajariladi?
4. "*" amali satrlar bilan ishlashda qanday vazifa bajaradi?



1. "gul", "is", "ton" qism so'zlaridan so'z yasang.
 2. Ismingizni ekranga 5 marta chiqarish dasturini tuzing.
 3. Sinf (masalan, 9) va ism (masalan, Dilshod) foydalanuvchi tomonidan kiritiladi.
- "Men 9-sinf o'quvchisi – Dilshodman" satrini chiqaruvchi dastur chizing.

37-DARS. AMALIY MASHG'ULOT

Maxsus belgilar: Pythonda boshqa tillardagi kabi quyidagi maxsus belgilar mavjud:

- \t – tabulyatsiya belgisi;
- \n – yangi satrga o'tish belgisi;
- \' – bittalik qo'shtirnoq belgisi;
- \" – ikkitalik qo'shtirnoq belgisi.

```
>>> print("Good morning.\nWelcome!")
```

```
Good Morning.
```

```
Welcome!
```

Misol. Buyruqlarni ketma-ket berish orqali oddiy shakllarni yasash mumkin:

```
>>> print('*' * 12)
>>> print('*' * 4 + ' ' * 4 + '*' * 4)
>>> print('*' * 1 + ' ' * 10 + '*' * 1)
>>> print('*' * 4 + ' ' * 4 + '*' * 4)
>>> print('*' * 12)
```

```
*****
*****
*
*****
*****
```



1. "GULTOJIXO'ROZ" so'zidan qism so'zlar ajratuvchi dastur tuzing.
2. TUZAMAN, PYTHON, DASTUR, TILIDA, MEN. Ushbu so'zlardan jumla tuzuvchi dastur tuzing.
3. Foydalanuvchi tomonidan kiritilgan matn uzunligini aniqlovchi dastur tuzing.
4. "*" lar yordamida uchburchak yasang.

```
*
**
***
****
```

5. "+" yordamida kvadrat yasang.

```
+++++
+++++
+++++
+++++
```

6. Belgilar yordamida pingvin yasang.

```
~
--
(O O)
/ V \
/( _ )\
^^ ^^
```

38-DARS. PYTHONDA OPERATOR VA IFODALAR

Avvalgi darslarimizda **input()** va **print()** operatorlari yordamida ma'lumotlarni dasturga kiritish hamda natijani ekranga chiqarishning eng oddiy usullarini ko'rib chiqqan edik. Endi **input()** va **print()** operatorlarining boshqa ish usullari bilan tanishasiz.



1. `input()` operatori qanday vazifa bajaradi?
2. `print()` operatori qanday vazifa bajaradi?
3. Ma'lumotlarni kiritishning qanday usullari mavjud?
4. Ma'lumotlarni chiqarishning qanday usullari mavjud?

Ma'lumotlarni kiritish usuli

split() usuli **input()** operatori orqali kiritilgan satrli ma'lumotlarni ajratuvchi belgi yordamida qismlarga ajratadi. Odatda, probel ajratuvchi belgi vazifasini bajaradi. Agar satr tarkibidagi qismlarni ajratish uchun boshqa belgi ishlatilgan bo'lsa, u holda bu belgini **split()** qavslari ichida ko'rsatish kerak bo'ladi.

Sintaksisi:
split(sep)

split – operator (yoki funksiya);
sep – qiymatlarni ajratuvchi belgi.

Misol:

```
>>> a=input().split()
book pen pencil notebook
>>> print(a)
['book', 'pen', 'pencil', 'notebook']
```

```
>>> a=input().split(';')
5;8;7;1;2
>>> print(a)
['5', '8', '7', '1', '2']
```

Ma'lumotlarni chiqarish usullari

print() ma'lumotlarni ekranga chiqarish yoki faylga yozish vazifasini bajaradi. Ma'lumotlarni chiqarish usullari yordamida ma'lumotni ixtiyoriy ko'rinishda aks ettirish mumkin. Buning uchun `print()` operatorining to'liq sintaksisidan foydalanish mumkin.

print('natija', sep = ' ', end= ' ')

sep = ' ' argumenti natija ma'lumotlarini ajratish uchun ishlatiladi. Ma'lumotlarni ajratishning ikkita usuli mavjud: defis "-" (uning o'rniga boshqa belgi ham qo'yish mumkin, masalan "+" yoki "*") hamda yangi satr (\n) belgisi orqali.

```
>>> a='Yulduz'
>>> b='14'
>>> c='years old'
>>> print(a,b,c, sep='+')
Yulduz+14+years old
>>> print(a,b,c, sep='\n')
Yulduz
14
years old
```

Eslab qoling!

end va **sep print()** funksiyasining argumentlari bo'lib, ma'lumotlarni chiqarish parametrlarini o'zgartirish uchun qo'llaniladi.

end= ' ' Natijaviy satr qanday belgi bilan tugashini belgilaydi.

```
>>> a = 'Yulduz'
>>> b = '14'
>>> c = 'years old'
>>> print(a,b,c, sep='-', end='.')
```

Yulduz-14-years old.

Masala. Anvarga 478 + 874 amalini dasturga kiritib, hisoblash topshirig'i berildi. U qanday dastur tuzishi lozim?

>>> a=input().split('+')	a=['478', '874'] satrlardan iborat qiymat.
478+874	
>>> a1 = int(a[0]) >>> a2 = int(a[1])	1-element 478 va 2-element 874 ni satrdan butun qiymatga aylantirib, a1 va a2 ga yuklaydi.
>>> b = a1+a2 >>> print(b, end=';')	478 + 874= yig'indini b ga yuklaydi. b ning oxiriga ";" qo'yib, ekranga chiqaradi.
1352;	

1. input() operatorining qanday usuli mavjud?
2. print() operatorining end='\n' argumenti nima maqsadda ishlatiladi?
3. print() operatorining sep argumenti nima maqsadda ishlatiladi?
4. input() operatorining split() usuli nima maqsadda ishlatiladi?



SAVOL VA
TOPSHIRIQLAR

1. Kutubxonaga ikkita kitob do'konidan kitoblar keltirildi. Birinchi do'kondan n dona kitob keltirildi. Ikkinchi do'kondan esa birinchi do'konga qaraganda k dona ko'p kitob keltirildi. Kutubxonaga jami qancha kitob keltirildi? Masalani yechish dasturini tuzing. n va k kattaliklar foydalanuvchi tomonidan kiritiladi.

2. Komilaga $854*89657*4587*425$ amalini dasturga kiritib, hisoblash topshirig'i berildi. U qanday dastur tuzishi lozim?



UY VAZIFASI

39-DARS. AMALIY MASHG'ULOT

Misol. k soni berilgan. Shu son xonalari sonining yig'indisini hisoblash dasturini tuzing ($0 < k < 9999$).

```
n = input("Enter number: k(0<k<9999) ")
n = int(n)
d1 = n % 10
d2 = n % 100 // 10
d3 = n % 1000 // 100
d4 = n // 1000
print("Result:", d1 + d2 + d3 + d4)
```

29

```
'8795'
8795
d1=8795%10=5
d2=8795%100//10=9
d3=8795%1000//100=7
d4=8795//1000=8
5+9+7+8=29
```




1. Bog'dagi olma daraxtidan Anvar a dona, Dilshod b dona va Mahmud c dona olma terdi. Bolalar tergan olmalarini o'zaro teng taqsimlashsa, har biriga qanchadan olma to'g'ri keladi? Qancha olma ortib qoladi? Har bir bolaning tergan olmasi soni foydalanuvchi tomonidan kiritiladi. Masala dasturini tuzing.

2. Bog'bon o'z bog'idan $n(23856)$ kg hosil oldi. Berilganlar asosida quyidagi topshiriqlar uchun dastur tuzing:

1) bog'bon hosilini tonna, sentner va kilogrammlarda ifodalang;

2) agar har bir yashikga 25 kg dan uzum solinsa, jami qancha yashikga uzum solingan bo'ladi?

3. Yuqoridagi masalada hosilning miqdori foydalanuvchi tomonidan kiritilgan hol uchun dastur tuzing.

4. `split()` usulidan foydalanib, "5746+4186+8426+8266" satrini dasturga kiriting va berilgan ifodani hisoblash dasturini tuzing.

5. F kuch ta'sirida jism a tezlanish oldi. Agar F va a ning qiymatlari quyidagicha bo'lsa, jism massasini hisoblash dasturini tuzing ($m=F/a$):

a) $F = 25, a = 45$;

b) $F = 12, a = 30$;

d) $F = 72, a = 90$;

e) $F = 150, a = 15$.

6. Quyidagi ifodalarni Python tilida yozib, ular natijasini hisoblash dasturini tuzing, bu yerda: $a = 14, b = 8, c = 452, r = 41$.

a) $S = a + b + ac$;

b) $P = \pi r^2 + ac$.

40-DARS. PYTHONDA SODDA MASALALARNI DASTURLASH

Dasturlash jarayonida asosan uch xil: *chiziqli*, *tarmoqlanuvchi* va *takrorlanuvchi* algoritmlardan foydalaniladi.

Bularning orasidan chiziqli algoritmlar, asosan, sodda masalalarni yechish jarayonida keng qo'llaniladi.



1. Chiziqli algoritmgga qanday misollarni keltira olasiz?

2. Chiziqli dastur deganda nimani tushunasiz?

3. Ma'lumot va qiymatlarni dasturga kiritish uchun qaysi operatoridan foydalaniladi?

Sodda masalalarni dasturlash tartibi

1. Asosiy ma'lumotlar va ularning turini aniqlash. O'zgaruvchilar uchun nom tanlash.

2. Natija qanday va qaysi turga mansub bo'lishini aniqlash. Natijani akslantiruvchi o'zgaruvchilar nomini tanlash.

Tayanch tushunchalar

Chiziqli algoritm – buyruqlarning qat'iy ketma-ketlikda tartib bilan bajarilishi.

3. Ma'lumotlarni kiritish, hisoblash va natijani ekranga chiqarish kabi qadamlardan iborat algoritmnı tuzish.

4. Tuzilgan algoritmgaga turli qiymatlar berib, uning to'g'riligini tekshirish.

Amallarning qat'iy ketma-ketlikda bajarilishi *chiziqli ijro* deb ataladi.

Chiziqli algoritm, shuningdek, amallarning shartsiz va takrorlanishsiz ketma-ket bajarilishining ifodasidir.

Chiziqli algoritmlarni dastur shaklida yozilishiga **chiziqli dastur** deyiladi.

Masala. Sinf xonasining eni 10 metr, bo'yi 12 metr. Xonaning yuzini topish dasturini tuzing. Eni va bo'yi qiymatlari foydalanuvchi tomonidan kiritilsin.

Kiruvchi ma'lumot	Hisoblash	Chiquvchi ma'lumot
10 12	$S=a*b=10*12=120$	120

Nº	Blok-sxema	Blok-sxema nomi	Dastur kodi
1		Algoritmning boshlanishi	
2		Kiritish bloki	<code>a=int(input())</code> <code>b=int(input())</code>
3		Bajarish bloki	<code>s=a*b</code>
4		Chiqarish bloki	<code>print(s)</code>
5		Algoritmning tugashi	

1. Chiziqli dastur nima?
2. Sodda masalalarni yechish necha bosqichda amalga oshiriladi?
3. Qaysi bosqich sodda masalalarni yechishning asosiy bosqichi hisoblanadi?



1. Trapetsiyaning ikkita asosi (a va b) hamda asosiga tushirilgan balandligi h berilgan. Trapetsiyaning yuzi s ni hisoblash dasturini tuzing. a , b va h foydalanuvchi tomonidan kiritiladi.

2. Teng tomonli uchburchakning tomoni a ga teng. Uning yuzi s ni hisoblash dasturini tuzing.

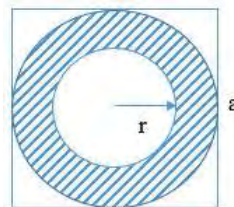
3. Kvadratning tomoni n ga teng. Uning yuzini hisoblash dasturini tuzing.

4. `split()` usulidan foydalanib, "5489*245*58*69*142*4587*54" satrini dasturga kiriting va berilgan ifodani hisoblang.



41-DARS. AMALIY MASHG'ULOT

Masala. Tomoni a ga teng kvadrat ichiga ikkita aylana chizilgan. Kichik aylananing radiusi r ga teng. Bo'yalgan yuza s ni topish dasturini tuzing.



No	Blok-sxema	Blok-sxema nomi	Dastur kodi
1	Boshlash	Algoritmning boshlanishi	
2	a, r	Kiritish bloki	$a = \text{int}(\text{input}())$ $r = \text{int}(\text{input}())$
3	$r_1 = a/2$ $s_1 = \text{PI} * r_1^{**2}$ $s_2 = \text{PI} * r^{**2}$ $s = s_1 - s_2$	Bajarish bloki	$r_1 = a/2$ $s_1 = \text{PI} * r_1^{**2}$ $s_2 = \text{PI} * r^{**2}$ $s = s_1 - s_2$
4	s	Chiqarish bloki	$\text{print}(s)$
5	Tamom	Algoritmning tugashi	



1. Tomoni a ga teng kvadrat ichiga doira chizilgan. Bo'yalgan soha yuzi s ni topish dasturini tuzing.

2. Foydalanuvchi tomonidan kiritilgan sondan avval va keyin keluvchi sonni ekranga chiqaruvchi dastur tuzing.

3. n nafar o'quvchi k dona olma terdi va olmalar ular o'rtasida teng taqsimlandi. Qoldiq olmalar savatchaga solindi.

a) Har bir o'quvchi qanchadan olma olgan?

b) Savatchaga qancha olma solingan?

d) n va k kattaliklar foydalanuvchi tomonidan kiritiladigan hol uchun dastur tuzing.

4. Foydalanuvchi tomonidan kiritilgan ikki xonalidan katta natural sonning oxirgi ikki raqamini topish dasturini tuzing.

5. Avtobus bir kunda n kilometr yo'l yuradi. m kilometr masofani bosib o'tishi uchun avtobus necha kun yurishi kerak? n va m kattaliklar foydalanuvchi tomonidan kiritiladi. Masalani yechish dasturini tuzing.

6. Chumolining bosib o'tgan yo'li millimetrlarda berilgan. Uni metr, santimetr va millimetrlarda ifodalang (masalan, $45\ 786 = 45$ metr 78 cm 6 mm).

7. Kubning tomoni a ga teng. Kubning hajmini topish dasturini tuzing.

8. a, b, c, d sonlari berilgan. Ularning o'rta arifmetigini topish dasturini tuzing.

9. Maktab ma'muriyati 3 ta sinf uchun yangi matematika xonasi tashkil etishga qaror qildi. Mashg'ulotlar bir vaqtda o'tkazilganligi sababli, har bir xonaga alohida parta sotib olinishi kerak. Bitta partada ikkitadan ortiq o'quvchi o'tira olmaydi. Har bir sinf o'quvchilari soni ma'lum bo'lsa, qancha parta sotib olish kerak? Foydalanuvchi tomonidan uchta qiymat – uchta sinf o'quvchilari soni kiritiladi.



42–43-DARSLAR. PYTHONDA MANTIQIY MASALALARNI DASTURLASH

Odatda, dastur tuzish jarayonida qaror qabul qilish uchun o'zgaruvchi, son va satrlarni taqqoslash, uning natijasiga ko'ra, keyingi qadamga o'tish kerak bo'ladi.

O'zgaruvchilarni taqqoslash uchun taqqoslash va mantiqiy amallardan foydalaniladi. Bu amallar ikkita *operand* qabul qiladi va natija sifatida *boolean* turidagi mantiqiy qiymatni qaytaradi: True (ifoda rost) yoki False (ifoda yolg'on).

Tayanch tushunchalar

Taqqoslash amallari – dastur mantiq'ini boshqarish va ikki yoki undan ortiq o'zgaruvchilarni taqqoslab, xulosa qabul qilish amallari.

1. Mantiqiy masalalarni yechishda qaysi amallardan foydalaniladi?
2. Taqqoslash amallarini bilasizmi?
3. Taqqoslash amallari natija sifatida qanday qiymatni qaytaradi?



BUNI BILASIZMI?

Taqqoslash amallari

Taqqoslash amallari ikkita qiymatni bir-biri bilan solishtirish uchun ishlatiladi. Taqqoslash amallari yordamida sodda shartlarni tuzish mumkin.

Amal	Tavsifi	Izoh
==	Aynan teng	Ikki operand teng bo'lsa True, aks holda False qiymatni qaytaradi.
!=	Teng emas	Ikki operand teng bo'lmasa True, aks holda False qiymatni qaytaradi.
<	Kichik	Agar birinchi operand ikkinchisidan kichik bo'lsa True, aks holda False qiymatni qaytaradi.
>	Katta	Agar birinchi operand ikkinchisidan katta bo'lsa True, aks holda False qiymatni qaytaradi.
<=	Kichik yoki teng	Agar birinchi operand ikkinchisidan kichik yoki teng bo'lsa True, aks holda False qiymatni qaytaradi.
>=	Katta yoki teng	Agar birinchi operand ikkinchisidan katta yoki teng bo'lsa True, aks holda False qiymatni qaytaradi.

Misol. Kutubxonaga 20 ta kitob keltirildi.

```
>>> num_book = 20
>>> num_book == 1
False
>>> num_book != 1
True
>>> num_book < 1
False
>>> num_book > 1
True
>>> num_book <= 10
False
>>> num_book >= 1
True
```

Eslab qoling!

Satr ichida qism satr yoki belgi mavjudligini aniqlash uchun **in** amalidan foydalaniladi.

```
>>> 'book' in 'Book shop'
True
>>> 'book' in 'Book shop'
False
```

Mantiqiy amallar

Dastur tuzish jarayonida murakkab shartli ifodalarni yozish uchun mantiqiy amallardan foydalaniladi.

Mantiqiy amallar dasturda buyruqlar bajarilish tartibini boshqarish imkoniyatini beradi hamda tarmoqlanuvchi va takrorlanuvchi operatorlar bilan birga qo'llaniladi.

Amal	Tavsifi	Izoh
and	Mantiqiy ko'paytirish	Murakkab ifodada qatnashgan barcha qism ifodalar True bo'lsa, ifodaning yakuniy qiymati True, aks holda False qiymatni qaytaradi.
or	Mantiqiy qo'shish	Murakkab ifodada qatnashgan barcha qism ifodalardan kamida bittasi True bo'lsa, ifodaning yakuniy qiymati True, aks holda False qiymatni qaytaradi.
not	Mantiqiy inkor	Ifodaning qiymati True bo'lsa, False qiymatni qaytaradi va aksincha.

Misol: Lolaning tug'ilgan kuni – 15 mart. Ushbu dastur mantiqiy amallar yordamida tug'ilgan kuni yoki aksincha ekanligini tekshiradi.

```
>>> day = 15
>>> month = 3
>>> day == 15 and month == 3
True
>>> day = 15
>>> month = 3
>>> not (day == 15 and month == 3)
False
```

and amali ikkala shart ham rost ekanligini tekshiradi.

not amali natijani uning inkoriga almashtiradi. Tug'ilgan kundan boshqa barcha kunlar uchun natija – False.

```
>>> day=15
>>> month=3
>>> (day==15 and month==3) or (day==1 and month==1)
True
```

or amali qavs ichidagi shartlardan kamida bittasi **True** bo'lsa, **True**ni qaytaradi.

Taqqoslash amallari yordamida satrlarni ham taqqoslash mumkin.

```
>>> name='Book shop'
>>> name == 'Book shop'
True
```

Satr to'liq mos kelganligi uchun rost qiymatni qaytaradi.

```
>>> name == 'book shop'
False
```

Birinchi harfi kichik harfda yozilgani uchun yolg'on qiymatni qaytaradi.

```
>>> name == 'Book shop '
False
```

Satr oxirida ortiqcha probellar qo'yilgani uchun yolg'on qiymatni qaytaradi.

1. Dastur tuzish jarayonida qarorlar nimaga asoslanib qabul qilinadi?
2. Taqqoslash amallariga qaysi amallar kiradi?
3. Mantiqiy amallar nima maqsadda qo'llaniladi?
4. Sodda va murakkab shartlar bir-biridan nimasi bilan farq qiladi?



1. Quyidagi mantiqiy ifodalar natijasini yozing:

- 1) $(3 > 5) \text{ and } (2 > 4)$
- 2) $(2 < 5) \text{ and } (3 > 0)$
- 3) $(4 > 2) \text{ or } (4 < 1)$
- 4) $(3 > 1) \text{ or } (5 > 0)$

2. Quyidagi mantiqiy ifodalar natijasini toping:

a)

```
>>> a = 20
>>> b = 28
>>> a > 17 and b = 28
```

b)

```
>>> a = 20
>>> b = 28
>>> c = True
>>> a > 17 and b > 22 and c
```



44-DARS. AMALIY MASHG'ULOT

Masala. Mantiqiy ifoda qiymatini toping: $a=10$; $b=a+4$; $a+3 \geq b-5$

```
>>> a=10
>>> b=a+4
>>> a+3>=b-5
True
```



1. Mantiqiy ifodalar qiymatini aniqlash uchun dastur tuzing va natijasini oling:

- 1) $a=8; b=a^{**}3; a < b/3$
- 2) $a=10; b=a*3; a <= b/3$
- 3) $a=8; b=a; a+b=2*b$
- 4) $a=8; b=a-4; a+3 >= b-2$

2. Mantiqiy ifodalar qiymatini aniqlang:

- 1) $(1 > 3)$ or $(4 < 0)$
- 2) not $(5 > 6)$
- 3) not $(6 > 5)$
- 4) $(2 = 0)$ or $(2 <> 2)$
- 5) $(2 = 0)$ and $(2 > 0)$
- 6) $(3 > 0)$ or $(2 > 0)$

3. Mantiqiy ifodalar natijasini toping:

1)

```
>>> a = 20
>>> c = False
>>> a > 17 or c
```

2)

```
>>> a = 20
>>> c = False
>>> not a > 17 or not c
```

3)

```
>>> a = True
>>> b = True
>>> c = False
>>> not(a and c) and (a or b) or c
```

4)

```
>>> a = 66
>>> b = 22
>>> c = 7
>>> not((a > b) or (b < c))
```

4. Mantiqiy ifodalar qiymatini aniqlash uchun dastur tuzing va natijasini oling:

- 1) $a <= 5$ or $a >= 0$ and $a < 3$
- 2) $x^{**}2+y > 0$ and $a=0.1$ or $(b > 3.7$ and $c != 4)$
- 3) $a < 1$ or $a > 0$ or not $x*x+x*x <= 1$
- 4) not(not(not($a > b$) or True) and False)

45-DARS. TARMOQLANUVCHI ALGORITMLARNI DASTURLASH. IF...ELSE OPERATORI

Mantiqiy ifodaning natijasiga asoslanib, dasturning qaysi qismi bajarilishi lozimligi haqida qaror qabul qilinadi. Bunday masalalar tarmoqlanuvchi algoritmlar yordamida hal etiladi.



1. Tarmoqlanuvchi algoritm deb nimaga aytiladi?
2. Shart qanday tekshiriladi?
3. Taqqoslash amallari natija sifatida qanday qiymatni qaytaradi?

Tarmoqlanuvchi algoritmlar – birorta shartga ko'ra buyruqlar ketma-ketligining bajarilishi yoki bajarilmasligini belgilovchi algoritm. Tarmoqlanuvchi algoritmlarda bir yoki bir necha shartlar tekshiriladi hamda rost yoki yolg'on qiymat qaytarishiga asoslanib, buyruqlar ketma-ketligi bajariladi.

Shartlarni tekshirish uchun barcha dasturlash tillari kabi Python dasturlash tilida ham shartli o'tish operatorlari mavjud.

if shartli o'tish operatori

Sintaksisi:

if shart:

buyruqlar_bloki

Tayanch tushunchalar

if operatori – faqat berilgan shart

rost bo'lgandagina muayyan

buyruqlar to'plamini bajaradigan

shartli operator.

if operatori tarkibidagi shart **True** (rost) qiymat qaytarsa, **buyruqlar_bloki** bajariladi. Agar yolg'on qiymat qaytarsa, **buyruqlar_bloki** bajarilmaydi.

buyruqlar_bloki **if** operatoridan keyingi satrda xat boshidan 4 ta probel qoldirib, keyin yoziladi.

buyruqlar_blokidagi buyruqlar alohida qatorda yoki bitta qatorda nuqta, vergul bilan ajratilgan holda yozilishi mumkin.

Nº	Blok-sxema	Blok-sxema nomi	Dastur kodi
1	Boshlash	Algoritmning boshlanishi	age=int(input('Enter your age?'))
2	age	Kiritish bloki	
3	age>18	Shart tekshirish bloki	if age>18:
4	msg='You can enter!'	Bajarish bloki	msg='You can enter!'
5	msg	Chiqarish bloki	print(msg)
6	Tamom	Algoritmning tugashi	

if-else shartli o'tish operatori

Sintaksisi:

if shart:

buyruqlar_bloki

else:

buyruqlar_bloki2

if operatori bilan birga **else** buyrug'ini ham qo'llash mumkin. Agar shart **True** (rost) qiymat qaytarsa, **buyruqlar_bloki** bajariladi, aks holda **buyruqlar_bloki2** bajariladi.

Misol: $y = \begin{cases} x - 3, & \text{agar } x > 6 \\ x, & \text{agar } x \leq 6 \end{cases}$ tenglamalar sistemasi dasturini tuzing.

```
x = int(input())
if x > 6:
    y = x-3
else:
    y=x
print(y)
```

8
5



1. Shartni tekshirish uchun qaysi operatoridan foydalaniladi?
2. Shartli o'tish operatorining umumiy ko'rinishi qanday?
3. Dastur tuzish jarayonida shart tekshirish nima uchun kerak?
4. Tarmoqlanish operatorining qisqa va to'liq ko'rinishi qanday?



1. Kiritilgan a soni musbat yoki manfiy ekanligini aniqlovchi dastur tuzing.
2. Kiritilgan a soni toq yoki juft ekanligini aniqlovchi dastur tuzing.
3. Tomonlari a va b ga teng to'g'ri to'rtburchak kvadrat ekanligini aniqlovchi dastur tuzing.
4. Kiritilgan a soni to'rt xonali son ekanligini aniqlovchi dastur tuzing.

46-DARS. AMALIY MASHG'ULOT

Masala. Kiritilgan son musbat bo'lsa, unga 2 ni qo'shuvchi, aks holda 2 ni ayiruvchi dastur tuzing va natijani chiqaring.

Nº	Blok-sxema	Blok-sxema nomi	Dastur kodi
1		Algoritmning boshlanishi	
2		Kiritish bloki	<code>a=int(input ())</code>
3		Shart tekshirish bloki	<code>if a > 0:</code>
4		Shart bajarilsa	<code>s = a + 2</code>
5		Shart bajarilmasa	<code>else:</code> <code>s = a - 2</code>
6		Chiqarish bloki	<code>print(s)</code>
7		Algoritmning tugashi	



1. Ikkita a va b son berilgan. Agar b son a sonidan kichik bo'lsa, u holda b ni nol bilan almashtiruvchi, aks holda b ni o'zgarishsiz qoldiruvchi dastur tuzing.
2. Berilgan a butun son noldan farqli b butun songa qoldiqsiz bo'linishi yoki bo'linmasligini aniqlovchi dastur tuzing.
3. Berilgan a butun son noldan farqli b butun songa qoldiqsiz bo'linsa, bu ikkala sonning yig'indisini, aks holda ko'paytmasini topish dasturini tuzing.
4. Uchta a , b va c son berilgan. Agar $a^2 - b^2 = c^2$ shart bajarilsa, ularning ko'paytmasini, aks holda yig'indisini hisoblovchi dastur tuzing.
5. Butun son berilgan. Agar u musbat bo'lsa, unga 1 sonini qo'shish, aks holda o'zini chiqarish dasturini tuzing.
6. Kiritilgan butun son musbat bo'lsa, uning 10 ga ko'paytmasini, aks holda o'zini chiqaruvchi dastur tuzing.

47-DARS. TARMOQLANUVCHI ALGORITMLARNI DASTURLASH. ELIF OPERATORI

Qo'yilgan masalani yechish jarayonida har doim ham bitta shart tekshirilmaydi. Shunday masalalar ham borki, ular yechimini topish uchun bir necha shart tekshirilishi kerak bo'ladi.

1. Shartli o'tish operatorining qanday ko'rinishlari mavjud?
2. Shartli o'tish operatorlari qanday ishlaydi?



BUNI BILASIZMI?

elif shartli o'tish operatori

Boshqa dasturlash tillarida bir necha shart tekshirilishi zarur bo'lgan hollar uchun CASE tanlash operatori ishlatiladi. Pythonda **CASE** tanlash operatori mavjud emasligi sababli, ko'p shartli masalalarni yechish uchun **elif operatori**dan foydalaniladi. **elif** – **else** va **if** so'zlarining kombinatsiyasi bo'lib, "aks holda agar" ma'nosini anglatadi.

Tayanch tushunchalar

CASE operatori – shartdan kelib chiqib, har bir shartga mos buyruqlar ketma-ketligini bajaradigan if ning takomillashgan ko'rinishi.

Sintaksisi:

if shart:

buyruqlar_bloki

elif shart1:

buyruqlar_bloki1

....

else:

buyruqlar_bloki2

shart True (rost) bo'lsa, **buyruqlar_bloki** bajariladi, aks holda ikkinchi shart – **shart1** tekshiriladi. **shart1** True (rost) bo'lsa, **buyruqlar_bloki1**, aks holda **buyruqlar_bloki2** bajariladi.

Misol: Kichik kalkulyator dasturini tuzing

```
a=int(input('a='))
b=int(input('b='))
amal=input('add/sub/mul/div:')
if amal=='add':
    c=a+b
elif amal=='sub':
    c=a-b
elif amal=='mul':
    c=a*b
elif amal=='div':
    c=a/b
else:
    c='Error'
print('Result = ', c)
```

```
a=8
b=4
add/sub/mul/div:add
Result = 12
a=72
b=8
add/sub/mul/div:div
Result = 64
a=2
b=4
add/sub/mul/div:deg
Result = Error
```

Ichma-ich joylashgan if shartli o'tish operatori

if shartli o'tish operatori tarkibida boshqa **if** shartli o'tish operatori mavjud bo'lishi mumkin. Bunday holatga *ichma-ich joylashgan shartli o'tish operatori* deyiladi. Ichki **if**ni ifodalash uchun tashqaridagiga nisbatan bitta xat boshi (4 ta probel) tashlab yozilishi shart, aks holda ifoda ichma-ich joylashmagan, alohida shart operatori hosil qilingan hisoblanadi.

Sintaksisi:

if shart:

buyruqlar_bloki

if shart1:

buyruqlar_bloki1

.....

else:

buyruqlar_bloki2

shart True (rost) bo'lsa, **buyruqlar_bloki** bajariladi va ikkinchi shart **shart1** tekshiriladi.

shart1 True (rost) bo'lsa, **buyruqlar_bloki1** bajariladi.

shart False (yolg'on) bo'lsa, **buyruqlar_bloki2** bajariladi.

Misol. Imtixon natijasini chiqarish dasturini tuzing.

```
result= int(input('Natijani kiriting(0-5 bahoda):'))
if result>=3:
    print('Imtihondan o`tdingiz!')
    if result>=5:
        print('Eng yuqori baho!')
```

```
Natijani kiriting(0-5 bahoda):5
Imtihondan o`tdingiz!
Eng yuqori baho!
```



1. Masalada faqatgina bitta shart tekshirilsa, qaysi operatoridan foydalaniladi?
2. Masalada bir necha shart tekshirilishi lozim bo'lsa-chi?
3. Ichma-ich joylashgan shartlarda ikkala if operatori bir chiziqda joylashsa, dastur to'g'ri ishlaydimi?
4. Tarmoqlanish operatorida buyruqlar ketma-ket ishtirok etsa, ular qanday xizmatchi so'zlar orasida yoziladi?



1. Ikki xonali sonning raqamlari toq ekanligini aniqlovchi dastur tuzing.
2. Uch xonali sonda bir xil raqamlar mavjudligini aniqlovchi dastur tuzing.
3. a va b sonidan qaysi biri juft ekanligini aniqlovchi dastur tuzing.
4. 3 ta butun son kiritildi. Qaysi biri juft ekanligini aniqlovchi dastur tuzing.

48-DARS. AMALIY MASHG'ULOT

Masala. Kiritilgan uchta sondan qanchasi musbat, qanchasi musbat emasligini aniqlovchi dastur tuzing

№	Blok-sxema	Blok-sxema nomi	Dastur kodi
1		Algoritmning boshlanishi	a=int(input ()) b=int(input ()) c=int(input ())
2		Kiritish bloki	k=0 d=0
3		Shart tekshirish bloki	if a>0: k+=1 else: d+=1
4		Shart bajarilsa	if b>0: k+=1 else: d+=1
5		Shart bajarilmasa	if c>0: k+=1 else: d+=1
6		Chiqarish bloki	print(k,d)
7		Algoritmning tugashi	

- Uchta a , b va c butun son berilgan. Ular orasidan musbatlari sonini topish dasturini tuzing.
- Uchta a , b va c butun son berilgan. Ular orasidan musbat va manfiylari sonini topish dasturini tuzing.
- Ikkita a va b son berilgan. Dastlab kattasini, keyin esa kichigini chiqaruvchi dastur tuzing.
- Uchta a , b va c butun son berilgan. Ulardan faqat musbatlari kvadratini hisoblab chiquvchi dastur tuzing.
- Kvadrat tenglamani yechish dasturini tuzing.
- Hafta raqami 1–7 oraliqda kiritilganda, hafta kunini ekranga chiqaruvchi dastur tuzing.
- a va b sonlar berilgan. Agar ular musbat va yig'indisi 100 dan katta bo'lsa, a sonining b songa nisbatini, ular musbat va yig'indisi 100 dan katta bo'lmasa, a ning b ga ko'paytmasini hisoblash dasturini tuzing.
- Ikkita son berilgan. Agar birinchi son ikkinchisidan katta bo'lsa, 1 ni, agar ikkinchi son birinchisidan katta bo'lsa, 2 ni, agar ikkalasi teng bo'lsa, 0 ni chiqaruvchi dastur tuzing.

49–50-DARSLAR. TAKRORLANUVCHI ALGORITMLARNI DASTURLASH. FOR OPERATORI

Berilgan masalani yechishda aynan bir xil amalni bir necha marta takrorlashga to'g'ri keladi. Bunday masalalarni yechishda takrorlanuvchi algoritmlardan foydalaniladi.



1. Bir necha marta takrorlanadigan kod qanday yoziladi?
2. Takrorlanuvchi algoritmlar qanday ishlaydi?
3. Sikllar nima uchun kerak?

Takrorlanuvchi algoritmlar – biror buyruqlar guruhining ma'lum marta yoki belgilangan shart bajarilgunga qadar takroran bajarilishi. Takrorlanuvchi algoritmlarga doir masalalarni dasturlashda sikl operatorlaridan foydalaniladi.

Masalan, n ta sonning musbatligini tekshirish uchun n marotaba bir xil amalni bajarish kerak bo'ladi. Bunday hollarda bitta amalni n marta yozishdan ko'ra, bitta kod blokida n ta sonni tekshirish uchun sikl operatorlaridan foydalangan afzal. Sikl operatorlari kodning takrorlanadigan buyruqlari uchun xizmat qiladi. Bu buyruqlarning ketma-ketligiga *siklning tanasi* deyiladi. Har bir takrorlanish esa *iteratsiya* deb ataladi.

Sikl operatorlari turlari

Sikl operatorlari nomi	Tavsifi	Vazifasi
for	Kodni muayyan marta takrorlaydi.	Takrorlanishlar soni avvaldan ma'lum bo'lganda qo'llaniladi.
while	Asosiy shart bajarilganda kodni takrorlaydi.	Takrorlanishlar soni noma'lum bo'lganda, kod hatto bir marta ham ishga tushmasligi mumkin. Kodni ishga tushirishdan avval shart tekshiriladi. Agar shart noto'g'ri bo'lsa, unda sikldagi kod ishga tushmaydi.

Qo'yilgan masalani yechishda sikllarning har ikkala turidan foydalanish mumkin, lekin berilgan shart uchun eng mos keladigan turni to'g'ri tanlay olish dasturning samaraliroq ishlashini ta'minlaydi.

for sikl operatori

Sintaksisi:

for i in range(start, stop, step):
sikl tanasi

Tayanch tushunchalar

for sikli – hisoblagichli (count-controlled) sikl. U faqat takrorlanishlar soni oldindan ma'lum bo'lganda foydalaniladi.

i – takrorlanishlar (iteratsiyalar) soni;
start – i ning boshlang'ich qiymati (ko'rsatilmasa, 0 deb qabul qiladi);
stop – i ning oxirgi qiymati (ko'rsatilishi shart);
step – qadam (ko'rsatilmasa, 1 deb qabul qiladi);

Misol. 0 dan 11 gacha bo'lgan juft sonlarni chiqarish dasturini tuzing.

```
for i in range(0,11,2):  
    print(i, end=';')
```

```
0;2;4;6;8;10
```

0 dan 11 gacha bo'lgan sonlarni
2 qadam bilan chiqaradi.

Misol. 10 gacha bo'lgan sonlarni chiqarish dasturini tuzing.

```
for i in range(10):  
    print(i, end=';')
```

```
0;1;2;3;4;5;6;7;8;9
```

0 dan 10 gacha bo'lgan sonlarni
1 qadam bilan chiqaradi.

Misol. 10 dan 1 gacha bo'lgan sonlarni –1 qadam bilan chiqarish dasturini tuzing.

```
for i in range(10,0,-1):  
    print(i, end=';')
```

```
10;9;8;7;6;5;4;3;2;1
```

10 dan 0 gacha bo'lgan sonlarni
–1 qadam bilan chiqaradi.

Ichma-ich joylashgan sikllar

Sikl ichida yana siklning ishlatilishiga ichma-ich joylashgan sikl deyiladi.

Sintaksisi:

```
for i in range(start1, stop1, step1):  
    for j in range(start2, stop2, step2):  
        sikl tanasi
```

i – 1-sikl takrorlanishlari soni;

j – 2-sikl takrorlanishlari soni;

start1 – i ning boshlang'ich qiymati (ko'rsatilmasa, 0 deb qabul qiladi);

stop1 – i ning oxirgi qiymati (ko'rsatilishi shart);

step1 – i ning qadami (ko'rsatilmasa, 1 deb qabul qiladi);

start2 – j ning boshlang'ich qiymati (ko'rsatilmasa, 0 deb qabul qiladi);

stop2 – j ning oxirgi qiymati (ko'rsatilishi shart);

step2 – j ning qadami (ko'rsatilmasa, 1 deb qabul qiladi).

Tashqi siklning har bir iteratsiyasi uchun j ta iteratsiya bajariladi. Tashqi siklning i ta iteratsiyasi uchun ichki siklning i*j ta iteratsiyasi bajariladi.

Misol.

```
n = 3
for i in range(1,n+1):
    for j in range(1,n+1):
        print(i,'x', j,'=', i*j)
```

```
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
```

i va j uchun 3 qadamlar soni kiritiladi.

0 ga ko'paytirishni chiqarmasligi uchun 1 dan boshlanadi, u holda qadamlar soni $n+1$ deb olinadi. Natija ekranga chiqariladi.

Tashqi siklning 1-takrorlanishida 3 marta ichki sikl bajarilib, natijani chiqardi.

Tashqi siklning 2-takrorlanishi.

Tashqi siklning 3-takrorlanishi.

SAVOL VA TOPSHIRIQLAR



1. Qaysi operator Python dasturlash tilida hisoblagich ko'rinishida ishlaydi?
2. Hisoblagich ko'rinishida ishlovchi sikl operatorining sintaksisi qanday bo'ladi?
3. Start, stop, step vazifalarini tushuntiring.
4. Qachon hisoblagich ko'rinishidagi sikl operatorlaridan foydalanib bo'lmaydi?
5. Ichma-ich joylashgan sikllarda ikkala for operatori bir chiziqda joylashsa, dastur to'g'ri ishlaydimi?

UY VAZIFASI



1. a va b sonlar berilgan. a dan b gacha bo'lgan barcha sonlarni chiqaruvchi dastur tuzing. Bu yerda $a \leq b$.
2. a va b sonlar berilgan. Agar $a < b$ bo'lsa, a dan b gacha bo'lgan sonlarni o'sish tartibida, aks holda kamayish tartibida chiqaring.
3. 1 dan 10 gacha bo'lgan natural sonlar kvadratlarini chiqaruvchi dastur tuzing.
4. 1 dan 10 gacha bo'lgan natural sonlar yig'indisini chiqaruvchi dastur tuzing.

51-DARS. AMALIY MASHG'ULOT

Misol. x ga karrali n ta sonni chiqarish dasturini tuzing.

Nº	Blok-sxema	Blok-sxema nomi	Dastur kodi
1		Algoritmning boshlanishi	
2		Kiritish bloki	<code>x = int(input ())</code> <code>n = int(input ())</code> <code>i = 0</code>
3		Sikl soni	<code>for i in range(n):</code>
4		Sikl tanasi Bajarish bloki	<code>s=i*x</code>
5		Chiqarish bloki	<code>print(s)</code>
6		Algoritmni tugatish	
	8 ga karrali 6 ta sonni chiqarish uchun n ga 6, x ga 8 qiymat berganimizda, 0 8 16 24 32 40 natijani chiqardi.		<code>x=8</code> <code>n=6</code> 0 8 16 24 32 40

1. Quyidagi dasturiy koddagi iteratsiyalar (takrorlashlar) sonini aniqlang.

1. `for i in range(487):`
`cms='Salom'`

2. `for i in range(2, 89):`
`cms=12`

3. `for i in range(-17, 15, 4):`
`cms='dastur'`

4. `for i in range(-58, 0):`
`cms=4789`

5. `for i in range(73, 161, 2):`
`cms=2020`

6. `for i in range(100, 1982):`
`cms='UZB'`

7. `for i in range(1874, 21):`
`cms='4782'`

8. `for i in range(-21, 21, 5):`
`cms='4782'`

2. 1 dan n gacha bo'lgan natural sonlar kvadratlari yig'indisini aniqlovchi dastur yozing.

3. $n \geq 2$ shartni qanoatlantiruvchi n soni berilgan. Ushbu ifodani hisoblovchi dastur tuzing. $d=1*2+2*3+\dots+(n-1)*n$

4. Bir nechta son berilgan. Ular orasida qancha 2 soni bor ekanligini aniqlovchi dastur tuzing.

5. $S=11+13+15+\dots+49$ yig'indini hisoblash dasturini tuzing.

6. n ta uchburchakni ekranga chiqaruvchi dastur tuzing. $n - 1$ dan 9 gacha bo'lgan natural sonlarni qabul qiladi.

```
*
**
***
****
```

52-DARS. NAZORAT ISHI

1. Butun son kiritilgan vaqtda sondan avval va keyin keluvchi sonni chiqaruvchi dastur tuzing. Dastur natijasi quyidagicha bo'lishi lozim.

Kiruvchi ma'lumot	Chiquvchi ma'lumot
254	254 dan avvalgi son bu – 253
	254 dan keyingi son bu – 255

2. Natural son berilgan. Shu son oxirgi raqamini topish dasturini tuzing.

3. Sutkaning n sekundi berilgan. Sutka boshlanganidan beri qancha soat (h), minut (min) va sekund (s) o'tganini aniqlash dasturini tuzing.

masalan, $n = 13257 = 3 \cdot 3600 + 40 \cdot 60 + 57$; $h = 3$ va $min = 40$.

4. Mahsulotning narxi s so'm. Ushbu mahsulotdan n ta sotib olish uchun qancha so'm sarflash kerak?

5. Musbat butun son berilgan. Ushbu sonning o'nliklar xonasidagi raqamini aniqlash dasturini tuzing.

6. Ismingizni kiritganda quyidagi xabarni chiqaruvchi dasturni tuzing. Dastur tuzishda `print()` operatorining `end()` usulidan foydalaning.

Kiruvchi ma'lumot	Chiquvchi ma'lumot
Lola	Xush kelibsiz, Lola!

53-DARS. TAKRORLANUVCHI ALGORITMLARNI DASTURLASH. WHILE OPERATORI

for sikl operatorini avvaldan takrorlanishlari soni aniq bo'lgan masalalarni yechish uchun qo'llash samara beradi. Lekin har doim ham takrorlanishlar sonini avvaldan bilib bo'lmaydi. Faqatgina shart tekshirish orqali sikl davom etishi yoki to'xtashi kerakligi haqida xulosa qilish uchun **while** sikl operatoridan foydalangan ma'qul.



1. Takrorlanishlar soni aniq bo'lmasa, qaysi sikldan foydalangan ma'qul?
2. **while** sikli qanday ishlaydi?

while sikl operatori

while sikl operatori shart ifodasi bajarilgan holatlar (True bo'lsa) uchun davom etadi, agar shart bajarilmasa (False bo'lsa), sikl o'z ishini to'xtatadi.

Sintaksisi:

while shart ifodasi:
sikl tanasi

shart ifodasi – sikl sharti;

sikl tanasi – bajarilishi kerak bo'lgan buyruqlar.

Tayanch tushunchalar

- while sikli** – berilgan shart rost bo'lgan holda sikl tanasi bajariladigan sikl turi.
- Agar sikl boshida shart bajarilmasa, u holda sikl ishga tushmaydi.

Misol.

```
res='ha'
while answer == 'ha':
    print('Foydalanishingiz mumkin')
    res=input('Ushbu dasturdan
foydalanasizmi? (ha/yo'q)')
print('Marhamat.')
```

Foydalanishingiz mumkin
Ushbu dasturdan foydalanasizmi? (ha/yo'q) **ha**
Foydalanishingiz mumkin
Ushbu dasturdan foydalanasizmi? (ha/yo'q) **ha**
Foydalanishingiz mumkin
Ushbu dasturdan foydalanasizmi? (ha/yo'q) **yo'q**
Marhamat.

Eslab qoling!

IDLE interfaol muhitida **cheksiz siklni to'xtatish** uchun, *Ctrl* tugmachasini bosib turgan holda *C* tugmachasi bosiladi yoki *Ctrl+C* birgalikda bir necha marta bosiladi. IDLE esa dasturni to'xtatish uchun so'rov yuboradi.

Misol. 1 dan n gacha bo'lgan sonlarning ko'paytmasini toping. $P=1*2*...*n=n!$

Nº	Blok-sxema	Blok-sxema nomi	Dastur kodi
1		Algoritmning boshlanishi	
2		Kiritish bloki	<code>n=int(input ())</code>
			<code>i=1</code> <code>p=1</code>
3		Shart tekshirish bloki	<code>while i<=n:</code>
4		Bajarish bloki	<code>P=P*i</code> <code>i+=1</code>
5		Chiqarish bloki	<code>print('1*...*n, '=', p)</code>
6		Algoritmni tugatish	
	<p>n ga 5 qiymat beramiz, i ning boshlang'ich qiymati 1 ga teng. Ko'paytma hisoblanayotganligi uchun p ning boshlang'ich qiymatini 1 deb olamiz. $1*1*2*3*4*5=120$ natijani chiqaradi.</p>		$n=5$ $1*...*5= 120$

Cheksiz sikl

while siklidagi shart ifodasi doimo bajarilsa (True qiymat qaytarsa), sikl hech qachon to'xtamaydi, ya'ni cheksiz davom etishi mumkin.

Cheksiz sikl yaratish juda ham oson bo'lib, shartning o'rniga True kalit so'zi kiritilsa kifoya.

Misol.

```
while True:
    res=input('So\'z kiriting:')
    print('davom eting')
```

So'z kiriting: kitob
davom eting
So'z kiriting:



1. Qaysi operator Python dasturlash tilida shartli sikl ko'rinishida ishlaydi?
2. Shart bo'yicha sikl operatorining hisoblagich ko'rinishidagi sikl operatoridan farqi nimada?
3. Shart bo'yicha sikl operatorining yana qanday turini bilasiz?



1. 0 dan 20 gacha bo'lgan juft sonlarni ekranga chiqaruvchi dastur tuzing.
2. n va k butun manfiy bo'lmagan sonlar berilgan. n va k qatnashgan ushbu ifodani hisoblang. $\frac{n!}{k!(n-k)!}$
3. a va b natural sonlar berilgan. a dan b gacha bo'lgan sonlar orasidan faqat juftlarini chiqaruvchi dastur tuzing. Bu yerda $a \leq b$.
4. n natural soni berilgan. Kvadrati n dan kichik bo'lgan barcha natural sonlarni chiqaruvchi dastur tuzing.

54-DARS. AMALIY MASHG'ULOT

Misol. $S=1*2+3*4+5*6+...+99*100$ ifodani hisoblash dasturini tuzing.

$$S=1*2+3*4+5*6+...+99*100 \Rightarrow \sum_{i=1}^{99} i*(i+1), i=i+2$$

Nº	Blok-sxema	Blok-sxema nomi	Dastur kodi
1		Algoritmning boshlanishi	
2		Kiritish bloki	$i=1$ $s=0$
3		Shart tekshirish bloki	while $i \leq 99$:
4		Bajarish bloki	$s=s+i*(i+1)$ $i+=2$
5		Chiqarish bloki	print(s)
6		Algoritmning tugashi	
<p>i ning boshlang'ich qiymati 1 ga teng. Yig'indi hisoblanayotganligi uchun s ning boshlang'ich qiymatini 0 deb olamiz. $0+1*2+3*4+...+99*100$</p>			169150

1. $S = 0,5 + 1,5 + 2,5 + \dots + 98,5 + 99,5$ ifodani hisoblash dasturini tuzing.
2. n natural sonining barcha bo'luvchilarini chiqaruvchi dastur tuzing.
3. Kiritilgan n soni qancha raqamdan iborat ekanligini aniqlovchi dastur tuzing (ko'rsatma: $n = n \% 10$ ifoda $n = 0$ bo'lguncha bajariladi).
4. Kiritilgan n soni raqamlari yig'indisini hisoblovchi dastur tuzing.
5. Kiritilgan n soni juft raqamlari sonini hisoblovchi dastur tuzing.
6. n natural soni berilgan. 1 dan n gacha bo'lgan natural sonlar ichida oxirgi raqami 3 ga karrali sonlarni chiqaruvchi dastur tuzing.

55-DARS. SIKLLARNI BOSHQARISH: CONTINUE, BREAK OPERATORLARI

Sikllarni boshqarish uchun **break** va **continue** kabi maxsus operatorlardan foydalaniladi. **break** operatori sikldan chiqish uchun ishlatiladi. **continue** operatori siklning navbatdagi iteratsiyasiga o'tish uchun ishlatiladi.

1. Cheksiz siklni to'xtatishning Ctrl+C dan boshqa usuli mavjudmi?
2. Sikl to'xtaganidan keyin uni davom ettirish mumkinmi?

Sikldan chiqish

Agar shart True qiymat qaytarsa ham, sikl ichida **break** operatoriga murojaat etilsa, u holda sikl ishi to'xtatiladi. Sikl tarkibidagi ixtiyoriy buyruq **break**ga murojaat etilganidan keyin bekor qilinadi.

Jayanch tushunchalar

break – sikl ishini to'xtatuvchi operator.

continue – joriy siklni o'tkazib yuborib, keyingisiga o'tuvchi operator.

Misol. Karra jadvali bo'yicha o'quvchilar bilimini tekshiramiz.

```
table=8
for i in range(1,11):
    print(table, 'x', i, '= ?' )
    pup=input()
    res=table*i
    if int(pup)==res:
        print('Barakalla!')
    else:
        print('Noto'gri, javob:', res)
print('Tugadi')
```

```
8 x 1 = ?
5
Noto'gri, javob: 8
```

i 1 dan 11 gacha bo'lgan 10 ta siklni hisoblaydi. Karra jadvali savolini chiqaradi. O'quvchi kiritgan javobni qabul qiladi. Ko'paytmani hisoblaydi. Natijani o'quvchi kiritgan javob bilan taqqoslaydi. To'g'ri bo'lsa, **Barakalla** yozuvini chiqaradi. Aks holda, noto'g'riligi haqida xabar berib, tog'ri javobni chiqaradi.

Sikldan chiqish uchun dastur kodiga **break** operatorini qo'shamiz. O'quvchi **bilmayman** so'zini yozgan vaqtda, sikl ishini to'xtatadi.

```

table=8
for i in range(1,11):
    print(table, 'x', i, '= ?' )
    pup=input()
    if pup=='bilmayman':
        break
    res=table*i
    if int(pup)==res:
        print('Barakalla!')
    else:
        print('Noto\'gri, javob:', res)
print('Tugadi')

```

```

8 x 1 = ?
8
Barakalla!
8 x 2 = ?
16
Barakalla!
8 x 3 = ?
bilmayman
Tugadi
>>>

```

Agar o'quvchi **bilmayman** deb javob bersa, dastur sikldan chiqadi va **Tugadi** yozuvini chiqaradi.

Siklni davom ettirish

continue operatori yordamida, sikldan chiqmagan holda, savolni o'tkazib yuborib, jarayonni davom ettirish mumkin. Agar ushbu operator sikl o'rtasida uchrasa, sikl oxirigacha bo'lgan keyingi buyruqlarni o'tkazib yuboradi va sikl o'z ishini keyingi iteratsiyadan davom ettiradi.

```

table=8
for i in range(1,11):
    print(table, 'x', i, '= ?' )
    pup=input()
    if pup=='bilmayman':
        break
    if pup=='keyingisi':
        print('Keyingi savol')
        continue
    res=table*i
    if int(pup)==res:
        print('Barakalla!')
    else:
        print('Noto\'gri, javob:', res)
print('Tugadi')

```

Agar o'quvchi **"keyingisi"** javobini kiritrsa, dastur siklni keyingisiga o'tkazib yuboradi.

```

8 x 1 = ?
8
Barakalla!
8 x 2 = ?
16
Barakalla!
8 x 3 = ?
keyingisi
Keyingi savol
8 x 4 = ?
32
Barakalla!
8 x 5 = ?

```

O'zgaruvchilarning o'zaro qiymat almashtirishi

Python dasturlash tilida o'zgaruvchilar qiymatini eng kam qadamlar bilan osongina o'zgartirish mumkin.

Masalan,

```
>>> a, b = 0, 1
```

```

a = 0
b = 1

```

Odatda, bu amal ikkita o'zgaruvchi qiymatini bir vaqtda o'zgartirish uchun kerak bo'ladi.

Boshqa dasturlash tilida	Python dasturlash tilida
<pre>a=1 b=2 tt=a a=b b=tt print(a,b)</pre>	<pre>a=1 b=2 a, b=b, a print(a, b)</pre>
21	21

1. Qaysi operatorlar sikl ishini boshqaradi?
2. *break* operatorining vazifasi nima?
3. *continue* operatorining vazifasi nima?
4. O'zgaruvchilarning o'zaro qiymat almashtirishi qanday amalga oshiriladi?



SAVOL VA
TOPSHIRIQLAR

1. Qo'shish, ayirish, ko'paytirish va bo'lish amallaridan iborat sodda kalkulyator dasturini tuzing.
2. Foydalanuvchi tomonidan kiritilgan sonlar yig'indisini hisoblash dasturini tuzing. Agar manfiy son kiritilsa, sikl o'z ishini to'xtatishi lozim.



UY VAZIFASI

56–57-DARSLAR. QISM DASTURLAR: FUNKSIYALAR VA PROTSEDURALAR

Dastur tuzish jarayonida ma'lum bir amallar majmuini dasturning turli qismlarida takrorlashga to'g'ri keladi. Dasturning mana shu amallar majmuini o'z ichiga olgan qismi **qism dastur** deb ataladi. Qism dasturlar ma'lum bir vazifani bajaradi, lekin alohida tizimni tashkil etmaydi.

Qism dasturga murojaat etilganda, unga murojaat etgan asosiy dastur to'xtaydi va boshqaruv qism dasturga o'tadi. Qism dastur bajarilishi tugaganidan so'ng, boshqaruv yana asosiy dasturga qaytadi.

1. Dasturlashda qism dasturlardan qanday foydalaniladi?
2. Qism dasturlar qanday ishlaydi?
3. Funksiya nima va u qanday ishlaydi?
4. Protsedura nima va u qanday ishlaydi?



BUNI BILASIZMI?

Asosiy dasturda qism dasturlarni chaqirish quyidagi imkoniyatlarni beradi:

- Qism dastur zarurat tug'ilganda chaqiriladi. U ayni bir kodni bir necha marta yozish zaruratini bartaraf qilib, butun dastur davomida ko'p marta foydalanilishi mumkin. Bu kodning blokligini oshiradi, tushunishni osonlashtiradi va xatolarni topishda yordam beradi.

- Xato bor yoki yo'qligini bitta kod blokining o'zida tekshirsa bo'ladi. Agar xato qism dasturda bo'lsa, faqat qism dasturning o'zini tuzatishga zarurat tug'iladi. Agar qism dasturdan foydalanmasdan, kod bir necha joyda takror-takror yozilsa, u holda butun dastur bo'ylab xatolarni qidirishga to'g'ri keladi.
- Kodni faqat bitta joyda yangilash kerak bo'ladi: Kiritilgan barcha tuzatishlar qism dastur chaqirilishi bilan amal qila boshlaydi.

Qism dasturning turlari

- **Funksiya** – ma'lum bir vazifani bajaruvchi, qandaydir nomga ega, bir yoki bir necha qiymatni qabul qiluvchi, ishni tugatganidan keyin esa asosiy dasturga bir yoki bir necha natija qiymatlarni qaytaruvchi qism dastur.
- **Protsedura** – funksiyaga o'xshash ko'p marta foydalanilishi mumkin bo'lgan qism dastur bo'lib, yagona farqli jihati hech qanday qiymatni qaytarmaydi.

Python dasturlash tilining har xil masalalarni yechishga mo'ljallangan bir necha foydali standart funksiyalari mavjud.

Standart funksiyalar

print() – foydalanuvchi uchun ma'lumotlarni chiqaradi. Masalan, turli ma'lumotlar va hisoblash natijalarini.

input() – print() funksiyasining zidi, foydalanuvchilar kiritgan ma'lumotlarni dasturga uzatadi.

randint() – tasodifiy sonni chiqaradi. Masalan, dasturda tasodifiy son kerak bo'lib qolganda ishlatiladi.

Keyinchalik standart funksiyalar bilan batafsilroq tanishib chiqamiz.

Funksiyani e'lon qilish va chaqirish

Har bir yaratilgan qism dasturga, xususan, funksiya hamda protseduraga albatta nom berish kerak va bu nom Pythonda define (ing. define – aniqlash) so'zidan olingan def kalit so'zi bilan boshlanadi.

Sintaksisi:

```
def funksiya_nomi ([parametrlar ro'yxati]):
    buyruqlar_bloki
```

def – funksiyaning e'lon qiluvchi kalit so'z.

funksiya_nomi – funksiya nomi.

parametrlar ro'yxati – ushbu ro'yxat bir necha parametrdan iborat bo'lishi mumkin va ular vergul bilan ajratib yoziladi.

buyruqlar_bloki – funksiya tanasi boshqa operatorlar kabi bitta xat boshi tashlab yozilishi shart.

Funksiya nomi orqali chaqirilganda uning tarkibidagi buyruqlar ketma-ketligi bajariladi. Shundan so'ng dastur funksiya chaqirilgan satrga qaytadi va shu satrdan keyingi buyruqlarga o'tadi.

Misol. Xabarni chiqarish.

```
def welcome():
    msg='Xayrli kun! '
    return msg
```

```
print(welcome())
```

```
Xayrli kun!
```

Protsedurani e'lon qilish

```
def msg():
    print('Bugun soat 14.00 da ota-
          onalar majlisi!')
```

```
>>> msg()
Bugun soat 14.00 da ota-onalar majlisi!
```

Funksiyaga qiymat uzatish

Funksiyaga qayta ishlashi uchun qiymatlar berish mumkin.

Misol. Aylananing radiusi kiritilganda, uning uzunligini topish dasturini tuzing.

```
def circle(r):
    PI=3.14
    len=2*PI*r
    return len
radius = int(input('Aylana radiusi: '))
uz= circle(radius)
print('Aylananing uzunligi: ', uz)
```

```
Aylananing radiusi: 4
Aylananing uzunligi: 25.136
```

Misol. n faktorialini hisoblash dasturini tuzing.

$S=1*2*3*...*n=n!$

```
def factor(n):
    res=1
    for i in range(2,n+1):
        res*=i
    return res
n=int(input('n sonini kiriting:'))
print(factor(n))
```

```
n sonini kiriting: 5
120
```

salomlashish nomli funksiya e'lon qilindi.

xabar o'zgaruvchisiga qiymat berish.

Funksiyaning vazifasi – xabar

o'zgaruvchisi qiymatini qaytarish.

Funksiyani chaqirib, ekranga chiqarish.

Pythonda protseduralar deyarli funksiyalardek yoziladi. Farqi shundaki, protseduralar hech qanday qiymatni qaytarmaydi. Quyida protseduraga misol keltirilgan:

aylana nomli funksiya e'lon qilindi,

uning qabul qiluvchi qiymati – r .

Aylananing uzunligi hisoblandi.

Funksiya aylana uzunligini qaytaradi.

Foydalanuvchi tomonidan kiritilgan

radiusni butun qiymatga o'zgartirish.

Aylana nomli funksiya chaqirilmoqda.

Aylana uzunligi chiqarilmoqda.

factor nomli funksiya e'lon qilindi.

Ko'paytmaning birinchi qiymati kiritildi.

Sikl 2 dan boshlab $n+1$ gacha,

ya'ni 1 marta takrorlanadi.

$res=1*2*...*n$

res natijasini qaytaradi.

n ga qiymat beriladi.

n faktorialni hisoblovchi funksiya

chaqirilib, ekranga natija chiqariladi.

Rekursiya

Funksiyaning o'zini o'zi chaqirishiga *rekursiya* deyiladi va bunday funksiyalar **rekursiv funksiyalar** deb ataladi.

Rekursiv funksiyalar dasturlashning kuchli mexanizmi hisoblanadi, lekin ular har doim ham samarali emas. Chunki aksariyat hollarda xatolarga yo'l qo'yadi. Xatolar ichidan eng ko'p tarqalgani – *cheksiz rekursiya*. Unda funksiyaning chaqiruv zanjiri cheksiz bo'lib, kompyuter bo'sh xotirasi tugamaguncha davom etaveradi. Cheksiz rekursiya ro'y berishining sabablari:

– rekursiyada shartni noto'g'ri qo'llash. Masalan, faktorialni hisoblashda if $n==0$ ni unutib qo'ysak, `factorial(0)` funksiyasi `factorial(-1)`ni, `factorial(-1)` funksiyasi esa `factorial(-2)` va hokazolarni chaqiradi;

– rekursiv funksiyani noto'g'ri parametr bilan chaqirish. Masalan, `factorial(n)` funksiya `factorial(n)`ni chaqirsa, yana cheksiz zanjir yuzaga keladi.

Shu boisdan, rekursiv funksiyani yaratishda rekursiyani yakunlash sharti hamda rekursiyani qachon, qanday tugatish haqida o'ylab ko'rish lozim.

```
def factor(n):
    if n==0:
        return 1
    else:
        res= n*factor(n-1)
        return res
n=int(input('n sonini kiriting: '))
print(factor(n))
```

```
n sonini kiriting: 5
120
```

factor nomli rekursiv funksiya e'lon qilindi.

$n==0$ da funksiya 1 ni qaytaradi, aks holda ishini davom ettiradi.

Funksiya o'zi o'zini chaqirib, $res=n*(n-1)*...*3*2*1$ ni hisoblaydi, qachonki `factor(0)` bo'lguncha.

`res` natijasini qaytaradi.

n ga qiymat beriladi.

n faktorialni hisoblovchi funksiya chaqirilib, ekranga natija chiqariladi.



1. Qism dastur nima?
2. Dasturda protsedura va funksiyalar qanday maqsadda qo'llaniladi?
3. Qism dasturning qanday turlari mavjud?
4. Qism dastur qanday afzalliklarga ega?
5. Protседura va funksiyaning farqi nimada?
6. Qachon funksiyaning o'rniga protsedurani qo'llash mumkin?



1. n va k butun musbat sonlar berilgan. n va k qatnashgan ushbu ifodani hisoblang.
$$\frac{n!}{k!(n-k)!}$$
Funksiyadan foydalaning.
2. n natural son berilgan. Kvadrati n dan kichik bo'lgan barcha natural sonlarni chiqaruvchi dastur tuzing.
3. Bir birlik uzunlik '-' ga teng. Berilgan n uzunlikdagi '-' belgidan iborat chiziq chizuvchi dastur tuzing. Protседuradan foydalaning.

Kiruvchi ma'lumot	Chiquvchi ma'lumot
n	n ta chiziq ('-')
5	-----

58-DARS. AMALIY MASHG'ULOT

Misol. Berilgan uchta a , b va c sonlardan eng kattasini topish dasturini tuzing. Funksiyadan foydalaning.

<pre>def max(a, b): if a > b: return a else: return b def max3(a, b, c): return max(max(a, b), c) a = int(input()) b = int(input()) c = int(input()) print(max3(a,b,c))</pre>	<p>a, b parametrlarni qabul qiluvchi max nomli funksiya e'lon qilindi.</p>
	Agar $a > b$ bo'lsa, funksiya a ni qaytaradi.
	Aks holda, funksiya b ni qaytaradi.
	a , b va c parametrlarni qabul qiluvchi max3 nomli funksiya e'lon qilindi.
	max3 nomli funksiya max funksiyasini $\max(a, b \text{ va } c)$ parametrlar bilan qaytarmoqda.
	a va b taqqoslanadi va eng kattasi topiladi, masalan, a , keyin bu eng katta a soni c bilan yana taqqoslanadi va funksiya ulardan kattasini qaytaradi.
	a , b va c ga qiymat beriladi.
	3 ta sonning kattasini topuvchi funksiya chaqirilib, ekranga natija chiqariladi.
<pre>89 78 58 89</pre>	

1. '*' belgisidan tomoni n ga teng kvadrat chizuvchi dastur tuzing. Protseduradan foydalaning.



Kiruvchi ma'lumot	Chiquvchi ma'lumot
n	$n \times n$ ta (*) dan iborat kvadrat
3	*** *** ***

2. Berilgan n sonining bo'luvchilarini bitta qatorda probel orqali ajratib chiqaruvchi dastur tuzing. Protseduradan foydalaning.

Kiruvchi ma'lumot	Chiquvchi ma'lumot
6	1 2 3 6

3. Berilgan n sonini rim raqamlarida ifodalovchi dastur tuzing. Protseduradan foydalaning.

Kiruvchi ma'lumot	Chiquvchi ma'lumot
125	CXXV

4. Berilgan n soni xonalarining yig'indisini hisoblash dasturini tuzing. Protseduradan foydalaning.

Kiruvchi ma'lumot	Chiquvchi ma'lumot
125	8

5. Sport musobaqasida sportchilarning chiqishlari hakamlar tomonidan ballik tizimda baholandi. Yakuniy ballni olish uchun eng yuqori va eng past ball chiqarib tashlandi va qolgan uchta ballning o'rtacha arifmetik qiymati hisoblandi. 5 nafar hakam tomonidan berilgan eng yuqori va eng past ballni hamda sportchi olgan ballni chiqaradigan dastur tuzing. Funksiyadan foydalaning.

Kiruvchi ma'lumot	Chiquvchi ma'lumot
1 2 3 4 5	15 3.00

6. Berilgan n soni xonalari raqamlarining sonini chiqaruvchi dastur tuzing.

Kiruvchi ma'lumot	Chiquvchi ma'lumot
647521	6

59–60-DARSLAR. FUNKSIYALAR VA O'ZGARUVCHILAR

Har bir o'zgaruvchi e'lon qilinganda, u dasturda qandaydir muhim vazifani bajaradi va dasturning qaysidir elementlari bilan bog'lanadi. Funksiya ichida e'lon qilingan o'zgaruvchilar (lokal o'zgaruvchilar) va funksiya tashqarisida e'lon qilingan o'zgaruvchilar (global o'zgaruvchilar)ning ahamiyatlari ham turlicha.



1. Dasturlashda qism dasturlardan qanday foydalaniladi?
2. Qism dasturlar qanday ishlaydi?
3. Funksiya nima va u qanday ishlaydi?
4. Protsedura nima va u qanday ishlaydi?

Lokal o'zgaruvchilar

Lokal o'zgaruvchilar faqatgina funksiya ichida ahamiyatga ega bo'lib, asosiy dastur va boshqa funksiyalar ichida ishlamaydi. Shuning uchun funksiya tashqarisida lokal o'zgaruvchiga murojaat etilganda, xatolik ro'y bergani haqida xabar chiqariladi.

Tayanch tushunchalar

- Lokal o'zgaruvchilar** – faqat o'zlari e'lon qilingan qism dasturda faol bo'ladigan o'zgaruvchilar.
- Global o'zgaruvchilar** – butun dastur davomida faol bo'ladigan o'zgaruvchilar.

Lokal o'zgaruvchilar – bu o'zlari e'lon qilingan qism dasturda faol bo'ladigan o'zgaruvchilar. Ularni muayyan funksiya doirasida qo'llash mumkin, shu sababli faqat shu funksiya doirasidagina amal qiladi. Lokal o'zgaruvchilardan foydalanish o'zgaruvchining qiymati dasturning boshqa qismlarida tasodifan o'zgarib qolishi xavfini kamaytiradi.

<pre>>>> def val(): a=10 print(a) >>> val()</pre>	<p>val nomli funksiya e'lon qilindi. Funksiya ichida <i>a</i> lokal o'zgaruvchi e'lon qilindi va unga qiymat berildi hamda <i>a</i> ning qiymatini ekranga chiqarish buyrug'i berildi.</p>
10	Asosiy dasturda val nomli funksiya chaqirilganda, ekranga <i>a</i> ning qiymatini chiqaradi.
<pre>>>> print(a)</pre>	Asosiy dasturda <i>a</i> ning qiymati ekranga chiqarish buyrug'i berilsa, xatolik haqidagi xabar chiqadi. Sababi <i>a</i> o'zgaruvchi – funksiya ichida e'lon qilingan lokal o'zgaruvchi, asosiy dasturda ahamiyatga ega emas.
<pre>Traceback (most recent call last): File "<pyshell#15>", line 1, in <module> print(a) NameError: name 'a' is not defined</pre>	

Global o'zgaruvchilar

Global o'zgaruvchilar butun dastur davomida faol bo'ladigan o'zgaruvchilardir. Ular qism dasturdan tashqarida, ya'ni asosiy dasturda e'lon qilinadi. Odatda, ularni modullarni import qilgandan keyin, kodning boshlanishida e'lon qilish kerak. Ularni odatiy o'zgaruvchilar kabi chaqirish mumkin.

<pre>>>> b=5 >>> def val2(): print(b) >>> val2()</pre>	<p><i>b</i> global o'zgaruvchi e'lon qilindi va 5 qiymat berildi.</p> <p>val2 nomli funksiya e'lon qilindi.</p> <p>Funksiya ichida <i>b</i> ning qiymatini ekranga chiqarish buyrug'i berildi.</p>
5	Asosiy dasturda val2 nomli funksiya chaqirilganda, ekranga <i>b</i> ning qiymatini chiqaradi.
<pre>>>> print(b)</pre>	<i>b</i> global o'zgaruvchiga dasturning ixtiyoriy qismida: asosiy dasturda ham, qism dasturda ham murojaat qilish mumkin va u natijani chiqaradi.
5	

Global o'zgaruvchining qiymatini o'zgartirish

Agar funksiya global o'zgaruvchining qiymatini o'zgartirishga to'g'ri kelsa, u holda funksiya "global" kalit so'zi yordamida o'zgaruvchini yana qayta e'lon qilish zarur.

<pre>>>> b=5 >>> def val3(): global b b=b+1 print(b) >>> val3()</pre>	<p><i>b</i> global o'zgaruvchi e'lon qilindi va 5 qiymat berildi.</p> <p>val3 nomli funksiya e'lon qilindi.</p> <p><i>b</i> o'zgaruvchining qiymatini o'zgartirish va uni butun dasturda ahamiyatli qilish uchun u funksiya ichida global sifatida qayta e'lon qilinadi.</p> <p>Funksiya ichida <i>b</i> ning qiymatini ekranga chiqarish buyrug'i berildi.</p>
6	Asosiy dasturda val3 nomli funksiya chaqirilganda, ekranga <i>b</i> ning qiymatini chiqaradi.


```
>>> print(b)
```

```
6
```

b global o'zgaruvchiga asosiy dasturda murojaat qilinganda ham natija sifatida 5 ni emas 6 ni chiqaradi, sababi u funksiya ichida global sifatida e'lon qilingan.

O'zgaruvchi funksiya parametr sifatida

Agar o'zgaruvchi funksiya parametr sifatida ishlatilsa, uning qiymati lokal o'zgaruvchining yangi qiymati sifatida qabul qilinadi.

```
>>> def val4(d):
```

```
    print(d)
    d=100
```

```
    print(d)
>>> c=200
>>> val4(c)
```

```
200
```

```
100
```

d ning qiymati **val4** nomli funksiya qaysi qiymat bilan chaqirilsa, shu qiymatga teng, ya'ni c ga teng.

Funksiya ichida parametr sifatida kelgan d ning qiymatini ekranga chiqarish buyrug'i berildi.

Lokal o'zgaruvchi d ga yangi qiymat berildi.

d lokal o'zgaruvchining qiymati ekranga chiqarildi.

Global o'zgaruvchi c e'lon qilindi.

Asosiy dasturda **val4** nomli funksiya c parametr bilan chaqirilganda, birinchi parametr sifatida kelgan 200 ni, keyin esa lokal o'zgaruvchiga yangi berilgan qiymat 100 ni chiqaradi.



1. Funksiyaning parametrlari nima?
2. Funksiyaning tanasida qaysi kodlar yoziladi?
3. Lokal o'zgaruvchi deb nimaga aytiladi?
4. Global o'zgaruvchi deb nimaga aytiladi?
5. Lokal va global o'zgaruvchilarning farqi nimada?



1. n natural soni berilgan. $S=1*5+2*6+3*7+...+n*(n+4)$ ifodani hisoblash dasturini protsedura yordamida tuzing.
2. a va b natural sonlari berilgan. a va b sonlaridan kattasini topish funksiyasini tuzing. Funksiyadan foydalanib, a , b va c sonlari ichidan kattasini topish dasturini tuzing.

61-DARS. AMALIY MASHG'ULOT

Misol. Berilgan ikkita sonning eng katta umumiy bo'luvchisi (EKUB)ni topish dasturini tuzing. Funksiyadan foydalaning.

Berilgan ikkita sonning eng katta umumiy bo'luvchisi (EKUB)ni topishning eng oddiy usuli maktab darsliklarida ko'rsatib o'tilgan bo'lib, bunda ikkala son tub bo'luvchilarga ajratiladi va ulardan umumiyliklari ko'paytirib chiqiladi.

$$60=2*2*3*5$$

$$21=3*7$$

$$\text{EKUB}(60,21)=3$$

Bu usul matematikada qulay, lekin dasturlashda undan foydalanish noqulay va sekin kechadi. Shu sababli, **Yevklid algoritmi**dan foydalaniladi.

Yevklid algoritmidan 2 ta usuldan foydalaniladi.

1-usul. Ikkita sonning kattasidan kichigi ayirib boriladi va ayirma kichik sondan kichik bo'lib qolsa, ularning o'rnini almashtiriladi.

1) $60 - 21 = 39$; 2) $39 - 21 = 18$; 3) $21 - 18 = 3$.

$EKUB(60, 21) = 3$

2-usul. Qadamlar sonini kamaytirish uchun ayirish o'rniga qoldiqli bo'lishdan foydalanish mumkin.

1. Ikkita sonning kattasini kichigiga bo'lib, qoldiq topiladi.

2. Ularning o'rnini almashtiriladi.

3. 1- va 2-qadamlar sonlardan biri nol bo'lguncha davom ettiriladi.

4. Oxirida qolgan son shu ikki son uchun EKUB bo'ladi.

$60/21 = 2 \cdot 21 + 18$ qoldiq

$21/18 = 1 \cdot 18 + 3$ qoldiq

$18/3 = 6 \cdot 3 + 0$ qoldiq

$EKUB(60, 21) = 3$

<pre>def ekub(a, b): while a != 0 and b != 0: if a > b: a %= b else: b %= a ekub_q = a + b return ekub_q a = int(input()) b = int(input()) print(ekub(a, b))</pre>	<p>a, b parametrlarini qabul qiluvchi ekub nomli funksiya e'lon qilindi.</p> <p>Sikl tanasi buyruqlari a va b 0 ga teng bo'lguncha takrorlanadi: 1-sikl. $a = 60; b = 21$ 2-sikl. $a = 18; b = 21$ 3-sikl. $a = 18; b = 3$ 4-sikl. $a = 0; b = 3$ sikldan chiqadi.</p> <p>Agar a soni b dan katta bo'lsa, a/b qoldig'ini a ga o'zlashtiradi: 1-sikl. $60 > 21 \rightarrow a = 60 \% 21 = 18$ 3-sikl. $18 > 3 \rightarrow a = 18 \% 3 = 0$</p> <p>Aks holda, b/a qoldig'ini b ga o'zlashtiradi: 2-sikl. $18 < 21 \rightarrow b = 21 \% 18 = 3$</p> <p>$ekub_q = 0 + 3 = 3$ EKUB ni hisoblaydi.</p> <p>ekub_q qiymatni qaytaradi.</p> <p>a, b ga qiymat beriladi.</p> <p>2 ta son EKUBining funksiyasi chaqirilib, ekranga natija chiqariladi.</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1. Berilgan to'rtta sonning eng kichigini topuvchi dastur tuzing. Buning uchun $min4(a, b, c, d)$ funksiyasini yarating.
2. Berilgan a (haqiqiy) sonining k (butun) darajasini topuvchi dastur tuzing. Buning uchun $daraja(a, k)$ funksiyasini yarating.
3. Inglizcha harf va raqamlardan iborat satr berilgan. Ushbu satr qancha raqamdan iboratligini hisoblovchi dastur tuzing.



Kiruvchi ma'lumot	Chiquvchi ma'lumot
jdf423h4545b5213b8u58hkj2k32	17

4. Berilgan sonni akslantiruvchi dastur tuzing. Masalan, 123 dan 321 ni hosil qiling. Funksiyadan foydalaning.

5. 0 bilan tugaydigan sonlar ketma-ketligi berilgan. Uning raqamlari yig'indisini sikl ishlatmagan holda hisoblash dasturini tuzing.

Masalan, $1790 = 17$

6. Tonna, kilogramm va grammlarda berilgan birlikni grammga aylantirish funksiyasini yozing.

Kiruvchi ma'lumot	Chiquvchi ma'lumot
tonna=14 kg=32 g=125	14032125 g

62–63-DARSLAR. PYTHON DASTURLASH TILI KUTUBXONASI

Har bir yangi dasturning kodini yozish ko'p vaqt talab qiladigan jarayon hisoblanadi. Shu sababli, tayyor qism dasturlardan foydalanish har bir dasturchi uchun qulaydir. Zamonaviy dasturlash tillarida bu jarayonni yengillashtirish uchun tayyor dastur kodlarini saqlovchi kutubxonalar mavjud.

Tayanch tushunchalar

Modullar – alohida faylda yozilgan bo'lib, turli dasturlarda qo'llanilishi mumkin bo'lgan kodlar majmui.



1. Dasturlash tillari yordamida hech qanday dastur kodini yozmasdan ham biror masalani hal etish mumkinmi?
2. Standart kutubxona qanday vazifa bajaradi?
3. Modullar nima?

Boshqa dasturlash tillari kabi Python dasturlash tilining standart kutubxonasi ham ko'plab tayyor kod fragmentlari (modullar, standart funksiyalar va b.)dan tarkib topgan. Python dasturlash tilini yanada takomillashtirish uchun foydalanuvchi tomonidan yozilgan modullarni kutubxonaning alohida qismiga yuklash ham mumkin.

Python dasturlash tilida juda ko'p modul mavjud bo'lib, ularning asosiyalarini ko'rib chiqamiz.

Eslab qoling!

Python dasturlash tili o'rnatgichidagi Batteries included (батарейки в комплекте – batareykasi bilan) izohi Python dasturlash tili majmuida ko'plab tayyor kodlar mavjudligini anglatadi.

Modul nomi	Modul tavsifi
math	Modul murakkab matematik ifodalarni hisoblash uchun qo'llaniladi.
random	Modul tasodifiy sonlarni tanlab oladi yoki ro'yxat elementlarini tasodifiy tartibda joylashtiradi.
tkinter	Modul foydalanuvchi va dastur o'rtasida o'zaro aloqa o'rnatish uchun oyna, tugmacha va boshqa grafik elementlardan foydalanish imkonini beradi.
datetime	Modul joriy sana va vaqtni ko'rsatish hamda sanalarni hisoblash, sanalar ustida amallar bajarish imkonini beradi.
socket	Modul kompyuterlarni internet orqali bir-biriga ulash uchun qo'llaniladi.
turtle	Modul ekranda chiziq va figuralarni chizish uchun qo'llaniladi.
locale	Modul sonlarni formatlashda belgilangan tartibni aniqlash muammosini hal qilish uchun qo'llaniladi.
decimal	Modul o'nli kasr sonlari bilan ishlash va ularni yaxlitlash uchun qo'llaniladi.
os	Modul katalog va fayllar bilan ishlash uchun bir qancha imkoniyatlarni beradi.
copy	Modul nusxalash bilan bog'liq masalalarni hal qilishga mo'ljallangan.
sys	Python interpretatorida dasturni bajaruvchi muhit hisoblanadi.

Dasturda modullardan foydalanish uchun avval uni dasturga yuklash kerak bo'ladi. Bu esa uning tarkibida saqlanuvchi kod fragmentiga murojaat qilishni talab qiladi. Modullarni dasturga yuklashning uch xil usuli mavjud.

1-usul. Modul funksiyalarini yuklab olishning bu usulida murojaat etilayotgan kod fragmentidan oldin modulning nomi ko'rsatilishi lozim. Bunday dasturlarni o'qish juda oson, chunki kod qaysi modulga tegishli ekanligini tez aniqlash mumkin.

Sintaksisi:

import modul fayli nomi

import modulni yuklovchi kalit so'z

<pre>>>> import random >>> random.randint(1,5)</pre>	Standart kutubxonadan <i>random</i> moduli funksiyalarini yuklab oldi.
3	Har bir funksiyadan oldin modulning nomi ko'rsatiladi.

2-usul. Modul funksiyalarini yuklab olishning bu usuli uncha katta bo'lmagan dasturlar uchun qulay. Katta dasturlarda esa dasturni tushunish qiyin kechadi, ya'ni funksiya qaysi modulga taalluqli ekanligini aniqlash uchun biroz urinish kerak bo'ladi.

Sintaksisi:

from modul fayli nomi
import *

from ... import * – modulni yuklovchi kalit so'z

Eslab qoling!

Pygame – ovoz va maxsus o'yin grafikasi bilan ishlash imkonini beruvchi, videoo'yinlar yaratish uchun mo'ljallangan Python kutubxonasi. Pythonni puxta o'rganib olsangiz, Pygameda dasturni erkin tuza olasiz.

<pre>>>> from random import * >>> randint(1,5)</pre>	Standart kutubxonadan <i>random</i> moduli funksiyalarini yuklab oldi.
4	Funksiya qaysi modulga taalluqli ekanligi ko'rsatilmaydi.

3-usul. Moduldan faqat funksiyalarning o'zini yuklab olish ham mumkin. Agar dasturdan faqat bitta funksiya kerak bo'lsa, u holda funksiyalarning barchasini emas, kerakli funksiyaning o'ziningina yuklab olgan ma'qul.

Sintaksisi:

from modul fayli nomi **import** funksiya nomi

<pre>>>> from random import randint >>> randint(1,5)</pre>	Standart kutubxonadan <i>random</i> modulining <i>randint</i> funksiyasi yuklab olindi. Funksiya qaysi modulga taalluqli ekanligi ko'rsatilmaydi.
3	

Mabodo, qaysi modulda qanday funksiyalar mavjud, ular nimaga xizmat qiladi singari savollar tug'lsa, Python kutubxonasining ma'lumotnomasini ko'rib chiqish mumkin. Dastur tuzish jarayonida vaqtni bekorga sarflamaslik uchun, standart kutubxona, modul va funksiyalar haqida batafsil bilish hamda tayyor kodlardan foydalanish muhim ahamiyatga ega.

Ma'lumotnomani ishga tushirish uchun menyular qatoridan **Help**ni bosib **Python Docs** bo'limi tanlanadi.

random moduli tasodifiy sonlarni generatsiya qilishni boshqaradi.

random modulining funksiyalari

Funksiyalar	Tavsifi
random(x)	0 dan 1 gacha tasodifiy sonlarni hosil qiladi.
randint(start, stop)	start dan stop gacha bo'lgan oraliqdagi tasodifiy sonlarni hosil qiladi.
randrange(start, stop, step)	qandaydir sonlar to'plamidan start dan stop gacha bo'lgan oraliqda step qadami bilan tasodifiy sonlarni hosil qiladi.



1. Dasturlash tili kutubxonasi nima?
2. Modul nima va u nima maqsadda ishlatiladi?
3. Qanday modullarni bilasiz?
4. Matematik hisoblashlarni amalga oshiruvchi modul qanday nomlanadi?



1. 0 va 1 ning oralig'idan 10 ta tasodifiy sonni chiqarish dasturini tuzing.
2. 0 va 1 ning oralig'idan 10 ta tasodifiy sonni chiqarish dasturini tuzing.
3. 10 va 10000 ning oralig'idan 5 ta tasodifiy sonni chiqarish dasturini tuzing.
4. 20 va 50 ning oralig'idan 2 qadam bilan 7 ta tasodifiy sonni chiqarish dasturini tuzing.

64-DARS. AMALIY MASHG'ULOT

Python kutubxonasining math moduli o'z ichiga matematik, trigonometrik va logarifmik amallarni bajaruvchi funksiyalarni qamrab olgan.

Funksiyalar	Tavsifi
$\text{ceil}(x)$	Eng yaqin katta butun songacha yaxlitlaydi. $\text{Ceil}(1.5)=2$, $\text{ceil}(-1.5)=-1$
$\text{round}(x, n)$	x sonini nuqtadan keyin n ta belgi qolgunga qadar yaxlitlaydi.
$\text{floor}(x)$	Eng yaqin kichik butun songacha yaxlitlaydi. $\text{floor}(1.5)=1$, $\text{floor}(-1.5)=-2$
$\text{round}(x)$	x sonini yaxlitlaydi.
$\text{log}(a, b)$	b asosga ko'ra a logarifmini hisoblaydi.
$\text{log10}(x)$	x sonining o'nli logarifmini hisoblaydi.
$\text{sqrt}(x)$	x ning kvadrat ildizini hisoblaydi.
$\text{pow}(x, n)$	x ning n -darajasini hisoblaydi.
$\text{factorial}(x)$	x faktorialni hisoblaydi.
$\text{abs}(x)$	x ning modulini hisoblaydi.
$\text{cos}(x)$	x ning kosinusini hisoblaydi.
$\text{sin}(x)$	x ning sinusini hisoblaydi.
$\text{tan}(x)$	x ning tangensini hisoblaydi.
$\text{acos}(x)$	x ning arkkosinusini hisoblaydi.
$\text{asin}(x)$	x ning arksinusini hisoblaydi.
$\text{atan}(x)$	x ning arktangensini hisoblaydi.
$\text{degrees}(x)$	Radiandan gradusga o'tkazadi.
$\text{radians}(x)$	Gradusdan radianga o'tkazadi.

Masala. Tomoni a ga teng bo'lgan teng tomonli uchburchakni yuzini hisoblash dasturini tuzing. a foydalanuvchi tomonidan kiritiladi.

$$s = \sqrt{\frac{3}{4}a^2}$$

```
from math import *
a = int(input())
s = sqrt(3/4)*pow(a,2)
print(s)
```

```
4
13.856406460551018
```



- Berilgan burchak yoyining uzunligini hisoblovchi dastur tuzing. Yoyning burchagi (gradusda) hamda radiusi foydalanuvchi tomonidan kiritiladi.
- $y = x * \cos x$ funksiyasi qiymatini hisoblovchi dastur tuzing. x foydalanuvchi tomonidan kiritiladi.
- Kvadrat tenglamaning ildizlarini hisoblash dasturini tuzing. a , b , c foydalanuvchi tomonidan kiritiladi.
- Berilgan haqiqiy sonning kasr qismini 1 dan 4 gacha bo'lgan aniqlikda yaxlitlang.
Berilgan son: 0.26598
Natija:
1-aniqlikda: 0.3 2-aniqlikda: 0.27
3-aniqlikda: 0.266 4-aniqlikda: 0.2660
- Doira sektorining yuzasini hisoblash dasturini tuzing. Doiraning radiusi hamda sektorning burchagi (gradusda) foydalanuvchi tomonidan kiritiladi.

65–66-DARSLAR. PYTHONDA FOYDALANUVCHI GRAFIK INTERFEYSI BILAN ISHLASH

Aksariyat dasturlash tillarida foydalanuvchi bilan o'zaro aloqani o'rnatish uchun boshqaruv elementlari: oyna, matnlar maydoni va tugmachalar ishlatiladi. Bular umumiy nom bilan foydalanuvchining **grafik interfeysi** (GUI – graphical user interface) deb ataladi.

Tayanch tushunchalar

Widget (vijet)– GUIga ega ilovani yaratish uchun foydalaniladigan tugmachalar yoki matnli maydonlar kabi interfeys elementlari.

BUNI BILASIZMI?



1. Foydalanuvchining grafik interfeysi nima?
2. Grafik interfeysli ilovalar qanday yaratiladi?

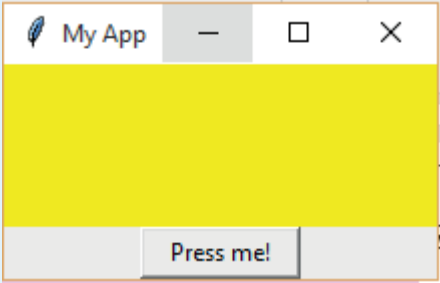
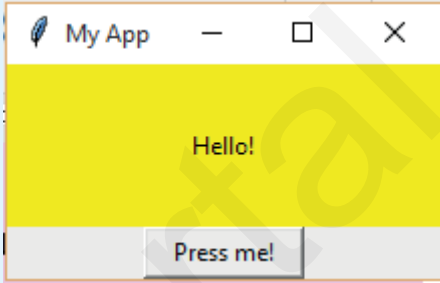
Barcha elementlar joylashadigan oyna GUIning asosi hisoblanadi. Oyna va uning elementlari (vijetlar)ni yaratish uchun Python standart kutubxonasining Tkinter modulidan foydalaniladi.

Tkinter – Pythondagi standart grafik kutubxona. Pythonni o'rnatganda kutubxona dasturning ichida birga taqdim etiladi. Python o'rnatilishi bilan GUIga ega ajoyib ilovalarni yaratish uchun zarur obyekt va usullardan foydalanish imkoniyati vujudga keladi. GUI ilovalarni yaratish uchun:

- Tkinter modulini import qilish;
- Tkinter asosiy oynasini yaratish;
- ilovaga bir yoki bir necha vijetni qo'shish;
- foydalanuvchi bajaradigan jarayonlarni tushunadigan va ularga javob qaytaradigan asosiy siklli kodga kirish lozim.

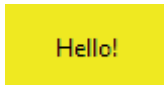


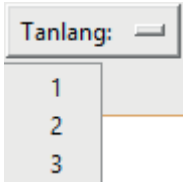
```
from tkinter import *  
  
window = Tk()  
  
window.title('My App')  
  
window.geometry('250x50')  
window.configure(background='yellow')  
  
my_label=Label(window, width=40,  
height=5, bg='yellow', text='')  
  
my_label.grid(row=0, column=0)  
  
def change_text():  
    my_label.config(text='Hello!')
```


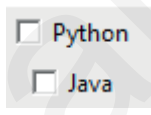
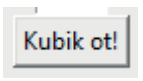

Kutubxonadan Tkinter modulini yuklab olish.
Tkinter oynasini yaratish.
Tkinter oynasi sarlavhasiga nom berish.
Tkinter oynasining o'lchami.
Tkinter oynasi fonining rangi.
Label funksiyasi yordamida eni 40, bo'yi 5, rangi sariq, bo'sh matnli maydon vijeti o'rnatiladi.
Matnli maydonni katakka joylashtirish 0-satr va 0-ustun.
Tugmachani bosganda chaqiriluvchi funksiyani e'lon qilish.

<pre>def change_text(): my_label.config(text='Hello!') my_button=Button(window, text="Press me!", width=10, command=change_text) my_button.grid(row=1, column=0) window.mainloop()</pre>	Tugmachani bosganda chaqiriluvchi funksiyani e'lon qilish.
	Button funksiyasi yordamida nomi Press me, eni 10 ga teng tugmacha o'rnatiladi. command = atributi yordamida tugmacha bosilganda, bajariluvchi funksiya ko'rsatiladi.
	Tugmachani katakka joylashtirish 1-satr, 0-ustun
	Asosiy siklga kirish
 <p>Dastur ishga tushganda, ushbu oyna chiqariladi.</p>	 <p>Tugmacha bosilganda, bo'sh matnli maydonda Hello! yozuvi paydo bo'ladi.</p>

Dastur ilovalarida foydalaniladigan barcha vijetlar `window = Tk()` va `window.mainloop()` buyruqlari orasida kiritiladi.

Tkinter modulidagi `grid()` usuli katakli koordinatalar tizimidan foydalangan holda vijetlarni kerakli koordinataga joylashtirish imkonini beradi.

Vijetlar	Dasturdagi ko'rinishi	Natijasi
Label()	<code>my_label=Label(window, width=40, height=5, bg='yellow', text="")</code> <code>my_label.grid(row=0, column=0)</code>	 Matnli maydon
Text	<code>text=Text(window, width=4, height=2)</code> <code>my_text_box.grid(row=0, column=1)</code>	 Natijani chiqarish uchun matnli maydon
Entry()	<code>my_text_box=Entry(window, width=20)</code> <code>my_text_box.grid(row=0, column=0)</code>	 Matn kiritiladigan maydon
OptionMenu()	<code>options=(1,2,3)</code> <code>my_variable_object=IntVar()</code> <code>my_variable_object.set('Tanlang:')</code> <code>my_dropdown=OptionMenu(window, my_variable_object, *options)</code> <code>my_dropdown.grid()</code>	 Tanlanadigan maydon

Radiobutton()	<pre>jinsi=StringVar() radio1=Radiobutton(window, text='Ayol', variable=jinsi, value='ayol') radio1.grid(row=3, column=0, sticky=W) radio1.select() radio2=Radiobutton(window, text='Erkak', variable=jinsi, value='erkak') radio2.grid(row=3, column=1, sticky=W) radio2.select()</pre>	
Checkbutton()	<pre>var1=IntVar() checkbox1=Checkbutton(window, text='Python', variable=var1) checkbox1.grid(row=0, column=0) var2=IntVar() checkbox2=Checkbutton(window, text='Java', variable=var2) checkbox2.grid(row=1, column=0)</pre>	
Button()	<pre>my_button=Button(window, text='Kubik ot!', command=kubik) my_button.grid(row=4, column=0)</pre>	
PhotoImage()	<pre>foto=PhotoImage(file='image/foto.png') my_label=Label(window, image=foto) my_label.grid(row=0, column=0)</pre>	



1. Foydalanuvchi grafik interfeysi (GUI) nima?
2. Vijet nima va u nima maqsadda ishlatiladi?
3. GUI ilovalarga vijet qanday qo'shiladi?
4. Ilova oynasida vijet qanday joylashtiriladi, qaysi usul yordamida?
5. Tugmachalar yordamida funksiya qanday ishga tushiriladi?



1. Rangi yashil, o'lchami 100x100 bo'lgan "Mening birinchi ilovam" nomli GUI oynasini yaratuvchi dastur kodini yozing. Oynada "Salom O'zbekiston" xabarini chiqaruvchi tugmachasini joylashtiring.
2. Rangi pushti, o'lchami 250x150 bo'lgan "Mevalar" nomli GUI oynasini yaratuvchi dastur kodini yozing. Oynada berilgan 4 ta mevdadan birini tanlash imkonini beruvchi vijetni joylashtiring.

67-DARS. AMALIY MASHG'ULOT

Misol. GUIdan foydalanib, 1 dan n gacha bo'lgan 10 ta tasodifiy sonni chiqaruvchi dastur tuzing. n foydalanuvchi tomonidan kiritiladi.

<pre>from tkinter import *</pre>	Kutubxonadan Tkinter modulini yuklab oladi.
<pre>from random import randint</pre>	Kutubxonadan <i>random</i> modulining <i>randint</i> funksiyasini yuklab oladi.
<pre>def tasodifiy():</pre>	Funksiyani e'lon qiladi.
<pre> number = int(textbox_input.get())</pre>	Matn maydoni orqali kiritilgan <i>textbox_input</i> o'zgaruvchisi qiymatini olib, butun songa aylantiradi.
<pre> textbox_output.delete(0.0, END)</pre>	Natijani chiqaruvchi maydon, ya'ni <i>textbox_output</i> o'zgaruvchisining qiymatini o'chiradi.
<pre> for i in range (1,11):</pre>	1 dan 11 gacha 10 ta sikl ishlaydi.
<pre> t_son = str (randint(1, number))</pre>	1 dan number gacha bo'lgan sonlar orasidan tasodifiy sonni topadi.
<pre> + '\n')</pre>	Tasodifiy sonni natija chiqaruvchi maydon, ya'ni textbox_output o'zgaruvchisiga tenglaydi.
<pre>textbox_output.insert(END, t_son)</pre>	Tkinter oynasini yaratish.
<pre>window = Tk()</pre>	Tkinter oynasi sarlavhasiga nom beradi.
<pre>window.title('Tasodifiy son')</pre>	Tkinter oynasi o'lchamini o'rnatadi.
<pre>window.geometry('250x250')</pre>	Tkinter oynasi fon rangini belgilaydi.
<pre>window.configure (background='yellow')</pre>	Tasodifiy son yuqori chegarasining maydoni nomini ko'rsatish uchun yorliq yaratadi.
<pre>input_label = Label (window,</pre>	Matnli maydonni katakka joylashtiradi: 0-satr va 0-ustun.
<pre> text='Son: ', bg='yellow')</pre>	Tasodifiy sonlarni chiqarish maydoni nomini ko'rsatish uchun yorliq yaratadi.
<pre>inpu t_label.grid (row=0, column=0)</pre>	Matnli maydonni katakka joylashtiradi: 2-satr va 0-ustun.
<pre>output_label = Label(window, text ='</pre>	Tasodifiy sonning yuqori chegarasini kiritish uchun matn maydoni yaratadi.
<pre>\nNatiya', bg='yellow')</pre>	Matnli maydonni katakka joylashtiradi: 1-satr va 0-ustun.
<pre>output_label.grid(row=2, column=0)</pre>	10 ta tasodifiy sonni chiqarish uchun matnli maydonni yaratadi.
<pre>textbox_input = Entry (window, width=5)</pre>	
<pre>textbox_input.grid (row=1, column=0)</pre>	
<pre>textbox_output = Text(window,</pre>	
<pre>height=10, width=6)</pre>	

```

textbox_output.
grid(row=3, column=0)

kubik_button = Button(window, text=
'Tasodifiy son', command=tasodifiy)

kubik_button.grid(row=1,
column=1)

window.mainloop()

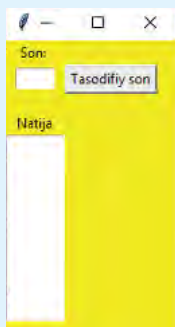
```

Matnli maydonni katakka joylashtiradi: 3-satr va 0-ustun.

"Tasodifiy son" nomli tasodifiy funksiyani chaqiradigan tugmacha yaratiladi.

Tugmachani katakka joylashtiradi: 1-satr, 1-ustun.

Asosiy siklga kirish



Dastur ishga tushganda, ushbu oyna chiqariladi.



Tugmacha bosilganda, 1 dan 15 gacha bo'lgan sonlar orasidan tasodifiylarini chiqaradi.



1. GUIDan foydalanib, berilgan songa karrali 10 ta sonni chiqaruvchi dastur tuzing. Berilgan son foydalanuvchi tomonidan kiritiladi.
2. GUIDan foydalanib, berilgan songa karrali sonlarni chiqaruvchi dastur tuzing. Berilgan son va karrali sonlarning soni foydalanuvchi tomonidan kiritiladi.
3. GUIDan foydalanib, ikkita a va b sonni qabul qiladigan, ab ni hisoblovchi dastur tuzing.
4. GUIDan foydalanib, n ta tub sonni chiqaruvchi dastur tuzing (1 ga va o'ziga bo'linadigan songa tub son deyiladi). n foydalanuvchi tomonidan kiritiladi.
5. GUIDan foydalanib, ikkita a va b sonni qabul qiladigan, ular EKUBini hisoblovchi dastur tuzing.
6. GUIDan foydalanib, ikkita a va b sonni qabul qiladigan, ularni EKUKini hisoblovchi dastur tuzing.

68-DARS. NAZORAT ISHI

1. 1 dan n gacha bo'lgan natural sonlar kubining yig'indisini topish dasturini tuzing.
2. $P=2*4*6*...*40$ ko'paytmani hisoblash dasturini tuzing.
3. 0 bilan tugovchi sonlar ketma-ketligi berilgan. Uni akslantiruvchi dastur tuzing. Masalan, 1230 dan 0321 hosil qiling.
4. Protsedura yordamida berilgan matndagi 'k' belgini 'q' belgiga, 't' belgini 'd' belgiga, 'n' belgini 'm' belgiga almashtiruvchi dastur tuzing

5. Berilgan ikki yoki undan ortiq sonning eng kichik umumiy karralisi (EKUK)ni topish dasturini tuzing. Funksiyadan foydalaning.

Kiruvchi ma'lumot	Chiquvchi ma'lumot
18 24	72

6. Soat, minut va sekunda berilgan birlikni sekundga aylantirish funksiyasini yozing.

Kiruvchi ma'lumot	Chiquvchi ma'lumot
soat=4 minut=15 sekund=60	15360 s

7. Canvas maydonida matn va rasmdan iborat otkritka hosil qiling.

8. GUIdan foydalanib, kalkulyator hosil qiling.

9. Shilliqqurt balandligi h metr daraxtning yuqorisiga sudralib chiqmoqda. U kunduzi a metr ko'tarilsa, kechasi b metr pastga tushadi. U daraxt uchiga necha kunda yetib boradi? h , a va b ($a > b$) qiymatlar foydalanuvchi tomonidan kiritiladi.

10. Ikki a va b son berilgan. Ulardan kattasini aniqlash dasturini tuzing.

11. Belgilar yordamida hosil qilingan pingvinni n marta chiqarish dasturini tuzing. $n - 1$ dan 4 gacha bo'lgan natural sonlarni qabul qiladi.



12. n ta kvadratni ekranga chiqaruvchi dastur tuzing. $n - 1$ dan 5 gacha bo'lgan natural sonlarni qabul qiladi.

```

+ + + + +
+ + + + +
+ + + + +
+ + + + +

```

FOYDALANILGAN ADABIYOTLAR VA WEB-SAYTLAR RO'YXATI

1. Boltayev B., Azamatov A., Asqarov A., Sodiqov M., Azamatova G. Informatika va hisoblash texnikasi asoslari. Umumiy o'rta ta'lim maktablarining 9-sinfi uchun darslik. Toshkent: "Cho'lpon" nomidagi NMIU, 2015. – 160 b.
2. Chris Roffey. Computer science. Programming book for Python. – USA: Cambridge university press. 2017, – p. 204
3. Chris Roffey. Python basics. Coding club. Level 1,2. – USA: Cambridge university press. 2012, – p. 85
4. Eric Matthes. Python crash course: a hands-on, project-based introduction to programming. – San-Francisco: No Starch Press, 2015. – p. 562
5. Matt Harrison. Illustrated guide to Python 3. 2017, – p. 267
6. Dan Bader. Python tricks the book. Anja Pircher Design, 2017, – p. 299
7. Jamie Chan. Learn python in one day and learn it well. – p. 132
8. Jake VanderPlas. A whirlwind tour of python. – USA: O'Reilly Media. 2016. – p. 98
9. Carol Vorderman. Computer Coding for Kids: A unique step-by-step visual guide, from binary code to building games. London: Dorling Kindersley Ltd, 2014, – p. 224.
10. Robert Sedgewick and Kevin Wayne. Algorithms. Fourth edition. Princeton University. First printing, March 2011.
11. Садыгов И. Дж., Махмудзаде Р. А., Исаева Н. Р. Информатика-11. Учебник для общеобразовательных школ. «Вакінағр» – Баку, 2011, 128 стр.
12. Алексеев Е. Г., Богатырев С. Д. Информатика. Мультимедийный электронный учебник. <http://inf.e-alekseev.ru/text/index.html>
13. Васильев А. Н. Python на примерах. – Санкт Петербург: Наука и техника, 2018, – 430 с.
14. Джейсон Бриггс. Python для детей. Самоучитель по программированию / пер. с англ. Станислава Ломакина; [науч. ред. Д. Абрамова]. – М.: Манн, Иванов и Фербер, 2017. – 320 с.
15. Кадиркулов Р. А., Рыскулбекова А. Информатика, 7 класс. <https://www.opiq.kz/Kit/Details/61>
16. <https://www.w3resource.com/python/>
17. <https://younglinux.info/python/task/>
18. <https://pythonru.com/>
19. <https://python-scripts.com/>
20. <https://www.rupython.com/>
21. <https://informatics.msk.ru/>
22. <https://pythonworld.ru/>
23. <http://pythoshka.ru/>

Fayziyeva Maxbuba Raximjonovna,
Sayfurov Dadajon Muxammedovich,
Xaytullayeva Nafisa Saxobiddinovna

O'quv nashri

INFORMATIKA VA AXBOROT TEXNOLOGIYALARI

Umumiy o'rta ta'lim maktablarining 9-sinfi uchun darslik

(O'zbek tilida)

Bosh muharrir:

R. Xidirova

Dizayner-rassomlar:

S. Doniyorov, K. Shadrin, Sh. To'raxonov, I. Matlab

Texnik muharrir:

S. Serenkov

Sahifalovchi:

K. Melnikova

Musahhihlar:

A. Umarova, D. Rahmonova, N. Oymatova

“Nashriyot uyi Tasvir”

Toshkent – 2020

Nashriyot litsenziyasi AI №292, 23.02.2017

Bosishga 2020-yil 21-sentabrda ruxsat etildi. Bichimi 60x84 1/8.

Qog'ozi silliqlangan-jilolanmagan.

Kegli 11, Roboto, Agency FB, a_EmpiricalNr garniturada
Ofset bosma usulida bosildi. Shartli bosma tabog'i 13,02.

Nashr tabog'i 15,23. Adadi 502 330 nusxa.

Buyurtma raqami:2165.

“Kolorpak” MChJ bosmaxonasida chop etildi.

Toshkent shahar, Elbek ko'chasi, 8-uy



COLORPACK

Ijaraga berilgan darslik holatini ko'rsatuvchi jadval

T/r	O'quvchining ismi, familiyasi	O'quv yili	Darslikning olingandagi holati	Sinf rahbarining imzosi	Darslikning topshirilgandagi holati	Sinf rahbarining imzosi
1						
2						
3						
4						
5						
6						

Darslik ijaraga berilib, o'quv yili yakunida qaytarib olinganda yuqoridagi jadval sinf rahbari tomonidan quyidagi baholash mezonlariga asosan to'ldiriladi.

Yangi	Darslikning birinchi marotaba foydalanishga berilgandagi holati.
Yaxshi	Muqova butun, darslikning asosiy qismidan ajralmagan. Barcha varaqlari mavjud, yirtilmagan, ko'chmagan, betlarida yozuv va chiziqlar yo'q.
Qoniqarli	Muqova ezilgan, birmuncha chizilib chetlari yedirilgan, darslikning asosiy qismidan ajralish holati bor, foydalanuvchi tomonidan qoniqarli ta'mirlangan. Ko'chgan varaqlari qayta ta'mirlangan, ayrim betlariga chizilgan.
Qoniqarsiz	Muqovaga chizilgan, yirtilgan, asosiy qismidan ajralgan yoki butunlay yo'q, qoniqarsiz ta'mirlangan. Betlari yirtilgan, varaqlari yetishmaydi, chizib, bo'yab tashlangan. Darslikni tiklab bo'lmaydi.

Sotuvga chiqarish taqiqlanadi



«NASHRIYOT UYI TASVIR»

ISBN 978-9943-5800-2-2



9 789943 580022