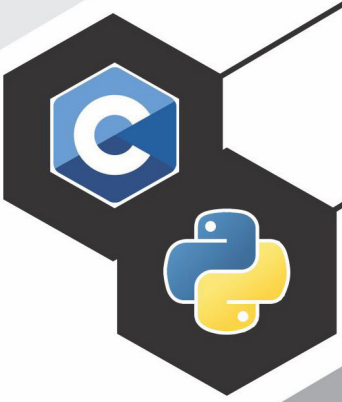
# DASTURLASH

R. Abdullayev
Sh. Raxmatov



## 0-qadam

# MATEMATIKA

Ushbu elektron kitob
**“Dasturlash 0-qadam: MATEMATIKA”**
kitobining bir qismi bo‘lib, kitob bilan oldindan tanishish maqsadida tekinga tarqatildi.

## MUNDARIJA

# Muallifdan

Ushbu kitob qo‘lingizdami, demak, siz dasturlashni o‘rganmoqchisiz yoki o‘rganib bo‘lgansiz. Agar o‘rganmoqchi bo‘lsangiz, o‘ylaymanki, kitob o‘qishga erinmaysiz.

O‘zim ham kitob o‘qisam, muallif yozgan gaplarini o‘qimasdim. Shuning uchun, gapni qisqa qilaman.

# Kitobni qanday o‘qiyman?

Kitobda dasturlashga aloqador bo‘lgan matematikadagi mavzular va ushbu mavzular dasturlashda qanday yechilishi haqida yoritilgan. Mundarijada faqat matematik mavzular ko‘rsatilgan lekin har bir mavzuda 💾 belgisidan keyin **C** va **Python** dasturlash tillaridagi dasturlar yoki mavzuga aloqador dasturlash haqida qo‘shimcha ma’lumotlar berilgan.

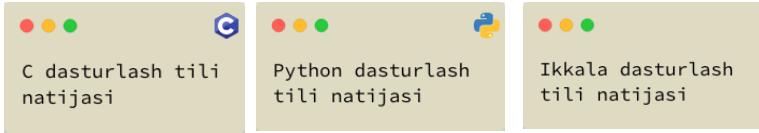
Berilgan dastur natijalari esa izoh shaklida yoki rasm shaklida ko‘rsatilgan bo‘lib, izoh shaklida kelsa – **C** dasturlash tilida // belgisidan keyin, **Python** dasturlash tilida # belgisidan keyin berilgan:

```
printf("%u", 7); // 7
```

```
print(7) # 7
```

*7 sonini ekranga chiqaradigan dastur qismlari va ularning natijasi.*

Rasm shaklidagi dastur natijasi esa uch xil ko‘rinishda bo‘lishi mumkin:




Agar natija birorta dasturlash tiliga tegishli bo‘lsa, ushbu dasturlash tilining logotipi keltiriladi. Agar ikkala dasturlash tilidagi natija bir xil bo‘lsa, natijada dasturlash tili logotiplari keltirilmaydi.

Ikkita dasturlash tilining keltirilish sababi esa, sizga dasturlash tillari aslida qanday ishlashi va qanday tuzulganligini tushunishingizga yordam beradi.

*Agar kitobni shu joyigacha o‘qigan bo‘lsangiz, demak, siz haqiqatdan ham dasturlashni o‘rganmoqchisiz.*

Kitobning davomini o‘qishdan oldin, qo‘lingizga ruchka va daftar oling. Chunki, kitobni o‘qish jarayonida sizda savollar paydo bo‘lishi tabiiy hol. Agar kitobda nimadir tushunarsiz bo‘lsa, darhol daftarga savolingizni va kitobning qavsi betida ekanligini yozib qo‘ying, kitobni o‘qishda davom eting.

Agar savollaringizga kitob o‘qish davomida ham javob topa olmasangiz, itboom.uz platformasidagi “Yordam” bo‘limiga kiring.

Har bir mavzuning oxirida **QR kodlar** berilgan bo‘lib, ushbu kodga mavzuga aloqador havola biriktirilgan. Agar siz kitobning PDF shaklini o‘qiyotgan bo‘lsangiz, **QR kod** ustiga bosish orqali, agar qog‘oz shaklini o‘qiyotgan bo‘lsangiz, telefoningizda **QR kod**ni o‘qish orqali ushbu havolaga o‘tishingiz mumkin. Havolada esa ushbu mavzuga aloqador masalalar va qo‘shimcha ma’lumotlar mavjud.

Ayrim mavzular tugallanmasdan mavzuga doir to‘ldiruvchi ma’lumotlarga havolalar keltirilgan. Ushbu havolalarga o‘tmasdan kitobni o‘qishda davom etmang. Chunki kitobdagi ma’lumotlar berilgan havoladagi ma’lumotlar bilan chambarchas bog‘liq.

Mavzularda ushbu mavzuga aloqador qiziq faktlar keltirilgan bo‘lib, faktlar asosan internet tarmog‘idagi eng mashhur wikipedia.org internet-ensiklopediyasidan olingan.

# ITBoom.uz platformasi

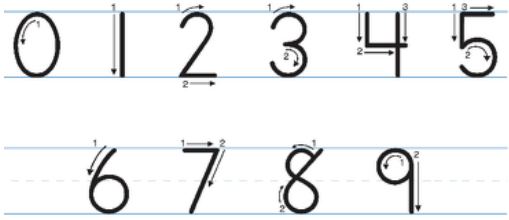
ITBoom.uz – bu web sayt bo‘lib, ushbu platformaning yaratilishining asosiy maqsadi, dasturlashni kompyutersiz, notebooksiz faqat telefonning o‘zida o‘rganish imkoniyatini berish. Bundan tashqari har bir mavzuga aloqador bo‘lgan masalalarni ham platformada ishlashingiz mumkin. Platforma esa sizning o‘zlashtirishingiz, qaysi masalalarni necha marotaba to‘g‘ri yoki noto‘g‘ri ishlaganingiz kabi ma’lumotlarni ham berib boradi.

ITBoom.uz platformasi va platformadan ro‘yxatdan o‘tish haqida to‘liq ma’lumotni quyidagi havola orqali olasiz:

# Raqamlar va sonlar

## Raqamlar

Raqamlar kundalik hayotimizda ishlatadigan eng ko‘p matemetik belgilardir. Raqamlarni sonlar bilan adashtirmaslik lozim. Raqam – bu belgi ya’ni sonlarni ifodalash uchun qo‘llaniladigan matematik belgilar.



Ularga 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 belgilari kiradi va **arab raqamlari** deyiladi. Shu belgilardan foydalanib sonlar hosil qilamiz. Sonlarni esa hayotda sanaladigan narsalarni ifodalash uchun qo‘llaymiz.

*Qiziq fakt: 0 raqami dastlab Hindistonda paydi bo‘lgan, Al-Xorazmiy esa 0 raqamidan algebrada qanday foydalanishni tushuntirib bergan.*

Tarixda raqamlarni nafaqat biz bilgan belgilar, balki so‘zlar va harflar oraqali ham ifodalashgan. Jumladan, rim raqamlari shu toifaga kiradi.

Dasturlash tilida raqamlar bilan ishlash juda oddiy. Klaviatura (tugmatag) orqali mos belgilarni bosib yozsa bo‘ladi.


Ya’ni, dasturlash tillarida ishlatiladigan raqamlar o‘zimiz ishlatadigan raqamlar bilan bir xil ma’noga ega.

Endi tasavvur qiling, biz oddiy bitta kichik dastur tuzdik. Bu dasturning vazifasi 7 raqamini ekranga chiqarish bo‘lsin. Raqamlarni kiritish klaviatura orqali amalga oshiriladi lekin dasturga kiritilgan raqamni ekranga chiqarish uchun maxsus funksiyalardan (funksiya haqida kitob oxirida yoritilgan) foydalanamiz.

Quyida ikkita dasturlash tilida kiritilgan 7 raqamini ekranga chiqarish funksiyalari ko‘rsatilgan.

```c
printf("%u", 7); // 7
```

```python
print(7) # 7
```

Ko‘rib turganingizdek har bir dasturlash tillarida bir xil vazifani bajaradigan o‘xshash funksiyalar juda ko‘p. Faqat ularning asosiy farqlari dasturlash tilining sintaksisidir, ya’ni yozilish qoidalaridir.

Yuqorida keltirilgan ikkita funksiya: printf va print haqida kitob davomida qo‘shimcha ma’lumotlar berilgan.

Mavzuga doir masalalar:

## Sonlar

Sonlarni hosil qilish juda oddiy, arab raqamlarini olamiz va xohlagan ketma-ketlikda joylashtiramiz. Hosil bo‘lgan raqamlar ketma-ketligiga **son** deyiladi. Masalan:

1; 23; 445; 556444; 145389089374; 17934745992792347927

Ko‘rib turganingizdek raqamlarning o‘zi ham son bo‘lib kelishi mumkin.

Dasturlash tilida sonlar bilan ishlash xuddi raqamlar bilan ishlash bilan bir xil.

Mavzuga doir masalalar:

## Sonlarni sinflarga ajratish

Raqamlarni ketma-ket yozib sonlarni hosil qildik ham deylik, lekin u yozilgan sonlar nima ma’no anglatishini insonlar tushunishi uchun bir nechta xususiyatlarini o‘rganishimiz kerak bo‘ladi.

Shulardan birinchisi, sondagi har bir raqamning joylashgan joyga nisbatan xona tushunchasi mavjud bo‘lib, ular o‘ngdan chapga (←) qarab o‘z nomiga ega (jadvalga qarang).

Ikkinchisi, o‘ngdan chapga qarab ajratilgan har 3 ta raqam guruhlari sonning sinfi deb nomlanadi va har bir sinf ham o‘z nomiga ega (jadvalga qarang).

| **Raqam o‘rni** | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|
| **Xona nomi** | yuz million-lar | o‘n million-lar | million-lar | yuz minglar | o‘ng minglar | minglar | yuzlar | o‘nlar | birlar |
| **Sinf nomi** | millionlar sinfi | | | mignlar sinfi | | | birlar sinfi | | |

| **Raqam o‘rni** | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 |
|---|---|---|---|---|---|---|---|---|---|
| **Xona nomi** | yuz kvad-rillionlar | o‘n kvad-rillionlar | kvad-rillionlar | yuz tril-lionlar | o‘n trillion-lar | trillion-lar | yuz mil-liard lar | o‘n mil-liard-lar | milliard-lar |
| **Sinf nomi** | kvadrillion sinfi | | | trillionlar sinfi | | | milliardlar sinfi | | |

Uchinchisi, berilgan sonning xonalari soni ushbu sondagi raqamlar soni bilan aniqlanadi. Masalan:

| Son | 1 | 22 | 4 568 | 1 554 578 | 1 256 389 456 751 |
|---|---|---|---|---|---|
| Xonalar soni | 1 xonali | 2 xonali | 4 xonali | 7 xonali | 13 xonali son |

Masalan, 63240023189745 soni berilgan bo‘lsa, ushbu songa nisbatan ayrim quyidagi xulosalarni chiqarish mumkin:

- berilgan son 14 xonali, chunki jami raqamlar soni 14 ta;
- sonning 5-xonasidagi raqam 8 ga teng, chunki o‘ngdan boshlab sanasak 5-o‘rinda turgan raqam 8 ga teng;
- sonning milliardlar xonasida turgan raqam 0 ga teng, chunki milliardlar xonasi o‘ngdan boshlab 10-xonada joylashgan, 10-xonada turgan raqam esa 0 ga teng;
- sonning millionlar sinfi qiymati 023 ga teng, chunki millionlar sinfi o‘ngdan boshlab 9-, 8- va 7-o‘rinda turgan raqamlar guruhi va qiymati 023 ga teng;

Matematikada xohlagan uzunlikdagi sonni yaratsa bo‘ladi, bunga hech qanday chegara mavjud emas. Lekin hayotda ishlatiladigan sonlar qaysidir ma‘noda chegaralangan, ya’ni juda katta sonlarni ishlatishning zarurati yo‘q.

Ayrim dasturlash tillarida sonlarni «_» (underscore) belgisi yordamida sinflarga ajratib ham yozish mumkin. Masalan, **Python** dasturlash tilida 12345678 sonni sinflarga ajratib yozib ekranga chiqarish quyidagicha:

```
printf("%u", 12345678); // 12345678
```

```
print(12_345_678) # 12345678
```

Ko‘rib turganingizdek **C** dasturlash tilida ushbu imkoniyat mavjud emas. **Python** dasturlash tilida esa, sonlarni sinflarga ajratib yozish, dasturdagi sonlarni o‘qilishini ancha osonlashtiradi.

Mavzuda doir masalalar:

## Sonlarning o‘qilishi

Oldingi mavzuda keltirilgan jadvalni yod olish shart. Chunki ushbu jadvaldan biz sonlarni to‘g‘ri o‘qish hamda yozishda foydalanamiz.

Agar biz 3 xonagacha bo‘lgan sonlarni to‘g‘ri yozishni o‘rgansak, katta sonlarni be’malol yoza olamiz. 3 xonali sonlarni yozish uchun bizga quyidagi jadval as qotadi:

| Xona raqami | Yuzlar | O‘nlar | Birlar |
|---|---|---|---|
| 0 | | | no‘l |
| 1 | bir yuz | o‘n | bir |
| 2 | ikki yuz | yigirma | ikki |
| 3 | uch yuz | o‘ttiz | uch |
| 4 | to‘rt yuz | qirq | to‘rt |
| 5 | besh yuz | ellik | besh |
| 6 | olti yuz | oltmish | olti |
| 7 | yetti yuz | yetmish | yetti |
| 8 | sakkiz yuz | sakson | sakkiz |
| 9 | to‘qqiz yuz | to‘qson | to‘qqiz |

Jadvaldan foydalanish, unchalik murakkab emas. 3 xonagacha bo‘lgan sonni olamiz va yuzlar, o‘nlar va birlar xonasida turgan raqamni jadvalda turgan so‘z bilan almashtiramiz:

- agar son 1 xonali bo‘lsa, yuqoridagi jadvaldan birlar ustunidan raqamga mos qiymatni o‘zini yozamiz: 0 – *no‘l*; 5 – *besh;*
- agar son 2 xonali bo‘lsa, dastlab o‘nlar xonasidagi raqamga mos qiymatni, keyin birlar xonasiga mos qiymatni olib yozamiz: 26 – *yigirma olti* (o‘nlar ustunidan 2 ga mos qiymat – *yigirma*, birlar ustunidan 6 ga mos qiymat – *olti*);
- agar son 3 xonali bo‘lsa, mos ravishda yuzlar, o‘nlar va birlar ustunidan mos qiymatlarni olib yozamiz: 152 – *bir yuz ellik ikki*, 568 – *besh yuz oltmish sakkiz;*

- agar son o‘rtasida yoki oxirida 0 raqami qatnashsa ushbu 0 raqami o‘rnida hech narsa yozilmaydi: 50 – *ellik* (*ellik no‘l* – xato), 100 – *bir yuz yoki yuz* (*yuz no‘l no‘l* – xato), 207 – *ikki yuz yetti* (*ikki yuz no‘l yetti* – xato), 970 – *to‘qqiz yuz yetmish* (*to‘qqiy yuz yetmish no‘l* – xato).

Agar 3 xonali songacha bo‘lgan sonlarni yozishni o‘rgangan bo‘lsangiz, endi katta sonlarni yozishni o‘rganamiz. Buning uchun quyidagi 3 ta qadamni bajarishimiz lozim:

1. Berilgan sonni sinflarga ajratamiz (oldingi mavzudagi qoida asosida).
2. Har bir sinfni 3 xonali songacha bo‘lgan sonlarni yozish qoidasi asosida yozib chiqamiz.
3. Har bir sinfning yozilgan qiymatidan keyin sinf nomini qo‘yamiz va natijalarni birlashtiramiz. Birlar sinfi va 3 ta raqam 0 ga teng bo‘lgan barcha sinf nomlarini tashlab ketamiz.

Masalan bizga 52360501000956 soni berilgan bo‘lsin. Ushbu sonni matn ko‘rinishiga o‘tkazish uchun yuqorida keltirilgan amallarni bajaramiz:

1. Sonni sinflarga ajratamiz: 52 360 501 000 956 .
2. Har bir sinfni yozilish shakliga o‘tkazamiz: 52 – ellik ikki (*trillionlar* sinfi) 360 – uch yuz oltmish (*milliardlar* sinfi) 501 – besh yuz bir (*millionlar* sinfi) 000 – no‘l (*minglar* sinfi) 956 – to‘qqiz yuz ellik olti (*birlar* sinfi).
3. Har bir sinf natijasidan keyin sinf nomini qo‘yib birlashtiramiz: ellik ikki *trillion* uch yuz oltmish *milliard* besh yuz bir *million* to‘qqiz yuz ellik olti.

Natijada minglar va birlar sinfini tashlab ketdik. Chunki minglar sinfida barcha raqamlar 0 ga teng. Birlar sinfi hech qachon yozilmaydi. Ya’ni «ellik ikki *trillion* uch yuz oltmish *milliard* besh yuz bir *million* ~~no‘l *ming*~~ to‘qqiz yuz ellik olti ~~*bir*~~» ko‘rinishida yozsak, xato hisoblanadi.

Ko‘p xonali sonlarda ham, yuz soniga o‘xshab «bir» so‘zini tashlab yozish mumkin.

| Son | Yozilishi |
|---|---|
| 1 000 | ming yoki bir ming |
| 1 000 000 | million yoki bir million |
| 1 000 000 000 | milliard yoki bir milliard |
| 1 000 000 000 000 | trillion yoki bir trillion |
| 1 000 000 000 000 000 | kvadrillion yoki bir kvadrillion |

Agar, sizda sonlarning yozilishi haqida tasavvur paydo bo‘lgan bo‘lsa, quyidagi havoladan foydalanib, har xil sonlar ustida tajribalar o‘tkazish bilan bilimingizni yanada mustahkamlashingiz mumkin:

«Raqamlar» mavzusida berilgan **C** dasturlash tilidagi printf funksiyasi aslida 0 dan 4 294 967 296 gacha bo‘lgan sonlarni ekranga chiqara oladi. Masalan, **C** dasturlash tilida 73 709 551 616 sonini ekranga chiqarmoqchi bo‘lsak, ekranda 695 107 584 kabi natijani ko‘ramiz:

```
printf("%u", 73709551616); // 695107584
```

Ko‘rib turganingizdek dastur kodida yozilgan son boshqa, ekranga chiqarilgan son boshqa. Buning sababi dastur ishlayotgan qurilma bilan bog‘liq. Hozircha bizga nimaga aynan bunday bo‘lgani muhim emas, biz faqat **C** dasturlash tilida sonlar bilan ishlayotganda shunday chegara borligini inobatga olib, kod yozsak yetarli.

Agar **C** dasturlash tilida katta sonlarni ekranga chiqarmoqchi bo‘lsak, printf funksiyamizda «%u» o‘rniga «%llu» qo‘yamiz va son oxirida LL qo‘shimchasini qo‘shamiz. **Python** dasturlash tilida hech narsa o‘zgarmaydi.

```
printf("%llu", 73709551616LL); // 73709551616
```

```
print(73709551616) # 73709551616
```

Ushbu o‘zgarish orqali **C** dasturlash tilida biz ekranga 18 446 744 073 709 551 616 gacha bo‘lgan sonlarni chiqara olamiz. Agar sonimiz berilgan sondan katta bo‘lsa, yana noto‘g‘ri natija chiqadi.

Umuman **C** dasturlash tili qurilmaga qarab 0 dan 4 294 967 296 yoki 18 446 744 073 709 551 616 songacha bo‘lgan (oxirgi sonlar kirmaydi) sonlar bilan hisob-kitob qila oladi. **Python** dasturlash tilida esa sonlar bilan ishlaganda deyarli hech qanday chegara mavjud emas.

Shu paytgacha faqat ekranga sonlarni chiqardik. Lekin dasturlar bilan ishlash mobaynida nafaqat sonlarni, balki so‘z va matnlarni ham ekranga chiqarishga to‘g‘ri keladi. Masalan «Salom ITBoom» matnini ekranga chiqarmoqchi bo‘lsak, quyidagicha yozamiz:

```
printf("Salom ITBoom"); // Salom ITBoom
```

```
print("Salom ITBoom") # Salom ITBoom
```

Matnlarni ekranga chiqarish xuddi sonlarni ekranga chiqarishga o‘xshaydi. Faqat **C** dasturlash tilida hech qanday %u, %llu ishlatmasdan, ekranga chiqarmoqchi bo‘lgan matnning o‘zini yozamiz.

Mavzuga doir masalalar:

## Rim raqamlari

*Ushbu mavzuda qo‘shish, ayirish amallari ishlatilgan. Agar sizga hisob-kitoblar og‘irlik qilsa, hozircha kalkulyatordan foydalanib turishingiz mumkin. Biz sonlarni qo‘shish va ayirishni «Butun sonlar» bobida o‘rganamiz.*

Rim raqamlari qadimgi rimliklar tomonidan qo‘llanilgan bo‘lib, hozirgi kunda ham asrlar, asar boblari, oylarning raqamlari, tartib sonlarini belgilash kabi hollarda qo‘llaniladi.

Rim raqamlari 7 ta lotin harflari I, V, X, L ,C, D, M dan iborat bo‘lib, ularning qiymatlari quyidagi jadvalda keltirilgan:

| I | V | X | L | C | D | M |
|---|---|---|---|---|---|---|
| 1 | 5 | 10 | 50 | 100 | 500 | 1000 |

Ayrim rim raqamlari boshqa rim raqamlari oldidan kelishi mumkin va u holatda qiymatlar ayiriladi:

| Rim raqami | IV | IX | XL | XC | CD | CM |
|---|---|---|---|---|---|---|
| Ifoda | 5 - 1 | 10 - 1 | 50 - 10 | 100 - 10 | 500 - 100 | 1000 - 100 |
| Qiymat | 4 | 9 | 40 | 90 | 400 | 900 |

Yuqoridagi jadvaldan tashqari, I, X, C va M rim raqamlari son ichida ko‘pi bilan uch marotaba ketma-ket kelishi mumkin va ularning qiymatlari qo‘shiladi:

| II | III | XX | XXX | CC | CCC | MM | MMM |
|---|---|---|---|---|---|---|---|
| 1+1 | 1+1+1 | 10+10 | 10+10+10 | 100+100 | 100+100+100 | 1000+1000 | 1000+1000+ +1000 |
| 2 | 3 | 20 | 30 | 200 | 300 | 2000 | 3000 |

Va oxirgi qoida yuqoridagi uchta jadvalda keltirilgan qiymatlar rim raqami ichida kelsa ularning qiymatlari qo‘shiladi.

*Qiziq fakt: rim raqamlarida 0 soni mavjud emas. Ushbu son o‘rnida «nulla» («nullus»), «nihil» («nil») so‘zlari (lotin tilida «hech narsa» ma’nosini anglatadi) qo‘llanilgan. Dasturlash tillaridagi «null», «nil» kalit so‘zlari ham hech narsa ma’nosida keladi.*

Aytaylik, bizga MMMDCCCXCIX rim raqami berilgan bo‘lsin. Uning son qiymatini topish uchun yuqoridagi uchta jadvaldan pastgisidan boshlab yuqoridagisiga qarab qiymatlarni almashtirishni boshlaymiz:

- son boshida uchinchi jadvaldagi MMM rim raqami bor, demak MMM = 3000;
- keyingisi, birinchi jadvalda D = 500;
- keyingisi, uchinchi jadvalda CCC = 300;
- keyingisi, ikkinchi jadvalda XC = 90;
- va oxirgisi ham ikkinchi jadvalda IX = 9.

Olingan barcha qiymatlarni qo‘shamiz: 3000 + 500 + 300 + 90 + 9 = 3899. Demak, MMMDCCCXCIX rim raqami 3899 qiymatga teng ekan.

Endi aksincha ya’ni sondan rim raqamiga o‘tkazmoqchi bo‘lsak, oldin sonni yuqoridagi uchta jadval qiymatlari yig‘indisi shakliga o‘tkazib olamiz va qiymatlarga mos rim raqamlarini qo‘yamiz. Masalan, 2689 sonini rim raqamiga o‘tkazish tartibi quyidagicha:

- dastlab minglar xonasiga qaraymiz. Bizning holatda 2 raqami turibdi. Demak, bu 2000 degani (minglar xonasida turgani uchun) va rim raqamida MM;
- endi yuzlar xonasiga qaraymiz. Bizning holatda 6 raqami turibdi va bu 600 degani. Lekin yuqoridagi jadvallarda 600 yo‘q. Shuning uchun 600 sonini oxirgi qoida asosida ikkita son yig‘indisi ko‘rinishida ifodalashimiz kerak. Demak, 600 soni 500 + 100 bo‘ladi va rim raqamida DC;
- o‘nlar xonasiga o‘tamiz. Bizni holatda 8 turibdi va bu 80 degani. 80 soni ham jadvallarimizda yo‘q, lekin jadvallarda mavjud ikkita son ko‘rinishida ifodalashimiz mumkin: 50 + 30 va bu rim raqamida LXXX bo‘ladi;
- va oxirgisi birlar xonasi va unda 9 raqami turibdi. 9 rim raqamida ikkinchi jadvalga asosan IX ga teng.

Demak, 2689 soni rim raqamida MMDCLXXXIX bo‘lar ekan, ya’ni minglar xonasidan boshlab hosil bo‘lgan rim raqamlarini to birlar xonasigacha birlashtirib yozamiz.

Agar rim raqamlari ishtirokidagi amallar qanday bajarilishi haqida tasavvur paydo bo‘lgan bo‘lsa, quyidagi havolaga o‘tib, har xil sonlar ustida tajriba o‘tkazib, bilimingizni yanada mustahkamlashingiz mumkin:

Mavzuga doir masalalar:

## Musbat va manfiy sonlar

Hayotda sanashda ishlatiladigan barcha sonlarimiz bu – musbat sonlar. Ya’ni 1, 2, 3 va h.k. Musbat sonlari +1, +2, +3 va h.k. kabi ham ifodalanadi, lekin ko‘p hollarda plyus (+) belgisi yozilmaydi.

Manfiy sonlar musbat sonlarga qarama-qarshi sonlar bo‘lib, plyus (+) belgisi o‘rnida minus (-) belgisi qo‘yib yozilgan sonlardir. Ular -1, -2, -3, -4, va h.k.

Manfiy sonlar haqida kitob davomida ma’lumotlar berilgan. Hozircha musbat sonlar: 1, 2, 3, ... va manfiy sonlar: -1, -2, -3 ... ekanligini bilishning o‘zi yetarli.

> *Qiziq fakt: Dastlab manfiy sonlardan (taxminan VII asrda) Hindistonda qarzlar va yetishmovchilikni ifodalashda qo‘llashgan.*

**C** dasturlash tilidagi manfiy sonlarni ekranga chiqarish uchun «%d» va «%lld» lardan foydalaniladi va manfiy sonlar matematikada qanday yozilsa, dasturlash tillarida ham xuddi shunday yoziladi.

E’tibor bering, siz ekranga chiqarmoqchi bo‘lgan natijangizdan kelib chiqib, **C** dasturlash tilida endi «%u», «%llu», «%d» va «%lld» lardan foydalanasiz. Bularning asosiy farqi «Sonlarning o‘qilishi» mavzusida yozilganidek, ekranga chiqariladigan sonning chegaralarida:

| | Dan | Gacha |
|---|---|---|
| %d | -2 147 483 648 | 2 147 483 648 |
| %u | 0 | 4 294 967 296 |
| %lld | -9 223 372 036 854 775 808 | 9 223 372 036 854 775 808 |
| %llu | 0 | 18 446 744 073 709 551 616 |

**Python** dasturlash tilida ancha oson, print funksiyasining o‘zi hech qanday o‘zgarishlarsiz musbat va manfiy sonlarni ekranga chiqara oladi.

```
printf("%d", -1522); // -1522
```

```
print(-1522) # -1522
```

```
printf("%lld", -84154787445LL); // -84154787445
```

```
print(-84154787445) # -84154787445
```

Mavzuda doir masalalar:

## Sonlar o‘qi

Bitta to‘g‘ri chiziq olamiz va uning teng uzunlikdagi qismlarga ajratamiz:

Endi shu kesimlarning o‘rtasiga 0 ni qo‘yamiz va o‘ngga qarab sanoq: 1,2,3, … sonlarini, chap tarafiga esa -1, -2, -3, … sonlarini yozib chiqamiz:

Hosil bo‘lgan shakl **sonlar o‘qi** deyiladi. Ko‘rib turganingizdek manfiy sonlar 0 dan chapda, musbat sonlar esa 0 dan o‘ng tarafda joylashgan.

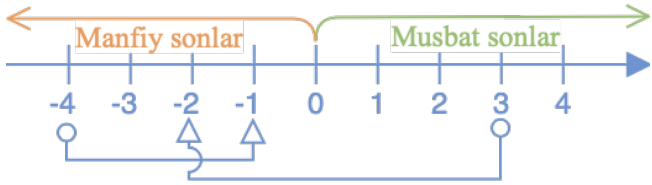
Ba’zida sonlar o‘qini birorta songa karrali sonlar ko‘rinishida ham ifodalash mumkin (karra so‘zi haqida sal keyinroq), ya’ni har bir kesimlarni 1 qiymatga emas 1 dan farqli qiymatga oshirib yozamiz:


Ushbu sonlar o‘qining har bir kesimi 4 ta birlikka oshirilgan.

Sonlar o‘qi keyingi mavzularimizda juda as qotadi. Umuman sonlar ustida bajariladigan qo‘shish, ayirish, sonning absolyut qiymati kabi amallarni sonlar o‘qi yordamida tushuntirish oson bo‘ladi. Bundan tashqari sonlarni taqqoslashni ham sonlar o‘qi yordamida tushuntirish juda oson. Masalan, bizga ikkita $a$ va $b$ noma’lum son berilgan bo‘lsin. Agar $a$ soni sonlar o‘qida $b$ sonidan chapda joylashgan bo‘lsa, demak $a$ soni $b$ sonidan kichik bo‘ladi va $a < b$ ko‘rinishida yoziladi. Agar $a$ soni $b$ sonidan sonlar o‘qida o‘ngda joylashgan bo‘lsa, demak $a$ soni $b$ sonidan katta bo‘ladi va $a > b$ ko‘rinishida yoziladi. Agar ikkala son bir-biriga teng bo‘lsa $a = b$ ko‘rinishida yoziladi.

*Yuqorida son o‘rnida kichik lotin* $a,b$ *harflarini ishlatdik. Bu degani* $a,b$ *harflari o‘rnida xohlagan son qo‘yilsa ham qonuniyat o‘zgarmasligini anglatadi va noma’lum son deyiladi. Bundan keyin agar ifodalarda lotin harflar kelib qolsa bilingki, harf o‘rnida siz xohlagan sonni qo‘yishingiz mumkin. Ayrim hollarda harflar o‘rnida qanday sonlar kelishi aniq ko‘rsatilgan bo‘ladi.*

Ushbu taqqoslash usulidan xulosa chiqarsak, 0 dan kichik har qanday son manfiy son $(a<0)$, 0 dan katta har qanday son esa musbat son $(a>0)$ ekanligi kelib chiqadi.




*-4 soni -1 sonidan kichik, chunki chapda joylashgan.*
*3 soni -2 sonidan katta, chunki o‘ngda joylashgan.*

Dasturlash tillarida bitta son va matnni chiqarishni ko‘rib chiqdik. Endi esa, dasturlash tillarida bir nechta sonni ekranga chiqarishni ko‘rib chiqamiz. Aslida u siz o‘ylagancha-lik murakkab emas.

**C** dasturlash tilidagi «%d», «%lld», «%u», «%llu»lardan foydalanib ekranga xohlagancha sonni chiqarishimiz mumkin. **Python** dasturlash tilida esa oddiy print funksiyasining o‘zida barcha sonlarni ketma-ket vergul bilan ajratgan holda yozamiz:

```
printf("%d", -1); // -1
```

```
print(-1) # -1
```

```c
printf("%d %u", -5, 7); // -5 7
```

```python
print(-5, 7) # -5 7
```

```c
printf("%d %u %lld", -55, 17, 4532984684LL);
// -55 16 4532984684
```

```python
print(-55, 17, 4532984684)
# -55 17 4532984684
```

Ikkala dasturlash tilida chiqarish funksiyalari natijalari bir xil bo‘lib, ular ekranda sonlarni funksiyalardan yozilgan ketma-ketlikda bo‘shliq (" ") belgisi bilan ajratilgan holda chiqaradi. **C** dasturlash tilida nechta sonni ekranga chiqarmoqchi bo‘lsak, printf funksiyasi birinchi parametrida shuncha son turiga mos «%d», «%lld», «%u», «%llu»larni yozishimiz shart.

Mavzuda doir masalalar:

## Sonlar to‘plami

Matematikada sonlar yozilishi shakli va qiymatidan kelib chiqib bir nechta to‘plamlarga ajratiladi. Ayrim son to‘plamlari o‘z ichiga boshqa son to‘plamlarini ham olishi mumkin.

Garchi dasturlash sonlar to‘plamiga to‘g‘ridan to‘gri bog‘liq bo‘lmasa-da, masalalar ishlaganda sonlar to‘plamiga murojaatlar bo‘ladi.

Quyidagi masalalarga e’tibor bering: berilgan ikkita natural sondan kattasini toping yoki haqiqiy sonning butun qismini va kasr qismi yig‘indisini hisoblang, shu kabi masalalarni yechishda sizda sonlar to‘plamiga qaysi sonlar kirishi haqida tushuncha bo‘lishi lozim.

## Natural sonlar to‘plami

Sanashda ishlatiladigan barcha sonlar to‘plamiga **natural sonlar to‘plami** deyiladi va ushbu sonlar to‘plamiga 1, 2, 3, 4, … va hakozo sonlar kiradi. Natural sonlar to‘plamini $\mathbb{N}$ yoki $\mathbb{N}^+$ lotin katta harfi bilan belgilanadi. Eng kichik natural son 1 ga teng. Eng katta natural son mavjud emas, ya’ni natural sonlar 1 dan boshlanib cheksizlikkacha ($\infty$) bo‘lgan sonlardir. Qisqacha quyidagicha yoziladi:

$$\mathbb{N} = \mathbb{N}^+ = \{1,2,3,4, ...,\infty\}$$

Natural sonlar to‘plami 0 sonidan ham boshlanishi mumkin va ushbu sonlar to‘plami $\mathbb{N}^0$ yoki $\mathbb{N}_0$ ko‘rinishida belgilanadi.

$$\mathbb{N}^0 = \mathbb{N}_0 = \{0,1,2,3,4, ...,\infty\}$$

## Butun sonlar to‘plami

Sonlar o‘qida joylashgan barcha sonlar to‘plamiga **butun sonlar to‘plami** deyiladi. Umuman barcha manfiy, barcha musbat (natural sonlar) va 0 sonidan iborat sonlar to‘plamidir. Butun sonlar to‘plamini $\mathbb{Z}$ lotin katta harfi bilan belgilanadi. Ushbu sonlar to‘plamida eng kichik butun son ham eng katta butun son ham mavjud emas. Qisqacha quyidagicha yoziladi:

$$\mathbb{Z} = \{-\infty,...,-4,-3,-2,-1,0,1,2,3,4, ...,\infty\}$$

## Ratsional sonlar to‘plami

O‘z ichiga barcha butun sonlarni hamda $\frac{a}{b}$ ko‘rinishida ifodalash mumkin bo‘lgan barcha sonlar to‘plamidan (oddiy kasrlar deyiladi. Kasrlar haqida keyinroq) iborat sonlarga **ratsional sonlar to‘plami** deyiladi. Bunda *a* butun, *b* esa natural

sonlardir. Ratsional sonlar to‘plamini $\mathbb{Q}$ lotin katta harfi bilan belgilanadi va qisqacha quyidagicha yoziladi:

$$\mathbb{Q}=\left\{\frac{a}{b}:a\in\mathbb{R},b\in\mathbb{N}^{+}\right\}$$

$\in$ – tegishlilik belgisi

## Irratsional sonlar to‘plami

Ratsional sonlar ko‘rinishida yozib bo‘lmaydigan sonlar to‘plamiga **irratsional sonlar to‘plami** deyiladi. Masalan **π** (pi) soni. Ushbu sonning qiymati 3.1415926… (uchta nuqta chekiz davom etishini bildiradi) ga teng va ushbu π sonini $\frac{a}{b}$ son ko‘rinishida yozib bo‘lmaydi. Shuning uchun ham bu son irratsional son hisoblanadi.

## Haqiqiy sonlar to‘plami

Haqiqiy sonlar to‘plamiga irratsional sonlar to‘plami va ratsional sonlar to‘plamidan iborat barcha sonlar kiradi. Haqiqiy sonlar to‘plami $\mathbb{R}$ lotin katta harfi bilan belgilanadi va qisqacha quyidagicha yoziladi:

$$\mathbb{R}=\{\mathbb{Q}+Irratsional\ sonlar\}$$

## Kompleks sonlar to‘plami

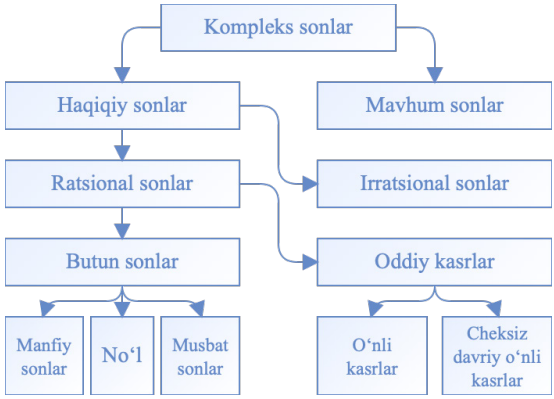
$a+bi$ ko‘rinishidagi sonlarga **kompleks sonlar** deyiladi. Bu yerda $i^2=-1$ bo‘lib, $a$ va $b$ lar haqiqiy sonlar, $i$ esa mavhum birlik. Oddiy holatda sonning kvadrati ($a^2$ ko‘rinishidagi sonlar) hech qachon manfiy son bo‘lmaydi (bu haqida keyingi bobda o‘tiladi).

Kompleks sonlar aslida biz uchun mavhum, lekin mavjud son bo‘lib, juda ko‘p masalalarni yechishda as qotadi.

Shaxsan o‘zim shuncha yildan beri dasturlar tuzib, biror marta kompleks sonlarni amalda ishlatmaganman, shuning uchun ham ushbu kitobda kompleks sonlar haqida ma’lumot berilmaydi.

> *Qiziq fakt: kompleks sonlar haqida dastlab italiyalik matematik Kardani (Gerolamo Cardano) «Buyuk san’at yoki algebraik qoidalar haqida» (Ars Magna, 1545-yil) kitobida so‘z yuritgan.*

Barcha sonlar to‘plamini quyidagicha ifodalash mumkin:




# Butun sonlar

## Butun sonlarni qo‘shish

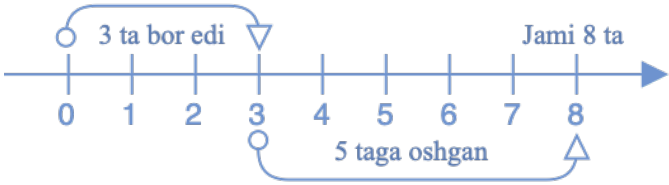
Tasavvur qiling, qo‘lingizdagi savatchada 3 ta olma (🍎🍎🍎) bor edi, uning ustiga yana 5 ta olma (🍎🍎🍎🍎🍎) qo‘shsangiz,

savatchada jami olmalar nechta bo‘ladi? Javob esa oddiy 8 ta. Ushbu masalaning matematik ifodalanishi quyidagicha:

$$3+5=8$$

Ko‘rib turganingizdek matematikada qo‘shish amali plyus (+) belgisi bilan yoziladi va biror bir son nechta birlikka ortganini bildiradi (necha marotaba bo‘lsa bu ko‘paytirish bo‘ladi, u haqida pastroqda).

Ushbu masalani sonlar o‘qi yordamida yechish quyidagicha: Dastlab savatchada mavjud bo‘lgan olmalar sonini sonlar o‘qida belgilaymiz va savatchaga nechta olma qo‘shgan bo‘lsak belgilagan sonimizda shuncha birlik o‘ngga sanaymiz. Sanashni tugatgandan keyin kelib qolgan nuqtamizdagi qiymat masalaning javobi bo‘ladi:

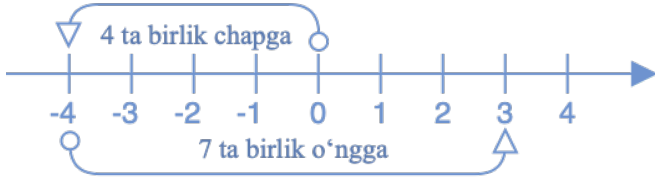



0 dan boshlab 3 ta birlik o‘ngga sanasak, savatchamizda dastlab nechta olma borligini aniqlaymiz. Keyin kelib qolgan nuqtamizdan 5 ta birlik yana o‘ngga sanaymiz. Sanashni tugatgandan keyingi 8 qiymati savatchada jami nechta olma borligini anglatadi.

Yuqoridagi qoidada manfiy songa musbat sonni qo‘shsak ham o‘rinli bo‘ladi, faqat manfiy sonda 0 dan boshlab chapga sanaymiz. Masalan:

$$-4+7=3$$

Sonlar o‘qida ifodalasak:




Yig‘indida ikkita sonning o‘rni almashishi natijani o‘zgartirmaydi. Ushbu qoida ko‘p xonali sonlar uchun ham o‘rinli.

Yuqoridagi ifodalarni quyidagicha yozsa bo‘ladi:

$$5+3=8 \quad va \quad 7+(-4)=7-4=3$$

Ikkinchi ifoda sonlarni ayirish ifodasi bo‘lgani uchun uni keyingi mavzuda o‘rganamiz.

> *Qiziq fakt: + va – belgilari nemislarning «algebrachilar» maktabida o‘ylab topilgan bo‘lib, dastlab Jon Vidman (Johannes Widmann)ning «Barcha savdogarlar uchun tez va aniq hisob» (nemischa «Behende und hüpsche Rechenung auff allen Kauffmanschafft») kitobida chop qilingan. Ushbu belgilargacha qo‘shish amali* ***p*** *(plus) va lotincha* ***et*** *(va), ayirish amali esa* ***m*** *harfi bilan belgilangan.*

Endi bizga uchta 1, 3, 6 sonlari berilgan va biz shu uchala sonni yig‘indisini hisoblashimiz kerak bo‘lsin. Buning uchun xohlasak oldin 1 ga 3 ni qo‘shib, keyin 6 ni qo‘shsak bo‘ladi: 1 + 3 = 4; 4 + 6 = 10. Yoki oldin 3 ga 6 ni qo‘shib, keyin 1 ni qo‘shsak bo‘ladi: 3 + 6 = 9; 9 + 1 = 10. Yoki 1 ga 6 ni qo‘shib, keyin 3 ni qo‘shsak ham natija o‘zgarmaydi: 1 + 6 = 7; 7 + 3 = 10. Ushbu holat **sonlarni guruhlash** deyiladi va qavslar yordamida quyidagicha yoziladi.

$$(1+3)+6=10; \quad 1+(3+6)=10; \quad (1+6)+3=10$$

> *Qavslar ichidagi ifoda har doim birinchi hisoblanadi, keyin qolgan amallar bajariladi.*

Agar bitta ifodada qavslar ichma-ich kelsa, ichkarida turgan qavsdan boshlab ifodalar hisoblana boshlaydi. Masalan:

$$3+\Big(4+\big(3+1\big)+\big(\big(4+6\big)+1\big)\Big)=3+\Big(4+4+\big(10+1\big)\Big)=$$

$$3+\big(8+11\big)=3+19=22$$

Sonlarni guruhlash, odatda hisoblashni osonlashtirish uchun kerak. Masalan, quyidagi ifoda berilgan bo‘lsin:

$$3+28+7+33+72$$

Ushbu ifodada sonlarni ketma-ket qo‘shib hisoblasak ham bo‘ladi. Lekin guruhlash usuli yordamida quyidagicha ko‘rinishga olib kelsak, hisoblash ancha osonlashadi:

$$\big(3+7\big)+\big(28+72\big)+33=\big(10\big)+\big(100\big)+33=143$$

Ko‘rib turganingizdek, guruhlangan sonlar 10, 100 bo‘ldi va oxirgi ifodani hisoblash ancha osonlashdi.

Xohlagan songa 0 ni qo‘shish ushbu son qiymatini o‘zgartirmaydi:

$$a+0=0+a=a$$

Kichik sonlarni sonlar o‘qi yordamida qo‘shish oson, lekin sonlar kattalashsa sonlar o‘qi yordamida qo‘shish deyarli ilojsiz. Shuning uchun ko‘p xonali sonlarni qo‘shishning ustun shakli mavjud bo‘lib quyidagi havola orqali o‘rganib olishingiz mumkin:

Agar ikkita sonni ustun shaklida qo‘shishni o‘rgangan bo‘lsangiz, quyidagi havolaga o‘tib, ixtiyory ikkita sonni ustun shaklida qo‘shish qadamlari ketma-ketligini o‘rganib, bilimingizni mustahkamlashingiz mumkin:

Dasturlash tillarida ikkita sonni qo‘shish uchun + (qo‘shish operatori)dan foydalaniladi va + hamda qavslar ifodada qanday kelsa, dasturlash tillarida ham xuddi shunday yoziladi. Masalan:

$$1+(2+8)+9999999999999$$

Ifodani dasturlash tillarida hisoblab ekranga chiqarmoqchi bo‘lsak quyidagicha yozamiz:

```
printf("%llu", 1 + (2 + 8) + 9999999999999LL);
// 10000000000010
```

```
print(1 + (2 + 8) + 9999999999999)
# 10000000000010
```

Faqat **C** dasturlash tilida, ifoda natijasi «Musbat va manfiy sonlar» mavzusida berilgan chegaradan oshib ketsa javob noto‘g‘ri chiqadi.

Endi bizga $3+4$ ifoda berilgan bo‘lsin. Ushbu ifodani dasturlash tillarida hisoblab ekranga natijani chiqarmoqchi bo‘lsak quyidagicha yozamiz:

```
printf("%d", 3 + 4); // 7
```

```
print(3 + 4) # 7
```

Dastur ishga tushganda ekranda bitta son, ya’ni faqat 7 soni ko‘rinadi. Bu to‘g‘ri, chunki $3+4=7$ ga teng va bitta 7 soni dastur foydalanuvchisi uchun tushunarsiz.

Agar biz natijani foydalanuvchiga tushunarli bo‘lishini xohlasak faqat ifoda natijasini emas, natija yoniga qo‘shimcha matnlarni ham chiqarishimiz kerak. Masalan:

Natija: 7 ga teng

ko‘rinishida ekranga chiqarmoqchi bo‘lsak quyidagicha yozamiz:

```c
printf("Natija: %d ga teng", 3 + 4);
// Natija: 7 ga teng
```

```python
print(f"Natija: {3 + 4} ga teng")
# Natija: 7 ga teng
```

**C** dasturlash tilida printf funksiyaning birinchi parametrida nima yozsangiz barchasi ekranga chiqadi, faqat birinchi parametr ichidagi «%d», «%u», «%lld», «%llu»lar o‘rniga keyingi parametrlarda kelgan ifodalar qiymatlari mos ravishda almashtiriladi.

**Python** dasturlash tilida esa matn boshiga f harfini qo‘yib, matn ichida {} sistemali qavslarda (ayrim manbalarda jingalak, figurali qavslar deyilgan) ifodalarni to‘g‘ridan to‘g‘ri yozish mumkin. Ifoda natijalari systemali qavslar bilan almashtirilib ekranga chiqariladi.

Agar:

3 + 4 ifoda natijasi 7 ga teng

ko‘rinishida chiqarmoqchi bo‘lsak quyidagicha yozamiz:

```c
printf("%d + %d ifoda natijasi %d ga teng",
    3, 4, 3 + 4); // 3 + 4 ifoda natijasi 7 ga teng
```

```python
print(f"{3} + {4} ifoda natijasi {3 + 4} ga teng")
# 3 + 4 ifoda natijasi 7 ga teng
```

Mavzuga doir masalalar:

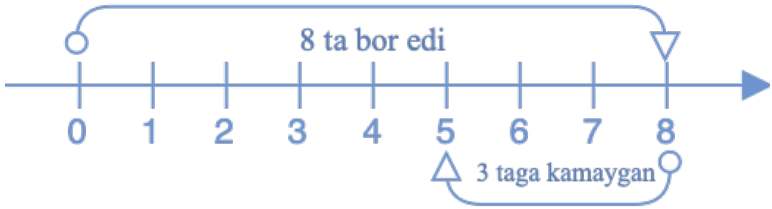
## Butun sonlarni ayirish

Qo‘shish amali birorta narsaning nechtaga ortganini bildirsa, ayirish amali nechtaga kamayganini anglatadi. Masalan,

sizda 8 ta olma (🍎🍎🍎🍎🍎🍎🍎🍎) bor edi. Shundan 3 tasini (🍎🍎🍎) yedingiz, ya'ni uchtaga kamaydi. Qolgan olmalar esa 5 ta (🍎🍎🍎🍎🍎). Ushbu masalaning matematik ifodalanishi quyidagicha:

$$8-3=5$$

Ko'rib turganingizdek ayirish amali minus (-) belgisi bilan yoziladi. Sonlar o'qida ushbu ifoda quyidagicha bo'ladi:

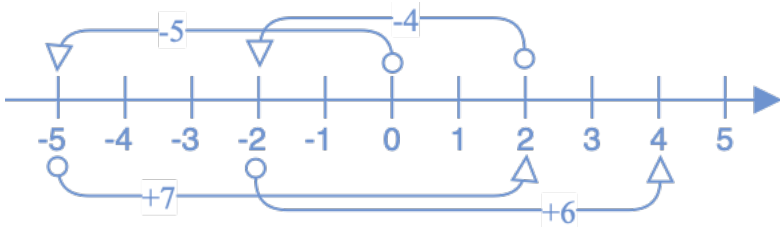



0 soni turgan nuqtadan boshlab, ifodadaning birinchi turgan soni musbat bo'lsa o'ngga qarab, manfiy bo'lsa chapga qarab shuncha birlik sanaymiz. Kelib qolgan nuqtadan boshlab, agar qo'shish amali bo'lsa o'ngga, ayirish amali bo'lsa chapga qarab, to ifoda tugamaguncha, shuncha birlik sanaymiz. Oxirida kelib qolgan nuqta qiymati ifoda natijasi bo'ladi.

Quyidagi ifodani sonlar o'qi yordamida hisoblaylik:

$$-5+7-4+6=?$$

Birinchi son manfiy, demak 0 dan boshlab 5 birlik chapga, keyin 7 birlik o'ngga, 4 birlik chapga va oxirgisi 6 birlik o'ngga:

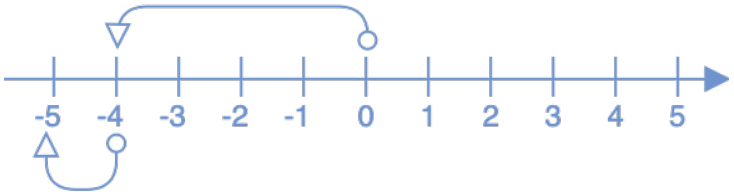


Ko‘rib turganingizdek yuqoridagi ifoda qiymati 4 ga teng ekan.

Sonlar o‘qi yordamida quyidagi ifodani hisoblaymiz:

$$-4-1=?$$

Birinchi son manfiy va ikkinchi ayirish amali. Demak, 0 dan boshlab, dastlab 4 ta birlik, keyin yana 1 ta birlik chapga yuramiz:




Javob -5 ekan. Ya’ni 4 ga 1 ni qo‘shib, oldiga minus (-) ishorasini yozsak ham natija bir xil bo‘lar ekan. Demak, quyidagi ifodalar teng kuchli:

$$-4-1=-(4+1)=-5$$

Qo‘shish amalidagi guruhlab hisoblash, ayirish amalida boshqacha bo‘ladi. Yuqoridagi ifodadan xulosa qilsak, oldida minus ishorasi bor qavs ochilganda, qavs ichidagi barcha ishoralar qarama-qarshisiga (minuslar plyuslarga, plyuslar minuslarga) o‘zgartiriladi va qavsni ochish deb nomlanadi. Agar minus ishorasini qavsdan tashqariga chiqarmoqchi bo‘lsak ham qavs ichidagi ishoralar qarama-qarshisiga almashtiriladi va qavsga olish yoki guruhlash deyiladi.

Har qanday $a$ va $b$ noma’lum sonlar uchun quyidagi qonuniyat o‘rinli bo‘ladi.

$$-(a)=(-a)=-a \qquad -(a+b)=-a-b$$

$$-(-a)=a \qquad -(a-b)=-a+b$$

$$a+(-b)=a-b \qquad -(-a+b)=a-b$$

$$a-(-b)=a+b \qquad -(-a-b)=a+b$$

Ushbu $-a-b=-(a+b)$ qonuniyatga e'tibor bering. Bu yerda manfiy sondan birorta sonni ayirmoqchi bo'lsak, oldin ularni qo'shib oldiga minus ishorasini qo'ysak ham qiymat o'zgarmas ekan, yuqoridagi misolda ko'rganimiz kabi.

Yuqoridagi qonuniyatlar ifodalar ichida kelib qolsa, osonroq yo'l bilan hisoblash uchun juda as qotadi. Shuning uchun ushbu qonuniyatlarni eslab qolish shart. Ushbu qonuniyatlar faqat ikkita son uchun emas, qavs ichidagi barcha sonlar uchun o'rinli. Masalan:

$$-(a+b-c+d-e)=-a-b+c-d+e$$

Yana bir misol:

$$-5-(4-8)+(-9-1)=-5-(-4)+(-10)=-5+4-10=-11$$

*Yana bir marotaba esga olamiz, qavslar ichidagi ifoda har doim birinchi bo'lib hisoblanadi, keyin qolgan amallar bajariladi. Agar qavslar ichma-ich bo'lsa, ichkaridagi qavslardan boshlab hisoblash amalga oshiriladi.*

Masalan:

$$8-\Big(4+(2-6)-\big((3-4)+8\big)\Big)=8-\Big(4+(-4)-\big((-1)+8\big)\Big)=$$

$$8-\Big(4-4-(-1+8)\Big)=8-\big(0-(7)\big)=8-(0-7))=8-(-7)=$$

$$8+7=15$$

Ko‘p xonali sonlarni ayirganda qo‘shish amaliga o‘xshab ustun shaklidan foydalanamiz. Faqat sonlarni ayirishda xotirada son saqlab turmaymiz, aksincha qo‘shni xonalardan qarz olamiz (to‘g‘ri ma’noda qarz olamiz :) ).

Ikki natural sonni ayirishni quydagi havoladagi video oraqali o‘rganishingiz mumkin:

*Qo‘shimcha tushuntirish: videoni ko‘rgan bo‘lsangiz qo‘shni xonadan qarz olib oldingi xonaga 10 qo‘shildi. Nega aynan 10 qo‘shiladi? Chunki sondagi yonma-yon xonalar nisbati o‘n martaga farq qiladi, ya’ni minglar xonasidan 1 degani bu aslida 1000 soni degani. Shu minglar xonasidan qarz olgan ming sonimizni yuzlar xonasiga 10 qilib qo‘shamiz. Chunki 10 ta 100 soni 1000 qiymatini beradi. Shu qoida boshqa yonma-yon xonalar uchun ham o‘rinli. Masalan, yuzlar xonasidan 1 bu aslida 100 degani va biz shu 100 ni o‘nlar xonasiga 10 qilib qo‘shamiz. Chunki 10 ta 10 – bu 100 ga teng.*

Agar ko‘p xonali sonlarni ayirishni tushungan bo‘lsangiz, quyidagi havolaga o‘tib har xil sonlarni kiritib, ikkita sonni ayirishning qadamlari ketma-ketligini o‘rganib, bilimingizni yanada mustahkamlashingiz mumkin:

Dasturlash tillarida ayirish qo‘shish amali bilan bir xil bo‘lib, ifoda matematikada qanday yozilsa dasturlash tillarida ham xuddi shunday yoziladi.

Bizga quyidagi ifoda berilgan bo‘lsin:

$$256456-\left(125487+\left(22-8889\right)\right)$$

Ushbu ifodani **C** va **Python** dasturlash tillari yordamida hisoblatib natijasini ekranga chiqarmoqchi bo‘lsak quyidagicha yozamiz:

```
printf("%d", 256456 - (125487 + (22 - 8889)));
// 139836
```

```
print(256456 - (125487 + (22 - 8889)))
# 139836
```

Ikkala dasturlash tilida natija bir xil va 139836 ga teng. Ushbu 139836 natijani ekranga chiqarish uchun ifodani hisoblab, keyin quyidagicha yozish ham mumkin edi:

```
printf("%d", 139836); // 139836
```

```
print(139836) # 139836
```

Lekin ushbu holatda siz ancha vaqtingizni sarflab qo‘yasiz, chunki ifodani hisoblashdan ko‘ra ushbu ifodani dasturlash tilida yozish ancha tez va qulay. Shunday ekan, keyingi masalalarda ham ifoda qanchalik sodda ko‘rinmasin ushbu ifodani dasturlash tili yordamida yozib uni dasturning o‘ziga hisoblatishga harakat qiling.

Mavzuga doir masalalar:

## Butun sonlarni ko‘paytirish

Tasavvur qiling, archalar quyidagi ko‘rinishida ekilgan bo‘lsin:

Agar sizga nechta archa ekilgan deb savol bersam, eng oson yo‘li bir boshidan boshlab barcha archalarni sanab chiqib 35 ta deysiz. Javob to‘g‘ri, lekin 35 ta archani sanashga siz ancha vaqt yo‘qotib qo‘yasiz. Agar archalarning nechta ustun va nechta qator qilib ekilgani ma’lum bo‘lsa, jami ekilgan archalar sonini topishning oson usuli bu ko‘paytirish amalidir.

Rasmdan ma’lum har bir qatorda 7 tadan archa bor ekan va bizda jami 5 qator mavjud, ya’ni:

1-qatorda: 7 ta

2-qatorda: 7 ta

3-qatorda: 7 ta

4-qatorda: 7 ta

5-qatorda: 7 ta

Endi barcha qatorlardagi archalarni qo‘shamiz va quyidagi ifoda hosil bo‘ladi:

$$7+7+7+7+7=35$$

Yoki shuning aksini qilsak, ya’ni bizda 7 ta ustun bor va har bir ustunda 5 tadan archa mavjud deb qarasak:

1-ustunda: 5 ta

2-ustunda: 5 ta

3-ustunda: 5 ta

4-ustunda: 5 ta

5-ustunda: 5 ta

6-ustunda: 5 ta

7-ustunda: 5 ta

Ushbu holatda ifoda quyidagicha bo‘ladi:

$$5+5+5+5+5+5+5=35$$

Agar ifodalarga e’tibor bersangiz, birinchi ifodada 7 soni 5 marotaba takrorlandi, ikkinchi ifodada 5 soni 7 marotaba takrorlandi, lekin qiymat o‘zgarmadi.

Umuman yuqoridagi ifodalarni qisqacha qilib quyidagicha yozsa bo‘ladi:

$$7+7+7+7+7=7\cdot 5=35 \qquad 5+5+5+5+5+5+5=5\cdot 7=35$$

7 va 5 sonlari o‘rtasida turgan belgi $\cdot$ (nuqta) matematikada ko‘paytirish belgisi deyiladi. Ba’zi manbalarda $\times$ ko‘rinishida ham keladi.

*Qiziq fakt: dastlab $\times$ ko‘paytirish belgisini Otred Uilyam (William Oughtred) 1631-yil muomalaga kiritgan. Ungacha esa M harfidan foydalanishgan.*

*Agar siz maktabda ko‘paytirish jadvalini (karra jadvalni) yodlamagan bo‘lsangiz, yodlashingizga to‘g‘ri keladi. Chunki yuqoridagi $7\cdot 5$ ifodani yoddan hisoblab ayta olishingiz kerak, 5 marta 7 raqamini qo‘shmasdan. Nimaga kerak deysizmi? Karra jadvaldagi 1 dan 9 gacha bo‘lgan ko‘paytmalardan foydalanib ko‘p xonali sonlarni bir-biriga ko‘paytiramiz.*

Yuqoridagilardan xulosa chiqarsak, $a$ sonining $b$ soniga ko‘mapytmasi bu – $b$ ta $a$ sonining yig‘indisiga teng qiymatdir:

$$a\cdot b=a+a+a+a+...+a \text{ (jami b ta a soni)}$$

Agar $b$ soni kichik bo‘lsa, yuqoridagi ifoda orqali ko‘paytmani hisoblash mumkin, lekin $b$ soni kattalashsa ko‘paytmani hisoblash juda qiyinlashadi.

$$6354\cdot 156=?$$

Yuqoridagi ifodani hisoblash uchun 6354 sonini o‘ziga 156 marta qo‘shish kerak. To‘g‘ri qo‘shib natijani chiqarsak bo‘ladi, ammo juda ko‘p vaqtimiz ketib qoladi.

Shuning uchun ko‘p xonali sonlarni ko‘paytirishda xuddi qo‘shish va ayirish kabi ustun usulidan foydalanamiz va quyidagi havolada o‘rganib olsangiz bo‘ladi:

Agar hali ham ko‘p xonali sonlarni ustun shaklida ko‘paytirishga qiynalayotgan bo‘lsangiz ushbu havola orqali xohlagan sonlarni ko‘paytirishni qadam-baqadam hisoblashni o‘rganishingiz mumkin:

Agar ikkita katta natural sonlarni ko‘paytirishni o‘rgangan bo‘lsangiz, quyidagi ifodani yoddan hisoblab ko‘ring:

$$3+5\cdot 4-7=?$$

Agar siz matematikadan xabaringiz bo‘lmasa, uni quyidagichi hisoblaysiz:

- Oldin 3 ga 5 ni qo‘shasiz: $3+5=8$
- Keyin 8 ni 4 ga ko‘paytirasiz: $8\cdot 4=32$
- Oxirida $32-7=25$. Lekin bu javob xato hisoblanadi.

Chunki ifoda ichida qo‘shish, ayirish va ko‘paytirish aralashib kelsa, oldin ko‘paytirish amallari hisoblanadi, keyin esa qo‘shish va ayirish. Demak, yuqoridagi ifoda quyidagicha hisoblanadi:

$$3+5\cdot 4-7=3+20-7=23-7=16$$

Agar ko‘paytirishlar ketma-ket kelib qolsa, siz xohlagan tartibda ko‘paytirishingiz mumkin, xuddi qo‘shish amalidagi guruhlash qoidasiga o‘xshab:

$$3\cdot 4\cdot 6=(3\cdot 4)\cdot 6=12\cdot 6=72 \text{ yoki } 3\cdot(4\cdot 6)=3\cdot 24=72$$

Agar sonni 10, 100, 1000 kabi, ya’ni oxiri faqat 0 bilan tugagan sonlarga ko‘paytirishda, natijani hosil qilish uchun

berilgan soning oxiriga shuncha 0 raqamini yozishning o‘zi kifoya. Masalan:

$$2356 \cdot 10 = 23560 \qquad 56 \cdot 1000 = 56000$$

Agar ikkita sonning oxiridagi 0 raqamlarini olib, qolgan sonlarni bir-biriga ko‘paytirib javob oxiriga ikkita sondagi barcha 0 raqamlarini yozsak ham qiymat o‘zgarmaydi.

Masalan: $270 \cdot 3200$ ifodani hisoblaylik. Bunda biz jami uchta 0 ni olib tashlab quyidagi ifodani hisoblaymiz:

$$27 \cdot 32 = 864$$

Endi 864 soni davomidan yuqorida olib qolingan uchta 0 raqamini yozamiz:

$$270 \cdot 3200 = 864000$$

Quyida natural sonlarga tegishli bo‘lgan qonuniyatlar keltirilgan $\left(a \in \mathbb{N}^+\right)$:

| | | | |
|---|---|---|---|
| $a \cdot 0 = 0$ | $a \cdot 1 = a$ | $a \cdot (-1) = -a$ | $a \cdot b = b \cdot a$ |
| $-a \cdot 0 = 0$ | $-a \cdot 1 = -a$ | $-a \cdot (-1) = a$ | $-a \cdot b = -(a \cdot b)$ |
| $0 \cdot a = 0$ | $1 \cdot a = a$ | $-1 \cdot a = -a$ | $a \cdot (-b) = -a \cdot b$ |
| | $1 \cdot (-a) = -a$ | $-1 \cdot (-a) = a$ | $-a \cdot (-b) = a \cdot b$ |

- Har qanday sonni (musbat, manfiy, kasr, haqiqiy, …) 0 ga ko‘paytirsak natija har doim 0 ga teng bo‘ladi (1-qonuniyat).
- Agar sonni 1 ga ko‘paytirsak har doim sonning o‘zi chiqadi (2-qoniniyat).
- Agar sonni -1 ga ko‘paytirsak sonning qiymati o‘zgarmaydi faqat ishorasi o‘zgaradi (3-qoniniyat).

- Agar musbat songa musbat sonni ko‘paytirsak har doim musbat chiqadi va ularning o‘rni almashishi natijani o‘zgartirmaydi (4-qonuniyat, 1-qator).
- Agar manfiy songa musbat son ko‘paytirilsa yoki musbat songa manfiy son ko‘paytirilsa natija ham manfiy bo‘ladi (4-qonuniyat, 2-,3-qatorlar).
- Agar manfiy songa manfiy son ko‘paytirilsa natija har doim musbat bo‘ladi (4-qoniniyat, oxirgi qator).

Yuqoridagi qonuniyatlar ifoda ichida ketma-ket kelsa, xuddi guruhlash usulidagiday bir chetdan boshlab hisoblanadi. Masalan:

$$-a\cdot(-b)\cdot(-c)=a\cdot b\cdot(-c)=-a\cdot b\cdot c$$

E’tibor bergan bo‘lsangiz, mavzuda manfiy songa musbat sonni ko‘paytirish haqida yozilmagan. Chunki biz yuqoridagi qonuniyatdan bilamizki, manfiy songa musbat sonni ko‘paytirsak oxirgi natija ham manfiy bo‘lar ekan: $-a\cdot b=-(a\cdot b)$.

Demak manfiy sonni musbat songa ko‘paytirmoqchi bo‘lsak, oldin ularning qiymatini ko‘paytirib natija oldiga minus ishorasini qo‘yamiz.

Biz yuqorida ikkita butun sonning ko‘paytmasi qonuniyatlarini ko‘rdik. Endi son va qavsli ifoda ko‘paytirilsa yoki qavsli ifoda bilan qavsli ifoda ko‘paytirlisa ular qanday yoyilishi (qavsdan chiqarish yoki qavsni ochish) qanday bo‘lishini o‘rganamiz.

Agar qavs oldida bitta son bo‘lsa, qavsni ochganda o‘sha son qavs ichidagi barcha sonlarga ko‘paytiriladi. Ishorasi o‘zgarishi yuqoridagi qonuniyat asosida amalga oshiriladi:

$$a\cdot(b+c)=a\cdot b+a\cdot c \qquad -a\cdot(-b-c)=a\cdot b+a\cdot c$$

$$-a\cdot(b+c)=-a\cdot b-a\cdot c \qquad -a\cdot(b-c)=-a\cdot b+a\cdot c$$

Agar qavsli ifoda qavsli ifodaga ko‘paytirilsa, birinchi qavs ichidagi har bir son (nechta bo‘lishining ahamiyat yo‘q)

keyingi qavs ichidagi barcha sonlarga ko‘paytiriladi. Ishorasi o‘zgarishi yuqoridagi qonuniyat asosida amalga oshiriladi:

$$(a+b)\cdot(c+d)=a\cdot c+a\cdot d+b\cdot c+b\cdot d$$

$$(a-b)\cdot(c+d)=a\cdot c+a\cdot d-b\cdot c-b\cdot d$$

$$(a+b)\cdot(c-d)=a\cdot c-a\cdot d+b\cdot c-b\cdot d$$

$$(a-b)\cdot(c-d)=a\cdot c-a\cdot d-b\cdot c+b\cdot d$$

*Yana bir marta eslaymiz, agar ifoda ichida qavslar kelsa oldin ushbu qavslar ichidagi ifodalar hisoblanadi. Ya’ni agar qavs ichida yana qavs va uning ichida qo‘shish, ayirish va ko‘paytirish amallari aralashgan ifoda bo‘lsa, demak siz ichki qavs ichidagi ko‘paytirish amallaridan hisoblashni boshlaysiz. Keyin ichki qavs ichidagi qo‘shish, ayirish amallari. Undan bitta oldin turgan qavs ham xuddi shu tartibda hisoblanadi.*

Masalan, quyidagi ifodani hisoblaylik:

$$-5\cdot(8-3)=?$$

Ushbu ifodani ikkita usulda hisoblash mumkin:

$$-5\cdot(8-3)=-5\cdot(5)=-5\cdot5=-25$$

$$-5\cdot(8-3)=-5\cdot8+5\cdot3=-40+15=-25$$

Ikkala usulda ham natija bir xil. 1-usulda oldin qavs ichini hisoblab, keyin ko‘paytma hisoblandi. 2-usulda oldin qavsni ochib, keyin yig‘indi hisoblandi. Qaysi usulni tanlash sizning qo‘lingizda. Agar qaysidir usul ifodani hisoblashni tezlashtirsa ushbu usuldan foydalanganingiz ma’qul.

Quyidagi ifoda berilgan bo‘lsin:

$$(9-5+2)\cdot(4-0)=?$$

Ushbu ifoda ham ikkita usulda yechiladi:

$$(9-5+2)\cdot(4-0)=(6)\cdot(4)=6\cdot 4=24$$

$$(9-5+2)\cdot(4-0)=9\cdot 4-9\cdot 0-5\cdot 4+5\cdot 0+2\cdot 4-2\cdot 0=$$

$$36-0-20+0+8-0=24$$

Ko‘rib turganingizdek ikkala usulda ham natija bir xil. Lekin ushbu ifodani 1-usulda ishlash ancha oson va biz yuqoridagi qavslarni ochish qoidalari to‘g‘ri ekanligiga ham ishonch hosil qildik.

Dasturlashda ifodalarni yozishda ko‘paytirish belgisi o‘rniga * (yulduzcha) qo‘yib yoziladi va qolgan qismi o‘zgarishsiz qoladi.

Masalan $7\cdot(4-5\cdot 2)+11\cdot(20\cdot 2-44)$ ifodaning **C** va **Python** dasturlash tillaridagi yozilishi quyidagicha:

```
printf("%d", 7 * (4 - 5 * 2) + 11 * (20 * 2 - 44));
// -86
```

```
print(7 * (4 - 5 * 2) + 11 * (20 * 2 - 44)) # -86
```

Faqat esdan chiqarmaslik lozim, **C** dasturlash tilida ifoda qiymati «Musbat va manfiy sonlar» mavzusidagi berilgan chegaradan chiqib ketib qolsa, javob noto‘g‘ri chiqadi. **Python** dasturlash tilida chegara mavjud emas.

Mavzuga doir masalalar:

## Butun sonlarni bo‘lish

Sonlarni bo‘lish amali aslida sonlarni ko‘paytirish amaliga teskari amal bo‘lib, bo‘lish birorta bir narsani teng qismlarga yoki guruhlarga ajratishni anglatadi. Masalan, 15 ta olmani 5 ta bolaga teng taqsimlasak har bir bolada 3 tadan olma bo‘ladi va matematik ifodasi quyidagicha:

$$15 : 5 = 3$$

$:$ bo‘lish belgisi bo‘lib, ba’zi manbalarda $\div$ belgisi bilan keladi. Yozma shaklda odatda $15\,/5$ ko‘rinishida ham yoziladi.

Endi $15 : 5$ ifodani hisoblashni o‘rgansak. Buning uchun 5 ta bolani ketma-ket qilib joylashtiramiz va 15 ta olmani har bir bolaga 1 tadan berishni boshlaymiz:

| | 1-bola | 2-bola | 3-bola | 4-bola | 5-bola |
|---|---|---|---|---|---|
| 1-qadamda | 1 ta olma | 1 ta olma | 1 ta olma | 1 ta olma | 1 ta olma |
| 2-qadamda | 2 ta olma | 2 ta olma | 2 ta olma | 2 ta olma | 2 ta olma |
| 3-qadamda | 3 ta olma | 3 ta olma | 3 ta olma | 3 ta olma | 3 ta olma |

Ko‘rib turganingizdek 1-qadamda har bir bolada 1 tadan olma bo‘ladi. 2-qadamda 2-tadan, 3-qadamda 3 tadan. Shu bilan olmalar tugaydi. Demak, 15 ta olmani beshta bola teng taqsimlab olsa, har bir bolaga 3 tadan olma tegarkan.

Yuqoridagi masalani teskarisini tuzsak ham bo‘ladi. Ya’ni beshta bolaning har birida 3 tadan olma bor bo‘lsa, ularda jami 15 ta olma mavjud bo‘ladi:

$$5 \cdot 3 = 5 + 5 + 5 = 15$$

Mavzu boshida yozganimizdek, bo‘lish amali ko‘paytirish amaliga teskari amal ekan, xuddi qo‘shish va ayirish amallari kabi. Umumiy qoida esa quyidagicha: $a$ sonini $b$ soniga bo‘lganda, shunda $c$ sonni topish kerakki, topilgan $c$ sonining $b$ soniga ko‘paytmasi $a$ sonini hosil qilsin. Ya‘ni:

$$a : b = c \text{ yoki } b \cdot c = a$$

Ushbu ifoda barcha sonlar uchun o‘rinli qoida, lekin $c$ sonini qanday topamiz? Har bitta sonni $b$ soniga ko‘paytirib chiqish juda ko‘p vaqtni oladi. Agar sonlar juda katta bo‘lsa buning deyarli imkoni yo‘q.

Shu yerda sizga bitta kichik sirni ochamiz. Agar siz xohlagan katta sonni 1, 2, 3, 4, 5, 6, 7, 8 va 9 sonlariga ko‘pay-

tira olsangiz, bilingki ikkita sonni bemalol bo‘la olasiz. Qanday deysizmi, quyidagi misolga e’tibor bering:

$$141248775 : 5649951 = ?$$

Bir qarashda murakkabga o‘xshaydi, lekin shu murakkab ifodani ham sodda usul bilan hisoblash mumkin. Buning uchun quyidagi ketma-ketlikni barajamiz:

1) Birinchi sonda chapdan boshlab to $5649951$ sonidan katta son hosil bo‘lguncha raqamlarni ajratib olamiz. Bizni holatda $141248775$ sonida chapdan boshlab, $5649951$ sonidan katta son hosil qilish uchun 14124877 sonini olishimiz kerak bo‘ladi. Qolib ketgan 5 raqami hozircha hisob-kitoblarda ishtirok etmaydi.

2) Endi 1 va 9 orasidan shunday ketma-ket ikkita son topishimiz kerak. Agar birinchi sonni $5649951$ soniga ko‘paytirsak, yangi hosil bo‘lgan 14124877 sonidan kichik bo‘lishi kerak, agar ikkinchi songa ko‘paytirsak 14124877 sonidan katta bo‘lishi kerak. Bizni holatda 2 va 3 sonlari bo‘ladi. $2 \cdot 5649951 = 11299902$ va natija 14124877 dan kichik $3 \cdot 5649951 = 16949853$ va natija 14124877 dan katta.

3) Topilgan 2 va 3 sonidan biz umumiy javobga 2 raqamini yozib turamiz, ya’ni kichik sonni. Hozircha javob 2.

   Endi yangi hosil qilingan 14124877 sonidan $2 \cdot 5649951$ ifodaning qiymatini ayiramiz:

$$14124877 - 2 \cdot 5649951 = 14124877 - 11299902 = 2824975$$

4) Endi hosil bo‘lgan $2824975$ soniga yuqorida qolib ketgan 5 raqamini oxiriga yozamiz va $28249755$ sonini hosil qilamiz. Xuddi 2-qadamdagiday ketma-ket ikkita sonni topishimiz kerak. Bizning

holatda 5 sonining o‘zi yetarli bo‘ladi, chunki: $5 \cdot 5649951 = 28249755$

5) Topilgan 5 sonini javob oxiriga yozamiz. Javob 25.

6) Birinchi sonimizda raqamlar tugadi demak:

$$141248775 : 5649951 = 25 \text{ yoki } 5649951 \cdot 25 = 141248775$$

Ko‘rib turganingizdek ikkita sonni bo‘lish uchun biz faqat sonni 1 dan 9 gacha ko‘paytirish va ayirish amallaridan foydalandik.

Yuqoridagi misol yordamida bo‘lish amali qanday qilinishini qisqacha ko‘rsatdik. Aslida bo‘lish amalida bir qancha qo‘shimcha shartlar ham mavjud. Shuning uchun quyidagi havolaga o‘tib video materiallarini yaxshilab o‘rganing.

*Qiziq fakt: Bo‘lish belgisi sifatida / belgisini Otred Uilyam, ÷ (lotincha «obelus») belgisini Rahn Iogann (nemischa Johann Heinrich Rahn, 1659-yilda) qo‘llashni taklif qilgan.*

«Butun sonlarni ko‘paytirish» mavzusida o‘rgangan barcha qonuniyatlarimiz va qavslarni ochish bo‘lish amali uchun ham o‘rinlidir, faqat ko‘paytirish belgisi o‘rniga bo‘lish amalini qo‘yib o‘rganamiz.

Yuqoridagi qonuniyatlarga bitta istisno mavjud, ya’ni sonni 0 ga bo‘lib bo‘lmaydi:

$$a : 0 = noma'lum\ son$$

Lekin 0 sonini xohlagan songa bo‘lish mumkin va natija har doim 0 ga teng: $0 : a = 0; a \neq 0$. $\neq$ teng emasligini bildiradigan belgi.

Agar ifodalarda ko‘paytirish va bo‘lish amali ketma-ket kelsa, ifodani chapdan boshlab o‘ngga qarab hisoblaymiz. Masalan quyidagi ifodaga e’tibor bering:

$$6 : 2 \cdot (1 + 2)$$

Ushbu ifoda internet tarmog‘ida juda mashhur. Ayrimlar javobni 9 chiqarsa ayrimlar 1 javobini chiqaradi. Aslida ikkalasi ham to‘g‘ri, lekin hozirgi kunda 9 javobi to‘g‘ri bo‘ladi. Chunki oldinlari ko‘paytirish amali bo‘lish amalidan oldin hisoblangan.

Ifoda esa quyidagicha hisoblanadi:

$$6 : 2 \cdot (1 + 2) = 6 : 2 \cdot (3) = 6 : 2 \cdot 3 = 3 \cdot 3 = 9$$

**C** dasturlash tillida bo‘lish operatori / (forward slash yoki slash) belgisi bilan yoziladi. **Python** dasturlash tilida esa ikkita: / va // (ketma-ket ikkita slash yozamiz). Bu ikkalasining farqini keyingi mavzularda ko‘rib chiqamiz.

Keling, endi quyidagi ifodaning dasturlash tillarida yozilishini ko‘ramiz:

$$25 + 90 : (20 + 10) - 14 \cdot 2$$

```
printf("%d", 25 + 90 / (20 + 10) - 14 * 2); // 0
```

```
print(25 + 90 // (20 + 10) - 14 * 2) # 0
```

Ko‘rib turganingizdek ifoda qanday yozilgan bo‘lsa hammasini shunday yozamiz, faqat : belgisi o‘rnida **C** dasturlash tilida bitta /, **Python** dasturlash tilida esa // yozib ketamiz. Qo‘shish, ayirish va ko‘paytirish amallari oldingi mavzularimizda o‘tganimizdek yoziladi.

Mavzuga doir masalalar:

Xurmatli o‘quvchi agar sizga kitobimiz ma’qul kelgan bo‘lsa, siz kitobni itboom.uz dan sotib olishingiz mumkin.