O`ZBEKISTON RESPUBLIKASI
OLIY VA O'RTA MAXSUS TA'LIM VAZIRLIGI
BUXORO DAVLAT UNIVERSITETI

M.A.Bobojonova, H.Sh.Rustamov

PYTHON DASTURLASH TILIDA MASALALAR VA UNING YECHIMLARI

O`QUV QO`LLANMA

5130200 – Amaliy matematika

5330100 – Kompyuter ilmlari va dasturlash texnologiyalari

5330200 – Axborot tizimlari va texnologiyalari

5111018 – Kasb ta`limi: Informatika va AT

Python dasturlash tilida masalalar va uning yechimlari[matn];O`quv qo`llanma/ M.A.Bobojonova, H.Sh.Rustamov. Python dasturlash tilida masalalar va uning yechimlari. O`quv qo`llanma – Buxoro, BuxDU, 2022, 240b.

Python dasturlash tili joriy yilda eng zamonaviy, eng sodda, dasturchilar tomonidan eng ko`p qo`llanilayotgan til sifatida e`tirof etildi. Amaliy jihatdan olib qaraydigan bo`lsak, Pythonda masalalar to`plamiga oid o`zbekcha kitoblarimiz sanoqli. Ularda axborot juda kam bo`lib, sayoz yozilgan. “Python dasturlash tilida masalalar va uning yechimlari” o`quv qo`llanmasi umumta`lim maktablari, akademik litsey hamda Oliy o`quv yurtlaridagi o`qitish effektivligini oshirishga, informatika darslarida Python tilini kengroq qo`llashga xizmat qiladi.

Mazkur o`quv qo`llanma Oliy ta`lim muassalarining **“**5130200 – Amaliy matematika**”** , **“**5330100 – Kompyuter ilmlari va dasturlash texnologiyalari**”** va **“**5330200 – Axborot tizimlari va texnologiyalari**”** ta`lim yo`nalishlarining **“**Algoritmik tillar va dasturlash**”** fanidan amaliyot darslarida foydalanish uchun tavsiya etiladi.

Taqrizchilar:

O.K. Soliyeva - Buxoro muhandislik texnologiyalari instituti Texnologik jarayonlarni boshqarishning axborot-kommunikatsiya tizimlari kafedrasi dotsenti;

S.S. Babayev - Buxoro davlat universiteti Amaliy matematika va dasturlash texnologiyalari kafedrasi dotsenti f.m.f.f.d.(PhD).

Ushbu o`quv qo`llanma Axborot Texnologiyalari fakulteti Amaliy matematika va dasturlash texnologiyalari kafedrasining 2022-yil 8-iyundagi 38-yig`ilishida ko`rib chiqildi va chop etishga ruxsat etildi.

MUNDARIJA

KIRISH

XXI asr axborot texnologiyalari asri. Dunyomizda qilinayotgan kashfiyotlar, olimlarning izlanishlari, fan-texnika yangiliklari fikrimizning yaqqol misolidir. Masalan, atom energetikasi, zamonaviy kompyuter texnologiyalarining xilma-xilligi, internet texnologiyalari va undan tashqari raqamli televideniyaning rivoji – bularning barchasi asrimizning katta yutuqlaridandir.

Bu borada muhtaram prezidentimiz Shavkat Mirziyoyev quyidagicha ta`kidlaganlar: *“BMT ning Yoshlar strategiyasida ta`kidlanganidek, “Yoshlar timsolida eng qimmatli va o`ta muhim resurslar mujassam bo`lib, unga har qancha investitsiya kiritsa arziydi, chunki bu sarmoyalar bir necha barobar ziyod bo`lib qaytadi”. Men ushbu g`oyat muhim fikrga to`liq qo`shilaman.*

Biz yoshlarimizda umrimizning ma`no-mazmuni, hayotimizning asosiy mevasi va samarasini ko`ramiz. Yangi O`zbekistonni azm-u shijoatli yoshlarimiz bilan birga bunyod etamiz!”

Bundan tashqari yurtimizda axborotlashtirish sohasiga, uning hayotimizga bevosita tadbiqiga kundan-kunga e’tibor kuchaymoqda. Fikrimizning yaqqol misoli sifatida Prezidentimizning 2020-yil 6-oktabrdagi PQ-4851-sonli “Axborot texnologiyalari sohasida ta`lim tizimini yanada takomillashtirish, ilmiy tadqiqotlarni rivojlantirish va ularni IT-industriya bilan integratsiya qilish chora-tadbirlari to`g`risida”gi qarori, shuningdek, O`zbekiston Respublikasini rivojlantirishning beshta ustuvor yo`nalishi bo`yicha Harakatlar strategiyasini “Ilm, ma`rifat va raqamli iqtisodiyotni rivojlantirish yili”da amalga oshirishga oid davlat dasturida belgilangan vazifalar, axborot texnologiyalari sohasidagi kadrlarni tayyorlash tizimini takomillashtirish “Raqamli O`zbekiston-2030” strategiyasini muvaffaqiyatli amalga oshirish maqsadida qilinayotgan ishlarni keltirishimiz mumkin.

Ushbu o`quv qo`llanma umumta`lim maktablari, akademik litsey hamda Oliy o`quv yurtlaridagi o`qitish effektivligini oshirishga, informatika darslarida Python tilini kengroq qo`llashga xizmat qiladi.

“Python dasturlash tilida masalalar va uning yechimlari” o`quv qo`llanmasi kirish, 6 bob, 22 paragraf, mundarija hamda foydanilgan adabiyotlar va internet resurlari ro`yxatidan iborat. O`quv qo`llanmada har bir dastur sharti, kerakli formula va chizmalar, yechimi va natijalari ko`rinishida berilgan.

1-bob “Dasturlash tillari tarixi” deb nomlanib, unda dasturlash tillari tarixi, turlari, Python dasturlash haqida ma`lumot hamda dasturlash asoslari haqida tushuncha berilgan.

2-bob “Chiziqli algoritmlarga doir dasturlar” bo`lib, unda eng oddiy masaladan murakkab chiziqli masalagacha yechimlari ko`rsatilgan. Ushbu bob arifmetik, satrli masalalar, shuningdek, sana va vaqtga, ro`yxat, kortej, lug`at va to`plamga, massivga, sinfga doir masalalardan tashkil topgan.

3-bob esa “Tarmoqlanuvchi algoritmlarga doir dasturlar” deb nomlanadi. U arifmetik, satrli, ro`yxat, lug`at va to`plamga doir masalalardan iborat.

Keyingi 4-bob “Takrorlanuvchi algoritmlarga doir dasturlar”dir. U ham arifmetik, satrli, ro`yxat, lug`atga doir masalalarni o`z ichiga olgan.

5-bob “Aralash dasturlar” deb nomlanadi. U o`zida arifmetik, satrli masalalarni hamda ro`yxatga, massivga doir masalalarni, qidiruv va tartiblash, rekursiya masalalarni jamlagan.

Oxirgi bob “Olimpiada masalalari” bo`lib hisoblanadi. Ushbu bobda qiyinroq masalalarning yechimlari ko`rsatilgan.

O`quv qo`llanmaning umumiy hajmi 240 betdan iborat bo`lib, 242 ta masala yechib ko`rsatilgan. Masalalarning 46 tasi olimpiada masalalari ekanligini ham alohida ta`kidlab o`tish kerak.

1-BOB. DASTURLASH TILLAR TARIXI

1-§. DASTURLASH TILLARI TURLARI

«Ikki tipdagi dasturlash tillari mavjud — birinchisi ustida hamma “urushadi”, boshqasida esa hech kim yozmaydi».

(B. Strustrup, C++ dasturlash tilining asoschisi)

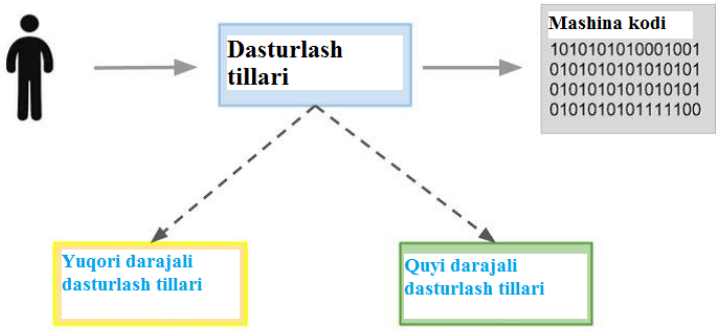
1.1-rasm. Markaziy protsessor tuzilishi

Imtihonni katta tomli kitobni o`qib topshirish mumkindir, lekin dastur yozish uchun bu yordam bermaydi. O`qishga qancha vaqt kerak bo`ladi? Musiqachilar ma`lum bir musiqa asbobini o`zlashtirish uchun har kuni 4 soatlik mashq qilish kerakligini aytishadi.

Dasturchining ishchi quroli komputer bo`lib, barcha hisoblashlar markaziy protsessorda bo`lib o`tadi. Dastur fayli doimiy xotirada (qattiq diskda) saqlanadi, bajarilish vaqtida esa vaqtinchalik (operativ) xotira yuklanadi.

Komputerda axborotni kiritish klaviatura (kiritish qurilmasi) orqali, chiqarish esa monitor (chiqarish qurilmasi) orqali amalga oshiriladi.

Komputer faqat ikki tipdagi signallar: 0 va 1 (mashina kodlari) bilan ishlay oladi. 1010101010010101010 ko`rinishdagi dasturni yozish inson uchun qiyin masala. Shuning uchun ham insoniyatga tushunarli bo`lgan dasturlash tillari bilan birgalikda komputerga tushunarli bo`lgan mashina tili translator-masalalar yaratildi.



1.2-rasm. Dasturlash tillari turlari

Mashina tillariga yaqin bo`lgan dasturlash tillari quyi darajali dasturlash tillari (Assembler), inson fikrlashiga yaqinroq bo`lgan dasturlash tillari yuqori darajali dasturlash tillari (Python, Java, C#) deb ataladi.

Dasturlash tillarining har birining ajoyib tarixi mavjud. Ular maxsus joyda emas, balki ishlab chiqaruvchilar oldida turgan konkret masalani bajarish orqali yaratilgan. Hozirgi kunda minglab dasturlash tillari yaratilgan, lekin ulardan bir nechtasigina asosiy rol o`ynaydi.

Avvalroq biz kompyuter bilan aloqaning boshlanishi mashina kodi ekanligini aytgan edik. Keyin, XX asrning 50-yillarida, mashina darajasiga eng yaqin bo'lgan past darajadagi **Assembler tili** paydo bo'ldi. U protsessor bilan bog'langan, shuning uchun uni o'rganish protsessor arxitekturasini o'rganishga o'xshaydi. Bugungi kunda ham assembler tilida yozilgan, xotira resurslari juda cheklangan dasturlar kichik qurilmalar(mikrokontrollerlar)ning ajralmas qismi hisoblanadi. Keyingi bosqich - matematik hisoblashlar uchun mo'ljallangan **Fortran tili**ning paydo bo'lishi. Ushbu davrda yaratilgan **Paskal tili** ham hanuzgacha maktablarda dasturlashni o'rgatishda asosiy til sifatida qo'llaniladi.

Dastlab operatsion tizim assembler tilida yozilgan bo‘lib, uni o‘zgartirish va o‘rganishni qiyinlashtirdi, keyin **D.Ritchi** tizim dasturlash uchun **C tilini** ishlab chiqdi va B.Kernigan bilan birgalikda UNIX tizimini shu tilda qayta yozdi. Keyinchalik, UNIX operatsion tizimi keng tarqaldi (hozirda uning GNU / Linux versiyalari ko'proq ma'lum) va u bilan birga ko'plab dasturchilar paydo bo'ldi, ular uchun C tili mahalliy bo'lib qoldi.

Keyingi bosqich (80-yillar) yirik sanoat dasturlarini yaratishni soddalashtirishi kerak bo'lgan ob'ektga yo'naltirilgan dasturlashning (OOP) paydo bo'lishi bilan tavsiflanadi. **B. Stroustrup** C tilining imkoniyatlaridan qoniqmadi, shuning uchun u OOP qo'shish orqali bu tilni kengaytiradi. Yangi til **C++** deb nomlandi.

90-yillarda shaxsiy kompyuterlar va Internet paydo bo'ldi, shuning uchun yangi texnologiyalar va dasturlash tillari talab qilindi. Bu vaqtda **Java tili** mashhur bo'ldi, tizimdagi biror narsani jiddiy buzmasdan eng qisqa vaqt ichida katta ilovalarni yozishni boshlash imkonini beradi. Ushbu til o'z dasturlarini ko'chirish qobiliyati bilan tavsiflanadi.

Java bilan bir vaqtda **Python tili** paydo bo'ladi. Til ishlab chiquvchisi, matematik **Guido van Rossum** uzoq vaqt davomida dasturlashni o'rgatish uchun ABC tilini ishlab chiqdi.

Birinchi bosqichda dasturchi "xom" ma'lumotlar to'plamiga ega. Bu, masalan, turli xil buxgalteriya hisobotlari, statistik ma'lumotlar va boshqalar bo'lishi mumkin. Bu ma'lumotlar tuzilishi va kompyuterda joylashtirilishi kerak. Keling, dasturni yozishni salat tayyorlash bilan taqqoslaylik: yuvilishi va kesilishi kerak bo'lgan "xom" sabzavotlar mavjud, ya'ni dasturning tuzilishi.

Vazifani alohida kichik vazifalarga bo'lish yaxshiroq natija beradi. Kichik muammolarni hal qilish, ularning ishlashiga ishonch hosil qilish va ularni birlashtirishni ilmiy tilda tahlil va sintez deyiladi. Tajriba bilan birga keyingi vazifalarni ko'rish va ajratib ko'rsatish qobiliyati keladi.

Keyinchalik, dasturchi algoritmni topshiriq asosida tuzilgan ma'lumotlarni qayta ishlashni amalga oshiradi. Ma'lumotlar strukturasini to'g'ri tanlash algoritmni yaratishga imkon beradi.

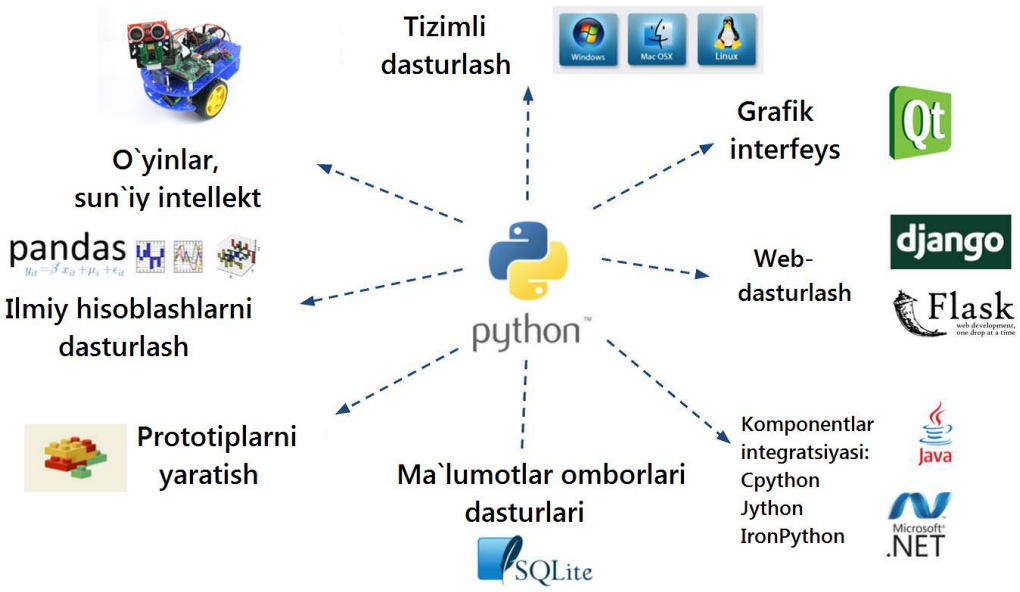
Algoritm ishlab chiqilgandan va dastur ishlagandan so'ng (uning ishi natijasida to'g'ri javob olinadi), siz chiroyli va qulay interfeys yaratishingiz mumkin.

Dasturlash tili alifbosi quyidagicha:

- Yordamchi so`zlar: print, len, type, def…
- Literal o`zgaruvchilar: number, strings…
- Operatorlar: + - * = , ; …
- Grammatika (sinkaksis)
- Foylanuvchi yaratgan so`zlar: variables, functions, classes…

2-§. PYTHON DASTURLASH TILI HAQIDA

Python faol qo'llaniladigan sohalarni qisqacha sanab o'tamiz:



1.3-rasm. Python tili sohalari

1. Tizimli dasturlash.
2. Grafik interfeysli dasturlarni ishlab Chiqarish.
3. Dinamik veb-saytlarni ishlab Chiqarish.
4. Komponentlarning integratsiyasi.
5. Ma'lumotlar bazalari bilan ishlash dasturlarini ishlab Chiqarish.
6. Tez prototiplash.
7. Ilmiy hisoblash uchun dasturlarni ishlab Chiqarish.
8. O'yinni ishlab Chiqarish.

Python dasturlarini ishga tushirish uchun bizga nima kerak? Bu savolga javob berishdan oldin dasturlarning kompyuterda qanday ishlashini ko'rib chiqamiz. Dasturlar operatsion tizim (Windows, Linux va boshqalar) tomonidan amalga oshiriladi. Operatsion tizimning vazifalariga dastur uchun resurslarni (RAM va boshqalarni) taqsimlash, kiritish-chiqarish qurilmalariga kirishni ta`qiqlash, ruxsat berish va boshqalar kiradi.

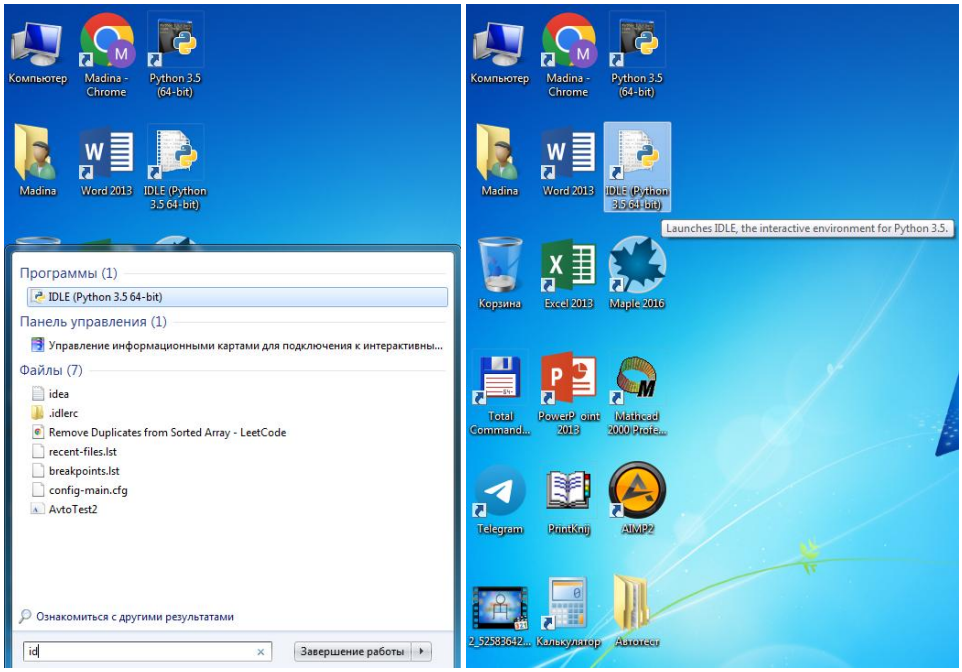
Python dasturlarini ishga tushirish uchun sizga Python tarjimoni (virtual mashina) kerak. Ushbu dastur operatsion tizimning barcha xususiyatlarini Python

dasturchisidan yashiradi, shuning uchun Windows tizimida yozilgan Python dasturini GNU/Linuxda ishga tushirishingiz va bir xil natija olishingiz mumkin.

Python dasturini kompyuterga o‘rnatish bir necha bosqichlardan iborat:

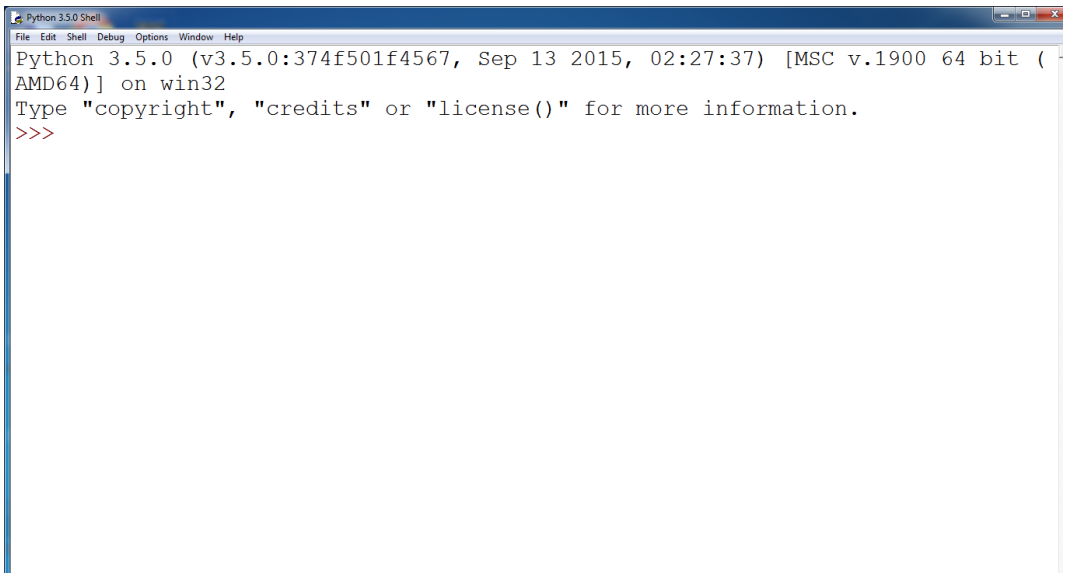
https://www.python.org/downloads/windows/ rasmiy veb saytiga kirib, kompyuteringizning texnik parametrlarini hisobga olgan holda eng so‘nggi versiyasini yuklab oling. Kompyuterning texnik parametrlariga, razryadi, (64 bit yoki 32 bit) qaysi operatsion sistema o‘rnatilganligi va boshqa parametrlar kiradi. Keltirilgan dasturlarni yechishda Python 3-versiyasi yoki undan keyingi versiyalar kerak bo`ladi.

IDLE ni ishga tushirish tartibi:



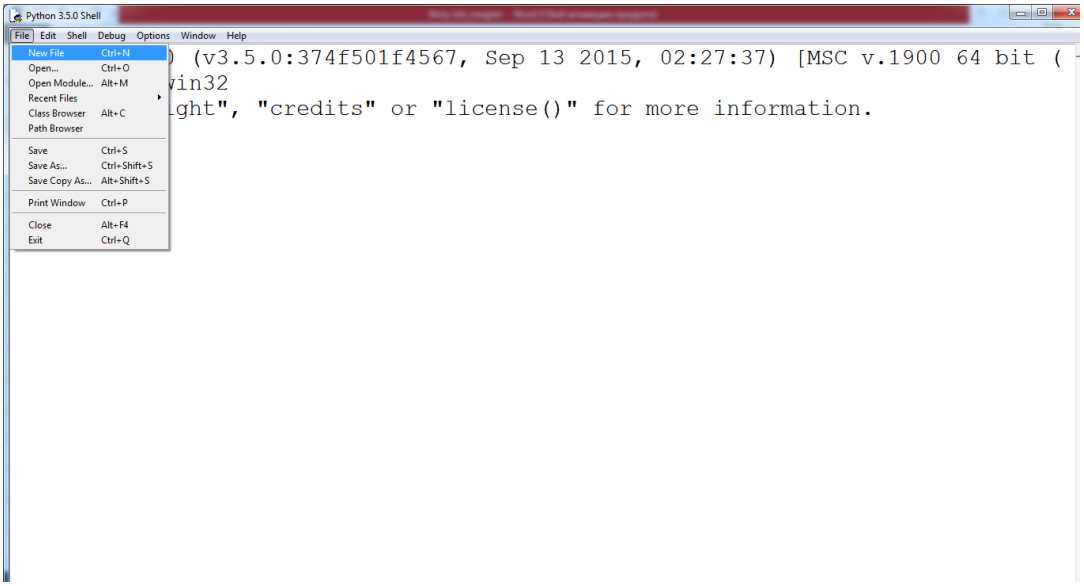
1.4-1.5-rasmlar. IDLE Python dasturini ishga tushirish

IDLE ni ishga tushirgandan so`ng quyidagi oyna paydo bo`ladi:



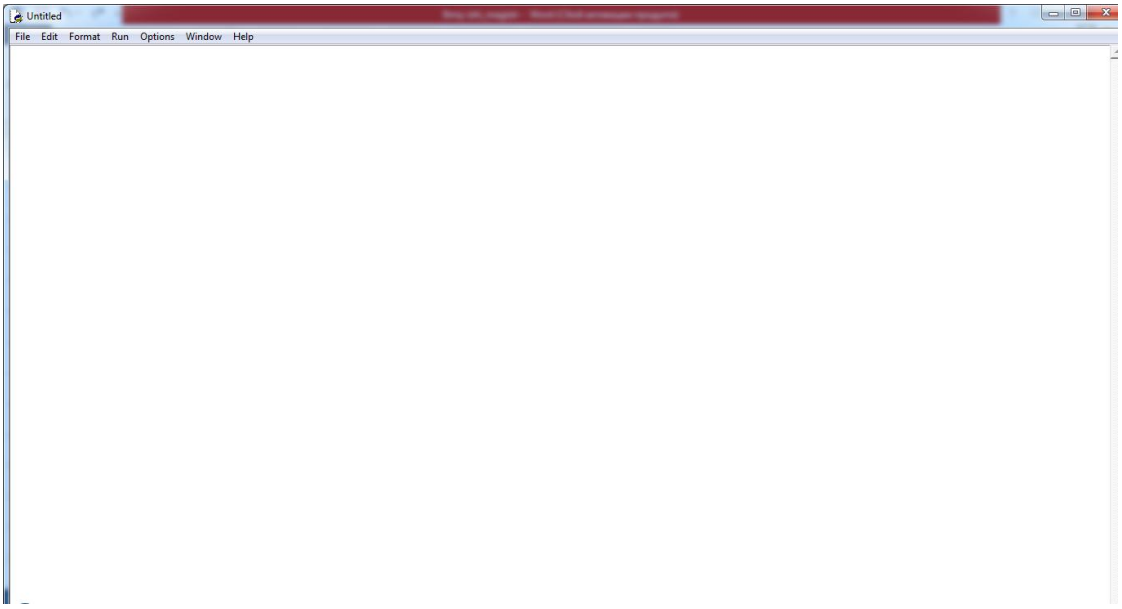
1.6-rasm. IDLE Python dasturining ishchi oynasi

IDLE da yangi fayl yaratish:



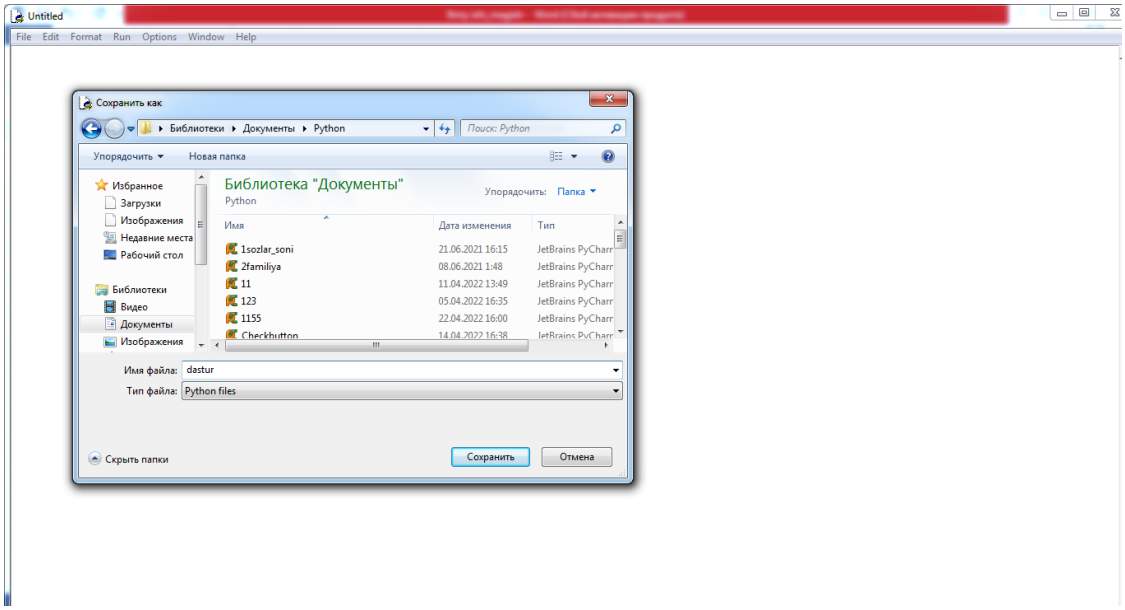
1.7-rasm. IDLE Pythonda yangi fayl yaratish

Untitled nomi ostida Yangi fayl yaratiladi:



1.8-rasm. Yangi fayl yaratish

Faylni saqlash uchun file-save yoki ctrl+s tugmalarini bosamiz. Kerakli papkani tanlab, faylimizga nom berib saqlaymiz. .py kengaytmali fayl paydo bo`ladi. masalan, dastur.py



1.9-rasm. Yangi faylni saqlash

O'rganishning eng boshida Pythonni oddiy interaktiv kalkulyator deb hisoblash mumkin. Interaktiv IDLE rejimida quyidagi matematik ifodalarning

qiymatlarini toping. Ifodani tugatgandan so'ng, kiritishni yakunlash va natijani ekranda ko'rsatish uchun Enter tugmasini bosing.

```
>>> 3.0 + 6
9.0
>>> 4 + 9
13
>>> 1 - 5
-4
>>> _ + 6
2
>>>
```

Oldingi misoldagi pastki chiziq oxirgi olingan natijani bildiradi.

Agar biror-bir sababga ko'ra buyruq yozishda xatoga yo'l qo'ysangiz, Python bu haqida xabar beradi:

```
>>> a
Traceback (most recent call last):
File "<pyshell#0>", line 1, in <module>
a
NameError: name 'a' is not defined
>>>
```

Xato qilishdan qo'rqmang! Python tuzatadi va nimaga e'tibor berish kerakligini aytadi. Linuxda o'rganish uchun IDLE muharririni o'rnatishingiz kerak bo'ladi:

```
sudo apt-get install idle3
```

Qiziqarli ma`lumotlar:

Python nima?

• Ilon.

• Britaniyalik kulguli guruhning nomi Monty Python.

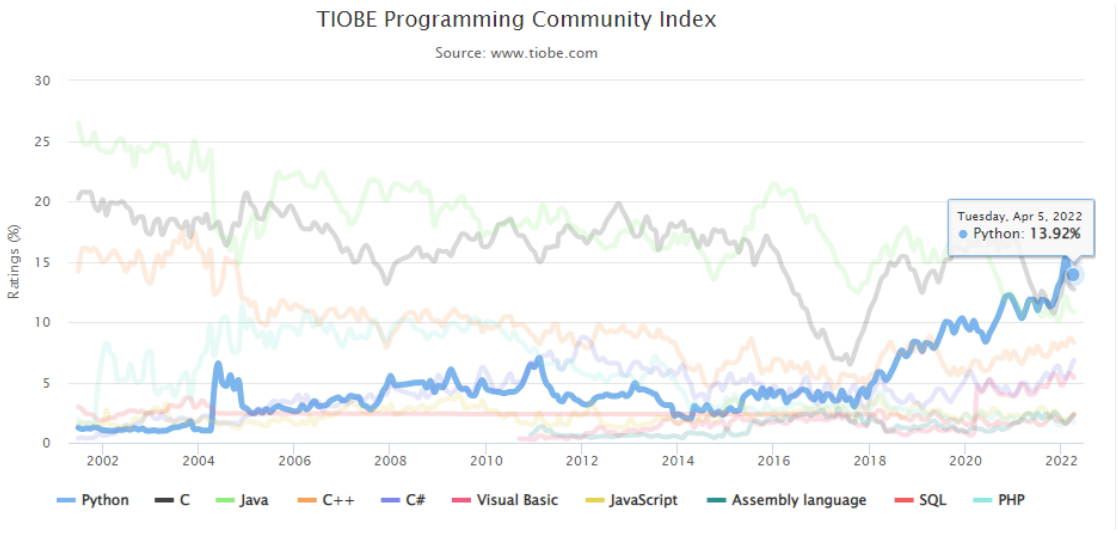
• Python dasturlash tilidagi kompilator/interpretator(a.k.a. CPython).

Apr 2022	Apr 2021	Change	Programming Language	Ratings	Change
1	3	↑	Python	13.92%	+2.88%
2	1	↓	C	12.71%	-1.61%
3	2	↓	Java	10.82%	-0.41%
4	4		C++	8.28%	+1.14%
5	5		C#	6.82%	+1.91%
6	6		Visual Basic	5.40%	+0.85%
7	7		JavaScript	2.41%	-0.03%
8	8		Assembly language	2.35%	+0.03%
9	10	↑	SQL	2.28%	+0.45%
10	9	↓	PHP	1.64%	-0.19%

1.10-rasm. TIOBE natijalari

Dasturlash tilining mashhurligini ishlab chiquvchilar orasida eng ommabop resurs - Stack Overflowdagi teglar sonining dinamikasi bilan kuzatish mumkin. Shunday qilib, grafikaga ko'ra, Pythonning o'sishi 2010 yilda boshlangan va 2015 yilda tezlashdi.

Pythonni eski til deb atash mumkin - u 1991 yilda, ya'ni deyarli 31 yil oldin paydo bo'lgan. Bu vaqt ichida u asta-sekin atrofiga katta jamoani to'pladi.



1.11-rasm. Dasturlash tillari reytingi

Python birinchi dasturlash tili sifatida ishonchli tavsiya etilishi mumkin. Gap shundaki, Python uzoq vaqtdan beri mavjud va u haqida juda ko'p yaxshi darsliklar yozilgan. U oddiy, "inson" tiliga o'xshash tushunarli sintaksisga ega va u xatolarni ham tushuntiradi.

Masalan, undagi ma'lumotlar turini ko'rsatish shart emas, faqat o'zgaruvchini e'lon qilish kifoya. Python kontekstdan uning butun son, haqiqiy son, mantiqiy yoki boshqa biror tipda ekanligini avtomatik aniqlaydi. Bu yangi boshlanuvchilar uchun katta afzallik.

Agar siz C++ da dasturlashtirgan bo‘lsangiz, biror joyda haqiqiy (float) sonini butun (int) songa o‘zgartirganingiz uchun dastur kompilyatsiya qilinmasligi qanchalik achinarli ekanini bilasiz. Python kodini o'qish juda oson.

Quyida "Yil dasturlash tili" mukofotining barcha g'oliblari ro'yxati keltirilgan. Mukofot bir yil ichida reytingi eng yuqori ko'tarilgan dasturlash tiliga beriladi.

Year	Winner
2021	Python
2020	Python
2019	C
2018	Python
2017	C
2016	Go
2015	Java
2014	JavaScript
2013	Transact-SQL
2012	Objective-C
2011	Objective-C
2010	Python
2009	Go
2008	C

1.12-rasm. Yil dasturlash tili

3-§. DASTURLASH FANIGA KIRISH

Python dasturi yordamchi so`zlar to'plamidan iborat. Har bir yordamchi so`z yangi qatorga joylashtiriladi. Misol uchun:

```
print(2 + 3)
print("Salom")
```

Natija:

```
5
Salom
```

Pythonda qator boshidan qo`yilgan bo`sh joylar(отступ, Tab orqali qo`yilgan) katta rol o'ynaydi. Noto'g'ri qo`yilgan bo`sh joylar xatolik beradi. Masalan, kod yuqoridagi bilan deyarli bir xil bo'lsa ham quyidagi holatda biz xatoga yo'l qo'yamiz:

```
print(2 + 3)
   print("Salom")
```

Xatolik beradi.

Shuning uchun, satrning boshida yangi yordamchi so`zlar qo'yishga arziydi. Bu Python va C# yoki Java kabi boshqa dasturlash tillari o'rtasidagi muhim farqlardan biridir.

Ammo shuni yodda tutish kerakki, ba'zi til konstruktsiyalari bir necha qatordan iborat bo'lishi mumkin. Masalan, agar shartli:

```
if 1 < 2:
   print("Salom")
```

natija: "Salom"

Bunday holda, agar 1 2 dan kichik bo'lsa, u holda "Salom" qatori ko'rsatiladi. Chunki print("Salom") iborasi if shartli konstruktsiyasining bir qismi sifatida ishlatiladi.

Python katta-kichik harflarga sezgir tildir, shuning uchun print, Print yoki PRINT ifodalari turli ifodalarni ifodalaydi. Agar konsolga chiqarish uchun print o'rniga, biz Printdan foydalansak:

```
Print("Salom")
```

Natija:

```
Traceback (most recent call last):
  File "C:\Users\Madina\Documents\Python\11.py", line 1, in <module>
    Print("Salom")
NameError: name 'Print' is not defined
```

Bunda biz hech qanday natijaga erishmaymiz.

Izohlar -Sharhlar kod qismi nima qilishini belgilash uchun ishlatiladi. Dastur bajarilganda, izohlarni e'tiborsiz qoldiradi, shuning uchun ular dasturning ishlashiga ta'sir qilmaydi. Pythondagi sharhlar blok va inline izohlarda keladi. Satr izohlaridan oldin funt belgisi - # qo'yiladi.

Ular alohida qatorda bo'lishi mumkin:

```
# Konsolga kirish
# Salom dunyo! xabari
print("Salom dunyo!")
```

Natija:

```
Salom dunyo!
```

belgisidan keyingi har qanday belgilar to'plami izohni bildiradi. Ya'ni, yuqoridagi misolda kodning birinchi ikki qatori sharhlardir.

Ular, shuningdek, yordamchi so`zlar bilan bir qatorda joylashgan bo'lishi mumkin:

```
print("Salom dunyo!")  # Konsolga xabarni chop etish
```

Blok izohlarida sharh matnidan oldin va keyin uchta bitta tirnoq qo'yiladi: "'sharh matni"'. Misol uchun:

```
'''
   Konsolga kirish
   Xabarni chop etish
'''
```

```
print("Salom dunyo!")
```

Asosiy funksiyalari

Python bir qator funksiyalarni taqdim etadi. Ulardan ba'zilari, ayniqsa, til o'rganishning dastlabki bosqichlarida juda tez-tez ishlatiladi, shuning uchun ularni ko'rib chiqaylik.

Konsolga ma'lumot chiqarishning asosiy funksiyasi print() funksiyasi hisoblanadi. Biz chiqarmoqchi bo'lgan satr ushbu funksiyaga argument sifatida uzatiladi:

```
print("Salom dunyo!")
```

Agar biz konsolga bir nechta qiymatlarni chop etishimiz kerak bo'lsa, ularni vergul bilan ajratilgan bosib chiqarish funksiyasiga o'tkazishimiz mumkin:

```
print("Familiyasi, ismi:", "Mustafoyev", "Muslimbek")
```

Agar print funksiyasi chiqarish uchun javobgar bo'lsa, u holda input funksiyasi axborotni kiritish uchun javobgardir. Bu funksiya kiritish so‘rovini ixtiyoriy parametr sifatida oladi va biz o‘zgaruvchida saqlashimiz mumkin bo‘lgan kirish qatorini qaytaradi:

```
ism = input("Ismingizni kiriting: ")
print("Salom", ism)
```

Natija:

```
Ismingizni kiriting: Madina
Salom Madina
```

2-BOB. CHIZIQLI ALGORITMLARGA DOIR DASTURLAR

4-§. ARIFMETIK MASALALAR

2.1-masala.

To`g`ri to`rtburchakning yuzasini hisoblash dasturini tuzing.

Yechim: $S = a \cdot b$;

bu yerda a-to`g`ri to`rtburchakning eni, b-to`g`ri to`rtburchakning bo`yi.

Dastur kodi	Dastur natijasi
a=23 b=17 s=a*b print("Yuza =", s)	Yuza = 391

2.2-masala

Radiusi berilgan doiraning yuzasi va uzunligini hisoblash dasturini tuzing.

Yechim: $S = \pi \cdot r^2, \quad l = 2\pi r$;

bu yerda, r-doira radiusi.

Dastur kodi	Dastur natijasi
r=7 pi=3.14 print("Yuzasi ", r*r*pi) print("Aylana uzunligi", 2*r*pi)	Yuzasi 153.86 Aylana uzunligi 43.96

2.3-masala

math kutubxonasidan foydalanib, 2.2-masalani yechamiz.

math —matematik funksiyalar kutubxonasi

Bu modul o‘z ichida matematikaga oid turli funksiyalar va o‘zgaruvchilarni saqlaydi. Keling, ularning ba‘zilari bilan tanishamiz:

π ning qiymati (Pi), e — natural logarifm asosi(e), trigonometrik funksiyalar (cos, sin, tangens, arccos), logarifmlar (log() va log10() funksiyalari), ildiz va daraja (sqrt(), pow() funksiyalari), sonlarni yaxlitlash (round(), ceil(), floor() funksiyalari) va hokazo.

Dastur kodi	Dastur natijasi
import math r=7 print("Yuzasi ", r*r*math.pi) print("Aylana uzunligi", 2*r*math.pi)	Yuzasi 153.93804002589985 Aylana uzunligi 43.982297150257104

2.4-masala

Asosi va balandligi berilgan uchburchakning yuzasini topish dasturini tuzing.

Yechim: $S = \frac{1}{2} \cdot a \cdot h$;

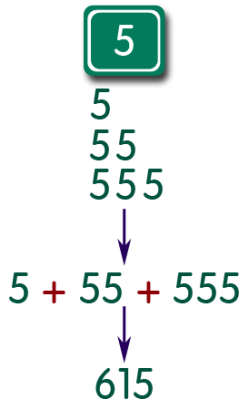
bu yerda, a - uchburchakning asosi, h - uchburchakning balandligi.

Dastur kodi	Dastur natijasi
a = int(input("Asos = ")) h = int(input("Balandlik = ")) s = a*h/2 print("Yuza =", s)	Asos = 5 Balandlik = 6 Yuza = 15.0

2.5-masala

Kiritilgan butun tipdagi n sonni n+nn+nnn ko`rinishida hisoblash dasturini tuzing.

Masalan, n=5 -> 5+55+555=615.



Dastur kodi	Dastur natijasi
a = int(input("Butun raqam kiriting: ")) n1 = int("%s" % a) n2 = int("%s%s" % (a,a)) n3 = int("%s%s%s" % (a,a,a)) print (n1+n2+n3)	Butun raqam kiriting: 5 615

2.6-masala

1 dan N gacha musbat butun songacha bo`lgan sonlarning yig`indisini hisoblash dasturini tuzing.

1-usul: arifmetik progressiyani qo`llash.

Yechim: $S = \frac{a_1+a_n}{2} \cdot n$; bu yerda $a_1 = 1$ va $a_n = n$ ga teng.

Dastur kodi	Dastur natijasi
n = int(input("N ni kiriting: ")) yigindi = (n * (n + 1)) / 2 print(n ,"gacha bo`lgan sonlarning yig`indisi: ", yigindi)	N ni kiriting: 8 8 gacha bo`lgan sonlarning yig`indisi: 36.0

2-usul: range funksiyasini qo`llash.

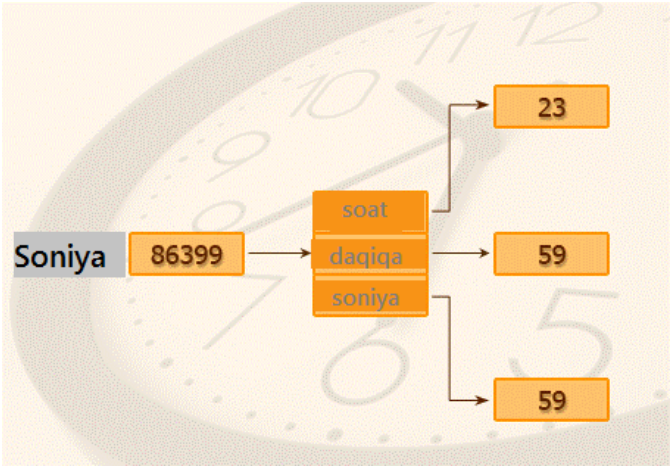
range() funksiyasi berilgan diapazonda bir qator sonlarni yaratishga imkon beradi. range() funksiyasini 3 xil chaqirish mumkin:

1. range(stop) - bitta argumentni oladi;
2. range(start, stop) - ikkita argumentni oladi;
3. range(start, stop, qadam) - uchta argumentni oladi.

Dastur kodi	Dastur natijasi
n = int(input("Butun son kiriting: ")) yigindi = sum(range(n+1)) print(n,"gacha sonlarning yig`indisi:",yigindi)	Butun son kiriting: 56 56 gacha sonlarning yig`indisi: 1596

2.7-masala

Python dasturlash tilida soniyadan kun, soat, daqiqa va soniyaga o`tkazish dasturini tuzing.



Dastur kodi	Dastur natijasi
vaqt = float(input("Sekund birligidagi vaqtni kiriting: ")) kun = vaqt // (24 * 3600) vaqt = vaqt % (24 * 3600) soat = vaqt // 3600 vaqt %= 3600	Sekund birligidagi vaqtni kiriting: 120000 kun, soat, daqiqa, soniya-> 1:9:20:0

daqiqa = vaqt // 60 vaqt %= 60 soniya = vaqt print("kun, soat, daqiqa, soniya-> %d:%d:%d:%d" % (kun, soat, daqiqa, soniya))	

2.8-masala.

Sonni kiritish uchun Python dasturini yozing, agar u son bo'lmasa, xato xabari paydo bo'lsin.

Dastur kodi	Dastur natijasi
x = 5.67 x_int = x.is_integer() print("x butun sonligini tekshirish!") print(x_int) y= 12.0 y_int = y.is_integer() print("y butun sonligini tekshirish!") print(y_int)	x butun sonligini tekshirish! False y butun sonligini tekshirish! True

Ushbu masalaning 2-usuli 4.3-masalada berilgan.

2.9-masala

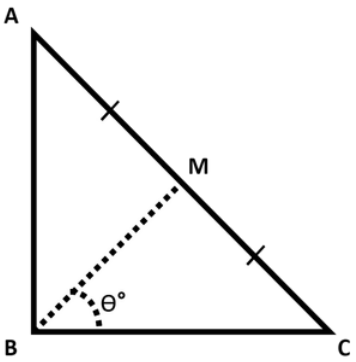
Satr sifatida ifodalangan ikkita musbat sonlar num1 va num2 berilgan bo'lsa, 1- va 2-sonning ko'paytmasini qaytaring.

Eslatma: Siz o'rnatilgan BigInteger kutubxonasidan foydalanmasligingiz yoki kiritilgan ma'lumotlarni to'g'ridan-to'g'ri butun songa aylantirmasligingiz kerak.

Dastur kodi	Dastur natijasi

def multiply(num1,num2):
  a=int(num1)*int(num2)
  return str(a)
 	
print(multiply("12","3"))
	36

2.10-masala



To`g`ri burchakli uchburchakda $\angle B = 90°$ yoki $\angle ABC = 90°$. M nuqta AC gipotenuzaning o`rta nuqtasi. Sizga AB va BC tomonlarning uzunliklari berilgan. $\angle MBC$ (rasmdagi $\theta°$)ning qiymatini gradusda topishingiz kerak.

Kiritish formati:

1-qatorda AB tomonining uzunligi berilgan.

2-qatorda BC tomonining uzunligi berilgan.

Shartlar:

$0 < AB \leq 100$

$0 < BC \leq 100$

AB va BC larning uzunliklari natural sonlar.

Chiqarish formati:

$\angle MBC$ ning gradus qiymati

Eslatma: Gradusni eng yaqin butun qiymatgacha yaxlitlang.

Kiritish namuna:

10

10

Chiqarish namuna:

45°

Dastur kodi	Dastur natijasi
import math y=int(input()) x=int(input()) tau=math.atan(y/x) print(int(round(math.degrees(tau),0)),sep='')	3 4 37

2.11-masala

a, b, c, d 4 ta butun sonni kiritib, $a^b + c^d$ natijani chiqarish uchun Python dasturini yozing.

Pythonda butun sonlar sizning mashinangiz xotirasi kabi katta hajmda bo`lishi mumkin. Hajm bo`yicha chegara yo`q. Ko`rinib turganidek, natijada b o`sishi bilan $a^b$ juda tez o`sadi. long int butun sonlar ustida bir necha hisoblashlarni olib boramiz.

Kiritish formati:

a, b, c, d 4 ta sonlarni alohida-alohida qatorlardan o`qilsin.

Shartlar

$1 \leq a \leq 1000$

$1 \leq b \leq 1000$

$1 \leq c \leq 1000$

$1 \leq d \leq 1000$

Chiqarish formati:

$a^b + c^d$ natija 1 qatorda chop etilsin

Kiritish namuna:

9

29

7

27

Chiqarish namuna:

47101944096086083692017432 32

Eslatma . Chiqarilgan natija $2^{63} - 1$ dan ancha katta, C++ dagi long long int yoki 64-bitli butun songa to`g`ri kelmaydi.

Dastur kodi	Dastur natijasi
a,b,c,d=int(input()),int(input()),int(input()),int(input()) print(a**b+c**d)	58 49 75 23 2558419075985223 2758592062951454 1067041053768046 8616890963178621 2850894772684234 8414403

2.12-masala

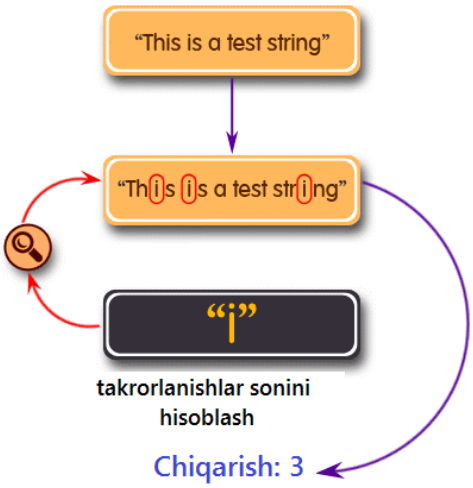
Ixtiyoriy haqiqiy sonning n-butun darajasini hisoblash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
import math def myPow(x,n): return math.pow(x,n) print(myPow(2.0000,10)) print(myPow(-5.68,25))	1024.0 -7.2228466238596e+18

5-§. SATRLI MASALALAR

2.13-masala

Pythonda original satrdagi berilgan simvollar sonini topuvchi dastur tuzing.



1-usul: count() funksiyasi orqali aniqlash.

Dastur kodi	Dastur natijasi
s = "Kamtarga kamol, manmanga zavol." print("Original satr:") print(s) print("Satrdagi 'a' simvollari soni:") print(s.count("a"))	Original satr: Kamtarga kamol, manmanga zavol. Satrdagi 'a' simvollari soni: 8

2-usul: collections.Counter() ni qo`llash.

Dastur kodi	Dastur natijasi
from collections import Counter s = "Kamtarga kamol, manmanga zavol." print("Original satr:") print(s) print("Satrdagi 'o' simvollari soni:") ctr = Counter(s) print (str(ctr['o']))	Original satr: Kamtarga kamol, manmanga zavol. Satrdagi 'o' simvollari soni: 2

Ushbu masalaning qolgan yechimlari 3.10-masalada berilgan.

2.14-masala

Pythonda berilgan simvolning ASCII jadvalidagi kodini chiqaruvchi dastur tuzing.

ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

ASCII (inglizcha: American Standard Code for Information Interchange) — bosma belgilar va boshqa maxsus kodlar uchun Amerika Qo‘shma Shtatlari standart kodlash jadvali.

ASCII o‘nlik sonlar, lotin va milliy alifbolar, tinish belgilari va boshqaruvchi belgilarni tasvirlovchi kodlashlarni o‘z ichiga oladi. Dastlab 7-bitlik qilib yaratilgan, keyinchalik 8-bitlik baytga o‘tkazilgan ASCII 8-bitlikning yarmi deb qabul qilina boshlandi. Komputerlarda odatda 8-bit va kod jadvalining ikkinchi yarmi bilan ishlangan ASCII kengaytmasi foydalaniladi.

Dastur kodi	Dastur natijasi
print() print(ord('a')) print(ord('A')) print(ord('1')) print(ord('@'))	 97 65 49 64

2.15-masala

Berilgan bo'sh bo'lmagan satrlar ro'yxatining uzunligini topish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
def test(strs): return [*map(len, strs)] strs = ['mushuk', 'mashina', 'yaqin', 'markaz'] print("Original satr:") print(strs) print("Bo'sh bo'lmagan satrlar ro'yxatining uzunligi:") print(test(strs)) strs = ['mushuk', 'it', 'xona', 'shirinlik', '', 'yedi', ''] print("\nOriginal satr:") print(strs) print("Bo'sh bo'lmagan satrlar ro'yxatining uzunligi:") print(test(strs))	Original satr: ['mushuk', 'mashina', 'yaqin', 'markaz'] Bo'sh bo'lmagan satrlar ro'yxatining uzunligi: [6, 7, 5, 6] Original satr: ['mushuk', 'it', 'xona', 'shirinlik', '', 'yedi', ''] Bo'sh bo'lmagan satrlar ro'yxatining uzunligi: [6, 2, 4, 9, 0, 4, 0]

2.16-masala

Berilgan satrdagi alohida belgilar to‘plamini topish uchun Python dasturini yozing (katta-kichik harflarga e’tibor berilmasin).

Dastur kodi	Dastur natijasi
def test(strs): return [*set(strs.lower())] strs = "SALOM" print("Original satr:",strs) print("Satrdagi belgilar:") print(test(strs)) strs = "Salom" print("\nOriginal satr:",strs) print("Satrdagi belgilar:") print(test(strs)) strs = "Ignoring case" print("\nOriginal satr:",strs) print("Satrdagi belgilar:") print(test(strs))	Original satr: SALOM Satrdagi belgilar: ['h', 'l', 'e', 'o'] Original satr: Salom Satrdagi belgilar: ['h', 'l', 'e', 'o'] Original satr: Ignoring case Satrdagi belgilar: [' ', 'r', 'i', 'c', 'g', 'o', 's', 'a', 'n', 'e']

2.17-masala

Berilgan butun son palindrom yoki yo'qligini tekshirish uchun Python dasturini yozing.

Eslatma: Butun son palindrom bo`lishi uchun oldindan yoki orqadan o`qilganda bir xil natija qaytarishi kerak. Masalan, 1331-palindrom. 9654-palindrom emas. Manfiy sonlar palindrom emas.

Dastur kodi	Dastur natijasi
def Palindrom(n): return str(n) == str(n)[::-1] print(Palindrom(100)) print(Palindrom(252)) print(Palindrom(-838))	False True False

Ushbu masalaning qolgan yechimlari 5.10 hamda 6.39- masalalarda berilgan.

2.18-masala

Satrdagi birinchi belgining barcha ko'rinishlari "$" ga o'zgartirish uchun Python dasturini yozing. (birinchi belgidan tashqari).

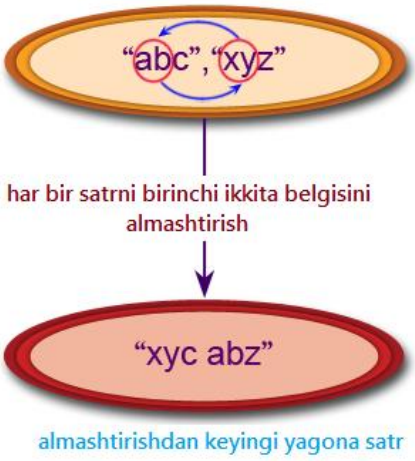
string.replace(1-qiymat,2-qiymat,hisoblagich)-satr metodi berilgan satrdagi ko`rsatilgan belgini(1-qiymat) parametr sifatida uzatilgan yangi qiymatga (2-qiymatga) o`zgartirib, yangi satrosti yaratib beradi.

Dastur kodi	Dastur natijasi

def belgi_uzgar(str1):
  belgi = str1[0]
  str1 = str1.replace(belgi, '$')
  str1 = belgi + str1[1:]
  return str1
print(belgi_uzgar('restart'))
	resta$t

2.19-masala

Bo'sh joy bilan ajratilgan ikkita berilgan satrdan bitta satr olish uchun Python dasturini yozing va har bir satrning birinchi ikkita belgisini almashtiring.

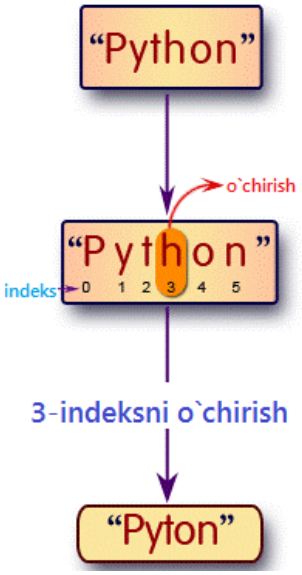


Dastur kodi	Dastur natijasi

def belgilar_mix(a, b):
  a1 = b[:2] + a[2:]
  b1 = a[:2] + b[2:]
 	
  return a1 + ' ' + b1
print(belgilar_mix('abc', 'xyz'))
	xyc abz

2.20-masala

Bo'sh bo'lmagan satrdan n-chi indeks belgisini olib tashlash uchun Python dasturini yozing.



Dastur kodi	Dastur natijasi
def belgi_uchir(str, n): qism1 = str[:n] qism2 = str[n+1:] return qism1 + qism2 print(belgi_uchir('Python', 0)) print(belgi_uchir('Python', 3)) print(belgi_uchir('Python', 5))	ython Pyton Pytho

2.21-masala

Berilgan satrni birinchi va oxirgi belgilarini almashtirib yangi satrga chiqarish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
def satr_uzgar(str1): return str1[-1:] + str1[1:-1] + str1[:1] print(satr_uzgar('abcd')) print(satr_uzgar('12345'))	dbca 52341

2.22-masala

Berilgan satrdan bo'sh joylarni olib tashlash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def probel_ochir(str1):
  str1 = str1.replace(' ','')
  return str1
 	
print(probel_ochir("O ` z b e k i s t o n"))
print(probel_ochir("a b c"))
	O`zbekiston
	abc

2.23-masala

Berilgan satrning takroriy belgilarni olib tashlash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
from collections import OrderedDict def dublikat_ochir(str1): return "".join(OrderedDict.fromkeys(str1)) print(dublikat_ochir("python dasturlash tili")) print(dublikat_ochir("amaliyot darsi"))	python dasurli amliyot drs

2.24-masala

So'zlarni qo'shish

Argument sifatida bir nechta so'zlarni oladigan va chiziqcha (-) bilan ajratilgan bu so'zlarning birlashtirilgan versiyasini qaytaradigan funksiyani yozishingiz kerak. Funktsiya argument sifatida har xil sonli so'zlarni olishi kerak.

Kiritish namuna:

this

is

great

Chiqarish namuna:

this-is-great

Eslatib o‘tamiz, funksiya parametri sifatida *args dan foydalanish ushbu funktsiyaga ixtiyoriy sonli argumentlarni o‘tkazish imkonini beradi.

Dastur kodi	Dastur natijasi
def concatenate(*args): return "-".join(args) print(concatenate("I", "love", "Python", "!")) print(concatenate("Python", "dasturlash", "tilini", "o`rganing", "!")	I-love-Python-! Python-dasturlash-tilini-o`rganing-!

6-§. SANA VA VAQTGA DOIR MASALALAR

2.25-masala

Joriy sana va vaqtni chiqarish dasturini tuzing.

datetime - sana va vaqt funksiyalari kutubxonasi

Bu modulga quyidagi tip obyektlari kiradi:

a) date-sanani saqlaydi;

b) time-vaqtni saqlaydi;

c) datetime-sana va vaqtni saqlaydi.

datetime.now(tz=None)-mahalliy joriy sana va vaqtni qaytaradi.

date.strftime(format)-sanani ko`rsatadigan ma`lum formatdagi satrni qaytaradi.

Dastur kodi	Dastur natijasi
import datetime hozir = datetime.datetime.now() print ("Joriy sana va vaqt: ") print (hozir.strftime("%Y-%m-%d %H:%M:%S"))	Joriy sana va vaqt: 2022-05-24 09:31:30

2.26-masala

Yil va oy kiritilgandan so`ng taqvimni chiqarish dasturini tuzing.

Dastur kodi	Dastur natijasi
import calendar y = int(input("Yilni kiriting: ")) m = int(input("Oyni kiriting: ")) print(calendar.month(y, m))	Yilni kiriting: 2022 Oyni kiriting: 8 August 2022 Mo Tu We Th Fr Sa Su 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

2.27-masala

Sanani (2022/1/1 dan 2022/12/31 gacha) o'qish va sana kunini chop etish Python dasturini yozing.

Dastur kodi	Dastur natijasi
from datetime import date print("Oy va sanani kiriting \n(bitta bo'sh joy bilan ajratilgan):") oy, sana = map(int, input().split()) hafta= {1:'Dushanba',2:'Seshanba',3:'Chorshanba',4:'Payshanba',5:'Juma', 6:'Shanba',7:'Yakshanba'} kun = date.isoweekday(date(2022, oy, sana)) print("Hafta kuni: ",hafta[kun])	Oy va sanani kiriting (bitta bo'sh joy bilan ajratilgan): 2 27 Hafta kuni: Yakshanba

2.28-masala

Ikkita sana beriladi. Shu sanalar orasidagi kunni hisoblash dasturini tuzing.

Dastur kodi	Dastur natijasi
from datetime import date sana_1 = date(2021, 10, 2) sana_2 = date(2021, 11, 30) kunlar = sana_2 - sana_1 print(kunlar.days)	59

2.29-masala

Muayyan millisekundlardan keyin berilgan funksiyani chaqirish Python dasturini yozing.

Dastur kodi	Dastur natijasi

from time import sleep
import math
def delay(fn, ms, *args):
  sleep(ms / 1000)
  return fn(*args)
print("Birozdan so`ng kvadrat ildiz chiqarish:")
print(delay(lambda x: math.sqrt(x), 100, 16))
print(delay(lambda x: math.sqrt(x), 1000, 100))
print(delay(lambda x: math.sqrt(x), 2000, 25100))
	Birozdan so`ng kvadrat
	ildiz chiqarish:
	4.0
	10.0
	158.42979517754858

7-§. RO`YXAT, KORTEJ, LUG`AT VA TO`PLAMGA DOIR MASALALAR

2.30-masala

Berilgan sonlar ketma-ketligini ro`yxat va kortejga o`tkazish dasturini tuzing.

Dastur kodi	Dastur natijasi
qiymat = input("Sonlarni kiriting(vergul bilan):\n") ruyxat = qiymat.split(",") kortej = tuple(ruyxat) print('Ro`yxat : ',ruyxat) print('Kortej : ',kortej)	Sonlarni kiriting(vergul bilan): 25,12,68,35,98 Ro`yxat : ['25', '12', '68', '35', '98'] Kortej : ('25', '12', '68', '35', '98')

2.31-masala

Ranglar ro`yxat ko`rinishida berilgan. Shu ro`yxatdagi birinchi va oxirgi rangni chiqarish dasturini tuzing.

Dastur kodi	Dastur natijasi
ranglar = ["Qizil","Yashil","Oq" ,"Qora"] print("%s %s"%(ranglar[0],ranglar[-1]))	Qizil Qora

2.32-masala

Takrorlanuvchiga hisoblagichlarni qo'shish dasturini tuzing. Hisoblagichlar takrorlanadigan ro'yxatni (yoki kortej, satr yoki lug'at va hokazo) qaytaradi va qaytarilgan obyektni sanab bo'ladi.

1-usul:

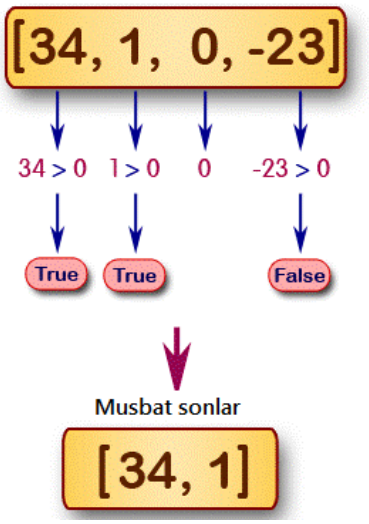
Dastur kodi	Dastur natijasi
satr="Python" l=enumerate(satr) print(list(l))	[(0, 'P'), (1, 'y'), (2, 't'), (3, 'h'), (4, 'o'), (5, 'n')]

2-usul:

Dastur kodi	Dastur natijasi
satr=["ko`k", "qizil", "qora", "yashil", "havorang"] l=enumerate(satr,10) print(list(l))	[(10, 'ko`k'), (11, 'qizil'), (12, 'qora'), (13, 'yashil'), (14, 'havorang')]

2.33-masala

Python dasturida berilgan ro`yxatdagi musbat sonlarni filtrlash dasturini tuzing.



Dastur kodi	Dastur natijasi
ruyxat = [34, 1, 0, -23, 12, -88] print("Ro`yxatdagi barcha sonlar:\n",ruyxat) yangi_ruyxat = list(filter(lambda x: x >0, ruyxat)) print("Ro`yxatdagi musbat sonlar:\n",yangi_ruyxat)	Ro`yxatdagi barcha sonlar: [34, 1, 0, -23, 12, -88] Ro`yxatdagi musbat sonlar: [34, 1, 12]

Ushbu masalaning qolgan yechimlari 5.3- masalada berilgan.

2.34-masala

Ranglar ro`yxati berilgan. 1-elementi o`chirilgandan so`ng, yangi ro`yxatni chiqarish dasturini tuzing.

1-usul:

Dastur kodi	Dastur natijasi
rang = ["Qizil", "Qora", "Yashil", "Oq", "Sabzirang"] print("Haqiqiy ro`yxat elementlari:") print(rang) del rang[0] print("1-rang o`chirilgandan so`ng:") print(rang)	Haqiqiy ro`yxat elementlari: ['Qizil', 'Qora', 'Yashil', 'Oq', 'Sabzirang'] 1-rang o`chirilgandan so`ng: ['Qora', 'Yashil', 'Oq', 'Sabzirang']

2-usul:

Dastur kodi	Dastur natijasi
rang = ["Qizil", "Qora", "Yashil", "Oq", "Sabzirang"] print("Haqiqiy ro`yxat elementlari:") print(rang) print("1-rang o`chirilgandan so`ng:") yangi_rang = rang[1:] print(yangi_rang)	Haqiqiy ro`yxat elementlari: ['Qizil', 'Qora', 'Yashil', 'Oq', 'Sabzirang'] 1-rang o`chirilgandan so`ng: ['Qora', 'Yashil', 'Oq', 'Sabzirang']

3-usul:

Dastur kodi	Dastur natijasi
rang = ["Qizil", "Qora", "Yashil", "Oq", "Sabzirang"] print("Joriy ro`yxat elementlari:") print(rang) print("1-elementi o`chirilgandan keyin:") rang.remove("Qizil") print(rang)	Joriy ro`yxat elementlari: ['Qizil', 'Qora', 'Yashil', 'Oq', 'Sabzirang'] 1-elementi o`chirilgandan keyin: ['Qora', 'Yashil', 'Oq', 'Sabzirang']

2.35-masala

Ro'yxat, to'plam, lug'at (qiymat) va katalogdan fayldan tasodifiy elementni tanlash uchun Python dasturini yozing. random.choice dan foydalaning.

random moduli tasodifiy sonlar, harflar, ketma-ketlik elementlarini tasodifiy tanlash funksiyalarini o`z ichiga oladi:

1) **random.choice(sequence)**-bo`sh bo`lmagan ketma-ketlikning ixtiyoriy elementi

2) **random.shuffle(sequence, [rand])**-ketma-ketlikni aralashtirib yuboradi (tartibning o'zini o'zgartiradi).
3) **random.random()**-(0;1) oralig`idagi ixtiyoriy son.
4) **random.randrange(start, stop, qadam)**-ketma-ketlikning ixtiyoriy sonini tanlab beradi.
5) **random.randint(A, B)**- $A \leq N \leq B$ oraliqdagi ixtiyoriy butun son.

Dastur kodi	Dastur natijasi
import random import os print("Ro`yxatdan tasodifiy elementni tanlang:") sonlar = [1, 2, 3, 4, 5] print(random.choice(sonlar)) print(random.choice(sonlar)) print(random.choice(sonlar)) print("\nTo`plamdan tasodifiy elementni tanlang:") sonlar = set([1, 2, 3, 4, 5]) print(random.choice(tuple(sonlar))) print(random.choice(tuple(sonlar))) print(random.choice(tuple(sonlar))) print("\nLug`atdan tasodifiy elementni tanlang:") d = {"a": 1, "b": 2, "c": 3, "d": 4, "e": 5} kalit = random.choice(list(d)) print(d[kalit]) kalit = random.choice(list(d)) print(d[kalit]) kalit = random.choice(list(d)) print(d[kalit]) print("\nKatalogdan tasodifiy elementni tanlang:") print(random.choice(os.listdir("/")))	Ro`yxatdan tasodifiy elementni tanlang: 1 5 2  To`plamdan tasodifiy elementni tanlang: 3 3 1 Lug`atdan tasodifiy elementni tanlang: 2 1 1 Katalogdan tasodifiy elementni tanlang: Program Files

2.36-masala

Tasodifiy sonlar generatorini yaratish uchun Python dasturini yozing, shuningdek, 1 dan tashqari 0 dan 1 gacha bo'lgan float sonini hosil qiling. random.random() dan foydalaning.

Dastur kodi	Dastur natijasi
import random print("Tasodifiy sonlar generatori:") print(random.Random().random()) print(random.Random(0).random()) print("\n0 dan 1 gacha float tipidagi tasodifiy son:") print(random.random())	Tasodifiy sonlar generatori: 0.8790728355726944 0.8444218515250481 0 dan 1 gacha float tipidagi tasodifiy sonlar: 0.6251105208700517

2.37-masala

Berilgan ro‘yxat elementlarini aralashtirish uchun Python dasturini yozing. random.shuffle() dan foydalaning.

Dastur kodi	Dastur natijasi
import random sonlar = [1, 2, 3, 4, 5] print("Original ro`yxat:") print(sonlar) random.shuffle(sonlar) print("Aralashtirilgan ro`yxat:") print(sonlar) suzlar = ['qizil', 'qora', 'yashil', 'ko`k'] print("\nOriginal ro`yxat:")	Original ro`yxat: [1, 2, 3, 4, 5] Aralashtirilgan ro`yxat: [3, 2, 1, 4, 5] Original ro`yxat: ['qizil', 'qora', 'yashil', 'ko`k'] Aralashtirilgan ro`yxat: ['yashil', 'ko`k', 'qora', 'qizil']

print(suzlar) random.shuffle(suzlar) print("Aralashtirilgan ro`yxat:") print(suzlar)	

2.38-masala

Berilgan ro‘yxatning sayoz nusxasini yaratish uchun Python dasturini yozing. copy.copy dan foydalaning.

copy moduli- Belgilash operatori obyektdan nusxa ko'chirmaydi, faqat ob'ektga havola yaratadi. Ushbu modul umumiy (tashqi, chuqur (to`la)) nusxa ko'chirish operatsiyalarini o`z ichiga oladi:

1) **copy.copy(x)-**x ning tashqi nusxasini qaytaradi
2) **copy.deepcopy(x)**-x ning to`la nusxasini qaytaradi.

Tashqi nusxa yangi kompozit obyektni yaratadi va keyin (agar iloji bo'lsa) unga asl nusxadagi obyektlarga havolalarni kiritadi.

Chuqur nusxa yangi kompozit obyektni yaratadi va keyin asl nusxadagi obyektlarning nusxalarini unga rekursiv ravishda kiritadi.

Dastur kodi	Dastur natijasi
import copy son = [1, [2, 3, 4]] print("Original ro`yxat: ", son) son2 = copy.copy(son) print("\nBerilgan ro`yxatning nusxasi:") print(son2) print("\nOriginal ro`yxat o`zgartirildi:") son[1][1] = 10 print(son) print("\n2-ro`yxat:") print(son2)	Original ro`yxat:  [1, [2, 3, 4]]  Berilgan ro`yxatning nusxasi: [1, [2, 3, 4]] Original ro`yxat o`zgartirildi: [1, [2, 10, 4]] 2-ro`yxat: [1, [2, 10, 4]]

sonlar = [[1], [2]] sonlar2 = copy.copy(sonlar) print("\nOriginal ro`yxat:") print(sonlar) print("\nBerilgan ro`yxatning nusxasi:") print(sonlar2) print("\nOriginal ro`yxat o`zgartirildi:") sonlar[0][0] = 0 print("\n1-ro`yxat:") print(sonlar) print("\n2-ro`yxat:") print(sonlar2)	Original ro`yxat: [[1], [2]]  Berilgan ro`yxatning nusxasi: [[1], [2]] Original ro`yxat o`zgartirildi: 1-ro`yxat: [[0], [2]]  2-ro`yxat: [[0], [2]]

2.39-masala

Berilgan ro‘yxatning chuqur nusxasini yaratish uchun Python dasturini yozing. copy.copy dan foydalaning.

Dastur kodi	Dastur natijasi
import copy son1 = [1, [2, 3, 4]] print("Original ro`yxat: ", son1) son2 = copy.deepcopy(son1) print("\nBerilgan ro`yxatning chuqur nusxasi:") print(son2) print("\nOriginal ro`yxat o`zgartirildi:") son1[1][1] = 10 print(son1) print("\n2-ro`yxatning nusxasi (Chuqur nusxa):") print(son2)	Original ro`yxat: [1, [2, 3, 4]]  Berilgan ro`yxatning chuqur nusxasi: [1, [2, 3, 4]] Original ro`yxat o`zgartirildi: [1, [2, 10, 4]]

sonlar1 = [[1, 2, 3], [4, 5, 6]] sonlar2 = copy.deepcopy(sonlar1) print("\nOriginal ro`yxat:") print(sonlar1) print("\nBerilgan ro`yxatning chuqur nusxasi:") print(sonlar2) print("\nOriginal ro`yxat o`zgartirildi:") sonlar1[0][2] = 55 sonlar1[1][1] = 77 print("\nOriginal ro`yxat:") print(sonlar1) print("\n2-ro`yxatning nusxasi (Chuqur nusxa):") print(sonlar2)	2-ro`yxatning nusxasi (Chuqur nusxa): [1, [2, 3, 4]]  Original ro`yxat: [[1, 2, 3], [4, 5, 6]] Berilgan ro`yxatning chuqur nusxasi: [[1, 2, 3], [4, 5, 6]]  Original ro`yxat o`zgartirildi:  Original ro`yxat: [[1, 2, 55], [4, 77, 6]] 2-ro`yxatning nusxasi (Chuqur nusxa): [[1, 2, 3], [4, 5, 6]]

2.40-masala

Berilgan lug'atning chuqur nusxasini yaratish uchun Python dasturini yozing. copy.copy dan foydalaning.

Dastur kodi	Dastur natijasi
import copy son1 = {"a":1, "b":2, 'cc':{"c":3}} print("Original lug`at:", son1) son2 = copy.copy(son1)	Original lug`at: {'b': 2, 'a': 1, 'cc': {'c': 3}} Berilgan lug`atning nusxasi: {'b': 2, 'a': 1, 'cc': {'c': 3}}

print("Berilgan lug`atning nusxasi:") print(son2) print("Original lug`at elementi o`zgartirildi:") son1["cc"]["c"] = 10 print(son1) print("2-lug`at(nusxasi):") print(son2) sonlar1 = {"x":1, "y":2, 'zz':{"z":3}} sonlar2 = copy.copy(sonlar1) print("Original lug`at:") print(sonlar1) print("Berilgan lug`atning nusxasi:") print(sonlar2) print("Original lug`at elementi o`zgartirildi:") sonlar1["zz"]["z"] = 10 print("1-lug`at:") print(sonlar1) print("2-lug`at(nusxasi):") print(sonlar2)	Original lug`at elementi o`zgartirildi: {'b': 2, 'a': 1, 'cc': {'c': 10}} 2-lug`at(nusxasi): {'b': 2, 'a': 1, 'cc': {'c': 10}} Original lug`at: {'x': 1, 'y': 2, 'zz': {'z': 3}} Berilgan lug`atning nusxasi: {'x': 1, 'y': 2, 'zz': {'z': 3}} Original lug`at elementi o`zgartirildi: 1-lug`at: {'x': 1, 'y': 2, 'zz': {'z': 10}} 2-lug`at(nusxasi): {'x': 1, 'y': 2, 'zz': {'z': 10}}

2.41-masala

Bo‘sh bo‘lmagan kortejlar ro‘yxatidan har bir kortejdagi oxirgi element ortib borishi bo`yicha tartiblangan ro‘yxatni olish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
def last(n): return n[-1] def oxirgi_element_tartiblash(tuples): return sorted(tuples, key=last) print(oxirgi_element_tartiblash([(1, 2), (4, 4), (2, 1)]))	[(2, 1), (1, 2), (4, 4)]

2.42-masala

Ikki ro'yxat orasidagi farqni olish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
ruyxat1 = [1, 3, 5, 7, 9] ruyxat2 =[1, 2, 4, 6, 7, 8] ruyxat1_ruyxat2_ayirma = list(set(ruyxat1) - set(ruyxat2)) ruyxat2_ruyxat1_ayirma = list(set(ruyxat2) - set(ruyxat1)) umumiy_farq = ruyxat1_ruyxat2_ayirma + ruyxat2_ruyxat1_ayirma print(umumiy_farq)	[9, 3, 5, 8, 2, 4, 6]

2.43-masala

Ro'yxatdagi bir xil qiymatlarni o`chirish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
ruyxat = [10, 20, 30, 40, 20, 50, 60, 40] print("Original ro`yxat : ",ruyxat) tuplam = set(ruyxat) ruyxat2 = list(tuplam) print("Bir xil elementlar o`chirildi: ",ruyxat2)	Original ro`yxat : [10, 20, 30, 40, 20, 50, 60, 40] Bir xil elementlar o`chirildi: [40, 10, 50, 20, 60, 30]

2.44-masala

1 dan n gacha bo'lgan oraliqda berilgan ro'yxatni birlashtirish orqali ro'yxat yaratish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
ruyxat = ['p', 'q'] n = 4 ruyxat2 = ['{}{}'.format(x, y) for y in range(1, n+1) for x in ruyxat] print(ruyxat2)	['p1', 'q1', 'p2', 'q2', 'p3', 'q3', 'p4', 'q4']

2.45-masala

Berilgan raqamlar ro'yxatining har bir sonini yaxlitlash uchun Python dasturini yozing va umumiy yig`indini ro'yxat uzunligiga ko'paytiring.

Dastur kodi	Dastur natijasi
sonlar = [22.4, 4.0, -16.22, -9.10, 17.50] print("Original ro`yxat: ", sonlar) print("Natija:") uzunlik=len(sonlar) print(sum(list(map(round,sonlar))* uzunlik))	Original ro`yxat: [22.4, 4.0, -16.22, -9.1, 17.5] Natija: 95

2.46-masala

Ro‘yxat elementlarini birlashtirish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
rang = ['qizil', 'yashil', 'sariq'] print('-'.join(rang)) print(''.join(rang))	qizil-yashil-sariq qizilyashilsariq

2.47-masala

Barcha nol raqamlarni berilgan raqamlar ro‘yxatining oxiriga ko‘chirish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def test(lst):
  result = sorted(lst, key=lambda x: not x)
  return result
nums = [3,4,0,0,0,9,10,7,4,4,5,3,0,0,2,9,7,1]
print("\nOriginal ro`yxat:")
print(nums)
print("\nBarcha nollarni ro'yxat oxiriga o'tkazish:")
print(test(nums))
	Original ro`yxat:
	[3, 4, 0, 0, 0, 9, 10, 7, 4, 4, 5, 3, 0, 0, 2, 9, 7, 1]
	
	Barcha nollarni ro'yxat oxiriga o'tkazish:
	[3, 4, 9, 10, 7, 4, 4, 5, 3, 2, 9, 7, 1, 0, 0, 0, 0, 0]

2.48-masala

Yangi lug'at yaratish uchun quyidagi lug'atlarni birlashtirish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
dic1={1:10, 2:20} dic2={3:30, 4:40} dic3={5:50,6:60} dic4 = {} for d in (dic1, dic2, dic3): dic4.update(d) print(dic4)	{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

2.49-masala

Lug'atda maksimal va minimal qiymatlarni olish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
tuplam = {'x':500, 'y':5874, 'z': 560} key_max = max(tuplam.keys(), key=(lambda k: tuplam[k])) key_min = min(tuplam.keys(), key=(lambda k: tuplam[k])) print('Eng katta qiymat: ',tuplam[key_max]) print('Eng kichik qiymat: ',tuplam[key_min])	Eng katta qiymat: 5874 Eng kichik qiymat: 500

2.50-masala

Obyekt maydonlaridan lug'at olish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

class lugat(object):
  def __init__(self):
  self.x = 'qizil'
  self.y = 'sariq'
  self.z = 'yashil'
  def do_nothing(self):
  pass
test = lugat()
print(test.__dict__)
	{'x': 'qizil', 'y': 'sariq', 'z': 'yashil'}

2.51-masala

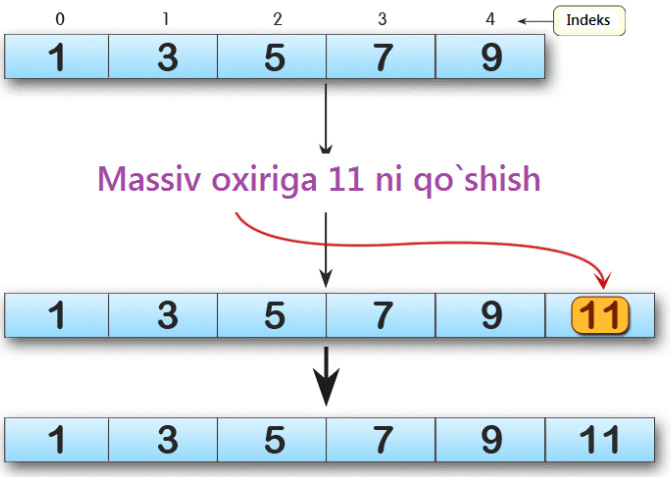
Lug`atlarning umumiy qiymatlarini qo'shib birlashtirish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
from collections import Counter d1 = {'a': 100, 'b': 200, 'c':300} d2 = {'a': 300, 'b': 200, 'd':400} d = Counter(d1) + Counter(d2) print(d)	Counter({'b': 400, 'd': 400, 'a': 400, 'c': 300})

8-§. MASSIVGA DOIR MASALALAR

2.52-masala

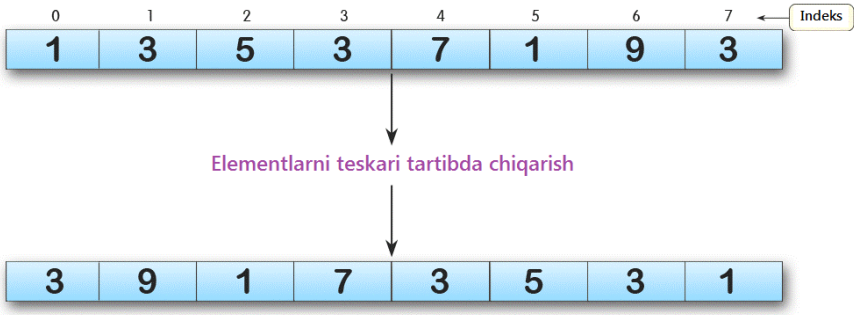
Massiv oxiriga yangi element qo'shish uchun Python dasturini yozing.



Dastur kodi	Dastur natijasi
from array import * array_num = array('i', [1, 3, 5, 7, 9]) print("Original massiv: "+str(array_num)) print("Massiv oxiriga 11 elementini qo`shish:") array_num.append(11) print("Yangi massiv: "+str(array_num))	Original massiv: array('i', [1, 3, 5, 7, 9]) Massiv oxiriga 11 elementini qo`shish: Yangi massiv: array('i', [1, 3, 5, 7, 9, 11])

2.53-masala

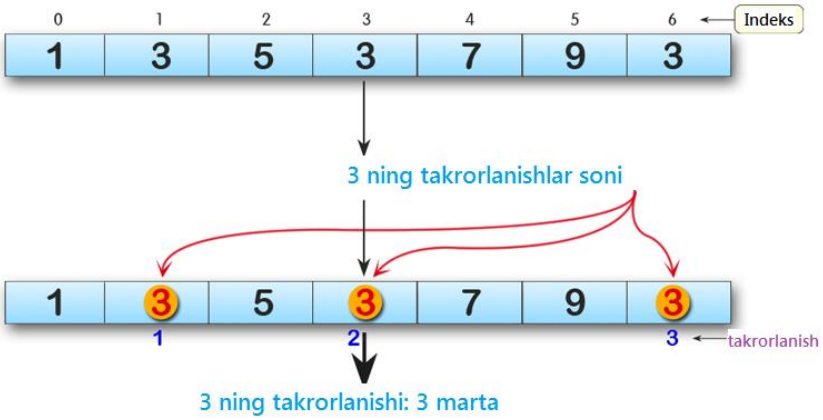
Massivdagi elementlarning tartibini o'zgartirish uchun Python dasturini yozing.



Dastur kodi	Dastur natijasi
from array import * array_num = array('i', [1, 3, 5, 3, 7, 1, 9, 3]) print("Original massiv: "+str(array_num)) array_num.reverse() print("Elementlarning tartibini o`zgartirish:") print(str(array_num))	Original massiv: array('i', [1, 3, 5, 3, 7, 1, 9, 3]) Elementlarning tartibini o`zgartirish: array('i', [3, 9, 1, 7, 3, 5, 3, 1])

2.54-masala

Massivda belgilangan elementning takrorlanish sonini olish uchun Python dasturini yozing.



Dastur kodi	Dastur natijasi
from array import * array_num = array('i', [1, 3, 5, 3, 7, 5, 9, 3]) print("Original massiv: "+str(array_num)) print("5 raqamining takrorlanishi: "+str(array_num.count(5)))	Original massiv: array('i', [1, 3, 5, 3, 7, 5, 9, 3]) 5 raqamining takrorlanishi: 2

2.55-masala

Belgilangan ro'yxatdagi elementlarni massivga qo'shish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
from array import * num_list = [1, 2, 6, -8] array_num = array('i', []) print("Ro`yxatdagi elementlar: " + str(num_list)) print("Ro'yxatdagi elementlarni qo'shish: ") array_num.fromlist(num_list) print("Massivdagi elementlar: "+str(array_num))	Ro`yxatdagi elementlar: [1, 2, 6, -8] Ro'yxatdagi elementlarni qo'shish: Massivdagi elementlar: array('i', [1, 2, 6, -8])

2.56-masala

Mavjud massivning ikkinchi elementidan oldin yangi element qo'shish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
from array import * array_num = array('i', [1, 3, 5, 7, 9]) print("Original massiv: "+str(array_num)) print("3 elementdan oldin 4 ni qo`shish:") array_num.insert(1, 4) print("Yangi massiv: "+str(array_num))	Original massiv: array('i', [1, 3, 5, 7, 9]) 3 elementdan oldin 4 ni qo`shish: Yangi massiv: array('i', [1, 4, 3, 5, 7, 9])

9-§. SINFGA DOIR MASALALAR

2.57-masala

Satrni so'zma-so'z o'zgartirish uchun Python sinfini yozing. Satrni so‘zma-so‘z teskari aylantiring.

Dastur kodi	Dastur natijasi

class py_solution:
  def reverse_words(self, s):
  return ' '.join(reversed(s.split()))
 	
print(py_solution().reverse_words('Python dasturlash tili'))
	tili dasturlash Python

2.58-masala

Get_String va print_String ikkita usuliga ega Python sinfini yozing. get_String foydalanuvchidan satrni qabul qiladi va print_String satrni katta harf bilan chop etadi.

Dastur kodi	Dastur natijasi

class IOString():
  def __init__(self):
  self.str1 = ""
 	
  def get_String(self):
  self.str1 = input()
 	
  def print_String(self):
  print(self.str1.upper())
	python dasturlash tili
	PYTHON DASTURLASH TILI

str1 = IOString() str1.get_String() str1.print_String()	

2.59-masala

Eni va bo`yi bo'yicha tuzilgan Rectangle nomli Python sinfini va to'rtburchakning yuzasini hisoblash dasturini yozing.

Dastur kodi	Dastur natijasi

class Rectangle():
  def __init__(self, l, w):
  self.length = l
  self.width = w
 	
  def rectangle_area(self):
  return self.length*self.width
 	
newRectangle = Rectangle(18, 12)
print(newRectangle.rectangle_area())
	216

2.60-masala

Radius bo'yicha qurilgan Circle nomli Python sinfini, doira maydoni va uzunligini hisoblash dasturini yozing.

Dastur kodi	Dastur natijasi

class Circle():
  def __init__(self, r):
  self.radius = r
	314.0
	62.800000000000004

def area(self):
  return self.radius**2*3.14
 	
def length(self):
  return 2*self.radius*3.14
 	
NewCircle = Circle(10)
print(NewCircle.area())
print(NewCircle.length())	

2.61-masala

Berilgan sinfdagi misollarni va kichik sinflarni tekshiring. Ikkita bo'sh sinfni, Student va Marksni yaratish uchun Python dasturini yozing. Shuningdek, ushbu sinflar o'rnatilgan obyekt sinfining pastki sinflari yoki yo'qligini tekshiring.

Dastur kodi	Dastur natijasi

class Student:
  pass
class Marks:
  pass
student1 = Student()
marks1 = Marks()
print(isinstance(student1, Student))
print(isinstance(marks1, Student))
print(isinstance(marks1, Marks))
print(isinstance(student1, Marks))
print("Sinfning sinfostilarini tekshiring.")
print(issubclass(Student, object))
print(issubclass(Marks, object))
	True
	False
	True
	False
	Sinfning
	sinfostilarini
	tekshiring.
	True
	True

3-BOB. TARMOQLANUVCHI ALGORITMLARGA DOIR DASTURLAR

10-§. ARIFMETIK MASALALAR

3.1-masala

Raqamlar ketma-ketligidan barcha raqamlar bir-biridan farq qilishini aniqlash uchun Python funksiyasini yozing.

Dastur kodi	Dastur natijasi

def test_unikal(ruyxat):
  if len(ruyxat) == len(set(ruyxat)):
  return True
  else:
  return False;
print(test_unikal([1,5,7,9]))
print(test_unikal([2,4,5,5,7,9]))
	True
	False

3.2-masala

Pythonda float tipidagi berilgan sonning musbat, manfiy yoki nol ekanligini tekshirish dasturini tuzing.

1-usul:

Dastur kodi	Dastur natijasi

son = float(input("Sonni kiriting: "))
if son > 0:
  print("Musbat son.")
elif son == 0:
  print("0 soni")
else:
  print("Manfiy son.")
	Sonni kiriting:
	85.658002
	Musbat son.

2-usul:

Dastur kodi	Dastur natijasi
n = float(input('Sonni kiriting: ')) print('Musbat son' if n > 0 else '0 soni' if n == 0 else 'Manfiy son')	Sonni kiriting: 6578411 Musbat son

3-usul:

Dastur kodi	Dastur natijasi

n = float(input("Sonni kiriting: "))
if n >= 0:
  if n == 0:
  print("0 soni")
  else:
  print("Musbat son")
else:
  print("Manfiy son")
	Sonni kiriting:
	6578411
	Musbat son

3.3-masala

Universitet sport kuni uchun ishtirokchilarning ballar jadvalini hisobga olgan holda, siz ikkinchi o'rinni egallagan ballni topishingiz kerak. Sizga ballar beriladi. Ularni ro'yxatda saqlang va ikkinchi o'rinni egallagan ballni toping.

Kirish formati:

Birinchi qatorda n. Ikkinchi qatorda har biri bo'sh joy bilan ajratilgan n ta butun sonlar massivi A[] mavjud.

Shartlar:

$2 \leq n \leq 10$

$-100 \leq A[i] \leq 100$

Chiqarish formati:

Ikkinchi o'rinning reytingini chop eting.

Dastur kodi	Dastur natijasi

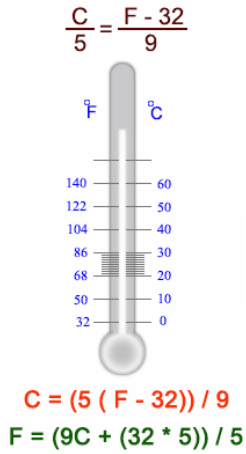
if __name__=='__main__':
n=int(input())
if n>=2 and n<=10:
arr=map(int,input().split())
new_set=sorted(set(arr))
print(new_set[-2])
else:
print("[2,10] oralig`idagi son kiriting")
	Kiritish:
	4
	5 6 10 8
	Chiqarish
	8

3.4-masala

Haroratni Selsiy, Farengeytga aylantiring. Farengeyt va santigrad bugungi kunda qo'llaniladigan ikkita harorat o'lchovidir. Farengeyt shkalasi nemis fizigi Daniel Gabriel Farengeyt tomonidan ishlab chiqilgan. Farengeyt shkalasida suv 32 daraja muzlaydi va 212 daraja qaynaydi.

C = (5/9) * (F - 32)

bu yerda F - Farengeyt harorati.



Dastur kodi	Dastur natijasi

temp = input("Harorat(45F, 102C va hk) : ")
degree = int(temp[:-1])
i_convention = temp[-1]
 	
if i_convention.upper() == "C":
  result = int(round((9 * degree) / 5 + 32))
  o_convention = "Farengeyt"
elif i_convention.upper() == "F":
  result = int(round((degree - 32) * 5 / 9))
  o_convention = "Selsiy"
else:
  print("Tegishli konventsiyani kiriting.")
  quit()
print(o_convention, " harorat", result, "darajaga teng.")
	Harorat(45F, 102C va hk) : 69C
	Farengeyt harorat 156 darajaga teng.
	
	Harorat(45F, 102C va hk) : 10F
	Selsiy harorat -12 darajaga teng.

3.5-masala

It yoshini it yillarida hisoblash uchun Python dasturini yozing.

Eslatma: Dastlabki ikki yil davomida it yili insonning 10,5 yiliga teng. Shundan so'ng har bir it yili insonning 4 yiliga to'g'ri keladi.

Dastur kodi	Dastur natijasi

h_age = int(input("It yoshini odam yilida kiriting: "))
 	
if h_age < 0:
print("Yosh musbat bo`lishi kerak.")
exit()
elif h_age <= 2:
d_age = h_age * 10.5
else:
d_age = 21 + (h_age - 2)*4
 	
print("It yoshi it yilida ", d_age, " ga teng.")
	It yoshini odam yilida kiriting: 5
	It yoshi it yilida 33 ga teng.

3.6-masala

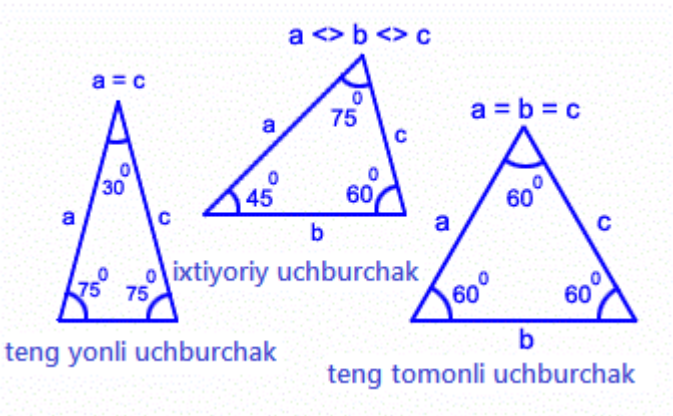
Uchburchakning teng tomonli, teng yonli yoki turli tomonli ekanligini tekshirish uchun Python dasturini yozing.

Eslatma:

Teng tomonli uchburchak - bu uchburchakning uch tomoni teng bo'lgan uchburchak.

Turli tomonli uchburchak - bu uchta teng bo'lmagan tomonlarga ega bo'lgan uchburchak.

Teng yonli uchburchak - bu (kamida) ikkita teng tomoni bo'lgan uchburchak.



Dastur kodi	Dastur natijasi

print("Uchburchak tomonlarini kiriting.")
x = int(input("x: "))
y = int(input("y: "))
z = int(input("z: "))
 	
if x == y == z:
    print("Teng tomonli uchburchak")
elif x==y or y==z or z==x:
    print("Teng yonli uchburchak")
else:
    print("Turli tomonli uchburchak")
	Uchburchak tomonlarini kiriting.
	x: 5
	y: 6
	z: 5
	Teng yonli uchburchak

3.7-masala

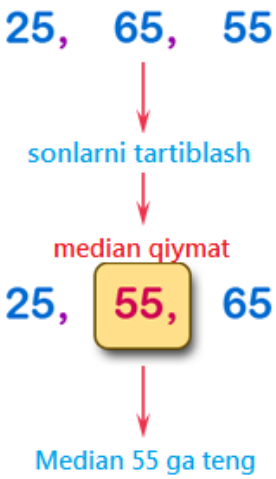
Berilgan yil uchun Xitoy zodiak belgisini ko'rsatish. Tug'ilgan yili uchun Xitoy Zodiak belgisini(muchal) ko'rsatish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
year = int(input("Tug`ilgan yilingizni kiriting: ")) if (year - 2000) % 12 == 0:   sign = 'Baliq' elif (year - 2000) % 12 == 1:   sign = 'Ilon' elif (year - 2000) % 12 == 2:   sign = 'Ot' elif (year - 2000) % 12 == 3:   sign = 'Qo`y' elif (year - 2000) % 12 == 4: sign = 'Maymun' elif (year - 2000) % 12 == 5:	Tug`ilgan yilingizni kiriting: 1993 Sizning muchalingiz : Tovuq

sign = 'Tovuq'
elif (year - 2000) % 12 == 6:
sign = 'It'
elif (year - 2000) % 12 == 7:
sign = 'To`ng`iz'
elif (year - 2000) % 12 == 8:
sign = 'Sichqon'
elif (year - 2000) % 12 == 9:
sign = 'Sigir'
elif (year - 2000) % 12 == 10:
sign = 'Yo`lbars'
else:
sign = 'Quyon'
 	
print("Sizning muchalingiz :",sign)	

3.8-masala

Uchta qiymatning medianini topish uchun Python dasturini yozing.



Dastur kodi	Dastur natijasi
a = float(input("1-sonni kiriting: ")) b = float(input("2-sonni kiriting: "))	1-sonni kiriting: 52 2-sonni kiriting: 17

c = float(input("3-sonni kiriting: "))
if a > b:
  if a < c:
  median = a
  elif b > c:
  median = b
  else:
  median = c
else:
  if a > c:
  median = a
  elif b < c:
  median = b
  else:
  median = c
 	
print("Bu sonlar mediani: ", median)
	3-sonni kiriting: 69
	Bu sonlar mediani: 52.0

3.9-masala

Pythonda kvadrat tenglamani yechimlarini topish uchun dastur kodini yozing.

Kiritish ma`lumotlari:

a,b,c-koeffitsientlar

Chiqarish ma`lumotlari:

x1,x2-kvadrat tenglamaning ildizlari

Dastur kodi	Dastur natijasi

import math
def kv_tenglama(a,b,c):
  d=b**2-4*a*c
  if d>0:
	Kiritish ma`lumotlari
	a ga qiymat kirit: 1
	b ga qiymat kirit: 3
	c ga qiymat kirit: 2

x1=(-b+math.sqrt(d))/(2*a)
x2=(-b-math.sqrt(d))/(2*a)
print(x1,x2)
elif d==0:
x1=(-b+math.sqrt(d))/(2*a)
x2=(-b-math.sqrt(d))/(2*a)
print(x1,x2)
else:
print("Haqiqiy ildizga ega emas!")
x=int(input('a ga qiymat kirit: '))
y=int(input('b ga qiymat kirit: '))
z=int(input('c ga qiymat kirit: '))
kv_tenglama(x,y,z)
	Chiqarish ma`lumotlari
	-1.0 -2.0

3.10-masala

Toq sonlarni oraliq diapazonda sanash uchun Python dasturini yozing. Past va yuqori ikkita manfiy bo'lmagan butun sonlar berilgan. Past va yuqori (shu jumladan) o'rtasidagi toq sonlar sonini qaytaring.

Dastur kodi	Dastur natijasi

def toq_son(min,max):
if min%2!=0:
if max%2!=0:
return (max-min)//2+1
else:
return (max-min)//2+1
else:
if max%2!=0:
return (max-min)//2+1
else:
	5
	627113176

return (max-min)//2+1 print(toq_son(1,10)) print(toq_son(10458161,1264684512))	

11-§. SATRLI MASALALAR

3.11-masala

Pythonda original satrdagi berilgan simvollar sonini topish dasturini tuzing.

1-usul: loop funksiyasini qo`llash.

Dastur kodi	Dastur natijasi

s = "Kamtarga kamol, manmanga zavol."
print("Original satr:")
print(s)
print("Satrdagi 'o' simvollari soni:")
ctr = 0
for c in s:
  if c == 'o':
  ctr = ctr + 1
print(ctr)
	Original satr:
	Kamtarga kamol,
	manmanga zavol.
	Satrdagi 'o'
	simvollari soni: 2

2-usul: Lambda funksiyasini qo`llash.

Dastur kodi	Dastur natijasi
s = "Kamtarga kamol, manmanga zavol." print("Original satr:") print(s) print("Satrdagi 'o' simvollari soni:") ctr = sum(map(lambda x : 1 if 'o' in x else 0, s)) print(ctr)	Original satr: Kamtarga kamol, manmanga zavol. Satrdagi 'o' simvollari soni: 2

3.12-masala

Berilgan satrdan dastlabki 2 va oxirgi 2 ta belgidan iborat satrni olish uchun Python dasturini yozing. Agar satr uzunligi 2 dan kichik bo'lsa, bo'sh satr qaytarsin.

Dastur kodi	Dastur natijasi

def satr_boshi_oxiri(str):
  if len(str) < 2:
  return ''
 	
  return str[0:2] + str[-2:]
 	
print(satr_boshi_oxiri('Python'))
print(satr_boshi_oxiri('Uz'))
print(satr_boshi_oxiri('w'))
	Pyon
	UzUz

3.13-masala

Uzunligi 3 dan katta berilgan satr oxiriga "ing" qo'shish uchun Python dasturini yozing. Agar berilgan satr "ing" bilan tugasa, uning o'rniga "ly" qo'shing. Agar satr uzunligi 3 dan kichik bo'lsa, uni o'zgarishsiz qoldiring.

Dastur kodi	Dastur natijasi

def satr_uzgar(str1):
  uzunlik = len(str1)
 	
  if uzunlik > 2:
  if str1[-3:] == 'ing':
  str1 += 'ly'
  else:
  str1 += 'ing'
	ab
	abcing
	stringly

return str1 print(satr_uzgar('ab')) print(satr_uzgar('abc')) print(satr_uzgar('string'))	

3.14-masala

Berilgan ikkita satrdan barcha umumiy belgilarni leksikografik tartibda topish uchun Python dasturini yozing. Agar umumiy harflar bo'lmasa, "Umumiy belgilar yo'q" ni chop eting.

Dastur kodi	Dastur natijasi
from collections import Counter def uxshash_belgi(str1,str2): d1 = Counter(str1) d2 = Counter(str2) uxshash_lugat = d1 & d2 if len(uxshash_lugat) == 0: return "Umumiy belgilar yo`q."  # umumiy belgilar ro`yxati uxshash_belgi = list(uxshash_lugat.elements()) uxshash_belgi = sorted(uxshash_belgi) return ''.join(uxshash_belgi) str1 = 'Python' str2 = 'PHP' print("Ikki satr: "+str1+', '+str2) print(uxshash_belgi(str1, str2)) str1 = 'Java'	Ikki satr: Python, PHP P Ikki satr: Java, PHP Umumiy belgilar yo`q.

str2 = 'PHP' print("Ikki satr: "+str1+', '+str2) print(uxshash_belgi(str1, str2))	

3.15-masala

Satrdan keltirilgan satrostining 1-indeksini chiqarish uchun Python dasturini yozing. Agar ushbu satrosti satrda mavjud bo`lmasa -1 ni qaytarsin.

Dastur kodi	Dastur natijasi

def strStr(haystack,needle):
  if not haystack:
  return 0
  elif needle in haystack:
  return haystack.index(needle)
  else:
  return -1
print(strStr("Uzbekistan","i"))
print(strStr("Uzbekistan","bek"))
print(strStr("Uzbekistan","os"))
	5
	2
	-1

3.16-masala

Oyda necha kun borligini aniqlash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
print("Oylar ro`yxati: Yanvar, Fevral, Mart, Aprel, May, Iyun, Iyul, Avgust, Sentabr, Oktabr, Noyabr, Dekabr") month_name = input("Oy nomini kiriting: ") if month_name == "Fevral":	Oylar ro`yxati: Yanvar, Fevral, Mart, Aprel, May, Iyun, Iyul, Avgust, Sentabr, Oktabr, Noyabr, Dekabr

print("Oyda 28/29 kun bor")
elif month_name in ("Aprel", "Iyun", "Sentabr", "Noyabr"):
print("Oyda 30 kun bor")
elif month_name in ("Yanvar", "Mart", "May", "Iyul", "Avgust", "Oktabr", "Dekabr"):
print("Oyda 31 kun bor")
else:
print("Oy nomi xato")
	Oy nomini kiriting:
	Avgust
	Oyda 31 kun bor

3.17-masala

Belgilangan tug'ilgan sana uchun astrologik belgini (burjni) ko'rsatish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
day = int(input("Tug`ilgan kuningizni kiriting: ")) month = input("Tug`ilgan oyingizni kiriting (mart, iyul va hk.): ") if month == 'dekabr': astro_sign = 'o`qotar' if (day < 22) else 'tog` echkisi' elif month == 'yanvar': astro_sign = 'tog` echkisi' if (day < 20) else 'qovg`a' elif month == 'fevral': astro_sign = 'qovg`a' if (day < 19) else 'baliq' elif month == 'mart': astro_sign = 'baliq' if (day < 21) else 'qo`y' elif month == 'aprel': astro_sign = 'qo`y' if (day < 20) else 'buzoq' elif month == 'may': astro_sign = 'buzoq' if (day < 21) else 'egizaklar' elif month == 'iyun':	Tug`ilgan kuningizni kiriting: 19 Tug`ilgan oyingizni kiriting (mart, iyul va hk.): noyabr Burjni aniqlash : chayon

astro_sign = 'egizaklar' if (day < 21) else 'qisqichbaqa'
elif month == 'iyul':
astro_sign = 'qisqichbaqa' if (day < 23) else 'arslon'
elif month == 'avgust':
astro_sign = 'arslon' if (day < 23) else 'parizod'
elif month == 'sentabr':
astro_sign = 'parizod' if (day < 23) else 'tarozi'
elif month == 'oktabr':
astro_sign = 'tarozi' if (day < 23) else 'chayon'
elif month == 'noyabr':
astro_sign = 'chayon' if (day < 22) else 'o`qotar'
print("Burjni aniqlash :",astro_sign)	

3.18-masala

Ro'yxatga olish tizimi

Web-sayt uchun ro'yxatdan o'tish shaklini tayyorlayapsiz. Formada 3 ta belgidan ortiq bo'lishi kerak bo'lgan nom maydoni mavjud. 4 ta belgidan kam bo'lgan har qanday nom noto'g'ri. Ismni kiritish sifatida qabul qilish uchun kodni to'ldiring va agar ism noto'g'ri bo'lsa, "invalid name" chiqadi. Agar nom haqiqiy bo'lsa, "account created" ni chiqaring.

Kiritish namuna: abc

Chiqarish namuna: invalid name

Dastur kodi	Dastur natijasi

try:
name=input()
if len(name)<=3:
raise ValueError("Invalid Name")
else:
print("Account Created")
	abc
	Invalid Name
	
	Nigina
	Account Created

except: print("Invalid Name")	

3.19-masala

Telefon raqamini tekshirish

Sizga raqam kiritiladi va u haqiqiy telefon raqami ekanligini tekshirishingiz kerak. Haqiqiy telefon raqami 8 ta raqamdan iborat bo'lib, 1, 8 yoki 9 bilan boshlanadi. Raqam haqiqiy bo'lsa "yaroqli" va "yaroqsiz" bo'lsa, chiqadi.

Dastur kodi	Dastur natijasi

import re
 	
inStr=input()
pattern ="\A([1|8|9])([0-9]{7})\Z"
match=re.match(pattern,inStr)
if match:
  print("Yaroqli")
else:
  print("Yaroqsiz")
	14587622
	Yaroqli
	
	9556653761
	Yaroqsiz

12-§. RO`YXAT, LUG`AT VA TO`PLAMGA DOIR MASALALAR

3.20-masala

Lug'atda berilgan kalit mavjudligini tekshirish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
d = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60} def kalit_mavjud(x): if x in d: print('Berilgan kalit lug`atda mavjud')   else:   print('Berilgan kalit lug`atda mavjud emas') kalit_mavjud(5) kalit_mavjud(9)	Berilgan kalit lug`atda mavjud Berilgan kalit lug`atda mavjud emas

3.21-masala

Lug'atdan kalitni olib tashlash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
myDict = {'a':1,'b':2,'c':3,'d':4} print(myDict) if 'a' in myDict: del myDict['a'] print(myDict)	{'a': 1, 'c': 3, 'b': 2, 'd': 4} {'c': 3, 'b': 2, 'd': 4}

3.22-masala

Turli xil butun sonlar to'plamidan barcha mumkin bo'lgan qism to'plamlarni olish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

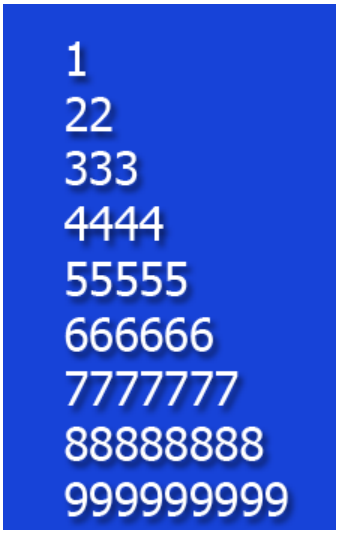
class py_solution:
  def sub_sets(self, sset):
  return self.subsetsRecur([], sorted(sset))
 	
  def subsetsRecur(self, current, sset):
  if sset:
  return self.subsetsRecur(current, sset[1:]) + self.subsetsRecur(current + [sset[0]], sset[1:])
  return [current]
 	
print(py_solution().sub_sets([4,5,6]))
	[[], [6], [5], [5, 6], [4], [4, 6], [4, 5], [4, 5, 6]]

4-BOB. TAKRORLANUVCHI ALGORITMLARGA DOIR DASTURLAR

13-§. ARIFMETIK MASALALAR

4.1-masala

Quyidagi naqshni qurish uchun Python dasturini ichkariga kiritilgan sikl raqamidan foydalanib yozing.



Dastur kodi	Dastur natijasi
for i in range(10): print(str(i) * i)	1 22 333 4444 55555 666666 7777777 88888888 999999999

4.2-masala

Ikki xonali toq sonlarning yig`indisini topish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

s=0
i=11
while i<=99:
  s=s+i
  i=i+2
print("2 xonali toq sonlar yig`indisi=",s)
	2 xonali toq sonlar yig`indisi=
	2475

4.3-masala

Sonni kiritish uchun Python dasturini yozing, agar u son bo'lmasa, xato xabari paydo bo'lsin.

Dastur kodi	Dastur natijasi

while True:
  try:
  a = int(input("Son kiriting: "))
  print("\nBu son.")
  break
  except ValueError:
  print("\nBu son emas. Yana urinib ko`ring...")
  print()
	Son kiriting: aad4
	
	Bu son emas. Yana urinib ko`ring...
	
	Son kiriting: 43132
	
	Bu son.

4.4-masala

Berilgan musbat sonning faktorialining oxiridagi nollar sonini topish uchun Python dasturini yozing. Son diapazoni(n): ($1 <= n <= 2*10^9$).

Dastur kodi	Dastur natijasi

def faktorial_nollari(n):
  x = n // 5
  y = x
  while x > 0:
  x /= 5
  y += int(x)
  return y
 	
print(faktorial_nollari(5))
print(faktorial_nollari(12))
print(faktorial_nollari(100))
	1
	2
	24

4.5-masala

Berilgan sonning eng katta va eng kichik raqamlarini topish uchun Python dasturini yozing.

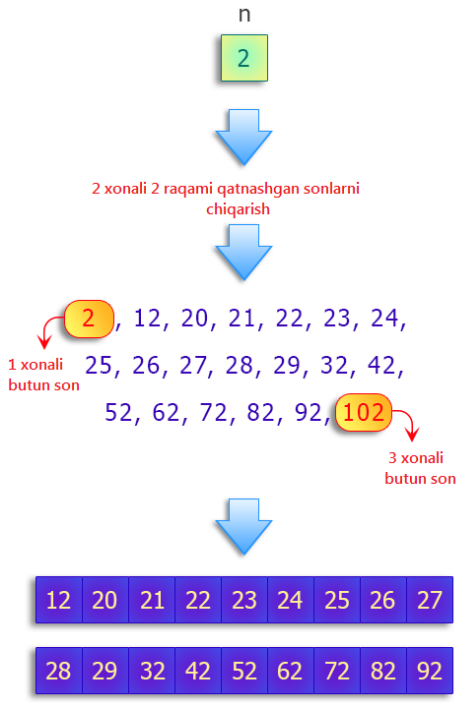
Dastur kodi	Dastur natijasi

def katta_kichik_raqam(n):
  katta_raqam = 0
  kichik_raqam = 9
  while (n):
  raqam = n % 10
  # eng katta raqam
  katta_raqam = max(raqam, katta_raqam)
  # eng kichik raqam
  kichik_raqam = min(raqam, kichik_raqam)
  n = n // 10
  return katta_raqam, kichik_raqam
n = 9387422
	Original Son:
	9387422
	Eng katta raqam: 9
	Eng kichik raqam: 2
	
	Original Son: 500
	Eng katta raqam: 5
	Eng kichik raqam: 0
	
	Original Son: 231548
	Eng katta raqam: 8
	Eng kichik raqam: 1

```
print("Original Son:", n)
natija = katta_kichik_raqam(n)
print("Eng katta raqam:", natija[0])
print("Eng kichik raqam:", natija[1])
n = 500
print("\nOriginal Son:", n)
natija = katta_kichik_raqam(n)
print("Eng katta raqam:", natija[0])
print("Eng kichik raqam:", natija[1])
n = 231548
print("\nOriginal Son:", n)
natija = katta_kichik_raqam(n)
print("Eng katta raqam:", natija[0])
print("Eng kichik raqam:", natija[1])
```

4.6-masala

n raqamli butun sonda 2 raqami ishtirok etgan sonlarni topish uchun Python dasturini yozing.



Dastur kodi	Dastur natijasi

def test(n):
  ans = []
  for i in range(10 ** (n - 1), 10 ** n):
  assert len(str(i)) == n
  if str(i).startswith("2") or str(i).endswith("2"):
  ans.append(i)
  return ans
n = 1
print("Xonalar soni:",n)
print("2 raqami qatnashadigan sonlar:")
print(test(n))
n = 2
print("\nXonalar soni:",n)
print("2 raqami qatnashadigan sonlar:")
print(test(n))
	Xonalar soni: 1
	2 raqami qatnashadigan sonlar:
	[2]
	
	Xonalar soni: 2
	2 raqami qatnashadigan sonlar:
	[12, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 32, 42, 52, 62, 72, 82, 92]

4.7-masala

for dan foydalanib, belgilangan naqshni yarating. for siklidan foydalanib, quyidagi naqshni yaratish uchun Python dasturini yozing.

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

Dastur kodi	Dastur natijasi
n=5; for i in range(n): for j in range(i): print ('* ', end="") print(") for i in range(n,0,-1): for j in range(i): print('* ', end="") print(")	* *

4.8-masala

Fibonachchi sonlari

0 dan 50 gacha Fibonachchi sonlarini olish uchun Python dasturini yozing. Fibonachchi ketma-ketligi sonlar qatoridir:

0, 1, 1, 2, 3, 5, 8, 13, 21, ...

Har bir keyingi raqam oldidagi ikkita sonni qo'shish orqali topiladi.

Kutilayotgan natija: 1 1 2 3 5 8 13 21 34

Dastur kodi	Dastur natijasi

x,y=0,1
 	
while y<50:
  print(y)
  x,y = y,x+y
	1
	1
	2
	3
	5
	8
	13
	21
	34

4.9-masala

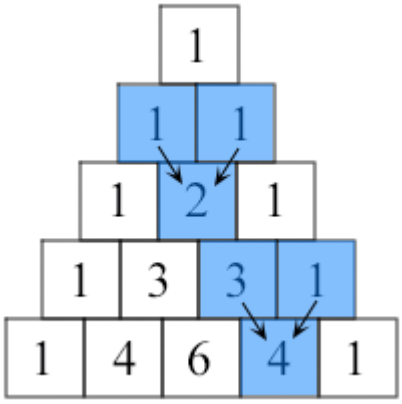
1 bit sonini aniqlash uchun Python dasturini yozing. Belgilanmagan butun sonni oladigan va undagi "1" bitlar sonini qaytaruvchi funksiyani yozing (hamming vazni deb ham ataladi).

Dastur kodi	Dastur natijasi

def hamming_vazni(n):
  c=0
  while n:
  n&=n-1
  c+=1
  return c
 	
print(hamming_vazni(1111011))
print(hamming_vazni(11000000000000000111))
print(hamming_vazni(11))
	12
	33
	3

4.10-masala

Paskal uchburchagining birinchi n qatorini chop etuvchi Python funksiyasini yozing. Paskal uchburchagi birinchi marta Blez Paskal tomonidan tasavvur qilingan arifmetik va geometrik figuradir. Har bir raqam yuqoridagi ikkita raqam birlashtirilgan. Paskal uchburchagi namunasi:



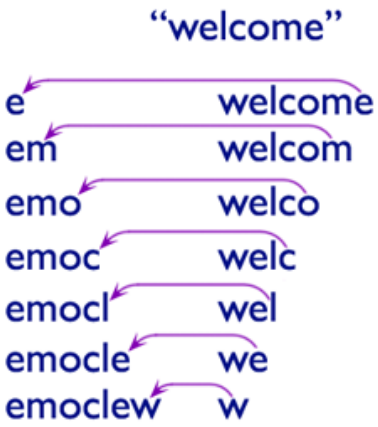
Dastur kodi	Dastur natijasi

def pascal_triangle(n):
  trow = [1]
  y = [0]
  for x in range(max(n,0)):
  print(trow)
  trow=[l+r for l,r in zip(trow+y, y+trow)]
  return n>=1
pascal_triangle(8)
	[1]
	[1, 1]
	[1, 2, 1]
	[1, 3, 3, 1]
	[1, 4, 6, 4, 1]
	[1, 5, 10, 10, 5, 1]
	[1, 6, 15, 20, 15, 6, 1]
	[1, 7, 21, 35, 35, 21, 7, 1]

14-§. SATRLI MASALALAR

4.11-masala

So'zni teskari aylantiring. Foydalanuvchidan so'zni qabul qiladigan Python dasturini yozing va uni teskari aylantiring.



Dastur kodi	Dastur natijasi
word = input("Satr kiriting: ") for char in range(len(word) - 1, -1, -1): print(word[char], end="") print("\n")	Satr kiriting: Amaliy matematika akitametam yilamA

4.12-masala

Ikki satrni qabul qiluvchi Python dasturini yozing va ikkinchi satrdagi harflar birinchi satrda mavjudligini tekshiring.

Dastur kodi	Dastur natijasi
def satrni_tekshir(str1, str2): return all([char in str1.lower() for char in str2.lower()]) print(satrni_tekshir("python", "ypth")) print(satrni_tekshir("python", "ypths"))	True False True False

print(satrni_tekshir("python", "ypthon")) print(satrni_tekshir("123456", "01234")) print(satrni_tekshir("123456", "1234"))	True

4.13-masala

Berilgan satrdan takroriy ketma-ket harflarsiz yangi satr yaratish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
def suzni_tugri_yoz(txt): return txt[0] + ''.join(txt[i] for i in range(1,len(txt)) if txt[i] != txt[i-1]) print(suzni_tugri_yoz("PPYYYTTHON")) print(suzni_tugri_yoz("PPyyythonnn")) print(suzni_tugri_yoz("Java")) print(suzni_tugri_yoz("PPPHHHPPP"))	PYTH ON Python Java PHP

4.14-masala

Tasodifiy alifbo tartibida belgi, alifbo qatori va belgilangan uzunlikdagi alfavit qatorini yaratish uchun Python dasturini yozing. random.choice dan foydalaning.

Dastur kodi	Dastur natijasi
import random import string print("Tasodifiy harf:") print(random.choice(string.ascii_letters)) print("\nTasodifiy alifbo qatori:") max_length = 255 str1 = "" for i in range(random.randint(1, max_length)):	Tasodifiy harf: j Tasodifiy alifbo qatori: PbFbiGMmZzlolpzwOF GTycBgsLhdfUduzhWV pzPNpRcSJhlqvGcjCmQ vsAcHoHlOwGPocLCO

str1 += random.choice(string.ascii_letters) print(str1) print("\nBerilgan uzunlikdagi alifbo qatori:") str1 = "" for i in range(10): str1 += random.choice(string.ascii_letters) print(str1)	gdbnADaCfQanQNLnWl FhXShOKaVstiv Berilgan uzunlikdagi alifbo qatori: LrXAdOIvGa

4.15-masala

Satr uzunligini hisoblash uchun Python dasturini yozing

Dastur kodi	Dastur natijasi

def satr_uzunlik(str1):
k = 0
for char in str1:
k += 1
return k
print(satr_uzunlik('Python'))
	6

4.16-masala

So'zlar ro'yxatini oladigan va eng uzun so'zni va eng uzunining uzunligini qaytarish uchun Python funksiyasini yozing.

Dastur kodi	Dastur natijasi
def uzun_suz_top(ruyxat): uzunlik = [] for n in ruyxat: uzunlik.append((len(n), n)) uzunlik.sort() return uzunlik[-1][0], uzunlik[-1][1]	Eng uzun so`z: Dasturlash Uzun so`zning uzunligi: 10

natija = uzun_suz_top(["PHP", "Amaliyot", "Dasturlash"]) print("Eng uzun so`z: ",natija[1]) print("Uzun so`zning uzunligi: ",natija[0])	

4.17-masala

Berilgan gapdagi har bir so'zning necha marta kelishini hisoblash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def word_count(str):
  counts = dict()
  words = str.split()
 	
  for word in words:
  if word in counts:
  counts[word] += 1
  else:
  counts[word] = 1
 	
  return counts
 	
print(word_count('Python dasturlash tili masalalar kitobi'))
	{'Python': 1,
	'tili': 1,
	'masalalar': 1,
	'kitobi': 1,
	'dasturlash': 1}

4.18-masala

Kitob sarlavhalari

Sizdan kitoblarni turkumlashtirish bo'yicha maxsus dastur yaratish so'ralgan, bu dasturda har bir kitobga uning nomiga qarab maxsus kod beriladi. Kod kitobning birinchi harfiga, keyin esa sarlavhadagi belgilar soniga teng. Masalan, "Harry Potter" kitobi uchun kod H12 bo'ladi, chunki u 12 belgidan iborat (bo'sh joy bilan birga). Sizga har biri alohida satrda yozilgan kitob nomlarini o'z ichiga olgan books.txt fayli taqdim etiladi. Sarlavhani birma-bir o'qing va har bir kitob uchun kodni alohida qatorga chiqaring.

Misol uchun, agar books.txt faylida:

Some book

Another book

Sizning dasturingiz quyidagi natijani chiqarishi kerak:

S9

A12

Fayl satrlarini o'z ichiga olgan ro'yxatni qaytaradigan readlines() usulini eslang. Shuni ham yodda tutingki, oxirgisidan tashqari barcha satrlar oxirida \n belgisi bor, bu belgilar soniga kiritilmasligi kerak.

Masalan, books.txt faylida “O`tgan Kunlar, Mehrobdan Chayon” berilgan. Dastur sizga “O13,M16” ni chiqarishi kerak.

Dastur kodi	Dastur natijasi

file = open("books.txt", "r")
 	
lines = file.readlines()
for i in lines:
  name = i.replace("\n", "")
  print(name[0] + str(len(name)))
 	
file.close()
	O13
	M16

15-§. RO`YXAT VA LUG`ATGA DOIR MASALALAR

4.19-masala

Pythonda ro`yxatdagi butun sonlarga mos keluvchi gistogramma yarating.

Gistogramma tipi: *
Gistogramma soni: [2,5]

**
*****

Dastur kodi	Dastur natijasi

def gistogramma(qiymat):
  for n in qiymat:
  belgi = ''
  muddat = n
  while(muddat > 0):
  belgi += '*'
  muddat = muddat - 1
  print(belgi)
 	
gistogramma([2,5,7,11])
	**
	*****
	*******
	**********

4.20-masala

1 dan 100 gacha bo'lgan 10 ta juft sonli ro'yxatni tasodifiy yaratish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
import random print(random.sample([i for i in range(1,100) if i%2==0], 10))	[40, 8, 54, 30, 24, 50, 42, 28, 16, 32]

4.21-masala

Berilgan raqamlar ro‘yxatidan uchta eng kichik musbat sonlar yig‘indisini hisoblash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
def uchta_son_yigindi(sonlar): return sum(sorted([x for x in sonlar if x >0])[:3]) son = [10, 20, 30, 40, 7] print("Original sonlar ro`yxati: ",son) print("3 kichik musbat sonlar yig`indisi: ",uchta_son_yigindi(son))	Original sonlar ro`yxati: [10, 20, 30, 40, 7] 3 kichik musbat sonlar yig`indisi: 37

4.22-masala

Qiymatlari 1 dan 20 gacha (ikkalasi ham kiritilgan) sonlar kvadratidan iborat bo'lgan birinchi va oxirgi 5 element ro'yxatini yaratish va chop etish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def printValues():
l = list()
for i in range(1,21):
l.append(i**2)
print(l[:5])
print(l[-5:])
printValues()
	[1, 4, 9, 16, 25]
	[256, 289, 324, 361, 400]

4.23-masala

Ro‘yxatdagi barcha elementlarni ko‘paytirish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def ruyxat_kupaytma(items):
  tot = 1
  for x in items:
  tot *= x
  return tot
print(ruyxat_kupaytma([1,2,-8]))
	-16

4.24-masala

print () funksiyasidan foydalanib, ichki ro'yxatlarni (har bir ro'yxat yangi qatorda) chop etish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
ranglar = [['Qizil'], ['Yashil'], ['Qora']] print('\n'.join([str(lst) for lst in ranglar]))	['Qizil'] ['Yashil'] ['Qora']

4.25-masala

Ikkala ro'yxatda mavjud bo'lgan elementlar ro'yxatini qaytaradigan Python dasturini yozing.

Dastur kodi	Dastur natijasi
def umum_element(x, y): return [item for item in x if item in y] print(umum_element([1, 2, 3, 5], [1, 5, 8]))	[1, 5]

4.26-masala

Berilgan satrlar ro‘yxatining barcha elementlari berilgan satrga teng yoki yo‘qligini tekshirish uchun Python dasturini yozing.

1-usul:

Dastur kodi	Dastur natijasi
rang1 = ["yashil", "sariq", "qora", "oq"] rang2 = ["yashil", "yashil", "yashil", "yashil"] print(all(c == 'oq' for c in rang1)) print(all(c == 'yashil' for c in rang2))	False True

2-usul:

Dastur kodi	Dastur natijasi
def hammasi_teng(son): return len(set(son)) == 1 print(hammasi_teng([1, 2, 3, 4, 5, 6])) print(hammasi_teng([4, 4, 4, 4, 4]))	False True

4.27-masala

(x, x*x) ko'rinishida (1 va n oralig'ida) raqamni o'z ichiga olgan lug'at yaratish va chop etish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
n=int(input("Sonni kiriting: ")) d = dict() for x in range(1,n+1): d[x]=x*x print(d)	Sonni kiriting: 5 {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

4.28-masala

Berilgan ro‘yxatdagi ma’lum belgilar bilan boshlangan so`zni topish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
def test(lst, char): result = [i for i in lst if i.startswith(char)] return result text = ["abcd", "abc", "bcd", "bkie", "dagfa", "acjd"] print("\nOriginal ro`yxat:") print(text) char = "a" print(char," belgisi bilan boshlangan so`z:") print(test(text, char)) char = "d" print(char," belgisi bilan boshlangan so`z:") print(test(text, char)) char = "w" print(char," belgisi bilan boshlangan so`z:") print(test(text, char))	Original ro`yxat: ['abcd', 'abc', 'bcd', 'bkie', 'dagfa', 'acjd'] a  belgisi bilan boshlangan so`z: ['abcd', 'abc', 'acjd'] d belgisi bilan boshlangan so`z: ['dagfa'] w  belgisi bilan boshlangan so`z: []

4.29-masala

Har bir harfni lug'atdagi boshqa kalitdan tanlab, barcha harf birikmalarini yaratish va ko'rsatish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
import itertools d ={'1':['a','b'], '2':['c','d']} for combo in itertools.product(*[d[k] for k in sorted(d.keys())]): print(''.join(combo))	ac ad bc bd

4.30-masala

Python lug'atlar ro'yxatidagi qiymatlarni birlashtirish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
from collections import Counter item_list = [{'item': 'item1', 'amount': 400}, {'item': 'item2', 'amount': 300}, {'item': 'item1', 'amount': 750}] result = Counter() for d in item_list: result[d['item']] += d['amount'] print(result)	Counter({'item1': 1150, 'item2': 300})

4.31-masala

Satrdan lug'at yaratish uchun Python dasturini yozing. Satrdagi harflar sonini kuzating.

Dastur kodi	Dastur natijasi
from collections import defaultdict, Counter str1 = 'Python dasturlash tili' my_dict = {} for letter in str1: my_dict[letter] = my_dict.get(letter, 0) + 1 print(my_dict)	{'s': 2, 'o': 1, 'i': 2, 'r': 1, 't': 3, 'l': 2, 'u': 1, 'n': 1, 'y': 1, 'd': 1, 'h': 2, 'P': 1, 'a': 2, ' ': 2}

4.32-masala

Ro‘yxatni kalitlarning ichki lug‘atiga aylantirish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

num_list = [1, 2, 3, 4]
new_dict = current = {}
for name in num_list:
  current[name] = {}
  current = current[name]
print(new_dict)
	{1: {2: {3: {4: {}}}}}

5-BOB. ARALASH DASTURLAR

16-§. ARIFMETIK MASALALAR

5.1-masala

Ikki xonali 5 ga bo`linadigan va 4 ga bo`linmaydigan sonlarni topish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

s=0
i=10
while i<=99:
  if (i%5==0) and (i%4!=0):
  s = s+i
  print(i,end=';')
  i=i+1
print("\ns =",s)
	10;15;25;30;35;45;50;55;65;70;75;85;90;95;
	s = 745

5.2-masala

Kiritilgan natural sonning raqamlar yig`indisini bir xonali son bo`lguncha raqamlar yig`indisini aniqlash dasturini tuzing. M: 345->12->3

Dastur kodi	Dastur natijasi

son=input('sonni kitiring: ')
def sum(son):
  if(int(son)>0):
  s=0
  for i in son:
  s+=int(i)
  if(s>10):
	sonni kitiring: 46151321
	5

sum(str(s)) else: print(s) else: print("Natural son kiriting!") sum(son)	

5.3-masala

Python dasturida berilgan ro`yxatdagi musbat sonlarni filtrlash dasturini tuzing.

1-usul:

Dastur kodi	Dastur natijasi
ruyxat = [34, 1, 0, -23, 12, -88] print("Ro`yxatdagi barcha sonlar:\n",ruyxat) print("Ro`yxatdagi musbat sonlar: ") for musbat_son in ruyxat: if musbat_son > 0: print(musbat_son, end = " ")	Ro`yxatdagi barcha sonlar: [34, 1, 0, -23, 12, -88] Ro`yxatdagi musbat sonlar: 34 1 12

2-usul:

Dastur kodi	Dastur natijasi
ruyxat = [34, 1, 0, -23, 12, -88] print("Ro`yxatdagi barcha sonlar:\n ",ruyxat) musbat_sonlar = [n for n in ruyxat if n> 0] print("Ro`yxatdagi musbat sonlar:\n",*musbat_sonlar)	Ro`yxatdagi barcha sonlar: [34, 1, 0, -23, 12, -88] Ro`yxatdagi musbat sonlar: 34 1 12

5.4-masala

Ro`yxatda 4 lar sonini aniqlab beradigan dastur tuzing.

Dastur kodi	Dastur natijasi

def ruyxat_4(sonlar):
  qiymat = 0
  for son in sonlar:
  if son == 4:
  qiymat = qiymat + 1
 	
  return qiymat
 	
print(ruyxat_4([1, 4, 6, 7, 4]))
print(ruyxat_4([1, 4, 6, 4, 7, 4]))
	2
	3

5.5-masala

n ta butun elementli massivning uchta elementi yig‘indisi nolni beradigan noyob (modul jihatdan takrorlanmaydigan) uchliklarni topish uchun Python dasturini yozing.

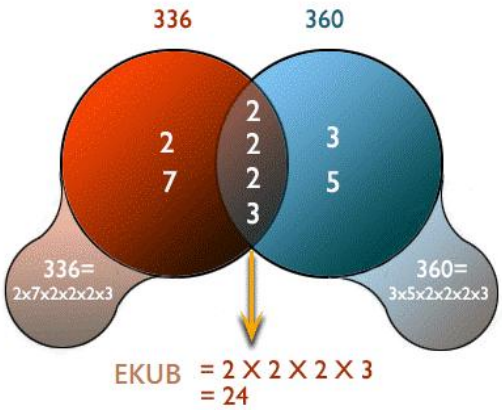
Dastur kodi	Dastur natijasi

def three_sum(nums):
  result = []
  nums.sort()
  for i in range(len(nums)-2):
  if i> 0 and nums[i] == nums[i-1]:
  continue
  l, r = i+1, len(nums)-1
  while l < r:
  s = nums[i] + nums[l] + nums[r]
	[(-6, 2, 4)]

```
    if s > 0:
      r -= 1
    elif s < 0:
       l += 1
    else:
      # three sum ning  yig`indisi
      result.append((nums[i], nums[l], nums[r]))
      # dublikatlarni o`chirish
      while l < r and nums[l] == nums[l+1]:
        l+=1
        while l < r and nums[r] == nums[r-1]:
          r -= 1
          l += 1
          r -= 1
        return result

x = [1, -6, 4, 2, -1, 2, 0, -2, 0 ]
print(three_sum(x))
```

5.6-masala

Berilgan juftlikdagi ikkita son orasidagi umumiy bo‘luvchilari sonini topish uchun Python dasturini yozing.

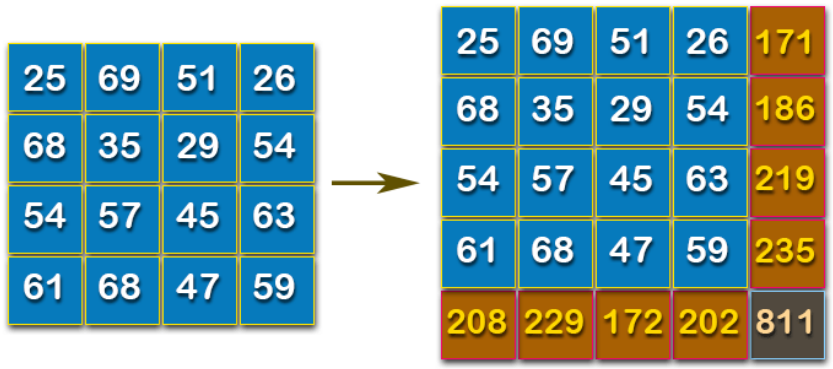


Dastur kodi	Dastur natijasi
def ekub(x, y): i=1 while(i<=x and i<=y): if(x%i==0 and y%i == 0): umumiy=i; i+=1 return umumiy; def um_bul_soni(x, y): n = ekub(x, y) natija = 0 z = int(n**0.5) i = 1 while(i <= z): if(n % i == 0): natija += 2 if(i == n/i): natija-=1 i+=1 return natija print("Umumiy bo`luvchilari soni: ",um_bul_soni(2, 4)) print("Umumiy bo`luvchilari soni: ",um_bul_soni(2, 8)) print("Umumiy bo`luvchilari soni: ",um_bul_soni(12, 24))	Umumiy bo`luvchilari soni: 2 Umumiy bo`luvchilari soni: 2 Umumiy bo`luvchilari soni: 6

5.7-masala

Sizning vazifangiz elektron jadval dasturining kichik qismini ishlab Chiqarishdir. Quyidagi rasmda ko'rsatilganidek, berilgan jadvalning ustunlari va

qatorlarini qo'shib, natijani yangi ustun(qator)ga chiqarish uchun Python dasturini yozing:



Dastur kodi	Dastur natijasi

while True:
  print("Qator/ustunlar sonini kiriting(Chiqarish uchun 0)")
  n = int(input())
  if n == 0:
  break
  print("Qiymatlarni kiriting:")
  x = []
  for i in range(n):
  x.append([int(num) for num in input().split()])
 	
  for i in range(n):
  sum = 0
  for j in range(n):
  sum += x[i][j]
  x[i].append(sum)
 	
  x.append([])
  for i in range(n + 1):
	Qator/ustunlar sonini kiriting(Chiqarish uchun 0)
	3
	Qiymatlarni kiriting:
	56 89 54
	25 12 36
	85 74 65
	Natija:
	56 89 54 199
	25 12 36 73
	85 74 65 224
	166 175 155 496
	Qator/ustunlar sonini kiriting(Chiqarish uchun 0)

```
        sum = 0
        for j in range(n):
            sum += x[j][i]
        x[n].append(sum)
    print("Natija:")
    for i in range(n + 1):
        for j in range(n + 1):
            print('{0:>5}'.format(x[i][j]), end="")
        print()
```

5.8-masala

Berilgan musbat sondan kichik tub sonlar sonini hisoblash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def tub_sonlar_soni(n):
  k = 0 # kichik tub sonlar soni
  for son in range(n):
  if son <= 1:
  continue
  for i in range(2, son):
  if (son % i) == 0:
  break
  else:
  k += 1
  return k
print(tub_sonlar_soni(10))
print(tub_sonlar_soni(100))
	4
	25

5.9-masala

32 bitli x berilgan bo'lsa, uning raqamlari teskari bo'lgan x ni qaytaring. Agar x ni teskari aylantirish qiymati 32 bitli ishorali butun son oralig'idan $[-2^{31}, 2^{31} - 1]$ tashqariga chiqsa, u holda 0 ni qaytaring. Sizga 64-bitli butun sonlarni (ishorali yoki ishorasiz) saqlashga ruxsat bermaydi deb faraz qiling.

Masalan, Example 1:

Input: x = 123

Output: 321

Example 2:

Input: x = -123

Output: -321

Example 3:

Input: x = 120

Output: 21

Cheklovlar: x $[-2^{31}, 2^{31} - 1]$ oralig`idagi butun son.

Dastur kodi	Dastur natijasi

def reverse(x):
  negative=x<0
  x=abs(x)
  reversed=0
  while x!=0:
  reversed=reversed*10+x%10
  x//=10
  if reversed>2**31-1:
  return 0
  return reversed if not negative else -reversed
 	
print(reverse(-958133))
	-331859

5.10-masala

Berilgan butun son palindrom yoki palindrom emasligini aniqlash uchun Python dasturini tuzing. Agar x palindrom butun son bo'lsa, True qaytaring. Butun son palindrom bo'lib, u oldinga o'qigandek orqaga o'qiladi. Masalan, 121 palindrom, 123 esa yo'q. Cheklovlar x [-2^{31}, 2^{31} - 1] oralig`idagi butun son.

Dastur kodi	Dastur natijasi

def isPalindrome(x):
  s=str(x)
  left,right=0, len(s)-1
  while left<right:
  if s[left]!=s[right]:
  return False
  left+=1
  right-=1
  return True
 	
print(isPalindrome(5335))
	True

5.11-masala

Butun sonni rim raqamlari orqali ifodalash(Integer to Roman) uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def intToRoman(num):
  roman=['I','IV','V','IX','X','XL','L','XC','C','CD','D','CM','M'][::-1]
  INT=(1000,900,500,400,100,90,50,40,10,9,5,4,1)
  s=""
  for i in range(len(INT)):
	LIX
	CXXIII
	MMDCC
	XLVIII

```
        count=num//INT[i]
        s+=roman[i]*count
        num-=INT[i]*count
    return s

print(intToRoman(59))
print(intToRoman(123))
print(intToRoman(2748))
```

5.12-masala

Bir qator sonlardan juft va toq sonlar sonini sanash. Bir qator raqamlardan juft va toq sonlar sonini hisoblash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

numbers = (1, 2, 3, 4, 5, 6, 7, 8, 9) # Kortejni e`lon qilish
count_odd = 0
count_even = 0
for x in numbers:
    if not x % 2:
        count_even+=1
    else:
        count_odd+=1
print("Juft sonlar soni :",count_even)
print("Toq sonlar soni :",count_odd)
	Juft sonlar soni :
	4
	Toq sonlar soni :
	5

5.13-masala

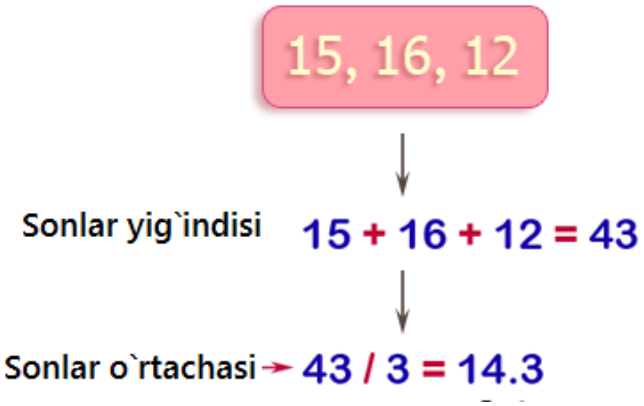
3 va 6 dan tashqari 0 dan 6 gacha barcha raqamlarni chop etadigan Python dasturini yozing. Eslatma: "continue" dan foydalaning.

Dastur kodi	Dastur natijasi

for x in range(6):
  if (x == 3 or x==6):
  continue
  print(x,end=' ')
print("\n")
	0 1 2 4 5

5.14-masala

n ta butun sonning yig‘indisini va o‘rtachasini hisoblang. n ta butun sonning yig‘indisi va o‘rtachasini hisoblash uchun Python dasturini yozing (foydalanuvchi kiritgan). Tugatish uchun 0 kiriting



Dastur kodi	Dastur natijasi

print("Butun sonlarni kiriting. Chiqarish uchun 0 ni kiriting.")
 	
count = 0
sum = 0.0
number = 1
 	
while number != 0:
  number = int(input(""))
  sum = sum + number
  count += 1
	Butun sonlarni kiriting. Chiqarish uchun 0 ni kiriting.
	5
	6
	2
	3

if count == 0:
print("Butun sonlarni kiriting: ")
else:
print("Sonlarning o'rtacha qiymati: ", sum / (count-1))
print("Sonlarning yig'indisi: ", sum)
	2
	1
	3
	2
	0
	Sonlarning o'rtacha qiymati: 3.0
	Sonlarning yig'indisi: 24.0

5.15-masala

Qavslar qatori

'(', ')', '{', '}', '[' va '] qavslar qatorining haqiqiyligini topish uchun Python sinfini yozing. Ushbu qavslar to'g'ri tartibda yaqin bo'lishi kerak, masalan, "()" va "()[]{}" to'g'ri, lekin "[)", "({[)]" va "{{{" noto'g'ri.

Dastur kodi	Dastur natijasi

class py_solution:
def is_valid_parenthese(self, str1):
stack, pchar = [], {"(": ")", "{": "}", "[": "]"}
for parenthese in str1:
if parenthese in pchar:
stack.append(parenthese)
elif len(stack) == 0 or pchar[stack.pop()] != parenthese:
return False
return len(stack) == 0
	True
	False
	True

print(py_solution().is_valid_parenthese("(){}[]")) print(py_solution().is_valid_parenthese("([{}]")) print(py_solution().is_valid_parenthese("()"))	

5.16-masala

Natural sonning tub ko`paytuvchilar ko`paytmasida chiqarish uchun Python dasturini yozing. Masalan, 100=2*2*5*5;

Dastur kodi	Dastur natijasi

n=int(input("Natural son kiriting: "))
i=2
print(n,end='=')
while n>1:
  if n % i==0:
  n=n // i
  if n!=1:
  print(i,end='*')
  else:
  print(i)
  else:
  i=i+1
	Natural son kiriting: 52
	52=2*2*13

5.17-masala

Berilgan n natural songacha bo'lgan tub sonlarni ekranga chiqarish dasturini yozing.

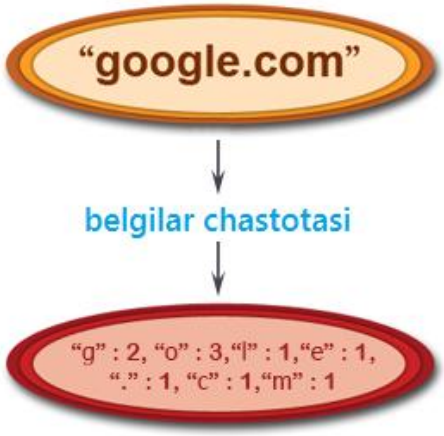
Dastur kodi	Dastur natijasi

n = int(input("Qaysi songacha bo'lgan tub sonlarni chiqarsin? "))
nums = []
for i in range(0, n + 1):
  nums.append(i)
nums[1] = 0
p = 2
while p < n:
  for i in range(p * 2, n + 1, p):
  nums[i] = 0
  p = p + 1
  while p < n and nums[p] == 0:
  p = p + 1
print(n," sonigacha bo'lgan tub sonlar:")
for i in nums:
  if nums[i] != 0:
  print(i, end=' ')
	Qaysi songacha bo'lgan tub sonlarni chiqarsin?
	50
	50 sonigacha bo'lgan tub sonlar:
	2 3 5 7 11 13 17 19 23 29 31 37 41 43 47

17-§. SATRLI MASALALAR

5.18-masala

Satrdagi belgilar sonini (belgi chastotasi) hisoblash uchun Python dasturini yozing.



Dastur kodi	Dastur natijasi

def belgi_chastota(str1):
  dict = {}
  for n in str1:
  kalit = dict.keys()
  if n in kalit:
  dict[n] += 1
  else:
  dict[n] = 1
  return dict
print(belgi_chastota('google.com'))
	{'.': 1, 'g': 2, 'c': 1, 'm': 1, 'e': 1, 'o': 3, 'l': 1}

5.19-masala

Agar birinchi 4 ta belgida kamida 2 ta bosh harf bo‘lsa, berilgan satrni barcha bosh harflarga aylantirish uchun Python funksiyasini yozing.

Dastur kodi	Dastur natijasi

def katta_harf(str1):
  num_upper = 0
  for letter in str1[:4]:
  if letter.upper() == letter:
  num_upper += 1
  if num_upper >= 2:
  return str1.upper()
  return str1
 	
print(katta_harf('Python'))
print(katta_harf('PyThon'))
	Python
	PYTHON

5.20-masala

Satrdagi takroriy belgilarni sanash uchun python dasturini yozing.

Dastur kodi	Dastur natijasi

import collections
str1 = 'thequickbrownfoxjumpsoverthelazydog'
d = collections.defaultdict(int)
for c in str1:
  d[c] += 1
 	
for c in sorted(d, key=d.get, reverse=True):
 if d[c] > 1:
  print('%s %d' % (c, d[c]))
	o 4
	e 3
	r 2
	t 2
	h 2
	u 2

5.21-masala

Berilgan so‘zlar ro‘yxatidan n dan uzun so‘zlar ro‘yxatini topish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def uzun_soz(n, str):
  word_len = []
  txt = str.split(" ")
  for x in txt:
  if len(x) > n:
  word_len.append(x)
  return word_len
print(uzun_soz(6, "Python dasturlash tili"))
	['dasturlash']

5.22-masala

Berilgan satr raqamlari yig'indisini hisoblash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def satr_raqam_sum(str1):
  s = 0
  for x in str1:
  if x.isdigit() == True:
  z = int(x)
  s = s + z
 	
  return s
print(satr_raqam_sum("123abcd45"))
print(satr_raqam_sum("abcd1234"))
	15
	10

5.23-masala

Berilgan satrdan ikkita satr yaratish uchun Python dasturini yozing. 1-qatorda 1 marta uchraydigan va 2-qatorda ko`p uchraydigan belgilardan iborat satr yarating.

Dastur kodi	Dastur natijasi
from collections import Counter def generateStrings(input): str_char_ctr = Counter(input) part1 = [key for (key,count) in str_char_ctr.items() if count==1] part2 = [key for (key,count) in str_char_ctr.items() if count>1] part1.sort() part2.sort() return part1,part2 input = "aabbcceffgh" s1, s2 = generateStrings(input) print(".join(s1)) print(".join(s2))	egh abcf

5.24-masala

Satr uzunligi 2 yoki undan ortiq, birinchi va oxirgi belgilar berilgan satrlar ro‘yxatidan bir xil bo‘lgan satrlar sonini hisoblash uchun Python dasturini yozing.

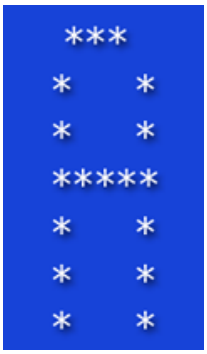
Dastur kodi	Dastur natijasi

def soz_topish(words):
ctr = 0
 	
for word in words:
if len(word) > 1 and word[0] == word[-1]:
	2

ctr += 1 return ctr print(soz_topish(['abc', 'xyz', 'aba', '1221']))	

5.25-masala

"A" harfi naqshini chop etish uchun Python dasturini yozing.

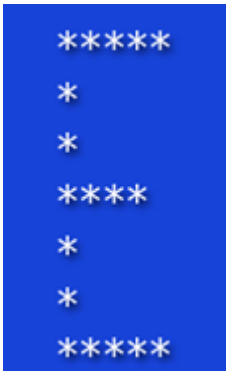


Dastur kodi	Dastur natijasi

result_str="";
for row in range(0,7):
for column in range(0,7):
if (((column == 1 or column == 5) and row != 0) or ((row == 0 or row == 3) and (column > 1 and column < 5))):
result_str=result_str+"*"
else:
result_str=result_str+" "
result_str=result_str+"\n"
print(result_str);
	***
	* *
	* *
	*****
	* *
	* *
	* *

5.26-masala

"E" harfi naqshini chop etish uchun Python dasturini yozing.

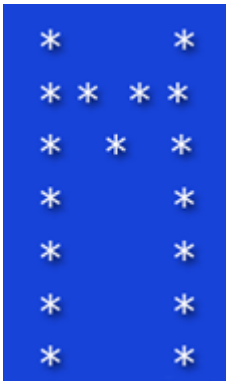


Dastur kodi	Dastur natijasi

result_str="";
for row in range(0,7):
  for column in range(0,7):
  if (column == 1 or ((row == 0 or row == 6) and (column > 1 and column < 6)) or (row == 3 and column > 1 and column < 5)):
  result_str=result_str+"*"
  else:
  result_str=result_str+" "
  result_str=result_str+"\n"
print(result_str);
	*****
	*
	*
	****
	*
	*
	*****

5.27-masala

"M" harfi naqshini chop etish uchun Python dasturini yozing.

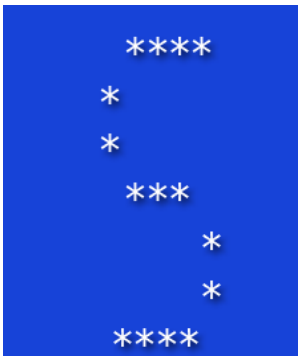


Dastur kodi	Dastur natijasi

result_str="";
for row in range(0,7):
  for column in range(0,7):
  if (column == 1 or column == 5 or (row == 2 and (column == 2 or column == 4)) or (row == 3 and column == 3)):
  result_str=result_str+"* "
  else:
  result_str=result_str+" "
  result_str=result_str+"\n"
print(result_str);
	* *
	* *
	* * * *
	* * *
	* *
	* *
	* *

5.28-masala

"S" harfi naqshini chop etish uchun Python dasturini yozing.



Dastur kodi	Dastur natijasi

result_str="";
for row in range(0,7):
  for column in range(0,7):
  if (((row == 0 or row == 3 or row == 6) and column > 1 and column < 5) or (column == 1 and (row == 1 or row == 2 or row == 6)) or (column == 5
	****
	*
	*
	***
	  *
	  *
	****

and (row == 0 or row == 4 or row == 5))):
result_str=result_str+"*"
else:
result_str=result_str+" "
result_str=result_str+"\n"
print(result_str);	

5.29-masala

Satrdagi raqamlar va harflar sonini hisoblang. Satrni qabul qiluvchi Python dasturini yozing va raqamlar va harflar sonini hisoblang.

Dastur kodi	Dastur natijasi

s = input("Satrni kiriting:")
d=l=0
for c in s:
if c.isdigit():
d=d+1
elif c.isalpha():
l=l+1
else:
pass
print("Harflar ", l)
print("Sonlar ", d)
	Satrni kiriting:2022yil 17may
	Harflar 6
	Sonlar 6

5.30-masala

Parolning haqiqiyligini tekshiring. Parolning haqiqiyligini tekshirish uchun Python dasturini yozing (foydalanuvchilar kiritishi).

Tasdiqlash:

[a-z] orasida kamida 1 harf va [A-Z] orasida 1 harf.

[0-9] orasida kamida 1 ta raqam.

[$#@] dan kamida 1 ta belgi.

Minimal uzunlik 6 belgi.

Maksimal uzunlik 16 belgi.

Dastur kodi	Dastur natijasi

import re
p= input("Parolni kiriting:")
x = True
while x:
  if (len(p)<6 or len(p)>12):
  break
  elif not re.search("[a-z]",p):
  break
  elif not re.search("[0-9]",p):
  break
  elif not re.search("[A-Z]",p):
  break
  elif not re.search("[$#@]",p):
  break
  elif re.search("\s",p):
  break
  else:
  print("Parol haqiqiy")
  x=False
  break
 	
if x:
  print("Parol haqiqiy emas")
	Parolni kiriting:gff0AS@131df
	Parol haqiqiy

18-§. MASSIVGA DOIR MASALALAR

5.31-masala

Har bir elementi * bo'lgan 3*4*6 3D massivni yaratish uchun Python dasturini yozing.

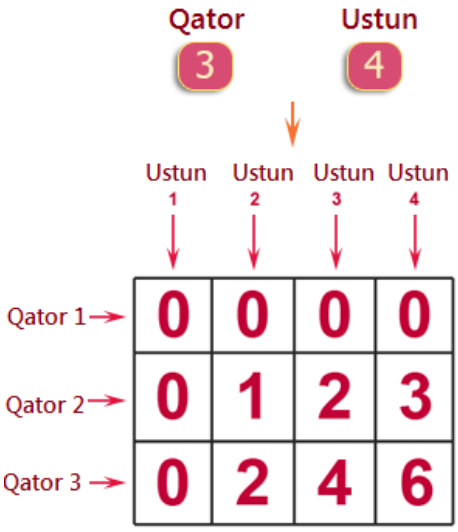
Dastur kodi	Dastur natijasi
array = [[['*' for col in range(6)] for col in range(4)] for row in range(3)] print(array)	[[['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*']], [['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*']], [['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*']]]

5.32-masala

Kirish sifatida ikkita m (satr) va n (ustun) raqamlarini oladigan va ikki o'lchovli massiv hosil qiluvchi Python dasturini yozing. Massivning i-qator va j-ustunidagi element qiymati i*j bo‘lishi kerak.

i = 0,1.., m-1

j = 0,1, n-1.

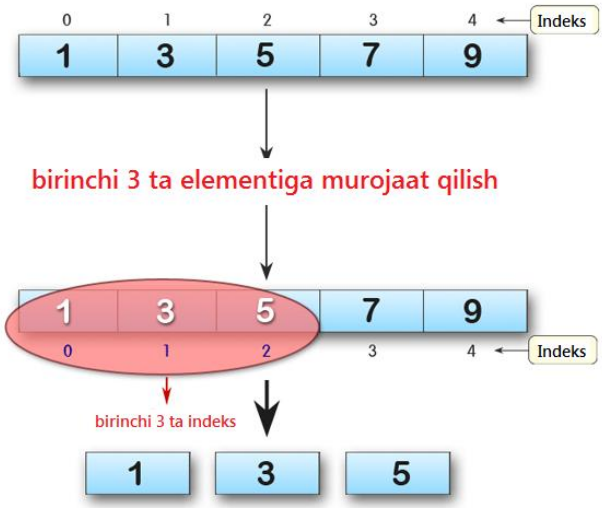


Dastur kodi	Dastur natijasi

row_num = int(input("Qatorlar sonini kiriting: "))
col_num = int(input("Ustunlar sonini kiriting: "))
multi_list = [[0 for col in range(col_num)] for row in range(row_num)]
 	
for row in range(row_num):
  for col in range(col_num):
  multi_list[row][col]= row*col
 	
print(multi_list)
	Qatorlar sonini kiriting: 2
	Ustunlar sonini kiriting: 3
	[[0, 0, 0], [0, 1, 2]]

5.33-masala

5 ta butun sonli massiv yaratish va massiv elementlarini ko‘rsatish uchun Python dasturini yozing. Bunda indekslar orqali alohida elementga kirish kerak.



Dastur kodi	Dastur natijasi
from array import * array_num = array('i', [1,3,5,7,9]) for i in array_num: print(i)	1 3 5 7

print("Birinchi uchta elementga alohida kirish") print(array_num[0]) print(array_num[1]) print(array_num[2])	9 Birinchi uchta elementga alohida kirish 1 3 5

5.34-masala

Berilgan n*m o`lchovli matritsani satr oldinga > orqaga > oldinga >... tartibda chop etish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def print_matrix(nums):
  flag = True
 	
  for line in nums:
 	
  if flag == True:
  i = 0
  while i < len(line):
  print(line[i])
  i += 1
  flag = False
 	
  else:
  i = -1
  while i > -1 * len(line) - 1:
  print(line[i])
  i = i - 1
  flag = True
	1
	2
	3
	4
	8
	7
	6
	5
	0
	6
	2
	8
	2
	0
	3
	2

print_matrix([[1, 2, 3, 4], [5, 6, 7, 8], [0, 6, 2, 8], [2, 3, 0, 2]])	

5.35-masala

Butun sonlar massivi va butun sonli targetni hisobga olsak, ikkita sonning indekslarini shunday qaytaringki, ularning yig`indisi targetga teng bo`lsin.

Siz har bir kirishda aynan bitta yechim bo'ladi deb taxmin qilishingiz mumkin va siz bir xil elementni ikki marta ishlata olmaysiz. Javobni istalgan tartibda qaytarishingiz mumkin.

Masalan,

Input: nums = [2,7,11,15], target = 9

Output: [0,1]

Tushuntirish: Chunki nums[0] + nums[1] == 9, biz [0, 1] ni qaytardik.

Cheklovlar:

2 <= nums.length <= 104

-109 <= nums[i] <= 109

-109 <= target <= 109

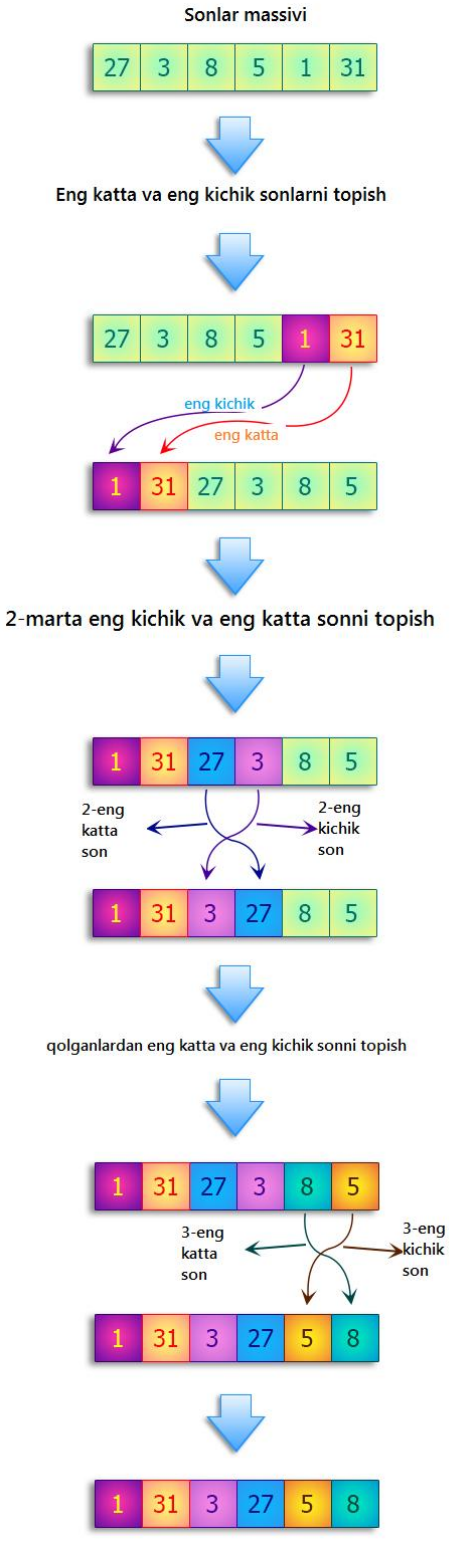
Faqat bitta to'g'ri javob mavjud.

Dastur kodi	Dastur natijasi

def twoSum(nums,target):
for i in range(len(nums)):
for j in range(i+1,len(nums)):
if nums[i]+nums[j]==target:
return i,j
 	
print(twoSum([1,4,9],10))
	(0, 2)

5.36-masala

Quyidagicha sonlar ro'yxatini topish uchun Python dasturini yozing: birinchi element eng kichik, ikkinchisi qolganlarning eng kattasi, uchinchisi qolganlarning eng kattasi, to'rtinchisi qolganlarning eng kichigi va hokazo.



Dastur kodi	Dastur natijasi
def test(sonlar): if len(sonlar) < 2: return sonlar natija = [] for i in range(len(sonlar)//2): natija.append(min(sonlar)) sonlar.remove(min(sonlar)) natija.append(max(sonlar)) sonlar.remove(max(sonlar)) if len(sonlar) > 0: natija.append(sonlar[0]) if len(natija) < 2*len(sonlar): natija.extend(sonlar[len(natija) // 2 + 1:len(natija) // 2 + 1 + len(sonlar) - len(natija)]) return natija son = [1, 3, 4, 5, 11] print("Original ro`yxat:") print(son) print("Yangi ro`yxat:") print(test(son)) son = [27, 3, 8, 5, 1, 31] print("\nOriginal ro`yxat:") print(son) print("Yangi ro`yxat:") print(test(son))	Original ro`yxat: [1, 3, 4, 5, 11] Yangi ro`yxat: [1, 11, 3, 5, 4] Original ro`yxat: [27, 3, 8, 5, 1, 31] Yangi ro`yxat: [1, 31, 3, 27, 5, 8]

5.37-masala

Agar butun son massivi nums va val butun sonini hisobga olsak, valning barcha sonlarini joyida olib tashlang. Elementlarning nisbiy tartibi o'zgarishi mumkin.

Ba'zi tillarda massiv uzunligini o'zgartirishning iloji bo'lmagani uchun siz natijani massivning birinchi qismiga raqamlar bilan joylashtirishingiz kerak. Rasmiy ravishda, agar dublikatlarni olib tashlaganingizdan so'ng, k element mavjud bo'lsa, u holda sonlarning birinchi k elementi yakuniy natijani saqlashi kerak. Birinchi k elementdan keyin nima qoldirganingiz muhim emas.

Yakuniy natijani sonlarning birinchi k uyasiga qo'ygandan keyin k ni qaytaring. Boshqa massiv uchun qo'shimcha joy ajratmang. Buni kiritish massivini O (1) qo'shimcha xotira bilan joyida o'zgartirish orqali qilishingiz kerak.

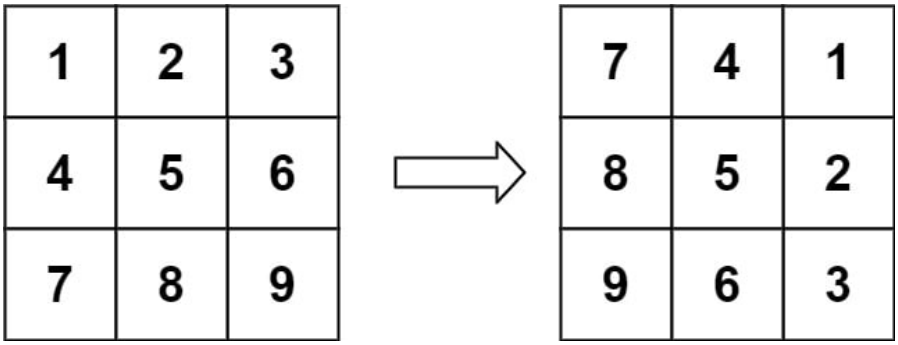
Dastur kodi	Dastur natijasi

def removeElement(nums):
  count=0
  for i in range(len(nums)):
  if nums[i]!=val:
  nums[count]=nums[i]
  count+=1
  return count
numbers=input().split()
nums=[int(i) for i in numbers]
val=int(input())
 	
print(removeElement(nums))
print(nums)
	Kiritish:
	2 2 3 3 4 5
	3
	Chiqarish:
	4
	[2, 2, 4, 5, 4, 5]

5.38-masala

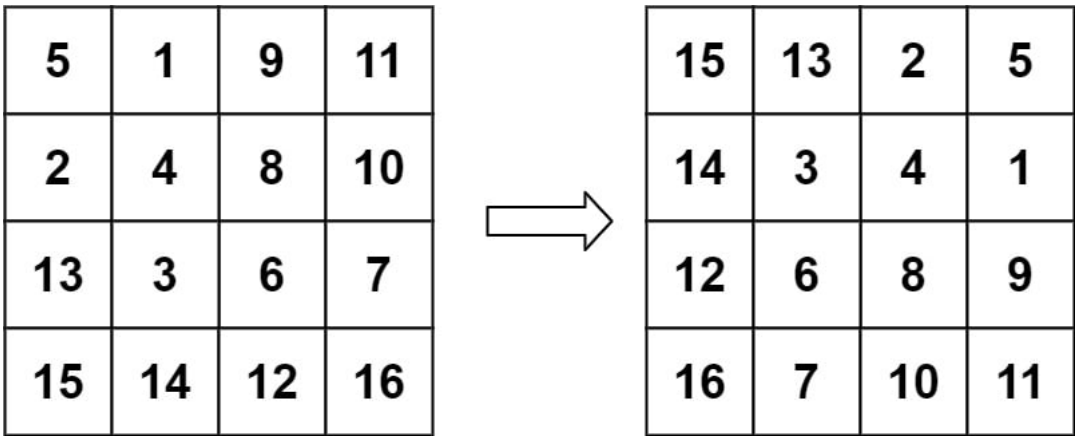
Sizga tasvirni ifodalovchi n x n 2D matritsasi beriladi, tasvirni 90 gradusga (soat yo'nalishi bo'yicha) aylantiring.

Tasvirni joyida aylantirishingiz kerak, ya'ni kirish 2D matritsasini to'g'ridan-to'g'ri o'zgartirishingiz kerak. Boshqa 2D matritsani ajratmang va aylanishni bajaring.



Kiritish: matrix = [[1,2,3],[4,5,6],[7,8,9]]

Chiqarish: [[7,4,1],[8,5,2],[9,6,3]]



Kiritish: matrix = [[5,1,9,11],[2,4,8,10],[13,3,6,7],[15,14,12,16]]

Chiqarish: [[15,13,2,5],[14,3,4,1],[12,6,8,9],[16,7,10,11]]

Dastur kodi	Dastur natijasi

def rotate(matrix):
matrix.reverse()
for i in range(len(matrix)):
for j in range(i):
matrix[i][j],matrix[j][i]=matrix[j][i],matrix[i][j]
return matrix
 	
print(rotate([[1,2,3],[4,5,6],[7,8,9]]))
	[[7, 4, 1],
	[8, 5, 2],
	[9, 6, 3]]

5.39-masala

Mos ravishda m va n o'lchamdagi nums1 va nums2 ikkita tartiblangan massivlar berilgan bo'lsa, ikkita tartiblangan massivning medianini qaytaring. Umumiy ish vaqtining murakkabligi O (log (m+n)) bo'lishi kerak.

Dastur kodi	Dastur natijasi

def findMedianSortedArrays(nums1,nums2):
l=nums1+nums2
l=sorted(l)
if len(l)%2==0:
return (l[(len(l)-1)//2]+l[(len(l)+1)//2])/2
else:
return l[(len(l)-1)//2]
 	
print(findMedianSortedArrays([1,2],[3,4]))
print(findMedianSortedArrays([0,1,2],[3,4,5]))
	2.5
	2.5

5.40-masala

Satrlar massivi orasidan eng uzun umumiy prefiks qatorini topish funksiyasini yozing. Agar umumiy prefiks bo'lmasa, bo'sh qatorini qaytaring.

Dastur kodi	Dastur natijasi

def longestCommonPrefix(strs):
  s=""
  if len(strs)==0:
  return s
  min_length=len(strs[0])
  for i in strs:
  min_length=min(min_length,len(i))
  for i in range(0,min_length):
  current=strs[0][i]
  for j in range(len(strs)):
  if strs[j][i]!=current:
  return s
  s+=current
  return s
 	
print(longestCommonPrefix(["flower","flight","flow"]))
print(longestCommonPrefix(["car","plane","bicycle"]))
	fl

5.41-masala

n haqiqiy sonlar to‘plamidan (massiv) yig‘indisi nolga teng bo‘lgan uchta elementni topish uchun Python sinfini yozing.

Dastur kodi	Dastur natijasi

class py_solution:
def threeSum(self, nums):
nums, result, i = sorted(nums), [], 0
while i < len(nums) - 2:
j, k = i + 1, len(nums) - 1
while j < k:
if nums[i] + nums[j] + nums[k] < 0:
j += 1
elif nums[i] + nums[j] + nums[k] > 0:
k -= 1
else:
result.append([nums[i], nums[j], nums[k]])
j, k = j + 1, k - 1
while j < k and nums[j] == nums[j - 1]:
j += 1
while j < k and nums[k] == nums[k + 1]:
k -= 1
i += 1
while i < len(nums) - 2 and nums[i] == nums[i - 1]:
i += 1
return result
 	
print(py_solution().threeSum([-25, -10, -7, -3, 2, 4, 8, 10]))
	[[-10, 2, 8], [-7, -3, 10]]

5.42-masala

Raqamli butun massiv berilgan bo‘lsa, ushbu uzunliklarning uchtasidan hosil bo‘lgan yuzasi nolga teng bo‘lmagan uchburchakning eng katta perimetrini qaytaring. Agar uchburchakni hosil qilib bo'lmasa, 0 ni qaytaring.

Dastur kodi	Dastur natijasi

def katta_perimetr(son):
  son.sort()
  n=len(son)
  for i in range(1,n-1):
  if son[n-i-2]+son[n-i-1]>son[n-i]:
  return son[n-i-2]+son[n-i-1]+son[n-i]
  return 0
 	
print(katta_perimetr([2,1,2]))
print(katta_perimetr([6,8,10]))
print(katta_perimetr([3,2,8]))
	5
	24
	0

5.43-masala

Massivni aylantirish

n ta elementdan iborat massivni k qadam bilan o‘ngga burang. Misol uchun, n = 7 va k = 3 qiymat kiritilganda [1,2,3,4,5,6,7] massivni [5,6,7,1,2,3,4] massivga aylantiriladi.

Dastur kodi	Dastur natijasi

class Solution(object):
  def rotate(self,num,k):
  n=len(nums)
  if n < 2 or k == 0:
  pass
  else:
	[5, 6, 7, 1, 2, 3, 4]

```
        if k >= n:
            k = k % n
        a= n - k
        self.reverse(nums, 0, a-1)
        self.reverse(nums, a, n-1)
        self.reverse(nums, 0, n-1)



    def reverse(self, nums, start, end):
        i = start
        j = end
        while i < j:
            nums[i], nums[j] = nums[j], nums[i]
            i +=1
            j -=1

if __name__ == "__main__":

    soln=Solution()
    nums=[1,2,3,4,5,6,7]
    soln.rotate(nums, 3)
    print(nums)
```

5.44-masala

Massiv raqamlari berilgan bo'lsa, barcha 0 larni oxirigacha ko'chirish funksiyasini yozing. Nolga teng bo'lmagan elementlarning nisbiy tartibini saqlang.

Masalan, berilgan sonlar = [0, 1, 0, 3, 12], funksiyani chaqirgandan so'ng, [1, 3, 12, 0, 0] sonlar bo'lishi kerak. Eslatma: Buni massiv nusxasini olmasdan joyida bajarishingiz kerak.

Dastur kodi	Dastur natijasi

def moveZeroes(nums):
  i=0
  j=0
  while j < len(nums):
  if nums[j] == 0:
  j += 1
  else:
  nums[i] = nums[j]
  i += 1
  j += 1
 	
  while i < len(nums):
  nums[i] = 0
  i += 1
  return nums
 	
print(moveZeroes([0,2,5,0,355,21,0,65]))
print(moveZeroes([0]))
print(moveZeroes([1,-1,2]))
	[2, 5, 355, 21, 65, 0, 0, 0]
	[0]
	[1, -1, 2]

5.45-masala

Butun sonlar massivi ‘nums’ va ‘k’ butun son berilgan. Massivdagi o`rni orasidagi farq k dan katta bo`lmagan hamda nums[i] = nums[j] shartni qanoatlantiruvchi o`zaro farqlanadigan massivning i va j indekslari mavjud yoki yo`qligini tekshirish uchun Python kodini yozing.

Misol 1: Kiritish: nums = [1,2,3,1], k = 3, Chiqarish: true

Misol 2: Kiritish: nums = [1,0,1,1], k = 1,Chiqarish: true

Misol 3: Kiritish: nums = [1,2,3,1,2,3], k = 2, Chiqarish: false

Dastur kodi	Dastur natijasi

def containsNearbyDuplicate(nums,k):
  if not nums:
  return False
  elif len(nums) == 1:
  return False
  elif len(nums) == 2:
  if nums[0] != nums[1]:
  return False
  else:
  if nums[0] == nums[1] and k>=1:
  return True
  else:
  return False
  else:
  index_dict={}
  for i in range(len(nums)):
  if nums[i] in index_dict:
  prev_index = index_dict[nums[i]]
  if i - prev_index <=k:
  return True
  index_dict[nums[i]] = i
  return False
 	
print(containsNearbyDuplicate([1,2,3,2,1,0],1))
print(containsNearbyDuplicate([1,2,3,2,1,0],2))
	False
	True

5.46-masala

n o'lchamli massivni hisobga olgan holda, ko'p uchraydigan elementni qaytaring. Ko‘p uchraydigan element ⌊n / 2⌋ martadan ko‘proq ko‘rinadigan elementdir. Ko'pchilik element har doim massivda mavjud deb taxmin qilishingiz mumkin.

Misol 1: Kiritish: nums = [3,2,3], Chiqarish: 3

Misol 2: Kiritish: nums = [2,2,1,1,1,2,2], Chiqarish: 2

Dastur kodi	Dastur natijasi

def majorityElement(nums):
  candidate = get_candidate(nums)
  candidate_count = 0
  if candidate != None:
  for entries in nums:
  if entries == candidate:
  candidate_count += 1
  if candidate_count >= len(nums)//2:
  return candidate
  else:
  return None
  else:
  return None
 	
def get_candidate(nums):
  count = 0
  candidate = None
  for entries in nums:
  if count == 0:
  candidate = entries
  count = 1
	3
	None

```
        else:
            if candidate == entries:
                count += 1
            else:
                count -= 1
    if count > 0:
        return candidate
    else:
        return None


print(majorityElement([3,2,3]))
print(majorityElement([2,5]))
```

19-§. RO`YXATGA DOIR MASALALAR

5.47-masala

Ro‘yxatdagi eng kichik sonni olish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def ruyxat_kichik_son(list):
  min = list[0]
  for a in list:
  if a < min:
  min = a
  return min
print(ruyxat_kichik_son([1, 2, -8, 0]))
	-8

5.48-masala

Ikkita ro‘yxat umumiy elementga ega yoki yo`qligini tekshirish uchun Python funksiyasini yozing.

Dastur kodi	Dastur natijasi

def umum_element(list1, list2):
  natija = False
  for x in list1:
  for y in list2:
  if x == y:
  natija = True
  return natija
print(umum_element([1,2,3,4,5], [5,6,7,8,9]))
print(umum_element([1,2,3,4,5], [6,7,8,9]))
	True
	None

5.49-masala

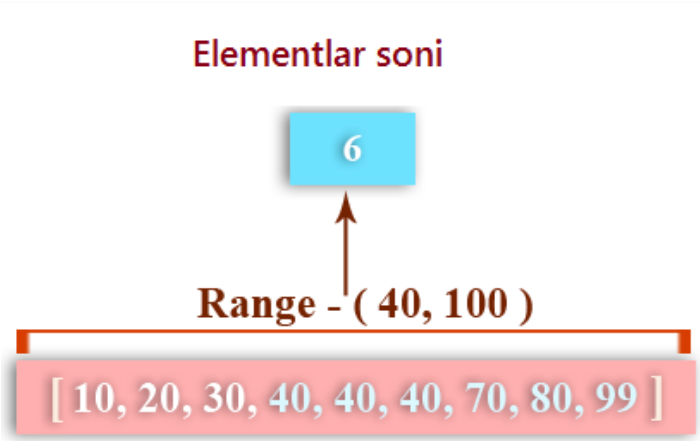
Berilgan satrdan ikkita satr yaratish uchun Python dasturini yozing. 1-qatorda 1 marta uchraydigan va 2-qatorda ko`p uchraydigan belgilardan iborat satr yarating.

Dastur kodi	Dastur natijasi

from collections import Counter
def generateStrings(input):
  str_char_ctr = Counter(input)
  part1 = [key for (key,count) in str_char_ctr.items() if count==1]
  part2 = [key for (key,count) in str_char_ctr.items() if count>1]
  part1.sort()
  part2.sort()
  return part1,part2
input = "aabbcceffgh"
s1, s2 = generateStrings(input)
print(''.join(s1))
print(''.join(s2))
	egh
	abcf

5.50-masala

Belgilangan diapazondagi ro'yxatdagi elementlar sonini hisoblash uchun Python dasturini yozing.

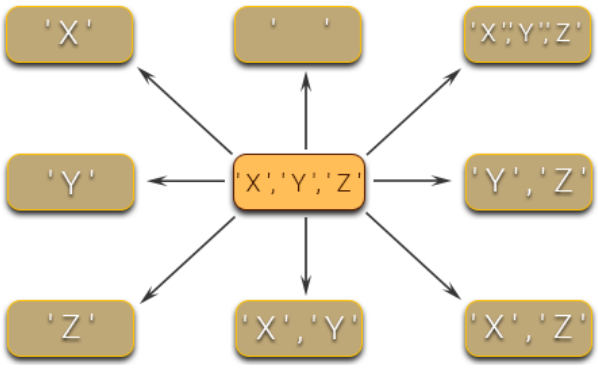


Dastur kodi	Dastur natijasi

def ruyxat_element_sanash(li, min, max):
    ctr = 0
    for x in li:
        if min <= x <= max:
            ctr += 1
    return ctr
 	
list1 = [10,20,30,40,40,40,70,80,99]
print(ruyxat_element_sanash(list1, 40, 100))
 	
list2 = ['a','b','c','d','e','f']
print(ruyxat_element_sanash(list2, 'a', 'e'))
	6
	5

5.51-masala

Ro'yxatning barcha ro'yxatostilarini yaratish uchun Python dasturini yozing.



Dastur kodi	Dastur natijasi
from itertools import combinations def sub_lists(my_list): subs = [] for i in range(0, len(my_list)+1):	Original ro`yxat: [10, 20, 30, 40] Ro`yxatning ro`yxatostilari:

temp = [list(x) for x in combinations(my_list, i)] if len(temp)>0: subs.extend(temp) return subs l1 = [10, 20, 30, 40] l2 = ['X', 'Y', 'Z'] print("Original ro`yxat:") print(l1) print("Ro`yxatning ro`yxatostilari:") print(sub_lists(l1)) print("\nOriginal ro`yxat:") print(l2) print("Ro`yxatning ro`yxatostilari:") print(sub_lists(l2))	[[], [10], [20], [30], [40], [10, 20], [10, 30], [10, 40], [20, 30], [20, 40], [30, 40], [10, 20, 30], [10, 20, 40], [10, 30, 40], [20, 30, 40], [10, 20, 30, 40]] Original ro`yxat: ['X', 'Y', 'Z'] Ro`yxatning ro`yxatostilari: [[], ['X'], ['Y'], ['Z'], ['X', 'Y'], ['X', 'Z'], ['Y', 'Z'], ['X', 'Y', 'Z']]

5.52-masala

Lug'atdan dublikatlarni olib tashlash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
student_data = {'id1': {'Nomi': ['Sara'], 'Sinf': ['V'], 'Fanlar': ['Ingliz tili, Matematika, Fan'] }, 'id2': {'Nomi': ['David'],	{'id1': {'Sinf': ['V'], 'Nomi': ['Sara'], 'Fanlar': ['Ingliz tili, Matematika, Fan']}, 'id2': {'Sinf': ['V'], 'Nomi': ['David'], 'Fanlar': ['Ingliz tili, Matematika, Fan']},

```
    'Sinf': ['V'],
    'Fanlar': ['Ingliz tili, Matematika, Fan']
   },
 'id3':
    {'Nomi': ['Sara'],
    'Sinf': ['V'],
    'Fanlar': ['Ingliz tili, Matematika, Fan']
   },
 'id4':
   {'Nomi': ['Surya'],
    'Sinf': ['V'],
    'Fanlar': ['Ingliz tili, Matematika, Fan']
   },
}

result = {}

for key,value in student_data.items():
    if value not in result.values():
        result[key] = value

print(result)
```

```
'id4': {'Sinf': ['V'],
'Nomi': ['Surya'],
'Fanlar': ['Ingliz tili,
Matematika, Fan']}}
```

5.53-masala

Taqdim etilgan lug'atda berilgan qiymatga ega bo'lgan barcha kalitlarni topish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def test(dict, val):
  return list(key for key, value in dict.items() if value == val)
students = {
  'Malika': 19,
  'Ruxshona': 20,
  'Dinara': 21,
  'Nargis': 20
}
print("\nOriginal lug`at elementlari:")
print(students)
print("\nBerilgan qiymatga ega kalitlar:")
print(test(students, 20))
	Original lug`at elementlari:
	{'Nargis': 20,
	'Malika': 19,
	'Ruxshona': 20,
	'Dinara': 21}
	
	Berilgan qiymatga ega kalitlar:
	['Nargis',
	'Ruxshona']

5.54-masala

Belgilangan lug'atni berilgan ro'yxatdan olib tashlash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def remove_dictionary(colors, r_id):
  colors[:] = [d for d in colors if d.get('id') != r_id]
  return colors
 	
colors = [{"id" : "#FF0000", "color" : "Red"},
  {"id" : "#800000", "color" : "Maroon"},
  {"id" : "#FFFF00", "color" : "Yellow"},
  {"id" : "#808000", "color" : "Olive"}]
print('Lug`atdagi original ro`yxat:')
print(colors)
	Lug`atdagi original ro`yxat:
	[{'color': 'Red', 'id': '#FF0000'}, {'color': 'Maroon', 'id': '#800000'}, {'color': 'Yellow', 'id': '#FFFF00'}, {'color': 'Olive', 'id': '#808000'}]

r_id = "#FF0000" print("\nLug`atdan ",r_id," id ni o`chirish:") print(remove_dictionary(colors, r_id))	Lug`atdan #FF0000 id ni o`chirish: [{'color': 'Maroon', 'id': '#800000'}, {'color': 'Yellow', 'id': '#FFFF00'}, {'color': 'Olive', 'id': '#808000'}]

5.55-masala

Lug'atni qiymatlar asosida filtrlash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
baho = {'Nigina Atayeva': 175, 'Yulduz G`aniyeva': 165, 'Adolat Yaxshiyeva': 190} print("Original Lug`at:") print(baho) print("170 dan katta qiymatli elementlar:") natija = {key:value for (key, value) in baho.items() if value >= 170} print(natija)	Original Lug`at: {'Yulduz G`aniyeva': 165, 'Adolat Yaxshiyeva': 190, 'Nigina Atayeva': 175} 170 dan katta qiymatli elementlar: {'Adolat Yaxshiyeva': 190, 'Nigina Atayeva': 175}

5.56-masala

Keling, ro'yxatni tushunish haqida bilib olaylik! Sizga kuboidning o'lchamlarini butun son bilan ifodalovchi uchta x,y,z berilgan. Yig'indisi n ga teng bo'lmagan 3D panjarada berilgan barcha i+j+k mumkin bo'lgan koordinatalar ro'yxatini chop eting. Bu yerda,

$0 \leq i \leq x$

$0 \leq j \leq y$

$0 \leq k \leq z$

Dastur kodi	Dastur natijasi

if __name__=='__main__':
  x=int(input())
  y=int(input())
  z=int(input())
  n=int(input())
  l=[]
  for i in range (x+1):
  for j in range (y+1):
  for k in range (z+1):
  if (i+j+k)!=n:
  l.append([i,j,k])
  print(l)
	1
	1
	1
	2
	[[0, 0, 0], [0, 0, 1], [0, 1, 0], [1, 0, 0], [1, 1, 1]]

20-§. QIDIRUV VA TARTIBLASH MASALALARI

5.57-masala

Ikkilik qidiruv uchun Python dasturini yozing. Kompyuter fanida ikkilik qidiruv yoki yarim intervalli qidiruv algoritmi tartiblangan massiv ichida ma`lum qiymatning o'rnini topadi.

Dastur kodi	Dastur natijasi

def binary_search(item_list,item):
    first = 0
    last = len(item_list)-1
    found = False
    while(first<=last and not found):
        mid = (first + last)//2
        if item_list[mid] == item :
            found = True
        else:
            if item < item_list[mid]:
                last = mid - 1
            else:
                first = mid + 1
    return found
 	
print(binary_search([1,2,3,5,8], 12))
print(binary_search([1,2,3,5,8], 2))
	False
	True

5.58-masala

Ketma-ket qidirish uchun Python dasturini yozing. Informatika fanida chiziqli qidiruv yoki ketma-ket qidiruv - bu ro'yxatdagi ma'lum bir qiymatni

topish usuli bo'lib, kerakli element topilmaguncha yoki ro'yxat tugamaguncha har bir elementni ketma-ketlikda tekshiradi.

Dastur kodi	Dastur natijasi

def Sequential_Search(dlist, item):
  pos = 0
  found = False
  while pos < len(dlist) and not found:
  if dlist[pos] == item:
  found = True
  else:
  pos = pos + 1
  return found, pos
print(Sequential_Search([11,23,58,31,56,77,43,12,65,19],31))
	(True, 3)

5.59-masala

Tartiblangan ro'yxatni ikkilik qidirish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def Ordered_binary_Search(olist, item):
 	
  if len(olist) == 0:
  return False
  else:
  midpoint = len(olist) // 2
  if olist[midpoint] == item:
  return True
  else:
  if item < olist[midpoint]:
	True
	False

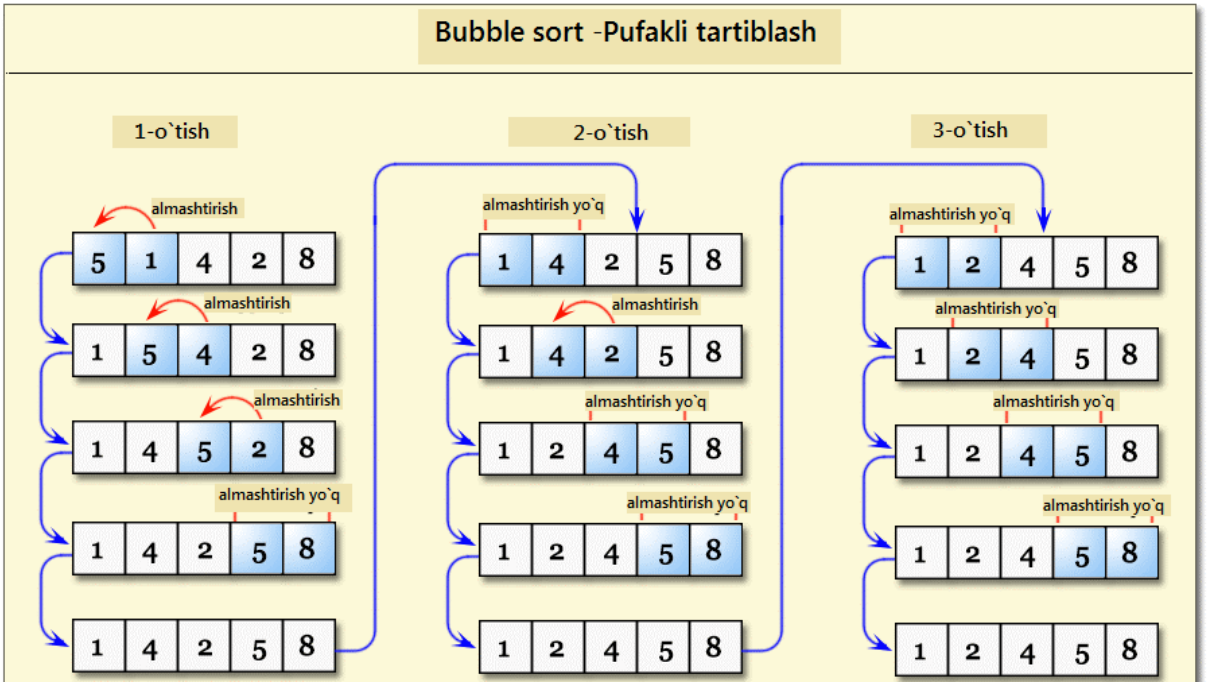
```
            return
binarySearch(olist[:midpoint], item)
        else:
            return
binarySearch(olist[midpoint+1:], item)

def binarySearch(alist, item):

    first = 0
    last = len(alist) - 1
    found = False

    while first <= last and not found:
        midpoint = (first + last) // 2
        if alist[midpoint] == item:
            found = True
        else:
            if item < alist[midpoint]:
                last = midpoint - 1
            else:
                first = midpoint + 1

    return found

print(Ordered_binary_Search([0, 1, 3,
8, 14, 18, 19, 34, 52], 3))
print(Ordered_binary_Search([0, 1, 3,
8, 14, 18, 19, 34, 52], 17))
```

5.60-masala

Bubble sort-Pufakli tartiblash

Bubble sort tartiblash algoritmidan foydalangan holda elementlar ro'yxatini tartiblash uchun Python dasturini yozing.

Eslatma: Vikipediyaga ko‘ra “Bubble sort, ba’zan pufakli tartiblash deb ataladi, bu oddiy tartiblash algoritmi bo‘lib, tartiblash kerak bo‘lgan ro‘yxat bo‘ylab qayta-qayta qadam qo‘yadi, har bir juft qo‘shni elementlarni taqqoslaydi va agar ular noto‘g‘ri tartibda bo‘lsa, ularni almashtiradi. Ro‘yxat bo‘yicha hech qanday almashtirish kerak bo‘lmaguncha takrorlanadi, bu ro‘yxat tartiblanganligini ko‘rsatadi.



Dastur kodi	Dastur natijasi

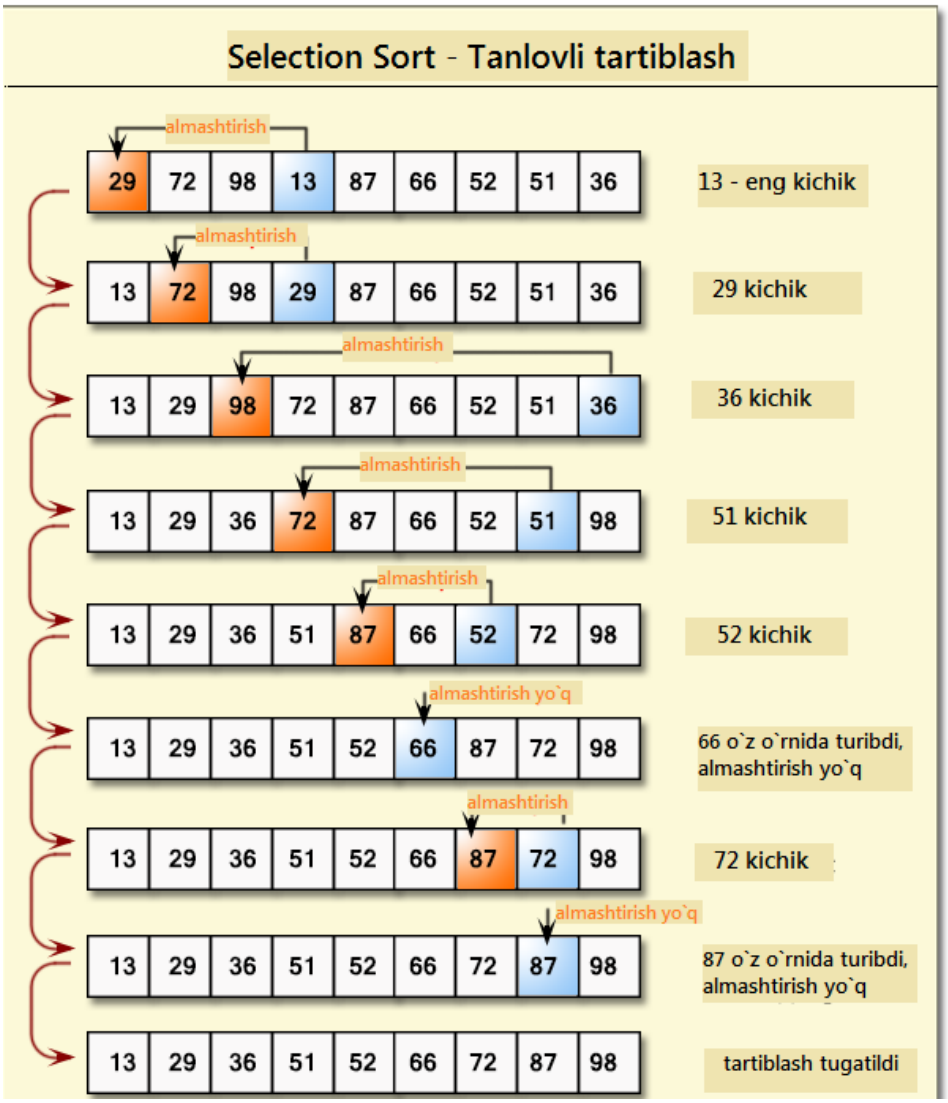
def bubbleSort(nlist):
  for passnum in range(len(nlist)-1,0,-1):
  for i in range(passnum):
  if nlist[i]>nlist[i+1]:
  temp = nlist[i]
  nlist[i] = nlist[i+1]
	[11, 15, 22, 23, 23, 42, 57, 68, 89]

nlist[i+1] = temp nlist = [89,23,15,68,57,42,23,11,22] bubbleSort(nlist) print(nlist)	

5.61-masala

Selection sort-Tanlovli tartiblash

Tanlovli tartiblash algoritmidan foydalangan holda elementlar ro'yxatini tartiblash uchun Python dasturini yozing. Eslatma: Tanlovli tartiblash ro‘yxat bo‘ylab har bir o‘tish uchun faqat bitta almashish orqali bajariladi, bu esa pufakli tartiblashdan yaxshiroqdir.



Dastur kodi	Dastur natijasi

def selectionSort(nlist):
  for fillslot in range(len(nlist)-1,0,-1):
  maxpos=0
  for location in range(1,fillslot+1):
  if nlist[location]>nlist[maxpos]:
  maxpos = location
 	
  temp = nlist[fillslot]
  nlist[fillslot] = nlist[maxpos]
  nlist[maxpos] = temp
 	
nlist = [89,23,15,68,57,42,23,11,22]
selectionSort(nlist)
print(nlist)
	[11, 15, 22, 23, 23, 42, 57, 68, 89]

5.62-masala

Sanoqni tartiblash algoritmi

Sanoqlarni tartiblash uchun Python dasturini yozing. Vikipediyaga ko'ra "Informatika fanida sanoqni tartiblash kichik butun sonlar bo'lgan kalitlarga ko'ra obyektlar to'plamini tartiblash algoritmidir; ya'ni bu butun sonlarni tartiblash algoritmidir. U har bir alohida kalitga ega bo'lgan obyektlar sonini hisoblash orqali ishlaydi.

Dastur kodi	Dastur natijasi

def counting_sort(array1, max_val):
  m = max_val + 1
  count = [0] * m
 	
  for a in array1:
	[1, 1, 1, 2, 2, 2, 2, 3, 3, 4, 7]

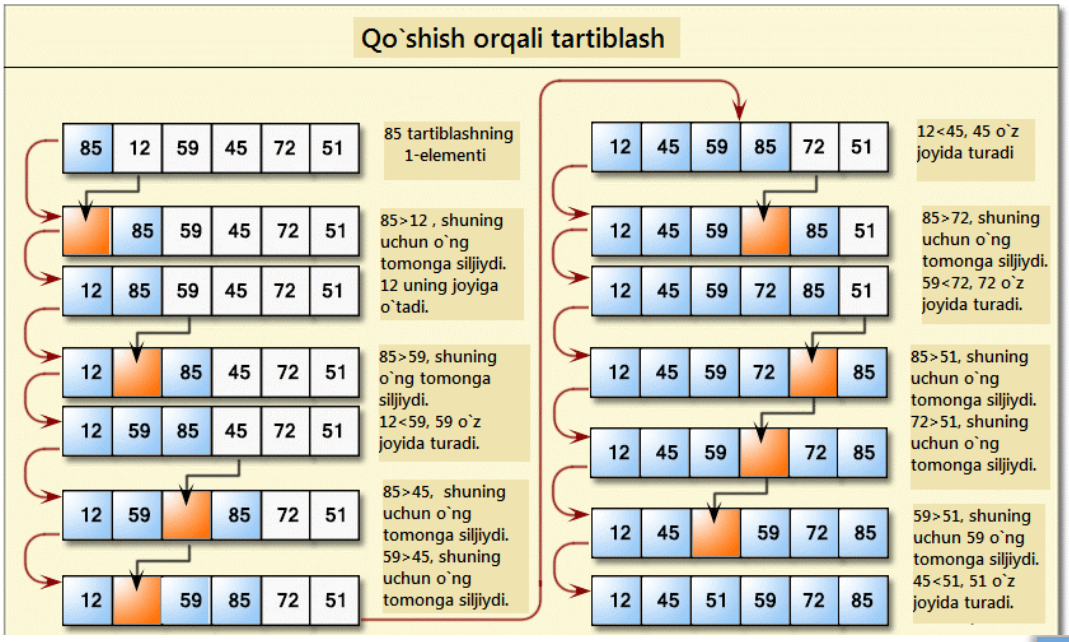
```
    # count occurences
        count[a] += 1
    i = 0
    for a in range(m):
        for c in range(count[a]):
            array1[i] = a
            i += 1
    return array1


print(counting_sort( [1, 2, 7, 3, 2, 1, 4,
2, 3, 2, 1], 7 ))
```

5.63-masala

Insertion Sort-Qo`shish orqali tartiblash

Qo`shish orqali tartiblash algoritmidan foydalangan holda elementlar ro'yxatini tartiblash uchun Python dasturini yozing.

Eslatma: Vikipediyaga ko'ra "Qo'shish orqali tartiblash bir vaqtning o'zida yakuniy saralangan massivni (yoki ro'yxatni) bitta elementni yaratadigan oddiy tartiblash algoritmidir. Bu tezkor saralash bo`lib, katta ro'yxatlarda unchalik samarali emas".



Dastur kodi	Dastur natijasi

def insertionSort(nlist):
  for index in range(1,len(nlist)):
 	
  currentvalue = nlist[index]
  position = index
 	
  while position>0 and nlist[position-1]>currentvalue:
  nlist[position]=nlist[position-1]
  position = position-1
 	
  nlist[position]=currentvalue
 	
nlist = [14,46,43,27,57,41,45,21,70]
insertionSort(nlist)
print(nlist)
	[14, 21, 27, 41, 43, 45, 46, 57, 70]

5.64-masala

Shell tartiblash

Shell sort algoritmidan foydalanib elementlar ro'yxatini tartiblash uchun Python dasturini yozing.

Eslatma : Vikipediyaga ko'ra "Shell sort yoki Shell usuli, o'z o'rnida taqqoslash turidir. Usul bir-biridan uzoqda joylashgan juft elementlar, so'ngra taqqoslanadigan elementlar orasidagi bo'shliqni bosqichma-bosqich qisqartiradigan tartiblashdan boshlanadi. Bir-biridan uzoqda joylashgan elementlardan boshlab, ba'zi bir joydan tashqari elementlarni oddiy yaqin qo'shni almashishdan ko'ra tezroq joyga ko'chirish mumkin."

Dastur kodi	Dastur natijasi
def shellSort(alist): sublistcount = len(alist)//2 while sublistcount > 0: for start_position in range(sublistcount): gap_InsertionSort(alist, start_position, sublistcount) print(sublistcount, " marta o`zgargandan keyin ro'yxat: ",nlist) sublistcount = sublistcount // 2 def gap_InsertionSort(nlist,start,gap): for i in range(start+gap,len(nlist),gap): current_value = nlist[i] position = i while position>=gap and nlist[position-gap]>current_value: nlist[position]=nlist[position-gap] position = position-gap nlist[position]=current_value nlist = [14,46,43,27,57,41,45,21,70] shellSort(nlist) print(nlist)	4 marta o`zgargandan keyin ro'yxat: [14, 41, 43, 21, 57, 46, 45, 27, 70] 2 marta o`zgargandan keyin ro'yxat: [14, 21, 43, 27, 45, 41, 57, 46, 70] 1 marta o`zgargandan keyin ro'yxat: [14, 21, 27, 41, 43, 45, 46, 57, 70] [14, 21, 27, 41, 43, 45, 46, 57, 70]

5.65-masala

Birlashtirish tartiblashi

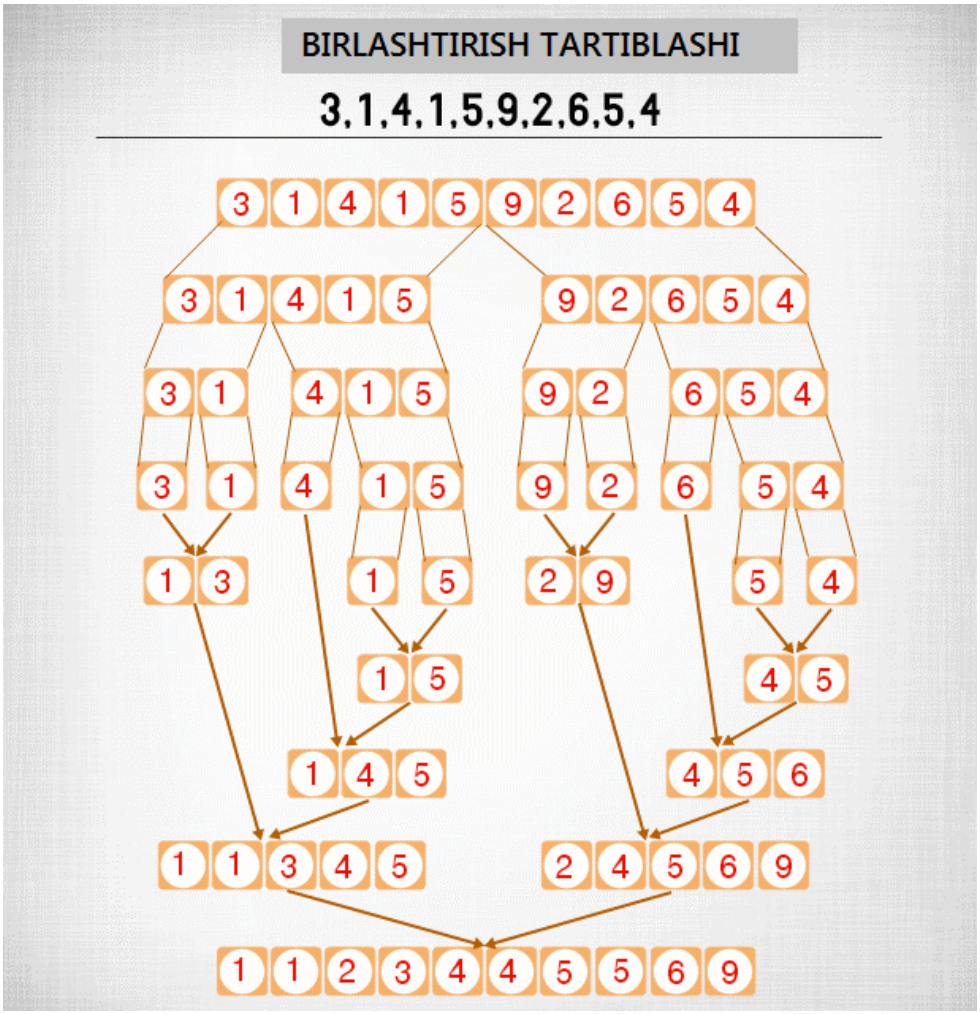
Birlashtirish tartiblashi algoritmidan foydalangan holda elementlar ro'yxatini tartiblash uchun Python dasturini yozing.

Eslatma: Vikipediyaga ko'ra “Birlashtirish tartiblashi O (n log n) bilan taqqoslashga asoslangan tartiblash algoritmidir”.

Kontseptual tarzda, birlashtirish tartibi quyidagicha ishlaydi:

Saralanmagan ro‘yxatni har birida 1 ta elementdan iborat bo‘lgan n ta pastki ro‘yxatlarga bo‘ling (1 ta elementdan iborat ro‘yxat tartiblangan hisoblanadi).

Faqat 1 ta pastki ro‘yxat qolmaguncha, yangi saralangan pastki ro‘yxatlarni yaratish uchun quyi ro‘yxatlarni qayta-qayta birlashtiring. Bu tartiblangan ro'yxat bo'ladi.



Dastur kodi	Dastur natijasi
def mergeSort(nlist): print("Bo`linish ",nlist) if len(nlist)>1: mid = len(nlist)//2 lefthalf = nlist[:mid] righthalf = nlist[mid:] mergeSort(lefthalf) mergeSort(righthalf) i=j=k=0 while i < len(lefthalf) and j < len(righthalf): if lefthalf[i] < righthalf[j]: nlist[k]=lefthalf[i] i=i+1 else: nlist[k]=righthalf[j] j=j+1 k=k+1 while i < len(lefthalf): nlist[k]=lefthalf[i] i=i+1 k=k+1 while j < len(righthalf): nlist[k]=righthalf[j] j=j+1 k=k+1 print("Qo`shilish ",nlist)	Bo`linish  [14, 43, 27] Bo`linish [14] Qo`shilish  [14] Bo`linish [43, 27] Bo`linish  [43] Qo`shilish [43] Bo`linish  [27] Qo`shilish [27] Qo`shilish  [27, 43] Qo`shilish [14, 27, 43] [14, 27, 43]

nlist = [14,43,27] mergeSort(nlist) print(nlist)	

5.66-masala

Tez tartiblash

Tez tartiblash algoritmi yordamida elementlar ro‘yxatini tartiblash uchun Python dasturini yozing.

Eslatma: Vikipediyaga ko'ra, "Quicksort - belgilangan har qanday turdagi elementlarni saralashi mumkin. Samarali amalga oshirishda bu barqaror tur emas, ya'ni teng tartiblash elementlarining nisbiy tartibi saqlanmaydi. Tezkor saralash massivda joyida ishlashi mumkin, bu saralashni amalga oshirish uchun kichik qo'shimcha xotira hajmini talab qiladi."

Dastur kodi	Dastur natijasi

def quickSort(data_list):
  quickSortHlp(data_list,0,len(data_list)-1)
 	
def quickSortHlp(data_list,first,last):
  if first < last:
 	
  splitpoint = partition(data_list,first,last)
 	
  quickSortHlp(data_list,first,splitpoint-1)
  quickSortHlp(data_list,splitpoint+1,last)
 	
def partition(data_list,first,last):
  pivotvalue = data_list[first]
	[17, 20, 26, 31, 44,
	54, 55, 77, 93]

```
  leftmark = first+1
  rightmark = last

  done = False
  while not done:

      while leftmark <= rightmark and
data_list[leftmark] <= pivotvalue:
          leftmark = leftmark + 1

      while data_list[rightmark] >= pivotvalue and
rightmark >= leftmark:
          rightmark = rightmark -1

      if rightmark < leftmark:
          done = True
      else:
          temp = data_list[leftmark]
          data_list[leftmark] = data_list[rightmark]
          data_list[rightmark] = temp

  temp = data_list[first]
  data_list[first] = data_list[rightmark]
  data_list[rightmark] = temp



  return rightmark

data_list = [54,26,93,17,77,31,44,55,20]
```

quickSort(data_list) print(data_list)	

5.67-masala

Tree sortdan foydalanib elementlar ro'yxatini tartiblang.

Dastur kodi	Dastur natijasi

class node():
 	
  def __init__(self, val):
  self.val = val
  self.left = None
  self.right = None
 	
  def insert(self,val):
  if self.val:
  if val < self.val:
  if self.left is None:
  self.left = node(val)
  else:
  self.left.insert(val)
  elif val > self.val:
  if self.right is None:
  self.right = node(val)
  else:
  self.right.insert(val)
  else:
  self.val = val
 	
def inorder(root, res):
	[1, 2, 5, 7, 14, 17, 19]

```
    # Rekursiya
    if root:
        inorder(root.left,res)
        res.append(root.val)
        inorder(root.right,res)


def treesort(arr):
    # BST qurish
    if len(arr) == 0:
        return arr
    root = node(arr[0])
    for i in range(1,len(arr)):
        root.insert(arr[i])
    # BSTni tartiblash
    res = []
    inorder(root,res)
    return res


print(treesort([7,1,5,2,19,14,17]))
```

5.68-masala

Vaqtni saralashdan foydalanib elementlar ro'yxatini tartiblash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def binary_search(lst, item, start, end):
  if start == end:
  if lst[start] > item:
  return start
  else:
	Sonlarni kiriting(vergul bilan):
	1,5,68,25,14,48
	[1, 5, 14, 25, 48, 68]

```
            return start + 1
    if start > end:
        return start

    mid = (start + end) // 2
    if lst[mid] < item:
        return binary_search(lst, item, mid
+ 1, end)
    elif lst[mid] > item:
        return binary_search(lst, item,
start, mid - 1)
    else:
        return mid
def insertion_sort(lst):
    length = len(lst)

    for index in range(1, length):
        value = lst[index]
        pos = binary_search(lst, value, 0,
index - 1)
        lst = lst[:pos] + [value] +
lst[pos:index] + lst[index+1:]
    return lst


def merge(left, right):
    if not left:
        return right

    if not right:
```

```
        return left

    if left[0] < right[0]:
        return [left[0]] + merge(left[1:],
right)

    return [right[0]] + merge(left,
right[1:])

def time_sort(lst):
    runs, sorted_runs = [], []
    length = len(lst)
    new_run = [lst[0]]
    sorted_array = []

    for i in range(1, length):
        if i == length - 1:
            new_run.append(lst[i])
            runs.append(new_run)
            break

        if lst[i] < lst[i - 1]:
            if not new_run:
                runs.append([lst[i - 1]])
                new_run.append(lst[i])
            else:
                runs.append(new_run)
                new_run = []
        else:
            new_run.append(lst[i])
```

for run in runs:
 	
sorted_runs.append(insertion_sort(run))
 	
for run in sorted_runs:
sorted_array =
merge(sorted_array, run)
 	
return sorted_array
 	
user_input = input("Sonlarni
kiriting(vergul bilan):\n").strip()
nums = [int(item) for item in
user_input.split(',')]
print(time_sort(nums))	

5.69-masala

Ko‘p tugmali tezkor saralash yordamida tartiblanmagan sonlarni tartiblang Vikipediyadan -Ko'p kalitli tezkor saralash:

Bu algoritm radix sort va quicksort birikmasidir. Massivdan (pivot) elementni tanlang va satrning birinchi belgisini (kalitini) ko'rib chiqing (multikey). Qolgan elementlarni uchta to'plamga bo'ling: mos belgilari pivot belgisidan kichik, teng va kattaroq bo'lganlar. Xuddi shu belgi bo'yicha "kamroq" va "kattaroq" bo'limlarini rekursiv tartiblash amalga oshiriladi.

Dastur kodi	Dastur natijasi
def quick_sort_3partition(sorting: list, left: int, right: int) -> None: if right <= left:	Original ro`yxat: [4, 3, 5, 1, 2]

```
        return
    a = i = left
    b = right
    pivot = sorting[left]
    while i <= b:
        if sorting[i] < pivot:
            sorting[a], sorting[i] = sorting[i], sorting[a]
            a += 1
            i += 1
        elif sorting[i] > pivot:
            sorting[b], sorting[i] = sorting[i], sorting[b]
            b -= 1
        else:
            i += 1
    quick_sort_3partition(sorting, left, a - 1)
    quick_sort_3partition(sorting, b + 1, right)
def three_way_radix_quicksort(sorting: list) ->
list:
    if len(sorting) <= 1:
        return sorting
    return (
        three_way_radix_quicksort([i for i in sorting
if i < sorting[0]])
        + [i for i in sorting if i == sorting[0]]
        + three_way_radix_quicksort([i for i in
sorting if i > sorting[0]])
    )
nums = [4, 3, 5, 1, 2]
print("\nOriginal ro`yxat:")
print(nums)
```

```
Random Pivot Quick
Sort tartiblash
qo'llanilgandan so'ng:
[1, 2, 3, 4, 5]

Original ro`yxat:
[5, 9, 10, 3, -4, 5, 178,
92, 46, -18, 0, 7]
Multi-key quicksort
saralashni
qo'llaganingizdan so'ng:
[-18, -4, 0, 3, 5, 5, 7, 9,
10, 46, 92, 178]

Original ro`yxat:
[1.1, 1, 0, -1, -1.1, 0.1]
Multi-key quicksort
saralashni
qo'llaganingizdan so'ng:
[-1.1, -1, 0, 0.1, 1, 1.1]

Original ro`yxat:
[1.1, 1, 0, -1, -1.1, 0.1]
Multi-key quicksort
saralashni
qo'llaganingizdan so'ng:
[1.1, -1.1, -1, 0, 0.1, 1]

Original ro`yxat:
['z', 'a', 'y', 'b', 'x', 'c']
```

print("Random Pivot Quick Sort tartiblash qo'llanilgandan so'ng:") quick_sort_3partition(nums, 0, len(nums)-1) print(nums) nums = [5, 9, 10, 3, -4, 5, 178, 92, 46, -18, 0, 7] print("\nOriginal ro`yxat:") print(nums) print("Multi-key quicksort saralashni qo'llaganingizdan so'ng:") quick_sort_3partition(nums, 0,  len(nums)-1) print(nums) nums = [1.1, 1, 0, -1, -1.1, .1] print("\nOriginal ro`yxat:") print(nums) print("Multi-key quicksort saralashni qo'llaganingizdan so'ng:") quick_sort_3partition(nums, 0, len(nums)-1) print(nums) nums = [1.1, 1, 0, -1, -1.1, .1] print("\nOriginal ro`yxat:") print(nums) print("Multi-key quicksort saralashni qo'llaganingizdan so'ng:") quick_sort_3partition(nums, 1,  len(nums)-1) print(nums) nums = ['z','a','y','b','x','c'] print("\nOriginal ro`yxat:") print(nums) print("Multi-key quicksort saralashni qo'llaganingizdan so'ng:")	Multi-key quicksort saralashni qo'llaganingizdan so'ng: ['a', 'b', 'c', 'x', 'y', 'z'] Original ro`yxat: ['z', 'a', 'y', 'b', 'x', 'c'] Multi-key quicksort saralashni qo'llaganingizdan so'ng: ['z', 'a', 'b', 'c', 'x', 'y']

```
quick_sort_3partition(nums, 0, len(nums)-1)
print(nums)
nums = ['z','a','y','b','x','c']
print("\nOriginal ro`yxat:")
print(nums)
print("Multi-key quicksort saralashni
qo'llaganingizdan so'ng:")
quick_sort_3partition(nums, 2,  len(nums)-1)
print(nums)
```

21-§. REKURSIYA MASALALARI

5.70-masala

Sonlar yig'indisini hisoblash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def list_sum(num_List):
  if len(num_List) == 1:
  return num_List[0]
  else:
  return num_List[0] + list_sum(num_List[1:])
 	
print(list_sum([6,8,9,12,15,16]))
	66

5.71-masala

Butun sonni istalgan sanoq sistemasiga o`tish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def to_string(n,base):
  conver_tString = "0123456789ABCDEF"
  if n < base:
  return conver_tString[n]
  else:
  return to_string(n//base,base) + conver_tString[n % base]
 	
print(to_string(28,2))
	11100

5.72-masala

Ro'yxat yig'indisini Rekursiya orqali hisoblaydigan Python dasturini yozing.

Dastur kodi	Dastur natijasi

def recursive_list_sum(data_list):
  total = 0
  for element in data_list:
  if type(element) == type([]):
  total = total + recursive_list_sum(element)
  else:
  total = total + element
 	
  return total
print(recursive_list_sum([1, 2, [3,4],[5,6]]))
	21

5.73-masala

Manfiy bo'lmagan butun sonning faktorialini olish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def factorial(n):
  if n <= 1:
  return 1
  else:
  return n * (factorial(n - 1))
 	
print(factorial(12))
	479001600

5.74-masala

Rekursiya yordamida Fibonachchi ketma-ketligini yechish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def Fibonachchi(n):
  if n < 0:
  print("Noto`g`ri kiritdingiz")
 	
  elif n == 0:
  return 0
 	
  elif n == 1 or n == 2:
  return 1
 	
  else:
  return Fibonachchi(n-1) + Fibonachchi(n-2)
 	
k=int(input('N-Fibbonachi soni? '))
print(Fibonachchi(k))
	N-Fibbonachi soni?
	7
	13

5.75-masala

Manfiy bo'lmagan butun sonning raqamlari yig'indisini olish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def sumDigits(n):
 if n == 0:
  return 0
 else:
  return n % 10 + sumDigits(int(n / 10))
	12
	9

print(sumDigits(345)) print(sumDigits(45))	

5.76-masala

n+(n-2)+(n-4)... (n-x =< 0 gacha) musbat butun sonlar yig‘indisini hisoblash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def sum_series(n):
  if n < 1:
  return 0
  else:
  return n + sum_series(n - 2)
 	
print(sum_series(5))
print(sum_series(10))
	9
	30

5.77-masala

n-1 ning garmonik yig'indisini hisoblash uchun Python dasturini yozing.
Eslatma: Garmonik yig'indi musbat butun sonlarning teskarilarining yig'indisidir.
Masalan,

$$1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\cdots$$

Dastur kodi	Dastur natijasi

def harmonic_sum(n):
  if n < 2:
  return 1
  else:
	1.5
	2.083333333333333

return 1 / n + (harmonic_sum(n - 1)) print(harmonic_sum(2)) print(harmonic_sum(4))	

5.78-masala

'a' ning 'b' darajasini hisoblash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def power(a,b):
    if b==0:
        return 1
    elif a==0:
        return 0
    elif b==1:
        return a
    else:
        return a*power(a,b-1)
 	
print(power(5,3))
	125

5.79-masala

Ikkita butun sonning eng katta umumiy bo‘luvchisini (EKUB) topish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi

def ekub(a, b):
    low = min(a, b)
    high = max(a, b)
	4

if low == 0:
return high
elif low == 1:
return 1
else:
return ekub(low, high%low)
print(ekub(60,52))	

5.80-masala

Berilgan n butun son, agar u ikkining darajasi bo'lsa, true qiymatini qaytaring. Aks holda, false qaytaring. Agar n == 2x bo'lgan x butun soni mavjud bo'lsa, n butun soni ikkining darajasidir.

Dastur kodi	Dastur natijasi

import math
def daraja_2(n):
if n==0:
return False
elif n==1:
return True
elif n%2!=0:
return False
else:
return daraja_2(n/2)
 	
print(daraja_2(1))
print(daraja_2(8))
print(daraja_2(255))
	True
	True
	False

5.81-masala

Agar n butun soni berilgan bo'lsa, u to'rtning darajasi bo'lsa, true qiymatini qaytaring. Aks holda, false qaytaring. Agar n == 4x bo'lgan x butun soni mavjud bo'lsa, n butun soni to'rtning darajasidir.

Dastur kodi	Dastur natijasi

def daraja_4(n):
  if n==0:
  return False
  elif n==1:
  return True
  else:
  return daraja_4(n/4)
 	
print(daraja_4(0))
print(daraja_4(4))
print(daraja_4(64))
	False
	True
	True

6-BOB. OLIMPIADA MASALALARI

22-§. OLIMPIADA MASALALARI

6.1-masala

Sizga butun sonlar qatori berilgan. Siz X-Y tekisligida (0,0) nuqtadan boshlaysiz va masofani[0] metr shimolga, so'ngra masofa[1] metr g'arbga, masofa[2] metr janubga, masofa[3] metrga o'tasiz. sharq va boshqalar. Boshqacha qilib aytganda, har bir harakatdan keyin sizning yo'nalishingiz soat miliga teskari o'zgaradi. Agar yo'lingiz o'z-o'zidan kesib o'tsa, true qiymatini qaytaring, agar u kesib o'tmasa, noto'g'ri.

Dastur kodi	Dastur natijasi

def kesishish(x):
  if x==None or len(x)<=3:
  return False
  else:
  for i in range(3,len(x)):
  if (x[i-3]>=x[i-1])and (x[i-1]<=x[i]):
  return True
  if (i>=4) and (x[i-4]+x[i]>=x[i-2]) and (x[i-3]==x[i-1]):
  return True
  if (i>=5) and (x[i-5]<=x[i-3]) and (x[i-4]<=x[i-2])and (x[i-1]<=x[i-3]) and (x[i-1]>=x[i-3]-x[i-5]) and (x[i]>=x[i-2]-x[i-4]) and (x[i]<=x[i-2]):
  return True
 	
  return False
print(kesishish([2,1,1,2]))
	True
	True
	False

print(kesishish([4,3,3,6])) print(kesishish([14,8,21,10]))	

6.2-masala

Uzunligi n bo‘lgan butun son massivi va maqsadli butun son berilgan bo‘lsa, yig‘indisi maqsadga eng yaqin bo‘ladigan tarzda uchta butun sonni toping. Uchta butun sonning yig'indisini qaytaring. Har bir kirishda aynan bitta yechim bo'ladi deb taxmin qilishingiz mumkin.

Kiritish namuna: nums = [-1,2,1,-4], target = 1

Chiqarish namuna: 2

Dastur kodi	Dastur natijasi

import sys
class Solution(object):
  def threeSumClosest(self,nums,target):
  if len(nums) in [0,1,2]:
  return 0
  else:
  min_diff=sys.maxsize
  result=0
  sorted_nums=sorted(nums)
  for i in range(len(nums)):
  start=i+1
  end=len(nums)-1
  while start<end:
 	
curr_sum=sorted_nums[i]+sorted_nums[start]+sorted_nums[end]
  diff=abs(curr_sum-target)
  if diff==0:
  return curr_sum
	2
	-4
	5

if diff<min_diff:
min_diff=diff
result=curr_sum
if curr_sum<=target:
start+=1
else:
end-=1
return result
 	
if __name__=="__main__":
 	
soln=Solution()
print(soln.threeSumClosest([-1,2,1,-4],1))
print(soln.threeSumClosest([-1,2,1,-4],-4))
print(soln.threeSumClosest([0,2,10,-5],4))	

6.3-masala

n ta butun sonli massiv va natija berilgan bo‘lsa, i indeksli uchlik sonini toping,

i, j, k sonlar[i] + sonlar[j] + sonlar[k] < natija

0 <= i < j < k < n

shartini qanoatlantirsin.

Masalan, berilgan sonlar = [-2, 0, 1, 3] va natija = 2.

Chiqarish ma`lumoti 2. Chunki yig'indilari 2 dan kichik bo'lgan ikkita uchlik bor:

[-2, 0, 1] [-2, 0, 3]

Dastur kodi	Dastur natijasi
class Solution(object): def threeSumSmaller(self,nums,target):	[(-4, -1, 2), (-4, 1, 2)]

```
        if len(nums) == 0 or len(nums) == 2 or len(nums) == 1:
            return len([])
        else:
            triplet_list = []
            sorted_nums = sorted(nums)
            for i in range(0,len(nums)-2):
                start = i+1
                end = len(nums)-1
                while start<end:

curr_sum=sorted_nums[i]+sorted_nums[start]+sorted_nums[end]
                    if curr_sum == target:
                        end -= 1
                    elif curr_sum < target:

triplet=(sorted_nums[i],sorted_nums[start],sorted_nums[end])
                        triplet_list.append(triplet)
                        start += 1
                    elif curr_sum > target:
                        end -= 1
                print(triplet_list)
                return len(triplet_list)

if __name__=="__main__":

    soln=Solution()
    print(soln.threeSumSmaller([-1,2,1,-4],1))
```

2

6.4-masala

Kamaymaydigan tartibda tartiblangan butun sonlar massivini hisobga olib, ikkita sonni topingki, ular yig`indisi target(berilgan son) ga teng bo`lsin. Bu ikki son numbers[index1] va numbers[index2] bo'lsin, bunda

1 <= indeks1 < indeks2 <= len(numbers).

Uzunligi 2 bo‘lgan butun son massivi [indeks1, indeks2] sifatida bittaga qo‘shilgan ikkita raqam, indeks1 va indeks2 indekslarini qaytaring. Sinovlar shunday yaratilganki, aynan bitta yechim mavjud. Xuddi shu elementni ikki marta ishlata olmaysiz.

Dastur kodi	Dastur natijasi

def twoSum(numbers, target):
  if len(numbers) == 0:
  return [-1]
  else:
  start = 0
  end = len(numbers) - 1
  while start < end:
  curr_sum = numbers[start] + numbers[end]
  if curr_sum == target:
  return [start+1, end+1]
  elif curr_sum < target:
  start +=1
  elif curr_sum > target:
  end -=1
  return[-1]
 	
print(twoSum([1,2,3,4],5))
print(twoSum([0,-1,4,8,2,10,-6],5))
	[1, 4]
	[-1]

6.5-masala

Ikki son yig`indisi. U quyidagilarni qo'llab-quvvatlashi kerak operatsiyalar: add (qo'shish) va find (topish).

add - raqamni ichki ma'lumotlar strukturasiga qo'shish.

find – Agar yig`indisi qiymatga teng bo'lgan raqamlar juftligi mavjud bo'lsa, toping.

Masalan, add (1); add (3); add (5);

find(4) -> true

find(7) -> false

Dastur kodi	Dastur natijasi

import collections
 	
class TwoSum(object):
  def __init__(self):
  self.__num_list=collections.defaultdict(int)
 	
  def add(self, number):
  self.__num_list[number] +=1
 	
  def find(self, value):
  if len(self.__num_list) == 0:
  return False
  else:
  for entries in self.__num_list.keys():
  target = value - entries
	False
	True

```
            if (target in self.__num_list) and
(entries != target or self.__num_list[target] > 1 ):
                return True
        return False


if __name__ == "__main__":

    twoSum=TwoSum()
    twoSum.add(6)
    twoSum.add(8)
    print(twoSum.find(10))

if __name__ == "__main__":

    twoSum=TwoSum()
    twoSum.add(2)
    twoSum.add(3)
    print(twoSum.find(5))
```

6.6-masala

Sarlavha kodlovchi

Sizga kitob nomlari bilan "books.txt" nomli fayl beriladi, ularning har biri alohida satrda. Kitob sarlavhalarini kodlash uchun siz sarlavhadagi har bir so'zning birinchi harflarini olishingiz va ularni birlashtirishingiz kerak. Masalan, “Game of Thrones” ("Taxtlar o'yini") kitobining nomi uchun kodlangan versiya "GoT" bo'lishi kerak.

Fayldan kitob nomini o'qish uchun dasturni yakunlang va har biri yangi satrda kodlangan versiyalarni chiqaring. book.txt faylida “O`tgan kunlar”, “Oltin Devor” kitob nomlari berilgan bo`lsa, natija OK,OD chiqadi.

1-usul:

Dastur kodi	Dastur natijasi

file=open('book.txt', 'r')
for line in file.readlines():
  words=line.split()
  encode=''
  for word in words:
  encode = encode + word[0]
  print(encode)
 	
file.close()
	OK
	OD

2-usul:

Dastur kodi	Dastur natijasi

with open('book.txt', 'r') as file:
  for line in file.readlines():
  words=line.split()
  encode=''
  for word in words:
  encode = encode + word[0]
  print(encode)
 	
file.close()
	OK
	OD

6.7-masala

Sharbat tayyorlovchi

Sizga nom va sig'im xususiyatlariga ega bo'lgan Sharbat sinfi beriladi. Ikkita sharbat obyektini yoqish va qo'shish uchun kodni to'ldirishingiz kerak, natijada ikkita sharbatning umumiy sig'imi va nomlari qo'shilgan yangi sharbat ob'ekti paydo bo'ladi.

Masalan, agar siz 1,0 sig'imli apelsin sharbati va 2,5 sig'imli olma sharbatini qo'shsangiz, hosil bo'lgan sharbat quyidagilarga ega bo'lishi kerak:
nomi: apelsin va olma
sig'im: 3,5. ismlar&belgisi yordamida birlashtirilgan.

Dastur kodi	Dastur natijasi

class Juice:
  def __init__(self,name,capacity):
  self.name = name
  self.capacity = capacity
 	
  def __str__(self):
  return (self.name + ' (' +str(self.capacity)+ 'L)')
  def __add__(self,other):
  return Juice(self.name + " & " +
other.name,self.capacity + other.capacity)
 	
a = Juice('Orange', 1.5)
b = Juice('Apple', 2.0)
result = a + b
print(result)
 	
a = Juice('Banana', 3.2)
b = Juice('Mandarin', 5.2)
result = a + b
print(result)
	Orange &
	Apple (3.5L)
	Banana &
	Mandarin
	(8.4L)

6.8-masala

Get Fizz, Buzz va FizzBuzz

1 dan 20 gacha bo'lgan butun sonlarni takrorlaydigan Python dasturini yozing. Uchga karrali son o'rniga "Fizz" va beshga karrali son o`rniga"Buzz" ni chop eting. Uch va beshga karrali raqamlar uchun "FizzBuzz" ni chop eting.

Dastur kodi	Dastur natijasi

for fizzbuzz in range(21):
  if fizzbuzz % 3 == 0 and fizzbuzz % 5 == 0:
  print("fizzbuzz")
  continue
  elif fizzbuzz % 3 == 0:
  print("fizz")
  continue
  elif fizzbuzz % 5 == 0:
  print("buzz")
  continue
  print(fizzbuzz)
	fizzbuzz
	1
	2
	fizz
	4
	buzz
	fizz
	7
	8
	fizz
	buzz
	11
	fizz
	13
	14
	fizzbuzz
	16
	17
	fizz
	19
	buzz

6.9-masala

Diyor va Shaxina “Mnion o`yini”ni o`ynamoqchi. Ikkala o'yinchiga ham bir xil S ipi beriladi.Ikkala o'yinchi ham S satrining harflari yordamida pastki qatorlarni yasashlari kerak.Styuart undosh tovushlardan boshlanadigan so‘zlarni yasashi kerak.Diyor unlilardan boshlanadigan so'zlarni yasashi kerak.

O'yin ikkala o'yinchi ham barcha mumkin bo'lgan pastki qatorlarni qilganda tugaydi. O'yinchi S satrida pastki qatorning har bir paydo bo'lishi uchun +1 ball oladi.

Masalan, S string =BANANA

Diyorning unli boshlang'ich so'zi=ANA

bu yerda ANA BANANAda ikki marta uchraydi. Shunday qilib, Diyor 2 ochko oladi.

Dastur kodi	Dastur natijasi

def minion_game(string):
  s=len(string)
  vowel=0
  consonant=0
  for i in range(s):
  if string[i] in 'AEIOU':
  vowel+=(s-i)
  else:
  consonant+=(s-i)
  if vowel < consonant:
  print('Shaxina '+str(consonant))
  elif vowel > consonant:
  print('Diyor '+str(vowel))
  else:
  print('')
 	
if __name__=='__main__':
  s=input()
  minion_game(s)
	ORANGE
	Diyor 11

6.10-masala

Statistika

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 15%)

Alisher ingliz tilini yoqtirmaydi, lekin barabanchi bo'lib qolish uchun har safar chorakda kamida to'rt ball olishga harakat qiladi. Joriy chorakda Alisher quyidagi qonuniyatni payqadi: oyning toq kunlarida u uch baho, juft kunlarda esa to'rt baho oladi. Bu baholarni qaysi kunlarda olganini ham eslaydi. Shuning uchun, u nechta uch va nechta to'rt borligini baholash uchun bu kunlarni qog'ozga yozdi. Alisher toq va juft raqamlarni turli qatorlarga joylashtirish orqali buni amalga oshirishga yordam bering. Alisher uchtadan kam bo'lmagan to'rtlar bo'lsa, 4 ball bilan hisoblashi mumkin.

Ma'lumotlarni kiritish

Birinchi qatorida bitta N son, butun massiv elementlari soni ($1 \leq N \leq 100$) mavjud. Ikkinchi qatorda berilgan massivni ifodalovchi N raqam mavjud. Har bir massiv elementi 1 dan 31 gacha natural sondir. Massivning barcha elementlari bo‘sh joy bilan ajratilgan.

Chiqarish

Birinchi qatorida siz Alisher uch baho qabul qilgan oy kunlariga mos keladigan raqamlarni chop etishingiz kerak, ikkinchi qatorda esa mos ravishda Alisher to'rt baho olgan oyning sanalarini joylashtirishingiz kerak. Uchinchi qatorda, agar Alisher to'rt olishi mumkin bo'lsa, "HA" va aks holda "YO'Q" deb yozing. Har bir satrda sonlar qanday kiritilgan bo'lsa, xuddi shunday tartibda chiqarilishi kerak. Chiqarishda raqamlar bo'sh joy bilan ajratiladi.

Dastur kodi	Dastur natijasi
a=int(input());toq='';juft='';t=0;j=0 b=list(map(str,input().strip().split())) for i in range(a):	Kiritish: 8 12 15 3 18 23 27 6 11

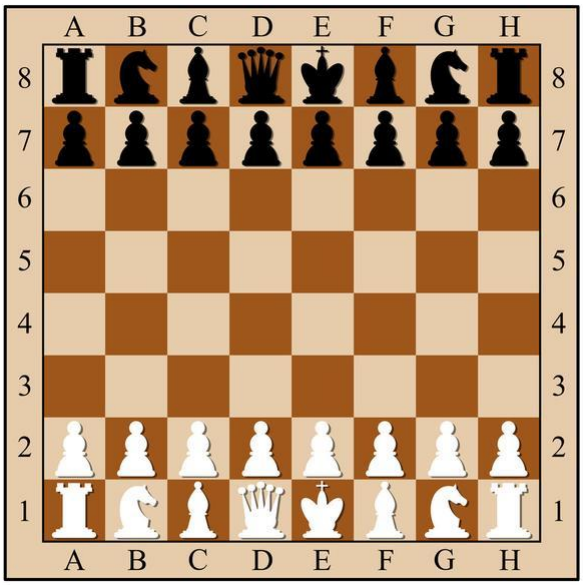
if int(b[i])%2==0:
juft=juft+b[i]+' '
j=j+1
else:
toq=toq+b[i]+' '
t=t+1
if j>=t:
total=(toq+'\n'+juft+'\n'+'YES')
else:
total=(toq+'\n'+juft+'\n'+'NO')
print(total)
	Chiqarish:
	15 3 23 27 11
	12 18 6
	NO

6.11-masala

Shaxmat

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 23%)

Shaxmat Yaqinda Nodirbek dasturlashni boshladi va shaxmat o'ynash bo'yicha o'z dasturini amalga oshirishga qaror qildi. Ammo u foydalanuvchi tomonidan amalga oshiriladigan ritsar(ot) harakatining to'g'riligini aniqlash muammosiga duch keldi. Bular. agar foydalanuvchi "C7-D5" qiymatini kiritsa, dastur buni haqiqiy harakat sifatida aniqlashi kerak, agar "E2-E4" kiritilgan bo'lsa, u holda harakat noto'g'ri. Shuningdek, siz kiritilgan yozuvning to'g'riligini tekshirishingiz kerak: agar, masalan, "D9-N5" kiritilgan bo'lsa, dastur ushbu yozuvni xato deb aniqlashi kerak. Unga ushbu testni o'tkazishga yordam bering!



Dastur kodi	Dastur natijasi

a=input()
if len(a)==5:
  if ord(a[0]) in range(65,73) and ord(a[3]) in range(65,73) and ord(a[1]) in range(49,57) and ord(a[4]) in range(49,57) and a[2]=='-':
  if abs(ord(a[0])-ord(a[3]))==1 and abs(ord(a[1])-ord(a[4]))==2 or abs(ord(a[0])-ord(a[3]))==2 and abs(ord(a[1])-ord(a[4]))==1:
  print('YES')
  else:
  print('NO')
  else:
  print('ERROR')
else:
  print('ERROR')
	D5-B4
	YES
	
	E9-A1
	ERROR

6.12-masala

Tenglama

Feruz maktabda kvadrat tenglamalarni o'rgangan va diskriminantni hisoblash orqali ularni qanday osonlik bilan yechish mumkinligini tushungan. Ammo Doston unga $A*X^3 + B*X^2 + C*X + D = 0$ ko‘rinishdagi kub tenglamalarni yechish usuli haqida gapirib berdi. Matematika fani bo‘yicha tanlov kursida Feruzga xuddi shunday turdagi yuzga yaqin tenglamalarni yechish taklif qilindi. Ammo, afsuski, Feruz Doston aytgan formulalarni unutdi. Ammo Feruz tenglamalarning barcha ildizlari butun sonlar ekanligini va [-100, 100] segmentida ekanligini bilar edi. Feruzga kub tenglamalarning ildizlarini topishga yordam beradigan dastur yozishga yordam bering!

Ma'lumotlarni kiritish

Bir qatori 4 ta raqamdan iborat: A, B, C va D kubik tenglamaning butun son koeffitsientlari. Har bir koeffitsient moduli 32768 dan kichik, $A \neq 0$.

Chiqarish

Bitta satrida berilgan kub tenglamaning barcha ildizlarini bo'sh joy bilan ajratib, o'sish tartibida chiqarish kerak. Bir nechta ildizlar faqat bir marta ko'rsatilishi kerak.

Dastur kodi	Dastur natijasi

a,b,c,d=map(int,input().split())
for x in range(-100,101):
  if a*x**3+b*x**2+c*x+d==0:
  print(x, end=" ")
	Kiritish:
	1 -7 -33 135
	Chiqarish:
	-5 3 9

6.13-masala

Ali va Vali ko'pincha turli xil mantiqiy o'yinlarni o'ynaydi. Ali yaqinda Valiga "Buqalar va sigirlar" yangi o'yini haqida gapirib berdi va endi ular bu o'yinni bir necha kun o'ynashadi. O'yinning mohiyati juda oddiy: Ali turli raqamlardan iborat to'rt xonali raqamni o'ylaydi. Vali mumkin bo'lgan variantlarni saralab, Ali tomonidan o'ylab topilgan raqamni taxmin qiladi. Har safar Vali o'zining to'rt xonali raqamining turli raqamlardan iborat variantini taklif qiladi va Ali Valiga maslahat beradi: u buqalar va sigirlarning sonini aytadi, shundan so'ng Vali maslahatni hisobga olib, raqamni taxmin qilishda davom etadi. u taxmin qiladi.

Buqalar - bu Vali tomonidan taklif qilingan raqamdagi raqamlar soni, qiymati bir-biriga mos keladigan va Ali tomonidan o'ylab topilgan raqamda to'g'ri pozitsiyada turadi. Sigirlar - qiymatga mos keladigan, ammo noto'g'ri holatda bo'lgan raqamlar soni.

Misol uchun, agar Ali 5671 raqami haqida o'ylagan bo'lsa va Vali 7251 variantini taklif qilgan bo'lsa, unda buqalar soni 1 ta (faqat 1 raqami o'z o'rnida), sigirlar soni esa 2 ta (faqat 7 va 5 raqamlari) o'z joylarida emas). Ali matematikada kuchli, lekin u ham xato qilishi mumkin. Aliga Ali taklif qilgan va Vali taklif qilgan buqalar va sigirlar sonini aytib beradigan dastur yozishga yordam bering.

Dastur kodi	Dastur natijasi

a,b=map(str,input().split())
x=[];y=[];s=0;m=0
for i in range(len(a)):
  if a[i]==b[i]:
  s=s+1
  else:
  x.append(a[i])
	Kiritish:
	2568 8596
	Chiqarish:
	1 2

```
        y.append(b[i])
for i in range(len(x)):
    if x[i] in y:
        m=m+1
print(s,m)
```

6.14-masala

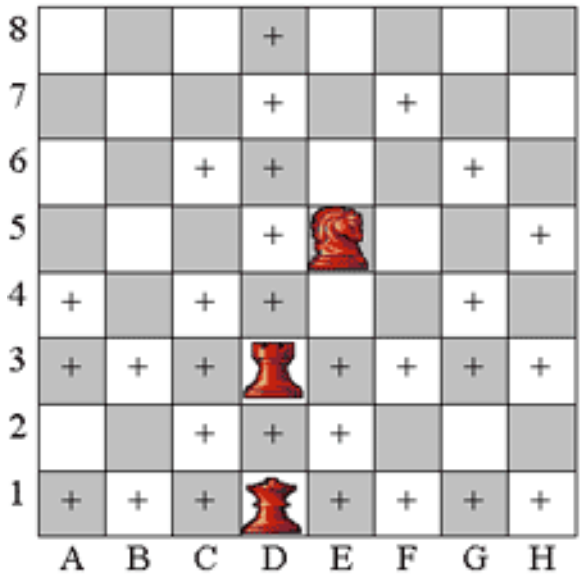
Ferz, rux va ot

8x8 shaxmat taxtasida uchta dona bor: Ferz, rux va ot. Hujum ostida bo'lgan bo'sh taxta maydonlarining sonini aniqlash talab qilinadi. Oddiylik uchun, shaxmat donalari boshqa donalarga "urilishi" mumkin deb taxmin qilamiz.

Kirish ma'lumotlari

Uchta sonni bo`sh joy orqali qayd etiladi: Farzin, rux va ot ning koordinatalari. Har bir koordinatalar bitta inglizcha harfdan (A dan H gacha) va bitta raqamdan (1dan 8gacha) iborat.

Chiqarish ma'lumotlarida ko'rsatilgan koordinatalardagi donalar yurishi mumkin bo'lgan bo'sh maydonlar sonini o'z ichiga olishi kerak.



Dastur kodi	Dastur natijasi

a=[]
#ferz
x=ord(s3[0])-65;y=int(s3[1])-1;
for i in range(8):
  a.append([0,0,0,0,0,0,0,0])
for i in range(8):
  for j in range(8):
  if (x==i)or(y==j)or(abs(x-i)==abs(y-j)):
  a[i][j]=1;
a[x][y]=-1
#ot
x=ord(s[0])-65;y=int(s[1])-1;
for i in range(8):
  for j in range(8):
  if (abs((x-i)*(y-j))==2)and(a[i][j]!=-1):
  a[i][j]=1;
a[x][y]=-1
#rux
x=ord(s1[0])-65;y=int(s1[1])-1;
for i in range(8):
  for j in range(8):
  if((x==i)or(y==j))and(a[i][j]!=-1):
  a[i][j]=1
a[x][y]=-1;
x=0
#sanash
for i in range(8):
  for j in range(8):
  if(a[i][j]==1):
	Kiritish:
	A4 B7 E3
	Chiqarish:
	34

x+=1 print(x)	

6.15-masala

Birliklar

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 16%)

Informatika darslarida, ehtimol, sizga raqamlarni bir sanoq tizimidan ikkinchisiga o'tkazish va shunga o'xshash boshqa operatsiyalarni bajarish o'rgatilgan. Bu bilimlarni namoyish qilish vaqti keldi. Berilgan sonning ikkilik sanoq tizimi ko‘rinishi va birlar sonini toping.

Kiritishda n ($0 \le n \le 2 \times 10^9$) butun son mavjud.

Chiqarish ma`lumotlari bitta satrida butun sonni ikkilik sanoq ko`rinishi va birlar sonini chiqarish kerak.

Dastur kodi	Dastur natijasi

a=int(input())
s=0
b=format(a, 'b')
print(b)
for i in range(len(str(b))):
  if b[i]=='1':
  s=s+1
  else:
  continue
print(s)
	Kiritish:
	25
	Chiqarish:
	11001
	3

6.16-masala

Folbinlik

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 13%)

Boshqa ko'plab qizlar singari, Nigora ham turli xil folbinlikni yaxshi ko'radi. Biroz oldinroq Nigora sonlarni taxmin qilishning yangi usulini o'rgandi - uni qiziqtirgan n natural soni uchun siz n qoldiqsiz bo'linadigan barcha raqamlarning yig'indisini hisoblashingiz kerak. Nigora arifmetikani umuman yoqtirmaydi va sizdan fol ochish jarayonini avtomatlashtiradigan dastur yozishingizni so'radi.

Kirish ma`lumotlarining bitta satrida Nigora beradigan n ($n \leq 1000$) natural soni mavjud.

Chiqarish ma`lumotlarida:

n sonining barcha natural bo'luvchilari yig'indisini chop eting.

Dastur kodi	Dastur natijasi

n=int(input())
s=0
for i in range(1,n+1):
  if n%i==0:
  s=s+i
print(s)
	Kiritish:
	8
	Chiqarish:
	15

6.17-masala

Ikki doira

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 17%)

Tekislikda 2 ta aylana berilgan. Ular kesishishi yoki kesishmasligini aniqlang.

Ma'lumotlarni kiritish: ikki qatordan iborat. Har bir satrda doira haqida ma'lumot mavjud - uning markazining x va y koordinatalari (absolut qiymati 5000 dan oshmaydigan butun sonlar) va radius (butun son $1 \leq r \leq 1000$).

Chiqarish ma`lumotlarida aylanalarda kamida bitta umumiy nuqta bo'lsa, "YES" ni, aks holda "YO'Q" ni chop eting.

1-usul:

Dastur kodi	Dastur natijasi

import math
x,y,r=map(int,input().split())
x1,y1,r1=map(int,input().split())
s=math.sqrt((x-x1)**2+(y-y1)**2)
if r<r1:
  r,r1=r1,r
if s+r1<r or s>r+r1:
  print('NO')
else:
  print('YES')
	Kiritish:
	0 0 5
	2 3 4
	Chiqarish:
	YES

2-usul:

Dastur kodi	Dastur natijasi

x1,y1,r1=map(int,input().split())
x2,y2,r2=map(int,input().split())
 	
if (r1+r2)*(r1+r2)>((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2)):
  print('Yes')
else:
  print('NO')
	Kiritish:
	0 0 2
	0 3 2
	Chiqarish:
	Yes

6.18-masala

O`qlar

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 20%)

Faqat '>', '<' va '-' belgilaridan iborat ketma-ketlik berilgan. Ushbu ketma-ketlikda yashiringan o'qlar sonini topish talab qilinadi. O`qlar ‘>>-->’ va ‘<--<<’ ko‘rinishdagi pastki qatorlardir.

Ma'lumotlarni kiritish: birinchi qatori '>', '<' va '-' (bo'shliqlarsiz) belgilardan iborat qatorni o'z ichiga oladi. Satr 250 ta belgidan ortiq emas.

Chiqarish ma`lumotlari: bitta satrida ko`rsatilgan o'qlarning sonini chiqarishi kerak.

Dastur kodi	Dastur natijasi

a=input();s=0;m=0
for i in range(len(a)):
  if len(a)<5:
  break
  elif i<len(a)-4:
  if a[i]=='>' and a[i+1]=='>' and a[i+2]=='-' and a[i+3]=='-' and a[i+4]=='>':
  s=s+1
  elif a[i]=='<' and a[i+1]=='-' and a[i+2]=='-' and a[i+3]=='<' and a[i+4]=='<':
  m=m+1
d=s+m
print(int(d))
	Kiritish:
	>>>>-----
	<<<>>>><<<<----
	>>-->>-<<<<--
	<<<<<<>>>>-----
	>>>>
	Chiqarish:
	2

6.19-masala

e soni

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 20%)

Chiqarish fayliga n ta xonagacha yaxlitlangan e sonini chiqaring. Bu masalada e soni aynan 2,718281828459045235360287 5 ga teng deb faraz qilamiz.

Ma'lumotlarni kiritishda n ($0 \leq n \leq 25$) butun son mavjud.

Chiqarish ma`lumotlarida muammoning javobi chiqaring.

Dastur kodi	Dastur natijasi

E='2.718281828459045235360287 5'
a=int(input())
if a==0:
  print(3)
elif a==25:
  print(E)
elif 4<int(E[a+2]):
  print((E[0:a+1])+str(int(E[a+1])+1))
else:
  print(E[0:a+2])
	Kiritish:
	6
	Chiqarish:
	2.718282

6.20-masala

Baxtli chipta

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 12%)

Jamoat transportidan foydalanasizmi? Siz yo'l haqini to'lab, raqamli(nomerlangan) chipta oldingiz. Baxtli chipta olti xonali raqamga ega bo'lgan chipta bo'lib, unda dastlabki uchta raqamning yig'indisi oxirgi uchtasining yig'indisiga teng. Bular. chipta raqami 385916 omadli, chunki 3+8+5=9+1+6. Chipta omadli yoki yo'qligini tekshiradigan dastur yozishingiz kerak.

Ma'lumotlarni kiritishda olti xonali o'nlik raqamdan iborat.

Chiqarish ma`lumotlarida agar chipta raqami N omadli bo'lsa, "YES" va aks holda "YO'Q" chiqaring.

Dastur kodi	Dastur natijasi

a=input()
if int(a[0])+int(a[1])+int(a[2])==int(a[3])+int(a[4])+int(a[5]):
  print('YES')
else:
  print('NO')
	Kiritish:
	256814
	Chiqarish:
	YES

6.21-masala

Oson hisoblash

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 25%)

n natural soni berilgan. Uni k sanoq sistemasiga o‘tkazish va bu sanoq sistemasidagi raqamlari ko‘paytmasi va raqamlari yig‘indisi o‘rtasidagi farqni topish kerak.

Masalan, n = 239, k = 8 bo'lsin. U holda n ning sakkizlik ko'rinishi 357 ga teng, masalaning javobi 3 × 5 × 7 - (3 + 5 + 7) = 90.

Ma'lumotlarni kiritish:ikkita natural sondan iborat: n va k ($1 \le n \le 109$, $2 \le k \le 10$). Bu raqamlarning ikkalasi ham o'nlik sanoq sistemasida berilgan. Chiqarish ma`lumotlariga masalaning javobini (o‘nli tizimida) chiqaring.

Dastur kodi	Dastur natijasi

a,b=map(int,input().split())
f='';d=1;k=0
while a>0:
  g=a%b
  k=k+a%b
	Kiritish:
	25 2
	Chiqarish:
	-3

a=a//b
f=str(g)+f
for i in range(len(f)):
if f[i]=='0':
d=0
break
else:
d=d*int(f[i])
e=d-k
print(e)	

6.22-masala

Basketbol

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 5%)

Basketbol uchrashuvining har 4 choraklik natijalari ma'lum. Uchrashuv g'olibini aniqlash kerak. Butun o'yin davomida eng ko'p ochko to'plagan jamoa g'alaba qozonadi.

Ma'lumotlarni kiritish: 4 qatorni o'z ichiga oladi, har bir satrda ikkita a va b butun sonlar mavjud - tegishli chorakdagi yakuniy ball. a - birinchi jamoaning chorakda to'plagan ochkolari soni, b - ikkinchi jamoaning chorakda to'plagan ochkolari soni. ($0 \le a, b \le 100$).

Chiqarish ma`lumotlarida g'olib jamoaning raqamini, durang o'ynagan taqdirda "DRAW" ni chop eting.

Dastur kodi	Dastur natijasi
a,b=map(int,input().split()) a1,b1=map(int,input().split()) a2,b2=map(int,input().split()) a3,b3=map(int,input().split())	Kiritish: 23 11 45 8 39 20

if a+a1+a2+a3>b+b1+b2+b3:
  print('1')
elif a+a1+a2+a3<b+b1+b2+b3:
  print('2')
else:
  print('DRAW')
	24 64
	Chiqarish:
	1

6.23-masala

Topishmoq

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 18%)

Muslimbek va Iymona aka va singildir. Muslimbek talaba, Iymona esa maktab o‘quvchisi. Muslimbek Iymonaga matematikadan yordam beradi. U ikkita natural son X va Y (X,Y≤1000) haqida o‘ylaydi va Iymona ularni taxmin qilishi kerak. Buning uchun Muslimbek ikkita maslahat beradi. U bu sonlarning yig'indisi S ni va ularning ko`paytmasi P ni aytadi. Iymonaga Muslimbek tomonidan o'ylab topilgan sonlarni taxmin qilishga yordam bering.

Ma'lumotlarni kiritish: bo'sh joy bilan ajratilgan ikkita S va P natural sonlarini o'z ichiga oladi.

Chiqarish ma`lumotlarida: Muslimbek tomonidan taklif qilingan ikkita X va Y sonlarini chop eting. Sonlar o`sish tartibida, bo'sh joy bilan ajratilishi kerak.

Dastur kodi	Dastur natijasi
x,y=map(int,input().split()) a=pow(x**2-4*y,1/2) b=(a+x)/2 c=x-b print(int(min(b,c)), int(max(b,c)))	Kiritish: 7 10 Chiqarish: 2 5

6.24-masala

Qator darajasi

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 30%)

$s = s_1s_2...s_n$ qator berilsin. Uni s^k ning k (k > 0) darajasi: $s^k = s_1s_2 . . .s_ns_1s_2 . . .s_n.....s_1s_2...s_n$ (k marta). Masalan, abc satrning uchinchi darajasi abcabcabc qatoridir.

s satrning k ildizi t (agar mavjud bo'lsa) t^k = s bo'ladigan qatordir. Sizning vazifangiz satrning darajasini yoki uning ildizini topadigan dastur yozishdir.

Ma'lumotlarni kiritish: birinchi qatori s qatorini o'z ichiga oladi, u ingliz alifbosining faqat kichik harflarini o'z ichiga oladi va 1000 dan oshmaydigan nolga teng bo'lmagan uzunlikka ega.

Kirish ma`lumotining ikkinchi qatorida $k \neq 0$, |k| butun son mavjud < 100001. Agar k > 0 bo'lsa, u holda s satrning k-darajasi, agar k < 0 bo'lsa, s dan |k| daraja ildizini topish kerak.

Chiqarish ma`lumotlarida vazifaga javob bo'lgan qatorni chiqaring. Agar javob uzunligi 1023 belgidan oshsa, faqat birinchi 1023 belgini chop eting. Agar kerakli qator mavjud bo'lmasa, NO SOLUTION ni chop eting.

Dastur kodi	Dastur natijasi

s=input()
k=int(input())
if k>0:
  if len(s)*k<1023:
  print(s*k)
  else:
  k=int(1023/len(s))+1
  s=s*k
  print(s[:1023])
else:
	abcdabcd
	2
	abcdabcdabcdabcd

x=s[:int(len(s)/abs(k))]
if x*abs(k)==s:
  print(x)
else:
  print('NO SOLUTION')	

6.25-masala

Shifrni ochish

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 28%)

Eng oddiy shifrning ishlashini ko'rib chiqing. Shifrlangan xabar kichik ingliz harflari va bo'sh joy belgisidan iborat. Shifrlash har bir belgi sodir bo'ladi. Biz har bir harfga raqam beramiz: a - 1, b - 2, ..., z - 26, ' ' - 27. Bundan tashqari, xabar modulidagi raqamga ramz indeksi qo'shiladi 27 va natija. qo'shish sanoq tizimida 27 asosli (0 , 1, …, Q bosh harf bilan) berilgan. Siz dekoder yozishingiz kerak.

Ma'lumotlarni kiritish: bitta satri 1 dan 255 ta belgigacha bo'lgan kodlangan qatorni o'z ichiga oladi. Satr katta harf bilan yoziladi.

Chiqarish ma`lumotlarining bitta satrida siz berilgan satrning shifrini ochishingiz kerak, ingliz alifbosi belgilari esa kichik harflarda chiqarilishi kerak.

Dastur kodi	Dastur natijasi
a=input() b=['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z',' '] for i in range(len(a)): print(b[(int(a[i],27)-i-2)%27],end='')	L7 MO test

6.26-masala

A ^ B ning oxirgi raqami

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 21%)

A^B soni oxirgi raqamini topadigan dastur yozish talab qilinadi.

Ma'lumotlarni kiritish: bitta satrdan iborat bo'lib, ikkita butun A va B bo'sh joy bilan ajratilgan ($1 \leq A,B \leq 10000$).

Chiqarish ma`lumotlarining yagona qatorida A^B ning oxirgi raqamini chop etish kerak.

Dastur kodi	Dastur natijasi
a,b=map(int,input().split()) d=(a**b)%10 print(int(d))	4 8 6

6.27-masala

Tangalar

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 8%)

Stolda n ta tanga bor. Ulardan ba'zilari sonli tarafida turibdi, ba'zilari esa gerb. Barcha tangalar bir xil tomonli bo`lishi uchun aylantirilishi kerak bo'lgan minimal tangalar sonini aniqlang.

Ma'lumotlarni kiritish: birinchi qatorida N natural son ($1 \leq N \leq 100$) - tangalar soni mavjud. Keyingi N qatorning har biri bitta butun sonni o'z ichiga oladi - agar tanga sonli tarafi bo'lsa 1 va gerb yuqorida bo'lsa 0.

Chiqarish ma`lumotlarida aylantirilishi kerak bo'lgan tangalarning minimal sonini chiqaradi.

Dastur kodi	Dastur natijasi

a=int(input());s=0
for i in range(a):
  b=int(input())
  if b==1:
  s=s+1
d=a-s
if d>s:
  print(s)
else:
  print(d)
	Kiritish:
	4
	1
	1
	1
	0
	Chiqarish:
	1

Masalaning 2-usuli 6.43- masalada keltirilgan.

6.28-masala

Rangli yomg'ir

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 26%)

Banana Respublikasida ko'priklar bilan bog'langan ko'plab tepaliklar mavjud. Kimyo zavodida avariya yuz berdi, natijada “Zovan” eksperimental o‘g‘it bug‘lanib ketdi. Ertasi kuni rangli yomg'ir yog'di va u faqat tepaliklardan o'tdi. Ba'zi joylarda qizil tomchilar, ba'zi joylarda ko'k va boshqa joylarda yashil rangga tushib, tepaliklarning mos rangga aylanishiga olib keldi. Banan respublikasi prezidentiga bu yoqdi, lekin tepaliklar orasidagi ko‘priklarni bo‘yashni xohladi, shunda ko‘priklar ular bog‘laydigan tepaliklar bilan bir xil rangga bo‘yalgan. Afsuski, agar tepaliklar turli xil ranglarda bo'lsa, unda ko'prikni

bu tarzda bo'yash mumkin bo'lmaydi. Bunday "yomon" ko'priklarning sonini hisoblang.

Ma'lumotlarni kiritish: birinchi qatorda N ($0 < N \leq 100$) - tepaliklar soni mavjud. Keyinchalik tepaliklar orasidagi ko'priklar mavjudligini tavsiflovchi qo'shnilik matritsasi keladi (1-ko'prik mavjud, 0-yo'q). Oxirgi satr bo'sh, oxirgi qatorda tepaliklar rangini ko'rsatuvchi N raqam mavjud: 1 - qizil; 2 - ko'k; 3 - yashil.

Chiqarish ma`lumotlariga "yomon" ko'priklar sonini chiqaring.

Masalan,

7

0 1 0 0 0 1 1

1 0 1 0 0 0 0

0 1 0 0 1 1 0

0 0 0 0 0 0 0

0 0 1 0 0 1 0

1 0 1 0 1 0 0

1 0 0 0 0 0 0

1 1 1 1 1 3 3

Javob: 4

Dastur kodi	Dastur natijasi

a=int(input())
m=[];s=0
for i in range(a):
  m.append(input())
v=input()
c=list(map(int,input().strip().split()))[:a+1]
for l in range(a):
  for k in range(a):
  if m[l][k*2]=='1' and c[l]!=c[k]:
	Kiritish:
	3
	0 1 0
	1 1 1
	0 0 1
	
	1 2 1
	
	Chiqarish:

s=s+1 print(int(s/2))	1

6.29-masala

Aholini ro'yxatga olish

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 15%)

Binoda N kishi istiqomat qiladi. Bir marta ular ushbu uyning barcha aholisini ro'yxatga olishni o'tkazishga qaror qilishdi va har bir ijarachining yoshi va jinsini ko'rsatadigan ro'yxatni tuzishdi. Eng keksa erkakning sonini topish talab qilinadi.

Ma'lumotlarni kiritishning birinchi qatorda N natural son, rezidentlar soni ($N \leq 100$) mavjud. Quyidagi N qatorda barcha rezidentlar to'g'risidagi ma'lumotlar mavjud: har bir qatorda ikkita butun son mavjud: V va S - insonning yoshi va jinsi ($1 \leq V \leq 100$, S - 0 yoki 1). S=1 qiymatiga erkak jinsi, S=0 ga esa ayol jinsi mos keladi.

Chiqarish ma`lumotlarida ro'yxatdagi eng keksa odamning raqamini o'z ichiga olishi kerak. Agar bunday ijarachilar bir nechta bo'lsa, unda eng kichik raqamni chop eting. Agar rezident erkak bo'lmasa, -1 ni chop eting.

Dastur kodi	Dastur natijasi

n=int(input())
mx=0;s=-1
for i in range(n):
  x,y=map(int,input().split())
  if y==1:
  if x>mx:
  mx=x
  s=i+1
print(s)
	Kiritish
	3
	15 1
	85 1
	25 0
	Chiqarish
	2

6.30-masala

5-sinf uchun tenglama!

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 20%)

Beshinchi sinf o'quvchilari uchun tenglama 5 ta belgidan iborat. Satrning ikkinchi belgisi yoki '+' (ortiqcha) yoki '-' (minus), to'rtinchi belgisi '=' (teng). Birinchi, uchinchi va beshinchi belgilarning aniq ikkitasi 0 dan 9 gacha bo'lgan raqamlar va bittasi noma'lumni ifodalovchi x harfidir.

Bu tenglamani x uchun yechish dasturini yozish talab qilinadi.

Ma'lumotlarni kiritishda tenglamani o'z ichiga olgan bitta qatordan iborat.

Chiqarish ma`lumotlarida butun son - x qiymatini chiqaring.

Dastur kodi	Dastur natijasi

a=input()
if a[0]=='x':
  if a[1]=='+':
  b=int(a[-1])-int(a[2])
  print(b)
  else:
  b=int(a[-1])+int(a[2])
  print(b)
elif a[2]=='x':
  if a[1]=='+':
  b=int(a[-1])-int(a[0])
  print(b)
  else:
  b=int(a[0])-int(a[-1])
  print(b)
else:
  if a[1]=='+':
	x-3=5
	8

b=int(a[2])+int(a[0]) print(b) else: b=int(a[0])-int(a[2]) print(b)	

6.31-masala

Baxtli chipta - 2

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 26%)

Chipta, agar raqamlar orasidagi to'g'ri chiziq orqali ikki qismga bo'linib, ulardagi raqamlar bir xil raqamli ildizga ega bo'lsa, omadli deb ataladi. Raqamning raqamli ildizini hisoblash uchun uning raqamlari qo'shiladi, agar natija 10 dan katta yoki teng bo'lsa, raqamlar 0 dan 9 gacha bo'lgan raqam olinmaguncha yana qo'shiladi va hokazo - bu raqamli ildiz. Misol uchun, 0015420 raqamiga ega chipta omadli, chunki uni 0015 va 420 raqamlari bilan bo'laklarga bo'lib, biz bu raqamlar uchun bir xil raqamli ildizlarga egamiz.

Berilgan raqamga ega chipta omadli yoki yo'qligini aniqlaydigan dastur yozish talab qilinadi.

Ma'lumotlarni kiritishda omadli chipta raqami mavjud. Raqam noldan boshlanishi mumkin va 1 dan 100 gacha raqamni o'z ichiga oladi.

Chiqarish matn faylida agar chipta omadli bo'lsa "YES" va aks holda "YO'Q" deb chop eting.

Dastur kodi	Dastur natijasi
a=input();t=False def f(x,y): m=0 n=0	00152040 YES

```
    while int(x)>9:
        for i in range(len(x)):
            n+=int(x[i])
        x=str(n)
        n=0
    while int(y)>9:
        for l in range(len(y)):
            m+=int(y[l])
        y=str(m)
        m=0
    if x==y:
        return True
    return False
for k in range(1,len(a)):
    x=a[0:k]
    y=a[k:]
    if f(x,y):
        t=True
        break
if t==True:
    print('YES')
else:
    print('NO')
```

6.32-masala

To'rtburchak - 2

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 27%)

To'rtburchakning uchta uchining koordinatalari berilgan. To'rtinchi uchining koordinatalarini aniqlash kerak.

Ma'lumotlarni kiritish: x1 y1 x2 y2 x3 y3 formatida tasodifiy tartibda uchta to'rtburchak cho'qqilarning bo'sh joydan ajratilgan koordinatalarini o'z ichiga oladi. Barcha raqamlar mutlaq qiymatda 1000 dan oshmaydigan butun sonlardir.

Chiqarish ma`lumotlarida bo'sh joy bilan ajratilgan to'rtburchakning to'rtinchi uchining koordinatalarini chiqarish kerak.

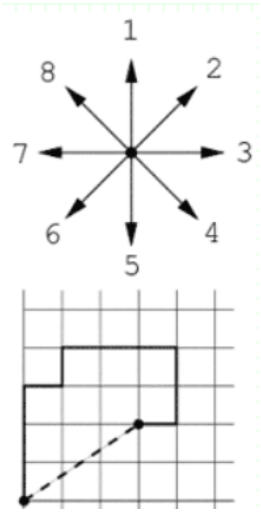
Dastur kodi	Dastur natijasi

x,y,x0,y0,x1,y1=map(int,input().split());m=[]
def f(x,y,x0,y0):
  s=pow((x-x0)**2+(y-y0)**2,1/2)
  return s
a=f(x,y,x0,y0)
b=f(x,y,x1,y1)
c=f(x1,y1,x0,y0)
m.append(f(x,y,x0,y0))
m.append(f(x,y,x1,y1))
m.append(f(x1,y1,x0,y0))
m.sort()
if a==m[0]:
  if b==m[2]:
  print(x1-x0+x,y1-y0+y)
  else:
  print(x1+x0-x,y1+y0-y)
elif b==m[0]:
  if a==m[2]:
  print(x0-x1+x,y0-y1+y)
  else:
  print(x0+x1-x,y0+y1-y)
else:
  if b==m[2]:
	1 4 8 3 7 6
	2 1

print(x-x0+x1,y-y0+y1) else: print(x+x0-x1,y+y0-y1)	

6.33-masala

Xazina



Qaroqchilar tomonidan ko'milgan xazinani topish oson: sizga faqat xarita kerak. Ma'lumki, qaroqchilar odatda kartalarni qo'lda chizishadi va harakatlar algoritmini tasvirlaydilar. Ushbu harakatlarning aksariyati sakkizta yo'nalishdan birida ma'lum miqdordagi qadamlarni bosib o'tishga to'g'ri keladi (1 - shimol, 2 - shimoli-sharqiy, 3 - sharq, 4 - janubi-sharq, 5 - janub, 6 - janubi-g'arbiy, 7 - g'arbiy, 8 - shimoli-g'arbiy) (rasmga qarang). Har qanday yo'nalishdagi qadam uzunligi 1 ga teng.

Bu yo'l bo'ylab sayohat qilish odatda atrofni ko'rishning ajoyib usuli hisoblanadi, ammo doimiy shoshqaloqlik davrida hech kimning bunga vaqti yo'q. Shuning uchun, xazina ovchilari xazina ko'milgan joyga to'g'ridan-to'g'ri borishni xohlashadi. Misol uchun, uch qadam shimolga, bir qadam sharqqa, bir qadam shimolga, uch qadam sharqqa, ikki qadam janubga va bir qadam g'arbga yurish o'rniga, taxminan 3,6 qadam yordamida to'g'ridan-to'g'ri yurish mumkin (rasmga qarang).

Siz qaroqchilarning yordamchi so`zlariga binoan xazina joylashgan joyni topishingiz kerak.

Ma'lumotlarni kiritish birinchi qatorida N raqami, ko'rsatkichlar soni ($1 \le N \le 40$) mavjud. Keyingi N qatorlar yordamchi so`zlarning o'zini o'z ichiga oladi - yo'nalish raqami (1 dan 8 gacha butun son) va qadamlar soni (1 dan 1000 gacha). Raqamlar bo'sh joy bilan ajratilgan.

Chiqarish ma`lumotlarida xazina ko‘milgan nuqtaning X va Y koordinatalarini (bo‘shliq bilan ajratilgan ikkita haqiqiy son) chop eting, OX o‘qi sharqqa, OY o‘qi esa o‘qga yo‘naltirilgan deb faraz qiling. shimol. Boshida xazina ovchisi kelib Chiqarishida turishi kerak. Koordinatalar 10^{-3} aniqlik bilan ko'rsatilishi kerak.

Dastur kodi	Dastur natijasi

a=int(input());x=0;y=0;c=[];d=[]
for i in range(a):
  w,q=map(int,input().split())
  c.append(w)
  d.append(q)
for l in range(a):
  if c[l]==1:
  y=y+d[l]
  elif c[l]==2:
  y=y+round(0.5**0.5,7)*d[l]
  x=x+round(0.5**0.5,7)*d[l]
  elif c[l]==3:
  x=x+d[l]
  elif c[l]==4:
  y=y-round(0.5**0.5,7)*d[l]
  x=x+round(0.5**0.5,7)*d[l]
  elif c[l]==5:
  y=y-d[l]
  elif c[l]==6:
  y=y-round(0.5**0.5,7)*d[l]
  x=x-round(0.5**0.5,7)*d[l]
  elif c[l]==7:
  x=x-d[l]
  else:
  y=y+round(0.5**0.5,7)*d[l]
	Kiritish
	4
	1 3
	3 0
	1 2
	5 3
	
	Chiqarish
	0 2

```
        x=x-round(0.5**0.5,7)*d[l]
if x%2==1:
    print('{x}.000 {y}.000')
else:
    print(round(x,3),round(y,3))
```

6.34-masala

Avtobus sayohati

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 14%)

Moskva shahar olimpiadasining tashkiliy qo'mitasi olimpiada ishtirokchilari uchun Moskva shahrining diqqatga sazovor joylariga sayohat uyushtirishga qaror qildi. Buning uchun balandligi 437 santimetr bo'lgan ikki qavatli avtobus buyurtma qilingan (Olimpiadada ishtirokchilar juda ko'p va ular oddiy avtobusga sig'maydi). Ekskursiya marshrutida N ta ko'prik mavjud. Hakamlar hay'ati va Olimpiada tashkiliy qo'mitasi baland ikki qavatli avtobus ulardan birining ostidan o'tmasligidan juda xavotirda. Ular ko'priklarning har birining aniq balandligini aniqlashga muvaffaq bo'lishdi. Avtobus ko'prik ostidan o'tishi mumkin, agar ko'prik balandligi avtobus balandligidan oshib ketgan bo'lsa.

Tashkilotchilarga ekskursiya xavfsiz tugashini aniqlashga yordam bering, agar bo'lmasa, voqea qaerda sodir bo'lishini aniqlashga yordam bering.

Ma'lumotlarni kiritish: birinchi qatorida N ($1 \leq N \leq 1000$) raqami mavjud. Ikkinchi qatorda 10000 dan oshmaydigan N natural sonlar, bo'shliqlar bilan ajratilgan - ko'priklarning balandligi avtobus yo'lida sodir bo'lish tartibida santimetrda.

Chiqarish ma`lumotlarining satrida, agar tur muvaffaqiyatli yakunlansa, "No Crash" iborasini chop etishingiz kerak. Agar baxtsiz hodisa ro'y bersa, siz "Crash k" xabarini ko'rsatishingiz kerak, bu erda k - avariya sodir bo'ladigan ko'prikning raqami.

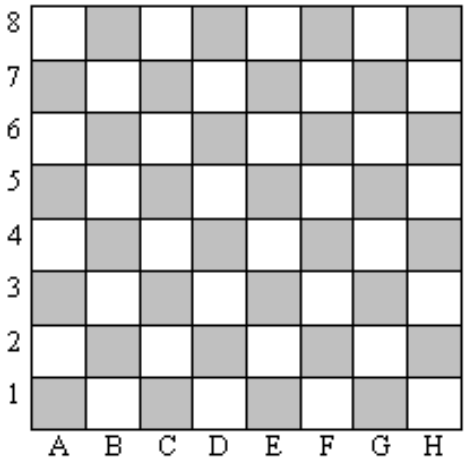
Dastur kodi	Dastur natijasi

a=int(input());bor=False;
b=list(map(int,input().strip().split()))[:a]
for i in range(a):
  if b[i]>437:
  bor=True
  elif b[i]<=437:
  bor=False
  break
if bor==True:
  print('No crash')
else:
  print('Crash', i+1)
	Kiritish
	2
	570 365
	Chiqarish
	Crash 2

6.35-masala

Shaxmat doskasi

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 15%)



Shaxmat kataklari oq yoki qora bo'lishi ma'lum. Har bir katak harf va raqamli kooordinatalari orqali aniqlanadi. Gorizontal A dan H gacha, vetikal 1 dan 8 gacha belgilanadi. A1 katak qora rangga ega. Berilgan koordinatalariga qarab katak rangini aniqlovchi dastur tuzing.

Kiritish: INPUT.TXT da katak koordinatasi – harf va raqam(bo'sh joysiz) yozilgan.

Chiqarish: OUTPUT.TXTda katak rangini "OQ" yoki "QORA" deb chiqarish kk.

Misollar:

№	INPUT.TXT	OUTPUT.TXT
1	C3	QORA
2	G8	OQ

Dastur kodi	Dastur natijasi

s=input()
if s[0]=='A':
  if int(s[1])%2==0:
  print('OQ')
  else:
  print('QORA')
elif s[0]=='B':
  if int(s[1])%2==1:
  print('OQ')
  else:
  print('QORA')
elif s[0]=='C':
  if int(s[1])%2==0:
  print('OQ')
  else:
  print('QORA')
elif s[0]=='D':
  if int(s[1])%2==1:
  print('OQ')
  else:
  print('QORA')
elif s[0]=='E':
  if int(s[1])%2==0:
	E6
	OQ

```
        print('OQ')
    else:
        print('QORA')
elif s[0]=='F':
    if int(s[1])%2==1:
        print('OQ')
    else:
        print('QORA')
elif s[0]=='G':
    if int(s[1])%2==0:
        print('OQ')
    else:
        print('QORA')
elif s[0]=='H':
    if int(s[1])%2==1:
        print('OQ')
    else:
        print('QORA')
else:
    print('ma`lumot xato kiritilgan')
```

6.36-masala

Vagonlar

(Vaqt: 1 sek. Xotira: 16 MB Qiyinchilik: 23%)

Har kuni poezd stantsiyasi dispetcheri ko'plab poezdlardagi vagonlarni ma'lum bir tartibda harakatlanishi uchun ularni qayta joylashtirishi kerak. Buning uchun dispetcher stansiyaga kelgan poyezdni tasodifiy joylarda ajratib qo‘yishi va hosil bo‘lgan bog‘lovchilarni bir yoki bir nechta vagonlardan ixtiyoriy tartibda o‘zgartirishi mumkin. Bitta bog'lovchidagi avtomobillarning tartibini o'zgartirib bo'lmaydi, shuningdek, bog'lovchidagi oxirgi mashina birinchi bo'lishi uchun butun bog'lovchini aylantirib bo'lmaydi.

Dispetcher sizdan poyezddagi vagonlarni kerakli tartibda joylashtirish uchun ajratilishi kerak bo'lgan vagonlar orasidagi minimal ulanishlar sonini aniqlashda yordam so'raydi.

Ma'lumotlarni kiritish: birinchi qatorida N ($1 \le N \le 100$) butun son mavjud. Ikkinchi qatorda 1 dan N gacha bo'lgan natural sonlarning almashtirilishi (ya'ni, 1 dan N gacha bo'lgan barcha natural sonlar ma'lum tartibda). Raqamlar bitta bo'sh joy bilan ajratilgan. Bu almashtirish stansiyaga kelayotgan poyezddagi vagonlar sonini belgilaydi. Stansiyadan ketayotgan poyezdda vagonlar o‘z raqamlari tartibida harakatlanishi talab qilinadi.

Chiqarish ma`lumotlariga ushbu poyezdda ajratilishi kerak bo'lgan vagonlar orasidagi ulanishlarning minimal soniga teng bitta butun sonni yozishi kerak, shunda ularni tartibda qayta joylashtirish mumkin.

Dastur kodi	Dastur natijasi

a=int(input());s=0
b=list(map(int,input().strip().split()))[:a]
for i in range(a):
  if i<a-1:
  if b[i]==b[i+1]-1:
	Kiritish
	4
	3 1 2 4
	Chiqarish
	2

s=s+1 c=a-1-s print(int(c))	

6.37-masala

Arifmetik progressiya

(Vaqt: 1 sek. Hajmi: 16 Mb. Qiyinligi: 15%)

Arifmetik progressiyaning 1- va 2-elementlari berilgan. Shu progressiyaning elementini tartib raqamiga ko'ra aniqlovchi dastur tuzing.

Kiritish: INPUT.TXT da 3 ta son berilgan – progressiyaning

1-elementi A1 ($1 \le A1 \le 1000$),

2-elementi A2 ($1 \le A2 \le 1000$)

so'ralgan element tartib raqami N ($1 \le N \le 1000$).

Chiqarish: OUTPUT.TXT da arifmetik progressiyaning N-elementi chiqarildi.

Misol:

№	INPUT.TXT	OUTPUT.TXT
1	1 5 3	9

Dastur kodi	Dastur natijasi
f=open('input.txt','r') fo=open('output.txt','w') l=list(map(int,f.read().split())) a1=l[0];d=l[1]-l[0];n=l[2] fo.write(str(a1+(n-1)*d)) f.close() fo.close()	input.txt 1 5 3 output.txt 9

6.38-masala

Aylanalar

(Vaqt: 1 sek. Hajmi: 16 Mb. Qiyinligi: 16%)

Bir kuni Internetda quyidagi rebus paydo bo'ldi:

157892 = 3

203516 = 2

409578 = 4

236271 = ?

Uni hech kim ishlay olmadi. Keyinchalik aniq bo'lishicha, "=" dan keyingi sonlar chap tomonda joylashgan sonni yozishda ishlatiluvchi "aylanacha"lar soni ekan.

Masalan: 8 da 2 ta. Sonni yozishda ishlatiluvchi "aylana"larni topish dasturini tuzing.

Kiritiluvchi qiymatlar: INPUT.TXT da N (0<=N<=10100) kiritilgan.

Chiqariluvchi qiymatlar: OUTPUT.TXT da shu N uchun "aylanacha"lar soni chiqarilsin.

Misollar:

№	INPUT.TXT	OUTPUT.TXT
1	157892	3
2	203516	2
3	409578	4
4	236271	1

Dastur kodi	Dastur natijasi
f=open('input.txt','r') ff=open('output.txt','w') n=f.readline() s=0 for i in range(len(n)):	input.txt 157892 output.txt 3

if n[i]=='0' or n[i]=='6' or n[i]=='9':
s=s+1
if n[i]=='8':
s=s+2
 	
ff.write(str(s))
ff.close()	

6.39-masala

4 xonali palindrom son

(Vaqt: 1 sek. Hajmi: 16 Mb. Qiyinligi: 10%)

4 xonali N natural son palindromligini, ya'ni o'ngdan chapga va chapdan o'ngga bir xilda o'qilishini tekshiring.

Kiritiluvchi qiymatlar: INPUT.TXT da N son (1000<=N<=9999) kiritilgan.

Chiqariluvchi qiymatlar: OUTPUT.TXT da N palindrom bo'lsa, "HA" so'zini, aks holda "YO'Q" so'zini chiqaring.

Misollar:

№	INPUT.TXT	OUTPUT.TXT
1	6116	YES
2	1231	NO

Dastur kodi	Dastur natijasi
f=open('input.txt','r') ff=open('output.txt','w') n=int(f.readline()) s=0 m=n while n>0:	input.txt 7557 output.txt YES

k=n%10
n=n//10
s=s*10+k
if m==s:
ff.write('YES')
else:
ff.write('NO')
 	
ff.close()	

6.40-masala

Nollar

(Vaqt: 1 sek. Hajmi: 16 Mb. Qiyinligi: 16%)

0 va 1 lardan iborat ketma-ketlikdagi uzluksiz kelgan 0lar ketma-ketligining eng uzun 0 lar soni topilsin.

Kiritish:INPUT.TXT da 1 va 0 lar ketma-ketligi probelsiz yozilgan. Raqamlar soni 100 tadan ortiq emas.

Chiqarish:OUTPUT.TXT da eng uzun 0lar zanjirining uzunligi chiqarilsin.

Misol:

№	INPUT.TXT	OUTPUT.TXT
1	00101110000110	4

Dastur kodi	Dastur natijasi
f=open('input.txt','r') ff=open('output.txt','w') n=f.readline() s=0 k=0 for i in range(len(n)):	input.txt 11100000000011101101011011111000000 output.txt 9

if n[i]=='0':
  s=s+1
else:
  if s>k:
  k=s
  s=0
ff.write(str(k))
ff.close()	

6.41-masala

Issiq havo

(Vaqt: 1 sek. Hajmi: 16 Mb. Qiyinligi: 17%)

Har bir kunida harorat 0ºC dan yuqori bo'lgan davrlar ichida eng uzun shunday kunlar zanjiridagi kunlar sonini chiqaring.

Kiritish: INPUT.TXT da 1-satrda N (1<=N<=100), keying satrda probel orqali N ta butun son yozilgan bo'lib, har bir son mos kunning haroratidir. Harorat – butun son bo'lib, -50 dan 50 gacha bo'lgan oraliqda joylashadi.

Chiqarish:OUTPUT.TXT da 0ºC dan eng yuqori haroratli kunlar zanjirining uzunligi chiqarilsin. Agar har kuni harorat nomusbat (0 dan katta emas) bo'lgan bo'lsa, natija sifatida 0 ni chiqaring.

Misollar:

№	INPUT.TXT	OUTPUT.TXT
1	6 -20 30 -40 50 10 -10	2
2	8 10 20 30 1 -10 1 2 3	4
3	5 -10 0 -10 0 -10	0

Dastur kodi	Dastur natijasi

f=open('input.txt','r')
ff=open('output.txt','w')
n=f.readline()
m=f.readline()
z=m.split(' ')
s=0
k=0
for i in range(len(z)):
  if int(z[i])>0:
  s=s+1
  else:
  if s>k:
  k=s
  s=0
 	
ff.write(str(k))
ff.close()
	input.txt
	5
	-4 3 15 20 -10
	output.txt
	3

6.42-masala

Ayniyat

(Vaqt: 1 sek. Hajmi: 16 Mb. Qiyinligi: 26%)

Siz Humoyunning uy vazifasini tekshirishingiz kerak, u qandaydir tenglik yozadi. Masalan: «2+3=5» to`g`ri, «23*7=421» bu noto`g`ri lekin korrekt. Ketma-ketlik korrekt bo`ladi: son, amal(+,-,*,/), son, tenglik belgisi, son ko`rinishda bo`lsa, aks holda nokorrekt, ya`ni son yetmasa yoki notanish belgi uchrasa. Masalan: «2*=3», «173» va «2+2=a» nokorrekt.

Kiritish:

INPUT.TXT faylida arifmetik ifoda beriladi. Barcha sonlar butun va absolut qiymati 30000 dan oshmaydi.Arifmetik ifodaning uzunligi 100 ta belgidan oshmaydi.

Chiqarish:

OUTPUT.TXT faylida to`g`ri bo`lsa «YES», korrekt lekin noto`g`ri bo`lsa «NO» va yozuvida xatolik bo`lsa «ERROR» chiqarsin.

Misol:

№	INPUT.TXT	OUTPUT.TXT
1	2+3=5	YES
2	3*7=20	NO
3	two plus three is five	ERROR

Dastur kodi	Dastur natijasi

f=open('input.txt','r')
ff=open('output.txt','w')
n=f.readline()
for i in range(len(n)):
  if (ord(n[i])==42 or ord(n[i])==43 or ord(n[i])==45 or ord(n[i])==61 or ord(n[i])==47):
  x=1
  elif ord(n[i])>47 and ord(n[i])<58:
  x=1
  else:
  x=0
  break
if x==1:
  for i in range(len(n)):
	input.txt
	8+2=10
	output.txt
	YES

```
        if n[i]=='/' and n[i+1]=='0':
            x=2


if x==1:
    z=n.split('=')
    k=eval(z[0])
    s=int(z[1])
    if k==s:
        ff.write('YES')
    else:
        ff.write('NO')
elif x==2:
   ff.write('NO')
else:
   ff.write('ERROR')


ff.close()
```

6.43-masala

Tangalar

(Vaqt: 1 sek. Hajmi: 16 Mb. Qiyinligi: 8%)

Stol ustida n ta tanga turibdi. Ularning ba’zilari so’mlik tarafi bilan, qolganlari gerb tarafi bilan yotibdi. Hamma tangalar bir xil tarafi bilan yotishi uchun aylantirib qo’yish lozim bo’lgan eng kam tangalar sonini aniqlang.

Kiritiluvchi qiymatlar: INPUT.TXT ning 1-satrida N natural son(1<=N<=100) – tangalar soni kiritilgan. Qolgan satrlarda 1 tadan butun son kirirtilgan– 1, agar tanga so’m tarafi bilan yotsa; 0 – agar gerb tarafi bilan yotsa

Chiqariluvchi qiymatlar: OUTPUT.TXT da aylantirish zarur bo’lgan minimal tangalar sonini chiqaring.

Misol:

№	INPUT.TXT	OUTPUT.TXT
1	5 1 0 1 1 0	2

Dastur kodi	Dastur natijasi

f=open('input.txt','r')
ff=open('output.txt','w')
n=int(f.readline())
k=0
s=0
for i in range(n):
  m=int(f.readline())
  if m==1:
  k=k+1
  else:
  s=s+1
if k>s:
  ff.write(str(s))
else:
  ff.write(str(k))
 	
ff.close()
	input.txt
	6
	1
	0
	1
	1
	0
	0
	output.txt
	3

6.44-masala

Kabisa yili

Taqvimga deyarli har to'rt yilda bir qo'shimcha kun 29 fevral sifatida qo'shiladi va bu kun kabisa kuni deb ataladi. U taqvimni to'g'rilaydi, chunki bizning sayyoramiz Quyosh atrofida aylanish uchun taxminan 365,25 kun davom etadi. Kabisa yili kabisa kunini o'z ichiga oladi.

Grigoriy taqvimida kabisa yillarini aniqlash uchun uchta shart qo'llaniladi:
Yilni teng ravishda 4 ga bo'lish mumkin bo`lsa, bu kabisa yili, agar:
Yilni teng ravishda 100 ga bo'lish mumkin bo`lsa, bu kabisa yili EMAS, agar:
Yil ham 400 ga bo`linsa, bu kabisa yilidir.

Demak, Grigoriy kalendarida 2000 va 2400 yillar kabisa yillari, 1800, 1900, 2100, 2200, 2300 va 2500 kabisa yillar EMAS. Manba
Vazifa
Bir yil berilgan, bu kabisa yili yoki yo'qligini aniqlang. Agar bu kabisa yili bo'lsa, mantiqiy rostni qaytaring, aks holda False qiymatini qaytaring.
E'tibor bering, taqdim etilgan kod stendlari STDIN dan o'qiydi va argumentlarni is_leap funktsiyasiga uzatadi. Faqat is_leap funktsiyasini bajarish kerak.

Dastur kodi	Dastur natijasi

def is_leap(year):
  leap=False
  leap=True if year%100==0 and year%400==0 else
True if year%100!=0 and year%4==0 else False
  return print(leap)
year=int(input())
is_leap(year)
	2016
	True

6.45-masala

Sizga ikkita to'plam beriladi A va B . Sizning vazifangiz A to'plam B to'plamning qism to'plami ekanligini aniqlashdir.

Agar A to'plam B to'plamning qism to'plami bo'lsa, "True" ni chop eting.

Agar A to‘plam B to‘plamning qism to‘plami bo‘lmasa, False chop eting.

Kiritish formati: Birinchi qatorda test holatlari T soni bo'ladi. Har bir test ishining birinchi qatori A to'plamdagi elementlar sonini o'z ichiga oladi. Har bir test ishining ikkinchi qatorida A to'plamning bo'sh joydan ajratilgan elementlari ko`rsatiladi. Har bir test ishining uchinchi qatori B to'plamdagi elementlar sonini o'z ichiga oladi. Har bir test ishining to'rtinchi qatorida B to'plamning bo'sh joydan ajratilgan elementlari ko`rsatiladi.

Shartlar:

0<T<21

0<Har bir to`plamning elementlari soni<1001

Chiqarish formati: Har bir test ishi uchun alohida satrlarda True yoki False chiqishi kerak.

Kiritish namuna:

3

1—holat:5

1 2 3 5 6

9

9 8 5 6 3 2 1 4 7

2—holat: 1

2

5

3 6 5 4 1

3—holat: 7

1 2 3 5 6 8 9

3

9 8 2

Chiqish namuna:

True

False

False

Dastur kodi	Dastur natijasi
for _ in range(int(input())): n=int(input()) a=set(input().split()) m=int(input()) b=set(input().split()) print(a.issubset(b))	2 4 1 2 5 6 5 3 2 1 5 6 True 2 1 3 5 1 4 5 8 9 False

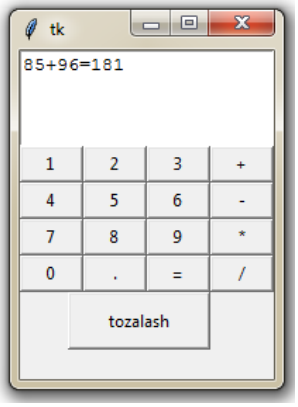
6.46-masala

Pythonda kalkulator yaratish uchun dastur yozing.

Tkinter —Pythonda grafik interfeys yaratish kutubxonasi bo`lib, vidjetlar uning asosi bo`lib hisoblanadi. Har bir vidjet sinf sifatida e`lon qilinadi. Keng qo`llaniladigan vidjetlar bilan tanishtirib o`tamiz:

Vidjet sinfi	Izoh
Label	Yangi oynaga matnni ko`rsatish yoki rasm qo`yish uchun ishlatiladi.
Button	Tugma yaratadi. Uni nomlash va bosilganda ma`lum amalni bajarishi mumkin.
Entry	1 qator matn kiritish vidjeti
Text	Ko`p qatorli matn kiritish vidjeti
Frame	Vidjetlarni guruhlash yoki vidjetlar orasiga masofa qo`yish uchun to`g`ri to`rtburchakdan iborat maydon

Dastur kodi	Dastur natijasi

from tkinter import *
 	
def qushish():
  s = text.get(1.0, END)
  k=float(len(s))
  text.insert(k,'+')
 	
def ayrish():
  s = text.get(1.0, END)
  k=float(len(s))
  text.insert(k,'-')
 	
def kopaytrish1():
  s = text.get(1.0, END)
  k=float(len(s))
  text.insert(k,'*')
	

```
def bolish1():
    s = text.get(1.0, END)
    k=float(len(s))
    text.insert(k,'/')
def b1():
    s = text.get(1.0, END)
    k=float(len(s))
    text.insert(k,'1')
def b2():
    s = text.get(1.0, END)
    k=float(len(s))
    text.insert(k,'2')
def b3():
    s = text.get(1.0, END)
    k=float(len(s))
    text.insert(k,'3')
def b4():
    s = text.get(1.0, END)
    k=float(len(s))
    text.insert(k,'4')
def b5():
    s = text.get(1.0, END)
    k=float(len(s))
    text.insert(k,'5')
def b6():
    s = text.get(1.0, END)
    k=float(len(s))
    text.insert(k,'6')
def b7():
    s = text.get(1.0, END)
```

```
    k=float(len(s))
    text.insert(k,'7')
def b8():
    s = text.get(1.0, END)
    k=float(len(s))
    text.insert(k,'8')
def b9():
    s = text.get(1.0, END)
    k=float(len(s))
    text.insert(k,'9')
def b0():
    s = text.get(1.0, END)
    k=float(len(s))
    text.insert(k,'0')
def nuqta():
    s = text.get(1.0, END)
    k=float(len(s))
    text.insert(k,'.')

def hisoblash():
     s = text.get(1.0,END)
     k=float(len(s)-1)
     x=eval(s)
     text.insert(k,'=')
     k=float(len(s)-1)
     text.insert(k,x)



def tozalash():
    text.delete(1.0, END)
```

```
root = Tk()

text = Text(width=22, height=4)
text.pack()

frame = Frame()
frame.pack()
Button(frame,
text="1",command=b1,width=5,height=1).grid(row=0,
column=0,sticky=E)
Button(frame,
text="2",command=b2,width=5,height=1).grid(row=0,
column=1,sticky=E)
Button(frame,
text="3",command=b3,width=5,height=1).grid(row=0,
column=2,sticky=E)
Button(frame,
text="4",command=b4,width=5,height=1).grid(row=1,
column=0,sticky=E)
Button(frame,
text="5",command=b5,width=5,height=1).grid(row=1,
column=1,sticky=E)
Button(frame,
text="6",command=b6,width=5,height=1).grid(row=1,
column=2,sticky=E)
Button(frame,
text="7",command=b7,width=5,height=1).grid(row=2,
column=0,sticky=E)
```

```
Button(frame,
text="8",command=b8,width=5,height=1).grid(row=2,
column=1,sticky=E)
Button(frame,
text="9",command=b9,width=5,height=1).grid(row=2,
column=2,sticky=E)
Button(frame,
text="+",command=qushish,width=5,height=1).grid(r
ow=0,column=3,sticky=E)
Button(frame, text="-
",command=ayrish,width=5,height=1).grid(row=1,col
umn=3,sticky=E)
Button(frame,
text="*",command=kopaytrish1,width=5,height=1).gri
d(row=2,column=3,sticky=E)
Button(frame,
text="/",command=bolish1,width=5,height=1).grid(ro
w=3,column=3,sticky=E)
Button(frame,
text="=",command=hisoblash,width=5,height=1).grid(
row=3,column=2,sticky=E)
Button(frame,
text="0",command=b0,width=5,height=1).grid(row=3,
column=0,sticky=E)
Button(frame,
text=".",command=nuqta,width=5,height=1).grid(row
=3,column=1,sticky=E)
Button(frame,
text="tozalash",command=tozalash,width=13,height=
2).grid(row=5,sticky=E,columnspan=3)
```

```
label = Label()
label.pack()


root.mainloop()
```

XULOSA

Xulosa o`rnida shuni aytish mumkinki, Python dasturlash tilida masalalar va ularning yechimlari o`quv qo`llanmasi kelajakda umumta`lim maktablari o`quvchilari va OTM talabalari uchun dasturlash tilini o`rganishda “ko`makdosh” vazifasini o`taydi.

Hozirgi zamonda dasturchi eng zamonaviy, eng hamyonbop va ilg`or kasblardan biri. Dasturchilik kasblariga Web-masalachi, Sistema adminstratori, IT operator, Flash animator va hokazolar kiradi.

Yaxshi dasturchi bo`lish uchun nima talab qilinadi deb o`ylaysiz? Buyuk dasturchilarimiz fikricha, dasturlashni o`rganishning eng yaxshi usuli qiyinroq dasturlar kodlarini o`rganib, ishlatib ko`rishdir.

Ushbu “Python dasturlash tilida masalalar va uning yechimlari to`plami o`quv qo`llanmasi” ham o`quvchilar va talabalarga dasturlash sirlarini o`rganishda hamroh bo`ladi.

O`quv qo`llanma tadbiq qilingandan so`ng o`rganuvchilarning dastur tuzish ko`nikmasi shakllanib, mustaqil dasturchi bo`la olishiga zamin yaratiladi.

FOYDALANILADIGAN ADABIYOTLAR VA INTERNET RESURLARI RO‘YXATI:

1. Mirziyoyev Sh.M. “Yangi O`zbekiston taraqqiyot strategiyasi”, Toshkent-2022, 245-bet.
2. Васильев А.Н., “Python на примерах. Практический курс по программированию”, Наука и техника, Санкт-Петербург, 2016, 235-243стр.
3. Gabor Szabo, “1000 Python Examples”, 2020, 140-165 pages
4. Sh.A. Mengliyev, O.A. Abdug‘aniev, S.Q. Shonazarov, D. Sh. To‘rayev “Python dasturlash tili”, Termiz-2021
5. https://metanit.com/python
6. https://pythonworld.ru/tipy-dannyx-v-python/slovari-dict-funkcii-i-metody-slovarej.html
7. https://ai.mohirdev.uz/
8. https://www.w3resource.com/
9. https://www.sololearn.com/
10. https://leetcode.com/
11. https://www.hackerrank.com/
12. www.tiobe.com/tiobe-index//
13. acmp.ru