

**O'ZBEKISTON RESPUBLIKASI OLIY TA'LIM, FAN VA  
INNOVATSIYALAR VAZIRLIGI**

**BUXORO DAVLAT UNIVERSITETI**

**M.A.BOBOJONOVA**

**PYTHON DASTURLASH TILI**



**O`QUV QO`LLANMA**

5130200 – Amaliy matematika

60610200-Axborot tizimlari va texnologiyalari (tarmoqlar  
va sohalar bo`yicha)

**“Durdona” nashriyoti  
Buxoro – 2023**

**UO'K 004.43(075.8)**

**32.973-018.1ya73**

**B 79**

**Bobojonova, M.A.**

**Python dasturlash tili [Matn] : o`quv qo`llanma / M.A. Bobojonova .-Buxoro: Sadriddin Salim Buxoriy, 2023.-108 b.**

**KBK 32.973-018.1ya73**

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`zbek tilidagi kitoblarimiz sanoqli. Ularda axborot juda kam bo`lib, sayoz yozilgan. “Python dasturlash tili” o`quv qo`llanmasi 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”, 60610200-Axborot tizimlari va texnologiyalari (tarmoqlar va sohalar bo`yicha) ta`lim yo`nalishining “Python dasturlash tili” fanidan amaliyot darslarida foydalanish uchun tavsiya etiladi.

#### **Taqrizchilar:**

**D.D.Axmedov** - Raqamli texnologiyalar va sun`iy intellektni rivojlantirish ilmiy-tadqiqot instituti katta ilmiy xodimi, texnika fanlari bo`yicha falsafa doktori

**I.I.Bakayev** - Buxoro davlat universiteti "Axborot tizimlari va raqamli texnologiyalar" kafedراس dotsenti, PhD.

**O`quv qo`llanma Buxoro davlat universitetining 2023-yil 27-iyundagi 354-sonli buyrug`iga asosan nashr etishga ruxsat berilgan. Ro`yxatga olish raqami 354-35.**

**ISBN 978-9910-9776-3-3**

## MUNDARIJA

Kirish .....	4
1-mavzu. Python dasturlash tili va uning dasturlash muhiti.....	7
2-mavzu. Python dasturlash tilida o'zgaruvchilar .....	12
3-mavzu. Pythonda ma'lumot turlari .....	15
4-mavzu. Pythonda operatorlar va ular bilan ishlash.....	18
5-mavzu. Pythonda tarmoqlanuvchi operatorlar.....	21
6-mavzu. Pythonda takrorlanish operatorlari.....	25
7-mavzu. Pythonda satrlar.....	29
8-mavzu. Pythonda satrlar (string funksiyalari) .....	32
9-mavzu. Pythonda ro'yxatlar va to'plamlar (list ma'lumot turi) .....	35
10-mavzu. Pythonda ro'yxatlar va to'plamlar (set ma'lumot turi).....	37
11-mavzu. Pythonda massivlar (bir o'lchovli massivlar) .....	39
12-mavzu. Pythonda massivlar (ikki o'lchovli massivlar) .....	42
13-mavzu. Pythonda funksiyalar.....	45
14-mavzu. Pythonda funksiyalar. Rekursiv funksiyalar.....	49
15-mavzu. Pythonda fayllar bilan ishlash.....	54
16-mavzu. Pythonda fayllar bilan ishlash.....	59
17-mavzu. Python standart kutubxonasi .....	65
18-mavzu. Python standart kutubxonasi .....	68
19-mavzu. Pythonda modullar (math, data time modullari).....	70
20-mavzu. Pythonda modullar (random moduli).....	74
21-mavzu. Pythonda sinflar va obyektlar .....	79
22-mavzu. Pythonda sinflar va obyektlar .....	84
23-mavzu. Voris tushunchasi .....	87
24-mavzu. Voris tushunchasi (ko'p merosxo'rlik) .....	90
25-26-mavzular. Grafik modular bilan ishlash.....	92
27-mavzu. Turtle moduli.....	94
28-mavzu. Turtle moduli.....	97
29-mavzu. Tkinter moduli (python tkinter basic) .....	100
30-mavzu. Tkinter moduli (python tkinter widgets).....	102
Xulosa.....	104
Foydalanilgan adabiyotlar va internet resurslari ro'yxati .....	105

## KIRISH

Zamonaviy dunyoda, hayotni axborot texnologiyalarisiz tasavvur qilishning iloji yo'q. Ular bizning hayotimizga qat'iy kirishdi, axborot texnologiyalari insoniyat hayotining barcha sohalarida qo'llaniladi, ayniqsa ikki tomonlama muhim rol o'ynaydi. Axborot texnologiyalari insoniyatning barcha to'plangan tajribasini amaliy foydalanish uchun mos formatlangan shaklda aks ettiradi. Shuningdek, u ijtimoiy jarayonlarni amalga oshirish uchun ilmiy bilim va materialistik tajribani jamlaydi, shu bilan birga mehnat, vaqt, energiya va moddiy xarajatlarni tejaydi.

"Bugun O'zbekiston barcha sohalarida o'zini namoyon qilmoqda. Axborot texnologiyalariga o'z vaqtida e'tibor qaratishimiz yaxshi natijalar bermoqda. Bitta dasturiy mahsulot davlatga korruptsiya, byurokratiyani yo'q qilish va odamlar uchun qulayliklar yaratishda katta yordamdir. Yodingizda bo'lsin, sizning ishingiz juda muhim", — dedi Shavkat Mirziyoyev.











O'zbekiston Respublikasi Prezidentining 2021 yil 17 fevraldagi PQ-4996-sonli "Sun'iy intellekt texnologiyalarini jadal joriy etish uchun shart-sharoitlar yaratish chora-tadbirlari to'g'risida"gi qarori, 2021 yil 26 avgustdagi PQ-5234-sonli "Sun'iy intellekt texnologiyalarini qo'llash bo'yicha maxsus rejimni joriy qilish chora-tadbirlari to'g'risida"gi qarori, 2020 yil 5 oktabrdagi PF-6079-sonli "«Raqamli O'zbekiston — 2030» strategiyasini tasdiqlash va uni samarali amalga oshirish chora-tadbirlari to'g'risida" farmoni yurtimizda Axborotlashtirish sohasiga bo'lgan katta e'tiborni ko'rsatadi.

O'quv qo'llanmaning asosini Python dasturlash tili bo'lgani uchun ushbu tilning tarixi hamda hozirgi reytingi haqida qisqacha to'xtalib o'tamiz.

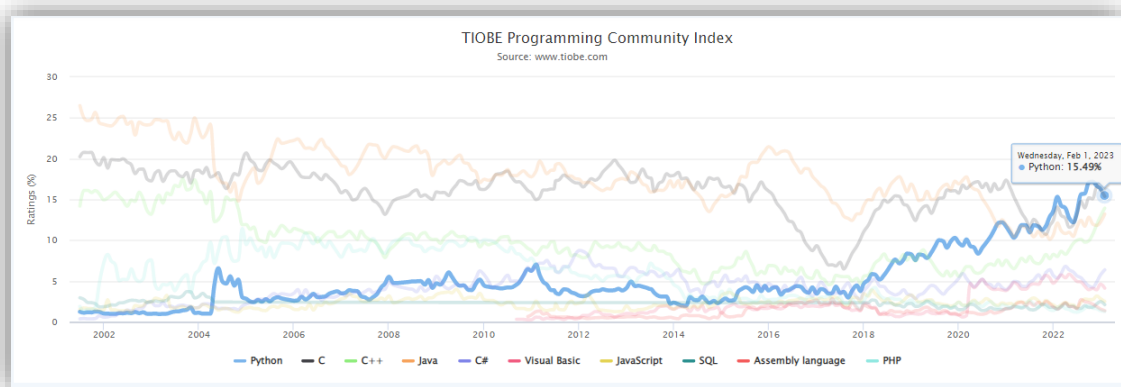
Bilamizki, dasturlash tilining mashhurligini ishlab chiquvchilar orasida eng ommabop resurs - Stack Overflowdagi teglar sonining dinamikasi bilan kuzatish mumkin. Shunday qilib, grafikaga ko'ra, Pythonning o'sishi 2010 yilda boshlangan va 2015 yilda tezlashdi.

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. 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.

Feb 2023	Feb 2022	Change	Programming Language	Ratings	Change
1	1		 Python	15.49%	+0.16%
2	2		 C	15.39%	+1.31%
3	4	▲	 C++	13.94%	+5.93%
4	3	▼	 Java	13.21%	+1.07%
5	5		 C#	6.38%	+1.01%
6	6		 Visual Basic	4.14%	-1.09%
7	7		 JavaScript	2.52%	+0.70%
8	10	▲	 SQL	2.12%	+0.58%
9	9		 Assembly language	1.38%	-0.21%
10	8	▼	 PHP	1.29%	-0.49%

1.1-rasm. TIOBE 2023 –yil fevral oyidagi natijalari



1.2-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.

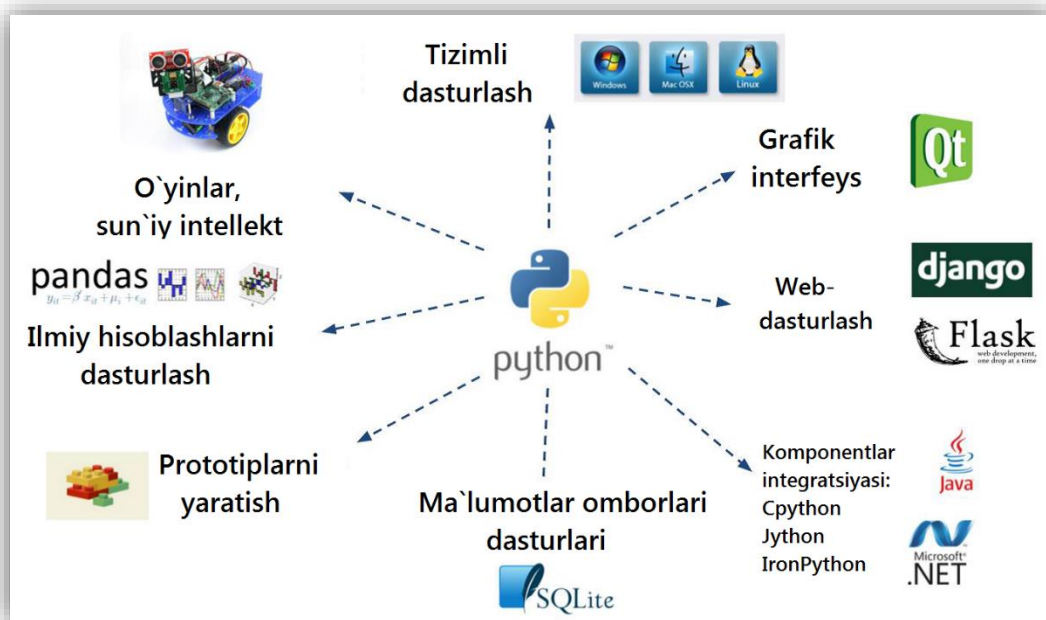
Ushbu o'quv qo'llanma Oliy o'quv yurtlaridagi o'qitish effektivligini oshirishga, Python tilini kengroq qo'llashga xizmat qiladi degan umiddamiz.

“Python dasturlash tili” o'quv qo'llanmasi kirish, 30 ta mavzu, mundarija hamda foydanilgan adabiyotlar va internet resurlari ro'yxatidan iborat. O'quv qo'llanmada dasturlar sharti, kerakli formula va chizmalar, yechimi va natijalari ko'rinishida berilgan.

O'quv qo'llanmaning umumiy hajmi 106 betdan iborat bo'lib, 80 dan ortiq masala yechimlari berilgan.

# 1-MAVZU. PYTHON DASTURLASH TILI VA UNING DASTURLASH MUHITI

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.

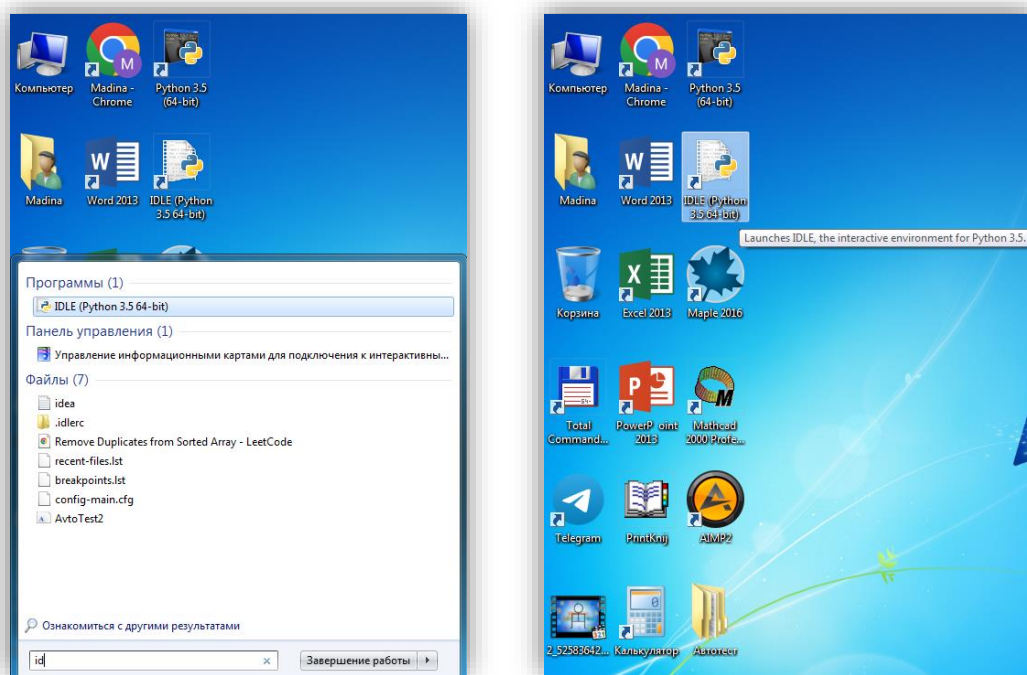
Python dasturlarini ishga tushirish uchun sizga Python tarjimoni (virtual mashina) kerak. Ushbu dastur operatsion tizimning barcha

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.

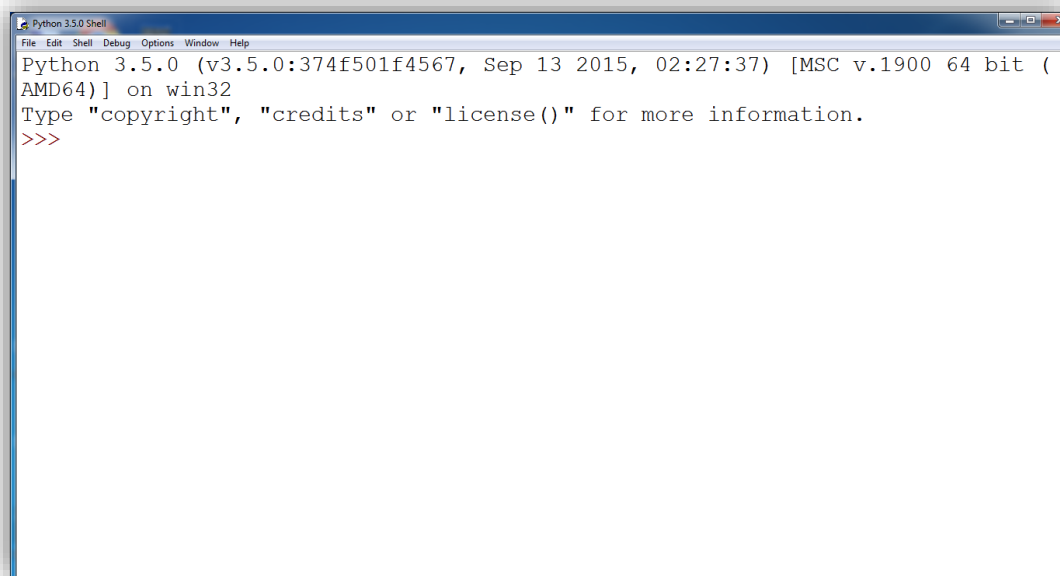
IDLE ni ishga tushirish tartibi:



1.4-1.5-rasmlar. IDLE Python dasturini ishga tushirish

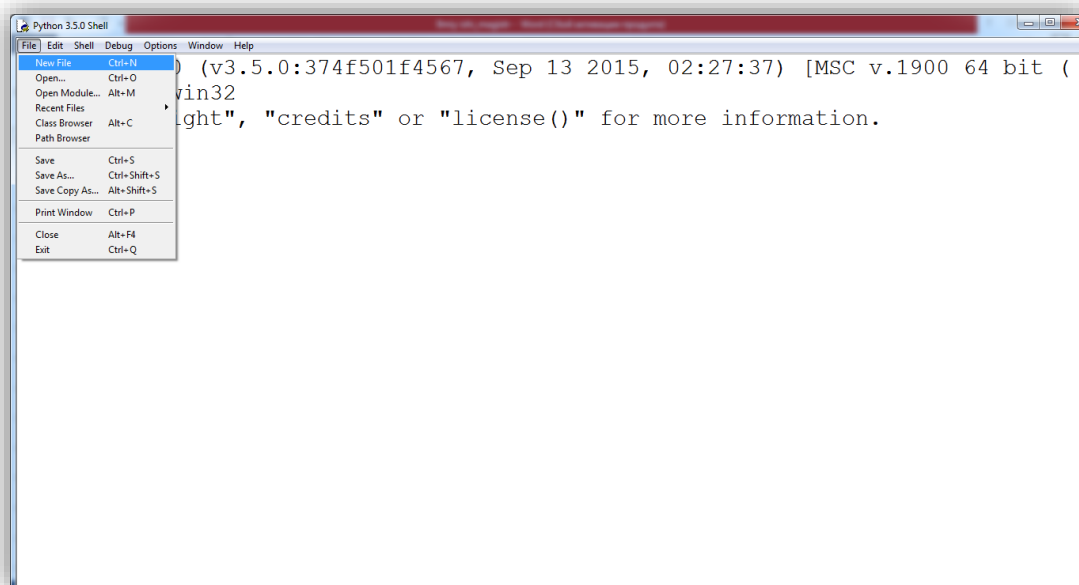
IDLE ni ishga tushirgandan so'ng quyidagi oyna paydo bo'ladi:





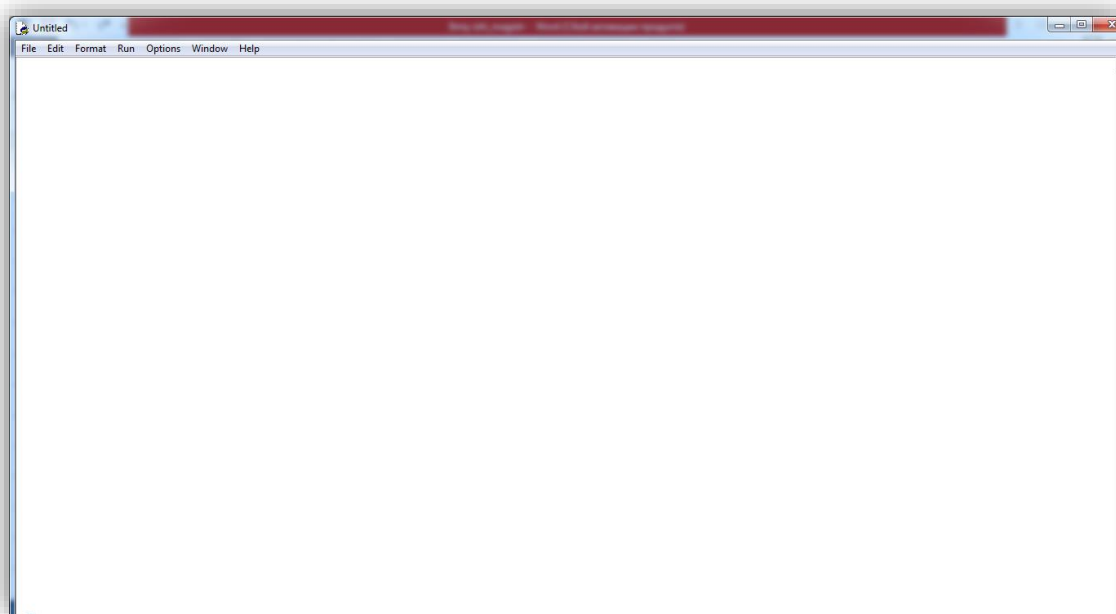
1.6-rasm. IDLE Python dasturining ishchi oynasi

IDLE da yangi fayl yaratish:



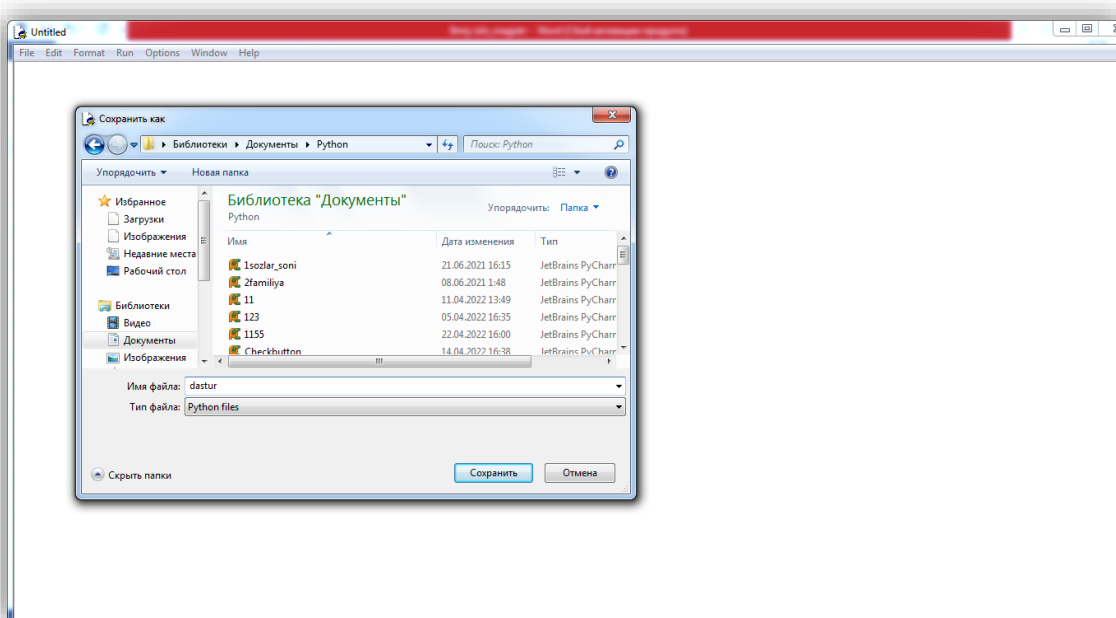
1.7-rasm. IDLE Pythonda yangi fayl yaratish

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

```
>>>
```

### **Mustaqil bajarish uchun topshiriqlar:**

1. "Python – interpretor dasturlash tili" deganda nimani tushunasiz?
2. Pythonda xotira qanday boshqariladi?
3. Python modullari nima?
4. Shaxsiy komputeringizga python dasturlash tilini yuklab oling.
5. Ushbu mavzuda berilgan ketma-ketlikda dasturni o`rnating.
6. Python dasturlash tilida yangi fayl ochib, 1-dastur.py deb nomlang.

## 2-MAVZU. PYTHON DASTURLASH TILIDA O'ZGARUVCHILAR

Dasturlash tili alifbosi quyidagicha:

- Yordamchi soʻzlar: print, len, type, def...
- Literal oʻzgaruvchilar: number, strings...
- Operatorlar: + - \* = , ; ...
- Grammatika (sinkaksis)
- Foylanuvchi yaratgan soʻzlar: variables, functions, classes...

O'zgaruvchi-bu dasturning oraliq yoki yakuniy natijasi saqlanishi mumkin bo'lgan eng oddiy nomlangan ma'lumotlar tuzilishi. Python - da o'zgaruvchini yaratish juda oson - "=" tayinlash operatori yordamida ma'lum bir identifikatorga qiymat berishingiz kerak.

```
a = 18
b = 3.1415926
c = "Salomlar"
d = [1, 2, 3, 4, 5]
```

Ushbu misollarda to'rtta o'zgaruvchidan foydalanildi: a o'zgaruvchisi int (butun son) turidagi qiymatni saqlaydi, float b tipidagi o'zgaruvchi (haqiqiy son), c tipidagi o'zgaruvchi satr (qator), ro'yxat, c tipidagi o'zgaruvchi (ro'yxat, bu holda uchta butun son).

Python dasturi yordamchi soʻzlar to'plamidan iborat. Har bir yordamchi soʻz 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.

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

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

Natija: `NameError: name 'Print' is not defined`

Bunda biz hech qanday natijaga erishmaymiz.

**Izohlar** - kod qismi nima qilishini belgilash uchun ishlatiladi. Dastur bajarilganda, izohlarni e'tiborsiz qoldiradi, shuning uchun ular dasturning ishlashiga ta'sir qilmaydi. Pythondagi izohlar 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 **izoh**ni bildiradi. Ya'ni, yuqoridagi misolda kodning birinchi ikki qatori izohlardir.

**Blok izohlarida** izoh matnidan oldin va keyin uchta bitta tirnoq qo'yiladi: `'''izoh matni'''`. Misol uchun:

```
'''  
    Konsolga kirish  
    Xabarni chop etish  
'''  
print("Salom dunyo!")
```

### Python 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:", "Mustafoeva",  
      "Iymona")
```

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: Muslimbek

Salom Muslimbek

### Mustaqil bajarish uchun topshiriqlar:

1. IDLE muhiti boshlang'ich oynasida quyidagi ifodalarning qiymatlarini hisoblang:
  - a)  $41115 + 31147 - 23321 = ?$
  - b)  $8 + 6 : (-2) - 2 \cdot (-11) = ?$
  - c)  $\frac{6,8 \cdot 0,04 \cdot 1,65}{3,3 \cdot 5,1 \cdot 0,16} = ?$
2. File->New File yoki Ctrl+N tugmalari yordamida yangi fayl yaratib, "SALOM" degan satrni chop eting.
3. Uch marta chiqarish operatoridan foydalangan holda, ism, familiya va sharifingizni ekranga chop eting.
4. Foydalanuvchidan ma'lumotlarni qanday olish mumkin?
5. Qaysi funksiya konsolga ma'lumotni chop etadi?
6. Python dasturlash tilidagi qaysi kutubxonalarni bilasiz?
7. Python dasturlash tilidagi operatorlar haqida ma'lumot bering.

### 3-MAVZU. PYTHONDA MA'LUMOT TURLARI

Ushbu mavzudan boshlab, masalalarning python dasturlash tilida tuzilgan kodlarini berib boramiz. Bu esa siz, aziz kitobxon, dasturlash tilini o'rganishingizda asosiy ko'makdosh bo'ladi degan umiddamiz!

Ma'lumot turlari python dasturlash tilidagi muhim tushunchadir. Har bir qiymat python dasturlash tilida o'z python ma'lumotlar turiga ega. Ma'lumot elementlarini tasniflash yoki ma'lumot qiymatini qandaydir ma'lumotlar toifasiga kiritish **ma'lumotlar turlari** deb nomlanadi.

Python o'zgaruvchining ma'lumotlar turini bilish uchun `type()` funksiyasini taqdim etadi. Raqamli ma'lumot turlariga `int`, `float`, `complex` tiplari kiradi.



#### 3.1-masala.

To'g'ri to'rtburchakning perimetrini hisoblash dasturini tuzing.  
Yechim:  $P = 2 \cdot (a + b)$ ; bu yerda  $a$ -to'g'ri to'rtburchakning eni va  $b$ -bo'yi.

#### Dastur kodi

```
a=23
b=17
p=2*(a+b)
print("Perimetr =", p)
```

#### Dastur natijasi

```
Perimetr = 80
```

#### 3.2-masala

Radiusi berilgan doiraning yuzasi va uzunligini hisoblash dasturini tuzing.

Yechim:  $S = \pi \cdot r^2$ ,  $l = 2\pi r$ ; bu yerda,  $r$ -doira radiusi.

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

### 3.3-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
<pre>n = int(input("N ni kiriting: ")) yigindi = (n * (n + 1)) / 2 print(n, "gacha yig'indi: ", yigindi)</pre>	<pre>N ni kiriting: 10 10 gacha yig'indi: 55.0</pre>

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
<pre>n = int(input("Butun son kiriting: ")) yigindi = sum(range(n+1)) print(n, "gacha yig'indi:", yigindi)</pre>	<pre>Butun son kiriting: 45 45 gacha yig'indi: 1035</pre>



### Mustaqil bajarish uchun topshiriqlar:

1. Uchburchakning uchlarining koordinatalari berilgan. Ular  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  hisoblanadi. Ikki nuqta orasidagi masofani topish formulasi va Geron formulasidan foydalanib uning perimetrini hamda yuzasini hisoblang.
2.  $a$  va  $b$  o'zgaruvchilardagi qiymatlarni almashtiradigan dastur tuzing.
3.  $a$ ,  $b$  va  $c$  o'zgaruvchilarining qiymatlari quyidagicha almashtiring:  $a \rightarrow b$ ,  
 $b \rightarrow c$ ,  $c \rightarrow a$  va chop eting.
4.  $a$ ,  $b$  va  $c$  o'zgaruvchilarining qiymatlari quyidagicha almashtiring:  $c \rightarrow b$ ,  
 $b \rightarrow a$ ,  $a \rightarrow c$  va chop eting.
5.  $x$  ning berilgan qiymatida  $y = 3x^6 - 6x^3 - 7$  ning qiymatini hisoblang.
6.  $x$  ning berilgan qiymatida  $y = 4(x-3)^6 - 7(x-3)^3 + 2$  ning qiymatini hisoblang.
7. Masofa 1 sm berilgan. sm larni metr larga aylantiring va butun qismini toping.
8. Tekislikda koordinatalari bilan berilgan ikki nuqta orasidagi masofani hisoblang.
9. Doiraning yuzi  $s$  berilgan. Uning diametri  $d$  ni va aylana uzunligi  $l$  ni hisoblang.
10. To'g'ri to'rtburchak shaklidagi parallelipipedning qirralari  $a$ ,  $b$ ,  $c$  berilgan.  
Uning hajmi  $v$  va sirtining yuzi  $s$  ni hisoblang.
11. Dastur natijasini toping:  

```
name = "John"
print('Hi, %s' % name)
```

## 4-MAVZU. PYTHONDA OPERATORLAR VA ULAR BILAN ISHLASH

**Operator**- ikkita operand o'rtasida ma'lum bir operatsiyani bajaradigan belgi. Muayyan dasturlash tilida operatorlar mantiqni dasturda qurish uchun asos bo'lib xizmat qiladi.

Python asosiy operatorlari:

- Arifmetik operatorlar
- Taqqoslash operatorlari
- Tayinlash Operatorlari
- Mantiqiy Operatorlar
- A'zolik Operatorlari

Ikki operand orasidagi arifmetik amallar **arifmetik operatorlar** yordamida amalga oshiriladi. Unga + (qo'shish), - (ayirish), \* (ko'paytirish), / (bo'lish), % (qoldikli bo'lish) va // (butun bo'lish) operatorlari hamda daraja (\*\*) operatori kiradi.

**Taqqoslash operatorlari** ikkita operandning qiymatlarini taqqoslaydi va mos ravishda True yoki False mantiqiy qiymatni qaytaradi.

**Tayinlash operatorlari** o'ng ifodaning qiymati belgilash operatorlari yordamida chap operandga beriladi. (=, +=, -=, \*=, //=, %=, \*\*=)

Qaror qabul qilish uchun natijalarni baholashda **mantiqiy operatorlardan** foydalanadi (and, or, not).

Python ma'lumotlar tuzilmasi ichidagi qiymatning a'ziligini Python **a'zolik operatorlari** yordamida tekshirish mumkin. Agar qiymat ma'lumotlar tarkibida bo'lsa, natija True bo'ladi, aks holda, u Falseni qaytaradi. (in, not in)

### 4.1-masala

Ikki butun son berilgan. Arifmetik operatorlar yordamida ularning yig'indisi, ayirmasi, ko'paytmasi, qoldig'ini va darajasini hisoblash dasturini tuzing.

Dastur kodi	Dastur natijasi
<code>a = 9</code>	<code>a + b= 13</code>
<code>b = 4</code>	<code>a - b= 5</code>
<code>add = a + b</code> <code># yig'indi</code>	<code>a * b= 36</code>

<code>sub = a - b</code>	<code># ayirma</code>	<code>a % b= 1</code>
<code>mul = a * b</code>	<code># ko`paytma</code>	<code>a ^ b= 6561</code>
<code>mod = a % b</code>	<code># qoldiqli bo`lish</code>	
<code>p = a ** b</code>	<code># daraja</code>	
<code># chop etish</code>		
<code>print("a + b=", add)</code>		
<code>print("a - b=", sub)</code>		
<code>print("a * b=", mul)</code>		
<code>print("a % b=", mod)</code>		
<code>print("a ^ b=", p)</code>		

#### 4.2-masala

$a_1, b_1, c_1, a_2, b_2, c_2$  koeffisientlar bilan berilgan, quyidagi ko'rinishga ega bo'lgan

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases}$$

chiziqli tenglamalar sistemasining yechimi hisoblang.

Masalan: 1 1 5 2 -1 4

Natija: 3 2

Dastur kodi	Dastur natijasi
<code>a1 = int(input("a1="))</code>	<code>a1=1</code>
<code>b1 = int(input("b1="))</code>	<code>b1=1</code>
<code>c1 = int(input("c1="))</code>	<code>c1=5</code>
<code>a2 = int(input("a2="))</code>	<code>a2=2</code>
<code>b2 = int(input("b2="))</code>	<code>b2=-1</code>
<code>c2 = int(input("c2="))</code>	<code>c2=4</code>
	<code>3.0 2.0</code>
<code>x = (b2 * c1 - b1 * c2) / (a1 * b2</code>	
<code>- a2 * b1)</code>	
<code>y = (a2 * c1 - a1 * c2) / (a2 * b1</code>	
<code>- a1 * b2)</code>	
<code>print(x,y)</code>	

#### 4.3-masala

Mantiqiy a va b o'zgaruvchilari berilgan. Mantiiy operatorlardan foydalangan holda a va b, a yoki b, a emas ifodalarning natijalarini toping.

Dastur kodi	Dastur natijasi
<pre> a = True b = False print(a and b) print(a or b) print(not a) </pre>	<pre> False True False </pre>

### Mustaqil bajarish uchun topshiriqlar:

1. Ikki xonali son berilgan. Dastlab uning chap qismidagi raqami so'ngra o'ng qismidagi raqami alohida-alohida qilib chop eting. Bu ishni bajarishda butunga bo'lish va qoldiqni hisoblash amallaridan foydalaning.
2. Sutkaning n-sekundi bo'lsa, sutka boshidan buyon necha soat o'tganligini aniqlang.
3. 1 yanvar seshanba bo'lib, hafta kunlari quyidagicha raqamlangan bo'lsa, 1-dushanba, 2-seshanba, 3-chorshanba, 4-payshanba, 5-juma, 6-shanba, 7-yakshanba bo'lsa, berilgan k (1-365) butun sonini yilning kuni deb hisoblab u haftaning qaysi kuniga to'g'ri kelishi aniqlang.
4. Biror yil berilgan. Shu yilning qaysi asrga tegishli ekanligi aniqlang, asr boshi quyidagicha hisoblanadi, misol: 20 asr 1900 yildan boshlanadi.
5. Birinchi avtomobilning tezligi v1, ikkinchisniki v2, ular orasidagi masofa s km. Avtomobillar bir-biriga tomon harakatlanayotgan bo'lsa, t vaqtdan keying ular orasidagi masofa hisoblang.
6. x, y sonlari berilgan. Ularni koordinatalar deb hisoblab 1- yoki 3-chorakda yotishi tekshirilsin.
7. Uchta butun son berilgan. Shu sonlarning uchburchakning tomonlarini tashkil etishi tekshirilsin.

## 5-MAVZU. PYTHONDA TARMOQLANUVCHI OPERATORLAR

Pythonda **shartli operatorlar** quyidagi operatorlar amalga oshiriladi:

1. **If operatori** ma'lum bir shartni tekshirish uchun ishlatiladi va agar shart to'g'ri bo'lsa, u if-blok deb nomlanuvchi kod blokini bajaradi.

```
if ifoda:
    #operatorlar
```

2. **If-else operatori** shartning False holatida bajariladigan if operatori bilan birlashtirilgan else blokini taqdim etadi. Agar shart to'g'ri bo'lsa, unda if-blok bajariladi. Aks holda, else-blok bajariladi.

```
if ifoda:
    #operatorlar
else:
    #else-block
```

3. **Elif operatori** bizga bir nechta shartlarni tekshirishga va ular orasidagi haqiqiy shartga qarab aniq operatorlar blokini bajarishga imkon beradi. Biroq, elif dan foydalanish ixtiyoriydir. Elif operatori C-dagi if-else-if ketma-ketlik kabi ishlaydi.

```
if ifoda 1:
    #operatorlar
elif ifoda 2:
    #operatorlar
elif ifoda 3:
    #operatorlar
else:
    #operatorlar
```

### 5.1-masala

Yil berilgan. Berilgan yilda necha kun borligini aniqlash dastur tuzing. Kabisa yili 366 kundan, boshqa yillar esa 365 kundan iborat.

#### Dastur kodi

```
yil=int(input("yil="))
if(yil%4==0):
    print("Kabisa yilida 366
kun bor")
```

#### Dastur natijasi

```
yil=2020
Kabisa yilida
366 kun bor
```

```
else:
    print("365 kun")
```

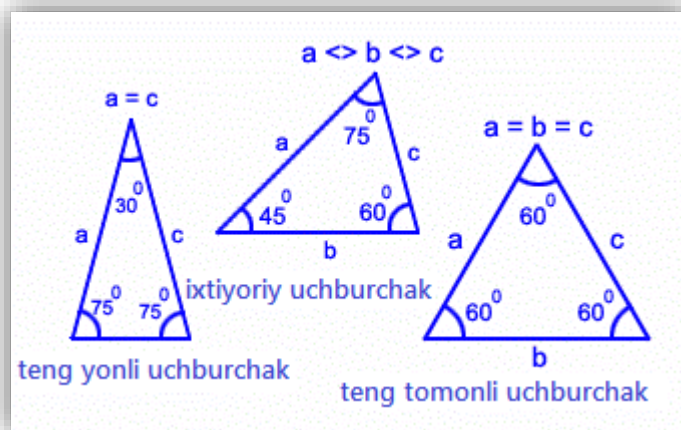
### 5.2-masala

To'g'ri sanani ifodalovchi kabisa bo'lmagan yilning (d) kuni va (m) oyi butun sonlar ko'rinishida berilgan. Berilgan sanadan oldingi sana toping  
 chop eting.  
 Masalan: 1 3  
 Natija: 28 2

Dastur kodi	Dastur natijasi
<pre>d=int(input("day=")) m=int(input("month=")) if (d&gt;1):     print(d-1, " ", m) elif(d==1):     x=m-1     if(x==1 or x==3 or x==5 or x==7 or x==8 or x==10 or x==12):         print("31 ", m-1)     if(x==4 or x==6 or x==9 or x==11):         print("30 ", m-1)     if(x==2):         print("28 ", m-1)</pre>	<pre>day=1 month=12 30  11</pre>

### 5.3-masala

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



5.1-rasm. Uchburchak turlari

#### Dastur kodi

```
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")
```

#### Dastur natijasi

```
Uchburchak
tomonlarini
kiriting.
x: 5
y: 6
z: 5
Teng yonli
uchburchak
```

#### 5.4-masala

Uchta butun son berilgan. Ulardan bittasi qolgan ikkitasidan ishoralari bilan farq qilsa, shu farq qiluvchi sonning tartib nomeri aniqlang.

Masalan: 4 -2

Natija: 1 2

#### Dastur kodi

```
a = float(input("1-son: "))
```

#### Dastur natijasi

```
1-son: -8
```

<pre> b = float(input("2-son: ")) c = float(input("3-son: "))  if ((a &gt; 0 and b &gt; 0 and c &gt; 0) or (a &lt; 0 and b &lt; 0 and c &lt; 0)):     print(0) elif(a * b &gt; 0):     print(3) elif(a * c &gt; 0):     print(2) else:     print(1) </pre>	<pre> 2-son: 5 3-son: -2 2 </pre>
--	-----------------------------------

### Mustaqil bajarish uchun topshiriqlar:

1. Butun son berilgan. Agar u musbat bo'lsa unga 1 ni qo'shing, aks holda o'zgarishsiz qoldirilsin. Olingan son chiqaring.
2. Butun son berilgan. Agar u manfiy bo'lsa 2 ni ayiring, 0 ga teng bo'lsa 10 bilan almashtiring. Olingan son chiqaring.
3. Uchta butun son berilgan. Ular orasidagi musbatlari soni toping.
4. Ikkita son berilgan. Ulardan kattasi chiqaring.
5. Ikkita son berilgan. Ulardan kichigining tartib raqami chiqaring.
6. Ikkita butun tipga tegishli a va b o'zgaruvchilar berilgan. Agar ularning qiymatlari teng bo'lmasa har bir o'zgaruvchiga qiymatlar yig'indisi berilsin, aks holda har bir o'zgaruvchining qiymati 0 bo'lsin. O'zgaruvchilarning natijaviy qiymatlari chiqaring.
7. Haqiqiy tipga tegishli x o'zgaruvchi berilgan. Uning qiymatiga mos keluvchi f funksiyaning qiymatini toping.

$$f = \begin{cases} -x, x \leq 0 \\ x^2, 0 < x < 2 \\ 4, x \geq 2 \end{cases}$$



## 6-MAVZU. PYTHONDA TAKRORLANISH OPERATORLARI

Biz **takrorlash operatorlari** yordamida bitta operator yoki operatorlar to'plamini qayta-qayta ishga tushirishimiz mumkin. Python dasturlash tilida quyidagi turdagi takrorlash operatorlari mavjud:

1. Pythonda **for operatori** ko'pincha ro'yxatlar, kortejlar yoki satrlar kabi takrorlanadigan obyektlar ustida takrorlash uchun ishlatiladi. Pythondagi for operatori ketma-ket elementlar bo'ylab o'tib, har safar kod blokini boshqaradi. For operatori sintaksisi:

```
for qiymat in ketma-ketlik:  
    #for bloki
```

2. **While operatori** kod blokining takrorlash iteratsiyasida berilgan shart, ya'ni shartli\_ifoda True bo'lganda, bajariladi. Agar biz iteratsiyani muddatidan oldin necha marta bajarishimizni bilmasak, noaniq takrorlash yozishimiz mumkin.

```
while shartli_ifoda:  
    #while bloki
```

### 6.1-masala

$a$  va  $b$  ( $a < b$ ) musbat butun sonlar berilgan.  $a$  dan  $b$  gacha bo'lgan hamma butun sonlarni son qiymati nechaga teng bo'lsa, shuncha martadan chiqaring.

Masalan: 3 sonini 3 marta ( $a$  va  $b$  ham kiradi)

#### Dastur kodi

#### Dastur natijasi

<code>a=int(input())</code>	2
<code>b=int(input())</code>	5
<code>for i in range(a,b+1):</code>	22
<code>for j in range(i):</code>	333
<code>print(i,</code>	4444
<code>end="")</code>	55555
<code>print()</code>	

### 6.2-masala

$n$  ( $n > 0$ ) butun son berilgan. Haqiqiy tipli  $a_k$  ketma-ketlik quyidagicha

aniqlanadi:  $a_0 = 1$ ,  $a_k = \frac{a_{k-1}+1}{k}$ ,  $k = 1, 2, \dots$  Ketma-ketlikning  $a_1, a_2, \dots, a_n$  hadlarini toping.

Dastur kodi	Dastur natijasi
<pre>n=int(input()) a0=1 for i in range(1,n+1):     a1=(a0+1)/i     print(a1)     a0=a1</pre>	<pre>2 2.0 1.5</pre>

### 6.3-masala

$n(n>0)$  butun son berilgan. Agar u 3 sonining darajasidan iborat bo'lsa *True*, aks holda *False* chiqaring.

Dastur kodi	Dastur natijasi
<pre>n=int(input("n=")) i=1 while i&lt;n:     i=i*3     if(i==n):         t=True     else:         t=False print(t)</pre>	<pre>n=243 True n=80 False</pre>

### 6.4-masala

$n$  va  $k$  musbat butun sonlari berilgan. Faqat qo'shish va ayirish amalidan foydalanib  $n$  ni  $k$  ga bo'lganda bo'linmaning butun hamda qoldiq qismini toping.

Dastur kodi	Dastur natijasi
<pre>n=int(input("n=")) k=int(input("k=")) b=0 while (n-k)&gt;0:     b=b+1     n=n-k print(b) print(n)</pre>	<pre>n=5 k=2 2 1</pre>

### Mustaqil bajarish uchun topshiriqlar:

1.  $k$  va  $n$  ( $n > 0$ ) butun sonlar berilgan.  $n$  marta  $k$  soni chiqaring.
2.  $a$  va  $b$  butun sonlar berilgan ( $a > b$ ).  $a$  va  $b$  sonlari orasidagi sonlarni o'sish tartibida chiqaring ( $a$  va  $b$  sonlari ham kiradi) hamda shu sonlar miqdori (soni)  $n$  chiqaring.
3.  $a$  va  $b$  butun sonlar berilgan ( $a < b$ ).  $a$  va  $b$  sonlari orasidagi sonlarni kamayish tartibida chiqaring ( $a$  va  $b$  sonlari ham kiradi) hamda shu sonlar miqdori (soni)  $n$  chiqaring.
4.  $n$  butun soni berilgan  $n^3 + (n+1)^3 + (n+2)^3 \dots + (2n)^3$ . (Yig'indi butun son).  
Yig'indi hisoblang.
5.  $n$  butun soni berilgan  $1, 1 \cdot 2, 1 \cdot 3, 1 \cdot \dots \{1, n\}$  ( $n$  ta ko'paytuvchi).  
Ko'paytma hisoblang.
6.  $n$  ( $n > 0$ ) butun soni berilgan.  $1, 1-1, 2+1, 3- \dots$  Ifodaning qiymati toping.  
Shart operatori qo'llanilmasin.
7.  $n$  ( $n > 0$ ) butun soni berilgan. Quyidagi formuladan foydalanib berilgan sonning kvadrati hisoblang:  $n^2 = 1 + 3 + 5 + \dots + (2n-1)$ . Har bir qadamdagi yig'indi chiqaring (natijada 1 dan  $n$  gacha bo'lgan butun sonlarning kvadrati chiqadi).
8.  $a$  haqiqiy va  $n$  butun sonlari berilgan ( $n > 0$ ).  $a^n = a \cdot a \cdot \dots \cdot a$ . ( $a$ ,  $n$  marta ko'paytirilgan)  $a$  ning  $n$ - darajasi hisoblang.
9.  $a$  va  $n$  sonlari berilgan. Bitta sikldan foydalanib  $a$  sonining 1 dan  $n$  gacha bo'lgan darajalari chiqaring.
10. 1 kg konfetning narxi haqiqiy sonda berilgan. 1, 2, ..., 10 kg konfetning bahosi chiqaring.
11. Kod natijasini toping:

```
for j in 'Salom! Men\ talabaman':  
    if j == '\\':  
        print("Topildi")  
        break  
else:  
    print ("Tugadi")
```

## 7-MAVZU. PYTHONDA SATRLAR

Pythonda **string** - bu bitta tirnoq, ikkita tirnoq yoki uch tirnoq ichidagi belgilar to'plami. Kompyuter belgilarni tushunmaydi, ichki sifatida u manipulyatsiya qilingan belgini 0 va 1 kombinatsiyasi sifatida saqlaydi. Har bir belgi ASCII yoki Unicode belgisida kodlangan.

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	

7.1-rasm. ASCII jadvali

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. Kompyuterlarda odatda 8-bit va kod jadvalining ikkinchi yarmi bilan ishlangan ASCII kengaytmasi foydalaniladi.

Boshqa tillar singari, Python satrlarini indekslash 0 dan boshlanadi. Masalan, "HELLO" satri quyidagi rasmda ko'rsatilganidek indekslanadi:

str = "HELLO"

H	E	L	L	O
---	---	---	---	---

0

1

2

3

4

str[0] = 'H'

str[:] = 'HELLO'

str[1] = 'E'

str[0:] = 'HELLO'

str[2] = 'L'

str[:5] = 'HELLO'

str[3] = 'L'

str[:3] = 'HEL'

str[4] = 'O'

str[0:2] = 'HE'

str[1:4] = 'ELL'

str = "HELLO"

H	E	L	L	O
---	---	---	---	---

-5

-4

-3

-2

-1

str[-1] = 'O'

str[-3:-1] = 'LL'

str[-2] = 'L'

str[-4:-1] = 'ELL'

str[-3] = 'L'

str[-5:-3] = 'HE'

str[-4] = 'E'

str[-4:] = 'ELLO'

str[-5] = 'H'

str[::-1] = 'OLLEH'

7.2-rasmi. Satrni indekslashga misol

### 7.1-masala

Bo'sh bo'lmagan satrdan n-indeks belgisini o'chirish uchun Python dasturini yozing.

#### Dastur kodi

#### Dastur natijasi

```
str="Buxoro"
n=int(input("indeks ="))
qism_1 = str[:n]
qism_2 = str[n+1:]
print(qism_1 + qism_2)
```

```
indeks =2
Buoro
```

### 7.2-masala

Satrdan takrorlangan belgilarni hisoblash uchun python dasturini yozing.

#### Dastur kodi

#### Dastur natijasi

```
import collections
str1 = 'Python dasturlash tili'
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]))
```

```
t 3
h 2
2
a 2
s 2
l 2
i 2
```

### 7.3-masala

Satrning o'rtasiga satr qo'shish uchun Python dasturini yozing.

#### Dastur kodi

#### Dastur natijasi

```
str = "[[]]"
word = "Python"
print(str[:2] + word +
      str[2:])
```

```
[[Python]]
```

#### Mustaqil bajarish uchun topshiriqlar:

1.  $c$  simvoli berilgan. Uning kodi chop eting.
2.  $n$  ( $31 < n < 127$ ) butun soni berilgan. Kodi  $n$  ga teng bo'lgan simvol chop eting.
3.  $c$  simvoli berilgan.  $c$  simvoldan (jadvalda) oldin va keyin joylashgan 2 ta simvol chop eting.
4.  $n$  ( $0 < n < 27$ ) soni berilgan. Lotin alfavitidagi birinchi  $n$  ta bosh hartflar chop eting.
5.  $n$  ( $0 < n < 27$ ) soni berilgan. Lotin alfavitidagi oxirgi  $n$  ta kichik hartflar chop eting.
6. Raqam yoki harfni aks ettiruvchi  $c$  simvol berilgan. Agar  $c$  raqam bo'lsa, "raqam", lotin harfi bo'lsa "lat", krill harfi bo'lsa "rus" satri chop eting.
7. Bo'sh bo'lmagan satr berilgan. Uning birinchi va oxirgi simvollarining kodlari chop eting.
8.  $n$  ( $n > 0$ ) butun son va  $c$  simvol berilgan.  $c$  simvoldan tuzilgan  $n$  uzunlikka ega satr chiqaring.
9.  $n$  ( $n > 0$ ) juft soni va  $c1$ ,  $c2$ , simvollar berilgan.  $c1$  dan boshlab navbat bilan almashib keluvchi  $n$  uzunlikka ega  $c1$  va  $c2$  dan tuzilgan ketma-ketlik chop eting.
10. Satr berilgan. Uni joylashish o'rniga nisbatan teskari tartibda chop eting.

## 8-MAVZU. PYTHONDA SATRLAR (STRING FUNKSIYALARI)

- **string.capitalize()** - satrning barcha probel bilan ajratilgan qismlarining birinchi belgilarini katta harfga o`giradi.
- **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.
- **string.count(string,begin,end)** - Bu boshlanish va tugash indeksleri orasidagi satrdagi substring paydo bo'lishining sonini hisoblaydi.
- **string.isalpha()** - Agar barcha belgilar alifbo belgilari bo'lsa True ni qaytaradi va kamida bitta belgi xato bo'lsa, aks holda False bo'ladi.
- **string.isdigit()** - Agar barcha belgilar raqamli belgilar bo'lsa True ni qaytaradi va kamida bitta belgi boshqa bo'lsa, False bo'ladi.
- **string.islower()** - satrning barcha belgisilari kichik harf bilan yozilgan bo'lsa True, aks holda False qaytaradi.
- **string.isupper()** - satrning barcha belgisilari kichik harf bilan yozilgan bo'lsa True, aks holda False qaytaradi.
- **len(string)** – satrning uzunligini, ya'ni belgilar sonini qaytaradi.
- **string.lower()** - satrning barcha belgilarini kichik harfga o`giradi.
- **string.upper()** - satrning barcha belgilarini katta harfga o`giradi.
- **string.split(str,num=string.count(str))** - Satrni ajratuvchi str ga muvofiq ajratadi. Agar ajratuvchi taqdim etilmasa, satr bo'sh joyga muvofiq bo'linadi. U ajratuvchi bilan birlashtirilgan substring ro'yxatini qaytaradi.

### 8.1-masala

S1 va S2 satrlari berilgan. S2 satrini S1 satrida takrorlanishlar sonini chiqarish uchun Python dasturini tuzing.



Dastur kodi	Dastur natijasi
<pre>s1=input("Matn : ") s2=input("takrorlanuvchi matn : ") n=0 while s1.find(s2)!=-1:     n=n+1  s1=s1[s1.find(s2)+len(s2):] print("matn ",n," marta takrorlandi")</pre>	<pre>Matn : assalomu aleykum      salom hammaga takrorlanuvchi matn : salom matn      2      marta takrorlandi</pre>

### 8.2-masala

Probel bilan ajratilgan va faqat katta harflar bilan terilgan o'zbekcha so'zlardan iborat satr berilgan. Satrdagi kamida 1 ta 'A' harfi bor so'zlar sonini aniqlash uchun Python dasturini tuzing.

Dastur kodi	Dastur natijasi
<pre>s=input("Matn:").split(" ") n=0 for i in s:     if i.find("A")&gt;=0:         n=n+1 print(n)</pre>	<pre>Matn:AXBOROT TIZIMLARI YO`NALISHI 3</pre>

### 8.3-masala

Probel bilan ajratilgan o'zbekcha so'zlardan iborat satr berilgan. Shu satrdagi so'zlarni quyidagicha o'zgartiring. Satrning oxirgi harfi bilan bir xil bo'lgan harflarni "." bilan almashtiruvchi dastur tuzing.

Dastur kodi	Dastur natijasi
<pre>str=input() m=str[-1] c="." for i in range(len(str)):     str=str.replace(m,c) print(str)</pre>	<pre>minimum .ini.u.</pre>

### **Mustaqil bajarish uchun topshiriqlar:**

1. Satr berilgan. Unda joylashgan raqamlar soni hisoblang. Satr berilgan. Undagi lotin alfavitining bosh harflari soni hisoblang.
2. Satr berilgan. Satrga kirmagan barcha lotin va rus harflarining soni hisoblang.
3. Satr berilgan. Berilgan satrdagi barcha bosh harflar kichik harflarga aylantirilsin.
4. Satr berilgan. Berilgan satrdagi barcha kichik harflar bosh harflarga aylantirilsin.
5. Satr berilgan. Berilgan satrdagi barcha bosh harflar kichik harflarga, kichik harflar bosh harflarga aylantirilsin.
6. Satr berilgan. Agar satr butun sondan iborat bo'lsa 1, haqiqiy sondan iborat bo'lsa 2, satrni son ko'rinishiga o'tkazib bo'lmasa 0 chop eting.
7. Butun musbat son berilgan. Bu sonni tasvirlovchi raqamlardan iborat simvollar chapdan o'ngga qaragan tartibda chop eting.
8. Butun musbat son berilgan. Bu sonni tasvirlovchi raqamlardan iborat simvollar o'ngdan chapga qaragan tartibda chop eting.
9. Butun musbat sonni tasvirlovchi satr berilgan. Bu sonning raqamlari yig'indisi hisoblang.

## 9-MAVZU. PYTHONDA RO'YXATLAR VA TO'PLAMLAR (LIST MA'LUMOT TURI)

Ro'yxat-bu [] belgisi bilan belgilanadigan va vergul bilan bo'linadigan elementlar guruhi. Ro'yxat har xil turdagi qiymatlar yoki elementlar to'plami sifatida belgilanishi mumkin:

```
list1 = [1, 2, "Python", "Program", 15.9]
list2 = ["Amy", "Ryan", "Henry", "Emma"]
```

Ro'yxat quyidagi xususiyatlarga ega:

- Ro'yxatlar tartiblangan;
- Ro'yxatning elementiga indeks orqali kirishi mumkin;
- Ro'yxatlar o'zgaruvchan turlardir, ya'ni ularni o'zgartirib bo'ladi.

Indekslash 0 dan boshlanadi. Ro'yxatning birinchi elementi 0-indeksda, ro'yxatning ikkinchi elementi 1-indeksda saqlanadi va hokazo.

Ro'yxatlar Pythondagi eng ko'p qirrali ma'lumotlar tuzilmalari, chunki ular o'zgaruvchan va tayinlash operatori yordamida yangilanishi mumkin. Python shuningdek, ro'yxatga qiymatlarni qo'shish uchun ishlatilishi mumkin bo'lgan append() va insert() usullarini taqdim etadi.

### 9.1-masala

Ro'yxatdagi elementlarning chastotasini olish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
<pre>import collections my_list = [1, 5, 1, 1, 6, 4, 4, 4, 5, 2] print("Original Ro'yxat : ", my_list) ctr = collections.Counter(my_list) print("Elementlar : ", ctr)</pre>	<pre>Original Ro'yxat : [1, 5, 1, 1, 6, 4, 4, 4, 5, 2] Elementlar      : Counter({1: 3, 4: 3, 5: 2, 6: 1, 2: 1})</pre>

### 9.2-masala

Berilgan aralash ro'yxatdagi butun sonlarni hisoblash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
<pre>def count_integer(list1):     ctr = 0     for i in list1:         if isinstance(i, int):             ctr = ctr + 1     return ctr  list1 = [1, 'abcd', 3, 1.2, 4, 'xyz', 5, 'pqr', 7, -5, - 12.22]  print("Original ro`yxat:") print(list1)  print("\nButun sonlar nechta:") print(count_integer(list1))</pre>	<p>Original ro`yxat: [1, 'abcd', 3, 1.2, 4, 'xyz', 5, 'pqr', 7, -5, -12.22]</p> <p>Butun sonlar nechta: 6</p>

### Mustaqil bajarish uchun topshiriqlar:

1. Berilgan ro'yxatdagi noyob raqamlarning mahsulotini hisoblash uchun Python dasturini yozing.
2. Bir xil uzunlikdagi bir nechta ro'yxatlarni aralashtirish uchun Python dasturini yozing.
3. Belgilar yoki satrlarni o'z ichiga olgan satrlarning berilgan ro'yxatidan so'zlarni olib tashlash uchun Python dasturini yozing
4. Belgilangan diapazon indeksleri orasidagi ro'yxatdagi raqamlar yig'indisini hisoblash uchun Python dasturini yozing.
5. Berilgan uchta ro'yxatda bir xil juftlikni hisoblash uchun Python dasturini yozing.
6. Berilgan raqamlar ro'yxatidagi maksimal va minimal qiymatlarning barcha indekslarini topish uchun Python dasturini yozing.

## 10-MAVZU. PYTHONDA RO'YXATLAR VA TO'PLAMLAR (SET MA'LUMOT TURI)

To'plamni vergul bilan ajratilgan o'zgarmas elementlarni figurali qavslar bilan yopish orqali yaratish mumkin {}. Python set () usuli orqali ketma-ketlikdan to'plamni yaratish uchun ishlatilishi mumkin.

```
Days = {"Monday", "Tuesday", "Wednesday",  
        "Thursday"}  
Days = set(["Monday", "Tuesday",  
            "Wednesday", "Thursday"])
```

### 10.1-masala

To'plam uzunligini topish uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
<pre>setn = {5, 5, 5, 5, 5, 5, 7} print("Original to`plam:") print(setn) print(type(setn)) print("to`plam uzunligi:") print(len(setn))</pre>	<pre>Original to`plam: {5, 7} &lt;class 'set'&gt; to`plam uzunligi: 2</pre>

### 10.2-masala

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

Dastur kodi	Dastur natijasi
<pre>A = set(["yashil", "ko`k",         "sariq"]) B = set(["oq", "sariq"]) print("Original to`plam:") print(A) print(B) r1 = A.difference(B) print("Ayirma A-B:") print(r1) r2 = B.difference(A) print("Ayirma B-A:") print(r2)</pre>	<pre>Original to`plam: {'yashil', 'sariq', 'ko`k'} {'oq', 'sariq'} Ayirma A-B: {'yashil', 'ko`k'} Ayirma B-A: {'oq'}</pre>

### **Mustaqil bajarish uchun topshiriqlar:**

1. To'plamdagi maksimal va minimal qiymatlarni topish uchun Python dasturini yozing.
2. To'plam uzunligini topish uchun Python dasturini yozing.
3. Berilgan qiymat to'plamda mavjudligini yoki yo'qligini tekshirish uchun Python dasturini yozing.
4. Berilgan ikkita to'plamda umumiy elementlar yo'qligini tekshirish uchun Python dasturini yozing.
5. Ikkinchi to'plamning birinchi to'plam bilan kesishishini olib tashlash uchun Python dasturini yozing.
6. Yig'indisi berilgan qiymatga teng bo'lgan ro'yxatdagi barcha juft elementlarni topadigan Python dasturini yozing.
7. Barcha noyob so'zlarni topish va berilgan satrlar ro'yxatidan paydo bo'lish chastotasini hisoblash uchun Python dasturini yozing.

## 11-MAVZU. PYTHONDA MASSIVLAR (BIR O'LCHOVLI MASSIVLAR)

**Massiv** C/C++, JavaScript va boshqalar kabi ko'pgina dasturlash tillarida mashhur. **Array**-bu bir xil turdagi bir nechta elementlar to'plami. U bir nechta qiymatlarni bitta o'zgaruvchida saqlash uchun ishlatiladi.

Massivni Pythonda **array** nomli modul yordamida boshqarish mumkin. Bu faqat ma'lum ma'lumotlar qiymatlarini manipulyatsiya qilishimiz kerak bo'lganda foydalidir.

Element-massivda saqlangan har bir element.

Indeks-elementning massivdagi joylashuvi, elementning holatini aniqlash uchun ishlatiladigan raqamli indeks.

### 11.1-masala

$n$  ( $n > 0$ ) butun son berilgan. Dastlabki  $n$  ta musbat toq sonlarni saqlaydigan  $n$  o'lchamli butun sonli massiv tashkil etilsin.

#### Dastur kodi

```
from array import*
n=int(input("n="))
m=[]
for i in range(1,2*n,2):
    m.append(i)
print(m)
```

#### Dastur natijasi

```
n=5
[1, 3, 5, 7, 9]
```

### 11.2-masala

Butun  $n$  ( $n > 1$ ) soni, arifmetik progressiyaning birinchi hadi  $a$  va uning ayirmasi  $d$  berilgan. Shulardan foydalanib o'zida arifmetik progressiyaning dastlabki  $n$  ta hadini saqlovchi massiv tashkil etilsin.

Dastur kodi	Dastur natijasi
<pre> from array import* n=int(input("n=")) a1=int(input("a1=")) d=int(input("d=")) m=[] for i in range(1,n+1,1):     m.append(a1)     a1=a1+d print(m) </pre>	<pre> n=5 a1=10 d=2 [10, 12, 14, 16, 18] </pre>

### 11.3-masala

$n$  o'lchamli massiv hamda  $k$  va  $l$  butun sonlari berilgan. ( $1 < k \leq l < n$ ).  $k$  indeksdan  $l$ -indeksgacha bo'lgan elementlardan boshqa barcha massiv elementlarining o'rta arifmetigi topilsin.

Dastur kodi	Dastur natijasi
<pre> from array import* n=int(input("n=")) k=int(input("k=")) l=int(input("l=")) m=[] s=0 s2=0 for i in range(0,n,1):     m.append(int(input()))     s=s+m[i] for j in range(k,l+1):     s2=s2+m[j] print((s-s2)/(n-(l-k+1))) </pre>	<pre> n=6 k=3 l=4 1 2 3 4 5 6 3.0 </pre>

### Mustaqil bajarish uchun topshiriqlar:

1.  $n$  ( $n > 0$ ) butun son berilgan. Dastlabki  $n$  ta musbat toq sonlarni saqlaydigan  $n$  o'lchamli butun sonli massiv tashkil eting.



2.  $n$  ( $n > 0$ ) butun son berilgan. 2 ning darajalarini saqlaydigan  $n$  o'lchamli butun sonli massiv tashkil eting.
3. Butun  $n$  ( $n > 1$ ) soni, arifmetik progressiyaning birinchi hadi  $a$  va uning ayirmasi  $d$  berilgan. Shulardan foydalanib o'zida arifmetik progressiyaning dastlabki  $n$  ta hadini saqlovchi massiv tashkil eting.
4.  $n$  ( $n > 1$ ) butun soni hamda birinchi hadi  $b$  va maxraji  $q$  bo'lgan geometric progressiya berilgan. Shulardan foydalanib o'zida geometrik progressiyaning dastlabki  $n$  ta hadini saqlovchi massiv tashkil eting.
5.  $n$  ( $n > 2$ ) butun soni berilgan.  $f_1=1$ ,  $f_2=1$ ,  $f_k=f_{k-2}+f_{k-1}$ ,  $k=3,4,\dots$   $f_k$  Fibonachchi sonlar ketma-ketligida birinchi  $n$  ta elementni o'z ichiga oladigan  $n$  o'lchamli butun sonli massiv ifodalansin va chop eting.
6.  $n$  ( $n > 2$ ),  $a$  va  $b$  butun sonlar berilgan. 1-elementi  $a$  ga, 2-elementi  $b$ , har bir keyingi elemeti barcha avvalgi elementlar (o'zidan oldingi barcha element) yig'indisiga teng bo'lgan  $n$  o'lchamli butun sonli massiv ifodalansin va chop eting.
7.  $n$  o'lchamli  $a$  massiv berilgan. Uning elementlari teskari tartibda chiqaring.

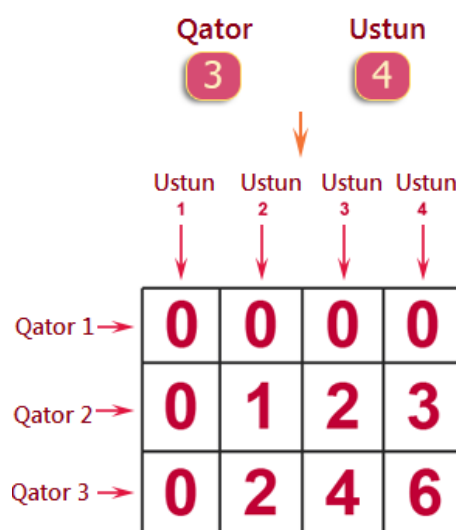
## 12-MAVZU. PYTHONDA MASSIVLAR (IKKI O'LCHOVLI MASSIVLAR)

### 12.1-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.

$i = 0, 1, \dots, m-1$

$j = 0, 1, \dots, n-1$ .



12.1-rasm

#### Dastur kodi

```
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)
```

#### Dastur natijasi

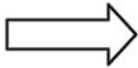
```
Qatorlar sonini kiriting: 2
Ustunlar sonini kiriting: 3
[[0, 0, 0],
 [0, 1, 2]]
```

### 12.2-masala

Sizga tasvirni ifodalovchi  $n \times n$  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.

1	2	3
4	5	6
7	8	9

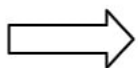


7	4	1
8	5	2
9	6	3

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

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

5	1	9	11
2	4	8	10
13	3	6	7
15	14	12	16



15	13	2	5
14	3	4	1
12	6	8	9
16	7	10	11

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]]
```

### Mustaqil bajarish uchun topshiriqlar:

1.  $n$  o'lchamli butun sonli massiv berilgan. Berilgan massivni indeksleri bo'yicha tartibida tartiblab, massivdagi juft sonlar va ularning miqdori  $k$  chiqaring.
2.  $n$  o'lchamli, butun sonli massiv berilgan. Berilgan massivdagi barcha toq sonlarni o'z ichiga oladigan elementlarni o'sish tartibida tartiblab, chop eting hamda ularning miqdori  $k$  aniqlang.
3.  $n$  o'lchamli butun sonli massiv berilgan. Massivdagi juft sonli elementlarining indekslarini o'sish tartibida, toq sonli elementlarining indekslarini kamayish tartibida tartiblab, massiv chop eting.
4.  $n$  o'lchamli  $a$  massiv va  $k(1 \leq k \leq n)$  butun soni berilgan. Massiv elementlari shart operatoridan foydalanmasdan quyidagi tartibda chop eting:  $a_k, a_{k-1}, a_{k-2}, \dots, a_1$ .
5.  $n$  o'lchamli  $a$  massiv berilgan ( $n$ -juft son). (indeksleri o'sish tartibida) Juft indeksdagi elementlari chiqaring.  $a_2, a_4, \dots, a_n$ . Shart operatoridan foydalanilmasin.
6.  $n$  o'lchamli  $a$  massiv berilgan ( $n$ -toq son). Massivning toq indeksida turgan elementlari indekslarini kamayish tartibida tartiblab chiqaring.  $a_n, a_{n-2}, a_{n-4}, \dots, a_1$  shart operatoridan foydalanilmasin.
7.  $n$  o'lchamli  $a$  massiv berilgan. Avval toq indeksdagi elementlar, keyin juft indeksdagi elementlar kamayish tartibida chop eting.

## 13-MAVZU. PYTHONDA FUNKSIYALAR

Pythonda funktsiyani yaratish **def** kalit so'z yordamida aniqlanadi:

```
def my_function():  
    print("Hello from a function")
```

Funktsiyani chaqirish uchun funktsiya nomidan, so'ngra qavsdan foydalaning:

```
def my_function():  
    print("Hello from a function")
```

```
my_function()
```

Axborot argument sifatida funktsiyalarga o'tkazilishi mumkin. Argumentlar funktsiya nomidan keyin, qavs ichida ko'rsatiladi. Siz xohlagancha argument qo'shishingiz mumkin, ularni vergul bilan ajrating. Quyidagi misolda bitta argument (ism) bo'lgan funktsiya mavjud. Funktsiya chaqirilganda, biz to'liq ismni chop etish uchun funktsiya ichida ishlatiladigan ism bo'ylab o'tamiz:

```
def my_function(fname):  
    print(fname + " Mustafoyev")
```

```
my_function("Muslimbek")
```

```
my_function("Oybek")
```

### 13.1-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

```
def isPalindrome(x):  
    s=str(x)  
    left,right=0, len(s)-  
1  
    while left<right:
```

#### Dastur natijasi

True

```

        if
s[left]!=s[right]:
        return False
        left+=1
        right-=1
        return True

print(isPalindrome(5335))

```

### 13.2-masala

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

#### Dastur kodi

```

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)):
    count=num//INT[i]
    s+=roman[i]*count
    num-=INT[i]*count
return s
print(intToRoman(59))
print(intToRoman(123))
print(intToRoman(2748))

```

#### Dastur natijasi

LIX  
CXXIII  
MMDCCXLVIII

### 13.3-masala

Satr uzunligini hisoblash uchun Python dasturini yozing

#### Dastur kodi

```

def satr_uzunlik(str1):
    k = 0

```

#### Dastur natijasi

6

```

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

```

### 13.4-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]
natiya = uzun_suz_top(["PHP",
"Amaliyot", "Dasturlash"])
print("Eng uzun so`z:
",natiya[1])
print("Uzun so`zning uzunligi:
",natiya[0])

```

```

Eng uzun
so`z:
Dasturlash
Uzun
so`zning
uzunligi:
10

```

### 13.5-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

```

```

{'Python': 1,
'tili': 1,
'masalalar':
1, 'kitobi':
1,
'dasturlash':
1}

```

```
return counts

print( word_count('Python
dasturlash tili masalalar
kitobi'))
```

### Mustaqil bajarish uchun topshiriqlar:

1. Berilgan  $x$  haqiqiy soni uchun  
 $x < 0$  da -1  
 $x = 0$  da 0 va  
 $x > 0$  da 1  
qiymat qaytaruvchi butun tipli  $sign(x)$  funksiyasi tasvirlansin. Bu funksiyadan foydalanib berilgan  $a$  va  $b$  sonlari uchun mos qiymatlarni oling.
2. Kvadrat tenglamaning ildizlari sonini aniqlaydigan  $RootsCount(a,b,c)$  butun tipli funksiya tasvirlansin ( $a, b, c$  - haqiqiy parametr  $a \neq 0$ ). Bu funksiyadan foydalanib berilgan 3 ta kvadrat tenglamaning har biri uchun ildizlari sonini aniqlang.
3.  $r$  radiusli doiraning yuzasini hisoblovchi haqiqiy tipli  $CircleS(r)$  funksiya tasvirlansin. Bu funksiyadan foydalanib radiuslari berilgan 3 ta doiraning har birining yuzasini hisoblang.
4. Koordinata boshida yotmaydigan (tekislikdagi) nuqtaning qaysi chorakda joylashganligini aniqlovchi butun tipli  $Quarter(x,y)$  funksiyasi tasvirlansin. Bu funksiyadan foydalanib, berilgan 3 ta koordinata boshida yotmaydigan nuqtalarning qaysi choraklarda joylashganligini aniqlang.
5.  $n$  ( $n > 1$ ) parametr tub son bo'lsa *true*, aks holda *false* qiymat qaytaruvchi mantiqiy tipli  $IsPrime(n)$  funksiya tasvirlansin. Har bir 1 dan katta bo'lgan 10 ta sondan iborat sonlar nabori berigan. Bu funksiyadan foydalanib berilgan nabordagi tub sonlar miqdorini aniqlang.



## 14-MAVZU. PYTHONDA FUNKSIYALAR. REKURSIV FUNKSIYALAR

### 14.1-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

### 14.2-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

### 14.3-masala

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

Dastur kodi	Dastur natijasi
<pre>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]]))</pre>	21

### 14.4-masala

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

Dastur kodi	Dastur natijasi
<pre>def factorial(n):     if n &lt;= 1:         return 1     else:         return n * (factorial(n - 1))  print(factorial(12))</pre>	479001600

### 14.5-masala

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

Dastur kodi	Dastur natijasi
<pre>def Fibonachchi(n):     if n &lt; 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?<br '))<br=""/>print(Fibonachchi(k))</pre>	<pre>N-Fibbonachi soni? 7 13</pre>

### 14.6-masala

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

Dastur kodi	Dastur natijasi
<pre>def sumDigits(n):     if n == 0:         return 0     else:         return n % 10 + sumDigits(int(n / 10))  print(sumDigits(345)) print(sumDigits(45))</pre>	<pre>12 9</pre>

### 14.7-masala

$n+(n-2)+(n-4)\dots (n-x \leq 0)$  gacha) musbat butun sonlar yig'indisini hisoblash uchun Python dasturini yozing.

Dastur kodi	Dastur natijasi
<pre>def sum_series(n):     if n &lt; 1:         return 0     else:         return n + sum_series(n - 2)  print(sum_series(5)) print(sum_series(10))</pre>	9 30

### 14.8-masala

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

Dastur kodi	Dastur natijasi
<pre>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))</pre>	125

### Mustaqil bajarish uchun topshiriqlar:

1.  $n$  faktorialni hisoblovchi haqiqiy tipli  $Fact(n)$  funksiyasini tuzing. Bu funksiyadan foydalanib berilgan 5 ta butun musbat sonning har biri uchun faktoriallarni hisoblang.

2. *MonthDays(m,y)* funksiyasidan foydalanib berilgan, to‘g‘ri  $(d,m,y)$  kun, oy, yil formatdagi sanadan keyingi kun sanasini hisoblovchi *NextDate(d,m,y)* funksiyasini tuzing. Prosedurani qo‘llab, berilgan 3 ta sanadan keying sanalarni toping.
3. Berilgan yilning  $m$ -oyi uchun *IsLeapYear* funksiyasidan foydalanib kunlar sonini hisoblovchi butun tipli *MonthDays(m,y)* funksiyasini tuzing. Bu funksiyadan foydalanib berilgan 2 ta turli butun musbat qiymatlar uchun funksiyaning qaytaradigan qiymatini chop eting.
4.  $f_k$  Fibonachchi sonlarining  $n$ -hadini hisoblaydigan butun tipli *Fib(n)* funksiyasini tuzing. Bu funksiyadan foydalanib  $n_1, n_2, \dots, n_5$  nomerlarga to‘g‘ri keluvchi Fibonachchi sonlarini toping.
5. Ikkita butun sonning eng katta umumiy bo‘luvchisini (EKUB) topish uchun Python dasturini yozing.
6. 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.
7. 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.

## 15-MAVZU. PYTHONDA FAYLLAR BILAN ISHLASH

Pythonda fayllar bilan ishlashning asosiy funksiyasi `open()` funksiyasidir.

`Open ()` funksiyasi ikkita parametрни oladi: **fayl nomi va rejimi**.

Faylni ochishning to'rt xil usuli (rejimi) mavjud:

- "r" - o'qish - standart qiymat. O'qish uchun faylni ochadi, agar fayl mavjud bo'lmasa xato
- "a" - Qo'shish-qo'shish uchun faylni ochadi, agar u mavjud bo'lmasa, faylni yaratadi
- "v" - yozish-yozish uchun faylni ochadi, agar u mavjud bo'lmasa, faylni yaratadi
- "x" - yaratish-belgilangan faylni yaratadi, agar fayl mavjud bo'lsa, xato qaytaradi

Bundan tashqari, siz faylni ikkilik yoki matn rejimi sifatida ko'rib chiqish kerakligini belgilashingiz mumkin:

- "t" - matn-standard qiymat. Matn rejimi
- "b" - ikkilik - ikkilik rejim (masalan, rasmlar)

O'qish uchun faylni ochish uchun fayl nomini ko'rsatish kifoya:

```
f = open("demofile.txt")  
f = open("demofile.txt", "rt")
```

### 15.1-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 aniqlashdastur tuzing.

Kiritish: INPUT.TXT da 3 ta son berilgan – progressiyaning

1-elementi  $A_1$  ( $1 \leq A_1 \leq 1000$ ),

2-elementi  $A_2$  ( $1 \leq A_2 \leq 1000$ )

so'ralgan element tartib raqami  $N$  ( $1 \leq N \leq 1000$ ).

Chiqarish: OUTPUT.TXT da arifmetik progressiyaning  $N$ -elementi chiqarildi.

Misol:

No	INPUT.TXT	OUTPUT.TXT
1	1 5 3	9

Dastur kodi	Dastur natijasi
<pre>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()</pre>	<pre>input.txt 1 5 3 output.txt 9</pre>

## 15.2-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 ( $1000 \leq N \leq 9999$ ) kiritilgan.

Chiqariluvchi qiymatlar: OUTPUT.TXT da N palindrom bo'lsa, "HA" so'zini, aks holda "YO'Q" so'zini chiqaring.

Misollar:

No	INPUT.TXT	OUTPUT.TXT
1	6116	YES
2	1231	NO

Dastur kodi	Dastur natijasi
<pre>f=open('input.txt','r') ff=open('output.txt','w') n=int(f.readline()) s=0 m=n while n&gt;0:     k=n%10     n=n//10     s=s*10+k if m==s:     ff.write('YES') else:     ff.write('NO')</pre>	<pre>input.txt 7557 output.txt YES</pre>

```
ff.close()
```

### 15.3-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 toping.

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 chiqaring.

Misol:

No	INPUT.TXT	OUTPUT.TXT
1	00101110000110	4

#### Dastur kodi

```
f=open('input.txt', 'r')
ff=open('output.txt', 'w')
n=f.readline()
s=0
k=0
for i in range(len(n)):
    if n[i]=='0':
        s=s+1
    else:
        if s>k:
            k=s
        s=0
ff.write(str(k))
ff.close()
```

#### Dastur natijasi

```
input.txt
11100000000011101101011011
1110000000
output.txt
9
```



## 15.4-masala

Issiq havo

(Vaqt: 1 sek. Hajmi: 16 Mb. Qiyinligi: 17%)

Har bir kunida harorat  $0^{\circ}\text{C}$  dan yuqori bo'lgan davrlar ichida eng uzun shunday kunlar zanjiridagi kunlar sonini chiqaring.

Kiritish: INPUT.TXT da 1-satrdan  $N$  ( $1 \leq N \leq 100$ ), keyingi 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  $0^{\circ}\text{C}$  dan eng yuqori haroratli kunlar zanjirining uzunligi chiqaring. Agar har kuni harorat nomusbat (0 dan katta emas) bo'lgan bo'lsa, natija sifatida 0 ni chiqaring.

Misollar:

No	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

```
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
```

### Dastur natijasi

```
input.txt
5
-4 3 15 20 -10
output.txt
3
```

```
ff.write(str(k))  
ff.close()
```

### Mustaqil bajarish uchun topshiriqlar:

1.  $s$  satr berilgan. Agar  $s$  fayl nomi bo'lishi uchun to'g'ri kelsa, shu nom bilan bo'sh fayl hosil qilinsin va *true* chop etilsin. Agar  $s$  nom bilan fayl hosil qilish mumkin bo'lmasa *false* chop eting.
2.  $s$  fayl nomi va  $n(n > 1)$  butun soni berilgan. Butun son joylashadigan  $s$  nomli fayl tuzilsin va unga  $n$  gacha bo'lgan juft musbat sonlar ketma-ketligini yozing. ( $n$  ham kiradi)
3.  $s$  fayl nomi va haqiqiy  $a, d$  sonlari berilgan. Tashqi faylga 1-hadi  $a$  ga ayirmasi  $d$  ga teng bo'lgan arifmetik progressiyaning dastlabki 10 ta hadining qiymatlarini yozing.
4. 4 ta faylning nomi berilgan. Joriy katalogda joylashgan shu nomlar bilan berilgan fayllar sonini toping.
5. Butun sonli fayl nomi berilgan. Fayldagi elementlar soni topilsin. Agar bunday nomdagi fayl topilmasa – 1 ni chiqaring.
6.  $k$  butun soni va o'zida manfiy bo'lmagan butun sonlarni saqlovchi fayl berilgan. Fayldagi  $k$ -element chiqarilsin. Bunday element topilmasa – 1 ni chop eting.
7. To'rttadan kam bo'lmagan elementga ega butun sonli fayl berilgan. Ushbu faylning 1-, 2-,  $n-1$ ,  $n$ -elementlarini chop eting.
8. Haqiqiy sonli 2 ta fayl nomi berilgan. Ma'lumki, ulardan biri mavjud va bo'sh emas, 2-si joriy katalogda yo'q. 2-fayl tuzilsin va 1-fayldagi 1- va oxirgi element nusxasini unga o'tkazing.

## 16-MAVZU. PYTHONDA FAYLLAR BILAN ISHLASH

### 16.1-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:

No	INPUT.TXT	OUTPUT.TXT
1	2+3=5	YES
2	3*7=20	NO
3	two plus three is five	ERROR

#### Dastur kodi

```
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:
```

#### Dastur natijasi

```
input.txt
8+2=10
output.txt
YES
```

```

        x=1
    else:
        x=0
        break
if x==1:
    for i in
        range(len(n)):
            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()

```

## 16.2-masala

### Tangalar

(Vaqt: 1 sek. Hajmi: 16 Mb. Qiyinligi: 8%)

Stol ustida  $n$  ta tanga turibdi. Ularning ba'zilar 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 ( $1 \leq N \leq 100$ ) – tangalar soni kiritilgan. Qolgan satrlarda 1 tadan butun son kiritilgan – 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:

No	INPUT.TXT	OUTPUT.TXT
1	5 1 0 1 1 0	2

#### Dastur kodi

```
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:
```

#### Dastur natijasi

```
input.txt
6
1
0
1
1
0
0
output.txt
3
```

```
ff.write(str(k))
```

```
ff.close()
```

### 16.3-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  $N$  ( $0 \leq N \leq 10100$ ) kiritilgan.

Chiqariluvchi qiymatlar: OUTPUT.TXT da shu  $N$  uchun “aylanacha”lar soni chiqaring.

Misollar:

No	INPUT.TXT	OUTPUT.TXT
1	157892	3
2	203516	2
3	409578	4
4	236271	1

#### Dastur kodi

```
f=open('input.txt','r')
ff=open('output.txt','w')
n=f.readline()
s=0
for i in range(len(n)):
    if n[i]=='0' or
       n[i]=='6' or
       n[i]=='9':
        s=s+1
```

#### Dastur natijasi

```
input.txt
157892
output.txt
3
```

```

        if n[i]=='8':
            s=s+2

ff.write(str(s))
ff.close()

```

## 16.4-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", "")

```

O13  
M16

```
print(name[0] +  
str(len(name)))  
  
file.close()
```

### Mustaqil bajarish uchun topshiriqlar:

1. Fayl nomi va butun musbat  $n$  va  $k$  sonlari berilgan. Ko'rsatilgan nomdagi matn faylini tuzing va uning  $n$  ta satrining har biriga  $k$  tadan «\*» yozing.
2. Fayl nomi va butun  $n$  ( $0 < n < 27$ ) soni berilgan. Ko'rsatilgan nomdagi matn fayl tuzilsin va unga  $n$  ta satr quyidagi ko'rinishda yozing: 1-satrga «a», 2-satrga «ab», 3-satrga «abc» va hokazo,  $n$  satrga  $n$  ta kichik lotin alifbosi harflari alfavit tartibida yozing.
3. Fayl nomi va butun  $n$  ( $0 < n < 27$ ) soni berilgan. Ko'rsatilgan nomdagi matn fayl tuzilsin va unga  $n$  ta satr quyidagi ko'rinishda yozing: Har bir  $k$ -satrga  $n$  tadan simvol bo'lib, simvollarning  $k$  tasi alfavit tartibidagi lotin bosh harflari qolganlari «\*» simvolidan iborat. Misol: «a\*\*\*», «ab\*\*», «abc\*», «abcd».  $n=4$ .
4. Matnli fayl berilgan. Unda joylashgan simvollar va satrlar soni chiqaring.
5.  $s$  satr va matnli fayl berilgan. Fayl oxiriga  $s$  satr qo'shing.
6. 2 ta matnli fayl berilgan. 1-fayl oxiriga 2-faylda saqlanayotgan ma'lumotlarni qo'shing.
7.  $s$  satr va matnli fayl berilgan. Fayl boshiga  $s$  satr qo'shing.



## 17-MAVZU. PYTHON STANDART KUTUBXONASI

Python dasturlash tili yildan-yilga ommalashib bormoqda. Bunga birinchi navbatda Pythonning sodda va tushunarli sintaksi sabab bo'lsa, ikkinchi va ehtimol eng ko'zga ko'ringan sabab bu Pythonning keng qamrovli kutubxonalar to'plamidir.

Kutubxona bu boshqalar tarafidan yozilgan tayyor funksiyalar va obyektlar to'plami.

### •**datetime** — SANA VA VAQT

Ushbu modul yordamida Pythonda sanalar bilan ishlashimiz mumkin. Moduldan foydalanishdan avval uni import qilamiz. Har gal moduldan foydalanishda datetime deb qayta yozmaslik uchun, import qilishda modulga dt nomini beramiz.

### •**pprint** - CHIROYLI PRINT

pprint moduli yordamida turli o'zgaruvchilarni chiroyli ko'rinishda konsolga chiqarishimiz mumkin. Bu bizga uzun lug'atlar, JSON fayllar yoki matnlar bilan ishlashda juda asqotadi.

Misol uchun, avvalgi darslarimizning birida yaratgan bemor.json faylini ochamiz va avval print() keyin pprint() yordamida konsolga chiqaramiz.

### •**RegEx** - ANDOZA YORDAMIDA MATN IZLASH

Pythondagi juda foydali modullardan biri bu **re** (*regular expressions*) moduli. Bu modul yordamida biror matn berilgan andozaga tushish, tushmalsigini tekshrib ko'rishimiz mumkin. Yoki berilgan andoza asosida matnlar orasidan kerakli matnlarni ajratib olish mumkin. Masalan,

```
from pprint import pprint
import json
filename = 'bemor.json'
with open(filename) as f:
    bemor = json.load(f)
pprint(bemor)
#Navbat pprint() funksiyasiga:
#pprint(bemor)
```

### 17.1-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
<pre>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&lt;end:  curr_sum=sorted_nums[i]+sorted_nums[start]+sorted_nums[end]                 diff=abs(curr_sum- target)                  if diff==0:                     return curr_sum                 if diff&lt;min_diff:                     min_diff=diff                     result=curr_sum                 if curr_sum&lt;=target:                     start+=1             else:</pre>	<p>2</p> <p>-4</p> <p>5</p>

```

                                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))

```

### Mustaqil bajarish uchun topshiriqlar:

1. Pprint moduli yordamida lug'at, ichki lug'at, lug'atlar lug'atlari ro'yxatini chop etish uchun Python dasturini yozing.
2. Pprint moduli yordamida lug'atni chop etishdan oldin uning kalitlarini saralash uchun Python dasturini yozing.
3. Pprint moduli yordamida ro'yxatni, lug'atni chop etishda chiqish kengligini belgilaydigan Python dasturini yozing.
4. Pprint moduli yordamida ichki lug'atni chop etishda chiziqni ko'rsatadigan Python dasturini yozing. Muharrirga boring "indent" (standart 1) har bir uyalash darajasi uchun qo'shilgan chuqurlik miqdorini belgilaydi.
5. Pprint () funksiyasidan foydalangan holda loyiha (PyPI-dan) haqida ma'lumot olish uchun Python dasturini yozing.
6. Pprint () funksiyasidan foydalangan holda loyiha (ya'ni PyPI) haqida ma'lumot olish uchun Python dasturini yozing. Natijani ma'lum darajada cheklang va tarkibning kengligini belgilang.

## 18-MAVZU. PYTHON STANDART KUTUBXONASI

### 18.1-masala

Kuchli parolni tekshirish uchun Python dasturini tuzing

#### Dastur kodi

```
import re
andoza = '^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[#?!@$%^&*~]).{8,}$'
msg = "Yangi parol kiriting"
msg += '(kamida 8 belgidan iborat, kamida 1 ta lotin bosh harf, 1 ta kichik harf, '
msg += '1 ta son va 1 ta maxsus belgi bo'lishi kerak):'
while True:
    password = input(msg)
    if re.match(andoza, password):
        print("Maxfiy so'z qabul qilindi")
        break
    else:
        print("Maxfiy so'z talabga javob bermadi")
```

#### Dastur natijasi

```
Yangi parol
kiriting(kamida
8 belgidan
iborat, kamida
1 ta lotin bosh
harf, 1 ta
kichik harf, 1
ta son va 1 ta
maxsus belgi
bo'lishi
kerak):
Aer34%s2
Maxfiy so'z
qabul qilindi
```

### Mustaqil bajarish uchun topshiriqlar:

1. Belgilangan raqamning keyingi va oldingi palindromlarini topish uchun Python dasturini yozing.
2. Floatni kasrga aylantirish uchun Python dasturini yozing.
3. Kvadratik funksiyaning ildizlarini topish uchun Python dasturini yozing.

4. O'nli raqamni ikkilik raqamga aylantirish uchun Python dasturini yozing.
5. E qiymatini hisoblash uchun Python dasturini yozing (2.718281827...) cheksiz qatordan foydalaning

## 19-MAVZU. PYTHONDA MODULLAR (MATH, DATA TIME MODULLARI)

Python kodlarini o'z ichiga olgan fayllar **Python modullari** deb ataladi. Ushbu kod sinflar, funksiyalar yoki o'zgaruvchilar bo'lishi mumkin va kerak bo'lganda oldindan belgilangan funksiyalarni taqdim etish orqali dasturchi vaqtini tejaydi. Bu bajariladigan kodni o'z ichiga olgan ".py" kengaytmali fayl. Odatda ishlatiladigan qurilgan modullar quyida keltirilgan:

- os
- sys
- data time
- math
- random
- JSON

### math —matematik funksiyalar kutubxonasi

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

$\pi$  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.

### 19.1-masala

Kvadratik funksiyaning ildizlarini topish uchun Python dasturini yozing.

#### Dastur kodi

```
from math import sqrt
print("Kvadrat tenglama : (a * x^2) + b*x + c=0")
a = float(input("a: "))
b = float(input("b: "))
c = float(input("c: "))
d = b**2 - 4*a*c
if d > 0:
    ildiz = 2
    x1 = ((-b) + sqrt(d)) / (2*a)
```

#### Dastur natijasi

```
Kvadrat
tenglama : (a *
x^2) + b*x +
c=0
a: 2
b: -5
c: 2
2 ta yechimga
ega: 2.000000
va 0.500000
```

```

        x2 = (((-b) -
sqrt(d))/(2*a))
        print("2 ta yechimga ega:
%f va %f" % (x1, x2))
elif d == 0:
    ildiz = 1
    x = (-b) / 2*a
    print("1 yechimga ega: ",
x)
else:
    ildiz = 0
    print("Ildizi yo`q,
chunki D < 0")
    exit()

```

### 19.2-masala

Math modulidan foydalanib, konusning yasovchisi, to`la sirti va hajmini topish uchun Python dasturini tuzing.

#### Dastur kodi

#### Dastur natijasi

```

import math
r = float(input('Radius: '))
h = float(input('Balandlik:
'))
#yasovchi
l = math.sqrt(r * r + h * h)
#To`la sirti
S = math.pi * r * (r + l)
# Hajm
Hajm = (1.0/3) * math.pi * r
* r * h
print("Konus yasovchisi =
%.2f " %l)
print("To`la sirti = %.2f "
%S)
print("Hajm = %.2f" %Hajm)

```

```

Radius: 5
Balandlik: 12
Konus
yasovchisi =
13.00
To`la sirti =
282.74
Hajm = 314.16

```

### **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.

### **19.3-masala**

Berilgan ikki yil oralig'ida necha marta oyning 1-kuni dushanbadan boshlanishini hisoblash Python dasturini yozing.

#### **Dastur kodi**

```
import datetime
from datetime import datetime
year1=int(input("year1= "))
year2=int(input("year2= "))
monday1 = 0
months = range(1,13)
for year in range(year1,year2):
    for month in months:
        if datetime(year, month,
1).weekday() == 0:
            monday1 += 1
print(monday1)
```

#### **Dastur natijasi**

```
year1= 2020
year2= 2022
4
```



### 19.4-masala

Tugʻilgan kunga necha kun qolganini aniqlash dasturini tuzing.

#### Dastur kodi

#### Dastur natijasi

```
import time
from datetime import date
today = date.today()
my_birthday = date(today.year, 7, 10)
if my_birthday < today:
    my_birthday =
my_birthday.replace(year=today.year+1)
time_to_birthday = abs(my_birthday -
today)
print(time_to_birthday.days, "kun
qoldi")
```

95 kun  
qoldi

#### Mustaqil bajarish uchun topshiriqlar:

1. Satrni datetimega aylantirish uchun Python dasturini yozing.
2. Pythonda joriy vaqtni olish uchun Python dasturini yozing.
3. Joriy sanadan besh kun oldingi sanani chop etish uchun Python dasturini yozing.
4. Belgilangan yil va oyning oxirgi kunini olish uchun Python dasturini yozing.
5. Berilgan oy va yildagi kunlar sonini olish uchun Python dasturini yozing.
6. Muntazam ko'pburchakning maydonini hisoblash uchun Python dasturini yozing.
7. Kubik tenglamaning ildizlarini topish uchun Python dasturini yozing.
8. O'nli raqamni ikkilik raqamga aylantirish uchun Python dasturini yozing.
9. Kompleks sonning haqiqiy va mavhum qismlarini chop etish uchun Python dasturini yozing.

## 20-MAVZU. PYTHONDA MODULLAR (RANDOM MODULI)

**Random moduli** - tasodifiy o'zgaruvchilarni yaratish uchun o'rnatilgan moduldir. Bu tasodifiy bir qator olish, ro'yxatdan tasodifiy elementlarni tanlash kabi tasodifiy harakatlarni amalga oshirish mumkin. Random modulda bir qator funksiyalar mavjud:

`getrandbits ()` - tasodifiy bitlarni ifodalovchi raqamni qaytaradi

`randrange ()` - berilgan qator o'rtasida tasodifiy raqamni qaytaradi

`randint ()` - berilgan qator o'rtasida tasodifiy raqamni qaytaradi

`shuffle ()` - ketma-ketlikni oladi va ketma-ketlikni tasodifiy tartibda qaytaradi

`sample()` - ketma-ketlikning berilgan namunasini qaytaradi

`random ()` - 0 va 1 o'rtasida tasodifiy float qator qaytadi

**copy moduli**- Belgilash operatori obyektidan 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 obyektни yaratadi va keyin (agar iloji bo'lsa) unga asl nusxadagi obyektlarga havolalarni kiritadi.

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

### 20.1-masala

Massiv elementlarini ixtiyoriy to'ldirib, so'ngra tartiblash uchun Python dasturini tuzing.

**Dastur kodi**

**Dastur  
natijasi**

<pre> from array import* import random  n=int(input("n=")) nlist = [] for i in range(n):  nlist.append(random.randint(1,100)) print(nlist) nlist.sort(reverse=False) print(nlist) </pre>	<pre> n=4 [61, 72, 27, 50] [27, 50, 61, 72] </pre>
--	--

## 20.2-masala

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

### Dastur kodi

### Dastur natijasi

<pre> import random nums = [1, 2, 3, 4, 5] print("Original ro`yxat:") print(nums) random.shuffle(nums) print("Shuffle ro`yxat:") print(nums) words = ['qizil', 'qora', 'yashil', 'ko`k'] print("\nOriginal ro`yxat:") print(words) random.shuffle(words) print("Shuffle ro`yxat:") print(words) </pre>	<pre> Original ro`yxat: [1, 2, 3, 4, 5] Shuffle ro`yxat: [4, 2, 1, 3, 5]  Original ro`yxat: ['qizil', 'qora', 'yashil', 'ko`k'] Shuffle ro`yxat: ['ko`k', 'qora', 'qizil', 'yashil'] </pre>
--	---

### 20.3-masala

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

Dastur kodi	Dastur natijasi
<pre>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) 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)</pre>	<pre>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]] 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]]</pre>

## 20.4-masala

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

Dastur kodi	Dastur natijasi
<pre>import copy son1 = [1, [2, 3, 4]] print("Original ro`yxat: ", son1) son2 = copy.deepcopy(son1) print("Berilgan ro`yxatning chuqur nusxasi:") print(son2) print("Original ro`yxat o`zgartirildi:") son1[1][1] = 10 print(son1) print("2-ro`yxatning nusxasi (Chuqur nusxa):") print(son2) sonlar1 = [[1, 2, 3], [4, 5, 6]] sonlar2 = copy.deepcopy(sonlar1) print("Original ro`yxat:") print(sonlar1) print("Berilgan ro`yxatning chuqur nusxasi:") print(sonlar2) print("Original ro`yxat o`zgartirildi:") sonlar1[0][2] = 55 sonlar1[1][1] = 77 print("Original ro`yxat:") print(sonlar1) print("2-ro`yxatning nusxasi (Chuqur nusxa):") print(sonlar2)</pre>	<pre>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]] 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:</pre>

	<pre>[[1, 2, 55], [4, 77, 6]] 2-ro'yxatning nusxasi (Chuqur nuxa): [[1, 2, 3], [4, 5, 6]]</pre>
--	---

### **Mustaqil bajarish uchun topshiriqlar:**

1. 0 va 70 ikki butun sonlar orasidagi tasodifiy qiymati 7 qadam orqali topish uchun Python dasturini yozing.
2. Katalogdan ro'yxat, to'plam, lug'at qiymati va fayldan tasodifiy elementni tanlash uchun Python dasturini yozing.
3. Tasodifiy alifbo belgilarini, alifbo satrlarini va belgilangan uzunlikdagi alifbo satrlarini yaratadigan Python dasturini yozing.
4. 0 va 6 (6 dan tashqari) sonlar o'rtasida tasodifiy butun sonlarni bir qadam bilan hosil qilish uchun bir Python dasturini yozing.
5. 0 dan 1 gacha float yaratish uchun Python dasturini yozing va ma'lum bir diapazonda tasodifiy float hosil qiling.
6. Tasodifiy butun sonlar ro'yxatini yaratish va ushbu ro'yxatdan bir nechta elementlarni tasodifiy tanlash uchun Python dasturini yozing.

## 21-MAVZU. PYTHONDA SINFLAR VA OBYEKTLAR

Python **obyektga yo'naltirilgan dasturlash tilidir**. Sinf obyekt konstruktoriga yoki obyektlarni yaratish uchun "reja" ga o'xshaydi. Sinf yarating Sinf yaratish uchun **class** kalit so'zidan foydalaning:

MyClass nomli sinf yarating, x o'zgaruvchi qo'shing:

```
class MyClass:  
    x = 5
```

P1 nomli obyekt yarating va x qiymatini chop eting:

```
p1 = MyClass()  
print(p1.x)
```

Sinflarning ma'nosini tushunish uchun biz o'rnatilgan `__init__()` funksiyasini tushunishimiz kerak. Barcha sinflar `__init__()` deb nomlangan funktsiyaga ega, bu sinf boshlanganda har doim bajariladi. Ob'ekt xususiyatlariga qiymatlarni belgilash uchun `__init__()` funksiyasidan yoki ob'ekt yaratilayotganda bajarilishi kerak bo'lgan boshqa operatsiyalardan foydalaning:

Misol Shaxs nomli sinf yarating, ism va yosh uchun qiymatlarni belgilash uchun `__init__()` funksiyasidan foydalaning:

```
class Person:  
    def __init__(self, name, age):  
        self.name = name  
        self.age = age
```

```
p1 = Person("John", 36)
```

```
print(p1.name)  
print(p1.age)
```

`__Str__()` funksiyasi sinf obyektini satr sifatida ko'rsatilganda nimani qaytarish kerakligini boshqaradi. `__Str__()` funksiyasi o'rnatilgan bo'lsa, obyekt string natija qaytaradi.

```
class Person:  
    def __init__(self, name, age):  
        self.name = name  
        self.age = age
```

```
p1 = Person("John", 36)
```

```
print(p1)
```

### 21.1-masala

Satrnı soʻzma-soʻz oʻzgartirish uchun Python sinfini yozing. Satrnı soʻzma-soʻz teskari aylantiring.

#### Dastur kodi

#### Dastur natijasi

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

```
tili
dasturla
sh
Python
```

### 21.2-masala

Get\_String va print\_String ikkita usuliga ega Python sinfini yozing. get\_String foydalanuvchidan satrnı qabul qiladi va print\_String satrnı 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())

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

```
python dasturlash
tili
PYTHON DASTURLASH
TILI
```



### 21.3-masala

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

Dastur kodi	Dastur natijasi
<pre>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())</pre>	216

### 21.4-masala

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

Dastur kodi	Dastur natijasi
<pre>class Circle():     def __init__(self, r):         self.radius = r      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())</pre>	314.0 62.8000000000000004

### 21.5-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
<pre>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))</pre>	<pre>True False True False Sinfning sinfostilarini tekshiring. True True</pre>

### **Mustaqil bajarish uchun topshiriqlar:**

1. Pi qiymatini n kasr soniga olish uchun Python loyihasini yarating.
2. Qiymatini olish uchun Python loyihasini yarating e ga n o'nlik kasrlar soni. Eslatma: raqamni kiriting va dastur ' n-raqamga e hosil qiladi.
3. Tasodifiy tanlangan raqamni taxmin qilish uchun Python loyihasini yarating.
4. Terminal oynasida rangli, qalin, kursiv, xira, miltillovchi (sekin/tez) matnni chop etishga yordam beradigan Python loyihasini yarating.
5. Nuqta yo'li yordamida ichki ob'ektdan xususiyat olish uchun Python loyihasini yarating.
6. Berilgan ma'lumotlar ro'yxatini bir nechta kichik bo'limlarga bo'lish uchun Python loyihasini yarating.
7. Tasodifiy ism (ism, to'liq ism va boshqalar) yaratish uchun Python loyihasini yarating. Profil va boshqalar.
8. Qiymatlar ro'yxatida oddiy statistikani bajarish uchun Python loyihasini yarating.

## 22-MAVZU. PYTHONDA SINFLAR VA OBYEKTLAR

### 22.1-masala

Qavslar qatori

'(', ')', '{', '}', '[' va ']' qavslar qatorining haqiqiylikini 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

<pre>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  print(py_solution().is_valid_parenthese( "(){}[]")) print(py_solution().is_valid_parenthese( "() [{} ]")) print(py_solution().is_valid_parenthese( "() ") )</pre>	<pre>True False True</pre>
---	----------------------------

### 22.2-masala

n ta butun sonli massiv va natija berilgan bo'lsa, i indeksli uchlik sonini toping,  
 $i, j, k$  sonlar  $i + sonlar[j] + sonlar[k] < natija$   
 $0 \leq i < j < k < n$   
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
<pre>class Solution(object):     def threeSumSmaller(self, nums, target):     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 &lt; end:  curr_sum = sorted_nums[i] + sorted_nums[start] + sorted_nums[end]                 if curr_sum == target:                     end -= 1                 elif curr_sum &lt; target:                     start += 1                 elif curr_sum &gt; target:                     end -= 1</pre>	<pre>[(-4, -1, 2), (- 4, 1, 2)] 2</pre>

```

        print(triplet_list)
        return
len(triplet_list)

if __name__=="__main__":

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

```

### Mustaqil bajarish uchun topshiriqlar:

1. Belgilangan kompaniyalarning joriy aktsiyalar narxini olish uchun Python loyihasini yarating.
2. Nomi va nashr etilgan yili, jurnalning hajmi va sahifalaridan foydalangan holda Google scholar-dan iqtibos olish uchun Python loyihasini yarating.
3. Berilgan shaharning ob-havo ma'lumotlarini ko'rsatish uchun Python loyihasini yarating.
4. Belgilangan belgi uchun kundalik munajjimlar bashoratini ko'rsatish uchun Python loyihasini yarating.
5. Belgilangan URL manzilidan sayt elektron pochta xabarlarini olish uchun Python loyihasini yarating.
6. BBC yangiliklarini olish uchun Python loyihasini yarating. Dan JSON formatida maqolalar ro'yxatini olib newsapi.org.
7. Haqiqatan ham veb-saytdan ish unvoni va joylashuvini olish uchun Python loyihasini yarating.

## 23-MAVZU. VORIS TUSHUNCHASI

**Voris** bir nechta sinflar uchun umumiy bo'lgan xatti-harakatlarni ajratib ko'rsatish va uni alohida shaxsga o'tkazish imkonini beradi. Yangi muammolarni hal qilish uchun mavjud koddan foydalanish, ya'ni voris kodni qayta ishlatish vositasidir! Voris eski sinfdan biroz farq qiladigan yangi sinfni olish imkonini beradi.

Voris mavjud sinf asosida yangi sinf yaratishga imkon beradi. Inkapsulyatsiya bilan bir qatorda, voris obyektga yo'naltirilgan dasturlashning asoslaridan biridir.

Vorisning asosiy tushunchalari subklass va superklass. Subklass superklassdan barcha ommaviy atributlar va usullarni voris qilib oladi. Super sinf, shuningdek, base (base class) yoki ota - ona (parent class), pastki sinf esa hosila (derived class) yoki bola (child class) deb nomlanadi.

Sinflarni voris qilib olish uchun sintaksis quyidagicha:

class subclass (superklass):

    subclass\_metodlari

Masalan, bizda insonni xususiyatlarini ko'rsatuvchi Person classi mavjud:

```
class Person:
    def __init__(self, name):
        self.__name = name    # inson nomi
    def name(self):
        return self.__name
    def display_info(self):
        print(f"Name: {self.__name} ")
```

Aytaylik, bizga ma'lum bir korxonada ishlaydigan ishchilar sinfi kerak. Biz noldan yangi sinf yaratishimiz mumkin, masalan, xodimlar sinfi:

```
class Employee:

    def __init__(self, name):
        self.__name = name

    def name(self):
        return self.__name
```

```

def display_info(self):
    print(f"Name: {self.__name} ")
def work(self):
    print(f"{self.name} works")

```

Biroq, xodimlar sinfi shaxs sinfi bilan bir xil atributlar va usullarga ega bo'lishi mumkin, chunki ishchi shaxsdir. Shunday qilib, yuqoridagi ishchi sinfida faqat works usuli qo'shiladi, qolgan barcha kodlar shaxs sinfining funktsional imkoniyatlarini takrorlaydi. Ammo bir sinfning funktsional imkoniyatlarini boshqasida takrorlamaslik uchun, bu holda merosni qo'llash yaxshiroqdir.

Shunday qilib, biz xodimlar sinfini shaxs sinfidan meros qilib olamiz:

```

class Person:
    def __init__(self, name):
        self.__name = name # odam nomi
    def name(self):
        return self.__name
    def display_info(self):
        print(f"Name: {self.__name} ")
class Employee(Person):
    def work(self):
        print(f"{self.name} works")
tom = Employee("Tom")
print(tom.name) # Tom
tom.display_info() # Name: Tom
tom.work() # Tom works

```

Xodimlar sinfi faqat work () usulini qo'shib, shaxs sinfining funktsionalligini to'liq qabul qiladi. Shunga ko'ra, Employee obyektini yaratishda biz Persondan meros bo'lib qolgan konstruktordan foydalanishimiz mumkin:

```
tom = Employee("Tom")
```

Shuningdek, meros qilib olingan atributlar / xususiyatlar va usullarga murojaat qilish mumkin:

```

print(tom.name) # Tom
tom.display_info() # Name: Tom

```



Ammo shuni ta'kidlash kerakki, `__name` kabi yopiq atributlar xodimlar uchun mavjud emas. Masalan, biz ish usulida shaxsiy `self.__name` atributiga murojaat qila olmaymiz:

```
def work(self):  
    print(f"{self.__name} works")    # ! Xato
```

### **Mustaqil bajarish uchun topshiriqlar:**

1. Shaxs nomli sinf yarating, ism va yosh uchun qiymatlarni belgilash uchun `__init__()` funksiyasidan foydalaning.
2. “Salom” satrini chop etadigan funksiyani joylashtiring va uni pl obyektida bajaring.
3. Ism va familiya atributlari va chop\_etish funksiyadan iborat shaxs nomli sinf yarating.
4. `super ()` funksiyasi orqali talaba sinfiga shaxs sinfidan barcha usullar va xususiyatlarni meros qilib oling.
5. Talabalar sinfiga bitiruv yili deb nomlangan atributni qo'shing.

## 24-MAVZU. VORIS TUSHUNCHASI (KO'P MEROSHO'RLIK)

Ko'p meros Python tilining o'ziga xos xususiyatlaridan biri bu ko'p merosni qo'llab-quvvatlashdir, ya'ni bitta sinfni bir nechta sinflardan meros qilib olish mumkin:

```
# Ishchi sinfi
class Employee:
    def work(self):
        print("Employee works")

# talaba sinfi
class Student:
    def study(self):
        print("Student studies")

class WorkingStudent(Employee, Student):
    # Ishchi va talaba sinfidan meros olish
    pass

# Ishlaydigan talaba sinfi
tom = WorkingStudent Employee ()
tom.work()          # ishlaydi
tom.study()         # talaba o`qiydi
```

Bu yerda firma xodimini ifodalovchi Employee sinfi va talaba sinfini ifodalovchi Student klassi aniqlanadi. Ishlayotgan talabani ifodalovchi ishchi talabalar sinfi hech qanday funksiyani aniqlamaydi, shuning uchun u pass operatorini belgilaydi. Ishchi talabalar sinfi faqat ikkita ishchi va talaba sinflaridan funksiyalarni meros qilib oladi. Shunga ko'ra, ushbu sinf obyektida biz ikkala sinfning usullarini chaqirishimiz mumkin.

Shu bilan birga, meros qilib olingan sinflar funksional jihatdan murakkabroq bo'lishi mumkin, masalan:

```
class Employee:
    def __init__(self, name):
        self.__name = name
    def name(self):
```

```

        return self.__name
    def work(self):
        print(f"{self.name} works")

class Student:
    def __init__(self, name):
        self.__name = name
    def name(self):
        return self.__name
    def study(self):
        print(f"{self.name} studies")

class WorkingStudent(Employee, Student):
    pass

tom = WorkingStudent("Tom")
tom.work()          # Tom ishlaydi
tom.study()         # Tom o`qiydi

```

### Mustaqil bajarish uchun topshiriqlar:

1. Ikkita 2 o'lchovli shakldan meros qilib, 3D shaklni ifodalovchi sinf yarating.
2. Asosiy sinfda 4 asosiy matematik amallarni (+, -, \*, /) bajaradigan funksiyalar yaratib, uni merosxo`r sinf orqali ishlating.
3. Hayvon nomli asosiy sinfning barcha atributlari va funksiyalarini it va quyon nomli merosxo`r sinflarida ishlating.

## 25-26-MAVZULAR. GRAFIK MODULAR BILAN ISHLASH

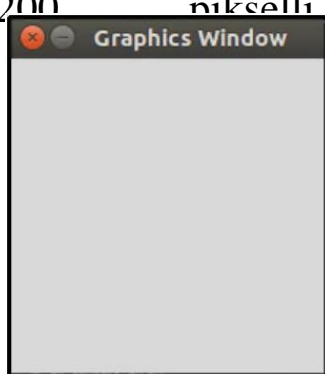
Grafika ko'p odamlar uchun dasturlashni yanada qiziqarli qiladi. Ushbu bo'limda John Zelle tomonidan Python dasturlash kitobida foydalanish uchun ishlab chiqilgan soddalashtirilgan grafik modul taqdim etilgan. Grafika yaratish bosqichlari:

- Grafika sinfidan foydalanish uchun kutubxonani Import qiling

- Oyna obyektini yarating
- Yaratilgan oynada obyektlarni chizish.
- Oynani yoping.

```
step 1: from graphics import *
step 2: win = GraphWin()
step 3: pt = Point(100, 50) #100 indicates
horizontal coordinate and 50 indicates
vertical coordinate pt.draw(win)
step 4: win.close()
```

Grafik oyna **GraphWin()** usuli yordamida boshqa oyna ostida yangi oyna yaratadigan oyna ichida grafik yaratishingiz mumkin. Ushbu oyna sukut bo'yicha "grafik oyna" deb nomlanadi va standart 200 X 200 pikseli Oynani yaratadi.



25.1-rasm Grafika oynasi

**GraphWin()** usuli uchta argumentni olishi mumkin:

- Grafik oyna uchun sarlavha;
- X-koordinata(grafik oynaning uzunligi);
- Y-koordinata(grafik oynaning balandligi).

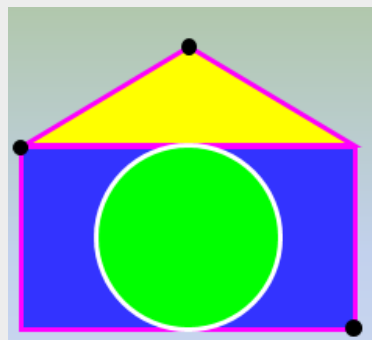
### 25.1-masala

Grafika yordamida uy rasmini hosil qiling.

#### Dastur kodi

```
from graph import *  
penColor("magenta")  
brushColor("blue")  
rectangle(100,100,300,200)  
brushColor("yellow")  
polygon([(100,100),  
        (200,50),  
        (300,100), (100,100)])  
penColor("white")  
brushColor("green")  
circle(200, 150, 50)  
run()
```

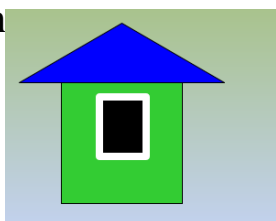
#### Dastur natijasi



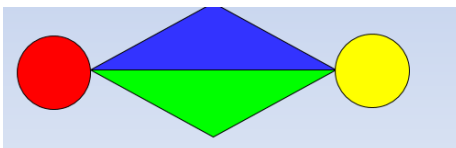
#### Mustaqil bajarish uchun topshiriqlar:

Grafika yordamida quyidagi shakllarni Python dasturlash tilida chizin

a)



b)



## 27-MAVZU. TURTLE MODULI

**Toshbaqa grafikasi**-bu bolalarga dasturlashni tanishtirishning mashhur usuli. Bu asl nusxaning bir qismi edi. Dasturlash tili logotipi Uolli Fyurzeyg, Seymour Papert va Sintiya Sulaymon 1967 yilda tomonidan ishlab chiqilgan.

Masalan, XY tekisligida (0, 0) boshlanadigan robot toshbaqani tasavvur qiling. import turtle keyin unga turtle.forward(15) buyrug'ini bering va u harakat qiladi (ekranda!). U qaragan yo'nalishda 15 piksel, harakatlanayotganda chiziq chizadi. turtle.right(25) va u joyida soat yo'nalishi bo'yicha 25 daraja aylanadi.

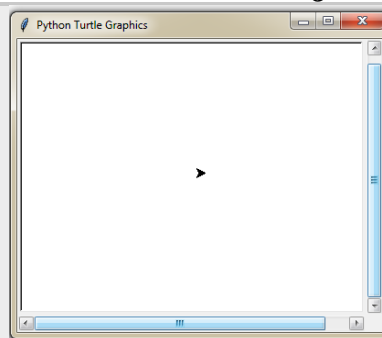
### 27.1-masala

400x320 ekranda toshbaqani hosil qiling.

#### Dastur kodi

```
from turtle import *
t = Turtle()
t.screen.setup(400,
320)
t.screen.exitonclick(
)
t.screen.mainloop()
```

#### Dastur natijasi



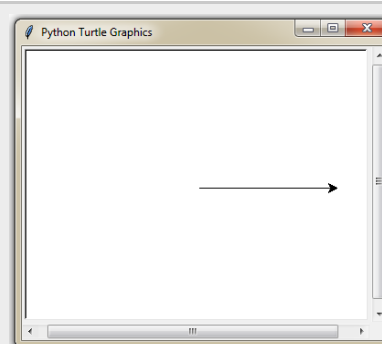
### 27.2-masala

Toshbaqa o'ng tomonga harakatini hosil qiling.

#### Dastur kodi

```
from turtle import *
t = Turtle()
t.screen.setup(400,
320)
t.fd(150)
t.screen.exitonclick(
)
t.screen.mainloop()
```

#### Dastur natijasi



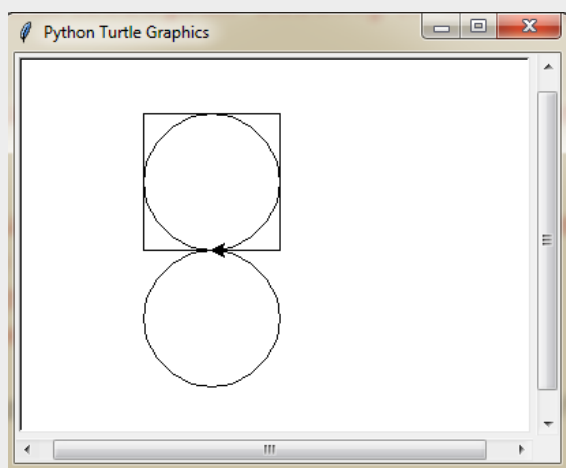
### 27.3-masala

Turtle moduli yordamida kvadrat va ikki aylanani chizing.

### Dastur kodi

```
from turtle import *
t = Turtle()
t.screen.setup(400,
300)
def sq_cr(side):
    for i in
range(4):
        t.left(90)
        t.fd(side)
        t.bk(side / 2)
        t.circle(side /
2, 360)
        t.left(180)
        t.circle(side /
2, 360)
sq_cr(100)
t.screen.exitonclick(
)
t.screen.mainloop()
```

### Dastur natijasi



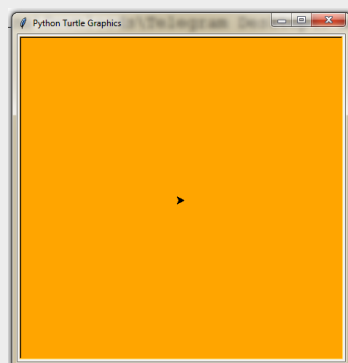
## 27.4-masala

Turtle moduli yordamida fonni sabzirangga bo'yash uchun dastur tuzing.

### Dastur kodi

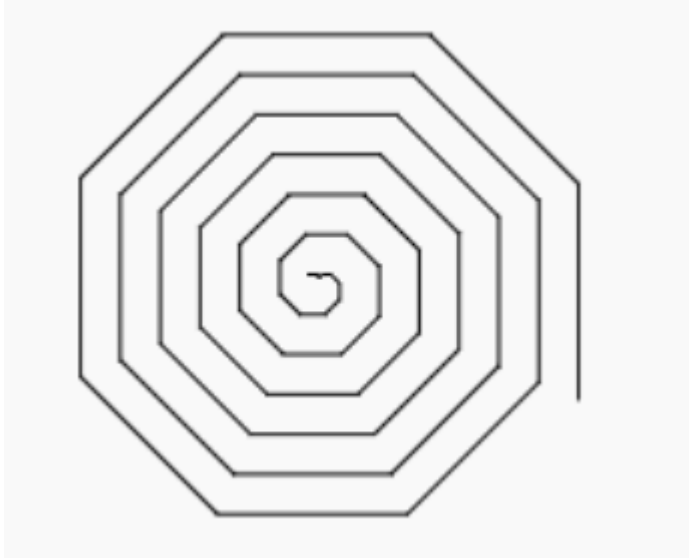
```
from turtle import *
t = Turtle()
t.screen.setup(450,
450)
t.screen.bgcolor("orange")
t.screen.exitonclick(
)
t.screen.mainloop()
```

### Dastur natijasi

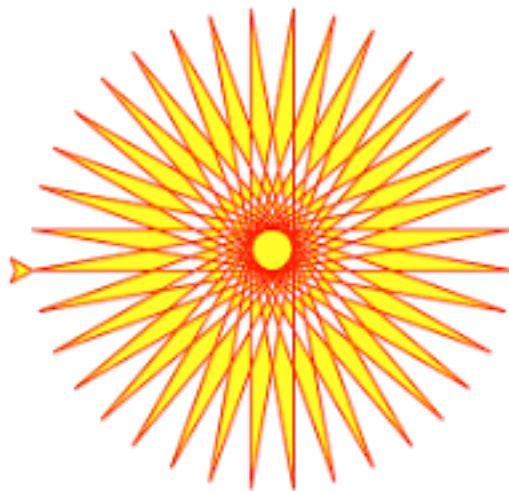


### Mustaqil bajarish uchun topshiriqlar:

1. Turtle moduli yordamida quyidagi shaklni chizing.



2. Turtle moduli yordamida quyidagi shaklni chizing.





## 28-MAVZU. TURTLE MODULI

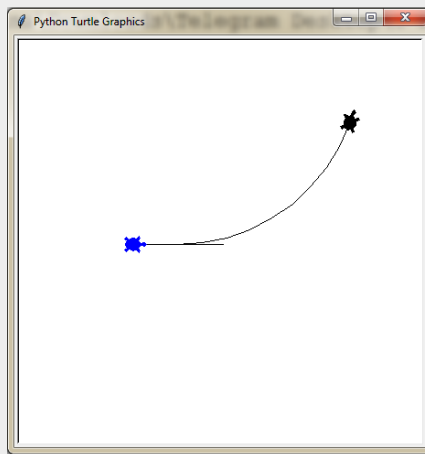
### 28.1-masala

Turtle moduli yordamida toshbaqani yoy bo'yicha harakatlantirish uchun dastur tuzing.

#### Dastur kodi

```
from turtle import *
t = Turtle()
t.screen.setup(450,
450)
t.shape("turtle")
t.goto(-100, 0);
t.color("blue")
t.stamp()
t.color("black")
t.up()
t.fd(50)
t.down()
t.circle(200, 70)
t.screen.exitonclick(
)
t.screen.mainloop()
```

#### Dastur natijasi



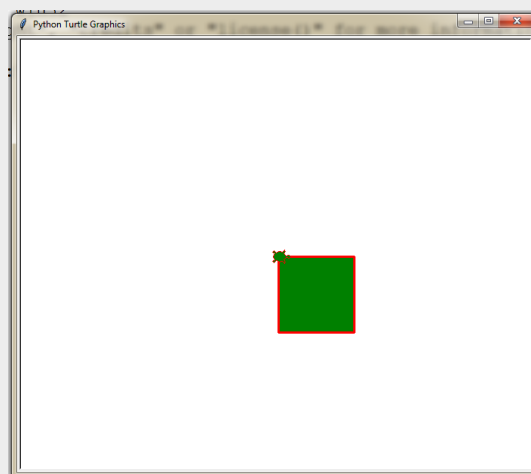
### 28.2-masala

Turtle moduli yordamida rangli kvadrat chizish uchun dastur tuzing.

#### Dastur kodi

```
from turtle import *
reset()
shape("turtle")
color("red", "green")
pensize(3)
begin_fill()
for i in range(4):
    fd(100)
    rt(90)
end_fill()
```

#### Dastur natijasi



### 28.3-masala

Turtle moduli yordamida qizil va sariq yoqlar yordamida rangli doira chizish uchun dastur tuzing.

#### Dastur kodi

```
import turtle
turtle.bgcolor("black")
)
turtle.tracer(50)
turtle.pensize(7)
turtle.goto(-40,-300)
c = ['red', 'yellow']
turtle.rt(4)
for i in range(300):

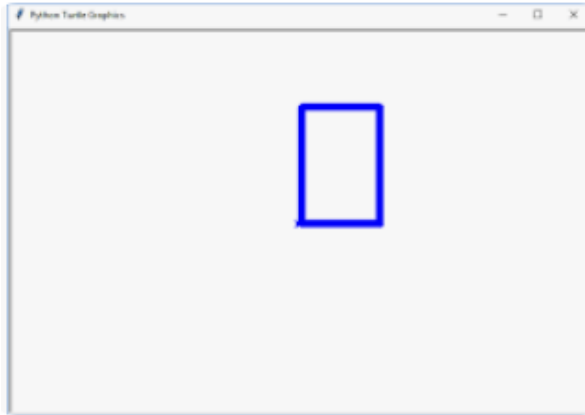
    turtle.color(c[i%2])
        turtle.up()
        turtle.circle(300-
i, -92)
        turtle.down()
        turtle.circle(300-
i, -92)
turtle.done()
```

#### Dastur natijasi



## Mustaqil bajarish uchun topshiriqlar:

1. Turtle moduli yordamida quyidagi shaklni chizing.



2. Turtle moduli yordamida quyidagi shaklni chizing.



## 29-MAVZU. TKINTER MODULI (PYTHON TKINTER BASIC)

**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
<b>Label</b>	Yangi oynaga matnni ko`rsatish yoki rasm qo`yish uchun ishlatiladi.
<b>Button</b>	Tugma yaratadi. Uni nomlash va bosilganda ma`lum amalni bajarishi mumkin.
<b>Entry</b>	1 qator matn kiritish vidjeti
<b>Text</b>	Ko`p qatorli matn kiritish vidjeti
<b>Frame</b>	Vidjetlarni guruhlash yoki vidjetlar orasiga masofa qo`yish uchun to`g`ri to`rtburchakdan iborat maydon

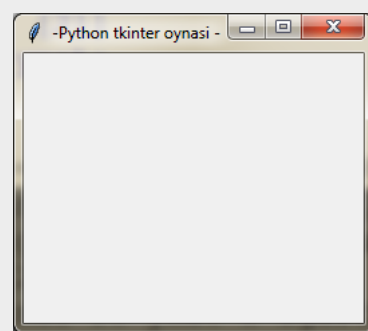
### 29.1-masala

Tkinter paketini import qilish va oyna yaratish va uning sarlavhasini o'rnatish uchun Python GUI dasturini yozing.

#### Dastur kodi

```
import tkinter as tk
# misol yaratish
parent = tk.Tk()
# Sarlavha qo`shish
parent.title("-Python tkinter oynasi -")
# GUI ni ishga tushirish
parent.mainloop()
```

#### Dastur natijasi



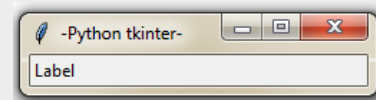
### 29.2-masala

Tkinter paketini import qilish va oyna yaratish uchun Python GUI dasturini yozing. Uning sarlavhasini o'rnatish va oynaga yorliq qo'shing.

### Dastur kodi

```
import tkinter as tk
parent = tk.Tk()
parent.title("-Python tkinter-")
my_label = tk.Label(parent,
text="Label")
my_label.grid(column=0, row=0)
parent.mainloop()
```

### Dastur natijasi



### Mustaqil bajarish uchun topshiriqlar:

1. Yorliq yaratish va tkinter moduli yordamida yorliq shrift uslubini (shrift nomi, qalin, o'lcham) o'zgartirish uchun Python GUI dasturini yozing.
2. Oyna yaratish uchun Python GUI dasturini yozing va tkinter moduli yordamida standart oyna hajmini o'rnatish.
3. Tkinter moduli yordamida oyna o'lchamini o'zgartirish uchun oyna yaratish va o'chirish uchun Python GUI dasturini yozing.

## 30-MAVZU. TKINTER MODULI (PYTHON TKINTER WIDGETS)

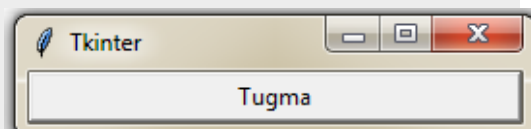
### 30.1-masala

Tkinter moduli yordamida bir tugmani kiritish uchun Python GUI dasturini yozing.

#### Dastur kodi

```
import tkinter as tk
parent = tk.Tk()
parent.title('Tkinter')
my_button =
tk.Button(parent,
text='Tugma', height=1,
width=35,
command=parent.destroy)
my_button.pack()
parent.mainloop()
```

#### Dastur natijasi



### 30.2-masala

Tkinter moduli yordamida tasdiqlash tugmasi vidjetini yaratish uchun Python GUI dasturini yozing.

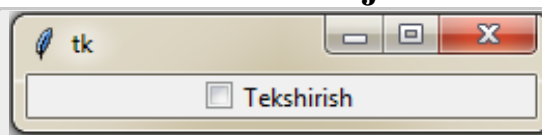
#### Dastur kodi

```
import tkinter as tk
from tkinter import ttk

root = tk.Tk()
my_boolean_var =
tk.BooleanVar()

my_checkbutton =
ttk.Checkbutton(
    text="Tekshirish",
    variable=my_boolean_var
)
my_checkbutton.pack()
```

#### Dastur natijasi



### **Mustaqil bajarish uchun topshiriqlar:**

1. Tkinter moduli yordamida Ikki tugma yaratish uchun Python GUI dasturini yozing. (salom va chiqish)
2. Tkinter moduli yordamida uchta variant bilan Combobox yaratish uchun Python GUI dasturini yozing.
3. Tkinter moduli yordamida matnli vidjet yaratish uchun Python GUI dasturini yozing. Boshiga satr kiriting, so'ngra joriy matnga satr kiriting. Matnning birinchi va oxirgi belgilarini o'chiring.
4. Tkinter moduli yordamida bir ScrolledText vidjetlarni yaratish uchun Python GUI dasturini yozing.
5. Tkinter moduli yordamida Listbox bar vidjetlarni yaratish uchun Python GUI dasturini yozing.

## XULOSA

Xulosa o`rnida shuni aytish mumkinki, Python dasturlash tilida masalalar va ularning yechimlari o`quv qo`llanmasi kelajakda 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 tili” 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.



## FOYDALANILGAN ADABIYOTLAR VA INTERNET RESURSLARI 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. O.M. Shukurov, F.Q. Qoraev, E.A.Eshboyev, B.H.Shovaliev “Dasturlashdan masalalar to‘plami”, Metodik qo‘llanma, Toshkent-2008.
6. <https://metanit.com/python>
7. <https://pythonworld.ru/tipy-dannyx-v-python/slovari-dict-funkcii-i-metody-slovarej.html>
8. <https://ai.mohirdev.uz/>
9. <https://www.w3resource.com/>
10. <https://www.sololearn.com/>
11. <https://leetcode.com/>
12. <https://www.hackerrank.com/>
13. <https://www.tiobe.com/tiobe-index/>
14. [acmp.ru](http://acmp.ru)
15. <https://docs.python.org/>





**M.A.BOBOJONOVA**

**PYTHON DASTURLASH TILI**

**O`QUV QO`LLANMA**

<i>Muharrir:</i>	<i>A. Qalandarov</i>
<i>Texnik muharrir:</i>	<i>G. Samiyeva</i>
<i>Musahhih:</i>	<i>Sh. Qahhorov</i>
<i>Sahifalovchi:</i>	<i>M. Bafoyeva</i>

Nashriyot litsenziyasi AI № 178. 08.12.2010. Original-maketdan bosishga ruxsat etildi: 19.07.2023. Bichimi 60x84. Kegli 16 shponli. «Times New Roman» garn. Ofset bosma usulida bosildi. Ofset bosma qog`ozi. Bosma tobog`i 6,7. Adadi 100. Buyurtma №380.

“Sadriddin Salim Buxoriy” MCHJ  
“Durdona” nashriyoti: Buxoro shahri Muhammad Iqbol ko`chasi, 11-uy.  
Bahosi kelishilgan narxda.

“Sadriddin Salim Buxoriy” MCHJ bosmaxonasida chop etildi.  
Buxoro shahri Muhammad Iqbol ko`chasi, 11-uy. Tel.: 0(365) 221-26-45