

```

print("bauhan academy")
print("berk batuhan devran")

#kaçış karakteri
print("18\\12\\1999")
print("istanbul\\nankara\\nizmir") # \n = new line
print("Berk Batuhan Devran")
print("Berk\\tBatuhan\\tDevran")
print('batuhan')
print("berk\\batuhan") # \\ ile kaçtık
print('batuhan\\'ın melekleri')

# matematiksel ifadeler
print(5+9)
print(11-5)
print(15/3)
print(15//3) # iki kez bölme işareti bölümün sadece tam kısmını gösterir
print(4*9)
print(15/2+(5*7)-9)

# Üs alma (**)
print(3**3)
print(2**10)
print(1024**0.1)

# Mod alma işlemi (%)
print(100%3)
print(101%4)
print(16%7.5)

#Özel durumlar
print(3.1+4.2)
print(0.1+2.2)

print("Berk Batuhan" "Devran")
print("Berk Batuhan", "Devran")
print("Berk Batuhan"+"Devran")
print("Batuhan "*3)

# r= raw string
print("C:\\Users\\batuhan")
print(r"C:\Users\batuhan")

print("Berk" "\n" "Devran")
print("Berk" +"\n"+ "Devran")
print(3*"Batuhan\n")

# ,end="..."
print("Berk Batuhan",end="")
print("DEVKAN")
print("Berk Batuhan",end="XXX")
print("DEVKAN")

# ,sep=""
print("Berk", "Batuhan", "Devran", sep="")
print("Berk", "Batuhan", "Devran", sep=" ")
print("Berk", "Batuhan", "Devran", sep="***")
print("Berk", "Batuhan", "Devran", sep="***")
print("Berk", "Batuhan", "Devran", sep="/")
print(19,19,1919, sep="/")

```

```
print("019","019","01919",sep="/")
print("019","019","01919",sep=":")

# *
print(*"BATUHAN")
print(*"BATUHAN",sep="/")
print(*"BATUHAN",sep=".")

# .format metodu

print("{}".format(10))
print("{}{}".format(10,25))
print("{} {}".format(10,25))
print("{}+{}={}".format(10,25,10+25))
print("{}+{}={}".format("berk","batuhan","devran"))

# comments yorumlar
# single line comments not= ctrl+/ maus ile seçilen kısmı yorum satırı
yapar
#print("batuhan")

# multiline comments
"""
print(10+9)
print(14+5)
"""

# TODO print ile alakalı çalışılacak
# TODO hesap makinesi ile ilgili çalışılacak

# help fonksiyonu

help(print)
help(help)
```

```

sayi1 = 1400
sayi2 = 412
print(sayi1+sayi2)
print(sayi1-sayi2)
print(sayi1*sayi2)
print(sayi1/sayi2)
print(sayi1%sayi2)

sayi1 = 5
sayi2 = 15

print(sayi1+sayi2)
print(sayi1-sayi2)
print(sayi1*sayi2)
print(sayi1/sayi2)
print(sayi1%sayi2)

#değişkenler alttre yada alfabe harferiyle başlar
__batuhan = "deneme"
print(__batuhan)

# değişken nasıl oluşturulur? print("değişken ismi".isidentifier())
print("batuhan".isidentifier())
#önceden tanımlanmış anahtar kelimeleri değişken olarak atayamazsın.

sayi1 = 159
sayi2 = 12
print(sayi1+sayi2)
print(sayi1-sayi2)
print(sayi1*sayi2)
print(sayi1/sayi2)
print(sayi1%sayi2)

sayi1 = 1000
print("sayı1",sayi1)

# değişkenler nerede depolanır?
sayi1 = 500
sayi2 = 500

print(sayi1)
print(id(sayi1))
print(id(sayi2))
print(id(500))
#python interpreter inde bütün sayılar depolanmıştır bu nedenle bir sayıya
değişken atandığında python interpreterindeki
#adres ram e iletilir sayı ve değişken aynı adreste depolanır

#everything is an object
tamsayi = 10
kesirliSayi = 15.5
print(type(tamsayi)) #int-> integer -> tamsayı
print(type(kesirliSayi)) #float-> kesirli sayı

print(tamsayi*3)
print(tamsayi+kesirliSayi)
format("{}+{}={}".format(tamsayi,kesirliSayi,tamsayi+kesirliSayi))

```

```

#str --> string -> karakter dizisi
isim = "batuhan"
print(type(isim))
print(id(isim))
isim = "berk"
print(id(isim))

print("berk batuhan devran")
isim = "berk batuhan"
soyisim = "devran"
print(isim,soyisim)

isim = "batuhan"
soyisim = "devran"
print(isim, soyisim)

fullname = isim + " "+soyisim
print(fullname)

# len() boşluk dahil karakter sayısını verir
print(len(fullname))

# fullname[1,2,...]
fullname = "Berk Batuhan Devran"
print(fullname)
print(fullname[0])

print(fullname[1:4]) # 1dahil ve 4. karakter dahil değil
değişmişisim = 'M'+fullname[1:4]+'EL'
print(değişmişisim)

fullname = değişmişisim+" "+fullname[5:]
print(fullname)

# 0--->18
# -19---->-1
fullname = "Berk Batuhan Devran"
print(len(fullname)) # len()---> lenght
print(fullname[18])
print(fullname[-1])
print(fullname[-19])

# len----length

print(fullname[-len(fullname)])
print(fullname[-1])
print(fullname[len(fullname)-1])

print(fullname)
print(fullname[1:])
print(fullname[5:])
print(fullname[:])
print(fullname[0:5])
print(fullname[-19:-14])
print(fullname[-6:])

# atlamalı yazdırma

print(fullname[0:5:1])

```

```
print(fullname[0:5:2]) # ikişer atlamalı yazdırmak
print(fullname[0:18:2]) # ikişer atlamalı yazdırmak
```

```
# tersten yazdırma
```

```
print(fullname[::-1])
print(fullname[18::-1])
print(fullname[0:5])
print(fullname[3::-1])
```

```
import sys
sayi1 = 12
print(sys.getsizeof(sayi1))
sayi1 = 123
print(sys.getsizeof(sayi1))
sayi1 = 1234
print(sys.getsizeof(sayi1))
sayi1 = 12345
print(sys.getsizeof(sayi1))
sayi1 = 123456
print(sys.getsizeof(sayi1))
sayi1 = 1234567
print(sys.getsizeof(sayi1))
sayi1 = 12345678
print(sys.getsizeof(sayi1))
sayi1 = 1234567812345678
print(sys.getsizeof(sayi1))
sayi1 = 12345678123456781234567812345678
print(sys.getsizeof(sayi1))
print(sys.getsizeof(sayi1), "bytes")
```

```
sayi1 = 10
sayi2 = 10
sayi3 = 10
print("sayi1 adres: {} sayi2 adres: {} sayi3: {}"
      .format(id(sayi1), id(sayi2), id(sayi3)))
```

```
isim1 = "batuhan"
isim2 = "batuhan"
print("isim1:{} isim2:{}".format(id(isim1), id(isim2)))
```

```
sayi3 = 19
sayi4 = sayi3
print(id(sayi1), id(sayi2), id(19), sep="--")
del sayi3
del sayi4
```

```
# print(sayi3)
# print(sayi4)
```

```
##### KONU= Typ conversion (Tip dönüşümü)
```

```
#print(15+"naber")--> hatalı
print("15"+"naber")
print("yaşınız"+str(30)) # integer ifade string çevrildi
sayi1 = 5
sayi2 = 5.5
```

```

karakterler = "185"
print(int(karakterler)+sayı1)  # + ifadesi aynı tür karakterleri toplar
# integer e çevirilen ifadenin içinde sadece sayı olmalıdır
print(float(int(karakterler)+sayı1))
sayı1 = float(5)
print(sayı1)
sayı2 = int(5.5)
print(sayı2)
print(17//6)  # //=int()

print(3*"5")
print(3*str(5))
print(int(str(555)*2))
print(int(3*str(5))*3)
print(float(str("189.2"))*2)
## ALT*SHIFT yapıp tön tuşarıyla istenilen ifadeyi oynatabiliriz,
istenilirse bu maus ile seçim yapılarak çoklu da yapılabilir

#print(int("55.6"))  hatalı çünkü type dönüşümü yapılırken içeri
gerçekten dönüştürülen type ye uymalı burada içeri float

"""
isim = input("lütfen bir isim giriniz")
print("isminiz:",isim)

sayı1 = int(input("lütfen 1. sayıyı giriniz")) !! input() un içerisi str
olarak çıkar
sayı2 = int(input("lütfen 2. sayıyı giriniz"))
print("sonuç:",sayı1+sayı2)

karakter = input("lütfen erkekseniz E kadınsanız K harfine basınız")
print(karakter)
karakter = input("lütfen erkekseniz E kadınsanız K harfine basınız")[0]
print(karakter)
"""
#exception handling
try:
    sayı1 = int(input("lütfen 1. sayıyı giriniz"))
    sayı2 = int(input("lütfen 2. sayıyı giriniz"))
    print("sonuç:",sayı1+sayı2)
except:
    print("lütfen bir sayı değeri giriniz")

sayı1 = float(input("Lütfen 1.sayıyı giriniz"))
sayı2 = float(input("lütfen 2. sayıyı giriniz"))
sayı3 = float(input("lütfen 3. sayıyı girin"))

print("{}+{}+{}={}".format(sayı1,sayı2,sayı3,sayı1+sayı2+sayı3))
print("{}-{}-{}={}".format(sayı1,sayı2,sayı3,sayı1-sayı2-sayı3))
print("{}*{}*{}={}".format(sayı1,sayı2,sayı3,sayı1*sayı2*sayı3))
print("{}{/{}{/{}={}".format(sayı1,sayı2,sayı3,sayı1/sayı2/sayı3))
print("{}({}+{}+{})/3={}".format(sayı1,sayı2,sayı3,(sayı1+sayı2+sayı3)/3))
##round()

print("{}({}+{}+{})/3={}".format(sayı1,sayı2,sayı3,round((sayı1+sayı2+sayı3)/
3,2)))

```

```

sayı1 = 140
sayı2 = 14
print(sayı1-sayı2)
print(sayı1+sayı2)
print(sayı1*sayı2)
print(sayı1/sayı2)
print(sayı1%sayı2)

sayı1 = 19
sayı2 = 1919
print(sayı1-sayı2)
print(sayı1+sayı2)
print(sayı1*sayı2)
print(sayı1/sayı2)
print(sayı1%sayı2)

# değişkenler harf ya da alttrem ile başlarlar

# true ya da false
print("123isim".isidentifier())
print("isim123".isidentifier())
print("print".isidentifier())
"""her ne kadar print fonksiyonuna true dese de bu bizim print fonksiyonuna
yai değeri atayabileceğimiz
anlamına gelmez. Bu nedenle pythonda daha önce tanımlanan fonksiyon ya da
metodlara değeri atayamayız."""

sayı1 = 19
sayı2 = 1919
print("çıkarna", sayı1-sayı2)
print("toplama", sayı1+sayı2)
print("çarpma", sayı1*sayı2)
print("bölme", sayı1/sayı2)
print("mod alma", sayı1%sayı2)

# id() değişkenin depolandığı adresi sorgulama
sayı1 = 500
sayı2 = 500
print(sayı1)
print(id(sayı1))
print(id(sayı2))

# type() değişkenin tipini sorgulamaya yarar
tamsayı = 19
kesirliSayı = 19.19
print(type(tamsayı)) # int--> integer tamsayı
print(type(kesirliSayı)) # float---> kesirli sayı
print(19*3)
print(tamsayı*3)
print(tamsayı+kesirliSayı)
print("{}+{}={}".format(tamsayı, kesirliSayı, kesirliSayı+tamsayı))

# string ---> str
isim = "Berk Batuhan"
soyisim = "Devran"
print(isim, soyisim)

fullname = isim+ " "+soyisim
print(fullname)

```

```

# len fonksiyonu
print(len(fullname))

fullname = "Berk Batuhan Devran"
print(fullname[0])
print(fullname[18])
# fullname[0] = "M" bu şekilde işlem yapılamıyor
print(fullname[0:9]) # 0 dahil 9 dahil değil

yeniisim= "M"+fullname[1:3]+"T"
print(yeniisim)

yenifullname = yeniisim+fullname[4:19]
print(yenifullname)
yenifullname = yeniisim+fullname[4:]
print(yenifullname)

# 0123... = -... -5,-4,-3,-2,-1
print(fullname[-1])
print(fullname[-19])
print(len(yenifullname))

#len--> length
print(fullname[-len(fullname)])
print(fullname[len(fullname)-1])

print(fullname[:])
print(fullname[0:18:2])
print(fullname[-19:])
print(fullname[0:4:1])
print(fullname[0:4:])
print(fullname[0:4:2])
print(fullname[:-1])
print(fullname[18::-1])

```



```
ad = "Berk Batuhan"
soyad = "Devran"
yaş = 21
print("{} , {} , {}".format("ad", "soyad", "yaş"))
print("{} , {} , {}".format(ad, soyad, yaş))
```

```
takim1 = "FENERBAHÇE"
takim2 = "GALATASARAY"
yenitakim1 = takim1[0:5]+" "+takim1[5:]
yenitakim2 = takim2[0:6]+" "+takim2[6:]
print(yenitakim1)
print(yenitakim2)
print(takim1,takim2, sep="\n")
```

```
ad = "berk batuhan"
soyad = "devran"
yaş = 21
yaşş = yaş+1

print(ad, soyad, yaş, yaşş)
print(id(ad), id(soyad), id(yaş), id(yaşş))
print(id(21))
print(id(22))
```

```
sayı1 = 16
sayı2 = 19
toplama = sayı1 + sayı2
çıkarma = sayı1 - sayı2
çarpma = sayı1 * sayı2
bölme = sayı1 / sayı2
tambölme = sayı1 // sayı2
modalma = sayı1 % sayı2
üs alma = sayı1 ** sayı2
```

```
del sayı1
del sayı2
print(toplama, çıkarma, çarpma, bölme, tambölme, modalma, üs alma, sep=" ")
```

```
şimdikizaman = 2020
doğum = 1999
print("yaş: {}".format(şimdikizaman-doğum))
```

```
sayı1 = float(input("Lütfen 1.sayıyı giriniz"))
sayı2 = float(input("Lütfen 2. sayıyı giriniz"))
sayı3 = float(input("Lütfen 3. sayıyı girin"))
try:
    print("{}+{}+{}={}".format(sayı1, sayı2, sayı3, sayı1+sayı2+sayı3))
    print("{}-{}-{}={}".format(sayı1, sayı2, sayı3, sayı1-sayı2-sayı3))
    print("{}*{}*{}={}".format(sayı1, sayı2, sayı3, sayı1*sayı2*sayı3))
    print("{} / {} / {} = {}".format(sayı1, sayı2, sayı3, sayı1/sayı2/sayı3))

print("({}+{}+{})/3={}".format(sayı1, sayı2, sayı3, (sayı1+sayı2+sayı3)/3))
```

```
print("({}+{}+{})/3={}".format(sayı1, sayı2, sayı3, round((sayı1+sayı2+sayı3)/3, 2)))
```

```
except ValueError:  
    print("l\u00fctfen bir say\u0131 de\u011feri giriniz")
```

```

"""Arithmetic operators"""
a=4
b=3
print(a+b) #addition
print(a-b) #subtraction
print(a*b) #multiplication
print(a/b) #division
print(a%b) #Moduls
print(a**b) #Exponensiations
print(a//b) #Floor division

"""Assignment Operators"""
#   = , +=, -=, *=, /=, %=, //=, **=

c = 5
isim = "batuhan"
print(c,isim)

c = c+3
print(c)

c += 3
print(c)

d = 10
print(d)
d = d-3
print(d)
d -= 3
print(d)

e = 15
print(e)
e *=4
print(e)

g = 75
g %=15
print(g)

h = 5
h **=2
print(h)

# temp

"""sayı1 = 10
sayı2 = 20
print("sayı1 {} sayı2 {}".format(sayı1,sayı2))
temp = sayı2
sayı2 = sayı1
sayı1 = temp
print("sayı1 {} sayı2 {}".format(sayı1,sayı2))

# temp kullanmadan
sayı1 = 10
sayı2 = 20
print("sayı1 {} sayı2 {}".format(sayı1,sayı2))
sayı1,sayı2 = sayı2,sayı1
print("sayı1 {} sayı2 {}".format(sayı1,sayı2))

```

```
# Örnek
#Kare ve dikdörtgenin alanını hesaplayan program
kenar1 = float(input("lütfen hesaplamak istediğiniz dikdörtgenin 1.
kenarını giriniz."))
kenar2 = float(input("lütfen hesaplamak istediğiniz dikdörtgenin 2.
kenarını giriniz."))
print("dikdörtgenin alanı: {}".format(kenar1*kenar2))
print("dikdörtgrnin çevresi: {}".format(2*(kenar1+kenar2)))
```

```
# Örnek
#Çemberin alanı ve çevresini bulan program yazınız
r = float(input("çemberin yarıçapını giriniz"))
print("çevre:",2*(3.14)*r)
print("alan:", (3.14)*r**24)
```

```
# Örnek
#Silindirin hacmini hesaplayan program
r,h = float(input("lütfen silindirin taban yarıçapını
giriniz")),float(input("lütfen silindirin yüksekliğini giriniz"))

print("Hacim:", (3.14)*r*r*h)
```

```
# Örnek
#ikinci dereceden denklemlerin köklerini bulan program

a = float(input("kareli terimin katsayısını giriniz"))
b = float(input("x li terimin katsayısını giriniz"))
c = float(input("x^0 li terimin katsayısını giriniz"))
delta = b**2-4*a*c
print("a*x*x+b*x+c")
print(delta)
print("Delta sıfırdan küçükse kökler sanaldır")
print("kök1:", (-b-(delta**0.5))/2*a)
print("kök2:", (-b+(delta**0.5))/2*a)
```

```
# Örnek
#Beş basamaklı sayının rakam değerleri toplamı
a = float(input("1.basamak"))
b = float(input("2.basamak"))
c = float(input("3.basamak"))
d = float(input("4.basamak"))
e = float(input("5.basamak"))
print("rakamlar toplamı:",a+b+c+d+e)
"""
```

```
"""Comparison Operators"""
```

```
# == != < > <= >=
```

```
a = 10
b = 15
c = 20
d = 20
print(a==b)
print(a!=b)
print(c == d)
print(c != d)
print(a<b)
```

```

print(a>b)
print(a>=b)
print(a<=b)
print("{} > {}".format(a,b),a>b)

isim = "batuhan"
print(isim=="batuhan")
print(isim=="berk")
print("A">"B") # Harfleri 1 den başlayarak numaralandırdığını düşün
print("batuhan">"berk")
print("devran"<"berk")
print("F">"f") # büyük harfler küçük harflerden önce gelir #ascii tablosunu
incele
print("Batuhan"<"batuhan")

# .lower() , .upper() # bütün harfleri büyük ya da küçük harflere çevirir
print("BATUHAN".lower())
print("Batuhan".lower()<"batuhan")
print("Batuhan".lower()=="batuhan")
print("Batuhan"<"batuhan") # alt tuşuna basıp mausla yeni yerler seçersen
imleç sayısını artırabilirsin

# Logical Operators
# OR Tablosu (veya)
# OR | k1 k2 | s
#-----
# | 0 0 | 0
# | 0 1 | 1
# | 1 0 | 1
# | 1 1 | 1

k1 = 5<6
k2 = 7<8
print(k1,k2)
print(k1 or k2)

a,b = 10,15
print(a<15 or b<20)
print(a>15 or b<20)
print(a<15 or b>20)
print(a>15 or b>20)

# AND tablosu ( ve )
# AND | k1 k2 | s
#-----
# | 0 0 | 0
# | 0 1 | 0
# | 1 0 | 0
# | 1 1 | 1

k1 = 5<6
k2 = 7<8
print(k1,k2)
print(k1 and k2)

a,b = 10,15
print(a<15 and b<20)
print(a>15 and b<20)
print(a<15 and b>20)
print(a>15 and b>20)

```

```

# NOT (değili) (True) != false (False) != True
anahtar = True
print("anahtar", anahtar)
anahtar = not anahtar
print("anahtar", anahtar)

# 0=false 0 dışındaki her değer true
sayı1 = 0
print(not sayı1)
sayı2 = 5
print(not sayı2)
#bool()
print(bool(-1))
print(bool(0))
print(bool(1.1))
print(bool("ahmet")) # içi dolu str ler true boş olnalar false, içinde
boşluk olanlar true
print(bool(""))
print(bool(" "))

# KISA DEVRE
# ASSOCIATIVITY
# OR Tablosu (veya)
# OR | k1 k2 | s
#-----
# | 0 0 | 0
# | 0 1 | 1
# | 1 0 | 1
# | 1 1 | 1

# AND tablosu ( ve )
# AND | k1 k2 | s
#-----
# | 0 0 | 0
# | 0 1 | 0
# | 1 0 | 0
# | 1 1 | 1

print(2<3 or 3<4 or 4<5 or 5<6)
print(2<3 or 3<4 or 4<5 or 5<6)
print(2>3 or 3>4 or 4>5 or 5>6)
print(2>3 or 3>4 or 4>5 or 5>6)
print(2<3 or 3>4 or 4<5 or 5<6)

print(2<3 and 3<4 and 4<5 and 5<6)
print(2<3 and 3<4 and 4<5 and 5<6)
print(2>3 and 3>4 and 4>5 and 5>6)
print(2>3 and 3>4 and 4>5 and 5>6)

sayı = int(input("ltfen 1-15 arasında sayı giriniz"))
sonuç = sayı>=11 and sayı<=15
print(sonuç)
# ÖRNEK
# Kullanıcı giriş belgelerini sorgulama

"""bir davette içeriye giriş için bir şifre belirlenmiştir. Bu şifre
"BATUHAN ACADAMY" dir. Ayrıca davete giriş için

```

kişinin 18 yaşı dahil olmak üzere 18-30 yaş arasında olması gerekmektedir. Kullanıcıdan yaş,şifre ve veli bilgilerini alarak u kullanıcının içerigirip giremeyeceğins gözlemleyiniz."""

```
"""şifre = input("şifreyi giriniz")
yaş = int(input("yaşınızı giriniz"))
sonuç = (şifre=="BATUHAN ACADEMY" and 18<=yaş<30) or (şifre=="BATUHAN
ACADEMY" and yaş<30 and veli=="evet")
veli = input("Veliniz yanınızda mı?")
print(sonuç)"""
```

```
""" UNARY OPERATORS (TEK TERİMLİ OPERATÖRLER) """
```

```
# + -
```

```
print(1++++++2)
```

```
sayı=++++5
```

```
print(sayı)
```

```
sayı=-sayı
```

```
print(sayı)
```

```
sayı--sayı
```

```
print(sayı)
```

```
""" IDENTITY OPERATORS (BENZERLİK OPERATÖRLERİ) """
```

```
#type(tip) ve value(değer) kontrolü
```

```
print(5 == 5)
```

```
#print(5 is 5)
```

```
print(type(5) is type(5) and 5==5)
```

```
#print(5 = 5.0)
```

```
#print(5 is 5.0)
```

```
print(type(5) is type(5.0) and 5 == 5.0)
```

```
print((2<3) == True)
```

```
print((2<3) is True)
```

```
print(5 == 5)
```

```
print(5 is not 5)
```

```
print(type(5) is not type(5) or 5 != 5)
```

```
print(5 == 5.0)
```

```
print( 5 is not 5.0)
```

```
print(type(5) is not type(5.0) or 5 != 5.0)
```

```
print(type(16) is type(16) and 16==16)
```

```
print(type(16) is type(16.0) and 16==16.0)
```

```
print(type(16) is type(15) and 16==15)
```

```
## is yerine isinstance kullan
```

```
print(isinstance(99,int))
```

```
print(isinstance(19.0,float))
```

```
print(isinstance(16,type(16)) and 16==16)
```

```
print(isinstance(16,type(16.0)) and 16==16.0)
```

```
print(isinstance(16,type(15)) and 16==15)
```

```
print(type(16) is not type(16) or 16!=16)
```

```
print(type(16) is not type(16.0) or 16!=16.0)
```

```
print(type(16) is not type(15) or 16!=15)
```

```
print(not isinstance(16,type(16)) or 16!=16)
```

```

print(not isinstance(16,type(16.0)) or 16!=16.0)
print(not isinstance(16,type(15)) or 16!=15)

#kullanıcıdan iki kelime alıp identitiy operatörlerle kıyasla

k1,k2 = input("1.Kelime") , input("2.Kelime") # tip+değer >is değer>==
print(k1==k2)
print(k1 is k2)
print(k1 != k2)
print(k1 is not k2 )

"""Bitwise operators"""

print(3&2) #bitwise and
print(9 | 14) #bitwise or
print(~2) #bitwise not (compliment) -A=A'+1
print(6^5) #bitwise xor "aynıysa 0 farklıysa 1
print(7>>2) #bitwise right shift
print(3<<2) #bitwise rihgt shift

#Membership Operators
liste = [1,2,3]
print(liste)
print(1 in liste)
print(4 in liste)
print(1 not in liste)
kelimelistesi = ["berk","batuhan","devran"]
print("berk" in kelimelistesi)
print("batuhan" not in kelimelistesi)

"""NUMBER SYSTEM CONVERSION"""

#Binay(ikilik taban)
print(bin(11))
print(bin(2))
print(bin(2*2))
print(bin(2*2*2*2))

#Decimal(onluk taban)
print(10)
print(10*10)
print(10*10*10)

#Octal(sekizlik taban)
print(oct(26))
print(oct(8))
print(oct(8*8))
print(oct(8*8*8))
print(oct(8*8*8*8))

#Hexadecimal(on altılık taban)
print(hex(11))
print(hex(12))
print(hex(13))
print(hex(14))
print(hex(15))
print(hex(16))

```



```
print(0b10)
print(0o10)
print(0x10)
```

#örnek:Kullanıcının girdiği sayıyı 2, 8 ve 16 tabanına karşılık gelen değerlerin çıktısını verenprogramı yaz!

```
sayı=int(input("lütfen bir sayı giriniz"))
print("binary {}".format(bin(sayı)))
print("octal {}".format(oct(sayı)))
print("hexadecimal {}".format(hex(sayı)))
```

```

"""None"""
sehir = None #BİR DEĞER VAR AMA BİZ O DEĞERİ BİLMİYORUZ
print("Değer", sehir, "Tip", type(sehir))

sehir = "istanbul"
print("Değer", sehir, "Tip", type(sehir))

"""Numeric(sayısal)"""
#integer
#float
#complex
#bool

say1 = 5
print(type(say1))
say1 = 2.5
print(type(say1))
say1 = 2+7j
print(type(say1))
say1 = True
print(type(say1))

say11 = 7.5
say12 = int(say11)
print(say11, type(say11))
print(say12, type(say12))

say13 = 10
say14 = float(say13)
print(say14, type(say14))

x = int(10)
y = float(15.0)
z = complex(2, 2)
t = bool(True)
print(x, y, z, t)

print(3<4)
anahtar = 3<4
print(anahtar, type(anahtar))

print(int(True))
print(int(False))
print(float(True))
print(float(False))

print(bool(10))
print(bool(1.2))
print(bool(-9))
print(bool(-8.9))

"""DİZİLER"""
#List
#Mutable(değiştirilebilir)
#Elemanların sıra numarası(sequence number) vardır
#[]

liste = [2, 6, 7, 9]
print(liste, type(liste))
print(*liste)

```

```

print(liste[3])
print(liste[3]*9)

liste = []
print(liste)
liste = list()
print(liste)
liste = list([1, 2, 3])
print(liste)

liste2 = [2, 3, 4]
print(liste2)
liste2[1]=15
print(liste2)

sayılar =[13, 16, 47, 79, 87, 99]
print(sayılar)
print(sayılar[0])
print(sayılar[4])
print(sayılar[-1])
print(len(sayılar))
print(sayılar[len(sayılar)-1])

isim = "berk batuhan"
isim = list(isim)
print(isim)
isimler = ["berk", "batuhan", "devran", "adı lazım değil"]
print(isimler)
print(isimler[-1])
print(isimler[len(isimler)-1])

karışıktpıçerenliste = [33, 45, "batuhan", True, None, {2,5}, {"595" :
"devran"}]
print(karışıktpıçerenliste)
print(karışıktpıçerenliste[-1])
print(type(karışıktpıçerenliste[-1]))
print(karışıktpıçerenliste[-1]["595"])
bilgiler = [isimler, sayılar]
print(bilgiler)
bilgiler = isimler+sayılar
print(bilgiler)
print(4*isimler)

isimler[2] = "fenerbahçe"
print(isimler)

sayılar[:2] = []
print(sayılar)
sayılar.append(123)
print(sayılar)
sayılar.insert(2,19)
print(sayılar)

sayılar.pop() # son elemanı siler
print(sayılar)
print(sayılar.pop()) #silinen elemanı yazdırır
x=sayılar.pop()
print(x)
print(sayılar.pop(2))
print(sayılar)
del sayılar[:2]

```

```

print(sayılar)
# del sayılar          sayıların tamamını siler

sayılar.extend([5, 12, 14])
print(sayılar)
sayılar.extend({5, 12, 14})
print(sayılar)
sayılar.extend("batuhan")
print(sayılar)
sayılar.extend({"batuhan": "595"})
print(sayılar)

sayılar = [1, 6, 0, 19, 67, 99]
print(min(sayılar))
print(max(sayılar))
print(sum(sayılar)) #toplamı verir
sayılar.sort(reverse=True)
print(sayılar)
print(sayılar[::-1])
sayılar.reverse()
print(sayılar)

#Tuple
#Immutable (değiştirilemez)
#Elemanların sıra numarası (sequence number) vardır
#()

tup = (4, 24, -67, 90)
print(tup, type(tup))
tup2 = ()
print(tup2)
tup3 = tuple((4, 6))
print(tup3)
print(tup[2])

tup = (14, 99, 67, 14)
print(tup)
print(type(tup))
print(tup[1])
print(tup[::1])
print(tup[::2])
print(tup[1::2])
print(tup.count(14))
print(tup.index(14))

tup9 = (1, 3, 5)
tup10 = (0, 7, "batuhan")
tup11 = tup9 + tup10
print(tup11)

#Set
#Mutable (değiştirilebilir)
#Elemanlarının sıra numarası (sequence number) yoktur
#İçerisinde aynı değerde elemanlar barındırmaz

bosSet = set()
print(type(bosSet))
bosSet2 = {}
print('set değil dict olur;', type(bosSet2))

```

```

set1 = {3, 5, 7, 9}
print(set1)
set1.add(111)
print(set1)
set1.remove(3)
print(set1)

set1 = {5, 7, 0, 9}
print(set1)
set1.remove(5)
print(set1)
set1.update({13, 18, 10})
print(set1)
set1.add(999)
print(set1)
harfler = set("batuhan")
print(harfler)
set2 = {9999, 66, 77, 0}
unionset = set2.union(set1)
print(unionset)

#String(karakter dizileri)
#immutable(değiştirilemez)
#Elemanların sıra numarası(sequence number) vardır.
# "", ''

karakter = "F"
print(karakter, type(karakter))
isim = "Batuhan"
print(isim[4])
"""isim[4] = "k"
print("isim")
"""# değiştirilemez

#Range
print(range(10))
print(list(range(0,10)))
print(list(range(1,10,2)))
print(*range(0,5))

#Dictionary(sözlük)
#mutable(değiştirilebilir)
#Elemanların sequence numberi yerine "KEY"leri vardır
#{'', ''}

bosSözlük = dict()
print("BOSSÖZLÜK;", bosSözlük, type(bosSözlük))
bosSözlük2 = {}
print("BOSSÖZLÜK2;", bosSözlük2, type(bosSözlük2))

sözlük = {'12211212121' : 'Berk Batuhan Devran', '43434343434' : 'Adı lazım değil'}
print(sözlük)
print(sözlük['12211212121'])
sözlük['12211212121'] = 'Batuhan'
print(sözlük)
print(*sözlük)
print(sözlük.keys())

```

```

print(sözlük.values())
print(sözlük.get('12211212121'))

sozluk = {1: "batuhan", 2: "berk", 3: "devran"}
print(sozluk)
print(sozluk[1])
print(sozluk.get(10)) #none
print(sozluk.get(10, "eleman bulunamadı"))
print(sozluk.keys())
print(sozluk.values())
print(sozluk.items())

emptyDict = {}
emptyDict2= dict()
anahtarlar= {"batuhan", "berk", "devran"}
print(anahtarlar)
değerler = ["Elektrik MÜH", "KONROL MÜH", "HABERLEŞME MÜH"]
bilgiler = dict(zip(anahtarlar, değerler))
print(bilgiler)
bilgiler["berrak"] = "yazılım müh"
bilgiler["berk"] = "kontrol ve otomasyon müh"
print(bilgiler)
del bilgiler["berrak"]
dersler = {"berk": "müh mat", "batuhan": ["matlab", "phyton"],
"devran": ["elmak", "plc", "scada"]}
print(dersler)
print(dersler["devran"][2])

sözlük1 = {123:"berk"}
sözlük2 = {1234:"batuhan"}#key 123 olsaydı berk yerine batuhanı islerdi
sözlük1.update(sözlük2)
print(sözlük1)
sözlük3 = sözlük1
print(sözlük1)
sözlük4 = {**sözlük3}
print(sözlük4)

### MATH MODÜLÜ

import math
print(math.sqrt(25))
print(math.pow(2,5))
print(math.pi)
print(math.ceil(2.3))
print(math.floor(2.3))
print(math.sin(math.pi/2))
print(math.cos(math.pi))

from math import * # * yerine istenen fonksiyon eklenebilir
print(tan(pi/2))
print(e)
e = 9
print(e)

import math as m
print(m.cos(0))
print(m.cos(pi))
print(m.cos(pi/2))

```


#COMMAND PROMPT

```
import sys

"""print("batuhan academy")
isim = input(print("lütfen bir isim giriniz"))

print(isim)
"""
#dosyaIsmi = sys.argv[]
print(sys.argv)

"""x = int(sys.argv[1])
y = int(sys.argv[2])
z = int(sys.argv[3])"""
```



```
"""DEBUG (HATA, AYIKLAMA, ONARMA) İŞLEMLERİ"""
```

```
isimler = ["berk", "batuhan", "devran", "ayşe", "fatma", "hayriye"]  
isimler.remove("batuhan")  
isimler.remove("berk")  
print(isimler)
```

```
sayı1 = 15  
sayı2 = 17  
sayı3 = 99  
print(sayı1+sayı2+sayı3)
```

```
sayı2 = 10  
print(sayı1+sayı2+sayı3)
```

```
sayı3 = 94  
print(sayı1+sayı2+sayı3)
```

```
sayı1 = 11  
print(sayı1+sayı2+sayı3)
```

```
sayı2 = 1  
print(sayı1+sayı2+sayı3)
```

```
"""CONTROL STATEMENTS (IF ELIF ELSE)"""
```

```
#Indentation (GİRİNTİ)
```

```
# If li ifadeden sonra : koyulmalıdır alt satırda kod en az bir tane boşluk bırakılarak yazılmalıdır
```

```
# if den sonra yazılan kod dizisi aynı hizada olmalıdır
```

```
# eğer if in hemen altındaki satıra kod yazılmayacaksa pass yazılarak hata almak engellenmelidir
```

```
if False:
```

```
    print('2. if blogu')
```

```
sayi = 5
```

```
if sayi>0:
```

```
    print(sayi, "pozitiftir")
```

```
if sayi<0:
```

```
    print (sayi, "negatiftir")
```

```
if sayi == 0:
```

```
    print(sayi, "nötrdür")
```

```
yaş = int(input("lütfen yaşınızı giriniz"))
```

```
if yaş>18:
```

```
    print("reşitisiniz")
```

```
if yaş<18 and yaş>0:
```

```
    print("reşit değilsiniz")
```

```
if yaş<0:
```

```
    print("lütfen yaşınızı doğru giriniz")
```

```
sayı = 8
```

```
sonuç = sayı % 2
```

```
if sonuç==0:
```

```
    print("sayı çifttir")
```

```
if sonuç==1:
```

```
    print("sayı tektir")
```

```
if sonuç == 0:
```

```
    print("sayı çifttir")
```

```
else: # sonucun 0 olmadığı her durum
```

```
    print("sayı tektir")
```

```
if yaş>=18:
```

```
    print("reşitsiniz")
```

```
else:
```

```
    if yaş>0:
```

```
        print("reşit değilsiniz")
```

```
    else:
```

```
        print("lütfen yaşınızı doğru giriniz")
```

```
vize = 49
```

```
final = 50
```

```
if final>=50:
```

```
    ort = vize*0.4+final*0.6
```

```
    if ort>50:
```

```
        print("hadi yine iyisin")
```

```
    else:
```

```
        print("başka bahara koçum")
```

```
yaş = 22
```

```
if yaş>18:
```

```
    print("reşitsiniz")
```

```
elif yaş>0 and yaş<=18:
```

```
    print("reşit değilsiniz")
```

```

else:
    print("l\u00fctfen ya\u015fınızı do\u011fru giriniz")

"""\u00d6rnek"""
#Kullan\u0131c\u0131dan 1 ve 4 arasında bir tamsayı alan ve bu sayının nasıl okundu\u011funu g\u00f6steren
program\u0131 yazınız

say\u0131 = int(input("l\u00fctfen 1-4 arasında bir rakam giriniz"))
if say\u0131==1:
    print("bir")
elif say\u0131==2:
    print("iki")
elif say\u0131==3:
    print("\u00fc\u00e7")
elif say\u0131==4:
    print("d\u00f6rt")
else:
    print("l\u00fctfen istenilen aralı\u011fa uygun bir say\u0131 giriniz")

"""\u00d6rnek"""
# Girilen ayın hangi mevsimde oldu\u011funu g\u00f6steren program\u0131 yazınız

ay = input("l\u00fctfen bir ay ismi giriniz")
ay = ay.replace('I', 'i')
ay = ay.replace('i', 'i')
ay = ay.lower()
if ay=="aralık" or ay=="ocak" or ay=="\u015ubat":
    print("k\u0131\u015f")
elif ay=="mart" or ay=="nisan" or ay=="may\u0131s":
    print("ilkbahar")
elif ay=="haziran" or ay=="temmuz" or ay=="a\u011fustos":
    print("yaz")
elif ay=="eyl\u00fcl" or ay=="ekim" or ay=="kasım":
    print("sonbahar")
else:
    print("l\u00fctfen bir ay ismi giriniz")

"""\u00d6rnek"""
# 6 katlı bir binanın her katında 3 daire bulunmaktadır. Girilen daire numarasına gre o
dairenin kat numarasını veren program\u0131 yazınız.

daireno = int(input("l\u00fctfen 1-18 arası bir say\u0131 giriniz"))
if daireno==1 or daireno==2 or daireno==3:
    print(1)
elif daireno==4 or daireno==5 or daireno==6:
    print(2)
elif daireno==7 or daireno==8 or daireno==9:
    print(3)
elif daireno==10 or daireno==11 or daireno==12:
    print(4)
elif daireno==13 or daireno==14 or daireno==15:
    print(5)
elif daireno==16 or daireno==17 or daireno==18:
    print(6)
else:
    print("l\u00fctfen isneilen aralı\u011fa uygun bir say\u0131 giriniz")

"""\u00d6rnek"""
#pilotluk sınavında adayların ilk a\u015famayı ge\u00e7ebilmeleri i\u00e7in kadınların 1.60 erkeklerin
1.70 boy sınırını ge\u00e7meleri istenmektedir..

boy= int(input("l\u00fctfen boyunuzu giriniz"))
cinsiyet= str(input("l\u00fctfen cinsiyetinizi giriniz"))
if cinsiyet == "Kadın" :
    if boy>=160:

```

```

        print("Mülakatı geçtiniz")
    else:
        print("Elendiniz")
elif cinsiyet == "erkek" :
    if boy>=170:
        print("mülakatı geçtiniz")
    else:
        print("elendiniz")

"""Örnek"""
#Kullanıcıdan aldığı paket sayısına göre ödeyeceği tutarı gösteren programı yazınız.
#Bir toptan satış mağazası
# 100 paket ve üzeri için satın alımda paket başına 10 tl
# 61-99 paket arası satın alımda paket başına 12 tl
# 30-60 paket arası satın alımda paket başına 15 tl
# 1-29 paket arası satın alımda paket başına 20 tl alıyorsa... programı yazınız.

paketsayısı= int(input("kaç paket alacaksınız?"))
toplamtutar= None
if paketsayısı>=100:
    toplamtutar = paketsayısı*10
elif paketsayısı>=61:
    toplamtutar = paketsayısı*12
elif paketsayısı>=30:
    toplamtutar = paketsayısı*15
elif paketsayısı>=1:
    toplamtutar = paketsayısı*20
else:
    print("lütfen adam gibi bir sayı giriniz")

if toplamtutar is not None:
    print("Ödemeniz gereken toplam tutar {}".format(toplamtutar))
else:
    print("Ödemeniz gereken bir tutar bulunmamaktadır")

### Kullanıcı 3 kenar bilgisi girmeli ve eğer bu 3 kenarla bir üç kenarla bir üçgen
çizilebiliyorsa
# bu üçgenin, ikizkenar ya da çeşitkenar olup olmadığını konsolda gösteren programı
yazınız.

kenar1, kenar2, kenar3 = int(input("lütfen kenar 1 uzunluğu giriniz")),
int(input("lütfen kenar 2 uzunluğu giriniz")), int(input("lütfen kenar 3 uzunluğu
giriniz"))

if abs(kenar1-kenar2) < kenar3 and kenar3 and kenar3 < kenar1+kenar2:
    if kenar1 == kenar2 and kenar1== kenar3:
        print("Eşkenar Üçgen")
    elif kenar1 == kenar2 or kenar2== kenar3 or kenar1==kenar3:
        print("İkizkenar üçgen")
    else:
        print("Çeşitkenar üçgen")
else:
    print("Bu üç kenar ile bir üçgen çizilemez")

# Bir önceki örneğin set veri tipi ile çözülmesi

kenar1, kenar2, kenar3 = int(input("lütfen kenar 1 uzunluğu giriniz")),
int(input("lütfen kenar 2 uzunluğu giriniz")), int(input("lütfen kenar 3 uzunluğu
giriniz"))
kenarlar = set([kenar1, kenar2, kenar3]) # set bir değeri iki defa depolamaz
print(kenarlar)
if abs(kenar1-kenar2) < kenar3 and kenar3 and kenar3 < kenar1+kenar2:
    if len(kenarlar)==1:
        print("eşkenar üçgen")

```

```
elif len(kenarlar)==2:  
    print("ikizkenar üçgen")  
else:  
    print("çeşitkenar üçgen")
```

##ascii tablosu

```
print(ord("A"))
print(ord("a"))
print(ord("p"))
print(ord("ş"))
```

girilen harfin büyük ya da küçük olduğunu söyleyen program

```
harf = input("lütfen bir harf giiniz")[0]
print(harf)

if ord(harf)>= 65 and ord(harf)<= 90:
    print("{} büyük harftir".format(harf))
elif ord(harf)>= 97 and ord(harf)<= 122:
    print("{} küçük harftir".format(harf))
else:
    print("harf gir dedik ulan")
```

LOOPS

```
i=1
while i < 5:
    print("berk batuhan devran")
    i = i + 1
print("döğü tamamlandı",i)
for i in range(4):
    print("berk")
print("döğü tamamlandı",i)
```

```
for i in range(4):
    print(i)
    print("berk")
```

```
for i in [1, 6, 7, 12]:
    print(i)
    print("berk batuhan")
```

```
for i in [1, 2, 3, 4]:
    print(i*i)
```

"""while loop (while döngüsü)"""

#döngüyü kontrol eden değişkenşn artırılma ya da azaltılma işlemi
#şart

```
i = 1
while i<5:
    print(i)
    i = i+1
```

```
i = 4
while i>0:
    print(i)
    i -= 1
```

```
i=0
while i<4:
    print("Ezber yok!", end=" ")
    j = 0
    while j<4:
        print("Mantık Var!", end=" ")
        j += 1
    i += 1
```

```
sayılar = [10, 20, 30, 40, 50]
index = 0
while index < len(sayılar):
    print(index, sayılar[index])
    index += 1
```

```
toplam = 0
i = 1
while i<= 10:
    toplam += i
    i += 1
print("toplam", toplam)
```

```
toplam = 0
i = 40
while i<= 50:
    toplam += i
    i += 1
print("toplam", toplam)
```

```
# 1-200 arasındaki sayılardan 5 ve 7 ye bölünebilenler
```

```
i = 1
while i<=200:
    if i % 5 == 0 and i % 7 == 0:
        print(i)
    i += 1
```

```
# 1 den n e kadar olan sayıların toplamını bulan program
```

```
n = int(input("1 den kaç e kadar olan sayıların toplamını istiyorsunuz"))
i = 1
toplam = 0
while i<=n :
    toplam = toplam + i
    i += 1
print("toplam", toplam)
```

```
# Bir önceki örneğin farklı bir çözümü
```

```
toplam = 0
while n>=0:
    toplam += n
    n -= 1
print("toplam", toplam)
```

```
# 5 ve 7 ye bölünemeyen sayılar
```

```
while i <= 200:
    if i % 5 != 0 and i % 7 != 0:
        print(i)
    i += 1
```

```
i = 0
while i< 6:
    print("&", end=" ")
    j = 0
    while j < 5:
        print("#", end=" ")
        j = j+1
    print()
    i = i+1
```

```
i = 0
while i < 6:
    j = 0
    while j< 5:
        print("&", end=" ")
        j += 1
    print()
    i += 1
```

```
günler = ["pazartesi" , "salı", "çarşamba", "perşembe", "cuma", "cumartesi", "pazar"]
```

```
i = 0
while i < 7:
    print(günler[i], end=" ")
    j = i+1
    while j<=31:
        print(j, end=" ")
        j += 7
    print()
    i += 1
```



```

# Kullanıcının girdiği sayının sayı değerlerinin toplamını fulan fonksiyon
"""sayı= int(input("lütfen bir sayı giriniz"))

sayıdeğerleri = []

while sayı>0:
    sayıdeğeri = sayı % 10
    sayı // 10
    sayıdeğerleri.append(sayıdeğeri)
print(sayıdeğerleri)

"""

# Faktoriyelini öğrenmek istediğiniz sayıyı giriniz
try:
    n = int(input("faktoriyelini bulmak istediğiniz sayıyı giriniz"))
    i = 1
    sonuc = 1
    if n>=0:
        while i < n:
            sonuc = sonuc*i
            i += 1

        print("{}!=".format(n),sonuc)
    else:
        print("adamı hasta etme")
except ValueError:
    print("lütfen bir tam sayı giriniz")

# Kullanıcının yemek seçimleri doğrultusunda toplam tutarı kullanıcıya söyleyen
programı yazınız.
print( """1. Adana/Urfa 20tl
        2. Beyti 25tl
        3. Tantuni 30tl
        4. Çorba 10tl
        5. Tatlı 15tl
        6. İçecek 5 tl
        7. Çıkış""")
toplamTutar = 0
while True:
    secim= int(input("lütfen seçiminizi tuşlayınız"))
    if secim > 0 and secim < 7:
        adet = int(input("kaç adet alırsınız"))
        if secim==1:
            toplamTutar+=20*adet
        elif secim==2:
            toplamTutar+=25*adet
        elif secim==3:
            toplamTutar+=30*adet
        elif secim==4:
            toplamTutar+=10*adet
        elif secim==5:
            toplamTutar+=15*adet
        elif secim==6:
            toplamTutar+=5*adet
        elif secim==7:
            print("siparişiniz için teşekkür ederiz")
            print("Toplamda ödenecek tutar:{}".format(toplamTutar))
            break

```

```
# While den farkı squence(dizi) ile çalışır
```

```
for i in range(4):
```

```
    print(i)
```

```
isim = "Berk Batuhan"
```

```
for i in isim:
```

```
    print(i)
```

```
for i in "Batuhan":
```

```
    print(i)
```

```
for harf in "Batuhan":
```

```
    print(harf, end=" ")
```

```
sayılar=[2,3,4,5]
```

```
for sayı in sayılar:
```

```
    print(sayı)
```

```
karışık=["a", 24.5, "batuhan"]
```

```
for ifade in karışık:
```

```
    print(ifade)
```

```
bilgiler = {'Batuhan':"Elektrik Mühendis", 'Berk':"Matematik Mühendisi"}
```

```
print(bilgiler.items())
```

```
for anahtar, değer in bilgiler.items():
```

```
    print(anahtar, değer)
```

```
benzersizsayılar= {-1, 3, 5, 7}
```

```
for sayı in benzersizsayılar:
```

```
    print(sayı, end=" ")
```

```
değişmeyensayılar=(2, 3, 4, 5, 6)
```

```
for sayı in değişmeyensayılar:
```

```
    print(sayı)
```

```
tuplelist=[(21,32), (41,53), (0,-11)]
```

```
for tup in tuplelist:
```

```
    print(tup)
```

```
for tupsayı1, tupsayı2 in tuplelist:
```

```
    print(tupsayı1, tupsayı2)
```

```
tuplelist=[(21,32,17), (41,53,17), (0,-11,17), (10,19,17), (12,90,17), (11,13,17)]
```

```
for tupe1, tupe2, tupe3 in tuplelist:
```

```
    print(tupe1,tupe2,tupe3)
```

```
sayılar = list(range(1,51))
```

```
print(sayılar)
```

```
print(sum(sayılar))
```

```
toplam = 0
```

```
for sayı in sayılar:
```

```
    toplam += sayı
```

```
print(toplam)
```

```
sayı1 = int(input("lütfen ilk sayıyı giriniz"))
```

```
sayı2 = int(input("lütfen ikinci sayıyı giriniz"))
```

```
sayılar = list(range(sayı1, sayı2+1))
```

```
toplam = 0
```

```
for sayı in sayılar:
```

```
    toplam += sayı
```

```
print(toplam, sum(sayılar))
```

```
# bir liste içerisindeki tek ve çift sayıları tutan ve bu elemanları liste içerisinde  
ayrı ayrı tutan programı yazınız.
```

```
sayılar = [2, 3, 4, 5, 65, 67, 78, 90, 12, 2, 6, 78, 65, 45, 43, 64, 87, 98]
```

```
ciftliste = []
```

```
tekliste = list()
```

```
for sayı in sayılar:
```

```
    if sayı %2 == 0:
```

```
        ciftliste.append(sayı)
```

```

else:
    tekliste.append(sayı)
print("Tek sayılar:{} ve {} tek sayı bulunmaktatadır.".format(tekliste, len(tekliste)))
print("Çift sayılar:{} ve {} çift sayı bulunmaktadır.".format(ciftliste,
len(ciftliste)))

# Şekildeki çıktıyı veren programı yazınız.
"""
*****
*****
*****
*****
*****"""
#1.yol
for i in range(5):
    for j in range(5):
        print("*", end="")
    print()
#2. yol
for i in range(5):
    print(5*"*", end="")
    print()

# Çarpım tablosunu veren programı yazınız.

for i in range(1,11):
    for j in range(1,11):
        print("{}*{}={}".format(i,j,j*i), end="          ")
    print()

# 1 ile 1001 arasında karekökü tam sayı olan sayılar hangileridir.
from math import sqrt
for i in range(1,1001):
    if int(sqrt(i))**2==i :
        print("karekök", i, sqrt(i), "i karekökü tam sayı olan bir sayıdır")

# Öğrenci bilgisi içeren model oluşturacağız. 5tane öğrencinin ayrı ayrı isim, numara,
2. vize notu ve 1 final notunu
# tutan bir dizi oluşturun.Daha sonra her öğrenci için dönem sonu notunu hesaplayın.(1.
vize %25, 2.vize %35 final%40)
# Derste çan sistemi uygulanmaktadır. 5öğrencinin dönem sonu ortalamasının üzerinde
kalan öğrenciler dersi geçmiş
# diğer öğreniler kalmı sayılacaktır

öğr1 = {"isim":"Batuhan", "vize notları":[75,45], "final notu": 85}
öğr2 = {"isim":"Aliss", "vize notları":[85,53], "final notu": 85}
öğr3 = {"isim":"Melis", "vize notları":[35,45], "final notu": 85}
öğr4 = {"isim":"Emily", "vize notları":[75,35], "final notu": 85}
öğr5 = {"isim":"Berk", "vize notları":[25,85], "final notu": 85}

ogrenciler = [öğr1, öğr2, öğr3, öğr4, öğr5]

ogrencilerindonemsonunottoplamı = 0
for ogrenci in ogrenciler:
    ogrDonemSonuNot = ogrenci["vize notları"][0]*0.25+ogrenci["vize
notları"][1]*0.35+ogrenci["final notu"]*0.40
ogrencilerindonemsonunottoplamı += ogrDonemSonuNot
ogrencidonemsonunotort =ogrencilerindonemsonunottoplamı/ len(ogrenciler)

for ogrenci in ogrenciler:
    ogrDonemSonuNot = ogrenci["vize notları"][0] * 0.25 + ogrenci["vize notları"][1] *
0.35 + ogrenci["final notu"] * 0.40
    if ogrDonemSonuNot>=ogrencidonemsonunotort:
        print("OğrenciAdı:{} ortalaması:{}")

```

```
Geçti".format(ogrenci["isim"],ogrDonemSonuNot))
    else:
        print("OgrenciAdı:{} ortalaması:{} Kaldı".format(ogrenci["isim"],
ogrDonemSonuNot))

bilgiler = [[16,"bursa","ipek"],[38, "kayseri", "mantı"], [41,"kocaeli","pişmaniye"]]
for plakakodu, isim, meshurürünü in bilgiler:
    print(plakakodu, isim, meshurürünü)
```

```
"""Break anahtar kelimesi"""
```

```
sayı = 0
while True:
    print(sayı)
    if sayı == 100:
        break
    sayı += 1

while True:
    n = float(input("lütfen bir sayı giriniz( çıkış değeri 0 dır)."))
    print("{}karesi = {}".format(n,n*n))
    if n ==0:
        print("programı kullandığınız için teşekkür ederiz")
        break
```

Bir stringi belirlenen bir harfe kadar yaz be bu harfin stringin kaçınca indexinde olduğunu bulan program.

```
index = 0
for harf in "Berk Batuhan":
    if harf == "K" or harf == "k":
        print("{} harfi {}. indexte bulunmaktadır".format(harf,index))
        break
    index += 1
"""Enumerate Anahtar kelimesi"""
```

```
enum = enumerate("BerK Batuhan")
print(enum)
print(list(enum))
```

```
for index, harf in enumerate("Berk Batuhan"):
    if harf.lower() == "k":
        print("{} harfi {}. indexte bulunmaktadır".format(harf, index))
        break
```

```
enum = enumerate("berk batuhan", 5)
print(enum)
print(list(enum))
```

Şifreyi doğru girmesi için kullanıcıya 3 hak tanıyan programı yazınız.
Doğru girmediği sürece şifre talep edecek, 3 defa yanlış girerse oturum sonlanacak.

```
secret_pass = "A1"
count_pass = 3
```

```
while True:
    sifre = input("lütfen şifreyi giriniz")
    if sifre==secret_pass:
        print("Sisteme Hoşgeldiniz")
        break
    count_pass -= 1
    if count_pass <= 0:
        print("$iifre girme hakkınız bitti.")
        break
```

Defter satan bir e-ticaret sitesinde defter satın almak isteyen bir müşteriye kaç adet defter almak istediğini soran
ve kullanıcıdan istediği defter sayısını stokla kontrol ederek kullanıcıya tepki veren programı yazınız.
5 defter alırsa 5 defa "1 adet defter sepete eklendi" desin. (stok miktarı başlangıçta 10 olsun)

```
stokMiktarı = 10
istenenDefterSayısı= int(input("lütfen satın almak istediğiniz defter sayısını giriniz"))
```

```
SepeteEklenenDefterSayısı= 0
```

```
for i in range(1,istenenDefterSayısı+1):
    if i>stokMiktarı:
        print("stopta istediğiniz adette defter bulunmamaktadır")
        break
    SepeteEklenenDefterSayısı +=1
    print("1 Adet Defter Sepete Eklenmiştir")
print(SepeteEklenenDefterSayısı,"adet defter sepete eklendi")
```

```
"""Continue anahtar kelimesi"""
```

```
for sayı in range(1,250):
    if sayı%5 ==0:
        print(sayı,end=" ")
print()
```

```
for sayı in range(1,250):
    if sayı % 5 !=0:
        continue
    print(sayı,end=" ")
```

1 ,le 100 arasındaki sadece 35 e bölünebilen sayıları continue anahtar kelimesi yardımıyla yazdınız.

```
for i in range(1,1001):
    if i % 35!=0:
        continue
    print(i,end=" ")
```

Banka Atm örneği

```
"""Kartın bir şifresi vardır
Kartın bakiyesi 500 Tl'dir.
3 defa yanlış şifre girilince kart bloke olacaktır.
Atm' nin işlem menüsünde para çekme, para yatırma, bakiye sorgulama ve kart iade işlemleri yapılmaktadır."""
```

```
sıfre= int(input("lütfen 4 basamaklı şifrenizi giriniz!"))
```

```
_sıfre = 1453
_bakiye = 500
sıfresayac = 3
login = False
```

```
while True:
    if login == False:
        sıfre = int(input("lütfen 4 basamaklı şifrenizi giriniz"))
    if sıfre == _sıfre:
        login = True
        print("""
```

```
1.Para çekme
2.Para Yatırma
3.Bakiye Sorgulama
4. Kart iade işlemleri""")
```

```

# Banka Atm örneği
"""Kartın bir şifresi vardır
Kartın bakiyesi 500 Tl'dir.
3 defa yanlış şifre girilince kart bloke olacaktır.
Atm' nin işlem menüsünde para çekme, para yatırma, bakiye sorgulama ve kart iade
işlemleri yapılmaktadır."""

_sifre = 1453
_bakiye = 500
sifresayac = 3
login = False
while True:
    if login == False:
        sifre = int(input("lütfen 4 basamaklı şifrenizi giriniz"))
    if sifre == _sifre:
        login = True
        print("""
1. Para çekme
2. Para Yatırma
3. Bakiye Sorgulama
4. Kart iade işlemleri""")
        secim = int(input("Hangi işlemi yapmak istiyorsunuz?"))
        if secim == 1:
            miktar = int(input("Kaç tl çekmek istiyorsunuz?"))
            if _bakiye < miktar:
                print("yeterli bakiyeniz bulunmamaktadır.")
                continue
            _bakiye -= miktar
        elif secim == 2:
            miktar = int(input("Kaç tl yatırmak istiyorsunuz?"))
            _bakiye += miktar
        elif secim == 3:
            print("bakiyeniz {}tl".format(_bakiye))
        elif secim == 4:
            print("yine bekleriz")
            break
        else:
            print("lütfen 1 ile 4 arasında bir rakam giriniz.")
    else:
        sifresayac -= 1
        if sifresayac < 1:
            print("kartınız bloke olmuştur")
            break

"""PASS anahtar kelimesi"""
# Normalde kod satırı içermesi gereken bir bloğun içerisini boş bıraktığımızda
programın çalışmasına
# engel olmaması için blok içerisinde pass anahtar kelimesini kullanırsınız.
# Kısaca daha sonradan çalışmaya devam etmek için o anlık pas geçmek gerekirse pass
anahtar kelimesi kullanılır.

for i in range(0,100):
    pass #sonrasında yazılanlar program tarafından okunur break gibi olmaz.

sayi = 5
if sayi == 5 :
    print(sayi)
else:
    pass

"""For else yapısı"""

sayilar=[1,3,9,5]

```

```

for sayi in sayilar:
    if sayi%2==0:
        print("{} çift sayıdır.".format(sayi))
        break
else:
    print("çift sayı bulunmamaktadır.")

"""ÖZEL ÖRNEK1"""
# Bir sayının asal sayı olup olmadığını kontrol eden programı for else yapısı ile yazınız.
sayi=7
from math import sqrt
for i in range(2, int(sqrt(sayi)+1)):
    if sayi %2 ==0:
        print("{} asal sayı değildir".format(sayi))
        break
else:
    print("{} asal sayıdır.".format(sayi))

"""ÖZEL ÖRNEK2"""
# 2 ile 1000 arasındaki sayıların asal olup olmadığını kontrol eden programı for else kullanarak yazınız.
for i in range(2,1001):
    for j in range(2, int(sqrt(i)+1)):
        if i%j==0:
            break
    else:
        print(i,end=" ")

"""EXTRA ÖRNEK"""
try:
    n = int(input("Lütfen faktoriyelini almak istediğiniz sayıyı giriniz.))
    if n<0:
        print("Lütfen bir doğal sayı giriniz.")
    else:
        sonuc = 1
        for i in range(1,n+1):
            sonuc *= i
        print("{}! = {}".format(n, sonuc))
except ValueError:
    print("lütfen bir tam sayı giriniz!")

"""EXTRA ÖRNEK Fibonacci"""
n1, n2 = 0, 1
count = 0
terimsayısı = int(input("lütfen kaç terim görüntülemek istediğinizi yazınız.))
if terimsayısı <= 0:
    print("lütfen pozitif bir terim sayısı giriniz")
elif terimsayısı == 1:
    print(n1)
else:
    while count < terimsayısı:
        print(n1, end=" ")
        n3 = n1 + n2
        n1=n2
        n2=n3
        count += 1          # CTRL+TAB yapınca python dosyaları arasında geçiş yapar

"""EXTRA ÖRNEK"""
# Kullanıcının girdiği iki tam sayının ebobu nu bulan programı yazınız.
x = int(input("lütfen ilk sayıyı giriniz"))
y = int(input("lütfen ikinci sayıyı giriniz"))

```



```
sayi1 = x
sayi2 = y
```

```
while y:
    x, y = y, x*y
print("EBOB({}, {}) = {}".format(sayi1, sayi2, x))
```

```
# For döngüsünün etkili kullanıldığı yerler.
```

```
sayılar1 = [3,5, 9, 11, 6, 1, 13, 15]
sayılar2 = sayılar1
sayılar3 = [sayı for sayı in sayılar1]
print(sayılar2)
print(sayılar3)
```

```
sayılar4 = [sayı ** 2 for sayı in sayılar1]
print(sayılar4)
```

```
isimler = ["fatih", "berk", "batuhan"], ["ayça", "sinem", "Handan"]
tekilisimler = list()
for cinsiyetegoreliste in isimler:
    for isim in cinsiyetegoreliste:
        tekilisimler.append(isim)
print(isimler)
```

```
tekilisimler = []
for i in range(0,len(isimler)):
    tekilisimler += [isim for isim in isimler[i]]
print(tekilisimler)
```

```
tekilisimler = []
for cinsiyetegoreisimler in isimler:
    tekilisimler += ["x"+ isim + "x" for isim in cinsiyetegoreisimler]
print(tekilisimler)
```

```
tekilisimler = [ isim for cinsiyetegoreisimler in isimler for isim in
cinsiyetegoreisimler]
```

```

#1
for i in range(5):
    for i in range(5):
        print("*", end="")
    print()#satırı atla demek

*****
*****
*****
*****
*****
*****
print("-----")
#2
k=1
for i in range(5):
    print("*"*k, end="")
    k += 1
    print()

for i in range(1,6):
    for j in range(i):
        print("*", end="")
    print()
****
*
**
***
****
*****
*****
print("-----")
#3
k=5
for i in range(5):
    print("*"*k, end=" ")
    k -= 1
    print()

for i in range(1,5):
    for j in range(5-i):
        print("*", end="")
    print()
****
*****
****
***
**
*

```

```

"""
print("-----")
#4

k=0
p=5
for i in range(0,6):
    print(" "*p, end="")
    p -= 1
    print(""*k)
    k += 1

n = 7
for i in range(1, n+1):
    for j in range(2*n - 2*i):
        print(end=" ")
    for j in range(0,i):
        print(end="* ")
    print()

n = 5
for i in range(i, n+1):
    print((2*n - 2*i)* " " + i*"* ")
"""
*
**
***
****
*****

      *
     * *
    * * *
   * * * *
  * * * * *
 * * * * *
* * * * *
"""

```

#5

```

i = 0
for i in range(1,6):
    print(" "*(2*i-2), end="")
    print("* "*(6-i))
    i += 1

n=5
for i in range(1, n+1):
    for j in range((2*i)-2):
        print(end=" ")

```

```

        for j in range(0,n+1-i):
            print(end="* ")
        print()

"""
* * * * *
* * * *
* * *
* *
*
* """

#6
n = 5
for i in range(0,n):
    for j in range(0,i+1):
        print("*", end=" ")
    print()
for i in range(n-1,0,-1):
    for j in range(i):
        print(end="* ")
    print()

n = 5
for i in range(1, 2*n):
    if i <= n:
        print(i*"* ")
    else:
        print((2*n-i)*"* ")

"""
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
* """

#7
n = int(input("lütfen bir sayı giriniz"))
for i in range(1,n):
    print(" "*n, end=" ")
    n -= 1
    print("* "*i)
n = 5
k = n+2
for i in range(n):
    for j in range(k):
        print(end=" ")

```

```

    k -= 1
    for j in range(i+1):
        print(end="* ")
    print()

"""
    *
  * *
 * * *
* * * *
* * * * *
* * * * * *
  *
  * *
 * * *
* * * *
* * * * *
"""

```

Process finished with exit code 0

#8

```

n = 6
k = n+2

for i in range(1,n+1):
    for j in range(k):
        print(end=" ")
    k -= 1
    for j in range(i):
        print(end="* ")
    print()
for i in range(n-1,0,-1):
    for j in range(k+2):
        print(end=" ")
    k += 1
    for j in range(i):
        print(end="* ")
    print()
"""
    *
  * *
 * * *
* * * *
* * * * *
* * * * * *
  * * * * *
   * * * *
    * * *
     * *
      *
"""

```

```

#9
n = 5
char = 65
for i in range(n):
    ch = chr(char)
    char += 1
    for j in range(0,i+1):
        print(ch, end=" ")
    print()

```

```

"""
A
B B
C C C
D D D D
E E E E E
"""

```

```

#10
n = 6
k = n + 2
for i in range(n):
    print(end=" "*k)
    k -= 1
    x = 65
    for j in range(0,i+1):
        ch = chr(x)
        print(ch, end=" ")
        x += 1
    print()
for i in range(n, 1,-1):
    print(end=" "*(k+2))
    k += 1
    x = 65
    for j in range(0,i-1):
        ch = chr(x)
        print(ch, end=" ")
        x += 1
    print()
"""

```

```

    A
  A B
A B C
A B C D
A B C D E
A B C D E F
  A B C D E
    A B C D
      A B C

```

A B

A

""

```

import array
sayılar = array.array("i", [-2, 2, 4, -7])
print(sayılar)
import array as arr
sayılar = arr.array("i", [-2, 2, 4, -7])
print(sayılar)

from array import *
sayılar = array("i", [-2, 2, 4, -7])
print(sayılar)
from array import *
sayılar = array("I", [9, 2, 4, 7]) # "I" ==> kesirli sayı ya da negatif
sayı eklenmez
print(sayılar)

print(sayılar[0])
print(sayılar.buffer_info()) #arrayın nerede tutulduğunu ve içerisinde kaç
tane eleman bulunduğunu verir.
print(sayılar.typecode) # generator tipini sorgular

for index in range(len(sayılar)):
    print(sayılar[index], end=" ")
harfler = array("u", ["f", "b", "s"]) # 9 2 4 7 array('u', 'fbs')
print(harfler)
print(harfler[0])

""" Generators """
from array import *
sayılar = array("i", [-2, 0, 2, 3, 4, 10]) #Generatörler sayesinde ram de
yer kaplamadan iş yapılabilir.
print([sayı for sayı in sayılar])
print((sayı for sayı in sayılar)) # generator obje oluşturuldu ramde yer
kaplamadı
print(list(sayı for sayı in sayılar))
print(set(sayı for sayı in sayılar))
print(tuple(sayı for sayı in sayılar))
print(sum(sayı for sayı in sayılar))
print(max(sayı for sayı in sayılar))

for sayı in (sayı*sayı for sayı in sayılar):
    print(sayı, end=" ") # Generatorler de bu şekilde kullanılabilir
print()
i = 0
while i < len(list(sayı*sayı for sayı in sayılar)):
    print(list(sayı*sayı for sayı in sayılar)[i], end=" ")
    i += 1
print()

sayılar3 = array(sayılar.typecode, (sayı for sayı in sayılar))
print(sayılar3)

```



```

for sayı in sayılar3:
    print(sayı, end=" ")
print()

i = 0
while i < len(sayılar3):
    print(sayılar[i], end=" ")
    i += 1
print()

generator = (sayı*sayı for sayı in sayılar)
i= 0
while i < len(sayılar3):
    print(sayılar[i], end=" ")
    i += 1
print()

""" Kullanıcıdan Array Elemanlarını Alma""" #arr.index()
from array import*
arr = array("i", [])
# printer(arr)
n = int(input("Diziye kaç tane eleman eklemek istiyorsunuz?"))
for i in range(n):
    sayı = int(input("lütfen eklemek istediğiniz sayıyı giriniz"))
    arr.append(sayı)
print(arr)

"""Array içerisindeki elemanı bulma"""
from array import array
arr = array("i", [2, 5, 8, -1, 0, 8, 76])
aranacakSayı = int(input("lütfen aramak istediğiniz sayıyı giriniz."))

i = 0
for sayı in arr:
    if sayı == akanacakSayı:
        print("{} değeri {}. indekste bulunmaktadır.".format(aranacakSayı,
i))
    else:
        print("aradığınız sayı bu array içerisinde bulunmamaktadır.")
        break # ilk bulunan indexte durmasını sağlar
    i += 1

aranacakSayı = int(input("lütfen aramak istediğiniz sayıyı giriniz."))
print("{} değeri {}. indekste bulunmaktadır.".format(aranacakSayı,
arr.index(aranacakSayı)))

""" Array İçerisindeki elemanı silme """
arr = array("i", [2, 5, 8, -1, 0, 75, 8, 75])
print(arr)

```

```

#Karşılaşılan ilk elemanı silme
arr.remove(8)
print(arr)
#Bütün değerleri silme
for i in arr:
    if i == 75:
        arr.remove(75)
print(arr)
arr = array("i", [2, 3, 4, 5, 6, 7, 2, 3, 4, 5, 2])
arr[5] = 100
print(arr)
for index, sayı in enumerate(arr):
    if sayı == 7:
        arr[index] = 100
print(arr)

```

""" Girilen değerin dizi içerisinde kaç defa geçtiğini bulmak """

```

from array import array
arr= array("i", [2, 3, 4, 5, 6, 7, 2, 3, 4, 5, 2, 7])
aranacakSayı = int(input("lütfen kaç tane geçtiğini bulmak istediğiniz sayıyı yazınız"))

```

```

#1.yol
count = 0

```

```

for i in arr:
    if i == akanacakSayı:
        count += 1
print("{} sayısı {} defa geçmektedir.".format(aranacakSayı, count))

```

```

#2.yol
aranacakSayıIndex = array("i", [])
for sayı in enumerate(arr):
    if sayı == akanacakSayı:
        akanacakSayıIndex.append(index)
print(aranacakSayıIndex)

```

```

#3.yol
print("count metodu: {} sayısı {} defa geçmektedir.".format(aranacakSayı, arr.count(aranacakSayı)))

```

ÖRNEK Ekleme, silme ve tersten sıralama işlemlerini metod kullanmadan uygulama.

1-) Boş bir dizi oluşturun. Bu diziye 7 tane eleman kullanıcı tarafından eklenecektir.

Ekleme işlemini append() metodu kullanmadan gerçekleştirin

2-) Bu dizide kullanıcının silmesini istediği değerlerin ailinmesini

sağlayın.

```
# Silme işlemini remove() metodunu kullanmadan gerçekleştirin.  
# 3-) Bu dizinin elemanlarının tersine çevrilmesini sağlayın.  
# Tersine çevirme işlemini reverse() metodunu kullanmadan  
gerçekleştirin.
```

```
# 1
```

```
from array import array  
dizi = array("i", [])  
for i in range(7):  
    sayı = int(input("lütfen eklemek istediğiniz sayıyı yazınız!"))  
    dizi += array("i", [sayı]) # iki array birbiriyle toplandı  
print(dizi)
```

```
delVal = int(input("lütfen silmek istediğiniz sayıyı giriniz"))  
degerSilinmişDizi = array("i", [])  
for i in dizi:  
    if i != delVal:  
        degerSilinmişDizi += array("i", [i])  
dizi = degerSilinmişDizi  
print(dizi)  
tersDizi = array(len(dizi))  
for i in range(len(dizi)-1, -1, -1):  
    tersDizi += array("i", [dizi[i]])  
dizi = tersDizi  
print(dizi)
```

```
""" Foksiyonların en önemli amacı aynı işlemi tekrar tekrar kodlamayı önlemektir"""
```

```
# fonksiyonlar tanımlanırken def anahtar kelimesi kullanılır.
```

```
def seslen():  
    print("sesleniyorum")  
    print("naber")  
seslen()
```

```
# fonksiyon çağırırken call anahtar kelimesi kullanılır.
```

```
""" Dönüş Değeri Olmayan Parametresiz Fonksiyonlar"""
```

```
def selamVer():  
    sonuc = 5 + 6  
    return sonuc # return(dönüş değeri) ortaya bir ürün çıkacak demektir.  
Ortaya sonuc ürünü çıkacak, return kullanılan fonksiyonlar tek başına kullanılarak yazdırılamaz.  
print(selamVer())
```

```
def selamVer():  
    sonuc = 5 + 6  
    print(sonuc) # dönüş değeri olmayan fonksiyonlar print içerisinde kullanılamaz.
```

```
selamVer()
```

```
def isimSoyle():  
    print("Mustafa Kemal ATATÜRK")
```

```
isimSoyle()
```

```
def isimSoyle2():  
    print("Berk Batuhan Devran")
```

```
def bilgiVer():  
    isimSoyle2() # fonksiyon içinde fonksiyon kullanılabilir  
    print("Yıldız Teknik Üniversitesi")  
    print("Matematik Mühendisi")  
    print("ANAMUR")
```

```
bilgiVer()
```

```
def topla(sayı1, sayı2): # parametrelili fonksiyon  
    print(sayı1 + sayı2)
```

```
topla(65,87)
```

```
def isimSoyle3(isim):
```

```

    print(isim)
isimSoyle3("berk")

from math import pi
print(pi)
def piSayısı():
    return pi
print(piSayısı())

def veriTabanınaBaglan():
    print("veri tabanına bağlanıyor")
    return "veri tabanına bağlandı"
print(veriTabanınaBaglan())

bilgi = veriTabanınaBaglan()
print(bilgi)

"""Dönüş Değeri Olan parametrelili(argümanlı) fonksiyonlar"""
print(abs(-9))
sayı = abs(-15)
print(sayı)

def carp(sayı1, sayı2):
    return sayı1 * sayı2
print(carp(4,6))
sonuc = carp(8, 9)
print(sonuc)

def mutlakDegerliCarpım(sayı1, sayı2):
    return abs(sayı1 * sayı2)
print(mutlakDegerliCarpım(-7, 9))

"""Birden Fazla Dönüş Değeri Olan Fonkşşyonlar"""
def ucgenKenarlarınıSoyle():
    return 3,4,5 # defaultta () isersen [] koyabilirsin...
birinciKenar = ucgenKenarlarınıSoyle()[0]
ikinciKenar = ucgenKenarlarınıSoyle()[1]
üçüncüKenar = ucgenKenarlarınıSoyle()[2]
print(ucgenKenarlarınıSoyle())
print(birinciKenar)
print(ikinciKenar)
print(üçüncüKenar)
from math import pi
def daireAlanVeCevreHesapla(r):
    return pi * (r**2), 2*pi*r
print(daireAlanVeCevreHesapla(100))
daireAlan = daireAlanVeCevreHesapla(100)[0]
daireCevre = daireAlanVeCevreHesapla(100)[1]

```

```
print(daireAlan)
print(daireCevre)
```

```
"""FONKSİYON ARGÜMANLARI"""
```

```
# Pass by value (Değer Yollama)
# Pass by reference (Adresi yollama)
```

```
def guncelle(sayi):
    print("Sayı id: ", id(sayi))
    sayi = 9
    print("Sayı id: ", id(sayi))
```

```
sayimiz = 10
guncelle(sayimiz)
print(sayimiz)
```

```
def listeGuncelle(liste):
    print("liste id:", id(liste))
    liste[0] = 100
    print("liste id:", id(liste))
```

```
listemiz = [0, 45, 75, 86, 98]
listeGuncelle(listemiz)
print(listemiz)
```

```
"""Types of Arguments (Argüman tipleri)"""
```

```
#Positional, keyword, default, variable, length
```

```
def cikar(a, b):
    sonuc = a - b
    print(sonuc)
    return sonuc
print(cikar(86, 54))
cikar(86, 54)
```

```
def ogrenciBilgileriSoyle(isim, numara, adres, sube=""):
    print("Öğrencinin ismi:", isim)
    print("Öğrencinin numarası:", numara)
    print("Öğrencinin şubesi:", adres)
    print("Öğrencinin adresi:", sube)
ogrenciBilgileriSoyle("Berk Batuhan Devran", "595", "Mersin/Anamur", "A")
# Eğer bir bilgi daha sonra girilecekse o bilgiyi default da boş olarak tanımlamalıyız
ogrenciBilgileriSoyle("Berk Batuhan Devran", "595", "Mersin/Anamur")
# ogrenciBilgileriSoyle("Berk Batuhan Devran", "Mersin/Anamur") # default da boş olmayan değer
```

için atama yapılmadan fonksiyon çalıştırıldığında program hata vriyor

```
def ogrenciBilgileriSoyle(isim, numara, sube, *adres):
    print("Öğrencinin ismi:", isim)
    print("Öğrencinin numarası:", numara)
    print("Öğrencinin şubesi:", sube)
    print("Öğrencinin adresi:", adres)
ogrenciBilgileriSoyle("Berk", "595", "A", "Mersin/ANAMUR",
"Mersin/TARSUS", "Mersin/AKDENİZ")
# yaptığımız *... değişikliği sonucu adres değeri sonsuz değer alabilir
```

```
def ogrenciBilgileriSoyle(isim, numara, sube, *adres):
    print("Öğrencinin ismi:", isim)
    print("Öğrencinin numarası:", numara)
    print("Öğrencinin şubesi:", sube)
    print("Öğrencinin adresi:", adres)
ogrenciBilgileriSoyle("Berk", "595", "A")
# *adres e değer yazmadığımız zaman program boş bir tuple gönderiyor.
```

```
def ogrenciBilgileriSoyle(isim, numara, sube, *adres):
    print("Öğrencinin ismi:", isim)
    print("Öğrencinin numarası:", numara)
    print("Öğrencinin şubesi:", sube)
    if len(adres) != 0:
        print("Öğrencinin adresi:", adres)
    else:
        print("adres girilmemiştir")
```

```
def ogrenciBilgileriSoyle(isim, numara, sube, adres):
    print("Öğrencinin ismi:", isim)
    print("Öğrencinin numarası:", numara)
    print("Öğrencinin şubesi:", sube)
    if len(adres) != 0:
        print("Öğrencinin adresi:", adres)
    else:
        print("adres girilmemiştir")
ogrenciBilgileriSoyle("Berk", "595", "A", "anamur")
ogrenciBilgileriSoyle(sube="b", isim="batuhan", adres="anamur",
numara="45")
```

```
def ogrenciBilgileriSoyle(isim, numara, adres, sube=""):
    print("Öğrencinin ismi:", isim)
    print("Öğrencinin numarası:", numara)
    print("Öğrencinin şubesi:", adres)
    print("Öğrencinin adresi:", sube)

ogrenciBilgileriSoyle(sube="b", isim="batuhan", adres="anamur",
numara="45")
#Bu kodun doğru çalışabilmesi için default değeri değiştirilen değerler
sube gibi sona alınmalıdır.
```

```
"""variable length"""
def mySum(*sayılar):
    sonuc = 0
    for sayi in sayılar:
        sonuc += sayi
    print(sonuc)
```

```
mySum(4, 5, 6, 7)
```

```
def ogretmenBilgileriSoyle(tcKimlikNo, **data):
    # isim
    # maaş
    # brans
    print(tcKimlikNo, data)

ogretmenBilgileriSoyle(11111111, isim = "Berk Batuhan", soyad = "DEVİRAN",
maas = 4354, brans = "Matematik")
# çıktı:
# 11111111 {'isim': 'Berk Batuhan', 'soyad': 'DEVİRAN', 'maas': 4354,
'brans': 'Matematik'}
```

```
def ogretmenBilgileriSoyle(tcKimlikNo, **data):
    # isim
    # maaş
    # brans
    print(data["isim"])
    print(data["soyad"])
    print(data["maas"])
    print(data["brans"])
    print(data.keys())
    print(data.values())
    print(data.items())
ogretmenBilgileriSoyle(11111111, isim = "Berk Batuhan", soyad = "DEVİRAN",
maas = 4354, brans = "Matematik")

#çıktı:
#Berk Batuhan
```



```

#DEVİRAN
#4354
#Matematik
#dict_keys(['isim', 'soyad', 'maas', 'brans'])
#dict_values(['Berk Batuhan', 'DEVİRAN', 4354, 'Matematik'])
#dict_items([('isim', 'Berk Batuhan'), ('soyad', 'DEVİRAN'), ('maas',
4354), ('brans', 'Matematik')])

""" değer önüne tek yıldız koyduğün zaman keywordsüz sonsuz eleman
alabilirsin
iki yıldız koyarsan keywordlü sonsuz eleman alabilirsin """

"""def toplamaYadaÇarpmaİşlemi(**sayılarVeİşlem):
    print(sayılarVeİşlem)
    if sayılarVeİşlem['isim'] == "toplama":
        sonuc = 0
        for sayı in sayılarVeİşlem['sayılar']:
            sonuc += sayı
            print(sonuc)
    elif sayılarVeİşlem["isim"] == "çarpma":
        sonuc = 1
        for sayı in sayılarVeİşlem['sayılar']:
            sonuc *= sayı
            print(sonuc)
    else:
        print("lütfeñ toplama veya çarpma işlem bilgileri giriniz ")
toplamaYadaÇarpmaİşlemi(sayılar = [4, 5, 5, 6, 7, 8], isim = "toplama")
toplamaYadaÇarpmaİşlemi(sayılar = [4, 5, 5, 6, 7, 8], isim = "çarpma")
"""

"""Varianle length arguments"""

def bilgileriGoster(*isimler, **yaslar):
    for isim in isimler:
        print(isim)
    for yas in yaslar:
        print(yas)
    for key in yaslar.keys():
        print(key)
    for value in yaslar.values():
        print(value)
    for item in yaslar.items():
        print(item)
bilgileriGoster("batuhan", "berk", "DEVİRAN", fatih = 28, kaan = 35, aysel
= 67)
# positional argümanlar keyword argümanlardan daha sonra yazılamaz.

def deneme(*args, isim):#positional argumanlar sonsuz değer alabilen
ifadelerden sonra gelirse mutlaka key kullanılarak değer atanmalı.

```

```

    print(args)
    print(isim)
deneme("fat", "sadsa", "sfds", isim="dasdass")
#Örnek1 Bir kullanıcının ismini, adını, yaşını ve cinsiyetini(opsiyonel)
olarak gösteren kodu yazınız.
#Kullanıcının cinsiyet bilgisi olmadığında konsolda cinsiyete dair
herhangi bir bilgilendirme olmamalıdır.
def showUserInfo(name, username, age, gender=None):
    if gender!=None:
        print(name, username, age, gender)#ctrl+d ile çoğalt alt+shift ve
yön tuşları ile aşağı taşı
    else:
        print(name, username, age)
showUserInfo("Berk", "Batuhan33", 26, "Heriff")
showUserInfo("batuhan", "berk3", 22)

```

```

def showUserInfo(name, username, age, gender=None):
    if gender!=None:
        print(name, username, age, gender)#ctrl+d ile çoğalt alt+shift ve
yön tuşları ile aşağı taşı
    else:
        print(name, username, age)
showUserInfo("Berk", "Batuhan33", 26, "Heriff")
showUserInfo("batuhan", "berk3", 22)

```

```

def showUserInfo(name, username, age, gender=None):
    if gender!=None:
        print(name, username, age, gender)#ctrl+d ile çoğalt alt+shift ve
yön tuşları ile aşağı taşı
    else:
        print(name, username, age)
showUserInfo("Berk", "Batuhan33", 26, "Heriff")
showUserInfo("batuhan", "berk3", 22, gender="herif")

```

#Örnek2 Sınırsız sayıda sayınının karekökünü hesaplayan fonksiyon.

```

from math import sqrt
def calcSqrt(*args):
    Liste = []
    for i in args:
        Liste.append(round(sqrt(i),4))

```

```

print(calcSqrt(4, 6, 67, 89))

```

Örnek3 Bir websitenin veri tabanında kullanıcılar kayıtlıdır. Ve bu kullanıcıların id değerleri vardır.
Yazılımcımız bu id ler arasında en küçük id yi 1 yapmak istemektedir. Ve 1 yapılan en küçük id nin azaldığı değer kadar
diğer id leri azaltmak istemektedir.
Id lerin azaltılmış haline ulaştıktan sonra kullanıcıların her birini

isim ve is değeri ile birlikte konsolda görmek istemektedir.
İlgili fonksiyonu yazınız.
Not:Fonksiyonda parametre olarak sadece keyword arguments alınacaktır.
Output: [('Berk', 1), ('Batuhan', 2)...]

```
def reduceIDandShowINFO(**kwargs):
    minVal = min(kwargs['ids'])
    reducedIDs = []
    for id in kwargs['ids']:
        reducedIDs.append(id - minVal + 1)
    infos = []
    for i in range(len(reducedIDs)):
        infos.append((kwargs['names'][i], reducedIDs[i]))
    return infos

print(reduceIDandShowINFO(ids=[100, 200, 300], names=["berk", "batuhan",
"devran"]))
```

#Örnek4

#Bir inşaat firmasında bir projenin başlangıç ve çalışan maliyetleri vardır. Bunun yanı sıra bilinmeyen maliyetler ve blinen extra # maliyetler de firma tarafından tutulmaktadır.
#Firmanın başlangıç, çalışan, bilinmeyen ve bilinen extra maliyetlerinin toplamını hesaplayan fonksiyonu yazınız.

```
def totalCost(initCost, employeCost, *unknownCosts, **extraKnownCosts):
    totalCostValue = initCost + employeCost
    for unknownCost in unknownCosts:
        totalCostValue += unknownCost
    for extraKnownCost in extraKnownCosts.values():
        totalCostValue += extraKnownCost
    print(totalCostValue)
```

```
totalCost(5, 10, 15, 45, 55, walls=100, doors=99)
```

"""LOCAL VARIABLE vs GLOBAL VARIABLE"""

```
a = 10 #global variable
```

```
def fonk1():
    a = 15 #local variable
    print(a)
fonk1()
def fonk2():
    global a
    a = 20
    print("fonk2 a değeri:", a, "adres:",id(a))
print("global a:", a, "Adres:", id(a))
```

```

fonk2()
print("global a:", a, "Adres:", id(a))

print("-----")
if True:
    y = 10
print(y)
while True:
    z = 20
    break

print(z)

for i in range(1):
    t = 30
print(t)
print("-----")
print(globals())#####

x = 100
print(x, id(x))
def fonk3():
    global x
    y = x
    y = 500
globals()['x']
fonk3()
print(x, id(x))
def fonk3():
    y = globals()['x']
    y = 500
fonk3()
print(x, id(x))

""" FONKSİYONLARDA VE DÖNGÜLERDE SONSUZLUK """
"""
import sys
sys.setrecursionlimit(200)# recursion özyineleme(kendi kendini çağırma)
demektir
print(sys.getrecursionlimit())

while True:
    print("while döngüsü")
def sonsuz():
    print("sonsuz gidiyoruz")
    sonsuz()
sonsuz()"""

```

#Örnek- Bir listeni in içindeki tek ve çift sayıların kaç tane olduğunu

hesaplayan fonksiyonu yazınız.

```
def countOddAndEvenNumber(l):
    countEven = 0
    countOdd = 0
    for i in l:
        if i % 2 == 0:
            countEven += 1
            continue
        countOdd += 1
    return countEven, countOdd
oddEvenList = [2, 4, 6, 78, 978, 56, 4674, 74, -66]
oddEvenCount = countOddAndEvenNumber(oddEvenList)
print(oddEvenCount)
print("Count of odd number in list:", oddEvenCount[0], "Count of even
number in list", oddEvenCount[1])
oddCount, evenCount = oddEvenCount
print(oddCount)
print(evenCount)
```

ÖRNEK: Fibonacci serisini istenilen eleman sayısına göre bastıran programı yazınız.

```
"""n = int(input("Fibonacci serisini kaçınıcı elemana kadar görüntülemek
istiyorsunuz?"))
first = 0
second = 1
third = first + second
print(first, second, third, end=" ")
for i in range(0, x-3):
    first = second
    second = third
    third = second + first
    print(third, end=" ")
"""
def fibo(n):
    if type(n) == int:
        if n <= 0:
            print("lütfeñ pozitif bir tamsayı giriniz")
        else:
            first = 0
            if n == 1:
                print(first)
                return # return döñgülerdeki break işlemini sağlar
            second = 1
            print(first, second, end=" ")
            for i in range(n - 2):
                third = first + second
                first = second
                second = third
                print(third, end=" ")
```

```

        print()
    else:
        print("lütfen pozitif bir tamsayı giriniz")
        print()

fibonacci(9)
fibonacci(45)

```

#Örnek: faktoriyel hesaplama

#iterative(tekrarlanan)

```

def faktoriyel(n):
    if type(n) == int and n>=0:
        sonuc = 1
        for i in range(2, n+1):
            sonuc *= i
        print(sonuc)
    else:
        return "lütfen doğal sayı giriniz"
faktoriyel(9)

```

#recursive(öz yinelenen)

```

def recfaktoriyel(n):
    if type(n) == int and n >= 0:
        if n==0:
            return 1
        return n * recfaktoriyel(n-1)
    else:
        return "lütfen doğal sayı giriniz"
print(recfaktoriyel(9))

```

"""Lambda anonymous function (ismi olmayan fonksiyonlar)"""

```

def kareAl(sayi):
    return sayi * sayi
print(type(kareAl))
print(kareAl(5))

```

```

lambdakareAl = lambda sayi : sayi * sayi
print(lambdakareAl)
print(lambdakareAl(8))

```

```

lambdatopla = lambda sayi1, sayi2 : sayi1 + sayi2
print(lambdatopla)
print(lambdatopla(7, 87))

```

```

kareAlFonksiyonu = kareAl
print(kareAlFonksiyonu(9))
print(type(kareAlFonksiyonu))

```

#Örnek1 Bir sayının küpünü alan bir fonksiyon yazın.

```
kupAl = lambda sayi: sayi*sayi*sayi
```

```
print(kupAl(9))
```

#Örnek2 Bir stringi tersten yazan anonymous fonksiyon yazın.

```
tersYaz = lambda str: str[::-1]
```

```
print(tersYaz("BERK BATUHAN DEVRAN"))
```

```
"""ÖZEL FONKSİYONLAR(map, filter, reduce)"""
```

#map() return değeri ne ise o elemanlar maplenmiş listeye eklenir. Normal liste ile aynı boyuttadır.

#map = eşlemek planlamak

```
sayilar = [8, 9, 23, 24, 13, 67, -8, 76, -78, -66]
```

```
def ikiIleCarp(liste):
```

```
    sonuc= []
```

```
    for sayi in liste:
```

```
        sonuc.append(sayi*2)
```

```
    return sonuc
```

```
print(ikiIleCarp(sayilar))
```

```
print("-----")
```

```
def ikiIleCarp(sayi):
```

```
    return 2*sayi
```

```
sayilar2 = list(map(ikiIleCarp,sayilar))
```

```
print(sayilar2)
```

```
sayilarınİkiKati = list(map(lambda sayi: sayi*2, sayilar))
```

```
print(sayilarınİkiKati)
```

#filter() filtrelenen veriler döner. (return bool olduğunda True ise

filrelenmiş listeye değeri ekler)

#filter=filtrelemek

```
sayilar = [2, 3, 4, 5, -6, -7, -9]
```

```
def pozitifMi(sayi):
```

```
    return sayi>0 #return burada if görevi yapar true dönerselisteğe ekleme yapılır.
```

```
pozitifSayilar = filter(pozitifMi, sayilar)
```

```
print(pozitifSayilar)
```

```
print(list(pozitifSayilar))
```

```
pozitifSayilar = list(filter(pozitifMi, sayilar))
```

```
negatifSayilar = [sayi for sayi in sayilar if sayi not in pozitifSayilar]
```

```
negatifSayilar2 = list(filter(lambda sayi: sayi < 0, sayilar))
```

```
print(negatifSayilar)
```

```
print(negatifSayilar2)
```

```
negatifSayilar.sort()#küçükten büyüğe sıralandı
```

```
print(negatifSayilar)
```

```
negatifSayilar.sort(reverse=True)#büyükten küçüğe sıralandı
```

```
print(negatifSayilar)
```

```
print("-----")
```

```

sayilar = [8, 9, 23, 24, 13, 67, -8, 76, -78, -66]
def cifSayi(sayi):
    return sayi % 2 == 0
çiftSayilar = list(filter(cifSayi, sayilar))
print(çiftSayilar)

```

```

tekSayilar = list(filter(lambda sayi: sayi % 2 != 0, sayilar))
print(tekSayilar)

```

```

# reduce() bir listedeki elemanları dönüş değeriyle beraber gezmeyi sağlar
# return edilen değer ve sıradaki eleman parametre olarak kullanılır.
# reduce = indirgemek
sayilar = [3, 2, 4, 6, -7]
sayilar2 = [3.2, 5.4, 7.8, 9.8]

```

```

def sayilarTopla(liste):
    sonuc = 0
    for sayi in liste:
        sonuc += sayi
    return sonuc
print(sayilarTopla(sayilar))
print(sayilarTopla(sayilar2))

```

```

from functools import reduce
toplama = reduce(lambda a, b : a + b, sayilar)
print(toplama)
def toplama(sayi1, sayi2):
    return sayi1 + sayi2
toplama3 = reduce(toplama, sayilar)
print(toplama3)

```

```

liste = [1, 2, 3, 4]
sonuc = reduce(lambda a,b : a*b, liste)
print(sonuc)

```

```

n = int(input("lütfen faktoriyelini öğrenmek istediğiniz sayıyı giriniz"))
liste = list(range(1, n+1))
sonuc = reduce(lambda a,b : a*b, liste)
print(sonuc)

```

#ÖRNEK

Bir liste içerisindeki en küçük ve en büyük sayıyı reduce fonksiyonunu kullanarak bulunuz. Bir de lambda kullanarak bulunuz.

```

def enBüyükSayi(a,b):
    if a>b:
        return a

```



```

        else:
            return b

liste = [4, 5, 6, -5, -7, 0, 99]
print(reduce(enBüyükSayı, liste))

print(reduce(lambda a, b: a if a>b else b, liste))

def enKüçükSayı(a,b):
    if a<b:
        return a
    else:
        return b
print(reduce(enKüçükSayı, liste))
print(reduce(lambda a, b: a if a<b else b, liste))

#ÖRNEK
#Notların katsayısını ve notları ayrı ayrı tutan 2 liste bulunmaktadır.
#Bu 2 listeyi kullanarak her bir not için dönem sonu notunu hesaplayın.
# NOT: Dönem sonu notlarının toplamı 50 üzerindeyse öğrenci geçmiş
altındaysa kalmış kabul edilmektedir.
katsayılar = [0.15, 0.25, 0.6]
notlar = [50, 70, 90]
def katsayıÇarp(katsayı,_not):
    return katsayı*_not
dönemSonuNotlar = list(map(katsayıÇarp, katsayılar, notlar))
dönemOrtalaması = reduce(lambda a,b: a+b, dönemSonuNotlar)
print(dönemOrtalaması)
if dönemOrtalaması>=50:
    print("geçtiniz")
else:
    print("kaldınız")

"""ZIP FONKSİYONU"""
numaralar = [123, 234, 456, 6577]
isimler = ["berk", "batuhan", "devran", "bbd"]
bilgiler = zip(numaralar, isimler)
print(bilgiler)
print(list(bilgiler))

yaslar = (23, 32, 12, 45)
bilgiler2 = zip(numaralar, isimler, yaslar)
#print(list(bilgiler2))# fazla gelen eleman eşlenmez

for bilgi in bilgiler2:
    print(bilgi)

bilgiler3 = zip(numaralar, isimler, yaslar)
for numara, isim, yas in bilgiler3:

```

```

    print("öğrenci no {} isim {} yaş {}".format(numara, isim, yas))

#ziplenerek oluşturulan bir liste sadece bir kez kullanılabilir

bilgilerSet = set(zip(numaralar, isimler, yaslar))
print(bilgilerSet)

"""bilgilerDict = dict(zip(numaralar, isimler, yaslar))
print(bilgilerDict)""" # sözlükler içli olarak dict edilemez

"""ENUMERATE FONKSİYONU"""

sayilar = (1, 2, 3, 4, 5)
print(sayilar)
print(list(enumerate(sayilar)))

enumerateSayılar = enumerate(sayilar)
print(enumerateSayılar)
print(list(enumerateSayılar))

enumerateSayılar = enumerate(sayilar)
print(tuple(enumerateSayılar))

def myEnumrate(list):
    count = 0
    newList = []
    for i in list:
        newList.append((count, i))
        count += 1
    return newList
print("myEnumerated list :", myEnumrate(sayilar))
print("myEnumerated list :", tuple(myEnumrate(sayilar)))

rehber = {'berk':5, 'batuhan':7}
print(list(enumerate(rehber)))# anahtar kelimeler tutulur değerler kabulur
print(list(enumerate(rehber.values()))) # bu şekilde de değerler üzerinden
bir numaralandırma işlemi yapılabilir

"""all() ve any() fonksiyonu"""
# all() bütün değerler için True ise True any() en az 1 değer için True
ise True döndürür.
print(all([True, True, False]))
print(all((True, True, False)))
print(all((True, True, True)))

print(any([True, True, False]))

```

```
print(any((True, True, False)))
print(any((True, True, True)))
```

```
def myAny(liste):
    for i in liste:
        if i == True:
            return True
    return False
```

```
def myAll(liste):
    for i in liste:
        if i == False:
            return False
    return True
```

```
print(myAll([True, True, False]))
print(myAll((True, True, False)))
print(myAll((True, True, True)))
```

```
print(myAny([True, True, False]))
print(myAny((True, True, False)))
print(myAny((True, True, True)))
```

```
"""Bir Fonksiyonun Fonksiyon döndürmesi"""
```

```
def bilgiVer(func):
```

```
    print("bilgi verildi")
    return func
```

```
def kullanıcıyıGoruntule():
```

```
    return "Berk Batuhan Devran"
```

```
print(bilgiVer(kullanıcıyıGoruntule))# bu şekilde kullanıcıyı görğntğle
fonksiyonunun sadece adres bilgisi alınabilir
print(bilgiVer(kullanıcıyıGoruntule()))
```

```
"""KKONU - DECORATORS (İstedğimiz şekilde fonksiyonları süslememizi
sağlar)"""
```

```
def funcInfo(func):
    func()
```

```
def soruSor():
```

```
    print("fonksiyonun çalışması başladı")
    print("Soru sordum")
    print("fonksiyonun çalışması bitti")
```

```
def cevapVer():
```

```
    print("fonksiyonun çalışması başladı")
    print("cevap verdim")
    print("fonksiyonun çalışması bitti")
```

```
soruSor()
```

```
cevapVer()
```

```

def funcInfo(func):
    print("fonksiyonun çalışması başladı")
    func()
    print("fonksiyonun çalışması bitti")
@funcInfo
def soruSor():
    print("Soru sordum")

def cevapVer():
    print("cevap verdim")

"""soruSor()
cevapVer()
funcInfo(soruSor)
funcInfo(cevapVer)"""
print(soruSor)#None ifadesi geldi--- funncInfo fonksiyonunun return
değerini verir. Fonksiyonların return değeri varsayılanda none dir.
print(type(soruSor))# type = str
print(cevapVer)#<function cevapVer at 0x000001F1BA5B5A60> ifadesi geldi

```

```

print("-----")
print("-----")
print("-----")

```

```

def funcInfo(func):
    print("fonksiyonun çalışması başladı")
    func()
    print("fonksiyonun çalışması bitti")
@funcInfo
def soruSor():
    print("Soru sordum")
@funcInfo
def cevapVer():
    print("cevap verdim")
# bu haliyle program çalıştırıldığı zaman biz çağırmasak da fonksiyonlar
çalışıyor

print("-----")

```

```

def funcInfo(func):
    def bilgiVer(): #inner fonksiyon
        print("fonksiyonun çalışması başladı")
        func()
        print("fonksiyonun çalışması bitti")
        return "fwfs"

```

```

        return bilgiVer
@funcInfo
def soruSor():
    print("Soru sordum")
    return "akfsdşjşvs"
@funcInfo
def cevapVer():
    print("cevap verdim")
# artık fonksiyonlar kendi kendine çağırılmıyor. :)

funcInfo(soruSor)()
funcInfo(cevapVer)()

print("-----")

print(soruSor) # <function funcInfo.<locals>.bilgiVer at
0x0000001D68B705E50>
print(soruSor())
"""
fonksiyonun çalışması başladı
Soru sordum
fonksiyonun çalışması bitti
fwfs"""
soruSor()
"""

fonksiyonun çalışması başladı
Soru sordum
fonksiyonun çalışması bitti
"""

cevapVer()
"""

fonksiyonun çalışması başladı
cevap verdim
fonksiyonun çalışması bitti"""

#Not:funcInfo(soruSor)=soruSor()

print("-----")

```

#ÖRNEK

```

def funcInfo(func):
    def inner():
        print("fonksiyonun çalışması başladı")
        func()
        print("fonksiyonun çalışması bitti")
    return inner #iner fonksiyonun çağırılmamasına() dikkat ediniz
def soruSor(isim, söz):
    print(isim, ":", söz)

```

```

def cevapVer(isim, söz):
    print(isim,":", söz)
soruSor("Batuhan", "Nasılsın?")
cevapVer("Kaan","İyiym, sen nasılsın?")

print("-----")

def funcInfo(func):
    def inner(isim, söz):
        print("Konuşma çalışması başladı")
        func(isim, söz)
        print("Konuşma çalışması bitti")
    return inner    #iner fonksiyonun çağırılmamasına() dikkat ediniz

def soruSor(isim, söz):
    print(isim,":", söz)

def cevapVer(isim, söz):
    print(isim,":", söz)

funcInfo(soruSor)# Bu ifade inner fonksiyonu tmsil eder ve burada inner fonksiyon çağırılmadığı() için program bir bilgi döndürmez
funcInfo(soruSor)("Batuhan", "Nasılsın?")

print("-----")
def funcInfo(func):
    def inner(*args, **kwargs):
        print("Konuşma çalışması başladı")
        func(*args, **kwargs)
        print("Konuşma çalışması bitti")
    return inner    #iner fonksiyonun çağırılmamasına() dikkat ediniz

def soruSor(isim, söz,yas):
    print(isim,":", söz, "yaşı", yas)

def cevapVer(isim, söz):
    print(isim,":", söz)

funcInfo(soruSor)# Bu ifade inner fonksiyonu tmsil eder ve burada inner fonksiyon çağırılmadığı() için program bir bilgi döndürmez
funcInfo(soruSor)("Batuhan", "Nasılsın?",yas=22)

# Örnek: Aşağıdaki çıktıyı veren programı dekoratör kullanarak yazınız.
"""
*****
%%%%%%%%%%%%%%
BERK BATUHAN DEVRAN

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*****
"""
```

```
def yüzdelikBas(func):
    def inner(*args, **kwargs):
        print("%"*30)
        func(*args, **kwargs)
        print("%"*30)
    return inner
```

```
def yıldızBas(func):
    def inner(*args, **kwargs):
        print("%"*30)
        func(*args, **kwargs)
        print("%"*30)
    return inner
```

```
def yazdir(isim):
    print(isim)
```

```
yıldızBas(yüzdelikBas(yazdir))("BERK BATUHAN DEVRAN")
```

```
print("-----")
```

```
def yüzdelikBas(func):
    def inner(*args, **kwargs):
        print("%"*30)
        func(*args, **kwargs)
        print("%"*30)
    return inner
```

```
def yıldızBas(func):
    def inner(*args, **kwargs):
        print("%"*30)
        func(*args, **kwargs)
        print("%"*30)
    return inner
```

```
@yüzdelikBas
@yıldızBas
def yazdir(isim):
    print(isim)
```

```
yazdir("BERK BATUHAN DEVRAN")
```

Örnek: Liste içerisindeki sayıların karesini ve karekökünü bulan 2 ayrı
fonksiyonun çalışma sürelerini
hesaplayan programı yazınız.

```
#import math
from math import sqrt
import time
print(time.time(), "saniye")# 1 ocak 1970 ten beri geçen saniye sayısı
print(time.time()/60, "dakika")
```

```
def calismaSuresiHesapla(func):
    def inner(*args, **kwargs):
        x = time.time()
        func(*args, **kwargs)
        y = time.time()
        print(y-x,"saniyede çalışmıştır")
    return inner
```

```
@calismaSuresiHesapla
def karekökAl(sayilar):
    sayilar = [sqrt(sayi) for sayi in sayilar]
    print(sayilar)
sayilar = list(range(0,1000))
karekökAl(sayilar)
print("-----")
@calismaSuresiHesapla
def kareAl(sayilar):
    sayilar = [sayi**2 for sayi in sayilar]
    print(sayilar)
sayilar = list(range(0,1000))
kareAl(sayilar)
```

#Örnek: İki sayıyı bölme işleminde otomatik olarak büyük olanı küçük olana bölen programı yazınız.

```
def buyuguBol(sayi1, sayi2):
    if sayi1 < sayi2:
        sayi1, sayi2 = sayi2, sayi1
    print(sayi1/ sayi2)
```

```
buyuguBol(12,2)
buyuguBol(2,12)
```

```
print("-----")
```

```
def form_bol(func):
    def inner(*args, **kwargs):
        sayi1 = args[0]
        sayi2 = args[1]
        if sayi1 < sayi2:
            sayi1, sayi2 = sayi2, sayi1
```



```

        func(sayi1, sayi2)
    return inner
@form_bol
def bol(sayi1, sayi2):
    print(sayi1/sayi2)
bol(3,6)
bol(6,3)

print("-----")
#Bir kelimenin palindrome olup olmadığını gösteren programı yazınız
"KABAK=KABAK"
def isPalindrome(s):
    return s == s[::-1]
kelime = "kabak"
sonuc = isPalindrome(kelime)
if sonuc: #sonuc == True: yazmanın kısaltması
    print(kelime, "palindromdur")
else:
    print(kelime, "palindrom değildir")
print(isPalindrome("kabak"))
print("-----")
#ÖRNEK: Bir kullanıcının isim, doğum tarihi, email ve şifre bilgisi
bulunmaktadır.
#     Bu isim, doğum tarihi, email ve şifre bilgisine "isim", "doğum
tarihi" ve "sifre" anahtar kelimeleriyle
#     ulaşılmak istenmektedir.
#     Bu kullanıcıların yaş ortalamasının üzerinde kalan kullanıcıyı
filtreleyen programı yazınız.

user1 = {"isim": "fatih", "doğumTarihi": 1990, "email": "ssgdsgs@gmail.com",
"sifre": 1234}
user2 = {"isim": "aysel", "doğumTarihi": 1992, "email": "fgrhr@gmail.com",
"sifre": 1235}
user3 = {"isim": "kaan", "doğumTarihi": 1994, "email": "kjhkgs@gmail.com",
"sifre": 1236}

users = [user1, user2, user3]
print(users)

def ortBul(users):
    toplam = 0
    for user in users:
        toplam += 2021 - user['doğumTarihi']
    return toplam/len(users)
usersYasOrt = ortBul(users)
def yasFiltrele(user):
    return 2021-user['doğumTarihi'] > usersYasOrt

ortUstuUsers = filter(yasFiltrele, users)
print(list(ortUstuUsers))

```

```
print("-----")
#ÖRNEK:Kullanıcıların bilgilerinin içerisinde "doğumTarihi" bilgisinşn
yanı sıra bir de "guncelYas"
# bilgisi tutulmak istenmektedir. Bunu sağlayan programı yazınız(map
fonksiyonu kullanınız).
#Yeni kullanıcılar listesi ile ilk önceki kullanıcılar listesi ayrı ayrı
tutulmaktadır. Var olan liste değişmemelidir.
#İpucu: python deep copy(araştırın).
```

```
user1 = {"isim":"fatih", "doğumTarihi":1990, "email":"ssgdsgs@gmail.com",
"sifre":1234}
user2 = {"isim":"aysel", "doğumTarihi":1992, "email":"fgrhr@gmail.com",
"sifre":1235}
user3 = {"isim":"kaan", "doğumTarihi":1994, "email":"kjhkgs@gmail.com",
"sifre":1236}
```

```
users = [user1, user2, user3]
```

```
"""# SÖZLÜĞE EKLEME YAPMAK
user1['guncelYas'] = 2021 - user1['doğumTarihi']
print(users)"""
```

```
#attribute=özellik
def attrEkle(user):
    user['guncelYas'] = 2021 - user['doğumTarihi']
    return user
"""
```

```
# shallow copy (yeni liste kopyalanan listenin adresi üzerinden
oluşturuluyor, yani newUsers ile users aynı adresi paylaşır.)
newUsers = list(map(attrEkle, users))
print(newUsers)
print(users)#istemese de users listesi de güncellendi
```

```
# deep copy (yeni liste kopyalanan listeden farklı bir adreste depoanır.)
# deep copy copy modülü kullanılarak gerçekleştirilir.
"""
```

```
import copy
newUsers = copy.deepcopy(users)
newUsers = list(map(attrEkle, users))
print(newUsers)
print(users)
print("-----")
#Örnek: Bir listede 0-50 arasındaki sayılar tutulmaktadır, bu sayıların
toplamını reduce fonksiyonunu kullanarak hesaplayınız
```

```
from functools import reduce
```

```

sayilar = [range(0,51)]
print(sayilar)
sayilar = [*range(0,51)]
print(sayilar)
def toplam(s1, s2):
    return s1 + s2
sayılarınToplamı = reduce(toplam, sayilar)
print(sayılarınToplamı)

print("-----")
#Örnek: bir sistem girişinde log kaydı tutan bir program yazılacaktır.
#İpucu: şu anki tarih ve saat bilgilerine erişmek için datetime modülünün
datetime sınıfı import edilmelidir.
"""OUTPUT - ÇIKTI
admin başarılı bir şekilde sisteme girdi. Tarih: 2021-09-11 03.11.33
asdmin adında bir giriş denendi.
Tarih: 2021-09-11 03.11.38 admin hatalı şifre girdi. Tarih:2021-09-11
03.11.46 """
from datetime import datetime
print(datetime.now())

def log(func):
    def inner(*args):
        message = func(*args)
        with open("Dosyalar/logs.txt", "a", encoding="utf-8") as dosya:
            dosya.write(message+ "Tarih: "+ str(datetime.now())+"\n")
        return inner

_username = "admin"
_password = 1234

@log
def sistemGiris(username, password):
    if username != _username:
        print("Böyle bir kullanıcı bulunmamaktadır")
        return username + "adında bir giriş denendi"
    elif password != _password:
        print("Hatalı şifre girdiniz")
        return username + "hatalı şifre girdi"
    else:
        print("Sisteme hoşgeldiniz")
        return username + "sisteme başarılı bir şekilde girdi"

sistemGiris("madmin",4321)
sistemGiris("admin", 1234)

```

```
"""import math
print(math.sqrt(16))
```

```
# as = alias
import math as m
print(m.sqrt(16))
```

```
from math import * #artık sadece fonksiyon ismi tazılarak kullanılabilir
print(sqrt(16))
```

```
from math import sqrt
print(sqrt(16))
```

```
from math import sqrt, factorial, floor, ceil
print(factorial(5))
print(floor(8,9))
print(ceil(9,1))"""
```

```
from math import sqrt
def sqrt(*args):
    return "Bu fonksiyon override edilmiştir"
print(sqrt(64))
```

```
import math
help(math)
print(dir(math))
```

```
import hesap
toplam = hesap.topla(5,6)
print(toplam)
daireCevre = hesap.daireCevreHesapla(8)
print(daireCevre)
daireAlan = hesap.daireCevreHesapla(7)
print(daireAlan)
print("-----")
#Konu: Special Variable olan __name__ değişkeninin işlevi
#     __name__ ana modülde kullanıldığında __main__ değerine sahiptir
#     Fakat başka modül import edildiğinde ve o modülde __name__
kullanıldıysa o zaman
#     kullanılan modülün ismine sahiptir.
```

```
#HATIRLATMA: Bir modül import edildiğinde edinilen modüldeki kodlar
kullanıldığı modülü içerisine yerleştirilir.
```

```
import hesap as h
print(h.__name__)
print(__name__)
print("-----")
#ÖRNEK: Sayı tahmini oyunu/Bilgisayar 1-50 arasında 1 sayı tutacak ve
```

kullanıcı bu sayıyı
tahmin etmeye çalışacak
Kullanıcının tahmin etmesi için sadece 5 hakkı vardır
Program oyun başlamadan önce 5 saniye beklemede kalmalıdır.(sleep)
#1-) time modülünün içerisinde bulunan sleep fonksiyonundan faydalanılacaktır.
sleep metodunun ne işe yaradığını ve nasıl çalıştığını kontrol edin.
#2-) random modülünün randint fonksiyonundan faydalanılacaktır.
randint fonksiyonunun ne işe yaradığını ve nasıl çalıştığını kontrol edin.

```
from time import sleep
from random import randint
print("Sayı tahmin oyunu başlıyor, lütfen bekleyiniz")
sleep(2)
sayi = randint(0,50)
print(sayi)
tahminHakkı = 5
while True:
    if tahminHakkı > 0:
        tahmin = int(input("lütfen tahmininizi giriniz"))
        tahminHakkı -=1

        if tahmin < sayi:
            print("lütfen sayıyı arttırınız")
        elif tahmin > 0:
            if tahminHakkı > 0:
                print("lütfen sayıyı azaltınız")
            else:
                print("kazandınız")
        else:
            print("tahmin hakkınız kalmamıştır")
            break
```