```
for i in range (5):
  for j in range (2):
    print (j)
```

```
 0
 1
 0
 1
 0
 1
 0
 1
 0
 1
```

```
try:
  i=1
  j=1
  while i<5:
    if (j<2):
      break
    i+=1
    j+=1
    print (j)
except:
  print("error")
finally:
  print('finally ')
```

```
 finally
```

```
class Bank:
    name=[]
    accountid=[]
    c=1
    balance=[]
    def create(self):
        self.name=input('enter your name hare: ')
        self.balance=int(input('enter amount atleast Rs 2000: '))
        if self.balance>=2000:
            print('account has been created with id',Bank.c,'on the name of',self.name,'havin
            Bank.name.append(self.name)
            Bank.balance.append(self.balance)
            Bank.c+=1
        else:
            print('Balance is less than 2000 Plz create again')
    def deposite(self):
```

```
            self.id=int(input("enter yout account id here: "))
            print('your name is',Bank.name[self.id-1])
            print('your balance is',Bank.balance[self.id-1])
            self.a=int(input('enter the amount you want to deposite here: '))
            self.d=Bank.balance[self.id-1]+self.a
            Bank.balance.pop(self.id-1)
            Bank.balance.insert(self.id-1,self.d)
            print(Bank.balance)
            print('your amount',self.a,' has been deposited and your new balance is',self.d)
        def withdraw(self):
            self.id=int(input("enter yout account id here: "))
            print('your name is',Bank.name[self.id-1])
            print('your balance is',Bank.balance[self.id-1])
            self.d=0
            while self.d<2000:
                self.a=int(input('enter the amount you want to withdraw here: '))
                self.d=Bank.balance[self.id-1]-self.a

                if self.d>2000:
                    break
                else:
                    print('Minimum balance cannot be less than 2000')
            Bank.balance.pop(self.id-1)
            Bank.balance.insert(self.id-1,self.d)
            print('your amount',self.a,' has been withdraw and your new balance is',self.d)
        def display(self):
            self.id=int(input("enter yout account id here: "))
            print('your name is',Bank.name[self.id-1])
            print('your balance is',Bank.balance[self.id-1])

    user=Bank()
    while True:
        print("Welcome to SudoSpark Bank")
        print("1.Create Bank Account")
        print("2.Deposite Cash")
        print("3.Withdrae Cash")
        print("4.Display Account Information")
        print("0.Exit")

        choice=int(input("Enter your choice:"))
        if choice==1:
            user.create()
        elif choice==2:
            user.deposite()
        elif choice==3:
            user.withdraw()
        elif choice==4:
            user.display()
        elif choice==0:
            break
        else:
```

```
            print('Invalid Choice')
```

```
      Welcome to SudoSpark Bank
      1.Create Bank Account
      2.Deposite Cash
      3.Withdrae Cash
      4.Display Account Information
      0.Exit
      Enter your choice:1
      enter your name hare: Amit
      enter amount atleast Rs 2000: 8000
      account has been created with id 1 on the name of Amit having amount 8000
      Welcome to SudoSpark Bank
      1.Create Bank Account
      2.Deposite Cash
      3.Withdrae Cash
      4.Display Account Information
      0.Exit
      Enter your choice:3
      enter yout account id here: 1
      your name is Amit
      your balance is 8000
      enter the amount you want to withdraw here: 4500
      your amount 4500  has been withdraw and your new balance is 3500
      Welcome to SudoSpark Bank
      1.Create Bank Account
      2.Deposite Cash
      3.Withdrae Cash
      4.Display Account Information
      0.Exit
      Enter your choice:0
```

```
import numpy as np


list=[10,20,30,40]

a=np.array(list)  #convert list to array


print(a)
print(type(a))  #ndarray = n dimensional array
```

```
      [10 20 30 40]
      <class 'numpy.ndarray'>
```

```
print(type(list))
```

```
<class 'list'>
```

```
a=[[10,20,30,66],[40,50,60,70],[45,654,564,45]]
b=np.array(a)
b
```

```
array([[ 10,  20,  30,  66],
       [ 40,  50,  60,  70],
       [ 45, 654, 564,  45]])
```

```
c=b.tolist() #convert array to list
```

```
print(c)
print(type(c))
```

```
[[10, 20, 30, 66], [40, 50, 60, 70], [45, 654, 564, 45]]
<class 'list'>
```

```
b.shape #shows rows and colums
```

```
(3, 4)
```

```
b.dtype #return datatype of each column
```

```
dtype('int64')
```

```
d=['cdac',2022,33.14,True]
e=np.array(d)
print(e)
print(type(e))
type(e[2])
print(e.dtype)
```

```
['cdac' '2022' '33.14' 'True']
<class 'numpy.ndarray'>
<U32
```

```
list
```

```
[10, 20, 30, 40]
```

```
arr=np.array(list)
print(arr)
```

```
[10 20 30 40]
```

```
a=np.zeros((2,2))
print(a)
b=np.ones((2,2))
print(b)
```

```
[[0. 0.]
 [0. 0.]]
[[1. 1.]
 [1. 1.]]
```

```
a=[[10,20,30,66],[40,50,60,70],[45,654,564,45]]
b=np.array(a)
print(b)
b.reshape(4,3)  # to reshape an array
```

```
[[ 10  20  30  66]
 [ 40  50  60  70]
 [ 45 654 564  45]]
array([[ 10,  20,  30],
       [ 66,  40,  50],
       [ 60,  70,  45],
       [654, 564,  45]])
```

```
c=b.reshape(2,3,2) # 2 arrays of dimension (2,3)
print(c)
```

```
[[[ 10  20]
  [ 30  66]
  [ 40  50]]

 [[ 60  70]
  [ 45 654]
  [564  45]]]
```

```
b
```

```
array([[ 10,  20,  30,  66],
       [ 40,  50,  60,  70],
       [ 45, 654, 564,  45]])
```

```
## Slicing
print(b[1:2,2:3])
```

```
[[60]]
```

```
print(b[::-1,::-1]) #reversing
```

```
[[ 45 564 654  45]
```

```
       [ 70  60  50  40]
       [ 66  30  20  10]]
```

```
print(b.sum(axis=0)) #sum of rows
```

```
     [ 95 724 654 181]
```

```
print(b.sum(axis=1)) #sum of colums
```

```
     [ 126  220 1308]
```

```
## Linspace
np.linspace(1,10)
```

```
     array([ 1.        ,  1.18367347,  1.36734694,  1.55102041,  1.73469388,
             1.91836735,  2.10204082,  2.28571429,  2.46938776,  2.65306122,
             2.83673469,  3.02040816,  3.20408163,  3.3877551 ,  3.57142857,
             3.75510204,  3.93877551,  4.12244898,  4.30612245,  4.48979592,
             4.67346939,  4.85714286,  5.04081633,  5.2244898 ,  5.40816327,
             5.59183673,  5.7755102 ,  5.95918367,  6.14285714,  6.32653061,
             6.51020408,  6.69387755,  6.87755102,  7.06122449,  7.24489796,
             7.42857143,  7.6122449 ,  7.79591837,  7.97959184,  8.16326531,
             8.34693878,  8.53061224,  8.71428571,  8.89795918,  9.08163265,
             9.26530612,  9.44897959,  9.63265306,  9.81632653, 10.        ])
```

```
np.linspace(1,10,4)
```

```
     array([ 1.,  4.,  7., 10.])
```

```
import psycopg2
```

```
     /usr/local/lib/python3.7/dist-packages/psycopg2/__init__.py:144: UserWarning: The psycop
       """)
```

```
 con=psycopg2.connect(database='student',user='postgres',password='pgdbda', host='122.171.13.
```

```
---------------------------------------------------------------------------
OperationalError                          Traceback (most recent call last)
<ipython-input-4-cfaae2e2d7ea> in <module>()
----> 1 con=psycopg2.connect(database='student',user='postgres',password='pgdbda',
host='122.171.13.244', port='5432')
```

```
import psycopg2
conn=psycopg2.connect(database='library',user='postgres',password='pgdbda',host='127.0.0.1',p
cur=conn.cursor()
cur.execute('''select *from books''')
print(cur.fetchall())
```

```
n1=float(input("Enter 1st no. : "))
n2=float(input("Enter 2nd no. : "))
n3=float(input("Enter 3rd no. : "))
if n1>n2 :

  if n1>n3:
    print("max is n1")
  elif n1==n3:
    print("n1 & n3, both are max & equal ")
  else:
    print("max is n3")


elif n2>n1 :


  if n2>n3:
    print("max is n2")
  elif n2==n3:
    print("n2 & n3, both max & are equal")
  else:
    print("max is n3")

elif n1==n2==n3:
  print("all are equal")


else:
  print("n1 & n2,both max & are equal")
```

```
Enter 1st no. : 6
Enter 2nd no. : 5
Enter 3rd no. : 3
max is n1
```

```
n1=float(input("Enter 1st no. : "))
n2=float(input("Enter 2nd no. : "))
```

```
n3=float(input("Enter 3rd no. : "))
if n1>n2 and n1>n3:
  print("n1 is greatest")
elif n2>n1 and n2>n3:
  print("n2 is greatest")

elif n1==n2==n3:
  print("all are equal")
elif n1==n2:
  print("n1 & n2 are max &  equal")
elif n1==n3:
  print("n1 & n3 are max & equal")
elif n2==n3:
  print("n2 & n3 are max & equal")
else:
  print("n3 is greatest")
```

```
Enter 1st no. : 5
Enter 2nd no. : 3
Enter 3rd no. : 5
n1 & n3 are max & equal
```

```
b=[12, 35, 9, 56, 24]
b[0], b[-1] = b[-1], b[0]
b
```

```
[24, 35, 9, 56, 12]
```

```
b=[12, 35, 9, 56, 24]
c=b.pop()
d=b.pop(0)
b.append(d)
b.insert(0,c)
b
```

```
[24, 35, 9, 56, 12]
```

```
a=[10,20,30,40,50]
a.insert(0,a.pop())
a.append(a[1])
a.remove(a[1])
print(a)
```

```
[50, 20, 30, 40, 10]
```

```
a= [1, 4, 5, 7, 8]
len(a)
```

```
5
```

```
a= [1, 4, 5, 7, 8]
c=0
for i in a:
  c+=1
print(c)
```

```
5
```

```
a= [1, 8, 5, 7, 8]
a[-1]="*#SADFDSFgh"

print(a.index(a[-1])+1)
```

```
5
```

```
a=[1,2,4,5,6,7]
def list_length(a):
  if not a:
    return 0
  else:
    return 1+list_length(a[1:])
#b=list_lenth(a)
list_length(a)
```

```
6
```

```
a= [1, 4, 5, 7, 8]
b=int(input("enter no. :"))
b in a
```

```
enter no. :5
True
```

```
a= [1, 4, 5, 7, 8]
b=int(input("enter no. :"))
```

```
while len(a) > 0:
  c=a.pop()
  if c==b:
    print(True)
  elif len(a) >0:
    a.pop()
  else:
    print(False)
```

```
 enter no. :5
 True
```

```
a=[1,2,3,4,5,6]
b=a.count(int(input()))
if b>0:
    print(True)
else:
  print(False)
```

```
 9
 False
```

```
a=[1,2,3,4,5,6]
a.clear()
a
```

```
 []
```

```
a=[1,2,3,4,5,6]
while len(a) > 0:
  c=a.pop()
a
```

```
 []
```

```
a=[1,2,3,4,5,6]
for i in range (len(a)):
  a.remove(a[0])
a
```

```
 []
```

```
a=[1,2,3,4,5,6]
a*=0
a
```

```
[]
```

```
a=[1,2,3,4,5,6]
del a[::]
a
```

```
[]
```

```
a=[1,2,3,4,5,6]
a.sort(reverse=True)
a
```

```
[6, 5, 4, 3, 2, 1]
```

```
a=[1,2,3,4,5,6]
a=a[::-1]
a
```

```
[6, 5, 4, 3, 2, 1]
```

```
a=[1,2,3,4,5,6]
b=[]
for i in range(len(a)):
  b.insert(0,a[i])
b
```

```
[6, 5, 4, 3, 2, 1]
```

```
a=[1,2,3,4,5,6]
b=[]
for i in range(len(a)-1,-1,-1):
  b.append(a[i])

b
```

```
[6, 5, 4, 3, 2, 1]
```

```
a=[1,2,3,4,5,6,10]
i=0
j=-1
for i in range(len(a)//2):
  a[j],a[i]=a[i],a[j]
  j-=1
a
```

```
[10, 6, 5, 4, 3, 2, 1]
```

```
a=[1,2,3,4,5,6,10]
sum(a)
```

```
31
```

```
a=[1,2,3,4,5,6,10]
b=0
for i in a:
  b=b+i
b
```

```
31
```

```
a=[1,2,3,4,5,6]
s=0
for i in range(len(a)):
  s+=a[i]

s
```

```
21
```

```
a=[1,2,3,4,5,6]
b=a[::-1]
c=0
s=0
for i in range(len(a)):
  s=a[i]+b[i]
  c+=s
c/2
```

```
21.0
```

```
a=[1,2,3,4,5,6,10]
a.sort(reverse=True)
a[1]
```

```
6
```

```
a=[1,2,3,4,5,7,6,10]
a.sort()
a[-2]
```

```
7
```

```
a=[1,2,3,4,5,8,10]
a.pop(a.index(max(a)))
max(a)
```

```
8
```

```
a=[1,2,3,4,5,6,10]
a.remove(max(a))
max(a)
```

```
6
```

```
a=[81, 52, 45, 10, 3, 2, 96]

a.sort(reverse=True)
a[:int(input())]
```

```
3
[96, 81, 52]
```

```
a=[81, 52, 45, 10, 3, 2, 96]
b=[]
for i in range(int(input())):
  b.append(max(a))
  a.remove(max(a))
b
```

```
4
[96, 81, 52, 45]
```

```
a=[81, 52, 45, 10, 3, 2, 96]
b=[]
for i in range(int(input())):

  b.append(a.pop(a.index(max(a))))
b
```

```
4
[96, 81, 52, 45]
```

```
a=[81, 52, 45, 10, 3, 2, 96]
```

```
81
```

```
a=[]

while True:

  b=input()
  if b=='':
    break
  else:
    a.append(int(b))
a
```

```
 4
 2
 8
 7

 [4, 2, 8, 7]
```