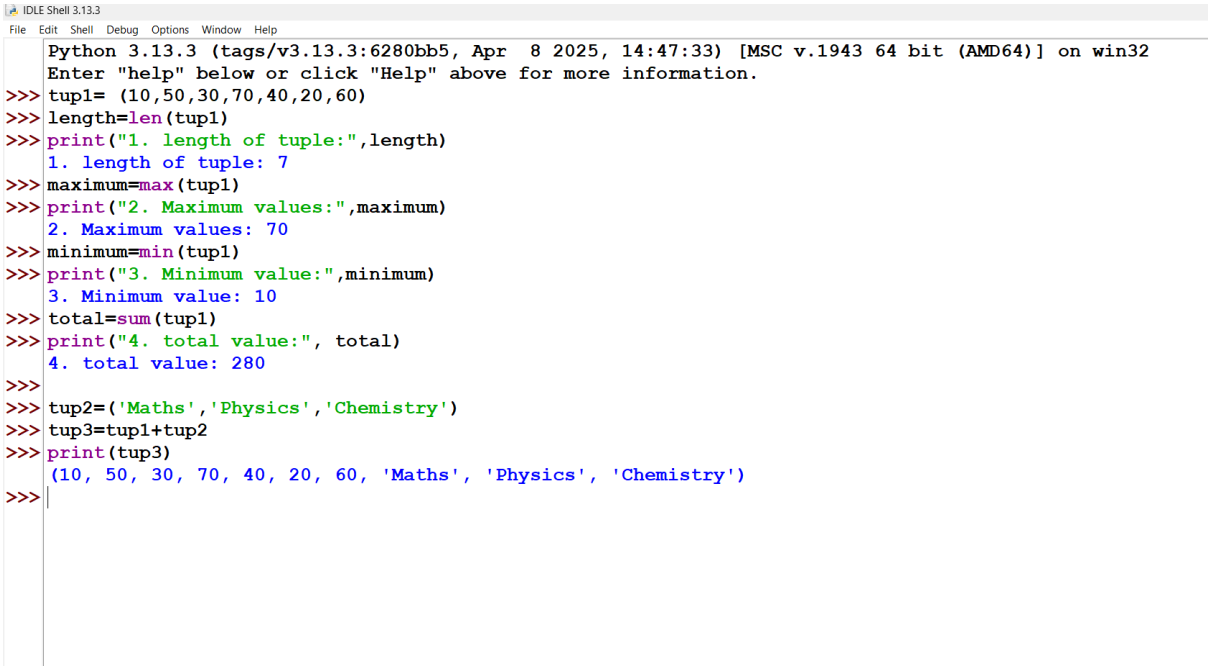
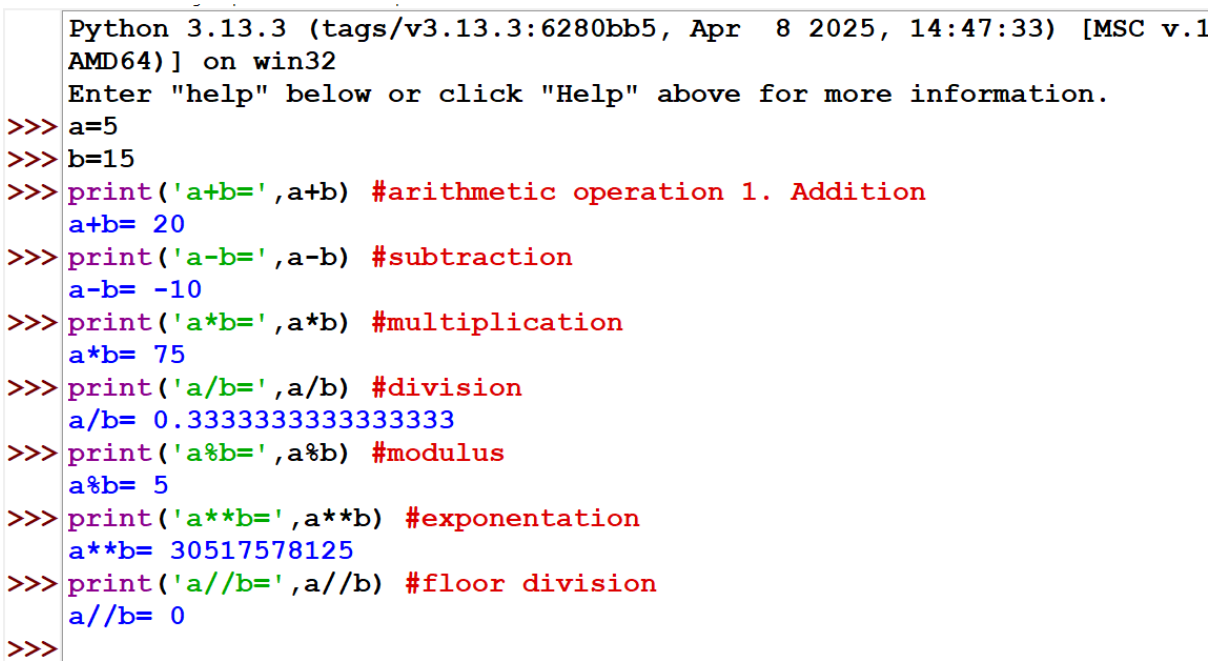
PYTHON QUESTIONS

Q1. Write Python Program to understand 5 types of built in functions to be done on Tuple.

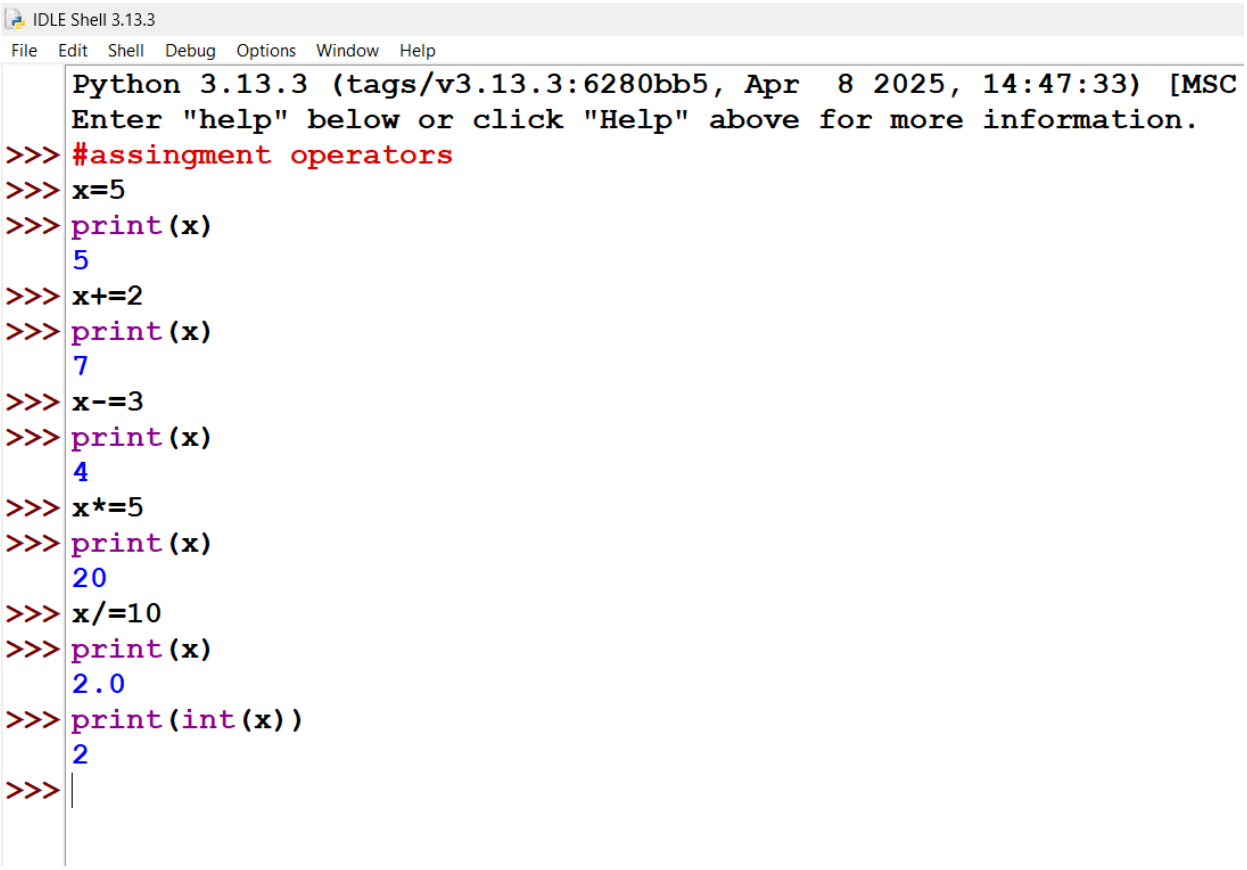


Q2. Write Python Program to understand Various types of operators in python.

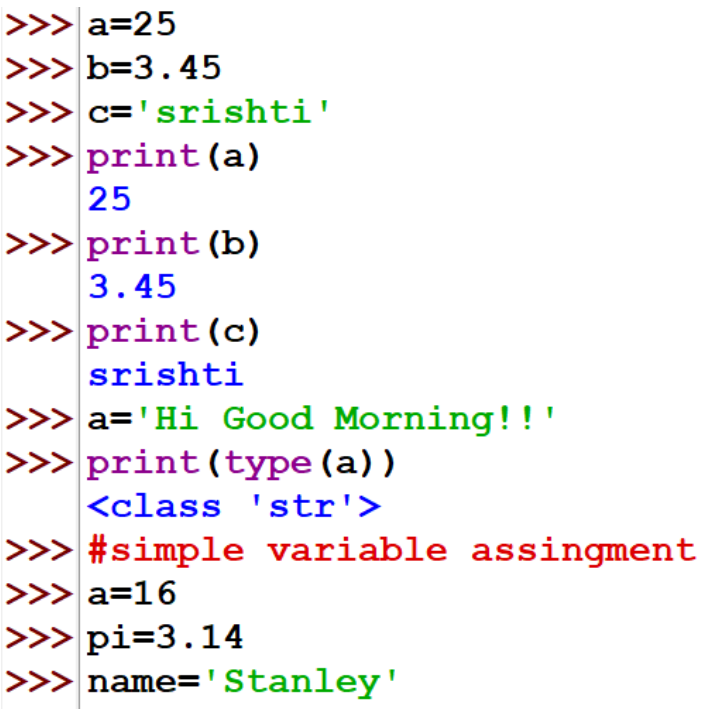


```
>>> a=20
>>> b=15
>>> print('a==b:',a==b)
    a==b: False
>>> print('a>=b:',a>=b)
    a>=b: True
>>> print('a<=b:',a<=b)
    a<=b: False
>>> print('a!=b:',a!=b)
    a!=b: True
>>> print('a>b:',a>b)
    a>b: True
>>> print('a<b:',a<b)
    a<b: False
>>> #comparision operators
```

```
>>> #logical operators
>>> a=True
>>> b=False
>>> print(a and b)
    False
>>> print(a or b)
    True
>>> print(not a)
    False
>>> print(not b)
    True
>>>
```



Q3. Write Python Program to understand Various types of variable assignments



```
>>> #multiple variable assingment
>>> a=10;pi=3.14;name='stanley'
>>> a,pi,name=10,3.14,'stanley'
>>> a=b=c=d=100
>>>
```

Q4. Write Python Program to understand sequencing, slicing, indexing operations to be done on string

```
text= 'Hi Good Morningg!!!'
for char in text:
    print(char,end=" ")
```

```
H i   G o o d   M o r n i n g g ! ! !
#sequencing
```

```
>>> #indexing
>>> text='Hello world, How are you?'
>>> print(text[5])

>>> print(text[9])
l
>>> print(text[-1])
?
>>> print(text[11])
,
>>> #slicing
>>> print(text[2:7])
llo w
>>> print(text[1:-1])
ello world, How are you
>>> print(text[-1:1])

>>> print(text[5:8])
 wo
>>>
```

Q5. Write Python Program to understand sequencing, slicing, indexing operations to be done on tuple and dictionary.

```
    Enter "help" below or click "He
>>> #tuples
>>> #indexing
>>> tup1=(10,50,20,40,90,30)
>>> print(tup1[0])
    10
>>> print(tup1[3])
    40
>>> print(tup1[-1])
    30
```

```
>>>
>>> tup2=(10,40,60,90,80)
>>> print(tup2[2:4])
    (60, 90)
>>> print(tup2[:-1])
    (10, 40, 60, 90)
>>> print(tup2[-1:])
    (80,)
>>> #slicing
```

```
>>>
>>> #sequencing
>>> tup3=(12,76,34,90,23,45)
>>> for item in tup3:
...     print(item,end=' ')
...
...
    12 76 34 90 23 45
>>>
```

```
>>> #dictionary
>>> #indexing
>>> d1={'name':'srishti','age':19,'fruit':'mango'}
>>> print(d1['name'])
srishti
>>> print(d1['age'])
19
```

```
#sequencing
dict_1={1:'drum',2:'flute',3:'violin'}
for key in dict_1:
    print(key,end=' ')
```

```
1 2 3
```

Q6. Write Python Program to understand Various types of built in functions to be done on string.

Q7. Write Python Program to understand Various types of loops (if-else, nested if else, elseif, for loop, while loop)

```
File  Edit  Shell  Debug  Options  Window  Help
Python 3.13.3 (tags/v3.13.3:6280bb5, Apr  8 2025, 14:4
AMD64)] on win32
Enter "help" below or click "Help" above for more info
>>> qty=int(input('enter quantity:'))
enter quantity:12
>>> price=float(input('enter price:'))
enter price:456.90
>>> if qty>1000:
...     dis=10
... else:
...     dis=0
...
>>> total=qty*price-qty*price*dis/100
>>> print('total=Rs.'+str(total))
total=Rs.5482.799999999999
>>> #if else
```

```
x=int(input('enter a number:'))#nested if-else
if x>=10:
    if x<=20:
        print('x is between 10 and 20')
    else:
        print('x is greater than 20')
else:
    print('x is less than 10')
```

```
IDLE Shell 3.13.3
File  Edit  Shell  Debug  Options  Window  Help
Python 3.13.3 (tags/v3.13.3:6280bb5, Apr  8 2
025, 14:47:33) [MSC v.1943 64 bit (AMD64)] on
 win32
Enter "help" below or click "Help" above for
more information.
>>>
===== RESTART: C:/Users/srish/AppData/Local/P
rograms/Python/Python313/ss.py ====
enter a number:12
x is between 10 and 20
>>>
===== RESTART: C:/Users/srish/AppData/Local/P
rograms/Python/Python313/ss.py ====
enter a number:5
x is less than 10
>>>
===== RESTART: C:/Users/srish/AppData/Local/P
rograms/Python/Python313/ss.py ====
enter a number:25
x is greater than 20
>>>
Ln: 15  Col: 0
```

ss.py - C:/Users/srish/AppData/Local/Programs/Python/Python313/ss.py (3.13.3)

File Edit Format Run Options Window Help

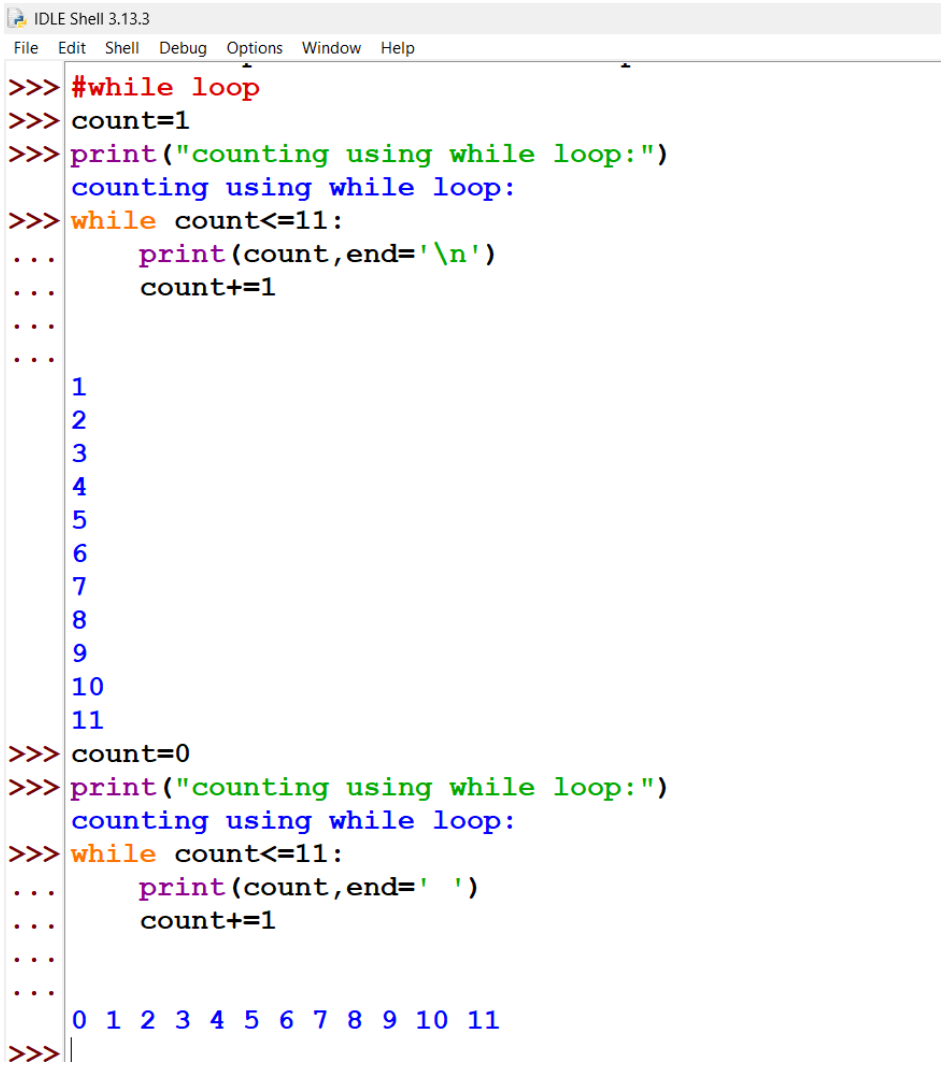
```
#elif statement

per=int(input("enter percentage"))
if per>=90:
    print('excellent!! first division!!')
elif per>=60:
    print('Good,second division')
elif per>=40:
    print('can do much better,third division')
else:
    print('FAILED')
```

IDLE Shell 3.13.3

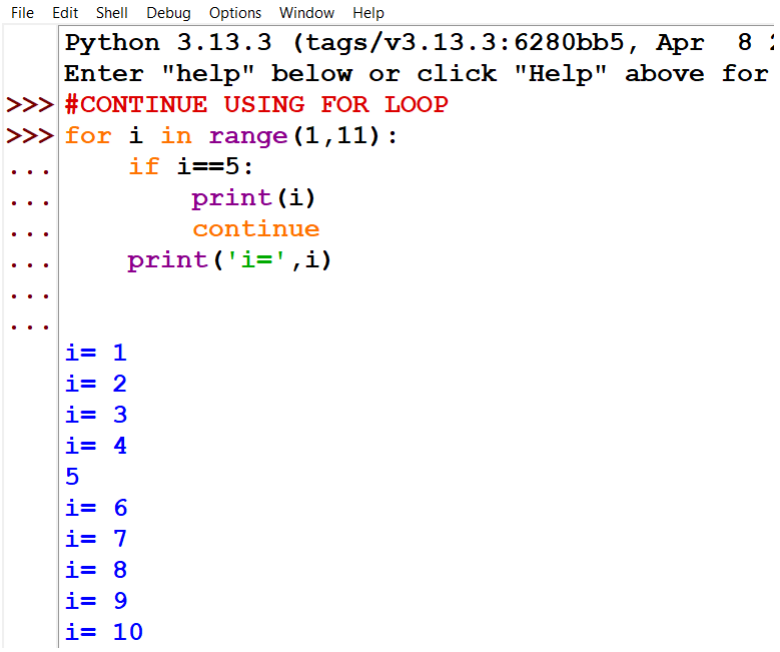
File Edit Shell Debug Options Window Help

```
Python 3.13.3 (tags/v3.13.3:6280bb5, Apr  8 2025, 14:
AMD64)] on win32
Enter "help" below or click "Help" above for more inf
>>>
===== RESTART: C:/Users/srish/AppData/Local/Programs/
enter percentage97
excellent!! first division!!
>>>
===== RESTART: C:/Users/srish/AppData/Local/Programs/
enter percentage66
Good,second division
>>>
===== RESTART: C:/Users/srish/AppData/Local/Programs/
enter percentage89
Good,second division
>>>
===== RESTART: C:/Users/srish/AppData/Local/Programs/
enter percentage35
FAILED
```

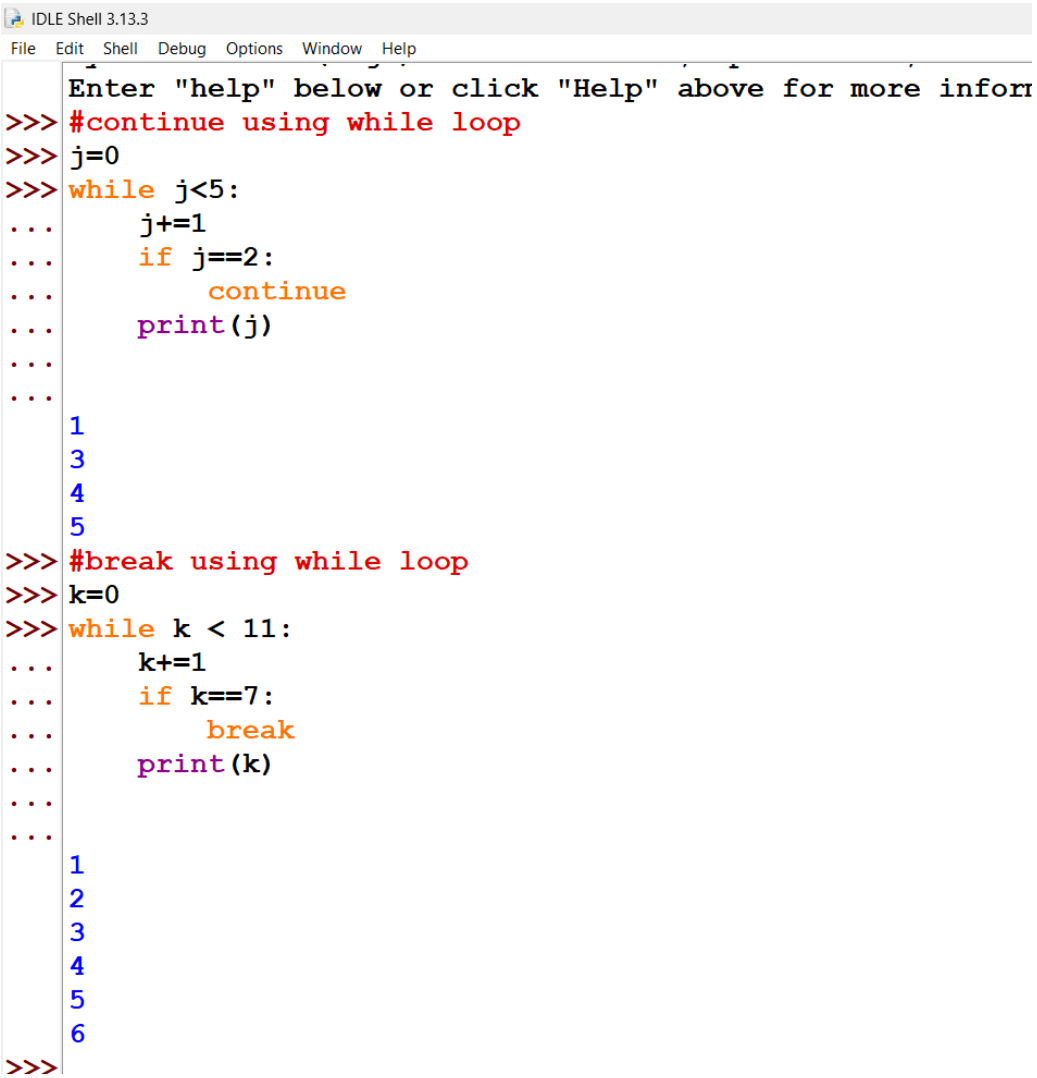
```
File  Edit  Shell  Debug  Options  Window  Help
Python 3.13.3 (tags/v3.13.3:6280bb5, Apr  8 2025, 
AMD64)] on win32
Enter "help" below or click "Help" above for more 
>>> #for loop
>>> print("numbers from 1 to 10")
numbers from 1 to 10
>>> for i in range(1,11):
...     print(i,end='\n')
...
...

1
2
3
4
5
6
7
8
9
10
>>> print("number from 1 to 10")
number from 1 to 10
>>> for i in range(1,11):
...     print(i,end=' ')
...
...

1 2 3 4 5 6 7 8 9 10
>>>
```

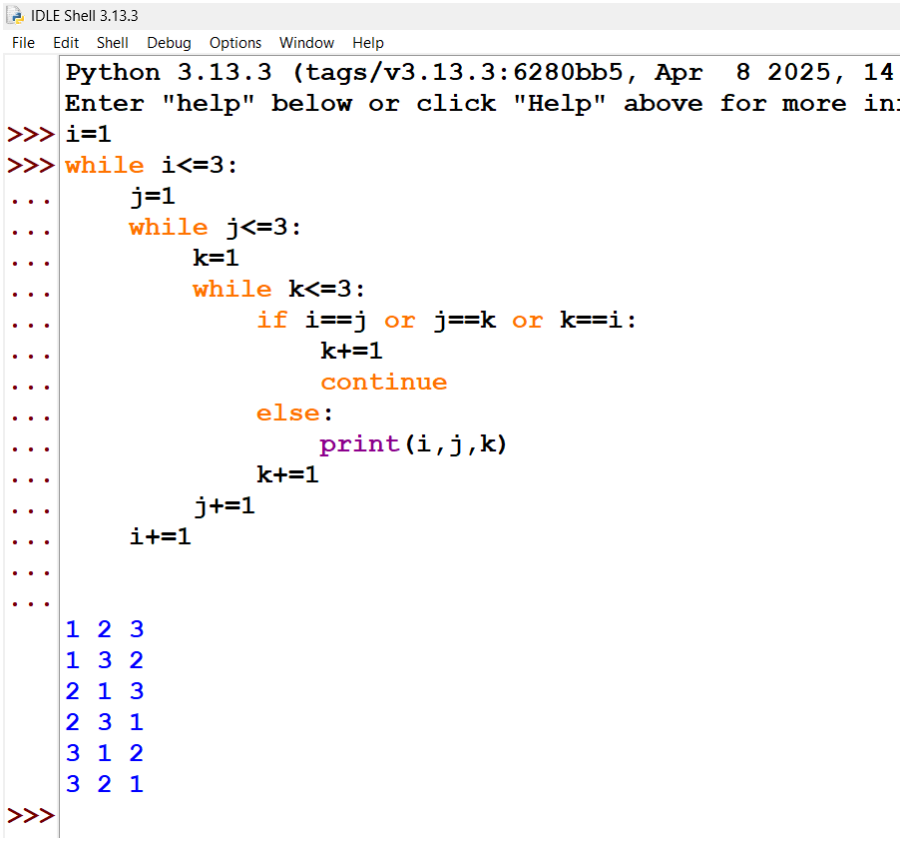


Q8. Write Python Program to understand Loops using continue & break statement.

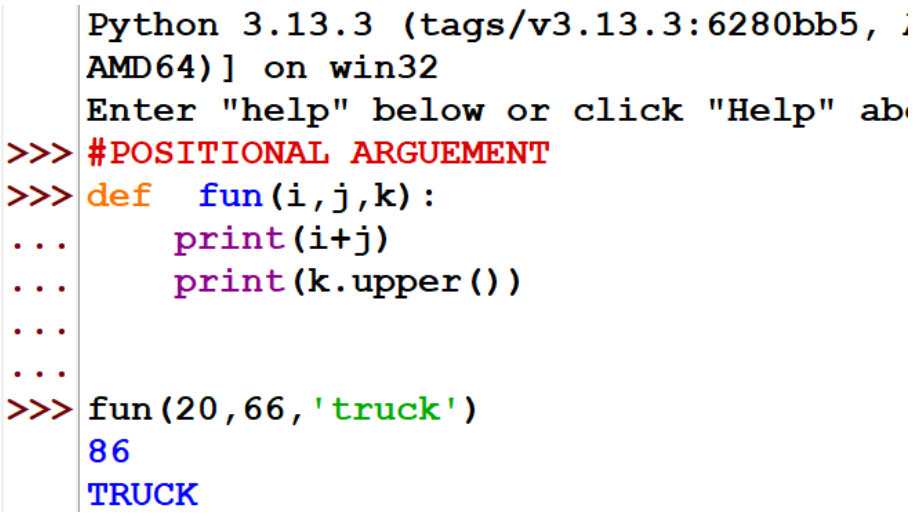


```
Python 3.13.3 (tags/v3.13.3:6280bb5, Apr  8 2025, 14:47:33) [MSC v.
1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>> #break using for loop
>>> for i in range(1,11):
...     if i==6:
...         break
...     print(i)
...
...

1
2
3
4
5
>>> #here using break statement it simply comes out of the loop when it
s condition is satisfied
```


IDLE Shell 3.13.3
File Edit Shell Debug Options Window Help
Enter "help" below or click "Help" above for more inform
>>> #continue using while loop
>>> j=0
>>> while j<5:
... j+=1
... if j==2:
... continue
... print(j)
... 1
3
4
5
>>> #break using while loop
>>> k=0
>>> while k < 11:
... k+=1
... if k==7:
... break
... print(k)
... 1
2
3
4
5
6
>>>



Q9. Write Python Program to understand Positional arguments and keyword arguments



```
>>> #keyword arguement
>>> def key(i,a,str):
...     print(i,a,str)
...
...
>>> key(a=23,i=9.8,str='jelly')
9.8 23 jelly
>>> key(str='jelly',i=23,a=9.5)
23 9.5 jelly
>>>
```

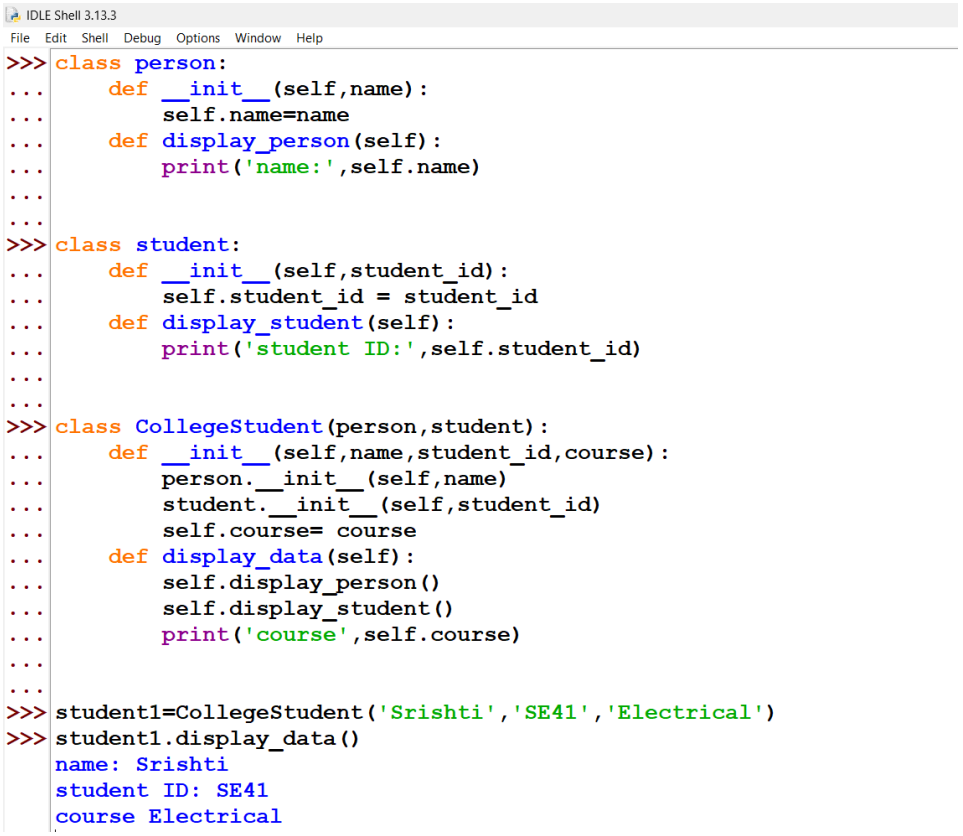
Q10. Write Python Program to understand Variable length positional argument & variable length keyword arguments.

```
IDLE Shell 3.13.3
File  Edit  Shell  Debug  Options  Window  Help
>>> #Variable length positional arguements
>>> def value(*args):
...     print()
...     for var in args:
...         print(var,end=' ')
...
...
>>> value(134)

134
>>> value(10,'srishti',78.56)

10 srishti 78.56
>>> #variable length keyword arguements
>>> def pin(**kwargs):
...     print()
...     for name,value in kwargs.items():
...         print(name,value,end=' ')
...
...
>>> pin(a=10)

a 10
>>> pin(a=25,b='srishti')

a 25 b srishti
>>> dct={'student':'srishti','age':19}
>>> pin(**dct)

student srishti age 19
```

Q11. Write Python Program to understand to create & return a list containing tuples of the form (x, x^2, x^3) for all x between 1 and 20.

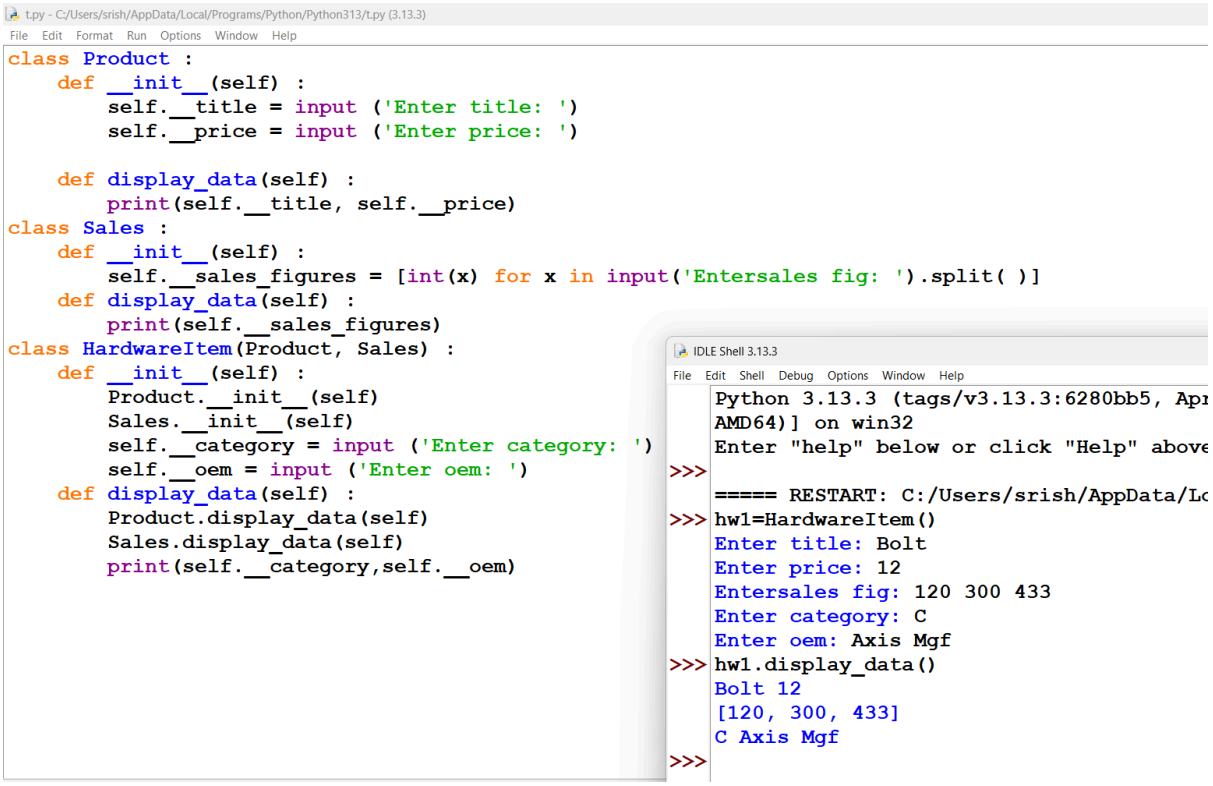
```
def gen_list():
    l1=list()            #or lst=[]
    for i in range(1,11):
        l1.append((i,i**2,i**3))
    return l1

l=gen_list()
print(l)
[(1, 1, 1), (2, 4, 8), (3, 9, 27), (4, 16, 64), (5, 25, 125), (6, 36, 216), (7, 49, 343), (8, 64, 512), (9, 81, 72
9), (10, 100, 1000)]
```

Q12. Write Python Program to understand to create user defined class for multiple inheritance.



```
IDLE Shell 3.13.3
File  Edit  Shell  Debug  Options  Window  Help
>>> class person:
...     def __init__(self,name):
...         self.name=name
...     def display_person(self):
...         print('name:',self.name)
...
...
>>> class student:
...     def __init__(self,student_id):
...         self.student_id = student_id
...     def display_student(self):
...         print('student ID:',self.student_id)
...
...
>>> class CollegeStudent(person,student):
...     def __init__(self,name,student_id,course):
...         person.__init__(self,name)
...         student.__init__(self,student_id)
...         self.course= course
...     def display_data(self):
...         self.display_person()
...         self.display_student()
...         print('course',self.course)
...
...
>>> student1=CollegeStudent('Srishti','SE41','Electrical')
>>> student1.display_data()
name: Srishti
student ID: SE41
course Electrical
```

Q13. Write Python Program to understand to understand multiple inheritance using product, sales & Hardware item classes.



```
t.py - C:/Users/srish/AppData/Local/Programs/Python/Python313/t.py (3.13.3)
File  Edit  Format  Run  Options  Window  Help
class Product :
    def __init__(self) :
        self.__title = input ('Enter title: ')
        self.__price = input ('Enter price: ')

    def display_data(self) :
        print(self.__title, self.__price)
class Sales :
    def __init__(self) :
        self.__sales_figures = [int(x) for x in input('Entersales fig: ').split( )]
    def display_data(self) :
        print(self.__sales_figures)
class HardwareItem(Product, Sales) :
    def __init__(self) :
        Product.__init__(self)
        Sales.__init__(self)
        self.__category = input ('Enter category: ')
        self.__oem = input ('Enter oem: ')
    def display_data(self) :
        Product.display_data(self)
        Sales.display_data(self)
        print(self.__category,self.__oem)
```

```
IDLE Shell 3.13.3
File  Edit  Shell  Debug  Options  Window  Help
Python 3.13.3 (tags/v3.13.3:6280bb5, Apr
AMD64)] on win32
Enter "help" below or click "Help" above
>>>
===== RESTART: C:/Users/srish/AppData/Lo
>>> hw1=HardwareItem()
Enter title: Bolt
Enter price: 12
Entersales fig: 120 300 433
Enter category: C
Enter oem: Axis Mgf
>>> hw1.display_data()
Bolt 12
[120, 300, 433]
C Axis Mgf
>>>
```

Q14. Write Python Program to understand Basics of numpy operations.

```
>>> import numpy as np
>>> a=np.array([[1,2,3],[4,5,6]])
>>> a.shape
(2, 3)
>>> a=np.array([1,2,3,4,5,6])
>>> a
array([1, 2, 3, 4, 5, 6])
>>> a[0]=50
>>> a
array([50,  2,  3,  4,  5,  6])
>>>
>>> a[:3]
array([50,  2,  3])
>>> b=a[:3]
>>> b
array([50,  2,  3])
>>> array[(6,7,8)]
```

```
>>> b=a[3:]
>>> b
array([4, 5, 6])
>>> b[0]=20
>>> b
array([20,  5,  6])
>>> a
array([50,  2,  3, 20,  5,  6])
>>>
>>>
>>> a=np.array([[1,2,3,4],[5,6,7,8],[9,10,11,12]])
>>> a
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12]])
>>> a.ndim
2
>>> a.shape
(3, 4)
>>> len(a.shape)==a.ndim
True
>>> a.size
12
>>> np.linspace(0,10,num=5)
array([ 0. ,  2.5,  5. ,  7.5, 10. ])
>>> x=np.ones(2,dtype=np.int64)
>>> x
array([1, 1])
>>> arr=np.array([7,3,4,8,1,2,9,5])
>>> np.sort(arr)
```

```
    array([1, 2, 3, 4, 5, 7, 8, 9])
>>> a=np.array([1,2,3,4,5])
>>> b=mp.array([6,7,8,9,10])
    Traceback (most recent call last):
      File "<pyshell#31>", line 1, in <module>
        b=mp.array([6,7,8,9,10])
    NameError: name 'mp' is not defined. Did you mean: 'np'?
>>> a=np.array([1,2,3,4,5])
>>> b=np.array([6,7,8,9,10])
>>> np.concatenate((a,b))
    array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
>>> a=np.arange(6)
>>> print(a)
    [0 1 2 3 4 5]
>>> b=a.reshape(3,2)
>>> print(b)
    [[0 1]
     [2 3]
     [4 5]]
>>> np.reshape(a,(1,6),order='C')
    array([[0, 1, 2, 3, 4, 5]])
>>> data=np.array([1,2])
>>> ones=np.ones(2,dtype=int)
>>> data+ones
    array([2, 3])
>>> data-ones
    array([0, 1])
>>> data*data
    array([1, 4])
>>> data/data
```

```
>>> data/data
    array([1., 1.])
>>> a=np.array([1,2,3,4,5,6,7])
>>> a.sum()
    np.int64(28)
>>> b=np.array([[1,2,3],[4,5,6]])
>>> b.sum(axis=0)
    array([5, 7, 9])
>>> b.sum(axiz=1)
    Traceback (most recent call last):
      File "<pyshell#50>", line 1, in <module>
        b.sum(axiz=1)
    TypeError: _sum() got an unexpected keyword argument 'axiz'. Did you
>>> b.sum(axis=1)
    array([ 6, 15])
>>> arr=np.array([1,2,3,4,5,6])
>>> reversed_arr=np.flip(arr)
>>> print('reversed array:',reversed_arr)
    reversed array: [6 5 4 3 2 1]
```

```
>>> arr_2d=np.array([[1,2,3,4],[5,6,7,8],[9,10,11,12]])
>>> reversed_arr=np.flip(arr_2d)
>>> print(reversed_arr)
[[12 11 10  9]
 [ 8  7  6  5]
 [ 4  3  2  1]]
>>> reversed_arr_rows=np.flip(arr_2d,axis=0)
>>> print(reversed_arr_rows)
[[ 9 10 11 12]
 [ 5  6  7  8]
 [ 1  2  3  4]]
>>> reversed_arr_columns=np.flip(arr_2d,axis=1)
>>> print(reversed_arr_columns)
[[ 4  3  2  1]
 [ 8  7  6  5]
 [12 11 10  9]]
>>>
```

Q15. Write Python Program to understand 5 types of built in functions to be done on List.

```
Enter "help" below or click "Help" above for more information.
>>> list1=[7,2,4,6,1,4,9,8,5]
>>> print('length:',len(list1))
length: 9
>>> print('maximum:',max(list1))
maximum: 9
>>> print('minimum:',min(list1))
minimum: 1
>>> print('sum:',sum(list1))
sum: 46
>>> print('any true?:',any(list1))
any true?: True
>>> print('all true?',all(list1))
all true? True
>>> #all checks non zero values. if any 0 it will give False as an output
>>> print('sorted:',sorted(list1))
sorted: [1, 2, 4, 4, 5, 6, 7, 8, 9]
>>> print('reversed:',reversed(list1))
reversed: <list_reverseiterator object at 0x000001F87E112E00>
>>> print('reversed:',list(reversed(list1)))
reversed: [5, 8, 9, 4, 1, 6, 4, 2, 7]
>>> del list1[4]
>>> print(list1)
[7, 2, 4, 6, 4, 9, 8, 5]
>>> del list1[:]
>>> print(list1)
[]
>>>
```

Q16. Write Python Program to understand Basics of Matplotlib operations.

Q17. Write Python Program to understand Basics of Scipy operations.

Q18. Write Python Program to understand Basics of GUI programming.

Q19. Write Python Program to understand Function in function.

```
def refact(n):
    if n==0:
        return 1
    else:
        p=n*refact(n-1)
    return p

num= int(input("enter any number:"))
enter any number:14
fact=refact(num)
print("factorial value=",fact)
factorial value= 87178291200
```

```
>>> def factorize(n,i):
...     if i<=n:
...         if n%i==0:
...             print(i,end=' , ')
...             n=n//i
...         else:
...             i+=1
...     factorize(n,i)
...
...
>>> num=int(input("enter a number:"))
    enter a number:34
>>> print('prime factors are:')
    prime factors are:
>>> factorize(num,2)
    2 , 17 , Traceback (most recent call last):
      File "<pyshell#23>", line 1, in <module>
        factorize(num,2)
      File "<pyshell#20>", line 8, in factorize
        factorize(n,i)
      File "<pyshell#20>", line 8, in factorize
        factorize(n,i)
      File "<pyshell#20>", line 8, in factorize
        factorize(n,i)
      [Previous line repeated 1023 more times]
    RecursionError: maximum recursion depth exceeded
```

```
>>> def rsum(num):
...     if num !=0:
...         digit=num%10
...         num=int(num/10)
...         sum=digit+rsum(num)
...     else:
...         return 0
...     return sum
...
>>> n=int(input("enter a number:"))
enter a number:278
>>> rs=rsum(n)
>>> print("sum of digits:",rs)
sum of digits: 17
```

```
>>> def fibo(old,current,terms):
...     if terms>=1:
...         new=old+current
...         print(f'{new}',end='\t')
...         terms-=1
...         fibo(current,new,terms)
...
...
>>> old=1
>>> current=1
>>> print(f'{old}',end='\t')
1
>>> print(f'{current}',end='\t')
1
>>> fibo(old,current,13)
2       3       5       8       13      21      34      55      89      144     233     377     610
>>>
```

Q20. Write a Python Program to understand database operations using sqlite3.

Q21. Write a Python program for file input output operations.

#NOTE:- JOH QUESTIONS KE SOLUTION NAHI HAI WHO APNI PYTHON FILE SE DEKH LENA :)