```
1. py: To display python version and run the program on the terminal.
2. filename.py: To create a python file.
3. python filename.py: To execute the python file.
4. exit(): To exit the terminal.
5. if 2<6:
     print("Yes")
   Prints "Yes" since the condition is true.
6. x=12
   y="Hello"
   z=3/5
   Datatypes do not have to be explicitly mentioned. They are created
   according to the values assigned.
7. #This is a comment: Single line comment..
8. """This
         is a
               multi-line
                           comment""": For multi-line comment.
9. x=str(7)
   y=int(7)
   z=float(7)
   Type casting a value type to another type.
   x will be '7', y will be 7 and z will be 7.0.
10.
   x=5
   x="Johnny"
   Value of x will be "Johnny" because it will take the value that has
   been recently assigned.
11.
   a="Memory"
   print(a[2])
   m will be displayed as the 2nd index of the string is displayed.
12.
   for a in "banana":
     print(a)
   Will display
   b
   a
   n
   a
   n
   a
   as the for loop is executed by each index of the string.
13.
   txt="Yellow or red"
   print("red" in txt)
   Will display True since "red" is present in the string txt. The
   keyword "in" is used to check if a string is a part of another
   string (substring).
14.
   txt="Yellow or red"
```

```
if (“red” in txt):
  print(“red” is present)
```

Will display “red” is present as the if condition is satisfied.

15.

```
txt=”Yellow or red”
print(“Yellow” not in txt)
```

Will display False since “Yellow” is present in the string txt. The keyword “not in” is used to check if a string is not a part of another string (substring).

16.

```
txt=”Yellow or red”
if (“Yellow” not in txt):
  print(“Yellow” is present)
```

Will display “red” is present as the if condition is satisfied.

17.

```
h=”Hello”
print(h.upper())
```

Will display HELLO as all the characters are turned to its uppercases.

18.

```
h=”Hello”
print(h.lower())
```

Will display hello as all the characters are turned into its lowercases.

19.

```
s=”Hello world”
print(s[2:7])
```

Displays llo w i.e. starting from index 2 and ending at index 6 as the last index is not included. This is called slicing operator.

20.

```
s=”Hello world”
print(s[2:])
```

Displays llo world i.e. starting from index 2 until the end of the string.

21.

```
s=”Hello world”
print(s[:8])
```

Displays Hello wo i.e. starting from the start of the string to the index 7 as the last index is excluded.

22.

```
s=”Hello world”
print(s[-7:-3])
```

Displays o wo i.e. starting from -7 that is the 7th index from the end to -4 i.e. 4th index from the end excluding -3rd.

23.

```
s=”Who are you?”
print(s.strip())
```

Will display Whoareyou? i.e. the string without any white spaces.

24.

```
s=”Hello”
```

```
print(s.replace(“H”,”B”))
```

Will display Bello as the letter “H” is replaced with the letter “B”.

25.

```
s=”Hello, World”
print(s.split(“,”))
```

Will display [‘Hello’, ‘World’] as the string is split from wherever there is a “,”.

26.

```
a=”Hello, “
b=”this is Simran.”
print(a+b)
```

Displays Hello, this is Simran. since the operator “+” is used to concatenate the 2 strings.

27.

```
x=3.0
y=”Red”
print(type(x))
print(type(y))
```

Will display float followed by string. This is used to display the datatype of the variable.

28.

```
x=4
y=2
print(x+y)
>>>6
print(x-y)
>>>2
print(x*y)
>>>8
print(x/y)
>>>2
print(x**y)
>>>16
```

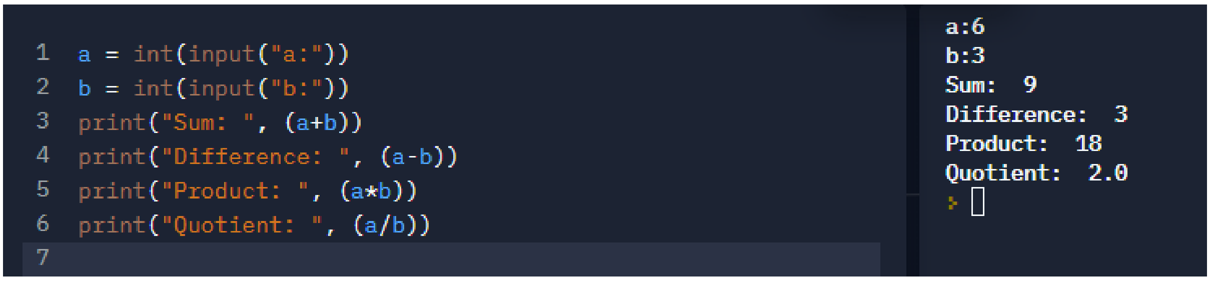
Addition, subtraction, multiplication, division and exponential operators have been used to give the actual output.

29.

```
d=3+6j
print(d.real)
>>>3.0
print(d.imag)
6.0
```

The real part of a complex number can be found by real and complex part by using imag.

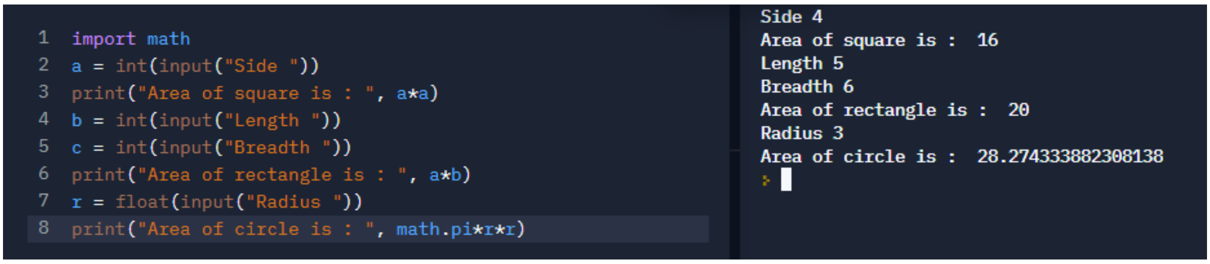
1.To perform addition, subtraction, multiplication and division operations.



```
a = int(input("a:"))
b = int(input("b:"))
print("Sum: ", (a+b))
print("Difference: ", (a-b))
print("Product: ", (a*b))
print("Quotient: ", (a/b))
```

```
a:6
b:3
Sum:  9
Difference:  3
Product:  18
Quotient:  2.0
```

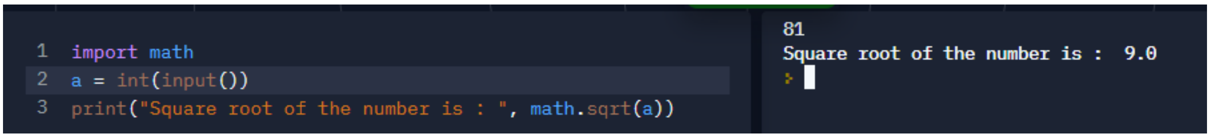
2. To find area of a square, rectangle and circle.



```
import math
a = int(input("Side "))
print("Area of square is : ", a*a)
b = int(input("Length "))
c = int(input("Breadth "))
print("Area of rectangle is : ", a*b)
r = float(input("Radius "))
print("Area of circle is : ", math.pi*r*r)
```

```
Side 4
Area of square is :  16
Length 5
Breadth 6
Area of rectangle is :  20
Radius 3
Area of circle is :  28.274333882308138
```

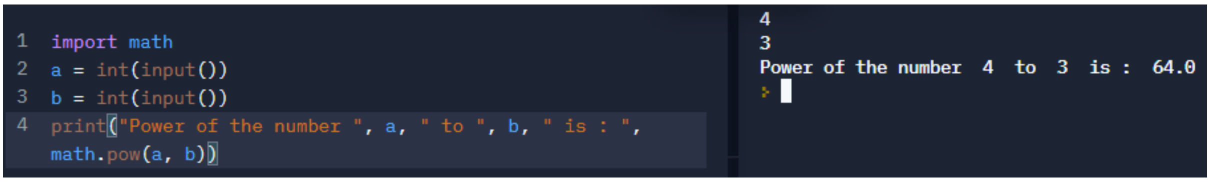
3. To find square root of a number.



```
import math
a = int(input())
print("Square root of the number is : ", math.sqrt(a))
```

```
81
Square root of the number is :  9.0
```

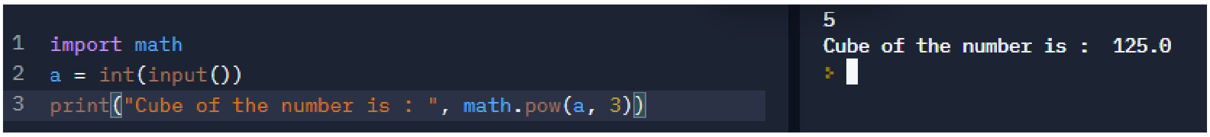
4. To find power of a number.



```
import math
a = int(input())
b = int(input())
print("Power of the number ", a, " to ", b, " is : ",
math.pow(a, b))
```

```
4
3
Power of the number  4  to  3  is :  64.0
```

5. To find cube of a number.

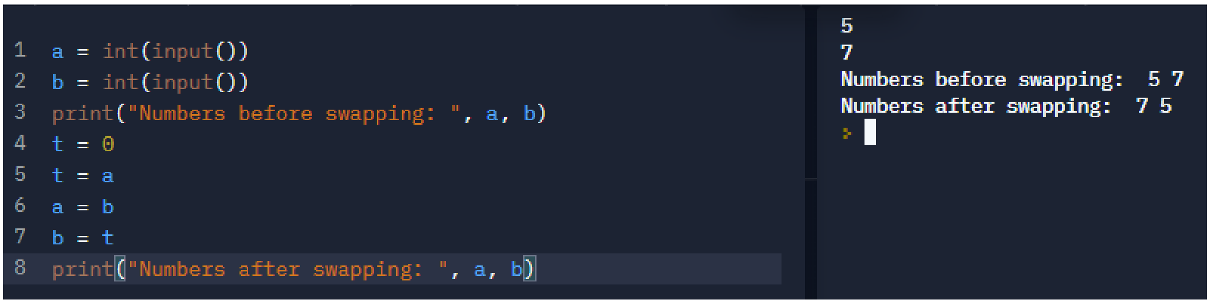


```
import math
a = int(input())
print("Cube of the number is : ", math.pow(a, 3))
```

```
5
Cube of the number is :  125.0
```

6. To swap two numbers with and without using a temporary variable.

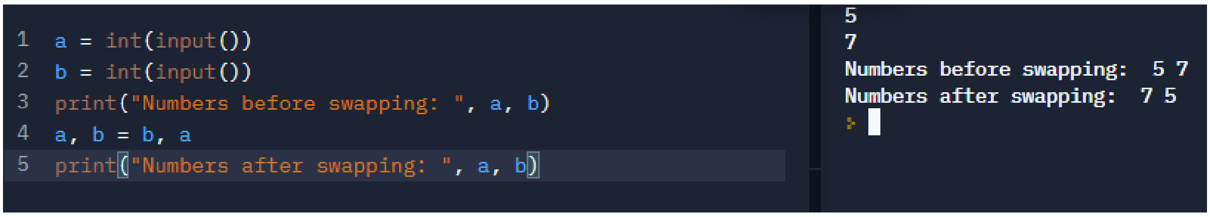
Using temporary variable:



```
a = int(input())
b = int(input())
print("Numbers before swapping: ", a, b)
t = 0
t = a
a = b
b = t
print("Numbers after swapping: ", a, b)
```

```
5
7
Numbers before swapping:  5 7
Numbers after swapping:  7 5
```

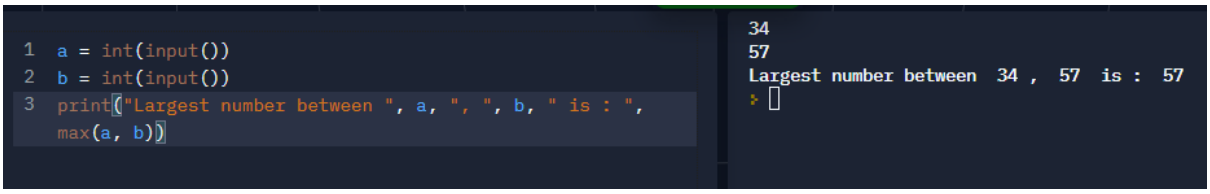
Without using temporary variable:



```
a = int(input())
b = int(input())
print("Numbers before swapping: ", a, b)
a, b = b, a
print("Numbers after swapping: ", a, b)
```

```
5
7
Numbers before swapping:  5 7
Numbers after swapping:  7 5
```

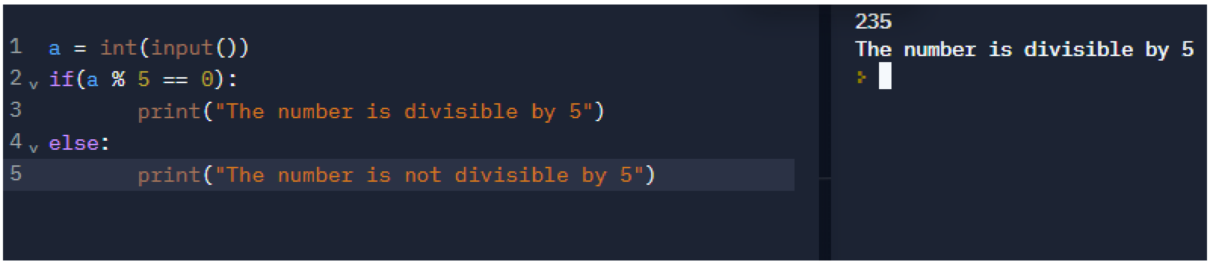
7. To find the largest of two numbers.



```
a = int(input())
b = int(input())
print("Largest number between ", a, ", ", b, " is : ", max(a, b))
```

```
34
57
Largest number between  34 ,  57  is :  57
```

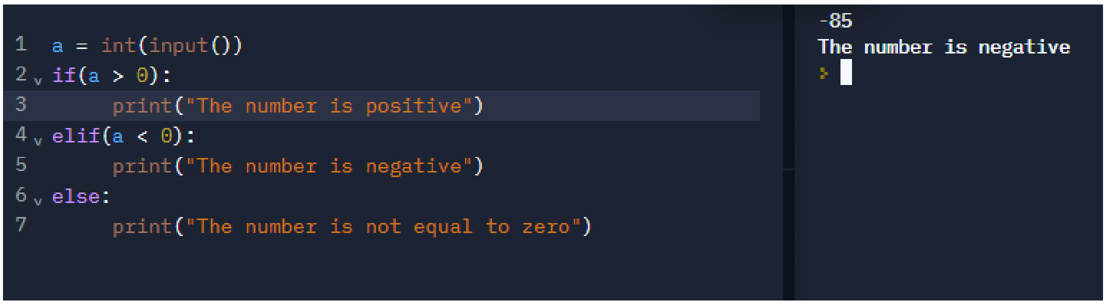
8. To check if a number is divisible by 5.



```
a = int(input())
if(a % 5 == 0):
        print("The number is divisible by 5")
else:
        print("The number is not divisible by 5")
```

```
235
The number is divisible by 5
```

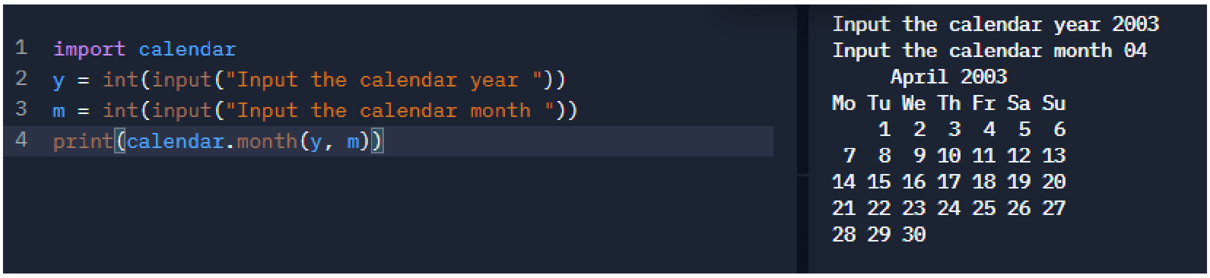
9. To check if a number is positive, negative or equal to zero.



```python
a = int(input())
if(a > 0):
        print("The number is positive")
elif(a < 0):
        print("The number is negative")
else:
        print("The number is not equal to zero")
```

```
-85
The number is negative
```

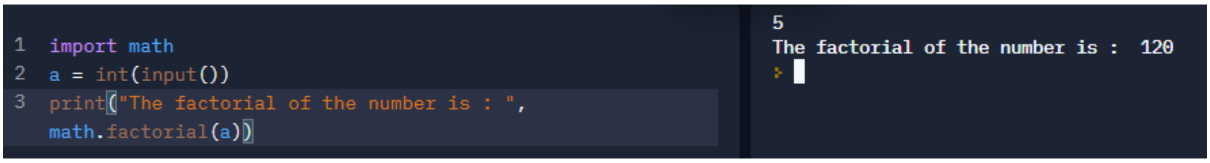
10. To print a calendar. import calendar



```python
import calendar
y = int(input("Input the calendar year "))
m = int(input("Input the calendar month "))
print(calendar.month(y, m))
```

```
Input the calendar year 2003
Input the calendar month 04
     April 2003
Mo Tu We Th Fr Sa Su
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30
```

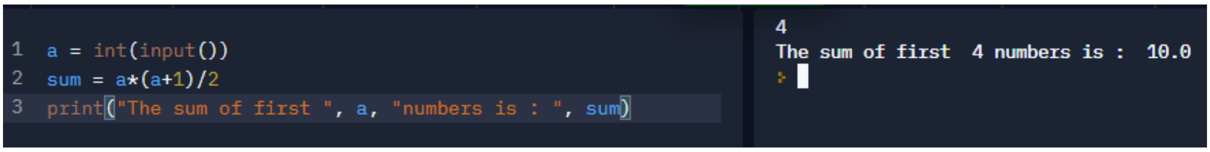
11. To find factorial of a number.



```python
import math
a = int(input())
print("The factorial of the number is : ", math.factorial(a))
```

```
5
The factorial of the number is :  120
```

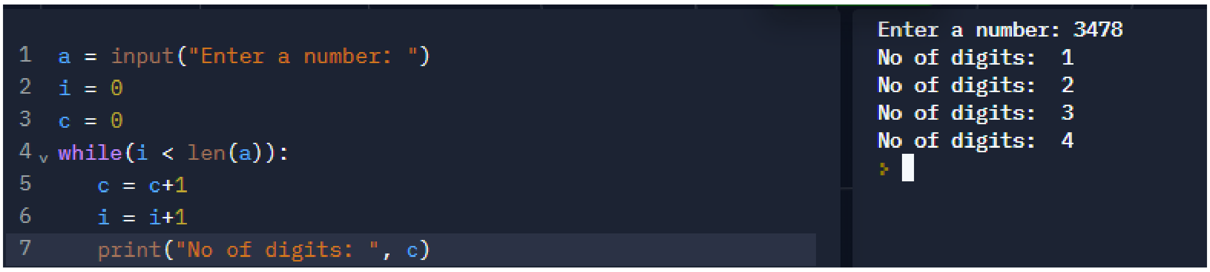
12. To find sum of first n positive integers.



```python
a = int(input())
sum = a*(a+1)/2
print("The sum of first ", a, "numbers is : ", sum)
```

```
4
The sum of first  4 numbers is :  10.0
```

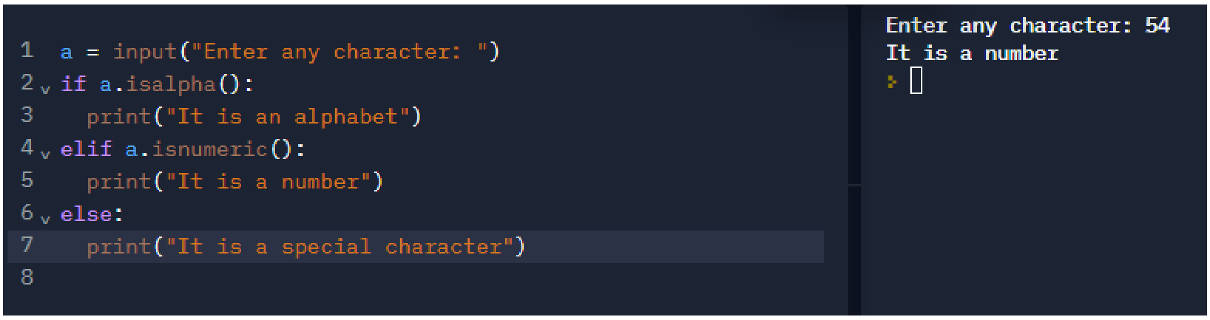
13. To count the number of digits in a number.



```python
a = input("Enter a number: ")
i = 0
c = 0
while(i < len(a)):
    c = c+1
    i = i+1
    print("No of digits: ", c)
```

```
Enter a number: 3478
No of digits:  1
No of digits:  2
No of digits:  3
No of digits:  4
```

14. To check if the input character is an alphabet, digit or a special character.

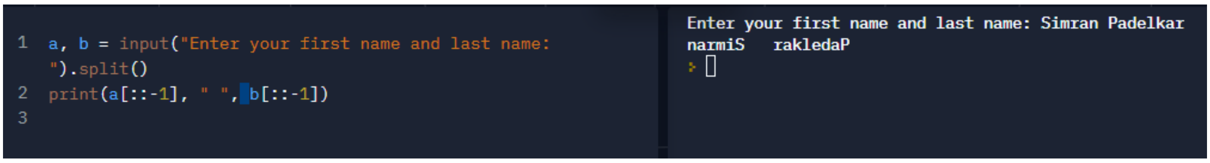


```python
a = input("Enter any character: ")
if a.isalpha():
    print("It is an alphabet")
elif a.isnumeric():
    print("It is a number")
else:
    print("It is a special character")

```

```
Enter any character: 54
It is a number
```

15. To print the input names in reverse order with a space in between.

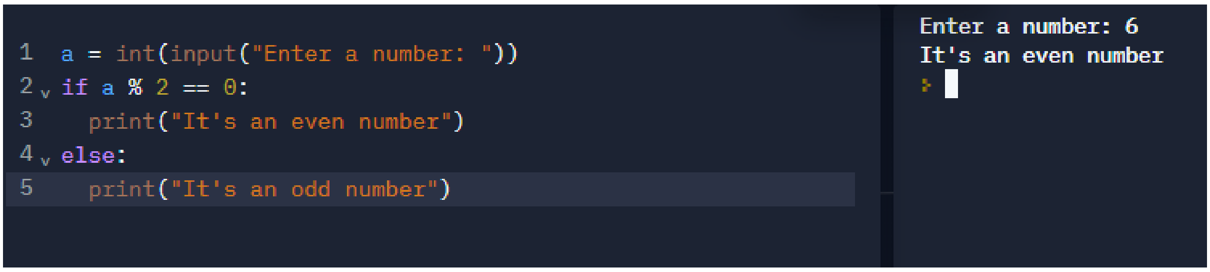


```python
a, b = input("Enter your first name and last name: ").split()
print(a[::-1], " ", b[::-1])

```

```
Enter your first name and last name: Simran Padelkar
narmiS   rakledaP
```

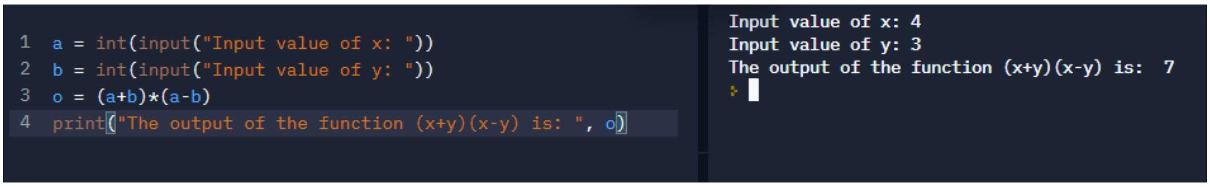
16. To check if a number is even or odd.



```python
a = int(input("Enter a number: "))
if a % 2 == 0:
    print("It's an even number")
else:
    print("It's an odd number")
```

```
Enter a number: 6
It's an even number
```

17. To solve the equation (x+1)(x-1).

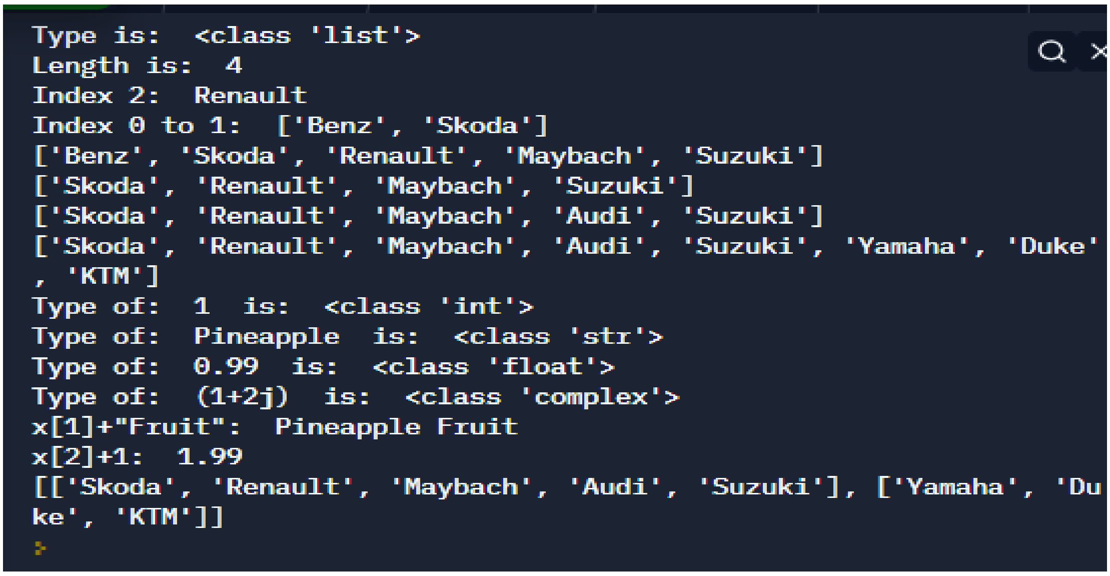


```python
a = int(input("Input value of x: "))
b = int(input("Input value of y: "))
o = (a+b)*(a-b)
print("The output of the function (x+y)(x-y) is: ", o)
```

```
Input value of x: 4
Input value of y: 3
The output of the function (x+y)(x-y) is:  7
```

LISTS:

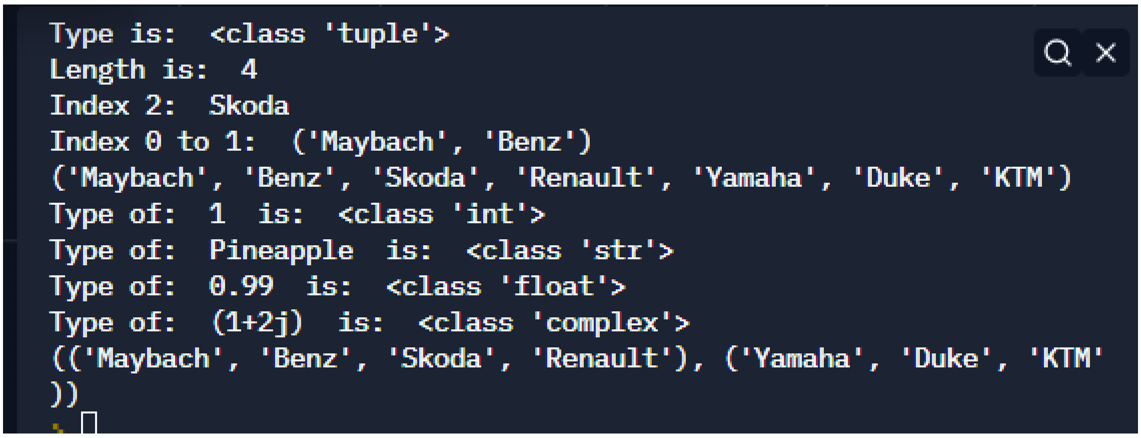
```
cars=['Benz','Skoda','Renault','Maybach']
#type
print("Type is: ",type(cars))
#length
print("Length is: ",len(cars))
#index and slicing
print("Index 2: ",cars[2])
print("Index 0 to 1: ",cars[:2])
#append
cars.append('Suzuki')
print(cars)
#remove
cars.remove('Benz')
print(cars)
#insert at an index
cars.insert(3,'Audi')
print(cars)
#combine 2 tuples
bikes=['Yamaha','Duke','KTM']
print(cars+bikes)
#multiple data types in a list
x=[1,'Pineapple',0.99,1+2j]
print("Type of: ",x[0]," is: ",type(x[0]))
print("Type of: ",x[1]," is: ",type(x[1]))
print("Type of: ",x[2]," is: ",type(x[2]))
print("Type of: ",x[3]," is: ",type(x[3]))
#change elements
x[1]+=" Fruit"
x[2]+=1
print("x[1]+\"Fruit\": ",x[1])
print("x[2]+1: ",x[2])
#nested list
nest=[cars,bikes]
print(nest)
```



```
Type is:  <class 'list'>
Length is:  4
Index 2:  Renault
Index 0 to 1:  ['Benz', 'Skoda']
['Benz', 'Skoda', 'Renault', 'Maybach', 'Suzuki']
['Skoda', 'Renault', 'Maybach', 'Suzuki']
['Skoda', 'Renault', 'Maybach', 'Audi', 'Suzuki']
['Skoda', 'Renault', 'Maybach', 'Audi', 'Suzuki', 'Yamaha', 'Duke'
, 'KTM']
Type of:  1  is:  <class 'int'>
Type of:  Pineapple  is:  <class 'str'>
Type of:  0.99  is:  <class 'float'>
Type of:  (1+2j)  is:  <class 'complex'>
x[1]+"Fruit":  Pineapple Fruit
x[2]+1:  1.99
[['Skoda', 'Renault', 'Maybach', 'Audi', 'Suzuki'], ['Yamaha', 'Du
ke', 'KTM']]
```

TUPLE:

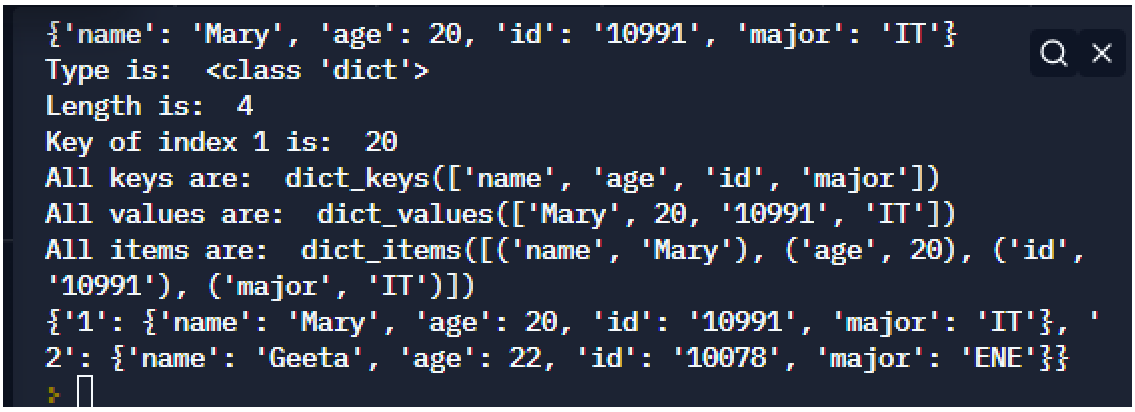
```
cars=('Maybach','Benz','Skoda','Renault')
#type
print("Type is: ",type(cars))
#length
print("Length is: ",len(cars))
#index and slicing
print("Index 2: ",cars[2])
print("Index 0 to 1: ",cars[:2])
#combine 2 tuples
bikes=('Yamaha','Duke','KTM')
print(cars+bikes)
#multiple data types in a tuple
x=(1,'Pineapple',0.99,1+2j)
print("Type of: ",x[0]," is: ",type(x[0]))
print("Type of: ",x[1]," is: ",type(x[1]))
print("Type of: ",x[2]," is: ",type(x[2]))
print("Type of: ",x[3]," is: ",type(x[3]))
#nested tuple
nest=(cars,bikes)
print(nest)
```



```
Type is:  <class 'tuple'>
Length is:  4
Index 2:  Skoda
Index 0 to 1:  ('Maybach', 'Benz')
('Maybach', 'Benz', 'Skoda', 'Renault', 'Yamaha', 'Duke', 'KTM')
Type of:  1  is:  <class 'int'>
Type of:  Pineapple  is:  <class 'str'>
Type of:  0.99  is:  <class 'float'>
Type of:  (1+2j)  is:  <class 'complex'>
(('Maybach', 'Benz', 'Skoda', 'Renault'), ('Yamaha', 'Duke', 'KTM'
))
```

DICTIONARY:

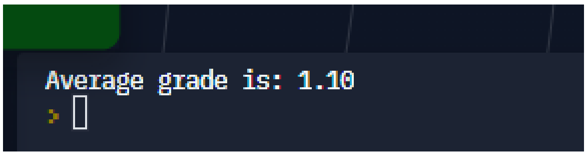
```
student={'name':'Mary','age':20,'id':'10991','major':'IT'}
print(student)
#type
print("Type is: ",type(student))
#length
print("Length is: ",len(student))
#value of a key
print("Key of index 1 is: ",student['age'])
#all keys
print("All keys are: ",student.keys())
#all values
print("All values are: ",student.values())
#all items
print("All items are: ",student.items())
#value can be another dictionary
student1={'name':'Geeta','age':22,'id':'10078','major':'ENE'}
students={'1':student,'2':student1}
print(students)
```



```
{'name': 'Mary', 'age': 20, 'id': '10991', 'major': 'IT'}
Type is:  <class 'dict'>
Length is:  4
Key of index 1 is:  20
All keys are:  dict_keys(['name', 'age', 'id', 'major'])
All values are:  dict_values(['Mary', 20, '10991', 'IT'])
All items are:  dict_items([('name', 'Mary'), ('age', 20), ('id',
'10991'), ('major', 'IT')])
{'1': {'name': 'Mary', 'age': 20, 'id': '10991', 'major': 'IT'}, '
2': {'name': 'Geeta', 'age': 22, 'id': '10078', 'major': 'ENE'}}
```

1.Average grade of a student

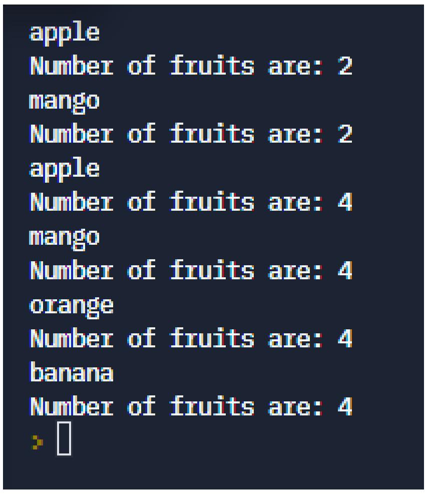
```
students= {'1': {"name":" Bob", "grade": 5.5},
'2': {"name":" Mary", "grade": 6.5},
'3': {"name":" David", "grade": 4.2},
'4': {"name":" John", "grade": 4.8},
'5': {"name":" Alex", "grade": 3.8}}
def AvgGrades(students):
  sum=0
  for i in students:
    sum=sum+students[i]['grade']
    avg=sum/len(students)
    return avg
a=AvgGrades(students)
print("Average grade is: %0.2f"%(a))
```



```
Average grade is: 1.10
>
```

2. Function using default arguments

```
def fruits(fruit=['apple','mango']):
  for i in fruit:
    print(i)
     print("Number of fruits are: %d" %(len(fruit)))
fruits()
fruit=['apple','mango','orange','banana']
fruits(fruit)
```

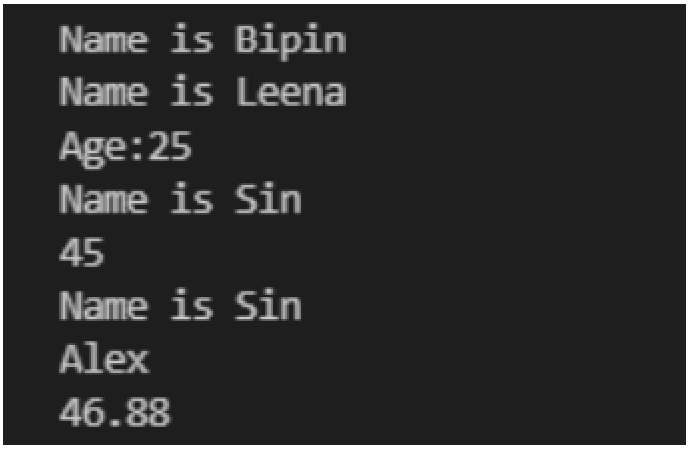


```
apple
Number of fruits are: 2
mango
Number of fruits are: 2
apple
Number of fruits are: 4
mango
Number of fruits are: 4
orange
Number of fruits are: 4
banana
Number of fruits are: 4
>
```

3. Function using variable length arguments

```
def Stu(name,*v):
 print("Name is "+name)
for i in v:
  print(i)
```

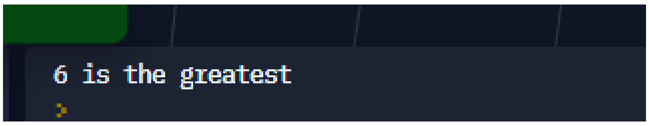
```
Stu("Bipin")
Stu("Leena",'Age:25')
Stu("Sin",45)
Stu("Sin","Alex",46.88)
```



```
Name is Bipin
Name is Leena
Age:25
Name is Sin
45
Name is Sin
Alex
46.88
```

4. Maximum of three numbers

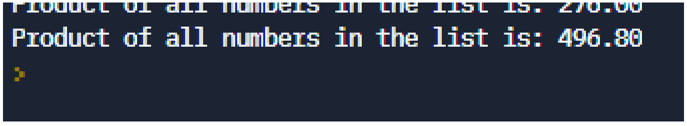
```
def max(a,b,c):
  if a>b:
    if a>c:
      print("%d is the greatest" %(a))
    else:
      print("%d is the greatest" %(c))
  else:
      print("%d is the greatest" %(b))
max(3,6,-2)
```



```
6 is the greatest
>
```

5. Product of all numbers in a list.

```
def pro(li):
  p=1
  for i in li:
    p=p*i
    print("Product of all numbers in the list is: %0.2f" %(p))
li=[1,2,3,5,9.2,1.8]
pro(li)
```



```
Product of all numbers in the list is: 276.00
Product of all numbers in the list is: 496.80
>
```

To print the light detected using LDR (0 toward dark and 1 toward light)

```
from gpiozero import LightSensor

ldr=LightSensor(17)

while True:

    ldr.wait_for_light()

    print("It's light")

    print(ldr.value)

    ldr.wait_for_dark()

    print("It's dark")

    print(ldr.value)
```

OUTPUT:

```
0.9762574999977005
0.9762574999977005
0.9762574999977005
0.9762574999977005
0.9762574999977005
0.9762574999977005
0.9762574999977005
0.9762574999977005
0.9762574999977005
0.9762574999977005
0.9762574999977005
0.9762574999977005
0.9762574999977005
0.9762574999977005
0.9762574999977005
0.9754057000018292
0.9754057000018292
0.9754057000018292
0.9754057000018292
0.9754057000018292
0.9754057000018292
```

```
0.9754057000018292

0.9754057000018292

0.9754057000018292

0.9754057000018292

0.9754057000018292

0.9754057000018292

0.9754057000018292

0.9754057000018292

0.9754057000018292

0.9747501999991073

0.9495484000021861

0.9495484000021861

0.9495484000021861

0.9495484000021861

0.9495484000021861

0.9495484000021861

0.9495484000021861

0.9495484000021861

0.9495484000021861

0.9495484000021861

0.8915893999992477

0.8915893999992477

0.8915893999992477

0.8915893999992477

0.8915893999992477

0.8915893999992477

0.8915893999992477

0.8915893999992477

0.8915893999992477

0.8915893999992477

0.8915893999992477

0.8235045000001264
```

```
0.8235045000001264
0.8235045000001264
0.8235045000001264
0.8235045000001264
0.7685065999917242
0.7685065999917242
0.7685065999917242
0.7685065999917242
0.7685065999917242
0.7685065999917242
0.6776569999951789
0.6776569999951789
0.6776569999951789
0.6776569999951789
0.6776569999951789
0.6776569999951789
0.5570391000001109
0.5570391000001109
0.5570391000001109
0.5570391000001109
0.5570391000001109
0.5570391000001109
0.5570391000001109
0.5570391000001109
0.5570391000001109
0.46430620000228373
0.46430620000228373
0.46430620000228373
0.46430620000228373
0.46430620000228373
0.46430620000228373
0.46430620000228373
```

```
0.46430620000228373

0.46430620000228373

0.46430620000228373

0.46430620000228373

0.4167248999929143

0.4167248999929143

0.4167248999929143

0.4167248999929143

0.4167248999929143

0.4167248999929143

0.4167248999929143

0.4167248999929143

0.4167248999929143

0.4167248999929143

0.4167248999929143

0.4167248999929143

0.32785299999977724

0.32785299999977724

0.32785299999977724

0.32785299999977724

0.32785299999977724

0.32785299999977724

0.32785299999977724

0.32785299999977724

0.32785299999977724

0.32785299999977724

0.32785299999977724

0.32785299999977724

0.2648384999938571

0.2648384999938571

0.2648384999938571

0.2648384999938571
```

```
0.2648384999938571
0.2648384999938571
0.2648384999938571
0.2648384999938571
0.2648384999938571
0.24195160000408578
0.24195160000408578
0.24195160000408578
0.24195160000408578
0.24195160000408578
0.24195160000408578
0.24195160000408578
0.24195160000408578
0.24195160000408578
0.24195160000408578
0.24195160000408578
0.19742399999904592
0.19742399999904592
0.19742399999904592
0.19742399999904592
0.19742399999904592
0.19742399999904592
0.19742399999904592
0.19742399999904592
0.19742399999904592
0.19742399999904592
0.19742399999904592
0.19742399999904592
0.16040009999278482
0.16040009999278482
0.16040009999278482
0.16040009999278482
```

```
0.16040009999278482

0.16040009999278482

0.16040009999278482

0.16040009999278482

0.16040009999278482

0.16040009999278482

0.16040009999278482

0.16040009999278482

0.16040009999278482

0.16040009999278482

0.16040009999278482

0.16040009999278482

0.16040009999278482

0.16040009999278482

0.16040009999278482

0.16040009999278482

0.16040009999278482

0.16040009999278482

0.16040009999278482

0.16040009999278482

0.1657686000053218

0.1657686000053218

0.1657686000053218

0.1657686000053218

0.1657686000053218

0.1657686000053218

0.1657686000053218

0.1657686000053218

0.1657686000053218

0.1657686000053218

0.1657686000053218

0.16040009999278482
```

```
0.16040009999278482
0.16040009999278482
0.16040009999278482
0.16040009999278482
0.16040009999278482
0.16040009999278482
0.16040009999278482
0.16040009999278482
0.16040009999278482
```

To print if it’s dark or light using LDR

```
from gpiozero import LightSensor
ldr=LightSensor(17)
while True:
    print(ldr.value)
    if(ldr.value<0.1):
        break
```

OUTPUT:

```
It's light
0.9728799999957118
It's dark
0.060667399999147165
It's light
0.10856300000432384
It's dark
0.039665600002990686
It's light
0.8164633999977013
```