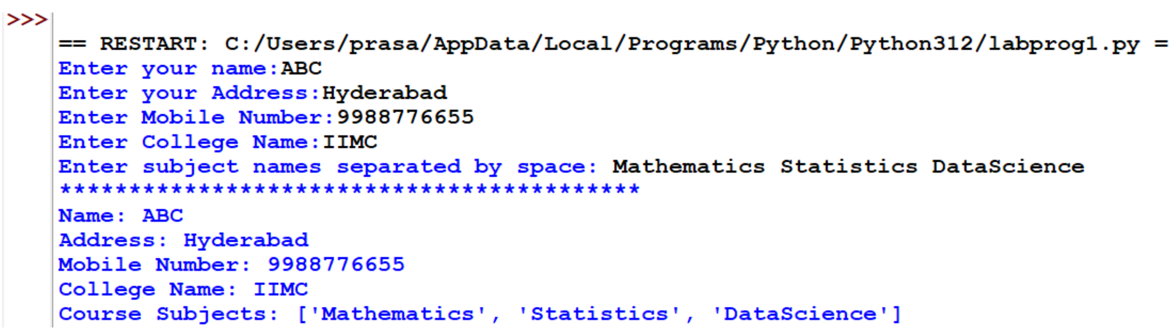
B.Sc. (Data Science) I Sem.
Problem Solving and Python Programming (Lab)

1. Program to display the information: Your name, Full Address, Mobile Number, College Name, Course Subjects

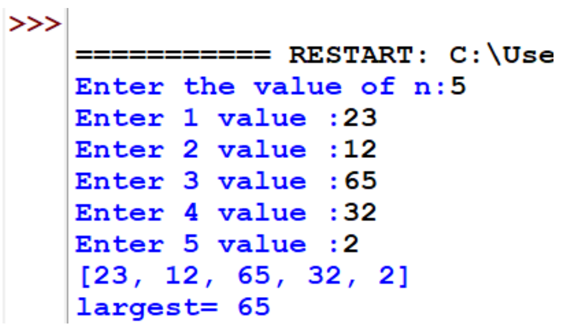
```
Name=input('Enter your name:')
Address=input('Enter your Address:')
Mobile=int(input('Enter Mobile Number:'))
College_Name=input('Enter College Name:')
Course_subjects=input("Enter subject names separated by space: ").split()
print('******************************************')
print('Name:',Name)
print('Address:',Address)
print('Mobile Number:',Mobile)
print('College Name:',College_Name)
print('Course Subjects:',Course_subjects)
```



```
>>>
== RESTART: C:/Users/prasa/AppData/Local/Programs/Python/Python312/labprog1.py =
Enter your name:ABC
Enter your Address:Hyderabad
Enter Mobile Number:9988776655
Enter College Name:IIMC
Enter subject names separated by space: Mathematics Statistics DataScience
******************************************
Name: ABC
Address: Hyderabad
Mobile Number: 9988776655
College Name: IIMC
Course Subjects: ['Mathematics', 'Statistics', 'DataScience']
```

2. Program to find the largest number among ‘n’ given numbers.

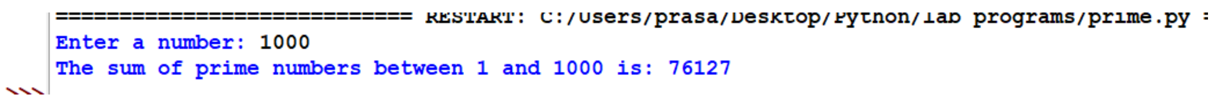
```
n=int(input("Enter the value of n:"))
numbers=[]
for i in range(n):
    val=int(input(f"Enter {i+1} value :"))
    numbers.append(val)
print(numbers)
big=numbers[0]
for i in numbers[1:]:
    if i>=big:
        big=i
print("largest=",big)
```



```
>>>
=========== RESTART: C:\Use
Enter the value of n:5
Enter 1 value :23
Enter 2 value :12
Enter 3 value :65
Enter 4 value :32
Enter 5 value :2
[23, 12, 65, 32, 2]
largest= 65
```

3. **Program to find the sum of all prime numbers between 1 and 1000.**

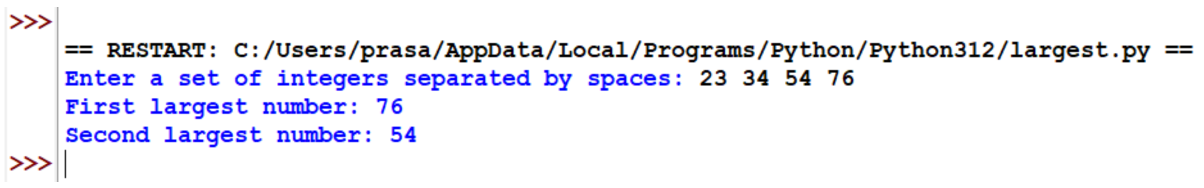
```
def is_prime(num):
    if num <= 1:
        return False
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            return False
    return True

def sum_of_primes(n):
    prime_sum = 0
    for num in range(2, n + 1):
        if is_prime(num):
            prime_sum += num
    return prime_sum

n = int(input("Enter a number: "))

result = sum_of_primes(n)
print(f"The sum of prime numbers between 1 and {n} is: {result}")
```



```
============================ RESTART: C:/Users/prasa/Desktop/Python/lab programs/prime.py =
Enter a number: 1000
The sum of prime numbers between 1 and 1000 is: 76127
```

4. **Program that reads set of integers and display 1st & 2nd largest numbers.**

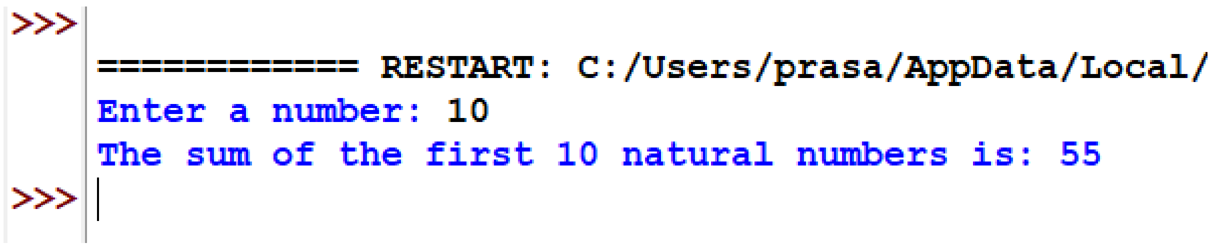
```
input_numbers = input("Enter a set of integers separated by spaces: ")
numbers = input_numbers.split()
first_largest = second_largest = None

for num_str in numbers:
    num = int(num_str)
    if first_largest is None or num > first_largest:
        second_largest = first_largest
        first_largest = num
    elif second_largest is None or (num > second_largest and num != first_largest):
        second_largest = num

if first_largest is not None and second_largest is not None:
    print("First largest number:", first_largest)
    print("Second largest number:", second_largest)
```



```
>>>
    == RESTART: C:/Users/prasa/AppData/Local/Programs/Python/Python312/largest.py ==
    Enter a set of integers separated by spaces: 23 34 54 76
    First largest number: 76
    Second largest number: 54
>>>
```

5. **Program to print the sum of first n natural numbers.**

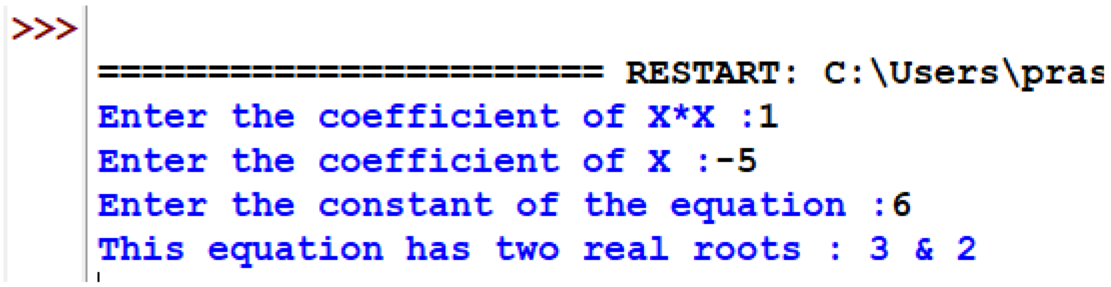
```
n = int(input("Enter a number: "))
sum = 0

for i in range(1, n + 1):
    sum+= i

print("The sum of the first", n, "natural numbers is:", sum)
```



```
>>>
    ============ RESTART: C:/Users/prasa/AppData/Local/
    Enter a number: 10
    The sum of the first 10 natural numbers is: 55
>>> |
```

6. **Program to find the roots of a quadratic equation.**

```
import math
a=int(input("Enter the coefficient of X*X :"))
b=int(input("Enter the coefficient of X :" ))
c=int(input("Enter the constant of the equation :"))
d=b*b-4*a*c
if d<0:
    print("This equation has no real roots :")
elif d==0:
    x=(-b)/2*a
    print("The equation has unique real root :",x)
else:
    dd=math.sqrt(d)
    x1=int((-b+dd)/2*a)
    x2=int((-b-dd)/2*a)
    print("This equation has two real roots :" ,x1,"&", x2)
```

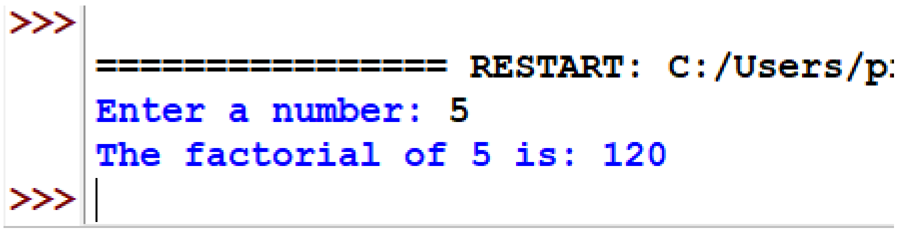


```
>>>
    ====================== RESTART: C:\Users\pras
    Enter the coefficient of X*X :1
    Enter the coefficient of X :-5
    Enter the constant of the equation :6
    This equation has two real roots : 3 & 2
```

7. **Python Program for both recursive and non-recursive functions to find the factorial of Positive integer.**

Non-Recursive:

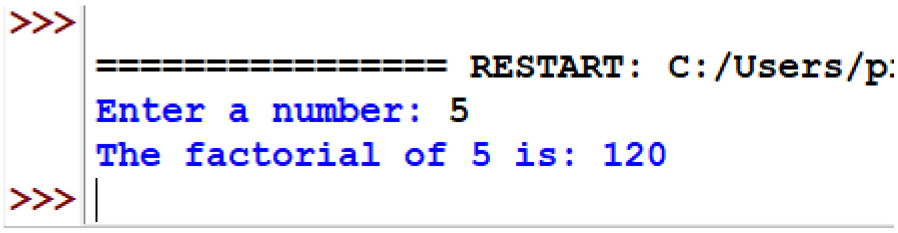
```
def ft(n):
    if n < 0:
        return

    result = 1
    for i in range(2, n + 1):
        result *= i

    return result


n = int(input("Enter a number: "))
fact = ft(n)
print(f"The factorial of {n} is: {fact}")
```



```
>>>
    ================ RESTART: C:/Users/p
    Enter a number: 5
    The factorial of 5 is: 120
>>>
```

Recursive:

```
def ft(n):
    if n < 0:
        return
    elif n == 0 or n == 1:
        return 1
    else:
        return n * ft(n - 1)

n = int(input("Enter a number: "))
fact = ft(n)
print(f"The factorial of {n} is: {fact}")
```

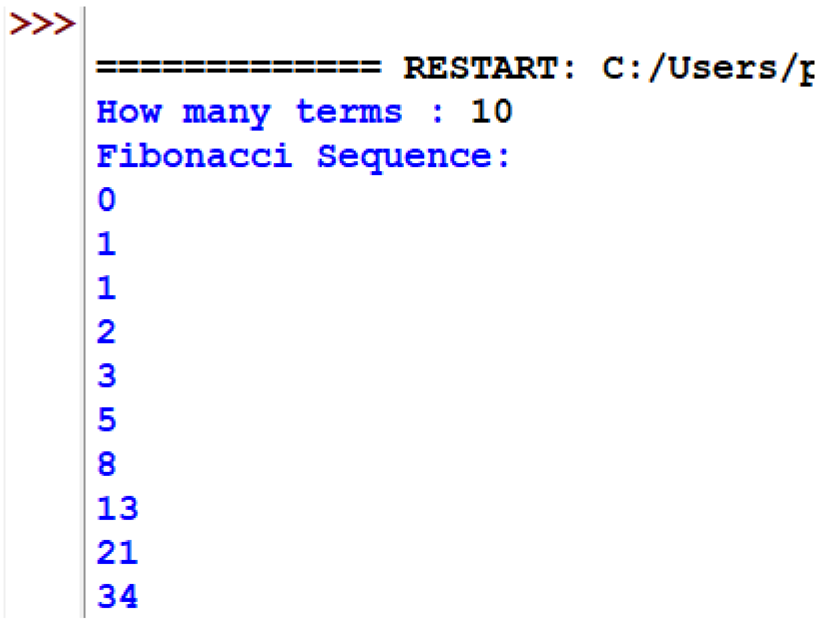


```
>>>
    ================ RESTART: C:/Users/p
    Enter a number: 5
    The factorial of 5 is: 120
>>>
```

8. **Python Program for both recursive and non-recursive functions to Print Fibonacci Sequence up to given number ‘n’.**

Non-Recursive:

```
n=int(input("How many terms : "))
n1,n2=0,1
count=0
if n <=0:
   print("Please enter a positive integer")
elif n==1:
   print("Fibonacci sequecnce upto",n," : ")
   print(n1)
else:
   print("Fibonacci Sequence:")
while count<n:
   print(n1)
   nth=n1+n2
   n1=n2
   n2=nth
   count+=1
```

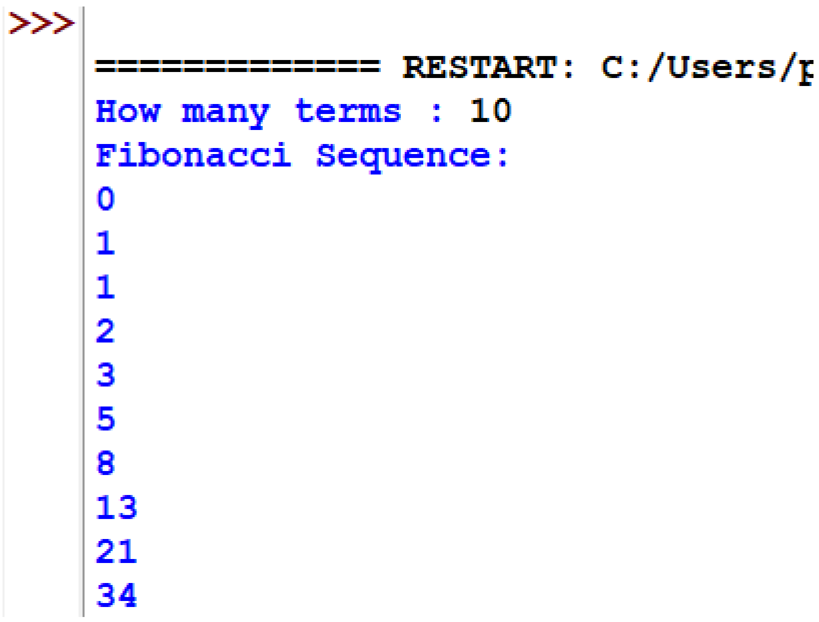


```
>>>
 ============= RESTART: C:/Users/p
 How many terms : 10
 Fibonacci Sequence:
 0
 1
 1
 2
 3
 5
 8
 13
 21
 34
```

Recursive:

```
def fibonacci(n):
   if n <= 1:
      return n
   else:
      return fibonacci(n - 1) + fibonacci(n - 2)

n = int(input("How many terms? "))
```

```
if n <= 0:
    print("Please enter a positive integer")
else:
    print("Fibonacci Sequence:")
    for i in range(n):
        print(fibonacci(i))
```

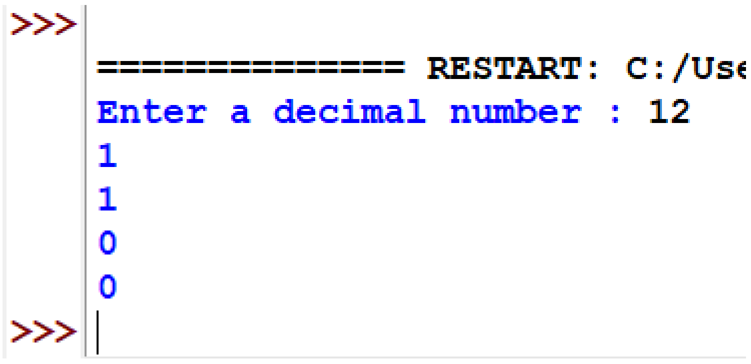


```
>>>
============= RESTART: C:/Users/p
How many terms : 10
Fibonacci Sequence:
0
1
1
2
3
5
8
13
21
34
```

9. **Python Program for both recursive and non-recursive functions to Convert decimal number to its binary equivalent.**

Non-Recursive:

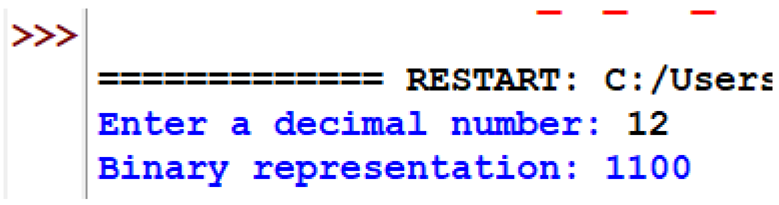
```
def dec_to_bin(n):
    if n>=1:
        dec_to_bin(n//2)
        print(n%2)
num=int(input("Enter a decimal number : "))
dec_to_bin(num)
```



```
>>>
============== RESTART: C:/Use
Enter a decimal number : 12
1
1
0
0
>>>
```

Recursive:

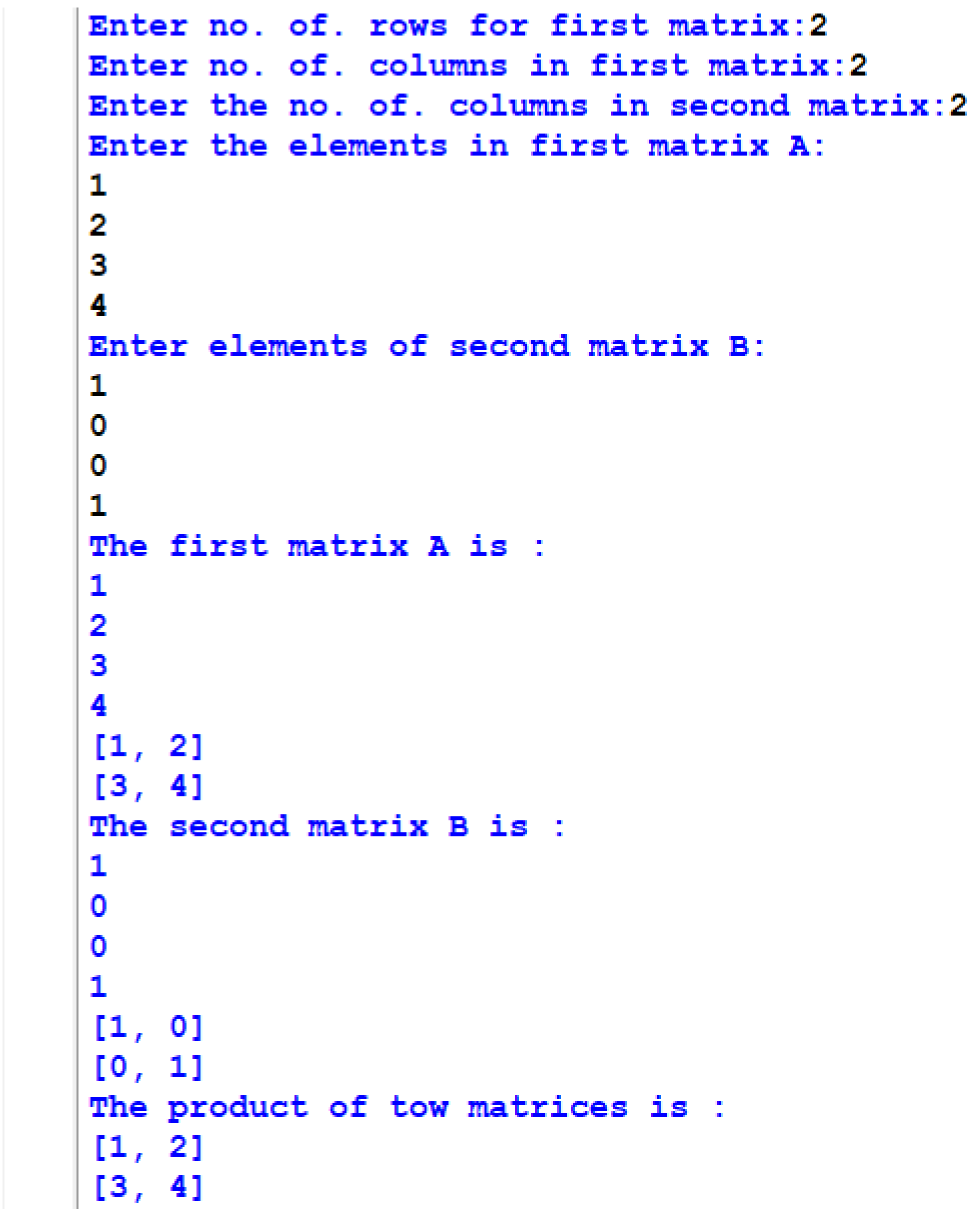
```
def dec_to_bin(n):
    if n == 0:
        return "0"
    elif n == 1:
        return "1"
    else:
        return dec_to_bin(n // 2) + str(n % 2)

num = int(input("Enter a decimal number: "))

if num < 0:
    print("Please enter a non-negative integer.")
else:
    print("Binary representation:", dec_to_bin(num))
```



```
>>>
    ============= RESTART: C:/Users
    Enter a decimal number: 12
    Binary representation: 1100
```

10. Program to find the product of two matrices.

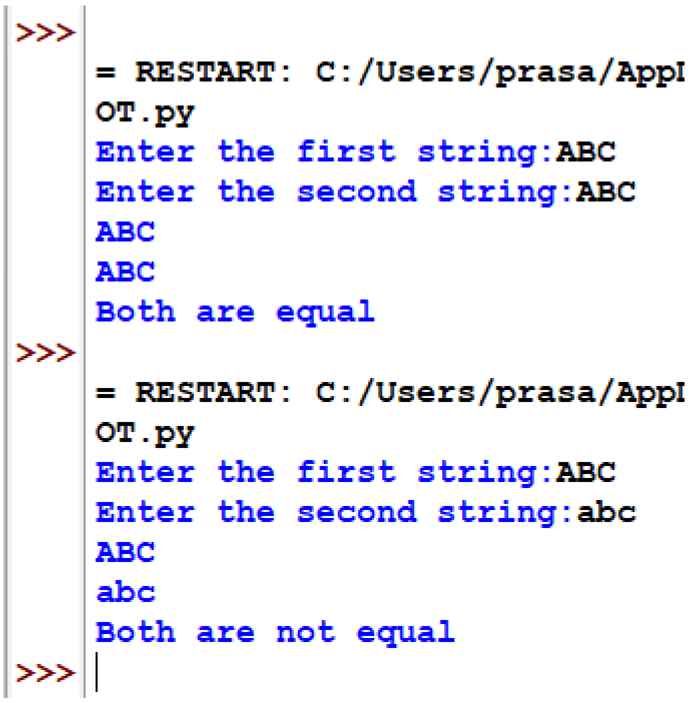
```
m=int(input("Enter no. of. rows for first matrix:"))
r=int(input("Enter no. of. columns in first matrix:"))
p=int(input("Enter the no. of. columns in second matrix:"))
print("Enter the elements in first matrix A:")
a=[[int(input())for j in range(p)]for i in range(m)]
print('Enter elements of second matrix B:')
b=[[int(input())for j in range(r)]for i in range(p)]
c=[[0 for j in range(r)]for i in range(m)]
print("The first matrix A is : ")
for i in range (m):
    for j in range(p):
        print(a[i][j])
for i in a:
    print(i)
print("The second matrix B is : ")
for i in range (p):
    for j in range(r):
        print(b[i][j])
for i in b:
    print(i)
print("The product of tow matrices is : ")
for i in range(m):
    for j in range(r):
        for k in range(len(c)):
            c[i][j] = c[i][j]+a[i][k]*b[k][j]
        # print(c[i][j])
for i in c:
    print(i)
```



```
Enter no. of. rows for first matrix:2
Enter no. of. columns in first matrix:2
Enter the no. of. columns in second matrix:2
Enter the elements in first matrix A:
1
2
3
4
Enter elements of second matrix B:
1
0
0
1
The first matrix A is :
1
2
3
4
[1, 2]
[3, 4]
The second matrix B is :
1
0
0
1
[1, 0]
[0, 1]
The product of tow matrices is :
[1, 2]
[3, 4]
```

11. Program that accepts two strings S1, S2 and finds whether they are equal are not

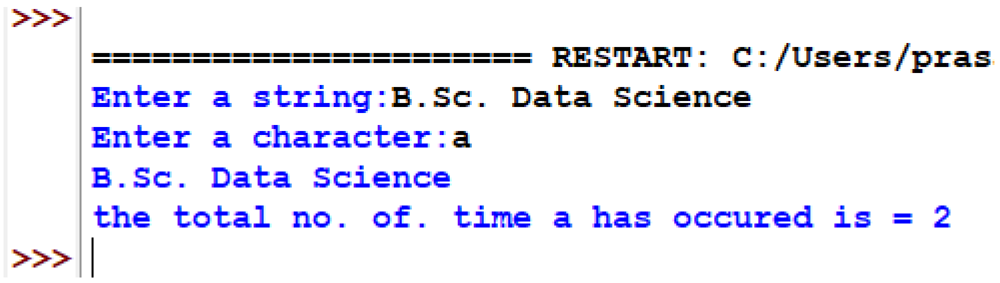
```
s1=input("Enter the first string:")
s2=input("Enter the second string:")
print(s1)
print(s2)
if s1==s
2:
    print("Both are equal")
else:
    print("Both are not equal")
```



```
>>>
    = RESTART: C:/Users/prasa/AppI
    OT.py
    Enter the first string:ABC
    Enter the second string:ABC
    ABC
    ABC
    Both are equal
>>>
    = RESTART: C:/Users/prasa/AppI
    OT.py
    Enter the first string:ABC
    Enter the second string:abc
    ABC
    abc
    Both are not equal
>>>
```

12. Program to count the number of occurrences of characters in a given string

```
string=input("Enter a string:")
character=input("Enter a character:")
print(string)
count=0
for  i in range(len(string)):
   if (string[i]==character):
      count=count+1
print(f'the total no. of. time {character} has occured is = {count}')
```

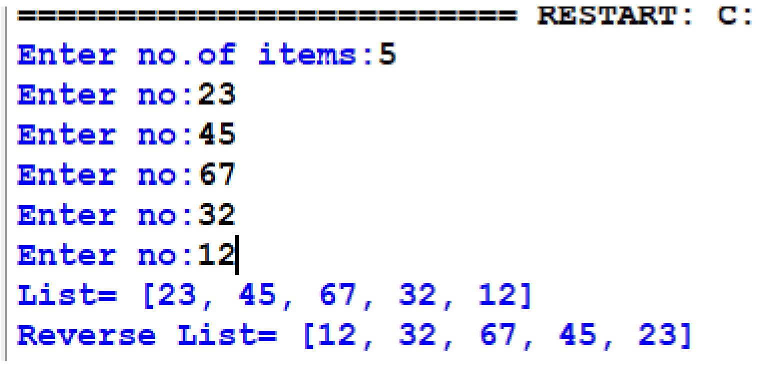


```
>>>
    ====================== RESTART: C:/Users/pras
    Enter a string:B.Sc. Data Science
    Enter a character:a
    B.Sc. Data Science
    the total no. of. time a has occured is = 2
>>>
```

13. Program to read the lists of numbers as l1, print the lists in reverse order without using reverse function.

```
list=[]
n=int(input('Enter no.of items:'))
i=1
while i<=n:
   n1=int(input('Enter no:'))
   list.append(n1)
   i+=1
print('List=',list)
len=len(list)-1
list1=[]
```

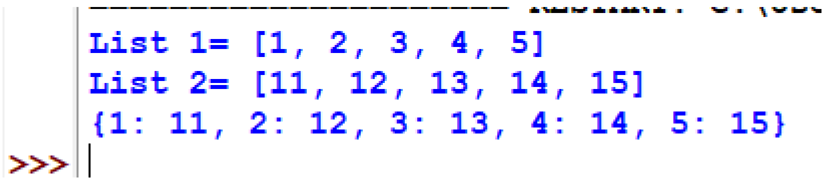
```
while len>=0:
    list1.append(list[len])
   len=len-1
print('Reverse List=',list1)
```



```
========================== RESTART: C:
Enter no.of items:5
Enter no:23
Enter no:45
Enter no:67
Enter no:32
Enter no:12
List= [23, 45, 67, 32, 12]
Reverse List= [12, 32, 67, 45, 23]
```

14. Program that combines lists L1 and L2 into a dictionary.

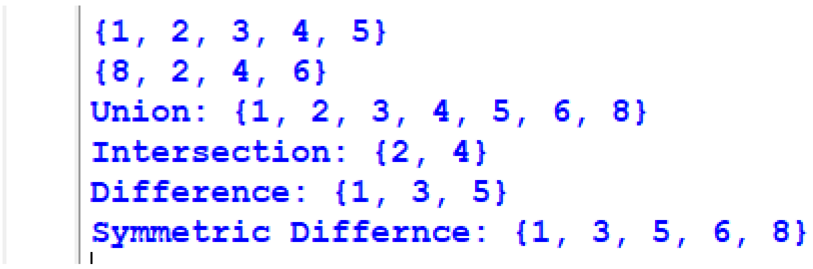
```
D=dict()
L1=[1,2,3,4,5]
L2=[11,12,13,14,15]
for i in range(len(L1)):
   D.update({L1[i]:L2[i]})
print("List 1=", L1)
print("List 2=", L2)
print(f'{D}')
```



```
List 1= [1, 2, 3, 4, 5]
List 2= [11, 12, 13, 14, 15]
{1: 11, 2: 12, 3: 13, 4: 14, 5: 15}
>>>
```

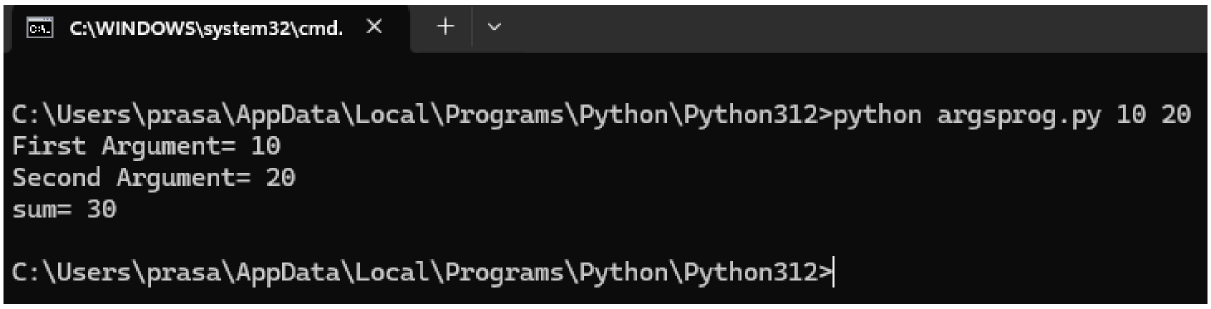
15. Program to find the Union, Intersection, Difference, Symmetric difference of any two sets.

```
A={1,2,3,4,5}
B={2,4,6,8}
print(A)
print(B)
print('Union:',A|B)
print('Intersection:',A&B)
print('Difference:',A-B)
print('Symmetric Differnce:',A^B)
```


```
{1, 2, 3, 4, 5}
{8, 2, 4, 6}
Union: {1, 2, 3, 4, 5, 6, 8}
Intersection: {2, 4}
Difference: {1, 3, 5}
Symmetric Differnce: {1, 3, 5, 6, 8}
```

16. Python Program that takes two integers as command line arguments and prints the sum of two integers.

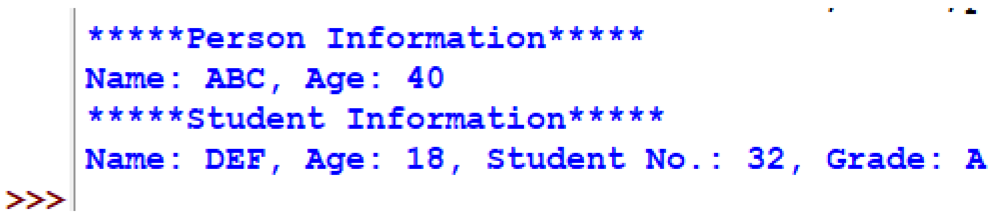
```
import sys

a = int(sys.argv[1])
b = int(sys.argv[2])

# Calculate the sum
c = a + b

print('First Argument=', a)
print('Second Argument=',b)
print('sum=',c)

# SAVE THE PROGRAM WITH THE NAME argsprog.py
# RUN COMMAND PROMPT(Uset Win+R and type cmd)
```



```
C:\Users\prasa\AppData\Local\Programs\Python\Python312>python argsprog.py 10 20
First Argument= 10
Second Argument= 20
sum= 30

C:\Users\prasa\AppData\Local\Programs\Python\Python312>
```

17. Program to implement the inheritance.

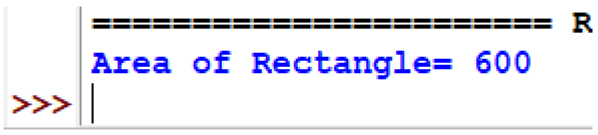
```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def display_info(self):
        return f"Name: {self.name}, Age: {self.age}"


class Student(Person):
    def __init__(self, name, age, sno, grade):
        super().__init__(name, age)
        self.sno = sno
        self.grade = grade

    def display_info(self):
        base_info = super().display_info()
        return f"{base_info}, Student No.: {self.sno}, Grade: {self.grade}"


if __name__ == "__main__":
    p = Person("ABC", 40)
    print("*****Person Information*****")
    print(p.display_info())

    s = Student("DEF", 18, 32, "A")
    print("*****Student Information*****")
    print(s.display_info())
```



```
*****Person Information*****
Name: ABC, Age: 40
*****Student Information*****
Name: DEF, Age: 18, Student No.: 32, Grade: A
>>>
```

18. Program to implement the polymorphism.

```
class Rectangle:
    def __init__(self, length, width):
        self.length = length
        self.width = width

    def calculate_area(self):
        return self.length * self.width

r = Rectangle(20,30)
print("Area of Rectangle=",r.calculate_area())
```



```
======================== R
Area of Rectangle= 600
>>>
```