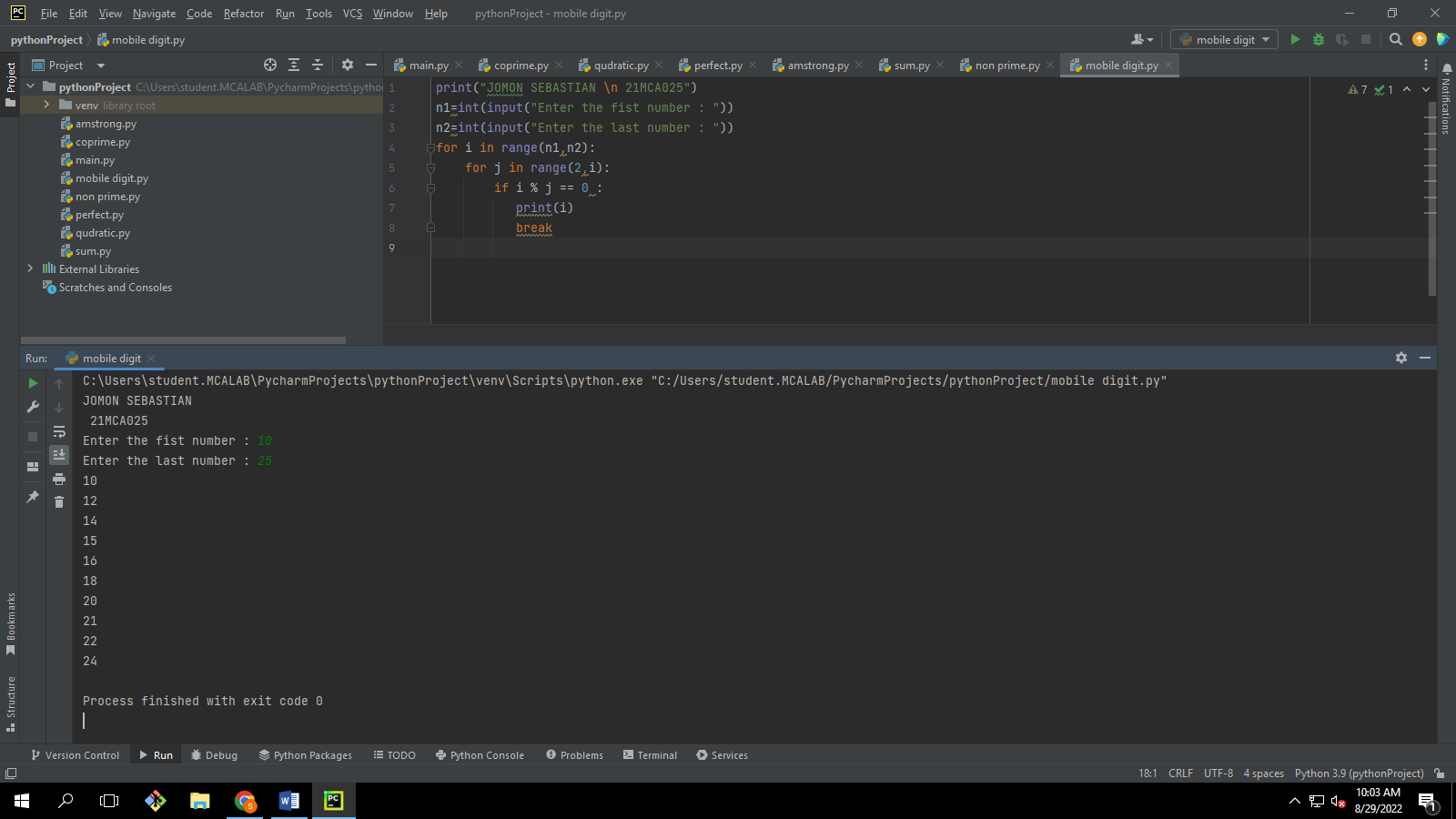
1. Program to Print all non-Prime Numbers in an Interval

print(“JOMON SEBASTIAN\n21MCA025”)  
n1=int(input("Enter the fist number : "))  
n2=int(input("Enter the last number : "))  
for i in range(n1,n2):  
 for j in range(2,i):  
 if i % j == 0 :  
 print(i)  
 break

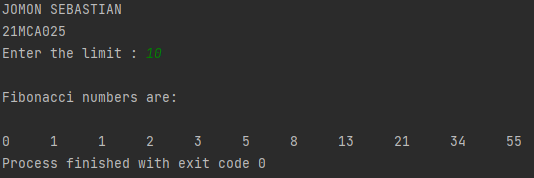
**OUTPUT**

****

1. Program to print the first N Fibonacci numbers.

print(“JOMON SEBASTIAN\n21MCA025”)  
n=int(input("Enter the limit : "))  
a=0  
b=1  
print("\nFibonacci numbers are: \n")  
print(a," ",end="")  
print(b," ",end="")  
  
for i in range(1,n):  
 c=a+b  
 a=b  
 b=c  
 print(c," ",end="")

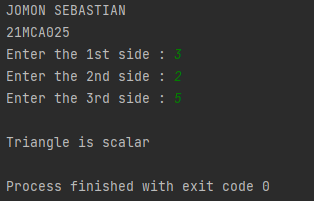
**OUTPUT**

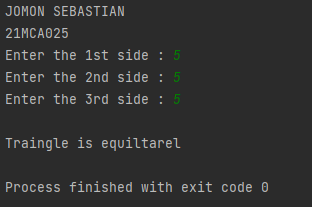
****

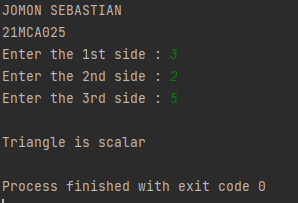
1. Given sides of a triangle, write a program to check whether given triangle is an isosceles, equilateral or scalene.

print(“JOMON SEBASTIAN\n21MCA025”)  
n1=int(input("Enter the 1st side : "))  
n2=int(input("Enter the 2nd side : "))  
n3=int(input("Enter the 3rd side : "))  
if n1==n2==n3:  
 print("\nTraingle is equiltarel")  
elif n1==n2 or n2==n3 or n1==n3:  
 print("\nTriangle is isosiles")  
else:  
 print("\nTriangle is scalar")

**OUTPUT**

****

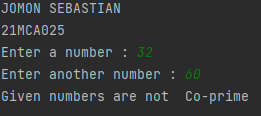
****

****

1. Program to check whether given pair of number is coprime

import math  
print(“JOMON SEBASTIAN\n21MCA025”)  
a=int(input("Enter a number : "))  
b=int(input("Enter another number : "))  
if math.gcd(a,b) == 1:  
 print("Given numbers are Co-prime")  
else:  
 print("Given numbers are not Co-prime ")

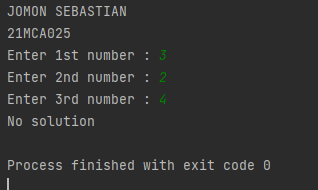
**OUTPUT**

****

1. Program to find the roots of a quadratic equation(rounded to 2 decimal places)

import math  
print(“JOMON SEBASTIAN\n21MCA025”)  
a=int(input("Enter 1st number : "))  
b=int(input("Enter 2nd number : "))  
c=int(input("Enter 3rd number : "))  
d=(b\*b)-(4\*a\*c)  
if d==0:  
 result=-b/2\*a  
 print ("The solution is : ",result)  
elif (d<0):  
 print("No solution")  
else:  
 result = (-b+math.sqrt(d))/(2\*a)  
 result1 = (-b-math.sqrt(d))/(2\*a)  
 print("The solutions are : ",result , result1)

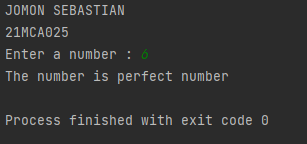
**OUTPUT**



1. Program to check whether a given number is perfect number or not (sum of factors =number)

print(“JOMON SEBASTIAN\n21MCA025”)  
a=int(input("Enter a number : "))  
b=0  
for i in range(1,a):  
 if a%i == 0:  
 b=b+i  
if a == b:  
 print("The number is perfect number ")  
else:  
 print("The number is not perfect ")

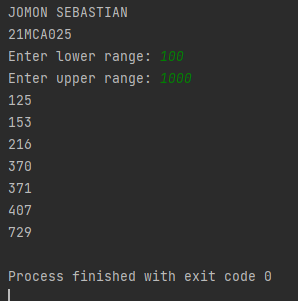
**OUTPUT**

****

1. Program to display amstrong numbers in an Interval

print(“JOMON SEBASTIAN\n21MCA025”)  
n1 = int(input("Enter the lower limit : "))  
n2 = int(input("Enter the upper limit : "))  
print("Amstrong numbers are:")  
  
for i in range(n1,n2):  
 count = len(str(i))  
 temp = i  
 sum = 0  
 while i > 0:  
 digit = i % 10  
 sum += digit \*\* count  
 i //= 10  
 if temp == sum:  
 print(temp)

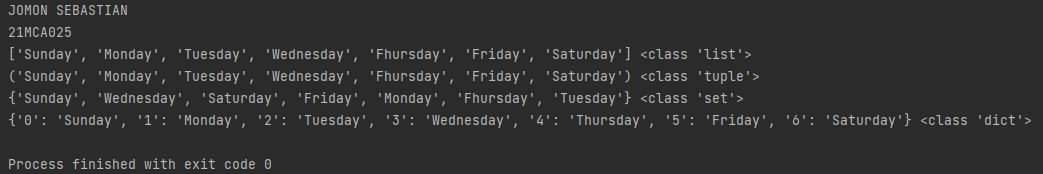
**OUTPUT**



1. Store and display the days of a week as a **List, Tuple, Dictionary, Set.** Also demonstrate different ways to store values in each of them. Display its type also.

print("JOYAL JOSE\n21MCA026")  
list1=['Sunday','Monday','Tuesday','Wednesday','Fhur sday','Friday','Saturday']  
tuples=('Sunday','Monday','Tuesday','Wednesday','Fhursday','Friday','Saturday')  
set1={'Sunday','Monday','Tuesday','Wednesday','Fhursday','Friday','Saturday'}  
dicts={'0':'Sunday','1':'Monday','2':'Tuesday','3':'Wednesday','4':'Thursday','5':'Friday','6':'Saturday'}  
  
print(list1,type(list1))  
print(tuples,type(tuples))  
print(set1,type(set1))  
print(dicts,type(dicts))

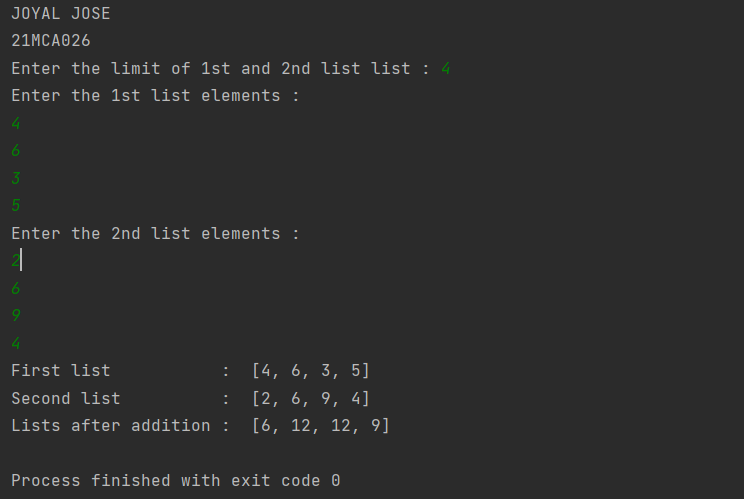
**OUTPUT**



1. Write a program to add elements of given 2 lists

print(“JOMON SEBASTIAN\n21MCA025”)  
n1=int(input("Enter the limit of 1st and 2nd list list : "))  
l1=[]  
print("Enter the 1st list elements : ")  
for i in range(0,n1):  
 x=int(input())  
 l1.append(x)  
l2 = []  
print("Enter the 2nd list elements : ")  
for i in range(0, n1):  
 y = int(input())  
 l2.append(y)  
l3=[]  
for i in range(0,len(l1)):  
 l3.append(l1[i] + l2[i])  
print("First list : ",l1)  
print("Second list : ",l2)  
print("Lists after addition : ",l3)

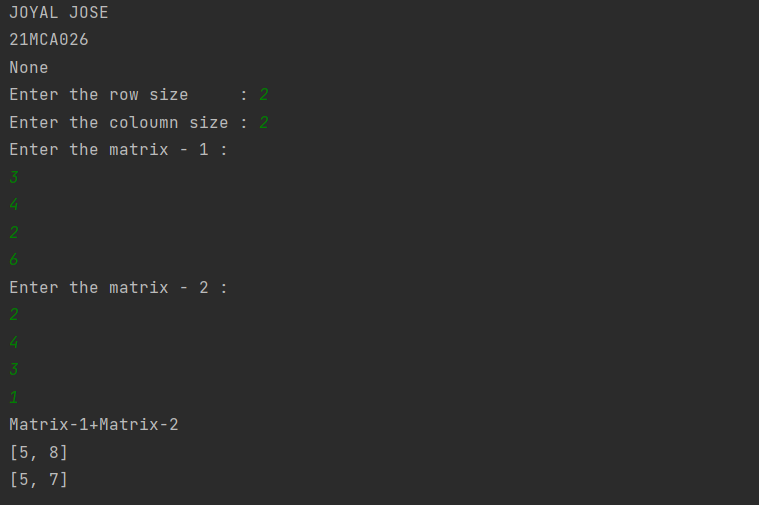
**OUTPUT**



1. Write a program to find the sum of 2 matrices using nested List.

print(print(“JOMON SEBASTIAN\n21MCA025”))  
n=int(input("Enter the row size : "))  
m=int(input("Enter the coloumn size : "))  
a=[]  
b=[]  
c=[]  
print("Enter the matrix - 1 :")  
for i in range(0,n):  
 a.append([])  
 for j in range(0,m):  
 p=int(input())  
 a[i].append(p)  
print("Enter the matrix - 2 :")  
for i in range(0,n):  
 b.append([])  
 for j in range(0,m):  
 q=int(input())  
 b[i].append(q)  
c=[[0 for i in range(m)] for i in range(n)]  
for i in range(n):  
 for j in range(m):  
 c[i][j]=a[i][j]+b[i][j]  
print("Matrix-1+Matrix-2")  
for r in c:  
 print(r)

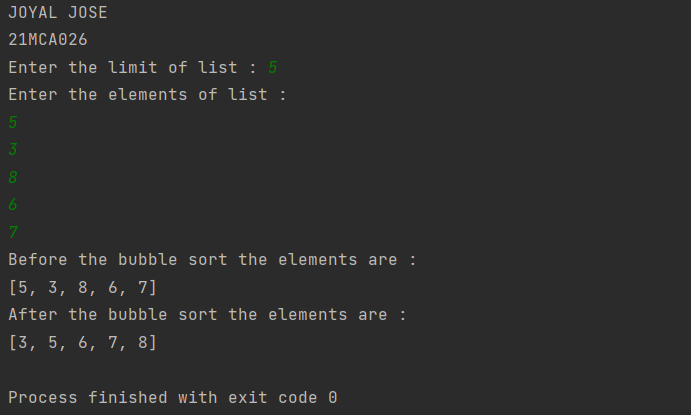
**OUTPUT**



1. Write a program to perform bubble sort on a given set of elements.

print(“JOMON SEBASTIAN\n21MCA025”)  
n=int(input("Enter the limit of list : "))  
a=[]  
print("Enter the elements of list : ")  
for i in range(n):  
 x=int(input())  
 a.append(x)  
print("Before the bubble sort the elements are : ")  
print(a)  
for i in range(n-1):  
 for j in range(n-1):  
 if (a[j]>a[j+1]):  
 temp=a[j]  
 a[j]=a[j+1]  
 a[j+1]=temp  
print("After the bubble sort the elements are : ")  
print(a)

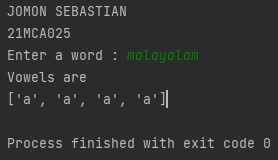
**OUTPUT**

****

1. Program to find the count of each vowel in a string (use dictionary)

print(“JOMON SEBASTIAN\n21MCA025”)  
vowels=[]  
vow={"a","A","e","E","i","I","o","O","u","U"}  
str=input("Enter a word : ")  
for i in str:  
 for x in vow:  
 if(i==x):  
 vowels.append(i)  
print("Vowels are")  
print(vowels)

**OUTPUT**



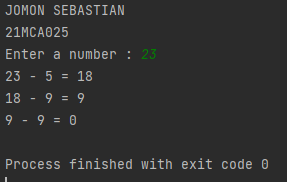
1. Write a Python program that accept a positive number and subtract from this number the sum of its digits and so on. Continues this operation until the number is positive (eg: 256->2+5+6=13

256-13=243

243-9=232…….

print(“JOMON SEBASTIAN\n21MCA025”)  
num=int(input("Enter a number : "))  
def digitsum(num):  
 sum=0  
 while num>0:  
 rem=num%10;  
 sum=sum+rem;  
 num=num//10  
 return sum  
  
while(num>0):  
 sum=digitsum(num)  
 print("{} - {} = {}".format(num,sum,num-sum))  
 num=num-sum

**OUTPUT**

****

1. Write a Python program that accepts a 10-digit mobile number, and find the digits

which are absent in a given mobile number

print("JOMON SEBASTIAN\n21MCA025")  
def absent\_digits(n):  
 all\_nums = set([0,1,2,3,4,5,6,7,8,9])  
 n = set([int(i) for i in n])  
 n = n.symmetric\_difference(all\_nums)  
 n = sorted(n)  
 return n  
print(absent\_digits([9,8,3,2,2,0

**OUTPUT**

