**CO1**

**Program-1**

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

print("SJC22MCA-2052 \n SONA MARIYA JOHN \n MCA 2022-2024 \n Output: ")

start=int(input("Enter the starting number:"))

end=int(input("Enter the ending number:"))

print("Non-prime numbers in the intervel",start,"to",end,"are:")

for num in range(start,end + 1):

if num > 1:

for i in range(2, num):

if(num % i) == 0:

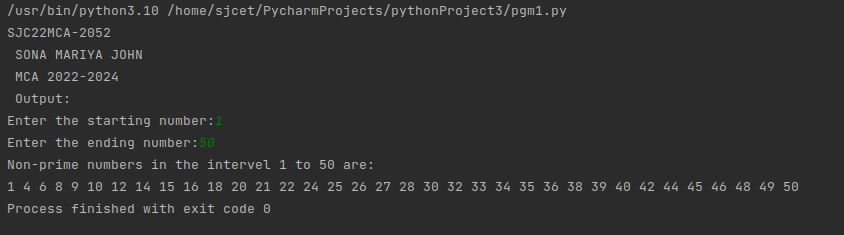
break

else:

continue

print(num,end=' ')

**Output**

****

**Program-2**

**Program to print the first N Fibonacci numbers.**

print("SJC22MCA-2052 \n SONA MARIYA JOHN \n MCA 2022-2024 \n Output: ")

n=int(input("Enter the number you want to print:"))

a=0

b=1

for i in range(0,n):

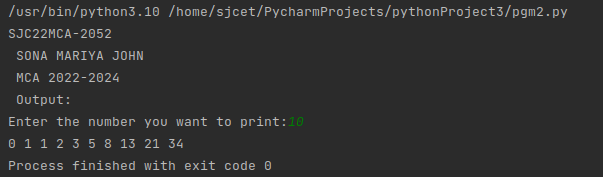
print(a,end=" ")

c=a+b

a=b

b=c

**Output**

****

**Program-3**

**Given sides of a triangle, write a program to check whether given triangle is an**

**isosceles, equilateral or scalene.**

print("SJC22MCA-2052 \n SONA MARIYA JOHN \n MCA 2022-2024 \n Output: ")

print("Input length of the triangle sides:")

x=int(input("Enter the 1st side length:"))

y=int(input("Enter the 2nd side length:"))

z=int(input("Enter the 3rd side length:"))

if x == y == z:

print("Equilateral triangle")

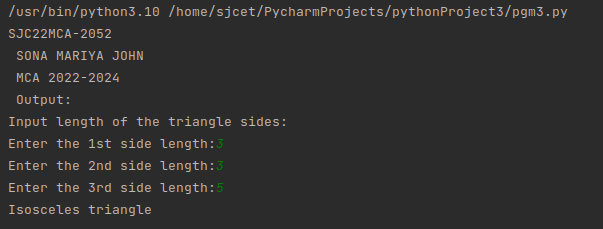
elif x == y or y == z or z == x:

print("Isosceles triangle")

else:

print("Scalene triangle")

**Output**

****

**Program-4**

**Program to check whether given pair of number is coprime**

import math

print("SJC22MCA-2052 \n SONA MARIYA JOHN \n MCA 2022-2024 \n Output: ")

num1 = int(input("Enter the first number:"))

num2 = int(input("Enter the second number:"))

gcd = math.gcd(num1, num2)

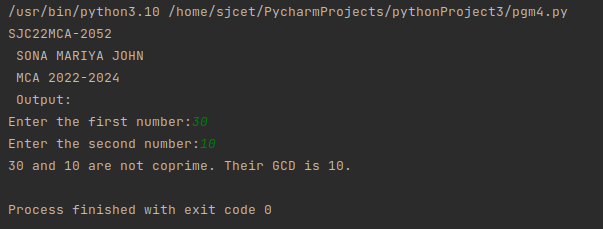
if gcd == 1:

print(f"{num1} and {num2} are coprime.")

else:

print(f"{num1} and {num2} are not coprime. Their GCD is {gcd}.")

**Output**

****

**Program-5**

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

import cmath

print("SJC22MCA-2052 \n SONA MARIYA JOHN \n MCA 2022-2024 \n Output: ")

a=float(input("Enter number a:"))

b=float(input("Enter the number b:"))

c=float(input("Enter the number c:"))

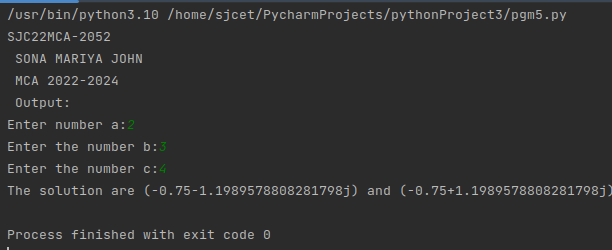
d=(b\*\*2) - (4\*a\*c)

sol1=(-b-cmath.sqrt(d))/(2\*a)

sol2=(-b+cmath.sqrt(d))/(2\*a)

print("The solution are {0} and {1}".format(sol1,sol2))

**Output**

****

**Program-6**

**Program to check whether a given number is perfect number or not(sum of factors**

**=number)**

print("SJC22MCA-2052 \n SONA MARIYA JOHN \n MCA 2022-2024 \n Output: ")

num=int(input("Enter the number"))

sum\_v=0

for i in range(1,num):

if(num%i==0):

sum\_v=sum\_v+i

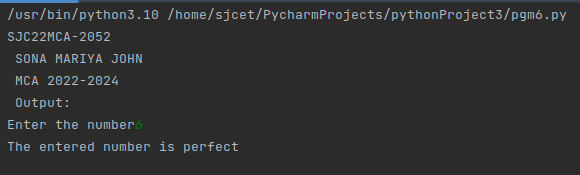
if(sum\_v==num):

print("The entered number is perfect")

else:

print("The entered number is not a perfect number")

**Output**

****

**Program-7**

**Program to display amstrong numbers upto 1000**

print("SJC22MCA-2052 \n SONA MARIYA JOHN \n MCA 2022-2024 \n Output: ")

lower=int(input("Enter the lower range"))

upper=int(input("Enter the upper range"))

for num in range(lower,upper+1):

sum=0

temp=num

while temp>0:

digit=temp%10

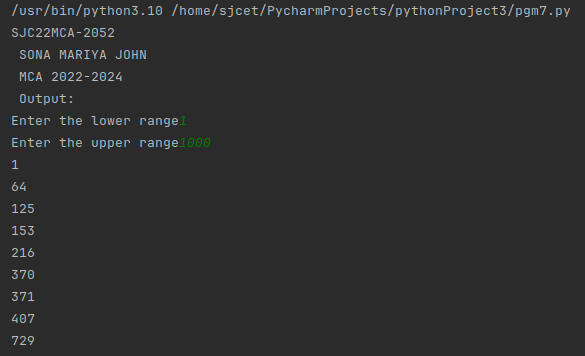
sum+=digit\*\*3

temp//=10

if num==sum:

print(num)

**Output**

****

**Program-8**

**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("SJC22MCA-2052 \n SONA MARIYA JOHN \n MCA 2022-2024 \n Output: ")

days\_list = ["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"]

print("List of days:",days\_list)

print("Types of days\_list:",type(days\_list))

days\_tuple=("Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday")

print("Tuple of days:",days\_tuple)

print("Types of days\_tuple:",type(days\_tuple))

days\_dict = {

1:"Monday",

2:"Tuesday",

3:"Wednesday",

4:"Thursday",

5:"Friday",

6:"Saturday",

7:"Sunday"

}

print("\nDictionary of days:",days\_dict)

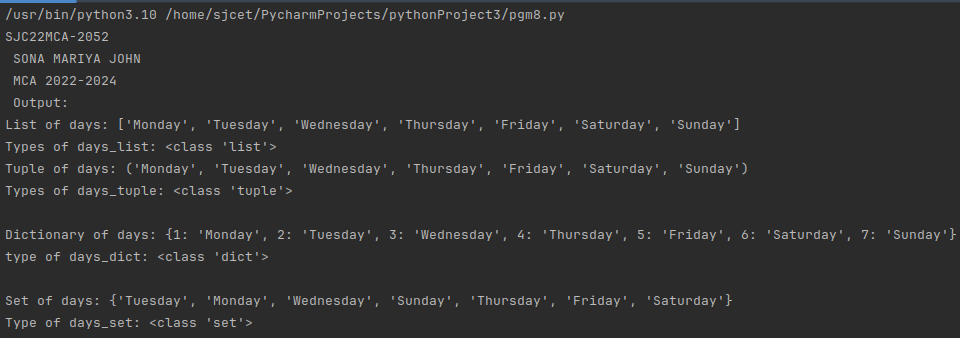
print("type of days\_dict:",type(days\_dict))

days\_set = {"Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"}

print("\nSet of days:",days\_set)

print("Type of days\_set:",type(days\_set))

**Output**

****

**Program-9**

**Write a program to add elements of given 2 lists**

lt1 = []

lt2 = []

lt3 = []

print("SJC22MCA-2052 \n SONA MARIYA JOHN \n MCA 2022-2024 \n Output: ")

items = int(input(" Enter the total number of the list elements: "))

print(" Enter the items into List 1 : ")

for i in range(1, items + 1):

num = int(input(" Enter the value of %d index is :" % i))

lt1.append(num)

print(" Enter the items into the List 2 : ")

for i in range(1, items + 1):

num = int(input(" Enter the value of %d index is :" % i))

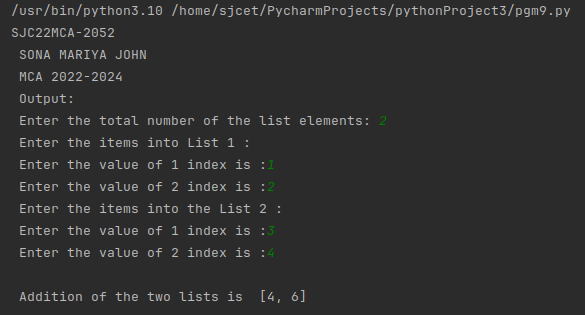
lt2.append(num)

for j in range(items):

lt3.append(lt1[j] + lt2[j])

print("\n Addition of the two lists is ", lt3)

**Output**

****

**Program-10**

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

print("SJC22MCA-2052 \n SONA MARIYA JOHN \n MCA 2022-2024 \n Output: ")

rows = int(input("Enter the number of rows: "))

cols = int(input("Enter the number of columns: "))

matrix1 = []

matrix2 = []

print("Enter elements for the first matrix:")

for i in range(rows):

row = []

for j in range(cols):

element = int(input(f"Enter element at position ({i+1},{j+1}): "))

row.append(element)

matrix1.append(row)

print("Enter elements for the second matrix:")

for i in range(rows):

row = []

for j in range(cols):

element = int(input(f"Enter element at position ({i+1},{j+1}): "))

row.append(element)

matrix2.append(row)

result\_matrix = []

for i in range(rows):

row = []

for j in range(cols):

sum\_element = matrix1[i][j] + matrix2[i][j]

row.append(sum\_element)

result\_matrix.append(row)

print("The sum of the two matrices is:")

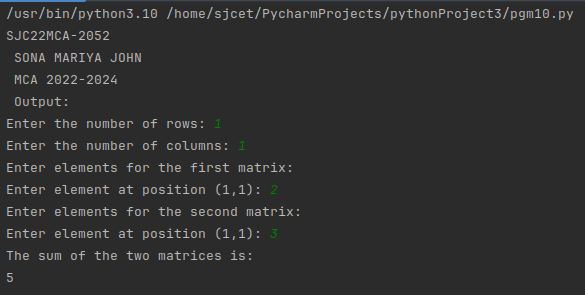
for i in range(rows):

for j in range(cols):

print(result\_matrix[i][j], end=" ")

print()

**Output**

****

**Program-11**

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

def bubble\_sort(arr):

n = len(arr)

for i in range(n):

swapped = False

for j in range(0, n - i - 1):

if arr[j] > arr[j + 1]:

arr[j], arr[j + 1] = arr[j + 1], arr[j]

swapped = True

if not swapped:

break

try:

print("SJC22MCA-2052 \n SONA MARIYA JOHN \n MCA 2022-2024 \n Output: ")

elements = input("Enter elements separated by spaces: ").split()

elements = [int(x) for x in elements]

bubble\_sort(elements)

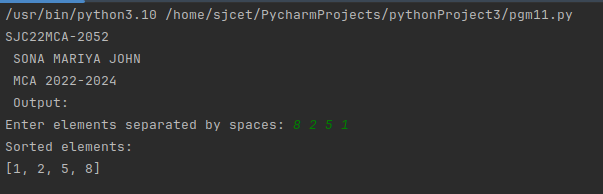
print("Sorted elements:")

print(elements)

except ValueError:

print("Invalid input. Please enter valid numbers separated by spaces.")

**Output**



**Program-12**

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

def count\_vowels(string):

vowel\_counts = {'A': 0, 'E': 0, 'I': 0, 'O': 0, 'U': 0}

string = string.upper()

for char in string:

if char in vowel\_counts:

vowel\_counts[char] += 1

return vowel\_counts

try:

print("SJC22MCA-2052 \n SONA MARIYA JOHN \n MCA 2022-2024 \n Output: ")

input\_string = input("Enter a string: ")

vowel\_counts = count\_vowels(input\_string)

print("Vowel counts:")

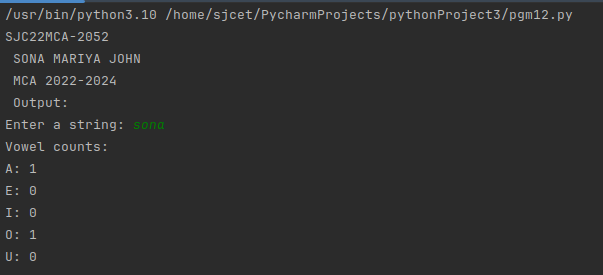
for vowel, count in vowel\_counts.items():

print(f"{vowel}: {count}")

except ValueError:

print("Invalid input. Please enter a valid string.")

**Output**

****

**Program-13**

**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-&gt;2+5+6=13**

**256-13=243**

**243-9=232……..**

def sum\_of\_digits(n):

digit\_sum = 0

while n > 0:

digit\_sum += n % 10

n //= 10

return digit\_sum

try:

print("SJC22MCA-2052 \n SONA MARIYA JOHN \n MCA 2022-2024 \n Output: ")

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

if num <= 0:

print("Please enter a positive number.")

else:

while num > 0:

digit\_sum = sum\_of\_digits(num)

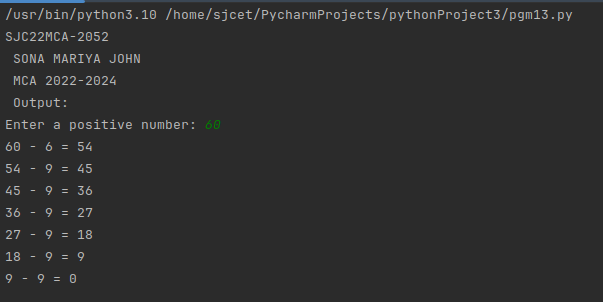
print(f"{num} - {digit\_sum} = {num - digit\_sum}")

num -= digit\_sum

except ValueError:

print("Invalid input. Please enter a valid positive number.")

**Output**

****

**Program-14**

**Write a Python program that accepts a 10 digit mobile number, and find the digits**

**which are absent in a given mobile number**

def find\_absent\_digits(mobile\_number):

all\_digits = set("0123456789")

mobile\_digits = set(mobile\_number)

absent\_digits = all\_digits - mobile\_digits

return sorted(list(absent\_digits))

try:

print("SJC22MCA-2052 \n SONA MARIYA JOHN \n MCA 2022-2024 \n Output: ")

mobile\_number = input("Enter a 10-digit mobile number: ")

if len(mobile\_number) == 10 and mobile\_number.isdigit():

absent\_digits = find\_absent\_digits(mobile\_number)

if absent\_digits:

print("Absent digits in the mobile number:", ', '.join(absent\_digits))

else:

print("The mobile number contains all digits from 0 to 9.")

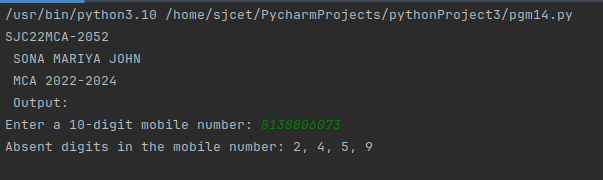
else:

print("Invalid input. Please enter a valid 10-digit mobile number.")

except ValueError:

print("Invalid input. Please enter a valid 10-digit mobile number.")

**Output**

****