CYCLE-1

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

print("SJC22MCA-2053 \nSREELAKSHMI M \nMCA 2022-2024 \nOutput: ")

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:

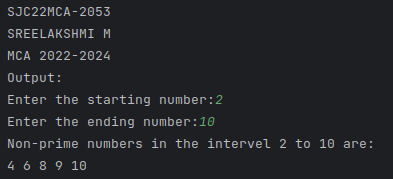
break

else:

continue

print(num,end=' ')

**OUTPUT**

****

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

print("SJC22MCA-2053 \nSREELAKSHMI M \nMCA 2022-2024 \nOutput: ")

n=int(input("Enter the number of Fibonacci numbers to print:"))

a=0

b=1

for i in range(0,n):

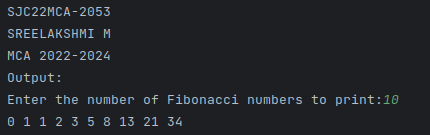
print(a,end=" ")

c=a+b

a=b

b=c

**OUTPUT**

****

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

print("SJC22MCA-2053 \nSreelakshmi M \n2022-2024 \nOutput:")

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

x = float(input("x: "))

y = float(input("y: "))

z = float(input("z: "))

if x == y == z:

print("Equilateral triangle")

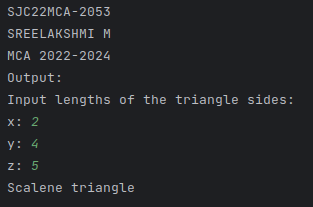
elif x==y or y==z or z==x:

print("isosceles triangle")

else:

print("Scalene triangle")

**OUTPUT**

****

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

import math

print("SJC22MCA-2053 \nSREELAKSHMI M \nMCA 2022-2024 \nOutput: ")

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

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

gcd = math.gcd(num1, num2)

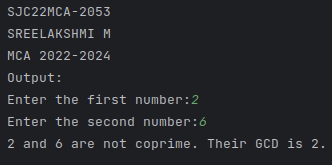
if gcd == 1:

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

else:

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

**OUTPUT**

****

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

import cmath

print("SJC22MCA-2053 \nSREELAKSHMI M \nMCA 2022-2024 \nOutput: ")

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

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

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

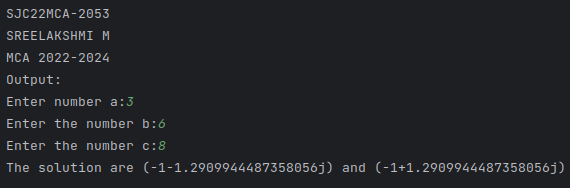
d=(b\*\*2) - (4\*a\*c)

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**

****

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

print("SJC22MCA-2053 \nSREELAKSHMI M \nMCA 2022-2024 \nOutput: ")

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

sum\_of\_factors = 0

for i in range(1,num):

if num % i == 0:

sum\_of\_factors += i

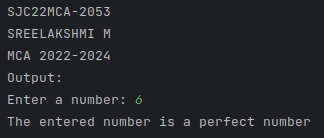
if sum\_of\_factors == num:

print("The entered number is a perfect number")

else:

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

**OUTPUT**

****

1. **Program to display amstrong numbers upto 1000.**

print("SJC22MCA-2053 \nSREELAKSHMI M \nMCA 2022-2024 \nOutput: ")

for num in range(1, 1001):

order = len(str(num))

sum = 0

temp = num

while temp > 0:

digit = temp % 10

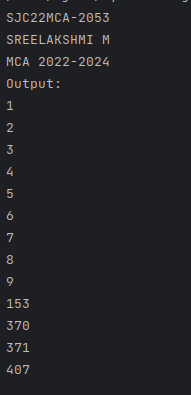
sum += digit \*\* order

temp //= 10

if num == sum:

print(num)

**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("SJC22MCA-2053 \nSREELAKSHMI M \nMCA 2022-2024 \nOutput: ")

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

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

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

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

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

print("Type 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**



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

print("SJC22MCA-2053 \nSREELAKSHMI M \nMCA 2022-2024 \nOutput: ")

def add\_lists(list1, list2):

if len(list1) != len(list2):

return None

result\_list = [a + b for a, b in zip(list1, list2)]

return result\_list

list1 = input("Enter elements for the first list (comma-separated): ").split(',')

list1 = [int(x) for x in list1]

list2 = input("Enter elements for the second list (comma-separated): ").split(',')

list2 = [int(x) for x in list2]

result = add\_lists(list1, list2)

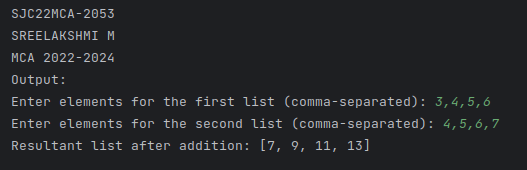
if result:

print("Resultant list after addition:", result)

else:

print("Lists are of different lengths. Addition not possible.")

**OUTPUT**

****

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

print("SJC22MCA-2053 \nSREELAKSHMI M \nMCA 2022-2024 \nOutput: ")

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:")

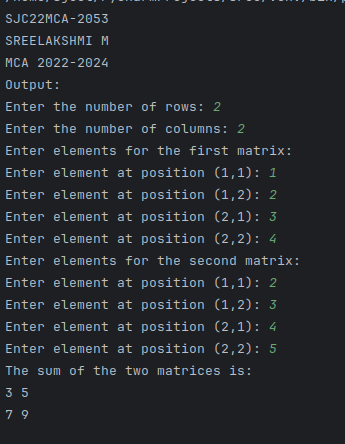
for i in range(rows):

for j in range(cols):

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

print()

**OUTPUT**



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

print("SJC22MCA-2053 \nSREELAKSHMI M \nMCA 2022-2024 \nOutput: ")

def bubble\_sort(arr):

n = len(arr)

for i in range(n):

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

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

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

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

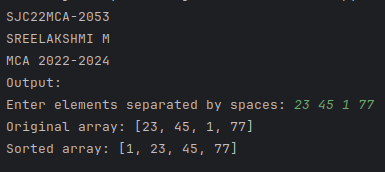
elements = [int(e) for e in elements]

print("Original array:", elements)

bubble\_sort(elements)

print("Sorted array:", elements)

**OUTPUT**



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

print("SJC22MCA-2053 \nSREELAKSHMI M \nMCA 2022-2024 \nOutput: ")

se=input("enter the sentence:");

string=se.lower()

print(string)

count=0

list1=["a","e","i","o","u"]

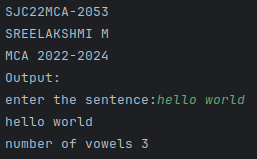
for char in string:

if char in list1:

count=count+1

print("number of vowels",count)

**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**

def sum\_of\_digits(n):

digit\_sum = 0

while n > 0:

digit\_sum += n % 10

n //= 10

return digit\_sum

try:

print("SJC22MCA-2053\nSREELAKSHMI M\nMCA 2022-2024\nOutput:")

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)

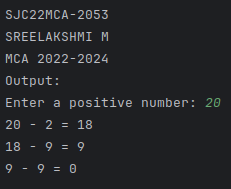
print(f"{num} - {digit\_sum} = {num - digit\_sum}")

num -= digit\_sum

except ValueError:

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

**OUTPUT**

****

1. **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-2053\nSREELAKSHMI M\nMCA 2022-2024\nOutput:")

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.")

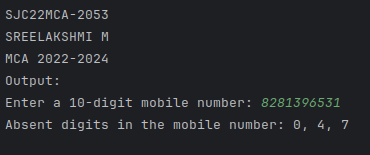
else:

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**

****