LAB CYCLE 1

PRGM NO:1: WRITE A PROGRAM TO PRINT ALL NON PRIME NO.'S IN AN INTERVAL?

CODE:

```
print("EMMANUEL.A \n MCA")
def is_prime(num):
  if num <= 1:
      return False
  for i in range (2,num):
      if num % i == 0:
          return False
  return True
start=int(input("Enter the starting number: "))
end=int(input("Enter the end number: "))
print("non-prime numbers between {start} and {end} is: ")
for num in range(start,end+1):
  if not is_prime(num):
      print(num,end=" ")
```

Output:

```
EMMANUEL.A
 MCA
Enter the starting number: 5
Enter the end number: 9
non-prime numbers between {start} and {end} is:
6 8 9
Process finished with exit code 0
```

PRGM.NO:2 FIND FIRST N FIBONACCI NUMBER?

CODE:

```
print("EMMANUEL \n MCA")
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:

```
EMMANUEL
 MCA
Enter the number you want to print: 8
0 1 1 2 3 5 8 13
Process finished with exit code 0
```

PROGRAM NO.3 Find What Type Of Triangle?

Code:

```
print("SJC22MCA-2023 : EMMANUEL")
print("MCA 2022-24")
print("**************************************************")
side1 = float(input("Enter the length of side 1: "))
side2 = float(input("Enter the length of side 2: "))
side3 = float(input("Enter the length of side 3: "))

if side1 + side2 > side3 and side1 + side3 > side2 and side2 + side3 > side1:
    if side1 == side2 == side3:
        print("It is an equilateral triangle.")
    elif side1 == side2 or side1 == side3 or side2 == side3:
        print("It is an isosceles triangle.")
    else:
        print("It is a scalene triangle.")
else:
    print("It is not a valid triangle.")
```

Output:

```
SJC22MCA-2035 : EMMANUEL
MCA 2022-24
**************************************************
Enter the length of side 1: 5
Enter the length of side 2: 3
Enter the length of side 3: 7
It is a scalene triangle.
```

Program no:4 find the root of quadratic equation?

Code:

```
import math
print("****************************************************")
print("SJC22MCA-2023 : EMMANUEL")
print("MCA 2022-24")
print("****************************************************")
a = float(input("Enter coefficient 'a': "))
b = float(input("Enter coefficient 'b': "))
c = float(input("Enter coefficient 'c': "))

discriminant = b**2 - 4*a*c

if discriminant >= 0:
    root1 = (-b + math.sqrt(discriminant)) / (2*a)
    root2 = (-b - math.sqrt(discriminant)) / (2*a)
    print(f"Root 1: {root1:.2f}")
    print(f"Root 2: {root2:.2f}")
else:
    print("No real roots")
```

Output:

```
SJC22MCA-2023  EMMANUEL
MCA 2022-24
****************************************************
Enter coefficient 'a': 2
Enter coefficient 'b': 4
Enter coefficient 'c': 8
No real roots
```

Program no:5 find number as coprime or not?

Code:

```
import math

print("**************************************************")
print("SJC22MCA-2023 :EMMANUEL ")
print("MCA 2022-24")
print("**************************************************")
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:

```
SJC22MCA-2023 : EMMANUEL
MCA 2022-24
**************************************************
Enter the first number: 6
Enter the second number: 9
6 and 9 are not coprime. Their GCD is 3.
```

Programno:6 sum of 2 matrix using nested list?

Code:

```
print("SJC22MCA-2023:EMMANUEL")
print("MCA 2022-24")
print("**************************************************")
def input_matrix(rows, cols):
```

```
    matrix = []
    for i in range(rows):
        row = []
        for j in range(cols):
            element = float(input(f"Enter the element at row {i+1}, column {j+1}: "))
            row.append(element)
        matrix.append(row)
    return matrix

def add_matrices(matrix1, matrix2):
    if len(matrix1) != len(matrix2) or len(matrix1[0]) != len(matrix2[0]):
        return None
    result = []
    for i in range(len(matrix1)):
        row = []
        for j in range(len(matrix1[0])):
            element = matrix1[i][j] + matrix2[i][j]
            row.append(element)
        result.append(row)
    return result

def display_matrix(matrix):
    for row in matrix:
        for element in row:
            print(element, end="\t")
        print()


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


print("Enter elements of the first matrix:")
matrix1 = input_matrix(rows, cols)


print("Enter elements of the second matrix:")
matrix2 = input_matrix(rows, cols)


result_matrix = add_matrices(matrix1, matrix2)

if result_matrix:
    print("\nMatrix 1:")
    display_matrix(matrix1)
    print("\nMatrix 2:")
```

```
    display_matrix(matrix2)
    print("\nSum of the matrices:")
    display_matrix(result_matrix)
else:
    print("Matrix addition is not possible. Matrices are of different sizes.")
```

Output:

```
Matrix 1:
1.0     2.0     3.0
4.0     5.0     6.0
7.0     6.0     5.0

Matrix 2:
4.0     3.0     2.0
1.0     2.0     3.0
4.0     5.0     6.0

Sum of the matrices:
5.0     5.0     5.0
5.0     7.0     9.0
```

Program no:7 Count vowel in string?

Code:

```
print("**************************************************")
print("SJC22MCA-2023 : EMMANUEL")
```

```python
print("MCA 2022-24")
print("**************************************************")
def count_vowels(input_string):

    vowel_counts = {'a': 0, 'e': 0, 'i': 0, 'o': 0, 'u': 0}


    input_string = input_string.lower()


    for char in input_string:
        if char in vowel_counts:
            vowel_counts[char] += 1

    return vowel_counts


input_string = input("Enter a string: ")


vowel_count = count_vowels(input_string)

for vowel, count in vowel_count.items():
    print(f"{vowel}: {count}")
```

Output:

```
SJC22MCA-2023 :EMMANUEL
MCA 2022-24
**************************************************
Enter a string: emmanuel
a: 1
e: 2
i: 0
o: 0
u: 1
```

Program no:8 Accept a positive number?

Code:

```
print("**************************************************")
print("SJC22MCA-2023:   EMMANUEL")
print("MCA 2022-24")
print("**************************************************")


def sum_of_digits(n):

    digit_sum = 0
    while n > 0:
        digit_sum += n % 10
        n //= 10
    return digit_sum



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

while num > 0:

    digit_sum = sum_of_digits(num)


    num -= digit_sum

    print(f"{num + digit_sum} - {digit_sum} = {num}")

print("The number is now positive or zero.")
```

Output:

```
SJC22MCA-2023 :EMMANUEL
MCA 2022-24
**************************************************
Enter a positive number: 20
20 - 2 = 18
18 - 9 = 9
9 - 9 = 0
The number is now positive or zero.
PS C:\Users\emman>
```

Program no: 9

Code:

```
print("**************************************************")
print("SJC22MCA-2023  :  EMMANUEL")
print("MCA 2022-24")
print("**************************************************")

list1 = [1, 2, 3, 4, 5]
list2 = [6, 7, 8, 9, 10]


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

    result = []


    for i in range(len(list1)):
        sum_element = list1[i] + list2[i]
        result.append(sum_element)

    print("Resultant list after adding elements:", result)
else:
    print("Lists are of different lengths. Cannot perform element-wise addition.")
```

output:

```
**************************************************
SJC22MCA-2023  :EMMANUEL
MCA 2022-24
**************************************************
Resultant list after adding elements: [7, 9, 11, 13, 15]
PS C:\Users\emman>
```

Program no:10 store and display days of a week?

Code:

```
print("**************************************************")
print("SJC22MCA-2023 :EMMANUEL")
print("MCA 2022-24")
print("**************************************************")
days_list = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday",
"Sunday"]

days_tuple = ("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday",
"Sunday")

days_dict = {
   1: "Monday",
   2: "Tuesday",
   3: "Wednesday",
   4: "Thursday",
   5: "Friday",
   6: "Saturday",
   7: "Sunday"
}

days_set = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday",
"Sunday"}


print("Type of days_list:", type(days_list))
```

```
print("Days in days_list:", days_list)

print("\nType of days_tuple:", type(days_tuple))
print("Days in days_tuple:", days_tuple)

print("\nType of days_dict:", type(days_dict))
print("Days in days_dict:", days_dict)

print("\nType of days_set:", type(days_set))
print("Days in days_set:", days_set)
```

Output:

```
SJC22MCA-2023  : EMMANUEL
MCA 2022-24
**************************************************
Type of days_list: <class 'list'>
Days in days_list: ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']

Type of days_tuple: <class 'tuple'>
Days in days_tuple: ('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday')

Type of days_dict: <class 'dict'>
Days in days_dict: {1: 'Monday', 2: 'Tuesday', 3: 'Wednesday', 4: 'Thursday', 5: 'Friday', 6: 'Saturday', 7: 'Sunday'}
```

Program no: 11

Code:

```
print("**************************************************")
print("SJC22MCA-2023 :EMMANUEL")
print("MCA 2022-24")
print("**************************************************")
def is_armstrong_number(n):

    num_digits = len(str(n))


    sum_of_powers = sum(int(digit) ** num_digits for digit in str(n))
```

```
    return n == sum_of_powers


for num in range(1, 1001):
    if is_armstrong_number(num):
        print(num)
```

Output:

```
2
3
4
5
6
7
8
9
153
370
371
407
```

<u>Program no.14</u>

<u>Enter mobile number which is absent?</u>

```
def find_absent_digits(mobile_number):
    all_digits = set('0123456789')
    mobile_digits = set(mobile_number)
    absent_digits = all_digits - mobile_digits
```

```
    return sorted(absent_digits)

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

if len(mobile_number) != 10:
    print("Please enter a valid 10-digit mobile number.")
else:
    absent_digits = find_absent_digits(mobile_number)
    if not absent_digits:
        print("All digits are present in the mobile number.")
    else:
        print("Absent digits:", ', '.join(absent_digits))
```

Output:

```
Enter a 10-digit mobile number: 8606129607
Absent digits: 3, 4, 5
```