**Name: Meet Raut**

**Batch: S21**

**Roll number: 2201084**

**PYTHON LAB**

EXPERIMENT 1-C

**AIM**:

To study about sequences and apply it to writing three programs:

1. Find the sum of series : 1+½+⅓ +....+1/N
2. To determine all the Pythagorean triplets between 1 and 50.
3. Find the sum of series : 1+x2/2 +x3/3+....+ xN/N

**THEORY:**

Python’s features like if-else, for-loops and nested loops help a lot for finding and simplifying problems on sequencing. These offer a robust framework for working with mathematical series. These tools empower you to explore, analyse, and manipulate series, gaining insights into diverse mathematical concepts and phenomena.

With Python's readability and simplicity, implementing mathematical algorithms becomes straightforward. Whether you're calculating Euler's series to understand exponential growth, generating Fibonacci numbers to model natural patterns, or finding Pythagorean triplets to explore geometric relationships, Python provides an intuitive environment for mathematical exploration and experimentation.

**CODE:**

# Write a program to find sum of series

print("Meet Raut    S21     2201084")

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

sum = 0

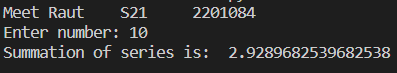
for i in range (1, n+1):

    sum = sum + (1/i)

    i = i + 1

print("Summation of series is: ", sum)

**OUTPUT :**



**CODE:**

print("Meet Raut    S21     2201084")

triplets = []

count = 0

print("1. With duplicates     2. Without duplicates")

choice = int(input("Your choice: "))

if(choice == 1):

     for a in range (1, 51):

          for b in range (1, 51):

               for c in range (1, 51):

                    if(a\*\*2 + b\*\*2 == c\*\*2):

                         count += 1

                         triplets.append((a,b,c))

elif(choice == 2):

     for a in range (1, 51):

          for b in range (a, 51):

               for c in range (1, 51):

                    if(a\*\*2 + b\*\*2 == c\*\*2):

                         count += 1

                         triplets.append((a,b,c))

else:

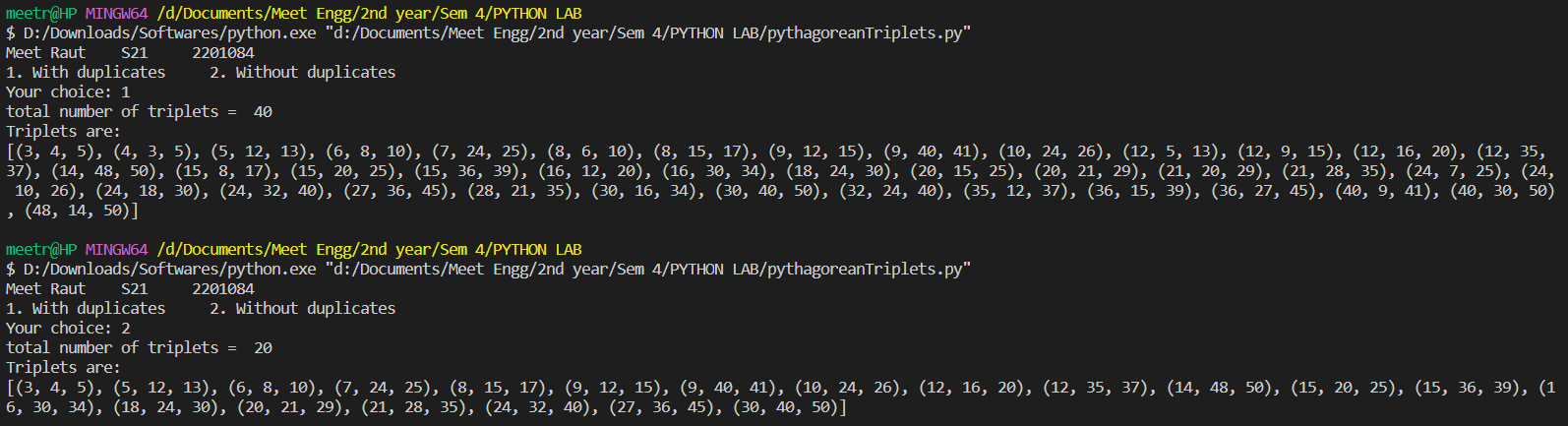
     print("Invalid input")

if(choice == 1 or choice == 2):

     print("total number of triplets = ", count, "\nTriplets are: ")

     print(triplets)

**OUTPUT :**



**CODE:**

# Write a program to print the sine and cosine series

import math

print("Meet Raut    S21     2201084")

print("\*\*\*MENU\*\*\*")

print("1. Sine series   2. Cosine series")

choice = int(input("Enter your choice: "))

x = int(input("Enter x: "))

if(choice == 1):

    sum = x

    num = 1

    print("This is Sine serie")

    for i in range (0, 84):

        #print(sum, end = " ")

        #sum = sum + ((-1)\*\*i) \* ((x\*\*num)/ math.factorial(num))

        print(((-1)\*\*i) \* ((x\*\*num)/ math.factorial(num)), end = " ")

        num = num + 2

elif(choice == 2):

    print("This is Cosine series")

    sum = 1

    num = 0

    for i in range (0, 84):

        #print(sum, end = " ")

        #sum = sum + ((-1)\*\*i) \* ((x\*\*num)/ math.factorial(num))

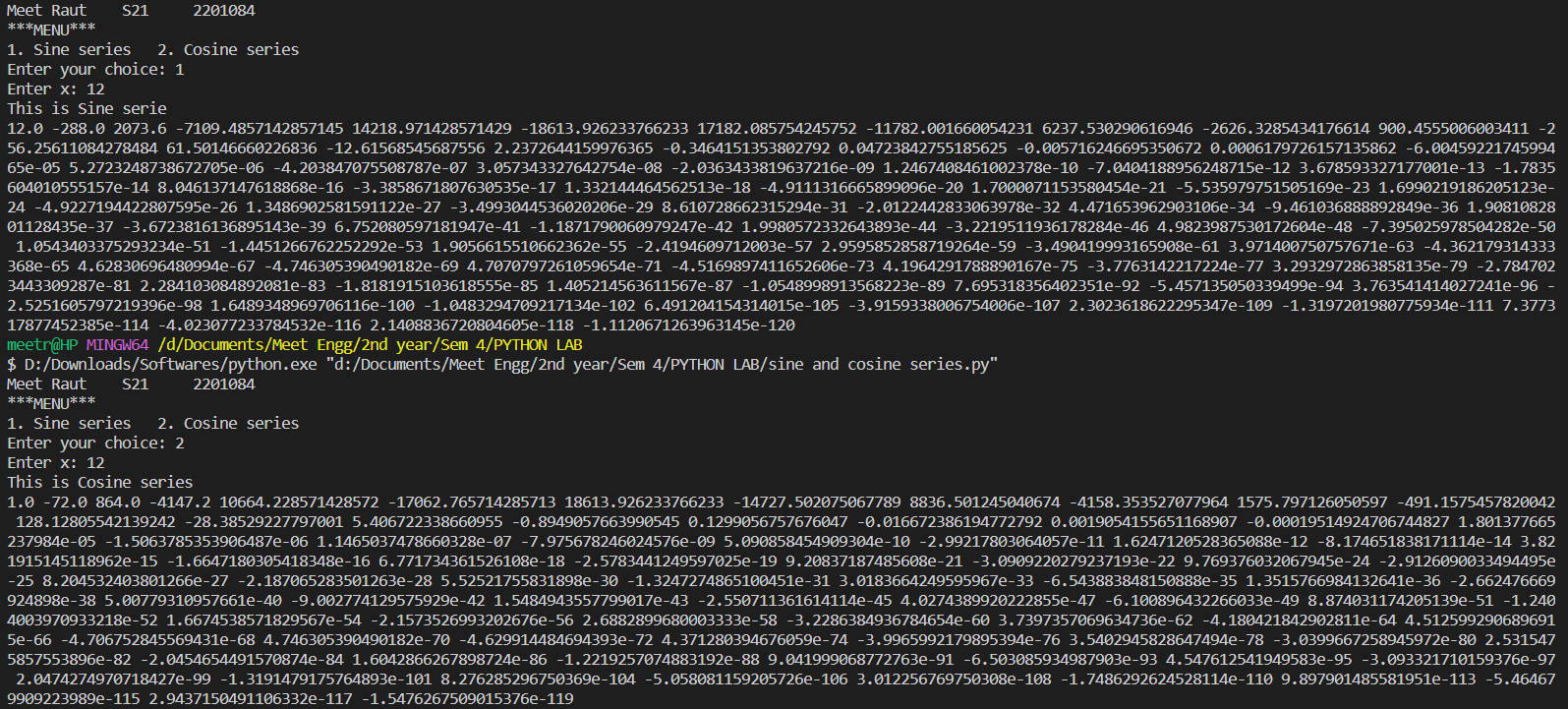
        print(((-1)\*\*i) \* ((x\*\*num)/ math.factorial(num)), end = " ")

        num = num + 2

else:

    print("Invalid choice")

**OUTPUT :**



**CONCLUSION :**

In conclusion, Python's versatility as a programming language, coupled with its rich set of features for handling loops, conditional statements, and functions, provides a powerful toolkit for working with mathematical series. From Euler's series to Fibonacci sequences and Pythagorean triplets, Python enables users to explore these mathematical phenomena with ease and clarity. By leveraging Python's simplicity and readability, mathematicians, scientists, and programmers can bridge the gap between abstract mathematical concepts and practical computational solutions. Whether for research, education, or creative exploration, Python serves as an invaluable ally in the journey to unlock the secrets hidden within mathematical series.