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 : $1+x^2/2 +x^3/3+....+ x^N/N$

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 :

```
Meet Raut    S21     2201084
Enter number: 10
Summation of series is:  2.9289682539682538
```

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 :

```
meetr@HP MINGW64 /d/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB
$ D:/Downloads/Softwares/python.exe "d:/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB/pythagoreanTriplets.py"
Meet Raut    S21     2201084
1. With duplicates     2. Without duplicates
Your choice: 1
total number of triplets =  40
Triplets are:
[(3, 4, 5), (4, 3, 5), (5, 12, 13), (6, 8, 10), (7, 24, 25), (8, 6, 10), (8, 15, 17), (9, 12, 15), (9, 40, 41), (10, 24, 26), (12, 5, 13), (12, 9, 15), (12, 16, 20), (12, 35,
37), (14, 48, 50), (15, 8, 17), (15, 20, 25), (15, 36, 39), (16, 12, 20), (16, 30, 34), (18, 24, 30), (20, 15, 25), (20, 21, 29), (21, 20, 29), (21, 28, 35), (24, 7, 25), (24,
 10, 26), (24, 18, 30), (24, 32, 40), (27, 36, 45), (28, 21, 35), (30, 16, 34), (30, 40, 50), (32, 24, 40), (35, 12, 37), (36, 15, 39), (36, 27, 45), (40, 9, 41), (40, 30, 50)
, (48, 14, 50)]

meetr@HP MINGW64 /d/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB
$ D:/Downloads/Softwares/python.exe "d:/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB/pythagoreanTriplets.py"
Meet Raut    S21     2201084
1. With duplicates     2. Without duplicates
Your choice: 2
total number of triplets =  20
Triplets are:
[(3, 4, 5), (5, 12, 13), (6, 8, 10), (7, 24, 25), (8, 15, 17), (9, 12, 15), (9, 40, 41), (10, 24, 26), (12, 16, 20), (12, 35, 37), (14, 48, 50), (15, 20, 25), (15, 36, 39), (1
6, 30, 34), (18, 24, 30), (20, 21, 29), (21, 28, 35), (24, 32, 40), (27, 36, 45), (30, 40, 50)]
```

CODE:

```python
# 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 :

```
Meet Raut    S21     2201084
***MENU***
1. Sine series   2. Cosine series
Enter your choice: 1
Enter x: 12
This is Sine serie
12.0 -288.0 2073.6 -7109.4857142857145 14218.971428571429 -18613.926233766233 17182.085754245752 -11782.001660054231 6237.530290616946 -2626.32854341766l4 900.4555006003411 -2
56.25611084278484 61.50146660226836 -12.615685456875S6 2.2372644159976365 -0.3464151353802792 0.04723842755185625 -0.005716246695350672 0.000617972615713586Z -6.004592217459946
65e-05 5.272324873867270Se-06 -4.203847075508787e-07 3.057343327642754e-08 -2.036343381963721Ge-09 1.2467408461002378e-10 -7.040418895624871Se-12 3.6785933271770Ole-13 -1.7835
604010555157e-14 8.0461371476188686-16 -3.3858671807630535e-17 1.3321444645625l3e-18 -4.9111316665899096e-20 1.7000071153580454e-21 -5.5359797515051696-23 1.6990219186205123e-
24 -4.9227194422807595e-26 1.3486902581591122e-27 -3.4993044536020206e-29 8.610728662315294e-31 -2.0122442833063978e-32 4.4716539629031O6e-34 -9.461036888892849e-36 1.90810828
01128435e-37 -3.6723816136895143e-39 6.752080597181947e-41 -1.1871790060979247e-42 1.9980572332643893e-44 -3.2219511936178284e-46 4.9823987530172604e-48 -7.395025978504282e-50
 1.0543403375293234e-51 -1.4451266762252292e-53 1.9056615510662362e-55 -2.4194609712003e-57 2.9595852858719264e-59 -3.4904199931659O8e-61 3.971400750757671e-63 -4.362179314333
368e-65 4.628306964809Ge-67 -4.7463059390490182e-69 4.7070797261059654e-71 -4.5169897411652606e-73 4.1964291788890167e-75 -3.7763142217224e-77 3.2932972863858135e-79 -2.784702
3443309287e-81 2.284103084892081e-83 -1.8181915103618555e-85 1.4052145636115Ge-87 -1.0548998913568223e-89 7.6953183564023Sle-92 -5.457135050339499e-94 3.763541414027241e-96 -
2.525160579721939Ge-98 1.6489348969706116e-100 -1.0483294709217134e-102 6.491204154314015e-105 -3.9159338006754006e-107 2.3023618622295347e-109 -1.3197201980775934e-111 7.3773
17877452385e-114 -4.0230772337845Z2e-116 2.1408836720804605e-118 -1.1120671263963145e-120
meetr@HP MINGW64 /d/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB
$ D:/Downloads/Softwares/python.exe "d:/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB/sine and cosine series.py"
Meet Raut    S21     2201084
***MENU***
1. Sine series   2. Cosine series
Enter your choice: 2
Enter x: 12
This is Cosine series
1.0 -72.0 864.0 -4147.2 10664.228571428572 -17062.765714285713 18613.926233766233 -14727.502075067789 8836.501245040674 -4158.353527077964 1575.797126050597 -491.1575457820042
 128.12805542139242 -28.38529227797001 5.406722338660955 -0.8949057663990545 0.1299056757676047 -0.016672386194772792 0.001905415565116890Z -0.000195149247067448Z7 1.801377665
237984e-05 -1.5063785353906487e-06 1.1465037478660328e-07 -7.975678246024576e-09 5.090858454909304e-10 -2.992178030640S7e-11 1.624712052836508Be-12 -8.174651838171114e-14 3.82
191514511896Ze-15 -1.664718030541834Be-16 6.771734361526108e-18 -2.578344124959702Se-19 9.208371874856O8e-21 -3.090922027923719Ze-22 9.769376032067945e-24 -2.912609003349449Se
-25 8.204532403801266e-27 -2.187065283501263e-28 5.525217558318980e-30 -1.324727486510045Ie-31 3.018366424959596Ze-33 -6.543883848150888e-35 1.351517669841326Ale-36 -2.662476669
924898e-38 5.007793109576Ole-40 -9.002774129575929e-42 1.548494355779901Ze-43 -2.550711361614114e-45 4.027438992022285Se-47 -6.100896432266033e-49 8.874031174205139e-51 -1.240
4003970933218e-52 1.667453857182956Ze-54 -2.157352699320267Ge-56 2.688289968000333Se-58 -3.228638493678465Ae-60 3.739735706963473Ge-62 -4.180421842902811e-64 4.512599290689691
5e-66 -4.706752845569431e-68 4.746305390490182e-70 -4.629914484694393e-72 4.371280394676059e-74 -3.996599217989539Ae-76 3.540294582864749Ae-78 -3.039966725894597Ze-80 2.531547
585755389Ge-82 -2.045465449157087Ae-84 1.604286626789872Ae-86 -1.221925707488319Ze-88 9.041999068772763e-91 -6.503085934987903e-93 4.547612541949583e-95 -3.093321710159376e-97
 2.047427497071842Ze-99 -1.319147917576489Ze-101 8.276285296750369e-104 -5.058081159205726e-106 3.012256769750308e-108 -1.748629262452811Ae-110 9.897901485581951e-113 -5.46467
9909223989e-115 2.943715049110633Ze-117 -1.547626750901537Ge-119
```

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.