Ex. 6 **EXPLORING USER-DEFINED FUNCTIONS**
Date: 26 February 2024

Aim:

To explore user-defined functions in Python by writing programs for the following:

a. Write a function that can generate Pythagorean triplets up to N. Obtain N from the user and pass it as an argument to your function.
b. Write a function to generate the sum of the sine series, sin (x) and another to generate the sum of the cosine series cos (x) up to N terms. These functions must have two arguments, x and N, where N is an optional argument. Here x is the angle in radians. Get the angle in degrees from the user and convert to radians. Pass this angle to the functions. The computation of the sum of the sine series also requires the computation of the factorial of a number, so write a function to compute factorial as well.
c. Write a menu-driven program to perform the following tasks. Write a separate function for each task.
    - Check if a given integer is an Armstrong number.
    - Check if a given integer is a Deficient number.
    - Check if a given integer is a Palindrome.
    - Exit the program.

Algorithm:

(a)

Step – 1: Get the ending condition (N) form the user.

Step – 2: With m as 2 and n as 1 start calculating the value of a,b and c using the equations.

Step – 3: Increment m until the a,b and c are always larger than 0 but lesser than N.

Step – 4: If m reaches its end condition, increment n and start m from 2.

Step – 5: Run through the loop antil incrementing n leads to values greater than N.

(b)

Step – 1: Get the value of x and N (number of terms) from the user.

Step – 2: Find the factorial of a number by recursively multiplying 1 less than the original number until 0.

Step – 3: To find the sin_Sum, get radians and number of terms as parameters and using a simple recursive equation calculate the sum and stop the recursive function call when the number of terms is met.

Step – 4: To find the cos_Sum, get radians and numbers of terms as parameters and using a simple recursive equation calculate the sum and stop the recursive function call when the number of terms is met.

(c)

Step – 1: Start a loop and ask the user for option to run a function.

Step – 2: Check Armstrong number by summing up the digits to the power of the number of digits.

Step – 3: Check for Deficient by summing up the factors of number and verify if it is smaller than the original number itself.

Step – 4: Check for palindrome by reversing the string of the number and comparing to the original string of the number.

Program:

(a)

```
def PTriplets(max):
    a = b = c = 0
    m = n = 1

    while(True):
        while(True):
            m+=1
            a = m**2 - n**2
            b = 2*m*n
            c = m**2 + n**2
            if(a >max or b >max or c>max):
                break
            if(a<=0 or b<=0 or c<=0):
                continue
            print(a,b,c)
        m = 2
        n+=1
        a = m**2 - n**2
        b = 2*m*n
        c = m**2 + n**2
        if(a >max or b >max or c>max):
            break
        if(a<=0 or b<=0 or c<=0):
            continue
        print(a,b,c)



N = int(input("Enter max : "))
PTriplets(N)
```

(b)

```
import math
def Fact(N):
    if(N<=1):
        return 1
    return N*Fact(N-1)

def SinSum(rad,terms,c = 0):
    if(terms == c):
        return 0
    else:
                    return math.pow(rad,(c*2+1))/ Fact(c*2+1) + math.pow(-1,c+1) *
(SinSum(rad,terms,c+1))

def CosSum(rad,terms,c=0):
    if(terms == c):
        return 0
    else:
        return math.pow(rad,(c*2))/ Fact(c*2) + math.pow(-1,c+1) * (CosSum(rad,terms,c+1))


N = int(input("Enter the number of terms : "))
deg = float(input("Enter the in degrees : "))
radians = deg*(3.14/180)
print(SinSum(radians,N))
print(CosSum(radians,N))
```

(c)

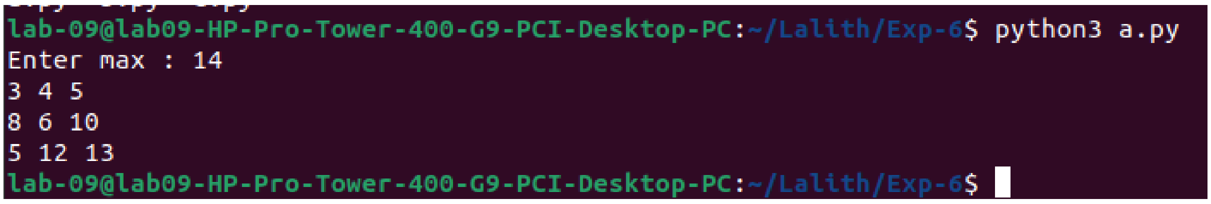
```
def Armstrong(num):
    if(num == sum([int(x)**len(str(num)) for x in str(num)])):
        return True
    return False

def Deficient(num):
    if(num > sum([int(x) for x in range(1,num) if(num%int(x)==0)] )):
        return True
    return False
def Palindrome(num):
    if(str(num) == str(num)[::-1]):
        return True
    return False
```

```
while True:
    c = int(input("\nEnter the option:\n\t1-Armstrong\n\t2-Deficient\n\t3-Palindrome\n\t4-Exit\n>>> "))
    if(0<c<4):N = int(input("\nEnter the number : "))
    if(c == 1):
        if(Armstrong(N)):
            print("The given number is an armstrong number")
        else:
            print("The number is not an armstrong number")
    elif(c== 2):
        if(Deficient(N)):
            print("the number is a deficient number")
        else:
            print("The Number is not deficient")
    elif(c==3):
        if(Palindrome(N)):
            print("The number is a palindrome")
        else:
            print("The number is not a palindrome")
    elif(c==4):
        print("Exiting program")
        break
    else:
        print("Wrong Option")
```

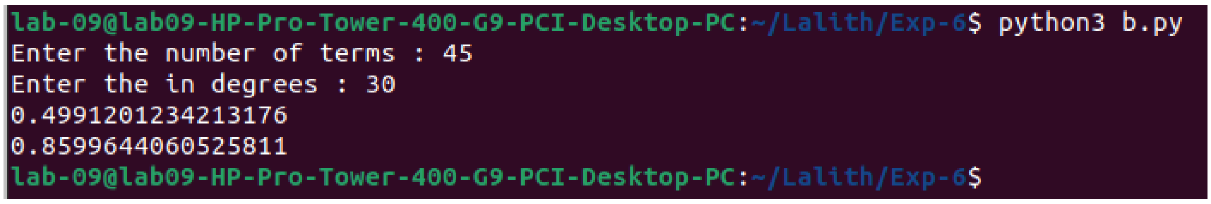
Screenshot of Output:

(a)



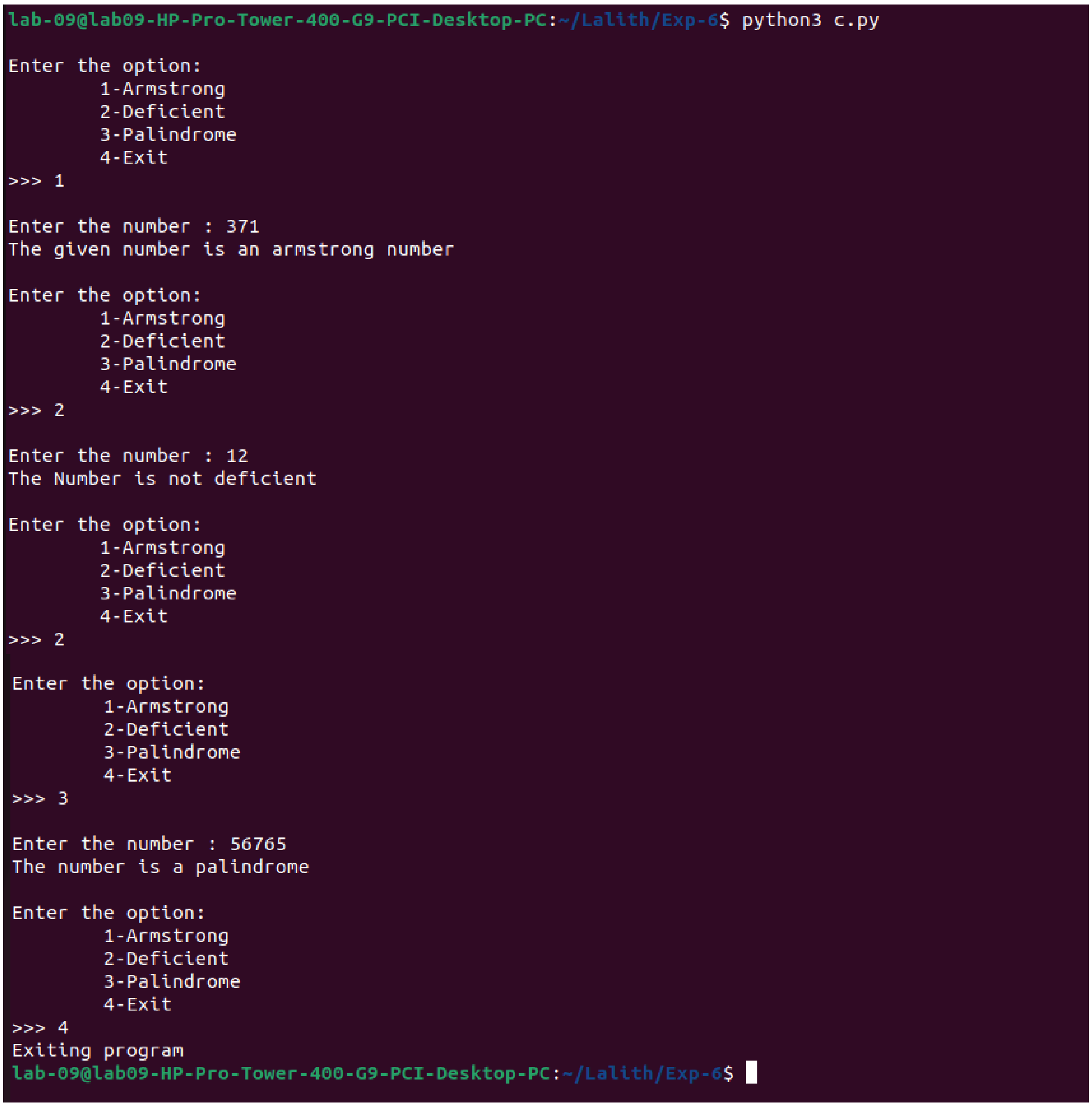
```
lab-09@lab09-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~/Lalith/Exp-6$ python3 a.py
Enter max : 14
3 4 5
8 6 10
5 12 13
lab-09@lab09-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~/Lalith/Exp-6$
```

(b)



```
lab-09@lab09-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~/Lalith/Exp-6$ python3 b.py
Enter the number of terms : 45
Enter the in degrees : 30
0.4991201234213176
0.8599644060525811
lab-09@lab09-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~/Lalith/Exp-6$
```

(c)



```
lab-09@lab09-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~/Lalith/Exp-6$ python3 c.py

Enter the option:
        1-Armstrong
        2-Deficient
        3-Palindrome
        4-Exit
>>> 1

Enter the number : 371
The given number is an armstrong number

Enter the option:
        1-Armstrong
        2-Deficient
        3-Palindrome
        4-Exit
>>> 2

Enter the number : 12
The Number is not deficient

Enter the option:
        1-Armstrong
        2-Deficient
        3-Palindrome
        4-Exit
>>> 2

Enter the option:
        1-Armstrong
        2-Deficient
        3-Palindrome
        4-Exit
>>> 3

Enter the number : 56765
The number is a palindrome

Enter the option:
        1-Armstrong
        2-Deficient
        3-Palindrome
        4-Exit
>>> 4
Exiting program
lab-09@lab09-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~/Lalith/Exp-6$ 
```

Result:

Thus, programs have been written and executed to explore user-defined functions in Python.