CYCLE 3

1. Program to find the factorial of a number

num = int(input())

factorial = 1

if num < 0:

     print("Sorry, factorial does not exist for negative numbers")

elif num == 0:

    print("The factorial of 0 is 1")

else:

     for i in range(1,num + 1):

      factorial = factorial\*i

    print("The factorial of",num,"is",factorial)

**OUTPUT:**

3

The factorial of 3 is 6

2. Generate Fibonacci series of N terms

nterms = int(input("How many terms? "))

n1, n2 = 0, 1

count = 0

if nterms <= 0:

    print("Please enter a positive integer")

elif nterms == 1:

    print("Fibonacci sequence upto",nterms,":")

    print(n1)

else:

    print("Fibonacci sequence:")

    while count < nterms:

      print(n1)

      nth = n1 + n2

      n1 = n2

      n2 = nth

      count += 1

**OUTPUT:**

How many terms? 8

Fibonacci sequence:

0

1

1

2

3

5

8

13

3. Find the sum of all items in a list.

total = 0

list1 = [11, 5, 17, 18, 23]

for ele in range(0, len(list1)):

     total = total + list1[ele]

  print("Sum of all elements in given list: ", total)

**OUTPUT:**

Sum of all elements in given list: 74

4. Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.

def perfectSquares(l, r):

     for i in range(l, r + 1):

      if (i\*\*(.5) == int(i\*\*(.5))):

      print(i, end=" ")

l = 2

r = 24

perfectSquares(l, r)

**OUTPUT:**

4 9 16

5. Display the given pyramid with step number accepted from user. Eg: N=4 1 2 4 3 6 9 4 8 12 16

rows = 6

for num in range(rows):

     for i in range(num):

      print(num, end=" ")

     print(" ")

**OUTPUT:**

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

6. Count the number of characters (character frequency) in a string.

test\_str = " sandra"

  all\_freq = {}

for i in test\_str:

     if i in all\_freq:

      all\_freq[i] += 1

     else:

     all\_freq[i] = 1

print ("Count of all characters in sandra is :\n "+  str(all\_freq))

**OUTPUT:**

Count of all characters in sandra is :

{'s': 1, 'a': 2, 'n': 1, 'd': 1, 'r': 1}

7. Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’

def add\_string(str1):

   length = len(str1)

   if length > 2:

     if str1[-3:] == 'ing':

     str1 += 'ly'

    else:

      str1 += 'ing'

   return str1

print(add\_string('ab'))

print(add\_string('abc'))

print(add\_string('string'))

**OUTPUT:**

ab

abcing

stringly

8. Accept a list of words and return length of longest word.

def find\_longest\_word(words\_list):

    word\_len = []

     for n in words\_list:

      word\_len.append((len(n), n))

     word\_len.sort()

     return word\_len[-1][0], word\_len[-1][1]

result = find\_longest\_word(["PHP", "Exercises", "Backend"])

print("\nLongest word: ",result[1])

print("Length of the longest word: ",result[0])

**OUTPUT:**

Longest word: Exercises

Length of the longest word: 9

9. Construct following pattern using nested loop \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

n=5;

for i in range(n):

     for j in range(i):

      print ('\* ', end="")

     print('')

for i in range(n,0,-1):

     for j in range(i):

      print('\* ', end="")

     print('')

**OUTPUT:**

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

10. Generate all factors of a number.

def print\_factors(x):

    print("The factors of",x,"are:")

    for i in range(1, x + 1):

      if x % i == 0:

      print(i)

num = 300

print\_factors(num)

**OUTPUT:**

The factors of 300 are:

1

2

3

4

5

6

10

12

15

20

25

30

50

60

75

100

150

300

11. Write lambda functions to find area of square, rectangle and triangle

import math

t\_peri = lambda p,q,r : p + q + r

r\_area = lambda len, ht : len\*ht

s\_area = lambda a : 2\*a

print("Perimeter of Triangle (10,20,15) is:", t\_peri(10,20,15))

print("Area of Rectangle (30,20) is:", r\_area(30,20))

print("Area of square (5) is:", s\_area(5))

**OUTPUT:**

Perimeter of Triangle (10,20,15) is: 45

Area of Rectangle (30,20) is: 600

Area of square (5) is: 10