

# BHARATI VIDYAPEETH’S

**INSTITUTE OF COMPUTER APPLICATIONS & MANAGEMENT**

(Affiliated to Guru Gobind Singh Indraprastha University, Approved by AICTE, New Delhi)

**Python Programming (MCA-106)**

Practical File

# Submitted To: Submitted By:

Dr. Rakhee Abhijeet Rana

01811604422

(Assistant Professor) MCA 22nd Sem, Sec 1

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Problem Description** | **Date of execution** | **Date of submission** |
| 1. | Implement Python Script to generate first N  natural numbers. |  |  |
| 2. | By considering the terms in the Fibonacci  sequence whose values do not exceed 1000,  find the sum of the even-valued terms |  |  |
| 3. | Write a program which makes use of function  to display all such numbers which are divisible  by 7 but are not a multiple of 5, between 1000  and 2000. |  |  |
| 4. | Write a function cumulative product to  compute cumulative product of a list of  numbers. |  |  |
| 5. | Write a function reverse to reverse a list.  Without using the reverse function. |  |  |
| 6. | Define a function which generates Fibonacci series up to n numbers using RECURSION. |  |  |
| 7. | With a given tuple (1, 2, 3, 4, 5, 6, 7, 8, 9, 10), write a program to print the first half values in one line and the last half values in one line. |  |  |
| 8. | Write a program to count the numbers of characters in the string and store them in a dictionary data structure. |  |  |
| 9. | Remove spaces from a string using recursion. |  |  |
| 10. | Write a program to compute the number of characters, words and lines in a file. |  |  |

1. Implement Python Script to generate first N natural numbers.

Code :

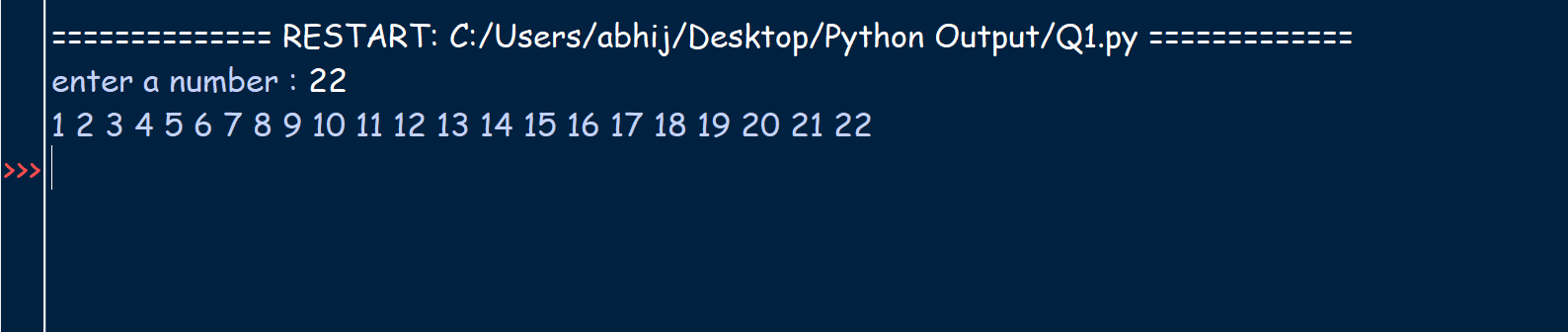
def generateNaturalNumbers (n) :

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

print(i, end=" ")

x = int(input("enter a number : "))

generateNaturalNumbers(x)



1. By considering the terms in the Fibonacci sequence whose values do not exceed 1000, find the sum of the even-valued terms.

Code :

def fibSum():

first\_term = 0

second\_term = 1

result = 0

while second\_term <= 1000 :

if second\_term % 2 == 0 :

result += second\_term

temp = second\_term

second\_term += first\_term

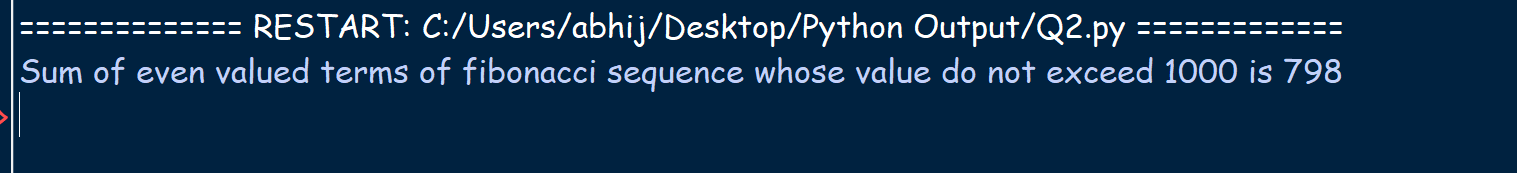
first\_term = temp

return result

x = fibSum()

print( f'Sum of even valued terms of fibonacci sequence whose value do not exceed 1000 is {x}')

Output :



1. Write a program which makes use of function to display all such numbers which are divisible by 7 but are not a multiple of 5, between 1000 and 2000.

Code :

def func ():

x = 1001

while ( x % 7 != 0) :

x += 1

for i in range (x, 2000, 7) :

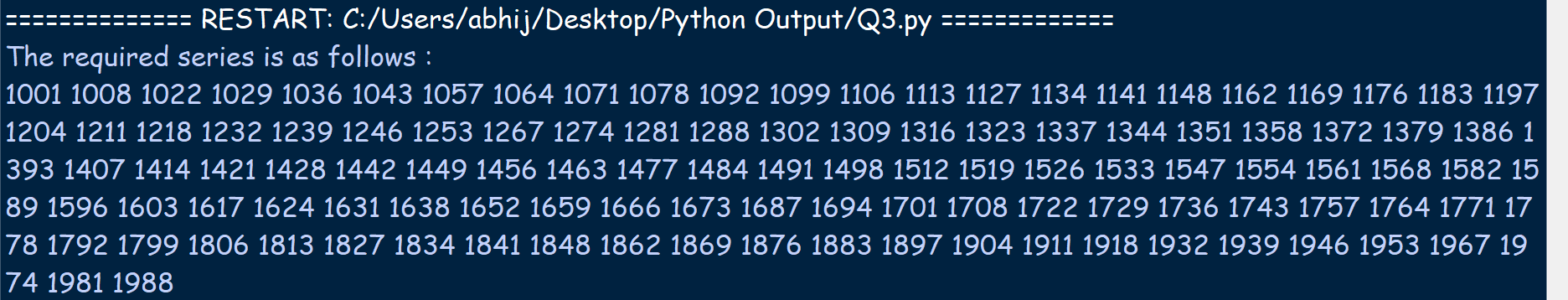
if(i % 5 != 0 ) :

print( i, end = " ")

print("The required series is as follows : ")

func()

Output :



1. Write a function cumulative\_product to compute cumulative product of a list of numbers.

Code :

def cummulative\_product ( list ):

res = 1

for ele in list :

res \*= ele

return res

length\_of\_arr = int(input('Enter the length of the array : '))

print('Enter the elements of the array')

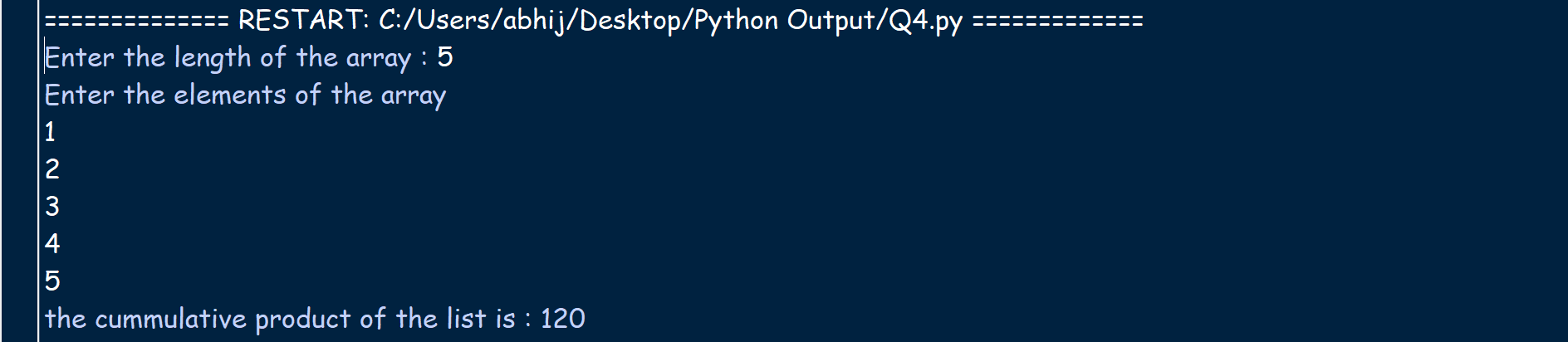
list = list()

for i in range (0, length\_of\_arr) :

list.append(int(input()))

print(f'the cummulative product of the list is : {cummulative\_product(list)}')

Output :



1. Write a function reverse to reverse a list. Without using the reverse function.

Code :

def reverseList(list) :

n = len(list)

mid = (n-1)//2

for i in range (0, mid) :

temp = list[i]

list[i] = list[n-1-i ]

list[ n-1-i ] = temp

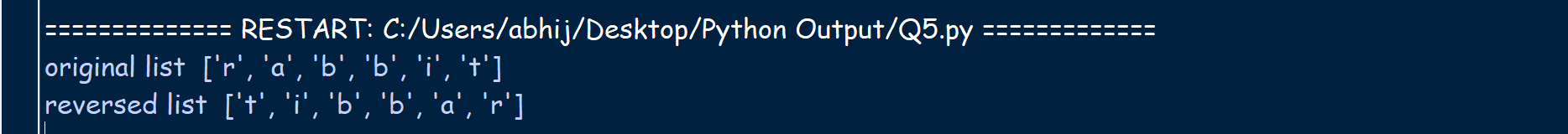
list= ['r', 'a', 'b', 'b', 'i', 't']

print( 'original list ', list )

reverseList(list)

print( 'reversed list ', list )

Output :



6. Define a function which generates Fibonacci series up to n numbers using RECURSION

Code :

def fib (n) :

    if  n == 0 :

        return 0

    if  n == 1 or n == 2  :

        return 1

    return fib(n-1) + fib(n-2)

def printFib(n):

    for i in range(0 ,n) :

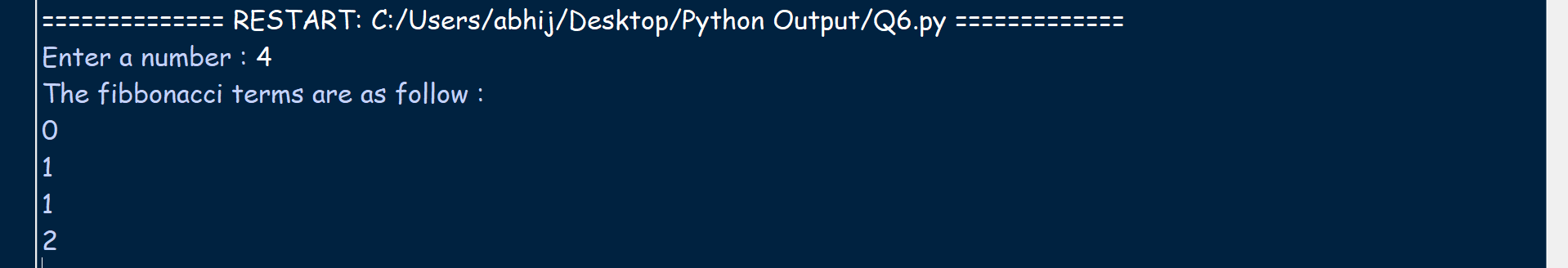
        print(fib(i))

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

print("The fibbonacci terms are as follow : ")

printFib(x)

Output:



7. With a given tuple (1, 2, 3, 4, 5, 6, 7, 8, 9, 10), write a program to print the first half values in one line and the last half values in one line.

Code :

def printTuple (t1 ):

    mid = len(t1)//2

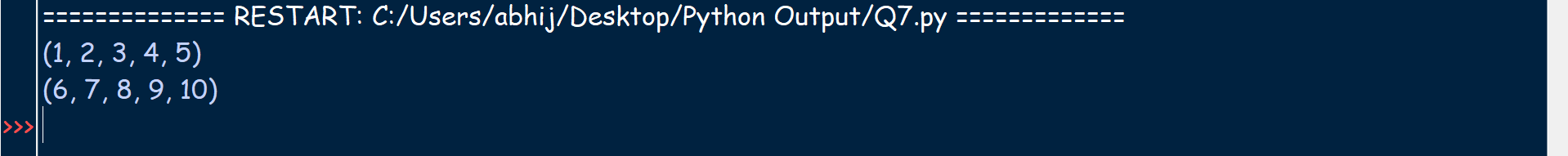
    print(t1[:mid])

    print(t1[mid:])

t1 = (1, 2,3, 4, 5, 6, 7, 8, 9, 10)

printTuple(t1)

Output:



8. Write a program to count the numbers of characters in the string and store them in a dictionary data structure

Code :

def createMap(str) :

    map = dict()

    for char in str:

        if char in map.keys() :

            map[char] = map[char] + 1

        else:

            map[char] = 1

    return map

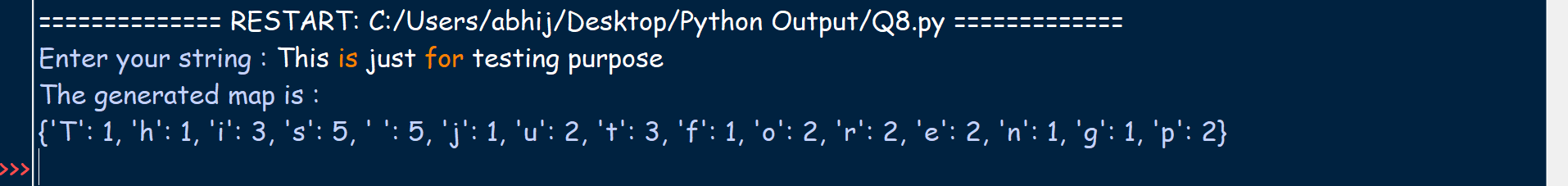
str = input("Enter your string : ")

print("The generated map is : ")

map = createMap(str)

print(map)

Output :



9. Remove spaces from string using recursion.

Code :

def stripSpaces( str ):

    if len(str ) == 0 : return ""

    if str[0] == " " :

        return stripSpaces ( str[1:] )

    else:

        return str[0:1] + stripSpaces(str[1:])

str = input("Enter a string : ")

print("String after striping spaces")

print(stripSpaces(str))

Ouptut :

