Python Assignment for kuber verma

Q1 - Write a decorator function for your taking input for you any kind of function you want to build , For example- you make a Fibonacci series function, in which your input range is been defined by the Decorator programs input,

Answer-

Fibonacci Series using Loop

Loops in Python allow us to execute a group of statements several times. Let's write a python program to implement Fibonacci Series using a loop.

```
1
2  # Enter number of terms needed                    #0,1,1,2,3,5....
3  a=int(input("Enter the terms"))
4  f=0                    #first element of series
5  s=1                    #second element of series
6  if a<=0:
7      print("The requested series is
   ",f)
8  else:
9      print(f,s,end=" ")
10     for x in range(2,a):
11         next=f+s
12         print(next,end=" ")
13         f=s
14         s=next
```

Output: Enter the terms 5 0 1 1 2 3

Another way to program the Fibonacci series generation is by using recursion. Let's dig deeper into it.

Python Program to Write Fibonacci Sequence Using Recursion

Recursion is the basic Python programming technique in which a function calls itself directly or indirectly. The corresponding function is called a **recursive function**. Using a recursive algorithm, certain problems can be solved quite easily. Let's see how to use recursion to print first 'n' numbers of the Fibonacci Series in Python.

Python Code:

```
def FibRecursion(n):
    if n <= 1:
        return n
    else:
        return(FibRecursion(n-1) + FibRecursion(n-2))
nterms = int(input("Enter the terms? "))  # take input from the user

if nterms <= 0:  # check if the number is valid
    print("Please enter a positive integer")
else:
    print("Fibonacci sequence:")
    for i in range(nterms):
        print(FibRecursion(i))
```

Output: How many terms: 5
0 1 1 2 3

Q 2 --MAKE a small generator program for returning Armstrong numbers in between 1-1000

In a generator object.

Answer-

Any three digit number is called an Armstrong number of sum of cube of its digits equals the number itself. In order to check if a number satisfies this condition, each digit from it is successively separated from right and its cube is cumulatively added. In the end if the sum is found to be equal to original number, it is called Armstrong number.

Example

Following Python code prints all armstrong numbers between 100 to 999

```
for num in range(100,1000):
  temp=num
  sum=0
  while temp>0:
    digit=temp%10
    sum=sum+digit**3
    temp=temp//10

  if sum==num:
    print (num)
```

Output

The output is list of armstrong numbers

153
370
371
407