Write a program non-recursive and recursive program to calculate Fibonacci numbers and analyze their time and space complexity.

This is a Python Program to find the fibonacci series without using recursion.

**Problem Description**

The program takes the first two numbers of the series along with the number of terms needed and prints the fibonacci series.

**Problem Solution**

1. Take the first two numbers of the series and the number of terms to be printed from the user.  
2. Print the first two numbers.  
3. Use a while loop to find the sum of the first two numbers and then proceed the fibonacci series.  
4. Print the fibonacci series till n-2 is greater than 0.  
5. Exit.

**Program/Source Code**

Here is source code of the Python Program to find the fibonacci series without using recursion. The program output is also shown below.

a=int(input("Enter the first number of the series "))

b=int(input("Enter the second number of the series "))

n=int(input("Enter the number of terms needed "))

**print**(a,b,end=" ")

**while**(n-2):

c=a+b

a=b

b=c

**print**(c,end=" ")

n=n-1

**Program Explanation**

1. User must enter the first two numbers of the series and the number of terms to be printed.  
2. The first two terms are printed outside the while loop.  
3. A while loop is used to find the sum of the first two terms and proceed the series by interchanging the variables.  
4. The value of n is decremented.  
5. The fibonacci series is printed till n-2 is greater than 0.

**Note: Join free Sanfoundry classes at**[**Telegram**](https://t.me/sanfoundryclasses/)**or [Youtube](https://www.youtube.com/c/SanfoundryOfficial/?sub_confirmation=1" \t "_blank)**

**Runtime Test Cases**

Case 1:

Enter the first number of the series 0

Enter the second number of the series 1

Enter the number of terms needed 4

0 1 1 2

Case 2:

Enter the first number of the series 2

Enter the second number of the series 4

Enter the number of terms needed 5

2 4 6 10 16

When it is required to find the Fibonacci series without using recursion technique, the input is taken from the user, and a ‘while’ loop is used to get the numbers in the sequence.

## **Example**

Below is a demonstration for the same −

first\_num = int(input("Enter the first number of the fibonacci series... "))

second\_num = int(input("Enter the second number of the fibonacci series... "))

num\_of\_terms = int(input("Enter the number of terms... "))

print(first\_num,second\_num)

print("The numbers in fibonacci series are : ")

while(num\_of\_terms-2):

   third\_num = first\_num + second\_num

   first\_num=second\_num

   second\_num=third\_num

   print(third\_num)

   num\_of\_terms=num\_of\_terms-1

## **Output**

Enter the first number of the fibonacci series... 2

Enter the second number of the fibonacci series... 8

Enter the number of terms... 8

2 8

The numbers in fibonacci series are :

10

18

28

46

74

120

## **Explanation**

* The first number and second number inputs are taken from the user.
* The number of terms is also taken from the user.
* The first and the second numbers are printed on the console.
* A while loop begins, and the below takes place −
* The first and second numbers are added and assigned to a third number.
* The second number is assigned to the third number.
* The third number is assigned to the second number.
* The third number is printed on the console.
* The number of terms is decremented by 1.

|  |
| --- |
| # Fibonacci Series using Dynamic Programming  **def** fibonacci(n):        # Taking 1st two fibonacci numbers as 0 and 1      f **=** [0, 1]      **for** i **in** range(2, n**+**1):          f.append(f[i**-**1] **+** f[i**-**2])  **return** f[n]    print(fibonacci(9)) |

**Output**

34

**Time complexity**: O(n) for given n

**Auxiliary space**: O(n)