

Mock Test > doliatius@protonmail.com

Full Name: Doli

Email: doliatius@protonmail.com

Test Name: Mock Test

 Taken On:
 20 Jan 2024 19:52:52 IST

 Time Taken:
 2 min 55 sec/ 20 min

Linkedin: https://www.linkedin.com/in/musab-oguz-

68990a200/

Invited by: Ankush

Invited on: 20 Jan 2024 19:52:44 IST

Skills Score:

Tags Score: Algorithms 120/120

Core CS 120/120

Dynamic Programming 120/120

Medium 120/120

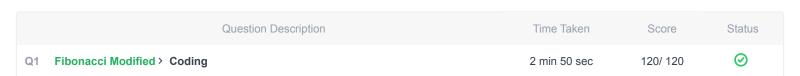
problem-solving 120/120

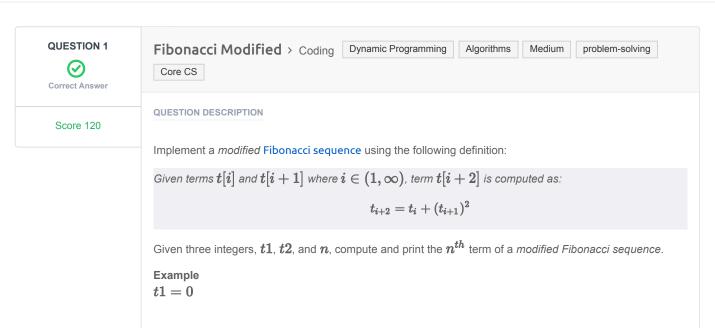


scored in **Mock Test** in 2 min 55 sec on 20 Jan 2024 19:52:52

Recruiter/Team Comments:

No Comments.





$$t2 = 1$$

$$n = 6$$

$$t3 = 0 + 1^{2} = 1$$

$$t4 = 1 + 1^{2} = 2$$

$$t5 = 1 + 2^{2} = 5$$

$$t6 = 2 + 5^{2} = 27$$

Return 27.

Function Description

Complete the $\it fibonacci Modified$ function in the editor below. It must return the $\it n^{th}$ number in the sequence.

fibonacciModified has the following parameter(s):

- int t1: an integer
- int t2: an integer
- int n: the iteration to report

Returns

ullet int: the n^{th} number in the sequence

Note: The value of t[n] may far exceed the range of a 64-bit integer. Many submission languages have libraries that can handle such large results but, for those that don't (e.g., C++), you will need to compensate for the size of the result.

Input Format

A single line of three space-separated integers, the values of t1, t2, and n.

Constraints

- $0 \le t1, t2 \le 2$
- $3 \le n \le 20$
- $oldsymbol{\cdot}$ t_n may far exceed the range of a 64-bit integer.

Sample Input

```
0 1 5
```

Sample Output

5

Explanation

The first two terms of the sequence are t1=0 and t2=1, which gives us a modified Fibonacci sequence of $\{0,1,1,2,5,27,\ldots\}$. The 5^{th} term is 5.

CANDIDATE ANSWER

Language used: Python 3

```
1
2 #
3 # Complete the 'fibonacciModified' function below.
4 #
5 # The function is expected to return an INTEGER.
6 # The function accepts following parameters:
7 # 1. INTEGER t1
8 # 2. INTEGER t2
9 # 3. INTEGER n
10 #
11
12 def fibonacciModified(t1, t2, n):
```

```
sys.set_int_max_str_digits(0)
14
       while n>2:
           sum = 0
            sum = t1 + pow(t2, 2)
            t1 = t2
            t2 = sum
            n -= 1
       return sum
   TESTCASE
               DIFFICULTY
                               TYPE
                                           STATUS
                                                      SCORE
                                                               TIME TAKEN
                                                                             MEMORY USED
                             Sample case
                                          Success
                                                         0
                                                                0.0196 sec
                                                                                 10.4 KB
  Testcase 1
                  Easy
  Testcase 2
                  Easy
                             Sample case
                                         Success
                                                        0
                                                                0.1174 sec
                                                                                 10.5 KB
  Testcase 3
                  Easy
                             Hidden case
                                                        15
                                                                0.0869 sec
                                                                                 10.7 KB
                                          Success
                                          Success
                                                                0.2499 sec
                                                                                 10.9 KB
  Testcase 4
                  Easy
                             Hidden case
                                                        15
   Testcase 5
                  Easy
                             Hidden case
                                          Success
                                                        15
                                                                0.0222 sec
                                                                                 10.7 KB
  Testcase 6
                  Easy
                             Hidden case
                                          Success
                                                        15
                                                                0.0199 sec
                                                                                 10.8 KB
   Testcase 7
                  Easy
                             Hidden case
                                          Success
                                                        15
                                                                0.2943 sec
                                                                                 10.8 KB
                                                                0.0222 sec
                                                                                 10.7 KB
  Testcase 8
                  Easy
                             Hidden case
                                          Success
                                                        15
```

Success

Success

15

15

 $0.0203 \, \text{sec}$

0.019 sec

10.8 KB

10.5 KB

No Comments

Testcase 9

Testcase 10

Easy

Easy

PDF generated at: 20 Jan 2024 14:27:42 UTC

Hidden case

Hidden case