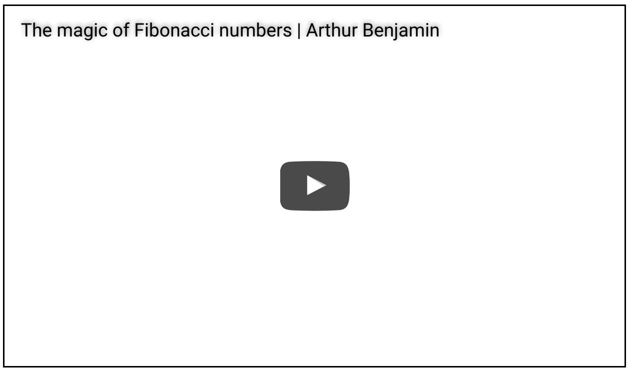
Fibonacci Numbers

by PRASHANTB1984

Objective
In this challenge, we learn about using the Fibonacci Function.

Resources
Here's a helpful video on the topic:



The Fibonacci Series

The Fibonacci sequence begins with 0 and 1. These are the first and second terms, respectively. After this, every element is the sum of the preceding elements:

```
Fibonacci(n) = Fibonacci(n-1) + Fibonacci(n-2)
```

Task
Given the starter code, complete the Fibonacci function to return the N^{th} term.

We start counting from $\mathit{Fibonacci}(1) = 0$. This might differ from some other notations that treats $\mathit{Fibonacci}(0) = 0$.

The overall equation is:

```
             = 0 , n = 1
Fibonacci(n) = 1 , n = 2
               Fibonacci(n-1) + Fibonacci(n-2)  , n > 2
```

Input Format

One line of input, the integer N.

Constraints

$0 < N <= 40$

Output Format

Output one integer, the N^{th} Fibonacci number.

Sample Input

```
3
```

Sample Output

```
1
```

Function Prototype
The starter code is provided for Scala. The code for accepting the input and displaying the output is provided. You will be provided the input parameter N, and you

need to return the N^{th} Fibonacci term.

Sample Input and Output Values for the Fibonacci Series

```
fibonacci(3) = (0+1) = 1
fibonacci(4) = (1+1) = 2
fibonacci(5) = (1+2) = 3
```

Requirements
Simple test cases can be cleared with a purely recursive function exponentially. To clear the more challenging test cases without violating the principles of functional programming, you might benefit from learning about the accumulator technique.

Submissions: 6978
Max Score: 2
Difficulty: Easy

Rate This Challenge:
☆☆☆☆☆

More

If you cannot find your favorite language here, please check out the algorithm domain. We support over 40 popular languages.

Current Buffer (saved locally, editable)

Racket

```
#lang racket
; Enter your code here. Read input from STDIN. Print output to STDOUT

(define (fib n)
  (fib-iter 1 0 0 1 n))
(define (fib-iter a b p q count)
  (cond ((= count 0) b)
        ((even? count)
         (fib-iter a
                   b
;                   <??> ; compute p’
;                   <??> ; compute q’
                   (+ (* q q) (* p p)) ; compute p'
                   (+ (* 2 p q) (* q q)) ; compute q’
                   (/ count 2)))
                   (else (fib-iter (+ (* b q) (* a q) (* a p))
                                   (+ (* b p) (* a q))
                                   p
                                   q
                                   (- count 1)))))

(displayln (fib (- (read) 1)))
```

Line: 1 Col: 1

Upload Code as File ☐ Test against custom input

Run Code Submit Code

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature