Recursion → Alternating Fibonacci numbers

1369 users solved this problem. Latest completion was about 7 hours ago.

Medium 9 minutes

Given the small integer **n (0 <= n <= 40)** you need to find the **n-th number** of the **alternating Fibonacci sequence.**

The sequence starts with `0, 1, -1, 2, -3, 5, -8, 13, -21, ...`

```
So, fib(0) = 0, fib(1) = 1 => fib(2) = -1.
```

Think of the recurrence relation and implement the method named `fib` in a recursive way. It's not efficient in the general but works well for small **n**.

Report a typo

Sample Input 1:

```
2
```

Sample Output 1:

```
-1
```

Sample Input 2:

```
3
```

Sample Output 2:

```
2
```

Write a program

Code Editor IDE

Java

```
import java.util.Scanner;

public class Main {

    public static long fib(long n) {
        // write your code here
        if (n <= 1) {
            return n;
        } else {
            return fib(n - 2) - fib(n - 1);
        }
    }

    /* Do not change code below */
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        System.out.println(fib(n));
    }
}

```

✓ Correct.

That's an awesome solution! What do you think about showing it off? Post it to Solutions so other learners can enjoy it too.

107 users liked this problem. **6** didn't like it. **What about you?**

Continue Solve again Solutions (66)

Time limit: 2 seconds Memory limit: 256 MB

Comments (11) Hints (8) Useful links (0) Solutions (66) Show discussion