Java → Basic syntax and simple programs → Methods → Recursion

Recursion → N-th power

Medium 11 minutes ?

1488 users solved this problem. Latest completion was about 14 hours ago.

It is possible to find a **n-th** power much quicker than by making **n** multiplications!

To do this you need to use the following recurrence relations:

$a^n = (a^2)^{n/2}$ for even n,

$a^n = a * a^{n-1}$ for odd n.

Implement the algorithm of quick exponentiating using a recursion method.

Report a typo

Sample Input 1:

```
2.0
1
```

Sample Output 1:

```
2.0
```

Sample Input 2:

```
1.5
10
```

Sample Output 2:

```
57.665
```

Write a program

Code Editor IDE

Java

```java
import java.util.Scanner;

public class Main {

    public static double pow(double a, long n) {
        // write your code here
        if (n == 1) {
            return a;
        } else if (n == 0) {
            return 1;
        } else {
            if (n % 2 == 0) {
                return pow(a * a, n / 2);
            } else {
                return a * pow(a, n - 1);
            }
        }
    }

    /* Do not change code below */
    public static void main(String[] args) {
        final Scanner scanner = new Scanner(System.in);
        final double a = Double.parseDouble(scanner.nextLine());
        final int n = Integer.parseInt(scanner.nextLine());
        System.out.println(pow(a, n));
        scanner.close();
    }
}

```

✓ 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.

113 users liked this problem. 4 didn't like it. **What about you?**

😍 🙂 😐 🙁 😡

Continue Solve again Solutions (70)

Time limit: 1 seconds Memory limit: 256 MB

Comments (3) Hints (12) Useful links (1) Solutions (70) Show discussion