https://course.acciojob.com/idle?question=f1316d07-891c-4d64-99d9-fccb4a8f940c

- EASY
- Max Score: 30 Points
-
-

Optimized power calculation

You are given two integers, `N` and `X`. You have to find `X` raised to the power `N` in `O(log(N))` time complexity using `recursion`.

Note Complete the given function and use recursion to solve this problem.

It is guaranteed that the test cases are made such that the answer does not overflow a 64-bit integer datatype.

Input Format

The first line contains two integers, `X` and `N`.

Output Format

Print the answer.

Example 1

Input

```
2 3
```

Output

```
8
```

Explanation

`2` raised to the power `3` is `8`.

Example 2

Input

```
10 5
```

Output

```
100000
```

Explanation

`10` raised to the power `5` is `100000`.

Constraints

1 <= X <= 10^8

1 <= N <= 64

Topic Tags

- **Recursion**

My code

```
// n java
import java.io.*;
import java.util.*;

class Main {
```

```
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

       int X,N;
        X = sc.nextInt();
        N = sc.nextInt();

        System.out.println(power(X,N));
    }

        public static long power(int x, int n)
    {
                //Write code here
                long ans=1;
                for(int i=0;i<n;i++)
                        ans=ans*x;
                return ans;
    }
}
```