

## 1204: 爬楼梯

### 题目描述

树老师爬楼梯，他可以每次走1级或者2级，输入楼梯的级数，求不同的走法数。

例如：楼梯一共有3级，他可以每次都走一级，或者第一次走一级，第二次走两级，也可以第一次走两级，第二次走一级，一共3种方法。

### 输入

输入包含若干行，每行包含一个正整数N，代表楼梯级数， $1 \leq N \leq 30$ 。

### 输出

不同的走法数，每一行输入对应一行输出。

### 输入样例

```
5
8
10
```

### 输出样例

```
8
34
89
```

### 分析

解决递归问题，需要你具备分工思想，我们换一个角度来考虑这个问题。

首先一共只有两种上楼梯的方法：

- 1、一次走一步
- 2、一次走两步

假设我现在要求到达第四阶楼梯的时候，我们的最后一步可能来自哪里呢？很容易想到，我们最后一步可能来自于第三阶（最后一次走一步），还可能来自于第二阶（最后一次走两步）。那么，问题是不是可以转化为：

到达第四阶楼梯的方法数 = 到达第三阶楼梯的方法数 + 到达第二阶楼梯的方法数

同理，继续推导：

到达第三阶楼梯的方法数 = 到达第二阶楼梯的方法数 + 到达第一阶楼梯的方法数

到达第二阶楼梯的方法数 = 到达第一阶楼梯的方法数 + 到达第零阶楼梯的方法数

最终我们可以得出：

到达第N阶楼梯的方法数 = 到达第N-1阶楼梯的方法数 + 到达第N-2阶楼梯的方法数

当然，到达第一阶楼梯和到达第零阶楼梯的数量很显然都是1。所以，最终爬楼梯的伪代码应该是这样的：

```
方法数(N)
{
    if(N == 1 || N == 0)
        return 1;
    return 方法数(N-1) + 方法数(N-2);
}
```

编码

```
#include<bits/stdc++.h>

using namespace std;

//计算爬楼梯的方法数
int calculate(int n) {
    //第1层和第0层的方法数均为1
    if (n == 1 || n == 0)
        return 1;
    return calculate(n - 1) + calculate(n - 2);
}

int main() {
    int n;
    while (cin >> n) {
        cout << calculate(n) << endl;
    }
    return 0;
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

