

## P1255 数楼梯

### 题目描述

楼梯有  $N$  阶，上楼可以一步上一阶，也可以一步上二阶。

编一个程序，计算共有多少种不同的走法。

### 输入格式

一个数字，楼梯数。

### 输出格式

输出走的方式总数。

### 输入样例

4

### 输出样例

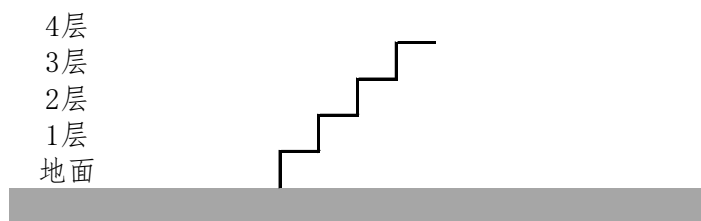
5

### 解析

本题存在两个问题：

- 1、楼梯方案数的计算
- 2、高精度

对于第一个问题，我们仔细观察下图。



想到达1层，一共只有1种途径，即从地面走到1层。

想到达2层，则有两种途径：第一种是直接从地面走到2层，另外一种则是从1层走上来。

想到达3层，我们也有两种途径，第一种是从1层跨越两层到达，另外一种是从2层跨越一层到达。那么到达1层有多少种方法呢？很明显是1种，那么到达2层有多少种方法呢？从上面可以看出一共是2种。

因此，到达3层的总方法数就是到达1层的总方法数加上到达2层的总方法数，答案为3。

继续推导，到达4层的总方案数就是达到2层的方案数加上到达3层的方案数，一共是2+3=5种。

最后，我们可以归纳出 $f(n) = f(n-1) + f(n-2)$ ，这是一个斐波那契数列。

即，想求出到达第n层的方案数，就需要计算出到达前n-1和前n-2层的方案数。

对于第二个问题，我们发现n的范围已经达到了5000，很明显，如果使用普通的long 或者 int 进行计算，结果会是一个天文数字，因此，本题需要使用高精度加法。

## 编码

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

using namespace std;
const int Maxn = 5000 + 1;
//缓存数组，避免对于某层进行重复运算
string record[Maxn];
//5000位的计算结果
int a[Maxn], b[Maxn];
int lenA, lenB;

string CountSum(string numA, string numB) {
    memset(a, 0, sizeof(a));
    memset(b, 0, sizeof(b));
    string res;
    lenA = numA.size();
    lenB = numB.size();
    for (int i = 0; i < lenA; i++) {
        a[i] = numA[lenA - 1 - i] - '0';
    }
    for (int i = 0; i < lenB; i++) {
        b[i] = numB[lenB - 1 - i] - '0';
    }
    lenA = max(lenA, lenB); //保证后续的计算按照位数最高的进行循环。
    for (int i = 0; i < lenA; i++) {
        a[i] += b[i]; //按位进行加法运算
    }
    for (int i = 0; i < lenA; i++) {
        a[i + 1] += a[i] / 10; //高位进位
        a[i] %= 10; //低位保留余数
    }
}
```

```

    }
    while (a[lenA]) { //循环判断最高位是不是需要进行进位
        a[lenA + 1] += a[lenA] / 10;
        a[lenA] %= 10;
        lenA++;
    }
    for (int i = lenA - 1; i >= 0; i--) { //逆序输出最后的结果;
        res += a[i] + '0';
    }
    return res;
}

//统计当前楼层共有几种走法
string CountFloor(int n) {
    if (record[n] != "") {
        return record[n];
    }

    //0,1,2,层可以预先设置
    if (n == 0)
        return "0";
    if (n == 1)
        return "1";
    if (n == 2)
        return "2";
    string res = CountSum(CountFloor(n - 1), CountFloor(n - 2));
    //到达当前楼层的走法数量: 等于到达前一层的走法总数+到达前两层的走法总数
    return record[n] = res;
}

int n;

int main(int argc, char **argv) {
    cin >> n;
    cout << CountFloor(n);
    return 0;
}

```

此外，本题还有一种更加简便的写法。

```

#include<bits/stdc++.h>

using namespace std;

////f[k][i]--第k阶台阶所对应的走法数
int f[5010][5010];

int n, len;

```

```

//高精加法
void Plus(int k) {
    //遍历每一位进行计算
    for (int i = 1; i <= len; i++) {
        f[k][i] = f[k - 1][i] + f[k - 2][i];
    }
    //进位
    for (int i = 1; i <= len; i++) {
        if (f[k][i] >= 10) {
            f[k][i + 1] += f[k][i] / 10;
            f[k][i] %= 10;
            if (f[k][len + 1] > 0) {
                len++;
            }
        }
    }
}

int main() {
    cin >> n;
    len = 1;
    f[1][1] = 1; //预处理
    f[2][1] = 2; //预处理
    //开始计算
    for (int i = 3; i <= n; i++) {
        Plus(i);
    }
    //输出
    for (int i = len; i >= 1; i--) {
        cout << f[n][i];
    }
    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。

