## 题目描述：

一只青蛙一次可以跳上1级台阶，也可以跳上2级。求该青蛙跳上一个n级的台阶总共有多少种跳法（先后次序不同算不同的结果）。

## 解题思路：

比较倾向于找规律的解法，f(1) = 1, f(2) = 2, f(3) = 3, f(4) = 5，  可以总结出f(n) = f(n-1) + f(n-2)的规律，但是为什么会出现这样的规律呢？假设现在6个台阶，我们可以从第5跳一步到6，这样的话有多少种方案跳到5就有多少种方案跳到6，另外我们也可以从4跳两步跳到6，跳到4有多少种方案的话，就有多少种方案跳到6，其他的不能从3跳到6什么的啦，所以最后就是f(6) = f(5) + f(4)；这样子也很好理解变态跳台阶的问题了。

## 算法描述：

class Solution {

public:

int jumpFloor(int number) {

if (number <= 0)

return 0;

if (number == 1)

return 1;

if (number == 2)

return 2;

if (number >= 3)

{

int i;

int f1 = 1, f2 = 2,f3;

for (i=3;i<=number;i++)

{

f3 = f1 + f2;

f1 = f2;

f2 = f3;

}

return f3;

}

}

};