/\*递归防坑指南

（1）递归结束条件，一般是最底层状态。但是适当多判断几个次底层状态，确保不会发生，后面的递推到你设置的第一底层状态还底层或者递推到你未设置的次底层状态，否则，递推无法终止进而不能回溯。

\*/

/\*递归方法计算一个数组内所有数的和

形参是：数组基地址，元素数目，和\*/

求和

long int Sum(int \*a, int n) {

return (0 >= n) ? 0 : a[n - 1] + Sum(a, n - 1);

}

斐波那契数列

/\*斐波那契数列,输入n，求数列第n项

f(0) = 0

f(1) = 1

f(2) = 1

f(3) = 2

...

...

f(n) = f(n-1)+f(n-2)

注：复习时请务必证明递归的时间复杂度是2的n次方！！！

\*/

long int f(int n) {

if (0 >= n)

return 0;

if (1 == n)

return 1;

else

return f(n - 1) + f(n - 2);

}

/\*当n = 40时，递归运行时间已经令人发指；

用循环高效解决斐波那契数列

设置数组a，预定义0号单元值为0,1号单元值为1

a[2] = a[0] + a[1]

a[i] = a[i - 2] + a[ i - 1]

一直把数组填充到n号单元，问题解决

\*/

青蛙跳台阶

/\*青蛙跳台阶问题\*/

/\*一只青蛙，一次可以跳1个台阶或2个台阶，问，青蛙跳上第100层

台阶有多少种跳法？

【问题补充说明】所谓跳上第n层台阶，意思是青蛙腿刚好落在第n层台阶上，

不能是青蛙在第n-1层台阶上，一下又跳2个台阶，这样不算是跳上第n层台阶，

算是跳上第n+1层台阶。

\*/

/\*思路：记f(n)是青蛙跳上第n层台阶的跳法总数

青蛙跳上第n个台阶,可分为两种情况：

1，青蛙的脚踩过第n-1个台阶

则青蛙最后一跳只能跳一个台阶，f(n) = f(n-1)

2，青蛙没踩过第n-1个台阶

则青蛙最后一跳只能跳2个台阶，f(n) = f(n-2)

综上，f(n) = f(n-1) + f(n-2)

\*/

#include<stdio.h>

long f(int n) {

if (n <= 0)

return 0;

if (1 == n)

return 1;

if (2 == n)

return 2;

else {

return f(n - 1) + f(n - 2);

}

}

int main() {

int n;

int i = 4;

while (i) {

printf("青蛙要跳到第几个台阶？\n");

scanf("%d", &n);

printf("青蛙共有%d种跳法。\n", f(n));

--i;

}

return 0;

}