

## B 下楼梯的和尚

### 题目描述:

---

从前有座山，山里有座庙，庙里有个老和尚 Island Close 和小和尚~~~o(∩\_∩)o~~~  
一天，Island Close 闲来无事，准备靠走下庙前面的楼梯来强身健体~! OwO! ~  
Island Close 一步可以向下走一个或两个或三个台阶 //我不会告诉你他一次向下走4个台阶会拉伤的 OwO  
Island Close 想看看他能下去的方法一共有多少种，//这句和上一句是重点。。。于是他就下了一次两次三次然后就累了，然后就请小和尚来帮忙了//再一次躺枪了!!!  
小和尚很懒，他才不想一次一次下楼梯呢。。。TAT\*2  
于是怎么做才能得到老和尚想要的答案呢?

### 输入格式:

---

多组测试数据。  
每组测试数据为一行，包含一个整数  $n$  ( $0 < n < 100000$ )，表示台阶总数;

### 输出格式:

---

对于每组测试数据，输出一行，包含一个整数，表示方法一共有多少种。  
结果请对 12211060 取模~~~OwO~~~

### sample in:

---

1  
2  
3

### sample out:

---

1  
2  
4

## 解题分析：

本题和 A 题在本质上唯一的区别，是数据量为  $1 \sim 100000$ 。由于斐波那契数列的递归算法的时间复杂度为  $O(2^n)$ ，显然不适用于上面那个大小的数据范围，因此本题应该使用迭代解决，也就是循环解决。

递推式  $F(n) = F(n-1) + F(n-2) + F(n-3)$

因此设置四个变量，轮换更新即可，具体参见下面代码。

## 参考代码:

```
#include <iostream>
#define m 12211060 //宏定义 m, 之后的程序中 m 的值即会变为 12211060
using namespace std;
int main()
{
    long long int t, a, b, c, sum;
    while(cin>>t)
    {
        a=1;
        b=2;
        c=4;
        if(t==1||t==2)
            cout<<t<<endl;
        else if(t==3)
            cout<<4<<endl;
        else
        {
            while(t>3) //迭代从第 4 个数才会开始, 次数为 t-3
            {
                t--;
                sum=(a%m+b%m+c%m)%m; //模除 12211060 防止数据溢出
                a=b;
                b=c;
                c=sum;
            }
            cout<<sum<<endl;
        }
    }
}
```