1. 用递归方法求解如下问题，计算到第n项为止。

$$f(x)=x-\frac{x^3}{3!}+\frac{x^5}{5!}-\frac{x^7}{7!}+\ \ldots\ldots$$

```
/*用递归方法求解如下问题，计算到第n项为止。
f(x)=x-x^3/3!+x^5/5!-x^7/7!+.....*/
#include<iostream>
#include<cmath>
using namespace std;
double f(int n,double x);
double fac(int n);
int main()
{
    int n,i;
    double x;
    cin>>n>>x;
    cout<<f(n,x)<<endl;
}
double f(int n,double x)//求多项式
{
    if(n==1){
        return x;
    }
    else{
        return pow(-1,n-1)*pow(x,2*n-1)/fac(2*n-1)+f(n-1,x);
    }
}
```

```
double fac(int n)//求阶乘，即多项式中的分母
{
    if(n==1){
        return 1;
    }
    else{
        return n*fac(n-1);
    }
}
```

2. 楼梯有n阶台阶，上楼可以一步上1阶，也可以一步上2阶，编一程序计算共有多少种不同的走法。例如，当n=3时，共有3种走法，即1+1+1，1+2，2+1。

```cpp
/*楼梯有n阶台阶，上楼可以一步上1阶，也可以一步上2阶，
编一程序计算共有多少种不同的走法。例如，当n=3时，共
有3种走法，即1+1+1， 1+2， 2+1。*/
#include<iostream>
#include<cmath>
typedef long long ll;
using namespace std;
ll f(int n);
int main()
{
    int n;
    cin>>n;
    cout<<f(n)<<endl;
}

ll f(int n)
{
    if(n==1){//台阶数为1时有1种走法
        return 1;
    }
    if(n==2){//台阶数为2时有2种走法
        return 2;
    }
    if(n==3){//台阶数为3时有3种走法
        return 3;
    }
    else{//通过找规律可发现n级台阶的走法等于n-1级台阶的走法加上n-2级台阶的走法
        return f(n-1)+f(n-2);
    }
}
```