**一、问题分析**

* 分析并确定要处理的对象（数据）是什么

2个整数

* 分析并确定要实现的功能是什么

接受从键盘输入的两个整数n，k

创建一个元素个数为n的环

模拟约瑟夫问题，让每个人报数，淘汰报的数满足条件的人，直至剩下最后一人

把最后剩余的人的编号输出到屏幕上

* 分析并确定处理后的结果如何显示

将最后剩下的那个人的编号直接输出到屏幕上

* 请用题目中样例，详细给出样例求解过程。

5=>n,2=>k

=>3

**二、数据结构和算法设计**

**STL的选用**

因为该问题的元素的编号是1、2、……、n依次增加，满足线性关系，然后因为该问题会不断遍历并删除元素，所以选用list，其删除元素比vector效率要高。

**算法思想的设计**

本程序由三个模块组成

**输入模块：**设计一模块void input(List<int>&L)，接受n ,k ，并创建一个list，给list每个元素赋值为编号

**计算模块：**设计一模块void count（List<int>&L），模拟约瑟夫不断淘汰的过程，直至list中只剩一个元素

**输出模块：**输出list中最后一模块的储存的值

**关键功能的算法步骤**

**输入模块**

1、接受输入数组n,k;

2、给list插入整数从1到n

void input(list<int>& ring)

{

int n,k;

cin>>n>>k;//从键盘输入

for(int i=0;i<n;i++)

ring.push\_back(i+1);

}

**计算模块**

1、不断遍历list，用一整型变量num储存所报的数

2、若num是k的倍数或者num最后一位为k，则删除当前元素

void count(list<int>& ring)

{

int num=1;//定义计数器，记录当前元素报的数

list<int>::iterator it=ring.begin();//定义迭代器，用来遍历list

while(ring.size()!=1)

{

if(num%k==0||(num-num/10\*10==k))//如果报数满足要求，就删除当前元素

{

it=ring.erase(it);

it--;

}

num++;

it++;//继续遍历

if(it==ring.end())//如果遍历到最后一个元素，则又回到开头，形成闭环

it=ring.begin();

}

}

**输出模块：**直接输出list中的元素的值

cout<<\*（list.begin()）;

**三、算法性能分析**

**输入模块：**调用函数void input(list<int>& ring)，时间复杂度为O(n)

**计算模块：**调用函数void count(list<int>& ring)，时间复杂度为O(n)

**输出模块：**调用函数cout，时间复杂度为O(1)