**实验2.2 游戏**

**计科1903 201914020128 陈旭**

1. **问题分析**

**1). 问题与功能描述**

**1.**需要处理的数据是两个整型(int型)数字n(小朋友人数)，k(判断用数)。

**2.**实现的功能是：先插入队列循环判断作为报数，当所报的数为k的倍数或者最后一位是k的时候标记为被淘汰并出队列，然后最终只剩一个小朋友未被淘汰的时候输出该小朋友的序号。

**3.**直接采用标准输出一个整数即可。

**2). 样例分析**

**1. 求解方法：**① 将n个小朋友，

② 对比每一个元素和它相邻的元素的差值绝对值，用一个数记录最小值，如果新的绝对值比前面的值要小，则记录的数被赋值为新的最小绝对值。最终输出这个用来记录最小值的数。

**2. 样例求解：**① 【样例一】

　　 输入：5 2  
1号小朋友报数1；

　2号小朋友报数2淘汰；  
　　3号小朋友报数3；  
　　4号小朋友报数4淘汰；  
　　5号小朋友报数5；  
　　1号小朋友报数6淘汰；  
　　3号小朋友报数7；  
　　5号小朋友报数8淘汰；

3号小朋友获胜。

② 【样例二】

输入：7 3。

1号小朋友报数1；

2号小朋友报数2；

　 3号小朋友报数3淘汰；

4号小朋友报数4；

5号小朋友报数5；

6号小朋友报数6淘汰；

7号小朋友报数7；

1号小朋友报数8；

2号小朋友报数9淘汰；

4号小朋友报数10；

5号小朋友报数11；

7号小朋友报数12淘汰；

1号小朋友报数13淘汰；

4号小朋友报数14；

5号小朋友报数15淘汰；

4号小朋友获胜

1. **抽象数据类型设计**

需要处理的数据对象的类型为整型，且具有序列性，并且需要多次循环来判断是否被淘汰，我们可以将这一特点归纳为先进先出的特性，因此我们采用STL中的queue(队列)来处理该题目。

1. 算法分析

* **算法思想**：引入queue头文件，使用序列式容器queue进行数据的存储。然后依

次压入队列，循环报数，如果符合上述两个条件之一的采用出队列操作； 不满足则重新压入队尾进行上述操作。

* **样例分析：**① 【样例一】

输入：5 2

1号小朋友报数1，弹出队列后压入队尾；

　　2号小朋友报数2出队列；

　　3号小朋友报数3，弹出队列后压入队尾；

　　4号小朋友报数4出队列；

　　5号小朋友报数5，弹出队列后压入队尾；

　　1号小朋友报数6出队列；

　　3号小朋友报数7，弹出队列后压入队尾；

　　5号小朋友报数8出队列；

3号小朋友获胜，输出3。

② 【样例二】

输入：7 3

1号小朋友报数1，弹出队列后压入队尾；

2号小朋友报数2，弹出队列后压入队尾；

　 3号小朋友报数3，出队列；

4号小朋友报数4，弹出队列后压入队尾；

5号小朋友报数5，弹出队列后压入队尾；

6号小朋友报数6，出队列；

7号小朋友报数7，弹出队列后压入队尾；

1号小朋友报数8，弹出队列后压入队尾；

2号小朋友报数9，出队列；

4号小朋友报数10，弹出队列后压入队尾；

5号小朋友报数11，弹出队列后压入队尾；

7号小朋友报数12，出队列；

1号小朋友报数13，出队列；

4号小朋友报数14，弹出队列后压入队尾；

5号小朋友报数15，出队列；

4号小朋友获胜，输出4。

* **关键功能的算法步骤：**关键步骤实现在一个while循环里。

while(listsize<1)

{

num=list.front(); //记录队首元素

if(order.lastnumber!=k&&order%k!=0) //判断是否满足游戏规则

{

list.push(num); //不能被淘汰的继续插入队尾。

}

order++;//报数加一

}

* **性能分析**

【空间复杂度】所调用的queue的内部函数，其空间复杂度为C1,循环中每往队列里面插入一个数，进行一次空间占用，开销为C2n;故整个程序的总空间复杂度为O(n)

【时间复杂度】先循环输入值并引入队列，时间开销为C1n，然后循环判断，分析知k次循环以内必然判断结束，则时间开销为C2n，综上时间复杂度为O(n)。