实验二 游戏

一、问题分析

1，处理对象

输入一行，包括两个整数*n*和*k。*

2，功能需求

有*n*个小朋友围成一圈玩游戏，小朋友从1至*n*编号，2号小朋友坐在1号小朋友的顺时针方向，3号小朋友坐在2号小朋友的顺时针方向，……，1号小朋友坐在*n*号小朋友的顺时针方向。

游戏开始，从1号小朋友开始顺时针报数，接下来每个小朋友的报数是上一个小朋友报的数加1。若一个小朋友报的数为*k*的倍数或其末位数（即数的个位）为*k*，则该小朋友被淘汰出局，不再参加以后的报数。当游戏中只剩下一个小朋友时，该小朋友获胜。  
　　给定*n*和*k*，请问最后获胜的小朋友编号为多少？

3，结果显示

输出一行，包含一个整数，表示获胜的小朋友编号。

4，样例分析

输入：

5 2

输出：

   3

样例说明：

  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。

二、数据结构与算法分析

1，抽象数据类型设计：

题设所给数据：对于所有评测用例，1 ≤ *n* ≤ 1000，1 ≤ *k* ≤ 9。因此，数据类型选用int整型。

2，物理数据对象设计

本次实验使用STL中的queue实现，物理数据对象为队列queue。

3，算法思想的设计

新建一个队列，把每个小朋友的编号放入，设置一个计数器，用于表示小朋友所报的编号，每次对编号进行筛选，符合条件的把队列头元素弹出，不符合条件的，把头元素放入队尾，直到队列中只剩下一个元素。

4，算法步骤

声明队列 q;

for 1到n

编号压入队列；

while(队列元素个数不为1){

if(满足筛选条件){

弹出队首元素；

} else {

队首元素放入队尾；

}

}

三、算法性能分析

1，时间复杂度

O（n）

2，空间开销

内存占用小，并且随时删除队列中的元素。空间开销基本为一。