

湖南大学

数据结构

课前资料搜集

主 题： 数据结构与算法设计

学生姓名： 刘大卫

学生学号： 201826010215

专业班级： 软件 1802

讨论周次： 第四周

一、需求分析

1. 题目内容

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。

例如，当 $n=5$ ， $k=2$ 时：

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。

给定 n 和 k ，请问最后获胜的小朋友编号为多少？

【输入形式】

输入一行，包括两个整数 n 和 k ，意义如题目所述。

【输出形式】

输出一行，包含一个整数，表示获胜的小朋友编号。

对于所有评测用例， $1 \leq n \leq 1000$ ， $1 \leq k \leq 9$ 。

2. 样例输入

5 2

7 3

1000 9

3. 样例输出

Case 1: 3

Case 2: 4

Case 3: 568

二、 分析

根据前面的经验，如果用数组来模拟报数过程，时间代价较大。

解决方法是采用 STL 里的队列 (queue)：每次都从队列中弹出队首元素，若满足条件则继续循环，不满足删除条件就把弹出的队首元素压入队尾，设置模拟报数器，不断重复这个过程：

```
cur = q.队首元素;  
if(满足条件){  
    q.pop();  
} else {  
    q.pop();  
    q.push(cur);
```

}提示：在队列这样的结构中，可调用一些辅助函数来设置。

有了这个代码，队列中的元素始终不超过 1000，执行步骤也比使用数组大大简化。反复使用出队入队的操作，使得数据更加灵活，操作更加简便。

提示：如果数据结构上的某一个操作很耗时，有时可以用加标记的方式处理。但同时，该数据结构的所有其他操作都要考虑这个标记。

三、 数据结构的重要意义

一般认为，一个数据结构是由数据元素依据某种逻辑联系组织起来的。对数据元素间逻辑关系的描述称为数据的逻辑结构；数据必须在计算机内存储，数据的存储结构是数据结构的实现形式，是其在计算机内的表示；此外讨论一个数据结构必须同时讨论在该类数据上执行的运算才有意义。一个逻辑数据结构可以有多种存储结构，且各种存储结构影响数据处理的效率。

在许多类型的程序的设计中，数据结构的选择是一个基本的设计考虑因素。许多大型系统的构造经验表明，系统实现的困难程度和系统构造的质量都严重的依赖于是否选择了最优的数据结构。许多时候，确定了数据结构后，算法就容易得到了。有些时候事情也会反过来，我们根据特定算法来选择数据结构与之适应。不论哪种情况，选择合适的数据结构都是非常重要的。

选择了数据结构，算法也随之确定，是数据而不是算法是系统构造的关键因素。这种洞见导致了许多种软件设计方法和程序设计语言的出现，面向对象的设计语言就是其中之一。

四、 如何调试数据结构

提示：简单的链式数据结构往往较为容易地解决问题。在包含多道题目的算法竞赛中，这一特点可以是选题的依据之一。

简单地说，测试的任务就是检查一份代码是否正确。如果找到了错误，最好还能提供一个让它错误的数据。有了错误数据之后，接下来的任务便是调试：看看程序为什么是错的。如果找到了错误，最好把它改对——至少对于刚才的错误数据能得到正确的结果。改对一组数据之后，可能还有其他错误，因此需要进一步测试；即使以前曾经正确的数据，也可能因为多次改动之后反而变错了，需要再次调试。总之，在编码结束后，为了确保程序的正确性，测试和调试往往要交替进行。

提示：测试的任务就是检查一份代码是否正确。如果找到了错误，最好还能提供一个让它出错的数据；调试的任务是找到错误原因并改正。改正一个错误之后有可能引入新的错误，因此调试和测试往往要交替进行。

如何测试上述代码的正确性呢？一个行之有效的方法是：再找一份完成同样功能的代码与之对比。对于本题来说，可以先写一个基于数组的版本。虽然这个版本会很慢，但正确性比较容易保证。接下来编写一个数据生成器（在第 5 章中曾介绍过这一技巧），并且反复执行下面的操作：生成随机数据，分别执行两个程序，比较它们的结果（俗称“对拍”）。合理地使用操作系统提供的脚本功能，可以自动完成对比测试。

提示：测试数据结构程序的常用方法是对拍：写一个功能相同但速度较慢的简易版本，再写一个数据生成器，不停对比快慢两个程序的输出。简易版本的代码越简单越好，因为重点不在效率，而在正确性。

如果发现让两个程序答案不一致的数据，最好别急着对它进行调试。可以尝试着减小数据生成器中的 n 和 m ，试图找到一组尽量简单的错误数据。一般来说，数据越简单，越容易调试。如果发现只有很大的数据才会出错，通常意味着程序在处理极限数据方面有问题，例如，`is_prime` 中遇到了“过大的 n ”，或者数组

开得不够大等。这些都是很实用的技巧。

五、 参考文献

[1] 刘汝佳. 算法竞赛入门经典[M] 第二版. 北京:清华大学出版社, 2009:144-147

[2] 洛谷网. UVA12657 Boxes in a Line. [HTML].

<https://www.luogu.org/problemnew/show/UVA12657>

[3] 百度百科 数据结构 （计算机存储、组织数据方式）[HTML].

<https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E7%BB%93%E6%9E%84/1450?fr=aladdin>