**数据结构课程设计**

**-银行业务**

# 题目描述

1. 项目内容

设某银行有A，B两个业务窗口，且处理业务的速度不一样，其中A窗口处理速度是B窗口的2倍----即当A窗口每处理完2个顾客是，B窗口处理完1个顾客。给定到达银行的顾客序列，请按照业务完成的顺序输出顾客序列。假定不考虑顾客信后到达的时间间隔，并且当不同窗口同时处理完2个顾客时，A窗口的顾客优先输出。

1. 项目功能要求

输入说明：输入为一行正整数，其中第一数字N（N<=1000）为顾客总数，后面跟着N位顾客的编号。编号为奇数的顾客需要到A窗口办理业务，为偶数的顾客则去B窗口。数字间以空格分隔。

输出说明：按照业务处理完成的顺序输出顾客的编号。数字键以空格分隔，但是最后一个编号不能有多余的空格。

# 总体思路

本题就是把输入的数据根据奇偶性分别加入到两个队列qA和qB中，然后这两个队列相互轮次输出，qA输出两个元素，qB输出一个元素，直到其中一个队列已经全部输出完毕，剩下的那个队列将剩余元素全部输出即可。

按照此思路先用STL标准库写一遍，做好错误输入和代码正确性的校验，再在此基础上用自定义的queue类替换标准库的队列容器，类成员函数可以按照STL库的逐一复刻需要的功能。

# 代码类设计

头文件myqueue.h包含了一个类定义，queue类。已知队列的存储表示有两种方式：一种是基于数组的存储表示，另一种是基于链表的存储表示。但由于在本项目问题中，优先给定了所包含的元素数量num，因此选择定长且存储空间小的数组存储格式。以下是自定义queue类的成员函数和变量：



# 具体实现

## 输入处理

因为此项目所有输入均是数字，所以可以将所有错误处理整合到一个函数中，传入参数有该变量的数值和该变量应该处于的区间范围，此即函数实现的功能，left和right是参数num正确的区间左右边界。顾客信息的输入用以i为循环变量的for循环即可，如果输入出现错误，则该函数会返回false，将循环变量i重新置零，并将队列qA和qB清空，重新输入。

## 出队列

目的是为了将两队列中所有数全部输出，因此外层while循环终止条件为qA和qB都为空。内部先进行qA的两次输出，输出先决条件都是qA队列不为空，再进行qB的两次输出，输出先决条件是qB队列不为空。因为输出是以空格为间隔，且最后一个数字后不能有空格，因此需要一个变量用于标记输出的个数，而此时用于记录元素总数量的变量num已经没有用处，可以直接拿来用作标记变量，每输出一个数就自减一，当num为1时，输出元素后不需要输出空格。

## queue类的实现

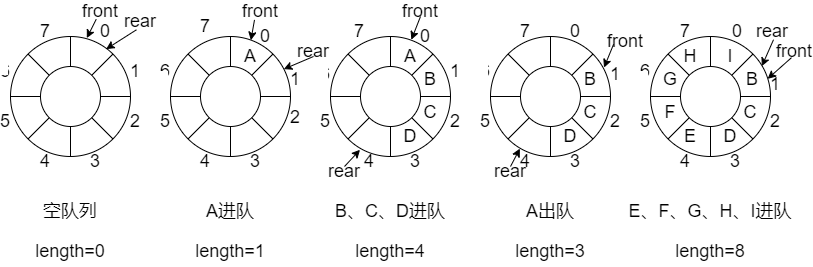
如果队列储存数据的数组下标为0的元素作为队列的头元素，即下一个要出队列的元素，那么每次都需要将数组内元素全体向前移动，这样会导致时间复杂度过高。因此我们选择添加头指针和尾指针用于标记队列的头元素和尾元素，用指针的移动代替数据的移动。如此，可得到如下对队列的操作：

1. clear():初始化这两个指针为0，清空队列也为这两个指针全赋值为0
2. push()：队列未满的前提下，当前尾指针指向的位置赋值为新增值，++
3. pop()：队列不为空的前提下，当前头指针直接自增
4. front()：返回当前头指针指向的数，若队列为空返回第一个值
5. isEmpty()：时，队列为空
6. isFull()：时队列满

如果这时加入新元素，就会产生“溢出”。但这种“溢出”很可能是假溢出，因为在数组的前端还可能有空余位置。为了充分利用数组中的存储空间，把数组的前端和后端连接到一起，形成一个环形的表，即从逻辑上将存储元素的表看做是一个环，成为循环队列。而要实现这个功能，只需要在或到达时，将其前进一格的位置设为0，具体操作可用除法取余做到：

1. 头指针进1：
2. 尾指针进1：

然而这样做，判断队列是否为空或为满的条件就重复了，都是，因此并不清楚是哪个指针加一后与另一指针相等，分不清到底是队列为空还是为满。但在该类中有变量length，添加数据时加一，弹出数据时减一，可根据它的数值作为队列为空或队列已满的判断依据。



以为为queue类部分功能代码：

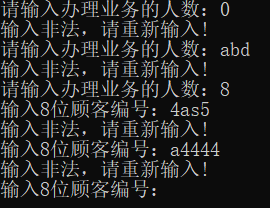
# 项目小结

由于是用数组作为存储类型的，因此空间复杂度为，且该项目中只需用到队列的入队、出队、清空和取头元素的操作，且每种操作时间消耗都为，因此总时间复杂度为。

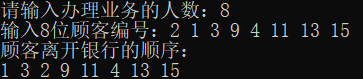
本项目的队列最大值已经定义，才开辟的队列空间。如果队列最大值未知或者元素变动比较大的情形下，则数组存储的方式在对溢出时，扩增存储空间方面将差强人意，因为需要重新开辟空间并再赋值。此时应该用链表来存储队列，不存在队列满而产生溢出的情况。

# 程序功能测试

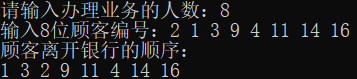
错误输入处理：



正常测试：A窗口人多



正常测试：B窗口人多



最小N

