**数据结构课程设计**

**项目说明文档**

**银行业务**

作 者 姓 名： 苏家铭

学 号： 2151299

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

**Tongji University**



**目录**

[1 项目分析 1](#_Toc177)

[1.1 项目背景 1](#_Toc26686)

[1.2项目要求 1](#_Toc3643)

[1.2.1 功能要求 1](#_Toc21769)

[1.2.2 输入要求 1](#_Toc10817)

[1.2.3 输出要求 1](#_Toc29152)

[1.2.4 项目实例 1](#_Toc14495)

[2 项目设计及实现 2](#_Toc28985)

[2.1 数据结构设计思路 2](#_Toc12246)

[2.2 类设计 2](#_Toc19951)

[2.2.1 队列 2](#_Toc27719)

[2.3 项目算法 3](#_Toc21509)

[2.3.1 实现思路 3](#_Toc3691)

[2.3.2 代码实现 4](#_Toc7307)

[3 项目测试 5](#_Toc17313)

[3.1 正常测试，A窗口人多 5](#_Toc32175)

[3.2 正常测试，B窗口人多 6](#_Toc5764)

[3.3 最小N 6](#_Toc14392)

[3.4 总人数N输入非法 6](#_Toc2799)

[3.5 编号序列有非法字符 7](#_Toc13827)

[4 算法性能分析 7](#_Toc11026)

[4.1 正确性 7](#_Toc2357)

[4.2 可使用性 7](#_Toc5706)

[4.3 可读性 7](#_Toc6724)

[4.4 效率 8](#_Toc26519)

[4.5 健壮性 8](#_Toc8969)

# 1 项目分析

## 项目背景

在日常生活中排队现象随处可见。每天下楼做核酸排起的长队、在食堂晚到一分钟人数便迅速增长的队伍、在上火车时人们有序排起的长队...对于这类生活中十分常见的现象，研究排队问题显得尤为重要。

插队、排错队伍等行为或叫人难堪或引发不满，诸如此类现象都是错误排队的严重后果。在不同特定情况下，若能借助计算机程序设计出一个最优的排队方案，那么社会将会更有序和谐。

本项目依托银行排队问题为背景，提出了一种特定情况下的排队问题，期望能得到该问题的解，即满足若干条件的排队结果。

## 1.2项目要求

### 1.2.1 功能要求

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个顾客时，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。

### 1.2.2 输入要求

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。

### 1.2.3 **输出要求**

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。

### 1.2.4 项目实例

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 输入 | 输出 | 说明 |
| 1 | 8 2 1 3 9 4 11 13 15 | 1 3 2 9 11 4 13 15 | 正常测试，A窗口人多 |
| 2 | 8 2 1 3 9 4 11 12 16 | 1 3 2 9 11 4 12 16 | 正常测试，B窗口人多 |
| 3 | 1 6 | 6 | 最小N |

# 2 **项目设计及实现**

## 2.1 **数据结构设计思路**

既然称之为银行排队，那么首先想到的便是队列这一数据结构。而队列这种具有先进先出的恰好可以满足本题的需求，对于编号为奇数的顾客进入qA队列（在A窗口前排队），编号为偶数的顾客进入qB队列（在B窗口前排队）。本项目基于已经实现的链表，完成了链式队列这种存储表示。适合于数据元素变动比较大的情况，不存在队列满而产生溢出的情况。

## 2.2 **类设计**

### 2.2.1 队列

队列(Queue)是只允许在一端进行插入，而在另一端进行删除的运算受限的线性表，允许插入的一端叫做队尾(back)，允许删除的一端叫做队头(front)。队列也叫做先进先出的线性表，即FIFO(First In/First Out).

而本项目基于已经实现了的双向循环链表，实现了链式队列。每次入队时在链表的末尾插入一个元素，出队时则删除链表的第一个元素。出队入队都可以通过链表的first指针的左右指针找到队首和队尾，则无需再定义last指针，也无需遍历一遍链表找队尾元素，即出队、入队的时间复杂度可以降为O(1).

队列类数据成员：

template<class T>

DblList<T> list;

该链式队列已经实现6种功能：入队、出队、置空、判空、返回队首元素、返回队列长度。

具体函数声明为：

* void push(const T& val);//入队
* void pop();//出队
* void makeEmpty();//置空
* bool empty()const;//判断空否
* int size()const;//返回队列大小
* const T& front();//返回队首元素

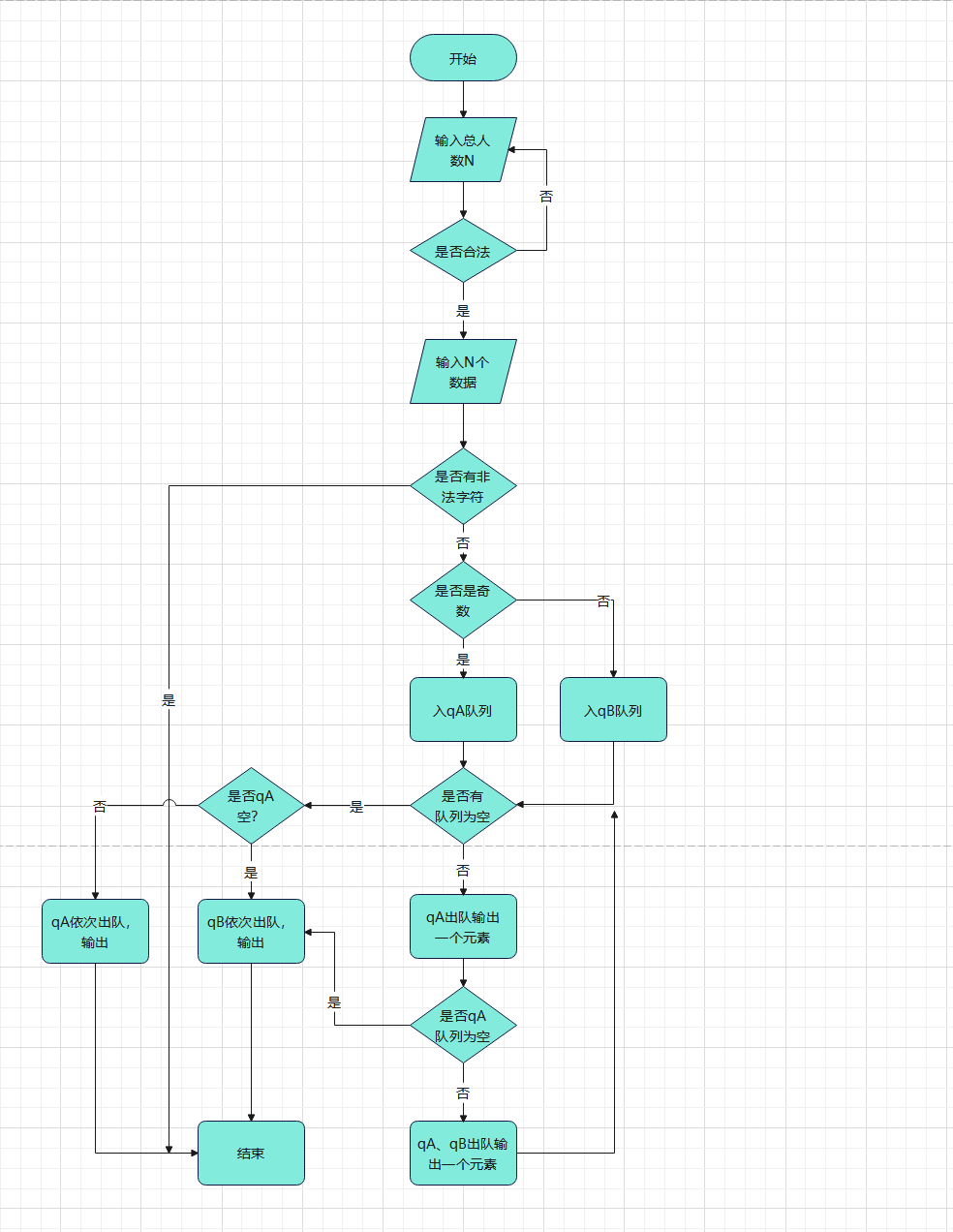
主要算法的时间复杂度：

* 入队：O(1)
* 出队：O(1)
* 置空：O(n)
* 判断空否：O(1)
* 返回队首元素：O(1)
* 返回队列大小：O(n)

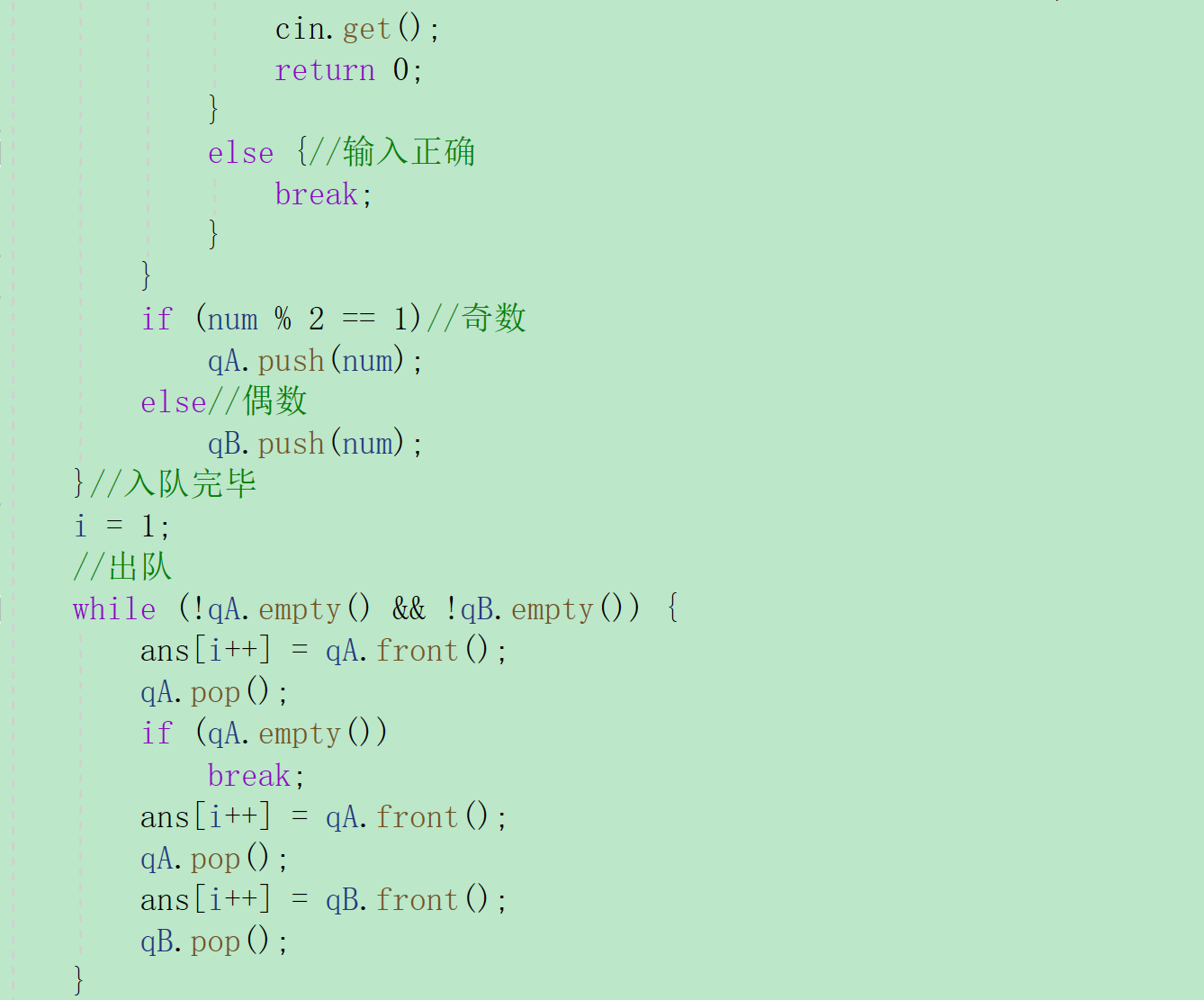
## 2.3 **项目算法**

### 2.3.1 实现思路

当有一位奇数编号的顾客来到A窗口，那么在编程中表示为该元素入qA队列；当一位偶数编号的顾客来到B窗口，那么可以相应地表示为该元素入B队列。根据题意，应当先排好队列再办理银行业务，也即先把所有元素入队，再开始按规则输出。具体流程图如下：



### 2.3.2 代**码实**现



# 3 项目测试

## 3.1 正常测试，A窗口人多

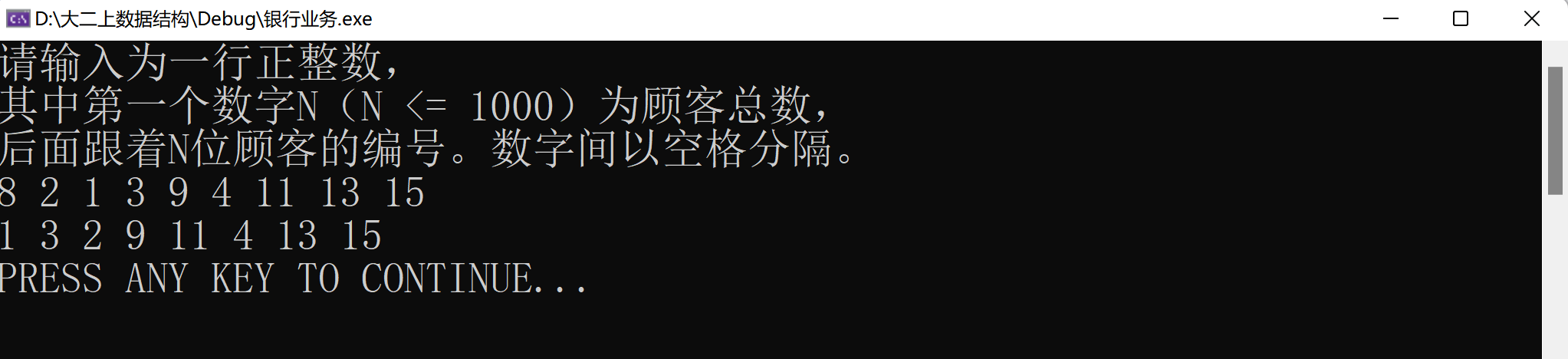
测试输入：

8 2 1 3 9 4 11 13 15

预期输出：

1 3 2 9 11 4 13 15

实验结果：



## 3.2 正常测试，B窗口人多

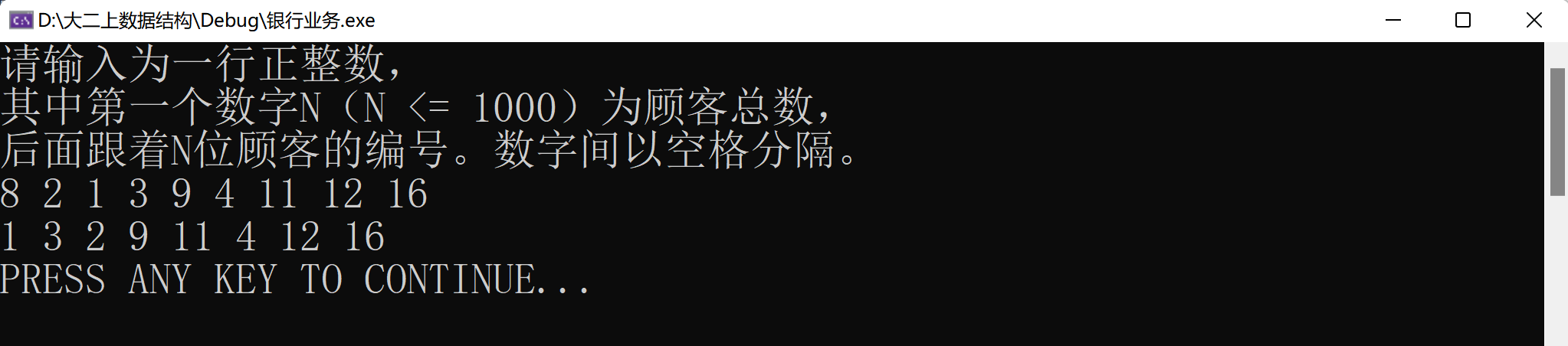
测试输入：

8 2 1 3 9 4 11 12 16

预期输出：

1 3 2 9 11 4 12 16

实验结果：



## 3.3 最小N

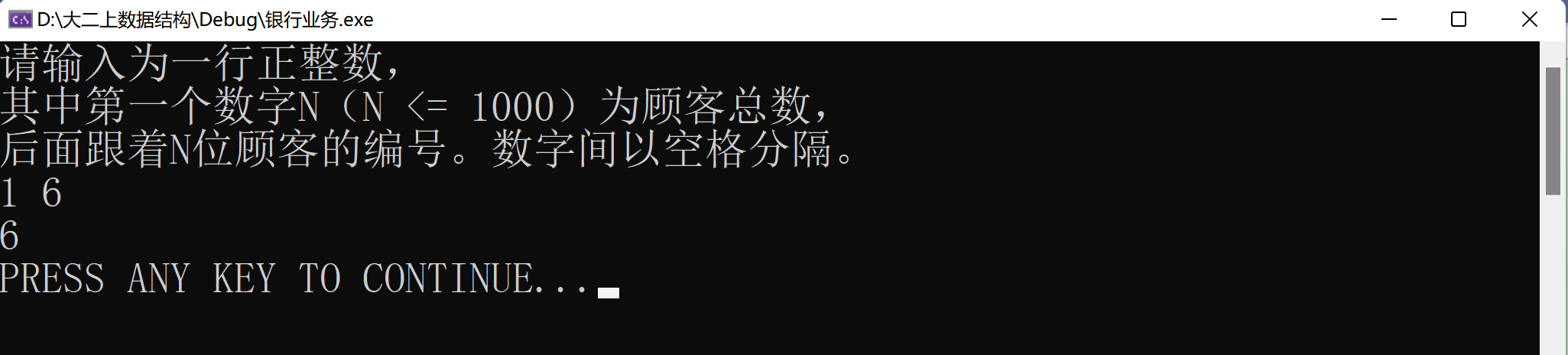
测试输入：

1 6

预期输出：

6

实验结果：



## 3.4 总人数N输入非法

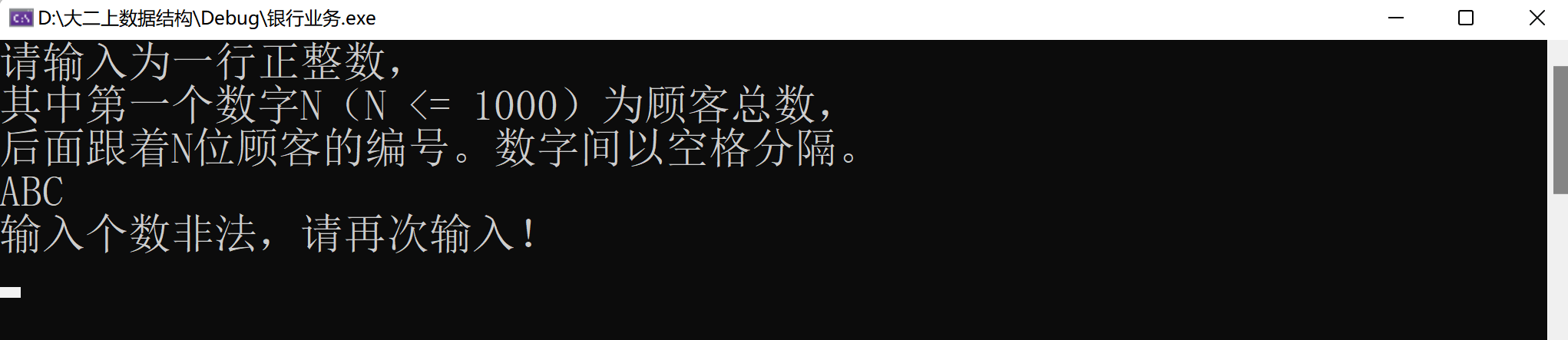
测试输入：

ABC

预期输出：

有错误提示，重新输入

实验结果：



## 3.5 编号序列有非法字符

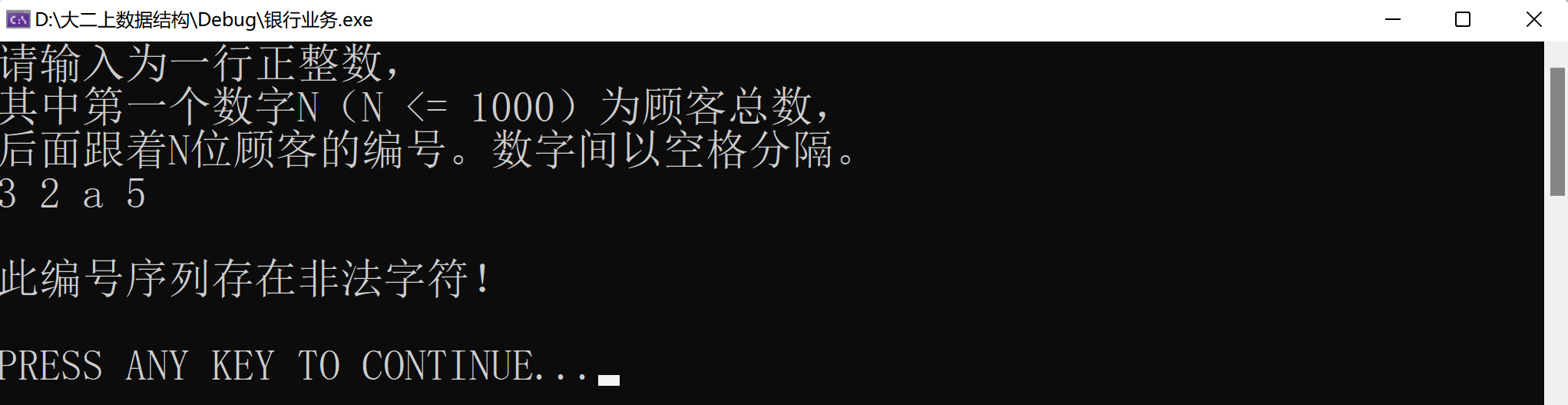
测试输入：

3 2 a 5

预期输出：

有错误提示，退出

实验结果：



# 4 算法性能分析

## 4.1 正确性

本算法能正确地执行预定的功能和性能要求，按照业务处理完成的顺序2基数1偶数顺序输出顾客的编号。数字键以空格分隔，并且最后一个编号没有多余的空格。

## 4.2 可使用性

本算法可以很方便地使用，只需输入总人数和序列即可很快求解。并且该算法有良好的界面和完备的用户文档。

## 4.3 可读性

本算法逻辑清晰、简单、且结构化，所有命名与函数名都具有实际含义，让人见名知义。且算法中包含了大量注释，简要说明了算法功能、输入与输出参数的使用规则、重要数据的作用、算法中各程序段完成的功能。

## 4.4 效率

队列的入队、出队、读取操作的时间复杂度已经降为O(1)，时间效率非常可观。

## 4.5 健壮性

本算法对于边界条件，诸如：最小N、输入N有误、输入编号序列有误的情况有相对应的输出提示，对于链式队列的底层数据结构链式表的结点空间申请是否成功也有相应的错误提示。