**项目说明文档**

**数据结构课程设计**

**——银行业务**

作者姓名：香宁雨

学号：1954098

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.题目要求及功能分析 2](#_Toc58629673)

[2.设计 2](#_Toc58629674)

[2.1数据结构设计 2](#_Toc58629675)

[2.2类结构设计 3](#_Toc58629676)

[2.3类的成员与方法 3](#_Toc58629677)

[2.4测试函数 5](#_Toc58629678)

[3.实现 5](#_Toc58629679)

[3.1链表类删除头节点：pop\_front()方法 5](#_Toc58629680)

[3.2链表类增加尾节点：push(type data)方法 6](#_Toc58629681)

[3.3链表类判断是否为空：empty()方法 6](#_Toc58629682)

[3.4链表类判断链表大小：size()方法 6](#_Toc58629683)

[3.5队列类获取头节点值：front()方法 7](#_Toc58629684)

[3.6核心函数：BussinessProcess(vector<int> customer) 8](#_Toc58629685)

[3.7print()函数 9](#_Toc58629687)

[4.输出测试 10](#_Toc58629688)

[4.1正常测试，A窗口人多 10](#_Toc58629689)

[4.2正常测试，B窗口人多 10](#_Toc58629690)

[4.3最小N 10](#_Toc58629691)

[4.4错误输入 10](#_Toc58629692)

# 1.题目要求及功能分析

题目要求：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。

项目功能要求：（1）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。

（2）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。

项目功能分析：先到达窗口的顾客优先被处理，first in first out，符合队列这种数据结构，所以本项目采用队列进行输入输出的处理。

# 2.设计

## 2.1数据结构设计

根据题目要求选择最合适的数据结构进行数据的处理与分析，题目中所描述的银行业务为一个先进先出的队列，所以本项目采用队列进行数据的处理，而为了简化队列的增添与删除操作并防止内存的溢出，本项目采用链表作为队列的基础结构进行队列的实现。

## 2.2类结构设计

队列是一种特殊的线性表，特殊之处在于它只允许在表的前端（front）进行删除操作，而在表的后端（rear）进行插入操作，和栈一样，队列是一种操作受限制的线性表。进行插入操作的端称为对位，进行删除操作的端成为队头。因为队列只允许在一端插入，在另一端删除，所以只有最早进入队列的元素才能从队列中删除，故队列又称为先进先出线性表。

## 2.3类的成员与方法

2.3.1链表类（List）

#### 2.3.2队列类（Queue）



## 2.4测试函数

为强化代码的封装性，充分体现c++语言的面向对象特性，本项目将测试函数封装为一个非main的函数，通过main函数进行调用，这样可以减少因为外部函数对内部函数调用的不正确性。



# 3.实现

3.1链表类删除头节点：pop\_front()方法

3.2链表类增加尾节点：push(type data)方法

## 3.3链表类判断是否为空：empty()方法

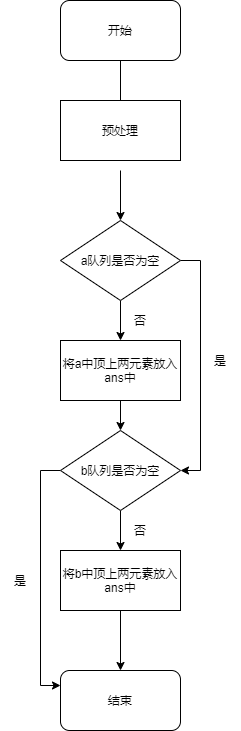


3.4链表类判断链表大小：size()方法

3.5队列类获取头节点值：front()方法

## 3.6核心函数：BussinessProcess(vector<int> customer)

#### 3.6.1流程图



3.6.2代码实现



## 3.7print()函数



# 4.输出测试

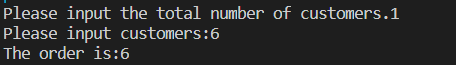
## 4.1正常测试，A窗口人多



## 4.2正常测试，B窗口人多



## 4.3最小N



## 4.4错误输入

