**理工学院计算机类课程实验报告**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 课程名称 | 数据结构 | | 实验名称 | 实验4 Project 队列 |
| 小组名称及成员 | 左康田 姜来 童明安 蔡杰铭 韦清秀 | | | |
| 专 业 | 软件工程 | | 年 级 | 2020级 |
| 班 级 | 软工1班 | | 实验时间 | 2021.4.20-2021.4.25 |
| 教师评语（或成绩） | | 教师签字：  年 月 日 | | |
| **一、实验目的**  （1）了解C项目编写方式  （2）熟悉队列的基本运算以及算法与程序之间的关系 | | | | |
| **二、实验设备（工具、材料、硬软件）及要求**  1、设备：计算机一台、Dev C++  2、具体要求：利用循环队列的基本算法，实现银行业务队列简单模拟。  （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窗口的两倍—即当A窗口处理完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，请按业务完成的顺序输出顾客序列。假定不考虑顾客先后到达的时间间隔，并且当不同窗口同时处理完2个顾客时，A窗口顾客优先输出。） | | | | |
| **三、实验过程**  要求：（1）列出实现的源代码、运行效果截图；（2）为源码添加必要的注释语句。另外，对实验进一步分析存在的问题提出解决办法，或总结实验所取得的经验均可写入报告中。**能力强者也可使用java语言实现代码。**  **/\*神仙打架\*/**  **//姜来**  **#include <stdio.h>**  **#include <stdlib.h>**  **#include "queue.h"**  **int main(int argc, char \*argv[]) {**  **Queue A,B;**  **ElemType e;**  **int n,i,x,temp;**  **//i是循环变量，从第一个客户遍历到最后一个客户，x:输入的元素**  **int result[10000];**  **initQueue(&A);//初始化A队列**  **initQueue(&B);//初始化B队列**  **printf("\n请输入正在等待服务的客户的人数:\n");**  **scanf("%d",&n);//n:客户数量**  **printf("请输入%d位顾客的编号：\n",n);**  **for(i=1;i<=n;i++){**  **scanf("%d",&x);**  **if(x%2!=0)**  **{**  **addQueue(&A,x);//若x为奇数则归到A队列中**  **}else**  **{**  **addQueue(&B,x);//否则归到B队列中**  **}**  **}**  **int flag=0,j;i=1;**  **/\*定义一个标志符，作用是任意队中第一次输出，**  **不用输出空格，直接输出数字，此后输出都要输出空格和数字。\*/**  **while(A.front!=A.rear&&B.front!=B.rear)**  **//当队列A，B都不为空时就继续输出**  **{**  **//A队列出两个**  **if(A.front!=A.rear) {**  **removeQueue(&A,&temp);**  **result[i++]=temp;**  **}**  **if(A.front!=A.rear){**  **removeQueue(&A,&temp);**  **result[i++]=temp;**  **}**  **//B队列出一个**  **removeQueue(&B,&temp);**  **result[i++]=temp;**    **}**  **while(A.front!=A.rear)**  **{**  **removeQueue(&A,&temp);**  **result[i++]=temp;**  **}**  **while(B.front!=B.rear)**  **{**  **removeQueue(&B,&temp);**  **result[i++]=temp;**  **}**  **for(i=1;i<=n;i++)**  **{ if(i==n)**  **printf("%d",result[i]);**  **else**  **printf("%d ",result[i]);**  **}**  **return 0;**  **}**  **#define MAXSIZE 1000//存储数据元素的最大个数为1000**  **typedef int ElemType;**  **typedef struct Queue{**  **ElemType data[MAXSIZE];**  **int front;**  **int rear; //队尾的下一个元素，也就是说这个队尾是虚拟的队尾**  **}Queue;**  **//初始化队列，即建立一个空队操作**  **void initQueue(Queue \*q)**  **{**  **q->front=0;// 队头**  **q->rear=0;//队尾**  **}**  **//入队,即增加**  **void addQueue(Queue \*q,ElemType e)**  **{**  **if((q->rear+1)%MAXSIZE==q->front)//队列满**  **{**  **printf("full\n");**  **return 0;**  **}**  **q->data[q->rear]=e;//将元素e赋值给队尾**  **q->rear=(q->rear+1)%MAXSIZE;**  **}**  **//出队**  **void removeQueue(Queue \*q,ElemType \*e)**  **{**  **if(q->front==q->rear)//队列空**  **{**  **printf("empty\n");**  **return 0;**  **}**  **\*e=q->data[q->front];//将队头元素赋值给e**  **q->front=(q->front+1)%MAXSIZE;**  **}**  **QQ图片20210425161253**  **本次作业由姜来同学负责分配任务，进行框架的梳理，编写结构体，并负责完成整个项目，而后童明安，韦清秀对代码进行注释，分析，理解，然后进行进一步分析，寻找不足之出，并加以完善。**  **错误：**   1. **两个对列，判断条件不完整** 2. **输出不够完善**   **QQ图片20210425161959**  **QQ图片20210425162752**  **两个对列如果只判断其中一个队列头尾指针不相等，最终的结果如上图所示，总会出现一个队列为空的现象，导致结果不是我们想要的，这是一点小失误！**  **QQ图片20210425163535**  **QQ图片20210425160927**  **看到光标一直闪烁，然后不给出结果，这是因为没有加输出进行调试，导致两组数据放到了一起，这是一个程序不完美的表现，没有一个好的代码习惯有时候会给人误解！**  **本次作业小结：**  **本小组对队列的一个代码实现比较陌生，导致整个作业耗时较长，不能完全懂代码，个人没有团队的意识，或者说团队意识不强，长此以往作业将会越来越差。这是一个不好的兆头。所以得有创新模式，总是用保守的方法解决问题，没有注入新鲜的血液进行循环，到时候总会漏洞百出，不利于个人的发展，更不利于一个团队的发展！**  **就本代码而言，巧妙的调用入栈出栈的函数，减少代码量，逻辑思维比较清楚；明确队头和队尾有一个指针是虚拟的指针，从而入栈出栈时赋值和申请空间的位置不会弄反；输出的时候保证末尾没有多余的空格，这是一个良好的习惯，长此以往下去，我们代码会越来越强，作业时间也会缩短！** | | | | |