**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 | 吴警 | | | |
| 专 业 | 软件工程 | | 年 级 | 2019 |
| 班 级 |  | | 实验时间 | 第5周 |
| 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。**  **源代码：**  **main.c：**  **#include <stdio.h>**  **#include <stdlib.h>**  **#include "queue.h"**  **int main(int argc, char \*argv[]) {**  **Queue A,B;//创建A,B队列模拟银行窗口**  **initQueue(&A);//初始化A,B队列**  **initQueue(&B);**  **int n,x,i,t;**  **int result[10100];//存放已办理序号**  **printf("num:");**  **scanf("%d",&n);//输入办理人数**  **for(i=1;i<=n;i++){**  **scanf("%d",&x);//输入办理序号**  **if(x%2){**  **inQueue(&A,x);//奇数号去A队列**  **}else{**  **inQueue(&B,x);//偶数号去B队列**  **}**  **}**  **i=1;**  **while(A.front!=A.rear&&B.front!=B.rear){//判断队列A，队列B是否为空**  **//将办理好的顾客序号存入数组**  **delQueue(&A,&t);**  **result[i++]=t;**  **if(A.front!=A.rear){**  **delQueue(&A,&t);**  **result[i++]=t;**  **}**  **delQueue(&B,&t);**  **result[i++]=t;**  **}**  **while(A.front!=A.rear){//判断队列A是否为空**  **delQueue(&A,&t);**  **result[i++]=t;//将办理好的顾客序号存入数组**  **}**  **while(B.front!=B.rear){//判断队列B是否为空**  **delQueue(&B,&t);**  **result[i++]=t;//将办理好的顾客序号存入数组**  **}**    **printf("%d",result[1]);**  **for(i=2;i<=n;i++){**  **printf(" %d",result[i]);//遍历输出数组**  **}**  **return 0;**  **}**  **队列结构体及队列相关函数：**  **queue.h**  **//队列结构体**  **#define MAXSIZE 10000**  **typedef int ElemType;**  **struct Queue{**  **ElemType data[MAXSIZE];**  **int front;//队头**  **int rear;//队尾**  **};**  **typedef struct Queue Queue;**  **//初始化队列**  **void initQueue(Queue \*q){**  **q->front=-1;**  **q->rear=-1;**  **}**  **//循环队列长度**  **int getSqLength(Queue Q){**  **return (Q.rear-Q.front+MAXSIZE)%MAXSIZE;**  **}**  **//入队操作，插入**  **void inQueue(Queue \*q,ElemType e){**  **if((q->rear+1)%MAXSIZE==q->front){//判断队列是否满**  **printf("full\n");**  **return;**  **}**  **q->rear=(q->rear+1)%MAXSIZE;**  **q->data[q->rear]=e;//元素e赋给队尾**  **}**  **//出队操作，删除**  **void delQueue(Queue \*q,ElemType \*e){**  **if(q->front==q->rear){//判断队列是否为空**  **printf("empty\n");**  **return;**  **}**  **q->front=(q->front+1)%MAXSIZE;**  **\*e=q->data[q->front];//将队头元素赋给e**  **}**  **运行结果：**    **实验过程中存在的问题：**  **·有一组测试结果存在错误。**  **·实验过程变量名称不一致导致错误。**  **改进措施：**  **·认真检查变量名称。**  **·对错误的实验结果认真思考。**  **收获：**  **·了解了队列的特征：队尾插入，队头删除。**  **·会对队列进行初始化操作。**  **·学习了循环队列相关知识，会利用取余的逻辑将队列循环。** | | | | |