《数据结构》课程实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 专业名称 | 计算机 | 年级 | 2017 | 班级 | 2 |
| 学生姓名 | 王汝芸 | 学号 | 201711010202 | 指导教师 | 郑志华 |
| 实验题目 | 实验3 队列的顺序表示和实现 | | | 提交时间 | 2018.4.25 |

一、实验目的和要求

例如：（1）熟悉C语言的上机环境，进一步掌握C语言的结构特点。

（2）掌握队列的顺序表示和实现

二、实验环境

Visual Studio2017

Windows 10

三、实验内容及实施

采用数组储存；宏定义N为队列长度，方便后期修改；

**实验3：队列的顺序表示和实现**

要求：构建一个循环队列, 实现下列操作

1、初始化队列(清空)；

2、入队；

3、出队；

4、求队列长度；

5、判断队列是否为空；

**模块图 略**

**【源程序】**

/\* 构建一个循环队列, 实现下列操作

1、初始化队列(清空)；

2、入队；

3、出队;

4、求队列长度；

5、判断队列是否为空；

\*/

#include<stdio.h>

#include<stdlib.h>

#define QUEUE\_SIZE 10

typedef struct queue

{

int front;

int rear;//rear指针指向队尾元素的下一个位置,当front等于rear时，此队列不是还剩一个元素，而是空队列

}Queue;

//求队列长度

int countLength(int \*a, int rear, int front)

{

int length = 0;

length = (rear - front + QUEUE\_SIZE) % QUEUE\_SIZE;

return length;

}

//入队

int enterQueue(int \* a, int rear, int front)

{

printf("----入队操作----\n");

int length;//计算空余位置个数

length = countLength(a, rear, front);

printf("\t队列还可输入%d个元素\n",QUEUE\_SIZE-length-1);

int n;//入队元素个数

printf("\t请输入入队元素个数：");

scanf\_s("%d", &n);

if (n < QUEUE\_SIZE - length-1)//判断队列未满,执行操作

{

printf("\t当前队列未满,请输入%d个整数\n",n);

for (int i = 0; i < n; i++)//入队

{

printf("\t请输入元素%d：", i + 1);

scanf\_s("%d", &a[rear%QUEUE\_SIZE]);//入队

rear = (rear + 1) % QUEUE\_SIZE;//rear后移

}

}

else

{

printf("----队满\n");

}

//打印入队操作完成后队列

printf("\t入队后队列为:");

for (int i = 0; i < QUEUE\_SIZE; i++)

{

if (a[i] == NULL)

{

continue;

}

else

{

printf(" %d", a[i]);

}

}

return rear;//入队只改变rear

putchar('\n');

}

//初始化队列

void clrarQueue(int \*a, int rear, int front)

{

printf("----程序开始前初始化队列----\n");

for (int i = 0; i < QUEUE\_SIZE; i++)

{

a[i] = NULL;

}

front = rear = 0;

putchar('\n');

}

//出队

int deleteQueue(int \*a, int rear, int front)

{

printf("\n----出队操作----\n");

int n;

printf("\t请输入出队元素个数：");

scanf\_s("%d", &n);

for (int i = 0; i < n; i++)

{

printf("\t%d出队 ", a[front]);

a[front] = NULL;

front++;

}

//打印出队完成后的队列

printf("\n\t出队后队列为：");

for (int i = 0; i < QUEUE\_SIZE; i++)

{

if (a[i] == NULL)

{

continue;

}

printf(" %d", a[i]);

}

putchar('\n');

return front;

}

//判断队列是否为空，若不为空，输出队列长度

void judgeVoid(int \*a, int rear, int front)

{

printf("\n----判断队列是否为空----\n");

if (rear == front)

{

printf("\t队列为空\n");

}

else

{

int length;

length = countLength(a, rear, front);

printf("\t队列不为空\n\t队列长度为：%d\n",length);

}

}

int main()

{

printf("----队列有%d个空间,可储存%d个元素----\n",QUEUE\_SIZE,QUEUE\_SIZE-1);

int a[QUEUE\_SIZE] = { NULL };

Queue \*p = (Queue\*)malloc(sizeof(Queue));

p->front = p->rear = 0;

putchar('\n');

//初始化队列

clrarQueue(a, p->rear, p->front);

p->front = p->rear = 0;

//入队

p->rear = enterQueue(a, p->rear, p->front);

printf("\n");

//出队操作

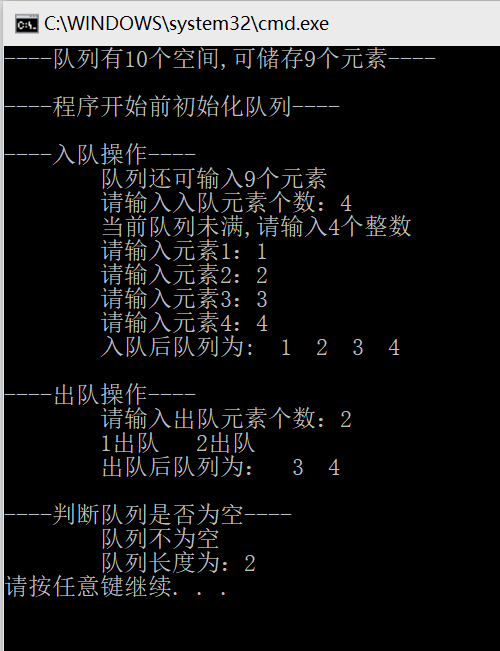
p->front = deleteQueue(a, p->rear, p->front);

//判断队列是否为空，若不为空，输出队列长度

judgeVoid(a,p->rear,p->front);

return 0;

}

**四、实验结果 (程序的执行结果)**

**五、实验讨论（可选）**

无