**实 验 报 告**

|  |  |  |
| --- | --- | --- |
| **课程名称**：数据结构 | **班级**：软工21101 | **实验成绩**： |
| **实验名称**：队列的实现与应用 | **学号**：2105006207 | **批阅教师签字：董傲霜** |
| **实验编号**：实验六 | **姓名**：方福涛 | **实验日期：**2022 年 10 月 21日 |
| **指导教师**：董傲霜 | **组号**：wuyanzu | **实验时间**： 时 分－ 时 分 |

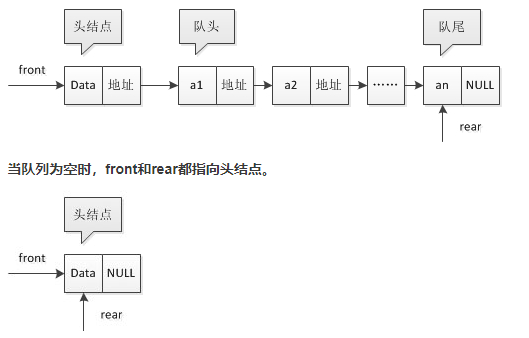
**一、实验设计思想**

1. 说明队列的抽象数据类型的特征；

（1） 新数据项的添加总发生在一端，通常称为“尾rear端”

(2) 存数据的一处总发生在另一端，通常称为“首front端”

1. 说明你想采用的队列的存储结构特点。



**二、程序说明**

1. 给出队列的存储结构定义及其属性含义的说明；

typedef int QElemType;

typedef struct QNode /\* 结点结构 \*/

{

QElemType data;

struct QNode \*next;

} QNode;

typedef struct /\* 队列的链表结构 \*/

{

QNode \*front; /\* 队头、队尾指针 \*/

QNode \*rear;

} LinkQueue;

1. 给出你设计的函数的结构说明（函数，参数，执行的结果）；

// 构造一个空队列q

LinkQueue \*InitQueue(LinkQueue \*q)

// 元素入队

LinkQueue \*EnQueue(LinkQueue \*q, QElemType e)

//元素出队

LinkQueue \*DeQueue(LinkQueue \*q)

//返回队头元素

void GetQHead(LinkQueue \*q)

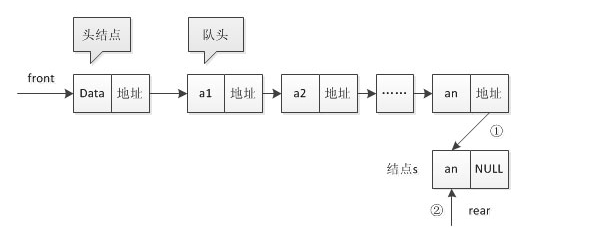
//求队列长度

void QueueLength(LinkQueue \*q)

//打印。带头结点，真正存储元素的位置从头结点下一位置（队头）开始！！！

void PrintQueue(LinkQueue \*q)

1. 说明入队列和出队列实现步骤（处理步骤，可用文字也可画流程图）。



**三、实验环境**

Windows 10、vscode

**四、实验过程分析**

实验调试过程中的问题及改正的事件举例说明（只要调程序就不可能没有错误，现象、原因、修正方法，可以截图但不能过多）。

总结易错点：

1.在用顺序及结构存储时，栈的大小是可以改变的，而循环队列的大小一般是不能改变的。  
 2.栈的特点是后进先出，但是并不是一组数据通过栈后都会逆置，只有在一次性进栈再出栈后才是，而队列无论什么过程都不会改变数据元素的顺序

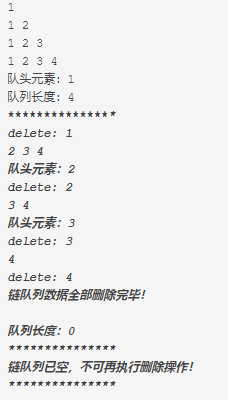
**五、实验结果总结**

1、阐述队列的入队列和出队列操作的算法时间复杂度和空间复杂度。

入队列的算法时间复杂度：O(1)、空间复杂度：O(1)

出队列的算法时间复杂度: O(1)、空间复杂度：O(1)

2、给出你的测试方法和测试截图（不能超过三个截图）；



**六、附录**

1. 意见和建议（没有可不写）。

2. 思考题：

回答以下问题：

1. 循环队列的队列空和满的标志是什么？

**当队头和队尾指针在同一位置时，队空。**

**队头指针在队尾指针的下一位置时，队满。**

1. 一个队列的循环队列设计时，数组空间的大小为MaxSize，当队列为满时队列中存储了多少个数据元素？

MAXSIZE - 1

**七、打分表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目 | 考核点 | 分数 | **得分** | 备注 |
| 程序 | 逻辑是否正确  程序可读性  创新点 | 50 | **0** |  |
| 报告完整性 | 实验过程阐述是否完整  测试数据设计是否合理  运行结果是否正确 | 40 | **0** |  |
| 调试问题及解决方法 | 是否对调试过程问题进行阐述 | 5 | **0** |  |
| 思考题目 | 回答是否正确 | 5 | **0** |  |
| 合计 |  | 100 | **0** |  |