

《数据结构》上机报告

2018 年 10 月 31 日

姓名：张天然 学号：1751237 班级：计 2 得分：

试验题目	循环队列和链队列
实验目的	理解顺序队列的逻辑结构和存储结构，熟练掌握顺序队列的相关操作 理解链队列的逻辑结构和存储结构，熟练掌握链队列的相关操作
问题描述	略
基本要求	定义顺序队列类型，实现队列的所有基本操作。 定义链队列类型，实现队列的所有基本操作。
数据结构设计	循环队列 private: QElemType *base; int front; int rear; 链队列 struct QNode{ QElemType data; QNode *next; }; private: QNode *front; QNode *rear;
功能(函数)说明	循环队列 LinkQueue1(); //构造函数 ~LinkQueue1(); //析构函数 int ClearQueue(); //清空队列 bool QueueEmpty(); //判断队列是否为空 bool QueueFull(); //判断队列是否为满 int QueueLength(); //返回队列长度 int EnQueue(QElemType e); //e入队 int DeQueue(QElemType &e); //用e返回出队元素 void display(); //遍历队列 链队列 LinkQueue(); //构造函数 ~LinkQueue(); //析构函数 int ClearQueue(); //清空队列 bool QueueEmpty(); //判断队列是否为空 int QueueLength(); //返回队列长度 int GetHead(QElemType &e); //返回队头元素 int EnQueue(QElemType e); //e入队

	<pre>int DeQueue(QElemType &e); //用e返回出队元素 void display(); //遍历队列</pre>
界面设计和使用说明	以 c++为开发语言，在 Visual Studio 2012 编译器上实现。 界面上显示功能菜单，根据提示按正确格式输入数据。
调试分析	
心得体会	<p>本次实验比较简单，主要是实现对队列的基本操作。实现循环队列时，需要特别注意 rear、front 和 MAXQSIZE 的关系。</p> <p>重要算法分析：</p> <pre>bool QueueFull() { if((rear+1)%MAXQSIZE==front) return 1; else return 0; } bool QueueEmpty() { if(rear==front)return 1; else return 0; }</pre> <p>这两个函数包含了循环队列中的重要算法，用来判断循环队列是否为满或空。这里少用了一个元素空间。</p>