

# 《数据结构》上机报告

2018 年 11 月 28 日

姓名：张天然 学号：1751237 班级：计 2 得分：

试验题目	静态链表
问题描述	静态链表（或称为数组链表），用一个一维数组 S 存储链表的所有结点。每个结点包含 data 和 cur 两个域，data 存储数据,cur(称作游标, int 类型)代替指针，存储链表的下一个结点在数组中的相对位置。数组前两个 s[0]存放链表头指针,s[1]存放闲置链表头指针；S[0].cur 初始为-1，表示空表。S[1].cur 初始为 2，表示第 1 个可分配的结点位置。S[MaxSize-1]=-1 表示无后继结点。初始 S[i].cur=i+1，表示所有未分配结点已形成备用链表，每当进行插入时，从备用链表中取得第 1 个结点作为待插入的新结点，删除时则将从链表中删除的结点链接到备用链表中。
基本要求	本题练习用静态链表实现线性表的基本操作，包括表的初始化，尾部追加一个元素、第 i 个元素位置前插入一个新的元素、删除第 i 个元素、查找某元素、判表空、判表满、表的遍历。元素数据类型是字符串。
数据结构设计	<pre>private:     struct LNode{         string data;         int cur;     };     LNode *s1s;     int maxlen;</pre>
功能(函数)说明	<pre>List();           // 构造函数 List(int size);    // 含参构造函数 - 创建最大长度为 size 的静态链表 int Size() const;  // 返回链表的长度 bool IsFull() const; // 返回链表是否已满 bool IsEmpty() const; // 返回链表是否已空 void Clear();      // 清空链表 void Print();      // 输出链表 int Retrieve( int position, string &amp;x ); // 获取链表第 position 位置的元素到 x，成功返回 0，否则返回 -1 int Replace( int position, const string &amp;x ); // 将链表第 position 位置的元素换为 x，成功返回 0，否则返回 -1 int Remove( int position, string &amp;x ); // 获取并删除链表第 position 位置的元素，成功返回 0，否则返回 -1 int Insert( int position, const string &amp;x ); // 将元素 x 插入到链表第 position 位置，成功返回 0，否则返回 -1</pre>

	<pre>int Append(const string &amp;x);           // 将元素 x 追加到链表尾部，成功返回 0，否则返回 -1</pre>
界面设计和使用说明	<p>以 c++为开发语言，在 Visual Studio 2012 编译器上实现。</p> <p>界面上显示执行简单测试程序后的结果。</p>
调试分析	<pre>9 5 jan feb mar apr may 0 : N/A : 2          1 : N/A : 7          2 : jan : 3 3 : feb : 4          4 : mar : 5          5 : apr : 6 6 : may : -1         7 : N/A : 8          8 : N/A : -1 2 jun jan jun feb mar apr may 1 jan oct -1 oct jun feb mar apr may oct 0 : N/A : 7          1 : N/A : 8          2 : oct : -1 3 : feb : 4          4 : mar : 5          5 : apr : 6 6 : may : 2          7 : jun : 3          8 : N/A : -1 请按任意键继续. . .</pre>
心得体会	<p>总结：</p> <p>静态链表相比于动态链表需要预先分配一个较大的空间，但仍保留了链式存储结构的主要优点。</p> <p>重要算法：</p> <pre>List::List(int size){     int i;     sls=new LNode[size+2];     sls[0].cur=sls[1].cur=-1;     sls[0].data="N/A";     for(i=1;i&lt;size+2;i++){         sls[i].cur=i+1;         sls[i].data="N/A";     }     sls[size+1].cur=-1;</pre>

	<pre>maxlen=size; }</pre> <p>此为建立长度为 <code>size</code> 的静态链表的函数，其中静态链表的逻辑关系非常重要。</p> <p>时间复杂度：O(size) 空间复杂度：O(size)</p>
--	---