

《数据结构课程实践》

**实验报告**

**实验名称：单链表**

**姓 名：唐子辉**

**学 号：19081630**

**专 业：通信工程**

**实验时间：2020年4月9日**

**杭州电子科技大学**

**通信工程学院**

**（实验报告要求：填写三四五六一共四部分内容，实验报告文件命名方式：*数据结构实验报告1-姓名-学号*，完成后上传电子版到校园网的网络教学平台，无需打印）**

1. **实验目的**

掌握顺序表的基本算法设计。

1. **实验内容**
2. 完成《数据结构教程第5版》（或《上机指导》）上机实验题2.2：实现单链表各种基本运算的算法；

\*要求：对照代码后的注释，理解程序

1. 在main()函数中，调用头插法（CreateListF()函数）和尾插法（CreateListR()函数），创建新链表，并输出结果进行比较；
2. （附加题）对任务1或者2中创建的某一个单链表{A1,B1,A2,B2,...,An,Bn}，编写一个算法将它拆分为两个单链表，分别为：{A1,A2,...,An}和{Bn,...,B2,B1}。\*要求：算法时间复杂度为O(n)，空间复杂度为O(1)。
3. **算法设计**
4. 实验内容2中，在main()函数中调用基本运算CreateListF()和CreateListR()创建单链表；

*int* main() {  
 LinkNode \*list;  
 ElemType e;  
 initList(list);  
 insertElem(list,1,'a');  
 insertElem(list,2,'b');  
 insertElem(list,3,'c');  
 insertElem(list,4,'d');  
 insertElem(list,5,'e');  
 dispList(list);  
 printf("the length of list : %d\n",getListLength(list));  
 printf("list %s\n",(isEmpty(list)?"is empty":"is not empty"));  
 getElemValue(list,3,e);  
 printf("the third elem in list is : %c\n",e);  
 printf("a locates :%d\n",getElemNumber(list,'a'));  
 insertElem(list,4,'f');  
 dispList(list);  
 destroyList(list);  
 printf("-------------附加题1--------------\n");  
 LinkNode \*Flist;  
 *char* a[5]={'q','w','e','r','t'};  
 LinkNode \*Rlist;  
 creatListF(Flist,a,5);  
 dispList(Flist);  
 creatListR(Rlist,a,5);  
 dispList(Rlist);  
 printf("-------------附加题2--------------\n");  
 dispList2(devideList(Rlist));  
 dispList(Rlist);  
}

头插法和尾插法创建链表都用了同一个字符数组：

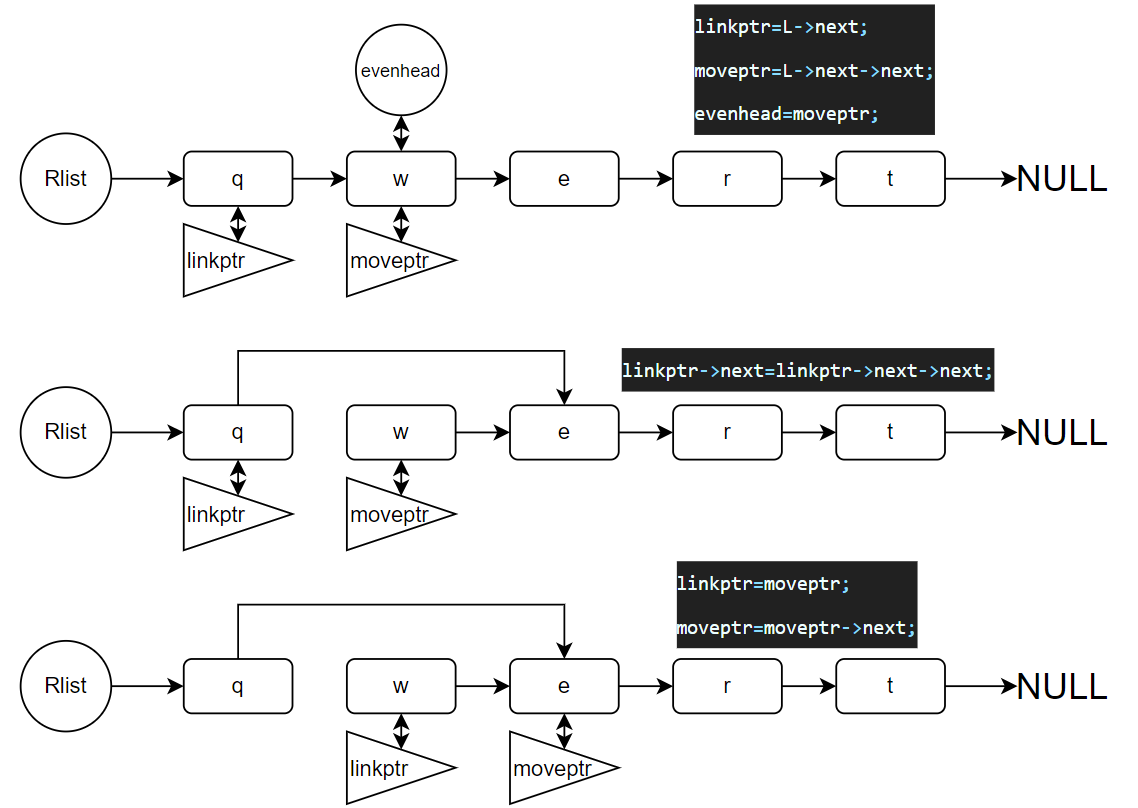
*char* a[5]={'q','w','e','r','t'};

结果可看到两种方法数据顺序是相反的。

1. 给出附加题的算法代码，并画示意图表示算法思想。

示意图：

单箭头表示该指针的next指向，而双向箭头表示赋值表示两者是相等的而不是说指针指向它。



创建返回节点的函数：

LinkNode\* devideList(LinkNode \*&L){  
 LinkNode \*linkptr,\*moveptr,\*evenhead;  
 linkptr=L->next;  
 moveptr=L->next->next;  
 evenhead=moveptr;  
 *while* (moveptr!=**NULL**)  
 {  
 linkptr->next=linkptr->next->next;  
 linkptr=moveptr;  
 moveptr=moveptr->next;  
 }  
 *return* evenhead;  
}

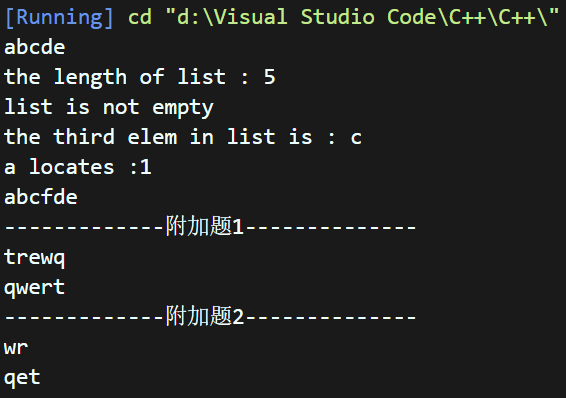
更改打印列表的函数（书本上输出函数会造成跳一个值打印到另外一个链表去了）：

*void* dispList2(LinkNode \*L){  
 LinkNode \*p=L;  
 *while* (p!=**NULL**)  
 {  
 printf("%c",p->data);  
 p=p->next;  
 }  
 printf("\n");  
}

调用函数（用尾插法得到的链表作为示例）：

dispList2(devideList(Rlist));  
dispList(Rlist);

1. **运行结果与分析（给出运行结果截图，如有必要，请加以简要说明）**



1. **思考题**
2. 顺序表和单链表的类型声明各有什么特点？

顺序表类型包括值数组（预设长度）和顺序表长度，单链表类型包括值域和指向下一个结点的指针域。

2. 顺序表和单链表都有ListInsert()函数，请根据代码说明两者不同之处。

顺序表插值可以利用索引值把插入点后的数据后移一位，然后在插值，并把长度加1；

单链表插值要为插入结点创建内存空间，并只能从头节点遍历到插入点的前一个，再把结点插入进去。

1. **实验小结（记录实验过程中遇到的主要问题和心得）**

附加题二如果打印链表用书上那个代码的话是不行的，因为那个是从传入的结点的后继节点开始打印，而题目链表是跨了一个结点连接的，就算把evenhead设置成前一个也会变成打印奇数列表，所以要自己写一个dispList2函数。也很简单，把L->next改成L就行了。