**第二章作业**

**第8周周一前提交电子档，由学习委员收齐打包给助教。**

**作业文件命名方式统一要求为：CS班号\_学号\_姓名\_2**

2.2 填空题

（1） 在顺序表中插入或删除一个元素，需要平均移动\_\_\_\_\_\_\_n/2\_\_\_\_\_元素，具体移动的元素个数与\_\_\_\_\_\_元素位置\_\_\_\_\_\_有关。

（2） 顺序表中逻辑上相邻的元素的物理位置\_\_\_\_是\_\_\_\_\_\_\_\_紧邻。 单链表中逻辑上相邻的元素的物理位置\_\_\_\_\_不一定\_\_\_\_\_\_\_紧邻。

（3）不带头结点的单链表中，除了首结点外，任一结点的存储位置由\_\_\_\_\_上一节点的指针域\_\_\_\_\_\_\_指示。

（4） 在单链表中设置头结点的作用是\_\_\_\_删除插入操作更加方便；统一空表和非空表的处理\_\_\_\_\_\_\_\_\_\_\_\_\_。

2.7 已知 L 是带表头结点的非空单链表，且 P 结点既不是首元结点，也不是尾元结点，试从下列提供的答案中选择合适的语句序列。

a. 删除 P 结点的直接后继结点的语句序列是\_\_11,3,14\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

b. 删除 P 结点的直接前驱结点的语句序列是\_\_\_\_\_10,12,8,11,3,14\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

c. 删除 P 结点的语句序列是\_\_\_\_10,12,7,3,14\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

d. 删除首元结点的语句序列是\_\_\_12,11,3,14\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

e. 删除尾元结点的语句序列是\_\_\_\_9,11,3,14\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

(1) P=P->next;

(2) P->next=P;

(3) P->next=P->next->next;

(4) P=P->next->next;

(5) while(P!=NULL) P=P->next;

(6) while(Q->next!=NULL) { P=Q; Q=Q->next; }

(7) while(P->next!=Q) P=P->next;

(8) while(P->next->next!=Q) P=P->next;

(9) while(P->next->next!=NULL) P=P->next;

(10) Q=P;

(11) Q=P->next;

(12) P=L;

(13) L=L->next;

(14) free(Q);

2.11 已知顺序表L中的数据元素递增有序，数据元素类型为int。相关定义如下：

#define LIST\_INIT\_SIZE 20

#define LISTINCREMENT 10

typedef int ElemType;

typedef struct

{ ElemType \*elem;

int length;

int listsize;

} SqList;

试写一算法，将x插入到顺序表的适当位置上，以保持该表的有序性。函数原型：

int insert(SqList &L,ElemType x);

注：严格按此函数原型，完成函数的定义。自行编写主函数等完成程序调试，作业提交时只需要（复制粘贴）提交函数insert即可，不需要其它部分代码。由助教在平台上测试函数的正确性，编译不通过不得分。部分正确按通过测试集的比例给出部分分。

**int** insert(SqList &L, ElemType x) {  
 **if** (L.length >= L.listsize) {  
 ElemType \*newbase;  
 newbase = (ElemType \*) realloc(L.elem, (L.listsize + **LISTINCREMENT**) \* **sizeof**(ElemType));  
 **if** (newbase == **NULL**) {  
 **return** 0;  
 }  
 L.elem = newbase;  
 L.listsize = L.listsize + **LISTINCREMENT**;  
 }  
 ElemType i, j, tmp1, tmp2;  
 **for** (i = 0; i < L.length; i++) {  
 **if** (x < L.elem[i]) {  
 tmp1 = L.elem[i];  
 L.elem[i] = x;  
 **for** (j = i + 1; j <= L.length; j++) {  
 tmp2 = L.elem[j];  
 L.elem[j] = tmp1;  
 tmp1 = tmp2;  
 }  
 **break**;  
 }  
 }  
 **if** (i==L.length)  
 L.elem[L.length]=x;  
 L.length += 1;  
 **return** 1;  
}

2.37 设以带头结点的双向循环链表表示的线性表L=（a1，a2，...，an），试编写一个时间复杂度为O(n)的算法,将L改编成L=（a1，a3，...，an，...，a4，a2）。

相关定义如下：

typedef int ElemType;

typedef struct Dnode

{

ElemType data;

struct Dnode \*prior,\*next;

} \*DuLinkList;

试写出算法函数。函数原型：

void adjust(DuLinkList L);

注：严格按此函数原型，完成函数的定义。自行编写主函数等完成程序调试，作业提交时只需要（复制粘贴）提交函数adjust即可，不需要其它部分代码。

**void** adjust(DuLinkList L) {  
 **int** i=0;  
 DuLinkList p,q,s,r;  
 q=L;  
 p=L;  
 p=p->next;  
 r=p;  
 **while** (p->next!=L){  
 s=p->next;  
 i++;  
 **if**(i%2==0){  
 q->prior=p;  
 p->next=q;  
 q=q->prior;  
 p=s;  
 r->next=s;  
 s->prior=r;  
 r=r->next;  
 } **else** p=p->next;  
 }  
 p->next=q;  
 q->prior=p;  
}