## 2.7 习题

### 2.7.1知识点：线性表的逻辑结构

**一、选择题**

1① 线性表L=（a, a,…,a）,下列说法正确的是 （D ）。

A．每个元素都有一个直接前驱和一个直接后继。

B．线性表中至少要有一个元素。

C．表中诸元素的排列顺序必须是由小到大或由大到小。

D．除第一个和最后一个元素外,其余每个元素都有一个且仅有一个直接前驱和直接后继。

2① 在线性表的下列运算中，不改变数据元素之间结构关系的运算是（ D ）。  
A．插入 B．删除  
C．排序 D．定位

3① 线性表是具有n 个（C ）的有限序列（n>0）。【清华大学 1998】

A．表元素 B．字符 C．数据元素 D．数据项 E．信息项

**二、判断题**

（ T ）1① 线性表中的每个结点最多只有一个前驱和一个后继。

（ F ）2① 线性表中的每个结点都至少有一个前驱结点和后继结点。

（ F ）3① 线性表是N个数的有限序列。

（ F）4① 同一线性表的数据元素可以具有不同的特性。

（ T ）5① 线性表的长度n就是表中数据元素的个数，当n=0时，称为空表。

（ T ）6① 线性表是一个相当灵活的数据结构，它的长度可根据需要增长或缩短。

（ F ）7① 对线性表中的数据元素只能进行访问，不能进行插入和删除操作。

### 2.7.2知识点：线性表的顺序存储结构

**一、选择题**

1① 在一个长度为n的顺序表中，在第i个元素（1 <= i <=n+1）之前插入一个新元素时需向后移动（ B ）个元素．

A．n-1 B．n-i+1 C．n-i-1 D．i

2① 若某线性表中最常用的操作是取第i个元素和找第i个元素的前趋元素，则采用（ D ）存储方式最节省时间。

A．单链表 B．双链表 C．单向循环 D．顺序表

3② 一个数组第一个元素的存储地址是100，每个元素的长度为2，则第5个元素的地址是（ B ）

A．110 B．108 C．100 D．120

4① 下述哪一条是顺序存储结构的优点（ A ）。【北方交通大学 2001】

A．存储密度大 B．插入运算方便

C．删除运算方便 D．可方便地用于各种逻辑结构的存储表示

5③ 若长度为n 的线性表采用顺序存储结构，在其第i 个位置插入一个新元素的算法的时间复杂度为（ C ）（1<=i<=n+1）。【北京航空航天大学 1999】

A．O（0） B．O（1） C．O（n） D．O（n）

6③ 对于顺序存储的线性表，访问结点和增加、删除结点的时间复杂度为（ C ）。【青岛大学 2000】

A．O（n） O（n） B．O（n） O（1）

C．O（1） O（n） D．O（1） O（1）

**二、填空题**

1① 线性表的顺序存储的缺点是在任意位置上\_\_\_插入\_\_\_\_\_数据与\_\_\_\_删除\_\_\_\_\_数据费时间。

2① 设一线性表的顺序存储，总存储容量为M，其元素存储位置的范围为\_\_0~M-1\_\_\_\_\_\_\_\_\_\_。

3① 向一个长度为n的向量中删除第i个元素（1≤i≤n）时，需向前移动\_\_\_\_n-i\_\_\_\_\_\_个元素。

**三、简答题**

1③ 已知线性表的存储结构为顺序表，阅读下列算法，并回答问题：

void f30 （SeqList \*L） {

int i，j;

for （i=j=0;i<L->length; i++）

if（L->data[i]>=0）{

if（i!=j）L->data[j]=L->data[i];

j++；

}

L->length=j;

}

（1）设线性表L=（21，-7，-8，19，0，-11，34，30，-10），写出执行f30（&L）后的L状态；(21,19,0,34,30)

（2）简述算法f30的功能。删除顺序表中小于0的元素

**四、编程题**

1④ 已知顺序表La中数据元素按非递减有序排列。试写一个算法，将x插入到La的合适位置上，保持该表的有序性。

### 2.7.3知识点：线性表的链式存储结构

**一、选择题**

1① 链表是一种采用（ B ）存储结构存储的线性表。

A．顺序 B．链式 C．星式 D．网状

2① 链接存储的存储结构所占存储空间（ A ）。

A．分两部分，一部分存放结点值，另一部分存放表示结点间关系的指针。

B．只有一部分，存放结点值。

C．只有一部分，存储表示结点间关系的指针。

D．分两部分，一部分存放结点值，另一部分存放结点所占单元数。

3① 线性表若采用链式存储结构时，要求内存中可用存储单元的地址（ D ）。

A．必须是连续的 B．部分地址必须是连续的

C．一定是不连续的 D．连续或不连续都可以

4① 线性表Ｌ在（ B ）情况下适用于使用链式结构实现。

A．需经常修改Ｌ中的结点值 B．需不断对Ｌ进行删除插入

C．Ｌ中含有大量的结点 D．Ｌ中结点结构复杂

5① 对单链表表示法,以下说法错误的是（C ）。

A．数据域用于存储线性表的一个数据元素。

B．指针域（或链域）用于存放一个指向本结点所含数据元素的直接后继所在结点的指针。

C．所有数据通过指针的链接而组织成单链表。

D．NULL称为空指针，它不指向任何结点只起标志作用。

6① 以下说法正确的是（D ）。

A．顺序存储方式的优点是存储密度大且插入、删除运算效率高

B．链表的每个结点中都恰好包含一个指针

C．线性表的顺序存储结构优于链式存储结构

D．顺序存储结构属于静态结构而链式结构属于动态结构

7① 以下说法错误的是（D ）。

A．求表长、定位这两种运算在采用顺序存储结构时实现的效率不比采用链式存储结构时实现的效率低

B．顺序存储的线性表可以随机存取

C．由于顺序存储要求连续的存储区域,所以在存储管理上不够灵活

D．线性表的链式存储结构优于顺序存储结构

8① 不带头结点的单链表head为空的判定条件是（ A ）。

A．head= =NULL B．head->next= =NULL

C．head->next= =head D．head!=NULL

9① 带头结点的单链表head为空的判定条件是（ B ）。

A．head= =NULL B．head->next= =NULL

C．head->next= =head D．head!=NULL

10② 在头指针为head的非空单循环链表中，指针p指向尾结点，下列关系成立的是（ A ）。

A．p->next= =head B．p->next->next= =head

C．p->next= =NULL D．p= =head

11② 在一个单链表中，已知q所指结点是p所指结点的前驱结点，若在q和p之间插入s结点，则执行语句（ C ）。

A．s->next=p->next;p->next=s; B．p->next=s->next;s->next=p;

C．q->next=s;s->next=p; D．p->next=s;s->next=q;

12② 在一个单链表中，若p所指结点不是最后结点，在p之后插入s结点，则应执行语句（ B ）。

A．s->next=p:p->next=s; B．s->next=p->next;p->next=s;

C．s->next=p->next;p=s; D．p->next=s;s->next=p;

13② 在一个单链表中，若删除p所指结点的后续结点，则应执行语句（ A ）。

A．p->next=p->next->next; B．p=p->next;p->next=p->next->next;

C．p->next=p->next; D．p=p->next->next;

14② 指针p、q和r依次指向某循环链表中三个相邻的结点，交换结点\*q和结点\*r在表中次序的程序段是（ A ）。  
 A．p->next=r； q->next=r->next； r->next=q；  
 B．p->next=r； r->next=q； q->next=r->next；  
 C．r->next=q； q->next=r->next； p->next=r；  
 D．r->next=q； p->next=r； q->next=r->next；

15① 链表不具有的特点是（ B ） 【福州大学 1998】

A．插入、删除不需要移动元素 B．可随机访问任一元素

C．不必事先估计存储空间 D．所需空间与线性长度成正比

16① 下面的叙述不正确的是（BC ）【南京理工大学 1996】

A．线性表在链式存储时，查找第i 个元素的时间同i 的值成正比

B．线性表在链式存储时，查找第i 个元素的时间同i 的值无关

C．线性表在顺序存储时，查找第i 个元素的时间同i 的值成正比

D．线性表在顺序存储时，查找第i 个元素的时间同i 的值无关

17① 下面关于线性表的叙述中，错误的是哪一个？（ B ）【北方交通大学 2001】

A．线性表采用顺序存储，必须占用一片连续的存储单元。

B．线性表采用顺序存储，便于进行插入和删除操作。

C．线性表采用链接存储，不必占用一片连续的存储单元。

D．线性表采用链接存储，便于插入和删除操作。

18① 在一个以 h 为头的单循环链中，p 指针指向链尾的条件是（A）【南京理工大学1998】

A．p->next=h B．p->next=NULL C．p->next->next=h D． p->data=-1

19② 若某线性表最常用的操作是存取任一指定序号的元素和在最后进行插入和删除运算，则利用（ A）存储方式最节省时间。【哈尔滨工业大学 2001】

A．顺序表 B．双链表 C．带头结点的双循环链表 D．单循环链表

20② 某线性表中最常用的操作是在最后一个元素之后插入一个元素和删除第一个元素，则采用（ D）存储方式最节省运算时间。【南开大学 2000】

A．单链表 B．仅有头指针的单循环链表 C．双链表 D．仅有尾指针的单循环链表

21② 设一个链表最常用的操作是在末尾插入结点和删除尾结点，则选用（ D ）最节省时间。【合肥工业大学 2000】

A．单链表 B．单循环链表 C．带尾指针的单循环链表 D．带头结点的双循环链表

22② 线性表（a, a,…,a）以链接方式存储时，访问第i 位置元素的时间复杂性为（C ）【中山大学 1999】

1. O（i） B．O（1） C．O（n） D．O（i-1）

23③ 完成在双循环链表结点p 之后插入s 的操作是（D ）。【北方交通大学 1999】

A．p->next:=s ; s->priou:=p; p->next->priou:=s ; s->next:=p->next;

B．p->next->priou:=s; p->next:=s; s->priou:=p; s->next:=p->next;

C．s->priou:=p; s->next:=p->next; p->next:=s; p->next->priou:=s ;

D．s->priou:=p; s->next:=p->next; p->next->priou:=s ; p->next:=s;

24③ 在双向循环链表中,在p 指针所指向的结点前插入一个指针q 所指向的新结点,其修改指针的操作是（ C ）。【北京邮电大学 1998】【青岛大学 2000】

注:双向链表的结点结构为（llink,data,rlink）。 供选择的答案：

A． p->llink=q； q->rlink=p； p->llink->rlink=q； q->llink=q；

B． p->llink=q； p->llink->rlink=q ； q->rlink= p； q->llink=p->llink；

C． q->rlink=p； q->llink=p->llink； p->llink->rlink=q; p->llink=q;

D． q->llink=p->llink；q->rlink:=p； p->llink=q；p->llink=q；

25③ 在双向链表存储结构中，删除p 所指的结点时需修改指针（ A）。【西安电子科技大学 1998】

A．p->prior->next=p->next p->next->prior=p->prior;

B．p->prior =p-> prior-> prior p-> prior-> next=p;

C．p-> prior-> prior:=p p-> next=p-> next -> next

D．p-> next =（p-> prior） -> prior p-> prior=p-> next-> next

**二、填空题**

1① 线性表的链式存储是用\_\_\_malloc\_\_\_\_\_\_语句实现空间单元动态分配。

2① 单链表是\_\_\_线性表\_\_\_\_\_的链接存储表示。

3① 头结点地址指针为L的循环单链表，空表的判别标志是\_\_\_L->next==NULL\_\_\_。

4① 在一个单链表中删除p所指结点时，应执行以下操作：

q=p->next;

p->data=p->next->data;

p->next=q->next\_\_\_\_\_\_\_\_\_;

free（q）;

5③ 下段程序的功能：有一头指针为head的链表,将new指针指向的节点插入到data域为7的节点的后边。将程序补充完整。

P = head;

while（P != NULL）

{ if （P ->data == 7）

/\*找到位置插入结点后跳出循环\*/

{ （1）\_new->next=p->next\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

（2） \_p->next=new\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

（3） \_\_\_\_\_\_break\_\_\_\_\_\_\_\_\_\_ ;}

else

（4）\_\_\_p=p->next\_\_\_\_\_\_\_\_\_\_\_\_\_\_; /\*指针后移\*/

}

if（P == NULL）

printf（“\n the position isn’t exist! ”）;

6③ 假设某个不设头指针的无头结点单向循环链表的长度大于1，s为指向链表中某个结点的指针。算法f 30的功能是，删除并返回链表中指针s所指结点的前驱。请在空缺处填入合适的内容，使其成为完整的算法。   
typedef struct node {  
 DataType data；   
 struct node \*next；   
}\*LinkList；   
DataType f 30（LinkList s） {  
 LinkList pre，p；   
 DataType e；   
 pre=s；   
 p=s->next；   
 while（（1）\_\_p->next!=s\_\_\_\_ ）{  
 pre=p;  
 （2）p=p->next\_\_\_\_\_\_\_\_\_\_\_\_ ;  
}  
 pre ->next=（3）\_\_p->next\_\_\_\_\_\_\_\_\_\_\_\_ ;  
 e=p->data;  
 free（p）;  
 return e;  
}

**三、判断题**

（F ）1① 单链表从任何一个结点出发，都能访问到所有结点。

（ F ）2① 线性表的每个结点只能是一个简单类型，而链表的每个结点可以是一个复杂类型。

**四、简答题**

1① 描述以下几个概念：顺序存储结构、链式存储结构、顺序表、有序表。

2② 描述以下三个概念的区别：头指针、头结点、首元结点。在单链表中设置头结点的作用是什么？

3② 线性表有两种存储结构：一是顺序表，二是链表，试问：

（1）如果有n个线性表同时共存，并且在处理过程中各表的长度会动态地发生变化，线性表的总数也会自动地改变。在此情况下，应选用哪种存储结构？为什么？链表

（2）若线性表的总数基本稳定，且很少进行插入和删除，但要求以最快的速度存取线性表中的元素，那么应采用哪种存取结构？为什么？顺序表

4③ 假设本测试中使用的链表如图2.45所示，结点定义如下：

struct List

{ int data;

struct List \*next;

};

typedef struct List Node;

typedef Node \*Link;

Link P,Q,R,S,head;

Link pointer,back,new;

对以下单链表分别执行下列程序段，要求分别画出结果图。

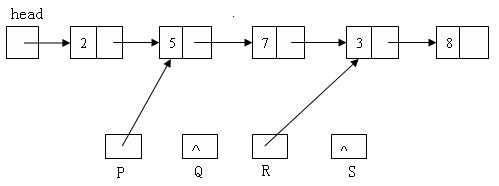


图2.45 4题图

1. Q=head->next->next;

Q指向7

（2）R->data=P->data;

3变5

（3）R->data=P->next->data;

3变7

（4）S=P;

while（S->next!=NULL）

{ S->data=S->data\*2; S=S->next;}

4 10 14 6 8

（5）S=P;

while（S!=NULL）

{ S->data=S->data\*2; S=S->next;}

4 10 14 6 16

5③ 假设本测试中使用的链表如图2.45所示，结点定义如第4题所示。画出执行如下程序段后各指针及链表的示意图。

head=（Link）malloc（sizeof（Node））;

head->data=0;

head->next=NULL;

P=head;

for（i=1;i<4;i++）

{ new=（Link）malloc（sizeof（Node））;

new->data=2\*i;

new->next=NULL;

P->next=new;

P=new;

}

创建了一个链表，数据元素为 0,2,4,6，并且p和new都指向尾结点

6③ 有一链表如下图所示，阅读程序给出程序的输出结果。

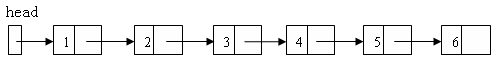


图2.46 6题图

P = head;

while（P != NULL）

{

printf（“\n data=%d”, P -> data）;

P = P->next;

if（ P != NULL）

P = P ->next;

}

Data=1

Data=3

Data=5

**五、编程题**

1④ 一个单链表，其头指针为head，编写一个函数计算数据域为x的节点个数。

2④ 已知单链表La中数据元素按非递减有序排列。按两种不同情况，分别写出算法，将元素x插入到La的合适位置上，保持该表的有序性：（1）La带头结点；（2）La不带头结点。

3④ 试写一个算法，实现顺序表的就地逆置，即在原表的存储空间将线性表（a,a,…,a,a）逆置为（a,a,…, a,a）。

4④ 设计一个算法，求A和B两个单链表表示的集合的交集、并集合差集。