一．是非题

1. 数据结构可用三元式表示（D，S，P）。其中：D是数据对象，S是D上的关系，P是对

D的基本操作集。()

2 简单地说,数据结构是带有结构的数据元素的集合。()

3 判断带头结点的非空循环单链表（头指针为L）中指针p所指结点是最后一个元素结点

的条件是：p->next==L。()

4 线性表的链式存储结构具有可直接存取表中任一元素的优点。()

5 线性表的顺序存储结构优于链式存储结构。()

6. 在单链表P指针所指结点之后插入S结点的操作是：

P->next= S ; S-> next = P->next;。()

7 对于插入、删除而言，线性表的链式存储优于顺序存储。()

8. 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。()

9. 栈和队列是操作上受限制的线性表。()

10. 队列是与线性表完全不同的一种数据结构。()

11. 队列是一种操作受限的线性表，凡对数据元素的操作仅限一端进行。()

12. 栈和队列也是线性表。如果需要，可对它们中的任一元素进行操作。()

13. 栈是限定仅在表头进行插入和表尾进行删除运算的线性表。()

选择题：

1 从逻辑上可以把数据结构分成( c )。

A. 动态结构和静态结构 B. 顺序组织和链接组织

C. 线性结构和非线性结构 D. 基本类型和组合类型

2 线性表L在( b )情况下适于使用链表结构实现。

A. 不需修改L的结构 B. 需不断对L进行删除、插入

C. 需经常修改L中结点值 D. L中含有大量结点

3 带头结点的单链表L为空的判断条件是 b 。

带头结点的循环链表L为空的判断条件是 。

A. L==null B. L->next==null

C. L->next==L D. L!=null

4 若顺序表中各结点的查找概率不等，则可用如下策略提高顺序查找的效率：若找到指定

的结点，将该结点与其后继（若存在）结点交换位置，使得经常被查找的结点逐渐移至

表尾。以下为据此策略编写的算法，请选择适当的内容，完成此功能。

顺序表的存储结构为：

typedef struct{

ElemType \*elem; //数据元素存储空间，0号单元作监视哨

int length; //表长度

}SSTable;

int search\_seq(SSTable ST,KeyType key)

{ //在顺序表ST中顺序查找关键字等于key的数据元素。

//若找到，则将该元素与其后继交换位置，并返回其在表中的位置，否则为0。

ST.elem[0].key=key;

i=ST.length;

while(ST.elem[i].key!=key) f ;

if( G )

{ST.elem[i]←→ST.elem[i+1]；

e ；

}

return i;

}

A. i>0 B. i>=0 C. i<ST.length D. i<=ST.length

E. i++ F. i-- G. A和C同时满足 H. B和D同时满足

5 若入栈顺序为A、B、C、D、E，则下列( d )出栈序列是不可能的。

A．A、B、C、D、E 　　　　B．B、C、D、A、E

C．C、D、B、E、A　　　 　D．D、E、C、A、B

6 递归程序可借助于( c )转化为非递归程序。

a.线性表 b.队列 c: 栈 d.数组

7 在下列数据结构中( c )具有先进先出（FIFO）特性，

( b )具有先进后出(FILO）特性。

a．线性表 b．栈 c．队列 d．广义表

8 若对编号为1，2，3的列车车厢依次通过扳道栈进行调度，不能得到 ( e ) 的序列。

a:1,2,3 b:1,3,2 c:2,1,3 d:2,3,1 e:3,1,2 f:3,2,1

9 在计算递归函数时，如不用递归过程，应借助于( b ) 这种数据结构。

A. 线性表 B. 栈 C. 队列 D. 双向队列

10 若带头结点的链表只设尾结点指针。下列选择中（ c ）最适用于队列。

A）单链表 B）双向链表 C循环单链表 D）双向循环链表

11 栈和队列的一个共同点是( c )。

A. 都是先进先出 B. 都是先进后出

C. 只允许在端点处插入和删除元素 D. 没有共同点

12 循环队列用数组A[0..m-1]存放其元素值，设头尾指针分别为front和rear，则当前队列中

的元素个数是( c )。

A. rear-front-1 B. Rear-front+1

C. (rear-front+m)%m D. Rear-front

13 如下关于串的陈述中，正确的是( a, c )。

A. 串是数据元素类型特殊的线性表 B. 串中的元素是字母

C. 串中若干个元素构成的子序列称为子串 D. 空串即为空格串

14 对字符串s=’data-structure’ 执行操作replace(s,substring(s,6,8),’bas’)

的结果是 ( d ) 。

a: ‘database’ b: ‘data-base’ c: ‘bas’ d: ‘data-basucture’

15 设有二维数组A 5 x 7 ,每一元素用相邻的4个字节存储，存储器按字节编址.

已知A的起始地址为100。则按行存储时，元素A06的第一个字节的地址是（d ）

按列存储时，元素A06的第一个字节的地址是（ a ）

a: 220 b: 200 c: 140 d: 124

16对广义表 A=（（a,(b)）,(c,()),d）执行操作gettail(gethead(gettail(A)))

的结果是：（ b ） 。

a:（） b: （（）） c: d d: (d)

17 假设用于通讯的电文仅由6个字符组成，字母在电文中出现的频率分别为7, 19, 22, 6, 32,

14。 若为这6个字母设计哈夫曼编码（设生成新的二叉树的规则是按给出的次序从左至

右的结合，新生成的二叉树总是插入在最右），则频率为7的字符编码是（ g ），频率

为32的字符编码是（ c ）。

a: 00 b: 01 c: 10 d: 11

e: 011 f: 110 g: 1110 h:1111

三．填空题

1 数据结构通常有下列4类基本结构：集合、 线性结构 、树型结构、图型结构。

2 设单链表中结点形式为 data next ，若单链表长度大于等于2，指针p指向表中某个结点且p->next非空，此时若要删除指针p所指的结点，可以通过如下方法进行：将p所指结点的后继的元素值复制到该结点，然后删除其后继结点。相应的语句序列为：

q = p->next ； p→data = q→data ； p→next = q->next ； free（q） ；

3 线性表的顺序存储结构是以 存储位置 来表示数据元素之间的逻辑关系的。

4 已知P是单链表中某一结点的指针，P既不是首元结点也不是尾元结点，Q是P 的 前驱

结点指针。当删除P结点时，链表的链接可用语句（ Q→next = P->next ）实现。

算法设计题

1 设有一个带头结点、元素按值递增有序的单链表，结点的类型定义如下：

typedef struct LNode

{ int data;

struct LNode \*next;

} LNode, \*LinkList;

编写算法，删除其中所有值相同的多余元素结点

2 某线性表中元素以降序排列，现要插入一个元素X，插入后该线性元素仍保持降序。线性表采用带头结点单链表方式存贮。请编写该插入算法。

3 编写在一有序顺序表中插入数据元素X的算法 INSERT（Ｌ，Ｘ）。

4 写一算法，Delete(linklist &L，X) ,删除单链表中所有值为X的结点。

单链表结点的类型定义如下：

typedef struct LNode {

int data;

struct LNode \*next;

} LNode, \*Linklist;

5 写一算法，Contrary(linklist &L) ,对一带头结点且仅设尾指针L的循环单链表就地逆置。（即表头变表尾，表尾变表头。）

6 已知线性表中的元素以值递增有序排列，并以带头结点的单链表作存储结构。试写一高效

的算法，删除表中所有值大于mink且小于maxk的元素（若表中存在这样的元素）同时释

放被删结点空间，并分析你的算法的时间复杂度。

单链表结点的类型定义如下：

typedef struct LNode {

int data;

struct LNode \*next;

} LNode, \*Linklist;

7 写一算法，将带头结点的有序单链表A和B合并成一新的有序表C。(注:不破坏A和B

的原有结构.)Merge(Linklist A, Linklist B, Linklist &C )

8 写一算法Oplinklist(linklist L,int i;int j )

删除单链表中第i个元素,并将之插入至原表中的第j个元素之前.

9 写出求单链表长度算法 int length(linklist L)

10 若将循环队列Q的结构定义为：

#define m 100 　//最大队列长度

typedef struct

{ QElemType \*base; 　//存储空间基址

int rear; 　//尾指针，若队列不空，指向队尾元素

int length;　　//当前队列的长度，即元素个数

} SqQueue;

试写出相应初始化、入队列和出队列的三个函数。