第 2 章 线性表

一、选择题

1. 下述哪一条是顺序存储结构的优点？（ ）

   A．存储密度大 B．插入运算方便

   C．删除运算方便 D．可方便的用于各种逻辑结构的存储表示

2. 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（ ）存储方式最节省时间。

   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

3. 对于顺序存储结构的线性表，增加、删除结点和访问结点的时间复杂度分别为（ ）

   A. O(n) O(n) B. O(n) O(1) C.O(1) O(n) D. O(1) O(1)

4. 以下（ ）是一个线性表。

   A.由 n 个实数组成的集合 B.由 100 个字符组成的序列

   C.所有整数组成的序列 D.邻接表

5. 从一个具有 2n 个结点的单链表中查找其值等于 x 的结点时，在查找成功的情况下，需平均比较（ ）个元素结点。

   A. (n+1)/2 B. n C. (2n+1)/2 D. (2n-1)/2

6. 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( )最节省时间。

   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

7. 在双向链表存储结构中，删除 p 所指的结点时须修改指针（ ）。

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

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

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

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

8. 下面的叙述不正确的是（ ）

   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

9. 静态链表中指针表示的是（ ）.

   A． 内存地址 B．数组下标 C．下一元素地址 D．左、右孩子地址

10. (1) 静态链表既有顺序存储的优点，又有动态链表的优点。所以，它存取表中第 i 个元素的时间与 i 无关。

    (2) 静态链表中能容纳的元素个数的最大数在表定义时就确定了，以后不能增加。

    (3) 静态链表与动态链表在元素的插入、删除上类似，不需做元素的移动。

    以上错误的是（ ）

A.（1），（2） B.（1） C.（1），（2）,(3) D.（2）

11. 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（ ）(1<=i<=n+1)。

A. O(0) B. O(1) C. O(n) D. $O(n^2)$

二、判断题

1. 线性表的特点是每个元素都有一个前驱和一个后继。（ ）
2. 所谓静态链表就是一直不发生变化的链表。（ ）
3. 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。（ ）
4. 与单链表相比，双链表的优点是可以更加灵活的访问前后相邻结点。（ ）
5. 为了很方便的插入和删除数据，可以使用双向链表存放数据。（ ）
6. 取线性表的第 i 个元素的时间同 i 的大小有关。（ ）
7. 对任何数据结构链式存储结构一定优于顺序存储结构。（ ）
8. 顺序存储方式只能用于存储线性结构。（ ）
9. 集合与线性表的区别在于是否按关键字排序。（ ）
10. 所谓静态链表就是一直不发生变化的链表。（ ）

三、填空题

1. 在一个长度为 n 的顺序表中第 i 个元素（1<=i<=n）之前插入一个元素时，需向后移动_____个元素，若是在第 i 个元素（1<=i<=n）之后插入一个元素，需向后移动_____个元素。
2. 设单链表的结点结构为(data,next)，next 为指针域，已知指针 px 指向单链表中 data 为 x 的结点，指针 py 指向 data 为 y 的新结点，若将结点 y 插入结点 x 之后，则需要执行以下语句:_____； _____;
3. 在一个长度为 n 的顺序表中第 i 个元素（1<=i<=n）之前插入一个元素时，需向后移动_____个元素。
4. 在单链表中设置头结点的作用是:

   _____

   _____。

5. 以下程序的功能是实现带附加头结点的单链表数据结点逆序连接，请填空完善之。

```
void reverse(pointer h)
 /* h 为附加头结点指针；类型 pointer 同算法设计第 3 题*/
{ pointer p,q;
  p=h->next;  h->next=NULL;
```

```
  while((1)_____)
  {q=p; p=p->next; q->next=h->next; h->next=(2)_____;   }
}
```

6. 一元稀疏多项式以循环单链表按降幂排列，结点有三个域，系数域 coef ，指数域 exp 和指针域 next；现对链表求一阶导数 ，链表的头指针为 ha，头结点的 exp 域为 －1。

```
 derivative(ha)
  { q=ha ;   pa=ha->next;
    while( (1)_____)
      { if ( (2)____) { (_(3)_); free(pa);  pa= ( (4)__);  }
        else{ pa->coef ( (5)____); pa->exp( (6)___); q=( (7)__);}
        pa=( (8)_____);
      }
  }
```

7. 下面是一个求两个集合 A 和 B 之差 C=A-B 的程序，即当且仅当 e 是 A 的一个元素，但不是 B 中的一个元素时，e 才是 C 中的一个元素。集合用有序链表实现，初始时，A，B 集合中的元素按递增排列，C 为空；操作完成后 A，B 保持不变，C 中元素按递增排列。下面的函数 append(last,e)是把值为 e 的新结点链接在由指针 last 指向的结点的后面，并返回新结点的地址；函数 difference(A,B)实现集合运算 A-B，并返回表示结果集合 C 的链表的首结点的地址。在执行 A-B 运算之前，用于表示结果集合的链表首先增加一个附加的表头结点，以便新结点的添加，当 A-B 运算执行完毕，再删除并释放表示结果集合的链表的表头结点。

```
 typedef struct node{ int element;  struct node *link;}NODE;
NODE  *A, *B, *C;
NODE  *append (NODE *last,int e)
{  last->link=(NODE*) malloc (sizeof(NODE));
   last->link->element=e;
   return(last->link);
 }
NODE *difference(NODE *A,NODE *B)
{NODE *C,*last;
 C=last=(NODE*) malloc (sizeof(NODE));
 while (1)___
    if (A->element<B->element) { last=append(last,A->element);  A=A->link;  }
  else  if (2) ___ { A=A->link;  B=B->link;  } ELSE  (3) ___ ;
  while (4) __
```

```
    { last=append(last,A->element); A=A->link;  }
   (5) ___;  last=C;    C=C->link;  free (last);   return (C);
  }
/*call form:C=difference(A,B);*/
```

8. 对于单链表，在两个结点之间插入一个新结点需修改的指针共 _____个，双向链表为 _____个.
9. 已知指针 p 指向单链表 L 中的某结点，则删除其后继结点的语句是：_____。（该结点非末尾结点)
10. 对于一个头指针为 head 的带头结点的单链表，判定该表为空表的语句为 _____;对于不带头结点的单链表，判定该表为空表的语句为 _____;
11. 根据线性表的链式存储结构中每一个结点包含的指针个数，将线性链表分成 _____和_____;而又根据指针的连接方式,链表又可分成_____和_____。
12. 下面是用 C 语言编写的对不带头结点的单链表进行就地逆置的算法，该算法用 L 返回逆置后的链表的头指针，试在空缺处填入适当的语句， 并在右侧空白处说明逆置完成时指针 p、q、L 所指向的位置（可画图说明）。

```
void   reverse(linklist&L){
    p=null;
    q=L;
    while(q! = null){
    (1) _____;
    q->next=p;
    p=q;
    (2) _____;
    }
    (3) _____;
}
```

13. 通过循环单链表来实现对一元稀疏多项式的求一阶导数，其中数据元素按降幂排列，结点有三个域,系数域 coef ,指数域 exp 和指针域 next;其中指向链表头结点的头指针为 head,

```
derivative(head){
    q = head ;
    pa = head->next;
    while((1)_____){
```

```
            if ((2)                    ) {
                    ((3)                    );
                    free(pa);
                     pa= ((4)                    );
            }
            else{
                     pa->coef = ((5)                    );
                    pa->exp = ((6)                    );
                    q=( (7)                    );
            }
            pa=((8)  _____                    );
        }
    }
```

四、应用题

（编写算法时不需要写出全部程序，伪代码即可。每个题要写出所使用数据结构的结构体定义，注意要有相应注释说明）

1. 设单链表结点指针域为 next，试写出删除链表中指针 p 所指结点的直接后继的 C 语言语句。

2. 设单链表中某指针 p 所指结点（即 p 结点）的数据域为 data，链指针域为 next，请写出在 p 结点之前插入 s 结点的操作。

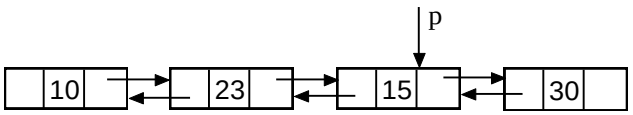
3. 设计一高效算法，将顺序表的所有元素逆置，要求算法的空间复杂度为 O（1）.

4. 从有序顺序表中删除所有其值重复的元素，使表中所有元素的值均不同。

5. 从顺序表中删除具有最小值的元素（假设唯一）并由函数返回被删元素的值，空出的位置由最后一个元素填补，若顺序表为空则显示出错并退出。

6. 给定两个单链表，编写算法找出两个链表的公共结点

7. 写出下图双链表中对换值为 23 和 15 的两个结点相互位置时修改指针的有关语句。（要写出结构体的定义，结点结构可参考：(llink,data,rlink)）



8. 编写一个算法，将两个递增有序的顺序表 L1，L2 合并成一个新的递增有序顺序表，返回结果顺序表 L

9. 设有一不带头结点的单链表，编程将链表按从小到大的顺序排序。(要求不用另外的数组或结点完成)

10. 设有两个从小到大排序的带头结点的有序链表。试编写求这两个链表交运算的算法（即 L1∩L2）。要求结果链表仍是从小到大排序，但无重复元素。

11. 设计算法判断带头结点的循环双链表是否对称

12. 两个整数序列 A = a1,a2,a3,...,am 和 B = b1,b2,b3,...,bn 分别存储再两个单链表中，设计一个算法，判断序列 B 是否是序列 A 的连续子序列。

五、算法设计

1. 编写程序用顺序表求解约瑟夫问题。

2. 假设有两个按元素值递增次序排列的线性表，均以单链表形式存储。请编写程序将这两个单链表归并为一个按元素值递减次序排列的单链表，并要求利用原来两个单链表的结点存放归并后的单链表。