一、填空（每空 1 分，共 13 分）

1.在顺序表中插入或删除一个元素，需要平均移动_表中一半_元素，具体移动的元素个数与_表长和该元素在表中的位置_有关。

2. 线性表中结点的集合是_有限_的，结点间的关系是_一对一_的。

3. **向一个长度为 n 的向量的第 i 个元素(1≤i≤n+1)之前插入一个元素时，需向后移动_n-i+1_个元素。**

4. **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。**

5. 在顺序表中访问任意一结点的时间复杂度均为_O(1)_，因此，顺序表也称为_随机存取_的数据结构。

6.顺序表中逻辑上相邻的元素的物理位置_必定_相邻。单链表中逻辑上相邻的元素的物理位置_不一定_相邻。

7.在单链表中，除了首元结点外，任一结点的存储位置由_其直接前驱结点的链域的值_指示。

8. 在 n 个结点的单链表中要删除已知结点*p，需找到它的_前驱结点的地址_，其时间复杂度为_O（n）_。

二、判断正误（在正确的说法后面打勾，反之打叉）（每小题 1 分，共 10 分）

（ × ）1. 链表的每个结点中都恰好包含一个指针。

答：错误。链表中的结点可含多个指针域，分别存放多个指针。例如，双向链表中的结点可以含有两个指针域，分别存放指向其直接前趋和直接后继结点的指针。

（ × ）2. 链表的物理存储结构具有同链表一样的顺序。错，链表的存储结构特点是无序，而链表的示意图有序。

（ × ）3. 链表的删除算法很简单，因为当删除链中某个结点后，计算机会自动地将后续的各个单元向前移动。错，链表的结点不会移动，只是指针内容改变。

（ × ）4. 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。

错，混淆了逻辑结构与物理结构，链表也是线性表！且即使是顺序表，也能存放记录型数据。

（ × ）5. 顺序表结构适宜于进行顺序存取，而链表适宜于进行随机存取。

错，正好说反了。顺序表才适合随机存取，链表恰恰适于“顺藤摸瓜”

（ × ）6. 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。

错，前一半正确，但后一半说法错误，那是链式存储的优点。顺序存储方式插入、删除运算效率较低，在表长为 n 的顺序表中，插入和删除一个数据元素，平均需移动表长一半个数的数据元素。

（ × ）7. 线性表在物理存储空间中也一定是连续的。

错，线性表有两种存储方式，顺序存储和链式存储。后者不要求连续存放。

（ × ）8. 线性表在顺序存储时，逻辑上相邻的元素未必在存储的物理位置次序上相邻。

错误。线性表有两种存储方式，在顺序存储时，逻辑上相邻的元素在存储的物理位置次序上也相邻。

（ × ）9. 顺序存储方式只能用于存储线性结构。

错误。顺序存储方式不仅能用于存储线性结构，还可以用来存放非线性结构，例如完全二叉树是属于非线性结构，但其最佳存储方式是顺序存储方式。（后一节介绍）

（ × ）10. 线性表的逻辑顺序与存储顺序总是一致的。

错，理由同 7。链式存储就无需一致。

三、单项选择题（每小题 1 分，共 10 分）

（ C ）1. 数据在计算机存储器内表示时，物理地址与逻辑地址相同并且是连续的，称之为：

（A）存储结构　　（B）逻辑结构　　（C）顺序存储结构　　（D）链式存储结构

（ B ）2.一个向量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_____
（A）110　　（B）108　　（C）100　　（D）120

（ A ）3. 在 n 个结点的顺序表中，算法的时间复杂度是 O（1）的操作是：
（A） 访问第 i 个结点（1≤i≤n）和求第 i 个结点的直接前驱（2≤i≤n）
（B） 在第 i 个结点后插入一个新结点（1≤i≤n）
（C） 删除第 i 个结点（1≤i≤n）
（D） 将 n 个结点从小到大排序

（ B ）4. 向一个有 127 个元素的顺序表中插入一个新元素并保持原来顺序不变，平均要移动__个元素
（A）8　　（B）63.5　　（C）63　　（D）7

（ A ）5. 链接存储的存储结构所占存储空间：
（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

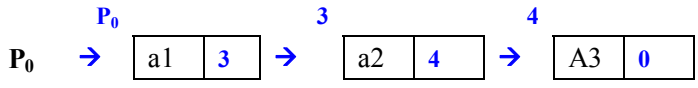
（ B ）6. 链表是一种采用_____存储结构存储的线性表；
（A）顺序　　（B）链式　　（C）星式　　（D）网状

（ D ）7. 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:
（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

（ B ）8. 线性表 L 在_____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

（ C ）9. 单链表的存储密度
（A）大于 1；　（B）等于 1；　（C）小于 1；　（D）不能确定

（ B ）10. 设 a1、a2、a3 为 3 个结点，整数 P_0，3，4 代表地址，则如下的链式存储结构称为



（A）循环链表　（B）单链表　（C）双向循环链表　（D）双向链表

四、简答题（每小题 5 分，共 10 分）

1.试比较顺序存储结构和链式存储结构的优缺点。在什么情况下用顺序表比链表好？

答：① 顺序存储时，相邻数据元素的存放地址也相邻（逻辑与物理统一）；要求内存中可用存储单元的地址必须是连续的。优点：存储密度大（=1？），存储空间利用率高。缺点：插入或删除元素时不方便。

②链式存储时，相邻数据元素可随意存放，但所占存储空间分两部分，一部分存放结点值，另一部分存放表示结点间关系的指针

优点：插入或删除元素时很方便，使用灵活。缺点：存储密度小（<1），存储空间利用率低。

顺序表适宜于做查找这样的静态操作；链表宜于做插入、删除这样的动态操作。

若线性表的长度变化不大，且其主要操作是查找，则采用顺序表；

若线性表的长度变化较大，且其主要操作是插入、删除操作，则采用链表。

2 .描述以下三个概念的区别：头指针、头结点、首元结点（第一个元素结点）。在单链表中设置头结点的作用是什么？

答：首元结点是指链表中存储线性表中第一个数据元素 a_1 的结点。为了操作方便，通常在链表的首元结点之前附设一个结点，称为头结点，该结点的数据域中不存储线性表的数据元素，其作用是为了对链表进行操作时，可以对空表、非空表的情况以及对首元结点进行统一处理。头指针是指向链表中第一个结点（或为头结点或为首元结点）的指针。若链表中附设头结点，则不管线性表是否为空表，头指针均不为空。否则表示空表的链表的头指针为空。这三个概念对单链表、双向链表和循环链表均适用。是否设置头结点，是不同的存储结构表示同一逻辑结构的问题。

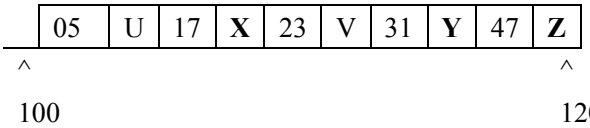
简而言之，

头指针是指向链表中第一个结点（或为头结点或为首元结点）的指针；

头结点是在链表的首元结点之前附设的一个结点；数据域内只放空表标志和表长等信息（内放头指针？那还得另配一个头指针!!!）

首元素结点是指链表中存储线性表中第一个数据元素 a_1 的结点。

五、线性表具有两种存储方式，即顺序方式和链接方式。现有一个具有五个元素的线性表 L={23，17，47，05，31}，若它以链接方式存储在下列 100～119 号地址空间中，每个结点由数据（占 2 个字节）和指针（占 2 个字节）组成，如下所示：



其中指针 X，Y，Z 的值分别为多少？该线性表的首结点起始地址为多少？末结点的起始地址为多少？（10 分）

答：X=__116__ Y=__0__ Z=__100__ 首址=__108__ 末址=__112__

六、阅读分析题（10 分）

指出以下算法中的错误和低效（即费时）之处，并将它改写为一个既正确又高效的算法。

```
Status DeleteK(SqList&a, int i, int k){
//本过程从顺序存储结构的线性表 a 中删除第 i 个元素起的 k 个元素
if ( i<1 || k<0 || i+k> a.length ) return INFEASIBLE;     //参数不合法
else{
   for(count = 1; count <k; count ++ ) {
   //删除一个元素
   for ( j = a.length; j>=i+1;   j--) a.elem[j-1] = a.elem[j];
   a.length - -;
   }
       return OK;
   } // DeleteK
```

```
注：上题涉及的类型定义如下：
# define LIST INIT SIZE 100
# define LISTINCREMENT 10
typedef struct {
   Elem Type       *elem;                  //存储空间基址
   Int              length;                //当前长度
   Int              listsize;              //当前分配的存储容量
   }SqList;
```

答：错误有两处：

① 参数不合法的判别条件不完整。例如表长为 10，若从第一位置（i=1）删除 10 个元素（k=10）,要求合理但会被判为非法。

合法的入口参数条件为（0<i≤a.length）^ (0≤k≤a.length-i)

应将 if (i<1 || k<0 || i+k> a.length) **return** INFEASIBLE

改为：if (!（(0<i≤a.length）^ (o≤k≤a.length-i)）) **return** INFEASIBLE

第二个 FOR 语句中，元素前移的次序错误。应将 for (j = a.length; j>=i+1;   j--) a.elem[j-1] = a.elem[j];

改为 for (j>=i+1; j = a.length; j++) a.elem[j-1] = a.elem[j];

七、编程题（每题 10 分，共 40 分）

1.写出在顺序存储结构下将线性表逆转的算法，要求使用最少的附加空间。

解：输入：长度为 n 的线性表数组 A(1:n)

输出：逆转后的长度为 n 的线性表数组 A(1:n)。

C 语言描述如下（其中 ET 为数据元素的类型）：

```
invsl(n,a)
int n;
ET a[];
{int k;
ET t;
for (k=1; k<=n/2; k++)
{t=a[k-1]; a[k-1]=a[n-k]; a[n-k]=t;}
return;
}
```

2.已知 L 是无表头结点的单链表，且 P 结点既不是首元结点，也不是尾元结点，请写出在 P 结点后插入 S 结点的核心语句序列。

答：此题答案不唯一，但若从已给定序列中挑选，则限制颇多。

(7) Q=P;

(11) P=L;

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

(10) P=Q;

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

P->next=S;

已知 P 结点，则不必“顺藤摸瓜”，直接链接即可。

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

(1) P->next=S;

3. 编写程序，将若干整数从键盘输入，以单链表形式存储起来，然后计算单链表中结点的个数（其中指针 P 指向该链表的第一个结点）。 注：统计结点个数是【省统考样题】的要求，也是教材 P60 4-6 计算链表长度的要求，编程又简单，很容易作为考题。

解：编写 C 程序如下(已上机通过)：

全局变量及函数提前说明：

```
--------------------------------
#include<stdio.h>
#include<stdlib.h>
typedef struct liuyu{int data;struct liuyu*link;}test;
liuyu *p,*q,*r,*head;
int m=sizeof(test);

void    main ()                 /*第一步，从键盘输入整数，不断添加到链表*/
{int i;
head=(test*)malloc(m);     /*m=sizeof(test);*/
p=head; i=0;
while (i!=-9999)
{ printf("/ninput an integer [stop by '-9999']:");
scanf("%d",&i);
p->data=i;                          /* input data    is saved       */
p->link=(test*)malloc(m);           /*m=sizeof(test));*/
q=p;
p=p->link;
}
q->link=NULL;                   /*原先用 p->link=NULL 似乎太晚！ */


p=head; i=0;                /*统计链表结点的个数并打印出来*/
while (p->link!=NULL)
{printf("%d",p->data);
p=p->link;
i++;
}
printf("\n node number=%d\n", i-1);   /*结点的个数不包括-9999*/
}
```

4. 请编写 26 个字母按特定字母值插入或删除的完整程序，可自行选用顺序存储或链表结构。

答：

```
#include<stdio.h>                    /*全局变量及函数提前说明：*/
#include<stdlib.h>
typedef struct liuyu{char data;struct liuyu*link;}test;
liuyu *p,*q,*r,*head;
int L;                               /*元素的个数*/
int m=sizeof(test);
void build();                        /*  主函数中会被调用的函数应当预先说明  */
void display();
int insert_char(char,char);          /*插入一个字母，在第字母 Y 之前，若无字母则加到末尾*/
int delet_char(char);                /*  删除元素 X，注意保存 X 的前趋元素指针！   */
/*----------------------------------------------------------*/
void build()                         /*字母链表的生成*/
{int i;
head=(test*)malloc(m);               /*m=sizeof(test);*/
p=head;
for(i=1;i<L;i++)
{ p->data=i+'a'-1;                    /*    'a'也可用其 ASCII 码 97 来表示      */
p->link=(test*)malloc(m);            /*m=sizeof(test));*/
p=p->link; }
p->data=i+'a'-1;
p->link=NULL;
}
/*----------------------------------------------------------*/
void display()               /*字母链表的输出*/
{p=head;
while (p->link!=NULL)
{ printf("%c",p->data);
p=p->link; }
printf("%c\n",p->data);
}
/*----------------------------------------------------------*/
int insert_char(char X,char Y)     /*插入一个字母 X 在某个字母 Y 之前，若找不到 Y 字母则加到末尾*/
{p=head;
r=(test*)malloc(m);
r->data=X;
if(head->data==Y)
   { head=r;
     r->link=p;     }
else{    while((p->data!=Y)&&(p->link!=NULL))    {q=p;    p=p->link;}
       if(p->data==Y)    { q->link=r; r->link=p; }
       else{p->link=r;r->link=NULL;}
       }
```

```
L++;
return(0);
}
/*-----------------------------------------------------------*/
int delet_char(char X)     /*  删除元素 X，注意保存 X 的前趋元素指针！   */
{ p=head;
if(head->data==X){head=head->link;free(p);}
else{ while((p->data!=X)&&(p->link!=NULL))
        {q=p;
         p=p->link;}
if(p->data==X)
        { q->link=p->link;
        free(p);     }
   else return(-1);
}
L--;
return(0);
}
/*-----------------------------------------------------------*/
void main(void)          /*字母线性表的生成和输出*/
{ L=26;
  build();
display();
printf("insert return value=%d\n",insert_char('L','W'));
display();
printf("delete return value=%d\n",delet_char('z'));
display();
}
```

附：屏幕上显示的执行结果是：

a b c d e f g h i j k l m n o p q r s t u v w x y z
insert return value=0
a b c d 9 e f g h i j k l m n o p q r s t u v w x y z L
delete return value=0
a b c d e f g h i j k l m n o p q r s t u v w x y L