第二章链表　自测卷答案　　姓名_____　班级_____

题号	一	二	三	四	五	六	七	总分
题分	13	10	10	10	7	10	40	100
得分								

一、填空（每空 1 分，共 13 分）

1. 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。
2. 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。
3. 【严题集 2.2①】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。
4. 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
5. 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）__。
6. 【严题集 2.6①】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. 单链表中每个节点都是由 __数据域__ 和 __指针域__ 构成。

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。

（ × ）2. 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”

（ × ）3. 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。

（ × ）4. 顺序存储方式只能用于存储线性结构。错。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）

（ × ）5. 链表的每个结点中都恰好包含一个指针。错。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。

（ × ）6. 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。

（ × ）7. 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。

（ × ）8. 线性表的逻辑顺序与物理顺序总是一致的。错，不一定一致。

（ × ）9. 单链表从任何一个结点出发，都能访问到所有结点。错，只有从头结点出发才行。

（ √ ）10. 单链表形式的队列，头指针 F 指向队列的第一个结点，尾指针 R 指向队列的最后一个节点。对的

三、单项选择题（每小题 1 分，共 10 分）

（ B ）1.一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_____

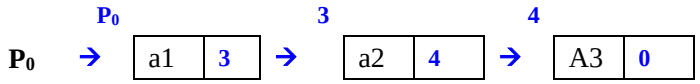
（A）110
（B）108
（C）100
（D）120

（ A ）2. 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

（ D ）3. 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

（ C ）4. 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 ）5 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

（ C ）6. 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

（ D ）7. 下列叙述中正确的是：
（A）非线性结构只能采用链式存储结构
（B）非线性结构只能用多重链表表示
（C）所有数据结构既可以采用顺序存储结构，也可以采用链式存储结构
（D）有的非线性结构也能采用顺序存储结构

（ C ）8. 栈底至栈顶依次存放元素 A、B、C、D，在第五个元素 E 入栈前，栈中元素可以出栈，则出栈序列可能是：

（A）ABCDE
（B）DCBEA
（C）DBCEA
（D）CDABE

（ A ）9．循环链表的主要优点是：
（A）不再需要头指针
（B）从表中任一结点出发都能访问到整个链表
（C）在进行插入、删除运算时，能更好的的保证链表不断开
（D）已知某个结点的位置后，能够容易的找到它

（ B ）10.一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

四、简答题（每小题 5 分，共 10 分）

1. 解释顺序存储结构和链式存储结构的特点，并比较顺序存储结构和链式存储结构的优缺点。

答：顺序结构存储时，相邻数据元素的存放地址也相邻，即逻辑结构和存储结构是统一的，，要求内存中存储单元的地址必须是连续的。

优点：一般情况下，存储密度大，存储空间利用率高。

缺点：（1）在做插入和删除操作时，需移动大量元素；

（2）由于难以估计，必须预先分配较大的空间，往往使存储空间不能得到充分利用；

（3）表的容量难以扩充。

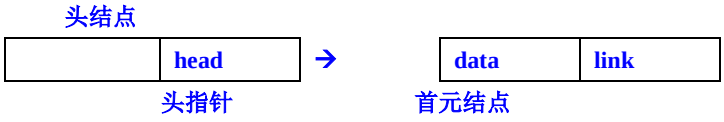
链式结构存储时，相邻数据元素可随意存放，所占空间分为两部分，一部分存放结点值，另一部分存放表示结点间关系的指针。

优点：插入和删除元素时很方便，使用灵活。

缺点：存储密度小，存储空间利用率低。

2．【严题集 2.1①】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第一个数据元素 a_1 的结点。

五、【软考题】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：

05	U	17	**X**	23	V	31	**Y**	47	**Z**

∧ ∧

100 120

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）

【严题集 2.10②】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.输入一个单向链表，判断链表是否有环？

答案：

```
bool hasCircle(Node *head,Node *&circleNode)
{
    Node *slow,*fast;
    slow = fast = head;
    while(fast != NULL && fast->next != NULL)
    {
        fast = fast->next->next;
        slow = slow->next;
        if(fast == slow)
        {
            circleNode = fast;
            return true;
        }
    }
    return false;
}
```

2. 【严题集 2.9】求链表的中间节点

答案：

```
Node* theMiddleNode(Node *head)
{
    if(head == NULL)
        return NULL;
    Node *slow,*fast;
    slow = fast = head;
    //在链表长度为偶数的情况下，返回中间两个节点的第一个，可以用下面的循环条件
    //while(fast && fast->next != NULL && fast->next->next != NULL)
    while(fast != NULL && fast->next != NULL)
    {
        fast = fast->next->next;
        slow = slow->next;
    }
    return slow;
```

3. 【严题集 2.11】查找链表第 K 个节点数据。

答案：

```
void    *searchRKthNode( List_t *list, unsigned int k ){
    if( list == NULL || list->head == NULL || k == 0)//k 是无符号数，因此需要判断是否为 0
        return NULL;
    ListElement_t *pFast = list->head;
    while( k > 1 && pFast != NULL ){
        --k;
        pFast = pFast->next;
    }
    //k>1 说明节点数少于 k 个，pFast=NULL 说明恰好链表为 k-1 个节点
    if( k > 1 || pFast == NULL )
        return NULL;
    ListElement_t *pSlow = list->head;
    while( pFast->next != NULL ){
        pFast = pFast->next;
        pSlow = pSlow->next;
    }
    return pSlow->data;
}
}
```

4. 倒序打印链表

答案：

```
int     ReversePrintList( List_t *list){
    if( list == NULL || list->head == NULL )
        return    INPUT_ERROR;

    Stack <ListElement_t *> st;
    ListElement_t *pNode = list->head;
    while( pNode != NULL ){
        st.push( pNode );
        pNode = pNode->next;
    }

    while( !st.empty() ){
        pNode = st.pop();
        print_func( pNode);
        st.pop();
    }
```