1. 实现一个单链表，满足如下操作：
2. 把数组A[8]={1,2,3,4,5,100,200,300}保存至单链表
3. 在单链表第6个位置插入数值586，并显示单链表此时新的情况，比如节点数
4. 然后对当前单链表第5个节点进行删除操作，并显示删除后单链表的状态
5. 显示所有操作的头结点地址。

解:

**void main()**

**{**

**LinkNode \*h; //注意，一般SqList为顺序表，Linklist为链表**

**int n=8;**

**ElemType a[]={**1,2,3,4,5,100,200,300**};**

**CreateListF(h,a[],n); //第（1）问，用头插法创建单链表**

Printf（h） //第（4）问，显示此时头结点地址

**ListInsert(h,6,586，n); //第（2）问，用头插法创建单链表**

//Printf（n）；

Printf（h） //第（4）问，显示此时头结点地址

**ListDelete（h,5）； //第（3）问，删除单链表第5个节点**

**DispList（h）； //第（3）问，显示单链表删除后的状态**

Printf（h） //第（4）问，显示此时头结点地址

**}**

**void CreateListF(LinkNode \*&L，ElemType a[]，int n) //头插法创建列表**

**{ LinkNode \*s;**

**int i;**

**L=(LinkNode \*)malloc(sizeof(LinkNode));**

**L->next=NULL; //创建头结点，其next域置为NULL**

**for (i=0;i<n;i++) //循环建立数据结点**

**{ s=(LinkNode \*)malloc(sizeof(LinkNode));**

**s->data=a[i]; //创建数据结点s**

**s->next=L->next; //将s插在原开始结点之前，头结点之后**

**L->next=s;**

**}**

**}**

**bool ListInsert(LinkNode \*&L，int i，ElemType e，int &m)**

**//在单链表第i个位置插入值为e的节点，并且返回节点数m；**

**{ int j=0;**

**LinkNode \*p=L，\*s; //p指向头结点，*j*置为0**

**while (j<i-1 && p!=NULL)**

**{ j++;**

**p=p->next;**

**}**

**if (p==NULL) //未找到第*i*-1个结点，返回false**

**return false;**

**else //找到第*i*-1个结点p，插入新结点并返回true**

**{**

**s=(LinkNode \*)malloc(sizeof(LinkNode));**

**s->data=e; //创建新结点s，其data域置为e**

**s->next=p->next; //将s插入到p之后**

**p->next=s;**

**m++; //只要插入一个新节点，节点数加一**

**return true;**

**}**

**}**

**bool ListDelete(LinkNode \*&L，int i)**

**//在第i个位置，删除节点**

**{ int j=0;**

**LinkNode \*p=L，\*q; //p指向头结点，j置为0**

**while (j<i-1 && p!=NULL) //查找第i-1个结点**

**{ j++;**

**p=p->next;**

**}**

**if (p==NULL) //未找到第*i*-1个结点，返回false**

**return false;**

**else //找到第*i*-1个结点p**

**{ q=p->next; //q指向第*i*个结点**

**if (q==NULL) //若不存在第i个结点，返回false**

**return false;**

**e=q->data;**

**p->next=q->next; //从单链表中删除q结点**

**free(q); //释放q结点**

**return true; //返回true表示成功删除第i个结点**

**}**

**}**

**void DispList(LinkNode \*L)**

**{**

**int i=1；**

**LinkNode \*p=L->next; //p指向开始结点**

**while (p!=NULL) //p不为NULL，输出p结点的data域**

**{**

**Printf(“第”+i+“个节点的值为：”)**

**printf("%d "，p->data);**

**p=p->next; //p移向下一个结点**

**i++；**

**}**

**printf("\n");**

**}**