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。

void main()

{

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

LinkNode \*Z

Create\_LinkList(a[],Z,8); //

}

//输入：数组、数组个数；输出：新单链表（我们链式存储每个节点都是一句malloc）

//方法名要有一定规范：queue（串）；List（线性表）；SqList（顺序表）；LinkList（链表）

//返回值+方法名（输入与输出）{}

ElemType Create\_LinkList(Elemtype a[])

{

ElemType a //新单链表地址

return a；//返回的新单链表地址

}

void Create\_LinkList(ElemType a[],LinkNode \*&L,int m) //头插法创建链表

{

//初始化链表，创建一个头结点

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

L—>next=null;

//头插法，先用一个结点打打底，只需要查到头结点就可以

for(int i=0;i<m;i++)

{

LinkNode \*S=（LinkNode \*）malloc（sizeof（LinkNode））； //创建一个新结点

S->data=a[i]; //对新结点赋值

S->next=L->next;

L->next=S;

}

}

void Create\_LinkList\_W(ElemType a[],LinkNode \*&L,int m) //尾插法创建链表

{

//初始化链表，创建一个头结点

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

L—>next=null;

//尾插法，先用一个结点打打底，只需要查到头结点就可以

//创建一个尾指针P，此指针永远指向尾结点

LinkNode \*P=L->next;

for(int i=0;i<m;i++)

{

LinkNode \*S=（LinkNode \*）malloc（sizeof（LinkNode））； //创建一个新结点

S->data=a[i]; //对新结点赋值

P->next=S;

P= P->next; //P永远指向尾结点

}

}