**一．实验目的**

掌握链表构造、查找、插入和合并等算法

**二．实验内容**

动态创建一个有5个数据元素的链表；

从键盘上输入一个新元素和插入位置，插入链表；

从键盘上输入一个删除位置，从链表中删除这个位置上的元素；

根据需要输出链表中的内容。

**三．实验步骤和实验结果**

**#include<stdio.h>**

**#include<stdlib.h>**

**typedef struct node{**

**int data;**

**struct node \*next;**

**}node;**

**void creartlist(node \*head,int size)//创建链表**

**{**

**node \*rear=head;**

**int i;**

**for(i=0;i<size;i++)**

**{**

**node \*newnode=(node\*)malloc(sizeof(node));**

**newnode->next=NULL;**

**scanf("%d",&newnode->data );**

**rear->next =newnode;**

**rear=newnode;**

**}**

**}**

**void travellist(node \*head)//遍历链表并输出**

**{**

**node \*p=head->next;**

**while(p!=NULL)**

**{**

**printf("%d",p->data);**

**p=p->next;**

**}**

**putchar('\n');**

**}**

**void insertlist(node \*head,int data)//插入链表元素**

**{**

**node \*newnode=(node\*)malloc(sizeof(node));**

**newnode->next=NULL;**

**newnode->data =data;**

**newnode->next =head->next ;**

**head->next=newnode;**

**}**

**void delist(node \*head,int n)//删除链表元素**

**{**

**node \*t=head,\*in;**

**int i=0;**

**while(i<n&&t!=NULL)**

**{**

**in =t;**

**t=t->next ;**

**i++;**

**}**

**if(t!=NULL)**

**{**

**in->next=t->next;**

**free(t);**

**}**

**else**

**{**

**printf("节点不存在");**

**}**

**}**

**int main()**

**{**

**node \*head=(node\*)malloc(sizeof(node));**

**head->next=NULL;**

**int n;**

**scanf("%d",&n);**

**creartlist(head,n);**

**travellist(head);**

**int x;**

**scanf("%d",&x);**

**insertlist(head,x);**

**travellist(head);**

**int y;**

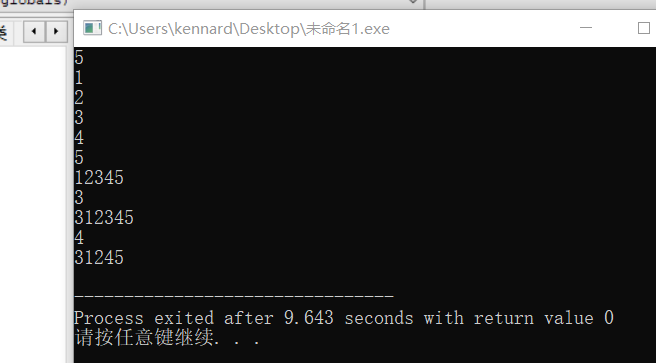
**scanf("%d",&y);**

**delist(head,y);**

**travellist(head);**

**return 0;**

**}**



图为创建一个五个元素的链表，并在头部插入元素，并且删除4号元素。

四．分析与讨论

1.链表依靠指针相连，在创建时需要注意这一点。

2.链表的插入时，先将新元素与插入位置后元素相连，再将插入位置前一元素的指针指向新元素，即先链接再断开。

3.链表的删除时，与链表的插入相同，要注意先将前一个元素与后相连，保证链表不会丢失，再断开。