**单链表**

**实验环境：C语言开发环境**

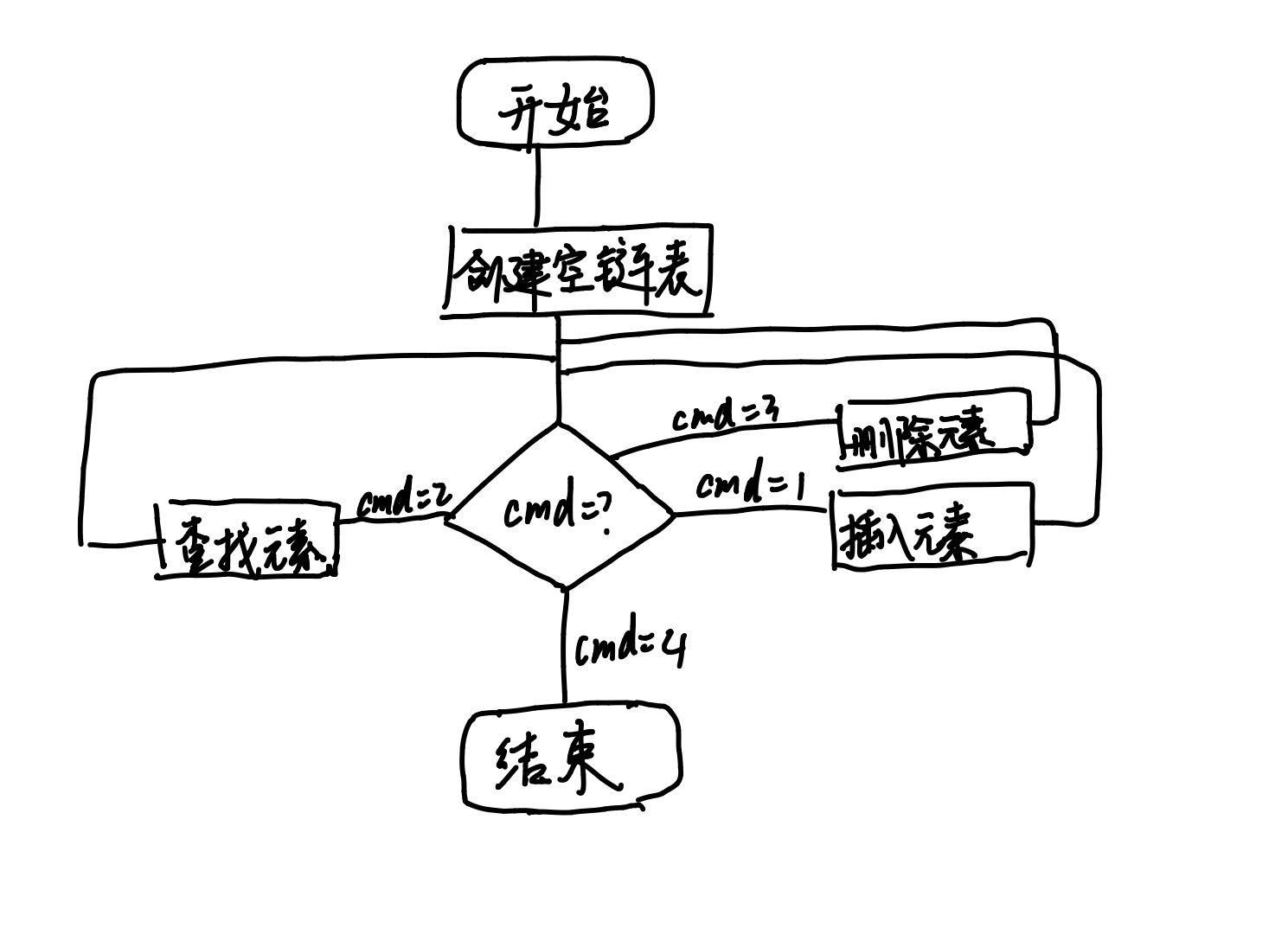
**实验目的：熟悉单链表的增删改查。**

**实验环境：**

**Win10**

**DevCpp**

**算法流程：**



**实验代码：**

#include<stdio.h>

#include<stdlib.h>

typedef int ElemType;

//结构体部分

typedef struct LNode

{

ElemType data;

struct LNode \*next;

}Linklist;

//初始化链表

Linklist \*InitList(Linklist \*L)

{

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

L->next = NULL;

return L;

}

//创建单链表

Linklist \*CreateList(int n)

{

int x,i;

Linklist \*L,\*r,\*p;

L = NULL;

L = InitList(L);

r = L;

for(i=0;i<n;i++)

{

scanf("%d",&x);

p = (Linklist \*)malloc(sizeof(Linklist));

p->data = x;

p->next = NULL;

r->next = p;

r = r->next;

}

return L;

}

//插入

int Insert(Linklist \*L,ElemType item,int x)

{

int i = 1;

Linklist \*p,\*t;

p = L;

t = (Linklist \*)malloc(sizeof(Linklist));

t->data = item;

if(L->next==NULL)

{

if(x==1)

{

L->next=t;

t->next=NULL;

return 1;

}

else

{

printf("操作失败！\n");

return 0;

}

}

while(p->next!=NULL&&i<x)

{

p = p->next;

i++;

}

if(p->next==NULL&&i<x)

{

printf("在%d位置处插入失败！\n",x);

return 0;

}

else

{

t->next = p->next;

p->next = t;

return 1;

}

}

//删除

int Delete(Linklist \*L,int x)

{

int i = 1;

Linklist \*p,\*q;

p = L;

if(L->next==NULL)

{

printf("单链表为空！\n");

return 0;

}

while(p->next!=NULL&&i<x)

{

p = p->next;

i++;

}

if(p->next==NULL)

{

printf("没有第%d个结点！\n",x);

return 0;

}

else

{

q = p->next;

p->next = p->next->next;

free(q);

return 1;

}

}

//查找

int Seek(Linklist \*L,ElemType x)

{

Linklist \*p;

int i = 1;

if(L->next==NULL)

{

printf("单链表为空！\n");

return 0;

}

else

{

p = L->next;

while(p!=NULL)

{

if(p->data!=x)

{

i++;

p = p->next;

}

else break;

}

if(p==NULL)

{

printf("没有数值%d！\n",x);

return 0;

}

else return i;

}

}

//输出

void output(Linklist \*L)

{

Linklist \*p;

p = L->next;

printf("该单链表为；\n");

for(;p!=NULL;p=p->next)

{

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

}

printf("\n");

}

int main()

{

int n;

Linklist \*L;

printf("请输入单链表元素的个数：");

scanf("%d",&n);

printf("请输入%d个元素：\n",n);

L = CreateList(n);

output(L);

printf("\t\tmenu\n1.插入元素\t2.查找元素\n3.删除元素\t4.退出\n");

int cmd;

scanf("%d",&cmd);

while(1){

if(cmd==1){

int m,x;

printf("请输入要插入的元素的值：");

scanf("%d",&m);

printf("请输入要插入的位置：");

scanf("%d",&x);

Insert(L,m,x);

}

else if(cmd==2){

int x;

printf("请输入要查找元素的值：");

scanf("%d",&x);

Seek(L,x);

}

else if(cmd==3){

int x;

printf("请输入要删除的位置：");

scanf("%d",&x);

Delete(L,x);

}

else if(cmd==4)

break;

output(L);

scanf("%d",&cmd);

system("cls");

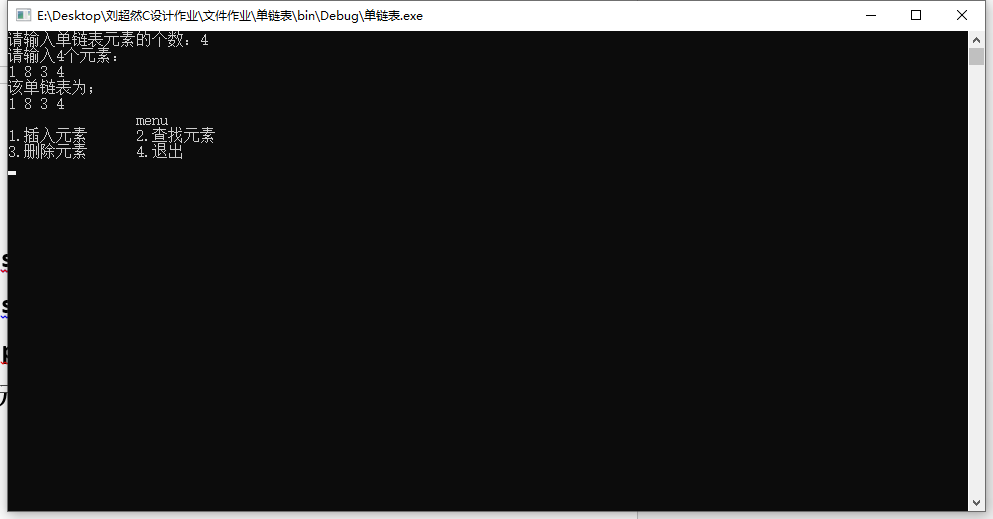
printf("\t\tmenu\n1.插入元素\t2.查找元素\n3.删除元素\t4.退出\n");

}

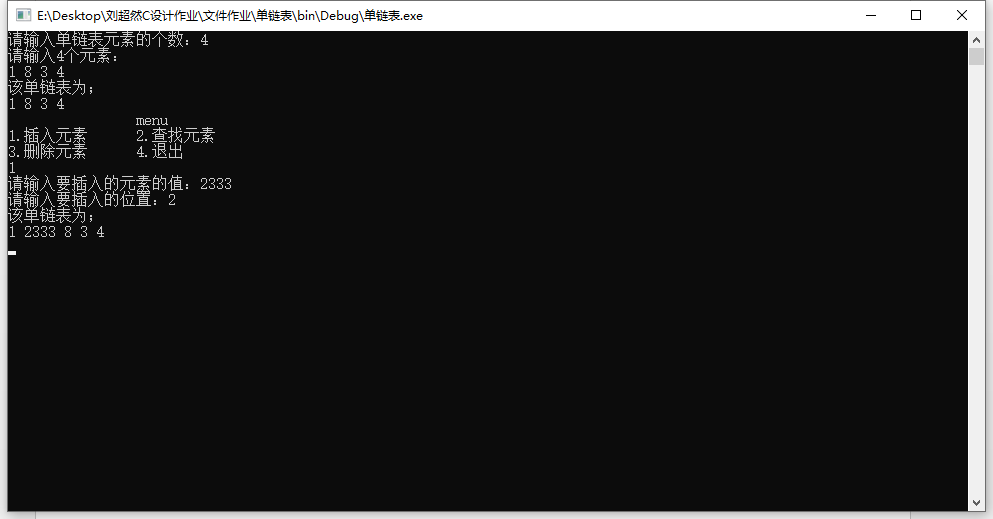
}

**运行截图：**

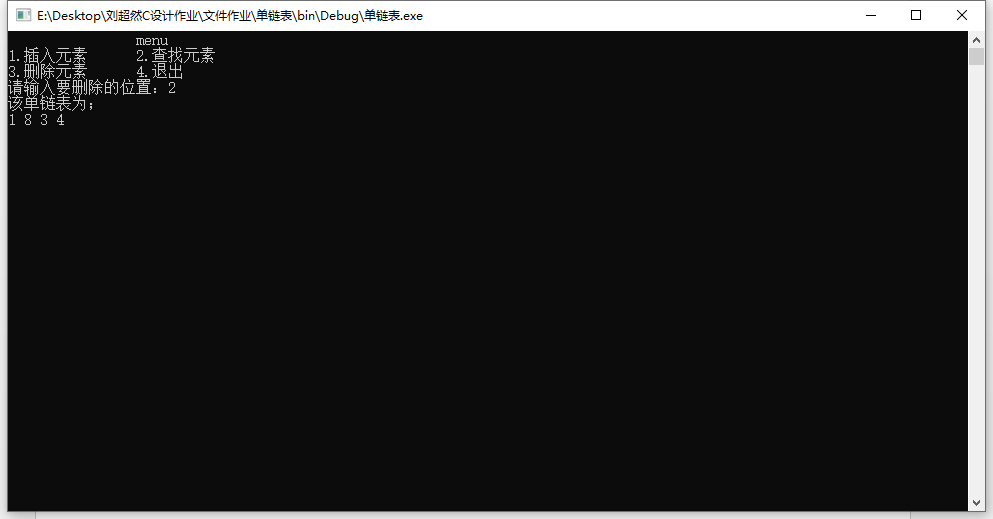
**单链表的创建：**



**插入元素：**



**删除元素：**



**查找元素：**

