**实验名称：**顺序表的基本操作

**实验目的：**

1、掌握线性表的顺序存储方式；

2、掌握顺序表的建立、插入、删除等操作。

**实验内容：**

1、初始化顺序表

2、实现数据元素的插入和删除

3、输出顺序表中的所有元素

#include<stdio.h>

#define MAXSIZE 100

//顺序表定义

typedef struct{

int \*elem;

int length;

}SqList;

/\*

函数功能：构造一个空的顺序表L

返回值：构造成功返回1，失败返回0

\*/

int init(SqList &L)

{

L.elem=new int[MAXSIZE];

if(!L.elem)

return 0;

L.length=0;

return 1;

}

/\*

函数功能：位置i前插入数据e

返回值：插入成功返回1，失败返回0

\*/

int insert(SqList &L, int i, int e)

{

if(i<1|| i>L.length+1)

{

printf("插入位置非法!\n");

return 0;

}

if(L.length==MAXSIZE)

{

printf("顺序表空间已满!\n");

return 0;

}

//补充完整

return 1;

}

/\*

函数功能：删除位置i的数据

返回值：删除成功返回1，失败返回0

\*/

int del(SqList &L, int i)

{

if(i<1 || i>L.length)

{

printf("删除位置非法!\n");

return 0;

}

//补充完整

return 1;

}

//函数功能：利用循环输出顺序表中的所有数据

void print(SqList L)

{

//补充完整

}

void main()

{

SqList L;

int i=1,x,e;

//1.调用init函数构造一个空顺序表L;

//补充完整

//2.从键盘输入数据插入顺序表(建表)，以-1结束;

printf("请依次输入顺序表的元素(以-1结束):");

scanf("%d",&x);

//补充完整

//3.调用del函数，删除位置i的数据;

printf("输入删除位置:");

scanf("%d",&i);

if(//补充完整)

printf("删除成功\n");

//4.调用print函数，输出顺序表的数据，验证前面步骤是否正确

//补充完整

}