# 2022-2023王道计算机练习：上机+笔试

## 1.typedef使用

<https://blog.csdn.net/weixin_46263778/article/details/126150684>

#include <stdio.h>

typedef struct student

{

int num;

char name[30];

char sex[30];

}stu, \* Studet;

//定义一个结构，里面有三个主体，在main中进行结构赋值，之后输出

typedef int INT;

int main()

{

stu s = { 180,"张珊","男" };

Studet p;

int n = 100;

p = &s;

printf("%d,身高:%d\n", n, p->num);

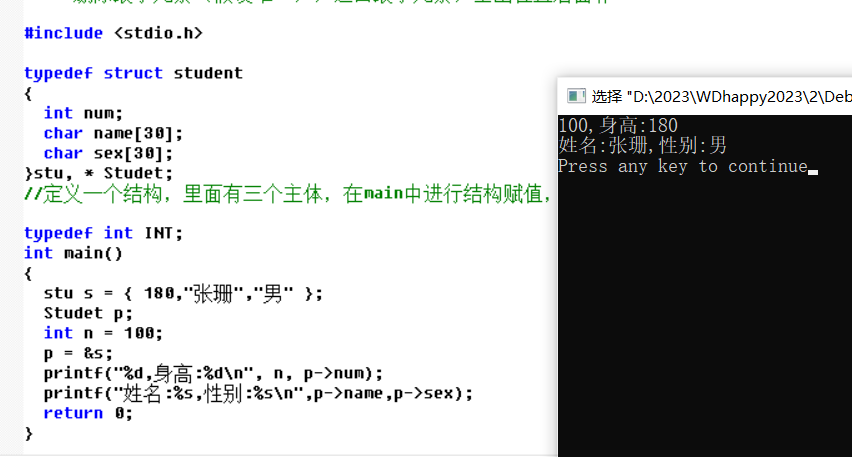
printf("姓名:%s,性别:%s\n",p->name,p->sex);

return 0;

}

2.

**\*是指针运算,&是取地址**



王道计算机练习：

顺序表的一系列操作：

顺序表插入（静态）：

#include <stdio.h>

#include <malloc.h>

#include <stdlib.h>

#define MaxSize 50

#define InitSize 100

typedef int ElemType;

typedef struct {

ElemType data[MaxSize]; //假定顺序表的元素类型ElemType

int length; //顺序表的当前长度

}SqList; //顺序表的类型定义

//静态分配。动态分配

typedef struct {

ElemType \*data; //指定动态分配数组的指针

int length; //数组的最大容量和当前个数

}SeqList; //动态分配数组顺序表的类型定义

//c:malloc,c++:new

//插入操作 i位置(1<=i<L.length+1)

bool ListInsert(SqList &L,int i,ElemType e){

if(i<1||i>L.length+1) //判断i的插入位置是否合法

return false;

if(L.length>=MaxSize) //判断存储空间是否已满。满则不插

return false;

for(int j=L.length;j>=i;j--) //开始是length-1~i,插入数据是i(j)~length-1+1

L.data[j]=L.data[j-1]; //i

L.data[i-1]=e;//i位置插入

L.length++;

return true;

}

int main()

{

SqList L;

int i=0;

//ElemType s[]={1,2,3,4,5};

//L.data[0]=1; //r C2106: '=' : left operand must be l-value

L.length=0;

for(i=0;i<4;i++){

L.data[i]=i;//赋初值

L.length++;

}

//L.data=(ElemType\*)malloc(ElemType)\*InitSize;//动态分配

int flag;

flag =ListInsert(L,2,6);//i代表插入位置是从1开始，不要和数组下标混淆

//printf("%d\n\n",flag);

//printf("%d\n\n",L.length);

for(i=0;i<L.length;i++){

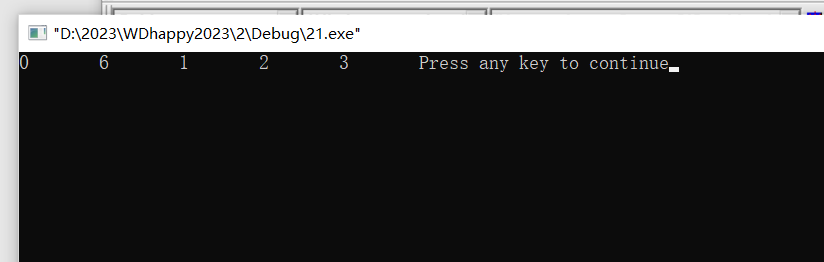
printf("%d\t",L.data[i]);

}

return 0;

}

在第二个位置插入6（位置2下标的1，在1 的位置插入，1之后元素向后移动）



对上面补充：（删除操作）

#include <stdio.h>

#include <malloc.h>

#include <stdlib.h>

#define MaxSize 50

#define InitSize 100

typedef int ElemType;

typedef struct {

ElemType data[MaxSize]; //假定顺序表的元素类型ElemType

int length; //顺序表的当前长度

}SqList; //顺序表的类型定义

//静态分配。动态分配

typedef struct {

ElemType \*data; //指定动态分配数组的指针

int length; //数组的最大容量和当前个数

}SeqList; //动态分配数组顺序表的类型定义

//c:malloc,c++:new

//11 插入操作 i位置(1<=i<L.length+1)

bool ListInsert(SqList &L,int i,ElemType e){

if(i<1||i>L.length+1) //判断i的插入位置是否合法

return false;

if(L.length>=MaxSize) //判断存储空间是否已满。满则不插

return false;

for(int j=L.length;j>=i;j--) //开始是length-1~i,插入数据是i(j)~length-1+1

L.data[j]=L.data[j-1]; //i后元素后移 i==>

L.data[i-1]=e;//i位置插入

L.length++;

return true;

}

//22 删除操作 i位置(1<=i<L.length)（范围和插入不同）

bool ListDelete(SqList &L,int i,ElemType &e){

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

return false;

e=L.data[i-1];//i位置删除，下标i-1+1后 即i后元素前移 i<==

for(int j=i;j<L.length;j++) //开始是length-1~i,插入数据是i(j)~length-1+1

L.data[j-1]=L.data[j]; //i

L.length--;

return true;

}

int main()

{

SqList L;

int i=0,e=0;

int flag=0;

L.length=0;

for(i=0;i<10;i++){

L.data[i]=i;//赋初值

L.length++; //length不能丢

}

ListInsert(L,2,121);//i代表插入位置是从1开始，不要和数组下标混淆

ListInsert(L,3,2333);

ListDelete(L,3,e);//一次性操作

for(i=0;i<L.length;i++){

printf("%d\t",L.data[i]);

}

return 0;

}

