**实验七指针类型实验（2学时）**

一、实验目的

1、熟练掌握掌握变量的直接访问方式和间接访问方式；

2、熟练掌握指针变量的定义与引用；

3、掌握指向一维数组的指针变量的定义和引用；

4、掌握指向二维数组元素的指针变量的定义和引用，指向二维数组行的指针变量的定义和引用。

5、掌握指针数组、指针的指针的定义与使用;

4、灵活运用指针类型处理实际问题。

**二、实验内容及要求**

1、实验内容包括8道题目，分为基本内容和选做内容两部分。要求：

（1）基本内容中的实验题目是必须要完成的，选做内容中的实验题目可以根据自己的情况选做或全做（越多越好，适当加分）；

（2）在实验之前编写出基本内容、选做内容中的所有程序，实验前检查（视未完成情况适当扣分）；

（3）对于设计型题目，要求画出N-S图；

（4）在上机实验过程中，注意总结和体会上机步骤（编辑、编译、连接、运行、调试、结果分析），注意培养观察、分析和排除编译错误、连接错误和程序逻辑错误的能力；

（5）上机结束后，应按要求认真书写实验报告（格式见附件）。

2、基本内容

（1）[验证型] 试分别输入并运行如下3个程序，然后分析结果。

**程序1**:

/\* 通过指针变量访问整型变量。

\*/

#include<stdio.h>

int main()

{

float a=100,b=10;

float \*pointer\_1, \*pointer\_2;

pointer\_1=&a;

pointer\_2=&b;

printf("a=%f,b=%f\n",a,b);

printf("\*pointer\_1=%f,\*pointer\_2=%f\n",

\*pointer\_1,\*pointer\_2);

return0;

}

**程序2**:

/\* 使用下标法和指针法引用数组元素

\*/

#include<stdio.h>

int main()

{

inta[10]; int \*p, i;

printf("Enter10 integer numbers:\n");

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

scanf("%d",&a[i]);

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

printf("%d",a[i]);

printf("%\n");

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

printf("%d ",\*(a+i));

printf("\n");

for(p=a;p<(a+10);p++)

printf("%d ",\*p);

printf("\n");

return0;

}

**程序3**:

/\* 二维数组的有关数据(地址和值)

\*/

#include<stdio.h>

int main()

{

inta[3][4]={2,4,6,8,10,12,14,16,18,20,22,24};

printf("%d,%d\n",a,\*a);

printf("%d,%d\n",a[0],\*(a+0));

printf("%d,%d\n",&a[0],&a[0][0]);

printf("%d,%d\n",a[1],a+1);

printf("%d,%d\n",&a[1][0],\*(a+1)+0);

printf("%d,%d\n",a[2],\*(a+2));

printf("%d,%d\n",&a[2],a+2);

printf("%d,%d\n",a[1][0],\*(\*(a+1)+0));

printf("%d,%d\n",\*a[2],\*(\*(a+2)+0));

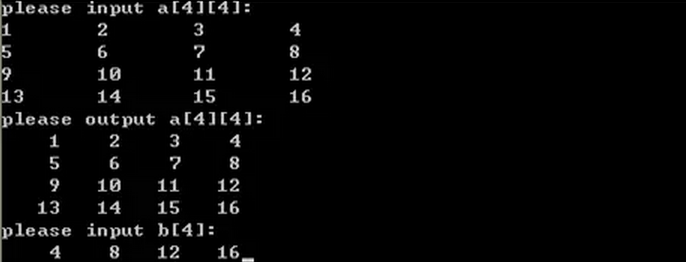
return 0;

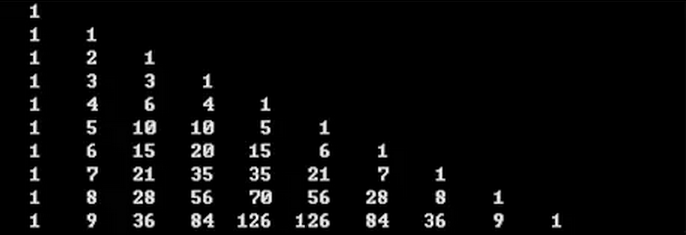
}

（2）[设计型] 输入n个整数，使用指针变量将这n个数按从小到大排序输出。运行结果示例如下：



（3）[设计型] 从键盘为数组a[4][4]输入数据，分别用指向数组元素的指针变量和指向一维数组的指针变量输入输出数组元素，求该数组每一行的最大值，将这些值保存在一维数组b[4]中。运行结果示例如下：

（4）[设计型]编写函数打印出10\*10的杨辉三角形（使用指针变量作函数参数）。运行结果示例如下：



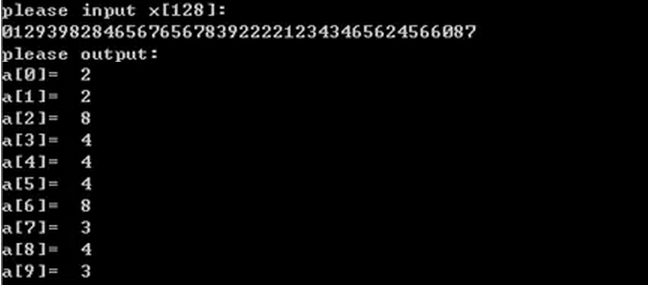
3、选做内容

（5）[设计型] 编写函数删除一维数组s中下标为k的元素的值。运行结果示例如下：



（6）[设计型] 编写函数把一个十进制正整数转换为二进制数。运行结果示例如下：

（7）[设计型] 编写函数实现输入一行数字字符，用数组元素作为计数器来统计每个数字字符的个数。用下标为0元素统计字符“0”的个数，用下标为1元素统计字符“1”的个数…。运行结果示例如下：



（8）[设计型] 编写程序从输入的若干字符串中找出最小的串输出。运行结果示例如下：



三、相关知识点及实验指导

（一）相关知识点

1、我们以前访问变量采用的都是**直接访问方式**，即在编译阶段系统给变量分配了相应的存储空间，同时也产生了一个变量名和内存地址关系的对照表，当我们通过变量名操作变量时，根据变量与内存地址对照表找到变量的地址进行相关的操作。在**间接访问方式**中，变量中保存的是另一个变量地址，通过该变量可以访问所保存地址空间中变量的内容，即访问其指向的变量。

2、指针变量定义的一般形式：类型说明符 \*指针变量名；

指针变量的引用方式：\*指针变量名

3、定义指向一维数组元素的指针变量的一般形式：

类型说明符 \*指针变量名=数组名；

通过\*(指针变量名++)可以访问数组中的其他元素。

4、二维数组是按行存储的，指向二维数组元素的指针变量的定义和定义指向一维数组元素的指针变量的定义相同。

定义指向二维数组行的指针变量的一般形式：

类型标识符 (\*指针变量名)[二维数组列数];

5、在C语言中，无论函数参数数据类型是什么，调用函数时实参和形参之间都是单向的值传递。当实参是普通变量时传递的是数值；当实参是变量的地址或指针变量时传递的是地址。但要注意实参可以是变量的地址也可以是指针变量，形参只能是指针变量（形参数组名按指针变量处理）。

6、数组名作函数参数实际上传递的数组的首地址（即指针），而不是将数组的所有元素传递给形参数组。函数调用时，实参将数组的首地址传递给形参指针，形参就指向了数组的首地址，通过间接访问方式，形参可以访问实参数组中的元素，更改数组元素的值。数组名做函数参数的实质是指针作函数参数。因此概括起来，传递一个数组实参和形参可以有4种形式，如下表所示：

数组名作函数参数的实参和形参对照表

|  |  |
| --- | --- |
| **实参类型** | **形参类型** |
| 数组名 | 数组名 |
| 数组名 | 指针变量 |
| 指向数组的指针变量 | 数组名 |
| 指向数组的指针变量 | 指针变量 |

7、将数组元素是指针的数组称为指针数组。

定义指针数组的一般形式：

类型说明符\*数组名[数组元素的个数]

一般情况下，利用数值型指针数组操作二维数组的情况很少，通常是利用字符指针数组操作二维字符数组。

8、将指针变量所指向的变量（也是指针变量）称作指向指针的指针。

定义指针的指针变量的一般形式：

类型说明符\*\*指针变量名

（二）实验指导

（略）

四、练习题

（1）编写程序把数组中的所有奇数放在另一个数组中。

（2）编写程序将方阵转置。

（3）编写函数将一个数插入一个有序数组中使数组中的数据仍然有序。

（4）找出方阵每列中的最小元素及其所在的行号。

（5）编写程序输入月份，输出该月的对应的英文名。