## 跟我学c练习题九

1. 程序分析。某同学的练习八第9题程序如下，请各位同学：
2. 指出程序中有问题之处，参考给出的表模板，在表中清楚地并说明原因，解决方法。
3. 改正并运行程序，给出测试结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 程序行 | 函数名 | 变量名 | 问题原因 | 解决方法 |
| 14 | input | p[100] | 该数组是局部变量，函数返回时数组所占的内存空间释放，只返回数组的首地址 | static int p[100]; |
| 19 | input | p[i] | scanf的格式中指定了int\*类型，而实际上传入了int类型 | scanf("%d",&p[i]); |

#include<stdio.h>

int \*input(int \*);

int \*findmax(int \*,int);

int main()

{

int \*p,len;

p=input(&len);

printf("Max=%d\n",\*(findmax(p,len)));

return(0);

}

//输入整数序列，返回该数组

int \*input(int \*len)

{

int p[100];

printf("输入序列长度：\n");

scanf("%d",len);

for(int i=0;i<\*len;i++){

printf("输入序列：\n");

scanf("%d",p[i]);

}

return(p);

}

//在整数序列中找最大值元素，返回该元素地址

int \*findmax(int \*p,int len)

{

int \*max=p;

for(int i=1;i<len;i++)if(\*max<p[i])max=&p[i];

return(max);

}

改正后的程序：

#include<stdio.h>

int \*input(int \*);

int \*findmax(int \*,int);

int main()

{

int \*p,len;

p=input(&len);

printf("Max=%d\n",\*(findmax(p,len)));

return(0);

}

//输入整数序列，返回该数组

int \*input(int \*len)

{

static int p[100];

printf("输入序列长度：\n");

scanf("%d",len);

for(int i=0;i<\*len;i++){

printf("输入序列：\n");

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

}

return(p);

}

//在整数序列中找最大值元素，返回该元素地址

int \*findmax(int \*p,int len)

{

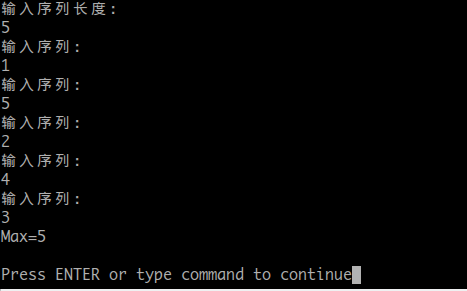
int \*max=p;

for(int i=1;i<len;i++)if(\*max<p[i])max=&p[i];

return(max);

}

测试结果：



1. 函数编程。主函数有2个整数变量i\_a和i\_b，而指针ip1指向i\_a，ip2指向i\_b，编程：
2. 以ip1和ip2作为实参，调用input()函数从键盘读入2个整数并通过ip1和ip2返回给主函数的i\_a和i\_b；
3. 以ip1和ip2作为实参，调用SwapPoint()函数，交换2个指针的指向（ip1指向i\_b，ip2指向i\_a）。
4. 主函数调用SwapPoint()函数之后，执行如下语句：

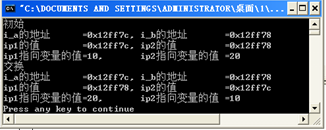
printf("交换\n");

printf("i\_a的地址=%#x, i\_b的地址=%#x\n",&i\_a,&i\_b);

printf("ip1的值=%#x, ip2的值=%#x\n",ip1,ip2);

printf("ip1指向变量的值=%d, ip2指向变量的值 =%d\n",\*ip1,\*ip2);

可在调用SwapPoint()之前，也插入上述语句，作交换指针指向的对比。参考如下：



1. 指针访问一维数组。用指针代替下标运算符操作数组：
2. input函数从健盘输入n个整数，返回数组及n；
3. add函数将数组每个元素值是奇数的加1，偶数则减1，最后输出数组的所有元素。
4. 程序调试练习。整型数组a定义如下，在表格第二列中说明第一列中的表示形式所表达的意义，并在第三列中写出它的数值（编一简短程序用DEBUG单步跟踪）。

int a[3][4];

|  |  |  |
| --- | --- | --- |
| 表示形式 | 含义 | 数值 |
| a | 二维数组名，数组首地址 | 0x7fffffffe3f0 |
| a[0],\*(a+0),\*a | 二维数组中下标为0的元素，即一个一维数组的首地址 | 0x7fffffffe3f0 |
| a+1 | 二维数组中下标为1的元素，即一个一维数组的首地址 | 0x7fffffffe400 |
| a[1],\*(a+1) | 二维数组中下标为1的元素，即一个一维数组的首地址 | 0x7fffffffe400 |
| a[1]+2,\*(a+1)+2,&a[1][2] | 二维数组中下标为1的一维数组中下标为2的元素的地址 | 0x7fffffffe408 |
| \*(a[1]+2),\*(\*(a+1)+2),a[1][2] | 二维数组中下标为1的一维数组中下标为2的元素的值 | 23 |

1. 二维数值型数组。编写一个程序，声明一个3x5的数组并初始化，数值随意。程序打印出数值，然后所有元素的数值翻倍，再次打印新的数值。用函数output()输出数组内容，函数Double()执行数值翻倍功能。数组名和行数作为参数传递给各函数。
2. 二维数值型数组。编写一个程序，用户输入三个数集，每个数集包括5个double值。程序实现下述功能：
3. 调用intput()把输入信息存入到一个3x5的数组中；
4. 调用average\_row()计算出每个数集（包含5个double数值）的平均值；
5. 调用averag\_All()计算所有数的平均值；
6. 调用max()找出这15个数中的最大值；
7. 调用list()函数打印出各个功能结果。

注：对于功能（2），需要编写计算并返回一维数组平均值的函数。循环三次实现功能（2）。对于其他功能，各函数应该把二维数组作为参数传递，并且完成功能（3）和（4）的函数应该向它的调用函数返回答案。

1. 指针数组排序。请把教材程序8.10中的函数comp的功能用冒泡算法实现（参考教材程序8.11）：
2. input从键盘任意输入5个英文单词（设每个单词字符串的长度小于20）返回到二维数组word
3. 指针数组p指向word
4. 调用排序函数bubbleSort，直接对p排序
5. bubbleSort函数不能改变各个字符串在word的位置
6. 按字典编辑顺序打印在屏幕上。