云南大学数学与统计学实验教学中心  
《高级语言程序设计》实验报告

|  |  |  |
| --- | --- | --- |
| 课程名称：程序设计和算法语言 | 学期：2016~2017学年上学期 | 成绩： |
| 指导教师：赵越 | 学生姓名：刘鹏 | 学生学号：20151910042 |
| 实验名称：指针程序设计(2) |  |  |
| 实验编号：No.08 | 实验日期：2017年8月20日 | 实验学时：2 |
| 学院：数学与统计学院 | 专业：信息与计算科学 | 年级：2015级 |

# 实验目的

1. 掌握指针变量的定义，会使用指针变量。
2. 掌握指针与变量、指针与数组、指针与字符串的关系。

# 实验环境

Windows10 Pro Workstation 17096；

Cygwin GCC编译器。

# 实验内容

## 知识准备

* 指针就是地址。指针变量是存放另一个变量的地址的变量，不要把指针和指针变量两个概念混淆。
* 定义指针变量后，一定要给指针变量赋初值，使用没有初始指向的指针是非常危险的。
* 指针变量作函数形参，可以接受来自实参的值（地址）。

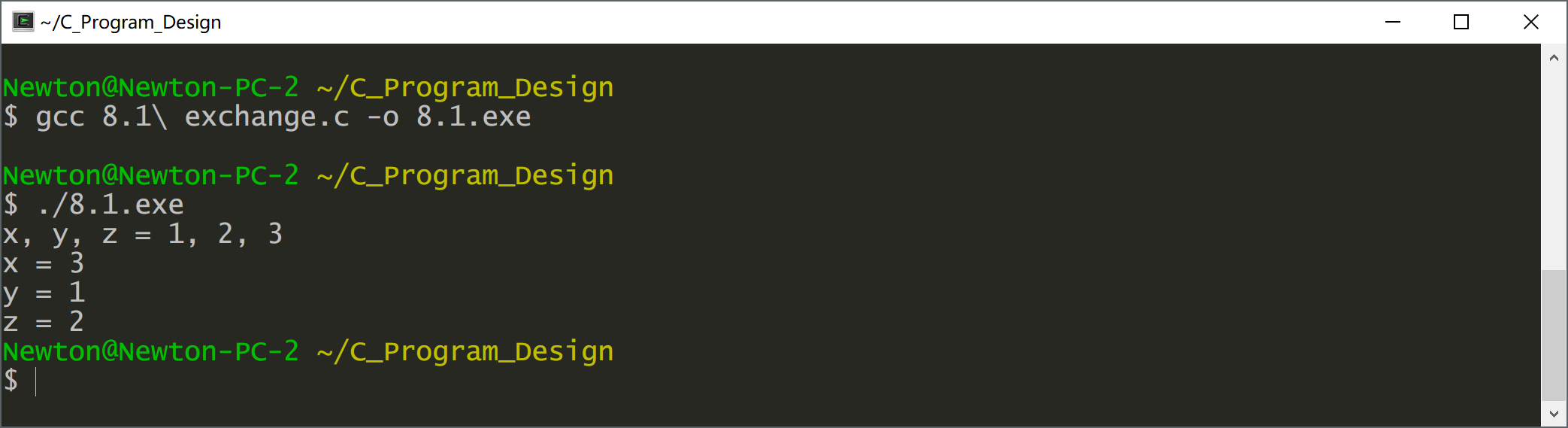
## 元素换位

有三个整数，，，设置三个指针变量，，，分别指向，，，然后通过指针变量使，，，三个变量交换顺序，即原来的值给，把的值给，的值给。，，的原值由键盘输入，要求输出，，的原值和新值。读懂下面程序，并上机运行，把结果打印出来。

### 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | /\*  \* filename: 8.1 exchange.c  \* property: test  \*/  #include <stdio.h>  int main**()** **{**  int x**,** y**,** z**,** t **;**  int **\***p1**,** **\***p2**,** **\***p3**;**  printf**(**"x, y, z = "**);**  scanf**(**"%d, %d, %d"**,** **&**x**,** **&**y**,** **&**z**);**  p1 **=** **&**x**;**  p2 **=** **&**y**;**  p3 **=** **&**z**;**  t **=** **\***p3**;**  **\***p3 **=** **\***p2**;**  **\***p2 **=** **\***p1**;**  **\***p1 **=** t**;**  printf**(**"x = %d\ny = %d\nz = %d"**,**x**,** y**,** z**);**  **return** 0**;**  **}** |

### 运行结果



请学生考虑第13~15行为什么不写成：\*p1 = &x;\*p2 = &y;\*p3 = &z;而第17行~20行不写成：t = p3;p3 = p2;p2 = p1;p1 = t。

答：写成\*p1=&x;\*p2=&y,\*p3=&z;是错误的，地址本身是一个数值，应该赋给一个变量。而对于指针变量而言，其变量名不带\*；

程序中的互换是发生在实际变量上的，只是通过指针变量进行了操作。由于输出的是变量的值，而且是直接引用，所以交换指针的值是行不通的，而且t没有定义指向，是个危险地址。

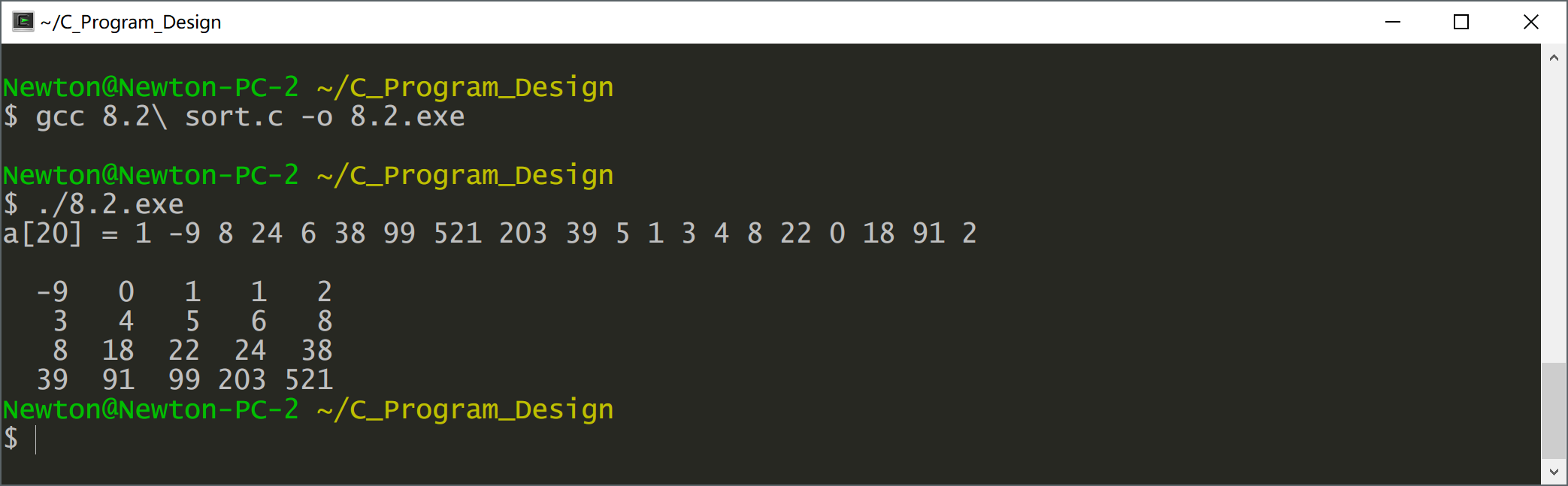
## 序列排序

编写一个函数，使20个整数由小到大的顺序排列。在主函数中输出排好序的数。本程序编写函数部分要求要用指针来解决。

### 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | /\*  \* filename: 8.2 sort.c  \* property: exercise  \*/  #include<stdio.h>  void sort**(**int array**[],** int n**)** **{**  int **\***p1**,** **\***p2**,** t**;**  **for** **(**p1 **=** array**;** p1 **<** array **+** **(**n **-** 1**);** p1**++)** **{**  **for** **(**p2 **=** p1 **+** 1**;** p2 **<** array **+** n**;** p2**++)** **{**  **if(\***p1 **>** **\***p2**)** **{**  t **=** **\***p1**;**  **\***p1 **=** **\***p2**;**  **\***p2 **=** t**;**  **}**  **}**  **}**  **}**  int main**()** **{**  int **\***p**,** a**[**20**],** i**;**  printf**(**"a[20] = "**);**  **for** **(**i **=** 0**;** i **<** 20**;** i**++)** **{**  scanf**(**"%d, "**,** **&**a**[**i**]);**  **}**  sort**(**a**,** 20**);**  **for** **(**p**=**a**;**p**<**a**+**20**;**p**++)** **{**  **if((**p **-** **&**a**[**0**])** **%** 5 **==** 0**)** **{**  printf**(**"\n"**);**  **}**  printf**(**"%4d"**,** **\***p**);**  **}**  **return** 0**;**  **}** |

### 运行结果



通过此题，学生可进一步理解指针的定义与赋初值的方法。可以思考一下选择排序与冒泡排序的区别，本质上都是两个循环，都是从左往右进行排序，只不过选择排序是间接通过下标进行记录，冒泡排序是直接进行交换，并且可能进行多次交换。

## 矩阵转置输出

有一个3\*4的矩阵，矩阵元素为：

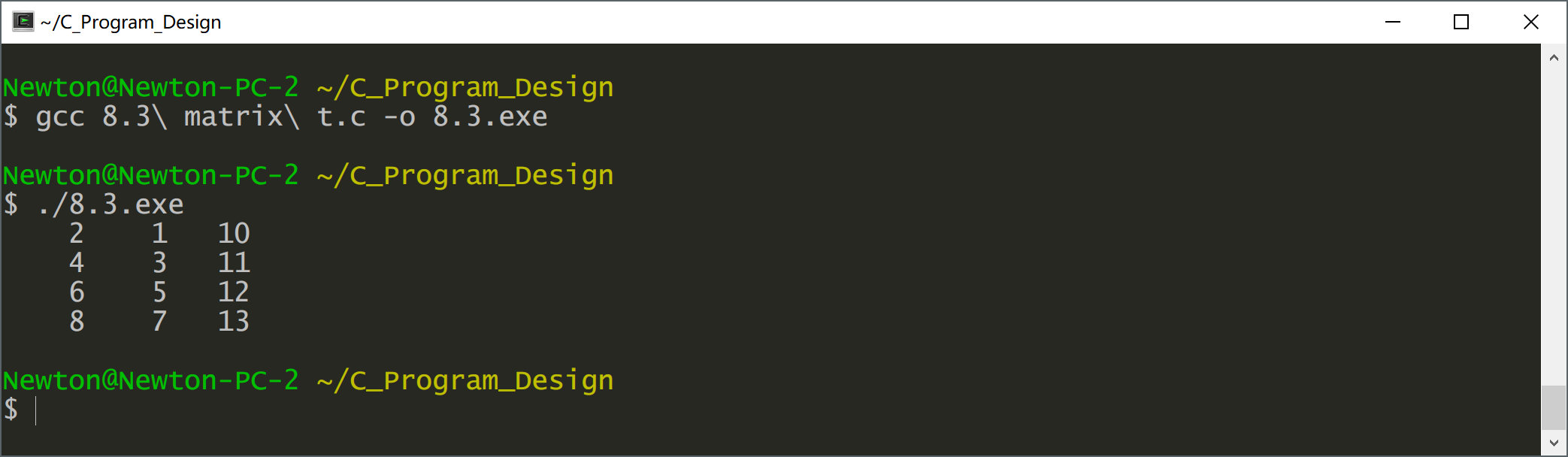


编写一程序实现矩阵的转置。上机调试并运行。

### 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | /\*  \* filename: 8.3 matrix t.c  \* property: exercise  \*/  #include <stdio.h>  void Transport**(**int **(\***a**)[**4**],** int **(\***b**)[**3**])** **{**  int i**,** j**;**  **for** **(**i **=** 0**;** i **<** 3**;** i**++)** **{**  **for** **(**j **=** 0**;** j **<** 4**;** j**++)** **{**  **\*(\*(**b **+** j**)** **+** i**)** **=** **\*(\*(**a **+** i**)** **+** j**);**  **}**  **}**  **}**  int main**()** **{**  int i**,** j**;**  int a**[**3**][**4**]** **=** **{{**2**,** 4**,** 6**,** 8**},** **{**1**,** 3**,** 5**,** 7**},** **{**10**,** 11**,** 12**,** 13**}};**  int b**[**4**][**3**];**  Transport**(**a**,** b**);**    **for** **(**i **=** 0**;** i **<** 4**;** i**++)** **{**  **for** **(**j **=** 0**;** j **<** 3**;** j**++)** **{**  printf**(**"%5d"**,** b**[**i**][**j**]);**  **}**  printf**(**"\n"**);**  **}**  **return** 0**;**  **}** |

### 运行结果



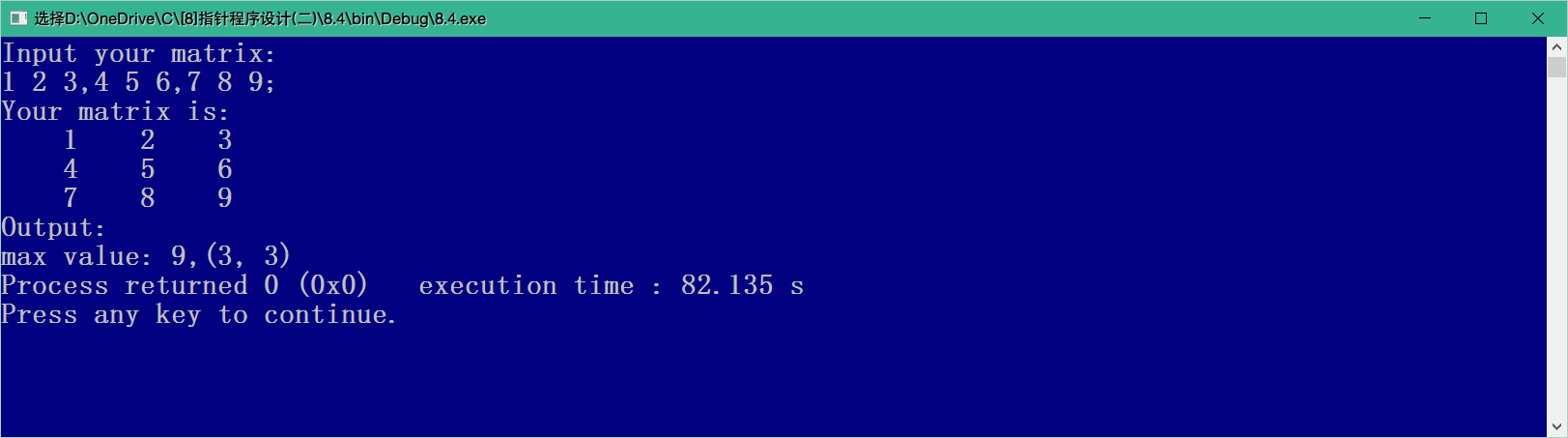
## 矩阵全主元

有一个整型二维数组，大小为，找出最大值所在的行和列，以及该最大值。此程序要求用一个函数实现最大值的寻找，并在函数中最好使用指针解决，和为该函数的形参，数组元素的值在主函数中输入。

### 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69 | /\* filename: 8.4 \*/  #include<stdio.h>  int rn**,**cn**,**r**,**c**,**maxval**;**  int a**[**20**][**20**];**  void input**()**  **{**  int i**,**j**;**char m**;**  rn**=**0**;**  cn**=**1**;**  printf**(**"Input your matrix:\n"**);**  **for(**i**=**0**;**i**<**20**;**i**++)**  **{**  cn**=**1**;**  **for(**j**=**0**;**j**<**20**;**j**++)**  **{**  scanf**(**"%d"**,&**a**[**i**][**j**]);**  scanf**(**"%c"**,&**m**);**  **if(**m**==**','**||**m**==**';'**)**  **{**  rn**++;**  **break;**  **}**  **else**  **{**  cn**++;**  **}**  **}**  **if(**m**==**';'**)**  **{**  **break;**  **}**  **}**  printf**(**"Your matrix is:\n"**);**  **for(**i**=**0**;**i**<**rn**;**i**++)**  **{**  **for(**j**=**0**;**j**<**cn**;**j**++)**  **{**  printf**(**"%5d"**,**a**[**i**][**j**]);**  **}**  printf**(**"\n"**);**  **}**  **}**  void max**()**  **{**  int i**,**j**;**  maxval**=**a**[**0**][**0**];**  **for(**i**=**0**;**i**<**rn**;**i**++)**  **{**  **for(**j**=**0**;**j**<**cn**;**j**++)**  **{**  **if(**maxval**<\*(\*(**a**+**i**)+**j**))**  **{**  maxval**=\*(\*(**a**+**i**)+**j**);**  r**=**i**+**1**;**  c**=**j**+**1**;**  **}**  **}**  **}**  **}**  int main**()**  **{**  input**();**  max**();**  printf**(**"Output:\nmax value: %d,(%d, %d)"**,**maxval**,**r**,**c**);**  **return** 0**;**  **}** |

### 运行结果



利用指针型参数，写一个函数，交换和的值。

用指针实现下列功能：从键盘输入一字符串,求此串的长度并输出。

## \*

在主函数中输入10个等长的字符串。用另一个函数对它们排序，然后输出这10个已排好序的字符串。要求用指针解决问题，编好程序上机运行。

## 程序排错

1．下列语句中有哪些上不正确的？

(1) int **\***p**=&**i**,**i**,**j**;**

必须先定义变量，之后将定义的变量的地址存入指针变量，这里的i定义应该在\*p之前

(2)float x**,**y**;**

int **\***p**;**

p**=&**x**;**

指针基类型与指针指向的变量类型不一致。

(3)int i**,**j**;**

i**=**10**;**j**=**20**;**

int **\***p**-&**i**,\***q**=&**j**;**

地址加减的生成值没有变量存放。

(4) int i**,**j**,\***p**[**3**]={**1**,**2**,**3**},\***q**=&**i**+**2**;**

指针数组是不能这样赋值的。

(5) int m,n,\*p,a[10];p=&a[10];

a[10]不是一个数组元素，取地址没有意义。

(6) int **\***p**=**128**;**

对指针进行赋值没有意义。

(7) int a=10,\*p=a;

应该是&a。

(8) fun()

{

int i,j;

static int \*pi=&i;

}

不会。

(9) int i=10,\*pi;\*pi=i;

赋值形似不对。

(10) char **\***ps**,**s**[**20**];** s**=**"good"**;** ps**=**s**;**

s是数组名字，是静态的，不能被赋值。

先想想，再上机试试，下面加程序输出什么结果。为什么？先看下面程序两种处理方式的差别，再上机验证一下你的判断.

### 程序代码

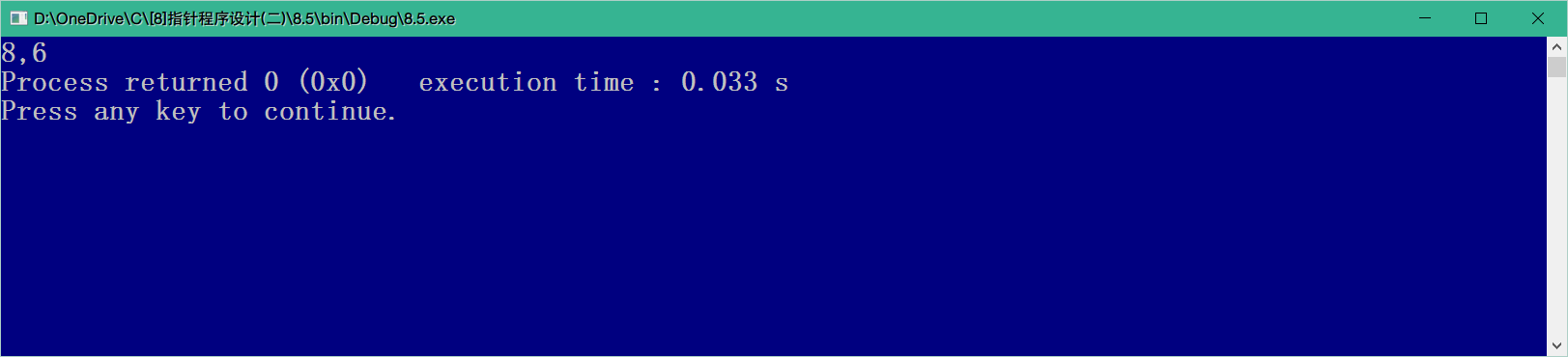
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | #include<stdio.h>  int main**()**  **{**  int i**,**a**[**10**],**b**[**10**],\***p**=**b**,\***q**=&**a**[**0**];**  **for** **(**i**=**0**;**i**<**10**;**i**++)**  **{**  scanf**(**"%d"**,&**a**[**i**]);**  **}**  **for** **(**i**=**0**;**i**<**10**;**i**++,**q**++)**  **{**  p**[**i**]=\***q**;**  **}**  **for** **(**i**=**0**;**i**<**10**;**i**++)**  **{**  printf**(**"%d"**,**a**[**i**]);**printf**(**"\n"**);**  **}**  **for(**i**=**0**;**i**<**10**;**i**++)**  **{**  printf**(**"%d"**,**b**[**i**]);**  **}**  printf**(**"\n"**);**  **for(**i**=**0**;**i**<**10**;**i**++,**p**++);**  **{**  printf**(**"%7d"**,\***p**);**  **}**  printf**(**"\n"**);**  **return** 0**;**  **}** |

# 实验过程

## 1题

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | /\* filename: 8.5 \*/  #include <stdio.h>  void swap**(**int **\***a**,**int **\***b**)**  **{**  int tmp**;**  tmp**=\***a**,\***a**=\***b**,\***b**=**tmp**;**  **}**  int main**()**  **{**  int a**=**6**,**b**=**8**;**  int **\***p1**=&**a**,\***p2**=&**b**;**  swap**(**p1**,**p2**);**  printf**(**"%d,%d"**,**a**,**b**);**  **return** 0**;**  **}** |

### 运行结果



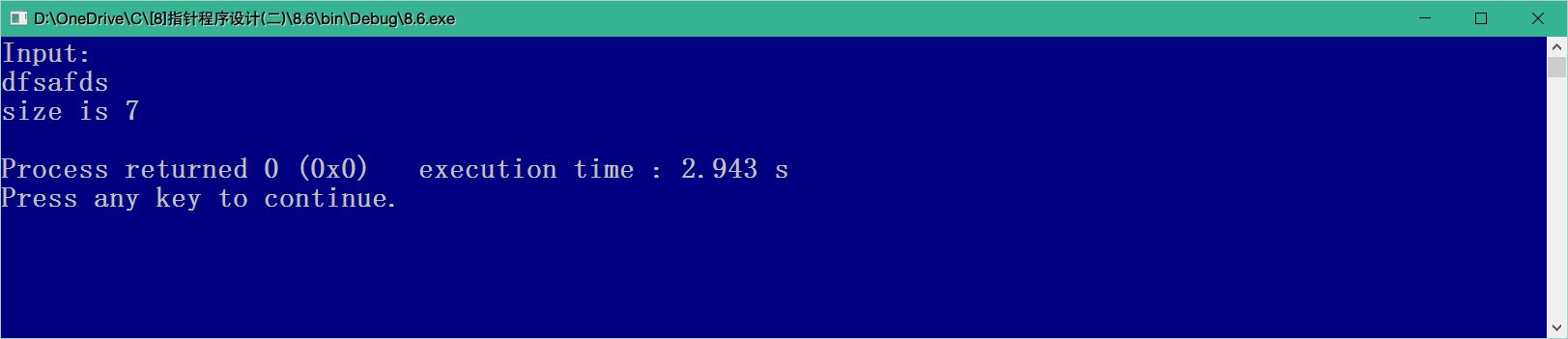
## 2题：

这个例子考察函数的用法，输入字符串是可以用scanf与gets的，不同的是，前者在定义的时候需要声明格式，%s只能输入没有空格的字符串，而%[^\n]就可以，一直输入（只要不超界），直到遇到换行符才停止。gets的用法与scanf的%[^\n]基本一致。



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | /\* filename: 8.6 \*/  #include<stdio.h>  void PrintSize**(**char a**[])**  **{**  char **\***p**;**  int n**=**0**;**  **for(**p**=**a**;\***p**!=**'\0'**;**p**++)**  **{**  n**++;**  **}**  printf**(**"size is %d\n"**,**n**);**  **}**  int main**()**  **{**  char a**[**30**];**  printf**(**"Input:\n"**);**  gets**(**a**);**  /\*scanf("%[^\n]",a);\*/  PrintSize**(**a**);**  **return** 0**;**  **}** |

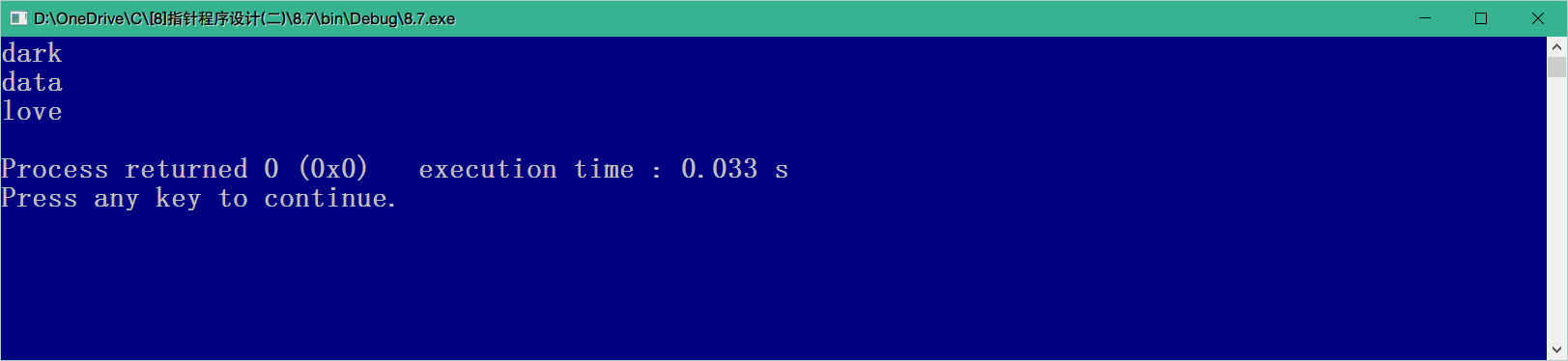
### 运行结果



## 3题:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | /\* filename: 8.7 \*/  #include<stdio.h>  #include<string.h>  void Sort**(**char **\***p**[])**  **{**  int i**,**j**;**  char **\***m**=NULL;**  **for(**i**=**0**;**i**<**2**;**i**++)**  **{**  **for(**j**=**i**+**1**;**j**<**3**;**j**++)**  **{**  **if(**strcmp**(**p**[**i**],**p**[**j**])>**0**)**  **{**  m**=**p**[**i**];**  p**[**i**]=**p**[**j**];**  p**[**j**]=**m**;**  **}**  **}**  **}**  **}**  void Print**(**char **\***name**[])**  **{**  int i**;**  **for(**i**=**0**;**i**<**3**;**i**++)**  **{**  printf**(**"%s\n"**,**name**[**i**]);**  **}**  **}**  int main**()**  **{**  char **\***name**[]={**"data"**,**"love"**,**"dark"**};**  Sort**(**name**);**  Print**(**name**);**  **return** 0**;**  **}** |

### 运行结果



# 实验总结

由于指针可以对地址进行存储，所以在函数较多的情况下，使用指针可以降低程序运行的内存占用。也由于这个，可以对非全局变量进行函数操作，特别是实参与形参是值传递的情况。

随着时间的推移，我的编译器选择历经轮转，从最初的古老的TC2.0，到Code::Blocks集成开发环境，再到Visual Studio 2017，最终还是回到了GNU平台上来，使用开源的一套库进行实验。在此期间，云南大学也从一个普通的211大学跻身双一流大学行列，高级语言程序设计这门课程是否也该升级一下？TC2.0这个编译器，界面十分古朴，在几十年前绝对算是一流的软件，但是现在，确实落后了，这主要是因为它无法编译在64位系统下运行的程序。但是TC2.0有很多的优势，比如完全可视化的编译过程，不会生成很多附加文件，这一点Visual Studio就太过专业化。经过对《UNIX环境高级编程》[1]这本书的学习，还有诸如*Harley Hahn's Guide to Unix and Linux*[2]这本书的阅读，我觉得基于Shell的UNIX环境似乎是最适合新手学习的。

本次实验，集中主要精力，在以前版本的基础上，对文档结构进行了重整，看起来自然了很多，目录也规范了很多。有关编程的规范性问题，参考林锐高质量C/C++编程指南的第一版[3]。

# 参考文献

1. Stevens, W.R. and S.A. Rago, *UNIX环境高级编程*. 2nd ed. 2005, 北京: 人民邮电出版社.

2. Hahn, H., *Harley Hahn's Guide to Unix and Linux*. 2009, New York: McGraw-Hill.

3. **林锐**, *高质量 C++/C 编程指南*. 1.0 ed. 2001.

# 教师评语