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：**数组**程序设计 |  |  |
| 实验编号：No.05 | 实验日期：2018年8月13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. 掌握数组的概念和使用方法。
2. 掌握数组初始化的方法。
3. 学会字符数组和字符串的应用。
4. 学会用数组名作函数的参数。
5. 掌握一维数组与二维数组的定义及其元素的引用方法。
6. 深刻体会数组与循环的关系。
7. 掌握利用一维数组和二维数组实现一些常用算法的编程技巧。
8. 进一步掌握动态调试的基本技能。

# 实验环境

Windows10 Pro Workstation 17096；

Code::Blocks 16.01 GCC集成开发环境；

Cygwin GCC编译器。

# 实验内容

## 有关概念

* + - 1. 只有静态数组和外部数组才能初始化。

1. 引用数组时，编译器对下标是否越界不作检查。如定义int a[5];在引用时出现a[5];不给出错信息，而是引a[4]下面一个单元的值。
2. 字符串放在字符数组中，一个字符串以'\0'结束，有一些字符串函数如strcpy，strcmp，strlen等可以方便进行字符串运算。
3. 如有如下定义：char \*str = "I love china";表示str是一个字符型指针变量，它的值是一个字符数据的地址。不要认为str是字符串变量，在其中存放一个字串"I love china"。
4. 用数组名作函数实参时，传到形参的是数组的首地址。

## 题1

定义三个数组

int a**[**5**];**

int b**[**2**][**2**];**

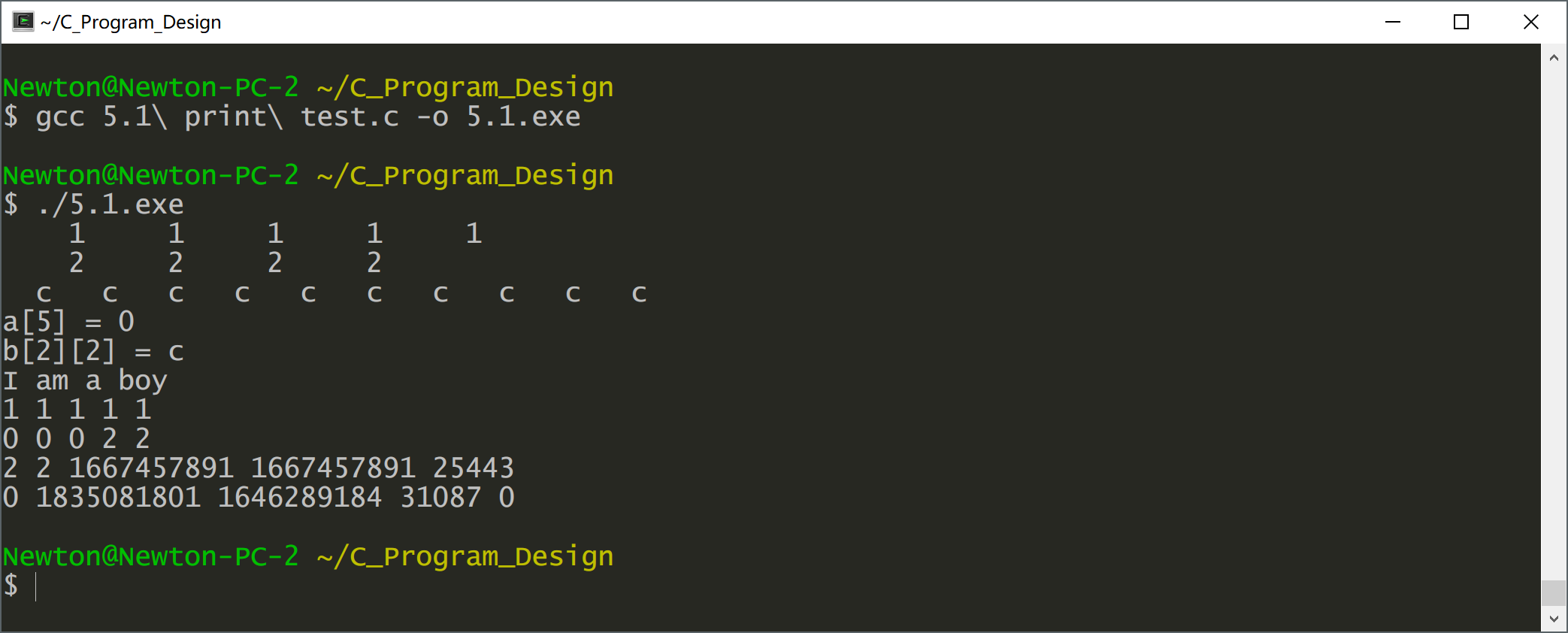
char c**[**10**];**

分别在函数体外和函数体内对它们进行初始化，然后输出它们的值。在程序中再加一语句，输出a**[**5**],** b**[**2**][**2**]**，分析结果。对c数组改为用赋值语句给各元素赋初值：c[0]～c[9]各元素分别为：'I'**,** ' '**,** 'a'**,** 'm'**,** ' '**,** 'b'**,** 'o'**,** 'y'。然后用printf**(**"%s"**,** c**)**输出字符串，分析结果。

### 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 | /\*  \* filename: 5.1 print test.c  \* property: test  \*/  #include <stdio.h>  int main**()** **{**  int i**,** x**,** y**;**  static int a**[**5**]** **=** **{**1**,** 1**,** 1**,** 1**,** 1**};**  static int b**[**2**][**2**]={{**2**,** 2**},** **{**2**,** 2**}};**  static char c**[**10**]** **=** **{**'c'**,** 'c'**,** 'c'**,** 'c'**,** 'c'**,** 'c'**,** 'c'**,** 'c'**,** 'c'**,** 'c'**};**  **for(**i **=** 0**;** i **<** 5**;** i**++)**  printf**(** "%5d "**,** a**[**i**]);**  printf**(**"\n"**);**  **for(**x **=** 0**;** x **<** 2**;** x**++)**  **for(**y **=** 0**;** y **<** 2**;** y**++)**  printf**(**"%5d "**,**b**[**x**][**y**]);**  printf**(**"\n"**);**  **for(**i **=** 0**;** i **<** 10**;** i**++)**  printf**(**"%3c "**,** c**[**i**]);**  printf**(**"\n"**);**  printf**(**"a[5] = %d\n"**,** a**[**5**]);**  printf**(**"b[2][2] = %c\n"**,** b**[**2**][**2**]);**  static char c2**[**10**]** **=** "I am a boy"**;**  printf**(**"%s\n"**,** c2**);**    **for(**int i**=**0**;** i **<** 20**;** i**++)** **{**  printf**(**"%d "**,** a**[**i**]);**  **if((**i**+**1**)** **%** 5 **==** 0**)** **{**  printf**(**"\n"**);**  **}**  **}**  **return** 0**;**  **}** |

### 运行结果



## 排序

有一数组，内放10个整数，要求找出最小的数和它的下标。然后把它和数组中最前面的元素对换位置。

编写程序，上机运行，并记录下结果。提示：数组的10个元素可用输入函数scanf()通过键盘输入进去，找出数组中最小的元素可通过循环语句和条件语句来实现。

设min是存放数组中最小元素的变量，array[k]为一个暂存单元。实现最前面的元素与最小元素对换可通过下面语句实现：

array**[**k**]=**array**[**0**];**

array**[**0**]=**min**;**

### 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 | #include<stdio.h>  int main**()**  **{**  int i**,**array**[**10**],**min**,**k**=**0**;**  printf**(**"Please input 10 data\n"**);**  **for(**i**=**0**;**i**<**10**;**i**++)**  scanf**(**"%d"**,&**array**[**i**]);**  printf**(**"Before exchang:\n"**);**  **for(**i**=**0**;**i**<**10**;**i**++)**  printf**(**"%5d"**,**array**[**i**]);**  min**=**array**[**0**];**  printf**(**"\n"**);**  **for(**i**=**1**;**i**<**10**;**i**++)**  **if(**min**>**array**[**i**])**  **{**  min**=**array**[**i**];**  k**=**i**;**  **}**  array**[**k**]=**array**[**0**];**  array**[**0**]=**min**;**  printf**(**"After exchange:\n"**);**  **for(**i**=**0**;**i**<**10**;**i**++)**  printf**(**"%5d"**,**array**[**i**]);**  printf**(**"\n"**);**  printf**(**"k=%d\t min=%d\n"**,**k**,**min**);**  **return** 0**;**  **}** |

### 运行结果

在一个已排好序的数列中（由小到大）再插入一个数，要求仍然有序。编程并上机运行。提示：编程时应考虑到插入的数的各种可能性（比原有所有的数大；比原有所有的数小；在最大数和最小数之间）。

### 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 | #include<stdio.h>  int main**()**  **{**  int i**,**n**;**  float a**,**x**[**20**],**y**[**21**];**  printf**(**"Please input n value\n"**);**  scanf**(**"%d"**,&**n**);**  printf**(**"Please input value (from small to big)\n"**);**  **for(**i**=**0**;**i**<**n**;**i**++)**  scanf**(**"%f"**,&**x**[**i**]);**  printf**(**"Insert value=?"**);**  scanf**(**"%f"**,&**a**);**  i**=**0**;**  **while(**a**>**x**[**i**]&&**i**<**n**)**  **{**  y**[**i**]=**x**[**i**];**  i**++;**  **}**  y**[**i**]=**a**;**  **for(**i**=**i**+**1**;**i**<**n**+**1**;**i**++)**  y**[**i**]=**x**[**i**-**1**];**  printf**(**"\n"**);**  **for(**i**=**0**;**i**<**n**+**1**;**i**++)**  **{**  printf**(**"%8.2f"**,**y**[**i**]);**  **if((**i**+**1**)%**5**==**0 **)** puts**(**"\n"**);**  **}**  **return** 0**;**  **}** |

### 运行结果

编写一程序，一班级有*n*名学生要求按他们姓名的顺序排列（按汉语拼音的字母顺序从小到大），并按序输出。

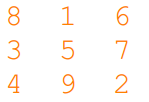
### 程序代码

|  |  |
| --- | --- |
| 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 | #include<stdio.h>  void strup**(**char str**[])**  int main**()**  **{**  char str**[**20**];**  char name**[**20**][**20**];**  int i**,**j**,**t**,**n**;**  printf**(**"Please input name number of sorting\n"**);**  scanf**(**"%d"**,&**n**);**  printf**(**"Please input name\n"**);**  **for(**i**=**0**;**i**<**n**;**i**++)**  **{**  gets**(**name**[**i**]);**  strup**(**name**[**i**]);**  **}**  **for(**i**=**0**;**i**<**n**;**i**++)**  **{**  **for(**j**=**i**+**1**;**j**<**n**;**j**++)**  **{**  **for(**k**=**0**;;**k**++)**  **if(**name**[**i**][**k**]<**name**[**j**][**k**])**  **break;**  **else**  **if(**name**[**i**][**k**]>**name**[**j**][**k**])**  **{**  strcpy**(**str**,**name**[**j**]);**  srycpy**(**name**[**j**],**name**[**i**]);**  strcpy**(**name**[**i**],**str**);**  **break;**  **}**  **}**  **}**  **for(**i**=**0**;**i**<**n**;**i**++)**  printf**(**"%s"**,**name**[**i**]);**  **}**  void strup**(**char str**[])**  **{**  int i**;**  **for(**i**=**0**;**str**[**i**]!=**'\o'**;**i**++)**  **if(**str**[**i**]>=**'a'**&&**str**[**i**]<=**'z'**)**  str**[**i**]=**str**[**i**]+**'A'**-**'a'**;**  **}** |

### 运行结果

## \* 打印魔方阵

所谓魔方阵是指，它的每行每一列的和与对角线之和均相等。例如，三阶魔方阵为



要求打印由到的自然数构成的魔方阵。

提示：魔方阵中各数排列规律为：

将“1”放在第一行中间一列；

从“2”开始直到止各数依次按下列规则存放：每一个数存放的行比前一个数的行数减1，列数加1；

如果上一数的行数为1，则下一个数的行数应为n（指最下一行）；

当上一个数的列数为n时，下一个数的列数应为1，行数减1。

如果按上面的规则 确定的位置上已有数，或上一个数是第1行第*n*列时，则把下一个数放在上一个数的下面。

|  |  |
| --- | --- |
| 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 | #include<stdio.h>  int main**()**  **{**  int a**[**16**][**16**],**i**,**j**,**k**,**p**,**m**,**n**;**  p**=**1**;** /\* 初始化 \*/  **while(**p**==**1**)**  **{**  printf**(**"Please input n:\n"**);**  scanf**(**"%d"**,&**n**);**  **if** **(**n**!=**0**)&&(**n**<=**15**)&&(**n**%**2**!=**0**)**  **{**  printf**(**"矩阵阶数是%d\n"**,**n**);**  p**=**0**;**  **}**  **}**  **for(**i**=**1**;**i**<=**n**;**i**++)**  **for(**j**=**1**;**j**<=**n**;**j**++)**  a**[**i**][**j**]=**0**;**  j**=**n**/**2**+**1**;** /\* 建立魔方面\*/  a**[**1**][**j**]=**1**;**  **for(**k**=**2**;**k**<=** n**\***n**;**k**++)**  **{**  i**=**i**-**1**;**  j**=**j**+**1**;**  **if((**i**<**1**)&&(**j**>**n**))**  **{**  i**=**i**+**2**;**  j**=**j**-**1**;**  **}**  **else**  **{**  **if(**i**<**1**)** i**=**n**;**  **if(**j**>**n**)** j**=**1**;**  **}**  **if(**a**[**i**][**j**]==**0**)**  a**[**i**][**j**]=**k**;**  **else**  **{**  i**=**i**+**2**;**  j**=**j**-**1**;**  a**[**i**][**j**]=**k**;**  **}**  **}**  **for(**i**=**1**;**i**<=**n**;**i**++)** /\* 输出 \*/  **{**  **for(**j**=**1**;**j**<=**n**;**j**++)**  **{**  printf**(**"%4d"**,**a**[**i**][**j**]);**  printf**(**"\n"**);**  **}**  **}**  **}** |

### 运行结果

## 元素移位

请按以下步骤实习和思考：

①分析程序及其特性。

②上机运行程序，查看运行结果是否正确？

③用动态跟踪查找错误原因，即按如下操作：

首先将光标移至t=a[9]的语句行上，按F4，再用Ctrl + F7操作将a数组的内容显示出来，然后将光标移至a[i]=a[i-1]的语句行上，不断按F4，观察a数组值的变化情况，以此分析并找出错误原因。

④改正错误后重新运行程序，直到结果正确为此。

⑤如果要用三次循环移位来实现将最后三个数移到前面，其余数依次往后移三个位置，则程序应该如何修改？

1. 用移位法将数组a中的最后一个数移到最前面，其余数依次往后移动一个位置。

### 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | #include<stdio.h>  int main**()**  **{**  int i**,**t**,**a**[**10**]={**0**,**1**,**2**,**3**,**4**,**5**,**6**,**7**,**8**,**9**};**  t**=**a**[**9**];**  **for(**i**=**1**;**i**<**10**;**i**++)**  a**[**i**]=**a**[**i**-**1**];**  a**[**0**]=**t**;**  printf**(**"\n"**);**  **for(**i**=**0**;**i**<**10**;**i**++)**  printf**(**"%d" **,**a**[**i**]);**  **}** |

## 成绩排序

输入n个学生的单科成绩，然后从高到低的顺序排序后输出。

①分析程序及其特性。

②上机编译程序，程序是否有语法错误？应如何修改？（数组a的长度可比n大些），改正错误后重新编译和运行程序，直到结果正确为此。

③你对选择排序算法的实现过程是否清楚了？若不清楚，请用动态跟踪的方法观察其实现过程，操作如下：

首先将光标移至 if语句行上，按F4，接着输入数据，当绿条第停留在if语句行时，用Ctrl-F7操作将a数组的内容显示出来，不断按F4，观察a数组值的变化情况，以此分析和领会算法的实现过程。

④输入冒泡排序程序，用动态跟踪观察其实现过程。

⑤如果要用三次循环移位来实现将最后三个数移到前面，其余数依次往后移三个位置，则程序应该如何修改？

以下是用选择法实现的排序。

### 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | #include<stdio.h>  int main**(** **)**  **{**  int i**,**j**,**t**,**n**,**a**[**n**];**  printf**(**"\n n=?"**);**  scanf**(**"%d"**,&**n**);**  printf**(**"input n numbers :\n"**);**  **for** **(**i**=**0**;**i**<**n**;**i**++)**  scanf**(**"%d"**,&**a**[**i**]);**  **for(**i**=**0**;**i**<**n**-**1**;**i**++)**  **for(**j**=**i**+**1**;**j**<**n**;**j**++)**  **if(**a**[**i**]<**a**[**j**])**  **{**t**=**a**[**i**];**a**[**i**]=**a**[**j**];**a**[**j**]=**t**;}**  printf**(**"the sorted numbers:\n"**);**  **for(**i**=**0**;**i**<**n**;**i**++)**  printf**(**"%4d"**,**a**[**i**]);**  **return** 0**;**  **}** |

## 矩阵操作

1．将矩阵Ａ[4][5]中值（行中所有数的和）为最大的那一行元素与首行元素对换。

具体要求如下：

①矩阵Ａ的数值从键盘输入。

②以矩阵的形式输出对换后的矩阵。

2．P141 7.6。

17个人围坐一圈，顺序编号为1, 2，3，…,17。现在从第一个人开始数起，每数到7时，这个人就从圈里出来，再从下一个数重新开始数1, 2，…，7，数到第7的这个人也从圈里出来，直到全部17个人从圈里出来为此。例如，前面站出来的4个人是7, 14, 4和12。编程输出从圈里出来的人的顺序。

## 程序排错

1．下列有关数组的说明或语句是否存在错误？

int a**[**3**,**4**];** a**[**3**]=**a**[**1**\***2**];**

int a**[**3**]={**10**,**8**,**3**,**4**}**

float a**[**4**];**a**[**0**]=**15**;**a**[**4**]=**100**;**

int I**,**j**,**k**[**3**]={**3**};**

char ch**[]={};**

## 排序

从键盘输入10整数，分别用选择法，交换法，插入法对它们进行由小到大（或由大到小）排序，且输出排序后的结果。

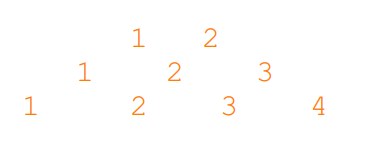
利用上题程序的排序结果，再从键盘输入一个整数，然后从排序的数组查找是否有与刚输入的数组相等的元素，若有，是第几个？若找不到，则显示相应的信息。

下面是一个输出10阶杨辉三角形的程序：

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 | #include<stdio.h>  #define N 10  int main**()**  **{**  int a**[**N**][**N**],**i**,**j**;**  **for(**i**=**1**;**i**<**N**;**i**++)** /\*形成主对角线和第1列元素\*/  **{**  a**[**i**][**1**]=**1**;**  a**[**i**][**i**]=**1**;**  **}**  **for(**i**=**3**;**i**<**N**;**i**++)** /\* 形成其余元素\*/  **for(**j**=**2**;**j**<=**i**-**1**;**j**++)**  a**[**i**][**j**]=**a**[**i**-**1**][**j**-**1**]+**a**[**i**-**1**][**j**];**  **for(**i**=**1**;**i**<**N**;**i**++)** /\* 输出\*/  **{**  **for(**j**=**1**;**j**<**30**-**2**\***i**;**j**++)**  printf**(**" "**);**  **for(**j**=**1**;**j**<=**i**;**j**++)**  printf**(**"%-4d"**,**a**[**i**][**j**]);**  printf**(**"\n"**);**  **}**  **return** 0**;**  **}** |

修改上述程序，使输出形式为



（一直到第10行）

## 矩阵转置

从键盘输入的矩阵（可以定义为5），输出此矩阵和转置后的矩阵。

## 矩阵鞍点

从键盘输入行列式，并输出此行列式；然后求所有的鞍点（某元素若是本行元素中的最大者，同时又是本列元素中最小者，则此元素称为鞍点）。最后输出这些鞍点及其对应坐标值。（若无鞍点，则显示无鞍点信息）。

## 回文字符串判断

从键盘上输入一字符串，并判断是否形成回文（即正序和逆序一样，如“abcd dcba”）。

## 字符串修改

5．按字符数组输入字符，将其中的英文字母都改成‘\*’，然后按字符串形式输出。

# 实验过程

1题：

这个题目看重两个方面，一是设计数组，二是以美观的形式输出数组。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | #include<stdio.h>  #define N 10  int main**()**  **{**  int a**[**N**][**N**],**i**,**j**;**  **for(**i**=**1**;**i**<**N**;**i**++)** /\*形成第1与第2列元素\*/  **{**  a**[**i**][**1**]=**1**;**  a**[**i**][**2**]=**2**;**  **}**  **for(**i**=**2**;**i**<**N**;**i**++)** /\* 形成其余元素\*/  **for(**j**=**3**;**j**<=**i**;**j**++)**  a**[**i**][**j**]=**a**[**i**-**1**][**j**-**1**]+**1**;**  **for(**i**=**1**;**i**<**N**;**i**++)** /\* 输出\*/  **{**  **for(**j**=**1**;**j**<**30**-**2**\***i**;**j**++)**  printf**(**" "**);**  **for(**j**=**1**;**j**<=**i**;**j**++)**  printf**(**"%-4d"**,**a**[**i**][**j**]);**  printf**(**"\n"**);**  **}**  **return** 0**;**  **}** |

2题：

互换单个元素。

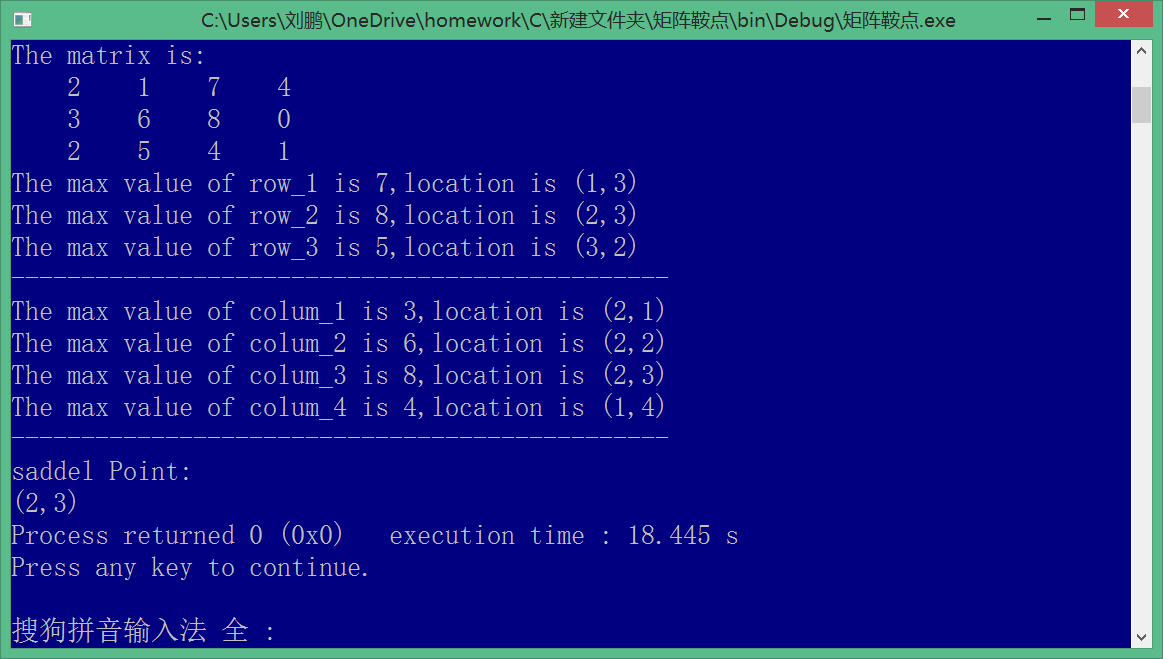
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | #include <stdio.h>  #define N 3  int main**()**  **{**  int a**[**N**][**N**],**b**[**N**][**N**],**i**,**j**;**  **for(**i**=**0**;**i**<=**N**-**1**;**i**++)**  **{**  **for(**j**=**0**;**j**<=**N**-**1**;**j**++)**  scanf**(**"%d"**,&**a**[**i**][**j**]);**  **}**  printf**(**"Matrix A is:\n"**);**  **for(**i**=**0**;**i**<=**N**-**1**;**i**++)**  **{**  **for(**j**=**0**;**j**<=**N**-**1**;**j**++)**  **{**  printf**(**"%5d"**,**a**[**i**][**j**]);**  b**[**j**][**i**]=**a**[**i**][**j**];**  **}**  printf**(**"\n"**);**  **}**  printf**(**"Matrix B is:\n"**);**  **for(**j**=**0**;**j**<=**N**-**1**;**j**++)**  **{**  **for(**i**=**0**;**i**<=**N**-**1**;**i**++)**  printf**(**"%5d"**,**b**[**j**][**i**]);**  printf**(**"\n"**);**  **}**  **return** 0**;**  **}** |

3题:



矩阵鞍点

|  |  |
| --- | --- |
| 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  70  71  72  73  74  75  76  77 | #include <stdio.h>  #define M 3 /\* 行\*/  #define N 4 /\* 列\*/  int main**()**  **{**  int compare**(**int a**,**int b**,**int c**,**int d**);**  printf**(**"-----------------------------------------------\n"**);**  printf**(**"Please input the elements of your matrix with ENTER to apart:\n"**);**  int a**[**M**][**N**],**i**,**j**;**  int max\_row**[**M**],**max\_colum**[**N**];**  int row**,**colum**;**  int row\_1**[**M**],**colum\_1**[**M**];** /\* 行的最大值有M个 \*/  int row\_2**[**N**],**colum\_2**[**N**];** /\* 列的最大值有N个 \*/  **for(**i**=**0**;**i**<=**M**-**1**;**i**++)** /\* 输入矩阵 \*/  **{**  **for(**j**=**0**;**j**<=**N**-**1**;**j**++)**  scanf**(**"%d"**,&**a**[**i**][**j**]);**  **}**  printf**(**"The matrix is:\n"**);**  **for(**i**=**0**;**i**<=**M**-**1**;**i**++)**  **{**  **for(**j**=**0**;**j**<=**N**-**1**;**j**++)**  printf**(**"%5d"**,**a**[**i**][**j**]);**  printf**(**"\n"**);**  **}**  **for(**i**=**0**;**i**<=**M**-**1**;**i**++)** /\*求每行的最大值及其坐标\*/  **{**  max\_row**[**i**]=**a**[**i**][**0**];**  **for(**j**=**1**;**j**<=**N**-**1**;**j**++)**  **{**  **if(**a**[**i**][**j**]>=**max\_row**[**i**])**  **{**  max\_row**[**i**]=**a**[**i**][**j**];**  row**=**i**;**  colum**=**j**;**  **}**  **}**  row\_1**[**i**]=**row**+**1**;** /\*得到了行最大的行坐标\*/  colum\_1**[**i**]=**colum**+**1**;** /\*得到了行最大的列坐标\*/  printf**(**"The max value of row\_%d is %d,location is (%d,%d)\n"**,**row**+**1**,**max\_row**[**i**],**row**+**1**,**colum**+**1**);**  **}**  printf**(**"-----------------------------------------------\n"**);**  **for(**j**=**0**;**j**<=**N**-**1**;**j**++)** /\*求每列的最大值及其坐标\*/  **{**  max\_colum**[**j**]=**a**[**0**][**j**];**  **for(**i**=**0**;**i**<=**M**-**1**;**i**++)**  **{**  **if(**a**[**i**][**j**]>=**max\_colum**[**j**])**  **{**  max\_colum**[**j**]=**a**[**i**][**j**];**  row**=**i**;**  colum**=**j**;**  **}**  **}**  row\_2**[**j**]=**row**+**1**;**  colum\_2**[**j**]=**colum**+**1**;**  printf**(**"The max value of colum\_%d is %d,location is (%d,%d)\n"**,**colum**+**1**,**max\_colum**[**j**],**row**+**1**,**colum**+**1**);**  **}**  printf**(**"-----------------------------------------------\n"**);**  printf**(**"saddel Point:\n"**);**  /\*得到了M+N个数字，然后核对是否有相同位置的，是的话，就是鞍点\*/  **for(**i**=**0**;**i**<=**M**-**1**;**i**++)**  **{**  **for(**j**=**0**;**j**<=**N**-**1**;**j**++)**  **{**  **if(**compare**(**row\_1**[**i**],**colum\_1**[**i**],**row\_2**[**j**],**colum\_2**[**j**])==**1**)**  printf**(**"(%d,%d)"**,**row\_1**[**i**],**colum\_1**[**i**]);**  **}**  **}**  **return** 0**;**  **}**  int compare**(**int a**,**int b**,**int c**,**int d**)**  **{**  int z**;**  z**=((**a**==**c**)&&(**b**==**d**))?** 1**:**0**;**  **return(**z**);**  **}** |



很明显，鞍点就是（2,3），8.

# 实验总结

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