**云南大学数学与统计学实验教学中心**

**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**：结构体与共用体程序设计 |
| **实验编号**：NO.10 | **实验日期**：2017年1月1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．掌握结构体类型变量与结构体数组的定义和使用。

2．学会使用结构体指针变量和结构体指针数组。

3．掌握链表的概念，初步学会对链表进行操作。

4．学会使用结构体指针变量。

5．巩固冒泡排序的方法。

6．掌握链表的基本概念及插入输出等常用操作方法。

7．熟悉共同体的概念与使用。

# 二、实验内容

1．输入出10个学生的学号、姓名和成绩，求出其中的高分者和低分者。

|  |  |
| --- | --- |
| 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 | /\* filename:10.1 \*/  #include<stdio.h>  struct student  **{**  int num**;**  char name**[**20**];**  int score**;**  **};**  int main**(** **)**  **{**  int i**;**  struct student st**,**stmax**,**stmin**;**  stmax**.**score**=**0**;** stmin**.**score**=**100**;**  printf**(**"\n input data"**);**  **for(**i**=**0**;**i**<**10**;**i**++)**  **{**  scanf**(**"%d%s%d"**,&**st**.**num**,**st**.**name**,&**st**.**score**);**  **if(**st**.**score**>**stmax**.**score**)**stmax**=**st**;**  **if(**st**.**score**<**stmin**.**score**)**stmin**=**st**;**  **}**  printf**(**"\n hight:%5d%15s%5d"**,**stmax**.**num**,**stmax**.**name**,**stmax**.**score**);**  printf**(**"\n low:%5d%15s%5d"**,**stmin**.**num**,**stmin**.**name**,**stmin**.**score**);**  **return** 0**;**  **}** |

①分析程序，上机运行程序。

②程序中，哪些是对结构体变量的成员引用，哪些是整体引用？

③对于此例来说，用结构体变量作为数据结构有何优越性？

1．有一学生情况如下表所示。编制一个C程序，用冒泡法对该学生情况表按成绩（grade）从低到高进行排序。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学号(num) | 姓名(name) | 性别(sex) | 年龄(age) | 成绩(grade) |
| 101 | Zhang | M | 19 | 95.6 |
| 102 | Wang | F | 18 | 92.4 |
| 103 | Zhao | M | 19 | 85.7 |
| 104 | Li | M | 20 | 96.3 |
| 105 | Gao | M | 19 | 90.2 |
| 106 | Lin | M | 18 | 91.5 |
| 107 | Ma | F | 17 | 98.7 |
| 108 | Zhen | M | 21 | 90.1 |
| 109 | Xu | M | 19 | 89.8 |
| 110 | Mao | F | 18 | 94.9 |

具体要求如下:

①结构体类型为

struct 结构体名

**{**

int num**;**

char name**[**8**];**

char sex**;**

int age**;**

double grade**;**

**};**

②在程序中用一个结构体指针数组，其中每一个指针元素指向结构体类型的各元素。

③在程序中，首先输出排序前的学生情况，然后输出排序后的结果，其格式如上表所示(表的框线不要)。

链表基本操作具体要求如下:

①初始时链表为空，即链表的头指针为空。

②对于上表所示的学生情况表，依次将每个学生的情况作为一个结点插入到单链表的链头(即当前插入的结点将成为第一个结点)。

③所有学生情况都插入到链表中后，从链头开始，依次输出链表中的各结点值(即每个学生的情况)。输出格式如同上表。

2.下表为学生学习情况，编写一个C程序，用冒泡法对此学生学习情况表按成绩（grade）从高到低进行排序。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学号(num) | 姓名(name) | 性别(sex) | 年龄(age) | 成绩(grade) |
| 101 | Wangyi | M | 16 | 82.7 |
| 102 | Zhaoyi | M | 17 | 99.0 |
| 103 | Liming | M | 15 | 85.6 |
| 104 | Gaoben | F | 16 | 77.8 |
| 105 | Chenping | F | 17 | 67.4 |
| 106 | Zhangjing | F | 16 | 99.5 |
| 107 | Handong | M | 15 | 82.7 |
| 108 | Mengguang | M | 16 | 60.5 |
| 109 | Xucong | F | 17 | 94.5 |
| 110 | Chengcheng | F | 16 | 96.7 |

对程序要求如下：

定义结构体类型为：

struct 结构体名

{

int num;

char name[15];

char sex;

int age;

double grade;

}

(2) 在程序中用一个结构体指针数组，其中每一个指针元素指向结构体类型数组的各元素。

(3）要求先输出排序前的情况，再输出排序后的结果（表的框线可以不要）。

方法提示：

冒泡排序过程为：从前到后扫描结构体数组中各元素，比较相邻两元素中成绩（grade）的大小，若发现逆序就交换，最后使最小者换到最后。第二轮同样可以把剩余元素中成绩最小者换到后面，剩余元素排序原理相同。

实际排序过程中，并不交换结构体数组的各元素，由于用指针指向各元素，所以只是交换指针数组中的各指针。因此，排序的最后结果，其结构体数组中的各元素的存贮顺序关系并没有改变，而只是按结构体指针数组中各指针元素顺序指向的结构体数组元素关于成绩（grade）是有序的。

\*3．链表基本操作

对程序要求如下：

初始化链表的头指针为空。

对上面的学生学习情况表，依次将每个学生的情况作为一个结点插入到单链表的表头（即当前插入的结点成为第一个结点）。每个学生的结点结构为：

Struct stu

**{**

int num**;**

char name**[**15**];**

char sex**;**

int age**;**

double grade**;**

struct stu **\***next**;**

**}**

建完链表后，从头开始，依次输出链表中各结点值（既每个学生学习情况）。

提示：为了给每个学生学习情况的结点P分配动态存储区，可用如下语句：P=(struct stu \*)malloc(sizeof(struct stu)); 其中P为结构体类型 struct stu的指针。为了使用库函数malloc()，要包含头文件“stdlib.h”.

**分析讨论：**

(1) 如用选择法排序，如何改动你的程序？简单讨论改进的选择法与普通的选择法的差别。

(2) 写出交换两个结构体类型数组元素的语句（不用指针），与你程序中用指针进行交换的方法进行比较，看看哪个更简单方便。

(3) 在建好的链表中若要删除num=103的结点，如何进行？写出相应的函数。

# 习题

1. 结构共用体的枚举型下述定义有哪些是不对的?

struct a

**{**

int a**;**

float x**;**

**}**

**答：没有分号。**

/\*----------------\*/

struct s

**{**

int x**,**y**;**

char c

**};**

struct s1**-**s2**;**

**答：自定义结构可以赋值，但是不能加减运算。**

/\*----------------\*/

strcut

**{**

char m**[**10**];**

int I**,**j**;**

**};** x**,**y**;**

**答：定义结构的时候定义变量，不需要再花括号后加分号。**

/\*----------------\*/

strcut str

**{**

string s**[**10**];**

char a**[**10**]**

**};**

**答：基本类型没有**string**。**

/\*----------------\*/

union data

**{**

int I**;**

char ch**;**

long int l**;**

**};**

union da**,**da2**;**

**答：定义的共用体类型变量是**data**，不是**union。

/\*----------------\*/

union s

**{**

int I**;**

char ch**;**

float x**;**

**};**

/\*----------------\*/

enum e1

**{**

a1**,**

z**,**

3**,**

b**,**

**};**

**答：枚举类型名不能用常量。**

/\*----------------\*/

enum e2

**{**

red**,**

yellow**=**8**,**

blue**=**7

**};**

**答：没错。**

/\*----------------\*/

enum e3

**{**

int a1**;**

float a2**;**

**};**

**答：枚举类型中不能有变量。**

/\*----------------\*/

union un

**{**

int I**,**j**,**k**;**

20**,**50**;**

float x**,**y**;**

**};**

**答：共用体成员必须是变量。**

阅读下面的程序,熟悉结构的访问方法

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | #include<stdio.h>  struct student  **{**  char name**[**10**]**  char sex**;**  float score**;**  **}**stu**={**"li fang "**,**'f'**,**98**};**  int main**()**  **{**  stuct student **\***p**;**  p**=&**stu**;**  printf**(**"\n &stu=%x,or%u"**,**p**,**p**);**  printf**(**"\n%s,%c”,%f"**,**sut**.**name**,** stu**.**sex**,** stu**.** Score**);**  printf**(**"\n %s, %c, %f"**,(\***p**).**name**,** **(\***p**).**sex**,(\***p**)** **.**score**);**  printf**(**"\n %s, %c, %f"**,**p**-**name**,**p**-**sex**,** p**-**score**);**  **return** 0**;**  **}** |

4．根据上题的提示,设计如下程序：某班30人,每人一个记录，包括姓名，学号和某门功课的成绩。从键盘输入（用记录数组形式访问）这些数据，然后对成绩进行自高而低排序，最后输出排序后的每条记录（用指针访问）

下面是一个将指针用于结构数据动态链表操作的实例。请初步熟悉它们的使用方法。

|  |  |
| --- | --- |
| 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 | /\* filename: 10.2 \*/  #include<stdio.h>  #include<stdlib.h>  #define LEN sizeof(struct Student)  struct Student  **{**  float score**;**  struct Student **\***next**;**  **};**  int n**;**  float x**;**  struct Student **\***Creat**()**  **{**  struct Student **\***head**;**  struct Student **\***r**,\***p**;**  n**=**0**;**  r**=(**struct Student **\*)**malloc**(**LEN**);**  p**=**r**;**  scanf**(**"%f"**,&**x**);**  **(**r**->**next**)=NULL;**  **return** head**;**  **}**  void Print**(**struct Student **\***head**)**  **{**  struct Student **\***p**;**  p**=**head**;**  **while** **(**p**!=NULL)**  **{**  printf**(**"%8.2f"**,**p**->**score**);**  p**=**p**->**next**;**  **}**  **}**  int main**()**  **{**  struct Student **\***head**;**  printf**(**"input score:\n"**);**  head**=**Creat**();**  Print**(**head**);**  **return** 0**;**  **}** |

/\* 这个程序有警告 \*/

参照上题设计程序,建立一个堆栈结构的单链表，便历这个链表，在某一转点后插入新结点，删除指定结点。

# 实验环境

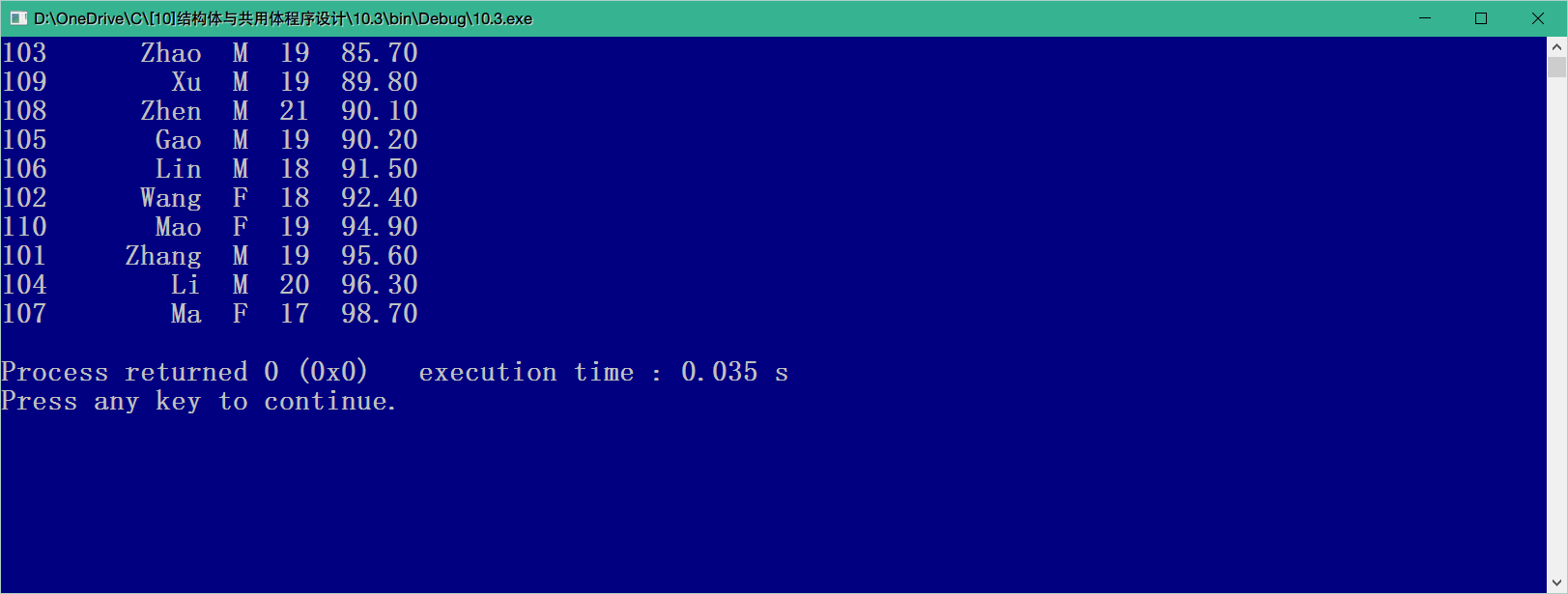
Windows10 Enterprise 1703中文版操作系统；

Turbo C 2.0与Code::Blocks 16.01集成开发环境。

# 三、实验过程

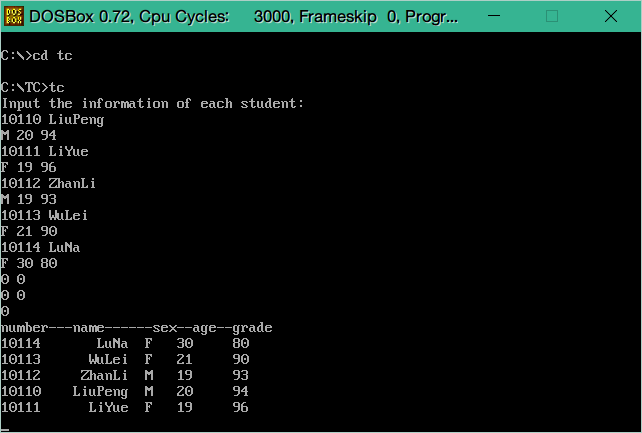
1题:

|  |  |
| --- | --- |
| 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  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 | /\* filename: 10.3 \*/  #include<stdio.h>  #include<stdlib.h>  struct Student  **{**  int num**;**  char name**[**8**];**  char sex**;**  int age**;**  double grade**;**  **};**  struct Student stu**[**10**]=**  **{**  **{**101**,**"Zhang"**,**'M'**,**19**,**95.6**},**  **{**102**,**"Wang"**,**'F'**,**18**,**92.4**},**  **{**103**,**"Zhao"**,**'M'**,**19**,**85.7**},**  **{**104**,**"Li"**,**'M'**,**20**,**96.3**},**  **{**105**,**"Gao"**,**'M'**,**19**,**90.2**},**  **{**106**,**"Lin"**,**'M'**,**18**,**91.5**},**  **{**107**,**"Ma"**,**'F'**,**17**,**98.7**},**  **{**108**,**"Zhen"**,**'M'**,**21**,**90.1**},**  **{**109**,**"Xu"**,**'M'**,**19**,**89.8**},**  **{**110**,**"Mao"**,**'F'**,**19**,**94.9**},**  **};**  void sort**(**struct Student **\***p**)**  **{**  int i**,**j**,**k**;**  struct Student temp**;**  **for(**i**=**0**;**i**<**10**-**1**;**i**++)**  **{**  k**=**i**;**  **for(**j**=**i**+**1**;**j**<**10**;**j**++)**  **{**  **if((**p**+**k**)->**grade **>** **(**p**+**j**)->**grade**)**  **{**  k**=**j**;**  **}**  **}**  **if(**i**!=**k**)**  **{**  temp**=\*(**p**+**k**);**  **\*(**p**+**k**)=\*(**p**+**i**);**  **\*(**p**+**i**)=**temp**;**  **}**  **}**  **}**  void print**(**struct Student **\***p**)**  **{**  int i**;**  **for(**i**=**0**;**i**<**10**;**i**++)**  **{**  printf**(**"%d %8s %c %d %3.2f\n"**,(**p**+**i**)->**num**,(**p**+**i**)->**name**,(**p**+**i**)->**sex**,(**p**+**i**)->**age**,(**p**+**i**)->**grade**);**  **}**  **}**  int main**()**  **{**  struct Student **\***p**;**  p**=**stu**;**  sort**(**p**);**  print**(**p**);**  **return** 0**;**  **}** |



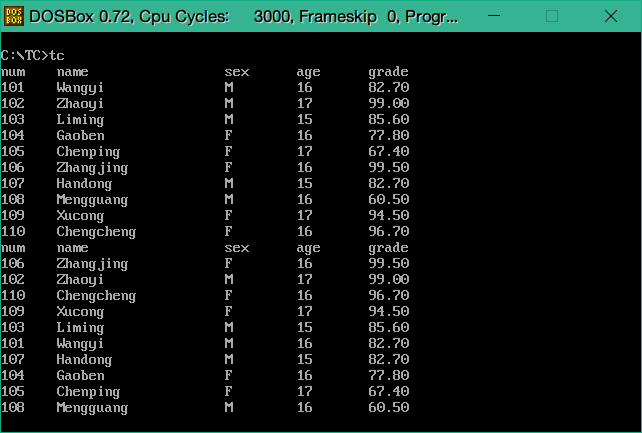
1题用动态链表实现：

|  |  |
| --- | --- |
| 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  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127 | /\* filename: 10.4 \*/  #include<stdio.h>  #include<stdlib.h>  #define LEN sizeof(struct Student)  struct Student  **{**  int num**;**  char name**[**8**];**  char sex**;**  int age**;**  int grade**;**  struct Student **\***next**;**  **}\*** head**=NULL;**  /\*--------\*/  struct Student **\***create**()** /\* run this block first \*/  **{**  int judgement**=**0**;**  struct Student **\***p1**,\***p2**=NULL;**  p1**=(**struct Student **\*)**malloc**(**LEN**);**  printf**(**"Input the information of each student:\n"**);**  scanf**(**"%d%s"**,&**p1**->**num**,**p1**->**name**);**  scanf**(**"%c"**,&**p1**->**sex**);**  scanf**(**"%c%d%d"**,&**p1**->**sex**,&**p1**->**age**,&**p1**->**grade**);**  **while(**p1**->**num**!=**0**)**  **{**  **if(**judgement**++==**0**)**  **{**  head**=**p1**;**  **}**  **else**  **{**  p2**->**next**=**p1**;**  **}**  p2**=**p1**;**  p1**=(**struct Student **\*)**malloc**(**LEN**);**  scanf**(**"%d%s"**,&**p1**->**num**,**p1**->**name**);**  scanf**(**"%c"**,&**p1**->**sex**);**  scanf**(**"%c%d%d"**,&**p1**->**sex**,&**p1**->**age**,&**p1**->**grade**);**  **}**  p2**->**next**=NULL;**  **return** head**;**  **}**  /\*--------\*/  void swap**(**struct Student **\***p**)**  **{**  int i**;**  struct Student temp**;**  **{**  temp**.**num **=** p**->**num**;**  p**->**num **=** p**->**next**->**num**;**  p**->**next**->**num **=** temp**.**num**;**  **}**  **{**  **for(**i**=**0**;**i**<**8**;**i**++)**  **{**  temp**.**name**[**i**]** **=** p**->**name**[**i**];**  p**->**name**[**i**]** **=** p**->**next**->**name**[**i**];**  p**->**next**->**name**[**i**]** **=** temp**.**name**[**i**];**  **}**  **}**  **{**  temp**.**sex **=** p**->**sex**;**  p**->**sex **=** p**->**next**->**sex**;**  p**->**next**->**sex **=** temp**.**sex**;**  **}**  **{**  temp**.**age **=** p**->**age**;**  p**->**age **=** p**->**next**->**age**;**  p**->**next**->**age **=** temp**.**age**;**  **}**  **{**  temp**.**grade **=** p**->**grade**;**  p**->**grade **=** p**->**next**->**grade**;**  p**->**next**->**grade **=** temp**.**grade**;**  **}**  **}**  /\*--------\*/  int count**()**  **{**  struct Student **\***pt **=** **NULL;**  int n**=**0**;**  **for(**pt**=**head**;**pt**!=NULL;**pt**=**pt**->**next**)**  **{**  n**+=**1**;**  **}**  **return** n**;**  **}**  /\*--------\*/  void sort**(**struct Student **\***p**)** /\* bubble sort \*/  **{** /\* from small to big \*/  int i**,**j**;**  int guard**=**count**();**  p**=**head**;**  **for(**i**=**0**;**i**<**guard**-**1**;**i**++)**  **{**  **for(**j**=**0**;**j**<**guard**-**i**;**j**++)**  **{**  **if(**p**->**grade **>** p**->**next**->**grade**)**  **{**  swap**(**p**);**  **}**  p**=**p**->**next**;**  **}**  p **=** head**;** /\* let this pointer come back \*/  **}**  p **=** head**;** /\* let this pointer come back \*/  **}**  /\*--------\*/  void output**(**struct Student **\***p**)**  **{**  printf**(**"number---name------sex--age--grade\n"**);**  **while(**p**->**num **!=** 0**)**  **{**  printf**(**"%d %8s %c %d %4d"**,**p**->**num**,**p**->**name**,**p**->**sex**,**p**->**age**,**p**->**grade**);**  p **=** p**->**next**;**  printf**(**"\n"**);**  **}**  **}**  /\*--------\*/  int main**()**  **{**  struct Student **\***p**;**  p**=**create**();** /\* p is the head of the linked array \*/  sort**(**p**);** /\* let p be the x of function sort \*/  output**(**p**);** /\* print every member of the array \*/  **return** 0**;**  **}** |



2.

|  |  |
| --- | --- |
| 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  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100 | /\* filename: 10.5 \*/  #include <stdio.h>  #include <stdlib.h>  /\* space \*/  **typedef** struct  **{**  int num**;**  char name**[**15**];**  char sex**;**  int age**;**  double grade**;**  **}**Info**;**  /\* space \*/  Info stu**[**10**]=**  **{**  **{**101**,**"Wangyi"**,**'M'**,**16**,**82.7**},**  **{**102**,**"Zhaoyi"**,**'M'**,**17**,**99.0**},**  **{**103**,**"Liming"**,**'M'**,**15**,**85.6**},**  **{**104**,**"Gaoben"**,**'F'**,**16**,**77.8**},**  **{**105**,**"Chenping"**,**'F'**,**17**,**67.4**},**  **{**106**,**"Zhangjing"**,**'F'**,**16**,**99.5**},**  **{**107**,**"Handong"**,**'M'**,**15**,**82.7**},**  **{**108**,**"Mengguang"**,**'M'**,**16**,**60.5**},**  **{**109**,**"Xucong"**,**'F'**,**17**,**94.5**},**  **{**110**,**"Chengcheng"**,**'F'**,**16**,**96.7**},**  **};**  /\* space \*/  void Swap**(**Info **\***p**)**  **{**  int i**;**  Info temp**;**  **{**  temp**.**num**=**p**->**num**;**  p**->**num**=(**p**+**1**)->**num**;**  **(**p**+**1**)->**num**=**temp**.**num**;**  **}**  **{**  **for(**i**=**0**;**i**<**15**;**i**++)**  **{**  temp**.**name**[**i**]=**p**->**name**[**i**];**  p**->**name**[**i**]=(**p**+**1**)->**name**[**i**];**  **(**p**+**1**)->**name**[**i**]=**temp**.**name**[**i**];**  **}**  **}**  **{**  temp**.**sex**=**p**->**sex**;**  p**->**sex**=(**p**+**1**)->**sex**;**  **(**p**+**1**)->**sex**=**temp**.**sex**;**  **}**  **{**  temp**.**age**=**p**->**age**;**  p**->**age**=(**p**+**1**)->**age**;**  **(**p**+**1**)->**age**=**temp**.**age**;**  **}**  **{**  temp**.**grade**=**p**->**grade**;**  p**->**grade**=(**p**+**1**)->**grade**;**  **(**p**+**1**)->**grade**=**temp**.**grade**;**  **}**  **}**  /\* space \*/  void BubbleSort**(**Info **\***p**[])**/\* from big to small \*/  **{**  int i**,**j**;**  **for(**i**=**0**;**i**<**9**;**i**++)**  **{**  **for(**j**=**0**;**j**<**10**-**i**;**j**++)**  **{**  **if(**p**[**j**]->**grade **<** p**[**j**+**1**]->**grade**)**  **{**  Swap**(**p**[**j**]);**  **}**  **}**  **}**  **}**  /\* space \*/  void Print**(**Info **\***p**[])**  **{**  int i**;**  printf**(**"num name sex age grade\n"**);**  **for(**i**=**0**;**i**<**10**;**i**++)**  **{**  printf**(**"%d %-15s %c %d %4.2f"**,**  p**[**i**]->**num**,**p**[**i**]->**name**,**p**[**i**]->**sex**,**p**[**i**]->**age**,**p**[**i**]->**grade**);**  printf**(**"\n"**);**  **}**  **}**  /\* space \*/  int main**()**  **{**  int i**;**  Info **\***p**[**10**];**  **for(**i**=**0**;**i**<**10**;**i**++)**  **{**  p**[**i**]=&**stu**[**i**];**  **}**  Print**(**p**);**  BubbleSort**(**p**);**  Print**(**p**);**  **return** 0**;**  **}** |



/\* 此程序在Code::Blocks 16.01下无法运行，但是在Turbo C 2.0下可以无警告运行 \*/

# 五、实验总结

有一个问题，就是编译器的改变，会导致程序不能编译。在TC2.0下，程序可以no warning, no error 编译链接运行，但是换到Code::Blocks上，也是无警告无错误，但是.exe文件就是不能运行。至今不知道为什么。

链表博大精深，应用广泛，能节约内存。很值得学习。

# 六、参考文献

[1]谭浩强，C 程序设计[M] (第四版)．北京：清华大学出版社，2010年6月（中国高等院校计算机基础教育课程体系规划教材）

[2]谭浩强，C 程序设计( 第四版 )学习辅导 ，北京：清华大学出版社，2010年7月（中国高等院校计算机基础教育课程体系规划教材）

# 七、教师评语