

面向 21 世纪高等院校教材

# Turbo C 程序设计上机指导

主	编	蒿社平	焦	锋	
参	编	胡承军	邢文建	王丽萍	

版权专有 侵权必究

---

图书在版编目(CIP)数据

Turbo C 程序设计上机指导 :面向 21 世纪高等院校教材 / 蒿社平, 焦锋  
主编. —北京: 北京理工大学出版社, 2006.2

ISBN 7-5640-0720-6

I. T... II. ①蒿...②焦... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 004667 号

---

---

出版发行 / 北京理工大学出版社

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010)68914775(办公室) 68944990(发行部) 68911084(读者服务部)

网 址 / <http://www.bitpress.com.cn>

电子邮箱 / [chiefeditor@bitpress.com.cn](mailto:chiefeditor@bitpress.com.cn)

经 销 / 全国各地新华书店

印 刷 / 北京圣瑞伦印刷厂

开 本 / 787 毫米×1092 毫米 1/16

印 张 / 12.5

字 数 / 288 千字

版 次 / 2006 年 2 月第 1 版 2006 年 2 月第 1 次印刷

印 数 / 1-6000 册

定 价 / 20.00 元

责任校对 / 郑兴玉

责任印制 / 李绍英

---

图书出现印装质量问题, 本社负责调换

# 前 言

信息技术已经成为推动一个国家经济发展的重要因素,信息技术的应用已经渗透到我们日常生活的各个领域。计算机技术作为信息技术的重要基础和组成部分,其作用显而易见。随着我国计算机应用的进一步普及和深入,具备计算机程序设计能力的高级技术人员已越来越受欢迎,计算机程序设计能力也成为衡量高级人才的重要指标之一。目前 C 语言程序设计课程是大学一年级必修的一门计算机课程。它通俗易学,容易上手,学生经过努力,在短期内就可以基本掌握计算机程序设计的基本知识,拥有计算机程序设计的操作技能和程序设计技能。同时,C 语言作为全国计算机等级考试二级考试中的一门可选语言,是用来衡量学生计算机程序设计水平的重要标准。

本书是专门为 C 语言课程上机实践提供指导的。由于计算机程序设计是一门以实践为主的课程,所以上机实践是学生掌握和提高计算机程序设计能力的重要环节,上机指导用书尤为重要。本书有以下主要特点:

第一,目标明确,重点突出。有关 C 语言的相关资料较多,侧重面也不尽相同。本书的内容紧扣大纲,同时也兼顾了全国计算机等级考试二级考试中关于 C 语言考试的范围和要求。另外,书中还考虑到学生将来的实际需求,在最后添加了综合程序设计编程的部分,以期使学生获得较大收益。

第二,在内容选取上力求简练。在本书的编写中,编者在追求知识完整性的同时,剔除了一些不常用的部分,选取和编写了一些具有代表性、普遍性的实验提供给学生,各个实验由浅入深,循序渐进。

本书共分为 11 个实验,各教学单位可根据自身情况进行选择。本书由蒿社平、焦锋、胡承军、邢文建、王丽萍编写。全书由蒿社平统稿、整理,张永强教授对全部书稿进行了审阅,并提出了许多宝贵的意见,在编写、出版过程中得到了河北工程大学有关领导的关心和支持,在此表示衷心感谢。

尽管本书在探索如何提高学生计算机程序设计能力方面做了不少努力,但由于编者水平有限,时间仓促,难免会有欠妥之处,恳请读者批评指正,我们将及时修订。

编 者

2005 年 12 月

# 目 录

实验一	TC 2.0 集成开发环境 .....	1
实验二	数据类型及顺序结构.....	6
实验三	选择结构程序设计 .....	12
实验四	循环结构程序设计 .....	19
实验五	函数 .....	27
实验六	数组 .....	39
实验七	指针(一) .....	46
实验八	指针(二) .....	57
实验九	结构体和共用体 .....	65
实验十	文件 .....	75
实验十一	综合程序设计 .....	87
附录一	全国计算机等级考试二级 C 语言考试大纲 .....	99
附录二	C 语言程序设计笔试测试题 .....	102
附录三	C 语言程序设计上机测试题 .....	164
附录四	ASCII 码表 .....	191

# 实验一 TC2.0 集成开发环境

## 一、知识要点

### 1. TC 简介

Turbo C是一个常用的、最基本的 C 语言工具，一般简称为 TC。它为 C 语言开发提供了操作便利的集成环境。源程序的输入、修改、调试及运行都可以在 TC 集成环境下完成，非常方便。TC 2.0系统非常小巧，但功能齐全。它主要支持 DOS 环境，因此在操作中无法使用鼠标，需要通过键盘操纵菜单或快捷键完成。其界面如图 1-1 所示。



图 1-1 TC 集成环境

菜单区：位于屏幕中的最上一行。包含许多不同功能的命令。通过按“↑”、“↓”、“←”、“→”4 个方向键可选择菜单命令。

编辑区：在此处输入和修改 C 语言源程序。

信息窗口：用于显示一些错误等信息。

热键提示区：位于屏幕中的最下一行。显示一些常用命令的热键（快捷键），以方便用户操作。

### 2. 主要菜单命令介绍

TC 2.0 包含 8 组菜单，每组中又包含若干菜单项，现将主要用到的菜单命令列于表 1-1。

表 1-1 TC 2.0 的主要菜单

一级菜单名	二级菜单名	说 明
File(文件)	Load	装入已经存在的程序文件（相当于打开文件）
	Pick	该命令将显示一个菜单，其中包含最近打开的 9 个文件名，用光标选择其中一个回车后即可装入
	New	清除编辑窗口中的程序，供输入一个新程序

续表

一级菜单名	二级菜单名	说 明
File(文件)	Save	保存当前程序到文件中
	Write to	把当前程序写到另外命名的文件上, 相当于Windows “文件” 菜单中的 “另存为” 命令
	Directory	显示当前目录中的内容
	Change dir	显示当前目录, 并可切换当前目录
	Os shell	暂时回到 DOS 方式, 可敲入 exit 命令返回 TC
	Quit	退出 TC 集成环境
Edit(编辑)		进入编辑状态
Run(运行)	Run	执行程序
	Program reset	终止调试方式对程序的运行
	Goto cursor	执行程序到光标所在行, 本命令及后续两条命令在调试程序时非常有用
	Trace into	单步执行程序, 即每选择一次该命令, 将执行一条语句
	Step over	单步执行程序, 与上条命令的区别在于, trace into 在遇到函数调用时将跟踪执行至函数的内部, 而该选项直接完成函数的调用
	User screen	查看程序的运行结果
Compile(编译)	Compile	把编辑窗口中的程序编译成目标文件
	Link	把编辑窗口中的程序连接成可执行文件
	Make	把编辑窗口中的程序经编译、连接生成可执行文件
Project (项目)		在开发比较大的 C 语言程序时才会用到此菜单
Options(选择项)		用于设置开发环境的工作方式, 很复杂, 设置错误将导致 TC 不能正常工作

### 3. 窗口的操作

如果想把编辑窗口扩大到整屏, 可按 F5 键, 这时信息窗口将被遮住, 再按 F5 键又可以恢复成上、下两个窗口。如果编辑窗口被扩大到整屏, 而又想查看信息窗口, 可使用 F6 键进行窗口切换。如果在程序执行时又打开了观察窗口, 使用 F6 键可以对 3 个窗口进行切换, 切换过程是按某一个方向循环进行的。如果在 Windows 中运行 TC, 所打开的窗口往往较小, 边框线也不对。

## 二、上机要求

### 1. 基本要求

(1) 熟悉 TC 环境并掌握常用菜单功能及命令键的使用方法。

- (2) 学会程序的编辑方法。
- (3) 掌握程序的编译、链接和运行。
- (4) 学会查看程序的运行结果。

## 2. 较高要求

- (1) 理解目录的概念。
- (2) 深入调试程序。

## 三、上机内容



### 1. TC 2.0 的启动

TC 2.0 的启动方法有很多种。

第一种方法：选择“开始”“程序”“MS-DOS 方式”命令后，系统进入 DOS 方式，此时，键入 tc 命令并按回车键即可进入 TC 2.0 的集成环境。

第二种方法：在 TC.EXE 程序所在的文件夹下，找到 TC.EXE 并双击执行。

上机考试时，一般系统均会设置好自动搜索路径，因此在考生目录下直接运行 TC 即可。

运行过程中，可选择  按钮，在汉字和英文字符方式之间进行切换；还可选择  按钮，将屏幕切换至“全屏幕”方式。

### 2. 编辑一个新程序

(1) 在 TC 环境中，按 F10 键后，可选中菜单 File；然后通过按回车键或“ ”键，可弹出如图 1-2 所示界面。

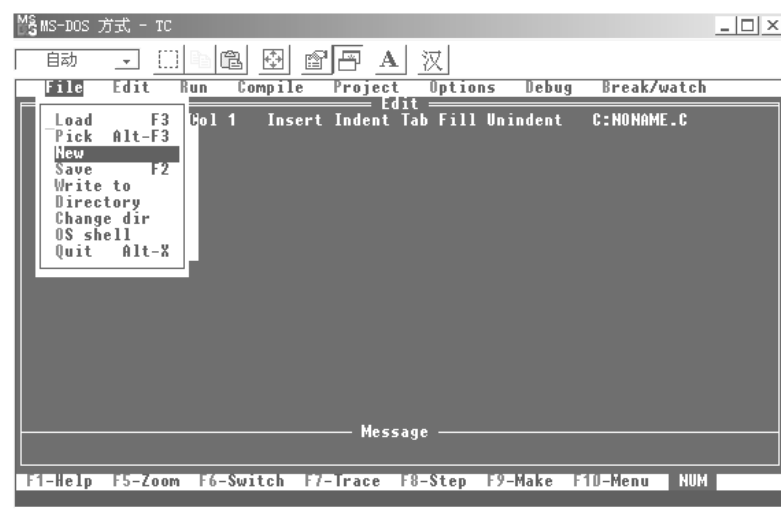


图 1-2 TC 的 File 菜单

(2) 将光标停在菜单 New 上并回车，进入新程序的编辑状态。在编辑区的左上角出现光标，此时可输入程序。

(3) 将下述一个简单的 C 程序输入编辑框，该程序的功能是在屏幕上显示一行文字，即 I am a student!

```
main()
{
```

```
printf("I am a student! ");  
}
```

注意：程序中的括弧、双引号和分号等所有标点符号均为半角字符，切勿输入错误。

### 3. 运行一个程序

选择菜单 Run Run 命令（如图 1-3 所示），或使用热键 Ctrl+F9，就可以运行一个程序。由于系统执行速度非常快，因此程序运行后很快就返回到编辑屏幕。



图 1-3 TC 的 Run 菜单

### 4. 查看程序的运行结果

有两种查看程序的运行结果的方法：第一种方法是，选择 Run User screen 命令，或使用热键 Alt+F5，系统暂时切换到运行结果屏幕，可显示该程序的运行结果，如图 1-4 所示，按任意键返回；第二种方法是，选择 File OS Shell 命令，系统返回到 DOS 方式下，此时，不仅可以看到运行结果，还可以执行一些简单的 DOS 命令。通过输入 exit 命令可返回到编辑环境。

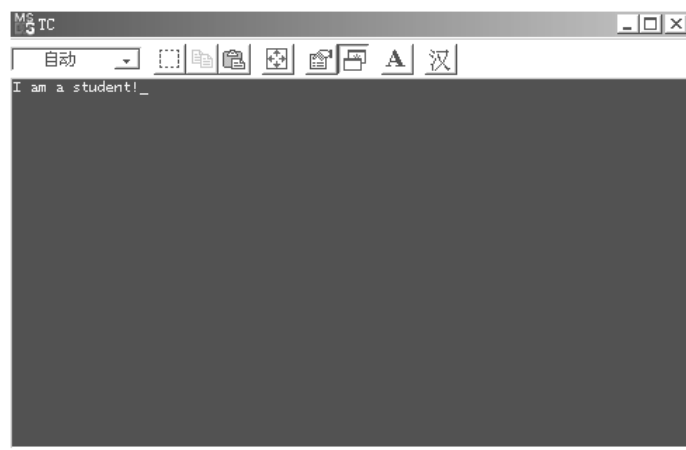


图 1-4 运行结果屏幕



#### 5. 保存程序

当输入程序后,可选择菜单 File Save 命令,或使用热键 F2 进行保存。如果当前编辑的是新程序,则系统将弹出一个对话框,提示用户为新程序输入一个名字;如输入 tcsy1-1.c 后回车,如果是已经存在(已经有文件名)的程序,则系统直接保存,不会有任何提示。

#### 6. 退出 TC 环境

选择菜单 File Quit 命令或使用热键 Alt+X 可退出 TC 环境,使计算机返回到操作系统的控制下。

#### 7. 打开已存在的程序

启动 TC,选择菜单 File Load 命令,或使用热键 F3 后,系统弹出一个对话框,如图 1-5 所示,此时,可直接输入程序文件名,如 tcsy1-1.c,也可回车后通过辅助窗口进行选择。如果程序源文件不在当前目录下,则必须输入文件的全路径。



图 1-5 对话框

#### 8. “另存为”命令

如要将正在编辑的程序文件 tcsy1-1.c 另存为 tcsy1-2.c,可选择菜单 File Write to 命令,在对话框中输入 tcsy1-2.c。

### 四、练习与思考

- (1) 输入一个简单的 C 程序,熟悉 TC 环境下基本的编辑方法和程序的运行方法。
- (2) main()的作用是什么?如果一个程序不包括 main()是否可以运行?
- (3) 如果程序在执行过程中无法结束(死机),则应如何强制结束它?如果直接关闭窗口正确吗?
- (4) 按上面介绍的步骤编辑一个新程序,并调试、运行。

程序功能:输入一个整数,并将这个整数显示在屏幕上。

程序如下:

---

```
main()
{
    int bl;                /*定义一个变量 bl,用于存放输入的整数*/
    scanf("%d",&bl);       /*输入一个整数,并存入变量 bl 中*/
    printf("%d",bl);       /*在屏幕上显示变量 bl 中存放的这个整数*/
}
```

---

注意:程序中的语句将在以后的章节中有详细介绍。

## 实验二 数据类型及顺序结构

### 一、知识要点

#### 1. 数据类型

数据类型有很多种，常见的有：整型(int, long int, unsigned int)、浮点型(float, double)、字符型(char)、指针型(\*)、无值型(void)、数组、枚举(enum)、联合(union)和结构(struct)。

#### 2. 常量

常量指在程序运行过程中不发生变化的量。它的数据类型有整型、实型和字符型。字符型常量包括字符常量和字符串常量，常量也可用标识符(const 或#define)定义。

#### 3. 变量

变量即在程序运行过程中发生变化的量。变量是有名字的标识符，在内存中占据一定的存储单元，并在该存储单元中存放变量的值。在使用变量时一定要先定义后使用，要用类型修饰符明确定义变量的数据类型。变量具有作用域，按照作用域的不同，可将变量分为局部变量和全局变量。

#### 4. 运算符

C 语言的运算符非常丰富，有算术运算符、关系与逻辑运算符、按位运算符和特殊运算符。

(1) 算术运算符和自加、自减运算符：+、-、\*、/、%、++、--。

(2) 关系与逻辑运算符：>、>=、<、<=、==、!=、!、&&、||。

(3) 按位运算符：~、&、|、^、<<、>>。

(4) 特殊运算符：?、:、&、\*、:、->、sizeof、[]、()。

#### 5. 表达式

用各种运算符连接起来的式子称为表达式，其格式为：<运算量><运算符><运算量>。

#### 6. 赋值语句

赋值语句具有计算和赋值功能，并可在内存中开辟由变量的数据类型决定的存储空间大小。

格式：<变量名>=<表达式>;

#### 7. 格式化输入/输出函数

(1) 格式化输入函数 scanf()。

格式：scanf(“<格式字符串>”，<地址表>);

功能：从标准输入设备（键盘）读取输入的信息。

说明：

地址表是需要读入数据的所有变量的地址，而不是变量本身。地址表中各变量间用“，”分开。

格式字符串包括两类：格式符和分隔符。

格式符：包括%c、%s、%d、%o、%x、%f、%e等。

分隔符：作用是当 scanf() 函数需要输入多个数据时用来分隔的，由格式符决定可是空格或逗号。

(2) 格式化输出函数 printf()。

格式：printf(“<格式字符串>”, <参量表>);

功能：向标准输出设备（显示器或打印机）按规定格式输出的信息。

说明：

参量表是要输出的一系列参数，参数个数应与格式字符串说明的输出参数格式个数一样多，且顺序一一对应，各参量间用“,” 分开。

格式字符串包括两部分内容：普通字符和格式字符。

普通字符：输出时按原样输出。

格式字符：是规定参数输出格式的格式字符，包括 %c、%s、%d、%u、%o、%x、%f、%e 等。

## 二、上机要求

### 1. 基本要求

- (1) 掌握数据类型、常量、变量、运算符和各种表达式的使用方法。
- (2) 掌握数据类型的定义和变量赋值的使用方法。
- (3) 掌握数据输入、输出时所用格式符的使用方法。
- (4) 熟悉各种运算符（特别是自加（++）和自减（--）运算符）的使用方法。
- (5) 熟练掌握调试 TC 源程序的方法和技巧。
- (6) 熟悉打开和保存程序的使用方法。

### 2. 较高要求

- (1) 更深刻理解表达式的计算过程和数据类型转换。
- (2) 验证各种输入、输出格式符及转义字符的使用。

## 三、上机内容

### 1. 定义变量类型、表达式及运算符的练习

- (1)  $x+a\%3*(x+y)\%2/4$  (设  $x=3.4$ ,  $a=7$ ,  $y=8.7$ )

先自己计算结果，再试着用程序求解，看得到的结果是否一致。

实现步骤：

进入 TC 环境。在 C 盘的 TC 文件夹下双击 TC.EXE 文件。

选择 File 中的 New 命令清除原内容，输入如下程序代码：

---

```
main()
{
    float  x=3.4,y=8.7, s;          /*定义变量数据类型并赋初值*/
    int    a=7;
    s= x+a%3*(int)(x+y)%2/4;        /*写出表达式并赋给变量 s，如果省略
                                     (int)，则结果大不同*/
    printf("s=%8.2f", s);           /*按规定的宽度输出变量 s 的值*/
}
```

---

注意：注释内容不输入。

编译程序：按下 Alt+C 键选择 Compile to OBJ 命令，或按 Alt+F9 键，对源程序进行编译，检查程序是否有语法错误。

运行程序：按 Alt+R 键选择 Run 命令，或按 Ctrl+F9 键运行程序。

查看运行结果：按 Alt+R 键选择 User Screen 命令，或按 Alt+F5 键，分析程序运行结果，注意观察与你计算的结果是否相同，按任意键返回编辑窗口。

运行结果是：s= 3.40。

保存程序：按 Alt+F 键选择 Save 命令，或按 F2 键，在弹出的对话框中输入文件名 tcsy2-1.c。

(2) (float)(a+b)/2+(int)x%(int)y (设 a=2, b=3, x=3.5, y=2.5)

实现步骤：

移动光标删除、修改、编辑字符，输入如下内容。

---

```
main()
{
    float x=3.5,y=2.5,s;           /*定义变量数据类型并赋初值*/
    int a=2,b=3;
    s=(float)(a+b)/2+(int)x%(int)y; /*计算并强制进行数据类型转换*/
    printf("s=%8.2f", s);          /*按规定的宽度输出变量 s 的值*/
}
```

---

运行程序。

查看运行结果。结果是：s= 3.50。

按 Alt+F 键选择 Write to 命令，在弹出的对话框中输入文件名 tcsy2-2.c 保存文件。

(3) 练习将整型值赋给字符型变量（即为 ASCII 码值），按格式输出。

操作方法同上。

---

```
main()
{
    char c1, c2;                   /*定义变量数据类型*/
    c1=68;                         /*给变量赋值*/
    c2=70;
    printf("%c %c\n", c1, c2);     /*用字符格式输出*/
    printf("%d, %d\n", c1, c2);    /*用整型数据格式输出*/
}
```

---

在该程序中说明了字符型数据在特定情况下可作为整型数据处理，整型数据有时也可以作为字符型数据处理。

运行结果是：

D F

68,70

(4) 若程序中有错误,则细心观察编译过程中提示的错误信息。  
操作方法同上。

---

```
main()
{
    float a,b; int c=5;           /*定义变量数据类型*/
    printf("%d",a%b);             /*错误,因 a,b 变量没赋值*/
    printf("%d",C);               /*错误,因 C 变量为大写字母*/
}
```

---

(5) 练习将小写字母转换为大写字母。

下列程序中因为 a, b 是字符型,值为 ASCII 码。如字符 a 的 ASCII 码值为 97,减去 32 为 65,即为字符 A 的 ASCII 码值。

输入如下程序代码。

---

```
main()
{
    char a,b;                     /*定义变量数据类型*/
    a='x';                        /*给变量赋值*/
    b='y';
    a=a-32;                       /*将小写字母转换为大写字母*/
    b=b-32;
    printf("%c,%c\n%d,%d\n",a,b,a,b); /*输出字符及其 ASCII 码值*/
}
```

---

运行程序,结果是:

X,Y

88,89

## 2. 转义字符的练习

(1) 输入如下程序代码。

---

```
main()
{
    char c1='a',c2='b',c3='c',c4='\101',c5='\116'; /*定义变量数据类型并赋初值*/
    printf("c1=%c c2=%c\tc3=%c\n",c1,c2,c3);      /*按格式输出, \t是制表位,同
    printf("\t\tc4=%c c5=%c",c4,c5);              Tab 键*/
}
```

---

(2) 程序运行结果是:

```
c1=a c2=b          c3=c
      c4=A c5=N
```

## 3. 宏定义和符号常量的练习

输入如下程序代码：

---

```

#define R 5                                /*用宏定义方法定义常量 R 值为 5 */
main()
{
    const PI=3.1415926;                    /*用符号常量方法定义常量 PI 的值*/
    float s;
    s=PI*R*R;                              /*计算圆的面积*/
    printf("s=%f\n",s);
}

```

---

## 4. 逻辑运算符的练习

输入如下程序代码：

---

```

main()
{
    int a=4,b=5,c=0,d;                    /*定义变量数据类型并赋初值*/
    d=!a&&!b||!c;                        /*因为 a 不为零，非 a(!a )值为 0，0 与
                                           任何值的逻辑运算结果均为 0；又因为 c
                                           为零，则 !c 值为 1，所以 d 值为 1 */

    printf("d=%d\n",d);
}

```

---

运行程序，结果是：d=1。

## 5. 自加、自减运算符的练习

观察程序的运行结果。

---

```

main()
{
    int i , j , m , n ;
    i=6;                                  /*给变量赋值*/
    j=9;
    m=++i;                                /*先执行 i 加 1，后存入 m 中*/
    n=j++;                                /*先将 j 的值存入 n 中，之后执
                                           行 j 加 1 并将值存入 j 中*/

    printf("%d,%d,%d,%d",i,j,m,n);        /*输出变量的值*/
    i=6;
    j=9;
    m=i++;
    n=++j;
    printf("%d,%d,%d,%d\n" , i,j,m,n);
    printf("i=%d,j=%d,m=%d,n=%d\n",i,j,m,n);
}

```

---

```
i=6;
j=9;
m=-i;
n=-j;
printf("%d,%d,%d,%d\n",i,j,m,n);
i=6;
j=9;
printf("%d,%d\n", i++, j++);
i=6;
j=9;
printf("%d,%d\n", ++i, ++j );
}
```

运行结果是：

7,10,7,9

7,10,6,10

i=7,j=10,m=6,n=10

5,8,5,8

6,9

7,10

#### 四、思考和练习题

(1) 编程，将“PassWord”进行加密。加密规则是：用原来的字母后面第3个字母代替原来的字母。例如，字母“a”后面第3个字母是“d”，“a”加密后为“d”。因此，“PassWord”经过加密后变为“SdxxZrug”。请编写一则程序，用赋初值的方法使c1、c2、c3、c4、c5、c6、c7、c8这8个变量的值分别为‘P’、‘a’、‘s’、‘s’、‘W’、‘o’、‘r’、‘d’，经过运算，使其分别变为‘S’、‘d’、‘v’、‘v’、‘Z’、‘r’、‘u’、‘g’，并输出。

(2) 编程，从键盘上输入一个字符，回显该字符并输出其ASCII码值。

## 实验三 选择结构程序设计

### 一、知识要点

选择结构是 C 程序的 3 种基本结构之一，它用来根据所指定的条件是否满足在两种或两种以上的条件中选择一种并执行相应的操作。C 语言中用来表达选择结构程序设计的语句有两种：if 语句和 switch 语句。

说明：选择结构中的判断结果是逻辑量。C 语言表示逻辑量的方法是：以 0 代表“假”，以非 0 代表“真”。

#### 1. if 语句

if 语句通过判断给出的条件是否满足来执行相应的操作。

格式 1：if (<条件>) <语句>;

格式 2：if (<条件>) { <语句序列>; }

这两种形式的语句是最基本的形式，为单边语句，它首先判断给出的条件是否为真，为真则执行，否则什么也不做。

格式 3：if (<条件>){ <语句序列 1>; }else { <语句序列 2>; }

这种形式的语句为双边语句，它首先判断给出的条件是否为真，是则执行语句序列 1，否则执行语句序列 2。

格式 4：if (<条件 1>){ <语句序列 1>; }  
else if (<条件 2>){ <语句序列 2>; }  
else if (<条件 3>){ <语句序列 3>; }

...

else { <语句序列 n+1>; }

这种形式的语句，它首先判断给出的条件 1 是否为真，是则执行语句序列 1，否则判断给出的条件 2 是否为真，……，直到最后。

#### 2. switch case 多重分支语句

格式：switch (<变量或表达式>)

```
{  
    case <常量表达式 1>: { <语句序列 1>; }  
    case <常量表达式 2>: { <语句序列 2>; }  
    ...  
    case <常量表达式 n>: { <语句序列 n>; }  
    default: { <语句序列 n+1>; }  
}
```

其中：

(1) 变量或表达式可以是数值型或字符型表达式；

(2) 语句序列可以是任意语句。若是一条语句，则不加花括号。若是复合语句，则要加花括号。



(3) 每一个 case 的常量表达式的值必须互不相同。

### 3. break 语句

break 语句一般放在复合语句的最后，可以使流程跳出 switch 结构。

## 二、上机要求

### 1. 基本要求

- (1) 熟练掌握条件语句的编程技巧。
- (2) 熟练掌握逻辑表达式的书写格式及使用方法。
- (3) 写出上机操作步骤和程序清单，并写出上机调试报告。

### 2. 较高要求

掌握一些简单的算法。

## 三、上机内容

### 1. 练习 if (<条件>) <语句>; 的用法

输入任意整数，判断其是否为奇数，并输出信息。

实现步骤：

- (1) 进入 TC 环境。
- (2) 输入如下程序代码。

---

```
main()
{
    int a ;
    printf("input number: ");          /*提示输入信息*/
    scanf("%d" , &a);                  /*输入任意整数*/
    if(a%2!=0)                          /*当 a 被 2 除余数不等于 0 时*/
        printf("%d is odd number\n " , a);    /*输出 a 是奇数的信息*/
}
```

---

### (3) 编译并运行程序。

屏幕将转入运行界面，并显示如下字符串：

input number:

此时，在光标处输入如下数据：

5

回车后，该程序的显示结果为：

5 is odd number

再运行一次程序，若输入 6，则屏幕上不显示任何信息。为什么？

(4) 保存程序：按 Alt+F 键选择 Save 命令或按 F2 键，在弹出的对话框中输入文件名 tcsy3-1.c。

### 2. 练习 if (<条件>) { <语句序列 1>; } else { <语句序列 2>; } 的用法

输入任意 3 个整数，输出其中的最大数和最小数。

实现步骤：

对上题进行编辑。

(1) 输入如下程序代码：(黑体部分为新添加的内容和语句)

```
main()
{
    int a, b, c, max, min;          /*定义变量, max 用于存放最大数, min
                                    用于存放最小数*/
    printf("input three number: "); /*提示输入 3 个数*/
    scanf("%d%d%d", &a, &b, &c);    /*输入任意 3 个整数, 之间用空格分隔*/
    if(a>b)                          /*先判断 a、b, 若 a 大于 b*/
    {
        max=a;                      /*则 a 放在 max 中, b 放在 min 中*/
        min=b;
    }
    else
    {                                /*否则 b 放在 max 中, a 放在 min 中*/
        max=b;
        min=a;
    }
    if(max<c)                        /* 再将 max 与 c 比较, 如果 max 小于 c,
                                    则将 c 存入 max 中*/
        max=c;
    else                             /*否则, 将 min 与 c 比较, 如果 min 大于
                                    c, 则将 c 存入 min 中*/
        if(min>c)
            min=c;
    printf("max=%d min=%d", max, min); /*输出 max 中的值和 min 中的值*/
}
```

(2) 编译并运行程序。

屏幕将转入运行界面, 并显示如下字符串:

input three number:

此时, 在光标处输入如下数据 (数据中间以空格分隔):

3 56 12

回车后, 该程序的显示结果为:

max=56 min=3

(3) 保存程序: 按 Alt+F 键选择 Write to 命令, 在弹出的对话框中输入文件名 tcsy3-2.c。

### 3. 练习逻辑表达式的用法

计算下列函数的值。

$$y = \begin{cases} x^2 + 2x - 6 & x < 0, x = -3 \\ x^2 - 5x + 6 & 0 < x < 10, x = 2, x = 3 \\ x^2 - x - 15 & x = 10 \end{cases}$$

实现步骤：

(1) 清除上题内容。

(2) 分析题意, 写出逻辑表达式。

对于数学式子  $x < 0, x \neq -3$ , 在 TC 中应写为  $x < 0 \ \&\& \ x \neq -3.0$ 。

对于数学式子  $0 \leq x < 10, x \neq 2, x \neq 3$  在 TC 中应写为  $x \geq 0 \ \&\& \ x \neq 2.0 \ \&\& \ x \neq 3.0 \ \&\& \ x < 10.0$ 。

对于数学式子  $x \geq 10$ , 在 TC 中应写为  $x \geq 10.0$ 。

(3) 输入如下程序代码。

---

```
main()
{
    float x, y;
    printf("input number x: ");           /*提示输入 x*/
    scanf("%f", &x);                     /*输入实型数 x 的值*/
    if(x < 0 && x != -3.0) y = x * x + 2 * x - 6; /*根据条件, 分别计算 y 的值*/
    if(x >= 0 && x != 2.0 && x != 3.0 && x < 10.0)
        y = x * x - 5 * x + 6;
    if(x >= 10.0)
        y = x * x - x - 15;
    printf("y=%f\n", y);                 /*输出实型数 y 的值*/
}
```

---

(4) 编译并运行程序。

屏幕将转入运行界面, 并显示如下字符串。

input number x:

此时, 在光标处输入如下数据:

-1.0

回车后, 该程序的显示结果为:

y=-7.000000

若再运行一次程序, 输入 5.0, 则输出结果为: y=6.000000。

(5) 保存程序: 按 Alt+F 键选择 Write to 命令, 在弹出的对话框中输入文件名 tcsy3-3.c。

#### 4. 练习 switch case 语句的用法

已知某公司员工的每月保底薪水为 600 元, 每月按所接工程的利润 profit (整数), 再给一部分奖金, 利润提成的关系如下 (计量单位: 元):

profit < 1000	没有提成;
1000 <= profit < 2000	提成 10%;
2000 <= profit < 5000	提成 15%;
5000 <= profit < 10000	提成 20%;
10000 <= profit	提成 25%。

实现步骤:

(1) 清除上题内容。

(2) 分析题意, 利用 switch case 语句, 写出开关变量 (等级) 的表达式。

等级 = (利润 - 1) / 1000, 求出整数, 因包括等于, 所以减 1。

根据等级的整数值，就可计算提成金额，程序中的粗体部分。

(3) 编辑输入如下程序代码。

编写初始化程序段，即定义变量，并给变量赋值。

---

```
main()
{
    long  int profit ;           /*定义利润 profit 变量为长整型*/
    int   grade , premium ;     /*grade 为等级 , premium 为提成金额*/
    float salary =600 ;         /*salary 为薪水 , 初值为 600
    printf("Input  profit: ");  /*提示输入利润信息*/
    scanf("%ld" , &profit) ;    /*输入长整型数利润的值*/
    grade=(profit-1)/ 1000 ;     /*计算等级的值*/
    ...
}
```

---

添加计算提成金额的程序。

---

```
switch(grade)
{ case  0: break ;              /*profit  1000 */
  case  1: premium = profit*0.1 ; break ; /*1000<profit  2000 */
  case  2:                    /*2000<profit  5000 */
  case  3:
  case  4: premium = profit*0.15 ; break ;
  case  5:                    /*5000<profit  10000 */
  case  6:
  case  7:
  case  8:
  case  9: premium = profit*0.2 ; break ;
  default: premium = profit*0.25 ;      /*10000<profit */
}
```

---

添加输出每月职工实发薪水的语句。

---

```
printf("salary=%.2f\n" , salary+ premium) ; /*输出当月薪水 */
```

---

(4) 编译并运行程序。

屏幕将转入运行界面，运行界面显示：

Input profit:

此时，在光标处输入如下利润数据：

2000

回车后，该程序显示的实发薪水为：

salary=800.00

如：再运行一次程序，输入利润数据为 3000，则输出结果为：salary=1049.00。

(5) 保存程序：按 Alt+F 键选择 Write to 命令，在弹出的对话框中输入文件名 tcsy3-4.c。

#### 四、思考和练习题

(1) 已知银行整存整取存款年息利率分别为：

一年期：2.25%；二年期：2.70%

三年期：3.24%；五年期：3.60%

活期：0.72%

要求：输入存钱的本金和期限月数，求到期时利息与本金的合计（银行到期利息计算公式：利息 = 本金 × 年息利率 × 存款年限）。程序中应当根据存款的年限确定利率。程序通过后保存文件名为 tcsy3-5.c。

例如：若在提示信息处输入:5000.0，12

则输出结果为：Total= 5090.00

---

```
#include <stdio.h>
main( )
{
    int year, month;
    float money, rate, interest, total;    /* 定义本金、月利率、利息和本利合计*/
    printf("Input money and month =?");
    scanf("%f, %d", &money, &month);    /* 输入本金和月数*/
    year=month/12;
    switch (year)                          /*根据年限确定利率*/
    {                                       /*编写程序段，计算 rate 的值*/

    }

    if (year>=1)
    {
        interest = money * rate * year;    /*计算利息*/
        interest = interest-interest*0.2;    /*计算税后利息*/
        total = money + interest;          /*计算本金加利息*/
    }
    printf("Total=%8.2f\n", total);        /*按规定格式输出*/
}
```

---

(2) 通过键盘输入字符，将输入的字符分为控制、数字、大写字母、小写字母和其他字符等 5 类。ASCII 码表中的前 32 个字符为控制字符，‘0’～‘9’为数字，‘A’～‘Z’为大写字母，‘a’～‘z’为小写字母和其他字符。

阅读下列程序，在【1】、【2】、【3】处填写合适的逻辑表达式，并上机调试程序。程序通

过后保存文件名为 tcsy3-6.c。

---

```
#include <stdio.h>                                /*因为要用到非格式化输入函数*/
main ()
{
    char c ;
    printf ("Enter a character:");
    c=getch();                                     /*输入任意字符*/
    if (c<0x20)                                    /*ASCII 码值在 0 ~ 20 之间*/
        printf ("\nIs a control character.\n");    /*输出是控制字符的信息*/
    else if (___【1】___)                           /*ASCII 码值在 48 ~ 57 之间*/
        printf ("\nIs a digit character.\n");      /*输出是数字字符的信息*/
    else if (___【2】___)                           /*ASCII 码值在 65 ~ 90 之间*/
        printf ("\nIs a captal character.\n");     /*输出是大写字母的信息*/
    else if (___【3】___)                           /*ASCII 码值在 97 ~ 122 之间*/
        printf ("\nIs a lower character.\n");      /*输出是小写字母的信息*/
    else
        printf ("\nIs an other character.\n");     /*输出是其他字符的信息*/
}
```

---

## 实验四 循环结构程序设计

### 一、知识要点

循环控制结构（又称为重复结构）是 C 程序中的另一个基本结构。在实际问题中，常常需要进行大量的重复处理，循环结构可以使我们只写很少的语句，而让计算机反复执行，从而完成大量类似的计算。

C 语言提供了 3 种基本的循环语句：for 语句、while 语句和 do-while 语句。

#### 1. for 语句

for 语句是循环控制结构中使用最广泛的一种循环控制语句，特别适合已知循环次数的情况。它的一般形式为：

```
for(<初始化>; <条件表达式>; <增量>)
{ 语句序列};
```

初始化总是一个赋值语句，它用来给循环控制变量赋初值；条件表达式是一个关系表达式，它决定什么时候退出循环；增量定义循环控制变量每循环一次后按什么方式变化。这 3 个部分之间用“；”分开。

说明：

(1) for 循环中的“语句序列”称为循环体，如是多条语句，要用“{”和“}”将参加循环的语句括起来。

(2) for 循环中的“初始化”、“条件表达式”和“增量”都是选择项，即可以缺省，但“；”不能缺省。如果省略了初始化，表示不对循环控制变量赋初值；省略了条件表达式，则不做其他处理时便成为死循环；省略了增量，则不对循环控制变量进行操作，这时可在语句体中加入修改循环控制变量的语句。

#### 2. while 语句

格式：while(条件) {语句序列}；

while 语句表示当条件为真时，便执行语句，直到条件为假才结束循环，并继续执行循环程序外的后续语句。

与 for 语句一样，while 语句总是在循环的头部检验条件，这就意味着循环可能什么也不执行就退出。

#### 3. do-while 语句

格式：

```
do
{ 语句序列};
while(条件);
```

do-while 语句与 while 语句的不同在于：它先执行循环中的语句，然后再判断条件是否为真，如果为真则继续循环；如果为假，则终止循环。因此，do-while 循环至少要执行一次循环体。同样当有多条语句参加循环时，要用花括号把它们括起来。

#### 4. 循环的嵌套

一个循环的循环体中有另一个循环叫循环嵌套。这种嵌套过程可以有很多重：一个循环外面仅包围一层循环叫二重循环，一个循环外面包围两层循环叫三重循环，一个循环外面包围多层循环叫多重循环。

3 种循环语句 for、while、do-while 可以互相嵌套自由组合。但要注意的是，各循环必须完整，相互之间绝不允许交叉。

#### 5. break 与 continue 语句

有时，需要在循环体中提前跳出循环，或者在满足某种条件下，不执行循环中剩下的语句，而立即从头开始新一轮循环，这时就要用到 break 或 continue 语句。

##### (1) break 语句。

在学习 switch 语句时，已经接触到 break 语句，在 case 子句执行完后，通过 break 语句使控制立即跳出 switch 结构。在循环语句中，break 语句的作用是在循环体中测试到应立即结束循环时，使控制立即跳出循环结构，转而执行循环语句后的语句。

##### (2) continue 语句。

continue 语句只能用于循环结构中，一旦执行了 continue 语句，程序就跳过循环体中位于该语句后的所有语句，提前结束本次循环周期并开始新一轮循环。

#### 6. 数组

数组是一个由若干同类型变量组成的集合，引用这些变量时可用同一名字。数组均由连续存储单元组成，最低地址对应于数组的第一个元素，最高地址对应于最后一个元素，数组可以是一维的，也可以是多维的。在程序中应先说明，后使用。在实验中详细介绍。

## 二、上机要求

### 1. 基本要求

- (1) 掌握循环语句的使用方法。
- (2) 掌握多重循环的使用方法及注意事项。
- (3) 掌握数组的使用方法。

### 2. 较高要求

掌握多种编程算法。

## 三、上机内容

### 1. 练习 for 循环语句的使用方法

编程计算  $n$  和  $n!$ 。

实现步骤：

(1) 分析题目， $n$  是计算累加，即  $1+2+3+\dots+n$ 。而  $n!$  是计算阶乘，即  $1*2*3*\dots*n$ 。这类计算用 for 循环语句比较合适。输入数据  $n$ ，用变量 sum 存放累加和，用变量 product 存放阶乘积。

(2) 进入 TC 环境。

(3) 编辑输入如下程序代码。

编写初始化程序，即定义变量，赋初值，并输入数据。



---

```

main()
{
    int n;
    long int sum=0, product=1;           /* sum 存放累加和, product
                                           存放阶乘积*/

    printf("input number n= : ");       /*提示输入信息*/
    scanf("%d", &n);                    /*输入任意整数 n*/

    ...
}

```

---

添加计算和输出部分的程序。

---

```

for(i=1; i<=n; i++)                    /*开始从 1 到 n 循环*/
{
    sum +=i;                            /*计算累加和*/
    product *=i;                        /*计算阶乘*/
}

printf("sum=%ld    %d!=%ld", sum, n, product);

```

---

#### (4) 运行程序。

屏幕将转入运行界面，并显示如下字符串：

input number n=:

此时，在光标处输入数据：

5

回车后，该程序的显示结果为：

sum=15, 5!=120。

#### (5) 保存程序：按 F2 键，在弹出的对话框中输入文件名 tcsy4-1.c。

#### 2. 练习 while 循环语句的使用方法

编程，利用格里高利公式求  $\pi$  的值，直到最后一项的绝对值小于等于  $10^{-6}$  为止。格里高利公式如下：

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \Delta$$

实现步骤：

(1) 分析题目，找出算法，后一项与前一项的关系是分母多 2，符号相异。用变量  $n$  作为计数器，用变量  $t$  作为符号位的变化，用变量  $p$  存放  $\frac{1}{n}$  的值。用当型循环判断  $p$  的值是否大于  $10^{-6}$ ，当小于等于  $10^{-6}$  时停止。

循环体中的计算应为：

$n=n+2$

$t=-t$

$p=1.0/n$

---

```
pi=pi+t*p
```

(2) 清除原内容。

(3) 编辑输入如下程序代码。

编写初始化程序段，定义变量类型并赋初值。

---

```
#include <stdio.h>
#include <math.h>
main()
{
    double pi=1.0, p=1.0;          /*pi 存储累加和, p 存储分数的值, 初值均赋 1*/
    long int n=1;
    int t=1;                        /*t 用于控制符号的正负变化*/
    ...
}
```

---

添加计算程序段，用当型循环判断最后一项的绝对值小于等于  $10^{-6}$  为止。

---

```
while (p>=1e-6)                    /*当 p>=1e-6 条件为真时, 执行循环体*/
{
    n = n + 2;
    p = 1.0/(float)n;               /*将 n 转换为实型数*/
    t = -t;                         /*改变 t 的正负*/
    pi = pi + t*p;
}
pi = pi * 4;                        /*因循环中计算的是  $\pi/4$  的值, 这里要求计算  $\pi$  */
printf("pi=%lf\n", pi);
```

---

(4) 运行程序。

通过按 Alt+F5 键查看屏幕显示结果为：

```
pi=3.141595
```

(5) 保存程序：按 Alt+F 键选择 Write to 命令，在弹出的对话框中输入文件名 tcsy4-2.c。

### 3. 练习 do-while 循环语句的使用方法

编程找出同时能被 n 和 m 整除的数。

实现步骤：

(1) 分析题目，从键盘上输入 n 和 m，用验证的方法找出数 k。即 k 是 n 和 m 的倍数。

(2) 清除原内容。

(3) 编写如下程序代码，粗体为找数程序段。

---

```
main()
{
    int  n, m, k=1;
    scanf("%d%d", &n, &m);          /*输入任意整数 n, m*/
    do                               /*当循环开始*/
```

---

---

```

{
    /*如 k 被 n 和 m 同时整除，则结束循环*/
    if (k%n==0 && k%m==0)break ;
    k++;
}while(k!=0) ;
printf("k=%d\n" , k);          /*输出倍数*/
}

```

---

(4) 运行程序。

屏幕将转入运行界面，运行界面无提示，等待用户输入数据，此时，在光标处输入如下两个数，用空格作为分隔符：

4 7

回车后，屏幕显示为：

k=28

(5) 保存程序：按 Alt+F 键选择 Write to 命令，在弹出的对话框中输入文件名 tcsy4-3.c。

#### 4. 练习多重循环的使用方法

编程计算用 3、4、5、6 四个数字，能组成多少个互不相同且无重复数字的三位数？都是多少？

可填在百位、十位、个位的数字都是 3、4、5、6。组成所有的排列后再去掉不满足条件的排列。

实现步骤：

(1) 清除原内容。

(2) 编写如下程序代码。因为要找出 3 位互不相同且无重复数字的数，此时要用三重循环来实现。

---

```

main()
{
    int i , j , k ;
    printf("\n");          /*输出一个空行*/
    for(i=3 ; i<7 ; i++)    /*外循环*/
        for(j=3 ; j<7 ; j++)
            for (k=3 ; k<7 ; k++)          /*内循环*/
            {
                if (i!=k && i!=j && j!=k)    /*确保 i、j、k 三位互不相同*/
                    printf("%d , %d , %d\n" , i , j , k);          /*输出无重复数字的三位数*/
            }
}

```

---

(3) 编译、运行程序。

(4) 按 Alt+F5 键查看运行结果。

(5) 保存程序：按 Alt+F 键选择 Write to 命令，在弹出的对话框中输入文件名 tcsy4-4.c。

## 5. 练习数组的使用方法

编程实现，用随机函数产生 10 个 200 ~ 300 之间的数，存入数组中并输出，之后挑选出是素数的数并统计个数，输出素数及个数。

实现步骤：

(1) 清除原内容。

(2) 输入如下程序代码。

编写头文件，因为要用到数学、时间、库函数。

---

```
#include <math.h>           /*头文件，因为要用到数学、时间、
#include <time.h>           库函数*/
#include <stdlib.h>
#include <stdio.h>
```

---

在程序中添加产生随机数存入数组中并输出数组元素的程序段(黑体部分为新添加的语句)。产生 200 ~ 300 之间随机数的公式为： $a[i] = \text{random}(99) + 200$ 。

---

```
...
main()
{
    int i, j, k=0, p, flag, a[10];           /*定义程序中用到的变量*/
    randomize();                             /*调用随机种子函数*/
    for(i=0; i<10; i++)                     /*循环产生 10 个随机数，并输出*/
    {
        a[i]=random(99)+200;
        printf("a(%d)=%d  ", i, a[i]);
        if(i%5==0)printf("\n");            /*每行输出 5 个数就换行*/
    }
    printf("\n");                           /*输出一个空行*/
    ...
```

---

挑选素数：如果  $a[i]$  能被 2 整除，则是偶数，继续判断下一个数，若不是，则循环继续进行，找另一个因子，若都不能被整除，则为素数(黑体部分为新添加的语句)。

---

```
...
for(i=0; i<10; i++)                       /*外循环*/
{
    if (a[i] % 2==0)                       /*判断 a[i] 是否为偶数，是则继续判断下
        continue;                         一个数*/
    flag=1;
    p=(int)(sqrt((double)(a[i])));        /*将 a[i] 开方*/
```

---

---

```

for(j=2;j<=p;j++)
    if ((a[i]%j)==0)
    {
        flag = 0 ;
        break;
    }
if (flag == 1)
{
    k++;
    printf("%d  ",a[i]);
    if(k%5==0)printf("\n");
}
printf("k=%d \n",k);
}

```

---

/\*内循环 ,查找 a[i] 是否能被另一个因子整除，是则退出内循环\*/

/\*内循环结束后，验证是否为素数，是则计数并输出\*/

/\*输出素数的个数\*/

---

(3) 编译，运行程序。

(4) 按 Alt+F5 键查看运行结果。

(5) 保存程序：按 Alt+F 键选择 Write to 命令，在弹出的对话框中输入文件名 tcsy4-5.c。

#### 四、思考和练习题

(1) 下列程序的功能是将两个字符串连接一起并输出。分析语句并填空，上机运行程序并观察结果，保存文件名为 tcsy4-6.c。

程序代码：

---

```

#include "stdio.h"
main()
{
    char a[]="acegikm" ;
    char b[]="bdfhjlnpq" ;
    char c[80] ;
    int i=0 , j=0 , k=0 ;
    while(a[i]!='\0'  【1】 b[j]!='\0')
    {
        if (a[i])
            c[k]= 【2】 ;
        else
            c[k]=b[j++] ;
        k++ ;
    }
    c[k]= 【3】 ;
    puts(c) ;
}

```

---

/\*定义字符串变量并赋值\*/

/\*用当型循环，【1】填写||\*/

/\*如果 a[i]非 0\*/

/\* 【2】填写 a[i++] \*/

/\*如果 a[i]为 0\*/

/\*连接 b 串\*/

/\* 【3】填写 ‘\0’ \*/

/\*输出两字符串连接后的新串\*/

---

(2) 阅读下列程序, 本程序的功能是输入任意一个数, 判断其是否为素数。

程序中给出了出错的位置, 请进行改正, 然后上机运行程序并观察结果, 保存文件名为 tcsy4-7.c。

---

```

#include <conio.h>
#include <stdio.h>
main( )
{   int   m , k ;
    printf( "\n Please enter m : " ) ;
    scanf("%d" , &m ) ;                /*输入任一数*/
    k = 2 ;
    while ( k <= m && (m%k))            /*用当型循环*/
    /*****found  error*****/
        k++                            /*应改为 k++ ; */
    /*****found  error*****/
    if (m = k )                        /*应改为 m==k*/
        printf( "YES\n" ) ;
    else
        printf( "NO!\n" ) ;
}

```

---

(3) 编程计算下列公式的值。

$$S = 1 \times \frac{(1)^2}{3} \times \frac{(4)^2}{3 \times 5} \times \frac{(6)^2}{5 \times 7} \times \Lambda \times \frac{(2k)^2}{(2k-1)(2k+1)}$$

当 k=10 时, S=1.533852。

## 实验五 函 数

### 一、知识要点

C 语言是通过函数来实现模块化程序设计的。所以 C 语言应用程序往往是由多个函数组成的，每个函数分别对应各自的功能模块。在程序中只有一个主函数 `main()`，可由多个子函数组成。函数是 C 源程序的基本模块，通过对函数模块的调用实现特定的功能。

C 语言不仅提供了极为丰富的库函数，还允许用户把自己的算法编写成一个个相对独立的函数模块，然后用调用的方法来使用函数。

可以说 C 程序的全部工作都是由各式各样的函数完成的，所以也把 C 语言称为函数式语言。由于采用了函数模块式的结构，C 语言易于实现结构化程序设计。程序的层次结构清晰，便于程序的编写、阅读和调试。

#### 1. 库函数

库函数由 C 系统提供，用户无须定义，也不必在程序中作类型说明，只需在程序前包含有该函数原型的头文件即可，在程序中直接调用。如前面用到过的 `printf`、`scanf`、`sqrt`、`randomize`、`random` 等函数均属此类。

#### 2. 用户自定义函数

用户自定义函数是由用户按需要编写的函数。对于用户自定义函数，不仅要在程序中定义函数本身，而且在主调函数模块中还必须对该被调函数进行类型说明，然后才能使用。

#### 3. 有返回值函数

此类函数被调用执行完后将向调用者返回一个执行结果，称为函数的返回值。如数学函数即属于此类函数。由用户定义的这种要返回函数值的函数，必须在函数定义和函数说明中明确给出返回值的类型。

#### 4. 无返回值函数

此类函数用于完成某项特定的处理任务，执行完成后不向调用者返回函数值。由于函数无须返回值，用户在定义此类函数时可指定它的返回为“空类型”，空类型的说明符为 `void`。

#### 5. 无参函数

在无参函数中，函数定义、函数说明及函数调用中均不带参数，主调函数和被调函数之间不进行参数传送。此类函数通常用来完成一组指定的功能，可以返回或不返回函数值。

#### 6. 有参函数

有参函数即为带参函数，在函数定义及函数说明中都有参数，称为形式参数(简称为形参)。在函数调用时也必须给出参数，称为实际参数(简称为实参)。进行函数调用时，主调函数将把实参的值传送给形参，供被调函数使用。

#### 7. 函数说明

当一个函数没有明确说明类型时，C 语言的编译程序自动将整型 (`int`) 作为这个函数的

默认类型，默认类型适用于很大一部分函数。当有必要返回其他类型数据时，首先必须给函数以明确的类型说明符，而且函数类型的说明必须处于对它的首次调用之前。

### 8. 返回语句

返回语句 `return` 有两个重要用途：第一，当执行到它时可使函数立即退出，即使程序返回到调用语句处继续进行；第二，它可以用来回送一个数值。

### 9. 局部变量

在函数内部定义的变量称为局部变量。局部变量仅由其被定义的模块内部的语句所访问。局部变量在自己的代码模块之外是不可知的。

### 10. 全局变量

全局变量贯穿整个程序，并且可被任何一个模块使用。在整个程序执行期间保持有效。全局变量定义在所有函数之外，可由函数内的任何表达式访问。

### 11. 动态存储变量

从变量的作用域原则出发，可将变量分为全局变量和局部变量；若从变量的生存期来分，可将变量分为动态存储变量和静态存储变量。

动态存储变量可以是函数的形式参数、局部变量、函数调用时的现场保护和返回地址。这些动态存储变量在函数调用时分配存储空间，函数结束时释放存储空间。

### 12. 静态存储变量

在编译时分配存储空间的变量称为静态存储变量，其定义形式为在变量定义的前面加上关键字 `static`。

## 二、上机要求

### 1. 基本要求

- (1) 掌握库函数和自定义函数的使用方法。
- (2) 掌握函数实参与形参的对应关系及“值传递”的方式。
- (3) 掌握全局变量、局部变量、动态变量、静态变量的使用方法。

### 2. 较高要求

掌握函数的嵌套调用和递归调用的方法。

## 三、上机内容

### 1. 练习库函数的用法

例 1：数学库函数的使用。

数学库函数包含在 `math.h` 头文件中，包括三角函数、指数、对数等。如：输入任意整数，计算并输出它的平方根、指数、对数的值。

实现步骤：

- (1) 进入 TC 环境。
- (2) 编辑输入如下程序代码。

---

```
#include <stdio.h>                /*启用输入、输出头文件*/
#include <math.h>                  /*启用数学库函数的头文件*/
main()
```

---



---

```
{
    int a;
    printf("input number: ");           /*提示输入信息*/
    scanf("%d",&a);                     /*输入任意整数*/
    printf("%5.2f\n",sqrt(a));          /*输出函数的值*/
    printf("%6.3f\n",exp(a));
    printf("%8.2f\n",log(a));
}
```

---

(3) 编译并运行程序。

屏幕将转入运行界面，运行界面显示：

input number:

此时，在光标处输入如下数据：

9

回车后，屏幕显示为：

3.00

813.084

2.20

(4) 将程序存为 tcsy5-1.c。

例 2：字符串的连接、复制、比较等库函数的使用。

如：从键盘上输入字符串，调用字符串库函数完成字符串的连接、复制、比较功能，并输出结果。实现步骤：

(1) 分析题目，从键盘上输入字符串，用 gets() 函数从键盘上读入一串字符，分别编程调用字符串库函数去实现。

(2) 练习字符串的连接。

编辑输入如下程序代码。（黑体字部分）

---

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
main()
{
    char str1[]="My name is ",str2[20];           /*定义字符型数组*/
    printf("input your name:\n");                 /*提示输入信息*/
    gets(str2);                                     /*从键盘输入字符串*/
    strcat(str1,str2);                               /*两字符串连接后存入 str1 中*/
    puts(str1);                                     /*输出字符串*/
}
```

---

运行程序。

屏幕将转入运行界面，运行界面显示：

input your name:

此时，在光标处输入如下字符串：

hao she ping

回车后，屏幕显示结果为：

My name is hao she ping

(3) 练习字符串的复制。将字符串的连接程序段加上注释符。

编辑输入如下程序代码。

---

```

...
char str1[20],str2[20];           /*定义字符型数组*/
printf("input string:\n");       /*提示输入信息*/
gets(str2);                      /*从键盘输入字符串*/
strcpy(str1,str2);               /*将字符串 str2 复制到 str1 中*/
puts(str1);                      /*输出字符串 str1*/
puts(str2);                      /*输出字符串 str2*/
...

```

---

运行程序。

屏幕将转入运行界面，运行界面显示：

input string:

此时，在光标处输入如下字符串：

this is a book!

回车后，屏幕显示结果为：

this is a book!

this is a book!

(4) 练习字符串的比较。将字符串的复制程序段加注释符。

编辑输入如下程序代码。

---

```

...
char str1[]="Computer Language";
char str2[20];
int k;
printf("input string:\n");       /*提示输入信息*/
gets(str2);                      /*从键盘输入字符串*/
k=strcmp(str1,str2);             /*将 str2 与 str1 比较，结果值存入 k 中*/
if(k==0)                         /*如果结果值为 0，则两串相等*/
    printf("str2=str1\n");       /*输出 str1=str2*/
if(k>0)                          /*如果结果值>0*/
    printf("str2>str1\n");       /*输出 str2>str1*/
if(k<0)                          /*如果结果值<0*/

```

---

---

```
printf("str2<str1\n");          /*输出 str2<str1*/
```

---

```
...
```

运行程序。

屏幕将转入运行界面，运行界面显示：

input string:

此时，在光标处输入如下字符串：

Language

回车后，屏幕显示结果为：

str2<str1

如再次运行程序，若输入：Computer Language，则屏幕显示结果为：

str2=str1

(5) 将程序另存为 tcsy5-2.c。按 Alt+F，选择 Write to，输入文件名。

## 2. 练习用户自定义函数

例 1：字符串库函数的使用。

字符串库函数包含在 string.h 头文件中，包括字符串运算的各种函数。

如：从键盘上输入任一字符，如果是小写字母则转换为大写字母，如果是大写字母则转换为小写字母，否则直接输出。

实现步骤：

(1) 分析题目，从键盘上输入任一字符，用 getchar()函数读入一个字符，用条件结构判断输入的字符是小写字母还是大写字母，用字符转换函数完成题目的要求。

(2) 清除原内容。

(3) 编辑输入如下程序代码。

编写判断输入字符是否为小写字母的函数 islower()。程序代码如下：

---

```
int islower(char str1)          /*因代码中给函数名赋予值 1，则应加 int 说明符，形参为字符型*/
{
    if(str1>=97 && str1<=122)   /*判断形参变量是否 ASCII 码在小写字母区间*/
        islower=1;             /*给函数名赋值 1*/
    return islower;             /*返回原调用函数处*/
}
```

---

编写判断输入字符是否为大写字母的函数 isupper()。程序代码如下：

---

```
int isupper(char str1)          /*因代码中给函数名赋 1 值，则应加 int 说明符，形参为字符型*/
{
    if(str1>=65 && str1<=90)    /*判断形参变量是否 ASCII 码在大写字母区间*/
        isupper =1;            /*给函数名赋值 1*/
    return isupper;             /*返回原调用函数处*/
}
```

---

编写主函数程序。定义变量，输入一个字符，进行判断并输出。

---

```

#include <stdio.h>                /*启用包含文件*/
#include <string.h>
#include <conio.h>
main()
{
    char str;                    /*定义字符型变量*/
    str=getchar();               /*输入一个字符*/
    if (islower(str))            /*调用自定义函数，判断是否是小写字母*/
        printf("%c\n", toupper(str)); /*是则调用系统函数，转换为大写字母并输出*/
    else
        if (isupper(str))        /*否则判断是否是大写字母*/
            printf("%c\n", tolower(str)); /*是则转换为小写字母并输出*/
        else
            printf("%c\n", str);    /*否则直接输出*/
}

```

---

(4) 编译并运行程序。

屏幕将转入运行界面，等待用户输入。

此时，在光标处输入任一字符：

a

回车后，屏幕显示为：

A

可多运行几次，观察结果。

(5) 将程序另存为 tcsy5-3.c。按 Alt+F，选择 Write to，输入文件名。

例 2：编写函数 fun()，将字符串 tt 中的大写字母都改为对应的小写字母，其他字符不变。

实现步骤：

(1) 分析题目，从键盘上输入任一字符串，用 gets() 函数读入字符串，调用用户自定义函数 fun()，并输出转换后的结果。

fun() 函数的功能：将字符串中的逐个字符进行比较并转换存入原相应的位置，因为是字符串，故 fun() 函数应定义为字符型指针。

(2) 清除原内容。

(3) 编写用户自定义函数代码。

---

```

#include <stdio.h>                /*定义头文件*/
#include <conio.h>
char *fun(char tt[])             /*定义字符指针型函数，形参也是字符型*/
{
    int i;

```

---

---

```

    for(i=0;tt[i];i++)                /*循环*/
    {
        if((tt[i]>='A')&&(tt[i]<='Z'))    /*判断是否是大写字母*/
            tt[i]+=32;                /*是则转换为小写字母并存入原位置*/
    }
    return(tt);                        /*返回字符串的内容*/
}
...

```

---

(4) 编写主函数代码。

---

```

main()
{
    char tt[81];                        /*定义字符串变量*/
    printf("\nPlease enter a string: ");
    gets(tt);                          /*输入任意字符串*/
    printf("\n%s\n",fun(tt));          /*调用函数，并输出*/
}

```

---

(5) 编译并运行程序。

(6) 输入字符串，观察结果。若输入 AbC , Def , 则输出:abc , def。

(7) 将程序保存为 tcsy5-6.c。

例 3：编程求 4 个数中的最大数。要求利用自定义函数 fun()求两数中的最大数。

实现步骤：

(1) 分析题目，主函数的功能是：从键盘上输入 4 个数，多次调用自定义函数 fun()，求出最大数并输出结果。

自定义函数 fun()的功能是：比较两个数的大小，并返回最大数。

(2) 清除原内容。

(3) 编写用户自定义函数代码。

---

```

int fun(int x,int y)                  /*定义函数是整型，形参也是整型*/
{
    int z;
    z=(x>=y? x:y);                    /*用条件语句实现*/
    return(z);                        /*返回两数中的较大数*/
}

```

---

(4) 编写主函数代码。

---

```

main()
{
    int a,b,c,d,max1;
    printf("\nPlease input numbers: ");

```

---

---

```

scanf("%d,%d,%d,%d",&a,&b,&c,&d);           /*输入 4 个整数*/
max1=fun(a,b);                             /*调用函数*/
max1=fun(max1,b);
max1=fun(max1,c);
max1=fun(max1,d);
printf("\nmax1=%d\n",max1);               /*输出最大数*/
}

```

---

(5) 编译并运行程序。

(6) 输入数据, 观察程序运行结果。若输入 23, 67, 889, 1, 则输出: max1=889。

(7) 将程序保存为 tcsy5-7.c。

例 4: 编程输入任意数, 利用自定义函数 fun() 验证其是否是素数。

实现步骤:

(1) 分析题目, 主函数的功能是: 从键盘上输入数据, 调用自定义函数 fun(), 根据返回的值判断是否是素数, 并输出信息。

自定义函数 fun() 的功能是: 利用循环的方法检验一个数是否有  $2 \sim \sqrt{n}$  中的数的约数, 有则为偶数, 否则循环到  $\sqrt{n}$ , 若没有找到, 则此数为素数, 并返回一个值, 由主函数输出信息。

(2) 清除原内容。

(3) 编写用户自定义函数代码。

---

```

#include <stdio.h>
#include <math.h>
int fun(int x)                             /*定义函数是整型, 形参也是整型*/
{
    int p,j;
    p=(int)(sqrt((double)x));              /*对 x 求平方根, 存入 p 中*/
    for(j=2;j<=p;j++)                     /*循环验证*/
        if ((x%j)==0)                    /*如果 x 被 j 整除有约数, 则不是素数, 返回 0*/
            return (0);
    return (1);                           /*如果无约数则是素数, 返回 1*/
}
...

```

---

(4) 编写主函数代码。

---

```

main()
{
    int a,k;
    printf("input number:");
    scanf("%d",&a);                       /*输入任意数*/
}

```

---

---

```

k=fun(a);                                /*调用函数，返回值赋给 k */
if (k==1)                                /*如果 k 为 1，则输入数是素数*/
    printf("\n%d is prime number\n",a);
else
    printf("\n%d is not prime number \n",a);    /*否则输入数不是素数*/
}

```

---

(5) 编译并运行程序。

(6) 输入数据，观察运行结果。若输入 23，则输出:23 is prime number；若输入 24，则输出:24 is not prime number。

(7) 将程序保存为 tcsy5-8.c。

例 5：编程求出一个  $2 \times M$  整型二维数组中最大元素的值。

实现步骤：

(1) 分析题目，主函数的功能是：定义并给二维数组赋值，输出自定义函数 fun() 的值。

自定义函数 fun() 的功能是：求二维数组中最大元素的值，并将此值返回主函数。

求二维数组中最大元素的算法是：利用循环逐行、逐列地比较，直到找出最大数。

(2) 清除原内容。

(3) 编写用户自定义函数代码。

---

```

#define M 4                                /*用宏定义 M 为 4*/
#include <stdio.h>
int fun(int a[][M])                        /*定义形参为整型二维数组*/
{
    int i,j,max1=a[0][0];                  /*将数组中的第 1 个元素赋给 max1*/
    for(i=0;i<2;i++)                      /*外循环为行*/
        for(j=0;j<M;j++)                  /*内循环为列*/
            if(max1<a[i][j])               /*如果 max1 小于某个元素，则将某个元素存入
            max1 中*/
                max1=a[i][j];
    return max1;                          /*循环结束后，返回最大数*/
}

```

---

(4) 编写主函数代码。

---

```

main()
{
    int arr[2][M]={5,8,3,45,76,-4,12,82};    /*定义数组并赋值*/
    printf("max=%d\n",fun(arr));              /*输出调用函数返回的值*/
}

```

---

(5) 编译并运行程序。

(6) 输出结果是 max=82。

(7) 将程序保存为 tcsy5-7.c。

#### 四、思考和练习题

(1) 请修改下面 fun 函数中的错误,使它得出正确的结果,此 fun 函数的功能是计算下列分数数列的前 n 项之和。

$$\frac{2}{1}, \frac{3}{2}, \frac{5}{3}, \frac{8}{5}, \frac{13}{8}, \frac{21}{13}, \Delta$$

当输入 n=5 时,则输出 8.391667。

程序代码:

---

```

#include <stdio.h>
#include <conio.h>
double fun(int n)
{
    int a,b,c,k;
    double s;
    s=0.0;                                /*给变量赋初值*/
    a=2;
    b=1;
    for(k=1;k<=n;k++)                    /*循环计算*/
    {
        /******found   error*****/
        s=s+a/b;                          /*正确的为 s=s+(double)a/b;*/
        c=a;
        /******found   error*****/
        a=b;                              /*正确的为 a=a+b;*/
        b=c;
    }
    return s;
}
main()
{
    int n=5;
    printf("sum=%lf\n",fun(n));          /*输出调用函数返回的值*/
}

```

---

(2) 请修改下面 fun 函数中的错误,使它得出正确的结果,此 fun 函数的功能是将浮点数保留 m 位小数(m 不大于 6),对第 m+1 位四舍五入,例如,输入 354.6789,保留 2 位小数应输出 354.68。



程序代码：

---

```

#include <stdio.h>
#include <conio.h>
float fun(float x,int n)
{
    float y=x;
    int k;
    /******found   error******/
    for(k=1;k<n;k++)                /*正确的为 k=0;*/
    {
        y*=10;
        x*=10;
    }
    y=(int)y;
    if(x-y>=0.5)
    /******found   error******/
        x++;                        /*正确的为 y++;*/
    for(k=0;k<n;k++)
        y/=10;
    /******found   error******/
    return x;                        /*正确的为 return y;*/
}
main()
{
    float x;
    int m;
    printf("input x:\n");
    scanf("%f",&x);                /*输入 354.6789*/
    printf("input m:\n");
    scanf("%d",&m);                /*输入 2*/
    printf("The result=%f\n",fun(x,m)); /*输出调用函数返回的值*/
}

```

---

(3) 编程将十进制数转换为二进制数。

(4) 用选择法对数组中的  $n$  个元素按从大到小的顺序进行排序。

(5) 下列程序中有错，请改正。假定整数数列中的数不重复，并存放在数组中。函数 fun 的功能是删除数列中值为  $x$  的元素， $n$  中存放的是数列中元素的个数。

---

```

#include <stdio.h>

```

---

```
#define N 20
fun(int *a,int n,int x)
{
    int p=0,i;
    a[n]=x;
    while( x!=a[p] )
        p=p+1;
    /******found error*****/
    if(P=n)
        return -1;
    else
    {
        for(i=p;i<n;i++)
            /******found error*****/
            a[i+1]=a[i];
        return n-1;
    }
}

main()
{
    int w[N]={-3,0,1,5,7,99,10,15,30,90},x,n,i;
    n=10;
    printf("The original data :\\n");
    for(i=0;i<n;i++)
        printf("%5d",w[i]);
    printf("\\nInput x (to delete): ");
    scanf("%d",&x);
    printf("Delete : %d\\n",x);
    n=fun(w,n,x);
    if ( n==-1 )
        printf("***Not be found!***\\n\\n");
    else
    {
        printf("The data after deleted:\\n");
        for(i=0;i<n;i++)
            printf("%5d",w[i]);
        printf("\\n\\n");
    }
}
```

---

## 实验六 数 组

### 一、知识要点

#### 1. 一维数组

数组是一组有序数据的集合，数组中每一个元素的类型相同。用数组名和下标可唯一确定数组中的元素。一维数组需要掌握它的定义、初始化以及数组元素的引用。

(1) 定义一维数组的方式。

类型说明符 数组名[常量表达式]；

如：int a[10]；

它表示定义一个数组，数组名为 a，有 10 个元素，每个元素的类型均为 int。这 10 个元素分别是：a[0]、a[1]、a[2]、a[3]、a[4]、...、a[8]、a[9]。

(2) 一维数组元素的引用。

C 语言规定，不能引用整个数组，只能逐个引用元素，元素引用方式：

数组名[下标]

如：a[0] = a[5] + a[7] - a[2\*3]；

(3) 一维数组的初始化。

对一维数组进行初始化，只需在定义时指定初始值，编译器把初值赋给数组变量。

如：static int a[10] = { 0,1,2,3,4,5,6,7,8,9}；

#### 2. 二维数组

(1) 定义二维数组的一般形式。

类型说明符 数组名[常量表达式][常量表达式]；

如：float a[2][3]；

它表示定义一个数组，数组名为 a，a 为 2×3(2 行 3 列)的数组，它有 6 个元素，每个元素的类型均为 float。这 6 个元素分别是：a[0][0]、a[0][1]、a[0][2]、a[1][0]、a[1][1]、a[1][2]。

(2) 二维数组的初始化。

分行赋值，如 static int a[3][4] = {{1,2,3,4},{5,6,7,8},{9,10,11,12}}；

全部数据写在一个大括号内，如：static int a[3][4] = {1,2,3,4,5,6,7,8,9,10,11,12}；

部分元素赋值，如：static int a[3][4] = {{1},{5},{9}}；

仅对 a[0][0]、a[1][0]、a[2][0]赋值，其余元素未赋值（对于静态数组，编译器自动为未赋值元素指定初值 0；对于动态数组，未赋值元素的初值是随机的）。

如果对全部元素赋初值，则第一维的长度可以不指定，但必须指定第二维的长度。

如：static int a[3][4]={1,2,3,4,5,6,7,8,9,10,11,12}；

与下面的定义等价：

static int a[ ][4]={1,2,3,4,5,6,7,8,9,10,11,12}；

### 3. 字符数组和字符串

C 语言中无字符串变量，字符串是存放在字符数组中的，而字符串又是一个整体，所以必须确定字符串的起始位置，C 语言以“\0”作为字符串结束标志。

#### (1) 字符数组的定义。

如 `char c[10];`

它表示定义 `c` 为字符数组，包含 10 个元素。

#### (2) 字符数组的初始化。

逐个元素初始化，如：`static char c[10] = {'I',' ','a','m',' ','h','a','p','p','y'};`

若初始化数据少于数组长度，则剩余元素自动为“空”（\0）。

如：`static char c[10] = {'c',' ','p','r','o','g','r','a','m'};`说明：9 个数组元素。

指定初值时，若未指定数组长度，则长度等于初值个数。

如：`static char c[] = {'I',' ','a','m',' ','h','a','p','p','y'};`

#### (3) 字符数组的引用。

引用字符数组的一个元素，可得到一个字符。

#### (4) 字符串。

C 语言中，字符串作为字符数组处理。字符数组可以用字符串来初始化。

如：`static char c[] = {"I am happy"};` 或 `static char c[] = "I am happy";`

字符串在存储时，系统自动在其后加上结束标志\0。但字符数组并不要求其最后一个元素是\0。

## 二、上机要求

### 1. 基本要求

(1) 掌握一维数组的定义、数组元素的引用以及初始化。

(2) 掌握二维数组的定义、数组元素的引用以及初始化。

(3) 掌握字符数组的定义、数组元素的引用以及初始化。

(4) 掌握常用字符串处理函数。

### 2. 较高要求

掌握三维以及多维数组的定义、数组元素的引用以及初始化。

## 三、上机内容

### 1. 熟悉一维数组的定义、数组元素的引用以及初始化

编写程序并调试、运行：输入 10 个数，用选择法对 10 个数排序（由小到大）。

测试数据：1 3 5 7 9 2 4 6 8 10（任意顺序的 10 个整数）

预期输出结果：2 3 4 5 6 7 8 9 10（按由小到大的顺序排列好的 10 个整数）

实现步骤：

(1) 进入 TC 环境。

(2) 输入如下程序。

编写初始化程序，即定义一个能存放 10 个元素的数组 `a`，并赋初值。

---

```
#define N 10
```

```
/*宏定义 N 的值为 10*/
```

---

---

```

main()
{
    int a[N];                /*定义数组 a*/
    int i,j,min,temp;        /* i,j 作为循环变量，min 存放当前最小值的下
                                标，temp 用做两变量交换时存放的临时变量*/

    printf(" input %d datas:\n",N); /*输入数据时的提示信息显示*/
    for(i=0;i<N;i++)
        scanf("%d",&a[i]);        /*用循环语句输入 10 个整数*/
}

```

---

添加输出程序，以便显示数组中的每一个元素，用于显示 10 个数据的原始顺序。

---

```

    for(i=0;i<N;i++)        /*用循环输出未排序前数组中的 10 个整数*/
        printf("%d  ",a[i]);
    printf("\n");           /*使光标显示在 10 个数的下一行*/

```

---

添加排序程序，用于对 10 个整数进行排序。

---

```

for(i=0;i<N-1;i++)        /*要进行 N-1 轮比较*/
{
    min=i;                 /*将剩余未排序元素中的第一个元素的下标赋
                                给 min，暂时作为最小值下标*/

    for(j=i+1;j<N;j++)     /*进行第 i+1 轮比较，比较后将未排序元素中的
                                最小值的下标赋给 min*/
        if(a[min]>a[j]) min=j;

    if(i!=min)             /*如果最小数不是未排序元素中的第一个元素，
                                则将最小数 a[min]与 a[i]交换*/
    {
        temp=a[i];
        a[i]= a[min];
        a[min]=temp;
    }
}

```

---

再次添加输出程序（黑体部分为添加的程序），用于显示排好序的数据，以便与原始顺序进行比较。

---

```

#define N 10
main()
{
    int a[N];
    int i,j,min,temp;
    printf(" input %d datas:\n",N);
    for(i=0;i<N;i++)

```

---

---

```

scanf("%d",&a[i]);
for(i=0;i<N;i++)
    printf("%d  ",a[i]);
printf("\n");
printf("\n");
for(i=0;i<N-1;i++)
{
    min=i;
    for(j=i+1;j<N;j++)
        if(a[min]>a[j]) min=j;
    if(i!=min)
    {
        temp=a[i];
        a[i]= a[min];
        a[min]=temp;
    }
}
for(i=0;i<N;i++)                /*用循环输出排好序的数组中的 10 个整数*/
    printf("%d  ",a[i]);
printf("\n");
}

```

---

### (3) 运行程序。

屏幕将转入运行界面，并显示如下字符串：

input 10 datas:

此时，在光标处输入如下数据：

1 3 5 7 9 2 4 6 8 10

回车后，该程序的显示结果为：

1 3 5 7 9 2 4 6 8 10

1 2 3 4 5 6 7 8 9 10

### (4) 保存该文件，文件名为 tcsy6-1.c，退出 TC。

#### 2. 熟悉二维数组的定义、数组元素的引用以及初始化

对于一个  $3 \times 4$  的矩阵： $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 9 & -8 & 7 & 6 \\ 8 & 10 & -5 & 2 \end{pmatrix}$ ，要求编写程序以求出其中最大的那个元素的值

及其所在的行号和列号。

预期输出结果：最大值为 10，所在的行为 2，列为 1（注意：得到的行列值是从 0 开

始的)

实现步骤：

(1) 进入 TC 环境。

(2) 然后编辑输入如下程序：

编写初始化程序，即定义一个 3×4 的二维数组 a，并初始化各元素。

---

```
main ()
{
    int i,j,row=0,column=0,max;          /* i,j 作为循环变量，row,column 用于
                                           存放行号和列号，初始值为 0 行 0 列，
                                           max 用于存放最大值*/
    static int a[3][4]={ { 1,2,3,4},{ 9,8,7,6},{ 8,10,-5,2} }; /*初始化数组 a，并赋初值*/
}
```

---

添加“求 3×4 矩阵的最大值”程序。

---

```
max = a[0][0];          /*将 a[0][0]作为临时最大值赋给 max*/
for(i=0; i<=2; i++)     /* 用两重循环遍历全部元素 */
    for(j=0; j<=3; j++)
        if (a[i][j] > max ) /*将临时最大值 max 与每一个元素 a[i][j]进行
                               比较，若 a[i][j]>max，把 a[i][j]作为新的临时最
                               大值，并将 i 和 j 分别保存在 row 和 column 中。
                               当全部元素比较完后，max 中存放的值即是整个
                               矩阵全部元素的最大值*/
        {
            max = a[i][j];
            row = i;
            column = j;
        }
```

---

添加输出程序，用于显示最大值及最大值所在的行和列（黑体部分为添加的程序）。

---

```
main ()
{
    int i,j,row=0,column=0,max;
    static int a[3][4]={ { 1,2,3,4},{ 9,8,7,6},{ -10,10,-5,2} };
    max = a[0][0];
    for(i=0; i<=2; i++)
        for(j=0; j<=3; j++)
            if (a[i][j] > max )
            {
                max = a[i][j];
                row = i;
                column = j;
            }
}
```

---

---

```
printf("max=%d, row=%d, column=%d\n", max, row, column); /*最大值及所在的行和列*/
}
```

---

(3) 运行程序。

该程序的显示结果为：max=10, row=2, column=1。

(4) 保存该文件，文件名为 tcsy6-2.c，退出 TC。

### 3. 熟悉字符数组的定义、数组元素的引用以及初始化

编写程序并调试、运行：输入一行字符，统计其中有多少个单词（单词间以空格分隔）。

测试数据：I am a boy

预期结果：有 4 个单词

实现步骤：

(1) 进入 TC 环境。

(2) 然后编辑输入如下程序。

定义一个字符数组 string，并编写输入一个字符串程序。

---

```
#include "stdio.h" /* gets()函数在该头文件中定义 */
main ()
{
    char string[81]; /*定义字符数组 string */
    int i, num = 0, word = 0; /*num 存放单词数(初值赋为 0)。word=0 表示前一字符为
                               空格，word=1 表示前一字符不是空格，word 初值为 0*/
    char c; /*c 用来临时存放当前从字符数组 string 中取出的字符*/
    gets(string); /*输入字符串语句，将用户输入的字符串保存到 string 中*/
}
```

---

在输入数据之后添加“统计单词个数”的程序。

---

```
for(i=0; (c=string[i])!= '\0'; i++) /*循环检测数组 string 中的每一个字符是否为空格*/
    if (c==' ') /*判断当前的字符是否为空格，如果当前字符是空格，则
        word = 0; 表示未出现新单词，并使 word = 0，num 不累加；如果
        else if (word == 0) 当前字符不是空格且前一个字符为空格（即 word = 0），
        { 则表示新单词出现，使 word = 1，num 加 1；如果当前字
        word = 1; 符不是空格且前一个字符为非空格（即 word = 1），则表
        num++; 示未出现新单词，同时 num 不累加*/
    }
```

---

添加“输出单词个数”程序（黑体部分为添加的程序语句）。

---

```
#include "stdio.h" /* gets()函数在该头文件定义 */
main ()
{
    char string[81];
    int i, num = 0, word = 0;
```

---



```
char c;
printf("Please input a string:\n");
gets(string);
for(i=0; (c=string[i]) != '\0'; i++)
    if (c==' ')
        word = 0;
    else if (word == 0)
    {
        word = 1;
        num++;
    }
printf("There are %d words in the line\n", num);    /*输出单词个数*/
}
```

(3) 运行程序。

屏幕将转入运行界面，运行界面显示：

Please input a string:

此时，在光标处输入如下字符串：

I am a boy.

回车后，该程序的显示结果为：

There are 4 words in the line

(4) 保存该文件，文件名为 tcsy6-3.c，退出 TC。

#### 四、练习与思考

(1) 编写一个程序，将两个一维数组中的对应元素的值相减后显示出来。

(2) 用冒泡法对 10 个整数排序。

(3) 编写一个程序，将两个  $5 \times 6$  的二维数组中的对应元素的值相加后显示出来。

(4) 已知 A 是一个  $3 \times 4$  的矩阵，B 是一个  $4 \times 5$  的矩阵，编写一个程序，求  $A \times B$  得到的新矩阵 C，并显示出来。

(5) 从键盘上输入一个字符串，并统计其中的英文字母（不区分大小写）、空格符、数字字符及其他字符的个数。

(6) 从键盘输入两个字符串，并将其首尾相接后输出。

## 实验七 指针（一）

### 一、知识要点

#### 1. 指针变量

一个变量的地址称为该变量的“指针”。如果一个变量专门用来存放另一变量的地址（指针）的，则称它为指针变量。

指针变量的定义格式：类型标识符 \*标识符；

```
如：int    a,*p1=&a；  
     float  x,*p2=&x；  
     char   ch,*p3=&ch；
```

指针也可以作为函数参数，它属于单值传递，调用函数时将实参的指针值传递给形参，调用结束后形参中的指针值会释放。

#### 2. 指针与数组

（1）指向一维数组元素的指针变量的定义与引用。

如果一个变量用来存放数组元素的地址，则该指针变量是指向数组元素的指针变量。

```
如：int  a[10],*p=a；
```

表示指针变量 p 存放了第一个数组元素的地址。

指向数组元素的指针变量可以指向任意一个数组元素，如果指针变量 p 指向某一个数组元素，则 p++指向下一个元素。

指向数组元素的指针变量也可以作为函数的参数，它属于地址传递，实参、形参既可以用数组名，也可以用指向数组元素的指针变量作参数，总之，实参数组和形参数组是同一个数组。

（2）指向多维数组的指针和指针变量。

指向二维数组元素的指针变量的定义。

```
如：int  a[2][3],*p=a[0]；
```

指向二维数组元素的指针变量可以指向任意一个数组元素，如果指针变量 p 指向某一个数组元素，则 p++指向下一个数组元素。

指向由 m 个数组成的一维数组的指针变量的定义。

定义格式为：类型标识符 (\*函数名)[数组长度]；

```
如：int a[2][3],(*p)[3]；p=a；
```

该指针变量 p 指向由 3 个数组成的一维数组，如果指针变量 p 指向二维数组某一行的首地址，则 p++指向下一行的首地址。

（3）指针与字符串

如果一维数组各元素存放的是字符类型的数据，该数组为字符数组，字符数组的首地址可以存放在字符指针中，通过指向字符串的指针来处理字符串。

## 二、上机要求

### 1. 基本要求

- (1) 掌握指针变量的定义与引用。
- (2) 掌握指针作为函数参数的方法。
- (3) 掌握指针与数组的使用方法。
- (4) 熟悉用 TC 集成环境来调试指针程序的方法。

### 2. 较高要求

掌握指向多维数组的指针变量的使用方法。

## 三、上机内容

### 1. 熟悉指针变量的定义和引用

编写程序并调试、运行：输入 a 和 b 两个整数，按从大至小的顺序输出 a 和 b。

测试数据：5，8

预期输出结果：8 5

实现步骤：

- (1) 进入 TC 环境。
- (2) 然后编辑输入如下程序：

编写初始化程序。定义两个指针变量,并赋初值。

---

```
main()
{
    int *p1,*p2,*p,a,b ;           /*定义指针变量 p1 , p2 以及整型变量 a , b*/
    p1=&a ; p2=&b ;                 /* p1 , p2 分别指向 a 和 b 两个整数变量的地址*/
    printf("Please input a,b : ") ; /*输入 a 和 b 两个整数的提示信息*/
    scanf ("%d,%d",&a,&b) ;        /*输入 a 和 b 两个整数*/
}
```

---

添加比较程序。

---

```
if(a<b)           /*如果 a<b , 则交换指针 p1、 p2 的值，使 p1 指向 a 和 b 两
{                  个整数中的大数，使 p2 指向 a 和 b 两个整数中的小数*/
    p=p1 ;
    p1=p2 ;
    p2=p ;
}
```

---

添加“按先大后小的顺序输出 a 和 b”的程序（黑体部分为添加的程序）。

---

```
main()
{
    int *p1,*p2,*p,a,b ;
```

---

---

```

    p1=&a ; p2=&b ;
    printf("Please input a,b : ");
    scanf ("%d,%d",&a,&b) ;
    if(a<b)
    {
        p=p1 ;
        p1=p2 ;
        p2=p ;
    }
    printf("max=%d,min=%d\n",*p1,*p2) ;    /* p1 所指向的变量的值为大数 , p2 所指
}                                           向的变量的值为小数*/

```

---

(3) 运行程序。

屏幕将转入运行界面，运行界面显示：

Please input a,b :

此时，在光标处输入如下数据：

5,8

回车后，该程序的显示结果为：

max=8,min=5

(4) 保存该文件，文件名为 tcsy7-1.c，退出 TC。

## 2. 熟悉指针作为函数参数的方法

编写程序并调试、运行：用指针作为函数参数来完成上题要求。

实现步骤：

(1) 进入 TC 环境。

(2) 然后编辑输入如下程序。

编写交换函数 swap ( )，其中指针变量 p1，p2 作为函数的参数，该函数能使指针变量 p1，p2 所指向的变量的值互换。程序如下：

---

```

swap(int *p1,int *p2)                /*函数参数为指向整数变量的指针变量 p1 和 p2*/
{
    int p ;                          /* p 是临时变量*/
    p=*p1 ;                          /* p1 , p2 所指向的两个整数变量值发生交换*/
    *p1= *p2 ;
    *p2=p ;
}

```

---

添加主程序，在主程序中调用 swap ( ) 函数。

---

```

main( )
{
    int a ,b ;

```

---

---

```

int *pointer_1,*pointer_2 ;
pointer_1=&a ; pointer_2=&b ;
printf("Please input a,b : ");
scanf("%d,%d",&a,&b);
if(a<b)                                /*如果 a<b ,则调用函数 swap ( ) ,该函数
    swap(pointer_1,pointer_2);          将 pointer-1 和 pointer-2 所指向的变量 a 和 b
                                        的值进行交换*/
printf("max=%d,min=%d\n",a,b);         /*输出 a 的值（大数）, b 的值（小数）*/
}

```

---

（3）运行程序。

屏幕将转入运行界面，运行界面显示：

Please input a,b :

此时，在光标处输入如下数据：

5,8

回车后，该程序的显示结果为：

max=8,min=5

（4）保存该文件，文件名为 tcsy7-2.c，退出 TC。

3. 熟悉指向一维数组元素的指针变量的定义与引用

编写程序并调试、运行：利用指向数组元素的指针变量输入、输出数组中的 10 个整数。

实现步骤：

（1）进入 TC 环境。

（2）然后编辑输入如下程序。

编写初始化程序。即定义指针变量，使它指向数组的第一个元素，并输入 10 个数。

---

```

main()
{
    int  a[10],*p,i ;                /*定义存放 10 个数的数组 a，指针变量 p */
    p=a ;                            /*指针变量 p 指向数组的第一个元素 */
    printf("Please input 10 datas : \n"); /*用于运行时提示用户输入的信息*/
    for (i=0 ; i<10 ; i++)           /*循环输入 10 个整数，指针变量 p 逐次指向数
        scanf("%d",p++);              组的下一个元素*/
    printf("\n");
}

```

---

添加“输出 10 个数”的程序（黑体部分为添加的程序）。

---

```

main()
{
    ...

```

---

---

```

printf("\n");
p=a;                                /*将指针变量 p 再指向数组的第一个元素*/
for(i=0; i<10; i++,p++)             /*循环输出 10 个整数，将指针变量 p 逐次指向数组的
    printf("%3d",*p);                下一个元素*/
printf("\n");
}

```

---

(3) 运行程序。

屏幕将转入运行界面，运行界面显示：

Please input 10 datas

此时，在光标处输入如下数据：

1 2 3 4 5 6 7 8 9 10

回车后，该程序的显示结果为：

1 2 3 4 5 6 7 8 9 10

(4) 如果显示的结果正确，则保存该文件，文件名为 tcsy7-3.c，退出 TC。

#### 4. 熟悉指向数组元素的指针变量作函数的参数的使用方法

编写程序并调试、运行：定义一个含有 10 个整型的数据，依次给数组元素输入任意 10 个数，然后按每行 5 个数顺序输出，最后再将 10 个数按相反顺序存放，并按每行 5 个数逆序输出。要求用指向数组元素的指针变量作函数参数来完成。

测试数据：1      2      3      4      5  
             6      7      8      9      10

预期输出结果：

10      9      8      7      6  
5      4      3      2      1

实现步骤：

(1) 进入 TC 环境。

(2) 然后编辑输入如下程序。

编写逆序函数 invert()。

---

```

void invert(int *x, int n) /*函数参数为指向数组的指针变量 x 和存放数据个数的变量 n*/
{
    int *p,t,*i,*j,m=n/2; /* p 存放中间数组元素的地址，t 是临时变量，i 从第一个元素
    i=x;                  开始逐次指向下一个元素 j 从最后一个元素开始逐次指向前一个元素。m 为数据的一半*/
    j=x+n-1;
    p=x+m;
    for (; i<p; i++,j--) /*通过循环将 n 个数按相反顺序存放*/
    {
        t=*i;
        *i=*j;
        *j=t;
    }
}

```

---

---

```

    }
    return ;
}

```

---

添加主程序。在主程序中输入 10 个数，并按每行 5 个数顺序输出（黑体部分为添加的程序）。

---

```

void invert(int *x , int n)
...
#define N 10
main ()
{
    int i,a[N],*p=a ;
    printf("Please input 10 datas : \n") ;
    for (i=0 ; i<N ; i++,p++)          /*循环输入 10 个数，并按每行 5 个数顺序输出*/
    {
        scanf("%d",&a[i]) ;
        if(i%5==0)
            printf("\n") ;
        printf ("%6d",a[i]) ;
    }
    printf("\n") ;
    printf("\n") ;
}

```

---

在主程序中添加调用 invert() 的程序（黑体部分为添加的程序）。

---

```

void invert(int *x , int n)
...
#define N 10
main ()
{
    ...
    p=a ;          /*将指针变量 p 指向数组的第一个元素*/
    invert(p,N) ;   /*函数调用时指针变量 p 和 N 为实参*/
}

```

---

添加输出程序。用于按相反顺序显示数据（黑体部分为添加的程序）。

---

```

void invert(int *x , int n)
...
#define N 10

```

---





---

添加输出程序。（黑体部分为添加的程序）

---

```
main()
{
    ...
    printf("Array a : \n");          /*用户输入时的提示信息*/
    for (p=a[0]; p<a[0]+6; p++)      /*循环输出数组中各元素值，并按每行 3 个数输出*/
    {
        printf ("%3d",*p);
        if(i%3==0) printf("\n");
        i++;
    }
}
```

---

（3）运行程序。

该程序的显示结果为：

Array a :

```
1  3  5
2  4  6
```

（4）保存该文件，文件名为 tcsy7-5.c，退出 TC。

6. 熟悉指向由 m 个数组成的一维数组的指针变量的使用方法

编写程序并调试、运行：输出二维数组 $\begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$ 中任一行任一列元素的值。

实现步骤：

（1）进入 TC 环境。

（2）然后编辑输入如下程序。

编写初始化程序。即定义二维数组和指向二维数组元素的指针及指向由 3 个数组成的一维数组的指针变量，程序如下：

---

```
main()
{
    static int a[2][3]={1,3,5,2,4,6};          /*定义二维数组 a，并进行初始化*/
    int *p1, k=1;                               /* p1 存放二维数组元素的地址,k 存放输出
                                                数据的个数（初始值为 1）*/
    int (*p2)[3], i,j;                          /*p2 用来存放由 3 个数组成的一维数组的首
                                                地址，i，j 存放行号和列号*/
}
```

---

添加“输出二维数组各元素”的程序（黑体部分为添加的程序）。



（3）运行程序。

屏幕将转入运行界面，运行界面显示：

Array a：

1 3 5

2 4 6

Please input i,j：

此时，在光标处输入如下数据：

1,2

回车后，屏幕显示为：

a[1][2]=6

（4）保存该文件，文件名为 tcsy7-6.c，退出 TC。

#### 7. 熟悉指针与字符串

编写程序并调试、运行：用字符指针求字符串长度并输出该字符串。

实现步骤：

（1）进入 TC 环境。

（2）然后编辑输入如下程序：

---

```
#include "stdio.h"
main()
{
    int len=0;                                /*定义变量 len，存放字符串的长度*/
    char *p="How are you!";                  /*定义指针变量 p 用来存放字符串的首地址*/

    while(*p!='\0')                          /*通过指针变量循环读取字符串中的
    {                                          字符，如果字符不为 '\0'，则 len
        len++; p++;                          加 1，直到遇到 '\0' 为止*/
    }
    p=p-len;                                  /*指针变量 p 再指向字符串的首地址*/
    printf("The string is %s ,its length is %d.\n",p,len); /*输出字符串及长度*/
}
```

---

（3）运行程序。

该程序的显示结果为：The string is How are you! , its length is 12.

（4）保存该文件，文件名为 tcsy7-7.c，退出 TC。

#### 四、练习与思考

（1）输入3个整数，要求设3个指针变量p1、p2、p3，使p1指向3个数中的最小者，p2指向3个数中的次小者，p3指向3个数中的最大者，然后按由小到大的顺序输出3个数。

(2) 编写函数add(int \*p1, int \*p2), 函数中把指针变量p1和p2所指的存储单元中的两个值相加, 然后将和作为函数值返回。在主程序中输入两个数给变量, 把变量的地址作为实参, 传送给对应的形参。

(3) 有一个长度为100的字符串, 将最后50个字符复制到另一个字符串中。利用指向数组的指针来完成。

(4) 输入n个数存放在数组a中, 编写函数删除下标为k的元素中的值。要求利用指向数组元素的指针变量作为函数参数。

(5) 已知如下程序, 编写函数程序段, 函数 fun 的功能是将 s 所指字符串中除了下标为奇数同时 ASCII 值也为奇数的字符之外, 其余的所有字符都删除。串中剩余字符所形成的一个新串放在 t 所指的数组中。

例如: 若 s 所指字符串中的内容为: ABCDEFG12345, 其中字符 A 的 ASCII 码值虽然为奇数, 但所在元素的下标为偶数, 因此必须删除, 而字符 1 的 ASCII 码值也为奇数, 但所在数组中的下标为奇数, 因此不应删除, 其他以此类推。最后 t 所指的数组的内容应是: 135。

---

```
#include <conio.h>
#include <stdio.h>
#include <string.h>
void fun(char *s, char t[])
{/*编写程序代码*/

}

main()
{
    char    s[100], t[100];
    clrscr();
    printf("\nPlease enter string S : ");
    scanf("%s", s);
    fun(s, t);
    printf("\nThe result is : %s\n", t);
}
```

---

## 实验八 指针（二）

### 一、知识要点

#### 1. 指针数组

如果一个数组中的元素均为指针类型的数据，则称为指针数组。

指针数组的定义形式：类型标识符 \*数组名[数组长度说明];

#### 2. 指针与函数

(1) 指向函数的指针变量。

定义形式：类型标识符 (\*指针变量名)();

如：int max(), (\*p)(); p=max;

该指针变量 p 指向 max() 函数，p 存放的是 max() 函数的入口地址，在函数调用时可以通过指针变量来调用。

(2) 返回指针值的函数。

如果一个函数的返回值为指针值，则函数的类型为指针类型。

返回指针值的函数定义形式：类型标识符 \*函数名(形参表);

#### 3. 指向指针的指针

如果指针变量指向一个专门存放指针的数组，则该指针变量为指向指针的指针。

指向指针的指针定义形式：类型标识符 \*\*标识符;

### 二、上机要求

#### 1. 基本要求

- (1) 掌握指针数组的使用方法。
- (2) 掌握指向函数的指针变量的使用方法。
- (3) 掌握返回指针值的函数的使用方法。
- (4) 掌握指向指针的指针的使用方法。
- (5) 熟悉用 TC 集成环境来调试指针程序的方法。

#### 2. 较高要求

掌握指针数组作为 main 函数的形参的使用方法。

### 三、上机内容

#### 1. 熟悉指针数组

编写程序并调试、运行：利用指针数组将 5 个字符串按字母顺序（由先到后）输出。

实现步骤：

- (1) 进入 TC 环境。
- (2) 然后编辑输入如下程序。

编写初始化程序，即定义指针数组 name，用来分别存放 5 个字符串的首地址。

---

```
#include "string.h"                                /* 包含库文件 string.h */
main()
{
    static char *name[ ]={"China","Japan",          /* 定义指针数组 name，并进行初始化，分别
        "Australia","Canada","America"};          存放 5 个字符串的首地址*/
    char *temp;                                       /* 定义指针数组 temp 为临时变量*/
    int i,j,k;                                        /* i,j 循环变量。k 存放当前最小串的下标*/
}
```

---

添加由小到大的排序程序，即对 5 个字符串按字母顺序排序（黑体部分为添加的程序）。

---

```
#include "string.h"
main()
{
    ...
    for(i=0;i<4;i++)                                /* 第 i 轮比较*/
    {
        k=i;                                       /* 初始化 k 为第 i 轮第一个串的下标*/
        for(j=i+1;j<5;j++)                          /* 第 i 轮 5-i-1 次循环中 k 始终存放当前最
            if(strcmp(name[k],name[j])>0) k=j;      小串的下标*/
            if(k!=i)                                /* 第 i 轮比较出的最小串 name[i]与
                {                                    name[k]交换*/
                    temp=name[i];
                    name[i]=name[k];
                    name[k]=temp;
                }
            }
    }
}
```

---

添加输出程序，显示排好序的 5 个字符串（黑体部分为添加的程序）。

---

```
#include "string.h"
main()
{
    static char *name[ ]={"China","Japan",
        "Australia","Canada","America"};
    char *temp;
    int i,j,k;
```

---

---

```
for(i=0;i<4;i++)
{
    k=i;
    for(j=i+1;j<5;j++)
        if(strcmp(name[k],name[j])>0) k=j;
    if(k!=i)
    {
        temp=name[i];
        name[i]=name[k];
        name[k]=temp;
    }
}
printf("The sorted strings : \n");
for(i=0;i<5;i++)
    printf("%s\n",name[i]);
}
```

---

（3）运行程序。

该程序的显示结果为（通过按 Alt+F5 键查看运行屏幕）：

The sorted strings :

America

Austrailia

Canada

China

Japan

（4）保存该文件，文件名为 tcsy8-1.c，退出 TC。

2. 掌握指向函数的指针变量的使用方法

编写程序并调试、运行：用指向函数的指针变量求 a 和 b 中的大者。

测试数据：4，8

预期输出结果：8

实现步骤：

（1）进入 TC 环境。

（2）输入如下程序。

编写求两个数中的大数的函数 max ()。程序如下：

---

```
max (int x,int y)
{
    int z;
    if (x>y) z=x;
    else z=y;
```

---

---

```

    return(z);
}

```

---

添加主程序，在主程序中定义指针变量 p（存放函数 max() 的入口地址）。

---

```

max (int x,int y)
...
main( )
{
    int max (int x,int y);           /* max ()函数的声明*/
    int (*p)( );                    /* 定义一个指向函数的指针变量 p*/
    int a,b,c;
    p=max;                          /*指针变量 p 指向 max ()函数*/
    printf("Please input a,b:");     /*输入 a 和 b 两个数的提示信息*/
    scanf("%d,%d",&a,&b);           /*输入 a 和 b 两个数*/
}

```

---

在主程序中添加利用指向函数的指针变量调用 max()，并输出两个数中的大数。

---

```

max (int x,int y)
{
    int z;
    if (x>y)  z=x;
    else z=y;
    return(z);
}
main( )
{
    int max (int x,int y);
    int (*p)( );
    int a,b,c;
    p=max;
    printf("Please input a,b:");
    scanf("%d,%d",&a,&b);
    c=(*p)(a,b);                    /*利用指针变量 p 调用函数*/
    printf("a=%d,b=%d,max=%d\n",a,b,c); /*输出 a 和 b 以及这两个数中的大数 c*/
}

```

---

(3) 运行程序。

屏幕将转入运行界面，运行界面显示：

Please input a,b:

此时，在光标处输入如下两个数：



4, 8

回车后，屏幕显示为：

a=4,b=8,max=8

(4) 保存该文件，文件名为 tcsy8-2.c，退出 TC。

### 3. 熟悉返回指针值的函数

编写程序并调试、运行：打印输出星期的号码及对应的英文名字，利用返回指针值的函数来完成。

实现步骤：

(1) 进入 TC 环境。

(2) 然后编辑输入如下程序。

编写函数 week\_name()，该函数根据星期的号码得到对应的英文名字。程序如下：

---

```
char * week_name(int n)                /*参数 n 存放星期的号码*/
{
    static char *name[]={"error","Monday", /*定义指针数组 name，数组中的每一个元
        "Tuesday","Wednesday","Thursday", 素分别存放 8 个字符串的首地址*/
        "Friday","Saturday","Sunday"};
    return((n<1||n>7)?name[0]:name[n]);    /*函数的返回值为某一个字符串的首地址*/
}
```

---

添加主程序，在主程序中调用函数 week\_name()，并输出星期的号码及对应的英文名字。

---

```
char * week_name(int n)
...
main()
{
    int week;                          /*定义 week 为循环变量 */
    char *week_name(int n);            /* 函数 week_name()的声明*/
    for(week=1; week<=7; week++)       /*循环输出星期的号码及对应的
        printf("week %d : %s\n", week, week_name(week)); 英文名字*/
}
```

---

(3) 运行程序。

该程序的显示结果为：

week 1 : Monday

week 2 : Tuesday

week 3 : Wednesday

week 4 : Thursday

week 5 : Friday

week 6 : Saturday

week 7 : Sunday

(4) 保存该文件，文件名为 tcsy8-3.c，退出 TC。

#### 4. 指向指针的指针

编写程序并调试、运行：使用指向指针的指针来输出 5 个字符串，实现过程如图 8-1 所示。

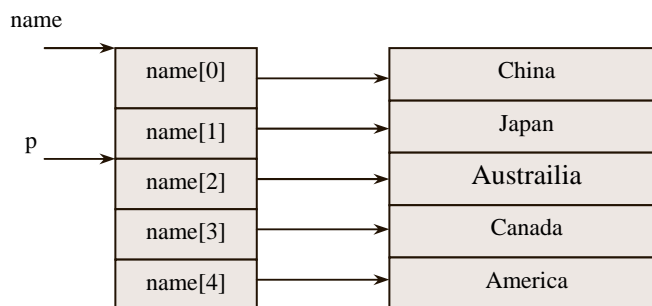


图 8-1 指针指向图

实现步骤：

(1) 进入 TC 环境。

(2) 然后编辑输入如下程序。

---

```

#include "string.h"
main()
{
    static char *name[ ]={"China","Japan",
        "Australia","Canada","America"};
    char **p;

    int i;
    printf("The five strings : \n");
    for(i=0;i<5;i++)
    {
        p=name+i;
        printf("%s\n",*p);
    }
}
  
```

/\*定义指针数组 name ,数组中的每一个元素分别存放 8 个字符串的首地址\*/

/\*定义指针变量 p , 用来存放指针数组 name 中数组元素的地址\*/

/\*利用指向指针的指针循环输出 5 个字符串\*/

---

(3) 运行程序。

该程序的显示结果为：

The five strings :

China

Japan

Australia

Canada

America

(4) 保存该文件，文件名为 tcsy8-4.c，退出 TC。

#### 四、练习与思考

(1) 利用指针数组显示下列菜单信息：

File  
Edit  
Run  
Compile  
Project  
Options  
Debug  
Break/watch

(2) 编写一个函数，实现求 n 个数的最大值。要求在主程序中从键盘输入 10 个整数，并定义一个指向函数的指针变量，通过函数指针调用该函数以找到其中的最大值并显示出来。

(3) 编写一个函数，实现将两个形参中较大数的地址作为函数值返回。在主程序中输入两个数给变量，把变量的地址作为实参，传送给对应的形参。

(4) 编写一则程序，要求输入月份号，输出该月份的英文名字。要求用返回指针值的函数处理。

(5) 用指向指针的指针方法对 n 个字符串排序并输出。

(6) 下列给定程序中，函数 fun()的功能是：将 M 行 N 列的二维数组中的数据按列的顺序依次放到一维数组中。

例如：二维数组中的数据为：

11	11	11	11
22	22	22	22
33	33	33	33

则一维数组中的内容应是：11 22 33 11 22 33 11 22 33 11 22 33。

补充实现此功能的 fun()函数。

---

```
#include <stdio.h>

void fun(int (*s)[4], int *b, int *n, int mm, int nn)
{

}

main()
{ int w[3][4]={ {33,33,33,33},{44,44,44,44},{55,55,55,55}},i,j;
  int a[12]={0}, n=0;
```

---

```
printf("The matrix:\n");
for(i=0; i<3; i++)
{
    for(j=0;j<4; j++)
        printf("%3d",w[i][j]);
    printf("\n");
}
fun(w,a,&n,3,4);
printf("The A array:\n");
for(i=0;i<n;i++)
    printf("%3d",a[i]);
printf("\n\n");
}
```

---

## 实验九 结构体和共用体

### 一、知识要点

#### 1. 结构体简介

结构体是把具有相互关系的不同类型的数据组成一个新的类型标识符。它包括结构体类型的说明，结构体类型的变量、数组和指针变量的定义，结构体变量、数组的初始化，结构体成员的引用。

其中，结构体类型说明的一般格式为：

```
struct    结构体名
{ 成员表列  };
```

结构体类型的变量、数组和指针变量的定义有 3 种方法：一是先声明结构体类型再定义变量名；二是在声明类型的同时定义变量；三是直接定义结构体类型变量。

结构体成员的引用可以采用 3 种形式：

- (1) 结构体变量名.成员名。
- (2) 指针变量名->成员名。
- (3) (\*指针变量名).成员名。

#### 2. 共用体简介

共用体又称为联合体，它使几种不同类型的变量共用一段存储空间。它包括共用体类型的说明，共用体类型变量、数组和指针变量的定义，共用体变量的赋值，共用体成员的引用。

其中，共用体类型说明的一般格式为：

```
union 共用体名
{ 成员表列  };
```

与结构体相似，共用体类型的变量、数组和指针变量的定义有 3 种方法：一是先声明共用体类型再定义变量名，二是在声明类型的同时定义变量，三是直接定义共用体类型变量。

一个共用体变量不是同时存放多个成员的值，而只能存放其中的一个值。共用体变量中起作用的是最近一次存入的成员变量的值，原有成员变量的值将被覆盖。

共用体成员的引用可以采用 3 种形式：

- (1) 共用体变量名.成员名。
- (2) 指针变量名->成员名。
- (3) (\*指针变量名).成员名。

#### 3. 利用结构体变量构成链表

链表是根据需要动态地进行存储分配的一种结构。图 9-1 为一个最简单的链表。

链表包括头指针 (head) 和结点，其中头指针是指向第一个结点的指针；结点实际上是一个结构体，它包括用户需要用的实际数据和下一个地址。

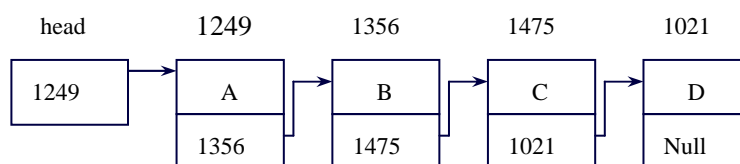


图 9-1 链表

## 二、上机要求

### 1. 基本要求

- (1) 掌握结构体类型的变量、数组和指针变量的定义和使用方法。
- (2) 掌握共用体的概念与使用方法。
- (3) 掌握用户自定义类型的特点和应用。

### 2. 较高要求

能利用结构体变量构造链表。

## 三、上机内容

### 1. 熟悉结构体类型变量的使用

编写程序并调试、运行：建立一个学生的信息，这个学生的数据包括学号、姓名及 3 门课程的成绩。要求从键盘输入这名学生的信息，并将这名学生的信息显示出来（见表 9-1）。

表 9-1 学生信息表

	学号	姓名	数学	语文	英语
测试数据	501	yufang	80	79	90
预期结果	501	yufang	80	79	90

实现步骤：

- (1) 进入 TC 环境。
- (2) 然后编辑输入如下程序。

定义一个结构体类型 student。

```

#define M 3                                /*定义符号常量 M 为 3，表示有 3 门课程*/
struct student                             /*定义一个结构体类型，含有 3 个成员*/
{
    int num;                               /*定义 num 变量，用于存放“学号”*/
    char name[20];                         /*定义 name 数组变量，用于存放“姓名”*/
    int score[M];                           /*定义 score 数组变量，存放 3 门“成绩”*/
};

```

添加“输入数据”程序（黑体部分为添加的程序）。

```

...
main()
{
    int i;                                /*定义循环变量 i*/
    struct student std;                  /*定义结构体变量 std*/
    printf("Please input num,name and %d scores:\n",M);
    scanf("%d%s",&std.num, std.name);    /*输入学生的学号、姓名*/
    for(i=0;i<M;i++)                      /*循环输入学生 3 门课的成绩*/
        scanf("%d",&std.score[i]);
}

```

添加“输出数据”程序（黑体部分为添加的程序）。

```

#define M 3
struct student
{
    int num;
    char name[20];
    int score[M];
};
main()
{
    int i;
    struct student std;
    printf("Please input num,name and %d scores:\n",M);
    scanf("%d%s",&std.num, std.name);
    for(i=0;i<M;i++)
        scanf("%d",&std.score[i]);
    printf("%6d%10s", std.num, std.name);    /*输出学号、姓名*/
    for(i=0;i<M;i++)                          /*循环输出 3 门课程的成绩*/
        printf ("%6d", std.score[i]);
    printf ("\n");
}

```

（3）运行程序。

屏幕将转入运行界面，并显示如下字符。

Please input num,name and 3 scores:

此时，在光标处输入如下数据：

501 yufang 80 79 90





---

```

char name[20];           /*定义 name 数组变量，用于存放“姓名”*/
int score[M];           /*定义 score 数组变量，存放“成绩”*/
}

```

---

添加“输入数据”程序。

---

```

#define N 5
...
main()
{
    int i,j;                /*定义循环变量 i,j*/
    struct student stud[N]; /*定义结构体数组 stud*/
    printf("Please input num,name and %d scores of %d students\n",M,N);
    for(i=0;i<N;i++)        /*循环输入 N 名学生的各成员*/
    {
        scanf("%d%s",&stud[i].num, stud[i].name); /*输入 i 名学生的学号、姓名*/
        for(j=0;j<M;j++)    /*循环输入 i 名学生的 M 门成绩*/
            scanf("%d",&stud[i].score[j]);
    }
}

```

---

添加“输出数据”程序（黑体部分为添加的程序）。

---

```

...
main()
{
    int i,j;
    struct student stud[N];
    printf("Please input num,name and %d scores of
           %d students\n",M,N);
    for(i=0;i<N;i++)
    {
        scanf("%d%s",&stud[i].num, stud[i].name);
        for(j=0;j<M;j++)
            scanf("%d",&stud[i].score[j]);
    }
    for(i=0;i<N;i++)
    {
        printf("%6d%10s", stud[i].num, stud[i].name);
        for(j=0;j<M;j++)

```

---

```

        printf ("%5d", stud[i].score[i]);
    printf ("\n");
}
}

```

/\*显示出第 i 名学生的各成绩后换行\*/

(3) 运行程序。

屏幕将转入运行界面，并显示如下字符：

Please input num,name and 3 scores of 5 students

此时，在光标处输入测试数据，输出结果为表 9-2。

表 9-2 测试数据

输入测试数据					输出结果				
501	Yufang	80	79	90	501	Yufang	80	79	90
502	Lihai	72	80	97	502	Lihai	72	80	97
503	Zhangyun	86	70	80	503	Zhangyun	86	70	80
504	Wanghong	86	74	68	504	Wanghong	86	74	68
505	xuewu	78	76	93	505	xuewu	78	76	93

(4) 保存该文件，文件名为 tcsy9-3.c，退出 TC。

#### 4. 熟悉共用体类型的使用

编写程序并调试、运行：设有若干个人的数据，其中有学生和教师。学生的数据中包括：姓名、号码、性别、职业、班级。教师的数据中包括：姓名、号码、性别、职业、职务。要求把它们放在同一表格（见表9-3）中。要求输入和输出有关学生和教师的两组数据。

表 9-3 学生、教师信息表

	name	num	sex	job	class / position
测试数据	Li	1011	f	s	501
	Wang	2085	m	t	prof

实现步骤：

(1) 进入 TC 环境。

(2) 然后编辑输入如下程序。

定义一个共用体类型：由于学生数据的 class(班级)和教师数据的 position(职务)类型不同，但在同一表格中，故可使用“共用体”数据结构。程序如下：

```

#include<stdio.h>
struct
{
    int num;
    char name[10];
    char sex;
}

```

/\*定义一个结构体类型\*/

/\*号码\*/

/\*姓名\*/

/\*性别\*/

---

```

char job;                                /*职业*/
union                                    /*定义一个共用体*/
{
    int class;                            /* 班级 */
    char position[10];                    /* 职务 */
}category;
}person[2];                              /* 2 个人员数据 */

```

---

编写“输入数据”的程序。

---

```

...
main()
{
    int n,i;
    char space;                           /*用于输入时数据之间的间隔(空格)*/

    printf("Input No.name sex job class/position:\n");
    for(i=0;i<2;i++)                      /*循环输入 2 个人员的数据 */
    {
        scanf("%d%c%s%c%c%c%c%c", &person[i].num, /*输入 i 名人员的号码、姓名、性
            &space, person[i].name,                  别、职业*/
            &space, &person[i].sex,
            &space, &person[i].job, &space);
        if (person[i].job=='s')               /*如果职业为's', 输入 i 名人员的
            scanf("%d", &person[i].category.class);   班级*/
        else if (person[i].job=='t')          /*如果职业为't', 输入 i 名人员的
            scanf("%s", person[i].category.position);  职务*/
        else printf("input error!");
    }
    printf("\n");
}

```

---

编写“输出数据”的程序（黑体部分为添加的程序）。

---

```

...
main()
{
    ...
    printf("No.   name       sex job class/position\n");
    for (i=0;i<2;i++)                /*循环输出 2 个人员的数据*/
    {

```

---

---

```

    if (person[i].job=='s')
        printf("%-6d%-10s%-5c%-6c%-6d\n",
                person[i].num,
                person[i].name,
                person[i].sex,
                person[i].job,
                person[i].category.class);
    else
        printf("%-6d%-10s%-5c%-6c%-6s\n",
                person[i].num,
                person[i].name,
                person[i].sex,
                person[i].job,
                person[i].category.position);
}
}

```

---

(3) 运行程序。

屏幕将转入运行界面，运行界面显示：

Input No.name sex job class/position:

此时，在光标处输入如下测试数据(数据之间用空格分隔)：

1011 Li F s 501

2085 Wang M t prof

回车后，程序的显示结果为：

No. name sex job class/position

1011 Li F s 501

2085 Wang M t prof

(4) 保存该文件，文件名为 tcsy9-4.c，退出 TC。

5. 利用结构体实现链表

编写程序并调试、运行：建立一个简单链表，它由 3 名学生数据（包括学号和成绩）的结点组成。输出各结点中的数据。

实现步骤：

(1) 进入 TC 环境。

(2) 然后编辑输入如下程序。

定义一个结构体类型 student。

---

```

#include "stdio.h"
struct student
{
    long num;

```

---

/\*包含库文件 stdio.h \*/

/\*定义一个结构体类型，含有 3 个成员\*/

/\*定义 num 变量，用于存放“学号”\*/

---

```

float score;                                /*定义 score 变量，用于存放“成绩”*/
struct student *next;                       /*定义指针变量 next，指向下一结点*/
};

```

---

添加“建立 3 个结点的链表”程序（黑体部分为添加的程序）

---

```

...
main( )
{
    struct student a,b,c,*head,*p;           /* a , b , c 为 3 个结点，head 为表头指针，
                                                p 为指向结点的指针*/
    a.num=99101;a.score=89.5;                 /*分别为 a , b , c 3 个结点的学号和成绩赋
    b.num=99103;b.score=90;                   初值*/
    c.num=99107;c.score=85;
    head=&a;                                   /*表头指针 head 指向 a 结点*/
    a.next=&b;                                 /*a 结点中的 next 指向 b 结点*/
    b.next=&c;                                 /*b 结点中的 next 指向 c 结点*/
    c.next=NULL;                             /* c 结点中的 next 为空，不指向任何结点*/
}

```

---

添加“输出各结点中的数据”的程序（黑体部分为添加的程序）

---

```

...
main( )
{
    ...
    p=head;                                  /*p 首先指向头结点*/
    printf(" No.   Score\n");
    do                                           /*循环输出 3 个结点的学号和成绩*/
    {
        printf("%ld%6.1f\n",p->num,p->score);
        p=p->next;                             /*p 指向下一个结点*/
    }
    while(p!=NULL);                             /*直到 p 所指向的下一个结点的 next 为
}                                                  空为止*/

```

---

(3) 运行程序。

该程序的显示结果为：

```

No.      Score
99101    89.5
99103    90.0
99107    85.0

```

(4) 保存该文件, 文件名为 tcsy9-5.c, 退出 TC。

#### 四、练习与思考

(1) 建立 10 名学生的信息表, 每个学生的数据包括学号、姓名及 3 门课程的成绩。要求从键盘输入这 10 名学生的信息, 并按照每一行显示一名学生信息的形式将 10 名学生的信息显示出来。

(2) 假设一名学生的信息表中包括学号、姓名、性别和一门课程的成绩。而成绩通常又可采用两种表示方法: 一种是五分制, 采用的是整型形式; 另一种是百分制, 采用的是浮点数形式。现要求编写一段程序, 输入一个学生的信息并显示出来。

(3) 建立一个链表, 每个结点包含: 工作证号(三位数)、姓名、职称(Prof、AssiProf、Lecture)和年龄。输入一个年龄, 如果链表中的结点所包含的年龄等于此年龄, 则将此结点删去。

# 实验十 文 件

## 一、知识要点

### 1. 文件

文件是指记录在外部介质上的数据的集合，以文件名作为访问文件的标识。文件所依附的介质一般是磁盘、磁带、光盘等。

文件的分类很多，按文件所依附的介质来分，有磁盘文件、磁带文件、内存文件、设备文件等。

按文件的内容区分，有源程序文件、目标文件、数据文件等。

按文件中的数据组织形式来分，数据文件可分为 ASCII 码文件和二进制文件。ASCII 码文件又称为文本文件 ( text )，其中每一个字节存放一个 ASCII 码。二进制文件把内存中的数据按其在内存中的存储形式存放在磁盘中。

按照操作系统对磁盘文件的读写方式，文件可以分为“缓冲文件系统”和“非缓冲文件系统”。缓冲文件系统：操作系统在内存中为每一个正在使用的文件开辟一个读写缓冲区。非缓冲文件系统：操作系统不开辟读写缓冲区。

### 2. 文件类型指针

要调用一个文件，需要有以下信息：文件当前的读写位置、与该文件对应的内存缓冲区的地址、缓冲区中未被处理的字符串、文件操作方式等。缓冲文件系统会为每一个文件系统开辟一个“文件信息区”，在 `stdio.h` 中，它被定义为 `FILE` 类型数据。

定义文件指针变量的一般形式为：

`FILE *文件结构指针变量名；`

例如：`FILE *fp；`

注意：只有通过文件指针，才能调用相应的文件。

### 3. 文件的打开与关闭及各种文件函数。

#### (1) 文件的打开 ( `fopen` 函数 )

文件操作的过程：对磁盘文件的操作顺序必须是：先打开，后读写，最后关闭。

打开文件的含义是以某种方式从磁盘中查找指定的文件或创建一个新文件。

`fopen` 函数的调用方式：`fopen ( 文件名, 使用方式 )；`

其中：文件名可以包含逻辑驱动器、路径、文件名、扩展名。使用方式 ( 见表 10-1 )。

表 10-1 文件打开方式及含义

文件打开方式	含义
"r" ( 只读 )	为输入打开一个文本文件
"w" ( 只写 )	为输出打开一个文本文件
"a" ( 追加 )	为追加打开一个文本文件

续表

文件打开方式	含义
"rb" (只读)	为输入打开一个二进制文件
"wb" (只写)	为输出打开一个二进制文件
"ab" (追加)	为追加打开一个二进制文件
"r+" (读写)	为读 / 写打开一个文本文件
"w+" (读写)	为读 / 写创建一个文本文件
"a+" (读写)	为读 / 写打开一个文本文件

例如：fp=fopen ("file1","r");

如果成功打开，返回一个指向被打开文件的文件信息区的起始地址；如果打开失败，返回一个 NULL 指针。

## (2) 文件的关闭 (fclose 函数)

关闭文件的作用是：使文件指针与文件脱离，同时刷新文件输入 / 输出缓冲区。

关闭文件的方法是：fclose (fp);

## (3) 读写函数。

fputc, fgetc 函数和 fputs, fgets 函数。

fputc 函数的调用方式：fputc (c, fp);

说明：把字符 c 写入文件 fp，成功时返回字符 c 的 ASCII 码，失败时返回 EOF。

fgetc 函数的调用方式：c=fgetc (fp);

说明：从文件 fp 中读一个字符，返回读得的字符。对于文本文件，遇文件尾时返回 EOF。

对于二进制文件，用 feof (fp) 判断是否遇到文件尾。feof (fp) = 1 说明遇到文件尾。

fputs 函数的调用方式：fputs (str, fp);

说明：把 str 写入 fp。

fgets 函数的调用方式：fgets (str, int n, fp);

说明：从文件 fp 中读取 n-1 个字节到 str，在 str 最后一个字节上加 '\0'。

fread 函数和 fwrite 函数。

fread 函数的调用方式：fread (buffer, size, count, fp);

说明：从文件 fp 中读入 count 次信息，每次读 size 字节，读入的信息存在 buffer 指针指向的缓冲区。函数返回值等于实际读入的次数（可能少于 count）。

fwrite 函数的调用方式：fwrite (buffer, size, count, fp);

说明：将以 buffer 地址开始的信息，写入 count 次，每次写 size 字节至文件 fp 中。函数返回值等于实际写入的次数（可能少于 count）。

fprintf 函数和 fscanf 函数。

fprintf 函数的调用方式：fprintf (文件指针，格式控制，变量列表);

fscanf 函数的调用方式：fscanf (文件指针，格式控制，变量地址列表);

除增加了“文件指针”参数外，fprintf 和 fscanf 函数与 printf 和 scanf 函数用法相同。



## 二、上机要求

### 1. 基本要求

- (1) 掌握文件和文件指针的概念。
- (2) 掌握缓冲文件系统中，文件的打开（fopen）与关闭（fclose）的使用方法。
- (3) 掌握缓冲文件系统中，利用 fputc 和 fgetc 函数编写对文件数据的读写方法。
- (4) 掌握缓冲文件系统中，利用 fputs 和 fgets 函数编写对文件数据的读写方法。
- (5) 掌握缓冲文件系统中，利用读文件函数（fread）和写文件函数（fwrite）编写对文件数据的读写方法。
- (6) 熟悉 TC 集成环境的调试文件程序的方法。

### 2. 较高要求

掌握非缓冲文件系统中文件的打开、关闭、读、写以及文件的定位等函数的使用。

## 三、上机内容

### 1. 熟悉 fputc 的使用

编写程序并上机调试：从键盘输入一些字符，逐个把它们送入磁盘文件 test.txt，并在屏幕上显示出来，直到从键盘输入#为止。

实现步骤：

- (1) 进入 TC 环境。
- (2) 输入如下程序。

编写“打开文件”和“关闭文件”语句。

---

```
#include "stdio.h"
main ( )
{
    FILE *fp ;                /* 定义文件指针 */
    fp=fopen ( "test.txt","w" ) ; /* 以“写”方式打开文本文件 test.txt */
    fclose ( fp );            /* 关闭文件 */
}
```

---

在程序中添加“输入字符”语句和“写文件”语句（黑体部分为新添加的语句）。

---

```
#include "stdio.h"
main ( )
{
    FILE *fp ;
    char ch ;                /*定义字符变量，用于存放读入的字符*/
    fp=fopen ( "test.txt","w" ) ;
    ch=getchar ( ) ;          /*从键盘读入字符*/
    while ( ch!='#')          /*循环语句，判断输入字符 ch 是否为“#”
    {
```

---

---

```

    putchar ( ch ) ;          /*将读入的字符显示在屏幕上，以便对比*/
    fputc ( ch,fp ) ;        /*将每一个输入的字符 ch 写入文件中*/
    ch=getchar ( ) ;
}
putchar ( '\n' ) ;
fclose ( fp ) ;
}

```

---

完善部分程序，增强程序的出错处理能力（黑体部分为新添加或修改的语句）。

---

```

#include "stdio.h"
main ( )
{
    FILE *fp ;
    char ch ;
    if ( ( fp=fopen ( filename,"w" ) ) ==NULL )          /* 判断打开文件操作是否成功，
    {                                                    如果不成功，则显示失败信息，同
        printf ( "Cannot open file %s\n",filename ) ;    时退出程序的执行*/
        exit ( 0 ) ;
    }
    ...

```

---

改写和添加部分语句，增强程序的灵活性（黑体部分为新添加或修改的语句）。

---

```

#include "stdio.h"
main ( )
{
    FILE *fp ;
    char ch ;
    char filename[10] ;          /* 定义变量，用来存放文件名*/
    printf ( "Input filename\n" ) ;
    scanf ( "%s", filename ) ;          /*从键盘输入文件名，并存放在
                                         filename 变量中 */

    if ( ( fp=fopen ( filename,"w" ) ) ==NULL )
    {
        printf ( "Cannot open file %s\n",filename ) ;
        exit ( 0 ) ;
    }
    printf ( "Input a string\n" ) ;
    ch=getchar ( ) ;
    while ( ch!='#' )

```

---

---

```

    {
        putchar ( ch ) ;
        fputc ( ch,fp ) ;
        ch=getchar ( ) ;
    }
    putchar ( '\n' ) ;

    fclose ( fp ) ;
}

```

---

(3) 运行程序。

屏幕将转入运行界面，此时，在光标处输入文件名 test.txt，然后回车，屏幕显示为

Input filename

test.txt

Input a string

再在光标处输入如下一串字符，直到输入#为止：

This is a c program.#

程序的显示结果为：

This is a c program.

(4) 保存该文件，文件名为 tcsy10-1.c，退出 TC。

(5) 查看结果。查看当前文件夹内是否生成了 test.txt 文件，并用写字板打开该文件，检查文件内容是否与自己从键盘输入的内容一致。

2. 熟悉 fgetc 的使用

编写程序并上机调试：统计磁盘文件 test.txt 中的字符个数。

实现步骤：

(1) 进入 TC 环境。

(2) 输入如下程序。

编写“打开文件”和“关闭文件”语句。

---

```

main ( )
{
    FILE *fp ;                               /* 定义文件指针 */
    int count=0 ;
    char filename[10] ;
    printf ( "Input filename\n" );
    scanf ( "%s", filename );
    if (( fp=fopen ( filename,"r" )) ==NULL ) /* 以“读”方式打开文本文件
    {                                           filename */
        printf ( "Cannot open file %s\n",filename );
        exit ( 0 );
    }
}

```

---

---

```

    }
    fclose ( fp );          /* 关闭文件 */
}

```

---

在程序中添加 “ 从 test.txt 文件中逐个读取字符，统计字符个数 ” 的语句（黑体部分为新添加的语句）。

---

```

#include "stdio.h"
main ( )
{
    FILE *fp ;
    int count=0 ;          /*定义变量 count，存放字符个数*/
    char filename[10] ;
    printf ( "Input filename\n" );
    scanf ( "%s", filename );
    if (( fp=fopen ( filename,"r" )) ==NULL )
    {
        printf ( "Cannot open file %s\n",filename );
        exit ( 0 );
    }
    while ( fgetc ( fp ) !=EOF )          /*循环从文件中读取字符,如果读取字
        count++ ;                        符不为 EOF,就使 count 加 1,直到遇
                                          文件尾 ( EOF ) 结束*/
    fclose ( fp );
}

```

---

在程序中添加输出语句，显示统计出的字符个数（黑体部分为新添加的语句）。

---

```

#include "stdio.h"
main ( )
{
    FILE *fp ;
    int count=0 ;
    char filename[10] ;
    printf ( "Input filename\n" );
    scanf ( "%s", filename );
    if (( fp=fopen ( filename,"r" )) ==NULL )
    {
        printf ( "Cannot open file %s\n",filename );
        exit ( 0 );
    }
    while ( fgetc ( fp ) !=EOF )

```

---

```

count++;
printf ( "It contains %d characters.\n",count );      /*输出统计出的字符个数*/
fclose ( fp );
}

```

(3) 运行程序。

屏幕将转入运行界面，运行界面显示：

Input filename

此时，在光标处输入如下文件名：

test.txt

回车后，程序的显示结果为：

It contains 20 characters.

(4) 保存该文件，文件名为 tcsy10-2.c，退出 TC。

3. 熟悉写文件函数 (fwrite) 的使用

编写程序并上机调试：建立 5 名学生信息表，已知每个学生的信息包括：姓名、学号、年龄、性别及 5 门课程的成绩。要求：从键盘上输入 5 名学生的信息；把学生信息输出到 sdata.dat 磁盘文件中。

设 5 名学生的姓名、学号、年龄、性别及 5 门课程的成绩如表 10-2 所示。

表 10-2 学生成绩表

姓名	学号	年龄	性别	数学	语文	英语	物理	化学
yufang	201	16	t	80	79	90	78	95
jiaoping	202	15	t	93	81	93	83	98
zhaojie	203	16	t	67	93	95	78	77
liqiang	204	16	f	72	80	72	66	84
Zhanghai	205	17	f	83	74	86	69	75

实现步骤：

(1) 进入 TC 环境。

(2) 输入如下程序。

定义一个结构体类型 student 以及结构体数组 st[N]。

```

#include "stdio.h"
#define N 5                                /*N 中存放学生数*/
#define M 5                                /*M 中存放科目数*/
struct student
{
    char name[20];                          /*姓名*/
    int num;                                /*学号*/
    int age;                                /*年龄*/
    char sex;                                /*性别*/

```

---

```

    int score[M] ;                                /*5 门课程的成绩*/
};
struct student st[N] ;                            /*定义结构体数组 st[] */

```

---

在程序中添加“输入学生信息”的程序（黑体部分为添加的程序）。

---

```

#include "stdio.h"
#define N 5
#define M 5
struct student
{
    char name[20] ;
    int num ;
    int age ;
    char sex ;
    int score[M] ;
};
struct student st[N] ;
main ( )
{
    FILE *fp ;                                    /*定义文件指针*/
    int i,j ;
    printf( "input Name Num Age Sex Math Chi Eng Phy Che:\n" ) ;
    for ( i=0 ; i<N ; i++ )                      /*循环输入学生的
                                                    姓名、学号、年龄、
                                                    性别及 5 门课程的
                                                    成绩*/
    {
        scanf ( "%s%d",st[i].name,&st[i].num ) ;
        scanf ( "%d%c",&st[i].age,&st[i].sex ) ;
        for ( j=0 ; j<M ; j++ )
            scanf ( "%d",&st[i].score[j] ) ;
    }
}

```

---

在程序中添加“打开文件”和“关闭文件”的语句（黑体部分为添加的程序）。

---

```

#include "stdio.h"
...
if(( fp=fopen( "sdata.dat","wb" ))==NULL ) /*以“写”方式打开文件*/
{
    printf ( "Cannot open file sdata.dat" ) ;
    exit ( 0 ) ;
}
else

```

---

}

```
#include "stdio.h"
```

```
#define M 5
```

 $\{$ 

```
int num ;
```

```
char sex ;
```

$$\} ;$$

```
main ( )
```

 $\{$ 

```
int i,j ;
```

```
for ( i=0 ; i<N ; i++ )
```

 $\{$ 

```
scanf ( "%d%c",&st[i].age,&st[i].sex );
```

scanf ( "%d",&amp;st[i].score[j] );

}

 $\{$ 

```
exit ( 0 );
```

}

 $\{$ 

```
/*循环输出5名学生信  
息到 sdata.dat 磁盘文  
件中*/
```

---

```

        fclose ( fp );
    }
}

```

---

(3) 运行程序。

屏幕将转入运行界面，此时，在光标处输入 5 名学生的信息如下：

yufang	201	16t	80	79	90	78	95
jiaoping	202	15t	93	81	93	83	98
zhaojie	203	16t	67	93	95	78	77
liqiang	204	16f	72	80	72	66	84
Zhanghai	205	17f	83	74	86	69	75

(4) 保存该文件，文件名为 tcsy10-3.c，退出 TC。

(5) 查看结果。查看当前文件夹内是否生成了 sdata.dat 文件，并用写字板打开该文件，检查文件内容是否与自己从键盘输入的内容一致。

4. 熟悉读文件函数 (fread) 的使用

编写程序并上机调试：为验证上题中输出到 sdata.dat 磁盘文件中的内容是否正确，将该文件中的数据按顺序读入到程序中，并显示出来。

实现步骤：

(1) 进入 TC 环境。

(2) 输入如下程序。

在程序中添加读入程序，即从 sdata.dat 文件中顺序读入数据到结构体数组中的程序 (黑体部分为添加的程序)。

---

```

#include "stdio.h"
#define N 5
#define M 5
struct student
{
    char name[20];
    int num;
    int age;
    char sex;
    int score[M];
};
struct student st[N];
main ( )
{
    FILE *fp;
    int i,j;

```

---



---

```

if ( ( fp=fopen ( "sdata.dat","rb" ) ) ==NULL )
{
    printf ( "Cannot open file sdata.dat" ) ;
    exit ( 0 ) ;
}
else
{
    for ( i=0 ; i<N ; i++ )                /*从 sdata.dat 文件中顺序读入
        fread ( &st[i],sizeof ( struct student ) ,1,fp ) ;    数据到结构体数组中*/
    fclose ( fp ) ;
}
}

```

---

在程序中添加“ 输出 5 名学生信息至显示屏幕上 ”的程序 ( 黑体部分为添加的程序 )。

---

```

#include "stdio.h"
...
main ( )
{
    FILE *fp ;
    int i,j ;
    if (( fp=fopen ( "sdata.dat","rb" )) ==NULL )
    {
        printf ( "Cannot open file sdata.dat" ) ;
        exit ( 0 ) ;
    }
    else
    {
        for ( i=0 ; i<N ; i++ )
            fread ( &st[i],sizeof ( struct student ) ,1,fp ) ;
        fclose ( fp ) ;
        printf( "Name Num Age Sex Math Chi Eng Phy Che\n" ) ;
        for ( i=0 ; i<N ; i++ )                /*循环输出 5 名学生信息
        {                                          至显示屏幕上*/
            printf ( "%s%5d",st[i].name, st[i].num ) ;
            printf ( "%3d%3c",st[i].age, st[i].sex ) ;
            for ( j=0 ; j<M ; j++ )
                printf ( "%4d", st[i].score[j] ) ;
            printf ( "\n" ) ;
        }
    }
}

```

---

---

```
}  
}
```

---

(3) 运行程序。

该程序的显示结果为：

Name	Num	Age	Sex	Math	Chi	Eng	Phy	Che
yufang	201	16	t	80	79	90	78	95
jiaoping	202	15	t	93	81	93	83	98
zhaojie	203	16	t	67	93	95	78	77
liqiang	204	16	f	72	80	72	66	84
Zhanghai	205	17	f	83	74	86	69	75

(4) 保存该文件，文件名为 tcsy10-4.c，退出 TC。

#### 四、练习与思考

(1) 文件打开和关闭的含义是什么？为什么要打开和关闭文件？

(2) 编写程序，将一个磁盘文件的内容复制到另一个磁盘文件。

(3) 从键盘上输入 100 个浮点数，并存入文件 fdata.dat 中，然后将写入文件 fdata.dat 中的浮点数读入到程序中，并显示出来。

(4) 编写程序，将两个文件中的内容合并到一个文件中并显示出来。

(5) 从键盘输入一个字符串，把它输出到磁盘文件 file1.dat 中，从磁盘文件 file1.dat 中读入一行字符到内存中，将其中的小写字母全部改成大写字母，然后输出到磁盘文件 file2.dat 中。

# 实验十一 综合程序设计

## 一、上机要求

能够综合运用前面所学的基本知识和所掌握的编程技能来解决一些实际问题。要求独立思考和编写程序，独立运行和调试程序。

## 二、上机内容

### 1. 编写四则运算自测程序并上机调试

编写一个小学生四则运算自测考试程序，该程序所完成的功能如下。

(1) 先出题；然后解答，如果解答正确，则显示“right”，如果解答不正确，则显示“wrong”。

(2) 该程序还可以预先设定有 N 道题，并且每题 10 分，做对 1 题加 10 分，满分为 N\*10。最后输出 N 道题的分数。

实现步骤：

(1) 进入 TC 环境。

(2) 输入如下程序。

编写初始化程序。

---

```
#include "stdio.h"
#define N 5                                /*共有 5 道四则运算题 */
main ( )
{
    int i=1,j=0;                            /*变量 i 用于统计出题数，
                                           j 用于统计做题正确的个数 */
    float x,y,z,result;                    /* x,y 为 两个数，z 为小学生给出的
                                           答案，result 为计算机算出的结果*/
    char ch;                               /*ch 为运算符 */
    printf ( "Notice : \n" );
    printf ( "Everyone is 10 scores.\n" );
    printf ( "The total score is %d \n",N*10 );
}
```

---

添加主体程序，即输入算术运算公式，并判断得数是否正确(黑体部分为添加的程序)。

---

```
#include "stdio.h"
...
while ( i<=N)                            /*循环输入算术式，并解答*/
{
```

---

---

```
printf ( "please input x,ch,y : \n" );

scanf ( "%f%c%f=%f",&x,&ch,&y,&z );    /*输入算术运算公式*/
switch ( ch )                          /*根据 ch 的内容判断所需要进行的
{                                       运算，如果运算符为 ' + '，则将变量
    case '+' : i++;                    i ( 存放出题个数 ) 加 1，并计算两个
        result=x+y;                    数的和，将结果赋给 result */
        if ( z==result )               /*如果 x 小学生给出的答案 与 result
        {                               相同，则显示 right，并将变量 j 加 1
            printf ( "right\n" );       ( 存放做对的个数 )，否则显示
            j++;                        wrong*/
        }
        else
            printf ( "wrong\n" );
        break;
    case '-' : i++;                    /*其他三种运算同加法运算*/
        result=x-y;
        if ( z==result )
        {
            printf ( "right\n" );
            j++;
        }
        else
            printf ( "wrong\n" );
        break;
    case '*' : i++;
        result=x*y;
        if ( z==result )
        {
            printf ( "right\n" );
            j++;
        }
        else
            printf ( "wrong\n" );
        break;
    case '/' : i++;
        result=x/y;
        if ( z==result )
        {
```

---

---

```

        printf ( "right\n" );
        j++;
    }
    else
        printf ( "wrong\n" );
        break;
    default : printf ( "Input data is error!\n" ); /*除了上述 4 种情况外，如果输入数
        break;                                  据有错，则输出错误信息*/
    }
}
}

```

---

在程序中添加输出程序，显示所得分数（黑体部分为添加的程序）。

---

```

#include "stdio.h"
...
while ( i<=N )
{
    ...
}
printf ( "Your score is %d\n", j*10 );          /*输出 N 道题所得分数*/
}

```

---

（3）运行程序。

屏幕将转入运行界面，运行界面显示：

Notice：

Everyone is 10 scores.

The total score is 50

please input x,ch,y：

此时，在光标处输入如下数据，然后回车。

3+2=5

接着显示的结果为：

right

please input x,ch,y：

再在光标处输入如下数据，然后回车。

15-6=11

接着显示的结果为：

wrong

please input x,ch,y：

再在光标处输入如下数据，然后回车。

5 × 9=45

接着显示的结果为：

right

please input x,ch,y：

再在光标处输入如下数据，然后回车。

5/4=1.25

接着显示的结果为：

right

please input x,ch,y：

再在光标处输入如下数据，然后回车。

15#6=21

接着显示的结果为：

Input data is error!

please input x,ch,y：

再在光标处输入如下数据，然后回车。

15+6=21

接着显示的结果为：

right

Your score is 40

(4) 保存该文件，文件名为 tcsy11-1.c，退出 TC。

## 2. 编写学生信息程序并上机调试

有 5 名学生，每名学生包括学号、姓名和 3 门课程的成绩。该程序要求在 main 函数中输入 5 名学生的学号、姓名和 3 门课程的成绩，然后在以下 4 种功能中选择所需要完成的功能。4 种功能如下。

(1) 在函数 print 中打印输出 5 名学生的学号、姓名和 3 门课程的成绩。

(2) 用户在输入学生序号（从 0 开始）后，能输出该学生的学号、姓名和 3 门课程的成绩，用函数 number 来实现。

(3) 用函数 fail 找出不及格成绩的学生的学号、姓名和 3 门课程的成绩。

(4) 用函数 maxu 找出成绩最高者的学号、姓名和 3 门课程的成绩。

实现步骤：

(1) 进入 TC 环境。

(2) 输入如下程序。

编写初始化程序。即定义符号常量和结构体类型，以及 4 个函数的声明。

---

```
#include "stdio.h"
#define N 5                                /*N 名学生*/
#define M 3                                /*M 门课程*/
#define format1 "%6d%10s"                /*将输出格式定义为符号常量*/
struct student                             /*定义结构体类型*/
```

---

---

```

{
    int num;                                /*学号*/
    char name[20];                          /*姓名*/
    int score[M];                          /*M 门课程的成绩*/
};
void print ( struct student *p );          /*print 函数原型声明*/
void number ( struct student *pointer, int n ); /*number 函数原型声明*/
void fail ( struct student *pointer );     /*fail 函数原型声明*/
void maxu ( struct student *pointer );     /*maxu 函数原型声明*/

```

---

编写主函数，在 main 函数中输入 5 名学生的学号、姓名和 3 门课程的成绩，并在主函数中选择调用 4 个函数。

---

```

#include "stdio.h"
...
void main ( )
{
    int i,j,a,snum;                        /*i,j 是循环变量 ,a 是功能号 ,snum
                                           是学生序号*/
                                           /*定义结构体数组*/

    struct student stu[N];
    printf ( "Please input num,name and %d scores of %d students\n",M,N );
    for ( i=0;i<N;i++)
    {
        scanf ( "%d%s",&stu[i].num,stu[i].name );
        for ( j=0;j<M;j++)
            scanf ( "%d",&stu[i].score[j] );    /*输入 5 名学生的学号、姓名
                                                    和 3 门课程的成绩*/
    }
    printf ( "-----\n" );
    printf ( "1-----print\n" );
    printf ( "2-----search by number\n" );
    printf ( "3-----search for failer \n" );
    printf ( "4-----search for maximal scores\n" );
    printf ( "0-----exit\n" );
    printf ( "-----\n" );
    while ( 1 )
    {
        printf ( "Please select function number : " );
        scanf ( "%d",&a );                    /*输入功能号*/
        switch ( a )                          /*根据功能号进行如下选择*/

```

---

---

```

{
    case 0 : exit ( 0 ) ;
               break;
    case 1 : print ( &stu[0] ) ;
               break;
    case 2 : printf( "\nInput a student\'s number : " );
               scanf ( "%d",&snum ) ;
               number ( &stu[0],snum ) ;
               break;
    case 3 : fail ( &stu[0] ) ;
               break;
    case 4 : maxu ( &stu[0] ) ;
               break;
}
}
}

```

---

编写 print 函数。

---

```

void print ( struct student *p )
{
    int i,j;
    struct student *pt;
    printf ( "\n num,name and %d  scores of %d students : \n",M,N );
    for ( i=0;i<N;i++ )
    {
        pt = p+i ;
        printf ( format1,pt->num, pt->name ) ;
        for ( j=0;j<M;j++ )
            printf ( "%5d",pt->score[j] ) ;
        printf ( "\n" ) ;
    }
    printf ( "\n" ) ;
    return ;
}

```

---



## 编写 number 函数。

---

<pre> void number ( struct student *pointer, int n ) {     int i,j;     struct student *pt;     pt = pointer+n;     printf ( "\n num,name and %d  scores of the %dth student : \n",M,n+1 ) ;     printf ( format1,pt-&gt;num, pt-&gt;name ) ;     for ( j=0;j&lt;M;j++ )         printf ( "%5d",pt-&gt;score[j] ) ;     printf ( "\n" ) ;     printf ( "\n" ) ;     return ; } </pre>	<pre> /*定义 number 函 数，参数为指向结 构体的指针和学生 序号 */ /*pt 指向序号为 n 的学生*/ /*通过指向结构体 的指针输出序号为 n 的学生的学号、 姓名和 M 门课程 的成绩*/ </pre>
---	--

---

## 编写 fail 函数。

---

<pre> void fail ( struct student *pointer ) {     int i,j,k,n=0;     struct student *pt;     printf ( "\n num,name and %d  scores of the fail students : \n",M ) ;     for ( i=0 ; i&lt;N ; i++ )     {         pt = pointer + i;         for ( j=0 ; j&lt;M ; j++ )             if ( pt-&gt;score[j]&lt;60 )             {                 printf ( format1,pt-&gt;num, pt-&gt;name ) ;                 for ( k=0;k&lt;M;k++ )                     printf ( "%5d",pt-&gt;score[j] ) ;                 printf ( "\n" ) ;                 n++;                 break;             }     }     if ( n==0 )         printf ( " All students pass!\n",M ) ;     printf ( "\n" ) ;     return ; } </pre>	<pre> /*定义 fail 函数，参 数为指向结构体的 指针 */ /* n 存放不及格人数 */ /*通过指向结构体 的指针循环输出不及 格学生的学号、姓名 和 M 门课程的成绩 */ /*统计不及格人数 */ /*如果不及格人数为 0，则显示全部通过 的信息 */ </pre>
---	---

---

---

}

---

编写 maxu 函数。

---

void maxu ( struct student *pointer )	/*定义 maxu 函数，参数为指向结
{	构体的指针 */
int i,j,max,temp=0;	/* max 存放成绩最高者的总成绩，
	temp 存放成绩最高者的序号 */
int sum[N];	/*数组存放每个学生的总成绩 */
struct student *pt,*p;	
for ( i=0;i<N;i++ )	/*每个学生的总成绩初始值置 0*/
sum[i]=0;	
p=pointer;	/*p 指向第一个学生*/
for ( i=0;i<N;i++ )	/*通过指向结构体的指针累加出
{	每个学生 M 门课程的成绩*/
p=p+i;	
for ( j=0;j<M;j++ )	
sum[i]=sum[i]+p->score[j];	
}	
for ( max=sum[0],i=1;i<N;i++ )	/*通过循环找出总成绩最高者及
if ( sum[i]>max )	最高者的序号并赋给 temp*/
{	
max=sum[i];	
temp=i;	
}	
pt = pointer+temp;	/*pt 指向总成绩最高的学生*/
printf ( "\n num,name and %d scores of the maximum score student : \n",M );	
printf ( format1,pt->num, pt->name );	/*输出总成绩最高学生的学号、姓
for ( j=0;j<M;j++ )	名和 M 门课程的成绩*/
printf ( "%5d",pt->score[j] );	
printf ( "\n" );	
printf ( "\n" );	
return ;	
}	

---

(3) 运行程序。

屏幕将转入运行界面，运行界面显示：

Please input num,name and 3 scores of 5 students

此时，在光标处输入5名学生的学号、姓名和3门课程的成绩，数据如下：

```
1 yufang      80 79 90
2 lihai       88 80 97
3 zhangyun    86 70 80
4 wanghong    43 74 68
5 xuewu       78 76 52
```

输出结果为：

```
-----
1-----print
2-----search by number
3-----search for failer
4-----search for maximal scores
0-----exit
-----
```

Please select function number：

在光标处输入功能号 1，然后回车，显示的结果为：

num,name and 3 scores of 5 students：

```
1      yufang      80  79  90
2      lihai       88  80  97
3      zhangyun    86  70  80
4      wanghong    43  74  68
5      xuewu       78  76  52
```

Please select function number：

在光标处输入功能号 2，然后回车，屏幕显示：

Input a student's number：

在光标处输入学生序号 0，然后回车，屏幕显示：

num,name and 3 scores of the 1th student：

```
1      yufang      80  79  90
```

Please select function number：

在光标处输入功能号 2，然后回车，屏幕显示：

Input a student's number：

在光标处输入学生序号 1，然后回车，屏幕显示：

num,name and 3 scores of the 1th student：

```
2      lihai      88  80  97
```

Please select function number：

在光标处输入功能号 3，然后回车，屏幕显示：

num,name and 3 scores of the fail students：

```
4      wanghong    43  74  68
5      xuewu       78  76  52
```

Please select function number：

在光标处输入功能号 4，然后回车，屏幕显示：

num,name and 3 scores of the maximum score student：

2        lihai    88    80    97

Please select function number：

在光标处输入功能号 0，然后回车，此时将退出输出屏幕。

(4) 保存该文件，文件名为 tcsy11-2.c，退出 TC。

### 三、练习与思考

(1) 对小学生四则运算自测考试程序进行修改,使程序所完成的功能为：先出题；然后解答；如果解答正确，则显示正确的信息；如果解答不正确，则显示错误的信息，同时还需要再重新解答，直到解答正确为止。重复上述过程。

参考程序如下：

```
#include "stdio.h"
#define N 3
main ( )
{
    int i=0;
    float x,y,z,result;
    char ch;
    printf ( "please input x,ch,y : " );
    scanf ( "%f%c%f= %f",&x,&ch,&y,&z );
    while ( 1 )
    {
        switch ( ch )
        {
            case '+' : result=x+y;
                if ( z==result )
                {
                    printf ( "right\n" );
                    i++;
                }
                else
                {
                    printf ( "wrong\n" );
                    printf ( "%f%c%f=",x,ch,y );
                    scanf ( "%f",&z );
                    continue;
                }
            break;
        }
    }
}
```

```
case '-': result=x-y;
if ( z==result )
{
    printf ( "right\n" );
    i++;
}
else
{
    printf ( "wrong\n" );
    printf ( "%f%c%f=",x,ch,y );
    scanf ( "%f",&z );
    continue;
}
break;
case '*': result=x*y;
if ( z==result )
{
    printf ( "right\n" );
    i++;
}
else
{
    printf ( "wrong\n" );
    printf ( "%f%c%f=",x,ch,y );
    scanf ( "%f",&z );
    continue;
}
break;
case '/': result=x/y;
if ( z==result )
{
    printf ( "right\n" );
    i++;
}
else
{
    printf ( "wrong\n" );
    printf ( "%f%c%f=",x,ch,y );
    scanf ( "%f",&z );
```

```
        continue;
    }
    break;
default : printf ( "Input  data is error!\n" ); break;
    }
if ( i>=N ) break;
printf ( "please input x,ch,y : " );
scanf ( "%f%c%f= %f",&x,&ch,&y,&z );
}
}
```

(2) 编写程序并上机调试：有 N 个学生，每个学生的信息包括学号、姓名和 M 门课程的成绩。该程序要求在 main 函数中输入 N 个学生的学号、姓名和 M 门课程的成绩，并需要完成如下功能。

把学生信息输出到 sdata.dat 磁盘文件中（用函数 stdout 实现）。

从 sdata.dat 磁盘文件中读取学生数据（用函数 stdin 实现）。

在函数 print 中打印输出 N 名学生的学号、姓名和 M 门课程的成绩。

用函数 average 输出每个学生的学号、姓名、M 门课程的成绩及平均成绩。

在函数 sort 中实现按成绩由高到低的顺序输出 N 名学生的学号、姓名和 M 门课程的成绩。

# 附录一 全国计算机等级考试二级 C 语言考试大纲

- 公共基础知识

## 基本要求

1. 掌握算法的基本概念。
2. 掌握基本数据结构及其操作。
3. 掌握基本排序和查找算法。
4. 掌握逐步求精的结构化程序设计方法。
5. 掌握软件工程的基本方法，具有初步应用相关技术进行软件开发的能力。
6. 掌握数据库的基本知识，了解关系数据库的设计。

## 考试内容

### 一、基本数据结构与算法

1. 算法的基本概念，算法复杂度的概念和意义（时间复杂度与空间复杂度）。
2. 数据结构的定义，数据的逻辑结构与存储结构，数据结构的图形表示，线性结构与非线性结构的概念。
3. 线性表的定义，线性表的顺序存储结构及其插入与删除运算。
4. 栈和队列的定义，栈和队列的顺序存储结构及其基本运算。
5. 线性单链表、双向链表与循环链表的结构及其基本运算。
6. 树的基本概念，二叉树的定义及其存储结构，二叉树的前序、中序和后序遍历。
7. 顺序查找与二分法查找算法，基本排序算法（交换类排序，选择类排序，插入类排序）。

### 二、程序设计基础

1. 程序设计方法与风格。
2. 结构化程序设计。
3. 面向对象的程序设计方法、对象、方法、属性及继承与多态性。

### 三、软件工程基础

1. 软件工程基本概念，软件生命周期概念，软件工具与软件开发环境。
2. 结构化分析方法，数据流图，数据字典，软件需求规格说明书。
3. 结构化设计方法，总体设计与详细设计。
4. 软件测试的方法，白盒测试与黑盒测试，测试用例设计，软件测试的实施，单元测试、集成测试和系统测试。
5. 程序的调试、静态调试与动态调试。

### 四、数据库设计基础

1. 数据库的基本概念：数据库，数据库管理系统，数据库系统。
2. 数据模型，实体联系模型及 E-R 图，从 E-R 图导出关系数据模型。
3. 关系代数运算，包括集合运算及选择、投影、连接运算，数据库规范化理论。
4. 数据库设计方法和步骤：需求分析、概念设计、逻辑设计和物理设计的相关策略。

### 考试方式

1. 公共基础知识的考试方式为笔试，与 C 语言程序设计的笔试部分合为一张试卷。公共基础知识部分占全卷的 30 分。

2. 公共基础知识有 10 道选择题和 5 道填空题。

### • C 语言程序设计

#### 基本要求

1. 熟悉 Turbo C 集成环境。
2. 熟练掌握结构化程序设计的方法，具有良好的程序设计风格。
3. 掌握程序设计中简单的数据结构和算法。
4. 在 Turbo C 的集成环境下，能够编写简单的 C 程序，并具有基本的纠错和调试程序的能力。

#### 考试内容

##### 一、C 语言的结构

1. 程序的构成，main 函数和其他函数。
2. 头文件，数据说明，函数的开始和结束标志。
3. 源程序的书写格式。
4. C 语言的风格。

##### 二、数据类型及其运算

1. C 语言的数据类型（基本类型、构造类型、指针类型、空类型）及其定义方法。
2. C 语言的运算符的种类、运算优先级和结合性。
3. 不同类型数据间的转换与运算。
4. C 语言的表达式类型（赋值表达式，算术表达式，关系表达式，逻辑表达式，条件表达式，逗号表达式）和求值规则。

##### 三、基本语句

1. 表达式语句，空语句，复合语句。
2. 数据的输入与输出，输入、输出函数的调用。
3. 复合语句。
4. go to 语句和语句标号的使用。

##### 四、选择结构程序设计

1. 用 if 语句实现选择结构。
2. 用 switch 语句实现多分支选择结构。
3. 选择结构的嵌套。

##### 五、循环结构程序设计

1. for 循环结构。
2. while 和 do-while 循环结构。
3. continue 语句和 break 语句。
4. 循环的嵌套。

##### 六、数组的定义和引用

1. 一维数组和 multidimensional 数组的定义、初始化和引用。



2. 字符串与字符数组。

#### 七、函数

1. 库函数的正确调用。

2. 函数的定义方法。

3. 函数的类型和返回值。

4. 形式参数与实在参数，参数值的传递。

5. 函数的正确调用、嵌套调用与递归调用。

6. 局部变量和全局变量。

7. 变量的存储类别（自动，静态，寄存器，外部），变量的作用域和生存期。

8. 内部函数与外部函数。

#### 八、编译预处理

1. 宏定义：不带参数的宏定义，带参数的宏定义。

2. “文件包含”处理。

#### 九、指针

1. 指针与指针变量的概念，指针与地址运算符。

2. 变量、数组、字符串、函数、结构体的指针以及指向变量、数组、字符串、函数、结构体的指针变量。通过指针引用以上各类型数据。

3. 用指针作函数参数。

4. 返回指针值的指针函数。

5. 指针数组，指向指针的指针，main 函数的命令行参数。

#### 十、结构体（即“结构”）与共用体（即“联合”）

1. 结构体和共用体类型数据的定义方法和引用方法。

2. 用指针和结构体构成链表，单向链表的建立、输出、删除与插入。

#### 十一、位运算

1. 位运算符的含义及使用。

2. 简单的位运算。

#### 十二、文件操作

只要求了解缓冲文件系统（即高级磁盘 I/O 系统），对非标准缓冲文件系统（即低级磁盘 I/O 系统）不作要求。

1. 文件类型指针（FILE 类型指针）。

2. 文件的打开与关闭（fopen，fclose）。

3. 文件的读写（fputc,fgetc,fputs,fgets,fread,frwrite,fprintf,fscanf 函数），文件的定位（rewind,fseek 函数）。

#### 考试方式

1. 笔试：120 分钟，满分 100 分，其中含公共基础知识部分的 30 分。

2. 上机：60 分钟，满分 100 分。

## 附录二 C 语言程序设计笔试测试题

### 笔试测试题一

一、选择题（1~10 每小题 2 分，11~50 每小题 1 分，共 60 分）

下列各题 A、B、C、D 四个选项中，只有一个选项是正确的。

1. 数据的存储结构是指\_\_\_\_\_。
  - A. 存储在外存中的数据
  - B. 数据所占的存储空间量
  - C. 数据在计算机中的顺序存储方式
  - D. 数据的逻辑结构在计算机中的表示
2. 下列关于栈的描述中错误的是\_\_\_\_\_。
  - A. 栈是先进后出的线性表
  - B. 栈只能顺序存储
  - C. 栈具有记忆作用
  - D. 对栈的插入与删除操作中，不需要改变栈底指针
3. 对于长度为  $n$  的线性表，在最坏情况下，下列各排序法所对应的比较次数中正确的是\_\_\_\_\_。
  - A. 冒泡排序为  $n/2$
  - B. 冒泡排序为  $n$
  - C. 快速排序为  $n$
  - D. 快速排序为  $n(n-1)/2$
4. 对长度为  $n$  的线性表进行顺序查找，在最坏情况下所需要的比较次数为\_\_\_\_\_。
  - A.  $\log_2 n$
  - B.  $n/2$
  - C.  $n$
  - D.  $n+1$
5. 下列对于线性链表的描述中正确的是\_\_\_\_\_。
  - A. 其存储空间不一定是连续的，且各元素的存储顺序是任意的
  - B. 其存储空间不一定是连续的，且前继元素一定存储在后继元素的前面
  - C. 其存储空间必须连续，且前继元素一定存储在后继元素的前面
  - D. 其存储空间必须连续，且各元素的存储顺序是任意的
6. 下列对于软件描述中正确的是\_\_\_\_\_。
  - A. 软件测试的目的是证明程序是否正确
  - B. 软件测试的目的是使程序运行结果正确
  - C. 软件测试的目的是尽可能多地发现程序中的错误
  - D. 软件测试的目的是使程序符合结构化原则
7. 为了使模块尽可能独立，要求\_\_\_\_\_。
  - A. 模块的内聚程序要尽量高，且各模块间的耦合程序要尽量强
  - B. 模块的内聚程序要尽量高，且各模块间的耦合程序要尽量弱

- C. 模块的内聚程序要尽量低, 且各模块间的耦合程序要尽量弱  
D. 模块的内聚程序要尽量低, 且各模块间的耦合程序要尽量强
8. 下列描述中正确的是\_\_\_\_\_。
- A. 程序就是软件  
B. 软件开发不受计算机系统的限制  
C. 软件既是逻辑实体, 又是物理实体  
D. 软件是程序、数据与相关文档的集合
9. 数据独立性是数据库技术的重要特点之一。所谓数据独立性是指\_\_\_\_\_。
- A. 数据与程序独立存放  
B. 不同的数据被存放在不同的文件中  
C. 不同的数据只能被对应的应用程序所使用  
D. 以上三种说法都不对
10. 用树形结构表示实体之间联系的模型是\_\_\_\_\_。
- A. 关系模型      B. 网状模型      C. 层次模型      D. 以上三个都是
11. 算法具有 5 个特性, 以下选项中不属于算法特性的是\_\_\_\_\_。
- A. 有穷性      B. 简洁性      C. 可行性      D. 确定性
12. 以下选项中可作为 C 语言合法常量的是\_\_\_\_\_。
- A. -80.      B. -080      C. -8e1.0      D. -80.0e
13. 以下叙述中正确的是\_\_\_\_\_。
- A. 用 C 程序实现的算法必须要有输入和输出操作  
B. 用 C 程序实现的算法可以没有输出但必须要有输入操作  
C. 用 C 程序实现的算法可以没有输入但必须要有输出操作  
D. 用 C 程序实现的算法可以既没有输入也没有输出操作
14. 以下不能定义为用户标识符的是\_\_\_\_\_。
- A. Main      B. \_0      C. \_int      D. sizeof
15. 以下选项中, 不能作为合法常量的是\_\_\_\_\_。
- A. 1.234e04      B. 1.234e0.4      C. 1.234e+4      D. 1.234e0
16. 数字字符 0 的 ASCII 值为 48, 若有以下程序:
- ```
main()
{
    char a='1',b='2';
    printf("%c",b++);
    printf("%d\n",b-a);
}
```
- 程序运行后的输出结果是\_\_\_\_\_。
- A. 3,2      B. 50,2      C. 2,2      D. 2,50
17. 有以下程序:
- ```
main()
{
```

```
int m=12,n=34;
printf("%d%d",m++,++n);
printf("%d%d\n",n++,++m);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 12353514      B. 12353513      C. 12343514      D. 12343513

18. 有定义语句: int b; char c[10]; ,则正确的输入语句是\_\_\_\_\_。

- A. scanf("%d%s",&b,&c);      B. scanf("%d%s",&b,c);  
C. scanf("%d%s",b,c);      D. scanf("%d%s",b,&c);

19. 有以下程序 :

```
main()
{
    int m,n,p;
    scanf("m=%dn=%dp=%d",&m,&n,&p);
    printf("%d%d%d\n",m,n,p);
}
```

若想从键盘上输入数据,使变量 m 中的值为 123,n 中的值为 456,p 中的值为 789,则正确的输入是\_\_\_\_\_。

- A. m=123n=456p=789      B. m=123 n=456 p=789  
C. m=123,n=456,p=789      D. 123    456    789

20. 有以下程序 :

```
main()
{
    int a,b,d=25;
    a=d/10%9;
    b=a&&(-1);
    printf("%d,%d\n",a,b);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 6,1      B. 2,1      C. 6,0      D. 2,0

21. 有以下程序 :

```
main()
{
    int i=1,j=2,k=3;
    if(i++==1&&(++j==3||k++==3))
        printf("%d %d %d\n",i,j,k);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 1 2 3      B. 2 3 4      C. 2 2 3      D. 2 3 3

22. 若整型变量 a、b、c、d 中的值依次为：1、4、3、2。则条件表达式  $a < b ? a : c < d ? c : d$  的值为\_\_\_\_\_。

- A. 1                  B. 2                  C. 3                  D. 4

23. 有以下程序：

```
main()
{
    int p[8]={11,12,13,14,15,16,17,18}, i=0,j=0;
    while(i++<7) if(p[i]%2)    j+=p[i];
    printf("%d\n",j);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 42                  B. 45                  C. 56                  D. 60

24. 有以下程序：

```
main()
{
    char a[7]="a0\0a0\0";int i,j;
    i=sizeof(a); j=strlen(a);
    printf("%d %d\n",i,j);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 2    2              B. 7    6              C. 7    2              D. 6    2

25. 以下能正确定义一维数组的选项是\_\_\_\_\_。

- A. `int a[5]={0,1,2,3,4,5};`                  B. `char a[]={0,1,2,3,4,5};`  
C. `char a={'A','B','C'};`                  D. `int a[5]="0123";`

26. 有以下程序：

```
int f1(int x,int y){return x>y?x:y;}
int f2(int x,int y){return x>y?y:x;}
main()
{
    int a=4,b=3,c=5,d=2,e,f,g;
    e=f2(f1(a,b),f1(c,d));
    f=f1(f2(a,b),f2(c,d));
    g=a+b+c+d-e-f;
    printf("%d,%d%d\n",e,f,g);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 4 , 3 , 7                  B. 3 , 4 , 7                  C. 5 , 2 , 7                  D. 2 , 5 , 7

27. 已有定义：`char a[]="xyz",b[]={ 'x','y','z' };,`以下叙述中正确的是\_\_\_\_\_。

- A. 数组 a 和数组 b 的长度相同                  B. a 数组的长度小于 b 数组的长度

C. a 数组的长度大于 b 数组的长度

D. 上述说法都不对

28. 有以下程序：

```
void f(int *x,int *y)
{
    int t;
    t=*x;*x=*y;*y=t;
}
main()
{
    int a[8]={1,2,3,4,5,6,7,8},i,*p,*q;
    p=a;q=&a[7];
    while(p<q)
    {
        f(p,q);
        p++;
        q--;
    }
    for(i=0;i<8;i++)
        printf("%d",a[i]);
}
```

程序运行后的输出结果是\_\_\_\_\_。

A. 8 , 2 , 3 , 4 , 5 , 6 , 7 , 1 ,

B. 5 , 6 , 7 , 8 , 1 , 2 , 3 , 4 ,

C. 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 ,

D. 8 , 7 , 6 , 5 , 4 , 3 , 2 , 1 ,

29. 有以下程序：

```
main ()
{
    int a[3][3], i, j, k=0;
    for (i=0, i<3; i++)
        for(j=0; j<3; j++)
        {
            a[i][j] = k;
            k++;
        }
    for (i=0; i<3; i++)
        printf ( "% d", a[1][i]);
}
```

程序运行后的输出结果是\_\_\_\_\_。

A. 0 1 2

B. 1 2 3

C. 2 3 4

D. 3 4 5

30. 以下叙述中错误的是\_\_\_\_\_。

- A. 对于 double 类型数组，不可以直接用数组名对数组进行整体输入或输出
- B. 数组名代表的是数组所占存储区的首地址，其值不可改变
- C. 当程序执行中，数组元素的下标超出所定义的下标范围时，系统将给出“下标越界”的出错信息
- D. 可以通过赋初值的方式确定数组元素的个数

31. 有以下程序：

```
#define N 20
fun(int a[],int n,int m)
{
    int i,j;
    for(i=m;i>=n;i--)
        a[i+1]=a[i];
}
main()
{
    int i,a[N]={1,2,3,4,5,6,7,8,9,10};
    fun(a,2,9);
    for(i=0;i<5;i++)
        printf("%d",a[i]);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 10234                      B. 12344                      C. 12334                      D. 12234

32. 有以下程序：

```
main()
{
    int a[3][2]={0},(*ptr)[2],i,j;
    for(i=0;i<2;i++)
    {
        ptr=a[i];
        scanf("%d",ptr);
        ptr++;
    }
    for(i=0;i<3;i++)
    {
        for(j=0;j<2;j++)
            printf("%2d",a[i][j]);
        printf("\n");
    }
}
```

```

    }

```

若运行时输入：1 2 3<回车>，则输出结果为\_\_\_\_\_。

- |           |        |        |        |
|-----------|--------|--------|--------|
| A. 产生错误信息 | B. 1 0 | C. 1 2 | D. 1 0 |
|           | 2 0    | 3 0    | 2 0    |
|           | 0 0    | 0 0    | 3 0    |

33. 有以下程序：

```

prt(int *m,int n)
{
    int i;
    for(i=0;i<n;i++)
        m++;
}
main()
{
    int a[]={1,2,3,4,5},i;
    prt(a,5);
    for(i=0;i<5;i++)
        printf("%d",a[i]);
}

```

程序运行后的输出结果是\_\_\_\_\_。

- |                        |                        |
|------------------------|------------------------|
| A. 1 , 2 , 3 , 4 , 5 , | B. 2 , 3 , 4 , 5 , 6 , |
| C. 3 , 4 , 5 , 6 , 7 , | D. 2 , 3 , 4 , 5 , 1 , |

34. 有以下程序：

```

main()
{
    int a[]={1,2,3,4,5,6,7,8,9,0},*p;
    for(p=a;p<a+10;p++)
        printf("%d",*p);
}

```

程序运行后的输出结果是\_\_\_\_\_。

- |   |
|---|
| A. 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 0 ,  |
| B. 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10 , 1 , |
| C. 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 ,  |
| D. 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 ,  |

35. 有以下程序：

```

#define P 3
#define F(int x) { return (P*x*x);}
main()
{

```



```
printf("%d\n",F(3+5));  
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 192                      B. 29                      C. 25                      D. 编译出错

36. 有以下程序：

```
main()  
{  
    int c=35;  
    printf("%d\n",c&c);  
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 0                      B. 70                      C. 35                      D. 1

37. 以下叙述中正确的是\_\_\_\_\_。

- A. 预处理命令行必须位于源文件的开头  
B. 在源文件的一行上可以有多条预处理命令  
C. 宏名必须用大写字母表示  
D. 宏替换不占用程序的运行时间

38. 若有以下说明和定义：

```
union dt  
{  
    int a;  
    char b;  
    double c;  
}data;
```

以下叙述中错误的是\_\_\_\_\_。

- A. data 的每个成员的起始地址都相同  
B. 变量 data 所占内存字节数与成员 c 所占字节数相等  
C. 程序段 data.a=5;printf("%f\n",data.c);输出结果为 5.000000  
D. data 可以作为函数的实参

39. 以下语句或语句组中，能正确进行字符串赋值的是\_\_\_\_\_。

- A. char \*sp; \*sp="right!";                      B. char s[10]; s="right!";  
C. char s[10]; \*s="right!";                      D. char \*sp="right!";

40. 设有如下说明：

```
typedef struct ST  
{  
    long a;  
    int b;  
    char c[2];  
} NEW;
```

则下面叙述中正确的是\_\_\_\_\_。

A. 以上的说明形式不正确

B. ST 是一个结构体类型

C. NEW 是一个结构体类型

D. NEW 是一个结构体变量

41. 有以下程序：

```
main()
{
    int a=1,b;
    for(b=1;b<=10;b++)
    {
        if(a>=8)    break;
        if(a%2==1)
            { a+=5; continue;}
        a-=3;
    }
    printf("%d\n",b);
}
```

程序运行后的输出结果是\_\_\_\_\_。

A. 3

B. 4

C. 5

D. 6

42. 有以下程序：

```
main()
{
    char s[]="159",*p;
    p=s;
    printf("%c",*p++);
    printf("%c",*p++);
}
```

程序运行后的输出结果是\_\_\_\_\_。

A. 15

B. 16

C. 12

D. 59

43. 有以下函数：

```
fun(char *a,char *b)
{
    while((*a!=\0')&&(*b!=\0')&&(*a==*b))
    {    a++;    b++;}
    return (*a-*b);
}
```

该函数的功能是\_\_\_\_\_。

A. 计算 a 和 b 所指字符串的长度之差

B. 将 b 所指的字符串复制到 a 所指的字符串中

C. 将 b 所指的字符串连接到 a 所指的字符串后面

D. 比较 a 和 b 所指的字符串的大小

44. 有以下程序：

```
main()
{
    int num[4][4]={ {1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}},i,j;
    for(i=0;i<4;i++)
    {
        for(j=1;j<=i;j++)
            printf("%c",' ');
        for(j=____; j<4;j++)
            printf("%4d",num[j]);
        printf("\n");
    }
}
```

若要按以下形式输出数组的右上三角

```
1  2  3  4
   6  7  8
      11 12
         16
```

则在程序中的横线处应填入的是\_\_\_\_\_。

A. i-1

B. i

C. i+1

D. 4-i

45. 有以下程序：

```
point(char *p)
{
    p+=3;
}
main()
{
    char b[4]={ 'a','b','c','d'},*p=b;
    point(p);
    printf("%c\n",*p);
}
```

程序运行后的输出结果是\_\_\_\_\_。

A. a

B. b

C. c

D. d

46. 程序中若有如下的说明和定义语句：

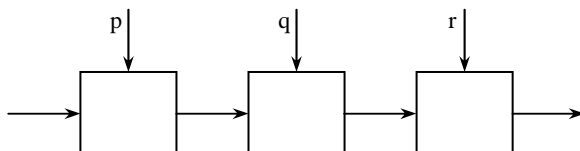
```
char fun(char *);
main()
{
    char *s="one",a[5]={0},(*f1)()=fun,ch;
```

```
...
}
```

则以下选项中对函数 fun 的正确调用语句是\_\_\_\_\_。

- A. (\*f1)(a);                      B. \*f1(\*s);                      C. fun(&a);                      D. ch=\*f1(s)

47. 有以下结构体说明和变量定义,如图所示,指针 p、q、r 分别指向此链表中三个连续结点。



```
struct node
{
    int data;
    struct node *next;
} *p,*q,*r;
```

现要将 q 所指结点从链表中删除,同时保持链表的连续,以下不能完成指定操作的语句是\_\_\_\_\_。

- A. p->next=q->next;                      B. p->next=p->next->next;  
C. p->next=r;                      D. p=q->next;

48. 以下对结构体类型变量 td 的定义中,错误的是\_\_\_\_\_。

- |  |  |
|--|--|
| <p>A. typedef struct aa</p> <pre>{     int n;     float m; }AA; AA td;</pre> | <p>B. struct aa</p> <pre>{     int n;     float m; } td; struct aa td;</pre> |
| <p>C. struct</p> <pre>{     int n;     float m; }aa; struct aa td;</pre>     | <p>D. struct</p> <pre>{     int n;     float m; }td;</pre>                   |

49. 以下与函数 fseek(fp,0L,SEEK\_SET)有相同作用的是\_\_\_\_\_。

- A. feof(fp)                      B. ftell(fp)                      C. fgetc(fp)                      D. rewind(fp)

50. 有以下程序:

```
#include "stdio.h"
void WriteStr(char *fn,char *str)
{
```

```
FILE *fp;
fp=fopen(fn,"W");
fputs(str,fp);
fclose(fp);
}
main()
{
    WriteStr("t1.dat","start");
    WriteStr("t1.dat","end");
}
```

程序运行后,文件 t1.dat 中的内容是\_\_\_\_\_。

- A. start                      B. end                      C. startend                      D. endrt

## 二、填空题（每空 2 分，共 40 分）

1. 某二叉树中，度为 2 的结点有 18 个，则该二叉树中有 【1】 个叶子结点。
2. 在面向对象的方法中，类的实例称为 【2】。
3. 诊断和改正程序中错误的工作通常称为 【3】。
4. 在关系数据库中，把数据表示成二维表，每一个二维表称为 【4】。
5. 问题处理方案的正确而完整的描述称为 【5】。
6. 在以下程序运行时，若从键盘输入：10 20 30<回车>，则输出结果是 【6】。

```
#include <stdio.h>
main()
{
    int i=0,j=0,k=0;
    scanf("%d%d%d",&i,&j,&k);
    printf("%d%d%d\n",i,j,k);
}
```

7. 以下程序运行后的输出结果是 【7】。

```
#define S(x) 4*x*x+1
main()
{
    int i=6,j=8;
    printf("%d\n",S(i+j));
}
```

8. 以下程序运行后的输出结果是 【8】。

```
main()
{
    int a=3,b=4,c=5,t=99;
    if(b<a&& a<c) t=a;a=c;c=t;
    if(a<c&& b<c) t=b;b=a;a=t;
    printf("%d%d%d\n",a,b,c);
}
```

```
}
```

9. 以下程序运行后的输出结果是 【9】。

```
main()
{
    int a,b,c;
    a=10;
    b=20;
    c=(a%b<1)||a/b>1;
    printf("%d %d %d\n",a,b,c);
}
```

10. 以下程序运行后的输出结果是 【10】。

```
main()
{
    char c1,c2;
    for(c1='0',c2='9';c1<c2;c1++,c2--)
        printf("%c%c",c1,c2);
    printf("\n");
}
```

11. 已知字符 A 的 ASCII 代码值为 65，在以下程序运行时，若从键盘输入：B33<回车>，则输出结果是 【11】。

```
#include "stdio.h"
main()
{
    char a,b;
    a=getchar();
    scanf("%d",&b);
    a=a-'A'+0;
    b=b*2;
    printf("%c %c\n",a,b);
}
```

12. 以下程序中，fun 函数的功能是求 3 行 4 列的二维数组每行元素中的最大值，请填空。

```
void fun(int, int, int (*)[4],int *) ;
main()
{
    int a[3][4]={ { 12,41,36,28 }, { 19,33,15,27 }, { 3,27,19,1 } },b[3],i;
    fun(3,4,a,b);
    for(i=0;i<3;i++)
        printf("%4d",b[i]);
    printf("\n");
}
```

```
}  
void fun(int m,int n,int ar[][4],int *br)  
{  
    int i,j,x;  
    for(i=0;i<m;i++)  
    {  
        x=ar[i][0];  
        for(j=0;j<n;j++)  
            if(x<ar[i][j])  
                x=ar[i][j];  
        【12】 =x;  
    }  
}
```

13. 以下程序运行后的输出结果是 【13】。

```
void swap(int x,int y)  
{  
    int t;  
    t=x;  
    x=y;  
    y=t;  
    printf("%d    %d    ",x,y);  
}  
main()  
{  
    int a=3,  
        b=4;  
    swap(a,b);  
    printf("%d    %d\n",a,b);  
}
```

14. 以下程序运行后的输出结果是 【14】。

```
#include "string.h"  
void fun(char *s,int p,int k)  
{  
    int i;  
    for(i=p;i<k-1;i++)  
        s[i]=s[i+2];  
}  
main()  
{
```

```
char s[]="abcdefg";  
fun(s,3,strlen(s));  
puts(s);  
}
```

15. 以下程序运行后的输出结果是 【15】。

```
#include "string.h"  
main()  
{  
    char ch[]="abc",x[3][4];  
    int i;  
    for(i=0;i<3;i++)  
        strcpy(x,ch);  
    for(i=0;i<3;i++)  
        printf("%s",x);  
    printf("\n");  
}
```

16. 以下程序运行后的输出结果是 【16】。

```
fun(int a)  
{  
    int b=0;  
    static int c=3;  
    b++;    c++;  
    return (a+b+c);  
}  
main()  
{  
    int i,a=5;  
    for(i=0;i<3;i++)  
        printf("%d %d ",i,fun(a));  
    printf("\n");  
}
```

17. 以下程序运行后的输出结果是 【17】。

```
struct NODE  
{  
    int k;  
    struct NODE *link;  
};  
main()
```



```
{
struct NODE m[5],*p=m,*q=m+4;
int i=0;
while(p!=q)
{
    p->k=++i;
    p++;
    q->k=i++;
    q--;
}
q->k=i;
for(i=0;i<5;i++)
    printf("%d",m[i]);
printf("\n");
}
```

18. 以下程序中函数 `huiwen` 的功能是检查一个字符串是否是回文，当字符串是回文时，函数返回字符串：`yes!`；否则函数返回字符串：`no!`，并在主函数中输出。所谓回文即正向与反向的拼写都一样，例如：`adgda`。请填空。

```
#include "string.h"
char *huiwen(char *str)
{
    char *p1,*p2;
    int i,t=0;
    p1=str;
    p2= 【18】;
    for(i=0;i<=strlen(str)/2;i++)
        if(*p1++!=*p2--)
        {
            t=1; break;
        }
    if(【19】)
        return ("yes!");
    else
        return ("no!");
}
main()
{
    char str[50];
```

```
printf("Input:");  
scanf("%s",str);  
printf("%s\n",  【20】 );  
}
```

## 笔试测试题二

一、选择题（1~10 每小题 2 分，11~50 每小题 1 分，共 60 分）

下列各题 A、B、C、D 四个选项中，只有一个选项是正确的。

1. 下列叙述中正确的是\_\_\_\_\_。  
A. 程序设计就是编制程序  
B. 程序的测试必须由程序员自己去完成  
C. 程序经调试、改错后还应进行再测试  
D. 程序经调试、改错后不应进行再测试
2. 下列数据结构中，能用二分法进行查找的是\_\_\_\_\_。  
A. 顺序存储的有序线性表  
B. 线性链表  
C. 二叉链表  
D. 有序线性链表
3. 下列关于栈的描述中正确的是\_\_\_\_\_。  
A. 在栈中只能插入元素而不能删除元素  
B. 在栈中只能删除元素而不能插入元素  
C. 栈是特殊的线性表，只能在一端插入或删除元素  
D. 栈是特殊的线性表，只能在一端插入元素，而在另一端删除元素
4. 下列叙述中正确的是\_\_\_\_\_。  
A. 一个逻辑数据结构只能有一种存储结构  
B. 数据的逻辑结构属于线性结构，存储结构属于非线性结构  
C. 一个逻辑数据结构可以有多种存储结构，且各种存储结构不影响数据处理的效率  
D. 一个逻辑数据结构可以有多种存储结构，且各种存储结构影响数据处理的效率
5. 下列叙述中正确的是\_\_\_\_\_。  
A. 软件工程只是解决软件项目的管理问题  
B. 软件工程主要解决软件产品的生产率问题  
C. 软件工程的主要思想是强调在软件开发过程中需要应用工程化原则  
D. 软件工程只是解决软件开发中的技术问题
6. 在软件设计中，不属于过程设计工具的是\_\_\_\_\_。  
A. PDL（过程设计语言）  
B. PAD 图  
C. N-S 图  
D. DFD 图
7. 下列叙述中正确的是\_\_\_\_\_。  
A. 软件交付使用后还需要进行维护  
B. 软件一旦交付使用就不需要再进行维护  
C. 软件交付使用后其生命周期就结束  
D. 软件维护是指修复程序中被破坏的指令
8. 数据库设计的根本目标是要解决\_\_\_\_\_。  
A. 数据共享问题  
B. 数据安全问题  
C. 大量数据存储空间  
D. 简化数据维护

9. 设有如下关系：

R		
A	B	C
1	1	2
2	2	3

S		
A	B	C
3	1	3

T		
A	B	C
1	1	2
2	2	3
3	1	3

则下列操作中正确的是\_\_\_\_\_。

- A.  $T=R \bowtie S$       B.  $T=R \cup S$       C.  $T=R \times S$       D.  $T=R \div S$

10. 数据库系统的核心是\_\_\_\_\_。

- A. 数据模型      B. 数据库管理系统      C. 数据库      D. 数据库管理员

11. 以下叙述中错误的是\_\_\_\_\_。

- A. 用户所定义的标识符允许使用关键字  
B. 用户所定义的标识符应尽量做到“见名知意”  
C. 用户所定义的标识符必须以字母或下划线开头  
D. 用户所定义的标识符中，大、小写字母代表不同的标识

12. 以下叙述中错误的是\_\_\_\_\_。

- A. C 语句必须以分号结束  
B. 复合语句在语法上被看做一条语句  
C. 空语句出现在任何位置都不会影响程序运行  
D. 赋值表达式末尾加分号就构成了赋值语句

13. 以下叙述中正确的是\_\_\_\_\_。

- A. 调用 printf 函数时，必须要有输出项  
B. 使用 putchar 函数时，必须在之前包含头文件 stdio.h  
C. C 语言编写的文件，整数可以以十二进制、八进制或十六进制的形式输出  
D. 使用 putchar 函数读入字符时，可以从键盘上输入字符所对应的 ASCII 码

14. 以下关于函数的叙述中正确的是\_\_\_\_\_。

- A. 每个函数都可以被其他函数调用（包括 main 函数）  
B. 每个函数都可以被单独编译  
C. 每个函数都可以单独运行  
D. 在一个函数内部可以定义另一个函数

15. 若有语句：char \*line[5];，以下叙述中正确的是\_\_\_\_\_。

- A. 定义 line 是一个数组，每个数组元素是一个基类型为 char 的指针变量  
B. 定义 line 是一个指针变量，该变量指向一个长度为 5 的字符型数组  
C. 定义 line 是一个指针数组，语句中的\*号称为间址运算符  
D. 定义 line 是一个指向字符型函数的指针

16. 有以下程序段：

```
typedef struct NODE
{int num; struct NODE *next;
}OLD;
```

以下叙述中正确的是\_\_\_\_\_。

- A. 以上的说明形式不规范      B. NODE 是一个结构体类型  
C. OLD 是一个结构体类型      D. OLD 是一个结构体变量
17. 以下叙述中错误的是\_\_\_\_\_。
- A. C 语言中对二进制文件的访问速度比文本文件快  
B. C 语言中, 随机文件以二进制代码形式存储数据  
C. 语句 FILE fp; 定义了一个名为 fp 的文件指针  
D. C 语言中的文本文件以 ASCII 码形式存储数据
18. 当把以下 4 个表达式用作 if 语句的控制表达式时, 其中有一个与其他 3 个含义不同, 该表达式是\_\_\_\_\_。
- A. k%2      B. k%2==1      C. (k%2)!=0      D. !k%2==1
19. 以下不能正确计算代数式  $\frac{1}{3}\sin^2(\frac{1}{2})$  值的 C 语言表达式是\_\_\_\_\_。
- A. 1/3\*sin(1/2)\*sin(1/2)      B. sin(0.5)\*sin(0.5)/3  
C. pow(sin(0.5),2)/3      D. 1/3.0\*pow(sin(1.0/2),2)
20. 以下不能正确定义且赋值的语句是\_\_\_\_\_。
- A. int n1=n2=10;      B. char c=32;      C. float f=f+1.1      D. double x=12.3E2.5
21. 以下程序的功能是: 给 r 输入数据后计算半径为 r 的圆面积 s。

```
main()
/*Beginning*/
{int r; float s;
scanf("%d",&r);
s=* π *r*r;    printf("s=%f\n",s);
}
```

程序在编译时出错, 出错的原因是\_\_\_\_\_。

- A. 注释语句书写位置错      B. 存放圆半径的变量 r 不应该定义为变量  
C. 输出语句中格式描述符不对      D. 计算圆面积的赋值语句中使用了非法变量
22. 设有定义: int k=1,m=2; float f=7; , 则以下选项中错误的表达式是\_\_\_\_\_。
- A. k=k>=k      B. -k++      C. k%int(f)      D. k>=f>=m
23. 设有定义: int a=2,b=3,c=4; , 则以下选项中值为 0 的表达式是\_\_\_\_\_。
- A. (!a==1)&&(!b==0)      B. (a<b)&&!c||1  
C. a&&b      D. a||(b+b)&&(c-a)
24. 有以下程序段:
- ```
int k=0,a=1,b=2,c=3;
k=a<b?b:a;    k=k>c?c:k;
```
- 执行该程序段后, k 的值是\_\_\_\_\_。
- A. 3      B. 2      C. 1      D. 0
25. 设变量 a、b、c、d 和 y 都已正确定义并赋值。若有以下 if 语句:
- ```
if(a<b)
if(c==d) y=0;
```

else y=1;

该语句所表示的含义是\_\_\_\_\_。

$$A. y = \begin{cases} 0 & a < b \text{ 且 } c = d \\ 1 & a = b \end{cases}$$

$$B. y = \begin{cases} 0 & a < b \text{ 且 } c = d \\ 1 & a = b \text{ 且 } c = d \end{cases}$$

$$C. y = \begin{cases} 0 & a < b \text{ 且 } c = d \\ 1 & a < b \text{ 且 } c = d \end{cases}$$

$$D. y = \begin{cases} 0 & a < b \text{ 且 } c = d \\ 1 & c = d \end{cases}$$

26. 有以下程序段：

```
int n,t=1,s=0;
scanf("%d",&n);
do{ s=s+t;t=t-2;} while(t!=n);
```

为使此程序不陷入死循环，从键盘输入的数据应该是\_\_\_\_\_。

A. 任意正奇数      B. 任意负偶数      C. 任意正偶数      D. 任意负奇数

27. 设变量已正确定义，则以下能正确计算  $f=n!$  的程序段是\_\_\_\_\_。

A. f=1;

B. f=0;

for(i=n;i>=2;i++) f\*=i;

for(i=1;i<=n;i--) f\*=i;

C. f=1;

D. f=1;

for(i=1;i<=n;i++) f\*=i;

for(i=n;i>1;i++) f\*=i;

28. 设有定义：int n1=0,n2,\*p=&n2,\*q=&n1;，以下赋值语句中与  $n2=n1$  语句等价的是\_\_\_\_\_。

A. \*p=\*q;

B. p=\*;

C. \*p=&n1;

D. p=\*q;

29. 若有定义：int x=0,\*p=&x;，则语句 printf("%d\n",\*p); 的输出结果是\_\_\_\_\_。

A. 随机数

B. 0

C. x 的地址

D. p 的地址

30. 设函数 fun 的定义形式为

```
void fun(char ch, float x)
{ ... }
```

则以下对函数 fun 的调用语句中正确的是\_\_\_\_\_。

A. fun("abc",3.0);

B. t=fun('D',16.5);

C. fun('65',2.8);

D. fun(32,32);

31. 有以下程序：

```
main()
{ int a[10]={1,2,3,4,5,6,7,8,9,10},*p=&a[3],*q=p+2;
printf("%d\n",*p+*q);
}
```

程序运行后的输出结果是\_\_\_\_\_。

A. 16

B. 10

C. 8

D. 6

32. 有以下程序：

```
main()
{ char p[]={'a','b','c'},q[]="abc";
printf("%d %d\n",sizeof(p), sizeof(q));
}
```

```
}
```

程序运行后的输出结果是\_\_\_\_\_。

A. 4 4

B. 3 3

C. 3 4

D. 4 3

33. 有以下程序：

```
#define f(x) (x*x)
main()
{   int  i1,i2;
    i1=f(8)/f(4); i2=f(4+4)/f(2+2);
    printf("%d, %d\n",i1, i2);
}
```

程序运行后的输出结果是\_\_\_\_\_。

A. 64,28

B. 4,4

C. 4,3

D. 64,64

34. 有以下程序：

```
#define f(x) (x*x)
main()
{   char a1='M',a2='m';
    printf("%c\n",(a1,a2));
}
```

以下叙述中正确的是\_\_\_\_\_。

A. 程序输出大写字母 M

B. 程序输出小写字母 m

C. 格式说明符不足，编译出错

D. 程序运行时产生出错信息

35. 有以下程序：

```
#include <stdio.h>
main()
{   char c1='1',c2='2';
    c1=getchar(); c2=getchar(); putchar(c1); putchar(c2);
}
```

当程序运行时输入：a<回车>后，以下叙述正确的是\_\_\_\_\_。

A. 变量 c1 被赋予字符 a，c2 被赋予回车符

B. 程序将等待用户输入第 2 个字符

C. 变量 c1 被赋予字符 a，c2 中仍是原有字符 2

D. 变量 c1 被赋予字符 a，c2 中无定值

36. 有以下程序：

```
main()
{
    int  k=5,n=0;
    while(k>0)
    {
        switch(k)
```

```
        {default: break;
          case 1:n+=k;
          case 2:
          case 3:n+=k;
        }
        k--;
    }
    printf("%d \n",n);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 0                      B. 4                      C. 6                      D. 7

37. 有以下程序：

```
main()
{
    int a[]={2,4,6,8,10},y=0,x,*p;
    p=&a[1];
    for(x=1;x<3;x++)
        y+=p[x];
    printf("%d \n",y);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 10                      B. 11                      C. 14                      D. 15

38. 有以下程序：

```
void sort(int a[],int n)
{
    int i,j,t;
    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;}
}

main()
{
    int aa[10]={1,2,3,4,5,6,7,8,9,10},i;
    clrscr();
    sort(aa+2,5);
    for(i=0;i<10;i++)printf("%d,",aa[i]);
    printf("\n");
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 1,2,3,4,5,6,7,8,9,10,                      B. 1,2,7,6,3,4,5,8,9,10,  
C. 1,2,7,6,5,4,3,8,9,10,                      D. 1,2,9,8,7,6,5,4,3,10,

39. 有以下程序：

```
void sum(int a[])
{
    a[0]=a[-1]+a[1];
}
main()
{
    int a[10]={1,2,3,4,5,6,7,8,9,10},i;
    sum(&a[2]);
    printf("%d\n",a[2]);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 6                      B. 7                      C. 5                      D. 8

40. 有以下程序：

```
void swap(int c0[],int c1[])
{
    int t;
    t=c0[0]; c0[0]=c1[0]; c1[0]=t;
}
void swap2(int *c0,int *c1)
{
    int t;
    t=*c0; *c0=*c1; *c1=t;
}
main()
{
    int a[2]={ 3, 5},b[2]={3,5};
    swap(a,a+1); swap2(&b[0],&b[1]);
    printf("%d %d %d %d\n",a[0], a[1], b[0], b[1]);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 3 5 5 3                      B. 5 3 3 5                      C. 3 5 3 5                      D. 5 3 5 3

41. 有以下程序：

```
#include <stdio.h>
main()
{
    char p[]={ 'a', 'b', 'c'}, q[10]={ 'a', 'b', 'c'};
```



```
printf("%d %d \n",strlen(p),strlen(q));  
}
```

以下叙述中正确的是\_\_\_\_\_。

- A. 在给 p 和 q 数组置初值时，系统会自动添加字符串结束符，故输出的长度都为 3
- B. 由于 p 数组中没有结束符，故长度不能确定；但 q 数组中字符串长度为 3
- C. 由于 q 数组中没有结束符，故长度不能确定；但 p 数组中字符串长度为 3
- D. 由于 p 和 q 数组中都没有字符串结束符，故长度都不能确定

42. 有以下程序，其中函数 f 的功能是将多个字符串按字典顺序排序：

```
#include <string.h>  
void f(char *p[],int n)  
{  
    char *t; int i,j;  
    for(i=0;i<n-1;i++)  
        for(j=i+1;j<n;j++)  
            if(strcmp(p[i],p[j]>0))  
            {  
                t=p[i];p[i]=p[j];p[j]=t;  
            }  
}  
main()  
{  
    char *p[5]={"abcd","aabdfg","dcdbe","cd"};  
    f(p,5);  
    printf("%d \n",strlen(p[1]));  
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 2              B. 3              C. 6              D. 4

43. 有以下程序：

```
#include <string.h>  
void f(char *s, char *t)  
{  
    char *k;  
    k=*s; *s=*t; *t=k;  
    s++; t--;  
    if(*s)  
        f(s,t);  
}  
main()  
{
```

```
char str[10]="abcdefg",*p;
p=str+strlen(str)/2+1;
f(p,p-2);
printf("%s\n",str);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. abcdefg      B. gfedcba      C. gbcdefa      D. abedcfg

44. 有以下程序：

```
float f1(float n)
{ return n*n; }
float f2(float n)
{ return 2*n; }
main()
{
    float (*p1)(float), (*p2)(float), (*t)(float), y1, y2;
    p1=f1; p2=f2;
    y1=p2(p1(2.0));
    t=p1; p1=p2; p2=t;
    y2=p2(p1(2.0));
    printf("%3.0f,%3.0f\n", y1, y2);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 8,16      B. 8,8      C. 16,16      D. 4,8

45. 有以下程序：

```
int a=2;
int f(int n)
{
    static int a=3;
    int t=0;
    if(n%2)
    {
        static int a=4;
        t+=a++;
    }
    else
    {
        static int a=5;
        t+=a++;
    }
}
```

```
    return t+a++;  
}  
main()  
{  
    int s=a,i;  
    for(i=0;i<3;i++)  
        s+=f(i);  
    printf("%d\n",s);  
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 26            B. 28            C. 29            D. 24

46. 有以下程序：

```
#include <string.h>  
struct STU  
{  
    int num;  
    float TotalScore;  
};  
void f(struct STU p)  
{  
    struct STU s[2]={ {20044,550},{20045,537} };  
    p.num=s[1].num; p.TotalScore=s[1].TotalScore;  
}  
main()  
{  
    struct STU s[2]={ {20041,703},{20042,580} };  
    f(s[0]);  
    printf("%d   %3.0f\n",s[0].num,s[0].TotalScore);  
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 20045 537    B. 20044 550    C. 20042 580    D. 20041 703

47. 有以下程序：

```
#include <string.h>  
struct STU  
{  
    char name[10];  
    int num;  
};
```

```
void f(char *name, int num)
{
    struct STU s[2]={{"SunDan",20044},{"Penghua",20045}};
    num=s[0].num;
    strcpy(name,s[0].name);
}
main()
{
    struct STU s[2]={{"YangSan",20041},{"LiSiGuo",20042}}, *p;
    p =&s[1];
    f(p->name, p->num);
    printf("%s %d\n", p->name, p->num);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- |                  |                  |
|------------------|------------------|
| A. SunDan 20042  | B. SunDan 20044  |
| C. LiSiGuo 20042 | D. YangSan 20041 |

48. 有以下程序：

```
struct STU
{
    char name[10];
    int num;
    float TotalScore;
};
void f(struct STU *p)
{
    struct STU s[2]={{"SunDan",20044,550},{"Penghua",20045,537}},*q=s;
    ++p;
    ++q;
    *p = *q;
}
main()
{
    struct STU s[3]={{"YangSan",20041,703},{"LiSiGuo",20042,580}};
    f(s);
    printf("%s %d %3.0f\n", s[1].name, s[1].num, s[1].TotalScore);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- |                      |                      |
|----------------------|----------------------|
| A. SunDan 20044 550  | B. Penghua 20045 537 |
| C. LiSiGuo 20042 580 | D. SunDan 20041 703  |

49. 以下程序的功能是进行位运算：

```
main()
{
    unsigned char a,b;
    a=7^3;
    b=~4 & 3;
    printf("%d  %d\n", a,b);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 4 3      B. 7 3      C. 7 0      D. 4 0

50. 有以下程序：

```
#include <stdio.h>
main()
{
    FILE *fp;
    int i,k,n;
    fp=fopen("data.dat","w+");
    for(i=1; i<6;i++)
    {
        fprintf(fp,"%d  ",i);
        if(i%3==0)fprintf(fp,"\n");
    }
    rewind(fp);
    fscanf(fp,"%d%d",&k,&n);printf("%d  %d\n", k,n);
    fclose(fp);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 0 0      B. 123 45      C. 1 4      D. 1 2

## 二、填空题（每空 2 分，共 40 分）

1. 数据管理技术发展过程经过人工管理、文件系统和数据库系统 3 个阶段，其中数据独立性最高的阶段是 【1】。
2. 算法复杂度主要包括时间复杂度和 【2】 复杂度。
3. 在进行模块测试时，要为每个被测试的模块另外设计两类模块：驱动模块和承接模块（桩模块）。其中 【3】 的作用是将测试数据传送给被测试的模块，并显示被测试模块所产生的结果。
4. 一棵二叉树第六层（根结点为第一层）的结点数最多为 【4】 个。
5. 数据结构分逻辑结构和存储结构，循环队列属于 【5】 结构。
6. 以下程序运行的输出结果是 【6】。

```
main()
```

```
{
    int x=0210;
    printf("%x\n",x);
}
```

7. 以下程序运行后的输出结果是 【7】。

```
main()
{
    int a=1,b=2,c=3;
    if(c=a)
        printf("%d\n",c);
    else
        printf("%d\n",b);
}
```

8. 已有定义：double \*p;，请写出完整的语句，利用 malloc 函数使 p 指向一个双精度型的动态存储单元 【8】。

9. 以下程序运行后的输出结果是 【9】。

```
main()
{
    char c; int n=100;
    float f=10; double x;
    x=f*n/(c=50);
    printf("%d %f\n",n,x);
}
```

10. 以下程序的功能是计算：s=1+12+123+1234+12345，请填空。

```
main()
{
    int t=0, s=0, i;
    for(i=1; i<=5; i++)
    {
        t=i+ 【10】 ;
        s=s+t;
    }
    printf("s=%d\n",s);
}
```

11. 已知字母 A 的 ASCII 码为 65。以下程序运行后的输出结果是 【11】。

```
main()
{
    char a,b;
    a='A'-'5'+7;
```

```
    b=a+'6'-'2';
    printf("%d  %c\n",a,b);
}
```

12. 有以下程序：

```
int sub(int n)
{
    return (n/10+n%10);
}
main()
{
    int x,y;
    scanf("%d",&x);
    y=sub(sub(sub(x)));
    printf("%d\n",y);
}
```

若运行时输入：1234<回车>，程序的输出结果是 【12】。

13. 以下函数 `sstrcat()` 的功能是实现字符串的连接，即将 `t` 所指字符串复制到 `s` 所指字符串的尾部。例如：`s` 所指字符串为 `abcd`，`t` 所指字符串为 `efgh`，函数调用后 `s` 所指字符串为 `abcdefgh`。请填空。

```
#include <string.h>
void sstrcat(char *s, char *t)
{
    int i,n;
    n=strlen(s);
    for (i=0; 【13】; i++)
        *(s+i+n)=*(t+i);
    *(s+i+n)= '\0'
}
```

14. 以下程序运行后的输出结果是 【14】。

```
#include <string.h>
char *ss(char *s)
{
    char *p,t;
    p=s+1; t=*s;
    while(*p)
    {
        *(p-1)= *p;
        p++;
    }
}
```

```
        *(p-1)=t;
        return s;
    }
    main()
    {
        char *p,str[10]= "abcdefgh";
        p=ss(str);
        printf("%s\n",p);
    }
```

15. 以下程序运行后的输出结果是 【15】。

```
int f(int a[], int n)
{
    if (n>=1)
        return f(a,n-1)+a[n-1];
    else
        return 0;
}
main()
{
    int aa[5]={ 1,2,3,4,5 },s;
    s=f(aa,5);
    printf("%d\n",s);
}
```

16. 以下程序运行后的输出结果是 【16】。

```
struct NODE
{
    int num;
    struct NODE *next;
};
main()
{
    struct NODE s[3]={ { 1, \0'},{ 2, \0'},{ 3, \0' } }, *p, *q, *r;
    int sum=0;
    s[0].next=s+1;
    s[1].next=s+2;
    s[2].next=s;
    p=s; q= p->next; r= q->next;
    sum+= q->next->num;
    sum+= r->next->next->num;
```



```
    printf("%d\n",sum);
}
```

17. 以下程序的功能是输出如下形式的方阵：

```
13  14  15  16
 9   10  11  12
 5   6   7   8
 1   2   3   4
```

请填空。

```
main()
{
    int i,j,k;
    for(j=4;j 【17】 ;j--)
    {
        for(i=1;i<=4;i++)
            {x=(j-1)*4+【18】; printf("%4d",k);}
        printf("\n");
    }
}
```

18. 以下函数 rotate 的功能是：将 a 所指的 N 行 N 列的二维数组中的最后一行放到 b 所指的二维数组的第 0 列中，把 a 所指的二维数组中的第 0 行放到 b 所指的二维数组的最后一列中。b 所指的二维数组中的其他数据不变。

```
#define N 4
void rotate(int a[][N], int b[][N])
{
    int i,
    for(i=0;i<N;i++)
    {
        b[i][N-1]= 【19】;
        【20】=a[N-1][i]
    };
}
```

### 笔试测试题三

一、选择题（1~10 每小题 2 分，11~50 每小题 1 分，共 60 分）

下列各题 A、B、C、D 四个选项中，只有一个选项是正确的。

1. 程序流程图（PFD）中的箭头代表的是\_\_\_\_\_。

A. 数据流            B. 控制流            C. 调用关系            D. 组成关系

2. 下列叙述中正确的是\_\_\_\_\_。

- A. 接口复杂的模块，其耦合程度一定低  
B. 耦合程度弱的模块，其内聚程度一定高  
C. 耦合程度弱的模块，其内聚程度一定低  
D. 以上三种说法都不对
3. 算法一般都可以用哪几种控制结构组合而成？\_\_\_\_\_
- A. 循环、分支、递归                      B. 顺序、循环、嵌套  
C. 循环、递归、选择                      D. 顺序、选择、循环
4. 下列叙述中正确的是\_\_\_\_\_。
- A. 软件就是程序清单                      B. 软件就是存放在计算机中的文件  
C. 软件应包括程序清单以及运行结果      D. 软件包括程序和文档
5. 检查软件产品是否符合需求定义的过程称为\_\_\_\_\_。
- A. 确认测试      B. 集成测试      C. 验证测试      D. 验收测试
6. 下面不属于软件设计原则的是\_\_\_\_\_。
- A. 抽象              B. 模块化              C. 自底向上      D. 信息隐蔽
7. 在下列选项中，哪个不是算法应该具有的基本特征？\_\_\_\_\_
- A. 确定性              B. 可行性              C. 无穷性              D. 拥有足够的情报
8. 希尔排序法属于哪一种类型的排序法？\_\_\_\_\_
- A. 交换类排序法      B. 插入类排序法      C. 选择类排序法      D. 建堆类排序法
9. 在计算机中，算法是指\_\_\_\_\_。
- A. 查询方法      B. 加工方法      C. 解题方案的准确而完整的描述      D. 排序方法
10. 栈和队列的共同点是\_\_\_\_\_。
- A. 都是先进后出                      B. 都是先进先出  
C. 只允许在端点处插入和删除元素      D. 没有共同点
11. 用 C 语言编写的代码程序\_\_\_\_\_。
- A. 可以立即执行                      B. 是一个源程序  
C. 经过编译即可执行                      D. 经过编译解释才能执行
12. 结构化程序由 3 种基本结构组成，3 种基本结构组成的算法是\_\_\_\_\_。
- A. 可以完成任何复杂的任务                      B. 只能完成部分复杂的任务  
C. 只能完成符合结构化的任务                      D. 只能完成一些简单的任务
13. 以下定义语句中正确的是\_\_\_\_\_。
- A. `char a='A',b='B';`                      B. `float a=b=10.0;`  
C. `int a=10, *b=&a;`                      D. `float *a, b=&a;`
14. 下列选项中，不能用作标识符的是\_\_\_\_\_。
- A. `_1234_`              B. `_1_2`              C. `int _2_`              D. `2_int_`
15. 有以下定义语句：
- `double a,b; int w; long c;`
- 若各变量已正确赋值，则下列选项中正确的表达式是\_\_\_\_\_。
- A. `a=a+b=b++`                      B. `w%((int)a+b)`  
C. `(c+w)%(int)a`                      D. `w=a=b`

16. 有以下程序：

```
main()
{int m=3,n=4,x;
  x=-m++;
  x=x+8/++n;
  printf("%d\n",x);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 3                                      B. 5                                      C. -1                                      D. -2

17. 有以下程序：

```
main()
{char a='a', b;
  printf("%c, ",++a);
  printf("%c\n",b=a++);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. b,b                                      B. b,c                                      C. a,b                                      D. a,c

18. 有以下程序：

```
main()
{int m=0256,n=256;
  printf("%o %o\n",m,n);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 0256 0400                              B. 0256 256                              C. 256 400                              D. 400 400

19. 有以下程序：

```
main()
{
  int a=666, b=888;
  printf("%d\n",a,b);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 错误信息                                      B. 666                                      C. 888                                      D. 666 , 888

20. 有以下程序：

```
main()
{
  int i;
  for(i=0;i<3;i++)
    switch(i)
    {
```

```

        case 0: printf("%d",i);
        case 2: printf("%d",i);
        default: printf("%d",i);
    }
}

```

程序运行后的输出结果是\_\_\_\_\_。

- A. 022111                      B. 021021                      C. 000122                      D. 012

21. 若  $x$  和  $y$  代表整型数，以下表达式中不能正确表示数学关系  $|x-y|<10$  的是\_\_\_\_\_。

- A.  $\text{abs}(x-y)<10$                       B.  $x-y<-10 \&\& x-y<10$   
 C.  $!(x-y)<-10 || !(x-y)<10$                       D.  $(x-y)*(x-y)<100$

22. 有以下程序：

```

main()
{
    int a=3, b=4, c=5, d=2;
    if(a>b)
        if(b>c)
            printf("%d",d++ +1);
        else
            printf("%d",++d +1);
    printf("%d\n",d);
}

```

程序运行后的输出结果是\_\_\_\_\_。

- A. 2              B. 3              C. 43              D. 44

23. 下列条件语句中，功能与其他语句不同的是\_\_\_\_\_。

- A.  $\text{if}(a) \text{ printf}("%d\\n",x); \text{ else printf}("%d\\n",y);$   
 B.  $\text{if}(a==0) \text{ printf}("%d\\n",y); \text{ else printf}("%d\\n",x);$   
 C.  $\text{if}(a!=0) \text{ printf}("%d\\n",x); \text{ else printf}("%d\\n",y);$   
 D.  $\text{if}(a==0) \text{ printf}("%d\\n",x); \text{ else printf}("%d\\n",y);$

24. 有以下程序：

```

main()
{
    int i=0, s=0;
    for(; ;)
    {
        if(i==3||i==5) continue;
        if(i==6) break;
        i++;
        s+=i;
    }
}

```

```
printf("%d\n",s);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 10                      B. 13                      C. 21                      D. 程序进入死循环

25. 若变量已正确定义，要求程序完成求 5! 的计算，不能完成此操作的程序段是\_\_\_\_\_。

- A. for(i=1,p=1;i<=5;i++) p\*=i;  
B. for(i=1;i<=5;i++) {p=1; p\*=i;}  
C. i=1;p=1; while(i<=5) {p\*=i; i++;}  
D. i=1;p=1; do {p\*=i; i++;} while(i<=5);

26. 有以下程序段：

```
main()
{
    char a,b,c,d;
    scanf("%c,%c,%d,%d",&a,&b,&c,&d);
    printf("%c,%c,%c,%c\n",a,b,c,d);
}
```

若运行时从键盘上输入：6,5,65,66<回车>，则输出结果是\_\_\_\_\_。

- A. 6,5,A,B                B. 6,5,65,66              C. 6,5,6,5                D. 6,5,6,6

27. 以下能正确定义二维数组的是\_\_\_\_\_。

- A. int a[][3];                                      B. int a[][3]={2\*3};  
C. int a[][3]={};                                      D. int a[2][3]={ {1},{2},{3,4}};

28. 有以下程序：

```
int f(int a)
{ return a%2; }
main()
{
    int s[8]={1,3,5,2,4,6}, i, d=0;
    for(i=0;f(s[i]); i++) d+=s[i];
    printf("%d\n",d);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. 9                      B. 11                      C. 19                      D. 21

29. 若有以下说明和语句：

```
int c[4][5], (*p)[5];
p=c;
```

能够正确引用 c 数组元素的是\_\_\_\_\_。

- A. p+1                      B. \*(p+3)                      C. \*(p+1)+3                      D. \*(p[0]+2)

30. 有以下程序：

```
main()
```

```

{
    int a=7,b=8, *p,*q,*r;
    p=&a;
    q=&b;
    r=p; p=q; q=r;
    printf("%d, %d, %d, %d\n", *p, *q, a, b);
}

```

程序运行后的输出结果是\_\_\_\_\_。

- A. 8,7,8,7      B. 7,8,7,8      C. 8,7,7,8      D. 7,8,8,7

31. s1 和 s2 已正确定义并分别指向两个字符串，若要求：当 s1 所指的字符串大于 s2 所指的字符串时，执行语句 S；则以下选项中正确的是\_\_\_\_\_。

- A. if(s1>s2) S;      B. if(strcmp(s1,s2)) S;  
C. if(strcmp(s2,s1)>0) S;      D. if(strcmp(s1,s2)>0) S;

32. 设有定义语句：

```
int x[6]={2,4,6,8,5,7}, *p=x, k;
```

要求依次输出数组 6 个元素中的值，不能完成此操作的语句是\_\_\_\_\_。

- A. for(k=0; k<6; k++) printf("%2d", \*(p++));  
B. for(k=0; k<6; k++) printf("%2d", \*(p+k));  
C. for(k=0; k<6; k++) printf("%2d", \*p++);  
D. for(k=0; k<6; k++) printf("%2d", \*(p++));

33. 有以下程序：

```

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

```

程序运行后的输出结果是\_\_\_\_\_。

- A. 运行后报错      B. 6 6      C. 6 11      D. 5 10

34. 设有以下定义和语句：

```
int a[3][2]={1,2,3,4,5,6}, *p[3];
p[0]=a[1];
```

则\*(p[0]+1)所代表的数组元素是\_\_\_\_\_。

- A. a[0][1]      B. a[1][0]      C. a[1][1]      D. a[1][2]

35. 有以下程序：

```

main()
{
    char str[][10]={ "China", "Beijing" }, *p=str;

```

```
printf("%s\n",p+10);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A. China                      B. Beijing                      C. ng                      D. ing

36. 有以下程序：

```
main()
{
    char s []="ABCD", *p;
    for(p=s+1; p<s+4; p++) printf("%s\n",p);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- |         |      |      |        |
|---------|------|------|--------|
| A. ABCD | B. A | C. B | D. BCD |
| BCD     | B    | C    | CD     |
| CD      | C    | D    | D      |
| D       | D    |      |        |

37. 在函数调用过程中，如果函数 funA 调用了函数 funB，函数 funB 又调用了函数 funA，则\_\_\_\_\_。

- A. 称为函数的直接递归调用  
B. 称为函数的间接递归调用  
C. 称为函数的循环调用  
D. C 语言中不允许这样的递归调用

38. 已有定义：int a[10],i, \*p;，则合法的赋值语句是\_\_\_\_\_。

- A. p=100;                      B. p=a[5];                      C. p=a[2]+2;                      D. p=a+2;

39. 以下叙述中正确的是\_\_\_\_\_。

- A. 局部变量说明为 static 存储类型，其生存期将得到延长  
B. 全局变量说明为 static 存储类型，其作用域将被扩大  
C. 任何存储类的变量在未赋初值时，其值都是不确定的  
D. 形参可以使用的存储类说明符与局部变量完全相同

40. 设有定义语句：char c1=92, c2=92;，则以下表达式中值为零的是\_\_\_\_\_。

- A. c1^c2                      B. c1&c2                      C. ~c2                      D. c1|c2

41. 设程序中对 fun 函数有如下说明：

```
void *fun();
```

此说明的含义是\_\_\_\_\_。

- A. fun 函数无返回值  
B. fun 函数的返回值可以是任意的数据类型  
C. fun 函数的返回值是无值的指针类型  
D. 指针 fun 指向一个函数，无返回值

42. 有以下程序：

```
main()
```

```

{
    char s[]="Yes\n/No", *ps=s;
    puts(ps+4);
    *(ps+4)=0;
    puts(s);
}

```

程序运行后的输出结果是\_\_\_\_\_。(选项 D 中的第一行是空行。)

A. n/No	B. /No	C. n/No	D.
Yes	Yes	Yes	/No
/No		/No	Yes

43. 有以下程序：

```

main()
{
    unsigned int a;
    int b=-1;
    a=b;
    printf("%u",a);
}

```

程序运行后的输出结果是\_\_\_\_\_。

A. -1	B. 65535	C. 32767	D. -32767
-------	----------	----------	-----------

44. 有以下程序：

```

void fun(int *a, int i, int j)
{
    int t;
    if(i<j)
    {
        t=a[i]; a[i]=a[j]; a[j]=t;
        i++; j--;
        fun(a,i,j);
    }
}

main()
{
    int x[]={2,6,1,8},i;
    fun(x,0,3);
    for(i=0;i<4;i++) printf("%2d",x[i]);
    printf("\n");
}

```

程序运行后的输出结果是\_\_\_\_\_。



A. 1 2 6 8      B. 8 6 2 1      C. 8 1 6 2      D. 8 6 1 2

45. 有以下说明和定义语句：

```
struct student;
{
    int age; char num[8];
};
struct student stu[3]={ {20, "200401"}, {21, "200402"}, {19, "200403"} };
struct student *p=stu;
```

以下选项中引用结构体变量成员的表达式错误的是\_\_\_\_\_。

A. (p++)>num      B. p->num      C. (\*p).num      D. stu[3].age

46. 有以下程序：

```
main()
{
    int x[]={1,3,5,7,2,4,6,0},i,j,k;
    for(i=0;i<3;i++)
        for(j=2;j>=i;j--)
            if (x[j+1]>x[j])
            {
                k=x[j];
                x[j]=x[j+1];
                x[j+1]=k;
            }
    for(i=0;i<3;i++)
        for(j=4;j<7;j++)
            if (x[j]>x[j+1])
            {
                k=x[j];
                x[j]=x[j+1];
                x[j+1]=k;
            }
    for(i=0;i<8;i++) printf("%d",x[i]);
    printf("\n");
}
```

程序运行后的输出结果是\_\_\_\_\_。

A. 75310246      B. 01234567      C. 76310462      D. 13570246

47. 有以下程序：

```
#include <stdio.h>
main()
{
```



50. A. p->next                      B. p                      C. s                      D. s->next

二、填空题（每空 2 分，共 40 分）

1. 数据结构包括数据的逻辑结构、数据的【1】以及对数据的操作运算。
2. 软件维护活动包括以下几类：改正性维护、适应性维护、【2】维护和预防性维护。
3. 软件的需求分析阶段的工作，可以概括为 4 个方面：【3】、需求分析、编写需求规格说明书和需求评审。
4. 栈的基本运算有 3 种：入栈、退栈和【4】。
5. 实现算法所需的存储单元多少和算法的工作量大小分别称为算法的【5】。
6. 以下程序段的输出结果是【6】。

```
int x=9; printf("%o\n",x);
}
```

7. 以下程序运行后的输出结果是【7】。

```
main()
{int a,b,c;
  a=25;
  b=025;
  c=0x25;
  printf("%d %d %d\n",a,b,c);
}
```

8. 以下程序运行后的输出结果是【8】。

```
main()
{int p[7]={ 11,13,14,15,16,17,18};
  int i=0,j=0;
  while(i<7&&p[i]%2==1)j+=p[i++];
  printf("%d\n",j);
}
```

9. 以下程序运行后的输出结果是【9】。

```
main()
{
  int x=1,y=0,a=0,b=0;
  switch(x)
  {case 1: switch(y)
    {
      case 0: a++; break;
      case 1: b++; break;
    }
    case 2: a++; b++;break;
  }
  printf("%d %d\n",a,b);
}
```

```
}
```

10. 以下程序运行后的输出结果是 【10】。

```
main()
{
    int a[4][4]={ {1,2,3,4},{5,6,7,8},{11,12,13,14},{15,16,17,18}};
    int k=0,j=0,s=0;
    while(k++ <4)
    {
        if (k==2||k==4)
            continue;
        j=0;
        do
        {
            s+=a[k][j];
            j++;
        } while(j<4);
    }
    printf("%d\n",s);
}
```

11. 以下程序运行后的输出结果是 【11】。

```
main()
{
    char a[]="Language",b[]="Programe";
    char *p1, *p2; int k;
    p1=a; p2=b;
    for(k=0;k<=7;k++)
        if (*(p1+k)== *(p2+k))
            printf("%c", *(p1+k));
}
```

12. 以下程序运行后的输出结果是 【12】。

```
main()
{
    char a[]="123456789",*p; int k=1;
    p=a;
    while(*p)
    {
        if (k%2==0) *p="*";
        p++;
        k++;
    }
}
```

```
    }  
    puts(a);  
}
```

13. 以下程序中，for 循环体执行的次数是 **【13】**。

```
#define N 2  
#define M N+1  
#define K M+1*M/2  
main()  
{  
    int j;  
    for(j=1;j<K;j++);  
    {...}  
    ...  
}
```

14. 以下程序通过函数 SunFun 求  $\sum_{x=0}^{10} f(x)$ 。这里  $f(x)=x^2+1$ ，由 F 函数实现。请填空。

```
main()  
{  
    printf("The sum=%d\n",SunFun(10));  
}  
SunFun(int n)  
{  
    int x,s=0;  
    for(x=0;x<=n;x++)  
        s+=F(【14】);  
    return s;  
}  
F(int x)  
{  
    return (【15】);  
}
```

15. 以下程序从终端读入数据到数组中，统计其中正数的个数，并计算它们的和。请填空。

```
main()  
{  
    int k, a[20], sum, count;  
    for(k=0;k<20;k++)  
        scanf("%d", 【16】);  
    for(k=0;k<20;k++)  
    {
```

```

        if(a[k]>0)
            count++;
        sum+= 【17】;
    }
    printf("%sum=d,count=%d\n",sum,count);
}

```

16. 以下程序中，函数 SumColumnMin 的功能是：求出 M 行 N 列的二维数组每列元素中的最小值，并计算它们的和值。和值通过形参传回主函数。请填空。

```

#define M 2
#define N 4
void SumColumnMin(int a[M][N], int *sum)
{
    int i,j,k,s=0;
    for (i=0;i<N;i++)
    {
        k=0;
        for (j=1;j<M;j++);
        if(a[k][i]>a[j][i])
            k=j-1;
        s+= 【18】;
    }
    【19】=s;
}
main()
{
    int x[M][N]={3,2,5,1,4,1,8,3},s;
    SumColumnMin (【20】);
    printf("%d\n",s);
}

```

## 笔试测试题四

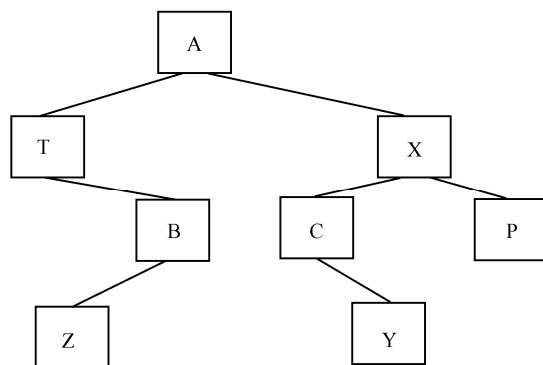
一、选择题（1~10 每小题 2 分，11~50 每小题 1 分，共 60 分）

下列各题 A、B、C、D 四个选项中，只有一个选项是正确的。

1. 下列叙述中正确的是\_\_\_\_\_。

- A. 算法的空间复杂度是指算法程序的长度
- B. 算法的效率只与所处理数据的规模有关，而与数据的存储结构无关
- C. 数据的逻辑结构是与存储结构一一对应的
- D. 上述三种说法都不对

2. 在结构化分析方法中, 数据字典的作用是\_\_\_\_\_。
- A. 存放所有需要处理的原始数据  
B. 存放所有处理的结果  
C. 存放所有的程序文件  
D. 描述系统中所用到的全部数据和文件的有关信息
3. 在下列数据管理的各个阶段中, 数据独立性最高的阶段是\_\_\_\_\_。
- A. 手工管理      B. 文件系统      C. 数据项管理      D. 数据库系统
4. 能将高级语言编写的源程序转换为目标程序的是\_\_\_\_\_。
- A. 链接程序      B. 解释程序      C. 编译程序      D. 编辑程序
5. 下列数据结构中具有记忆作用的是\_\_\_\_\_。
- A. 栈      B. 队列      C. 有序表      D. 二叉树
6. 下列二叉树进行后序遍历的结果是\_\_\_\_\_。



- A. ZBTYCPXA      B. ATBZXCYP      C. ZBTACYXP      D. ATBZXCYPY
7. 下列关于关系运算的叙述中正确的是\_\_\_\_\_。
- A. 投影、选择、连接是从二维表的行的方向来进行运算  
B. 并、交、差是从二维表的列的方向来进行运算  
C. 投影、选择、连接是从二维表的列的方向来进行运算  
D. 以上三种说法都不对
8. 在关系数据库中, 用来表示实体之间联系的是\_\_\_\_\_。
- A. 树结构      B. 网结构      C. 线性表      D. 二维表
9. 数据库概念设计中由分散到集中的设计方法是\_\_\_\_\_。
- A. 视图设计      B. 视图集成设计      C. 集中式模式设计      D. 分散式模式设计
10. 下列说法中, 正确的是\_\_\_\_\_。
- A. 结构化方法中, 各软件模块间的关系应该作到高耦合、低内聚  
B. 软件测试是为了验证软件的正确性  
C. 软件维护包括校正性维护、适应性维护、完整性维护和预防性维护  
D. 软件文档可分为用户文档、管理员文档、设计文档和系统文档
11. C 语言程序的基本单位是\_\_\_\_\_。
- A. 字符      B. 程序行      C. 语句      D. 函数
12. 以下叙述中正确的是\_\_\_\_\_。

- A. C 语言的源程序不必通过编译就可以直接运行  
B. C 语言中的每条可执行语句最终都将被转换成二进制的机器指令  
C. C 源程序经编译形成的二进制代码可以直接运行  
D. C 语言中的函数不可以单独进行编译
13. 以下符合 C 语言语法的实型常量是\_\_\_\_\_。  
A. 1.2E0.5                      B. 3.14.159E                      C. .5E-3                      D. E15
14. 以下 4 组用户定义标识符中，全部合法的一组是\_\_\_\_\_。  
A. \_main                      B. if                      C. txt                      D. int  
    enclude                      -max                      REAL                      k\_2  
    sin                      turbo                      3COM                      \_001
15. 若以下选项中的变量已正确定义，则正确的赋值语句是\_\_\_\_\_。  
A. x1=26.8%3                      B. 1+2=x2                      C. x3=0x12                      D. x4=1+2=3;
16. 设有以下定义：  
    int    a=0;  
    double b=1.25;  
    char c='A';  
    #define d    2  
    则下面语句中错误的是\_\_\_\_\_。  
A. a++;                      B. b++;                      C. c++;                      D. d++;
17. 设有定义：float a=2,b=4,h=3;，以下 C 语言表达式与代数式计算结果不相符的是\_\_\_\_\_。  
A. (a+b)\*h/2                      B. (1/2)\*(a+b)\*h                      C. (a+b)\*h\*1/2                      D. h/2\*(a+b)
18. 有以下程序：  
    main()  
    {  
        int x=102 , y=012;  
        printf("%2d,%2d\n",x,y);  
    }  
    执行后输出结果是\_\_\_\_\_。  
A. 10,01                      B. 002,12                      C. 102,10                      D. 02,10
19. 以下 4 个选项中，不能看做一条语句的是\_\_\_\_\_。  
A. {}                      B. a=0,b=0,c=0;                      C. if(a>0);                      D. if(b==0) m=1;n=2;
20. 设有定义：int a,\*pa=&a ;，以下 scanf 语句中能正确为变量 a 读入数据的是\_\_\_\_\_。  
A. scanf("%d",pa);                      B. scanf("%d",a);  
C. scanf("%d",&pa);                      D. scanf("%d",\*pa);
21. 以下程序段中与语句 k=a>b?(b>c?1:0):0 ; 功能等价的是\_\_\_\_\_。  
A. if((a>b)&&(b>c)) k=1;                      B. if((a>b)||(b>c)) k=1;  
    else k=0;  
C. if(a<=b) k=0;                      D. if(a>b) k=1;  
    else if(b<=c) k=1;                      else if(b>c) k=1;



```
else k=0;
```

22. 有以下程序：

```
main()  
{ char k; int i;  
  for(i=1;i<3;i++)  
  { scanf("%c",&k);  
    switch(k)  
    { case '0': printf("another\n");  
      case '1': printf("number\n");  
    }  
  }  
}
```

程序运行时，从键盘输入：01<回车>，程序执行后的输出结果是\_\_\_\_\_。

- |            |            |            |           |
|------------|------------|------------|-----------|
| A. another | B. another | C. another | D. number |
| number     | number     | number     | number    |
| another    | number     |            |           |

23. 有以下程序：

```
main()  
{ int x=0,y=5,z=3;  
  while(z-->0&&++x<5) y=y-1;  
  printf("%d,%d,%d\n",x,y,z);  
}
```

程序执行后的输出结果是\_\_\_\_\_。

- |          |           |           |            |
|----------|-----------|-----------|------------|
| A. 3,2,0 | B. 3,2,-1 | C. 4,3,-1 | D. 5,-2,-5 |
|----------|-----------|-----------|------------|

24. 有以下程序：

```
main()  
{ int i,s=0;  
  for(i=1;i<10;i+=2) s+=i+1;  
  printf("%d\n",s);  
}
```

程序执行后的输出结果是\_\_\_\_\_。

- |                     |                      |
|---------------------|----------------------|
| A. 自然数 1 ~ 9 的累加和   | B. 自然数 1 ~ 10 的累加和   |
| C. 自然数 1 ~ 9 中的奇数之和 | D. 自然数 1 ~ 10 中的偶数之和 |

25. 有以下程序：

```
main()  
{ int i,n=0;  
  for(i=2;i<5;i++)  
  { do  
    { if(i%3) continue;
```

```

        n++;
    } while(!i);
    n++;
}
printf("n=%d\n",n);
}

```

程序执行后的输出结果是\_\_\_\_\_。

- A. n=5                      B. n=2                      C. n=3                      D. n=4

26. 若程序中定义了以下函数：

```

double myadd(double a,double b)
{ return (a+b);}

```

并将其放在调用语句之后，则在调用之前应该对该函数进行说明，以下选项中错误的说明是\_\_\_\_\_。

- A. double myadd(double a,b);                      B. double myadd(double,double);  
C. double myadd(double b,double a);              D. double myadd(double x,double y);

27. 有以下程序：

```

char fun(char x , char y)
{ if(x<y) return x;
  return y;
}
main( )
{ int a='9',b='8',c='7';
  printf("%c\n",fun(fun(a,b),fun(b,c)));
}

```

程序的执行结果是\_\_\_\_\_。

- A. 函数调用出错              B. 8                      C. 9                      D. 7

28. 设有定义：int n=0,\*p=&n,\*\*q=&p；，则以下选项中，正确的赋值语句是\_\_\_\_\_。

- A. p=1;                      B. \*q=2;                      C. q=p;                      D. \*p=5;

29. 有以下程序：

```

void f(int v , int w)
{ int t;
  t=v;v=w;w=t;
}
main( )
{ int x=1,y=3,z=2;
  if(x>y) f(x,y);
  else if(y>z) f(y,z);
  else if(x,z);
  printf("%d,%d,%d\n",x,y,z);
}

```

```
}
```

程序执行后的输出结果是\_\_\_\_\_。

- A. 1,2,3                  B. 3,1,2                  C. 1,3,2                  D. 2,3,1

30. 有以下程序段：

```
int a[10]={1,2,3,4,5,6,7,8,9,10},*p=&a[3],b;
```

```
b=p[5];
```

b 中的值是\_\_\_\_\_。

- A. 5                  B. 6                  C. 8                  D. 9

31. 有以下程序：

```
main( )
```

```
{ char a[]="abcdefg",b[10]="abcdefg";
```

```
printf("%d %d\n",sizeof(a),sizeof(b));
```

```
}
```

程序执行后的输出结果是\_\_\_\_\_。

- A. 7 7                  B. 8 8                  C. 8 10                  D. 10 10

32. 有以下程序：

```
void swap1(int c[])
```

```
{ int t;
```

```
t=c[0];c[0]=c[1];c[1]=t;
```

```
}
```

```
void swap2(int c0,int c1)
```

```
{ int t;
```

```
t=c0;c0=c1;c1=t;
```

```
}
```

```
main( )
```

```
{ int a[2]={3,5},b[2]={3,5};
```

```
swap1(a); swap2(b[0],b[1]);
```

```
printf("%d %d %d %d\n",a[0],a[1],b[0],b[1]);
```

```
}
```

其输出结果是\_\_\_\_\_。

- A. 5 3 5 3                  B. 5 3 3 5                  C. 3 5 3 5                  D. 3 5 5 3

33. 有以下程序：

```
void sum(int *a)
```

```
{ a[0]=a[1];}
```

```
main( )
```

```
{ int aa[10]={1,2,3,4,5,6,7,8,9,10},i;
```

```
for(i=2;i>=0;i--) sum(&aa[i]);
```

```
printf("%d\n",aa[0]);
```

```
}
```

其输出结果是\_\_\_\_\_。

- A. 4                      B. 3                      C. 2                      D. 1

34. 有以下程序：

```
int f(int b[][4])
{ int i,j,s=0;
  for(j=0;j<4;j++)
  { i=j;
    if(i>2) i=3-j;
    s+=b[i][j];
  }
  return s;
}

main( )
{ int a[4][4]={ {1,2,3,4},{0,2,4,5},{3,6,9,12},{3,2,1,0}};
  printf("%d\n",f(a));
}
```

其输出结果是\_\_\_\_\_。

- A. 12                      B. 11                      C. 18                      D. 16

35. 有以下定义：

```
#include <stdio.h>
char a[10],*b=a;
```

下列选项中不能给数组 a 输入字符串的语句是\_\_\_\_\_。

- A. gets(a)      B. gets(a[0])      C. gets(&a[0]);      D. gets(b);

36. 有以下程序：

```
main( )
{ char *p[10]={ "abc","aabdfg","dcdbe","abbd","cd"};
  printf("%d\n",strlen(p[4]));
}
```

其输出结果是\_\_\_\_\_。

- A. 2                      B. 3                      C. 4                      D. 5

37. 有以下程序：

```
int a=2;
int f(int *a)
{ return (*a)++; }
main( )
{ int s=0;
  { int a=5;
    s+=f(&a);
  }
}
```

```
s+=f(&a);
printf("%d\n",s);
}
```

其输出结果是\_\_\_\_\_。

- A. 10                      B. 9                      C. 7                      D. 8

38. 有以下程序：

```
#define f(x) x*x
main()
{ int i;
  i=f(4+4)/f(2+2);
  printf("%d\n",i);
}
```

其输出结果是 (     )

- A. 28                      B. 22                      C. 16                      D. 4

39. 设有以下语句：

```
typedef struct S
{ int g; char h;} T;
```

则下列叙述中正确的是\_\_\_\_\_。

- A. 可用 S 定义结构体变量                      B. 可以用 T 定义结构体变量  
C. S 是 struct 类型的变量                      D. T 是 struct S 类型的变量

40. 有以下程序：

```
struct STU
{
    char name[10];
    int num;
};
void f1(struct STU c)
{ struct STU b={"LiSiGuo",2042};
  c=b;
}
void f2(struct STU *c)
{ struct STU b={"SunDan",2044};
  *c=b;
}
main()
{ struct STU a={"YangSan",2041},b={"WangYin",2043};
  f1(a);f2(&b);
  printf("%d %d\n",a.num,b.num);
}
```

其输出结果是\_\_\_\_\_。

- A. 2041 2044    B. 2041 2043    C. 2042 2044    D. 2042 2043

41. 有以下程序：

```
main()
{ unsigned char a,b;
  a=4|3;
  b=4&3;
  printf("%d %d\n",a,b);
}
```

其输出结果是\_\_\_\_\_。

- A. 7 0    B. 0 7    C. 1 1    D. 43 0

42. 下面程序的功能是输出以下形式的金字塔图案：

```

      *
     ***
    *****
   ********
  main()
  { int i,j;
    for(i=1;i<=4;i++)
      { for(j=1;j<=4-i;j++) printf(" ");
        for(j=1;j<=_____;j++) printf("*");
        printf("\n");
      }
  }
```

在横线处应填入的是\_\_\_\_\_。

- A. i    B. 2\*i-1    C. 2\*i+1    D. i+2

43. 有以下程序：

```
void sort(int a[],int n)
{ int i,j,t;
  for(i=0;i<n-1;i+=2)
    for(j=i+2;j<n;j+=2)
      if(a[i]<a[j]) { t=a[i];a[i]=a[j];a[j]=t;}
}
main()
{ int aa[10]={1,2,3,4,5,6,7,8,9,10},i;
  sort(aa,10);
  for(i=0;i<10;i++) printf("%d,",aa[i]);
  printf("\n");
}
```

其输出结果是\_\_\_\_\_。

- A. 1,2,3,4,5,6,7,8,9,10,      B. 10,9,8,7,6,5,4,3,2,1,  
C. 9,2,7,4,5,6,3,8,1,10,      D. 1,10,3,8,5,6,7,4,9,2,

44. 以下程序段中，不能正确赋值（编译时系统会提示错误）的是\_\_\_\_\_。

- A. char s[10]="abcdefg";      B. char t[]="abcdefg",\*s=t;  
C. char s[10];s="abcdefg";      D. char s[10];strcpy(s,"abcdefg");

45. 有以下程序：

```
#include <string.h>
main(int argc, char *argv[])
{
    int i, len=0;
    for(i=1; i<argc; i+=2) len+=strlen(argv[i]);
    printf("%d\n", len);
}
```

经编译链接后生成的可执行文件是 ex.exe，若运行时输入以下带参数的命令行：

ex abcd efg h3 k44

执行后的输出结果是\_\_\_\_\_。

- A. 14      B. 12      C. 8      D. 6

46. 有以下程序：

```
void f(int a[], int i, int j)
{
    int t;
    if(i<j)
    {
        t=a[i]; a[i]=a[j]; a[j]=t;
        f(a, i+1, j-1);
    }
}

main()
{
    int i, aa[5]={1,2,3,4,5};
    f(aa, 0, 4);
    for(i=0; i<5; i++) printf("%d", aa[i]); printf("\n");
}
```

其输出结果是\_\_\_\_\_。

- A. 5,4,3,2,1,      B. 5,2,3,4,1,      C. 1,2,3,4,5,      D. 1,2,3,4,5,

47. 有以下程序：

```
struct STU
{
    char name[10];
    int num;
    int score;
};

main()
```

```

{   struct STU   s[5]={{"YangSan",20041,703},{ "LiSiGuo",20042,580},
    {"wangYin",20043,680},{ "SunDan",20044,550},
    {"Penghua",20045,537}},*p[5],*t;
    int i,j;
    for(i=0;i<5;i++)   p[i]=&s[i];
    for(i=0;i<4;i++)
        for(j=i+1;j<5;j++)
            if(p[i]->score>p[j]->score)
                { t=p[i];p[i]=p[j];p[j]=t;}
    printf("%d   %d\n",s[1].score,p[1]->score);
}

```

其输出结果是\_\_\_\_\_。

- A. 550 550      B. 680 680      C. 580 550      D. 580 680

48. 有以下程序：

```

#include <stdlib.h>
struct NODE
{ int num;
  struct NODE *next;
};
main( )
{   struct NODE *p,*q,*r;
    int sum=0;
    p=(struct NODE *)malloc(sizeof(struct NODE));
    q=(struct NODE *)malloc(sizeof(struct NODE));
    r=(struct NODE *)malloc(sizeof(struct NODE));
    p->num=1;q->num=2;r->num=3;
    p->next=q;q->next=r;r->next=NULL;
    sum+=q->next->num;sum+=p->num;
    printf("%d\n",sum);
}

```

其输出结果是\_\_\_\_\_。

- A. 3              B. 4              C. 5              D. 6

49. 有以下程序：

```

#include <stdio.h>
main( )
{   FILE *fp;   int i,k=0,n=0;
    fp=fopen("d1.dat","w");
    for(i=1;i<4;i++)   printf(fp,"%d",i);
    fclose(fp);
}

```



```
fp=fopen("d1.dat","r");
fscanf(fp,"%d%d",&k,&n);    printf("%d %d\n",k,n);
fclose(fp);
}
```

其输出结果是\_\_\_\_\_。

- A. 1 2                      B. 123 0                      C. 1 23                      D. 0 0

50. 有以下程序：

```
#include <stdio.h>
main( )
{ FILE *fp;  int i,a[4]={1,2,3,4},b;
  fp=fopen("data.dat","wb");
  for(i=0;i<4;i++)  fwrite(&a[i],sizeof(int),1,fp);
  fclose(fp);
  fp=fopen("data.dat","rb") ;
  fseek(fp,-2L*sizeof(int),SEEK_END); /*使位置指针从文件尾向前移 2*sizeof(int)字节 */
  fread(&b,sizeof(int),1,fp);/*从文件中读取 sizeof(int)字节的数据到变量 b 中*/
  fclose(fp);
  printf("%d\n",b);
}
```

其输出结果是\_\_\_\_\_。

- A. 2                      B. 1                      C. 4                      D. 3

二、填空题（每空 2 分，共 40 分）

1. 在结构化设计方法中，数据流图表达了问题中的数据流与加工之间的关系，并且，每一个【1】实际上对应一个处理模块。
2. 由于软件的复杂性和高成本，使大型软件的生产出现了很大困难，人们称之为【2】。
3. 在关系运算中，【3】运算是在指定的关系中选取所有满足给定条件的元组，构成一个新的关系，而这个新的关系是原关系的一个子集。
4. 在面向对象的程序设计中，用来请求对象执行某一处理或答某些信息的要求称为【4】。
5. 程序测试的方法分为静态分析和动态分析。使用测试用例在计算机上运行程序，使程序在运行过程中暴露错误，这种方法称为【5】。
6. 已知字符 A 的 ASCII 码值为 65，以下语句的输出结果是【6】。

```
char ch='B';
printf("%c %d\n",ch,ch);
```

7. 有以下语句段：

```
int  n1=10,n2=20;
printf("【7】",n1,n2);
```

要求按以下格式输出 n1 和 n2 的值，每个输出行从第 1 列开始，请填空。

```
n1=10
n2=20
```

8. 有以下程序：

```
main()
{
    int t=1,i=5;
    for(;i>=0;i--)
        t*=i;
    printf("%d\n",t);
}
```

其输出结果是 【8】。

9. 有以下程序：

```
main()
{
    int n=0,m=1,x=2;
    if(!n)    x-=1;
    if(m)     x-=2;
    if(x)     x-=3;
    printf("%d\n",x);
}
```

其输出结果是 【9】。

10. 有以下程序：

```
#include <stdio.h>
main()
{
    char ch1,ch2;  int n1,n2;
    ch1=getchar();
    ch2=getchar();
    n1=ch1-'0';
    n2=n1*10+(ch2-'0');
    printf("%d\n",n2);
}
```

程序运行时输入：12<回车>，执行程序后的输出结果是 【10】。

11. 有以下程序：

```
void f( int y,int *x)
{
    y=y+*x;
    *x=*x+y;
}
main()
{
```

```
int x=2,y=4;
f(y,&x);
printf("%d    %d\n",x,y);
}
```

其输出结果是 **【11】**。

12. 以下的函数 fun 的功能是计算  $x^n$  :

```
double fun(double x,int n)
{
    int i;
    double y=1;
    for(i=1;i<=n;i++)
        y=y*x;
    return y;
}
```

在主函数中已经正确定义 m、a、b 变量并赋值,若调用 fun 函数计算:  $m=a^4+b^4-(a+b)^3$ ,则实现这一计算的函数调用语句为 **【12】**。

13. 以下程序的功能是完成单向链表的建立与输出。

```
Struct data
{
    int num;
    struct data *next;
}
#define    LEN    sizeof(struct data)
#define    NULL    0
main()
{
    struct data *p1, *p2,*head;
    p1=p2=(struct data *)malloc(LEN);
    scanf("%d",&p1->num);
    head=p1;
    while(p1->num!=-999)
    {
        p1=(struct data *)malloc(LEN);
        p2->num=p1;
        scanf("%d", 【13】);
        p2=p1;
    }
    p1->num=NULL;
    p1=head;
```

```
while(【14】!=NULL)
{
    printf("%2d", p1->num);
    p1=p1->num;
}
```

14. 以下 sstrcpy( )函数实现字符串复制,即将 t 所指字符串复制到 s 所指向内存空间中,形成一个新的字符串 s。请填空。

```
void sstrcpy(char *s,char *t)
{ while(*s++=【15】); }
main( )
{
    char str1[100],str2[]="abcdefgh";
    sstrcpy(str1,str2);
    printf("%s\n",str1);
}
```

15. 下列程序的运行结果是【16】。

```
#include <string.h>
char *ss(char *s)
{
    return s+strlen(s)/2;
}
main( )
{
    char *p,*str="abcdefgh";
    p=ss(str);
    printf("%s\n",p);
}
```

16. 下面程序的运行结果是【17】。

```
int f( int a[], int n)
{
    if(n>1)
        return a[0]+f(&a[1],n-1);
    else
        return a[0];
}
main ( )
{
    int aa[3]={ 1,2,3},s;
```

```
s=f(&aa[0],3);
printf("%d\n",s);
}
```

17. 以下程序中给指针 p 分配了 3 个 double 型动态内存单元，请填空。

```
#include <stdlib.h>
main ( )
{
    double *p;
    p=(double *) malloc(【18】);
    p[0]=1.5;
    p[1]=2.5;
    p[2]=3.5;
    printf("%f%f%f\n",p[0],p[1],p[2]);
}
```

18. 以下程序的运行结果是【19】。

```
#include <string.h>
typedef struct student
{
    char name[10];
    long sno;
    float score;
}STU;
main( )
{
    STU a={"zhangsan",2001,95},b={"Shangxian",2002,90},c={"Anhua",2003,95},d,*p=&d;
    d=a;
    if(strcmp(a.name,b.name)>0)
        d=b;
    if(strcmp(c.name,d.name)>0)
        d=c;
    printf("%ld%s\n",d.sno,p->name);
}
```

19. 以下 sum 函数的功能是计算下列级数之和。

$$s = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \Lambda + \frac{x^n}{n!}$$

请给函数中的各变量正确赋初值。

```
double sum( double x, int n )
{
    int i; double a,b,s;
```

```

    【20】
    for( i=1;i<=n;i++)
    {
        a=a*x;
        b=b*i;
        s=s+a/b;
    }
    return  s;
}

```

## 笔试测试题一答案

### 一、选择题

1.D	2.B	3.D	4.C	5.A	6.C	7.B	8.D	9.D	10.C
11.B	12.A	13.C	14.D	15.B	16.C	17.A	18.B	19.A	20.B
21.D	22.A	23.B	24.C	25.B	26.A	27.C	28.D	29.D	30.C
31.C	32.B	33.A	34.A	35.D	36.C	37.D	38.C	39.D	40.C
41.B	42.A	43.D	44.B	45.A	46.A	47.D	48.C	49.D	50.B

### 二、填空题

【1】 19	【2】 对象	【3】 程序调试	【4】 关系
【5】 算法	【6】 10300	【7】 81	【8】 4599
【9】 10    20    0	【10】 0918273645	【11】 1    B	【12】 br[i]
【13】 4    3    3    4		【14】 abcfg	【15】 abcabcabc
【16】 0    10    1    11    2    12		【17】 13431	
【18】 str+strlen(str)-1		【19】 t==0 或 !t	【20】 huiwen(str)

## 笔试测试题二答案

### 一、选择题

1.C	2.A	3.C	4.D	5.C	6.D	7.A	8.A	9.B	10.B
11.A	12.C	13.B	14.B	15.A	16.B	17.C	18.D	19.A	20.B
21.D	22.C	23.A	24.B	25.C	26.D	27.C	28.A	29.B	30.B
31.B	32.C	33.C	34.B	35.A	36.D	37.C	38.C	39.A	40.D
41.A	42.A	43.B	44.A	45.C	46.D	47.A	48.B	49.A	50.D

### 二、填空题

【1】 数据库系统	【2】 空间	【3】 驱动模块	【4】 32
-----------	--------	----------	--------

- |   |              |                |                |
|---|--------------|----------------|----------------|
| 【5】存储   | 【6】88        | 【7】1           |                |
| 【8】 $p=(double*)(\text{malloc}(\text{sizeof}((double)))\text{或}(\text{double}*)(\text{malloc}(\text{sizeof}((8))))$ |              |                |                |
| 【9】2 20.000000  | 【10】 $t*10$  | 【11】67 G       | 【12】10         |
| 【13】*t  | 【14】bcdefgha | 【15】15         | 【16】5          |
| 【17】 $>0$ 或 $\geq 1$  | 【18】i        | 【19】 $a[0][i]$ | 【20】 $b[i][0]$ |

### 笔试测试题三答案

#### 一、选择题

- |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|
| 1.B  | 2.B  | 3.D  | 4.B  | 5.A  | 6.C  | 7.C  | 8.B  | 9.C  | 10.C |
| 11.B | 12.C | 13.C | 14.D | 15.C | 16.D | 17.A | 18.C | 19.B | 20.C |
| 21.C | 22.A | 23.D | 24.D | 25.B | 26.A | 27.C | 28.A | 29.D | 30.C |
| 31.D | 32.D | 33.A | 34.C | 35.B | 36.D | 37.B | 38.D | 39.A | 40.A |
| 41.C | 42.B | 43.B | 44.C | 45.D | 46.A | 47.C | 48.C | 49.A | 50.B |

#### 二、填空题

- |                |                |              |                  |
|----------------|----------------|--------------|------------------|
| 【1】存储结构        | 【2】完善性         | 【3】需求获取      | 【4】读栈顶元素         |
| 【5】空间复杂度和时间复杂度 | 【6】11          | 【7】25 21 37  | 【8】24            |
| 【9】2 1         | 【10】92         | 【11】gae      | 【12】 $1*3*5*7*9$ |
| 【13】4          | 【14】x          | 【15】 $x*x+1$ | 【16】 $\&a[k]$    |
| 【17】 $a[k]$    | 【18】 $a[k][i]$ | 【19】*sum     | 【20】x,&s         |

### 笔试测试题四答案

#### 一、选择题

- |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|
| 1.D  | 2.D  | 3.D  | 4.C  | 5.A  | 6.A  | 7.C  | 8.D  | 9.B  | 10.C |
| 11.D | 12.B | 13.C | 14.A | 15.C | 16.D | 17.B | 18.C | 19.D | 20.A |
| 21.A | 22.B | 23.B | 24.D | 25.D | 26.A | 27.D | 28.D | 29.C | 30.D |
| 31.C | 32.B | 33.A | 34.D | 35.B | 36.A | 37.C | 38.A | 39.D | 40.A |
| 41.A | 42.B | 43.C | 44.C | 45.D | 46.A | 47.C | 48.B | 49.B | 50.D |

#### 二、填空题

- |   |                           |                                |                                       |
|---|---------------------------|--------------------------------|---------------------------------------|
| 【1】加工   | 【2】软件微机                   | 【3】选择                          | 【4】消息                                 |
| 【5】动态分析   | 【6】B 66                   | 【7】 $n1=\%d\backslash nn2=\%d$ | 【8】0                                  |
| 【9】-4   | 【10】12                    | 【11】8 4                        |                                       |
| 【12】 $\text{fun}(a,4)+\text{fun}(b,4)-\text{fun}(a+b,3);$ | 【13】 $\&p1->\text{num}$   | 【14】 $p1->\text{num}$          |                                       |
| 【15】*t++  | 【16】efgh                  | 【17】6                          | 【18】 $3*\text{sizeof}(\text{double})$ |
| 【19】2002Shangxian   | 【20】 $a=1.0;b=1.0;s=1.0;$ |                                |                                       |

## 附录三 C 语言程序设计上机测试题

### 上机测试题一

#### 1. 填空题

以下程序中，函数 fun 的功能是：求出 s 所指字符串中最后一次出现的 t 所指字符串的地址，通过函数值返回。在主函数中输出从此地址开始的字符串。若未找到，则函数值为 NULL。例如，若字符串中的内容为“abcabdabfgh”，t 中的内容为“ab”，则输出结果是“abfgh”。当字符串中的内容为“abcabdabfgh”，t 中的内容为“abe”，则程序输出未找到信息：not be found!。

请勿改动主函数 main 和其他函数中的任何内容，仅在填空处填入所编写的语句。

```
#include <conio.h>
#include <stdio.h>
#include <string.h>
char * fun (char *s, char *t)
{
    char *p, *r, *a;
    a = 【 1 】;
    while (*s)
    {
        p = s;
        r = t;
        while (*r)
            if (【 2 】)
            {
                r++;
                p++;
            }
        else
            【 3 】;
        if (*r == '\0') a = s;
        s++;
    }
    return a;
}
```



```

main()
{
    char    s[100], t[100], *p;
    clrscr();
    printf("\nPlease enter string s :"); scanf("%s", s );
    printf("\nPlease enter substring t :"); scanf("%s", t );
    p = fun( s, t );
    if ( p ) printf("\nThe result is :  %s\n", p);
    else     printf("\nNot found !\n" );
}

```

答案：

【1】 s

【2】 \*r == \*p

【3】 break

解析：

【1】先将 s 字符串的首地址赋给 a，如 s 字符串不包含 t 字符串，则 a 字符串的起点是 s 字符串。

【2】只有当每一位的字符都相同时才执行。

【3】如有字符则退出当前循环。

## 2. 改错题

下列给定程序中，函数 fun 的功能是：求整数 x 的 y 次方的低 3 位值。例如：整数 5 的 6 次方为 12625，此值的低 3 位值为 625。

请改正下列 fun 函数中指定位置的错误，使它能得出正确的结果。

注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构。

```

#include <stdio.h>

long fun(int x,int y,long *p )
{ int i;
    long t=1;
    /*****found error*****/
    for(i=1; i<y; i++)
        t=t*x;
    *p=t;
    /*****found error*****/
    t=t/1000;
    return t;
}

main()
{ long t,r; int x,y;
    printf("\nInput x and y: "); scanf("%ld%ld",&x,&y);
    t=fun(x,y,&r);
}

```

```
printf("\n\nx=%d, y=%d, r=%ld, last=%ld\n\n", x, y, r, t );
}
```

答案：

(1) 错误：for(i=1; i<y; i++)                      正确：for(i=1; i<=y; i++)

(2) 错误：t=t/1000;                                  正确：t=t%1000;

解析：

错误 1：因为要计算 x 的 y 次方，所以应计算到 y。

错误 2：因为除是缩小倍数，取余是整除后取的余数部分。

### 3. 编程题

请编写一个函数 fun()，其功能是：找出一维整型数组元素中最大的值和它所在的下标，并将它们通过形参传回。数组元素中的值已在主函数中赋予。

主函数中 x 是数组名，n 是 x 中的数据个数，max 存放最大值，index 存放最大值所在元素的下标。

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的花括号中填入所编写的若干语句。

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
void fun(int a[], int n, int *max, int *d )
```

```
{/*编写程序代码*/
```

```
}
```

```
main()
```

```
{
```

```
int i, x[20], max, index, n = 10;
```

```
randomize();
```

```
for (i=0; i < n; i++)
```

```
{
```

```
    x[i] = rand()%50;
```

```
    printf("%4d", x[i]);
```

```
}
```

```
printf("\n");
```

```
fun( x, n, &max, &index);
```

```
printf("Max =%5d , Index =%4d\n", max, index );
```

```
}
```

答案：

```
int k;
*max=a[0];
for (k=1;k<n;k++)
    if(*max<a[k])
    {
        *max=a[k];
        *d=k;
    }
```

解析：

本题的解题思路是将第一个数存放在 max 的地址内,用循环比较的方法去验证之后的所有数,如最大数小,则将该数存入最大 max 的地址中,并将下标存到另一个地址变量中,因为需要返回调用函数中,所以使用地址变量。

## 上机测试题二

### 1. 填空题

下列给定程序中,函数 fun()的功能是:按顺序给 s 所指数组中的元素赋予从 2 开始的偶数,然后再按顺序对每 5 个元素求一个平均值,并将这些值依次存放在 w 所指的数组中。若 s 所指数组中元素的个数不是 5 的倍数,多余部分忽略不计。例如,s 所指数组有 14 个元素,则只对前 10 个元素进行处理,不对最后的 4 个元素求平均值。

```
#include <stdio.h>
#define SIZE 20
fun (double *s,double *w)
{
    int k,i;
    double sum;
    for(k=2,i=0;i<SIZE;i++)
    {
        s[i]=k;
        k+=2;
    }
    sum=0.0;
    for(k=0,i=0;i<SIZE;i++)
    {
        sum+=s[i];
        if((i+1) 【1】 5==0)
        {
            w[k]=sum/5;
            sum=0.0;
        }
    }
}
```

```
        k++;
    }
}
    【2】 k;
}
main()
{
    double a[SIZE],b[SIZE/5];
    int i, k;
    k=fun(a,b);
    printf("The original data:\n");
    for(i=0;i<SIZE;i++)
    {
        if(i%5==0)
            printf("\n");
        printf("%4.0f ",a[i]);
    }
    printf("\n\nThe result:\n");
    for(i=0;i<k;i++)
        printf("%6.2f", 【3】);
    printf("\n\n");
}
```

答案：

【1】 %

【2】 return

【3】 b[i]

解析：

【1】根据题意，这里是执行按顺序对每 5 个元素求一个平均值的操作，所以应该使用取余符号“%”，如果是 5 的倍数，则该式子的 sum 值清零。

【2】根据 C 语言的规定，除了使用关键字 void 的任何一个子函数都应该有返回值，所以应该使用关键字 return 把变量 k 的值返回给主函数。

【3】由题意知，这里是把存在数组 b 中的内容依次循环输出，所以后面的变量名应该使用 b[i]。

## 2. 改错题

下列给定程序中，函数 fun()的功能是：逐个比较 a,b 两个字符串对应位置中的字符，把 ASCII 值较小或相等的字符依次存放到 c 数组中，形成一个新的字符串。例如：a 中的字符串为 fshADfg，b 中的字符串为 sdAEdi，则 c 中的字符串应为 fdAADf。请改正程序中的错误，使它能得到正确结果。

注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构。

```
#include <stdio.h>
#include <string.h>
void fun(char *p,char *q,char *c)
{
    int k=0;
    while(*p||*q)
    /*****found error*****/
    {
        if (*p<=*q)
            c[k]=*q;
        else
            c[k]=*p;
        if(*p)
            p++;
        if(*q)
            q++;
    /*****found error*****/
        k++;
    }
}
main()
{
    char a[10]="fshADfg",b[10]="sdAEdi",c[80]={'\0'};
    fun(a,b,c);
    printf("The string a:");
    puts(a);
    printf("The string b:");
    puts(b);
    printf("The result :");
    puts(c);
}
```

答案：

(1) 错误：if (\*p<=\*q)                      正确：if (\*p>=\*q)

(2) 错误：k++                                正确：k++;

解析：

错误 1：由于要把 ASCII 值较小或相等的字符存放到 c 数组中，故 if 语句的关系运算符应为“>=”。

错误 2：C 语言规定，每个语句都要用分号（；）结束。

### 3. 编程题

请编写一个函数 void fun(int tt[M][N], int pp[N]), tt 指向一个 M 行 N 列的二维数组, 求出二维数组每列中最大元素, 并依次放入 pp 所指的一维数组中。二维数组中的数已在主函数中给出。

请勿改动主函数 main 和其他函数中的任何内容, 仅在函数 fun 的花括号中填入所编写的若干语句。

```
#include    <conio.h>
#include    <stdio.h>
#define     M    3
#define     N    4
void fun(int tt[M][N],int pp[N])
{ /*编写程序代码*/

}
main()
{
    int t[M][N]={ { 68, 32, 54, 12},{ 14, 24, 88, 58},{ 42, 22, 44, 56} };
    int p[N],i,j,k;
    clrscr();
    printf("The riginal data is:\n");
    for(i=0;i<M;i++)
    {
        for(j=0;j<N;j++)
            printf("%6d",t[i][j]);
        printf("\n");
    }
    fun(t,p);
    printf("\nThe result  is:\n");
    for(k=0;k<N;k++)
        printf("%4d",p[k]);
    printf("\n");
}
```

答案 :

```
int i,j,max;
for(j=0;j<N;j++)
{
```

```

max=tt[0][j];          /*假设各列中的第一个元素最大*/
for(i=0;i<M;i++)
    if(tt[i][j]>max)
        max=tt[i][j];
        /*如各列中的其他元素比最大值还大，则将该元素存入 max 变量中*/
pp[j]=max;            /*将各列的最大值依次放入 pp 数组中*/
}

```

解析：

本题中函数的功能是求出二维数组中每列的最大元素。首先，假设各列中的第一个元素最大，然后利用行标值的移动来依次取得各列中其他元素的值，并与假设的最大值进行比较，如果遇到更大的，则把这个更大的元素看做当前该列中最大的元素，继续与该列中其他元素比较。

### 上机测试题三

#### 1. 填空题

以下程序中，函数fun的功能是：从整数10~65之间，选出能被3整除，且有一位上的数是5的数，并把这些数放在b所指的数组中，这些数的个数作为函数值返回。

请勿改动主函数 main 和其他函数中的任何内容，仅在填空处填入编写的语句。

```

#include <stdio.h>
fun( int  *b )
{
    int  k,a1,a2,i=0;
    for(k=10; k<=65; k++)
    {
        a2=__【1】__; /*a2存放十位数*/
        a1=k-a2*10; /*a1存放个位数*/
        if((k%3==0 && a2==5) __【2】 (k%3==0 && a1==5))
        {
            b[i]=k;
            i++;
        }
    }
    return __【3】__;
}

main( )
{
    int  a[100],k,m;
    m=fun( a );
    printf("The result is :\n");
    for(k=0; k<m; k++) printf("%4d",a[k]);  printf("\n");
}

```

答案：

【1】a2=k/10;

【2】||

【3】i

解析：

【1】因为a2存放十位数，所以将k除10，即可得到十位上的数。

【2】因为要选出能被3整除，且有一位上的数是5，所以条件应是或的关系。

【3】在子函数程序中，i是累计满足条件数的个数的，因此应返回i的值。

## 2. 改错题

以下程序中，函数 fun 的功能是实现两个整数的交换。例如，给 a 和 b 分别输入 60 和 65，则输出为：a=65 b=60。请改正函数 fun 中指定位置的错误，使它能得出正确的结果。

注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构。

```
#include <stdio.h>
#include <conio.h>
/*****found error*****/
void fun ( int a, b )
{ int t;
/*****found error*****/
    t = b; b = a; a = t;
}
main ( )
{ int a, b;
    clrscr();
    printf ( "Enter a, b : "); scanf ( "%d%d", &a, &b );
    fun ( &a, &b );
    printf ( " a = %d b = %d\n ", a, b );
}
```

答案：

(1) 错误：void fun ( int a, b )

正确：void fun ( int \*a, int \*b )

(2) 错误：t = b; b = a; a = t;

正确：t = \*b; \*b = \*a; \*a = t;

解析：

错误 1：因为在主函数中是将地址传给形参的，所以 fun 函数的形参在说明时应为指针型。

错误 2：因为 a, b 在子函数说明中是指针型，在使用时因为 t 是简单变量，所以应将内容交换，返回主调函数时是变量地址。

## 3. 编程题

请编写一个函数 fun()，其功能是：将所有大于 1 而小于整数 m 的非素数存入 xx 所指的数组中，并将非素数的个数通过 k 传回。例如：若输入 17，则输出 9 和 4,6,8,9,10,12,14,15,16。



请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的花括号中填入所编写的若干语句。

```
#include <conio.h>
```

```
#include <stdio.h>
```

```
void fun( int m, int *k, int xx[] )
```

```
{ /*编写程序代码*/
```

```
}
```

```
main()
```

```
{
```

```
    int m, n, zz[100];
```

```
    clrscr();
```

```
    printf( "\nPlease enter an integer number between 10 and 100: " );
```

```
    scanf( "%d", &n );
```

```
    fun( n, &m, zz );
```

```
    printf( "\n\nThere are %d non-prime numbers less than %d:", m, n );
```

```
    for( n = 0; n < m; n++ )
```

```
        printf( "\n   %4d", zz[n] );
```

```
}
```

答案：

```
    int i,j,a=0,p;
```

```
    for(i=4;i<m;i++)
```

```
    {
```

```
        if(i % 2 == 0)
```

```
        {
```

```
            xx[a]=i;
```

```
            a++;
```

```
            continue;
```

```
        }
```

```
        p=1;
```

```
        for(j=2;j<(i/2);j++)
```

```
            if(i%j==0)
```

```
                p=0;
```

```
    if(p==0)
    {
        xx[a]=i;
        a++;
    }
}
*k=a;
```

解析：

本题的解题思路是：因为要从 1 ~ m(m 为整数)中查找所需的数，则应用循环语句，从 4 开始，如该数是 2 的倍数，则是偶数，应存入 xx 所指数组中，并计数，继续找下一个数；如该数不是 2 的倍数，则是奇数，就要去验证它有无约数，如果有也应存入 xx 所指数组中，并计数，直到循环结束。因为 xx 是数组，可直接将地址带回主函数，非素数的个数通过 k 指针传回。

## 上机测试题四

### 1. 填空题

下列给定程序中，函数 fun()的功能是：将长整型数中每一位上为奇数的数依次取出，构成一个新数放在 t 中。高位仍在高位，低位仍在低位。例如：当 s 中的数为 87653142 时，t 中的数为 7531。

请勿改动主函数 main 和其他函数中的任何内容，仅在填空处填入所编写的语句。

```
#include <stdio.h>
#include <conio.h>
void fun(long s,long *t)
{
    int d;
    long s1=1;
    *t=0;
    while(s>0)
    {
        d=s __【1】__ 10;
        if(d%2!=0)
        {
            *t=d*s1+*t;
            s1*=__【2】__;
        }
        s/=__【3】__;
    }
}
main()
```

```

{
    long s, t;
    clrscr();
    printf("\nPlease enter s: ");
    scanf("%ld",&s);
    fun(s,&t);
    printf("The result is :%ld\n",t);
}

```

答案：

【1】%

【2】10

【3】10

解析：

【1】根据题意，此处是由低到高逐位提取变量 d 的每一位，然后再进行判断，所以可以推出这里应该使用取余符号“%”。

【2】根据题意，这里是把提取出来满足条件的数字依次从低到高填入一个新的变量 t 中，所以进行的操作应该是将每一个数字都循环乘以 10，再加上原来的部分，例如，依次得到数字 321，最后应该输出 123。

【3】由前面的算法可以看出，这里应该是进行了一次提取操作，所以应该舍弃最低位——一个位，即把变量 s 除以 10。

## 2. 改错题

下列给定程序中函数 fun()的功能是：将 M 行 N 列的二维数组中的字符数据按列的顺序依次存放到一个字符串中。例如：二维数组中的字符数据为：

```

M  M  M  M
A  A  A  A
G  G  G  G

```

则字符串中的内容应是：MAGMAGMAGMAG。请改正程序中的错误，使它能得到正确结果。

注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构。

```

#include <stdio.h>
#define      M      3
#define      N      4
void fun(char  (*s)[N], char *b)
{
    int  i,j;
    /*****found  error*****/
    for(i=1; i<=N; i)
        for(j=0;j<M; j++)
        {
            /*****found  error*****/

```

```

        *b=(*s)[j];
        b++;
    }
}
main()
{
    char a[100],w[M][N]={ {'M','M','M','M'},{'A','A','A','A'},{'G','G','G','G'}};
    int i,j;
    printf("The matrix:\n");
    for(i=0; i<M; i++)
    {   for(j=0;j<N; j++)printf("%3c",w[i][j]);
        printf("\n");
    }
    fun(w,a);
    printf("The A string:\n");puts(a);
    printf("\n\n");
}

```

答案：

(1) 错误：for(i=1; i<=N; i)

正确：for(i=0; i<N; i)

(2) 错误：\*b=(\*s)[j];

正确：\*b=s[j][i];

解析：

错误 1：因为 C 语言数组下标下界是从 0 开始的，所以 i=1 应改为 i=0，数组下标上界要小于 N，不能等于 N。

错误 2：因为是二维数组，指针地址又要按列排出顺序，所以使用数组元素赋值比较直观。

### 3. 编程题

编写函数 fun()，它的功能是：求 n 以内(不包括 n)同时能被 5 与 11 整除的所有自然数之和的平方根 s，并作为函数值返回。例如：n 为 1000 时，函数值应为 s=96.979379。

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的花括号中填入所编写的若干语句。

```

#include    <conio.h>
#include    <math.h>
#include    <stdio.h>
double fun(int n)
{   /*编写程序代码*/

```

```

}

```

```
main()
{
    clrscr();
    printf("s=%f\n", fun(1000));
}
```

答案：

```
double s=0.0;
int i;
/*从 0 ~ n 中找到既能被 5 整除同时又能被 11 整除的数，并对这些数进行求和*/
for(i=0; i<n; i++)
    if(i%5==0 && i%11==0)
        s=s+i;
s=sqrt(s);          /*对和求平方根*/
return s;
```

解析：

本题的解题思路是：逐个取得从 0 ~ n 之间的所有数，对每次取得的数进行条件判断，条件是既能被 5 整除同时又能被 11 整除，注意：这两个条件要求同时成立，因此用到了“&&”运算符。若满足条件，该数就被累加到 s 中去，求出所有符合条件的数后，用 sqrt() 函数(包含于头文件 <math.h> 中)对 s 求平方根。

## 上机测试题五

### 1. 填空题

学生的记录由学号和成绩组成，N 名学生的数据已在主函数中放入结构体数组 s 中，下列给定的程序中，函数 fun() 的功能是：把分数最高的学生数据放在 h 所指的数组中。注意：分数高的学生可能不只一个，函数返回分数最高学生的人数。

请勿改动主函数 main 和其他函数中的任何内容，仅在填空处填入所编写的语句。

```
#include <stdio.h>
#define 【1】 16
typedef 【2】
{
    char num[10];
    int s;
}STREC;
int fun (STREC *a, STREC *b)
{
    int i,j=0,max=a[0].s;
    for(i=0; i<N; i++)
```

```

        if(max<a[i].s) max=a[i].s;
    for(i=0;i<N;i++)
        if(max==a[i].s) b[ j_【3】 ]=a[i];
    return j;
}
main ()
{
    STREC s[N]={{"GA005",85},{"GA003",76},{"GA002",69},{"GA004",85},
        {"GA001",91},{"GA007",72},{"GA008",64},{"GA006",87},
        {"GA015",85},{"GA013",91},{"GA012",64},{"GA014",91},
        {"GA011",66},{"GA017",64},{"GA018",64},{"GA016",72}};
    STREC h[N];
    int i, n;
    FILE *out;
    n=fun(s,h);
    printf("The %d highest score :\n",n);
    for (i=0; i<n; i++)
        printf("%s %4d\n ",h[i].num,h[i].s);
    printf("\n");
    out=fopen("out45.dat", "w");
    fprintf(out, "%d\n",n);
    for(i=0; i<n; i++)
        fprintf(out, "%4d\n ",h[i].s);
    fclose(out);
}

```

答案：

【1】N

【2】struct

【3】++

解析：

【1】从 C 语言的学习中知道 #define 语句可以在函数的开头定义一些常量 ,其格式为“ #define 变量名 常量值 ”, 注意末尾不需要分号。

【2】根据题意，这里是一个结构的定义，很明显应该使用关键字 struct。

【3】由前面的算法可知，第 2 个循环的作用是找出最高成绩的学生记录，并存入 b 中，因为分数高的学生可能不只一个，所以这里的变量 j 也需要每次赋值之后加 1。

## 2. 改错题

下列给定程序中函数 fun()的功能是：计算正整数 num 的各位上的数字的平方和。例如：若输入 352，则输出应该是  $3^2+5^2+2^2=38$ ；若输入 328，则输出应该是  $3^2+2^2+8^2=77$ 。请改正程序中的错误，使它能得到正确的结果。

注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构。

```
#include <stdio.h>
#include <conio.h>
long fun(long num)
{
    /******found error******/
    long k=1;
    do
    {
        k+=(num%10)*(num%10);
        num/=10;
    }while(num)
    /******found error******/
    return (k);
}
main()
{
    long n;
    clrscr();
    printf("\Please enter a number:");
    scanf("%ld",&n);
    printf("\n%ld\n",fun(n));
}
```

答案：

【1】错误：long k=1;            正确：long k=0;

【2】错误：while(num)          正确：while(num);

解析：

错误 1：k 用来存放各位数字的平方和，初值应为 0。

错误 2：do-while 语句的语法，while()后应加分号。

### 3. 编程题

请编写函数 fun()，它的功能是：求 Fibonacci 数列中小于 t 的最大的一个数，结果由函数返回。

其中 Fibonacci 数列 F(n)的定义为

$$F(0) = 0, F(1) = 1$$

$$F(n) = F(n-1) + F(n-2)$$

例如：t=1000 时，函数值为 987。

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的花括号中填入所编写的若干语句。

```
#include <conio.h>
#include <math.h>
```

```
#include <stdio.h>
int fun(int t)
{ /*编写程序代码*/

}

main()
{
    int n;
    clrscr();
    n=1000;
    printf("n=%d, f=%d\n",n, fun(n));
}
```

答案：

```
int a=1,b=1,c=0,i; /*a 代表第 n-2 项, b 代表第 n-1 项, c 代表第 n 项*/
/*下列循环中若求得的数 c 比指定比较数小,则计算下一个 Fibonacci 数,对 a,b 重新置数*/
do
{
    c=a+b;
    a=b;
    b=c;
} while (c<t); /*如果求得的数 c 比指定比较数大,退出循环*/
c=a; /*此时数 c 的前一个 Fibonacci 数为小于指定比较数的最大的数*/
return c;
}
```

解析：

根据所给的数列定义,不难发现,该数列最终的结果是由两个数列之和组成的,所以可以在循环内部始终把 c 看成是前两项之和(即第 n 项),而 a 始终代表第 n-2 项, b 始终代表第 n-1 项(通过不断地重新赋值来实现)。应注意,退出循环时得到的数 c 是大于指定比较数的最小的数,而它前一个数就是小于指定比较数的最大的数。

## 上机测试题六

### 1. 填空题

以下程序中,函数 fun 的功能是:将单链表的元素从小到大排序。即若原链表元素为:10, 4, 2, 8, 6, 排序后链表元素为:2, 4, 6, 8, 10。

请勿改动主函数 main 和其他函数中的任何内容,仅在填空处填入所编写的语句。



```
#include <stdio.h>
#include <stdlib.h>
typedef struct node
{
    int    data;
    struct node *next;
} NODE;
void fun(NODE *h)
{
    NODE *p, *q;
    int t;
    p = h;
    while (p)
    {
        q = 【1】;
        while (【2】)
        {
            if (p->data > q->data)
            {
                t = p->data;
                p->data = q->data;
                q->data = t;
            }
            q = q->next;
        }
        p = 【3】;
    }
}
void main()
{
    NODE *h=NULL, *p, *q;
    int i, a[6] = {0, 10, 4, 2, 8, 6 };
    for (i=1; i<=5; i++)
    {
        q = (NODE *)malloc(sizeof(NODE));
        q->data = a[i];
        q->next = NULL;
        if (h == NULL)
            h = p = q;
```

```

        else
        {
            p->next = q;
            p = q;
        }
    }
    printf("排序前 : ");
    p = h;
    while (p)
    {
        printf("%d ", p->data);
        p = p->next;
    }
    fun(h);
    printf("\n 排序后 : ");
    p = h;
    while (p)
    {
        printf("%d ", p->data);
        p = p->next;
    }
}

```

答案：

【1】p->next

【2】q 或 q!=NULL

【3】p->next 或 (\*p).next 或 p[0].next

解析：

【1】将被比较结点 p 后的第一个结点赋给 q，并依次取出 q 以后的结点与结点 p 进行比较。

【2】比较时依次取出下一个结点，并判断是否到达链尾。

【3】将被比较结点 p 的下一个结点作为新的比较结点，然后进行新一轮的排序。

## 2. 改错题

下列给定程序中，函数 fun()的功能是：将字符串 s 中位于偶数位置的字符或 ASCII 码为奇数的字符放入字符串 t 中(规定第一个字符放在第 0 位中)。例如：字符串中的数据为 ADFESHDI，则输出应当是 AFESDI。请改正程序中的错误，使它能得到正确结果。

注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构。

```

#include    <conio.h>
#include    <stdio.h>
#include    <string.h>
#define    N    80

```

```

/*****found  error*****/
void fun(char s, char t[])
{
    int i, j=0;
    for(i=0; i<strlen(s);i++)
/*****found  error*****/
        if(i%2==0||s[i]%2!=0)
            t[j++]=s[i];
    t[j]='\0';
}
main()
{
    char s[N], t[N];
    clrscr();
    printf("\nPlease enter string s :");
    gets(s);
    fun(s,t);
    printf("\nThe result is :%s\n",t);
}

```

答案：

- (1) 错误：void fun(char s, char t[])  
 正确：void fun(char \*s, char t[])
- (2) 错误：if(i%2=0||s[i]%2!=0)  
 正确：if(i%2==0||s[i]%2!=0)

解析：

错误 1：由于本题中函数的功能是对字符串进行处理，而不是对单个字符进行处理，因此，函数的参数应为字符串指针。

错误 2：if 语句中要用关系运算符，而不是赋值运算符。

### 3. 编程题

请编写一个函数 fun()，它的功能是：计算并输出给定整数 n 的所有因子(不包括 1 与自身)的平方和(规定 n 的值不大于 100)。例如：若输入 n 的值为 56，则输出为 sum=1113。

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的花括号中填入所编写的若干语句。

```

#include <stdio.h>

long fun(int n)
{ /*编写程序代码*/

```

```

}

main()
{
    int n;
    long sum;
    printf("Input n:");
    scanf("%d", &n);
    sum=fun(n);
    printf("sum=%ld\n", sum);
}

```

答案：

```

int i;
long s=0;
for(i=2;i<=n-1;i++)          /*从 2 ~ n-1 中找 n 的所有因子*/
    if(n%i==0)
        s+=i*i;              /*将所有因子求平方加*/
return s;                    /*将平方和返回*/

```

解析：

本题的解题思路是用  $n$  逐个去除以  $2 \sim n-1$  之间的所有数，如果  $n$  能被除尽，则把所得到的一个因子的平方累加到  $s$  中去。

## 上机测试题七

### 1. 填空题

下列给定程序中，函数fun()的功能是：将ss所指字符串中所有下标为奇数位置的字母转换为大写(若该位置上不是字母，则不转换)。例如，若输入abc45eFg，则应输出aBc45EFg。

请勿改动主函数 main 和其他函数中的任何内容，仅在填空处填入所编写的语句。

```

#include <conio.h>
#include <stdio.h>
#include <string.h>
void fun(char *ss)
{
    int i;
    for(i=0;ss[i]!='\0';i++)
        if(i%2==1 && ss[i]>='a' && ss[i]<='z')
            【1】
}

```

```

main()
{
    【2】 *wf;
    char tt[81],s[10]="abc4Efg" 【3】
    printf("\nPlease enter an string within 80 characters:\n");
    gets(tt);
    printf("\n\nAfter changing, the string\n%s",tt);
    fun(tt);
    printf("\nbecomes\n%s\n",tt);
    wf=fopen("out.dat","w");
    fun(s);
    fprintf(wf,"%s",s);
    fclose(wf);
}

```

答案：

【1】 ss[i]=ss[i]-32;

【2】 FILE

【3】;

解析：

【1】从C语言的学习中知道，只要将小写字母减去32则可转换成大写字母，将大写字母加上32则转换成小写字母。

【2】根据主函数中fprintf(wf,"%s",s);和fclose(wf);两条语句的操作知道，wf是一个文件型的指针，所以应该使用关键字FILE对其进行声明。

【3】这里是说明并给字符数组进行初始化，所以应该使用“;”作为语句结束的标志，这是C语言的规定。

## 2. 改错题

下列给定程序中，函数 fun()的功能是：根据整型形参 m，计算如下公式的值。

$$y = 1 - \frac{1}{2 \times 2} + \frac{1}{3 \times 3} - \frac{1}{4 \times 4} + K + \frac{(-1)^{m+1}}{m \times m}$$

例如：若 m 中的值为 5，则应输出 0.838611。

请改正程序中的错误，使它能得到正确结果。

注意：不要改动main函数，不得增行或删行，也不得更改程序的结构。

```

#include <conio.h>
#include <stdio.h>
double fun(int m)
{
    double y=1.0;
    /*****found error*****/
    int j=0, i;

```

```

    for(i=2; i<=m; i++)
    {
        j=-j;
        /*****found  error*****/
        y+=j/(i * i);
    }
return(y);
}
main()
{
    int  n=5;
    clrscr();
    printf("\nThe result is %lf\n",fun(n));
}

```

答案：

- (1) 错误：int j=0;                      正确：int j=1;  
 (2) 错误：y+=j/(i \* i);              正确：y+=j/(float)(i \* i);

解析：

错误 1：因为题中公式是加、减相间的运算，通过 j 来实现，所以这里 j 应赋初值为 1。

错误 2：因为 j, i 的数据类型都是整型，而 y 是双精度型，因此做除法运算时要进行类型转换。

### 3. 编程题

请编写函数 fun，其功能是：将两个两位数的正整数 a, b 合并形成一个整数，放在 c 中。合并的方式是：将 a 数的十位和个位数依次放在 c 数个位和十位上，b 数的十位和个位数依次放在 c 数的百位和千位上。例如，当 a=16, b=35，调用该函数后，c=5361。

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的花括号中填入所编写的若干语句。

```

#include<stdio.h>
void fun(int a,int b,long *c)
{ /*编写程序代码*/

```

```

}
main()
{
    int a,b;
    long  c;
    clrscr();
    printf("Input a,b;");

```

```
scanf("%d%d",&a, &b);
fun(a,b,&c);
printf("The result is:%ld\n",c);
}
```

答案：

```
*c=(b%10)*1000+(b/10)*100+(a%10)*10+a/10;
```

解析：

语句 $*c=(b\%10)*1000+(b/10)*100+(a\%10)*10+a/10$ ;是将 a 数的十位和个位数依次放在 c 数个位和十位上, b 数的十位和个位数依次放在 c 数的百位和千位上。

注意：“/”和“%”的用法（“/”是整除，求除后的整数；“%”是求除后的余数）。

## 上机测试题八

### 1. 填空题

下列给定程序中，函数 fun()的功能是：将 s 所指字符串中的字母转换为按字母序列的后续字母(但 Z 转化为 A, z 转化为 a)，其他字符不变。

请勿改动主函数 main 和其他函数中的任何内容，仅在填空处填入所编写的语句。

```
#include <stdio.h>
#include <ctype.h>
#include <conio.h>
void fun(char *s)
{
    while(【1】)
    {
        if(*s>='A' && *s<='Z' || *s>='a' && *s<='z')
        {
            if(*s=='Z')
                *s='A';
            else if(*s=='z')
                *s='a';
            else
                *s+=【2】;
        }
        【3】;
    }
}
main()
{
    char s[80];
```

```

printf("\n Enter a string with length<80:\n\n");
gets (s);
printf("\n The string:\n\n");
puts(s);
fun(s);
printf("\n\n The Cords : \n\n");
puts(s);
}

```

答案：

【1】\*s

【2】1

【3】s++

解析：

【1】在 C 语言中，while 语句的结束条件是括号中的表达式结果为 0，而一个字符串的结尾标志为'\0'，即满足循环的结束条件，所以一般情况下，C 语言中使用字符串的结尾作为结束循环的标志；s 为指向一个数组的指针，\*s 为取出 s 指向的内容。

【2】根据题意，若该字母不是'z'或'Z'，则将该字母的 ASCII 码值加 1。

【3】根据题意，要对字符串中所有字母进行遍历，若使用指针进行这一过程则应该使用循环，即每次都对该指针进行加 1 操作。

## 2. 改错题

下列给定程序中，函数 fun 的功能是：按以下递归公式求函数值。

$$\text{Fun}(n) = \begin{cases} 15 \\ \text{Fun}(n-1) \times 2 \end{cases}$$

例如：当给 n 输入 5 时，函数值为 240；当给 n 输入 3 时，函数值为 60。

请改正程序中的错误，使它能得到正确结果。

注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构。

```

#include    <stdio.h>

/*****found  error*****/
fun(int n);
{
    int  c;

/*****found  error*****/

    if(n=1)
        c=15;
    else
        c=fun(n-1)*2;
    return(c);
}

```



```
main()
{
    int n;
    printf("Enter n:");
    scanf("%d",&n);
    printf("The result :%d\n\n",fun(n));
}
```

答案：

(1) 错误：fun(int n);      正确：fun(int n)

(2) 错误：if(n=1)      正确：if(n==1)

解析：

错误 1：该行是函数的首部，不是一条语句，因此不能以分号结束。

错误 2：if 后面应该紧跟一个条件判断表达式，若写成 if(n=1)，则说明该表达式的值永远为真，因为 n=1 是赋值语句，也就是说这个条件永远成立，没有递归执行。

### 3. 编程题

请编写函数 fun()，对长度为 7 个字符的字符串，除首、尾字符外，将其余 5 个字符按 ASCII 码的值升序排列。例如：原来的字符串为 BdsihAd，则排序后输出为 BAdhisd。

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的花括号中填入所编写的若干语句。

```
#include <stdio.h>
#include <ctype.h>
#include <conio.h>
int fun(char *s, int num)
{ /*编写程序代码*/
```

```

}
main()
{
    char s[10];
    clrscr();
    printf("输入 7 个字符的字符串:");
    gets(s);
    fun(s,7);
    printf("\n%s",s);
}
```

答案：

```
char t;
```

```
int i, j;
for(i=1; i<num-2; i++)          /*下标值从 1 开始,用循环依次取得字符串中的字符*/
    for(j=i+1; j<num-1; j++)    /*将字符与其后的每个字符比较*/
        if(s[i]>s[j])           /*如果后面字符的 ASCII 码值小于该字符的 ASCII 码值*/
            { t=s[i];          /*则交换这两个字符*/
              s[i]=s[j];
              s[j]=t;
            }
```

解析：

本题利用循环来控制数组元素的操作，首、尾元素除外，因此，数组的下标值要从 1 开始，用循环变量 i 来依次取得数组中的元素，用数组中的元素 s[i]和 s[j]进行比较，如果后一个元素的 ASCII 码值小于前一个元素的 ASCII 码值，则交换这两个数组元素的值。

## 附录四 ASCII 码表

ASC 值	字符	ASC 值	字符	ASC 值	字符	ASC 值	字符
000	NUL	032	空格符	064	@	096	.
001	SOH	033	!	065	A	097	a
002	STX	034	"	066	B	098	b
003	ETX	035	#	067	B	099	c
004	EOT	036	\$	068	D	100	d
005	ENQ	037	%	069	E	101	e
006	ACK	038	&	070	F	102	f
007	BEL	039	'	071	G	103	g
008	BS	040	(	072	H	104	h
009	HT	041	)	073	I	105	i
010	LF	042	*	074	J	106	j
011	VT	043	+	075	K	107	k
012	FF	044	,	076	L	108	l
013	CR	045	-	077	M	109	m
014	SO	046	.	078	N	110	n
015	SI	047	/	079	O	111	o
016	DLE	048	0	080	P	112	p
017	DC1	049	1	081	Q	113	q
018	DC2	050	2	082	R	114	r
019	DC3	051	3	083	S	115	s
020	DC4	052	4	084	T	116	t
021	NAK	053	5	085	U	117	u
022	SYN	054	6	086	V	118	v
023	STB	055	7	087	W	119	w
024	CAN	056	8	088	X	120	x
025	EM	057	9	089	Y	121	y
026	SUB	058	:	090	Z	122	z
027	ESO	059	;	091	[	123	{
028	FS	060	<	092	\	124	
029	GS	061	=	093	]	125	}
030	RS	062	>	094	^	126	~
031	US	063	?	095	_	127	DEL