

语
景
人
全

C 语言程序设计

目录

1. [c 语言概述](#)
2. [程序的灵魂—算法](#)
3. [数据类型、运算符与表达式](#)
4. [最简单的 C 程序设计—顺序程序设计](#)
5. [分支结构程序](#)
6. [循环控制](#)
7. [数组](#)
8. [函数](#)
9. [预处理命令](#)
10. [指针](#)
11. [结构体与共用体](#)
12. [位运算](#)
13. [文件](#)

参考资料

1. [turbo c2.0 函数大全](#)

[a](#) [b](#) [c](#) [d](#) [e](#) [f](#) [g](#) [h](#) [i](#) [k](#) [l](#) [m](#) [n](#) [o](#) [p](#) [q](#) [r](#)
[s](#) [t](#) [u](#) [v](#) [w](#)

2. [turbo c2.0 字符屏幕函数](#)

3. [turbo c2.0 图形函数](#)

实例

1. [c 语言 100 例](#)
2. [南开 c 语言上机 100 题](#)
3. [南开 100 题二级 C](#)
4. [二级 c 语言试题及答案](#)

C 语言教程

1 C 语言概述

1.1 C 语言的发展过程

C 语言是在 70 年代初问世的。一九七八年由美国电话电报公司(AT&T)贝尔实验室正式发表了 C 语言。同时由 B.W.Kernighan 和 D.M.Ritchie 合著了著名的“THE C PROGRAMMING LANGUAGE”一书。通常简称为《K&R》，也有人称之为《K&R》标准。但是，在《K&R》中并没有定义一个完整的标准 C 语言，后来由美国国家标准协会(American National Standards Institute)在此基础上制定了一个 C 语言标准，于一九八三年发表。通常称之为 ANSI C。

1.2 当代最优秀的程序设计语言

早期的 C 语言主要是用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐为人们认识，到了八十年代，C 开始进入其它操作系统，并很快在各类大、中、小和微型计算机上得到了广泛的使用，成为当代最优秀的程序设计语言之一。

1.3 C 语言版本

目前最流行的 C 语言有以下几种：

- Microsoft C 或称 MS C
- Borland Turbo C 或称 Turbo C
- AT&T C

这些 C 语言版本不仅实现了 ANSI C 标准，而且在此基础上各自作了一些扩充，使之更加方便、完美。

1.4 C 语言的特点

1 • C 语言简洁、紧凑，使用方便、灵活。ANSI C 一共只有 32 个关键字：

auto	break	case	char	const	continue	default
do	double	else	enum	extern	float	for
goto	if	int	long	register	return	short
signed	static	sizeof	struct	switch	typedef	union
unsigned	void	volatile	while			

9 种控制语句，程序书写自由，主要用小写字母表示，压缩了一切不必要的成分。
Turbo C 扩充了 11 个关键字：

asm _cs _ds _es _ss cdecl far
huge interrupt near pascal

注意：在 C 语言中，关键字都是小写的。

2• 运算符丰富。共有 34 种。C 把括号、赋值、逗号等都作为运算符处理。从而使 C 的运算类型极为丰富，可以实现其他高级语言难以实现的运算。

3• 数据结构类型丰富。

4• 具有结构化的控制语句。

5• 语法限制不太严格，程序设计自由度大。

6• C 语言允许直接访问物理地址，能进行位（bit）操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。因此有人把它称为中级语言。

7• 生成目标代码质量高，程序执行效率高。

8• 与汇编语言相比，用 C 语言写的程序可移植性好。

但是，C 语言对程序员要求也高，程序员用 C 写程序会感到限制少、灵活性大，功能强，但较其他高级语言在学习上要困难一些。

1.5 面向对象的程序设计语言

在 C 的基础上，一九八三年又由贝尔实验室的 Bjarne Stroustrup 推出了 C++。C++ 进一步扩充和完善了 C 语言，成为一种面向对象的程序设计语言。C++ 目前流行的最新版本是 Borland C++, Symantec C++ 和 Microsoft VisualC++。

C++ 提出了一些更为深入的概念，它所支持的这些面向对象的概念容易将问题空间直接地映射到程序空间，为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法。因而也增加了整个语言的复杂性，掌握起来有一定难度。

1.6 C 和 C++

但是，C 是 C++ 的基础，C++ 语言和 C 语言在很多方面是兼容的。因此，掌握了 C 语言，再进一步学习 C++ 就能以一种熟悉的语法来学习面向对象的语言，从而达到事半功倍的目的。

1.7 简单的 C 程序介绍

为了说明 C 语言源程序结构的特点，先看以下几个程序。这几个程序由简到难，表现了 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍，但可从这些例子中了解到组成一个 C 源程序的基本部分和书写格式。

【例 1.1】

```
main()
{
    printf("世界，您好！\n");
}
```

- main 是主函数的函数名，表示这是一个主函数。
- 每一个 C 源程序都必须有，且只能有一个主函数(main 函数)。

- 函数调用语句，printf 函数的功能是把要输出的内容送到显示器去显示。
- printf 函数是一个由系统定义的标准函数，可在程序中直接调用。

【例 1.2】

```
#include<math.h>
#include<stdio.h>
main()
{
    double x,s;
    printf("input number:\n");
    scanf("%lf",&x);
    s=sin(x);
    printf("sine of %lf is %lf\n",x,s);
}
```

- include 称为文件包含命令
- 扩展名为.h 的文件称为头文件
- 定义两个实数变量，以被后面程序使用
- 显示提示信息
- 从键盘获得一个实数 x
- 求 x 的正弦,并把它赋给变量 s
- 显示程序运算结果
- main 函数结束

程序的功能是从键盘输入一个数 x，求 x 的正弦值，然后输出结果。在 main()之前的两行称为预处理命令(详见后面)。预处理命令还有其它几种，这里的 include 称为文件包含命令，其意义是把尖括号<>或引号""内指定的文件包含到本程序来，成为本程序的一部分。被包含的文件通常是由系统提供的，其扩展名为.h。因此也称为头文件或首部文件。C 语言的头文件中包括了各个标准库函数的函数原型。因此，凡是在程序中调用一个库函数时，都必须包含该函数原型所在的头文件。在本例中，使用了三个库函数：输入函数 scanf，正弦函数 sin,输出函数 printf。sin 函数是数学函数，其头文件为 math.h 文件，因此在程序的主函数前用 include 命令包含了 math.h。scanf 和 printf 是标准输入输出函数，其头文件为 stdio.h，在主函数前也用 include 命令包含了 stdio.h 文件。

需要说明的是，C 语言规定对 scanf 和 printf 这两个函数可以省去对其头文件的包含命令。所以在本例中也可以删去第二行的包含命令#include<stdio.h>。

同样，在例 1.1 中使用了 printf 函数，也省略了包含命令。

在例题中的主函数体中又分为两部分，一部分为说明部分，另一部为分执行部分。说明是指变量的类型说明。例题 1.1 中未使用任何变量，因此无说明部分。C 语言规定，源程序中所有用到的变量都必须先说明，后使用，否则将会出错。这一点是编译型高级程序设计语言的一个特点，与解释型的 BASIC 语言是不同的。说明部分是 C 源程序结构中很重要的组成部分。本例中使用了两个变量 x，s，用来表示输入的自变量和 sin 函数值。由于 sin 函数要求这两个量必须是双精度浮点型，故用类型说明符 double 来说明这两个变量。说明部分后的四行为执行部分或称为执行语句部分，用以完成程序的功能。执行部分的第一行是输出语句，调用 printf 函数在显示器上输出提示字符串，请操作人员输入自变量 x 的值。第二行为输入语句，调用 scanf 函数，接受键盘上输入的数并存入变量 x 中。第三行是调用 sin 函数并把函数值送到变量 s 中。第四行是用 printf 函数输出变量 s 的值，即 x 的正弦值。程序结束。

运行本程序时，首先在显示器屏幕上给出提示串 `input number`，这是由执行部分的第一行完成的。用户在提示下从键盘上键入某一数，如 5，按下回车键，接着在屏幕上给出计算结果。

1.8 输入和输出函数

在前两个例子中用到了输入和输出函数 `scanf` 和 `printf`，在以后要详细介绍。这里我们先简单介绍一下它们的格式，以便下面使用。

`scanf` 和 `printf` 这两个函数分别称为格式输入函数和格式输出函数。其意义是按指定的格式输入输出值。因此，这两个函数在括号中的参数表都由以下两部分组成：

“格式控制串”，参数表

格式控制串是一个字符串，必须用双引号括起来，它表示了输入输出量的数据类型。各种类型的格式表示法可参阅第三章。在 `printf` 函数中还可以在格式控制串内出现非格式控制字符，这时在显示屏幕上将原文照印。参数表中给出了输入或输出的量。当有多个量时，用逗号间隔。例如：

```
printf("sine of %lf is %lf\n",x,s);
```

其中 `%lf` 为格式字符，表示按双精度浮点数处理。它在格式串中两次出现，对应了 `x` 和 `s` 两个变量。其余字符为非格式字符则照原样输出在屏幕上。

【例 1.3】

```
main()                /*主函数*/
{
    int x,y,z;          /*变量说明*/
    int max(int a,int b); /*函数说明*/
    printf("input two numbers:\n");
    scanf("%d%d",&x,&y); /*输入 x,y 值*/
    z=max(x,y);          /*调用 max 函数*/
    printf("maximum=%d",z); /*输出*/
}
int max(int a,int b)    /*定义 max 函数*/
{
    if(a>b)return a;else return b; /*把结果返回主调函数*/
}
```

上面例中程序的功能是由用户输入两个整数，程序执行后输出其中较大的数。本程序由两个函数组成，主函数和 `max` 函数。函数之间是并列关系。可从主函数中调用其它函数。`max` 函数的功能是比较两个数，然后把较大的数返回给主函数。`max` 函数是一个用户自定义函数。因此在主函数中要给出说明(程序第三行)。可见，在程序的说明部分中，不仅可以有变量说明，还可以有函数说明。关于函数的详细内容将在以后第五章介绍。在程序的每行后用 `/*` 和 `*/` 括起来的内容为注释部分，程序不执行注释部分。

上例中程序的执行过程是，首先在屏幕上显示提示串，请用户输入两个数，回车后由 `scanf` 函数语句接收这两个数送入变量 `x,y` 中，然后调用 `max` 函数，并把 `x,y` 的值传送给 `max` 函数的参数 `a,b`。在 `max` 函数中比较 `a,b` 的大小，把大者返回给主函数的变量 `z`，最后在屏幕上输出 `z` 的值。

1.9 C 源程序的结构特点

- 1.一个C语言源程序可以由一个或多个源文件组成。
- 2.每个源文件可由一个或多个函数组成。
- 3.一个源程序不论由多少个文件组成，都有一个且只能有一个 `main` 函数，即主函数。
- 4.源程序中可以有预处理命令(`include` 命令仅为其中的一种)，预处理命令通常应放在源文件或源程序的最前面。
- 5.每一个说明，每一个语句都必须以分号结尾。但预处理命令，函数头和花括号 “`{}`” 之后不能加分号。
- 6.标识符，关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符，也可不再加空格来间隔。

1.10 书写程序时应遵循的规则

从书写清晰，便于阅读，理解，维护的角度出发，在书写程序时应遵循以下规则：

- 1.一个说明或一个语句占一行。
 - 2.用 `{}` 括起来的部分，通常表示了程序的某一层结构。`{}` 一般与该结构语句的第一个字母对齐，并单独占一行。
 - 3.低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写。以便看起来更加清晰，增加程序的可读性。
- 在编程时应力求遵循这些规则，以养成良好的编程风格。

1.11 C 语言的字符集

字符是组成语言的最基本的元素。C 语言字符集由字母，数字，空格，标点和特殊字符组成。在字符常量，字符串常量和注释中还可以使用汉字或其它可表示的图形符号。

1.字母

小写字母 `a~z` 共 26 个

大写字母 `A~Z` 共 26 个

2.数字

`0~9` 共 10 个

3.空白符

空格符、制表符、换行符等统称为空白符。空白符只在字符常量和字符串常量中起作用。在其它地方出现时，只起间隔作用，编译程序对它们忽略不计。因此在程序中使用空白符与否，对程序的编译不发生影响，但在程序中适当的地方使用空白符将增加程序的清晰性和可读性。

4.标点和特殊字符

1.12 C 语言词汇

在 C 语言中使用的词汇分为六类：标识符，关键字，运算符，分隔符，常量，注释符等。

1.标识符

在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名由系统定义外，其余都由用户自定义。C 规定，标识符只能是字母(A~Z, a~z)、数字(0~9)、下划线(_)组成的字符串，并且其第一个字符必须是字母或下划线。

以下标识符是合法的：

a, x, x3, BOOK_1, sum5

以下标识符是非法的：

3s 以数字开头

s*T 出现非法字符*

-3x 以减号开头

bowy-1 出现非法字符-(减号)

在使用标识符时还必须注意以下几点：

(1)标准 C 不限制标识符的长度，但它受各种版本的 C 语言编译系统限制，同时也受到具体机器的限制。例如在某版本 C 中规定标识符前八位有效，当两个标识符前八位相同时，则被认为是同一个标识符。

(2)在标识符中，大小写是有区别的。例如 BOOK 和 book 是两个不同的标识符。

(3)标识符虽然可由程序员随意定义，但标识符是用于标识某个量的符号。因此，命名应尽量有相应的意义，以便于阅读理解，作到“顾名思义”。

2.关键字

关键字是由 C 语言规定的具有特定意义的字符串，通常也称为保留字。用户定义的标识符不应与关键字相同。C 语言的关键字分为以下几类：

(1)类型说明符

用于定义、说明变量、函数或其它数据结构的类型。如前面例题中用到的 int, double 等

(2)语句定义符

用于表示一个语句的功能。如例 1.3 中用到的 if else 就是条件语句的语句定义符。

(3)预处理命令字

用于表示一个预处理命令。如前面各例中用到的 include。

3.运算符

C 语言中含有相当丰富的运算符。运算符与变量，函数一起组成表达式，表示各种运算功能。运算符由一个或多个字符组成。

4.分隔符

在 C 语言中采用的分隔符有逗号和空格两种。逗号主要用在类型说明和函数参数表中，分隔各个变量。空格多用于语句各单词之间，作间隔符。在关键字，标识符之间必须要有一个以上的空格符作间隔，否则将会出现语法错误，例如把 int a;写成 inta;C 编译器会把 inta 当成一个标识符处理，其结果必然出错。

5.常量

C 语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量、转义字符等多种。在后面章节中将专门给予介绍。

6.注释符

C 语言的注释符是以“/*”开头并以“*/”结尾的串。在“/*”和“*/”之间的即为注释。程序编译时，不对注释作任何处理。注释可出现在程序中的任何位置。注释用来向用户提示或解释程序的意义。在调试程序中对暂不使用的语句也可用注释符括起来，使翻译跳过不作处理，待调试结束后再去掉注释符。

1.13 Turbo C 2.0 集成开发环境的使用

1.13.1 Turbo C 2.0 简介和启动

我们上机实习和将来考试都是使用 Borland Turbo C 2.0 这个版本。该系统是 DOS 操作系统支持下的软件，在 windows 98 环境下，可以在 DOS 窗口下运行。

我们机房是在 D 盘根目录下建立一个 TC 子目录下安装 Turbo C 2.0 系统的。TC 下还建立了两个子目录 LIB 和 INCLUDE, LIB 子目录中存放库文件, INCLUDE 子目录中存放所有头文件。

在 DOS 环境下或在 windows 98 的 DOS 窗口下运行 Turbo C 2.0 时，只要在 TC 子目录下键入 TC 并回车即可进入 Turbo C 2.0 集成开发环境。

在 windows 98 环境下，也可以选运行菜单，然后键入 d:\tc\tc 即可，也可以在 tc 文件夹找到 tc.exe 文件，然后用鼠标双击该文件名也可进入 Turbo C 2.0 集成开发环境。

Turbo C 是美国 Borland 公司的产品, Borland 公司是一家专门从事软件开发、研制的大公司。该公司相继推出了一套 Turbo 系列软件, 如 Turbo BASIC, Turbo Pascal, Turbo Prolog, 这些软件很受用户欢迎。该公司在 1987 年首次推出 TurboC 1.0 产品, 其中使用了全然一新的集成开发环境, 即使用了一系列下拉式菜单, 将文本编辑、程序编译、连接以及程序运行一体化, 大大方便了程序的开发。1988 年, Borland 公司又推出 Turbo C1.5 版本, 增加了图形库和文本窗口函数库等, 而 Turbo C 2.0 则是该公司 1989 年出版的。Turbo C2.0 在原来集成开发环境的基础上增加了查错功能, 并可以在 Tiny 模式下直接生成 .COM (数据、代码、堆栈处在同一 64K 内存中) 文件。还可对数学协处理器 (支持 8087/80287/80387 等) 进行仿真。

Borland 公司后来又推出了面向对象的程序软件包 Turbo C++, 它继承发展 Turbo C 2.0 的集成开发环境, 并包含了面向对象的基本思想和设计方法。1991 年为了适用 Microsoft 公司的 Windows 3.0 版本, Borland 公司又将 Turbo C++ 作了更新, 即 Turbo C 的新一代产品 Borland C++ 也已经问世了。

1.13.2 Turbo C 2.0 集成开发环境

进入 Turbo C 2.0 集成开发环境中后, 屏幕上显示:



其中项上一行为 Turbo C 2.0 主菜单，中间窗口为编辑区，接下来是信息窗口，最底下一行为参考行。这四个窗口构成了 Turbo C 2.0 的主屏幕，以后的编辑、编译、调试以及运行都将在这个主屏幕中进行。

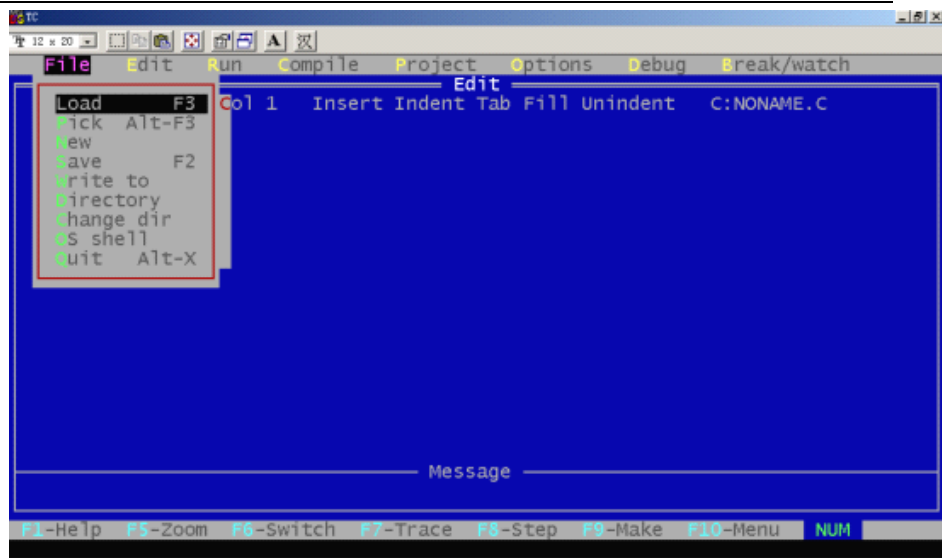
主菜单在 Turbo C 2.0 主屏幕项上一行，显示下列内容：

File Edit Run Compile Project Options Debug Break/watch

除 Edit 外，其它各项均有子菜单，只要用 Alt 加上某项中第一个字母，就可进入该项的子菜单中。

1.13.3 File 菜单

按 Alt+F 可进入 File 菜单，如图：



File 菜单的子菜单共有 9 项，分别叙述如下：

1. Load：装入一个文件，可用类似 DOS 的通配符(如*.C)来进行列表选择。也可装入其它

扩展名的文件, 只要给出文件名(或只给路径)即可。该项的热键为 F3, 即只要按 F3 即可进入该项, 而不需要先进入 File 菜单再选此项。

2. **Pick:** 将最近装入编辑窗口的 8 个文件列成一个表让用户选择, 选择后将该程序装入编辑区, 并将光标置在上次修改过的地方。其热键为 Alt-F3。
3. **New:** 新建文件, 缺省文件名为 NONAME.C, 存盘时可改名。
4. **Save:** 将编辑区中的文件存盘, 若文件名是 NONAME.C 时, 将询问是否更改文件名, 其热键为 F2。
5. **Write to:** 可由用户给出文件名将编辑区中的文件存盘, 若该文件已存在, 则询问要不要覆盖。
6. **Directory:** 显示目录及目录中的文件, 并可由用户选择。
7. **Change dir:** 显示当前默认目录, 用户可以改变默认目录。
8. **Os shell:** 暂时退出 Turbo C 2.0 到 DOS 提示符下, 此时可以运行 DOS 命令, 若想回到 Turbo C 2.0 中, 只要在 DOS 状态下键入 EXIT 即可。
9. **Quit:** 退出 Turbo C 2.0, 返回到 DOS 操作系统中, 其热键为 Alt+X。

说明:

以上各项可用光标键移动色棒进行选择, 回车则执行。也可用每一项的第一个大写字母直接选择。若要退到主菜单或从它的下一级菜单列表框退回均可用 Esc 键, Turbo C 2.0 所有菜单均采用这种方法进行操作, 以下不再说明。

1.13.4 Edit 菜单

按 Alt+E 可进入编辑菜单, 若再回车, 则光标出现在编辑窗口, 此时用户可以 进行文本编辑。编辑方法基本与 wordstar 相同, 可用 F1 键获得有关编辑方法的帮助信息。

1. 与编辑有关的功能键如下:

F1 获得 Turbo C 2.0 编辑命令的帮助信息;
F5 扩大编辑窗口到整个屏幕;
F6 在编辑窗口与信息窗口之间进行切换;
F10 从编辑窗口转到主菜单。

2. 编辑命令简介:

PageUp 向前翻页
PageDn 向后翻页
Home 将光标移到所在行的开始
End 将光标移到所在行的结尾
Ctrl+Y 删除光标所在的一行
Ctrl+T 删除光标所在处的一个词 (从光标上的字符开始)
Ctrl+KB 设置块开始
Ctrl+KK 设置块结尾
Ctrl+KV 块移动
Ctrl+KC 块拷贝
Ctrl+KY 块删除
Ctrl+KR 读文件
Ctrl+KW 存文件
Ctrl+KP 块文件打印
Ctrl+F1 如果光标所在处为 Turbo C 2.0 库函数, 则获得有关该函数的帮助信息

Ctrl+Q[查找 Turbo C 2.0 双界符的后匹配符

Ctrl+Q] 查找 Turbo C 2.0 双界符的前匹配符

说明:

- 1) Turbo C 2.0 的双界符包括以下几种符号:
 - a) 花括号 {和}
 - b) 尖括号 <和>
 - c) 圆括号 (和)
 - d) 方括号 [和]
 - e) 注释符 /*和*/
 - f) 双引号 "
 - g) 单引号 '
- 2) Turbo C 2.0 在编辑文件时还有一种功能, 就是能够自动缩进, 即光标定位和上一个非空字符对齐。在编辑窗口中, Ctrl+OI 为自动缩进开关的控制键。

1.13.5 Run 菜单

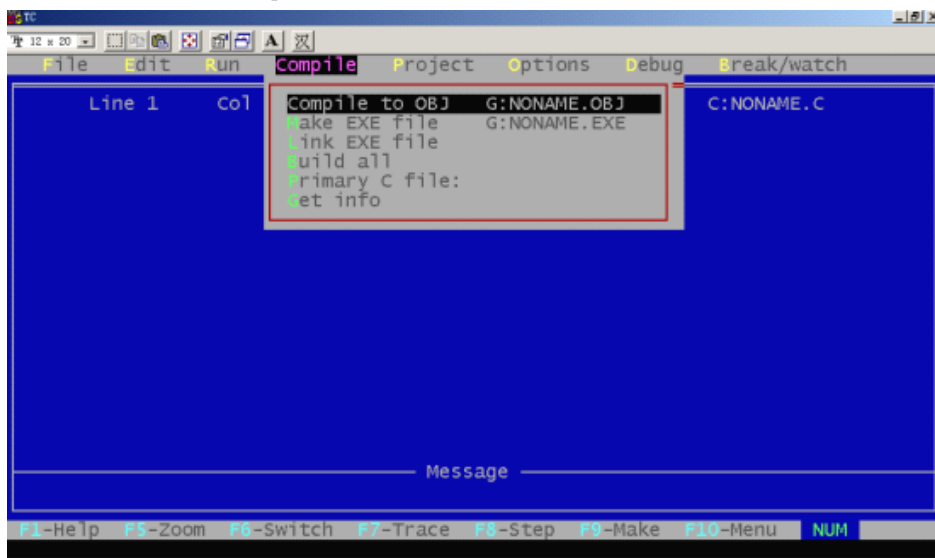
按 Alt+R 可进入 Run 菜单, 该菜单有以下各项, 如图所示:



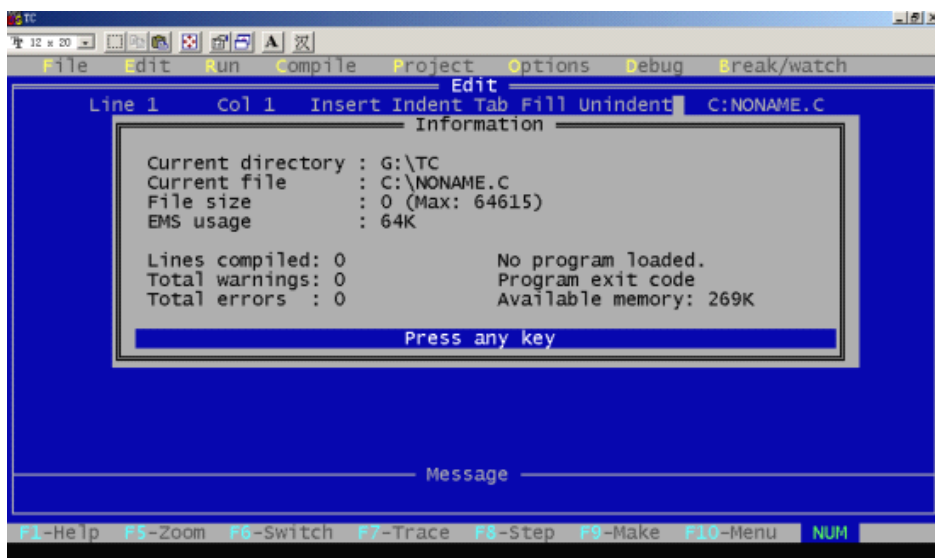
1. Run: 运行由 Project/Project name 项指定的文件名或当前编辑区的文件。如果对上次编译后的源代码未做过修改, 则直接运行到下一个断点(没有断点则运行到结束)。否则先进行编译、连接后才运行, 其热键为 Ctrl+F9。
2. Program reset: 中止当前的调试, 释放分给程序的空间, 其热键为 Ctrl+F2。
3. Go to cursor: 调试程序时使用, 选择该项可使程序运行到光标所在行。光标所在行必须为一条可执行语句, 否则提示错误。其热键为 F4。
4. Trace into: 在执行一条调用其它用户定义的子函数时, 若用 Trace into 项, 则执行长条将跟踪到该子函数内部去执行, 其热键为 F7。
5. Step over: 执行当前函数的下一条语句, 即使用户函数调用, 执行长条也不会跟踪进函数内部, 其热键为 F8。
6. User screen: 显示程序运行时在屏幕上显示的结果。其热键为 Alt+F5。

1.13.6 Compile 菜单

按 Alt+C 可进入 Compile 菜单, 该菜单有以下几个内容, 如图所示:

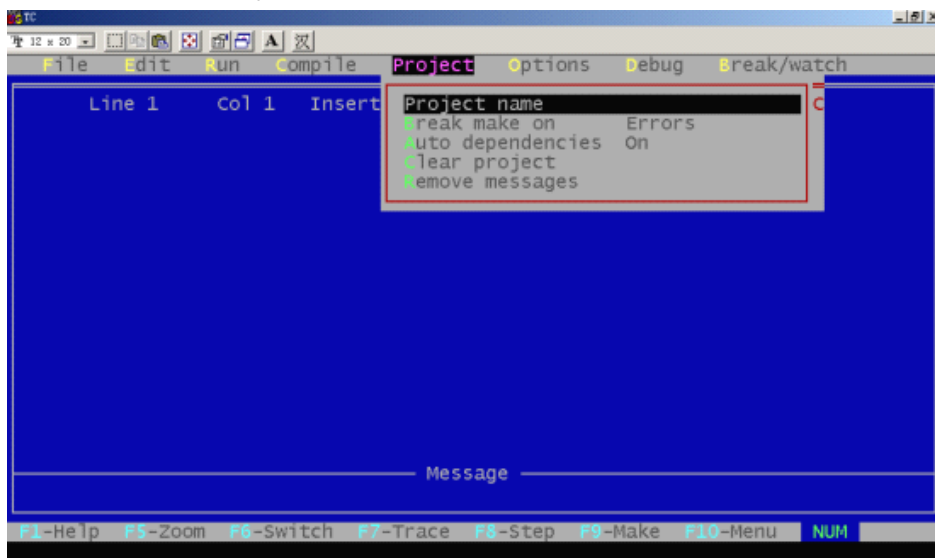


1. **Compile to OBJ**: 将一个 C 源文件编译生成.OBJ 目标文件, 同时显示生成的文件名。其热键为 Alt+F9。
2. **Make EXE file**: 此命令生成一个.EXE 的文件, 并显示生成的.EXE 文件名。其中.EXE 文件名是下面几项之一:
 - 1) 由 Project/Project name 说明的项目文件名。
 - 2) 若没有项目文件名, 则由 Primary C file 说明的源文件。
 - 3) 若以上两项都没有文件名, 则为当前窗口的文件名。
3. **Link EXE file**: 把当前.OBJ 文件及库文件连接在一起生成.EXE 文件。
4. **Build all**: 重新编译项目里的所有文件, 并进行装配生成.EXE 文件。该命令不作过时检查(上面的几条命令要作过时检查, 即如果目前项目里源文件的日期和时间与目标文件相同或更早, 则拒绝对源文件进行编译)。
5. **Primary C file**: 当在该项中指定了主文件后, 在以后的编译中, 如没有项目文件名则编译此项中规定的主 C 文件, 如果编译中有错误, 则将此文件调入编辑窗口, 不管目前窗口中是不是主 C 文件。
6. **Get info**: 获得有关当前路径、源文件名、源文件字节大小、编译中的错误数目、可用空间等信息, 如图:



1.13.7 Project 菜单

按 Alt+P 可进入 Project 菜单, 该菜单包括以下内容, 如图所示:



1. Project name: 项目名具有.PRJ的扩展名, 其中包括将要编译、连接的文件名。例如有一个程序由 file1.c, file2.c, file3.c 组成, 要将这 3 个文件编译装配成一个 file.exe 的执行文件, 可以先建立一个 file.prj 的项目文件, 其内容如下:

file1.c

file2.c

file3.c

此时将 file.prj 放入 Project name 项中, 以后进行编译时将自动对项目文件中规定的三个源文件分别进行编译。然后连接成 file.exe 文件。如果其中有些文件已经编译成.OBJ 文件, 而

又没有修改过,可直接写上.OBJ 扩展名。此时将不再编译而只进行连接。

例如:

```
file1.obj  
file2.c  
file3.c
```

将不对 file1.c 进行编译,而直接连接。

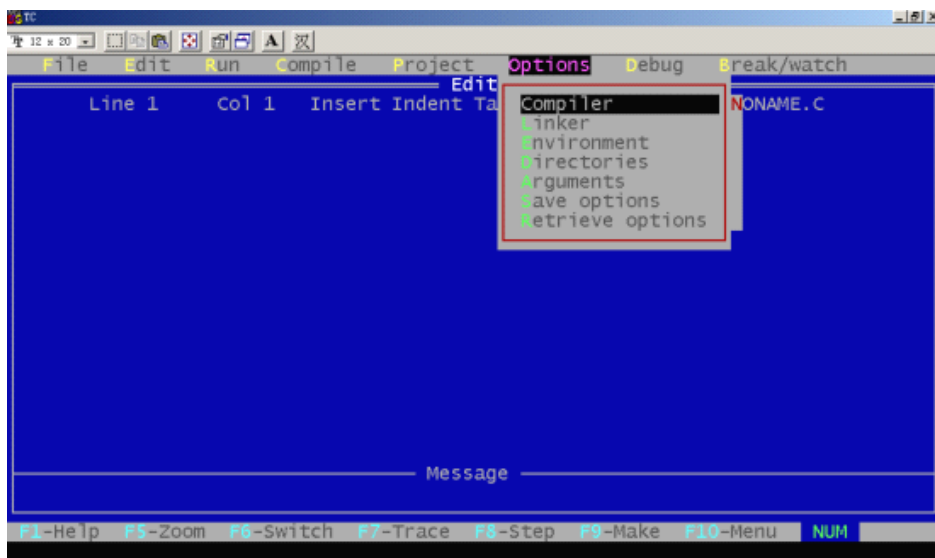
说明:

当项目文件中的每个文件无扩展名时,均按源文件对待,另外,其中的文件也可以是库文件,但必须写上扩展名.LIB。

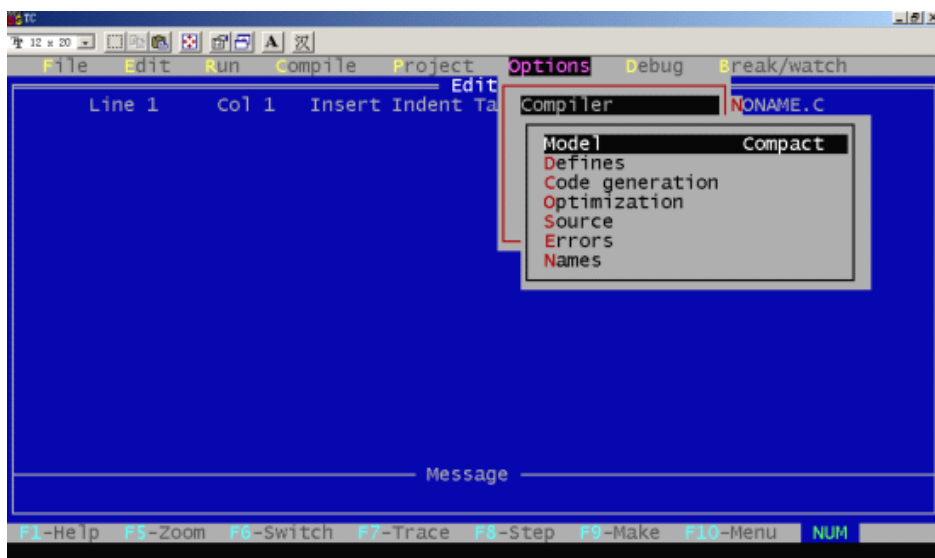
2. Break make on: 由用户选择是否有 Warning、Errors、Fatal Errors 时或 Link 之前退出 Make 编译。
3. Auto dependencies: 当开关置为 on, 编译时将检查源文件与对应的.OBJ 文件日期和时间, 否则不进行检查。
4. Clear project: 清除 Project/Project name 中的项目文件名。
5. Remove messages: 把错误信息从信息窗口中清除掉。

1.13.8 Options 菜单

按 Alt+O 可进入 Options 菜单,该菜单对初学者来说要谨慎使用,该菜单有以下几个内容,如图所示:



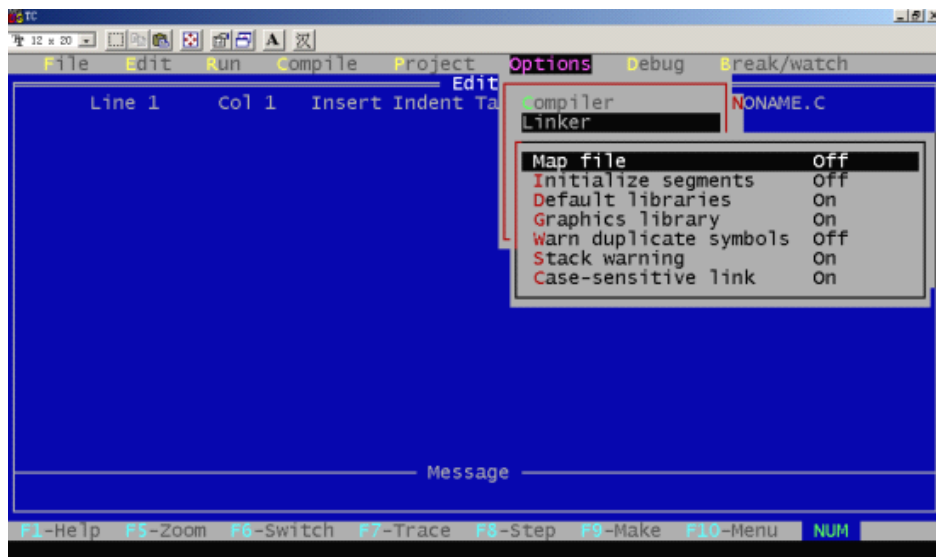
1. Compiler: 本项选择又有许多子菜单,可以让用户选择硬件配置、存储模型、调试技术、代码优化、对话信息控制和宏定义。这些子菜单如图所示:



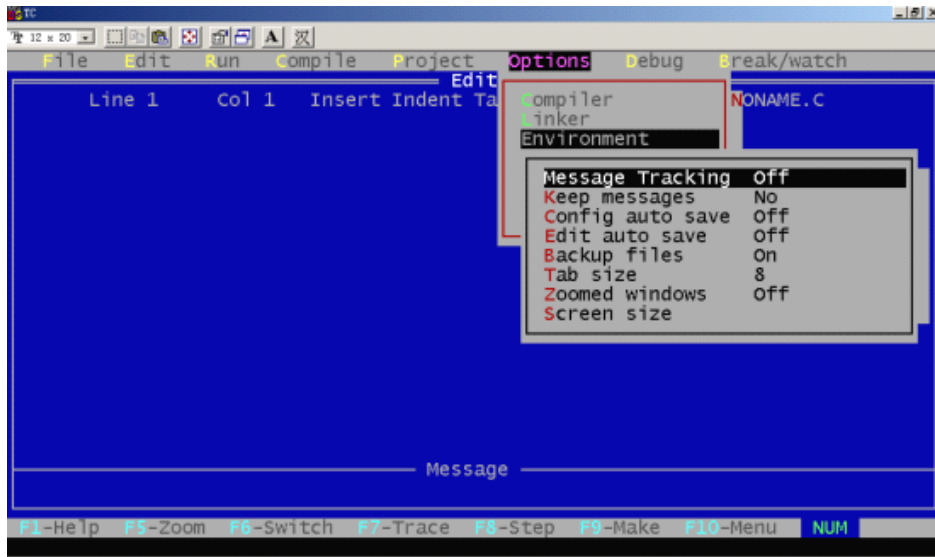
- 1) Model: 共有 Tiny, small, medium, compact, large, huge 六种不同模式可由用户选择。
- 2) Define: 打开一个宏定义框, 用户可输入宏定义。多重定义可用分号, 赋值可用等号。
- 3) Code generation: 它又有许多任选项, 这些任选项告诉编译器产生什么样的目标代码。
 - ✧ Calling convention 可选择 C 或 Pascal 方式传递参数。
 - ✧ Instruction set 可选择 8088/8086 或 80186/80286 指令系列。
 - ✧ Floating point 可选择仿真浮点、数学协处理器浮点或无浮点运算。
 - ✧ Default char type 规定 char 的类型。
 - ✧ Alignment 规定地址对准原则。
 - ✧ Merge duplicate strings 作优化用, 将重复的字符串合并在一起。
 - ✧ Standard stack frame 产生一个标准的栈结构。
 - ✧ Test stack overflow 产生一段程序运行时检测堆栈溢出的代码。
 - ✧ Line number 在 .OBJ 文件中放进行号以供调试时用。
 - ✧ OBJ debug information 在 .OBJ 文件中产生调试信息。
- 4) Optimization: 它又有许多任选项。
 - ✧ Optimize for 选择是对程序小型化还是对程序速度进行优化处理。
 - ✧ Use register variable 用来选择是否允许使用寄存器变量。
 - ✧ Register optimization 尽可能使用寄存器变量以减少过多的取数操作。
 - ✧ Jump optimization 通过去除多余的跳转和调整循环与开关语句的办法, 压缩代码。
- 5) Source: 它又有许多任选项。
 - ✧ Identifier length 说明标识符有效字符的个数, 默认为 32 个。
 - ✧ Nested comments 是否允许嵌套注释。
 - ✧ ANSI keywords only 是只允许 ANSI 关键字还是也允许 Turbo C2.0 关键字。
- 6) Error
 - ✧ Error stop after 多少个错误时停止编译, 默认为 25 个。
 - ✧ Warning stop after 多少个警告错误时停止编译, 默认为 100 个。

- ✧ Display warning
 - ✧ Portability warning 移植性警告错误。
 - ✧ ANSI Violations 侵犯了 ANSI 关键字的警告错误。
 - ✧ Common error 常见的警告错误。
 - ✧ Less common error 少见的警告错误。
- 7) Names : 用于改变段(segment)、组(group) 和类(class)的名字,默认值为 CODE, DATA, BSS。

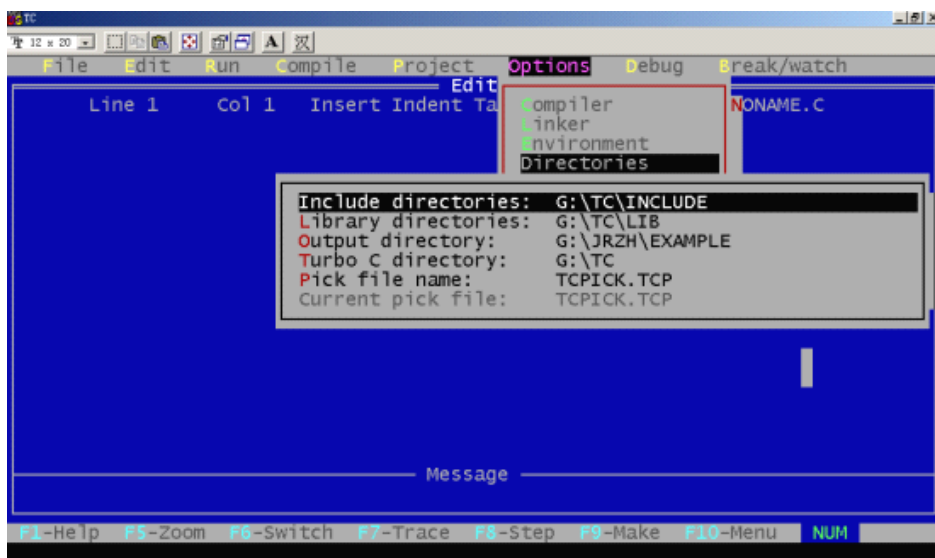
2. Linker: 本菜单设置有关连接的选择项, 它有以下内容, 如图所示:



- 1) Map file menu 选择是否产生.MAP 文件。
 - 2) Initialize segments 是否在连接时初始化没有初始化的段。
 - 3) Devault libraries 是否在连接其它编译程序产生的目标文件时去寻找其缺省库。
 - 4) Graphics library 是否连接 graphics 库中的函数。
 - 5) Warn duplicate symbols 当有重复符号时产生警告信息。
 - 6) Stack warinig 是否让连接程序产生 No stack 的警告信息。
 - 7) Case-sensitive link 是否区分大、小写字。
3. Environment: 菜单规定是否对某些文件自动存盘及制表键和屏幕大小的设置, 它有以下内容, 如图所示:



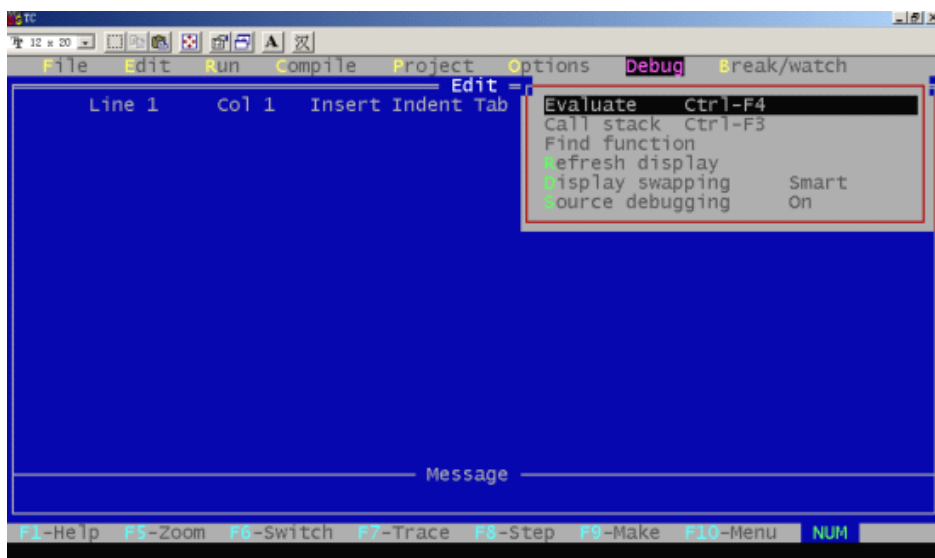
- 1) Message tracking:
 - ✧ Current file 跟踪在编辑窗口中的文件错误。
 - ✧ All files 跟踪所有文件错误。
 - ✧ Off不跟踪。
 - 2) Keep message: 编译前是否清除 Message 窗口中的信息。
 - 3) Config auto save: 选 on 时, 在 Run, Shell 或退出集成开发环境之前,如果 Turbo C 2.0 的配置被改过, 则所做的改动将存入配置文件中。选 off 时不存。
 - 4) Edit auto save: 是否在 Run 或 Shell 之前, 自动存储编辑的源文件。
 - 5) Backup file: 是否在源文件存盘时产生后备文件(.BAK 文件)。
 - 6) Tab size: 设置制表键大小, 默认为 8。
 - 7) Zoomed windows: 将现行活动窗口放大到整个屏幕, 其热键为 F5。
 - 8) Screen size 设置屏幕文本大小。
4. Directories: 规定编译、连接所需文件的路径, 有下列各项, 如图所示:



- (1) Include directories: 包含文件的路径, 多个子目录用";"分开。
 - (2) Library directories: 库文件路径, 多个子目录用";"分开。
 - (3) Output directory: 输出文件(.OBJ, .EXE, .MAP 文件)的目录。
 - (4) Turbo C directory: Turbo C 所在的目录。
 - (5) Pick file name: 定义加载的 pick 文件名, 如不定义则从 currentpick file 中取。
5. Arguments: 允许用户使用命令行参数。
 6. Save options: 保存所有选择的编译、连接、调试和项目到配置文件中, 缺省的配置文件为 TCCONFIG.TC。
 7. Retrive options 装入一个配置文件到 TC 中, TC 将使用该文件的选择项。

1.13.9 Debug 菜单

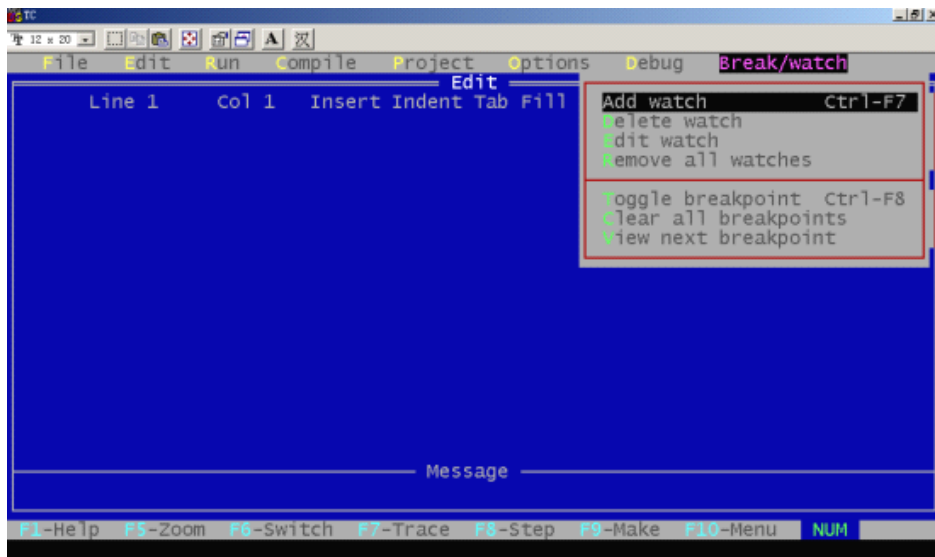
按 Alt+D 可选择 Debug 菜单, 该菜单主要用于查错, 它包括以下内容, 如图所示:



1. Evaluate
 - 1) Expression 要计算结果的表达式。
 - 2) Result 显示表达式的计算结果。
 - 3) New value 赋给新值。
2. Call stack: 该项不可接触。而在 Turbo C debugger 时用于检查堆栈情况。
3. Find function 在运行 Turbo C debugger 时用于显示规定的函数。
4. Refresh display 如果编辑窗口偶然被用户窗口重写了可用此恢复编辑窗口的内容。

1.13.10 Break/watch 菜单

按 Alt+B 可进入 Break/watch 菜单, 该菜单有以下内容, 如图所示:



1. Add watch: 向监视窗口插入一监视表达式。

2. **Delete watch:** 从监视窗口中删除当前的监视表达式。
3. **Edit watch:** 在监视窗口中编辑一个监视表达式。
4. **Remove all : watches** 从监视窗口中删除所有的监视表达式。
5. **Toggle breakpoint:** 对光标所在的行设置或清除断点。
6. **Clear all breakpoints:** 清除所有断点。
7. **View next breakpoint:** 将光标移动到下一个断点处。

1.13.11 Turbo C 2.0 的配置文件

所谓配置文件是包含 Turbo C 2.0 有关信息的文件，其中存有编译、连接的选择和路径等信息。可以用下述方法建立 Turbo C 2.0 的配置：

1. 建立用户自命名的配置文件：可以从 Options 菜单中选择 Options/Save options 命令，将当前集成开发环境的所有配置存入一个由用户命名的配置文件中。下次启动 TC 时只要在 DOS 下键入：
 tc/c<用户命名的配置文件名>
 就会按这个配置文件中的内容作为 Turbo C 2.0 的选择。
2. 若设置 Options/Environment/Config auto save 为 on，则退出集成开发环境时，当前的设置会自动存放到 Turbo C 2.0 配置文件 TCCONFIG.TC 中。Turbo C 在启动时会自动寻找这个配置文件。
3. 用 TCINST 设置 Turbo C 的有关配置，并将结果存入 TC.EXE 中。Turbo C 在启动时，若没有找到配置文件，则取 TC.EXE 中的缺省值。

2	程序的灵魂—算法.....	1
2.1	算法的概念.....	1
2.2	简单算法举例.....	1
2.3	算法的特性.....	4
2.4	怎样表示一个算法.....	4
2.4.1	用自然语言表示算法.....	4
2.4.2	用流程图表示算法.....	4
2.4.3	三种基本结构和改进的流程图.....	8
2.4.4	用 N-S 流程图表示算法.....	9
2.4.5	用伪代码表示算法.....	10
2.4.6	用计算机语言表示算法.....	11
2.5	结构化程序设计方法.....	11

2 程序的灵魂—算法

一个程序应包括：

- 对数据的描述。在程序中要指定数据的类型和数据的组织形式，即数据结构（data structure）。
- 对操作的描述。即操作步骤，也就是算法（algorithm）。

Nikiklaus Wirth 提出的公式：

数据结构+算法=程序

教材认为：

程序=算法+数据结构+程序设计方法+语言工具和环境

这 4 个方面是一个程序涉及人员所应具备的知识。

本课程的目的是使同学知道怎样编写一个 C 程序，进行编写程序的初步训练，因此，只介绍算法的初步知识。

2.1 算法的概念

做任何事情都有一定的步骤。为解决一个问题而采取的方法和步骤，就称为算法。

- 计算机算法：计算机能够执行的算法。
- 计算机算法可分为两大类：
 - 数值运算算法：求解数值；
 - 非数值运算算法：事务管理领域。

2.2 简单算法举例

【例 2.1】求 $1 \times 2 \times 3 \times 4 \times 5$ 。

最原始方法：

步骤 1：先求 1×2 ，得到结果 2。

步骤 2: 将步骤 1 得到的乘积 2 乘以 3, 得到结果 6。

步骤 3: 将 6 再乘以 4, 得 24。

步骤 4: 将 24 再乘以 5, 得 120。

这样的算法虽然正确, 但太繁。

改进的算法:

S1: 使 $t=1$

S2: 使 $i=2$

S3: 使 $t \times i$, 乘积仍然放在在变量 t 中, 可表示为 $t \times i \rightarrow t$

S4: 使 i 的值+1, 即 $i+1 \rightarrow i$

S5: 如果 $i \leq 5$, 返回重新执行步骤 S3 以及其后的 S4 和 S5; 否则, 算法结束。

如果计算 100! 只需将 S5: 若 $i \leq 5$ 改成 $i \leq 100$ 即可。

如果该求 $1 \times 3 \times 5 \times 7 \times 9 \times 11$, 算法也只需做很少的改动:

S1: $1 \rightarrow t$

S2: $3 \rightarrow i$

S3: $t \times i \rightarrow t$

S4: $i+2 \rightarrow i$

S5: 若 $i \leq 11$, 返回 S3, 否则, 结束。

该算法不仅正确, 而且是计算机较好的算法, 因为计算机是高速运算的自动机器, 实现循环轻而易举。

思考: 若将 S5 写成: S5: 若 $i < 11$, 返回 S3; 否则, 结束。

【例 2.2】有 50 个学生, 要求将他们之中成绩在 80 分以上者打印出来。

如果, n 表示学生学号, n_i 表示第 i 个学生学号; g 表示学生成绩, g_i 表示第 i 个学生成绩; 则算法可表示如下:

S1: $1 \rightarrow i$

S2: 如果 $g_i \geq 80$, 则打印 n_i 和 g_i , 否则不打印

S3: $i+1 \rightarrow i$

S4: 若 $i \leq 50$, 返回 S2, 否则, 结束。

【例 2.3】判定 2000 — 2500 年中的每一年是否闰年, 将结果输出。

闰年的条件:

- 1) 能被 4 整除, 但不能被 100 整除的年份;
- 2) 能被 100 整除, 又能被 400 整除的年份;

设 y 为被检测的年份, 则算法可表示如下:

S1: $2000 \rightarrow y$

S2:若 y 不能被 4 整除, 则输出 y “不是闰年”, 然后转到 S6

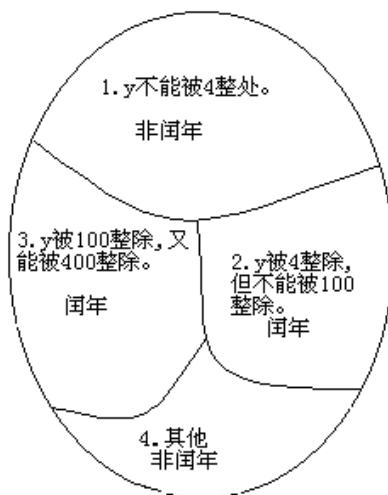
S3:若 y 能被 4 整除, 不能被 100 整除, 则输出 y “是闰年”, 然后转到 S6

S4:若 y 能被 100 整除, 又能被 400 整除, 输出 y “是闰年” 否则输出 y “不是闰年”, 然后转到 S6

S5:输出 y “不是闰年”。

S6: $y+1 \rightarrow y$

S7:当 $y \leq 2500$ 时, 返回 S2 继续执行, 否则, 结束。



【例 2.4】求 $1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots + \frac{1}{99} - \frac{1}{100}$ 。

算法可表示如下:

S1: $sigh=1$

S2: $sum=1$

S3: $deno=2$

S4: $sigh=(-1) \times sigh$

S5: $term = sigh \times (1/deno)$

S6: $term = sum + term$

S7: $deno = deno + 1$

S8:若 $deno \leq 100$, 返回 S4; 否则, 结束。

【例 2.5】对一个大于或等于 3 的正整数, 判断它是不是一个素数。

算法可表示如下:

S1: 输入 n 的值

S2: $i \leftarrow 2$

S3: n 被 i 除, 得余数 r

S4: 如果 $r=0$, 表示 n 能被 i 整除, 则打印 n “不是素数”, 算法结束; 否则执行 S5

S5: $i+1 \rightarrow i$

S6: 如果 $i \leq n-1$, 返回 S3; 否则打印 n “是素数”; 然后算法结束。

改进:

S6: 如果 $i \leq \sqrt{n}$, 返回 S3; 否则打印 n “是素数”; 然后算法结束。

2.3 算法的特性

- 有穷性: 一个算法应包含有限的操作步骤而不能是无限的。
- 确定性: 算法中每一个步骤应当是确定的, 而不能应当是含糊的、模棱两可的。
- 有零个或多个输入。
- 有一个或多个输出。
- 有效性: 算法中每一个步骤应当能有效地执行, 并得到确定的结果。

对于程序设计人员, 必须会设计算法, 并根据算法写出程序。

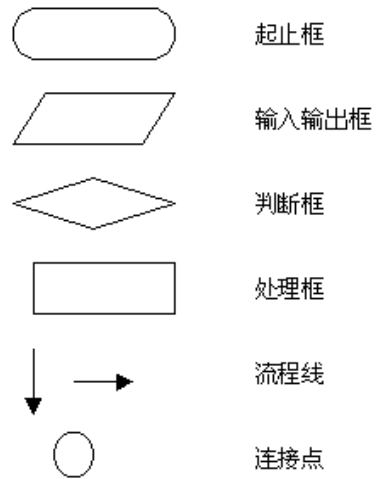
2.4 怎样表示一个算法

2.4.1 用自然语言表示算法

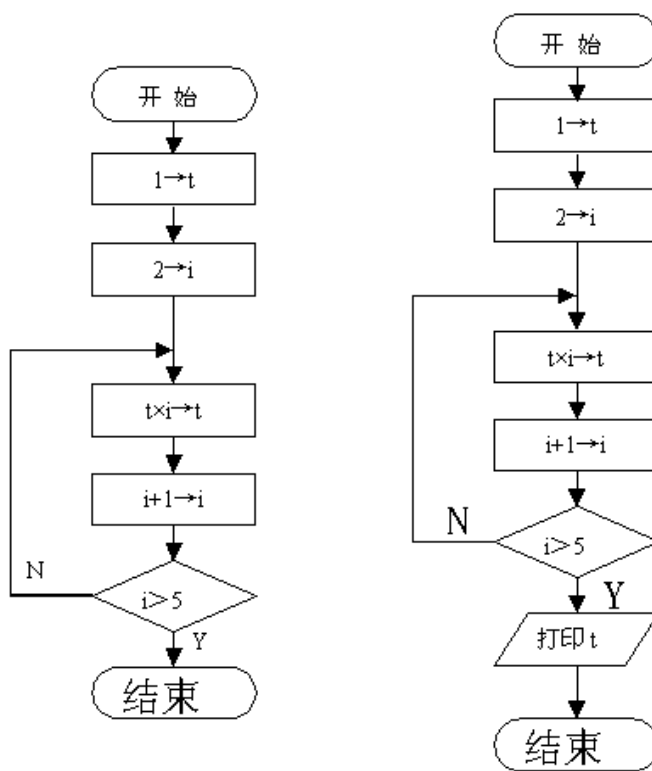
除了很简单的问题, 一般不用自然语言表示算法。

2.4.2 用流程图表示算法

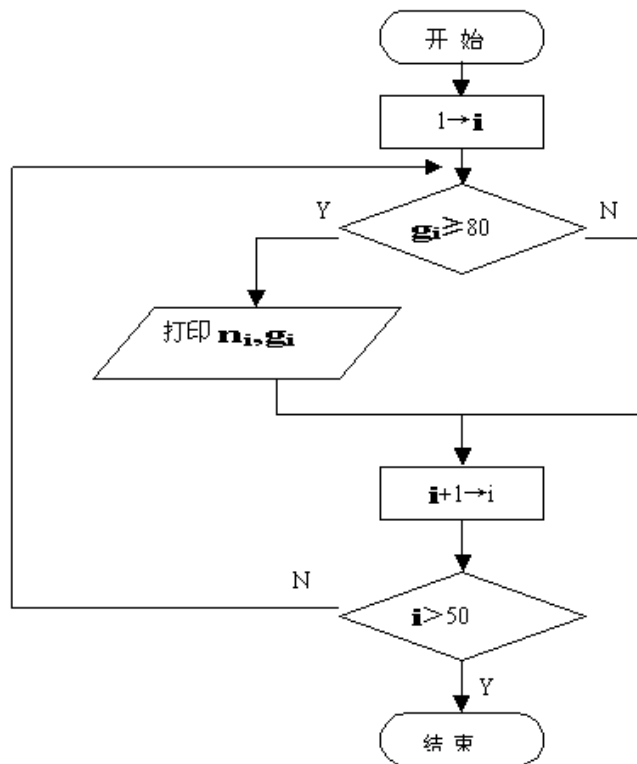
流程图表示算法, 直观形象, 易于理解。



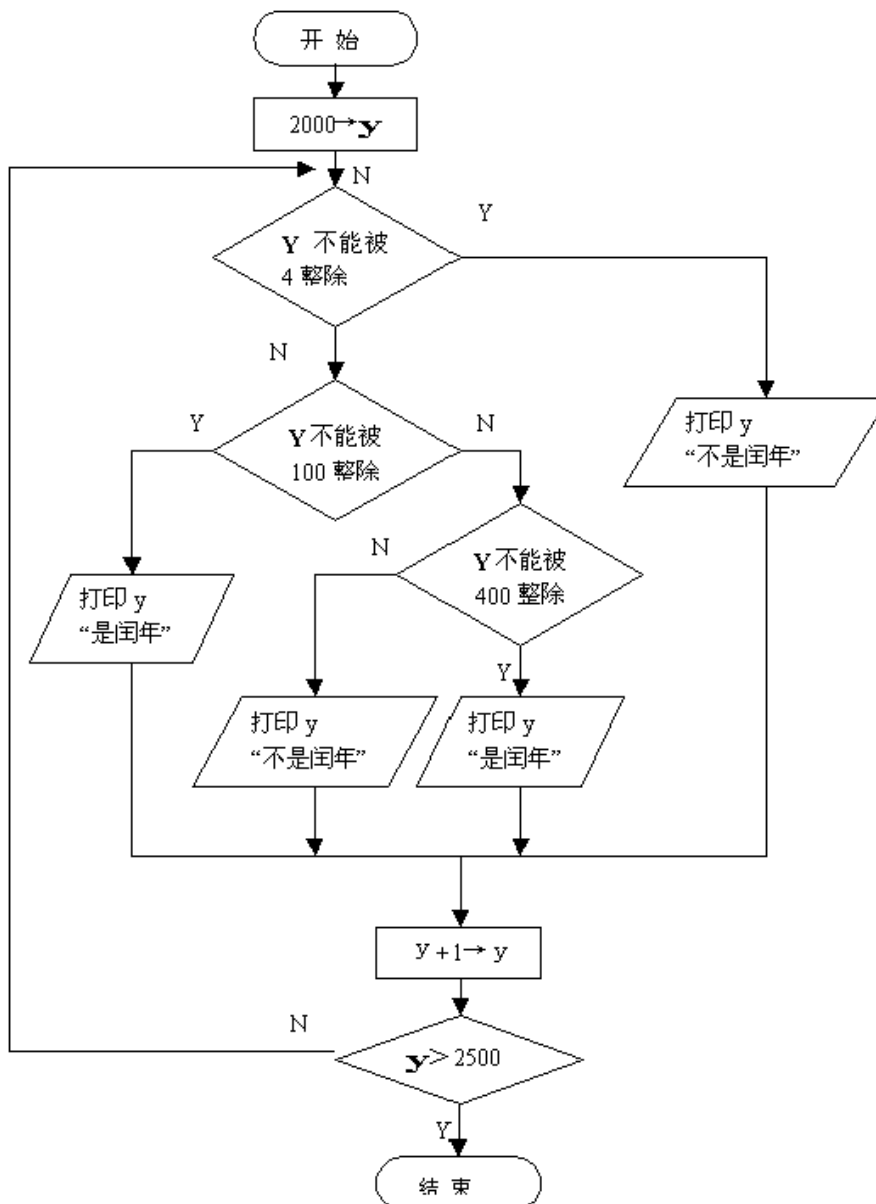
【例 2.6】将例 2.1 求 5! 的算用流程图表示。



【例 2.7】将例 2.2 的算用流程图表示。



【例 2.8】将例 2.3 判定闰年的算用流程图表示。



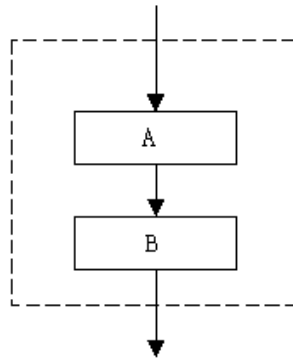
【例 2.9】将例 2.4 求 $1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots + \frac{1}{99} - \frac{1}{100}$ 的算用流程图表示。

一个流程图包括：

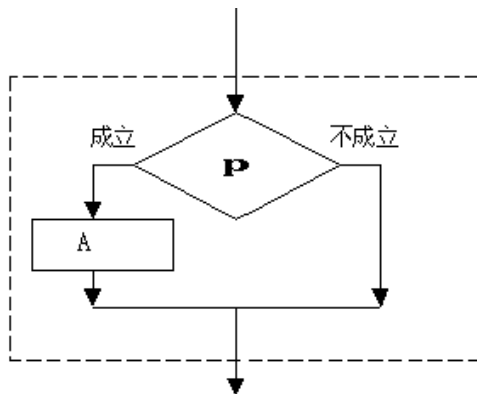
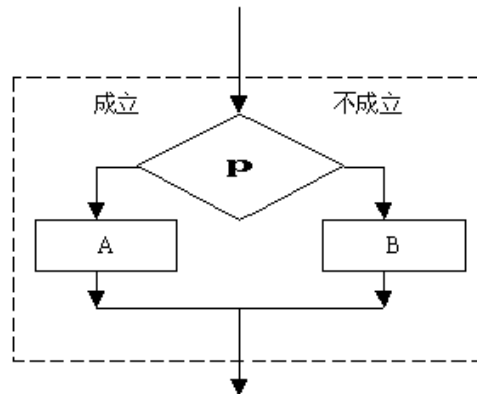
1. 表示相应操作的框；
2. 带箭头的流程线；
3. 框内外必要的文字说明。

2.4.3 三种基本结构和改进的流程图

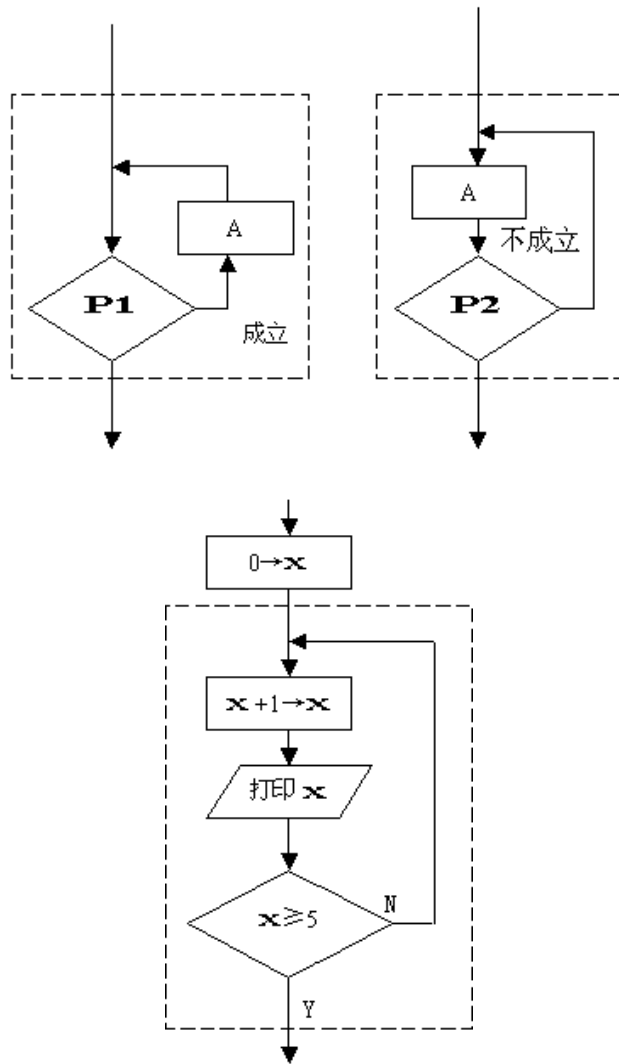
1. 顺序结构：



2. 选择结构：



3. 循环结构



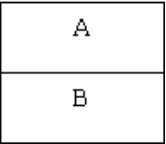
三种基本结构的共同特点：

- 只有一个入口；
- 只有一个出口；
- 结构内的每一部分都有机会被执行到；
- 结构内不存在“死循环”。

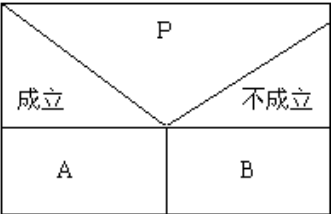
2.4.4 用 N-S 流程图表示算法

1973 年美国学者提出了一种新型流程图：N-S 流程图。

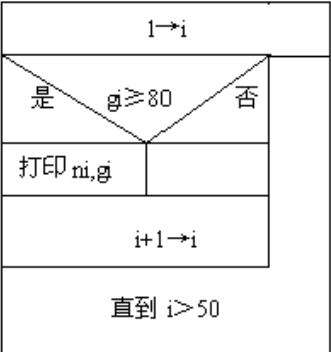
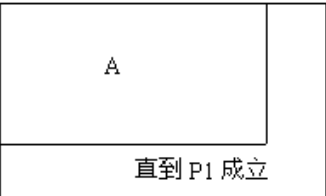
顺序结构：



选择结构:



循环结构:



2.4.5 用伪代码表示算法

伪代码使用介于自然语言和计算机语言之间的文字和符号来描述算法。

2.4.6 用计算机语言表示算法

- 我们的任务是用计算机解题，就是用计算机实现算法；
- 用计算机语言表示算法必须严格遵循所用语言的语法规则。

【例 2.20】求 $1 \times 2 \times 3 \times 4 \times 5$ 用 C 语言表示。

```
main()
{int i,t;
  t=1;
  i=2;
  while(i<=5)
  {t=t*i;
    i=i+1;
  }
  printf("%d",t);
}
```



【例 2.21】求级数的值。

```
main()
{
  int sigh=1;
  float deno=2.0,sum=1.0,term;
  while(deno<=100)
  { sigh= -sigh;
    term= sigh/ deno;
    sum=sum+term;
    deno=deno+1;
  }
  printf("%f",sum);
}
```



2.5 结构化程序设计方法

- 自顶向下；
- 逐步细化；
- 模块化设计；
- 结构化编码。

3	数据类型、运算符与表达式.....	1
3.1	C 语言的数据类型.....	1
3.2	常量与变量.....	3
3.2.1	常量和符号常量.....	3
3.2.2	变量.....	3
3.3	整型数据.....	4
3.3.1	整型常量的表示方法.....	4
3.3.2	整型变量.....	5
3.4	实型数据.....	7
3.4.1	实型常量的表示方法.....	7
3.4.2	实型变量.....	8
3.4.3	实型常数的类型.....	9
3.5	字符型数据.....	9
3.5.1	字符常量.....	9
3.5.2	转义字符.....	9
3.5.3	字符变量.....	10
3.5.4	字符数据在内存中的存储形式及使用方法.....	10
3.5.5	字符串常量.....	11
3.5.6	符号常量.....	12
3.6	变量赋初值.....	12
3.7	各类数值型数据之间的混合运算.....	13
3.8	算术运算符和算术表达式.....	14
3.8.1	C 运算符简介.....	14
3.8.2	算术运算符和算术表达式.....	15
3.9	赋值运算符和赋值表达式.....	17
3.10	逗号运算符和逗号表达式.....	18
3.11	小结.....	19
3.11.1	C 的数据类型.....	19
3.11.2	基本类型的分类及特点.....	19
3.11.3	常量后缀.....	19
3.11.4	常量类型.....	19
3.11.5	数据类型转换.....	19
3.11.6	运算符优先级和结合性.....	20
3.11.7	表达式.....	20

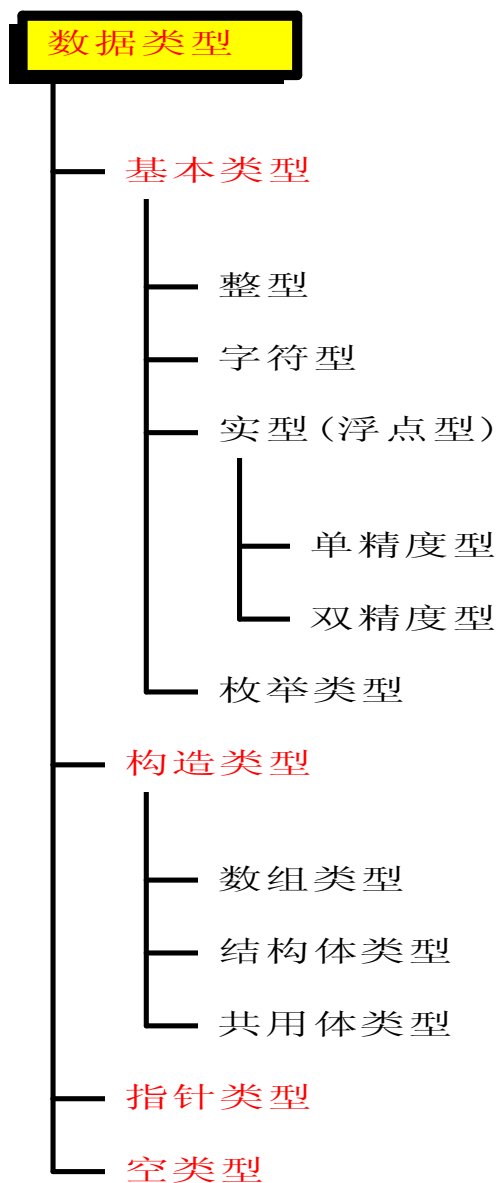
3 数据类型、运算符与表达式

3.1 C 语言的数据类型

在第一章中，我们已经看到程序中使用的各种变量都应预先加以定义，即先定义，后使用。对变量的定义可以包括三个方面：

- 数据类型
- 存储类型
- 作用域

在本章中，我们只介绍数据类型的说明。其它说明在以后各章中陆续介绍。所谓数据类型是按被定义变量的性质，表示形式，占据存储空间的大小，构造特点来划分的。在 C 语言中，数据类型可分为：基本数据类型，构造数据类型，指针类型，空类型四大类。



1. 基本数据类型：基本数据类型最主要的特点是，其值不可以再分解为其它类型。也就是说，基本数据类型是自我说明的。
2. 构造数据类型：构造数据类型是根据已定义的一个或多个数据类型用构造的方法来定义的。也就是说，一个构造类型的值可以分解成若干个“成员”或“元素”。每个“成员”

都是一个基本数据类型或又是一个构造类型。在 C 语言中，构造类型有以下几种：

- 数组类型
 - 结构体类型
 - 共用体（联合）类型
3. 指针类型：指针是一种特殊的，同时又是具有重要作用的数据类型。其值用来表示某个变量在内存存储器中的地址。虽然指针变量的取值类似于整型量，但这是两个类型完全不同的量，因此不能混为一谈。
 4. 空类型：在调用函数值时，通常应向调用者返回一个函数值。这个返回的函数值是具有一定的数据类型的，应在函数定义及函数说明中给以说明，例如在例题中给出的 `max` 函数定义中，函数头为：`int max(int a,int b);`其中“`int`”类型说明符即表示该函数的返回值为整型量。又如例题中，使用了库函数 `sin`，由于系统规定其函数返回值为双精度浮点型，因此在赋值语句 `s=sin(x);`中，`s` 也必须是双精度浮点型，以便与 `sin` 函数的返回值一致。所以在说明部分，把 `s` 说明为双精度浮点型。但是，也有一类函数，调用后并不需要向调用者返回函数值，这种函数可以定义为“空类型”。其类型说明符为 `void`。在后面函数中还要详细介绍。

在本章中，我们先介绍基本数据类型中的整型、浮点型和字符型。其余类型在以后各章中陆续介绍。

3.2 常量与变量

对于基本数据类型量，按其取值是否可改变又分为常量和变量两种。在程序执行过程中，其值不发生改变的量称为常量，其值可变的量称为变量。它们可与数据类型结合起来分类。例如，可分为整型常量、整型变量、浮点常量、浮点变量、字符常量、字符变量、枚举常量、枚举变量。在程序中，常量是可以不经说明而直接引用的，而变量则必须先定义后使用。整型量包括整型常量、整型变量。

3.2.1 常量和符号常量

在程序执行过程中，其值不发生改变的量称为常量。

- 直接常量(字面常量):
 - 整型常量：12、0、-3;
 - 实型常量：4.6、-1.23;
 - 字符常量：'a'、'b'。
- 标识符：用来标识变量名、符号常量名、函数名、数组名、类型名、文件名的有效字符序列。
- 符号常量：用标识符代表一个常量。在 C 语言中，可以用一个标识符来表示一个常量，称之为符号常量。

符号常量在使用之前必须先定义，其一般形式为：

```
#define 标识符 常量
```

其中 `#define` 也是一条预处理命令（预处理命令都以“`#`”开头），称为宏定义命令（在后面预处理程序中将进一步介绍），其功能是把该标识符定义为其后的常量值。一经定义，以后在程序中所有出现该标识符的地方均代之以该常量值。

- 习惯上符号常量的标识符用大写字母，变量标识符用小写字母，以示区别。

【例 3.1】符号常量的使用。

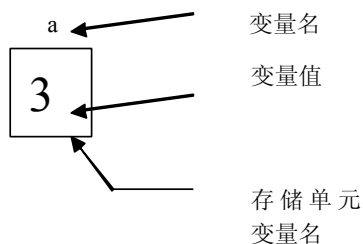
```
#define PRICE 30
main()
{
    int num,total;
    num=10;
    total=num* PRICE;
    printf("total=%d",total);
}
```



- 用标识符代表一个常量，称为符号常量。
- 符号常量与变量不同，它的值在其作用域内不能改变，也不能再被赋值。
- 使用符号常量的好处是：
 - 含义清楚；
 - 能做到“一改全改”。

3.2.2 变量

其值可以改变的量称为变量。一个变量应该有一个名字，在内存中占据一定的存储单元。变量定义必须放在变量使用之前。一般放在函数体的开头部分。要区分变量名和变量值是两个不同的概念。



3.3 整型数据

3.3.1 整型常量的表示方法

整型常量就是整常数。在 C 语言中，使用的整常数有八进制、十六进制和十进制三种。

- 1) 十进制整常数：十进制整常数没有前缀。其数码为 0~9。

以下各数是合法的十进制整常数：

237、-568、65535、1627；

以下各数不是合法的十进制整常数：

023 (不能有前导 0)、23D (含有非十进制数码)。

在程序中是根据前缀来区分各种进制数的。因此在书写常数时不要把前缀弄错造成

结果不正确。

- 2) 八进制整数：八进制整数必须以 0 开头，即以 0 作为八进制数的前缀。数码取值为 0~7。八进制数通常是无符号数。

以下各数是合法的八进制数：

015(十进制为 13)、0101(十进制为 65)、0177777(十进制为 65535)；

以下各数不是合法的八进制数：

256(无前缀 0)、03A2(包含了非八进制数码)、-0127(出现了负号)。

- 3) 十六进制整数：十六进制整数的前缀为 0X 或 0x。其数码取值为 0~9, A~F 或 a~f。

以下各数是合法的十六进制整数：

0X2A(十进制为 42)、0XA0(十进制为 160)、0XFFFF(十进制为 65535)；

以下各数不是合法的十六进制整数：

5A(无前缀 0X)、0X3H(含有非十六进制数码)。

- 4) 整型常数的后缀：在 16 位字长的机器上，基本整型的长度也为 16 位，因此表示的数的范围也是有限的。十进制无符号整常数的范围为 0~65535，有符号数为 -32768~+32767。八进制无符号数的表示范围为 0~0177777。十六进制无符号数的表示范围为 0X0~0XFFFF 或 0x0~0xFFFF。如果使用的数超过了上述范围，就必须用长整型数来表示。长整型数是用后缀“L”或“l”来表示的。

例如：

十进制长整常数：

158L(十进制为 158)、358000L(十进制为 358000)；

八进制长整常数：

012L(十进制为 10)、077L(十进制为 63)、0200000L(十进制为 65536)；

十六进制长整常数：

0X15L(十进制为 21)、0XA5L(十进制为 165)、0X10000L(十进制为 65536)。

长整数 158L 和基本整常数 158 在数值上并无区别。但对 158L，因为是长整型量，C 编译系统将为它分配 4 个字节存储空间。而对 158，因为是基本整型，只分配 2 个字节的存储空间。因此在运算和输出格式上要予以注意，避免出错。

无符号数也可用后缀表示，整型常数的无符号数的后缀为“U”或“u”。

例如：

358u, 0x38Au, 235Lu 均为无符号数。

前缀，后缀可同时使用以表示各种类型的数。如 0XA5Lu 表示十六进制无符号长整数 A5，其十进制为 165。

3.3.2 整型变量

1. 整型数据在内存中的存放形式

如果定义了一个整型变量 i：

```
int i;
i=10;
```

i 10

0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

数值是以补码表示的：

- 正数的补码和原码相同；
- 负数的补码：将该数的绝对值的二进制形式按位取反再加 1。

例如：

求-10 的补码：

10 的原码：

0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

取反：

1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

再加 1，得-10 的补码：

1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

由此可知，左面的第一位是表示符号的。

2. 整型变量的分类

- 1) 基本型：类型说明符为 `int`，在内存中占 2 个字节。
- 2) 短整型：类型说明符为 `short int` 或 `short`。所占字节和取值范围均与基本型相同。
- 3) 长整型：类型说明符为 `long int` 或 `long`，在内存中占 4 个字节。
- 4) 无符号型：类型说明符为 `unsigned`。

无符号型又可与上述三种类型匹配而构成：

- 无符号基本型：类型说明符为 `unsigned int` 或 `unsigned`。
- 无符号短整型：类型说明符为 `unsigned short`。
- 无符号长整型：类型说明符为 `unsigned long`。

各种无符号类型量所占的内存空间字节数与相应的有符号类型量相同。但由于省去了符号位，故不能表示负数。

有符号整型变量：最大表示 32767

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

无符号整型变量：最大表示 65535

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

下表列出了 Turbo C 中各类整型量所分配的内存字节数及数的表示范围。

类型说明符	数的范围		字节数
<code>int</code>	-32768~32767	即 $-2^{15} \sim (2^{15}-1)$	2
<code>unsigned int</code>	0~65535	即 $0 \sim (2^{16}-1)$	2
<code>short int</code>	-32768~32767	即 $-2^{15} \sim (2^{15}-1)$	2
<code>unsigned short int</code>	0~65535	即 $0 \sim (2^{16}-1)$	2
<code>long int</code>	-2147483648~2147483647	即 $-2^{31} \sim (2^{31}-1)$	4
<code>unsigned long</code>	0~4294967295	即 $0 \sim (2^{32}-1)$	4

以 13 为例：

`int` 型：

00	00	00	00	00	00	11	01
----	----	----	----	----	----	----	----

short int 型:

0	0	0	0	0	0	1	1	0	1
---	---	---	---	---	---	---	---	---	---

long int 型:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

unsigned int 型:

0	0	0	0	0	0	1	1	0	1
---	---	---	---	---	---	---	---	---	---

unsigned short int 型:

0	0	0	0	0	0	1	1	0	1
---	---	---	---	---	---	---	---	---	---

unsigned long int 型:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

3. 整型变量的定义

变量定义的一般形式为:

类型说明符 变量名标识符, 变量名标识符, ...;

例如:

int a,b,c; (a,b,c 为整型变量)

long x,y; (x,y 为长整型变量)

unsigned p,q; (p,q 为无符号整型变量)

在书写变量定义时, 应注意以下几点:

- 允许在一个类型说明符后, 定义多个相同类型的变量。各变量名之间用逗号间隔。类型说明符与变量名之间至少用一个空格间隔。
- 最后一个变量名之后必须以 “;” 号结尾。
- 变量定义必须放在变量使用之前。一般放在函数体的开头部分。

【例 3.2】整型变量的定义与使用。

```
main()
{
    int a,b,c,d;
    unsigned u;
    a=12;b=-24;u=10;
    c=a+u;d=b+u;
    printf("a+u=%d,b+u=%d\n",c,d);
}
```



4. 整型数据的溢出

【例 3.3】整型数据的溢出。

```
main()
{
    int a,b;
    a=32767;
    b=a+1;
    printf("%d,%d\n",a,b);
}
```



32767:

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

-32768

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

【例 3.4】

```
main(){
    long x,y;
    int a,b,c,d;
    x=5;
    y=6;
    a=7;
    b=8;
    c=x+a;
    d=y+b;
    printf("c=x+a=%d,d=y+b=%d\n",c,d);
}
```



从程序中可以看到：x, y 是长整型变量，a, b 是基本整型变量。它们之间允许进行运算，运算结果为长整型。但 c, d 被定义为基本整型，因此最后结果为基本整型。本例说明，不同类型的量可以参与运算并相互赋值。其中的类型转换是由编译系统自动完成的。有关类型转换的规则将在以后介绍。

3.4 实型数据

3.4.1 实型常量的表示方法

实型也称为浮点型。实型常量也称为实数或者浮点数。在 C 语言中，实数只采用十进制。它有二种形式：十进制小数形式，指数形式。

- 1) 十进制数形式：由数码 0~9 和小数点组成。

例如：

0.0、25.0、5.789、0.13、5.0、300.、-267.8230

等均为合法的实数。注意，必须有小数点。

- 2) 指数形式：由十进制数，加阶码标志“e”或“E”以及阶码（只能为整数，可以带符号）组成。

其一般形式为：

$aE n$ (a 为十进制数，n 为十进制整数)

其值为 $a \times 10^n$ 。

如：

2.1E5 (等于 2.1×10^5)

3.7E-2 (等于 3.7×10^{-2})

0.5E7 (等于 0.5×10^7)
 -2.8E-2 (等于 -2.8×10^{-2})

以下不是合法的实数:

345 (无小数点)
 E7 (阶码标志 E 之前无数字)
 -5 (无阶码标志)
 53.-E3 (负号位置不对)
 2.7E (无阶码)

标准 C 允许浮点数使用后缀。后缀为 “f” 或 “F” 即表示该数为浮点数。如 356f 和 356. 是等价的。

【例 3.5】说明了这种情况。

```
main(){
    printf("%f\n",356.);
    printf("%f\n",356);
    printf("%f\n",356f);
}
```



3.4.2 实型变量

1. 实型数据在内存中的存放形式

实型数据一般占 4 个字节 (32 位) 内存空间。按指数形式存储。实数 3.14159 在内存中的存放形式如下:

+	.314159	1
数符	小数部分	指数

- 小数部分占的位 (bit) 数愈多, 数的有效数字愈多, 精度愈高。
- 指数部分占的位数愈多, 则能表示的数值范围愈大。

2. 实型变量的分类

实型变量分为: 单精度 (float 型)、双精度 (double 型) 和长双精度 (long double 型) 三类。

在 Turbo C 中单精度型占 4 个字节 (32 位) 内存空间, 其数值范围为 $3.4\text{E}-38 \sim 3.4\text{E}+38$, 只能提供七位有效数字。双精度型占 8 个字节 (64 位) 内存空间, 其数值范围为 $1.7\text{E}-308 \sim 1.7\text{E}+308$, 可提供 16 位有效数字。

类型说明符	比特数 (字节数)	有效数字	数的范围
float	32 (4)	6~7	$10^{-37} \sim 10^{38}$
double	64(8)	15~16	$10^{-307} \sim 10^{308}$
long double	80 (10)	18~19	$10^{-4931} \sim 10^{4932}$

实型变量定义的格式和书写规则与整型相同。

例如:

float x,y; (x,y 为单精度实型量)
 double a,b,c; (a,b,c 为双精度实型量)

3. 实型数据的舍入误差

由于实型变量是由有限的存储单元组成的,因此能提供的有效数字总是有限的。如下例。

【例 3.6】实型数据的舍入误差。

```
main()
{float a,b;
a=123456.789e5;
b=a+20
printf("%f\n",a);
printf("%f\n",b);
}
```



注意: $1.0/3*3$ 的结果并不等于 1。

【例 3.7】

```
main()
{
float a;
double b;
a=33333.33333;
b=33333.33333333333333;
printf("%f\n%f\n",a,b);
}
```



- 从本例可以看出,由于 **a** 是单精度浮点型,有效位数只有七位。而整数已占五位,故小数二位后之后均为无效数字。
- **b** 是双精度型,有效位为十六位。但 Turbo C 规定小数后最多保留六位,其余部分四舍五入。

3.4.3 实型常数的类型

实型常数不分单、双精度,都按双精度 **double** 型处理。

3.5 字符型数据

字符型数据包括字符常量和字符变量。

3.5.1 字符常量

字符常量是用单引号括起来的一个字符。

例如:

'a'、'b'、'='、'+'、'？'

都是合法字符常量。

在 C 语言中，字符常量有以下特点：

- 1) 字符常量只能用单引号括起来，不能用双引号或其它括号。
- 2) 字符常量只能是单个字符，不能是字符串。
- 3) 字符可以是字符集中任意字符。但数字被定义为字符型之后就不能参与数值运算。
如 '5' 和 5 是不同的。'5' 是字符常量，不能参与运算。

3.5.2 转义字符

转义字符是一种特殊的字符常量。转义字符以反斜线“\”开头，后跟一个或几个字符。转义字符具有特定的含义，不同于字符原有的意义，故称“转义”字符。例如，在前面各例题 `printf` 函数的格式串中用到的“\n”就是一个转义字符，其意义是“回车换行”。转义字符主要用来表示那些用一般字符不便于表示的控制代码。

常用的转义字符及其含义

转义字符	转义字符的意义	ASCII 代码
\n	回车换行	10
\t	横向跳到下一制表位置	9
\b	退格	8
\r	回车	13
\f	走纸换页	12
\\	反斜线符“\”	92
\'	单引号符	39
\”	双引号符	34
\a	鸣铃	7
\ddd	1~3 位八进制数所代表的字符	
\xhh	1~2 位十六进制数所代表的字符	

广义地讲，C 语言字符集中的任何一个字符均可用转义字符来表示。表中的 \ddd 和 \xhh 正是为此而提出的。ddd 和 hh 分别为八进制和十六进制的 ASCII 代码。如 \101 表示字母“A”，\102 表示字母“B”，\134 表示反斜线，\XOA 表示换行等。

【例 3.8】转义字符的使用。

```
main()
{
    printf(" ab c\tde\r\n");
    printf("hijk\tL\bM\n");
}
```



3.5.3 字符变量

字符变量用来存储字符常量，即单个字符。

字符变量的类型说明符是 `char`。字符变量类型定义的格式和书写规则都与整型变量相同。例如：

```
char a,b;
```

3.5.4 字符数据在内存中的存储形式及使用方法

每个字符变量被分配一个字节的内存空间，因此只能存放一个字符。字符值是以 ASCII 码的形式存放在变量的内存单元之中的。

如 x 的十进制 ASCII 码是 120，y 的十进制 ASCII 码是 121。对字符变量 a,b 赋予 'x' 和 'y' 值：

```
a='x';
```

```
b='y';
```

实际上是在 a,b 两个单元内存放 120 和 121 的二进制代码：

a:

0	1	1	1	1	0	0	0
---	---	---	---	---	---	---	---

b:

0	1	1	1	1	0	0	1
---	---	---	---	---	---	---	---

所以也可以把它们看成是整型量。C 语言允许对整型变量赋以字符值，也允许对字符变量赋以整型值。在输出时，允许把字符变量按整型量输出，也允许把整型量按字符量输出。

整型量为二字节量，字符量为单字节量，当整型量按字符型量处理时，只有低八位字节参与处理。

【例 3.9】向字符变量赋以整数。

```
main()
{
    char a,b;
    a=120;
    b=121;
    printf("%c,%c\n",a,b);
    printf("%d,%d\n",a,b);
}
```



本程序中定义 a, b 为字符型，但在赋值语句中赋以整型值。从结果看，a, b 值的输出形式取决于 printf 函数格式串中的格式符，当格式符为 "c" 时，对应输出的变量值为字符，当格式符为 "d" 时，对应输出的变量值为整数。

【例 3.10】

```
main()
{
    char a,b;
    a='a';
    b='b';
    a=a-32;
    b=b-32;
    printf("%c,%c\n%d,%d\n",a,b,a,b);
}
```



本例中，a，b 被说明为字符变量并赋予字符值，C 语言允许字符变量参与数值运算，即用字符的 ASCII 码参与运算。由于大小写字母的 ASCII 码相差 32，因此运算后把小写字母换成大写字母。然后分别以整型和字符型输出。

3.5.5 字符串常量

字符串常量是由一对双引号括起的字符序列。例如：“CHINA”，“C program”，“\$12.5”等都是合法的字符串常量。

字符串常量和字符常量是不同的量。它们之间主要有以下区别：

- 1) 字符常量由单引号括起来，字符串常量由双引号括起来。
- 2) 字符常量只能是单个字符，字符串常量则可以含一个或多个字符。
- 3) 可以把一个字符常量赋予一个字符变量，但不能把一个字符串常量赋予一个字符变量。在 C 语言中没有相应的字符串变量。这是与 BASIC 语言不同的。但是可以用一个字符数组来存放一个字符串常量。在数组一章内予以介绍。
- 4) 字符常量占一个字节的内存空间。字符串常量占的内存字节数等于字符串中字节数加 1。增加的一个字节中存放字符“\0” (ASCII 码为 0)。这是字符串结束的标志。

例如：

字符串 “C program” 在内存中所占的字节为：

C		p	r	o	g	r	a	m	\0
---	--	---	---	---	---	---	---	---	----

字符常量'a'和字符串常量"a"虽然都只有一个字符，但在内存中的情况是不同的。

'a'在内存中占一个字节，可表示为：

a

"a"在内存中占二个字节，可表示为：

a	\0
---	----

3.6 变量赋初值

在程序中常常需要对变量赋初值，以便使用变量。语言程序中可有多种方法为变量提供初值。本小节先介绍在作变量定义的同时给变量赋以初值的方法。这种方法称为初始化。在变量定义中赋初值的一般形式为：

类型说明符 变量 1= 值 1，变量 2= 值 2，……；

例如：

```
int a=3;
```

```
int b,c=5;
```

```
float x=3.2,y=3,f,z=0.75;
```

```
char ch1='K',ch2='P';
```

应注意，在定义中不允许连续赋值，如 a=b=c=5 是不合法的。

【例 3.11】

```
main()
{
```

```

int a=3,b,c=5;
b=a+c;
printf("a=%d,b=%d,c=%d\n",a,b,c);
}

```

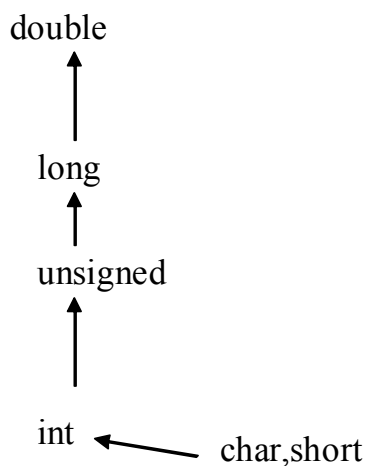


3.7 各类数值型数据之间的混合运算

变量的数据类型是可以转换的。转换的方法有两种，一种是自动转换，一种是强制转换。自动转换发生在不同数据类型的量混合运算时，由编译系统自动完成。自动转换遵循以下规则：

- 1) 若参与运算量的类型不同，则先转换成同一类型，然后进行运算。
- 2) 转换按数据长度增加的方向进行，以保证精度不降低。如 `int` 型和 `long` 型运算时，先把 `int` 量转成 `long` 型后再进行运算。
- 3) 所有的浮点运算都是以双精度进行的，即使仅含 `float` 单精度量运算的表达式，也要先转换成 `double` 型，再作运算。
- 4) `char` 型和 `short` 型参与运算时，必须先转换成 `int` 型。
- 5) 在赋值运算中，赋值号两边量的数据类型不同时，赋值号右边量的类型将转换为左边量的类型。如果右边量的数据类型长度左边长时，将丢失一部分数据，这样会降低精度，丢失的部分按四舍五入向前舍入。

下图表示了类型自动转换的规则。



【例 3.12】

```

main()
{
float PI=3.14159;
int s,r=5;
s=r*r*PI;
}

```

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



本例程序中，PI 为实型；s，r 为整型。在执行 `s=r*r*PI` 语句时，r 和 PI 都转换成 double 型计算，结果也为 double 型。但由于 s 为整型，故赋值结果仍为整型，舍去了小数部分。

强制类型转换

强制类型转换是通过类型转换运算来实现的。

其一般形式为：

(类型说明符) (表达式)

其功能是把表达式的运算结果强制转换成类型说明符所表示的类型。

例如：

(float) a 把 a 转换为实型

(int)(x+y) 把 x+y 的结果转换为整型

在使用强制转换时应注意以下问题：

- 1) 类型说明符和表达式都必须加括号(单个变量可以不加括号)，如把 `(int)(x+y)` 写成 `(int)x+y` 则成了把 x 转换成 int 型之后再与 y 相加了。
- 2) 无论是强制转换或是自动转换，都只是为了本次运算的需要而对变量的数据长度进行的临时性转换，而不改变数据说明时对该变量定义的类型。

【例 3.13】

```
main()
{
    float f=5.75;
    printf("(int)f=%d,f=%f\n",(int)f,f);
}
```



本例表明，f 虽强制转为 int 型，但只在运算中起作用，是临时的，而 f 本身的类型并不改变。因此，`(int)f` 的值为 5(删去了小数)而 f 的值仍为 5.75。

3.8 算术运算符和算术表达式

C 语言中运算符和表达式数量之多，在高级语言中是少见的。正是丰富的运算符和表达式使 C 语言功能十分完善。这也是 C 语言的主要特点之一。

C 语言的运算符不仅具有不同的优先级，而且还有一个特点，就是它的结合性。在表达式中，各运算量参与运算的先后顺序不仅要遵守运算符优先级别的规定，还要受运算符结合性的制约，以便确定是自左向右进行运算还是自右向左进行运算。这种结合性是其它高级语言的运算符所没有的，因此也增加了 C 语言的复杂性。

3.8.1 C 运算符简介

C 语言的运算符可分为以下几类：

1. 算术运算符:用于各类数值运算。包括加(+)、减(-)、乘(*)、除(/)、求余(或称模运算，%)、

- 自增(++)、自减(--)共七种。
2. 关系运算符:用于比较运算。包括大于(>)、小于(<)、等于(==)、大于等于(>=)、小于等于(<=)和不等不等于(!=)六种。
 3. 逻辑运算符:用于逻辑运算。包括与(&&)、或(||)、非(!)三种。
 4. 位操作运算符:参与运算的量,按二进制位进行运算。包括位与(&)、位或(|)、位非(~)、位异或(^)、左移(<<)、右移(>>)六种。
 5. 赋值运算符:用于赋值运算,分为简单赋值(=)、复合算术赋值(+=, -=, *=, /=, %=)和复合位运算赋值(&=, |=, ^=, >>=, <<=)三类共十一种。
 6. 条件运算符:这是一个三目运算符,用于条件求值(?:)。
 7. 逗号运算符:用于把若干表达式组合成一个表达式(,)。
 8. 指针运算符:用于取内容(*)和取地址(&)二种运算。
 9. 求字节数运算符:用于计算数据类型所占的字节数(sizeof)。
 10. 特殊运算符:有括号(), 下标[], 成员(→, .)等几种。

3.8.2 算术运算符和算术表达式

1. 基本的算术运算符

- 加法运算符“+”:加法运算符为双目运算符,即应有两个量参与加法运算。如 $a+b$, $4+8$ 等。具有右结合性。
- 减法运算符“-”:减法运算符为双目运算符。但“-”也可作负值运算符,此时为单目运算,如 $-x$, -5 等具有左结合性。
- 乘法运算符“*”:双目运算,具有左结合性。
- 除法运算符“/”:双目运算具有左结合性。参与运算量均为整型时,结果也为整型,舍去小数。如果运算量中有一个是实型,则结果为双精度实型。

【例 3.14】

```
main(){
    printf("\n\n%d,%d\n",20/7,-20/7);
    printf("%f,%f\n",20.0/7,-20.0/7);
}
```



本例中, $20/7$, $-20/7$ 的结果均为整型,小数全部舍去。而 $20.0/7$ 和 $-20.0/7$ 由于有实数参与运算,因此结果也为实型。

- 求余运算符(模运算符)“%”:双目运算,具有左结合性。要求参与运算的量均为整型。求余运算的结果等于两数相除后的余数。

【例 3.15】

```
main(){
    printf("%d\n",100%3);
}
```



本例输出 100 除以 3 所得的余数 1。

2. 算术表达式和运算符的优先级和结合性

表达式是由常量、变量、函数和运算符组合起来的式子。一个表达式有一个值及其类型,

它们等于计算表达式所得结果的值和类型。表达式求值按运算符的优先级和结合性规定的顺序进行。单个的常量、变量、函数可以看作是表达式的特例。

算术表达式是由算术运算符和括号连接起来的式子。

- **算术表达式：**用算术运算符和括号将运算对象（也称操作数）连接起来的、符合 C 语法规则的式子。

以下是算术表达式的例子：

```
a+b
(a*2) / c
(x+r)*8-(a+b) / 7
++I
sin(x)+sin(y)
(++i)-(j++)+(k--)
```

- **运算符的优先级：**C 语言中，运算符的运算优先级共分为 15 级。1 级最高，15 级最低。在表达式中，优先级较高的先于优先级较低的进行运算。而在一个运算量两侧的运算符优先级相同时，则按运算符的结合性所规定的结合方向处理。
- **运算符的结合性：**C 语言中各运算符的结合性分为两种，即左结合性(自左至右)和右结合性(自右至左)。例如算术运算符的结合性是自左至右，即先左后右。如有表达式 $x-y+z$ 则 y 应先与“-”号结合，执行 $x-y$ 运算，然后再执行 $+z$ 的运算。这种自左至右的结合方向就称为“左结合性”。而自右至左的结合方向称为“右结合性”。最典型的右结合性运算符是赋值运算符。如 $x=y=z$ ，由于“=”的右结合性，应先执行 $y=z$ 再执行 $x=(y=z)$ 运算。C 语言运算符中有不少为右结合性，应注意区别，以避免理解错误。

3. 强制类型转换运算符

其一般形式为：

(类型说明符) (表达式)

其功能是把表达式的运算结果强制转换成类型说明符所表示的类型。

例如：

```
(float) a    把 a 转换为实型
(int)(x+y)   把 x+y 的结果转换为整型
```

4. 自增、自减运算符

自增 1，自减 1 运算符：自增 1 运算符记为“++”，其功能是使变量的值自增 1。

自减 1 运算符记为“--”，其功能是使变量值自减 1。

自增 1，自减 1 运算符均为单目运算，都具有右结合性。可有以下几种形式：

```
++i    i 自增 1 后再参与其它运算。
--i    i 自减 1 后再参与其它运算。
i++    i 参与运算后，i 的值再自增 1。
i--    i 参与运算后，i 的值再自减 1。
```

在理解和使用上容易出错的是 $i++$ 和 $i--$ 。特别是当它们出在较复杂的表达式或语句中时，常常难于弄清，因此应仔细分析。

【例 3.16】

```
main()
{
    int i=8;
    printf("%d\n",++i);
    printf("%d\n",--i);
}
```

```
printf("%d\n",i++);
printf("%d\n",i--);
printf("%d\n",-i++);
printf("%d\n",-i--);
}
```



i 的初值为 8，第 2 行 i 加 1 后输出故为 9；第 3 行减 1 后输出故为 8；第 4 行输出 i 为 8 之后再加 1(为 9)；第 5 行输出 i 为 9 之后再减 1(为 8)；第 6 行输出 -8 之后再加 1(为 9)，第 7 行输出 -9 之后再减 1(为 8)。

【例 3.17】

```
main()
{
    int i=5,j=5,p,q;
    p=(i++)+(i++)+(i++);
    q=(++j)+(++j)+(++j);
    printf("%d,%d,%d,%d",p,q,i,j);
}
```



这个程序中，对 $P=(i++)+(i++)+(i++)$ 应理解为三个 i 相加，故 P 值为 15。然后 i 再自增 1 三次相当于加 3 故 i 的最后值为 8。而对于 q 的值则不然， $q=(++j)+(++j)+(++j)$ 应理解为 q 先自增 1，再参与运算，由于 q 自增 1 三次后值为 8，三个 8 相加的和为 24，j 的最后值仍为 8。

3.9 赋值运算符和赋值表达式

1. 赋值运算符

简单赋值运算符和表达式:简单赋值运算符记为“=”。由“=”连接的式子称为赋值表达式。其一般形式为:

变量=表达式

例如:

$x=a+b$

$w=\sin(a)+\sin(b)$

$y=i++-j$

赋值表达式的功能是计算表达式的值再赋予左边的变量。赋值运算符具有右结合性。因此

$a=b=c=5$

可理解为

$a=(b=(c=5))$

在其它高级语言中，赋值构成了一个语句，称为赋值语句。而在 C 中，把“=”定义为运算符，从而组成赋值表达式。凡是表达式可以出现的地方均可出现赋值表达式。

例如，式子:

$x=(a=5)+(b=8)$

是合法的。它的意义是把 5 赋予 a, 8 赋予 b, 再把 a,b 相加, 和赋予 x, 故 x 应等于 13。

在 C 语言中也可以组成赋值语句, 按照 C 语言规定, 任何表达式在其末尾加上分号就构成语句。因此如

```
x=8;a=b=c=5;
```

都是赋值语句, 在前面各例中我们已大量使用过了。

2. 类型转换

如果赋值运算符两边的数据类型不相同, 系统将自动进行类型转换, 即把赋值号右边的类型换成左边的类型。具体规定如下:

- 1) 实型赋予整型, 舍去小数部分。前面的例子已经说明了这种情况。
- 2) 整型赋予实型, 数值不变, 但将以浮点形式存放, 即增加小数部分(小数部分的值为 0)。
- 3) 字符型赋予整型, 由于字符型为一个字节, 而整型为二个字节, 故将字符的 ASCII 码值放到整型量的低八位中, 高八位为 0。整型赋予字符型, 只把低八位赋予字符量。

【例 3.18】

```
main(){
    int a,b=322;
    float x,y=8.88;
    char c1='k',c2;
    a=y;
    x=b;
    a=c1;
    c2=b;
    printf("%d,%f,%d,%c",a,x,a,c2);
}
```



本例表明了上述赋值运算中类型转换的规则。a 为整型, 赋予实型量 y 值 8.88 后只取整数 8。x 为实型, 赋予整型量 b 值 322, 后增加了小数部分。字符型量 c1 赋予 a 变为整型, 整型量 b 赋予 c2 后取其低八位成为字符型(b 的低八位为 01000010, 即十进制 66, 按 ASCII 码对应于字符 B)。

3. 复合的赋值运算符

在赋值符 “=” 之前加上其它二目运算符可构成复合赋值符。如 +=, -=, *=, /=, %=, <<=, >>=, &=, ^=, |=。

构成复合赋值表达式的一般形式为:

变量 双目运算符=表达式

它等效于

变量=变量 运算符 表达式

例如:

```
a+=5      等价于 a=a+5
x*=y+7    等价于 x=x*(y+7)
r%=p      等价于 r=r%p
```

复合赋值符这种写法, 对初学者可能不习惯, 但十分有利于编译处理, 能提高编译效率并产生质量较高的目标代码。

3.10 逗号运算符和逗号表达式

在 C 语言中逗号“,”也是一种运算符,称为逗号运算符。其功能是把两个表达式连接起来组成一个表达式,称为逗号表达式。

其一般形式为:

表达式 1, 表达式 2

其求值过程是分别求两个表达式的值,并以表达式 2 的值作为整个逗号表达式的值。

【例 3.19】

```
main(){
    int a=2,b=4,c=6,x,y;
    y=(x=a+b),(b+c);
    printf("y=%d,x=%d",y,x);
}
```



本例中, y 等于整个逗号表达式的值,也就是表达式 2 的值, x 是第一个表达式的值。对于逗号表达式还要说明两点:

1) 逗号表达式一般形式中的表达式 1 和表达式 2 也可以又是逗号表达式。

例如:

表达式 1, (表达式 2, 表达式 3)

形成了嵌套情形。因此可以把逗号表达式扩展为以下形式:

表达式 1, 表达式 2, …表达式 n

整个逗号表达式的值等于表达式 n 的值。

2) 程序中使用逗号表达式,通常是要分别求逗号表达式内各表达式的值,并不一定要求整个逗号表达式的值。

并不是在所有出现逗号的地方都组成逗号表达式,如在变量说明中,函数参数表中逗号只是用作各变量之间的间隔符。

3.11 小结

3.11.1 C 的数据类型

基本类型,构造类型,指针类型,空类型

3.11.2 基本类型的分类及特点

	类型说明符	字节	数值范围
字符型	char	1	C 字符集
基本整型	int	2	-32768~32767
短整型	short int	2	-32768~32767
长整型	long int	4	-214783648~214783647

无符号型	unsigned	2	0~65535
无符号长整型	unsigned long	4	0~4294967295
单精度实型	float	4	3/4E-38~3/4E+38
双精度实型	double	8	1/7E-308~1/7E+308

3.11.3 常量后缀

L 或 l 长整型
U 或 u 无符号数
F 或 f 浮点数

3.11.4 常量类型

整数，长整数，无符号数，浮点数，字符，字符串，符号常数，转义字符。

3.11.5 数据类型转换

- 自动转换:在不同类型数据的混合运算中，由系统自动实现转换，由少字节类型向多字节类型转换。不同类型的量相互赋值时也由系统自动进行转换，把赋值号右边的类型转换为左边的类型。
- 强制转换:由强制转换运算符完成转换。

3.11.6 运算符优先级和结合性

一般而言，单目运算符优先级较高，赋值运算符优先级低。算术运算符优先级较高，关系和逻辑运算符优先级较低。多数运算符具有左结合性，单目运算符、三目运算符、赋值运算符具有右结合性。

3.11.7 表达式

表达式是由运算符连接常量、变量、函数所组成的式子。每个表达式都有一个值和类型。表达式求值按运算符的优先级和结合性所规定的顺序进行。

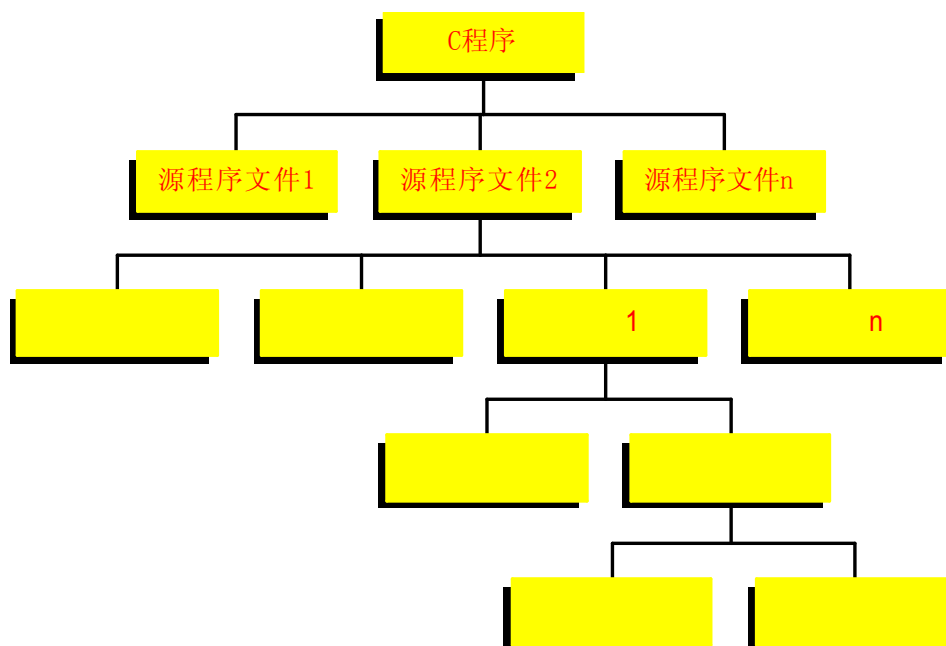
4	最简单的 C 程序设计—顺序程序设计.....	1
4.1	C 语句概述.....	1
4.2	赋值语句.....	3
4.3	数据输入输出的概念及在 C 语言中的实现.....	4
4.4	字符数据的输入输出.....	4
4.4.1	putchar 函数（字符输出函数）.....	4
4.4.2	getchar 函数（键盘输入函数）.....	5
4.5	格式输入与输出.....	5
4.5.1	printf 函数（格式输出函数）.....	5
4.5.2	scanf 函数(格式输入函数).....	8
4.6	顺序结构程序设计举例.....	12

4 最简单的 C 程序设计—顺序程序设计

从程序流程的角度来看，程序可以分为三种基本结构，即顺序结构、分支结构、循环结构。这三种基本结构可以组成所有的各种复杂程序。C 语言提供了多种语句来实现这些程序结构。本章介绍这些基本语句及其在顺序结构中的应用，使读者对 C 程序有一个初步的认识，为后面各章的学习打下基础。

4.1 C 语句概述

C 程序的结构：



C 程序的执行部分是由语句组成的。程序的功能也是由执行语句实现的。

C 语句可分为以下五类：

- 1) 表达式语句
- 2) 函数调用语句
- 3) 控制语句
- 4) 复合语句
- 5) 空语句

1. 表达式语句：表达式语句由表达式加上分号 “;” 组成。

其一般形式为：

表达式；

执行表达式语句就是计算表达式的值。

例如：

`x=y+z;` 赋值语句；

`y+z;` 加法运算语句，但计算结果不能保留，无实际意义；

`i++;` 自增 1 语句，i 值增 1。

2. 函数调用语句：由函数名、实际参数加上分号 “;” 组成。

其一般形式为：

函数名(实际参数表)；

执行函数语句就是调用函数体并把实际参数赋予函数定义中的形式参数，然后执行被调函数体中的语句，求取函数值（在后面函数中再详细介绍）。

例如：

`printf("C Program");`调用库函数，输出字符串。

3. 控制语句：控制语句用于控制程序的流程， 以实现程序的各种结构方式。它们由特定的语句定义符组成。C 语言有九种控制语句。 可分成以下三类：

- 1) 条件判断语句：if 语句、switch 语句；
- 2) 循环执行语句：do while 语句、while 语句、for 语句；
- 3) 转向语句：break 语句、goto 语句、continue 语句、return 语句。

4. 复合语句：把多个语句用括号 {} 括起来组成的一个语句称复合语句。

在程序中应把复合语句看成是单条语句，而不是多条语句。

例如：

```
{ x=y+z;
  a=b+c;
  printf("%d%d", x, a);
}
```

是一条复合语句。

复合语句内的各条语句都必须以分号 “;” 结尾，在括号 “}” 外不能加分号。

5. 空语句：只有分号 “;” 组成的语句称为空语句。空语句是什么也不执行的语句。在程序中空语句可用来作空循环体。

例如

```
while(getchar() != '\n')
;
```

本语句的功能是，只要从键盘输入的字符不是回车则重新输入。

这里的循环体为空语句。

4.2 赋值语句

赋值语句是由赋值表达式再加上分号构成的表达式语句。

其一般形式为：

变量=表达式；

赋值语句的功能和特点都与赋值表达式相同。它是程序中使用最多的语句之一。

在赋值语句的使用中需要注意以下几点：

1. 由于在赋值符“=”右边的表达式也可以又是一个赋值表达式，因此，下述形式

变量=(变量=表达式)；

是成立的，从而形成嵌套的情形。

其展开之后的一般形式为：

变量=变量=...=表达式；

例如：

```
a=b=c=d=e=5;
```

按照赋值运算符的右接合性，因此实际上等效于：

```
e=5;
```

```
d=e;
```

```
c=d;
```

```
b=c;
```

```
a=b;
```

2. 注意在变量说明中给变量赋初值和赋值语句的区别。

给变量赋初值是变量说明的一部分，赋初值后的变量与其后的其它同类变量之间仍必须用逗号间隔，而赋值语句则必须用分号结尾。

例如：

```
int a=5, b, c;
```

3. 在变量说明中，不允许连续给多个变量赋初值。

如下述说明是错误的：

```
int a=b=c=5
```

必须写为

```
int a=5, b=5, c=5;
```

而赋值语句允许连续赋值。

4. 注意赋值表达式和赋值语句的区别。

赋值表达式是一种表达式，它可以出现在任何允许表达式出现的地方，而赋值语句则不能。

下述语句是合法的：

```
if((x=y+5)>0) z=x;
```

语句的功能是，若表达式 $x=y+5$ 大于 0 则 $z=x$ 。

下述语句是非法的：

```
if((x=y+5;)>0) z=x;
```

因为 $x=y+5$; 是语句，不能出现在表达式中。

4.3 数据输入输出的概念及在 C 语言中的实现

- 1) 所谓输入输出是以计算机为主体而言的。
- 2) 本章介绍的是向标准输出设备显示器输出数据的语句。
- 3) 在 C 语言中，所有的数据输入 / 输出都是由库函数完成的。因此都是函数语句。
- 4) 在使用 C 语言库函数时，要用预编译命令

```
#include
```

将有关“头文件”包括到源文件中。

使用标准输入输出库函数时要用到“stdio.h”文件，因此源文件开头应有以下预编译命令：

```
#include<stdio.h>
```

或

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

stdio 是 standard input & output 的意思。

- 5) 考虑到 printf 和 scanf 函数使用频繁，系统允许在使用这两个函数时可不加

```
#include<stdio.h>
```

或

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

4.4 字符数据的输入输出

4.4.1 putchar 函数（字符输出函数）

putchar 函数是字符输出函数，其功能是在显示器上输出单个字符。

其一般形式为：

```
putchar(字符变量)
```

例如：

```
putchar('A');    (输出大写字母 A)
putchar(x);      (输出字符变量 x 的值)
putchar('\101'); (也是输出字符 A)
putchar('\n');   (换行)
```

对控制字符则执行控制功能，不在屏幕上显示。

使用本函数前必须要用文件包含命令：

```
#include<stdio.h>
```

或

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

【例 4.1】输出单个字符。

```
#include<stdio.h>
```

```
main() {
```

```
    char a='B', b='o', c='k';
```

```
    putchar(a);putchar(b);putchar(b);putchar(c);putchar('\t');
```

```
    putchar(a);putchar(b);
```

```
    putchar(' \n' );  
    putchar(b);putchar(c);  
}
```



4.4.2 getchar 函数（键盘输入函数）

getchar 函数的功能是从键盘上输入一个字符。

其一般形式为：

```
getchar();
```

通常把输入的字符赋予一个字符变量，构成赋值语句，如：

```
char c;  
c=getchar();
```

【例 4.2】输入单个字符。

```
#include<stdio.h>  
void main() {  
    char c;  
    printf("input a character\n");  
    c=getchar();  
    putchar(c);  
}
```



使用 getchar 函数还应注意几个问题：

- 1) getchar 函数只能接受单个字符，输入数字也按字符处理。输入多于一个字符时，只接收第一个字符。
- 2) 使用本函数前必须包含文件“stdio.h”。
- 3) 在 TC 屏幕下运行含本函数程序时，将退出 TC 屏幕进入用户屏幕等待用户输入。输入完毕再返回 TC 屏幕。
- 4) 程序最后两行可用下面两行的任意一行代替：

```
putchar(getchar());  
printf("%c", getchar());
```

4.5 格式输入与输出

4.5.1 printf 函数（格式输出函数）

printf 函数称为**格式输出函数**，其关键字最末一个字母 f 即为“格式”(format)之意。

其功能是按用户指定的格式，把指定的数据显示到显示器屏幕上。在前面的例题中我们已多次使用过这个函数。

1. printf 函数调用的一般形式

printf 函数是一个标准库函数, 它的函数原型在头文件“stdio.h”中。但作为一个特例, 不要求在使用 printf 函数之前必须包含 stdio.h 文件。

printf 函数调用的一般形式为:

printf(“格式控制字符串”, 输出表列)

其中格式控制字符串用于指定输出格式。格式控制串可由格式字符串和非格式字符串两种组成。格式字符串是以%开头的字符串, 在%后面跟有各种格式字符, 以说明输出数据的类型、形式、长度、小数位数等。如:

“%d”表示按十进制整型输出;

“%ld”表示按十进制长整型输出;

“%c”表示按字符型输出等。

非格式字符串在输出时原样照印, 在显示中起提示作用。

输出表列中给出了各个输出项, 要求格式字符串和各输出项在数量和类型上应该一一对应。

【例 4.3】

```
main()
{
    int a=88,b=89;
    printf("%d %d\n",a,b);
    printf("%d,%d\n",a,b);
    printf("%c,%c\n",a,b);
    printf("a=%d,b=%d",a,b);
}
```



本例中四次输出了 a, b 的值, 但由于格式控制串不同, 输出的结果也不相同。第四行的输出语句格式控制串中, 两格式串%d 之间加了一个空格(非格式字符), 所以输出的 a, b 值之间有一个空格。第五行的 printf 语句格式控制串中加入的是非格式字符逗号, 因此输出的 a, b 值之间加了一个逗号。第六行的格式串要求按字符型输出 a, b 值。第七行中为了提示输出结果又增加了非格式字符串。

2. 格式字符串

在 Turbo C 中格式字符串的一般形式为:

[标志][输出最小宽度][.精度][长度]类型

其中方括号[]中的项为可选项。

各项的意义介绍如下:

1) 类型: 类型字符用以表示输出数据的类型, 其格式符和意义如下表所示:

格式字符	意 义
d	以十进制形式输出带符号整数(正数不输出符号)
o	以八进制形式输出无符号整数(不输出前缀 0)
x, X	以十六进制形式输出无符号整数(不输出前缀 0x)
u	以十进制形式输出无符号整数
f	以小数形式输出单、双精度实数
e, E	以指数形式输出单、双精度实数
g, G	以%f 或%e 中较短的输出宽度输出单、双精度实数
c	输出单个字符

s	输出字符串
---	-------

2) 标志: 标志字符为-、+、#、空格四种, 其意义下表所示:

标 志	意 义
-	结果左对齐, 右边填充格
+	输出符号(正号或负号)
空格	输出值为正时冠以空格, 为负时冠以负号
#	对 c, s, d, u 类无影响; 对 o 类, 在输出时加前缀 0; 对 x 类, 在输出时加前缀 0x; 对 e, g, f 类当结果有小数时才给出小数点

3) 输出最小宽度: 用十进制整数来表示输出的最少位数。若实际位数多于定义的宽度, 则按实际位数输出, 若实际位数少于定义的宽度则补以空格或 0。

4) 精度: 精度格式符以“.”开头, 后跟十进制整数。本项的意义是: 如果输出数字, 则表示小数的位数; 如果输出的是字符, 则表示输出字符的个数; 若实际位数大于所定义的精度数, 则截去超过的部分。

5. 长度: 长度格式符为 h, l 两种, h 表示按短整型量输出, l 表示按长整型量输出。

【例 4.4】

```
main()
{
    int a=15;
    float b=123.1234567;
    double c=12345678.1234567;
    char d='p';
    printf("a=%d,%5d,%o,%x\n", a, a, a, a);
    printf("b=%f,%lf,%5.4lf,%e\n", b, b, b, b);
    printf("c=%lf,%f,%8.4lf\n", c, c, c);
    printf("d=%c,%8c\n", d, d);
}
```



本例第七行中以四种格式输出整型变量 a 的值, 其中“%5d”要求输出宽度为 5, 而 a 值为 15 只有两位故补三个空格。第八行中以四种格式输出实型量 b 的值。其中“%f”和“%lf”格式的输出生相同, 说明“l”符对“f”类型无影响。“%5.4lf”指定输出宽度为 5, 精度为 4, 由于实际长度超过 5 故应该按实际位数输出, 小数位数超过 4 位部分被截去。第九行输出双精度实数, “%8.4lf”由于指定精度为 4 位故截去了超过 4 位的部分。第十行输出字符 d, 其中“%8c”指定输出宽度为 8 故在输出字符 p 之前补加 7 个空格。

使用 printf 函数时还要注意一个问题, 那就是输出表列中的求值顺序。不同的编译系统不一定相同, 可以从左到右, 也可从右到左。Turbo C 是按从右到左进行的。请看下面两个例子:

【例 4.5】

```
main() {
    int i=8;
    printf("%d\n%d\n%d\n%d\n%d\n%d\n%d\n", ++i, --i, i++, i--, -i++, -i--);
}
```



【例 4.6】

```
main() {
    int i=8;
    printf("%d\n", ++i);
    printf("%d\n", --i);
    printf("%d\n", i++);
    printf("%d\n", i--);
    printf("%d\n", -i++);
    printf("%d\n", -i--);
}
```



这两个程序的区别是用一个 printf 语句和多个 printf 语句输出。但从结果可以看出是不同的。为什么结果会不同呢？就是因为 printf 函数对输出表中各量求值的顺序是自右至左进行的。在第一例中，先对最后一项“-i--”求值，结果为-8，然后 i 自减 1 后为 7。再对“-i++”项求值得-7，然后 i 自增 1 后为 8。再对“-i--”项求值得 8，然后 i 再自减 1 后为 7。再求“-i++”项得 7，然后 i 再自增 1 后为 8。再求“-i--”项，i 先自减 1 后输出，输出值为 7。最后才求输出表列中的第一项“++i”，此时 i 自增 1 后输出 8。

但是必须注意，求值顺序虽是自右至左，但是输出顺序还是从左至右，因此得到的结果是上述输出结果。

4.5.2 scanf 函数(格式输入函数)

scanf 函数称为格式输入函数，即按用户指定的格式从键盘上把数据输入到指定的变量之中。

1. scanf 函数的一般形式

scanf 函数是一个标准库函数，它的函数原型在头文件“stdio.h”中，与 printf 函数相同，C 语言也允许在使用 scanf 函数之前不必包含 stdio.h 文件。

scanf 函数的一般形式为：

scanf(“格式控制字符串”，地址表列)；

其中，格式控制字符串的作用与 printf 函数相同，但不能显示非格式字符串，也就是不能显示提示字符串。地址表列中给出各变量的地址。地址是由地址运算符“&”后跟变量名组成的。

例如：

&a, &b

分别表示变量 a 和变量 b 的地址。

这个地址就是编译系统在内存中给 a, b 变量分配的地址。在 C 语言中，使用了地址这个概念，这是与其它语言不同的。应该把变量的值和变量的地址这两个不同的概念区别开来。变量的地址是 C 编译系统分配的，用户不必关心具体的地址是多少。

变量的地址和变量值的关系如下：

在赋值表达式中给变量赋值，如：

a=567

则，a 为变量名，567 是变量的值，&a 是变量 a 的地址。

但在赋值号左边是变量名，不能写地址，而 scanf 函数在本质上也是给变量赋值，但要

求写变量的地址，如&a。这两者在形式上是不同的。&是一个取地址运算符，&a 是一个表达式，其功能是求变量的地址。

【例 4.7】

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



在本例中，由于 scanf 函数本身不能显示提示串，故先用 printf 语句在屏幕上输出提示，请用户输入 a、b、c 的值。执行 scanf 语句，则退出 TC 屏幕进入用户屏幕等待用户输入。用户输入 7 8 9 后按下回车键，此时，系统又将返回 TC 屏幕。在 scanf 语句的格式串中由于没有非格式字符在“%d%d%d”之间作输入时的间隔，因此在输入时要用一个以上的空格或回车键作为每两个输入数之间的间隔。如：

7 8 9
或
7
8
9

2. 格式字符串

格式字符串的一般形式为：

%[*][输入数据宽度][长度]类型

其中有方括号[]的项为任选项。各项的意义如下：

- 1) 类型：表示输入数据的类型，其格式符和意义如下表所示。

格式	字符意义
d	输入十进制整数
o	输入八进制整数
x	输入十六进制整数
u	输入无符号十进制整数
f 或 e	输入实型数(用小数形式或指数形式)
c	输入单个字符
s	输入字符串

- 2) “*”符:用以表示该输入项,读入后不赋予相应的变量，即跳过该输入值。

如：

```
scanf("%d %*d %d",&a,&b);
```

当输入为：1 2 3 时，把 1 赋予 a，2 被跳过，3 赋予 b。

- 3) 宽度:用十进制整数指定输入的宽度(即字符数)。

例如：

```
scanf("%5d",&a);
```

输入：12345678

只把 12345 赋予变量 a，其余部分被截去。

又如：

```
scanf("%4d%4d", &a, &b);
```

输入: 12345678

将把 1234 赋予 a, 而把 5678 赋予 b。

- 4) 长度: 长度格式符为 l 和 h, l 表示输入长整型数据(如 %ld) 和双精度浮点数(如 %lf)。h 表示输入短整型数据。

使用 scanf 函数还必须注意以下几点:

- 1) scanf 函数中没有精度控制, 如: scanf("%5.2f", &a); 是非法的。不能企图用此语句输入小数为 2 位的实数。
- 2) scanf 中要求给出变量地址, 如给出变量名则会出错。如 scanf("%d", a); 是非法的, 应改为 scanf("%d", &a); 才是合法的。
- 3) 在输入多个数值数据时, 若格式控制串中没有非格式字符作输入数据之间的间隔则可用空格, TAB 或回车作间隔。C 编译在碰到空格, TAB, 回车或非法数据(如对 "%d" 输入 "12A" 时, A 即为非法数据)时即认为该数据结束。
- 4) 在输入字符串数据时, 若格式控制串中无非格式字符, 则认为所有输入的字符均为有效字符。

例如:

```
scanf("%c%c%c", &a, &b, &c);
```

输入为:

```
d e f
```

则把 'd' 赋予 a, ' ' 赋予 b, 'e' 赋予 c。

只有当输入为:

```
def
```

时, 才能把 'd' 赋予 a, 'e' 赋予 b, 'f' 赋予 c。

如果在格式控制中加入空格作为间隔,

如:

```
scanf("%c %c %c", &a, &b, &c);
```

则输入时各数据之间可加空格。

【例 4.8】

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



由于 scanf 函数 "%c%c" 中没有空格, 输入 M N, 结果输出只有 M。而输入改为 MN 时则可输出 MN 两字符。

【例 4.9】

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

}



本例表示 scanf 格式控制串“%c %c”之间有空格时，输入的数据之间可以有空格间隔。

5) 如果格式控制串中有非格式字符则输入时也要输入该非格式字符。

例如：

```
scanf("%d,%d,%d",&a,&b,&c);
```

其中用非格式符“,”作间隔符，故输入时应为：

5,6,7

又如：

```
scanf("a=%d,b=%d,c=%d",&a,&b,&c);
```

则输入应为：

a=5,b=6,c=7

6) 如输入的数据与输出的类型不一致时，虽然编译能够通过，但结果将不正确。

【例 4.10】

```
main() {
    int a;
    printf("input a number\n");
    scanf("%d",&a);
    printf("%ld",a);
}
```



由于输入数据类型为整型，而输出语句的格式串中说明为长整型，因此输出结果和输入数据不符。如改动程序如下：

【例 4.11】

```
main() {
    long a;
    printf("input a long integer\n");
    scanf("%ld",&a);
    printf("%ld",a);
}
```



运行结果为：

```
input a long integer
1234567890
1234567890
```

当输入数据改为长整型后，输入输出数据相等。

【例 4.12】

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



```
printf("%d, %d, %d\n%c, %c, %c\n", a, b, c, a-32, b-32, c-32);
}
```



输入三个小写字母，输出其 ASCII 码和对应的大写字母。

【例 4.13】

```
main() {
    int a;
    long b;
    float f;
    double d;
    char c;
    printf("\nint:%d\nlong:%d\nfloat:%d\ndouble:%d\nchar:%d\n", sizeof(a), sizeof(b),
        sizeof(f), sizeof(d), sizeof(c));
}
```



输出各种数据类型的字节长度。

4.6 顺序结构程序设计举例

【例 4.14】输入三角形的三边长，求三角形面积。

已知三角形的三边长 a, b, c ，则该三角形的面积公式为：

$$area = \sqrt{s(s-a)(s-b)(s-c)},$$

其中 $s = (a+b+c)/2$

源程序如下：

```
#include<math.h>
main()
{
    float a,b,c,s,area;
    scanf("%f,%f,%f",&a,&b,&c);
    s=1.0/2*(a+b+c);
    area=sqrt(s*(s-a)*(s-b)*(s-c));
    printf("a=%7.2f,b=%7.2f,c=%7.2f,s=%7.2f\n",a,b,c,s);
    printf("area=%7.2f\n",area);
}
```



【例 4.15】求 $ax^2+bx+c=0$ 方程的根， a, b, c 由键盘输入，设 $b^2-4ac>0$ 。

求根公式为：

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$x_{1,2} = \frac{-b}{2a}$$

$$\text{令 } q = \frac{\sqrt{b^2 - 4ac}}{2a}, \quad q = \frac{\sqrt{b^2 - 4ac}}{2a}$$

则 $x_1 = p + q$

$x_2 = p - q$

源程序如下：

```
#include <math.h>
```

```
main()
```

```
{
```

```
    float a,b,c,disc,x1,x2,p,q;
```

```
    scanf("a=%f,b=%f,c=%f",&a,&b,&c);
```

```
    disc=b*b-4*a*c;
```

```
    p=-b/(2*a);
```

```
    q=sqrt(disc)/(2*a);
```

```
    x1=p+q;x2=p-q;
```

```
    printf("\nx1=%5.2f\nx2=%5.2f\n",x1,x2);
```

```
}
```



5	分支结构程序.....	1
5.1	关系运算符和表达式.....	1
5.1.1	关系运算符及其优先次序.....	1
5.1.2	关系表达式.....	1
5.2	逻辑运算符和表达式.....	2
5.2.1	逻辑运算符极其优先次序.....	2
5.2.2	逻辑运算的值.....	3
5.2.3	逻辑表达式.....	3
5.3	if 语句.....	4
5.3.1	if 语句的三种形式.....	4
5.3.2	if 语句的嵌套.....	7
5.3.3	条件运算符和条件表达式.....	9
5.4	switch 语句.....	10
5.5	程序举例.....	11

5 分支结构程序

5.1 关系运算符和表达式

在程序中经常需要比较两个量的大小关系，以决定程序下一步的工作。比较两个量的运算符称为关系运算符。

5.1.1 关系运算符及其优先次序

在 C 语言中有以下关系运算符：

- 1) < 小于
- 2) <= 小于或等于
- 3) > 大于
- 4) >= 大于或等于
- 5) == 等于
- 6) != 不等于

关系运算符都是双目运算符，其结合性均为左结合。关系运算符的优先级低于算术运算符，高于赋值运算符。在六个关系运算符中，<, <=, >, >= 的优先级相同，高于 == 和 !=，== 和 != 的优先级相同。

5.1.2 关系表达式

关系表达式的一般形式为：

表达式 关系运算符 表达式

例如：

a+b>c-d

```

x>3/2
'a'+1<c
-i-5*j==k+1

```

都是合法的关系表达式。由于表达式也可以又是关系表达式。因此也允许出现嵌套的情况。例如：

```

a>(b>c)
a!=(c==d)

```

等。

关系表达式的值是“真”和“假”，用“1”和“0”表示。

如：

5>0 的值为“真”，即为 1。

(a=3)>(b=5) 由于 3>5 不成立，故其值为假，即为 0。

【例 5.1】

```

main() {
    char c='k';
    int i=1, j=2, k=3;
    float x=3e+5, y=0.85;
    printf("%d,%d\n", 'a'+5<c, -i-2*j>=k+1);
    printf("%d,%d\n", 1<j<5, x-5.25<=x+y);
    printf("%d,%d\n", i+j+k==2*j, k==j==i+5);
}

```



在本例中求出了各种关系运算符的值。字符变量是以它对应的 ASCII 码参与运算的。对于含多个关系运算符的表达式，如 $k==j==i+5$ ，根据运算符的左结合性，先计算 $k==j$ ，该式不成立，其值为 0，再计算 $0==i+5$ ，也不成立，故表达式值为 0。

5.2 逻辑运算符和表达式

5.2.1 逻辑运算符及其优先次序

C 语言中提供了三种逻辑运算符：

- 1) && 与运算
- 2) || 或运算
- 3) ! 非运算

与运算符&&和或运算符||均为双目运算符。具有左结合性。非运算符!为单目运算符，具有右结合性。逻辑运算符和其它运算符优先级的关系可表示如下：

! (非) \rightarrow && (与) \rightarrow || (或)

! (非)
 算术运算符
 关系运算符
 && 和 ||
 赋值运算符



“&&”和“||”低于关系运算符，“!”高于算术运算符。

按照运算符的优先顺序可以得出：

$a > b \ \&\& \ c > d$ 等价于 $(a > b) \&\& (c > d)$
 $!b == c \ || \ d < a$ 等价于 $((!b) == c) \ || \ (d < a)$
 $a + b > c \&\& x + y < b$ 等价于 $((a + b) > c) \&\& ((x + y) < b)$

5.2.2 逻辑运算的值

逻辑运算的值也为“真”和“假”两种，用“1”和“0”来表示。其求值规则如下：

1. 与运算 &&：参与运算的两个量都为真时，结果才为真，否则为假。

例如：

$5 > 0 \ \&\& \ 4 > 2$

由于 $5 > 0$ 为真， $4 > 2$ 也为真，相与的结果也为真。

2. 或运算 ||：参与运算的两个量只要有一个为真，结果就为真。两个量都为假时，结果为假。

例如：

$5 > 0 \ || \ 5 > 8$

由于 $5 > 0$ 为真，相或的结果也就为真。

3. 非运算 !：参与运算量为真时，结果为假；参与运算量为假时，结果为真。

例如：

$!(5 > 0)$

的结果为假。

虽然 C 编译在给出逻辑运算值时，以“1”代表“真”，“0”代表“假”。但反过来在判断一个量是为“真”还是为“假”时，以“0”代表“假”，以非“0”的数值作为“真”。例如：

由于 5 和 3 均为非“0”因此 $5 \&\& 3$ 的值为“真”，即为 1。

又如：

$5 \ || \ 0$ 的值为“真”，即为 1。

5.2.3 逻辑表达式

逻辑表达式的一般形式为：

表达式 逻辑运算符 表达式

其中的表达式可以又是逻辑表达式，从而组成了嵌套的情形。

例如：

$(a \&\& b) \&\& c$

根据逻辑运算符的左结合性，上式也可写为：

`a&&b&&c`

逻辑表达式的值是式中各种逻辑运算的最后值，以“1”和“0”分别代表“真”和“假”。

【例 5.2】

```
main() {
    char c='k';
    int i=1, j=2, k=3;
    float x=3e+5, y=0.85;
    printf("%d,%d\n", !x*!y, !!!x);
    printf("%d,%d\n", x||i&&j-3, i<j&&x<y);
    printf("%d,%d\n", i==5&&c&&(j=8), x+y||i+j+k);
}
```



本例中!x 和!y 分别为 0，!x*!y 也为 0，故其输出值为 0。由于 x 为非 0，故!!!x 的逻辑值为 0。对 x|| i && j-3 式，先计算 j-3 的值为非 0，再求 i && j-3 的逻辑值为 1，故 x|| i&&j-3 的逻辑值为 1。对 i<j&&x<y 式，由于 i<j 的值为 1，而 x<y 为 0 故表达式的值为 1，0 相与，最后为 0，对 i==5&&c&&(j=8)式，由于 i==5 为假，即值为 0，该表达式由两个与运算组成，所以整个表达式的值为 0。对于式 x+ y||i+j+k 由于 x+y 的值为非 0，故整个或表达式的值为 1。

5.3 if 语句

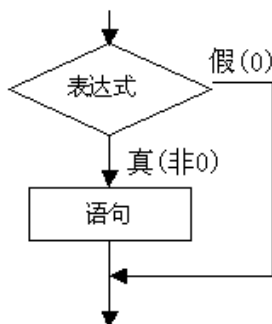
用 if 语句可以构成分支结构。它根据给定的条件进行判断，以决定执行某个分支程序段。C 语言的 if 语句有三种基本形式。

5.3.1 if 语句的三种形式

1. 第一种形式为基本形式：if

if(表达式) 语句

其语义是：如果表达式的值为真，则执行其后的语句， 否则不执行该语句。其过程可表示为下图。



【例 5.3】

```

main() {
    int a,b,max;
    printf("\n input two numbers:  ");
    scanf("%d%d",&a,&b);
    max=a;
    if (max<b) max=b;
    printf("max=%d",max);
}

```



本例程序中，输入两个数 a, b。把 a 先赋予变量 max，再用 if 语句判别 max 和 b 的大小，如 max 小于 b，则把 b 赋予 max。因此 max 中总是大数，最后输出 max 的值。

2. 第二种形式为：if-else

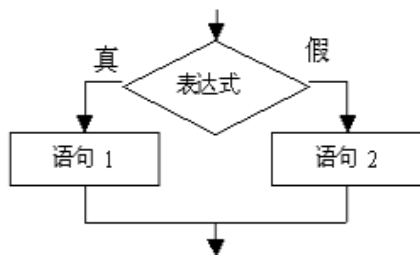
if(表达式)

语句 1;

else

语句 2;

其语义是：如果表达式的值为真，则执行语句 1，否则执行语句 2。其执行过程可表示为下图。



【例 5.4】

```

main() {
    int a, b;
    printf("input two numbers:  ");
    scanf("%d%d",&a,&b);
    if(a>b)
        printf("max=%d\n",a);
    else
        printf("max=%d\n",b);
}

```



输入两个整数，输出其中的大数。

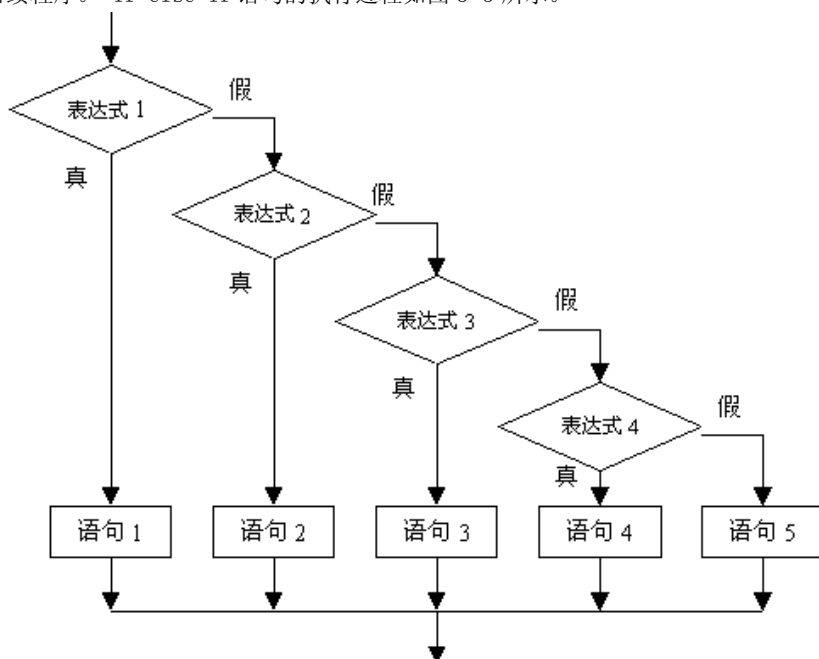
改用 if-else 语句判别 a, b 的大小，若 a 大，则输出 a，否则输出 b。

3. 第三种形式为 if-else-if 形式

前二种形式的 if 语句一般都用于两个分支的情况。当有多个分支选择时，可采用 if-else-if 语句，其一般形式为：

```
if(表达式 1)
    语句 1;
else if(表达式 2)
    语句 2;
else if(表达式 3)
    语句 3;
...
else if(表达式 m)
    语句 m;
else
    语句 n;
```

其语义是：依次判断表达式的值，当出现某个值为真时，则执行其对应的语句。然后跳到整个 if 语句之外继续执行程序。如果所有的表达式均为假，则执行语句 n。然后继续执行后续程序。if-else-if 语句的执行过程如图 3-3 所示。



【例 5.5】

```
#include "stdio.h"
main() {
    char c;
    printf("input a character:  ");
    c=getchar();
    if(c<32)
        printf("This is a control character\n");
    else if(c>='0' && c<='9')
```



```

    printf("This is a digit\n");
else if(c>='A' && c<='Z')
    printf("This is a capital letter\n");
else if(c>='a' && c<='z')
    printf("This is a small letter\n");
else
    printf("This is an other character\n");
}

```



本例要求判别键盘输入字符的类别。可以根据输入字符的 ASCII 码来判别类型。由 ASCII 码表可知 ASCII 值小于 32 的为控制字符。在“0”和“9”之间的为数字，在“A”和“Z”之间为大写字母，在“a”和“z”之间为小写字母，其余则为其它字符。这是一个多分支选择的问题，用 if-else-if 语句编程，判断输入字符 ASCII 码所在的范围，分别给出不同的输出。例如输入为“g”，输出显示它为小写字母。

4. 在使用 if 语句中还应注意以下问题：

- 1) 在三种形式的 if 语句中，在 if 关键字之后均为表达式。该表达式通常是逻辑表达式或关系表达式，但也可以是其它表达式，如赋值表达式等，甚至也可以是一个变量。

例如：

```
if(a=5) 语句;
```

```
if(b) 语句;
```

都是允许的。只要表达式的值为非 0，即为“真”。

如在：

```
if(a=5)...
```

中表达式的值永远为非 0，所以其后的语句总是要执行的，当然这种情况在程序中不一定会出现，但在语法上是合法的。

又如，有程序段：

```

if(a=b)
    printf("%d",a);
else
    printf("a=0");

```

本语句的语义是，把 b 值赋予 a，如为非 0 则输出该值，否则输出“a=0”字符串。这种用法在程序中是经常出现的。

- 2) 在 if 语句中，条件判断表达式必须用括号括起来，在语句之后必须加分号。
- 3) 在 if 语句的三种形式中，所有的语句应为单个语句，如果要想在满足条件时执行一组(多个)语句，则必须把这一组语句用 {} 括起来组成一个复合语句。但要注意的是在 } 之后不能再加分号。

例如：

```

if(a>b)
{a++;
 b++;}
else
{a=0;
 b=10;}

```

5.3.2 if 语句的嵌套

当 if 语句中的执行语句又是 if 语句时，则构成了 if 语句嵌套的情形。

其一般形式可表示如下：

```
if(表达式)
```

```
if 语句;
```

或者为

```
if(表达式)
```

```
if 语句;
```

```
else
```

```
if 语句;
```

在嵌套内的 if 语句可能又是 if-else 型的，这将会出现多个 if 和多个 else 重叠的情况，这时要特别注意 if 和 else 的配对问题。

例如：

```
if(表达式 1)
```

```
if(表达式 2)
```

```
语句 1;
```

```
else
```

```
语句 2;
```

其中的 else 究竟是与哪一个 if 配对呢？

应该理解为：

```
if(表达式 1)
    if(表达式 2)
        语句 1;
    else
        语句 2;
```

还是应理解为：

```
if(表达式 1)
    if(表达式 2)
        语句 1;
else
    语句 2;
```

为了避免这种二义性，C 语言规定，else 总是与它前面最近的 if 配对，因此对上述例子应按前一种情况理解。

【例 5.6】

```
main() {
    int a, b;
    printf("please input A,B:   ");
    scanf("%d%d", &a, &b);
    if(a!=b)
    if(a>b) printf("A>B\n");
    else    printf("A<B\n");
    else    printf("A=B\n");
}
```

}



比较两个数的大小关系。

本例中用了 if 语句的嵌套结构。采用嵌套结构实质上是为了进行多分支选择，实际上有三种选择即 A>B、A<B 或 A=B。这种问题用 if-else-if 语句也可以完成。而且程序更加清晰。因此，在一般情况下较少使用 if 语句的嵌套结构。以使程序更便于阅读理解。

【例 5.7】

```
main() {
    int a, b;
    printf("please input A,B:      ");
    scanf("%d%d", &a, &b);
    if(a==b) printf("A=B\n");
    else if(a>b) printf("A>B\n");
    else printf("A<B\n");
}
```



5.3.3 条件运算符和条件表达式

如果在条件语句中，只执行单个的赋值语句时，常可使用条件表达式来实现。不但使程序简洁，也提高了运行效率。

条件运算符为?和:，它是一个三目运算符，即有三个参与运算的量。

由条件运算符组成条件表达式的一般形式为：

表达式 1? 表达式 2: 表达式 3

其求值规则为：如果表达式 1 的值为真，则以表达式 2 的值作为条件表达式的值，否则以表达式 3 的值作为整个条件表达式的值。

条件表达式通常用于赋值语句之中。

例如条件语句：

```
if(a>b) max=a;
else max=b;
```

可用条件表达式写为

```
max=(a>b)?a:b;
```

执行该语句的语义是：如 a>b 为真，则把 a 赋予 max，否则把 b 赋予 max。

使用条件表达式时，还应注意以下几点：

- 1) 条件运算符的运算优先级低于关系运算符和算术运算符，但高于赋值符。

因此

```
max=(a>b)?a:b
```

可以去掉括号而写为

```
max=a>b?a:b
```

- 2) 条件运算符?和: 是一对运算符，不能分开单独使用。
- 3) 条件运算符的结合方向是自右至左。

例如：

```
a>b?a:c>d?c:d
```

应理解为

```
a>b?a:(c>d?c:d)
```

这也就是条件表达式嵌套的情形，即其中的表达式 3 又是一个条件表达式。

【例 5.8】

```
main() {
    int a, b, max;
    printf("\n input two numbers:  ");
    scanf("%d%d", &a, &b);
    printf("max=%d", a>b?a:b);
}
```



用条件表达式对上例重新编程，输出两个数中的大数。

5.4 switch 语句

C 语言还提供了另一种用于多分支选择的 switch 语句，其一般形式为：

```
switch(表达式) {
    case 常量表达式 1: 语句 1;
    case 常量表达式 2: 语句 2;
    ...
    case 常量表达式 n: 语句 n;
    default          : 语句 n+1;
}
```

其语义是：计算表达式的值。并逐个与其后的常量表达式值相比较，当表达式的值与某个常量表达式的值相等时，即执行其后的语句，然后不再进行判断，继续执行后面所有 case 后的语句。如表达式的值与所有 case 后的常量表达式均不相同时，则执行 default 后的语句。

【例 4.9】

```
main() {
    int a;
    printf("input integer number:  ");
    scanf("%d", &a);
    switch (a) {
        case 1:printf("Monday\n");
        case 2:printf("Tuesday\n");
        case 3:printf("Wednesday\n");
        case 4:printf("Thursday\n");
        case 5:printf("Friday\n");
        case 6:printf("Saturday\n");
        case 7:printf("Sunday\n");
        default:printf("error\n");
    }
}
```

}



本程序是要求输入一个数字，输出一个英文单词。但是当输入 3 之后，却执行了 case3 以及以后的所有语句，输出了 Wednesday 及以后的所有单词。这当然是不希望。为什么会出现这种情况呢？这恰恰反应了 switch 语句的一个特点。在 switch 语句中，“case 常量表达式”只相当于一个语句标号，表达式的值和某标号相等则转向该标号执行，但不能在执行完该标号的语句后自动跳出整个 switch 语句，所以出现了继续执行所有后面 case 语句的情况。这是与前面介绍的 if 语句完全不同的，应特别注意。为了避免上述情况，C 语言还提供了一种 break 语句，专用于跳出 switch 语句，break 语句只有关键字 break，没有参数。在后面还将详细介绍。修改例题的程序，在每一 case 语句之后增加 break 语句，使每一次执行之后均可跳出 switch 语句，从而避免输出不应有的结果。

【例 4.10】

```
main() {
    int a;
    printf("input integer number:  ");
    scanf("%d", &a);
    switch (a) {
        case 1: printf("Monday\n"); break;
        case 2: printf("Tuesday\n"); break;
        case 3: printf("Wednesday\n"); break;
        case 4: printf("Thursday\n"); break;
        case 5: printf("Friday\n"); break;
        case 6: printf("Saturday\n"); break;
        case 7: printf("Sunday\n"); break;
        default: printf("error\n");
    }
}
```



在使用 switch 语句时还应注意以下几点：

- 1) 在 case 后的各常量表达式的值不能相同，否则会出现错误。
- 2) 在 case 后，允许有多个语句，可以不用 {} 括起来。
- 3) 各 case 和 default 子句的先后顺序可以变动，而不会影响程序执行结果。
- 4) default 子句可以省略不用。

5.5 程序举例

【例 4.11】输入三个整数，输出最大数和最小数。

```
main() {
    int a, b, c, max, min;
    printf("input three numbers:  ");
    scanf("%d%d%d", &a, &b, &c);
    if (a > b)
```

```

        {max=a;min=b;}
    else
        {max=b;min=a;}
    if(max<c)
        max=c;
    else
        if(min>c)
            min=c;
    printf("max=%d\nmin=%d", max, min);
}

```



本程序中，首先比较输入的 a, b 的大小，并把大数装入 max，小数装入 min 中，然后再与 c 比较，若 max 小于 c，则把 c 赋予 max；如果 c 小于 min，则把 c 赋予 min。因此 max 内总是最大数，而 min 内总是最小数。最后输出 max 和 min 的值即可。

【例 4.12】 计算器程序。用户输入运算数和四则运算符，输出计算结果。

```

main() {
    float a, b;
    char c;
    printf("input expression: a+(-,*,/)b \n");
    scanf("%f%c%f", &a, &c, &b);
    switch(c) {
        case '+': printf("%f\n", a+b); break;
        case '-': printf("%f\n", a-b); break;
        case '*': printf("%f\n", a*b); break;
        case '/': printf("%f\n", a/b); break;
        default: printf("input error\n");
    }
}

```



本例可用于四则运算求值。switch 语句用于判断运算符，然后输出运算值。当输入运算符不是 +, -, *, / 时给出错误提示。

6 循环控制.....	1
6.1 概述.....	1
6.2 goto 语句以及用 goto 语句构成循环.....	1
6.3 while 语句.....	2
6.4 do-while 语句.....	4
6.5 for 语句.....	6
6.6 循环的嵌套.....	9
6.7 几种循环的比较.....	9
6.8 break 和 continue 语句.....	9
6.8.1 break 语句.....	9
6.8.2 continue 语句.....	10
6.9 程序举例.....	11

6 循环控制

6.1 概述

循环结构是程序中一种很重要的结构。其特点是，在给定条件成立时，反复执行某程序段，直到条件不成立为止。给定的条件称为循环条件，反复执行的程序段称为循环体。C 语言提供了多种循环语句，可以组成各种不同形式的循环结构。

- 1) 用 goto 语句和 if 语句构成循环；
- 2) 用 while 语句；
- 3) 用 do-while 语句；
- 4) 用 for 语句；

6.2 goto 语句以及用 goto 语句构成循环

goto 语句是一种无条件转移语句，与 BASIC 中的 goto 语句相似。goto 语句的使用格式为：

goto 语句标号；

其中标号是一个有效的标识符，这个标识符加上一个“:”一起出现在函数内某处，执行 goto 语句后，程序将跳转到该标号处并执行其后的语句。另外标号必须与 goto 语句同处于一个函数中，但可以不在一个循环层中。通常 goto 语句与 if 条件语句连用，当满足某一条件时，程序跳到标号处运行。

goto 语句通常不用，主要因为它将使程序层次不清，且不易读，但在多层嵌套退出时，用 goto 语句则比较合理。

【例 6.1】用 goto 语句和 if 语句构成循环， $\sum_{n=1}^{100} n$ 。

```
main()
{
```

```

    int i, sum=0;
    i=1;
loop:  if(i<=100)
        {sum=sum+i;
          i++;
          goto loop;}
    printf("%d\n", sum);
}

```



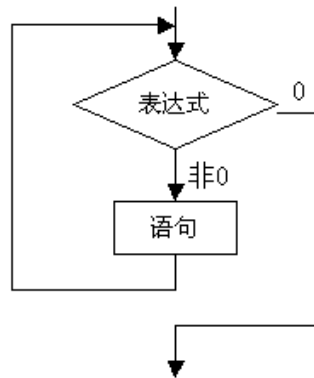
6.3 while 语句

while 语句的一般形式为：

while(表达式) 语句

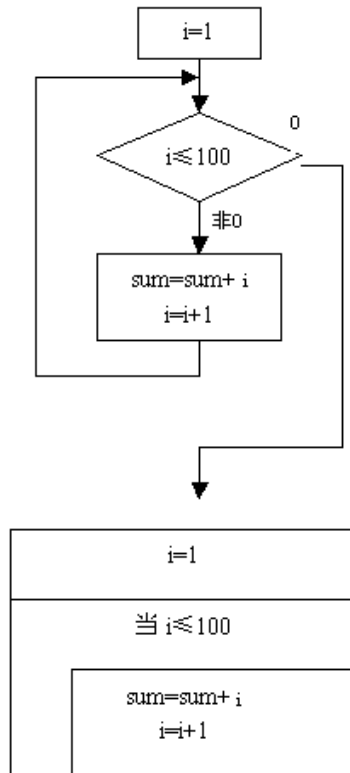
其中表达式是循环条件，语句为循环体。

while 语句的语义是：计算表达式的值，当值为真(非 0)时，执行循环体语句。其执行过程可用下图表示。



【例 6.2】用 while 语句求 $\sum_{n=1}^{100} n$ 。

用传统流程图和 N-S 结构流程图表示算法，见图：



```

main()
{
    int i, sum=0;
    i=1;
    while(i<=100)
    {
        sum=sum+i;
        i++;
    }
    printf("%d\n", sum);
}

```



【例 6.3】统计从键盘输入一行字符的个数。

```

#include <stdio.h>
main() {
    int n=0;
    printf("input a string:\n");
    while(getchar()!='\n') n++;
    printf("%d", n);
}

```



本例程序中的循环条件为 `getchar() != '\n'`，其意义是，只要从键盘输入的字符不是回车就继续循环。循环体 `n++` 完成对输入字符个数计数。从而程序实现了对输入一行字符的字符个数计数。

使用 `while` 语句应注意以下几点：

- 1) `while` 语句中的表达式一般是关系表达或逻辑表达式，只要表达式的值为真 (非 0) 即可继续循环。

【例 6.4】

```
main() {
    int a=0,n;
    printf("\n input n:  ");
    scanf("%d",&n);
    while (n-->0)
        printf("%d ",a++*2);
}
```



本例程序将执行 `n` 次循环，每执行一次，`n` 值减 1。循环体输出表达式 `a++*2` 的值。该表达式等效于 `(a*2; a++)`。

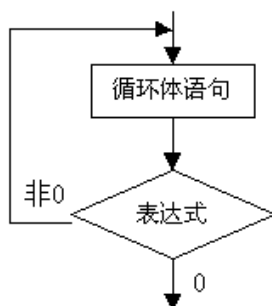
- 2) 循环体如包括有一个以上的语句，则必须用 `{}` 括起来，组成复合语句。

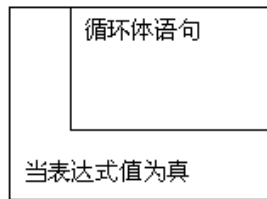
6.4 do-while 语句

`do-while` 语句的一般形式为：

```
do
    语句
while(表达式);
```

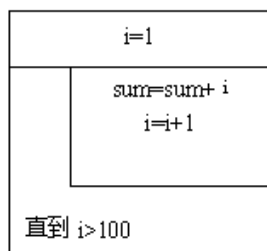
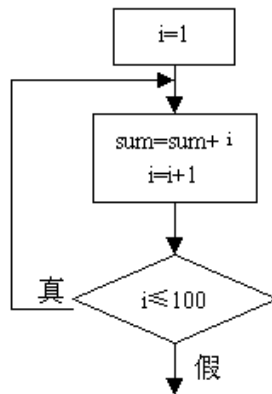
这个循环与 `while` 循环的不同在于：它先执行循环中的语句，然后再判断表达式是否为真，如果为真则继续循环；如果为假，则终止循环。因此，`do-while` 循环至少要执行一次循环语句。其执行过程可用下图表示。





【例 6.5】用 do-while 语句求 $\sum_{n=1}^{100} n$ 。

用传统流程图和 N-S 结构流程图表示算法，见图：



```
main()
{
    int i,sum=0;
    i=1;
    do
    {
        sum=sum+i;
        i++;
    }
    while(i<=100)
    printf("%d\n",sum);
}
```



同样当有许多语句参加循环时，要用“{”和“}”把它们括起来。

【例 6.6】while 和 do-while 循环比较。

```
(1) main()
{
    int sum=0, i;
    scanf("%d", &i);
    while(i<=10)
    {
        sum=sum+i;
        i++;
    }
    printf("sum=%d", sum);
}
```



```
(2) main()
{
    int sum=0, i;
    scanf("%d", &i);
    do
    {
        sum=sum+i;
        i++;
    }
    while(i<=10);
    printf("sum=%d", sum);
}
```



6.5 for 语句

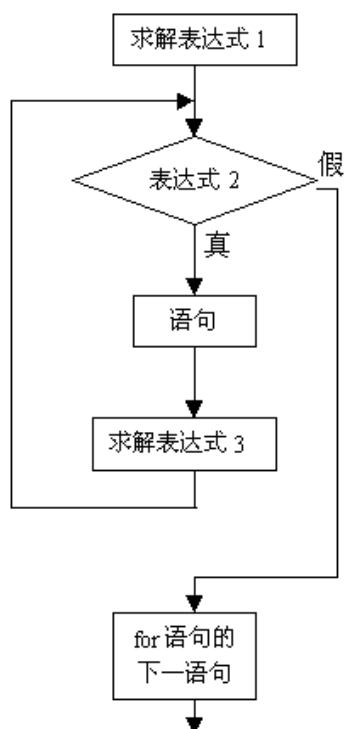
在 C 语言中，for 语句使用最为灵活，它完全可以取代 while 语句。它的一般形式为：

for(表达式 1; 表达式 2; 表达式 3) 语句

它的执行过程如下：

- 1) 先求解表达式 1。
- 2) 求解表达式 2，若其值为真（非 0），则执行 for 语句中指定的内嵌语句，然后执行下面第 3) 步；若其值为假（0），则结束循环，转到第 5) 步。
- 3) 求解表达式 3。
- 4) 转回上面第 2) 步继续执行。
- 5) 循环结束，执行 for 语句下面的一个语句。

其执行过程可用下图表示。



for 语句最简单的应用形式也是最容易理解的形式如下：

for(循环变量赋初值; 循环条件; 循环变量增量) 语句

循环变量赋初值总是一个赋值语句，它用来给循环控制变量赋初值；循环条件是一个关系表达式，它决定什么时候退出循环；循环变量增量，定义循环控制变量每循环一次后按什么方式变化。这三个部分之间用“；”分开。

例如：

```
for(i=1; i<=100; i++) sum=sum+i;
```

先给 i 赋初值 1，判断 i 是否小于等于 100，若是则执行语句，之后值增加 1。再重新判断，直到条件为假，即 i>100 时，结束循环。

相当于：

```
i=1;
while (i<=100)
{
    sum=sum+i;
    i++;
}
```

对于 for 循环中语句的一般形式，就是如下的 while 循环形式：

```
表达式 1;
while (表达式 2)
{
    语句
    表达式 3;
}
```

注意：

- 1) for 循环中的“表达式 1（循环变量赋初值）”、“表达式 2(循环条件)”和“表达式 3(循

环变量增量)”都是选择项，即可以缺省，但“;”不能缺省。

- 2) 省略了“表达式 1 (循环变量赋初值)”，表示不对循环控制变量赋初值。
- 3) 省略了“表达式 2 (循环条件)”，则不做其它处理时便成为死循环。

例如：

```
for (i=1;;i++) sum=sum+i;
```

相当于：

```
i=1;
while (1)
{sum=sum+i;
 i++;}
```

- 4) 省略了“表达式 3 (循环变量增量)”，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。

例如：

```
for (i=1;i<=100;)
{sum=sum+i;
 i++;}
```

- 5) 省略了“表达式 1 (循环变量赋初值)”和“表达式 3 (循环变量增量)”。

例如：

```
for (;i<=100;)
{sum=sum+i;
 i++;}
```

相当于：

```
while (i<=100)
{sum=sum+i;
 i++;}
```

- 6) 3 个表达式都可以省略。

例如：

```
for (;; )语句
```

相当于：

```
while (1) 语句
```

- 7) 表达式 1 可以是设置循环变量的初值的赋值表达式，也可以是其他表达式。

例如：

```
for (sum=0;i<=100;i++) sum=sum+i;
```

- 8) 表达式 1 和表达式 3 可以是一个简单表达式也可以是逗号表达式。

```
for (sum=0, i=1;i<=100;i++) sum=sum+i;
```

或：

```
for (i=0, j=100;i<=100;i++, j--) k=i+j;
```

- 9) 表达式 2 一般是关系表达式或逻辑表达式，但也可是数值表达式或字符表达式，只要其值非零，就执行循环体。

例如：

```
for (i=0; (c=getchar())!='\n'; i+=c);
```

又如：

```
for (; (c=getchar())!='\n';)
printf("%c", c);
```

6.6 循环的嵌套

【例 6.7】

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



6.7 几种循环的比较

- 1) 四种循环都可以用来处理同一个问题，一般可以互相代替。但一般不提倡用 `goto` 型循环。
- 2) `while` 和 `do-while` 循环，循环体中应包括使循环趋于结束的语句。`for` 语句功能最强。
- 3) 用 `while` 和 `do-while` 循环时，循环变量初始化的操作应在 `while` 和 `do-while` 语句之前完成，而 `for` 语句可以在表达式 1 中实现循环变量的初始化。

6.8 break 和 continue 语句

6.8.1 break 语句

`break` 语句通常用在循环语句和开关语句中。当 `break` 用于开关语句 `switch` 中时, 可使程序跳出 `switch` 而执行 `switch` 以后的语句; 如果没有 `break` 语句, 则将成为一个死循环而无法退出。`break` 在 `switch` 中的用法已在前面介绍开关语句时的例子中碰到, 这里不再举例。

当 `break` 语句用于 `do-while`、`for`、`while` 循环语句中时, 可使程序终止循环而执行循环后面的语句, 通常 `break` 语句总是与 `if` 语句联在一起。即满足条件时便跳出循环。

【例 6.8】

```
main()
{
    int i=0;
    char c;
    while(1)                /*设置循环*/
    {
        c='0';              /*变量赋初值*/
    }
}
```

```

while(c!=13&& c!=27) /*键盘接收字符直到按回车或 Esc 键*/
{
    c=getch();
    printf("%c\n", c);
}
if(c==27)
    break;          /*判断若按 Esc 键则退出循环*/
i++;
printf("The No. is %d\n", i);
}
printf("The end");
}

```



注意:

- 1) break 语句对 if-else 的条件语句不起作用。
- 2) 在多层循环中, 一个 break 语句只向外跳一层。

6.8.2 continue 语句

continue 语句的作用是跳过循环本中剩余的语句而强行执行下一次循环。continue 语句只用在 for、while、do-while 等循环体中, 常与 if 条件语句一起使用, 用来加速循环。其执行过程可用下图表示。

1) while(表达式 1)

```

{ .....
    if(表达式 2) break;
    .....
}

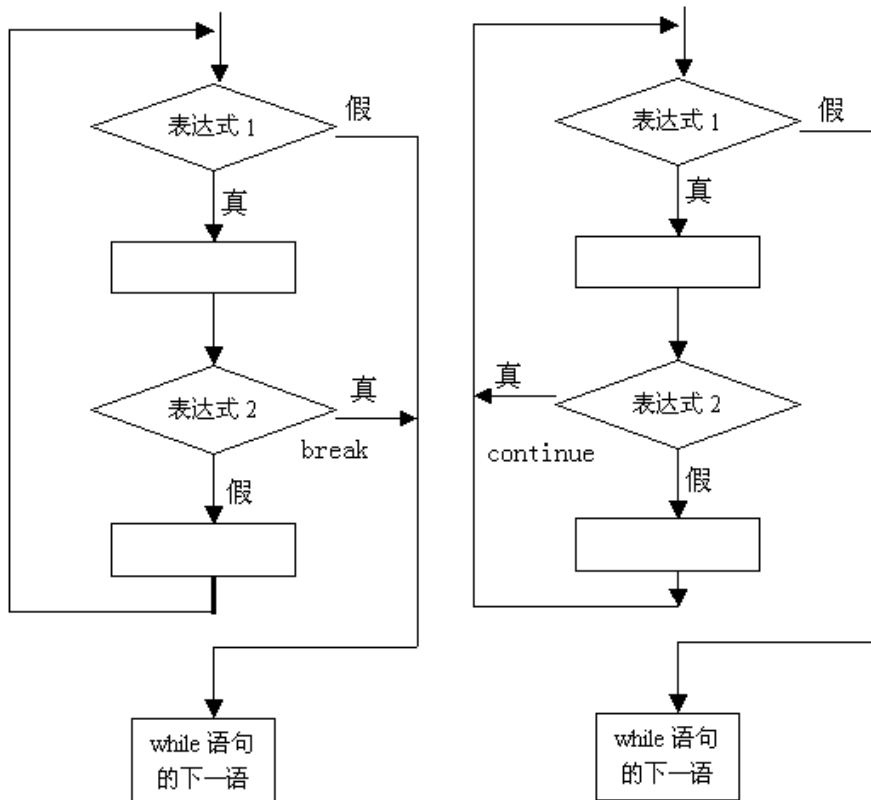
```

2) while(表达式 1)

```

{ .....
    if(表达式 2) continue;
    .....
}

```

【例 6.9】

```

main()
{
    char c;
    while(c!=13)    /*不是回车符则循环*/
    {
        c=getch();
        if(c==0X1B)
            continue; /*若按 Esc 键不输出便进行下次循环*/
        printf("%c\n", c);
    }
}

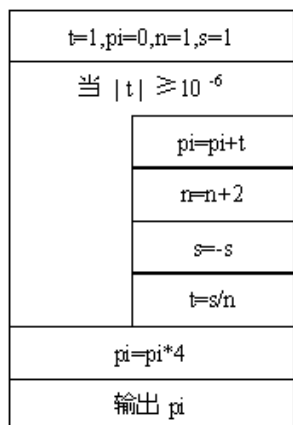
```



6.9 程序举例

【例 6.10】用 $\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$ 公式求 π 。

N-S 流程图:



```
#include<math.h>
main()
{
    int s;
    float n, t, pi;
    t=1, pi=0; n=1.0; s=1;
    while(fabs(t)>1e-6)
    {
        pi=pi+t;
        n=n+2;
        s=-s;
        t=s/n;
    }
    pi=pi*4;
```

```
printf("pi=%10.6f\n", pi);
}
```



【例 6.11】判断 m 是否素数。

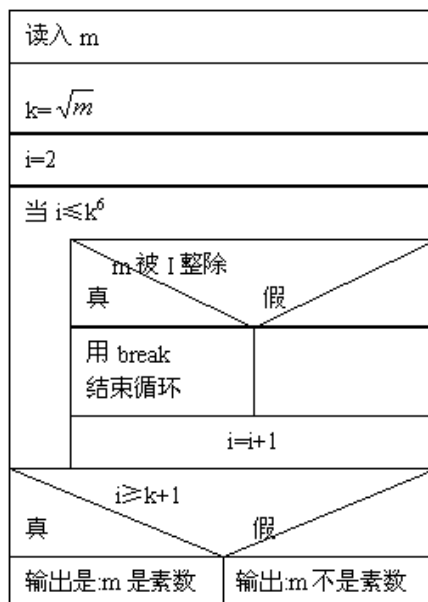
N-S 流程图:

```
#include<math.h>
main()
{
    int m, i, k;
    scanf("%d", &m);
    k=sqrt(m);
    for(i=2; i<=k; i++)
        if(m%i==0) break;
    if(i>=k+1)
        printf("%d is a prime number\n", m);
    else
        printf("%d is not a prime number\n", m);
}
```



【例 6.12】求 100 至 200 间的全部素数。

```
#include<math.h>
main()
{
    int m, i, k, n=0;
    for(m=101; m<=200; m=m+2)
    {
        k=sqrt(m);
        for(i=2; i<=k; i++)
```



```
        if (m%i==0) break;
        if (i>=k+1)
            {printf("%d",m);
              n=n+1;}
        if (n%n==0) printf("\n");
    }
    printf("\n");
}
```



7	数组.....	1
7.1	一维数组的定义和引用.....	1
7.1.1	一维数组的定义方式.....	1
7.1.2	一维数组元素的引用.....	2
7.1.3	一维数组的初始化.....	4
7.1.4	一维数组程序举例.....	4
7.2	二维数组的定义和引用.....	6
7.2.1	二维数组的定义.....	6
7.2.2	二维数组元素的引用.....	6
7.2.3	二维数组的初始化.....	7
7.2.4	二维数组程序举例.....	9
7.3	字符数组.....	9
7.3.1	字符数组的定义.....	9
7.3.2	字符数组的初始化.....	9
7.3.3	字符数组的引用.....	10
7.3.4	字符串和字符串结束标志.....	10
7.3.5	字符数组的输入输出.....	10
7.3.6	字符串处理函数.....	12
7.4	程序举例.....	14
7.5	本章小结.....	17

7 数组

在程序设计中，为了处理方便，把具有相同类型的若干变量按有序的形式组织起来。这些按序排列的同类数据元素的集合称为数组。在 C 语言中，数组属于构造数据类型。一个数组可以分解为多个数组元素，这些数组元素可以是基本数据类型或是构造类型。因此按数组元素的类型不同，数组又可分为数值数组、字符数组、指针数组、结构数组等各种类别。本章介绍数值数组和字符数组，其余的在以后各章陆续介绍。

7.1 一维数组的定义和引用

7.1.1 一维数组的定义方式

在 C 语言中使用数组必须先进行定义。

一维数组的定义方式为：

类型说明符 数组名 [常量表达式]；

其中：

类型说明符是任一种基本数据类型或构造数据类型。

数组名是用户定义的数组标识符。

方括号中的常量表达式表示数据元素的个数，也称为数组的长度。

例如：

```
int a[10];           说明整型数组 a，有 10 个元素。
float b[10],c[20];   说明实型数组 b，有 10 个元素，实型数组 c，有 20 个元素。
char ch[20];         说明字符数组 ch，有 20 个元素。
```

对于数组类型说明应注意以下几点：

- 1) 数组的类型实际是指数组元素的取值类型。对于同一个数组，其所有元素的数据类型都是相同的。
- 2) 数组名的书写规则应符合标识符的书写规定。
- 3) 数组名不能与其它变量名相同。

例如：

```
main()
{
    int a;
    float a[10];
    .....
}
```

是错误的。

- 4) 方括号中常量表达式表示数组元素的个数，如 a[5]表示数组 a 有 5 个元素。但是其下标从 0 开始计算。因此 5 个元素分别为 a[0],a[1],a[2],a[3],a[4]。
- 5) 不能在方括号中用变量来表示元素的个数，但是可以是符号常数或常量表达式。

例如：

```
#define FD 5
main()
{
    int a[3+2],b[7+FD];
    .....
}
```

是合法的。

但是下述说明方式是错误的。

```
main()
{
    int n=5;
    int a[n];
    .....
}
```

- 6) 允许在同一个类型说明中，说明多个数组和多个变量。

例如：

```
int a,b,c,d,k1[10],k2[20];
```

7.1.2 一维数组元素的引用

数组元素是组成数组的基本单元。数组元素也是一种变量，其标识方法为数组名后跟一个下标。下标表示了元素在数组中的顺序号。

数组元素的一般形式为：

数组名[下标]

其中下标只能为整型常量或整型表达式。如为小数时，C 编译将自动取整。

例如：

```
a[5]
a[i+j]
a[i++]
```

都是合法的数组元素。

数组元素通常也称为下标变量。必须先定义数组， 才能使用下标变量。在 C 语言中只能逐个地使用下标变量，而不能一次引用整个数组。

例如，输出有 10 个元素的数组必须使用循环语句逐个输出各下标变量：

```
for(i=0; i<10; i++)
    printf("%d", a[i]);
```

而不能用一个语句输出整个数组。

下面的写法是错误的：

```
printf("%d", a);
```

【例 7.1】

```
main()
{
    int i, a[10];
    for(i=0; i<=9; i++)
        a[i]=i;
    for(i=9; i>=0; i--)
        printf("%d ", a[i]);
}
```



【例 7.2】

```
main()
{
    int i, a[10];
    for(i=0; i<10; i++)
        a[i]=i;
    for(i=9; i>=0; i--)
        printf("%d", a[i]);
}
```



【例 7.3】

```
main()
{
    int i, a[10];
    for(i=0; i<10; i++)
        a[i]=2*i+1;
    for(i=0; i<=9; i++)
        printf("%d ", a[i]);
}
```

```
printf("\n%d %d\n", a[5.2], a[5.8]);
}
```



本例中用一个循环语句给 a 数组各元素送入奇数值，然后用第二个循环语句输出各个奇数。在第一个 for 语句中，表达式 3 省略了。在下标变量中使用了表达式 i++，用以修改循环变量。当然第二个 for 语句也可以这样作，C 语言允许用表达式表示下标。程序中最后一个 printf 语句输出了两次 a[5] 的值，可以看出当下标不为整数时将自动取整。

7.1.3 一维数组的初始化

给数组赋值的方法除了用赋值语句对数组元素逐个赋值外，还可采用初始化赋值和动态赋值的方法。

数组初始化赋值是指在数组定义时给数组元素赋予初值。数组初始化是在编译阶段进行的。这样将减少运行时间，提高效率。

初始化赋值的一般形式为：

类型说明符 数组名[常量表达式]={值, 值.....值};

其中在 { } 中的各数据值即为各元素的初值，各值之间用逗号间隔。

例如：

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

相当于 a[0]=0; a[1]=1... a[9]=9;

C 语言对数组的初始化赋值还有以下几点规定：

- 1) 可以只给部分元素赋初值。

当 { } 中值的个数少于元素个数时，只给前面部分元素赋值。

例如：

```
int a[10]={0, 1, 2, 3, 4};
```

表示只给 a[0]~a[4] 5 个元素赋值，而后 5 个元素自动赋 0 值。

- 2) 只能给元素逐个赋值，不能给数组整体赋值。

例如给十个元素全部赋 1 值，只能写为：

```
int a[10]={1, 1, 1, 1, 1, 1, 1, 1, 1, 1};
```

而不能写为：

```
int a[10]=1;
```

- 3) 如给全部元素赋值，则在数组说明中，可以不给出数组元素的个数。

例如：

```
int a[5]={1, 2, 3, 4, 5};
```

可写为：

```
int a[]={1, 2, 3, 4, 5};
```

7.1.4 一维数组程序举例

可以在程序执行过程中，对数组作动态赋值。这时可用循环语句配合 scanf 函数逐个对数组元素赋值。

【例 7.4】

```

main()
{
    int i,max,a[10];
    printf("input 10 numbers:\n");
    for(i=0;i<10;i++)
        scanf("%d",&a[i]);
    max=a[0];
    for(i=1;i<10;i++)
        if(a[i]>max) max=a[i];
    printf("maxmum=%d\n",max);
}

```



本例程序中第一个 for 语句逐个输入 10 个数到数组 a 中。然后把 a[0] 送入 max 中。在第二个 for 语句中，从 a[1] 到 a[9] 逐个与 max 中的内容比较，若比 max 的值大，则把该下标变量送入 max 中，因此 max 总是在已比较过的下标变量中为最大者。比较结束，输出 max 的值。

【例 7.5】

```

main()
{
    int i, j, p, q, s, a[10];
    printf("\n input 10 numbers:\n");
    for(i=0;i<10;i++)
        scanf("%d",&a[i]);
    for(i=0;i<10;i++) {
        p=i;q=a[i];
        for(j=i+1;j<10;j++)
            if(q<a[j]) { p=j;q=a[j]; }
        if(i!=p)
            {s=a[i];
             a[i]=a[p];
             a[p]=s; }
        printf("%d",a[i]);
    }
}

```



本例程序中用了两个并列的 for 循环语句，在第二个 for 语句中又嵌套了一个循环语句。第一个 for 语句用于输入 10 个元素的初值。第二个 for 语句用于排序。本程序的排序采用逐个比较的方法进行。在 i 次循环时，把第一个元素的下标 i 赋于 p，而把该下标变量值 a[i] 赋于 q。然后进入小循环，从 a[i+1] 起到最后一个元素止逐个与 a[i] 作比较，有比 a[i] 大者则将其下标送 p，元素值送 q。一次循环结束后，p 即为最大元素的下标，q 则为该元素值。若此时 i≠p，说明 p,q 值均已不是进入小循环之前所赋之值，则交换 a[i] 和 a[p] 之值。此时 a[i] 为已排序完毕的元素。输出该值之后转入下一次循环。对 i+1 以后各个元

素排序。

7.2 二维数组的定义和引用

7.2.1 二维数组的定义

前面介绍的数组只有一个下标，称为一维数组，其数组元素也称为单下标变量。在实际问题中有很多量是二维的或多维的，因此 C 语言允许构造多维数组。多维数组元素有多个下标，以标识它在数组中的位置，所以也称为多下标变量。本小节只介绍二维数组，多维数组可由二维数组类推而得到。

二维数组定义的一般形式是：

类型说明符 数组名[常量表达式 1][常量表达式 2]

其中常量表达式 1 表示第一维下标的长度，常量表达式 2 表示第二维下标的长度。

例如：

```
int a[3][4];
```

说明了一个三行四列的数组，数组名为 a，其下标变量的类型为整型。该数组的下标变量共有 3×4 个，即：

```
a[0][0], a[0][1], a[0][2], a[0][3]
```

```
a[1][0], a[1][1], a[1][2], a[1][3]
```

```
a[2][0], a[2][1], a[2][2], a[2][3]
```

二维数组在概念上是二维的，即是说其下标在两个方向上变化，下标变量在数组中的位置也处于一个平面之中，而不是象一维数组只是一个向量。但是，实际的硬件存储器却是连续编址的，也就是说存储器单元是按一维线性排列的。如何在一维存储器中存放二维数组，可有两种方式：一种是按行排列，即放完一行之后顺次放入第二行。另一种是按列排列，即放完一列之后再顺次放入第二列。在 C 语言中，二维数组是按行排列的。

即：

先存放 a[0] 行，再存放 a[1] 行，最后存放 a[2] 行。每行中有四个元素也是依次存放。由于数组 a 说明为 int 类型，该类型占两个字节的内存空间，所以每个元素均占有两个字节。

7.2.2 二维数组元素的引用

二维数组的元素也称为双下标变量，其表示的形式为：

数组名[下标][下标]

其中下标应为整型常量或整型表达式。

例如：

```
a[3][4]
```

表示 a 数组三行四列的元素。

下标变量和数组说明在形式中有些相似，但这两者具有完全不同的含义。数组说明的方括号中给出的是某一维的长度，即可取下标的最大值；而数组元素中的下标是该元素在数组中的位置标识。前者只能是常量，后者可以是常量，变量或表达式。

【例 7.6】一个学习小组有 5 个人，每个人有三门课的考试成绩。求全组分科的平均成绩和各科总平均成绩。

	张	王	李	赵	周
Math	80	61	59	85	76
C	75	65	63	87	77
Foxpro	92	71	70	90	85

可设一个二维数组 `a[5][3]` 存放五个人三门课的成绩。再设一个一维数组 `v[3]` 存放所求得各分科平均成绩，设变量 `average` 为全组各科总平均成绩。编程如下：

```
main()
{
    int i, j, s=0, average, v[3], a[5][3];
    printf("input score\n");
    for(i=0; i<3; i++)
    {
        for(j=0; j<5; j++)
        { scanf("%d", &a[j][i]);
          s=s+a[j][i]; }
        v[i]=s/5;
        s=0;
    }
    average = (v[0]+v[1]+v[2])/3;
    printf("math:%d\nc language:%d\ndbase:%d\n", v[0], v[1], v[2]);
    printf("total:%d\n", average );
}
```



程序中首先用了一个双重循环。在内循环中依次读入某一门课程各个学生的成绩，并把这些成绩累加起来，退出内循环后再把该累加成绩除以 5 送入 `v[i]` 之中，这就是该门课程的平均成绩。外循环共循环三次，分别求出三门课各自的平均成绩并存放在 `v` 数组之中。退出外循环之后，把 `v[0]`, `v[1]`, `v[2]` 相加除以 3 即得到各科总平均成绩。最后按题意输出各个成绩。

7.2.3 二维数组的初始化

二维数组初始化也是在类型说明时给各下标变量赋以初值。二维数组可按行分段赋值，也可按行连续赋值。

例如对数组 `a[5][3]`：

1) 按行分段赋值可写为：

```
int a[5][3]={ {80, 75, 92}, {61, 65, 71}, {59, 63, 70}, {85, 87, 90}, {76, 77, 85} };
```

2) 按行连续赋值可写为：

```
int a[5][3]={ 80, 75, 92, 61, 65, 71, 59, 63, 70, 85, 87, 90, 76, 77, 85};
```

这两种赋初值的结果是完全相同的。

【例 7.7】

```
main()
{
```

```

int i, j, s=0, average, v[3];
int a[5][3]={ {80, 75, 92}, {61, 65, 71}, {59, 63, 70}, {85, 87, 90}, {76, 77, 85} };
for(i=0; i<3; i++)
{
    for(j=0; j<5; j++)
        s=s+a[j][i];
    v[i]=s/5;
    s=0;
}
average=(v[0]+v[1]+v[2])/3;
printf("math:%d\nc language:%d\ndFoxpro:%d\n", v[0], v[1], v[2]);
printf("total:%d\n", average);
}

```



对于二维数组初始化赋值还有以下说明：

- 1) 可以只对部分元素赋初值，未赋初值的元素自动取 0 值。

例如：

```
int a[3][3]={ {1}, {2}, {3} };

```

是对每一行的第一列元素赋值，未赋值的元素取 0 值。赋值后各元素的值为：

```
1 0 0

```

```
2 0 0

```

```
3 0 0

```

```
int a [3][3]={ {0, 1}, {0, 0, 2}, {3} };

```

赋值后的元素值为：

```
0 1 0

```

```
0 0 2

```

```
3 0 0

```

- 2) 如对全部元素赋初值，则第一维的长度可以不给出。

例如：

```
int a[3][3]={1, 2, 3, 4, 5, 6, 7, 8, 9};

```

可以写为：

```
int a[][3]={1, 2, 3, 4, 5, 6, 7, 8, 9};

```

- 3) 数组是一种构造类型的数据。二维数组可以看作是由一维数组的嵌套而构成的。设一维数组的每个元素都又是一个数组，就组成了二维数组。当然，前提是各元素类型必须相同。根据这样的分析，一个二维数组也可以分解为多个一维数组。C 语言允许这种分解。

如二维数组 a[3][4]，可分解为三个一维数组，其数组名分别为：

```
a[0]

```

```
a[1]

```

```
a[2]

```

对这三个一维数组不需另作说明即可使用。这三个一维数组都有 4 个元素，例如：

一维数组 a[0] 的元素为 a[0][0], a[0][1], a[0][2], a[0][3]。

必须强调的是，a[0], a[1], a[2] 不能当作下标变量使用，它们是数组名，不是一个单纯的下标变量。

7.2.4 二维数组程序举例

7.3 字符数组

用来存放字符量的数组称为字符数组。

7.3.1 字符数组的定义

形式与前面介绍的数值数组相同。

例如：

```
char c[10];
```

由于字符型和整型通用，也可以定义为 `int c[10]` 但这时每个数组元素占 2 个字节的内存单元。

字符数组也可以是二维或多维数组。

例如：

```
char c[5][10];
```

即为二维字符数组。

7.3.2 字符数组的初始化

字符数组也允许在定义时作初始化赋值。

例如：

```
char c[10]={ 'c', ' ', 'p', 'r', 'o', 'g', 'r', 'a', 'm' };
```

赋值后各元素的值为：

数组 C	c[0] 的值为 'c'
	c[1] 的值为 ' '
	c[2] 的值为 'p'
	c[3] 的值为 'r'
	c[4] 的值为 'o'
	c[5] 的值为 'g'
	c[6] 的值为 'r'
	c[7] 的值为 'a'
	c[8] 的值为 'm'

其中 `c[9]` 未赋值，由的值为 'p' 系统自动赋予 0 值。

当对全体元素赋初值时也可以省去长度说明。

例如：

```
char c[]={ 'c', ' ', 'p', 'r', 'o', 'g', 'r', 'a', 'm' };
```

这时 C 数组的长度自动定为 9。

7.3.3 字符数组的引用

【例 7.8】

```
main()
{
    int i, j;
    char a[][5]={{'B','A','S','I','C'},{'d','B','A','S','E'}};
    for(i=0;i<=1;i++)
    {
        for(j=0;j<=4;j++)
            printf("%c",a[i][j]);
        printf("\n");
    }
}
```



本例的二维字符数组由于在初始化时全部元素都赋以初值，因此一维下标的长度可以不加以说明。

7.3.4 字符串和字符串结束标志

在 C 语言中没有专门的字符串变量，通常用一个字符数组来存放一个字符串。前面介绍字符串常量时，已说明字符串总是以 '\0' 作为串的结束符。因此当把一个字符串存入一个数组时，也把结束符 '\0' 存入数组，并以此作为该字符串是否结束的标志。有了 '\0' 标志后，就不必再用字符数组的长度来判断字符串的长度了。

C 语言允许用字符串的方式对数组作初始化赋值。

例如：

```
char c[]={'c',' ','p','r','o','g','r','a','m'};
```

可写为：

```
char c[]="C program";
```

或去掉 {} 写为：

```
char c[]="C program";
```

用字符串方式赋值比用字符逐个赋值要多占一个字节，用于存放字符串结束标志 '\0'。上面的数组 c 在内存中的实际存放情况为：

C		p	r	o	g	r	a	m	\0
---	--	---	---	---	---	---	---	---	----

'\0' 是由 C 编译系统自动加上的。由于采用了 '\0' 标志，所以在用字符串赋初值时一般无须指定数组的长度，而由系统自行处理。

7.3.5 字符数组的输入输出

在采用字符串方式后，字符数组的输入输出将变得简单方便。

除了上述用字符串赋初值的办法外，还可用 printf 函数和 scanf 函数一次性输出输入

一个字符数组中的字符串，而不必使用循环语句逐个地输入输出每个字符。

【例 7.9】

```
main()
{
    char c[]="BASIC\ndBASE";
    printf("%s\n",c);
}
```



注意在本例的 printf 函数中，使用的格式字符串为“%s”，表示输出的是一个字符串。而在输出表列中给出数组名则可。不能写为：

```
printf("%s",c[]);
```

【例 7.10】

```
main()
{
    char st[15];
    printf("input string:\n");
    scanf("%s",st);
    printf("%s\n",st);
}
```



本例中由于定义数组长度为 15，因此输入的字符串长度必须小于 15，以留出一个字节用于存放字符串结束标志‘\0’。应该说明的是，对一个字符数组，如果不作初始化赋值，则必须说明数组长度。还应该特别注意的是，当用 scanf 函数输入字符串时，字符串中不能含有空格，否则将以空格作为串的结束符。

例如当输入的字符串中含有空格时，运行情况为：

```
input string:
this is a book
输出为:
this
```

从输出结果可以看出空格以后的字符都未能输出。为了避免这种情况，可多设几个字符数组分段存放含空格的串。

程序可改写如下：

【例 7.11】

```
main()
{
    char st1[6],st2[6],st3[6],st4[6];
    printf("input string:\n");
    scanf("%s%s%s%s",st1,st2,st3,st4);
    printf("%s %s %s %s\n",st1,st2,st3,st4);
}
```



本程序分别设了四个数组，输入的一行字符的空格分段分别装入四个数组。然后分别输出这四个数组中的字符串。

在前面介绍过，scanf 的各输入项必须以地址方式出现，如 &a,&b 等。但在前例中却是以数组名方式出现的，这是为什么呢？

这是由于在 C 语言中规定，数组名就代表了该数组的首地址。整个数组是以首地址开头的一块连续的内存单元。

如有字符数组 char c[10]，在内存可表示如图。

C[0]	C[1]	C[2]	C[3]	C[4]	C[5]	C[6]	C[7]	C[8]	C[9]
------	------	------	------	------	------	------	------	------	------

设数组 c 的首地址为 2000，也就是说 c[0] 单元地址为 2000。则数组名 c 就代表这个首地址。因此在 c 前面不能再加地址运算符&。如写作 scanf("%s",&c);则是错误的。在执行函数 printf("%s",c) 时，按数组名 c 找到首地址，然后逐个输出数组中各个字符直到遇到字符串终止标志'\0' 为止。

7.3.6 字符串处理函数

C 语言提供了丰富的字符串处理函数，大致可分为字符串的输入、输出、合并、修改、比较、转换、复制、搜索几类。使用这些函数可大大减轻编程的负担。用于输入输出的字符串函数，在使用前应包含头文件"stdio.h"，使用其它字符串函数则应包含头文件"string.h"。

下面介绍几个最常用的字符串函数。

1. 字符串输出函数 puts

格式： puts (字符数组名)

功能：把字符数组中的字符串输出到显示器。即在屏幕上显示该字符串。

【例 7.12】

```
#include "stdio.h"
main()
{
    char c[]="BASIC\ndBASE";
    puts(c);
}
```



从程序中可以看出 puts 函数中可以使用转义字符，因此输出结果成为两行。puts 函数完全可以由 printf 函数取代。当需要按一定格式输出时，通常使用 printf 函数。

2. 字符串输入函数 gets

格式： gets (字符数组名)

功能：从标准输入设备键盘上输入一个字符串。

本函数得到一个函数值，即为该字符数组的首地址。

【例 7.13】

```
#include "stdio.h"
main()
{
    char st[15];
```

```
printf("input string:\n");
gets(st);
puts(st);
}
```



可以看出当输入的字符串中含有空格时，输出仍为全部字符串。说明 gets 函数并不以空格作为字符串输入结束的标志，而只以回车作为输入结束。这是与 scanf 函数不同的。

3. 字符串连接函数 strcat

格式： strcat (字符数组名 1, 字符数组名 2)

功能：把字符数组 2 中的字符串连接到字符数组 1 中字符串的后面，并删去字符串 1 后的串标志“\0”。本函数返回值是字符数组 1 的首地址。

【例 7.14】

```
#include "string.h"
main()
{
    static char st1[30]="My name is ";
    int st2[10];
    printf("input your name:\n");
    gets(st2);
    strcat(st1,st2);
    puts(st1);
}
```



本程序把初始化赋值的字符数组与动态赋值的字符串连接起来。要注意的是，字符数组 1 应定义足够的长度，否则不能全部装入被连接的字符串。

4. 字符串拷贝函数 strcpy

格式： strcpy (字符数组名 1, 字符数组名 2)

功能：把字符数组 2 中的字符串拷贝到字符数组 1 中。串结束标志“\0”也一同拷贝。

字符数组名 2，也可以是一个字符串常量。这时相当于把一个字符串赋予一个字符数组。

【例 7.15】

```
#include "string.h"
main()
{
    char st1[15], st2[]="C Language";
    strcpy(st1, st2);
    puts(st1); printf("\n");
}
```



本函数要求字符数组 1 应有足够的长度，否则不能全部装入所拷贝的字符串。

5. 字符串比较函数 strcmp

格式: `strcmp`(字符数组名 1, 字符数组名 2)

功能: 按照 ASCII 码顺序比较两个数组中的字符串, 并由函数返回值返回比较结果。

字符串 1 = 字符串 2, 返回值 = 0;

字符串 1 > 字符串 2, 返回值 > 0;

字符串 1 < 字符串 2, 返回值 < 0。

本函数也可用于比较两个字符串常量, 或比较数组和字符串常量。

【例 7.16】

```
#include "string.h"
main()
{ int k;
  static char st1[15], st2[] = "C Language";
  printf("input a string:\n");
  gets(st1);
  k = strcmp(st1, st2);
  if(k == 0) printf("st1=st2\n");
  if(k > 0) printf("st1>st2\n");
  if(k < 0) printf("st1<st2\n");
}
```



本程序中把输入的字符串和数组 `st2` 中的串比较, 比较结果返回到 `k` 中, 根据 `k` 值再输出结果提示串。当输入为 `dbase` 时, 由 ASCII 码可知 `"dbase"` 大于 `"C Language"` 故 `k > 0`, 输出结果 `"st1>st2"`。

6. 测字符串长度函数 `strlen`

格式: `strlen`(字符数组名)

功能: 测字符串的实际长度(不含字符串结束标志 `'\0'`) 并作为函数返回值。

【例 7.17】

```
#include "string.h"
main()
{ int k;
  static char st[] = "C language";
  k = strlen(st);
  printf("The length of the string is %d\n", k);
}
```



7.4 程序举例

【例 7.18】把一个整数按大小顺序插入已排好序的数组中。

为了把一个数按大小插入已排好序的数组中, 应首先确定排序是从大到小还是从小到大进行的。设排序是从大到小进行的, 则把欲插入的数与数组中各数逐个比较, 当找到第一个比插入数小的元素 `i` 时, 该元素之前即为插入位置。然后从数组最后一个元素开始到该元素为止, 逐个后移一个单元。最后把插入数赋予元素 `i` 即可。如果被插入数比所有的元素值

都小则插入最后位置。

```
main()
{
    int i, j, p, q, s, n, a[11]={127, 3, 6, 28, 54, 68, 87, 105, 162, 18};
    for(i=0; i<10; i++)
    {
        p=i; q=a[i];
        for(j=i+1; j<10; j++)
            if(q<a[j]) {p=j; q=a[j];}
        if(p!=i)
        {
            s=a[i];
            a[i]=a[p];
            a[p]=s;
        }
        printf("%d ", a[i]);
    }
    printf("\ninput number:\n");
    scanf("%d", &n);
    for(i=0; i<10; i++)
        if(n>a[i])
        {
            for(s=9; s>=i; s--) a[s+1]=a[s];
            break;
        }
    a[i]=n;
    for(i=0; i<=10; i++)
        printf("%d ", a[i]);
    printf("\n");
}
```



本程序首先对数组 a 中的 10 个数从大到小排序并输出排序结果。然后输入要插入的整数 n。再用一个 for 语句把 n 和数组元素逐个比较，如果发现有 $n > a[i]$ 时，则由一个内循环把 i 以下各元素值顺次后移一个单元。后移应从后向前进行（从 a[9] 开始到 a[i] 为止）。后移结束跳出外循环。插入点为 i，把 n 赋予 a[i] 即可。如所有的元素均大于被插入数，则并未进行过后移工作。此时 $i=10$ ，结果是把 n 赋予 a[10]。最后一个循环输出插入数后的数组各元素值。

程序运行时，输入数 47。从结果中可以看出 47 已插入到 54 和 28 之间。

【例 7.19】在二维数组 a 中选出各行最大的元素组成一个一维数组 b。

```
a=( 3  16 87  65
    4  32 11 108
   10 25 12  37)
b=(87 108 37)
```

本题的编程思路是，在数组 A 的每一行中寻找最大的元素，找到之后把该值赋予数组 B 相应的元素即可。程序如下：

```
main()
```

```

{
    int a[][4]={3,16,87,65,4,32,11,108,10,25,12,27};
    int b[3],i,j,l;
    for(i=0;i<=2;i++)
        { l=a[i][0];
          for(j=1;j<=3;j++)
              if(a[i][j]>l) l=a[i][j];
          b[i]=l;}
    printf("\narray a:\n");
    for(i=0;i<=2;i++)
        { for(j=0;j<=3;j++)
          printf("%5d",a[i][j]);
          printf("\n");
          printf("\narray b:\n");
          for(i=0;i<=2;i++)
              printf("%5d",b[i]);
          printf("\n");
        }
}

```



程序中第一个 for 语句中又嵌套了一个 for 语句组成了双重循环。外循环控制逐行处理,并把每行的第 0 列元素赋予 l。进入内循环后,把 l 与后面各列元素比较,并把比 l 大者赋予 l。内循环结束时 l 即为该行最大的元素,然后把 l 值赋予 b[i]。等外循环全部完成时,数组 b 中已装入了 a 各行中的最大值。后面的两个 for 语句分别输出数组 a 和数组 b。

【例 7.20】输入五个国家的名称按字母顺序排列输出。

本题编程思路如下:五个国家名应由一个二维字符数组来处理。然而 C 语言规定可以把一个二维数组当成多个一维数组处理。因此本题又可以按五个一维数组处理,而每一个一维数组就是一个国家名字符串。用字符串比较函数比较各一维数组的大小,并排序,输出结果即可。

编程如下:

```

main()
{
    char st[20],cs[5][20];
    int i,j,p;
    printf("input country's name:\n");
    for(i=0;i<5;i++)
        gets(cs[i]);
    printf("\n");
    for(i=0;i<5;i++)
        { p=i;strcpy(st,cs[i]);
          for(j=i+1;j<5;j++)
              if(strcmp(cs[j],st)<0) {p=j;strcpy(st,cs[j]);}
          if(p!=i)
              {

```

```

strcpy(st, cs[i]);
strcpy(cs[i], cs[p]);
strcpy(cs[p], st);
}
puts(cs[i]);}printf("\n");
}

```



本程序的第一个 for 语句中，用 gets 函数输入五个国家名字符串。上面说过 C 语言允许把一个二维数组按多个一维数组处理，本程序说明 cs[5][20] 为二维字符数组，可分为五个一维数组 cs[0]，cs[1]，cs[2]，cs[3]，cs[4]。因此在 gets 函数中使用 cs[i] 是合法的。在第二个 for 语句中又嵌套了一个 for 语句组成双重循环。这个双重循环完成按字母顺序排序的工作。在外层循环中把字符串数组 cs[i] 中的国名字符串拷贝到数组 st 中，并把下标 i 赋予 p。进入内层循环后，把 st 与 cs[i] 以后的各字符串作比较，若有比 st 小者则把该字符串拷贝到 st 中，并把其下标赋予 p。内循环完成后如 p 不等于 i 说明有比 cs[i] 更小的字符串出现，因此交换 cs[i] 和 st 的内容。至此已确定了数组 cs 的第 i 号元素的排序值。然后输出该字符串。在外循环全部完成之后即完成全部排序和输出。

7.5 本章小结

1. 数组是程序设计中最常用的数据结构。数组可分为数值数组(整数组，实数组)，字符数组以及后面将要介绍的指针数组，结构数组等。
2. 数组可以是一维的，二维的或多维的。
3. 数组类型说明由类型说明符、数组名、数组长度(数组元素个数)三部分组成。数组元素又称为下标变量。数组的类型是指下标变量取值的类型。
4. 对数组的赋值可以用数组初始化赋值，输入函数动态赋值和赋值语句赋值三种方法实现。对数值数组不能用赋值语句整体赋值、输入或输出，而必须用循环语句逐个对数组元素进行操作。

8	函 数	1
8.1	概述	1
8.2	函数定义的一般形式	3
8.3	函数的参数和函数的值	4
8.3.1	形式参数和实际参数	4
8.3.2	函数的返回值	5
8.4	函数的调用	6
8.4.1	函数调用的一般形式	6
8.4.2	函数调用的方式	6
8.4.3	被调用函数的声明和函数原型	7
8.5	函数的嵌套调用	8
8.6	函数的递归调用	10
8.7	数组作为函数参数	12
8.8	局部变量和全局变量	17
8.8.1	局部变量	17
8.8.2	全局变量	19
8.9	变量的存储类别	20
8.9.1	动态存储方式与静态动态存储方式	20
8.9.2	auto 变量	21
8.9.3	用 static 声明局部变量	21
8.9.4	register 变量	22
8.9.5	用 extern 声明外部变量	23

8 函 数

8.1 概述

在前面已经介绍过，C 源程序是由函数组成的。虽然在前面各章的程序中大都只有一个主函数 `main()`，但实用程序往往由多个函数组成。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 语言中的函数相当于其它高级语言的子程序。C 语言不仅提供了极为丰富的库函数(如 Turbo C，MS C 都提供了三百多个库函数)，还允许用户建立自己定义的函数。用户可把自己的算法编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。使程序的层次结构清晰，便于程序的编写、阅读、调试。

在 C 语言中可从不同的角度对函数分类。

1. 从函数定义的角度看，函数可分为库函数和用户定义函数两种。

1) 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。在前面各章的例题中反复用到 `printf`、`scanf`、`getchar`、`putchar`、`gets`、`puts`、`strcat` 等函数均属此类。

- 2) 用户定义函数：由用户按需要写的函数。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。
2. C 语言的函数兼有其它语言中的函数和过程两种功能，从这个角度看，又可把函数分为有返回值函数和无返回值函数两种。
- 1) 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返回值的类型。
- 2) 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。这类函数类似于其它语言的过程。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”。
3. 从主调函数和被调函数之间数据传送的角度看又可分为无参函数和有参函数两种。
- 1) 无参函数：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。
- 2) 有参函数：也称为带参函数。在函数定义及函数说明时都有参数，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。
4. C 语言提供了极为丰富的库函数，这些库函数又可从功能角度作以下分类。
- 1) 字符类型分类函数：用于对字符按 ASCII 码分类：字母，数字，控制字符，分隔符，大小写字母等。
- 2) 转换函数：用于字符或字符串的转换；在字符量和各类数字量(整型，实型等)之间进行转换；在大、小写之间进行转换。
- 3) 目录路径函数：用于文件目录和路径操作。
- 4) 诊断函数：用于内部错误检测。
- 5) 图形函数：用于屏幕管理和各种图形功能。
- 6) 输入输出函数：用于完成输入输出功能。
- 7) 接口函数：用于与 DOS, BIOS 和硬件的接口。
- 8) 字符串函数：用于字符串操作和处理。
- 9) 内存管理函数：用于内存管理。
- 10) 数学函数：用于数学函数计算。
- 11) 日期和时间函数：用于日期，时间转换操作。
- 12) 进程控制函数：用于进程管理和控制。
- 13) 其它函数：用于其它各种功能。

以上各类函数不仅数量多，而且有的还需要硬件知识才会使用，因此要想全部掌握则需要一个较长的学习过程。应首先掌握一些最基本、最常用的函数，再逐步深入。由于课时关系，我们只介绍了很少一部分库函数，其余部分读者可根据需要查阅有关手册。

还应该指出的是，在 C 语言中，所有的函数定义，包括主函数 main 在内，都是平行的。也就是说，在一个函数的函数体内，不能再定义另一个函数，即不能嵌套定义。但是函数之间允许相互调用，也允许嵌套调用。习惯上把调用者称为主调函数。函数还可以自己调用自己，称为递归调用。

main 函数是主函数，它可以调用其它函数，而不允许被其它函数调用。因此，C 程序的执行总是从 main 函数开始，完成对其它函数的调用后再返回到 main 函数，最后由 main 函数结束整个程序。一个 C 源程序必须有，也只能有一个主函数 main。

8.2 函数定义的一般形式

1. 无参函数的定义形式

```
类型标识符 函数名()  
{声明部分  
  语句  
}
```

其中类型标识符和函数名称为函数头。类型标识符指明了本函数的类型，函数的类型实际上是函数返回值的类型。该类型标识符与前面介绍的各种说明符相同。函数名是由用户定义的标识符，函数名后有一个空括号，其中无参数，但括号不可少。

{ } 中的内容称为函数体。在函数体中 **声明部分**，是对函数体内部所用到的变量的类型说明。

在很多情况下都不要求无参函数有返回值，此时函数类型符可以写为 void。

我们可以改写一个函数定义：

```
void Hello()  
{  
    printf ("Hello, world \n");  
}
```

这里，只把 main 改为 Hello 作为函数名，其余不变。Hello 函数是一个无参函数，当被其它函数调用时，输出 Hello world 字符串。

2. 有参函数定义的一般形式

```
类型标识符 函数名(形式参数表列)  
{声明部分  
  语句  
}
```

有参函数比无参函数多了一个内容，即形式参数表列。在形参表中给出的参数称为形式参数，它们可以是各种类型的变量，各参数之间用逗号间隔。在进行函数调用时，主调函数将赋予这些形式参数实际的值。形参既然是变量，必须在形参表中给出形参的类型说明。

例如，定义一个函数，用于求两个数中的大数，可写为：

```
int max(int a, int b)  
{  
    if (a>b) return a;  
    else return b;  
}
```

第一行说明 max 函数是一个整型函数，其返回的函数值是一个整数。形参为 a, b, 均为整型量。a, b 的具体值是由主调函数在调用时传送过来的。在 { } 中的函数体内，除形参外没有使用其它变量，因此只有语句而没有声明部分。在 max 函数体中的 return 语句是把 a (或 b) 的值作为函数的值返回给主调函数。有返回值函数中至少应有一个 return 语句。

在 C 程序中，一个函数的定义可以放在任意位置，既可放在主函数 main 之前，也可放在 main 之后。

例如：

可把 max 函数置在 main 之后，也可以把它放在 main 之前。修改后的程序如下所示。

【例 8.1】

```
int max(int a, int b)
{
    if(a>b) return a;
    else return b;
}
main()
{
    int max(int a, int b);
    int x, y, z;
    printf("input two numbers:\n");
    scanf("%d%d", &x, &y);
    z=max(x, y);
    printf("maxmum=%d", z);
}
```



现在我们可以从函数定义、函数说明及函数调用的角度来分析整个程序，从中进一步了解函数的各种特点。

程序的第 1 行至第 5 行为 max 函数定义。进入主函数后，因为准备调用 max 函数，故先对 max 函数进行说明(程序第 8 行)。函数定义和函数说明并不是一回事，在后面还要专门讨论。可以看出函数说明与函数定义中的函数头部分相同，但是末尾要加分号。程序第 12 行为调用 max 函数，并把 x, y 中的值传送给 max 的形参 a, b。max 函数执行的结果(a 或 b)将返回给变量 z。最后由主函数输出 z 的值。

8.3 函数的参数和函数的值

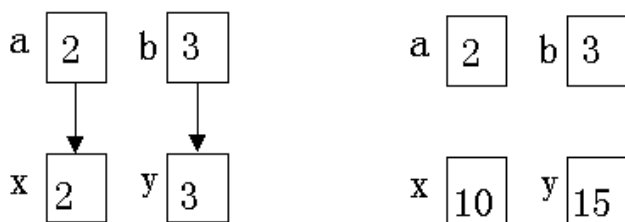
8.3.1 形式参数和实际参数

前面已经介绍过，函数的参数分为形参和实参两种。在本小节中，进一步介绍形参、实参的特点和两者的关系。形参出现在函数定义中，在整个函数体内都可以使用，离开该函数则不能使用。实参出现在主调函数中，进入被调函数后，实参变量也不能使用。形参和实参的功能是作数据传送。发生函数调用时，主调函数把实参的值传送给被调函数的形参从而实现主调函数向被调函数的数据传送。

函数的形参和实参具有以下特点：

1. 形参变量只有在被调用时才分配内存单元，在调用结束时，即刻释放所分配的内存单元。因此，形参只有在函数内部有效。函数调用结束返回主调函数后则不能再使用该形参变量。
2. 实参可以是常量、变量、表达式、函数等，无论实参是何种类型的量，在进行函数调用时，它们都必须具有确定的值，以便把这些值传送给形参。因此应预先用赋值，输入等办法使实参获得确定值。
3. 实参和形参在数量上，类型上，顺序上应严格一致，否则会发生类型不匹配”的错误。
4. 函数调用中发生的数据传送是单向的。即只能把实参的值传送给形参，而不能把形参的值反向地传送给实参。因此在函数调用过程中，形参的值发生改变，而实参中的值不

会变化。



【例 8.2】可以说明这个问题。

```
main()
{
    int n;
    printf("input number\n");
    scanf("%d", &n);
    s(n);
    printf("n=%d\n", n);
}

int s(int n)
{
    int i;
    for(i=n-1; i>=1; i--)
        n=n+i;
    printf("n=%d\n", n);
}
```



本程序中定义了一个函数 s ，该函数的功能是求 $\sum n_i$ 的值。在主函数中输入 n 值，并作为实参，在调用时传送给 s 函数的形参量 n （注意，本例的形参变量和实参变量的标识符都为 n ，但这是两个不同的量，各自的作用域不同）。在主函数中用 `printf` 语句输出一次 n 值，这个 n 值是实参 n 的值。在函数 s 中也用 `printf` 语句输出了一次 n 值，这个 n 值是形参最后取得的 n 值 0。从运行情况看，输入 n 值为 100。即实参 n 的值为 100。把此值传给函数 s 时，形参 n 的初值也为 100，在执行函数过程中，形参 n 的值变为 5050。返回主函数之后，输出实参 n 的值仍为 100。可见实参的值不随形参的变化而变化。

8.3.2 函数的返回值

函数的值是指函数被调用之后，执行函数体中的程序段所取得的并返回给主调函数的值。如调用正弦函数取得正弦值，调用例 8.1 的 `max` 函数取得的最大数等。对函数的值（或称函数返回值）有以下一些说明：

- 1) 函数的值只能通过 `return` 语句返回主调函数。

`return` 语句的一般形式为：

return 表达式；

或者为：

return (表达式);

该语句的功能是计算表达式的值，并返回给主调函数。在函数中允许有多个 return 语句，但每次调用只能有一个 return 语句被执行，因此只能返回一个函数值。

- 2) 函数值的类型和函数定义中函数的类型应保持一致。如果两者不一致，则以函数类型为准，自动进行类型转换。
- 3) 如函数值为整型，在函数定义时可以省去类型说明。
- 4) 不返回函数值的函数，可以明确定义为“空类型”，类型说明符为“void”。如例 8.2 中函数 s 并不向主函数返回函数值，因此可定义为：

```
void s(int n)
{ .....
}
```

一旦函数被定义为空类型后，就不能在主调函数中使用被调函数的函数值了。

例如，在定义 s 为空类型后，在主函数中写下述语句

```
sum=s(n);
```

就是错误的。

为了使程序有良好的可读性并减少出错，凡不要求返回值的函数都应定义为空类型。

8.4 函数的调用

8.4.1 函数调用的一般形式

前面已经说过，在程序中是通过对函数的调用来执行函数体的，其过程与其它语言的子程序调用相似。

C 语言中，函数调用的一般形式为：

函数名(实际参数表)

对无参函数调用时则无实际参数表。实际参数表中的参数可以是常数，变量或其它构造类型数据及表达式。各实参之间用逗号分隔。

8.4.2 函数调用的方式

在 C 语言中，可以用以下几种方式调用函数：

1. 函数表达式：函数作为表达式中的一项出现在表达式中，以函数返回值参与表达式的运算。这种方式要求函数是有返回值的。例如：z=max(x, y) 是一个赋值表达式，把 max 的返回值赋予变量 z。
2. 函数语句：函数调用的一般形式加上分号即构成函数语句。例如：printf("%d", a); scanf("%d", &b); 都是以函数语句的方式调用函数。
3. 函数实参：函数作为另一个函数调用的实际参数出现。这种情况是把该函数的返回值作为实参进行传送，因此要求该函数必须是有返回值的。例如：printf("%d", max(x, y)); 即是把 max 调用的返回值又作为 printf 函数的实参来使用的。在函数调用中还应该注意的一个问题是求值顺序的问题。所谓求值顺序是指对实

参表中各量是自左至右使用呢，还是自右至左使用。对此，各系统的规定不一定相同。

介绍 printf 函数时已提到过，这里从函数调用的角度再强调一下。

【例 8.3】

```
main()
{
    int i=8;
    printf("%d\n%d\n%d\n%d\n", ++i, --i, i++, i--);
}
```



如按照从右至左的顺序求值。运行结果应为：

8
7
7
8

如对 printf 语句中的 ++i, --i, i++, i-- 从左至右求值，结果应为：

9
8
8
9

应特别注意的是，无论是从左至右求值，还是自右至左求值，其输出顺序都是不变的，即输出顺序总是和实参表中实参的顺序相同。由于 Turbo C 现定是自右至左求值，所以结果为 8, 7, 7, 8。上述问题如还不理解，上机一试就明白了。

8.4.3 被调用函数的声明和函数原型

在主调函数中调用某函数之前应对该被调函数进行说明（声明），这与使用变量之前要先进行变量说明是一样的。在主调函数中对被调函数作说明的目的是使编译系统知道被调函数返回值的类型，以便在主调函数中按此种类型对返回值作相应的处理。

其一般形式为：

类型说明符 被调函数名(类型 形参, 类型 形参...);

或为：

类型说明符 被调函数名(类型, 类型...);

括号内给出了形参的类型和形参名，或只给出形参类型。这便于编译系统进行检错，以防止可能出现的错误。

例 8.1 main 函数中对 max 函数的说明为：

```
int max(int a, int b);
```

或写为：

```
int max(int, int);
```

C 语言中又规定在以下几种情况时可以省去主调函数中对被调函数的函数说明。

- 1) 如果被调函数的返回值是整型或字符型时，可以不对被调函数作说明，而直接调用。这时系统将自动对被调函数返回值按整型处理。例 8.2 的主函数中未对函数 s 作说明而直接调用即属此种情形。
- 2) 当被调函数的函数定义出现在主调函数之前时，在主调函数中也可以不对被调函数

再作说明而直接调用。例如例 8.1 中，函数 max 的定义放在 main 函数之前，因此可在 main 函数中省去对 max 函数的函数说明 `int max(int a, int b)`。

- 3) 如在所有函数定义之前，在函数外预先说明了各个函数的类型，则在以后的各主调函数中，可不再对被调函数作说明。例如：

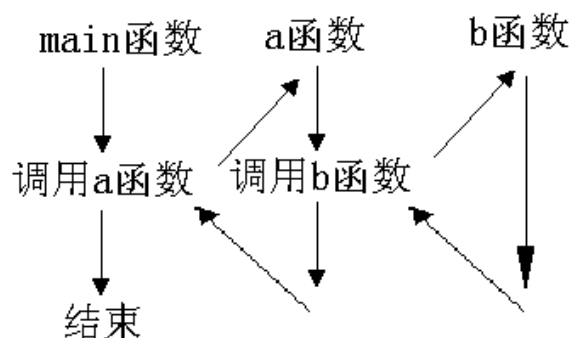
```
char str(int a);
float f(float b);
main()
{
    .....
}
char str(int a)
{
    .....
}
float f(float b)
{
    .....
}
```

其中第一，二行对 str 函数和 f 函数预先作了说明。因此在以后各函数中无须对 str 和 f 函数再作说明就可直接调用。

- 4) 对库函数的调用不需要再作说明，但必须把该函数的头文件用 include 命令包含在源文件前部。

8.5 函数的嵌套调用

C 语言中不允许作嵌套的函数定义。因此各函数之间是平行的，不存在上一级函数和下级函数的问题。但是 C 语言允许在一个函数的定义中出现对另一个函数的调用。这样就出现了函数的嵌套调用。即在被调函数中又调用其它函数。这与其它语言的子程序嵌套的情形是类似的。其关系可表示如图。



图表示了两层嵌套的情形。其执行过程是：执行 main 函数中调用 a 函数的语句时，即转去执行 a 函数，在 a 函数中调用 b 函数时，又转去执行 b 函数，b 函数执行完毕返回 a 函数的断点继续执行，a 函数执行完毕返回 main 函数的断点继续执行。

【例 8.4】 计算 $s=2^2!+3^2!$

本题可编写两个函数，一个是用来计算平方值的函数 f1，另一个是用来计算阶乘值的函数 f2。主函数先调 f1 计算出平方值，再在 f1 中以平方值为实参，调用 f2 计算其阶乘值，然后返回 f1，再返回主函数，在循环程序中计算累加和。

```
long f1(int p)
{
    int k;
    long r;
    long f2(int);
    k=p*p;
    r=f2(k);
    return r;
}
long f2(int q)
{
    long c=1;
    int i;
    for(i=1;i<=q;i++)
        c=c*i;
    return c;
}
main()
{
    int i;
    long s=0;
    for (i=2;i<=3;i++)
        s=s+f1(i);
    printf("\ns=%ld\n", s);
}
```



在程序中，函数 f1 和 f2 均为长整型，都在主函数之前定义，故不必再在主函数中对 f1 和 f2 加以说明。在主程序中，执行循环程序依次把 i 值作为实参调用函数 f1 求 i^2 值。

在 f1 中又发生对函数 f2 的调用，这时是把 i^2 的值作为实参去调 f2，在 f2 中完成求 $i^2!$

的计算。f2 执行完毕把 C 值(即 $i^2!$)返回给 f1，再由 f1 返回主函数实现累加。至此，由函数的嵌套调用实现了题目的要求。由于数值很大，所以函数和一些变量的类型都说明为长整型，否则会造成计算错误。

8.6 函数的递归调用

一个函数在它的函数体内调用它自身称为递归调用。这种函数称为递归函数。C 语言允许函数的递归调用。在递归调用中，主调函数又是被调函数。执行递归函数将反复调用其自身，每调用一次就进入新的一层。

例如有函数 f 如下：

```
int f(int x)
{
    int y;
    z=f(y);
    return z;
}
```

这个函数是一个递归函数。但是运行该函数将无休止地调用其自身，这当然是不正确的。为了防止递归调用无终止地进行，必须在函数内有终止递归调用的手段。常用的办法是加条件判断，满足某种条件后就不再作递归调用，然后逐层返回。下面举例说明递归调用的执行过程。

【例 8.5】用递归法计算 $n!$

用递归法计算 $n!$ 可用下述公式表示：

$$\begin{aligned} n! &= 1 & (n=0, 1) \\ n \times (n-1)! & & (n>1) \end{aligned}$$

按公式可编程如下：

```
long ff(int n)
{
    long f;
    if(n<0) printf("n<0,input error");
    else if(n==0||n==1) f=1;
    else f=ff(n-1)*n;
    return(f);
}

main()
{
    int n;
    long y;
    printf("\ninput a inteager number:\n");
    scanf("%d",&n);
    y=ff(n);
    printf("%d!=%ld",n,y);
}
```



程序中给出的函数 ff 是一个递归函数。主函数调用 ff 后即进入函数 ff 执行，如果 $n<0, n==0$ 或 $n=1$ 时都将结束函数的执行，否则就递归调用 ff 函数自身。由于每次递归调用的实参为 $n-1$ ，即把 $n-1$ 的值赋予形参 n ，最后当 $n-1$ 的值为 1 时再作递归调用，形参 n 的值

也为 1，将使递归终止。然后可逐层退回。

下面我们再举例说明该过程。设执行本程序时输入为 5，即求 $5!$ 。在主函数中的调用语句即为 $y=ff(5)$ ，进入 ff 函数后，由于 $n=5$ ，不等于 0 或 1，故应执行 $f=ff(n-1)*n$ ，即 $f=ff(5-1)*5$ 。该语句对 ff 作递归调用即 $ff(4)$ 。

进行四次递归调用后， ff 函数形参取得的值变为 1，故不再继续递归调用而开始逐层返回主调函数。 $ff(1)$ 的函数返回值为 1， $ff(2)$ 的返回值为 $1*2=2$ ， $ff(3)$ 的返回值为 $2*3=6$ ， $ff(4)$ 的返回值为 $6*4=24$ ，最后返回值 $ff(5)$ 为 $24*5=120$ 。

例 8.5 也可以不用递归的方法来完成。如可以用递推法，即从 1 开始乘以 2，再乘以 3... 直到 n 。递推法比递归法更容易理解和实现。但是有些问题则只能用递归算法才能实现。典型的问题是 Hanoi 塔问题。

【例 8.6】Hanoi 塔问题

一块板上有三根针，A，B，C。A 针上套有 64 个大小不等的圆盘，大的在下，小的在上。如图 5.4 所示。要把这 64 个圆盘从 A 针移动到 C 针上，每次只能移动一个圆盘，移动可以借助 B 针进行。但在任何时候，任何针上的圆盘都必须保持大盘在下，小盘在上。求移动的步骤。

本题算法分析如下，设 A 上有 n 个盘子。

如果 $n=1$ ，则将圆盘从 A 直接移动到 C。

如果 $n=2$ ，则：

1. 将 A 上的 $n-1$ (等于 1) 个圆盘移到 B 上；
2. 再将 A 上的一个圆盘移到 C 上；
3. 最后将 B 上的 $n-1$ (等于 1) 个圆盘移到 C 上。

如果 $n=3$ ，则：

A. 将 A 上的 $n-1$ (等于 2，令其为 n') 个圆盘移到 B (借助于 C)，步骤如下：

- (1) 将 A 上的 $n'-1$ (等于 1) 个圆盘移到 C 上。
- (2) 将 A 上的一个圆盘移到 B。
- (3) 将 C 上的 $n'-1$ (等于 1) 个圆盘移到 B。

B. 将 A 上的一个圆盘移到 C。

C. 将 B 上的 $n-1$ (等于 2，令其为 n') 个圆盘移到 C (借助 A)，步骤如下：

- (1) 将 B 上的 $n'-1$ (等于 1) 个圆盘移到 A。
- (2) 将 B 上的一个盘子移到 C。
- (3) 将 A 上的 $n'-1$ (等于 1) 个圆盘移到 C。

到此，完成了三个圆盘的移动过程。

从上面分析可以看出，当 n 大于等于 2 时，移动的过程可分解为三个步骤：

第一步 把 A 上的 $n-1$ 个圆盘移到 B 上；

第二步 把 A 上的一个圆盘移到 C 上；

第三步 把 B 上的 $n-1$ 个圆盘移到 C 上；其中第一步和第三步是类同的。

当 $n=3$ 时，第一步和第三步又分解为类同的三步，即把 $n'-1$ 个圆盘从一个针移到另一个针上，这里的 $n'=n-1$ 。显然这是一个递归过程，据此算法可编程如下：

```
move(int n, int x, int y, int z)
{
    if (n==1)
        printf("%c-->%c\n", x, z);
    else
```

```

    {
        move(n-1, x, z, y);
        printf("%c-->%c\n", x, z);
        move(n-1, y, x, z);
    }
}
main()
{
    int h;
    printf("\ninput number:\n");
    scanf("%d", &h);
    printf("the step to moving %2d disks:\n", h);
    move(h, 'a', 'b', 'c');
}

```



从程序中可以看出, move 函数是一个递归函数, 它有四个形参 n, x, y, z 。 n 表示圆盘数, x, y, z 分别表示三根针。move 函数的功能是把 x 上的 n 个圆盘移动到 z 上。当 $n=1$ 时, 直接把 x 上的圆盘移至 z 上, 输出 $x \rightarrow z$ 。如 $n \neq 1$ 则分为三步: 递归调用 move 函数, 把 $n-1$ 个圆盘从 x 移到 y ; 输出 $x \rightarrow z$; 递归调用 move 函数, 把 $n-1$ 个圆盘从 y 移到 z 。在递归调用过程中 $n=n-1$, 故 n 的值逐次递减, 最后 $n=1$ 时, 终止递归, 逐层返回。当 $n=4$ 时程序运行的结果为:

```

input number:
4
the step to moving 4 disks:
a→b
a→c
b→c
a→b
c→a
c→b
a→b
a→c
b→c
b→a
c→a
b→c
a→b
a→c
b→c

```

8.7 数组作为函数参数

数组可以作为函数的参数使用, 进行数据传送。数组用作函数参数有两种形式, 一种是

把数组元素(下标变量)作为实参使用;另一种是把数组名作为函数的形参和实参使用。

1. 数组元素作函数实参

数组元素就是下标变量,它与普通变量并无区别。因此它作为函数实参使用与普通变量是完全相同的,在发生函数调用时,把作为实参的数组元素的值传送给形参,实现单向的值传送。例 5.4 说明了这种情况。

【例 8.7】判别一个整数数组中各元素的值,若大于 0 则输出该值,若小于等于 0 则输出 0 值。编程如下:

```
void nzp(int v)
{
    if(v>0)
        printf("%d ",v);
    else
        printf("%d ",0);
}

main()
{
    int a[5],i;
    printf("input 5 numbers\n");
    for(i=0;i<5;i++)
        {scanf("%d",&a[i]);
         nzp(a[i]);}
}
```



本程序中首先定义一个无返回值函数 nzp,并说明其形参 v 为整型变量。在函数体中根据 v 值输出相应的结果。在 main 函数中用一个 for 语句输入数组各元素,每输入一个就以该元素作实参调用一次 nzp 函数,即把 a[i] 的值传送给形参 v,供 nzp 函数使用。

2. 数组名作为函数参数

用数组名作函数参数与用数组元素作实参有几点不同:

- 1) 用数组元素作实参时,只要数组类型和函数的形参变量的类型一致,那么作为下标变量的数组元素的类型也和函数形参变量的类型是一致的。因此,并不要求函数的形参也是下标变量。换句话说,对数组元素的处理是按普通变量对待的。用数组名作函数参数时,则要求形参和相对应的实参都必须是类型相同的数组,都必须有明确的数组说明。当形参和实参二者不一致时,即会发生错误。
- 2) 在普通变量或下标变量作函数参数时,形参变量和实参变量是由编译系统分配的两个不同的内存单元。在函数调用时发生的值传送是把实参变量的值赋予形参变量。在用数组名作函数参数时,不是进行值的传送,即不是把实参数组的每一个元素的值都赋予形参数组的各个元素。因为实际上形参数组并不存在,编译系统不为形参数组分配内存。那么,数据的传送是如何实现的呢?在我们曾介绍过,数组名就是数组的首地址。因此在数组名作函数参数时所进行的传送只是地址的传送,也就是说把实参数组的首地址赋予形参数组名。形参数组名取得该首地址之后,也就等于有了实在的数组。实际上是形参数组和实参数组为同一数组,共同拥有一段内存空间。

	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
起始地址 2000	2	4	6	8	10	12	14	16	18	20
	b[0]	b[1]	b[2]	b[3]	b[4]	b[5]	b[6]	b[7]	b[8]	b[9]

上图说明了这种情形。图中设 a 为实参数组，类型为整型。a 占有以 2000 为首地址的一块内存区。b 为形参数组名。当发生函数调用时，进行地址传送，把实参数组 a 的首地址传送给形参数组名 b，于是 b 也取得该地址 2000。于是 a，b 两数组共同占有以 2000 为首地址的一段连续内存单元。从图中还可以看出 a 和 b 下标相同的元素实际上也占相同的两个内存单元（整型数组每个元素占二字节）。例如 a[0] 和 b[0] 都占用 2000 和 2001 单元，当然 a[0] 等于 b[0]。类推则有 a[i] 等于 b[i]。

【例 8.8】数组 a 中存放了一个学生 5 门课程的成绩，求平均成绩。

```
float aver(float a[5])
{
    int i;
    float av,s=a[0];
    for(i=1;i<5;i++)
        s=s+a[i];
    av=s/5;
    return av;
}

void main()
{
    float sco[5],av;
    int i;
    printf("\ninput 5 scores:\n");
    for(i=0;i<5;i++)
        scanf("%f",&sco[i]);
    av=aver(sco);
    printf("average score is %5.2f",av);
}
```



本程序首先定义了一个实型函数 aver，有一个形参为实型数组 a，长度为 5。在函数 aver 中，把各元素值相加求出平均值，返回给主函数。主函数 main 中首先完成数组 sco 的输入，然后以 sco 作为实参调用 aver 函数，函数返回值送 av，最后输出 av 值。从运行情况可以看出，程序实现了所要求的功能。

3) 前面已经讨论过，在变量作函数参数时，所进行的值传送是单向的。即只能从实参传向形参，不能从形参传回实参。形参的初值和实参相同，而形参的值发生改变后，实参并不变化，两者的终值是不同的。而当用数组名作函数参数时，情况则不同。由于实际上形参和实参为同一数组，因此当形参数组发生变化时，实参数组也随之变化。当然这种情况不能理解为发生了“双向”的值传递。但从实际情况来看，调用函数之后实参数组的值将由于形参数组值的变化而变化。为了说明这种情况，把例 5.4 改为例 5.6 的形式。

【例 8.9】题目同 8.7 例。改用数组名作函数参数。

```
void nzp(int a[5])
{
    int i;
    printf("\nvalues of array a are:\n");
    for(i=0;i<5;i++)
    {
        if(a[i]<0) a[i]=0;
        printf("%d ",a[i]);
    }
}

main()
{
    int b[5],i;
    printf("\ninput 5 numbers:\n");
    for(i=0;i<5;i++)
        scanf("%d",&b[i]);
    printf("initial values of array b are:\n");
    for(i=0;i<5;i++)
        printf("%d ",b[i]);
    nzp(b);
    printf("\nlast values of array b are:\n");
    for(i=0;i<5;i++)
        printf("%d ",b[i]);
}
```



本程序中函数 nzp 的形参为整数组 a，长度为 5。主函数中实参数组 b 也为整型，长度也为 5。在主函数中首先输入数组 b 的值，然后输出数组 b 的初始值。然后以数组名 b 为实参调用 nzp 函数。在 nzp 中，按要求把负值单元清 0，并输出形参数组 a 的值。返回主函数之后，再次输出数组 b 的值。从运行结果可以看出，数组 b 的初值和终值是不同的，数组 b 的终值和数组 a 是相同的。这说明实参形参为同一数组，它们的值同时得以改变。

用数组名作为函数参数时还应注意以下几点：

- a. 形参数组和实参数组的类型必须一致，否则将引起错误。
- b. 形参数组和实参数组的长度可以不相同，因为在调用时，只传送首地址而不检查形参数组的长度。当形参数组的长度与实参数组不一致时，虽不至于出现语法错误（编译能通过），但程序执行结果将与实际不符，这是应予以注意的。

【例 8.10】如把例 8.9 修改如下：

```
void nzp(int a[8])
{
    int i;
    printf("\nvalues of array aare:\n");
    for(i=0;i<8;i++)
    {
```

```

        if(a[i]<0)a[i]=0;
        printf("%d ",a[i]);
    }
}
main()
{
    int b[5],i;
    printf("\ninput 5 numbers:\n");
    for(i=0;i<5;i++)
        scanf("%d",&b[i]);
    printf("initial values of array b are:\n");
    for(i=0;i<5;i++)
        printf("%d ",b[i]);
    nzp(b);
    printf("\nlast values of array b are:\n");
    for(i=0;i<5;i++)
        printf("%d ",b[i]);
}

```



本程序与例 8.9 程序比, nzp 函数的形参数组长度改为 8, 函数体中, for 语句的循环条件也改为 i<8。因此, 形参数组 a 和实参数组 b 的长度不一致。编译能够通过, 但从结果看, 数组 a 的元素 a[5], a[6], a[7]显然是无意义的。

- c. 在函数形参表中, 允许不给出形参数组的长度, 或用一个变量来表示数组元素的个数。

例如, 可以写为:

```
void nzp(int a[])
```

或写为

```
void nzp(int a[], int n)
```

其中形参数组 a 没有给出长度, 而由 n 值动态地表示数组的长度。n 的值由主调函数的实参进行传送。

由此, 例 8.10 又可改为例 8.11 的形式。

【例 8.11】

```

void nzp(int a[], int n)
{
    int i;
    printf("\nvalues of array a are:\n");
    for(i=0;i<n;i++)
    {
        if(a[i]<0) a[i]=0;
        printf("%d ",a[i]);
    }
}
main()

```

```

{
    int b[5], i;
    printf("\ninput 5 numbers:\n");
    for(i=0; i<5; i++)
        scanf("%d", &b[i]);
    printf("initial values of array b are:\n");
    for(i=0; i<5; i++)
        printf("%d ", b[i]);
    nzp(b, 5);
    printf("\nlast values of array b are:\n");
    for(i=0; i<5; i++)
        printf("%d ", b[i]);
}

```



本程序 nzp 函数形参数组 a 没有给出长度，由 n 动态确定该长度。在 main 函数中，函数调用语句为 nzp(b, 5)，其中实参 5 将赋予形参 n 作为形参数组的长度。

- d. 多维数组也可以作为函数的参数。在函数定义时对形参数组可以指定每一维的长度，也可省去第一维的长度。因此，以下写法都是合法的。

```

int MA(int a[3][10])
或
int MA(int a[][10])。

```

8.8 局部变量和全局变量

在讨论函数的形参变量时曾经提到，形参变量只在被调用期间才分配内存单元，调用结束立即释放。这一点表明形参变量只有在函数内才是有效的，离开该函数就不能再使用了。这种变量有效性的范围称变量的作用域。不仅对于形参变量，C 语言中所有的量都有自己的作用域。变量说明的方式不同，其作用域也不同。C 语言中的变量，按作用域范围可分为两种，即局部变量和全局变量。

8.8.1 局部变量

局部变量也称为内部变量。局部变量是在函数内作定义说明的。其作用域仅限于函数内，离开该函数后再使用这种变量是非法的。

例如：

```

int f1(int a)          /*函数 f1*/
{
    int b, c;
    .....
}
a, b, c 有效
int f2(int x)          /*函数 f2*/

```

```

{
int y, z;
.....
}
x, y, z 有效
main()
{
    int m, n;
    .....
}
m, n 有效

```

在函数 f1 内定义了三个变量, a 为形参, b, c 为一般变量。在 f1 的范围内 a, b, c 有效, 或者说 a, b, c 变量的作用域限于 f1 内。同理, x, y, z 的作用域限于 f2 内。m, n 的作用域限于 main 函数内。关于局部变量的作用域还要说明以下几点:

- 1) 主函数中定义的变量也只能在主函数中使用, 不能在其它函数中使用。同时, 主函数中也不能使用其它函数中定义的变量。因为主函数也是一个函数, 它与其它函数是平行关系。这一点是与其它语言不同的, 应予以注意。
- 2) 形参变量是属于被调函数的局部变量, 实参变量是属于主调函数的局部变量。
- 3) 允许在不同的函数中使用相同的变量名, 它们代表不同的对象, 分配不同的单元, 互不干扰, 也不会发生混淆。如在前例中, 形参和实参的变量名都为 n, 是完全允许的。
- 4) 在复合语句中也可定义变量, 其作用域只在复合语句范围内。

例如:

```

main()
{
    int s, a;
    .....
    {
        int b;
        s=a+b;
        .....          /*b 作用域*/
    }
    .....          /*s, a 作用域*/
}

```

【例 8.12】

```

main()
{
    int i=2, j=3, k;
    k=i+j;
    {
        int k=8;
        printf("%d\n", k);
    }
    printf("%d\n", k);
}

```

}



本程序在 main 中定义了 i, j, k 三个变量，其中 k 未赋初值。而在复合语句内又定义了一个变量 k，并赋初值为 8。应该注意这两个 k 不是同一个变量。在复合语句外由 main 定义的 k 起作用，而在复合语句内则由在复合语句内定义的 k 起作用。因此程序第 4 行的 k 为 main 所定义，其值应为 5。第 7 行输出 k 值，该行在复合语句内，由复合语句内定义的 k 起作用，其初值为 8，故输出值为 8，第 9 行输出 i, k 值。i 是在整个程序中有效的，第 7 行对 i 赋值为 3，故以输出也为 3。而第 9 行已在复合语句之外，输出的 k 应为 main 所定义的 k，此 k 值由第 4 行已获得为 5，故输出也为 5。

8.8.2 全局变量

全局变量也称为外部变量，它是在函数外部定义的变量。它不属于哪一个函数，它属于一个源程序文件。其作用域是整个源程序。在函数中使用全局变量，一般应作全局变量说明。只有在函数内经过说明的全局变量才能使用。全局变量的说明符为 extern。但在一个函数之前定义的全局变量，在该函数内使用可不再加以说明。

例如：

```
int a, b;          /*外部变量*/
void f1()          /*函数 f1*/
{
    .....
}
float x, y;        /*外部变量*/
int fz()           /*函数 fz*/
{
    .....
}
main()             /*主函数*/
{
    .....
}
```

从上例可以看出 a、b、x、y 都是在函数外部定义的外部变量，都是全局变量。但 x, y 定义在函数 f1 之后，而在 f1 内又无对 x, y 的说明，所以它们在 f1 内无效。a, b 定义在源程序最前面，因此在 f1, f2 及 main 内不加说明也可使用。

【例 8.13】输入正方体的长宽高 l, w, h。求体积及三个面 x*y, x*z, y*z 的面积。

```
int s1, s2, s3;
int vs( int a, int b, int c)
{
    int v;
    v=a*b*c;
    s1=a*b;
    s2=b*c;
    s3=a*c;
```

```

    return v;
}
main()
{
    int v, l, w, h;
    printf("\ninput length,width and height\n");
    scanf("%d%d%d", &l, &w, &h);
    v=vs(l, w, h);
    printf("\nv=%d, s1=%d, s2=%d, s3=%d\n", v, s1, s2, s3);
}

```



【例 8.14】外部变量与局部变量同名。

```

int a=3, b=5;    /*a, b 为外部变量*/
max(int a, int b) /*a, b 为外部变量*/
{int c;
  c=a>b?a:b;
  return(c);
}
main()
{int a=8;
  printf("%d\n", max(a, b));
}

```



如果同一个源文件中，外部变量与局部变量同名，则在局部变量的作用范围内，外部变量被“屏蔽”，即它不起作用。

8.9 变量的存储类别

8.9.1 动态存储方式与静态动态存储方式

前面已经介绍了，从变量的作用域（即从空间）角度来分，可以分为**全局变量**和**局部变量**。

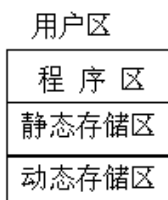
从另一个角度，从变量值存在的作时间（即生存期）角度来分，可以分为**静态存储方式**和**动态存储方式**。

静态存储方式：是指在程序运行期间分配固定的存储空间的方式。

动态存储方式：是在程序运行期间根据需要进行动态的分配存储空间的方式。

用户存储空间可以分为三个部分：

- 1) 程序区；
- 2) 静态存储区；
- 3) 动态存储区；



全局变量全部存放在静态存储区，在程序开始执行时给全局变量分配存储区，程序行完毕就释放。在程序执行过程中它们占据固定的存储单元，而不动态地进行分配和释放；

动态存储区存放以下数据：

- 1) 函数形式参数；
- 2) 自动变量（未加 `static` 声明的局部变量）；
- 3) 函数调用时的现场保护和返回地址；

对以上这些数据，在函数开始调用时分配动态存储空间，函数结束时释放这些空间。

在 C 语言中，每个变量和函数有两个属性：数据类型和数据的存储类别。

8.9.2 auto 变量

函数中的局部变量，如不专门声明为 `static` 存储类别，都是动态地分配存储空间的，数据存储在动态存储区中。函数中的形参和在函数中定义的变量（包括在复合语句中定义的变量），都属此类，在调用该函数时系统会给它们分配存储空间，在函数调用结束时就自动释放这些存储空间。这类局部变量称为自动变量。自动变量用关键字 `auto` 作存储类别的声明。

例如：

```
int f(int a)          /*定义 f 函数，a 为参数*/
{auto int b,c=3;      /*定义 b, c 自动变量*/
    .....
}
```

a 是形参，b, c 是自动变量，对 c 赋初值 3。执行完 f 函数后，自动释放 a, b, c 所占的存储单元。

关键字 `auto` 可以省略，`auto` 不写则隐含定为“自动存储类别”，属于动态存储方式。

8.9.3 用 static 声明局部变量

有时希望函数中的局部变量的值在函数调用结束后不消失而保留原值，这时就应该指定局部变量为“静态局部变量”，用关键字 `static` 进行声明。

【例 8.15】考察静态局部变量的值。

```
f(int a)
{auto b=0;
  static c=3;
  b=b+1;
  c=c+1;
  return(a+b+c);
}
main()
```

```
{int a=2,i;
  for(i=0;i<3;i++)
    printf("%d",f(a));
}
```



对静态局部变量的说明：

- 1) 静态局部变量属于静态存储类别，在静态存储区内分配存储单元。在程序整个运行期间都不释放。而自动变量（即动态局部变量）属于动态存储类别，占动态存储空间，函数调用结束后即释放。
- 2) 静态局部变量在编译时赋初值，即只赋初值一次；而对自动变量赋初值是在函数调用时进行，每调用一次函数重新给一次初值，相当于执行一次赋值语句。
- 3) 如果在定义局部变量时不赋初值的话，则对静态局部变量来说，编译时自动赋初值 0（对数值型变量）或空字符（对字符变量）。而对自动变量来说，如果不赋初值则它的值是一个不确定的值。

【例 8.16】打印 1 到 5 的阶乘值。

```
int fac(int n)
{static int f=1;
  f=f*n;
  return(f);
}
main()
{int i;
  for(i=1;i<=5;i++)
    printf("%d!=%d\n",i,fac(i));
}
```



8.9.4 register 变量

为了提高效率，C 语言允许将局部变量得值放在 CPU 中的寄存器中，这种变量叫“寄存器变量”，用关键字 register 作声明。

【例 8.17】使用寄存器变量。

```
int fac(int n)
{register int i,f=1;
  for(i=1;i<=n;i++)
    f=f*i;
  return(f);
}
main()
{int i;
  for(i=0;i<=5;i++)
    printf("%d!=%d\n",i,fac(i));
}
```

}



说明:

- 1) 只有局部自动变量和形式参数可以作为寄存器变量;
- 2) 一个计算机系统上的寄存器数目有限, 不能定义任意多个寄存器变量;
- 3) 局部静态变量不能定义为寄存器变量。

8.9.5 用 **extern** 声明外部变量

外部变量(即全局变量)是在函数的外部定义的, 它的作用域为从变量定义处开始, 到本程序文件的末尾。如果外部变量不在文件的开头定义, 其有效的作用范围只限于定义处到文件终了。如果在定义点之前的函数想引用该外部变量, 则应该在引用之前用关键字 **extern** 对该变量作“外部变量声明”。表示该变量是一个已经定义的外部变量。有了此声明, 就可以从“声明”处起, 合法地使用该外部变量。

【例 8.18】用 **extern** 声明外部变量, 扩展程序文件中的作用域。

```
int max(int x, int y)
{
    int z;
    z = x > y ? x : y;
    return(z);
}

main()
{
    extern A, B;
    printf("%d\n", max(A, B));
}

int A=13, B=-8;
```



说明: 在本程序文件的最后 1 行定义了外部变量 A, B, 但由于外部变量定义的位置在函数 **main** 之后, 因此本来在 **main** 函数中不能引用外部变量 A, B。现在我们在 **main** 函数中用 **extern** 对 A 和 B 进行“外部变量声明”, 就可以从“声明”处起, 合法地使用该外部变量 A 和 B。

9	预处理命令.....	1
9.1	概述.....	1
9.2	宏定义.....	1
9.2.1	无参宏定义.....	1
9.2.2	带参宏定义.....	4
9.3	文件包含.....	8
9.4	条件编译.....	9
9.5	本章小结.....	11

9 预处理命令

9.1 概述

在前面各章中，已多次使用过以“#”开头的预处理命令。如包含命令#include，宏定义命令#define等。在源程序中这些命令都放在函数之外，而且一般都放在源文件的前面，它们称为预处理部分。

所谓预处理是指在进行编译的第一遍扫描(词法扫描和语法分析)之前所作的工作。预处理是C语言的一个重要功能，它由预处理程序负责完成。当对一个源文件进行编译时，系统将自动引用预处理程序对源程序中的预处理部分作处理，处理完毕自动进入对源程序的编译。

C语言提供了多种预处理功能，如宏定义、文件包含、条件编译等。合理地使用预处理功能编写的程序便于阅读、修改、移植和调试，也有利于模块化程序设计。本章介绍常用的几种预处理功能。

9.2 宏定义

在C语言源程序中允许用一个标识符来表示一个字符串，称为“宏”。被定义为“宏”的标识符称为“宏名”。在编译预处理时，对程序中所有出现的“宏名”，都用宏定义中的字符串去代换，这称为“宏代换”或“宏展开”。

宏定义是由源程序中的宏定义命令完成的。宏代换是由预处理程序自动完成的。

在C语言中，“宏”分为有参数和无参数两种。下面分别讨论这两种“宏”的定义和调用。

9.2.1 无参宏定义

无参宏的宏名后不带参数。

其定义的一般形式为：

#define 标识符 字符串

其中的“#”表示这是一条预处理命令。凡是以“#”开头的均为预处理命令。“define”为宏定义命令。“标识符”为所定义的宏名。“字符串”可以是常数、表达式、格式串等。

在前面介绍过的符号常量的定义就是一种无参宏定义。此外，常对程序中反复使用的表

达式进行宏定义。

例如：

```
#define M (y*y+3*y)
```

它的作用是指定标识符 M 来代替表达式 (y*y+3*y)。在编写源程序时，所有的 (y*y+3*y) 都可由 M 代替，而对源程序作编译时，将先由预处理程序进行宏代换，即用 (y*y+3*y) 表达式去置换所有的宏名 M，然后再进行编译。

【例 9.1】

```
#define M (y*y+3*y)
```

```
main() {
    int s,y;
    printf("input a number: ");
    scanf("%d",&y);
    s=3*M+4*M+5*M;
    printf("s=%d\n",s);
}
```



上例程序中首先进行宏定义，定义 M 来替代表达式 (y*y+3*y)，在 s=3*M+4*M+5*M 中作了宏调用。在预处理时经宏展开后该语句变为：

$$s=3*(y*y+3*y)+4*(y*y+3*y)+5*(y*y+3*y);$$

但要注意的是，在宏定义中表达式 (y*y+3*y) 两边的括号不能少。否则会发生错误。如当作以下定义后：

```
#define M y*y+3*y
```

在宏展开时将得到下述语句：

$$s=3*y*y+3*y+4*y*y+3*y+5*y*y+3*y;$$

这相当于：

$$3y^2+3y+4y^2+3y+5y^2+3y;$$

显然与原题意要求不符。计算结果当然是错误的。因此在作宏定义时必须十分注意。应保证在宏代换之后不发生错误。

对于宏定义还要说明以下几点：

- 1) 宏定义是用宏名来表示一个字符串，在宏展开时又以该字符串取代宏名，这只是一种简单的代换，字符串中可以含任何字符，可以是常数，也可以是表达式，预处理程序对它不作任何检查。如有错误，只能在编译已被宏展开后的源程序时发现。
- 2) 宏定义不是说明或语句，在行末不必加分号，如加上分号则连分号也一起置换。
- 3) 宏定义必须写在函数之外，其作用域为宏定义命令起到源程序结束。如要终止其作用域可使用 # undef 命令。

例如：

```
#define PI 3.14159
```

```
main()
```

```
{
```

```
.....
```

```

    }
    #undef PI
    f1()
    {
        .....
    }

```

表示PI 只在main 函数中有效，在 f1 中无效。

4) 宏名在源程序中若用引号括起来，则预处理程序不对其作宏代换。

【例 9.2】

```

#define OK 100
main()
{
    printf("OK");
    printf("\n");
}

```



上例中定义宏名 OK 表示 100，但在 printf 语句中 OK 被引号括起来，因此不作宏代换。程序的运行结果为：OK 这表示把“OK”当字符串处理。

5) 宏定义允许嵌套，在宏定义的字符串中可以使用已经定义的宏名。在宏展开时由预处理程序层层代换。

例如：

```

#define PI 3.1415926
#define S PI*y*y          /* PI 是已定义的宏名*/

```

对语句：

```
printf("%f",S);
```

在宏代换后变为：

```
printf("%f",3.1415926*y*y);
```

6) 习惯上宏名用大写字母表示，以便于与变量区别。但也允许用小写字母。

7) 可用宏定义表示数据类型，使书写方便。

例如：

```
#define STU struct stu
```

在程序中可用 STU 作变量说明：

```
STU body[5],*p;
```

```
#define INTEGER int
```

在程序中即可用 INTEGER 作整型变量说明：

```
INTEGER a,b;
```

应注意用宏定义表示数据类型和用 typedef 定义数据说明符的区别。

宏定义只是简单的字符串代换，是在预处理完成的，而 typedef 是在编译时处理的，它不是作简单的代换，而是对类型说明符重新命名。被命名的标识符具有类型定义说明的功能。请看下面的例子：

```

#define PIN1 int *
typedef (int *) PIN2;

```

从形式上看这两者相似，但在实际使用中却不相同。

下面用 PIN1, PIN2 说明变量时就可以看出它们的区别:

PIN1 a,b;在宏代换后变成:

```
int *a,b;
```

表示 a 是指向整型的指针变量, 而 b 是整型变量。

然而:

```
PIN2 a,b;
```

表示 a, b 都是指向整型的指针变量。因为 PIN2 是一个类型说明符。由这个例子可见, 宏定义虽然也可表示数据类型, 但毕竟是作字符代换。在使用时要分外小心, 以避出错。

8) 对“输出格式”作宏定义, 可以减少书写麻烦。

【例 9.3】中就采用了这种方法。

```
#define P printf
#define D "%d\n"
#define F "%f\n"
main() {
    int a=5, c=8, e=11;
    float b=3.8, d=9.7, f=21.08;
    P(D F, a, b);
    P(D F, c, d);
    P(D F, e, f);
}
```



9.2.2 带参宏定义

C 语言允许宏带有参数。在宏定义中的参数称为形式参数, 在宏调用中的参数称为实际参数。

对带参数的宏, 在调用中, 不仅要宏展开, 而且要用实参去代换形参。

带参宏定义的一般形式为:

```
#define 宏名(形参表) 字符串
```

在字符串中含有各个形参。

带参宏调用的一般形式为:

```
宏名(实参表);
```

例如:

```
#define M(y) y*y+3*y      /*宏定义*/
.....
k=M(5);                  /*宏调用*/
.....
```

在宏调用时, 用实参 5 去代替形参 y, 经预处理宏展开后的语句为:

```
k=5*5+3*5
```

【例 9.4】

```
#define MAX(a,b) (a>b)?a:b
main() {
    int x,y,max;
```

```
printf("input two numbers:  ");
scanf("%d%d",&x,&y);
max=MAX(x,y);
printf("max=%d\n",max);
}
```



上例程序的第一行进行带参宏定义，用宏名 MAX 表示条件表达式 $(a>b)?a:b$ ，形参 a, b 均出现在条件表达式中。程序第七行 `max=MAX(x, y)` 为宏调用，实参 x, y，将代换形参 a, b。宏展开后该语句为：

```
max=(x>y)?x:y;
```

用于计算 x, y 中的大数。

对于带参的宏定义有以下问题需要说明：

1. 带参宏定义中，宏名和形参表之间不能有空格出现。

例如把：

```
#define MAX(a, b) (a>b)?a:b
```

写为：

```
#define MAX (a, b) (a>b)?a:b
```

将被认为是无参宏定义，宏名 MAX 代表字符串 `(a, b) (a>b)?a:b`。宏展开时，宏调用语句：

```
max=MAX(x, y);
```

将变为：

```
max=(a, b) (a>b)?a:b(x, y);
```

这显然是错误的。

2. 在带参宏定义中，形式参数不分配内存单元，因此不必作类型定义。而宏调用中的实参有具体的值。要用它们去代换形参，因此必须作类型说明。这是与函数中的情况不同的。在函数中，形参和实参是两个不同的量，各有自己的作用域，调用时要把实参值赋予形参，进行“值传递”。而在带参宏中，只是符号代换，不存在值传递的问题。
3. 在宏定义中的形参是标识符，而宏调用中的实参可以是表达式。

【例 9.5】

```
#define SQ(y) (y)*(y)
main() {
    int a, sq;
    printf("input a number:  ");
    scanf("%d",&a);
    sq=SQ(a+1);
    printf("sq=%d\n", sq);
}
```



上例中第一行为宏定义，形参为 y。程序第七行宏调用中实参为 `a+1`，是一个表达式，在宏展开时，用 `a+1` 代换 y，再用 `(y)*(y)` 代换 SQ，得到如下语句：

```
sq=(a+1)*(a+1);
```

这与函数的调用是不同的，函数调用时要把实参表达式的值求出来再赋予形参。而宏代

换中对实参表达式不作计算直接地照原样代换。

4. 在宏定义中, 字符串内的形参通常要用括号括起来以避免出错。在上例中的宏定义中 $(y)*(y)$ 表达式的 y 都用括号括起来, 因此结果是正确的。如果去掉括号, 把程序改为以下形式:

【例 9.6】

```
#define SQ(y) y*y
main() {
    int a, sq;
    printf("input a number:  ");
    scanf("%d", &a);
    sq=SQ(a+1);
    printf("sq=%d\n", sq);
}
```



运行结果为:

```
input a number:3
sq=7
```

同样输入 3, 但结果却是不一样的。问题在哪里呢? 这是由于代换只作符号代换而不作其它处理而造成的。宏代换后将得到以下语句:

```
sq=a+1*a+1;
```

由于 a 为 3 故 sq 的值为 7。这显然与题意相违, 因此参数两边的括号是不能少的。即使在参数两边加括号还是不够的, 请看下面程序:

【例 9.7】

```
#define SQ(y) (y)*(y)
main() {
    int a, sq;
    printf("input a number:  ");
    scanf("%d", &a);
    sq=160/SQ(a+1);
    printf("sq=%d\n", sq);
}
```



本程序与前例相比, 只把宏调用语句改为:

```
sq=160/SQ(a+1);
```

运行本程序如输入值仍为 3 时, 希望结果为 10。但实际运行的结果如下:

```
input a number:3
sq=160
```

为什么会得这样的结果呢? 分析宏调用语句, 在宏代换之后变为:

```
sq=160/(a+1)*(a+1);
```

a 为 3 时, 由于 $"/"$ 和 $"*"$ 运算符优先级和结合性相同, 则先作 $160/(3+1)$ 得 40, 再作 $40*(3+1)$ 最后得 160。为了得到正确答案应在宏定义中的整个字符串外加括号, 程序修改如下:

【例 9.8】

```
#define SQ(y) ((y)*(y))
main() {
    int a, sq;
    printf("input a number:  ");
    scanf("%d", &a);
    sq=160/SQ(a+1);
    printf("sq=%d\n", sq);
}
```



以上讨论说明，对于宏定义不仅应在参数两侧加括号，也应在整个字符串外加括号。

5. 带参的宏和带参函数很相似，但有本质上的不同，除上面已谈到的各点外，把同一表达式用函数处理与用宏处理两者的结果有可能是不同的。

【例 9.9】

```
main() {
    int i=1;
    while(i<=5)
        printf("%d\n", SQ(i++));
}
SQ(int y)
{
    return((y)*(y));
}
```



【例 9.10】

```
#define SQ(y) ((y)*(y))
main() {
    int i=1;
    while(i<=5)
        printf("%d\n", SQ(i++));
}
```



在例 9.9 中函数名为 SQ，形参为 Y，函数体表达式为 ((y)*(y))。在例 9.10 中宏名为 SQ，形参也为 y，字符串表达式为 (y)*(y))。例 9.9 的函数调用为 SQ(i++)，例 9.10 的宏调用为 SQ(i++)，实参也是相同的。从输出结果来看，却大不相同。

分析如下：在例 9.9 中，函数调用是把实参 i 值传给形参 y 后自增 1。然后输出函数值。因而要循环 5 次。输出 1~5 的平方值。而在例 9.10 中宏调用时，只作代换。SQ(i++) 被代换为 ((i++)*(i++))。在第一次循环时，由于 i 等于 1，其计算过程为：表达式中前一个 i 初值为 1，然后 i 自增 1 变为 2，因此表达式中第 2 个 i 初值为 2，两相乘的结果也为 2，然后 i 值再自增 1，得 3。在第二次循环时，i 值已有初值为 3，因此表达式中前一个 i 为 3，后一个 i 为 4，乘积为 12，然后 i 再自增 1 变为 5。进入第三次循环，由于 i 值已为 5，所以这将是最后一次循环。计算表达式的值为 5*6 等于 30。i 值再自增 1 变为 6，不再满足循

环条件，停止循环。

从以上分析可以看出函数调用和宏调用二者在形式上相似，在本质上是完全不同的。

6. 宏定义也可用来定义多个语句，在宏调用时，把这些语句又代换到源程序内。看下面的例子。

【例 9.11】

```
#define SSSV(s1, s2, s3, v) s1=l*w;s2=l*h;s3=w*h;v=w*l*h;
main() {
    int l=3, w=4, h=5, sa, sb, sc, vv;
    SSSV(sa, sb, sc, vv);
    printf("sa=%d\nsb=%d\nsc=%d\nvv=%d\n", sa, sb, sc, vv);
}
```



程序第一行为宏定义，用宏名 SSSV 表示 4 个赋值语句，4 个形参分别为 4 个赋值符左部的变量。在宏调用时，把 4 个语句展开并用实参代替形参。使计算结果送入实参之中。

9.3 文件包含

文件包含是 C 预处理程序的另一个重要功能。

文件包含命令的一般形式为：

```
#include"文件名"
```

在前面我们已多次用此命令包含过库函数的头文件。例如：

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

```
#include"math.h"
```

文件包含命令的功能是把指定的文件插入该命令行位置取代该命令行，从而把指定的文件和当前的源程序文件连成一个源文件。

在程序设计中，文件包含是很有用的。一个大的程序可以分为多个模块，由多个程序员分别编程。有些公用的符号常量或宏定义等可单独组成一个文件，在其它文件的开头用包含命令包含该文件即可使用。这样，可避免在每个文件开头都去书写那些公用量，从而节省时间，并减少出错。

对文件包含命令还要说明以下几点：

1. 包含命令中的文件名可以用双引号括起来，也可以用尖括号括起来。例如以下写法都是允许的：

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

```
#include<math.h>
```

但是这两种形式是有区别的：使用尖括号表示在包含文件目录中去查找（包含目录是由用户在设置环境时设置的），而不在源文件目录去查找；

使用双引号则表示首先在当前的源文件目录中查找，若未找到才到包含目录中去查找。用户编程时可根据自己文件所在的目录来选择某一种命令形式。

2. 一个 include 命令只能指定一个被包含文件，若有多个文件要包含，则需用多个 include 命令。
3. 文件包含允许嵌套，即在一个被包含的文件中又可以包含另一个文件。

9.4 条件编译

预处理程序提供了条件编译的功能。可以按不同的条件去编译不同的程序部分，因而产生不同的目标代码文件。这对于程序的移植和调试是很有用的。

条件编译有三种形式，下面分别介绍：

1. 第一种形式：

```
#ifdef 标识符
    程序段 1
#else
    程序段 2
#endif
```

它的功能是，如果标识符已被 #define 命令定义过则对程序段 1 进行编译；否则对程序段 2 进行编译。如果没有程序段 2(它为空)，本格式中的#else 可以没有，即可以写为：

```
#ifdef 标识符
    程序段
#endif
```

【例 9.12】

```
#define NUM ok
main() {
    struct stu
    {
        int num;
        char *name;
        char sex;
        float score;
    } *ps;
    ps=(struct stu*)malloc(sizeof(struct stu));
    ps->num=102;
    ps->name="Zhang ping";
    ps->sex='M';
    ps->score=62.5;
    #ifdef NUM
    printf("Number=%d\nScore=%f\n", ps->num, ps->score);
    #else
    printf("Name=%s\nSex=%c\n", ps->name, ps->sex);
    #endif
    free(ps);
}
```



由于在程序的第 16 行插入了条件编译预处理命令，因此要根据 NUM 是否被定义过来决定编译那一个 printf 语句。而在程序的第一行已对 NUM 作过宏定义，因此应对第一个 printf

语句作编译故运行结果是输出了学号和成绩。

在程序的第一行宏定义中，定义 NUM 表示字符串 OK，其实也可以为任何字符串，甚至不给出任何字符串，写为：

```
#define NUM
```

也具有同样的意义。只有取消程序的第一行才会去编译第二个 printf 语句。读者可上机试作。

2. 第二种形式：

```
#ifndef 标识符
    程序段 1
#else
    程序段 2
#endif
```

与第一种形式的区别是将“ifdef”改为“ifndef”。它的功能是，如果标识符未被#define 命令定义过则对程序段 1 进行编译，否则对程序段 2 进行编译。这与第一种形式的功能正相反。

3. 第三种形式：

```
#if 常量表达式
    程序段 1
#else
    程序段 2
#endif
```

它的功能是，如常量表达式的值为真(非 0)，则对程序段 1 进行编译，否则对程序段 2 进行编译。因此可以使程序在不同条件下，完成不同的功能。

【例 9.13】

```
#define R 1
main() {
    float c, r, s;
    printf("input a number: ");
    scanf("%f", &c);
    #if R
        r=3.14159*c*c;
        printf("area of round is: %f\n", r);
    #else
        s=c*c;
        printf("area of square is: %f\n", s);
    #endif
}
```



本例中采用了第三种形式的条件编译。在程序第一行宏定义中，定义 R 为 1，因此在条件编译时，常量表达式的值为真，故计算并输出圆面积。

上面介绍的条件编译当然也可以用条件语句来实现。但是用条件语句将会对整个源程序进行编译，生成的目标代码程序很长，而采用条件编译，则根据条件只编译其中的程序段 1 或程序段 2，生成的目标程序较短。如果条件选择的程序段很长，采用条件编译的方法是

十分必要的。

9.5 本章小结

1. 预处理功能是 C 语言特有的功能，它是在对源程序正式编译前由预处理程序完成的。程序员在程序中用预处理命令来调用这些功能。
2. 宏定义是用一个标识符来表示一个字符串，这个字符串可以是常量、变量或表达式。在宏调用中将用该字符串代换宏名。
3. 宏定义可以带有参数，宏调用时是以实参代换形参。而不是“值传送”。
4. 为了避免宏代换时发生错误，宏定义中的字符串应加括号，字符串中出现的形式参数两边也应加括号。
5. 文件包含是预处理的一个重要功能，它可用来把多个源文件连接成一个源文件进行编译，结果将生成一个目标文件。
6. 条件编译允许只编译源程序中满足条件的程序段，使生成的目标程序较短，从而减少了内存的开销并提高了程序的效率。
7. 使用预处理功能便于程序的修改、阅读、移植和调试，也便于实现模块化程序设计。

10	指针.....	1
10.1	地址指针的基本概念.....	1
10.2	变量的指针和指向变量的指针变量.....	2
10.2.1	定义一个指针变量.....	3
10.2.2	指针变量的引用.....	3
10.2.3	指针变量作为函数参数.....	7
10.2.4	指针变量几个问题的进一步说明.....	10
10.3	数组指针和指向数组的指针变量.....	13
10.3.1	指向数组元素的指针.....	13
10.3.2	通过指针引用数组元素.....	14
10.3.3	数组名作函数参数.....	16
10.3.4	指向多维数组的指针和指针变量.....	22
10.4	字符串的指针指向字符串的指针变量.....	25
10.4.1	字符串的表示形式.....	25
10.4.2	使用字符串指针变量与字符数组的区别.....	28
10.5	函数指针变量.....	29
10.6	指针型函数.....	30
10.7	指针数组和指向指针的指针.....	31
10.7.1	指针数组的概念.....	31
10.7.2	指向指针的指针.....	34
10.7.3	main 函数的参数.....	36
10.8	有关指针的数据类型和指针运算的小结.....	37
10.8.1	有关指针的数据类型的小结.....	37
10.8.2	指针运算的小结.....	37
10.8.3	void 指针类型.....	38

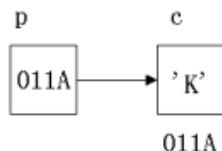
10 指针

指针是C语言中广泛使用的一种数据类型。运用指针编程是C语言最主要的风格之一。利用指针变量可以表示各种数据结构；能很方便地使用数组和字符串；并能象汇编语言一样处理内存地址，从而编出精练而高效的程序。指针极大地丰富了C语言的功能。学习指针是学习C语言中最重要的一环，能否正确理解和使用指针是我们是否掌握C语言的一个标志。同时，指针也是C语言中最为困难的一部分，在学习除了要正确理解基本概念，还必须要多编程，上机调试。只要作到这些，指针也是不难掌握的。

10.1 地址指针的基本概念

在计算机中，所有的数据都是存放在存储器中的。一般把存储器中的一个字节称为一个内存单元，不同的数据类型所占用的内存单元数不等，如整型量占2个单元，字符型占1个单元等，在前面已有详细的介绍。为了正确地访问这些内存单元，必须为每个内存单元编上号。根据一个内存单元的编号即可准确地找到该内存单元。内存单元的编号也叫做地址。

既然根据内存单元的编号或地址就可以找到所需的内存单元，所以通常也把这个地址称为指针。内存单元的指针和内存单元的内容是两个不同的概念。可以用一个通俗的例子来说明它们之间的关系。我们到银行去存取款时，银行工作人员将根据我们的帐号去找我们的存款单，找到之后在存单上写入存款、取款的金额。在这里，帐号就是存单的指针，存款数是存单的内容。对于一个内存单元来说，单元的地址即为指针，其中存放的数据才是该单元的内容。在C语言中，允许用一个变量来存放指针，这种变量称为指针变量。因此，一个指针变量的值就是某个内存单元的地址或称为某内存单元的指针。



图中，设有字符变量C，其内容为“K”（ASCII码为十进制数75），C占用了011A号单元（地址用十六进制数表示）。设有指针变量P，内容为011A，这种情况我们称为P指向变量C，或说P是指向变量C的指针。

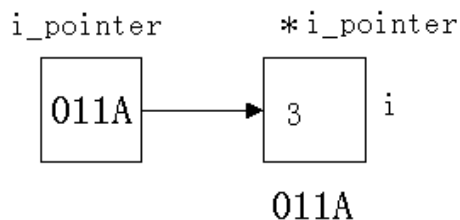
严格地说，一个指针是一个地址，是一个常量。而一个指针变量却可以被赋予不同的指针值，是变量。但常把指针变量简称为指针。为了避免混淆，我们约定：“指针”是指地址，是常量，“指针变量”是指取值为地址的变量。定义指针的目的是为了通过指针去访问内存单元。

既然指针变量的值是一个地址，那么这个地址不仅可以是变量的地址，也可以是其它数据结构的地址。在一个指针变量中存放一个数组或一个函数的首地址有何意义呢？因为数组或函数都是连续存放的。通过访问指针变量取得了数组或函数的首地址，也就找到了该数组或函数。这样一来，凡是出现数组、函数的地方都可以用一个指针变量来表示，只要该指针变量中赋予数组或函数的首地址即可。这样做，将会使程序的概念十分清楚，程序本身也精练，高效。在C语言中，一种数据类型或数据结构往往都占有一组连续的内存单元。用“地址”这个概念并不能很好地描述一种数据类型或数据结构，而“指针”虽然实际上也是一个地址，但它却是一个数据结构的首地址，它是“指向”一个数据结构的，因而概念更为清楚，表示更为明确。这也是引入“指针”概念的一个重要原因。

10.2 变量的指针和指向变量的指针变量

变量的**指针**就是变量的**地址**。存放变量**地址**的变量是**指针变量**。即在C语言中，允许用一个变量来存放指针，这种变量称为指针变量。因此，一个指针变量的值就是某个变量的地址或称为某变量的指针。

为了表示指针变量和它所指向的变量之间的关系，在程序中用“*”符号表示“指向”，例如，i_pointer代表指针变量，而*i_pointer是i_pointer所指向的变量。



因此，下面两个语句作用相同：

```
i=3;
```

```
*i_pointer=3;
```

第二个语句的含义是将 3 赋给指针变量 `i_pointer` 所指向的变量。

10.2.1 定义一个指针变量

对指针变量的定义包括三个内容：

- (1) 指针类型说明，即定义变量为一个指针变量；
- (2) 指针变量名；
- (3) 变量值(指针)所指向的变量的数据类型。

其一般形式为：

类型说明符 *变量名；

其中，*表示这是一个指针变量，变量名即为定义的指针变量名，类型说明符表示本指针变量所指向的变量的数据类型。

例如： `int *p1;`

表示 `p1` 是一个指针变量，它的值是某个整型变量的地址。或者说 `p1` 指向一个整型变量。

至于 `p1` 究竟指向哪一个整型变量，应由向 `p1` 赋予的地址来决定。

再如：

```
int *p2;          /*p2 是指向整型变量的指针变量*/
```

```
float *p3;        /*p3 是指向浮点变量的指针变量*/
```

```
char *p4;         /*p4 是指向字符变量的指针变量*/
```

应该注意的是，一个指针变量只能指向同类型的变量，如 `P3` 只能指向浮点变量，不能时而指向一个浮点变量，时而又指向一个字符变量。

10.2.2 指针变量的引用

指针变量同普通变量一样，使用之前不仅要定义说明，而且必须赋予具体的值。未经赋值的指针变量不能使用，否则将造成系统混乱，甚至死机。指针变量的赋值只能赋予地址，决不能赋予任何其它数据，否则将引起错误。在 C 语言中，变量的地址是由编译系统分配的，对用户完全透明，用户不知道变量的具体地址。

两个有关的运算符：

- 1) `&`:取地址运算符。
- 2) `*`: 指针运算符（或称“间接访问”运算符）。

C 语言中提供了地址运算符 `&` 来表示变量的地址。

其一般形式为：

`&变量名;`

如 `&a` 表示变量 `a` 的地址，`&b` 表示变量 `b` 的地址。变量本身必须预先说明。

设有指向整型变量的指针变量 `p`，如要把整型变量 `a` 的地址赋予 `p` 可以有以下两种方式：

- (1) 指针变量初始化的方法

```
int a;
```

```
int *p=&a;
```

- (2) 赋值语句的方法

```
int a;  
int *p;  
p=&a;
```

不允许把一个数赋予指针变量，故下面的赋值是错误的：

```
int *p;  
p=1000;
```

被赋值的指针变量前不能再加 “*”说明符，如写为 *p=&a 也是错误的。

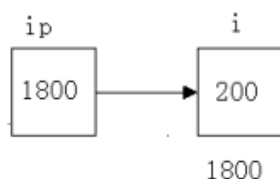
假设：

```
int i=200, x;  
int *ip;
```

我们定义了两个整型变量 i, x, 还定义了一个指向整型数的指针变量 ip。i, x 中可存放整数, 而 ip 中只能存放整型变量的地址。我们可以把 i 的地址赋给 ip:

```
ip=&i;
```

此时指针变量 ip 指向整型变量 i, 假设变量 i 的地址为 1800, 这个赋值可形象理解为下图所示的联系。



以后我们便可以通过指针变量 ip 间接访问变量 i, 例如:

```
x=*ip;
```

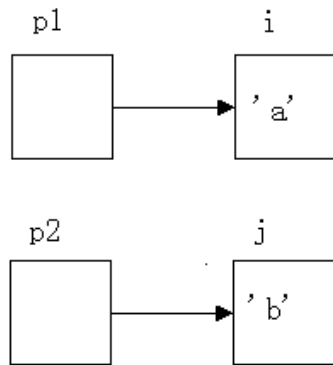
运算符*访问以 ip 为地址的存储区域, 而 ip 中存放的是变量 i 的地址, 因此, *ip 访问的是地址为 1800 的存储区域(因为是整数, 实际上是从 1800 开始的两个字节), 它就是 i 所占用的存储区域, 所以上面的赋值表达式等价于

```
x=i;
```

另外, 指针变量和一般变量一样, 存放在它们之中的值是可以改变的, 也就是说可以改变它们的指向, 假设

```
int i, j, *p1, *p2;  
i='a';  
j='b';  
p1=&i;  
p2=&j;
```

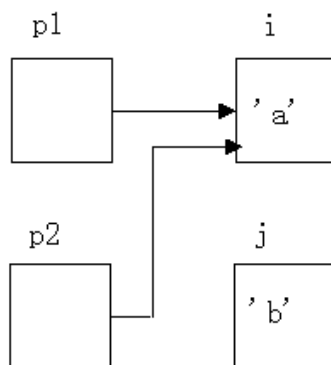
则建立如下图所示的联系:



这时赋值表达式:

`p2=p1`

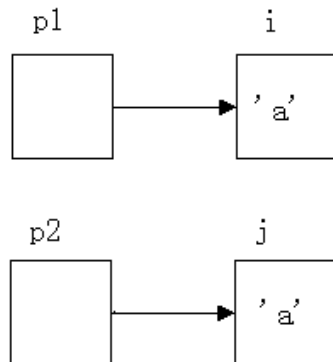
就使 p2 与 p1 指向同一对象 i, 此时 *p2 就等价于 i, 而不是 j, 图所示:



如果执行如下表达式:

`*p2=*p1;`

则表示把 p1 指向的内容赋给 p2 所指的区域, 此时就变成图所示



通过指针访问它所指向的一个变量是以间接访问的形式进行的, 所以比直接访问一个变量要费时间, 而且不直观, 因为通过指针要访问哪一个变量, 取决于指针的值(即指向), 例如

“*p2=*p1;”实际上就是“j=i;”,前者不仅速度慢而且目的不明。但由于指针是变量,我们可以通过改变它们的指向,以间接访问不同的变量,这给程序员带来灵活性,也使程序代码编写得更为简洁和有效。

指针变量可出现在表达式中, 设

```
int x, y, *px=&x;
```

指针变量 px 指向整数 x, 则 *px 可出现在 x 能出现的任何地方。例如:

```
y=*px+5; /*表示把 x 的内容加 5 并赋给 y*/
```

```
y=++*px; /**px 的内容加上 1 之后赋给 y, ++*px 相当于++(*px)*/
```

```
y=*px++; /*相当于 y=*px; px++*/
```

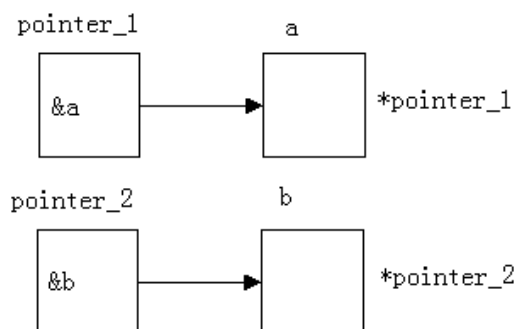
【例 10.1】

```
main()
{ int a,b;
  int *pointer_1, *pointer_2;
  a=100;b=10;
  pointer_1=&a;
  pointer_2=&b;
  printf("%d,%d\n",a,b);
  printf("%d,%d\n",*pointer_1,*pointer_2);
}
```



对程序的说明:

- 1) 在开头处虽然定义了两个指针变量 pointer_1 和 pointer_2, 但它们并未指向任何一个整型变量。只是提供两个指针变量, 规定它们可以指向整型变量。程序第 5、6 行的作用就是使 pointer_1 指向 a, pointer_2 指向 b。



- 2) 最后一行的 *pointer_1 和 *pointer_2 就是变量 a 和 b。最后两个 printf 函数作用是相同的。
- 3) 程序中有两处出现 *pointer_1 和 *pointer_2, 请区分它们的不同含义。
- 4) 程序第 5、6 行的 “pointer_1=&a” 和 “pointer_2=&b” 不能写成 “*pointer_1=&a” 和 “*pointer_2=&b”。

请对下面再的关于 “&” 和 “*” 的问题进行考虑:

- 1) 如果已经执行了 “pointer_1=&a;” 语句, 则 &*pointer_1 是什么含义?

2) *a 含义是什么?

3) (pointer_1)++和 pointer_1++的区别?

【例 10.2】输入 a 和 b 两个整数，按先大后小的顺序输出 a 和 b。

```
main()
{
    int *p1,*p2,*p,a,b;
    scanf("%d,%d",&a,&b);
    p1=&a;p2=&b;
    if(a<b)
        {p=p1;p1=p2;p2=p;}
    printf("\na=%d,b=%d\n",a,b);
    printf("max=%d,min=%d\n",*p1,*p2);
}
```



10.2.3 指针变量作为函数参数

函数的参数不仅可以是整型、实型、字符型等数据，还可以是指针类型。它的作用是将一个变量的地址传送到另一个函数中。

【例 10.3】题目同例 10.2，即输入的两个整数按大小顺序输出。今用函数处理，而且用指针类型的数据作函数参数。

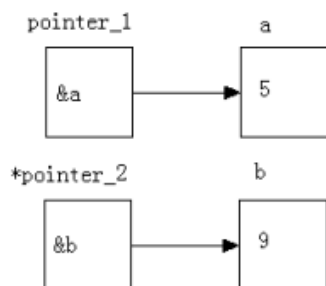
```
swap(int *p1,int *p2)
{
    int temp;
    temp=*p1;
    *p1=*p2;
    *p2=temp;
}

main()
{
    int a,b;
    int *pointer_1,*pointer_2;
    scanf("%d,%d",&a,&b);
    pointer_1=&a;pointer_2=&b;
    if(a<b) swap(pointer_1,pointer_2);
    printf("\n%d,%d\n",a,b);
}
```

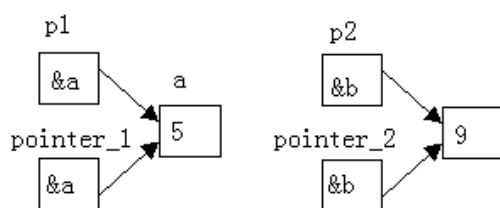


对程序的说明：

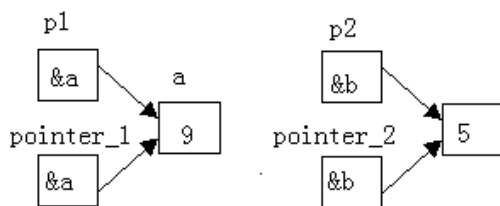
swap 是用户定义的函数，它的作用是交换两个变量（a 和 b）的值。swap 函数的形参 p1、p2 是指针变量。程序运行时，先执行 main 函数，输入 a 和 b 的值。然后将 a 和 b 的地址分别赋给指针变量 pointer_1 和 pointer_2，使 pointer_1 指向 a，pointer_2 指向 b。



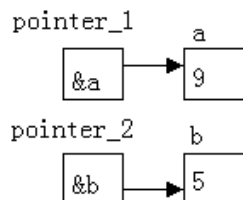
接着执行 if 语句，由于 $a < b$ ，因此执行 swap 函数。注意实参 pointer_1 和 pointer_2 是指针变量，在函数调用时，将实参变量的值传递给形参变量。采取的依然是“值传递”方式。因此虚实结合后形参 p1 的值为 &a，p2 的值为 &b。这时 p1 和 pointer_1 指向变量 a，p2 和 pointer_2 指向变量 b。



接着执行 swap 函数的函数体使 *p1 和 *p2 的值互换，也就是使 a 和 b 的值互换。



函数调用结束后，p1 和 p2 不复存在（已释放）如图。



最后在 main 函数中输出的 a 和 b 的值是已经过交换的值。

请注意交换 *p1 和 *p2 的值是如何实现的。请找出下列程序段的错误：

```

swap(int *p1, int *p2)
{
    int *temp;
    *temp = *p1;    /*此语句有问题*/
    *p1 = *p2;
    *p2 = *temp;
}
  
```

```

}
```

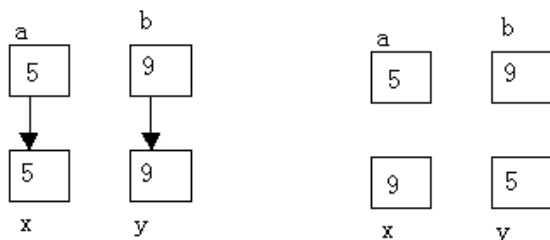
请考虑下面的函数能否实现 a 和 b 互换。

```

swap(int x, int y)
{
    int temp;
    temp=x;
    x=y;
    y=temp;
}

```

如果在 main 函数中用“swap(a, b);”调用 swap 函数，会有什么结果呢？请看下图所示。



【例 10.4】请注意，不能企图通过改变指针形参的值而使指针实参的值改变。

```

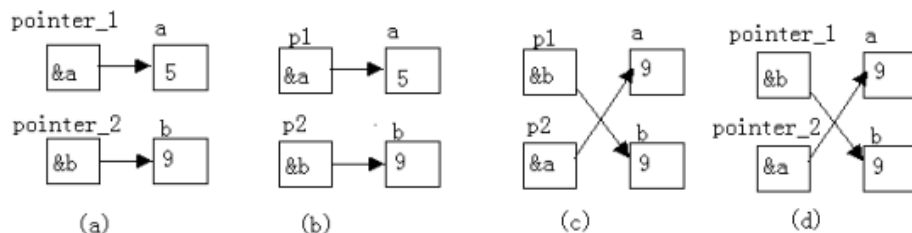
swap(int *p1, int *p2)
{
    int *p;
    p=p1;
    p1=p2;
    p2=p;
}

main()
{
    int a, b;
    int *pointer_1, *pointer_2;
    scanf("%d, %d", &a, &b);
    pointer_1=&a; pointer_2=&b;
    if(a<b) swap(pointer_1, pointer_2);
    printf("\n%d, %d\n", *pointer_1, *pointer_2);
}

```



其中的问题在于不能实现如图所示的第四步（d）。



【例 10.5】输入 a、b、c 3 个整数，按大小顺序输出。

```

swap(int *pt1, int *pt2)
{
    int temp;
    temp=*pt1;
    *pt1=*pt2;
    *pt2=temp;
}

exchange(int *q1, int *q2, int *q3)
{
    if(*q1<*q2) swap(q1, q2);
    if(*q1<*q3) swap(q1, q3);
    if(*q2<*q3) swap(q2, q3);
}

main()
{
    int a, b, c, *p1, *p2, *p3;
    scanf("%d, %d, %d", &a, &b, &c);
    p1=&a; p2=&b; p3=&c;
    exchange(p1, p2, p3);
    printf("\n%d, %d, %d\n", a, b, c);
}

```



10.2.4 指针变量几个问题的进一步说明

指针变量可以进行某些运算，但其运算的种类是有限的。它只能进行赋值运算和部分算术运算及关系运算。

1. 指针运算符

- 1) 取地址运算符&: 取地址运算符&是单目运算符，其结合性为自右至左，其功能是取变量的地址。在 scanf 函数及前面介绍指针变量赋值中，我们已经了解并使用了&运算符。
- 2) 取内容运算符*: 取内容运算符*是单目运算符，其结合性为自右至左，用来表示指针变量所指的变量。在*运算符之后跟的变量必须是指针变量。

需要注意的是指针运算符*和指针变量说明中的指针说明符*不是一回事。在指针变量说明中，“*”是类型说明符，表示其后的变量是指针类型。而表达式中出现的“*”则是一个运算符用以表示指针变量所指的变量。

【例 10.6】

```

main() {
    int a=5, *p=&a;
    printf ("%d", *p);
}

```



表示指针变量 p 取得了整型变量 a 的地址。printf("%d", *p) 语句表示输出变量 a 的值。

2. 指针变量的运算

- 1) 赋值运算: 指针变量的赋值运算有以下几种形式。

- ① 指针变量初始化赋值，前面已作介绍。
 ② 把一个变量的地址赋予指向相同数据类型的指针变量。

例如：

```
int a, *pa;
pa=&a;    /*把整型变量 a 的地址赋予整型指针变量 pa*/
```

- ③ 把一个指针变量的值赋予指向相同类型变量的另一个指针变量。

如：

```
int a, *pa=&a, *pb;
pb=pa;    /*把 a 的地址赋予指针变量 pb*/
```

由于 pa, pb 均为指向整型变量的指针变量，因此可以相互赋值。

- ④ 把数组的首地址赋予指向数组的指针变量。

例如：

```
int a[5], *pa;
pa=a;
(数组名表示数组的首地址，故可赋予指向数组的指针变量 pa)
```

也可写为：

```
pa=&a[0]; /*数组第一个元素的地址也是整个数组的首地址，
```

也可赋予 pa*/

当然也可采取初始化赋值的方法：

```
int a[5], *pa=a;
```

- ⑤ 把字符串的首地址赋予指向字符类型的指针变量。

例如：

```
char *pc;
pc="C Language";
```

或用初始化赋值的方法写为：

```
char *pc="C Language";
```

这里应说明的是并不是把整个字符串装入指针变量，而是把存放该字符串的字符数组的首地址装入指针变量。在后面还将详细介绍。

- ⑥ 把函数的入口地址赋予指向函数的指针变量。

例如：

```
int (*pf)();
pf=f;    /*f 为函数名*/
```

2) 加减算术运算

对于指向数组的指针变量，可以加上或减去一个整数 n。设 pa 是指向数组 a 的指针变量，则 pa+n, pa-n, pa++, ++pa, pa--, --pa 运算都是合法的。指针变量加或减一个整数 n 的意义是把指针指向的当前位置(指向某数组元素)向前或向后移动 n 个位置。应该注意，数组指针变量向前或向后移动一个位置和地址加 1 或减 1 在概念上是不同的。因为数组可以有不同的类型，各种类型的数组元素所占的字节长度是不同的。如指针变量加 1，即向后移动 1 个位置表示指针变量指向下一个数据元素的首地址。而不是在原地址基础上加 1。例如：

```
int a[5], *pa;
pa=a;    /*pa 指向数组 a，也是指向 a[0]*/
pa=pa+2; /*pa 指向 a[2]，即 pa 的值为 &a[2]*/
```

指针变量的加减运算只能对数组指针变量进行，对指向其它类型变量的指针变量作加减运算是毫无意义的。

- 3) 两个指针变量之间的运算：只有指向同一数组的两个指针变量之间才能进行运算，否则运算毫无意义。

① 两指针变量相减：两指针变量相减所得之差是两个指针所指数组元素之间相差的元素个数。实际上是两个指针值(地址)相减之差再除以该数组元素的长度(字节数)。

例如 pf1 和 pf2 是指向同一浮点数组的两个指针变量，设 pf1 的值为 2010H，pf2 的值为 2000H，而浮点数组每个元素占 4 个字节，所以 pf1-pf2 的结果为 (2000H-2010H)/4=-4，表示 pf1 和 pf2 之间相差 4 个元素。两个指针变量不能进行加法运算。例如，pf1+pf2 是什么意思呢？毫无实际意义。

② 两指针变量进行关系运算：指向同一数组的两指针变量进行关系运算可表示它们所指数组元素之间的关系。

例如：

pf1==pf2 表示 pf1 和 pf2 指向同一数组元素；

pf1>pf2 表示 pf1 处于高地址位置；

pf1<pf2 表示 pf2 处于低地址位置。

指针变量还可以与 0 比较。

设 p 为指针变量，则 p==0 表明 p 是空指针，它不指向任何变量；

p!=0 表示 p 不是空指针。

空指针是由对指针变量赋予 0 值而得到的。

例如：

```
#define NULL 0
```

```
int *p=NULL;
```

对指针变量赋 0 值和不赋值是不同的。指针变量未赋值时，可以是任意值，是不能使用的。否则将造成意外错误。而指针变量赋 0 值后，则可以使用，只是它不指向具体的变量而已。

【例 10.7】

```
main() {
    int a=10,b=20,s,t,*pa,*pb; /*说明 pa,pb 为整型指针变量*/
    pa=&a;                    /*给指针变量 pa 赋值, pa 指向变量 a*/
    pb=&b;                    /*给指针变量 pb 赋值, pb 指向变量 b*/
    s=*pa+*pb;               /*求 a+b 之和, (*pa 就是 a, *pb 就是 b)*/
    t=*pa**pb;               /*本行是求 a*b 之积*/
    printf("a=%d\nb=%d\na+b=%d\na*b=%d\n", a, b, a+b, a*b);
    printf("s=%d\nt=%d\n", s, t);
}
```



【例 10.8】

```
main() {
    int a,b,c,*pmax,*pmin;    /*pmax,pmin 为整型指针变量*/
    printf("input three numbers:\n"); /*输入提示*/
    scanf("%d%d%d",&a,&b,&c); /*输入三个数字*/
    if(a>b) {                 /*如果第一个数字大于第二个数字... */
        pmax=&a;              /*指针变量赋值*/
        pmin=&b;              /*指针变量赋值*/
    }
```

```

else{
    pmax=&b;                /*指针变量赋值*/
    pmin=&a;}               /*指针变量赋值*/
if(c>*pmax) pmax=&c;        /*判断并赋值*/
if(c<*pmin) pmin=&c;        /*判断并赋值*/
printf("max=%d\nmin=%d\n",*pmax,*pmin); /*输出结果*/
}

```



10.3 数组指针和指向数组的指针变量

一个变量有一个地址，一个数组包含若干元素，每个数组元素都在内存中占用存储单元，它们都有相应的地址。所谓数组的指针是指数组的起始地址，数组元素的指针是数组元素的地址。

10.3.1 指向数组元素的指针

一个数组是由连续的一块内存单元组成的。数组名就是这块连续内存单元的首地址。一个数组也是由各个数组元素(下标变量)组成的。每个数组元素按其类型不同占有几个连续的内存单元。一个数组元素的首地址也是指它所占有的几个内存单元的首地址。

定义一个指向数组元素的指针变量的方法，与以前介绍的指针变量相同。

例如：

```

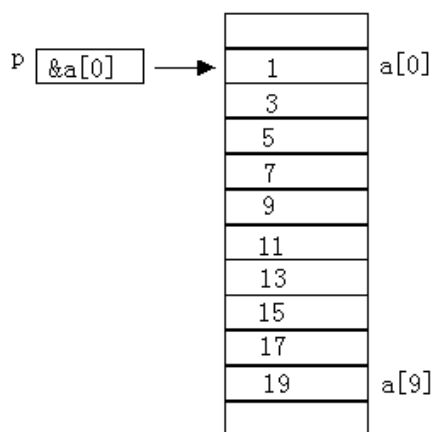
int a[10];    /*定义 a 为包含 10 个整型数据的数组*/
int *p;       /*定义 p 为指向整型变量的指针*/

```

应当注意，因为数组为 int 型，所以指针变量也应为指向 int 型的指针变量。下面是对指针变量赋值：

```
p=&a[0];
```

把 a[0] 元素的地址赋给指针变量 p。也就是说，p 指向 a 数组的第 0 号元素。



C 语言规定，数组名代表数组的首地址，也就是第 0 号元素的地址。因此，下面两个语句等价：

```
p=&a[0];
```

```
p=a;
```

在定义指针变量时可以赋给初值：

```
int *p=&a[0];
```

它等效于：

```
int *p;
```

```
p=&a[0];
```

当然定义时也可以写成：

```
int *p=a;
```

从图中我们可以看出有以下关系：

p, a, &a[0] 均指向同一单元，它们是数组 a 的首地址，也是 0 号元素 a[0] 的首地址。应该说明的是 p 是变量，而 a, &a[0] 都是常量。在编程时应予以注意。

数组指针变量说明的一般形式为：

类型说明符 *指针变量名；

其中类型说明符表示所指数组的类型。从一般形式可以看出指向数组的指针变量和指向普通变量的指针变量的说明是相同的。

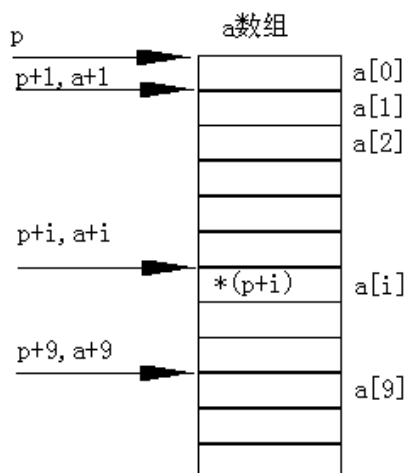
10.3.2 通过指针引用数组元素

C 语言规定：如果指针变量 p 已指向数组中的一个元素，则 p+1 指向同一数组中的下一个元素。

引入指针变量后，就可以用两种方法来访问数组元素了。

如果 p 的初值为 &a[0]，则：

- 1) p+i 和 a+i 就是 a[i] 的地址，或者说它们指向 a 数组的第 i 个元素。



- 2) *(p+i) 或 *(a+i) 就是 p+i 或 a+i 所指向的数组元素，即 a[i]。例如，*(p+5) 或 *(a+5) 就是 a[5]。

- 3) 指向数组的指针变量也可以带下标，如 p[i] 与 *(p+i) 等价。

根据以上叙述，引用一个数组元素可以用：

- 1) 下标法，即用 `a[i]` 形式访问数组元素。在前面介绍数组时都是采用这种方法。
- 2) 指针法，即采用 `*(a+i)` 或 `*(p+i)` 形式，用间接访问的方法来访问数组元素，其中 `a` 是数组名，`p` 是指向数组的指针变量，其初值 `p=a`。

【例 10.9】输出数组中的全部元素。（下标法）

```
main() {
    int a[10], i;
    for(i=0; i<10; i++)
        a[i]=i;
    for(i=0; i<5; i++)
        printf("a[%d]=%d\n", i, a[i]);
}
```



【例 10.10】输出数组中的全部元素。（通过数组名计算元素的地址，找出元素的值）

```
main() {
    int a[10], i;
    for(i=0; i<10; i++)
        *(a+i)=i;
    for(i=0; i<10; i++)
        printf("a[%d]=%d\n", i, *(a+i));
}
```



【例 10.11】输出数组中的全部元素。（用指针变量指向元素）

```
main() {
    int a[10], i, *p;
    p=a;
    for(i=0; i<10; i++)
        *(p+i)=i;
    for(i=0; i<10; i++)
        printf("a[%d]=%d\n", i, *(p+i));
}
```



【例 10.12】

```
main() {
    int a[10], i, *p=a;
    for(i=0; i<10; i++) {
        *p=i;
        printf("a[%d]=%d\n", i++, *p++);
    }
}
```



几个注意的问题:

- 1) 指针变量可以实现本身的值的改变。如 `p++` 是合法的; 而 `a++` 是错误的。因为 `a` 是数组名, 它是数组的首地址, 是常量。
- 2) 要注意指针变量的当前值。请看下面的程序。

【例 10.13】找出错误。

```
main() {
    int *p, i, a[10];
    p=a;
    for(i=0;i<10;i++)
        *p+=i;
    for(i=0;i<10;i++)
        printf("a[%d]=%d\n", i, *p++);
}
```



【例 10.14】改正。

```
main() {
    int *p, i, a[10];
    p=a;
    for(i=0;i<10;i++)
        *p+=i;
    p=a;
    for(i=0;i<10;i++)
        printf("a[%d]=%d\n", i, *p++);
}
```



- 3) 从上例可以看出, 虽然定义数组时指定它包含 10 个元素, 但指针变量可以指到数组以后的内存单元, 系统并不认为非法。
- 4) `*p++`, 由于 `++` 和 `*` 同优先级, 结合方向自右而左, 等价于 `*(p++)`。
- 5) `*(p++)` 与 `*(++p)` 作用不同。若 `p` 的初值为 `a`, 则 `*(p++)` 等价 `a[0]`, `*(++p)` 等价 `a[1]`。
- 6) `(*p)++` 表示 `p` 所指向的元素值加 1。
- 7) 如果 `p` 当前指向 `a` 数组中的第 `i` 个元素, 则
 - `*(p--)` 相当于 `a[i--]`;
 - `*(++p)` 相当于 `a[++i]`;
 - `*(--p)` 相当于 `a[--i]`。

10.3.3 数组名作函数参数

数组名可以作函数的实参和形参。如:

```
main()
{int array[10];
    .....
    .....
```

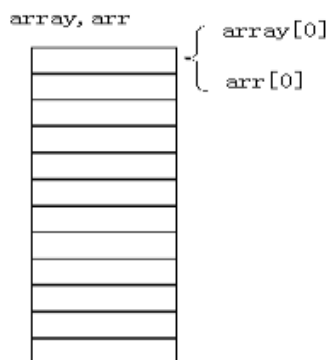
```

f(array, 10);
.....
.....
}

f(int arr[], int n);
{
.....
.....
}

```

array 为实参数组名，arr 为形参数组名。在学习指针变量之后就更容易理解这个问题了。数组名就是数组的首地址，实参向形参传送数组名实际上就是传送数组的地址，形参得到该地址后也指向同一数组。这就好象同一件物品有两个彼此不同的名称一样。



同样，指针变量的值也是地址，数组指针变量的值即为数组的首地址，当然也可作为函数的参数使用。

【例 10.15】

```

float aver(float *pa);
main() {
    float sco[5], av, *sp;
    int i;
    sp=sco;
    printf("\ninput 5 scores:\n");
    for(i=0; i<5; i++) scanf("%f", &sco[i]);
    av=aver(sp);
    printf("average score is %5.2f", av);
}
float aver(float *pa)
{
    int i;
    float av, s=0;
    for(i=0; i<5; i++) s=s+*pa++;
    av=s/5;
    return av;
}

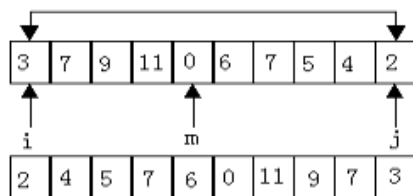
```

}



【例 10.16】将数组 a 中的 n 个整数按相反顺序存放。

算法为：将 $a[0]$ 与 $a[n-1]$ 对换，再 $a[1]$ 与 $a[n-2]$ 对换....., 直到将 $a[(n-1)/2]$ 与 $a[n-\text{int}((n-1)/2)]$ 对换。今用循环处理此问题，设两个“位置指示变量” i 和 j ， i 的初值为 0， j 的初值为 $n-1$ 。将 $a[i]$ 与 $a[j]$ 交换，然后使 i 的值加 1， j 的值减 1，再将 $a[i]$ 与 $a[j]$ 交换，直到 $i=(n-1)/2$ 为止，如图所示。



程序如下：

```
void inv(int x[], int n)    /*形参x是数组名*/
{
    int temp, i, j, m=(n-1)/2;
    for(i=0; i<=m; i++)
    {j=n-1-i;
      temp=x[i]; x[i]=x[j]; x[j]=temp;}
    return;
}

main()
{int i, a[10]={3, 7, 9, 11, 0, 6, 7, 5, 4, 2};
  printf("The original array:\n");
  for(i=0; i<10; i++)
    printf("%d, ", a[i]);
  printf("\n");
  inv(a, 10);
  printf("The array has been inverted:\n");
  for(i=0; i<10; i++)
    printf("%d, ", a[i]);
  printf("\n");
}
```



对此程序可以作一些改动。将函数 `inv` 中的形参 x 改成指针变量。

【例 10.17】对例 10.16 可以作一些改动。将函数 `inv` 中的形参 x 改成指针变量。

程序如下：

```
void inv(int *x, int n)    /*形参x为指针变量*/
{
    int *p, temp, *i, *j, m=(n-1)/2;
```



```

i=x; j=x+n-1; p=x+m;
for(; i<=p; i++, j--)
    {temp=*i; *i=*j; *j=temp;}
return;
}
main()
{int i, a[10]={3, 7, 9, 11, 0, 6, 7, 5, 4, 2};
printf("The original array:\n");
for(i=0; i<10; i++)
    printf("%d, ", a[i]);
printf("\n");
inv(a, 10);
printf("The array has been inverted:\n");
for(i=0; i<10; i++)
    printf("%d, ", a[i]);
printf("\n");
}

```



运行情况与前一程序相同。

【例 10.18】从 0 个数中找出其中最大值和最小值。

调用一个函数只能得到一个返回值，今用全局变量在函数之间“传递”数据。程序如下：

```

int max, min; /*全局变量*/
void max_min_value(int array[], int n)
{int *p, *array_end;
array_end=array+n;
max=min=*array;
for(p=array+1; p<array_end; p++)
    if(*p>max) max=*p;
    else if (*p<min) min=*p;
return;
}
main()
{int i, number[10];
printf("enter 10 integer umbers:\n");
for(i=0; i<10; i++)
    scanf("%d", &number[i]);
max_min_value(number, 10);
printf("\nmax=%d, min=%d\n", max, min);
}

```



说明：

- 1) 在函数 max_min_value 中求出的最大值和最小值放在 max 和 min 中。由于它们是全局，

因此在主函数中可以直接使用。

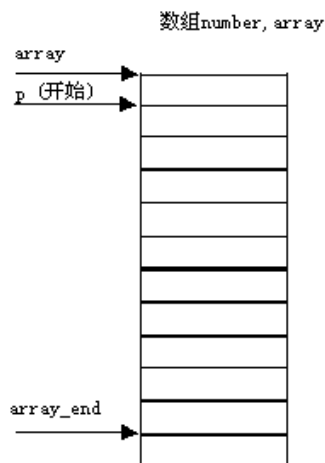
- 2) 函数 `max_min_value` 中的语句:

```
max=min=*array;
```

`array` 是数组名, 它接收从实参传来的数组 `number` 的首地址。

`*array` 相当于 `*(&array[0])`。上述语句与 `max=min=array[0];` 等价。

- 3) 在执行 `for` 循环时, `p` 的初值为 `array+1`, 也就是使 `p` 指向 `array[1]`。以后每次执行 `p++`, 使 `p` 指向下一个元素。每次将 `*p` 和 `max` 与 `min` 比较。将大者放入 `max`, 小者放 `min`。



- 4) 函数 `max_min_value` 的形参 `array` 可以改为指针变量类型。实参也可以不用数组名, 而用指针变量传递地址。

【例 10.19】程序可改为:

```
int max, min;          /*全局变量*/
void max_min_value(int *array, int n)
{int *p, *array_end;
 array_end=array+n;
 max=min=*array;
 for(p=array+1;p<array_end;p++)
     if(*p>max)max=*p;
     else if (*p<min)min=*p;
 return;
}
main()
{int i, number[10], *p;
 p=number;              /*使p指向number数组*/

 printf("enter 10 integer umbers:\n");

 for(i=0;i<10;i++, p++)
     scanf("%d", p);
 p=number;
 max_min_value(p, 10);
```

```
printf("\nmax=%d,min=%d\n",max,min);
}
```



归纳起来，如果有一个实参数组，想在函数中改变此数组的元素的值，实参与形参的对应关系有以下 4 种：

1) 形参和实参都是数组名。

```
main()                                f(int x[], int n)
{int a[10];                           {
    .....                             .....
    f(a, 10)                          }
    .....
}
```

a 和 x 指的是同一组数组。

2) 实用数组，形参用指针变量。

```
main()                                f(int *x, int n)
{int a[10];                           {
    .....                             .....
    f(a, 10)                          }
    .....
}
```

3) 实参、型参都用指针变量。

4) 实参为指针变量，型参为数组名。

【例 10.20】用实参指针变量改写将 n 个整数按相反顺序存放。

```
void inv(int *x, int n)
{int *p, m, temp, *i, *j;
  m=(n-1)/2;
  i=x; j=x+n-1; p=x+m;
  for(; i<=p; i++, j--)
    {temp=*i; *i=*j; *j=temp;}
  return;
}

main()
{int i, arr[10]={3, 7, 9, 11, 0, 6, 7, 5, 4, 2}, *p;
  p=arr;
  printf("The original array:\n");
  for(i=0; i<10; i++, p++)
    printf("%d, ", *p);
  printf("\n");
  p=arr;
  inv(p, 10);
  printf("The array has been inverted:\n");
  for(p=arr; p<arr+10; p++)
    printf("%d, ", *p);
  printf("\n");
}
```

}



注意：main 函数中的指针变量 p 是有确定值的。即如果用指针变作实参，必须现使指针变量有确定值，指向一个已定义的数组。

【例 10.21】用选择法对 10 个整数排序。

```
main()
{int *p, i, a[10]={3, 7, 9, 11, 0, 6, 7, 5, 4, 2};
  printf("The original array:\n");
  for(i=0; i<10; i++)
    printf("%d, ", a[i]);
  printf("\n");
  p=a;
  sort(p, 10);
  for(p=a, i=0; i<10; i++)
    {printf("%d  ", *p); p++;}
  printf("\n");
}

sort(int x[], int n)
{int i, j, k, t;
  for(i=0; i<n-1; i++)
    {k=i;
     for(j=i+1; j<n; j++)
       if(x[j]>x[k]) k=j;
     if(k!=i)
       {t=x[i]; x[i]=x[k]; x[k]=t;}
    }
}
```



说明：函数 sort 用数组名作为形参，也可改为用指针变量，这时函数的首部可以改为：sort(int *x, int n) 其他可一律不改。

10.3.4 指向多维数组的指针和指针变量

本小节以二维数组为例介绍多维数组的指针变量。

1. 多维数组的地址

设有整型二维数组 `a[3][4]` 如下：

```
0  1  2  3
4  5  6  7
8  9 10 11
```

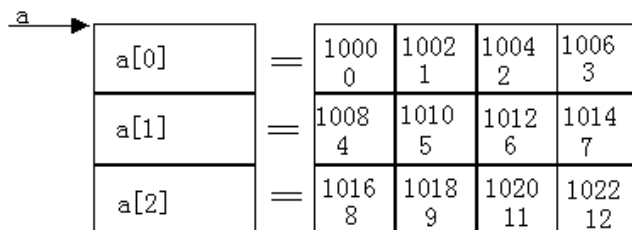
它的定义为：

```
int a[3][4]={{0, 1, 2, 3}, {4, 5, 6, 7}, {8, 9, 10, 11}}
```

设数组 `a` 的首地址为 1000，各下标变量的首地址及其值如图所示。

1000 0	1002 1	1004 2	1006 3
1008 4	1010 5	1012 6	1014 7
1016 8	1018 9	1020 11	1022 12

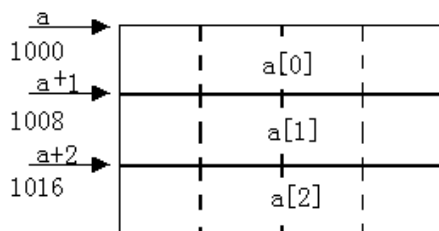
前面介绍过，C 语言允许把一个二维数组分解为多个一维数组来处理。因此数组 `a` 可分解为三个一维数组，即 `a[0]`，`a[1]`，`a[2]`。每一个一维数组又含有四个元素。



例如 `a[0]` 数组，含有 `a[0][0]`，`a[0][1]`，`a[0][2]`，`a[0][3]` 四个元素。

数组及数组元素的地址表示如下：

从二维数组的角度来看，`a` 是二维数组名，`a` 代表整个二维数组的首地址，也是二维数组 0 行的首地址，等于 1000。`a+1` 代表第一行的首地址，等于 1008。如图：



`a[0]` 是第一个一维数组的数组名和首地址，因此也为 1000。`*(a+0)` 或 `*a` 是与 `a[0]` 等效的，它表示一维数组 `a[0]` 0 号元素的首地址，也为 1000。`&a[0][0]` 是二维数组 `a` 的 0 行 0 列元素首地址，同样是 1000。因此，`a`，`a[0]`，`*(a+0)`，`*a`，`&a[0][0]` 是相等的。

同理，`a+1` 是二维数组 1 行的首地址，等于 1008。`a[1]` 是第二个一维数组的数组名和首

地址，因此也为 1008。 $\&a[1][0]$ 是二维数组 a 的 1 行 0 列元素地址，也是 1008。因此 $a+1, a[1], *(a+1), \&a[1][0]$ 是等同的。

由此可得出： $a+i, a[i], *(a+i), \&a[i][0]$ 是等同的。

此外， $\&a[i]$ 和 $a[i]$ 也是等同的。因为在二维数组中不能把 $\&a[i]$ 理解为元素 $a[i]$ 的地址，不存在元素 $a[i]$ 。C 语言规定，它是一种地址计算方法，表示数组 a 第 i 行首地址。由此，我们得出： $a[i], \&a[i], *(a+i)$ 和 $a+i$ 也都是等同的。

另外， $a[0]$ 也可以看成是 $a[0]+0$ ，是一维数组 $a[0]$ 的 0 号元素的首地址，而 $a[0]+1$ 则是 $a[0]$ 的 1 号元素首地址，由此可得出 $a[i]+j$ 则是一维数组 $a[i]$ 的 j 号元素首地址，它等于 $\&a[i][j]$ 。

	$a[0]$	$a[0]+1$	$a[0]+2$	$a[0]+3$
a	↓	↓	↓	↓
$a+1$	1000 0	1002 1	1004 2	1006 3
$a+2$	1008 4	1010 5	1012 6	1014 7
	1016 8	1018 9	1020 11	1022 12

由 $a[i]=*(a+i)$ 得 $a[i]+j=*(a+i)+j$ 。由于 $*(a+i)+j$ 是二维数组 a 的 i 行 j 列元素的首地址，所以，该元素的值等于 $*(*(a+i)+j)$ 。

【例 10.22】

```
main() {
    int a[3][4]={0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11};
    printf("%d, ", a);
    printf("%d, ", *a);
    printf("%d, ", a[0]);
    printf("%d, ", &a[0]);
    printf("%d\n", &a[0][0]);
    printf("%d, ", a+1);
    printf("%d, ", *(a+1));
    printf("%d, ", a[1]);
    printf("%d, ", &a[1]);
    printf("%d\n", &a[1][0]);
    printf("%d, ", a+2);
    printf("%d, ", *(a+2));
    printf("%d, ", a[2]);
    printf("%d, ", &a[2]);
    printf("%d\n", &a[2][0]);
    printf("%d, ", a[1]+1);
    printf("%d\n", *(a+1)+1);
    printf("%d, %d\n", *(a[1]+1), *((a+1)+1));
}
```



2. 指向多维数组的指针变量

把二维数组 a 分解为一维数组 $a[0], a[1], a[2]$ 之后, 设 p 为指向二维数组的指针变量。可定义为:

```
int (*p)[4]
```

它表示 p 是一个指针变量, 它指向包含 4 个元素的一维数组。若指向第一个一维数组 $a[0]$, 其值等于 $a, a[0]$, 或 $\&a[0][0]$ 等。而 $p+i$ 则指向一维数组 $a[i]$ 。从前面的分析可得出 $*(p+i)+j$ 是二维数组 i 行 j 列的元素的地址, 而 $*(*(p+i)+j)$ 则是 i 行 j 列元素的值。

二维数组指针变量说明的一般形式为:

类型说明符 (*指针变量名)[长度]

其中“类型说明符”为所指数组的数据类型。“*”表示其后的变量是指针类型。“长度”表示二维数组分解为多个一维数组时, 一维数组的长度, 也就是二维数组的列数。应注意“(*指针变量名)”两边的括号不可少, 如缺少括号则表示是指针数组(本章后面介绍), 意义就完全不同了。

【例 10.23】

```
main() {
    int a[3][4]={0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11};
    int (*p)[4];
    int i, j;
    p=a;
    for(i=0;i<3;i++)
        {for(j=0;j<4;j++) printf("%2d  ",*(*(p+i)+j));
          printf("\n");}
```



10.4 字符串的指针指向字符串的指针变量

10.4.1 字符串的表示形式

在 C 语言中, 可以用两种方法访问一个字符串。

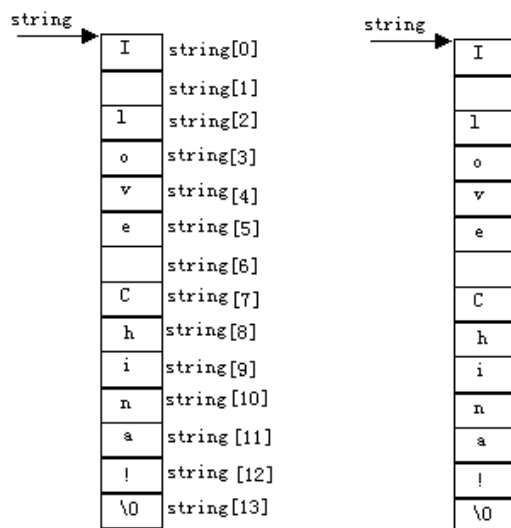
1) 用字符数组存放一个字符串, 然后输出该字符串。

【例 10.24】

```
main() {
    char string[]="I love China!";
    printf("%s\n", string);
}
```



说明: 和前面介绍的数组属性一样, `string` 是数组名, 它代表字符数组的首地址。



2) 用字符串指针指向一个字符串。

【例 10.25】

```
main() {
    char *string="I love China!";
    printf("%s\n", string);
}
```



字符串指针变量的定义说明与指向字符变量的指针变量说明是相同的。只能按对指针变量的赋值不同来区别。对指向字符变量的指针变量应赋予该字符变量的地址。

如：

```
char c, *p=&c;
```

表示 p 是一个指向字符变量 c 的指针变量。

而：

```
char *s="C Language";
```

则表示 s 是一个指向字符串的指针变量。把字符串的首地址赋予 s。

上例中，首先定义 string 是一个字符指针变量，然后把字符串的首地址赋予 string(应写出整个字符串，以便编译系统把该串装入连续的一块内存单元)，并把首地址送入 string。程序中的：

```
char *ps="C Language";
```

等效于：

```
char *ps;
```

```
ps="C Language";
```

【例 10.26】输出字符串中 n 个字符后的所有字符。

```
main() {
    char *ps="this is a book";
    int n=10;
    ps=ps+n;
```



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



运行结果为：

```
book
```

在程序中对 ps 初始化时，即把字符串首地址赋予 ps，当 ps=ps+10 之后，ps 指向字符“b”，因此输出为“book”。

【例 10.27】在输入的字符串中查找有无‘k’字符。

```
main() {
    char st[20], *ps;
    int i;
    printf("input a string:\n");
    ps=st;
    scanf("%s", ps);
    for(i=0; ps[i]!='\0'; i++)
        if(ps[i]=='k') {
            printf("there is a 'k' in the string\n");
            break;
        }
    if(ps[i]=='\0') printf("There is no 'k' in the string\n");
}
```



【例 10.28】本例是将指针变量指向一个格式字符串，用在 printf 函数中，用于输出二维数组的各种地址表示的值。但在 printf 语句中用指针变量 PF 代替了格式串。这也是程序中常用的方法。

```
main() {
    static int a[3][4]={0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11};
    char *PF;
    PF="%d, %d, %d, %d, %d\n";
    printf(PF, a, *a, a[0], &a[0], &a[0][0]);
    printf(PF, a+1, *(a+1), a[1], &a[1], &a[1][0]);
    printf(PF, a+2, *(a+2), a[2], &a[2], &a[2][0]);
    printf("%d, %d\n", a[1]+1, *(a+1)+1);
    printf("%d, %d\n", *(a[1]+1), (*(a+1)+1));
}
```



【例 10.29】本例是把字符串指针作为函数参数的使用。要求把一个字符串的内容复制到另一个字符串中，并且不能使用 strcpy 函数。函数 cprstr 的形参为两个字符指针变量。pss 指向源字符串，pds 指向目标字符串。注意表达式：(*pds=*pss)!='\0' 的用法。

```
cpystr(char *pss, char *pds) {
    while((*pds=*pss)!='\0') {
```

```

        pds++;
        pss++; }
    }
main() {
    char *pa="CHINA", b[10], *pb;
    pb=b;
    cpystr(pa, pb);
    printf("string a=%s\nstring b=%s\n", pa, pb);
}

```



在本例中，程序完成了两项工作：一是把 pss 指向的源字符串复制到 pds 所指向的目标字符串中，二是判断所复制的字符是否为 '\0'，若是则表明源字符串结束，不再循环。否则，pds 和 pss 都加 1，指向下一字符。在主函数中，以指针变量 pa, pb 为实参，分别取得确定值后调用 cprstr 函数。由于采用的指针变量 pa 和 pss, pb 和 pds 均指向同一字符串，因此在主函数和 cprstr 函数中均可使用这些字符串。也可以把 cprstr 函数简化为以下形式：

```

cprstr(char *pss, char *pds)
{while ((*pds++=*pss++)!='\0');}

```

即把指针的移动和赋值合并在一个语句中。进一步分析还可发现 '\0' 的 ASCII 码为 0，对于 while 语句只看表达式的值为非 0 就循环，为 0 则结束循环，因此也可省去 "\0" 这一判断部分，而写为以下形式：

```

cprstr(char *pss, char *pds)
{while (*pds++=*pss++);}

```

表达式的意义可解释为，源字符串向目标字符串赋值，移动指针，若所赋值为非 0 则循环，否则结束循环。这样使程序更加简洁。

【例 10.30】简化后的程序如下所示。

```

cpystr(char *pss, char *pds) {
    while(*pds++=*pss++);
}
main() {
    char *pa="CHINA", b[10], *pb;
    pb=b;
    cpystr(pa, pb);
    printf("string a=%s\nstring b=%s\n", pa, pb);
}

```



10.4.2 使用字符串指针变量与字符数组的区别

用字符数组和字符指针变量都可实现字符串的存储和运算。但是两者是有区别的。在使用时应注意以下几个问题：

1. 字符串指针变量本身是一个变量，用于存放字符串的首地址。而字符串本身是存放在以该首地址为首的一块连续的内存空间中并以 '\0' 作为串的结束。字符数组是

由于若干个数组元素组成的，它可用来存放整个字符串。

2. 对字符串指针方式

```
char *ps="C Language";
```

可以写为：

```
char *ps;
ps="C Language";
```

而对数组方式：

```
static char st[]={ "C Language" };
```

不能写为：

```
char st[20];
st={ "C Language" };
```

而只能对字符数组的各元素逐个赋值。

从以上几点可以看出字符串指针变量与字符数组在使用时的区别，同时也可看出使用指针变量更加方便。

前面说过，当一个指针变量在未取得确定地址前使用是危险的，容易引起错误。但是对指针变量直接赋值是可以的。因为 C 系统对指针变量赋值时要给以确定的地址。

因此，

```
char *ps="C Langage";
```

或者

```
char *ps;
ps="C Language";
```

都是合法的。

10.5 函数指针变量

在 C 语言中，一个函数总是占用一段连续的内存区，而函数名就是该函数所占内存区的首地址。我们可以把函数的这个首地址(或称入口地址)赋予一个指针变量，使该指针变量指向该函数。然后通过指针变量就可以找到并调用这个函数。我们把这种指向函数的指针变量称为“函数指针变量”。

函数指针变量定义的一般形式为：

```
类型说明符 (*指针变量名)();
```

其中“类型说明符”表示被指函数的返回值的类型。“(* 指针变量名)”表示“*”后面的变量是定义的指针变量。最后的空括号表示指针变量所指的是一个函数。

例如：

```
int (*pf)();
```

表示 pf 是一个指向函数入口的指针变量，该函数的返回值(函数值)是整型。

【例 10.31】本例用来说明用指针形式实现对函数调用的方法。

```
int max(int a, int b) {
    if(a>b) return a;
    else return b;
}

main() {
    int max(int a, int b);
    int (*pmax)();
```

```

int x, y, z;
pmax=max;
printf("input two numbers:\n");
scanf("%d%d", &x, &y);
z=(*pmax)(x, y);
printf("maxumum=%d", z);
}

```



从上述程序可以看出，函数指针变量形式调用函数的步骤如下：

- 1) 先定义函数指针变量，如后一程序中第 9 行 `int (*pmax)();` 定义 `pmax` 为函数指针变量。
- 2) 把被调函数的入口地址(函数名)赋予该函数指针变量，如程序中第 11 行 `pmax=max;`
- 3) 用函数指针变量形式调用函数，如程序第 14 行 `z=(*pmax)(x, y);`
- 4) 调用函数的一般形式为：

(*指针变量名) (实参表)

使用函数指针变量还应注意以下两点：

- a) 函数指针变量不能进行算术运算，这是与数组指针变量不同的。数组指针变量加减一个整数可使指针移动指向后面或前面的数组元素，而函数指针的移动是毫无意义的。
- b) 函数调用中“(*指针变量名)”的两边的括号不可少，其中的*不应该理解为求值运算，在此处它只是一种表示符号。

10.6 指针型函数

前面我们介绍过，所谓函数类型是指函数返回值的类型。在 C 语言中允许一个函数的返回值是一个指针(即地址)，这种返回指针值的函数称为指针型函数。

定义指针型函数的一般形式为：

```

类型说明符 *函数名(形参表)
{
    ..... /*函数体*/
}

```

其中函数名之前加了“*”号表明这是一个指针型函数，即返回值是一个指针。类型说明符表示了返回的指针值所指向的数据类型。

如：

```

int *ap(int x, int y)
{
    ..... /*函数体*/
}

```

表示 `ap` 是一个返回指针值的指针型函数，它返回的指针指向一个整型变量。

【例 10.32】本程序是通过指针函数，输入一个 1~7 之间的整数，输出对应的星期名。

```

main() {
    int i;
    char *day_name(int n);
    printf("input Day No:\n");
}

```

```

scanf("%d",&i);
if(i<0) exit(1);
printf("Day No:%2d-->%s\n",i,day_name(i));
}
char *day_name(int n){
    static char *name[]={ "Illegal day",
                           "Monday",
                           "Tuesday",
                           "Wednesday",
                           "Thursday",
                           "Friday",
                           "Saturday",
                           "Sunday"};
    return((n<1||n>7) ? name[0] : name[n]);
}

```



本例中定义了一个指针型函数 `day_name`，它的返回值指向一个字符串。该函数中定义了一个静态指针数组 `name`。`name` 数组初始化赋值为八个字符串，分别表示各个星期名及出错提示。形参 `n` 表示与星期名所对应的整数。在主函数中，把输入的整数 `i` 作为实参，在 `printf` 语句中调用 `day_name` 函数并把 `i` 值传送给形参 `n`。`day_name` 函数中的 `return` 语句包含一个条件表达式，`n` 值若大于 7 或小于 1 则把 `name[0]` 指针返回主函数输出出错提示字符串“Illegal day”。否则返回主函数输出对应的星期名。主函数中的第 7 行是个条件语句，其语义是，如输入为负数 (`i<0`) 则中止程序运行退出程序。`exit` 是一个库函数，`exit(1)` 表示发生错误后退出程序，`exit(0)` 表示正常退出。

应该特别注意的是函数指针变量和指针型函数这两者在写法和意义上的区别。如 `int (*p)()` 和 `int *p()` 是两个完全不同的量。

`int (*p)()` 是一个变量说明，说明 `p` 是一个指向函数入口的指针变量，该函数的返回值是整型量，`(*p)` 的两边的括号不能少。

`int *p()` 则不是变量说明而是函数说明，说明 `p` 是一个指针型函数，其返回值是一个指向整型量的指针，`*p` 两边没有括号。作为函数说明，在括号内最好写入形式参数，这样便于与变量说明区别。

对于指针型函数定义，`int *p()` 只是函数头部分，一般还应该有函数体部分。

10.7 指针数组和指向指针的指针

10.7.1 指针数组的概念

一个数组的元素值为指针则是指针数组。指针数组是一组有序的指针的集合。指针数组的所有元素都必须是具有相同存储类型和指向相同数据类型的指针变量。

指针数组说明的一般形式为：

类型说明符 *数组名[数组长度]

其中类型说明符为指针值所指向的变量的类型。

例如：

```
int *pa[3]
```

表示 pa 是一个指针数组，它有三个数组元素，每个元素值都是一个指针，指向整型变量。

【例 10.33】通常可用一个指针数组来指向一个二维数组。指针数组中的每个元素被赋予二维数组每一行的首地址，因此也可理解为指向一个一维数组。

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



本例程序中，pa 是一个指针数组，三个元素分别指向二维数组 a 的各行。然后用循环语句输出指定的数组元素。其中 *a[i] 表示 i 行 0 列元素值；*(*(a+i)+i) 表示 i 行 i 列的元素值；*pa[i] 表示 i 行 0 列元素值；由于 p 与 a[0] 相同，故 p[i] 表示 0 行 i 列的值；*(p+i) 表示 0 行 i 列的值。读者可仔细领会元素值的各种不同的表示方法。

应该注意指针数组和二维数组指针变量的区别。这两者虽然都可用来表示二维数组，但是其表示方法和意义是不同的。

二维数组指针变量是单个的变量，其一般形式中“(*指针变量名)”两边的括号不可少。而指针数组类型表示的是多个指针（一组有序指针）在一般形式中“*指针数组名”两边不能有括号。

例如：

```
int (*p)[3];
```

表示一个指向二维数组的指针变量。该二维数组的列数为 3 或分解为一维数组的长度为 3。

```
int *p[3]
```

表示 p 是一个指针数组，有三个下标变量 p[0]，p[1]，p[2] 均为指针变量。

指针数组也常用来表示一组字符串，这时指针数组的每个元素被赋予一个字符串的首地址。指向字符串的指针数组的初始化更为简单。例如在例 10.32 中即采用指针数组来表示一组字符串。其初始化赋值为：

```
char *name[]={ "Illegal day",
                "Monday",
                "Tuesday",
                "Wednesday",
                "Thursday",
                "Friday",
                "Saturday",
                "Sunday"};
```

完成这个初始化赋值之后，name[0] 即指向字符串 "Illegal day"，name[1] 指向

"Monday".....。

指针数组也可以用作函数参数。

【例 10.34】指针数组作指针型函数的参数。在本例主函数中，定义了一个指针数组 name，并对 name 作了初始化赋值。其每个元素都指向一个字符串。然后又以 name 作为实参调用指针型函数 day_name，在调用时把数组名 name 赋予形参变量 name，输入的整数 i 作为第二个实参赋予形参 n。在 day_name 函数中定义了两个指针变量 pp1 和 pp2，pp1 被赋予 name[0] 的值(即*name)，pp2 被赋予 name[n]的值即*(name+n)。由条件表达式决定返回 pp1 或 pp2 指针给主函数中的指针变量 ps。最后输出 i 和 ps 的值。

```
main() {
    static char *name[]={ "Illegal day",
                           "Monday",
                           "Tuesday",
                           "Wednesday",
                           "Thursday",
                           "Friday",
                           "Saturday",
                           "Sunday"};

    char *ps;
    int i;
    char *day_name(char *name[], int n);
    printf("input Day No:\n");
    scanf("%d",&i);
    if(i<0) exit(1);
    ps=day_name(name, i);
    printf("Day No:%2d-->%s\n", i, ps);
}

char *day_name(char *name[], int n)
{
    char *pp1,*pp2;
    pp1=name;
    pp2=*(name+n);
    return((n<1||n>7)? pp1:pp2);
}
```



【例 10.35】输入 5 个国名并按字母顺序排列后输出。现编程如下：

```
#include "string.h"
main() {
    void sort(char *name[], int n);
    void print(char *name[], int n);
    static char *name[]={ "CHINA", "AMERICA", "AUSTRALIA",
                           "FRANCE", "GERMAN"};

    int n=5;
    sort(name, n);
```

```
    print(name,n);
}
void sort(char *name[],int n){
    char *pt;
    int i,j,k;
    for(i=0;i<n-1;i++){
        k=i;
        for(j=i+1;j<n;j++){
            if(strcmp(name[k],name[j])>0) k=j;
        }
        if(k!=i){
            pt=name[i];
            name[i]=name[k];
            name[k]=pt;
        }
    }
}
void print(char *name[],int n){
    int i;
    for (i=0;i<n;i++) printf("%s\n",name[i]);
}
```



说明:

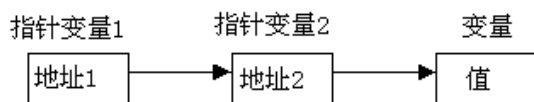
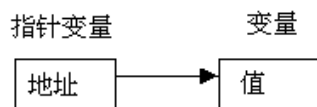
在以前的例子中采用了普通的排序方法，逐个比较之后交换字符串的位置。交换字符串的物理位置是通过字符串复制函数完成的。反复的交换将使程序执行的速度很慢，同时由于各字符串(国名)的长度不同，又增加了存储管理的负担。用指针数组能很好地解决这些问题。把所有的字符串存放在一个数组中，把这些字符串的首地址放在一个指针数组中，当需要交换两个字符串时，只须交换指针数组相应两元素的内容(地址)即可，而不必交换字符串本身。

本程序定义了两个函数，一个名为 sort 完成排序，其形参为指针数组 name，即为待排序的各字符串数组的指针。形参 n 为字符串的个数。另一个函数名为 print，用于排序后字符串的输出，其形参与 sort 的形参相同。主函数 main 中，定义了指针数组 name 并作了初始化赋值。然后分别调用 sort 函数和 print 函数完成排序和输出。值得说明的是在 sort 函数中，对两个字符串比较，采用了 strcmp 函数，strcmp 函数允许参与比较的字符串以指针方式出现。name[k] 和 name[j] 均为指针，因此是合法的。字符串比较后需要交换时，只交换指针数组元素的值，而不交换具体的字符串，这样将大大减少时间的开销，提高了运行效率。

10.7.2 指向指针的指针

如果一个指针变量存放的又是另一个指针变量的地址，则称这个指针变量为指向指针的指针变量。

在前面已经介绍过，通过指针访问变量称为间接访问。由于指针变量直接指向变量，所以称为“单级间址”。而如果通过指向指针的指针变量来访问变量则构成“二级间址”。



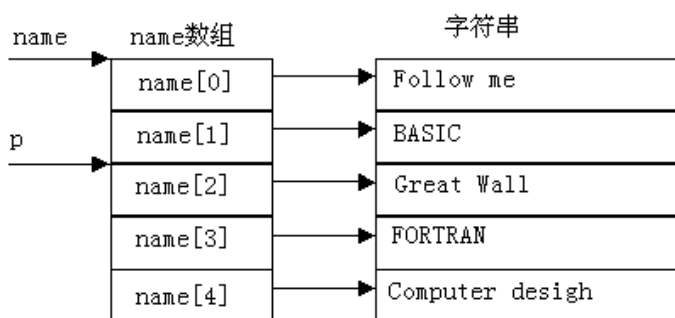
从下图可以看到，name 是一个指针数组，它的每一个元素是一个指针型数据，其值为地址。Name 是一个数据，它的每一个元素都有相应的地址。数组名 name 代表该指针数组的首地址。name+1 是 mane[i] 的地址。name+1 就是指向指针型数据的指针（地址）。还可以设置一个指针变量 p，使它指向指针数组元素。P 就是指向指针型数据的指针变量。

怎样定义一个指向指针型数据的指针变量呢？如下：

```
char **p;
```

p 前面有两个*号，相当于*(*p)。显然*p 是指针变量的定义形式，如果没有最前面的*，那就是定义了一个指向字符数据的指针变量。现在它前面又有一个*号，表示指针变量 p 是指向一个字符指针型变量的。*p 就是 p 所指向的另一个指针变量。

从下图可以看到，name 是一个指针数组，它的每一个元素是一个指针型数据，其值为地址。name 是一个数组，它的每一个元素都有相应的地址。数组名 name 代表该指针数组的首地址。name+1 是 mane[i] 的地址。name+1 就是指向指针型数据的指针（地址）。还可以设置一个指针变量 p，使它指向指针数组元素。P 就是指向指针型数据的指针变量。



如果有：

```
p=name+2;
printf("%o\n", *p);
printf("%s\n", *p);
```

则，第一个 printf 函数语句输出 name[2] 的值（它是一个地址），第二个 printf 函数语句以字符串形式（%s）输出字符串“Great Wall”。

【例 10.36】使用指向指针的指针。

```
main()
{char *name[]={"Follow me", "BASIC", "Great Wall", "FORTRAN", "Computer designh"};
  char **p;
  int i;
  for(i=0; i<5; i++)
  {p=name+i;
```

```

    printf("%s\n", *p);
}
}

```



说明:

p 是指向指针的指针变量。

【例 10.37】一个指针数组的元素指向数据的简单例子。

```

main()
{static int a[5]={1, 3, 5, 7, 9};
  int *num[5]={&a[0], &a[1], &a[2], &a[3], &a[4]};
  int **p, i;
  p=num;
  for(i=0; i<5; i++)
    {printf("%d\t", **p); p++;}
}

```



说明:

指针数组的元素只能存放地址。

10.7.3 main 函数的参数

前面介绍的 main 函数都是不带参数的。因此 main 后的括号都是空括号。实际上, main 函数可以带参数, 这个参数可以认为是 main 函数的形式参数。C 语言规定 main 函数的参数只能有两个, 习惯上这两个参数写为 argc 和 argv。因此, main 函数的函数头可写为:

```
main (argc, argv)
```

C 语言还规定 argc(第一个形参)必须是整型变量, argv(第二个形参)必须是指向字符串的指针数组。加上形参说明后, main 函数的函数头应写为:

```
main (int argc, char *argv[])
```

由于 main 函数不能被其它函数调用, 因此不可能在程序内部取得实际值。那么, 在何处把实参值赋予 main 函数的形参呢? 实际上, main 函数的参数值是从操作系统命令行上获得的。当我们要运行一个可执行文件时, 在 DOS 提示符下键入文件名, 再输入实际参数即可把这些实参传送到 main 的形参中去。

DOS 提示符下命令行的一般形式为:

```
C:\>可执行文件名 参数 参数.....;
```

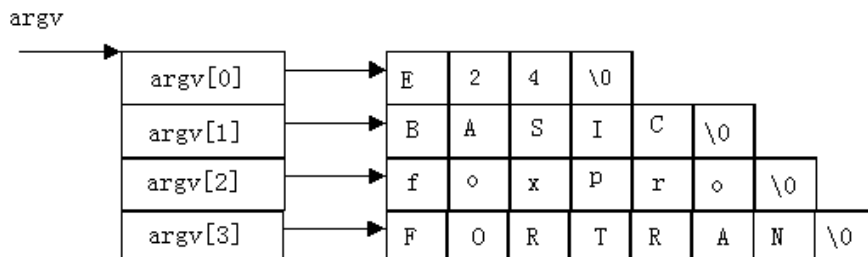
但是应该特别注意的是, main 的两个形参和命令行中的参数在位置上不是一一对应的。因为, main 的形参只有二个, 而命令行中的参数个数原则上未加限制。argc 参数表示了命令行中参数的个数(注意: 文件名本身也算一个参数), argv 的值是在输入命令行时由系统按实际参数的个数自动赋予的。

例如有命令行为:

```
C:\>E24 BASIC foxpro FORTRAN
```

由于文件名 E24 本身也算一个参数, 所以共有 4 个参数, 因此 argc 取得的值为 4。argv 参数是字符串指针数组, 其各元素值为命令行中各字符串(参数均按字符串处理)的首地址。指

针数组的长度即为参数个数。数组元素初值由系统自动赋予。其表示如图所示：



【例 10.38】

```
main(int argc, char *argv) {
    while(argc-->1)
        printf("%s\n", *++argv);
}
```



本例是显示命令行中输入的参数。如果上例的可执行文件名为 e24.exe，存放在 A 驱动器的盘内。因此输入的命令行为：

C:\>a:e24 BASIC foxpro FORTRAN

则运行结果为：

```
BASIC
foxpro
FORTRAN
```

该行共有 4 个参数，执行 main 时，argc 的初值即为 4。argv 的 4 个元素分为 4 个字符串的首地址。执行 while 语句，每循环一次 argv 值减 1，当 argv 等于 1 时停止循环，共循环三次，因此共可输出三个参数。在 printf 函数中，由于打印项 *++argv 是先加 1 再打印，故第一次打印的是 argv[1] 所指的字符串 BASIC。第二、三次循环分别打印后二个字符串。而参数 e24 是文件名，不必输出。

10.8 有关指针的数据类型和指针运算的小结

10.8.1 有关指针的数据类型的小结

定义	含 义
int i;	定义整型变量 i
int *p	p 为指向整型数据的指针变量
int a[n];	定义整型数组 a，它有 n 个元素
int *p[n];	定义指针数组 p，它由 n 个指向整型数据的指针元素组成
int (*p)[n];	p 为指向含 n 个元素的一维数组的指针变量
int f();	f 为带回整型函数值的函数
int *p();	p 为带回一个指针的函数，该指针指向整型数据
int (*p)();	p 为指向函数的指针，该函数返回一个整型值

```
int **p;
```

P 是一个指针变量，它指向一个指向整型数据的指针变量

10.8.2 指针运算的小结

现把全部指针运算列出如下：

- 1) 指针变量加（减）一个整数：

例如：p++、p--、p+i、p-i、p+=i、p-=i

一个指针变量加（减）一个整数并不是简单地将原值加（减）一个整数，而是将该指针变量的原值（是一个地址）和它指向的变量所占用的内存单元字节数加（减）。

- 2) 指针变量赋值：将一个变量的地址赋给一个指针变量。

p=&a; (将变量 a 的地址赋给 p)

p=array; (将数组 array 的首地址赋给 p)

p=&array[i]; (将数组 array 第 i 个元素的地址赋给 p)

p=max; (max 为已定义的函数，将 max 的入口地址赋给 p)

p1=p2; (p1 和 p2 都是指针变量，将 p2 的值赋给 p1)

注意：不能如下：

p=1000;

- 3) 指针变量可以有空值，即该指针变量不指向任何变量：

p=NULL;

- 4) 两个指针变量可以相减：如果两个指针变量指向同一个数组的元素，则两个指针变量值之差是两个指针之间的元素个数。

- 5) 两个指针变量比较：如果两个指针变量指向同一个数组的元素，则两个指针变量可以进行比较。指向前面的元素的指针变量“小于”指向后面的元素的指针变量。

10.8.3 void 指针类型

ANSI 新标准增加了一种“void”指针类型，即可以定义一个指针变量，但不指定它是指向哪一种类型数据。

11	结构体与共用体.....	1
11.1	定义一个结构的一般形式.....	1
11.2	结构类型变量的说明.....	2
11.3	结构变量成员的表示方法.....	4
11.4	结构变量的赋值.....	4
11.5	结构变量的初始化.....	5
11.6	结构数组的定义.....	5
11.7	结构指针变量的说明和使用.....	7
11.7.1	指向结构变量的指针.....	7
11.7.2	指向结构数组的指针.....	9
11.7.3	结构指针变量作函数参数.....	10
11.8	动态存储分配.....	11
11.9	链表的概念.....	12
11.10	枚举类型.....	14
11.10.1	枚举类型的定义和枚举变量的说明.....	14
11.10.2	枚举类型变量的赋值和使用.....	15
11.11	类型定义符 typedef.....	16

11 结构体与共用体

11.1 定义一个结构的一般形式

在实际问题中，一组数据往往具有不同的数据类型。例如，在学生登记表中，姓名应为字符型；学号可为整型或字符型；年龄应为整型；性别应为字符型；成绩可为整型或实型。显然不能用一个数组来存放这一组数据。因为数组中各元素的类型和长度都必须一致，以便于编译系统处理。为了解决这个问题，C 语言中给出了另一种构造数据类型——“结构（structure）”或叫“结构体”。它相当于其它高级语言中的记录。“结构”是一种构造类型，它是由若干“成员”组成的。每一个成员可以是一个基本数据类型或者又是一个构造类型。结构既是一种“构造”而成的数据类型，那么在说明和使用之前必须先定义它，也就是构造它。如同在说明和调用函数之前要先定义函数一样。

定义一个结构的一般形式为：

```
struct 结构名
{成员表列};
```

成员表列由若干个成员组成，每个成员都是该结构的一个组成部分。对每个成员也必须作类型说明，其形式为：

```
类型说明符 成员名;
```

成员名的命名应符合标识符的书写规定。例如：

```
struct stu
{
    int num;
    char name[20];
    char sex;
```

```
float score;  
};
```

在这个结构定义中，结构名为 `stu`，该结构由 4 个成员组成。第一个成员为 `num`，整型变量；第二个成员为 `name`，字符数组；第三个成员为 `sex`，字符变量；第四个成员为 `score`，实型变量。应注意在括号后的分号是不可少的。结构定义之后，即可进行变量说明。凡说明为结构 `stu` 的变量都由上述 4 个成员组成。由此可见，结构是一种复杂的数据类型，是数目固定，类型不同的若干有序变量的集合。

11.2 结构类型变量的说明

说明结构变量有以下三种方法。以上面定义的 `stu` 为例来加以说明。

1. 先定义结构，再说明结构变量。

如：

```
struct stu  
{  
    int num;  
    char name[20];  
    char sex;  
    float score;  
};  
struct stu boy1, boy2;
```

说明了两个变量 `boy1` 和 `boy2` 为 `stu` 结构类型。也可以用宏定义使一个符号常量来表示一个结构类型。

例如：

```
#define STU struct stu  
STU  
{  
    int num;  
    char name[20];  
    char sex;  
    float score;  
};  
STU boy1, boy2;
```

2. 在定义结构类型的同时说明结构变量。

例如：

```
struct stu  
{  
    int num;  
    char name[20];  
    char sex;  
    float score;  
} boy1, boy2;
```

这种形式的说明的一般形式为：

struct 结构名

```
{
    成员表列
} 变量名表列;
```

3. 直接说明结构变量。

例如：

```
struct
{
    int num;
    char name[20];
    char sex;
    float score;
} boy1, boy2;
```

这种形式的说明的一般形式为：

```
struct
{
    成员表列
} 变量名表列;
```

第三种方法与第二种方法的区别在于第三种方法中省去了结构名，而直接给出结构变量。三种方法中说明的 boy1, boy2 变量都具有下图所示的结构。

num	name	sex	score

说明了 boy1, boy2 变量为 stu 类型后，即可向这两个变量中的各个成员赋值。在上述 stu 结构定义中，所有的成员都是基本数据类型或数组类型。

成员也可以又是一个结构，即构成了嵌套的结构。例如，下图给出了另一个数据结构。

num	name	sex	birthday			score
			month	day	year	

按图可给出以下结构定义：

```
struct date
{
    int month;
    int day;
    int year;
};
struct{
    int num;
    char name[20];
    char sex;
    struct date birthday;
    float score;
} boy1, boy2;
```

首先定义一个结构 date，由 month(月)、day(日)、year(年) 三个成员组成。在定义

并说明变量 boy1 和 boy2 时，其中的成员 birthday 被说明为 data 结构类型。成员名可与程序中其它变量同名，互不干扰。

11.3 结构变量成员表示方法

在程序中使用结构变量时，往往不把它作为一个整体来使用。在 ANSI C 中除了允许具有相同类型的结构变量相互赋值以外，一般对结构变量的使用，包括赋值、输入、输出、运算等都是通过结构变量的成员来实现的。

表示结构变量成员的一般形式是：

结构变量名.成员名

例如：

boy1.num 即第一个人的学号

boy2.sex 即第二个人的性别

如果成员本身又是一个结构则必须逐级找到最低级的成员才能使用。

例如：

boy1.birthday.month

即第一个人出生的月份成员可以在程序中单独使用，与普通变量完全相同。

11.4 结构变量的赋值

结构变量的赋值就是给各成员赋值。可用输入语句或赋值语句来完成。

【例 11.1】给结构变量赋值并输出其值。

```
main()
{
    struct stu
    {
        int num;
        char *name;
        char sex;
        float score;
    } boy1, boy2;
    boy1.num=102;
    boy1.name="Zhang ping";
    printf("input sex and score\n");
    scanf("%c %f",&boy1.sex,&boy1.score);
    boy2=boy1;
    printf("Number=%d\nName=%s\n", boy2.num, boy2.name);
    printf("Sex=%c\nScore=%f\n", boy2.sex, boy2.score);
}
```



本程序中用赋值语句给 num 和 name 两个成员赋值，name 是一个字符串指针变量。用 scanf 函数动态地输入 sex 和 score 成员值，然后把 boy1 的所有成员的值整体赋予 boy2。

最后分别输出 boy2 的各个成员值。本例表示了结构变量的赋值、输入和输出的方法。

11.5 结构变量的初始化

和其他类型变量一样，对结构变量可以在定义时进行初始化赋值。

【例 11.2】对结构变量初始化。

```
main()
{
    struct stu    /*定义结构*/
    {
        int num;
        char *name;
        char sex;
        float score;
    } boy2, boy1={102, "Zhang ping", 'M', 78.5};
    boy2=boy1;
    printf("Number=%d\nName=%s\n", boy2.num, boy2.name);
    printf("Sex=%c\nScore=%f\n", boy2.sex, boy2.score);
}
```



本例中，boy2, boy1 均被定义为外部结构变量，并对 boy1 作了初始化赋值。在 main 函数中，把 boy1 的值整体赋予 boy2，然后用两个 printf 语句输出 boy2 各成员的值。

11.6 结构数组的定义

数组的元素也可以是结构类型的。因此可以构成结构型数组。结构数组的每一个元素都是具有相同结构类型的下标结构变量。在实际应用中，经常用结构数组来表示具有相同数据结构的一个群体。如一个班的学生档案，一个车间职工的工资表等。

方法和结构变量相似，只需说明它为数组类型即可。

例如：

```
struct stu
{
    int num;
    char *name;
    char sex;
    float score;
} boy[5];
```

定义了一个结构数组 boy，共有 5 个元素，boy[0]~boy[4]。每个数组元素都具有 struct stu 的结构形式。对结构数组可以作初始化赋值。

例如：

```
struct stu
{
```

```

    int num;
    char *name;
    char sex;
    float score;
}boy[5]={
    {101,"Li ping","M",45},
    {102,"Zhang ping","M",62.5},
    {103,"He fang","F",92.5},
    {104,"Cheng ling","F",87},
    {105,"Wang ming","M",58};
}

```

当对全部元素作初始化赋值时，也可不给出数组长度。

【例 11.3】计算学生的平均成绩和不及格的人数。

```

struct stu
{
    int num;
    char *name;
    char sex;
    float score;
}boy[5]={
    {101,"Li ping",'M',45},
    {102,"Zhang ping",'M',62.5},
    {103,"He fang",'F',92.5},
    {104,"Cheng ling",'F',87},
    {105,"Wang ming",'M',58},
};

main()
{
    int i,c=0;
    float ave,s=0;
    for(i=0;i<5;i++)
    {
        s+=boy[i].score;
        if(boy[i].score<60) c+=1;
    }
    printf("s=%f\n",s);
    ave=s/5;
    printf("average=%f\ncount=%d\n",ave,c);
}

```



本例程序中定义了一个外部结构数组 boy，共 5 个元素，并作了初始化赋值。在 main 函数中用 for 语句逐个累加各元素的 score 成员值存于 s 之中，如 score 的值小于 60 (不及格) 即计数器 C 加 1，循环完毕后计算平均成绩，并输出全班总分，平均分及不及格人数。

11.7 结构指针变量的说明和使用

11.7.1 指向结构变量的指针

当然也可在定义 `stu` 结构时同时说明 `psu`。与前面讨论的各类指针变量相同，结构指针变量也必须要先赋值后才能使用。

赋值是把结构变量的首地址赋予该指针变量，不能把结构名赋予该指针变量。如果 boy 是被说明为 stu 类型的结构变量，则：

```
pstu=&boy
```

是正确的，而：

```
pstu=&stu
```

是错误的。

结构名和结构变量是两个不同的概念，不能混淆。结构名只能表示一个结构形式，编译系统并不对它分配内存空间。只有当某变量被说明为这种类型的结构时，才对该变量分配存储空间。因此上面 &stu 这种写法是错误的，不可能去取一个结构名的首地址。有了结构指针变量，就能更方便地访问结构变量的各个成员。

其访问的一般形式为：

(*结构指针变量). 成员名

或为：

结构指针变量->成员名

例如：

```
(*pstu). num
```

或者：

```
pstu->num
```

应该注意 (*pstu) 两侧的括号不可少，因为成员符“.”的优先级高于“*”。如去掉括号写作 *pstu.num 则等效于 *(pstu.num)，这样，意义就完全不对了。

下面通过例子来说明结构指针变量的具体说明和使用方法。

【例 11.5】

```
struct stu
{
    int num;
    char *name;
    char sex;
    float score;
} boy1={102,"Zhang ping",'M',78.5},*pstu;

main()
{
    pstu=&boy1;
    printf("Number=%d\nName=%s\n",boy1.num,boy1.name);
    printf("Sex=%c\nScore=%f\n\n",boy1.sex,boy1.score);
    printf("Number=%d\nName=%s\n",(*pstu).num,(*pstu).name);
    printf("Sex=%c\nScore=%f\n\n",(*pstu).sex,(*pstu).score);
    printf("Number=%d\nName=%s\n",pstu->num,pstu->name);
    printf("Sex=%c\nScore=%f\n\n",pstu->sex,pstu->score);
}
```



本例程序定义了一个结构 stu，定义了 stu 类型结构变量 boy1 并作了初始化赋值，还定义了一个指向 stu 类型结构的指针变量 pstu。在 main 函数中，pstu 被赋予 boy1 的地址，因此 pstu 指向 boy1。然后在 printf 语句内用三种形式输出 boy1 的各个成员值。从运行结

果可以看出：

结构变量. 成员名

(*结构指针变量). 成员名

结构指针变量->成员名

这三种用于表示结构成员的形式是完全等效的。

11.7.2 指向结构数组的指针

指针变量可以指向一个结构数组，这时结构指针变量的值是整个结构数组的首地址。结构指针变量也可指向结构数组的一个元素，这时结构指针变量的值是该结构数组元素的首地址。

设 ps 为指向结构数组的指针变量，则 ps 也指向该结构数组的 0 号元素，ps+1 指向 1 号元素，ps+i 则指向 i 号元素。这与普通数组的情况是一致的。

【例 11.6】用指针变量输出结构数组。

```
struct stu
{
    int num;
    char *name;
    char sex;
    float score;
}boy[5]={
    {101,"Zhou ping",'M',45},
    {102,"Zhang ping",'M',62.5},
    {103,"Liou fang",'F',92.5},
    {104,"Cheng ling",'F',87},
    {105,"Wang ming",'M',58},
};

main()
{
    struct stu *ps;
    printf("No\tName\t\t\tSex\tScore\t\n");
    for(ps=boy;ps<boy+5;ps++)
        printf("%d\t%s\t\t%c\t%f\t\n",ps->num,ps->name,ps->sex,ps->score);
}
```



在程序中，定义了 stu 结构类型的外部数组 boy 并作了初始化赋值。在 main 函数内定义 ps 为指向 stu 类型的指针。在循环语句 for 的表达式 1 中，ps 被赋予 boy 的首地址，然后循环 5 次，输出 boy 数组中各成员值。

应该注意的是，一个结构指针变量虽然可以用来访问结构变量或结构数组元素的成员，但是，不能使它指向一个成员。也就是说不允许取一个成员的地址来赋予它。因此，下面的赋值是错误的。

```
ps=&boy[1].sex;
```

而只能是：

```
ps=boy; (赋予数组首地址)
```

或者是：

```
ps=&boy[0]; (赋予 0 号元素首地址)
```

11.7.3 结构指针变量作函数参数

在 ANSI C 标准中允许用结构变量作函数参数进行整体传送。但是这种传送要将全部成员逐个传送，特别是成员为数组时将会使传送的时间和空间开销很大，严重地降低了程序的效率。因此最好的办法就是使用指针，即用指针变量作函数参数进行传送。这时由实参传向形参的只是地址，从而减少了时间和空间的开销。

【例 11.7】计算一组学生的平均成绩和不及格人数。用结构指针变量作函数参数编程。

```
struct stu
{
    int num;
    char *name;
    char sex;
    float score;}boy[5]={
        {101,"Li ping",'M',45},
        {102,"Zhang ping",'M',62.5},
        {103,"He fang",'F',92.5},
        {104,"Cheng ling",'F',87},
        {105,"Wang ming",'M',58},
    };
main()
{
    struct stu *ps;
    void ave(struct stu *ps);
    ps=boy;
    ave(ps);
}
void ave(struct stu *ps)
{
    int c=0,i;
    float ave,s=0;
    for(i=0;i<5;i++,ps++)
    {
        s+=ps->score;
        if(ps->score<60) c+=1;
    }
    printf("s=%f\n",s);
    ave=s/5;
    printf("average=%f\ncount=%d\n",ave,c);
}
```



本程序中定义了函数 ave，其形参为结构指针变量 ps。boy 被定义为外部结构数组，因此在整个源程序中有效。在 main 函数中定义说明了结构指针变量 ps，并把 boy 的首地址赋予它，使 ps 指向 boy 数组。然后以 ps 作实参调用函数 ave。在函数 ave 中完成计算平均成绩和统计不及格人数的工作并输出结果。

由于本程序全部采用指针变量作运算和处理，故速度更快，程序效率更高。

11.8 动态存储分配

在数组一章中，曾介绍过数组的长度是预先定义好的，在整个程序中固定不变。C 语言中不允许动态数组类型。

例如：

```
int n;
scanf("%d", &n);
int a[n];
```

用变量表示长度，想对数组的大小作动态说明，这是错误的。但是在实际的编程中，往往会发生这种情况，即所需的内存空间取决于实际输入的数据，而无法预先确定。对于这种问题，用数组的办法很难解决。为了解决上述问题，C 语言提供了一些内存管理函数，这些内存管理函数可以按需要动态地分配内存空间，也可把不再使用的空间回收待用，为有效地利用内存资源提供了手段。

常用的内存管理函数有以下三个：

1. 分配内存空间函数 malloc

调用形式：

(类型说明符*) malloc(size)

功能：在内存的动态存储区中分配一块长度为“size”字节的连续区域。函数的返回值为该区域的首地址。

“类型说明符”表示把该区域用于何种数据类型。

(类型说明符*)表示把返回值强制转换为该类型指针。

“size”是一个无符号数。

例如：

```
pc=(char *)malloc(100);
```

表示分配 100 个字节的内存空间，并强制转换为字符数组类型，函数的返回值为指向该字符数组的指针，把该指针赋予指针变量 pc。

2. 分配内存空间函数 calloc

calloc 也用于分配内存空间。

调用形式：

(类型说明符*) calloc(n, size)

功能：在内存动态存储区中分配 n 块长度为“size”字节的连续区域。函数的返回值为该区域的首地址。

(类型说明符*)用于强制类型转换。

calloc 函数与 malloc 函数的区别仅在于一次可以分配 n 块区域。

例如：

```
ps=(struct stu*)calloc(2, sizeof(struct stu));
```

其中的 `sizeof(struct stu)` 是求 `stu` 的结构长度。因此该语句的意思是：按 `stu` 的长度分配 2 块连续区域，强制转换为 `stu` 类型，并把其首地址赋予指针变量 `ps`。

2. 释放内存空间函数 `free`

调用形式：

`free(void*ptr);`

功能：释放 `ptr` 所指向的一块内存空间，`ptr` 是一个任意类型的指针变量，它指向被释放区域的首地址。被释放区应是由 `malloc` 或 `calloc` 函数所分配的区域。

【例 11.8】分配一块区域，输入一个学生数据。

```
main()
{
    struct stu
    {
        int num;
        char *name;
        char sex;
        float score;
    } *ps;
    ps=(struct stu*)malloc(sizeof(struct stu));
    ps->num=102;
    ps->name="Zhang ping";
    ps->sex='M';
    ps->score=62.5;
    printf("Number=%d\nName=%s\n", ps->num, ps->name);
    printf("Sex=%c\nScore=%f\n", ps->sex, ps->score);
    free(ps);
}
```



本例中，定义了结构 `stu`，定义了 `stu` 类型指针变量 `ps`。然后分配一块 `stu` 大内存区，并把首地址赋予 `ps`，使 `ps` 指向该区域。再以 `ps` 为指向结构的指针变量对各成员赋值，并用 `printf` 输出各成员值。最后用 `free` 函数释放 `ps` 指向的内存空间。整个程序包含了申请内存空间、使用内存空间、释放内存空间三个步骤，实现存储空间的动态分配。

11.9 链表的概念

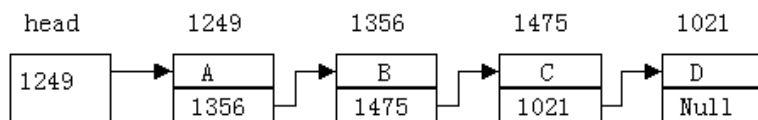
在例 7.8 中采用了动态分配的办法为一个结构分配内存空间。每一次分配一块空间用来存放一个学生的数据，我们可称之为一个结点。有多少个学生就应该申请分配多少块内存空间，也就是说要建立多少个结点。当然用结构数组也可以完成上述工作，但如果预先不能准确把握学生人数，也就无法确定数组大小。而且当学生留级、退学之后也不能把该元素占用的空间从数组中释放出来。

用动态存储的方法可以很好地解决这些问题。有一个学生就分配一个结点，无须预先确定学生的准确人数，某学生退学，可删去该结点，并释放该结点占用的存储空间。从而节约了宝贵的内存资源。另一方面，用数组的方法必须占用一块连续的内存区域。而使用动态分配时，每个结点之间可以是不连续的（结点内是连续的）。结点之间的联系可以用指针实现。

即在结点结构中定义一个成员项用来存放下一结点的首地址，这个用于存放地址的成员，常把它称为指针域。

可在第一个结点的指针域内存入第二个结点的首地址，在第二个结点的指针域内又存放第三个结点的首地址，如此串连下去直到最后一个结点。最后一个结点因无后续结点连接，其指针域可赋为 0。这样一种连接方式，在数据结构中称为“链表”。

下图为最简单链表的示意图。



图中，第 0 个结点称为头结点，它存放有第一个结点的首地址，它没有数据，只是一个指针变量。以下的每个结点都分为两个域，一个是数据域，存放各种实际的数据，如学号 num，姓名 name，性别 sex 和成绩 score 等。另一个域为指针域，存放下一结点的首地址。链表中的每一个结点都是同一种结构类型。

例如，一个存放学生学号和成绩的结点应为以下结构：

```

struct stu
{
    int num;
    int score;
    struct stu *next;
}
  
```

前两个成员项组成数据域，后一个成员项 next 构成指针域，它是一个指向 stu 类型结构的指针变量。

链表的基本操作对链表的主要操作有以下几种：

1. 建立链表；
2. 结构的查找与输出；
3. 插入一个结点；
4. 删除一个结点；

下面通过例题来说明这些操作。

【例 11.9】建立一个三个结点的链表，存放学生数据。为简单起见，我们假定学生数据结构中只有学号和年龄两项。可编写一个建立链表的函数 creat。程序如下：

```

#define NULL 0
#define TYPE struct stu
#define LEN sizeof (struct stu)
struct stu
{
    int num;
    int age;
    struct stu *next;
};
TYPE *creat(int n)
{
    struct stu *head,*pf,*pb;
    int i;
    for(i=0;i<n;i++)
  
```

```

{
    pb=(TYPE*) malloc (LEN);
    printf("input Number and Age\n");
    scanf ("%d%d", &pb->num, &pb->age);
    if(i==0)
        pf=head=pb;
    else pf->next=pb;
    pb->next=NULL;
    pf=pb;
}
return(head);
}

```

在函数外首先用宏定义对三个符号常量作了定义。这里用 TYPE 表示 struct stu，用 LEN 表示 sizeof(struct stu) 主要的目的是为了在以下程序内减少书写并使阅读更加方便。结构 stu 定义为外部类型，程序中的各个函数均可使用该定义。

creat 函数用于建立一个有 n 个结点的链表，它是一个指针函数，它返回的指针指向 stu 结构。在 creat 函数内定义了三个 stu 结构的指针变量。head 为头指针，pf 为指向两相邻结点的前一结点的指针变量。pb 为后一结点的指针变量。

11.10 枚举类型

在实际问题中，有些变量的取值被限定在一个有限的范围内。例如，一个星期内只有七天，一年只有十二个月，一个班每周有六门课程等等。如果把这些量说明为整型，字符型或其它类型显然是不妥当的。为此，C 语言提供了一种称为“枚举”的类型。在“枚举”类型的定义中列举出所有可能的取值，被说明为该“枚举”类型的变量取值不能超过定义的范围。应该说明的是，枚举类型是一种基本数据类型，而不是一种构造类型，因为它不能再分解为任何基本类型。

11.10.1 枚举类型的定义和枚举变量的说明

1. 枚举的定义 枚举类型定义的一般形式为：

```
enum 枚举名 { 枚举值表 };
```

在枚举值表中应罗列出所有可用值。这些值也称为枚举元素。

例如：

该枚举名为 weekday，枚举值共有 7 个，即一周中的七天。凡被说明为 weekday 类型变量的取值只能是七天中的某一天。

2. 枚举变量的说明

如同结构和联合一样，枚举变量也可用不同的方式说明，即先定义后说明，同时定义说明或直接说明。

设有变量 a, b, c 被说明为上述的 weekday，可采用下述任一种方式：

```
enum weekday { sun, mon, tue, wed, thu, fri, sat };
enum weekday a, b, c;
```

或者为：

```
enum weekday { sun, mon, tue, wed, thu, fri, sat } a, b, c;
```

或者为：

```
enum { sun, mon, tue, wed, thu, fri, sat } a, b, c;
```

11.10.2 枚举类型变量的赋值和使用

枚举类型在使用中有以下规定：

1. 枚举值是常量，不是变量。不能在程序中用赋值语句再对它赋值。

例如对枚举 weekday 的元素再作以下赋值：

```
sun=5;
```

```
mon=2;
```

```
sun=mon;
```

都是错误的。

2. 枚举元素本身由系统定义了一个表示序号的数值，从 0 开始顺序定义为 0, 1, 2...

如在 weekday 中，sun 值为 0，mon 值为 1，..., sat 值为 6。

【例 11.10】

```
main() {
    enum weekday
    { sun, mon, tue, wed, thu, fri, sat } a, b, c;
    a=sun;
    b=mon;
    c=tue;
    printf("%d,%d,%d", a, b, c);
}
```



说明：

只能把枚举值赋予枚举变量，不能把元素的数值直接赋予枚举变量。如：

```
a=sum;
```

```
b=mon;
```

是正确的。而：

```
a=0;
```

```
b=1;
```

是错误的。如一定要把数值赋予枚举变量，则必须用强制类型转换。

如：

```
a=(enum weekday)2;
```

其意义是将顺序号为 2 的枚举元素赋予枚举变量 a，相当于：

```
a=tue;
```

还应该说明的是枚举元素不是字符常量也不是字符串常量，使用时不要加单、双引号。

【例 11.11】

```
main() {
    enum body
    { a, b, c, d } month[31], j;
```

```

int i;
j=a;
for(i=1;i<=30;i++){
    month[i]=j;
    j++;
    if (j>d) j=a;
}
for(i=1;i<=30;i++){
    switch(month[i])
    {
        case a:printf(" %2d  %c\t",i,'a'); break;
        case b:printf(" %2d  %c\t",i,'b'); break;
        case c:printf(" %2d  %c\t",i,'c'); break;
        case d:printf(" %2d  %c\t",i,'d'); break;
        default:break;
    }
}
printf("\n");
}

```



11.11 类型定义符 typedef

C 语言不仅提供了丰富的数据类型，而且还允许由用户自己定义类型说明符，也就是说允许由用户为数据类型取“别名”。类型定义符 typedef 即可用来完成此功能。例如，有整型量 a, b, 其说明如下：

```
int a, b;
```

其中 int 是整型变量的类型说明符。int 的完整写法为 integer，为了增加程序的可读性，可把整型说明符用 typedef 定义为：

```
typedef int INTEGER
```

这以后就可用 INTEGER 来代替 int 作整型变量的类型说明了。

例如：

```
INTEGER a, b;
```

它等效于：

```
int a, b;
```

用 typedef 定义数组、指针、结构等类型将带来很大的方便，不仅使程序书写简单而且使意义更为明确，因而增强了可读性。

例如：

```
typedef char NAME[20];
```

表示 NAME 是字符数组类型，数组长度为 20。然后可用 NAME 说明变量，如：

```
NAME a1, a2, s1, s2;
```

完全等效于：

```
char a1[20], a2[20], s1[20], s2[20]
```

又如：

```
typedef struct stu
{ char name[20];
  int age;
  char sex;
} STU;
```

定义 STU 表示 stu 的结构类型，然后可用 STU 来说明结构变量：

```
STU body1, body2;
```

typedef 定义的一般形式为：

```
typedef 原类型名 新类型名
```

其中原类型名中含有定义部分，新类型名一般用大写表示，以便于区别。

有时也可用宏定义来代替 typedef 的功能，但是宏定义是由预处理完成的，而 typedef 则是在编译时完成的，后者更为灵活方便。

12	位运算.....	1
12.1	位运算符 C 语言提供了六种位运算符:	1
12.1.1	按位与运算.....	1
12.1.2	按位或运算.....	2
12.1.3	按位异或运算.....	2
12.1.4	求反运算.....	3
12.1.5	左移运算.....	3
12.1.6	右移运算.....	3
12.2	位域 (位段)	4
12.3	本章小结.....	6

12 位运算

前面介绍的各种运算都是以字节作为最基本位进行的。但在很多系统程序中常要求在位 (bit) 一级进行运算或处理。C 语言提供了位运算的功能, 这使得 C 语言也能像汇编语言一样用来编写系统程序。

12.1 位运算符 C 语言提供了六种位运算符:

&	按位与
	按位或
^	按位异或
~	取反
<<	左移
>>	右移

12.1.1 按位与运算

按位与运算符“&”是双目运算符。其功能是参与运算的两数各对应的二进位相与。只有对应的两个二进位均为 1 时, 结果位才为 1, 否则为 0。参与运算的数以补码方式出现。

例如: 9&5 可写算式如下:

00001001	(9 的二进制补码)
&00000101	(5 的二进制补码)
00000001	(1 的二进制补码)

可见 9&5=1。

按位与运算通常用来对某些位清 0 或保留某些位。例如把 a 的高八位清 0, 保留低八位, 可作 a&255 运算 (255 的二进制数为 0000000011111111)。

【例 12.1】

```
main() {
    int a=9, b=5, c;
```

```

c=a&b;
printf("a=%d\nb=%d\nc=%d\n", a, b, c);
}

```



12.1.2 按位或运算

按位或运算符“|”是双目运算符。其功能是参与运算的两数各对应的二进位相或。只要对应的二个二进位有一个为1时，结果位就为1。参与运算的两个数均以补码出现。

例如：9|5 可写算式如下：

```

00001001
|00000101
00001101      (十进制为 13) 可见 9|5=13

```

【例 12.2】

```

main() {
    int a=9, b=5, c;
    c=a|b;
    printf("a=%d\nb=%d\nc=%d\n", a, b, c);
}

```



12.1.3 按位异或运算

按位异或运算符“^”是双目运算符。其功能是参与运算的两数各对应的二进位相异或，当两对应的二进位相异时，结果为1。参与运算数仍以补码出现，例如 9^5 可写成算式如下：

```

00001001
^00000101
00001100      (十进制为 12)

```

【例 12.3】

```

main() {
    int a=9;
    a=a^5;
    printf("a=%d\n", a);
}

```



12.1.4 求反运算

求反运算符 \sim 为单目运算符，具有右结合性。其功能是对参与运算的数的各二进制位按位求反。

例如 ~ 9 的运算为：

$\sim(0000000000001001)$ 结果为：1111111111110110

12.1.5 左移运算

左移运算符“ \ll ”是双目运算符。其功能把“ \ll ”左边的运算数的各二进制位全部左移若干位，由“ \ll ”右边的数指定移动的位数，高位丢弃，低位补0。

例如：

$a \ll 4$

指把 a 的各二进制位向左移动 4 位。如 $a=00000011$ (十进制 3)，左移 4 位后为 00110000 (十进制 48)。

12.1.6 右移运算

右移运算符“ \gg ”是双目运算符。其功能是把“ \gg ”左边的运算数的各二进制位全部右移若干位，“ \gg ”右边的数指定移动的位数。

例如：

设 $a=15$,

$a \gg 2$

表示把 000001111 右移为 00000011 (十进制 3)。

应该说明的是，对于有符号数，在右移时，符号位将随同移动。当为正数时，最高位补0，而为负数时，符号位为1，最高位是补0或是补1取决于编译系统的规定。Turbo C 和很多系统规定为补1。

【例 12.4】

```
main() {
    unsigned a,b;
    printf("input a number: ");
    scanf("%d",&a);
    b=a>>5;
    b=b&15;
    printf("a=%d\tb=%d\n",a,b);
}
```



请再看一例！

【例 12.5】

```
main() {
    char a='a',b='b';
```



```

int p, c, d;
p=a;
p=(p<<8)|b;
d=p&0xff;
c=(p&0xff00)>>8;
printf("a=%d\nb=%d\nc=%d\nd=%d\n", a, b, c, d);
}

```



12.2 位域（位段）

有些信息在存储时，并不需要占用一个完整的字节，而只需占几个或一个二进制位。例如在存放一个开关量时，只有 0 和 1 两种状态，用一位二进制位即可。为了节省存储空间，并使处理简便，C 语言又提供了一种数据结构，称为“位域”或“位段”。

所谓“位域”是把一个字节中的二进制位划分为几个不同的区域，并说明每个区域的位数。每个域有一个域名，允许在程序中按域名进行操作。这样就可以把几个不同的对象用一个字节的二进制位域来表示。

1. 位域的定义和位域变量的说明

位域定义与结构定义相仿，其形式为：

```

struct 位域结构名
{ 位域列表 };

```

其中位域列表的形式为：

```

类型说明符 位域名：位域长度

```

例如：

```

struct bs
{
    int a:8;
    int b:2;
    int c:6;
};

```

位域变量的说明与结构变量说明的方式相同。可采用先定义后说明，同时定义说明或者直接说明这三种方式。

例如：

```

struct bs
{
    int a:8;
    int b:2;
    int c:6;
}data;

```

说明 data 为 bs 变量，共占两个字节。其中位域 a 占 8 位，位域 b 占 2 位，位域 c 占 6 位。

对于位域的定义尚有以下几点说明：

- 1) 一个位域必须存储在同一个字节中，不能跨两个字节。如一个字节所剩空间不够存

放另一位域时,应从下一单元起存放该位域。也可以有意使某位域从下一单元开始。

例如:

```
struct bs
{
    unsigned a:4
    unsigned :0      /*空域*/
    unsigned b:4      /*从下一单元开始存放*/
    unsigned c:4
}
```

在这个位域定义中, a 占第一字节的 4 位, 后 4 位填 0 表示不使用, b 从第二字节开始, 占用 4 位, c 占用 4 位。

- 2) 由于位域不允许跨两个字节, 因此位域的长度不能大于一个字节的长度, 也就是说不能超过 8 位二进制。
- 3) 位域可以无位域名, 这时它只用来作填充或调整位置。无名的位域是不能使用的。

例如:

```
struct k
{
    int a:1
    int :2      /*该 2 位不能使用*/
    int b:3
    int c:2
};
```

从以上分析可以看出, 位域在本质上就是一种结构类型, 不过其成员是按二进制分配的。

2. 位域的使用

位域的使用和结构成员的使用相同, 其一般形式为:

位域变量名·位域名

位域允许用各种格式输出。

【例 12.6】

```
main() {
    struct bs
    {
        unsigned a:1;
        unsigned b:3;
        unsigned c:4;
    } bit,*pbit;
    bit.a=1;
    bit.b=7;
    bit.c=15;
    printf("%d,%d,%d\n",bit.a,bit.b,bit.c);
    pbit=&bit;
    pbit->a=0;
    pbit->b&=3;
    pbit->c|=1;
    printf("%d,%d,%d\n",pbit->a,pbit->b,pbit->c);
}
```



```
}
```

上例程序中定义了位域结构 `bs`，三个位域为 `a, b, c`。说明了 `bs` 类型的变量 `bit` 和指向 `bs` 类型的指针变量 `pbit`。这表示位域也是可以使用指针的。程序的 9、10、11 三行分别给三个位域赋值(应注意赋值不能超过该位域的允许范围)。程序第 12 行以整型量格式输出三个域的内容。第 13 行把位域变量 `bit` 的地址送给指针变量 `pbit`。第 14 行用指针方式给位域 `a` 重新赋值，赋为 0。第 15 行使用了复合的位运算符“&=”，该行相当于：

```
pbit->b=pbit->b&3
```

位域 `b` 中原有值为 7，与 3 作按位与运算的结果为 3(111&011=011, 十进制值为 3)。同样，程序第 16 行中使用了复合位运算符“|=”，相当于：

```
pbit->c=pbit->c|1
```

其结果为 15。程序第 17 行用指针方式输出了这三个域的值。

12.3 本章小结

1. 位运算是 C 语言的一种特殊运算功能，它是以二进制位为单位进行运算的。位运算符只有逻辑运算和移位运算两类。位运算符可以与赋值符一起组成复合赋值符。如 `&=`, `|=`, `^=`, `>>=`, `<<=` 等。
2. 利用位运算可以完成汇编语言的某些功能，如置位，位清零，移位等。还可进行数据的压缩存储和并行运算。
3. 位域在本质上也是结构类型，不过它的成员按二进制位分配内存。其定义、说明及使用的方式都与结构相同。
4. 位域提供了一种手段，使得可在高级语言中实现数据的压缩，节省了存储空间，同时也提高了程序的效率。

13	文件.....	1
13.1	C 文件概述.....	1
13.2	文件指针.....	2
13.3	文件的打开与关闭.....	2
13.3.1	文件的打开 (fopen 函数).....	2
13.3.2	文件关闭函数 (fclose 函数)	4
13.4	文件的读写.....	4
13.4.1	字符读写函数 fgetc 和 fputc.....	4
13.4.2	字符串读写函数 fgets 和 fputs.....	8
13.4.3	数据块读写函数 fread 和 fwrite.....	9
13.4.4	格式化读写函数 fscanf 和 fprintf.....	11
13.5	文件的随机读写.....	12
13.5.1	文件定位.....	12
13.5.2	文件的随机读写.....	13
13.6	文件检测函数.....	14
13.6.1	文件结束检测函数 feof 函数.....	14
13.6.2	读写文件出错检测函数.....	14
13.6.3	文件出错标志和文件结束标志置 0 函数.....	14
13.7	C 库文件.....	14
13.8	本章小结.....	15

13 文件

13.1 C 文件概述

所谓“文件”是指一组相关数据的有序集合。这个数据集有一个名称，叫做文件名。实际上在前面的各章中我们已经多次使用了文件，例如源程序文件、目标文件、可执行文件、库文件（头文件）等。

文件通常是驻留在外部介质(如磁盘等)上的，在使用时才调入内存中来。从不同的角度可对文件作不同的分类。从用户的角度看，文件可分为普通文件和设备文件两种。

普通文件是指驻留在磁盘或其它外部介质上的一个有序数据集，可以是源文件、目标文件、可执行程序；也可以是一组待输入处理的原始数据，或者是一组输出的结果。对于源文件、目标文件、可执行程序可以称作程序文件，对输入输出数据可称作数据文件。

设备文件是指与主机相联的各种外部设备，如显示器、打印机、键盘等。在操作系统中，把外部设备也看作是一个文件来进行管理，把它们的输入、输出等同于对磁盘文件的读和写。

通常把显示器定义为标准输出文件，一般情况下在屏幕上显示有关信息就是向标准输出文件输出。如前面经常使用的 printf, putchar 函数就是这类输出。

键盘通常被指定标准的输入文件，从键盘上输入就意味着从标准输入文件上输入数据。scanf, getchar 函数就属于这类输入。

从文件编码的方式来看，文件可分为 ASCII 码文件和二进制码文件两种。ASCII 文件也称为文本文件，这种文件在磁盘存放时每个字符对应一个字节，用于存放对应的 ASCII 码。

例如，数 5678 的存储形式为：

ASCII 码： 00110101 00110110 00110111 00111000

	↓	↓	↓	↓
十进制码：	5	6	7	8

共占用 4 个字节。

ASCII 码文件可在屏幕上按字符显示，例如源程序文件就是 ASCII 文件，用 DOS 命令 TYPE 可显示文件的内容。由于是按字符显示，因此能读懂文件内容。

二进制文件是按二进制的编码方式来存放文件的。

例如，数 5678 的存储形式为：

00010110 00101110

只占二个字节。二进制文件虽然也可在屏幕上显示，但其内容无法读懂。C 系统在处理这些文件时，并不区分类型，都看成是字符流，按字节进行处理。

输入输出字符流的开始和结束只由程序控制而不受物理符号(如回车符)的控制。因此也把这种文件称作“流式文件”。

本章讨论流式文件的打开、关闭、读、写、定位等各种操作。

13.2 文件指针

在 C 语言中用一个指针变量指向一个文件，这个指针称为文件指针。通过文件指针就可对它所指的文件进行各种操作。

定义说明文件指针的一般形式为：

FILE *指针变量标识符；

其中 FILE 应为大写，它实际上是由系统定义的一个结构，该结构中含有文件名、文件状态和文件当前位置等信息。在编写源程序时不必关心 FILE 结构的细节。

例如：

FILE *fp；

表示 fp 是指向 FILE 结构的指针变量，通过 fp 即可找存放某个文件信息的结构变量，然后按结构变量提供的信息找到该文件，实施对文件的操作。习惯上也笼统地把 fp 称为指向一个文件的指针。

13.3 文件的打开与关闭

文件在进行读写操作之前要先打开，使用完毕要关闭。所谓打开文件，实际上是建立文件的各种有关信息，并使文件指针指向该文件，以便进行其它操作。关闭文件则断开指针与文件之间的联系，也就禁止再对该文件进行操作。

在 C 语言中，文件操作都是由库函数来完成的。在本章内将介绍主要的文件操作函数。

13.3.1 文件的打开(fopen 函数)

fopen 函数用来打开一个文件，其调用的一般形式为：

文件指针名=fopen(文件名, 使用文件方式)；

其中，

“文件指针名”必须是被说明为 FILE 类型的指针变量；

“文件名”是被打开文件的文件名；

“使用文件方式”是指文件的类型和操作要求。

“文件名”是字符串常量或字符串数组。

例如：

```
FILE *fp;
fp=("file a","r");
```

其意义是在当前目录下打开文件 file a，只允许进行“读”操作，并使 fp 指向该文件。

又如：

```
FILE *fphzk
fphzk=("c:\\hzk16","rb")
```

其意义是打开 C 驱动器磁盘的根目录下的文件 hzk16，这是一个二进制文件，只允许按二进制方式进行读操作。两个反斜线“\\”中的第一个表示转义字符，第二个表示根目录。

使用文件的方式共有 12 种，下面给出了它们的符号和意义。

文件使用方式	意义
"rt"	只读打开一个文本文件，只允许读数据
"wt"	只写打开或建立一个文本文件，只允许写数据
"at"	追加打开一个文本文件，并在文件末尾写数据
"rb"	只读打开一个二进制文件，只允许读数据
"wb"	只写打开或建立一个二进制文件，只允许写数据
"ab"	追加打开一个二进制文件，并在文件末尾写数据
"rt+"	读写打开一个文本文件，允许读和写
"wt+"	读写打开或建立一个文本文件，允许读写
"at+"	读写打开一个文本文件，允许读，或在文件末追加数据
"rb+"	读写打开一个二进制文件，允许读和写
"wb+"	读写打开或建立一个二进制文件，允许读和写
"ab+"	读写打开一个二进制文件，允许读，或在文件末追加数据

对于文件使用方式有以下几点说明：

- 1) 文件使用方式由 r, w, a, t, b, + 六个字符拼成，各字符的含义是：

```
r(read):      读
w(write):     写
a(append):    追加
t(text):      文本文件，可省略不写
b(banary):    二进制文件
+:            读和写
```

- 2) 凡用“r”打开一个文件时，该文件必须已经存在，且只能从该文件读出。
- 3) 用“w”打开的文件只能向该文件写入。若打开的文件不存在，则以指定的文件名建立该文件，若打开的文件已经存在，则将该文件删去，重建一个新文件。
- 4) 若要向一个已存在的文件追加新的信息，只能用“a”方式打开文件。但此时该文件必须是存在的，否则将会出错。
- 5) 在打开一个文件时，如果出错，fopen 将返回一个空指针值 NULL。在程序中可以用这一信息来判别是否完成打开文件的工作，并作相应的处理。因此常用以下程序段打开文件：
- 6)

```
if((fp=fopen("c:\\hzk16","rb"))==NULL)
{
```

```
printf("\nerror on open c:\\hzk16 file!");
getch();
exit(1);
}
```

这段程序的意义是，如果返回的指针为空，表示不能打开 C 盘根目录下的 hzk16 文件，则给出提示信息“error on open c:\\ hzk16 file!”，下一行 getch() 的功能是从键盘输入一个字符，但不在屏幕上显示。在这里，该行的作用是等待，只有当用户从键盘敲任一键时，程序才继续执行，因此用户可利用这个等待时间阅读出错提示。敲键后执行 exit(1) 退出程序。

- 7) 把一个文本文件读入内存时，要将 ASCII 码转换成二进制码，而把文件以文本方式写入磁盘时，也要把二进制码转换成 ASCII 码，因此文本文件的读写要花费较多的转换时间。对二进制文件的读写不存在这种转换。
- 8) 标准输入文件(键盘)，标准输出文件(显示器)，标准出错输出(出错信息)是由系统打开的，可直接使用。

13.3.2 文件关闭函数（fclose 函数）

文件一旦使用完毕，应用关闭文件函数把文件关闭，以避免文件的数据丢失等错误。

fclose 函数调用的一般形式是：

```
fclose(文件指针);
```

例如：

```
fclose(fp);
```

正常完成关闭文件操作时，fclose 函数返回值为 0。如返回非零值则表示有错误发生。

13.4 文件的读写

对文件的读和写是最常用的文件操作。在 C 语言中提供了多种文件读写的函数：

- 字符读写函数：fgetc 和 fputc
- 字符串读写函数：fgets 和 fputs
- 数据块读写函数：fread 和 fwrite
- 格式化读写函数：fscanf 和 fprintf

下面分别予以介绍。使用以上函数都要求包含头文件 stdio.h。

13.4.1 字符读写函数 fgetc 和 fputc

字符读写函数是以字符(字节)为单位的读写函数。每次可从文件读出或向文件写入一个字符。

1. 读字符函数 fgetc

fgetc 函数的功能是从指定的文件中读一个字符，函数调用的形式为：

```
字符变量=fgetc(文件指针);
```

例如：

```
ch=fgetc(fp);
```

其意义是从打开的文件 fp 中读取一个字符并送入 ch 中。

对于 fgetc 函数的使用有以下几点说明：

- 1) 在 fgetc 函数调用中，读取的文件必须是以读或读写方式打开的。
- 2) 读取字符的结果也可以不向字符变量赋值，
例如：
fgetc(fp);
但是读出的字符不能保存。
- 3) 在文件内部有一个位置指针。用来指向文件的当前读写字节。在文件打开时，该指针总是指向文件的第一个字节。使用 fgetc 函数后，该位置指针将向后移动一个字节。因此可连续多次使用 fgetc 函数，读取多个字符。应注意文件指针和文件内部的位置指针不是一回事。文件指针是指向整个文件的，须在程序中定义说明，只要不重新赋值，文件指针的值是不变的。文件内部的位置指针用以指示文件内部的当前读写位置，每读写一次，该指针均向后移动，它不需在程序中定义说明，而是由系统自动设置的。

【例 13.1】读入文件 cl.doc，在屏幕上输出。

```
#include<stdio.h>
main()
{
    FILE *fp;
    char ch;
    if((fp=fopen("d:\\jrzh\\example\\cl.txt", "rt"))==NULL)
    {
        printf("\nCannot open file strike any key exit!");
        getch();
        exit(1);
    }
    ch=fgetc(fp);
    while(ch!=EOF)
    {
        putchar(ch);
        ch=fgetc(fp);
    }
    fclose(fp);
}
```



本例程序的功能是从文件中逐个读取字符，在屏幕上显示。程序定义了文件指针 fp，以读文本文件方式打开文件“d:\\jrzh\\example\\ex1_1.c”，并使 fp 指向该文件。如打开文件出错，给出提示并退出程序。程序第 12 行先读出一个字符，然后进入循环，只要读出的字符不是文件结束标志（每个文件末有一结束标志 EOF）就把该字符显示在屏幕上，再读入下一字符。每读一次，文件内部的位置指针向后移动一个字符，文件结束时，该指针指向 EOF。执行本程序将显示整个文件。

2. 写字符函数 fputc

fputc 函数的功能是把一个字符写入指定的文件中，函数调用的形式为：

fputc(字符量, 文件指针);

其中, 待写入的字符量可以是字符常量或变量, 例如:

```
fputc('a', fp);
```

其意义是把字符 a 写入 fp 所指向的文件中。

对于 fputc 函数的使用也要说明几点:

- 1) 被写入的文件可以用写、读写、追加方式打开, 用写或读写方式打开一个已存在的文件时将清除原有的文件内容, 写入字符从文件首开始。如需保留原有文件内容, 希望写入的字符以文件末开始存放, 必须以追加方式打开文件。被写入的文件若不存在, 则创建该文件。
- 2) 每写入一个字符, 文件内部位置指针向后移动一个字节。
- 3) fputc 函数有一个返回值, 如写入成功则返回写入的字符, 否则返回一个 EOF。可用此来判断写入是否成功。

【例 13.2】从键盘输入一行字符, 写入一个文件, 再把该文件内容读出显示在屏幕上。

```
#include<stdio.h>
main()
{
    FILE *fp;
    char ch;
    if((fp=fopen("d:\\jrzh\\example\\string", "wt+"))==NULL)
    {
        printf("Cannot open file strike any key exit!");
        getch();
        exit(1);
    }
    printf("input a string:\n");
    ch=getchar();
    while (ch!='\n')
    {
        fputc(ch, fp);
        ch=getchar();
    }
    rewind(fp);
    ch=fgetc(fp);
    while(ch!=EOF)
    {
        putchar(ch);
        ch=fgetc(fp);
    }
    printf("\n");
    fclose(fp);
}
```



程序中第 6 行以读写文本文件方式打开文件 string。程序第 13 行从键盘读入一个字符

后进入循环，当读入字符不为回车符时，则把该字符写入文件之中，然后继续从键盘读入下一字符。每输入一个字符，文件内部位置指针向后移动一个字节。写入完毕，该指针已指向文件末。如要把文件从头读出，须把指针移向文件头，程序第 19 行 rewind 函数用于把 fp 所指文件的内部位置指针移到文件头。第 20 至 25 行用于读出文件中的一行内容。

【例 13.3】把命令行参数中的前一个文件名标识的文件，复制到后一个文件名标识的文件中，如命令行中只有一个文件名则把该文件写到标准输出文件(显示器)中。

```
#include<stdio.h>
main(int argc, char *argv[])
{
    FILE *fp1, *fp2;
    char ch;
    if(argc==1)
    {
        printf("have not enter file name strike any key exit");
        getch();
        exit(0);
    }
    if((fp1=fopen(argv[1], "rt"))==NULL)
    {
        printf("Cannot open %s\n", argv[1]);
        getch();
        exit(1);
    }
    if(argc==2) fp2=stdout;
    else if((fp2=fopen(argv[2], "wt+"))==NULL)
    {
        printf("Cannot open %s\n", argv[1]);
        getch();
        exit(1);
    }
    while((ch=fgetc(fp1))!=EOF)
        fputc(ch, fp2);
    fclose(fp1);
    fclose(fp2);
}
```



本程序为带参的 main 函数。程序中定义了两个文件指针 fp1 和 fp2，分别指向命令行参数中给出的文件。如命令行参数中没有给出文件名，则给出提示信息。程序第 18 行表示如果只给出一个文件名，则使 fp2 指向标准输出文件(即显示器)。程序第 25 行至 28 行用循环语句逐个读出文件 1 中的字符再送到文件 2 中。再次运行时，给出了一个文件名，故输出给标准输出文件 stdout，即在显示器上显示文件内容。第三次运行，给出了二个文件名，因此把 string 中的内容读出，写入到 OK 之中。可用 DOS 命令 type 显示 OK 的内容。

13.4.2 字符串读写函数 fgets 和 fputs

1. 读字符串函数 fgets

函数的功能是从指定的文件中读一个字符串到字符数组中，函数调用的形式为：

fgets(字符数组名, n, 文件指针);

其中的 n 是一个正整数。表示从文件中读出的字符串不超过 n-1 个字符。在读入的最后一个字符后加上串结束标志 '\0'。

例如：

```
fgets(str, n, fp);
```

的意义是从 fp 所指的文件中读出 n-1 个字符送入字符数组 str 中。

【例 13.4】从 string 文件中读入一个含 10 个字符的字符串。

```
#include<stdio.h>
main()
{
    FILE *fp;
    char str[11];
    if((fp=fopen("d:\\jrzh\\example\\string", "rt"))==NULL)
    {
        printf("\nCannot open file strike any key exit!");
        getch();
        exit(1);
    }
    fgets(str, 11, fp);
    printf("\n%s\n", str);
    fclose(fp);
}
```



本例定义了一个字符数组 str 共 11 个字节，在以读文本文件方式打开文件 string 后，从中读出 10 个字符送入 str 数组，在数组最后一个单元内将加上 '\0'，然后在屏幕上显示输出 str 数组。输出的十个字符正是例 13.1 程序的前十个字符。

对 fgets 函数有两点说明：

- 1) 在读出 n-1 个字符之前，如遇到了换行符或 EOF，则读出结束。
- 2) fgets 函数也有返回值，其返回值是字符数组的首地址。

2. 写字符串函数 fputs

fputs 函数的功能是向指定的文件写入一个字符串，其调用形式为：

fputs(字符串, 文件指针);

其中字符串可以是字符串常量，也可以是字符数组名，或指针变量，例如：

```
fputs("abcd", fp);
```

其意义是把字符串 "abcd" 写入 fp 所指的文件之中。

【例 13.5】在例 13.2 中建立的文件 string 中追加一个字符串。

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

```

{
    FILE *fp;
    char ch, st[20];
    if((fp=fopen("string", "at+"))==NULL)
    {
        printf("Cannot open file strike any key exit!");
        getch();
        exit(1);
    }

    printf("input a string:\n");
    scanf("%s", st);
    fputs(st, fp);
    rewind(fp);
    ch=fgetc(fp);
    while(ch!=EOF)
    {
        putchar(ch);
        ch=fgetc(fp);
    }
    printf("\n");
    fclose(fp);
}

```



本例要求在 string 文件末加写字符串，因此，在程序第 6 行以追加读写文本文件的方式打开文件 string。然后输入字符串，并用 fputs 函数把该串写入文件 string。在程序 15 行用 rewind 函数把文件内部位置指针移到文件首。再进入循环逐个显示当前文件中的全部内容。

13.4.3 数据块读写函数 fread 和 fwrite

C 语言还提供了用于整块数据的读写函数。可用来读写一组数据，如一个数组元素，一个结构变量的值等。

读数据块函数调用的一般形式为：

```
fread(buffer, size, count, fp);
```

写数据块函数调用的一般形式为：

```
fwrite(buffer, size, count, fp);
```

其中：

buffer 是一个指针，在 fread 函数中，它表示存放输入数据的首地址。在 fwrite 函数中，它表示存放输出数据的首地址。

size 表示数据块的字节数。

count 表示要读写的数据块块数。

fp 表示文件指针。

例如：

```
fread(fa, 4, 5, fp);
```

其意义是从 fp 所指的文件中，每次读 4 个字节(一个实数)送入实数组 fa 中，连续读 5 次，即读 5 个实数到 fa 中。

【例 13.6】从键盘输入两个学生数据，写入一个文件中，再读出这两个学生的数据显示在屏幕上。

```
#include<stdio.h>
struct stu
{
    char name[10];
    int num;
    int age;
    char addr[15];
} boya[2], boyb[2], *pp, *qq;
main()
{
    FILE *fp;
    char ch;
    int i;
    pp=boya;
    qq=boyb;
    if((fp=fopen("d:\\jrzh\\example\\stu_list", "wb+"))==NULL)
    {
        printf("Cannot open file strike any key exit!");
        getch();
        exit(1);
    }
    printf("\ninput data\n");
    for(i=0; i<2; i++, pp++)
        scanf("%s%d%d%s", pp->name, &pp->num, &pp->age, pp->addr);
    pp=boya;
    fwrite(pp, sizeof(struct stu), 2, fp);
    rewind(fp);
    fread(qq, sizeof(struct stu), 2, fp);
    printf("\n\nname\tnumber\t\tage\t\taddr\n");
    for(i=0; i<2; i++, qq++)
        printf("%s\t%5d\t7d\t\t%s\n", qq->name, qq->num, qq->age, qq->addr);
    fclose(fp);
}
```



本例程序定义了一个结构 stu，说明了两个结构数组 boya 和 boyb 以及两个结构指针变量 pp 和 qq。pp 指向 boya，qq 指向 boyb。程序第 16 行以读写方式打开二进制文件“stu_list”，输入二个学生数据之后，写入该文件中，然后把文件内部位置指针移到文件首，读出两块学生数据后，在屏幕上显示。

13.4.4 格式化读写函数 fscanf 和 fprintf

fscanf 函数, fprintf 函数与前面使用的 scanf 和 printf 函数的功能相似, 都是格式化读写函数。两者的区别在于 fscanf 函数和 fprintf 函数的读写对象不是键盘和显示器, 而是磁盘文件。

这两个函数的调用格式为:

fscanf(文件指针, 格式字符串, 输入表列);

fprintf(文件指针, 格式字符串, 输出表列);

例如:

```
fscanf(fp, "%d%s", &i, s);
```

```
fprintf(fp, "%d%c", j, ch);
```

用 fscanf 和 fprintf 函数也可以完成例 10.6 的问题。修改后的程序如例 10.7 所示。

【例 13.7】用 fscanf 和 fprintf 函数成例 10.6 的问题。

```
#include<stdio.h>
struct stu
{
    char name[10];
    int num;
    int age;
    char addr[15];
}boya[2], boyb[2], *pp, *qq;
main()
{
    FILE *fp;
    char ch;
    int i;
    pp=boya;
    qq=boyb;
    if((fp=fopen("stu_list", "wb+"))==NULL)
    {
        printf("Cannot open file strike any key exit!");
        getch();
        exit(1);
    }
    printf("\ninput data\n");
    for(i=0; i<2; i++, pp++)
        scanf("%s%d%d%s", pp->name, &pp->num, &pp->age, pp->addr);
    pp=boya;
    for(i=0; i<2; i++, pp++)
        fprintf(fp, "%s %d %d %s\n", pp->name, pp->num, pp->age, pp->
            addr);
    rewind(fp);
    for(i=0; i<2; i++, qq++)
```

```

    fscanf(fp, "%s %d %d %s\n", qq->name, &qq->num, &qq->age, qq->addr);
    printf("\nnname\tnumber      age      addr\n");
    qq=boyb;
    for(i=0; i<2; i++, qq++)
        printf("%s\t%5d %7d      %s\n", qq->name, qq->num, qq->age,
                qq->addr);
    fclose(fp);
}

```



与例 10.6 相比，本程序中 `fscanf` 和 `fprintf` 函数每次只能读写一个结构数组元素，因此采用了循环语句来读写全部数组元素。还要注意指针变量 `pp, qq` 由于循环改变了它们的值，因此在程序的 25 和 32 行分别对它们重新赋予了数组的首地址。

13.5 文件的随机读写

前面介绍的对文件的读写方式都是顺序读写，即读写文件只能从头开始，顺序读写各个数据。但在实际问题中常要求只读写文件中某一指定的部分。为了解决这个问题可移动文件内部的位置指针到需要读写的位置，再进行读写，这种读写称为随机读写。

实现随机读写的关键是要按要求移动位置指针，这称为文件的定位。

13.5.1 文件定位

移动文件内部位置指针的函数主要有两个，即 `rewind` 函数和 `fseek` 函数。

`rewind` 函数前面已多次使用过，其调用形式为：

`rewind(文件指针);`

它的功能是把文件内部的位置指针移到文件首。

下面主要介绍 `fseek` 函数。

`fseek` 函数用来移动文件内部位置指针，其调用形式为：

`fseek(文件指针, 位移量, 起始点);`

其中：

“文件指针”指向被移动的文件。

“位移量”表示移动的字节数，要求位移量是 `long` 型数据，以便在文件长度大于 64KB 时不会出错。当用常量表示位移量时，要求加后缀“`L`”。

“起始点”表示从何处开始计算位移量，规定的起始点有三种：文件首，当前位置和文件尾。其表示方法如下表。

起始点	表示符号	数字表示
文件首	SEEK_SET	0
当前位置	SEEK_CUR	1
文件末尾	SEEK_END	2

例如：

```
fseek(fp, 100L, 0);
```

其意义是把位置指针移到离文件首 100 个字节处。

还要说明的是 fseek 函数一般用于二进制文件。在文本文件中由于要进行转换，故往往计算的位置会出现错误。

13.5.2 文件的随机读写

在移动位置指针之后，即可用前面介绍的任一种读写函数进行读写。由于一般是读写一个数据块，因此常用 fread 和 fwrite 函数。

下面用例题来说明文件的随机读写。

【例 13.8】在学生文件 stu_list 中读出第二个学生的数据。

```
#include<stdio.h>
struct stu
{
    char name[10];
    int num;
    int age;
    char addr[15];
}boy,*qq;
main()
{
    FILE *fp;
    char ch;
    int i=1;
    qq=&boy;
    if((fp=fopen("stu_list","rb"))==NULL)
    {
        printf("Cannot open file strike any key exit!");
        getch();
        exit(1);
    }
    rewind(fp);
    fseek(fp,i*sizeof(struct stu),0);
    fread(qq,sizeof(struct stu),1,fp);
    printf("\n\nname\tnumber\t\tage\t\taddr\n");
    printf("%s\t%5d\t%7d\t\t%s\n",qq->name,qq->num,qq->age,
        qq->addr);
}
```



文件 stu_list 已由例 13.6 的程序建立，本程序用随机读出的方法读出第二个学生的数据。程序中定义 boy 为 stu 类型变量，qq 为指向 boy 的指针。以读二进制文件方式打开文件，程序第 22 行移动文件位置指针。其中的 i 值为 1，表示从文件头开始，移动一个 stu 类型的长度，然后再读出的数据即为第二个学生的数据。

13.6 文件检测函数

C 语言中常用的文件检测函数有以下几个。

13.6.1 文件结束检测函数 `fEOF` 函数

调用格式：

`fEOF(文件指针);`

功能：判断文件是否处于文件结束位置，如文件结束，则返回值为 1，否则为 0。

13.6.2 读写文件出错检测函数

`ferror` 函数调用格式：

`ferror(文件指针);`

功能：检查文件在用各种输入输出函数进行读写时是否出错。如 `ferror` 返回值为 0 表示未出错，否则表示有错。

13.6.3 文件出错标志和文件结束标志置 0 函数

`clearerr` 函数调用格式：

`clearerr(文件指针);`

功能：本函数用于清除出错标志和文件结束标志，使它们为 0 值。

13.7 C 库文件

C 系统提供了丰富的系统文件，称为库文件，C 的库文件分为两类，一类是扩展名为“.h”的文件，称为头文件，在前面的包含命令中我们已多次使用过。在“.h”文件中包含了常量定义、类型定义、宏定义、函数原型以及各种编译选择设置等信息。另一类是函数库，包括了各种函数的目标代码，供用户在程序中调用。通常在程序中调用一个库函数时，要在调用之前包含该函数原型所在的“.h”文件。

下面给出 Turbo C 的全部“.h”文件。

Turbo C 头文件

- `ALLOC.H` 说明内存管理函数(分配、释放等)。
- `ASSERT.H` 定义 `assert` 调试宏。
- `BIOS.H` 说明调用 IBM-PC ROM BIOS 子程序的各个函数。
- `CONIO.H` 说明调用 DOS 控制台 I/O 子程序的各个函数。
- `CTYPE.H` 包含有关字符分类及转换的名类信息(如 `isalpha` 和 `toascii` 等)。
- `DIR.H` 包含有关目录和路径的结构、宏定义和函数。
- `DOS.H` 定义和说明 MSDOS 和 8086 调用的一些常量和函数。
- `ERRON.H` 定义错误代码的助记符。
- `FCNTL.H` 定义在与 `open` 库子程序连接时的符号常量。

- `FLOAT.H` 包含有关浮点运算的一些参数和函数。
- `GRAPHICS.H` 说明有关图形功能的各个函数，图形错误代码的常量定义，正对不同驱动程序的各种颜色值，及函数用到的一些特殊结构。
- `IO.H` 包含低级 I/O 子程序的结构和说明。
- `LIMIT.H` 包含各环境参数、编译时间限制、数的范围等信息。
- `MATH.H` 说明数学运算函数，还定了 `HUGE_VAL` 宏，说明了 `matherr` 和 `matherr` 子程序用到的特殊结构。
- `MEM.H` 说明一些内存操作函数(其中大多数也在 `STRING.H` 中说明)。
- `PROCESS.H` 说明进程管理的各个函数，`spawn...`和 `EXEC ...`函数的结构说明。
- `SETJMP.H` 定义 `longjmp` 和 `setjmp` 函数用到的 `jmp buf` 类型，说明这两个函数。
- `SHARE.H` 定义文件共享函数的参数。
- `SIGNAL.H` 定义 `SIG[ZZ(Z) [ZZ)]IGN` 和 `SIG[ZZ(Z) [ZZ)]DFL` 常量，说明 `raise` 和 `signal` 两个函数。
- `STDARG.H` 定义读函数参数表的宏。(如 `vprintf`, `vscanf` 函数)。
- `STDDEF.H` 定义一些公共数据类型和宏。
- `STDIO.H` 定义 Kernighan 和 Ritchie 在 Unix System V 中定义的标准和扩展的类型和宏。还定义标准 I/O 预定义流: `stdin`, `stdout` 和 `stderr`，说明 I/O 流子程序。
- `STDLIB.H` 说明一些常用的子程序: 转换子程序、搜索/ 排序子程序等。
- `STRING.H` 说明一些串操作和内存操作函数。
- `SYS\STAT.H` 定义在打开和创建文件时用到的一些符号常量。
- `SYS\TYPES.H` 说明 `ftime` 函数和 `timeb` 结构。
- `SYS\TIME.H` 定义时间的类型 `time[ZZ(Z) [ZZ)]t`。
- `TIME.H` 定义时间转换子程序 `asctime`, `localtime` 和 `gmtime` 的结构, `ctime`、`difftime`、`gmtime`、`localtime` 和 `stime` 用到的类型，并提供这些函数的原型。
- `VALUE.H` 定义一些重要常量，包括依赖于机器硬件的和为与 Unix System V 相兼容而说明的一些常量，包括浮点和双精度值的范围。

13.8 本章小结

1. C 系统把文件当作一个“流”，按字节进行处理。
2. C 文件按编码方式分为二进制文件和 ASCII 文件。
3. C 语言中，用文件指针标识文件，当一个文件被 打开时，可取得该文件指针。
4. 文件在读写之前必须打开，读写结束必须关闭。
5. 文件可按只读、只写、读写、追加四种操作方式打开，同时还必须指定文件的类型是二进制文件还是文本文件。
6. 文件可按字节，字符串，数据块为单位读写，文件也可按指定的格式进行读写。
7. 文件内部的位置指针可指示当前的读写位置，移动该指针可以对文件实现随机读写。

turbo c2.0 函数大全

函数名: abort

功 能: 异常终止一个进程

用 法: void abort(void);

程序例:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main(void)
```

```
{
```

```
    printf("Calling abort()\n");
```

```
    abort();
```

```
    return 0; /* This is never reached */
```

```
}
```

函数名: abs

功 能: 求整数的绝对值

用 法: int abs(int i);

程序例:

```
#include <stdio.h>
```

```
#include <math.h>
```

```
int main(void)
```

```
{
```

```
    int number = -1234;
```

```
    printf("number: %d   absolute value: %d\n", number, abs(number));
```

```
    return 0;
```

```
}
```

函数名: absread, abswrite

功 能: 绝对磁盘扇区读、写数据

用 法: int absread(int drive, int nsects, int sectno, void *buffer);

int abswrite(int drive, int nsects, int tsectno, void *buffer);

程序例:

```
/* absread example */
```

```
#include <stdio.h>
#include <conio.h>
#include <process.h>
#include <dos.h>

int main(void)
{
    int i, strt, ch_out, sector;
    char buf[512];

    printf("Insert a diskette into drive A and press any key\n");
    getch();
    sector = 0;
    if (absread(0, 1, sector, &buf) != 0)
    {
        perror("Disk problem");
        exit(1);
    }
    printf("Read OK\n");
    strt = 3;
    for (i=0; i<80; i++)
    {
        ch_out = buf[strt+i];
        putchar(ch_out);
    }
    printf("\n");
    return(0);
}
```

函数名: access

功 能: 确定文件的访问权限

用 法: int access(const char *filename, int amode);

程序例:

```
#include <stdio.h>
#include <io.h>
```

```
int file_exists(char *filename);
```

```
int main(void)
```

```
{
    printf("Does NOTEXIST.FIL exist: %s\n",
```

```
    file_exists("NOTEXISTS.FIL") ? "YES" : "NO");  
    return 0;  
}
```

```
int file_exists(char *filename)  
{  
    return (access(filename, 0) == 0);  
}
```

函数名: acos

功 能: 反余弦函数

用 法: double acos(double x);

程序例:

```
#include <stdio.h>  
#include <math.h>
```

```
int main(void)  
{  
    double result;  
    double x = 0.5;  
  
    result = acos(x);  
    printf("The arc cosine of %lf is %lf\n", x, result);  
    return 0;  
}
```

函数名: allocmem

功 能: 分配 DOS 存储段

用 法: int allocmem(unsigned size, unsigned *seg);

程序例:

```
#include <dos.h>  
#include <alloc.h>  
#include <stdio.h>
```

```
int main(void)  
{  
    unsigned int size, segp;  
    int stat;  
  
    size = 64; /* (64 x 16) = 1024 bytes */  
    stat = allocmem(size, &segp);
```

```
if (stat == -1)
    printf("Allocated memory at segment: %x\n", segp);
else
    printf("Failed: maximum number of paragraphs available is %u\n",
        stat);

return 0;
}
```

函数名: arc

功 能: 画一弧线

用 法: void far arc(int x, int y, int stangle, int endangle, int radius);

程序例:

```
#include <graphics.h>
```

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main(void)
```

```
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy;
    int stangle = 45, endangle = 135;
    int radius = 100;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();    /* an error occurred */
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();

        exit(1);    /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;
```

```
setcolor(getmaxcolor());

/* draw arc */
arc(midx, midy, stangle, endangle, radius);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: asctime

功 能: 转换日期和时间 为 ASCII 码

用 法: char *asctime(const struct tm *tblock);

程序例:

```
#include <stdio.h>
#include <string.h>
#include <time.h>
```

```
int main(void)
{
    struct tm t;
    char str[80];

    /* sample loading of tm structure */

    t.tm_sec    = 1; /* Seconds */
    t.tm_min    = 30; /* Minutes */
    t.tm_hour   = 9;  /* Hour */
    t.tm_mday   = 22; /* Day of the Month */
    t.tm_mon    = 11; /* Month */
    t.tm_year   = 56; /* Year - does not include century */
    t.tm_wday   = 4;  /* Day of the week */
    t.tm_yday   = 0;  /* Does not show in asctime */
    t.tm_isdst  = 0;  /* Is Daylight SavTime; does not show in asctime */

    /* converts structure to null terminated
    string */

    strcpy(str, asctime(&t));
    printf("%s\n", str);
}
```

```
    return 0;
}
```

函数名: asin

功 能: 反正弦函数

用 法: double asin(double x);

程序例:

```
#include <stdio.h>
```

```
#include <math.h>
```

```
int main(void)
{
    double result;
    double x = 0.5;

    result = asin(x);
    printf("The arc sin of %lf is %lf\n", x, result);
    return(0);
}
```

函数名: assert

功 能: 测试一个条件并可能使程序终止

用 法: void assert(int test);

程序例:

```
#include <assert.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct ITEM {
    int key;
    int value;
};
```

```
/* add item to list, make sure list is not null */
```

```
void additem(struct ITEM *itemptr) {
    assert(itemptr != NULL);
    /* add item to list */
}
```



```
int main(void)
{
    additem(NULL);
    return 0;
}
```

函数名: atan

功 能: 反正切函数

用 法: double atan(double x);

程序例:

```
#include <stdio.h>
#include <math.h>
```

```
int main(void)
{
    double result;
    double x = 0.5;

    result = atan(x);
    printf("The arc tangent of %lf is %lf\n", x, result);
    return(0);
}
```

函数名: atan2

功 能: 计算 Y/X 的反正切值

用 法: double atan2(double y, double x);

程序例:

```
#include <stdio.h>
#include <math.h>
```

```
int main(void)
{
    double result;
    double x = 90.0, y = 45.0;

    result = atan2(y, x);
    printf("The arc tangent ratio of %lf is %lf\n", (y / x), result);
    return 0;
}
```

```
}
```

函数名: atexit

功 能: 注册终止函数

用 法: int atexit(atexit_t func);

程序例:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
void exit_fn1(void)
```

```
{  
    printf("Exit function #1 called\n");  
}
```

```
void exit_fn2(void)
```

```
{  
    printf("Exit function #2 called\n");  
}
```

```
int main(void)
```

```
{  
    /* post exit function #1 */  
    atexit(exit_fn1);  
    /* post exit function #2 */  
    atexit(exit_fn2);  
    return 0;  
}
```

函数名: atof

功 能: 把字符串转换成浮点数

用 法: double atof(const char *nptr);

程序例:

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{  
    float f;  
    char *str = "12345.67";
```

```
f = atof(str);
printf("string = %s float = %f\n", str, f);
return 0;
}
```

函数名: atoi

功 能: 把字符串转换成长整型数

用 法: int atoi(const char *nptr);

程序例:

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
int main(void)
{
    int n;
    char *str = "12345.67";

    n = atoi(str);
    printf("string = %s integer = %d\n", str, n);
    return 0;
}
```

函数名: atol

功 能: 把字符串转换成长整型数

用 法: long atol(const char *nptr);

程序例:

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
int main(void)
{
    long l;
    char *str = "98765432";

    l = atol(str);
    printf("string = %s integer = %ld\n", str, l);
    return(0);
}
```

函数名: bar

功 能: 画一个二维条形图

用 法: void far bar(int left, int top, int right, int bottom);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy, i;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* loop through the fill patterns */
    for (i=SOLID_FILL; i<USER_FILL; i++)
    {
        /* set the fill style */
        setfillstyle(i, getmaxcolor());

        /* draw the bar */
        bar(midx-50, midy-50, midx+50,
            midy+50);

        getch();
    }
}
```

```
}

/* clean up */
closegraph();
return 0;
}
```

函数名: bar3d

功 能: 画一个三维条形图

用 法: void far bar3d(int left, int top, int right, int bottom,
int depth, int topflag);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy, i;

    /* initialize graphics, local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* loop through the fill patterns */
```

```
for (i=EMPTY_FILL; i<USER_FILL; i++)
{
    /* set the fill style */
    setfillstyle(i, getmaxcolor());

    /* draw the 3-d bar */
    bar3d(midx-50, midy-50, midx+50, midy+50, 10, 1);

    getch();
}

/* clean up */
closegraph();
return 0;
}
```

函数名: bdos

功 能: DOS 系统调用

用 法: int bdos(int dosfun, unsigned dosdx, unsigned dosal);

程序例:

```
#include <stdio.h>
#include <dos.h>

/* Get current drive as 'A', 'B', ... */
char current_drive(void)
{
    char curdrive;

    /* Get current disk as 0, 1, ... */
    curdrive = bdos(0x19, 0, 0);
    return('A' + curdrive);
}

int main(void)
{
    printf("The current drive is %c:\n", current_drive());
    return 0;
}
```

函数名: bdosptr

功 能: DOS 系统调用

用 法: int bdosptr(int dosfun, void *argument, unsigned dosal);

程序例:

```
#include <string.h>
#include <stdio.h>
#include <dir.h>
#include <dos.h>
#include <errno.h>
#include <stdlib.h>

#define BUFLLEN 80

int main(void)
{
    char buffer[BUFLLEN];
    int test;

    printf("Enter full pathname of a directory\n");
    gets(buffer);

    test = bdosptr(0x3B,buffer,0);
    if(test)
    {
        printf("DOS error message: %d\n", errno);
        /* See errno.h for error listings */
        exit(1);
    }

    getcwd(buffer, BUFLLEN);
    printf("The current directory is: %s\n", buffer);

    return 0;
}
```

函数名: bioscom

功 能: 串行 I/O 通信

用 法: int bioscom(int cmd, char abyte, int port);

程序例:

```
#include <bios.h>
#include <conio.h>

#define COM1          0
#define DATA_READY 0x100
#define TRUE          1
#define FALSE         0

#define SETTINGS ( 0x80 | 0x02 | 0x00 | 0x00)

int main(void)
{
    int in, out, status, DONE = FALSE;

    bioscom(0, SETTINGS, COM1);
    cprintf("... BIOSCOM [ESC] to exit ...\n");
    while (!DONE)
    {
        status = bioscom(3, 0, COM1);
        if (status & DATA_READY)
        if ((out = bioscom(2, 0, COM1) & 0x7F) != 0)
            putchar(out);
        if (kbhit())
        {
            if ((in = getch()) == '\x1B')
                DONE = TRUE;
            bioscom(1, in, COM1);
        }
    }
    return 0;
}
```

函数名: biosdisk

功 能: 软硬盘 I/O

用 法: int biosdisk(int cmd, int drive, int head, int track, int sector
int nsects, void *buffer);

程序例:

```
#include <bios.h>
```



```
#include <stdio.h>

int main(void)
{
    int result;
    char buffer[512];

    printf("Testing to see if drive a: is ready\n");
    result = biosdisk(4,0,0,0,0,1,buffer);
    result &= 0x02;
    (result) ? (printf("Drive A: Ready\n")) :
              (printf("Drive A: Not Ready\n"));

    return 0;
}
```

函数名: biosequip

功 能: 检查设备

用 法: int biosequip(void);

程序例:

```
#include <bios.h>
#include <stdio.h>

int main(void)
{
    int result;
    char buffer[512];

    printf("Testing to see if drive a: is ready\n");
    result = biosdisk(4,0,0,0,0,1,buffer);
    result &= 0x02;
    (result) ? (printf("Drive A: Ready\n")) :
              (printf("Drive A: Not Ready\n"));

    return 0;
}
```

函数名: bioskey

功 能: 直接使用 BIOS 服务的键盘接口

用 法: int bioskey(int cmd);

程序例:

```
#include <stdio.h>
#include <bios.h>
#include <ctype.h>

#define RIGHT 0x01
#define LEFT 0x02
#define CTRL 0x04
#define ALT 0x08

int main(void)
{
    int key, modifiers;

    /* function 1 returns 0 until a key is pressed */
    while (bioskey(1) == 0);

    /* function 0 returns the key that is waiting */
    key = bioskey(0);

    /* use function 2 to determine if shift keys were used */
    modifiers = bioskey(2);
    if (modifiers)
    {
        printf("[");
        if (modifiers & RIGHT) printf("RIGHT");
        if (modifiers & LEFT) printf("LEFT");
        if (modifiers & CTRL) printf("CTRL");
        if (modifiers & ALT) printf("ALT");
        printf("]");
    }
    /* print out the character read */
    if (isalnum(key & 0xFF))
        printf("%c\n", key);
    else
        printf("%#02x\n", key);
    return 0;
}
```

函数名: biosmemory

功 能: 返回存储块大小

用 法: int biosmemory(void);

程序例:

```
#include <stdio.h>
#include <bios.h>

int main(void)
{
    int memory_size;

    memory_size = biosmemory(); /* returns value up to 640K */
    printf("RAM size = %dK\n",memory_size);
    return 0;
}
```

函数名: biosprint

功 能: 直接使用 BIOS 服务的打印机 I/O

用 法: int biosprint(int cmd, int byte, int port);

程序例:

```
#include <stdio.h>
#include <conio.h>
#include <bios.h>

int main(void)
{
    #define STATUS 2 /* printer status command */
    #define PORTNUM 0 /* port number for LPT1 */

    int status, abyte=0;

    printf("Please turn off your printer. Press any key to continue\n");
    getch();
    status = biosprint(STATUS, abyte, PORTNUM);
    if (status & 0x01)
        printf("Device time out.\n");
    if (status & 0x08)
        printf("I/O error.\n");
}
```

```
    if (status & 0x10)
        printf("Selected.\n");
    if (status & 0x20)
        printf("Out of paper.\n");

    if (status & 0x40)
        printf("Acknowledge.\n");
    if (status & 0x80)
        printf("Not busy.\n");

    return 0;
}
```

函数名: biostime

功 能: 读取或设置 BIOS 时间

用 法: long biostime(int cmd, long newtime);

程序例:

```
#include <stdio.h>
#include <bios.h>
#include <time.h>
#include <conio.h>

int main(void)
{
    long bios_time;

    clrscr();
    cprintf("The number of clock ticks since midnight is:\r\n");
    cprintf("The number of seconds since midnight is:\r\n");
    cprintf("The number of minutes since midnight is:\r\n");
    cprintf("The number of hours since midnight is:\r\n");
    cprintf("\r\nPress any key to quit:");
    while(!kbhit())
    {
        bios_time = biostime(0, 0L);

        gotoxy(50, 1);
        cprintf("%lu", bios_time);
    }
}
```

```
    gotoxy(50, 2);
    cprintf("%.4f", bios_time / CLK_TCK);

    gotoxy(50, 3);
    cprintf("%.4f", bios_time / CLK_TCK / 60);

    gotoxy(50, 4);
    cprintf("%.4f", bios_time / CLK_TCK / 3600);
}
return 0;
}
```

函数名: brk

功 能: 改变数据段空间分配

用 法: int brk(void *endds);

程序例:

```
#include <stdio.h>
#include <alloc.h>

int main(void)
{
    char *ptr;

    printf("Changing allocation with brk()\n");
    ptr = malloc(1);
    printf("Before brk() call: %lu bytes free\n", coreleft());
    brk(ptr+1000);
    printf(" After brk() call: %lu bytes free\n", coreleft());
    return 0;
}
```

函数名: bsearch

功 能: 二分法搜索

用 法: void *bsearch(const void *key, const void *base, size_t *nelem,
size_t width, int(*fcmp)(const void *, const *));

程序例:

```
#include <stdlib.h>
#include <stdio.h>

#define NELEMS(arr) (sizeof(arr) / sizeof(arr[0]))

int numarray[] = {123, 145, 512, 627, 800, 933};

int numeric (const int *p1, const int *p2)
{
    return(*p1 - *p2);
}

int lookup(int key)
{
    int *itemptr;

    /* The cast of (int*)(const void *,const void*)
       is needed to avoid a type mismatch error at
       compile time */
    itemptr = bsearch (&key, numarray, NELEMS(numarray),
        sizeof(int), (int*)(const void *,const void *)numeric);
    return (itemptr != NULL);
}

int main(void)
{
    if (lookup(512))
        printf("512 is in the table.\n");
    else
        printf("512 isn't in the table.\n");

    return 0;
}
```

函数名: cabs

功 能: 计算复数的绝对值

用 法: double cabs(struct complex z);

程序例:

```
#include <stdio.h>
#include <math.h>
```

```
int main(void)
{
```

```
struct complex z;
double val;

z.x = 2.0;
z.y = 1.0;
val = cabs(z);

printf("The absolute value of %.2lf i %.2lf j is %.2lf", z.x, z.y, val);
return 0;
}
```

函数名: calloc

功 能: 分配主存储器

用 法: void *calloc(size_t nelem, size_t elsize);

程序例:

```
#include <stdio.h>
#include <alloc.h>

int main(void)
{
    char *str = NULL;

    /* allocate memory for string */
    str = calloc(10, sizeof(char));

    /* copy "Hello" into string */
    strcpy(str, "Hello");

    /* display string */
    printf("String is %s\n", str);

    /* free memory */
    free(str);

    return 0;
}
```

函数名: `ceil`

功 能: 向上舍入

用 法: `double ceil(double x);`

程序例:

```
#include <math.h>
#include <stdio.h>

int main(void)
{
    double number = 123.54;
    double down, up;

    down = floor(number);
    up = ceil(number);

    printf("original number      %5.2lf\n", number);
    printf("number rounded down %5.2lf\n", down);
    printf("number rounded up    %5.2lf\n", up);

    return 0;
}
```

函数名: `cgets`

功 能: 从控制台读字符串

用 法: `char *cgets(char *str);`

程序例:

```
#include <stdio.h>
#include <conio.h>

int main(void)
{
    char buffer[83];
    char *p;

    /* There's space for 80 characters plus the NULL terminator */
    buffer[0] = 81;

    printf("Input some chars:");
    p = cgets(buffer);
}
```



```
printf("\ncgets read %d characters: \"%s\\n\"", buffer[1], p);
printf("The returned pointer is %p, buffer[0] is at %p\\n", p, &buffer);

/* Leave room for 5 characters plus the NULL terminator */
buffer[0] = 6;

printf("Input some chars:");
p = cgets(buffer);
printf("\ncgets read %d characters: \"%s\\n\"", buffer[1], p);
printf("The returned pointer is %p, buffer[0] is at %p\\n", p, &buffer);
return 0;
}
```

函数名: `chdir`

功 能: 改变工作目录

用 法: `int chdir(const char *path);`

程序例:

```
#include <stdio.h>
#include <stdlib.h>
#include <dir.h>

char old_dir[MAXDIR];
char new_dir[MAXDIR];

int main(void)
{
    if (getcurdir(0, old_dir))
    {
        perror("getcurdir()");
        exit(1);
    }
    printf("Current directory is: \\%s\\n", old_dir);

    if (chdir("\\\\"))
    {
        perror("chdir()");
        exit(1);
    }

    if (getcurdir(0, new_dir))
```

```
{
    perror("getcurdir()");
    exit(1);
}
printf("Current directory is now: \\%s\\n", new_dir);

printf("\\nChanging back to original directory: \\%s\\n", old_dir);
if (chdir(old_dir))
{
    perror("chdir()");
    exit(1);
}

return 0;
}
```

函数名: _chmod, chmod

功 能: 改变文件的访问方式

用 法: int chmod(const char *filename, int permiss);

程序例:

```
#include <sys\stat.h>
```

```
#include <stdio.h>
```

```
#include <io.h>
```

```
void make_read_only(char *filename);
```

```
int main(void)
```

```
{
    make_read_only("NOTEXIST.FIL");
    make_read_only("MYFILE.FIL");
    return 0;
}
```

```
void make_read_only(char *filename)
```

```
{
    int stat;

    stat = chmod(filename, S_IREAD);
    if (stat)
        printf("Couldn't make %s read-only\\n", filename);
    else
```

```
    printf("Made %s read-only\n", filename);  
}
```

函数名: chsize

功 能: 改变文件大小

用 法: int chsize(int handle, long size);

程序例:

```
#include <string.h>  
#include <fcntl.h>  
#include <io.h>  
  
int main(void)  
{  
    int handle;  
    char buf[11] = "0123456789";  
  
    /* create text file containing 10 bytes */  
    handle = open("DUMMY.FIL", O_CREAT);  
    write(handle, buf, strlen(buf));  
  
    /* truncate the file to 5 bytes in size */  
    chsize(handle, 5);  
  
    /* close the file */  
    close(handle);  
    return 0;  
}
```

函数名: circle

功 能: 在给定半径以(x, y)为圆心画圆

用 法: void far circle(int x, int y, int radius);

程序例:

```
#include <graphics.h>  
#include <stdlib.h>  
#include <stdio.h>  
#include <conio.h>
```

```
int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy;
    int radius = 100;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;
    setcolor(getmaxcolor());

    /* draw the circle */
    circle(midx, midy, radius);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

函数名: cleardevice

功 能: 清除图形屏幕

用 法: void far cleardevice(void);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
```

```
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;
    setcolor(getmaxcolor());

    /* for centering screen messages */
    setttextjustify(CENTER_TEXT, CENTER_TEXT);

    /* output a message to the screen */
    outtextxy(midx, midy, "press any key to clear the screen:");

    /* wait for a key */
    getch();

    /* clear the screen */
    cleardevice();

    /* output another message */
    outtextxy(midx, midy, "press any key to quit:");

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

```
}
```

函数名: `clearerr`

功 能: 复位错误标志

用 法: `void clearerr(FILE *stream);`

程序例:

```
#include <stdio.h>

int main(void)
{
    FILE *fp;
    char ch;

    /* open a file for writing */
    fp = fopen("DUMMY.FIL", "w");

    /* force an error condition by attempting to read */
    ch = fgetc(fp);
    printf("%c\n", ch);

    if (ferror(fp))
    {
        /* display an error message */
        printf("Error reading from DUMMY.FIL\n");

        /* reset the error and EOF indicators */
        clearerr(fp);
    }

    fclose(fp);
    return 0;
}
```

函数名: `clearviewport`

功 能: 清除图形视区

用 法: `void far clearviewport(void);`

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

#define CLIP_ON 1    /* activates clipping in viewport */

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int ht;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    setcolor(getmaxcolor());
    ht = textheight("W");

    /* message in default full-screen viewport */
    outtextxy(0, 0, "* <-- (0, 0) in default viewport");

    /* create a smaller viewport */
    setviewport(50, 50, getmaxx()-50, getmaxy()-50, CLIP_ON);

    /* display some messages */
    outtextxy(0, 0, "* <-- (0, 0) in smaller viewport");
    outtextxy(0, 2*ht, "Press any key to clear viewport:");

    /* wait for a key */
    getch();

    /* clear the viewport */
```

```
clearviewport();

/* output another message */
outtextxy(0, 0, "Press any key to quit:");

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: `_close`, `close`

功 能: 关闭文件句柄

用 法: `int close(int handle);`

程序例:

```
#include <string.h>
#include <stdio.h>
#include <fcntl.h>
#include <io.h>

main()
{
    int handle;
    char buf[11] = "0123456789";

    /* create a file containing 10 bytes */
    handle = open("NEW.FIL", O_CREAT);
    if (handle > -1)
    {
        write(handle, buf, strlen(buf));

        /* close the file */
        close(handle);
    }
    else
    {
        printf("Error opening file\n");
    }
    return 0;
}
```


函数名: `clock`

功 能: 确定处理器时间

用 法: `clock_t clock(void);`

程序例:

```
#include <time.h>
#include <stdio.h>
#include <dos.h>

int main(void)
{
    clock_t start, end;
    start = clock();

    delay(2000);

    end = clock();
    printf("The time was: %f\n", (end - start) / CLK_TCK);

    return 0;
}
```

函数名: `closegraph`

功 能: 关闭图形系统

用 法: `void far closegraph(void);`

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int x, y;
```

```
/* initialize graphics mode */
initgraph(&gdriver, &gmode, "");

/* read result of initialization */
errorcode = graphresult();

if (errorcode != grOk) /* an error
    occurred */
{
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

x = getmaxx() / 2;
y = getmaxy() / 2;

/* output a message */
settextjustify(CENTER_TEXT, CENTER_TEXT);
outtextxy(x, y, "Press a key to close the graphics system:");

/* wait for a key */
getch();

/* closes down the graphics system */
closegraph();

printf("We're now back in text mode.\n");
printf("Press any key to halt:");
getch();
return 0;
}
```

函数名: `clreol`

功 能: 在文本窗口中清除字符到行末

用 法: `void clreol(void);`

程序例:

```
#include <conio.h>
```

```
int main(void)

{
    clrscr();
    cprintf("The function CLREOL clears all characters from the\r\n");
    cprintf("cursor position to the end of the line within the\r\n");
    cprintf("current text window, without moving the cursor.\r\n");
    cprintf("Press any key to continue . . .");
    gotoxy(14, 4);
    getch();

    clreol();
    getch();

    return 0;
}
```

函数名: clrscr

功 能: 清除文本模式窗口

用 法: void clrscr(void);

程序例:

```
#include <conio.h>
```

```
int main(void)
{
    int i;

    clrscr();
    for (i = 0; i < 20; i++)
        cprintf("%d\r\n", i);
    cprintf("\r\nPress any key to clear screen");
    getch();

    clrscr();
    cprintf("The screen has been cleared!");
    getch();

    return 0;
}
```

函数名: `coreleft`

功 能: 返回未使用内存的大小

用 法: `unsigned coreleft(void);`

程序例:

```
#include <stdio.h>
#include <alloc.h>

int main(void)
{
    printf("The difference between the highest allocated block and\n");
    printf("the top of the heap is: %lu bytes\n", (unsigned long) coreleft());

    return 0;
}
```

函数名: `cos`

功 能: 余弦函数

用 法: `double cos(double x);`

程序例:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    double result;
    double x = 0.5;

    result = cos(x);
    printf("The cosine of %lf is %lf\n", x, result);
    return 0;
}
```

函数名: `cosh`

功 能: 双曲余弦函数

用 法: `double cosh(double x);`

程序例:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    double result;
    double x = 0.5;

    result = cosh(x);
    printf("The hyperboic cosine of %lf is %lf\n", x, result);
    return 0;
}
```

函数名: `country`

功 能: 返回与国家有关的信息

用 法: `struct COUNTRY *country(int countrycode, struct country *country);`

程序例:

```
#include <dos.h>
#include <stdio.h>

#define USA 0

int main(void)
{
    struct COUNTRY country_info;

    country(USA, &country_info);
    printf("The currency symbol for the USA is: %s\n",
        country_info.co_curr);
    return 0;
}
```

函数名: `cprintf`

功 能: 送格式化输出至屏幕

用 法: int cprintf(const char *format[, argument, ...]);
程序例:

```
#include <conio.h>

int main(void)
{
    /* clear the screen */
    clrscr();

    /* create a text window */
    window(10, 10, 80, 25);

    /* output some text in the window */
    cprintf("Hello world\r\n");

    /* wait for a key */
    getch();
    return 0;
}
```

函数名: cputs
功 能: 写字符到屏幕
用 法: void cputs(const char *string);
程序例:

```
#include <conio.h>

int main(void)
{
    /* clear the screen */
    clrscr();

    /* create a text window */
    window(10, 10, 80, 25);

    /* output some text in the window */
    cputs("This is within the window\r\n");

    /* wait for a key */
    getch();
}
```

```
    return 0;
}
```

函数名: `_creat` `creat`

功 能: 创建一个新文件或重写一个已存在的文件

用 法: `int creat (const char *filename, int permiss);`

程序例:

```
#include <sys\stat.h>
#include <string.h>
#include <fcntl.h>
#include <io.h>

int main(void)
{
    int handle;
    char buf[11] = "0123456789";

    /* change the default file mode from text to binary */
    _fmode = O_BINARY;

    /* create a binary file for reading and writing */
    handle = creat("DUMMY.FIL", S_IREAD | S_IWRITE);

    /* write 10 bytes to the file */
    write(handle, buf, strlen(buf));

    /* close the file */
    close(handle);
    return 0;
}
```

函数名: `creatnew`

功 能: 创建一个新文件

用 法: `int creatnew(const char *filename, int attrib);`

程序例:

```
#include <string.h>
#include <stdio.h>
#include <errno.h>
```

```
#include <dos.h>
#include <io.h>

int main(void)
{
    int handle;
    char buf[11] = "0123456789";

    /* attempt to create a file that doesn't already exist */
    handle = creatnew("DUMMY.FIL", 0);

    if (handle == -1)
        printf("DUMMY.FIL already exists.\n");
    else
    {
        printf("DUMMY.FIL successfully created.\n");
        write(handle, buf, strlen(buf));
        close(handle);
    }
    return 0;
}
```

函数名: creattemp

功 能: 创建一个新文件或重写一个已存在的文件

用 法: int creattemp(const char *filename, int attrib);

程序例:

```
#include <string.h>
#include <stdio.h>
#include <io.h>

int main(void)
{
    int handle;
    char pathname[128];

    strcpy(pathname, "\\");

    /* create a unique file in the root directory */
    handle = creattemp(pathname, 0);
```



```
    printf("%s was the unique file created.\n", pathname);
    close(handle);
    return 0;
}
```

函数名: `cscanf`

功 能: 从控制台执行格式化输入

用 法: `int cscanf(char *format[,argument, ...]);`

程序例:

```
#include <conio.h>

int main(void)
{
    char string[80];

    /* clear the screen */
    clrscr();

    /* Prompt the user for input */
    cprintf("Enter a string with no spaces.");

    /* read the input */
    cscanf("%s", string);

    /* display what was read */
    cprintf("\r\nThe string entered is: %s", string);
    return 0;
}
```

函数名: `ctime`

功 能: 把日期和时间转换为字符串

用 法: `char *ctime(const time_t *time);`

程序例:

```
#include <stdio.h>
#include <time.h>
```

```
int main(void)
{
    time_t t;

    time(&t);
    printf("Today's date and time: %s\n", ctime(&t));
    return 0;
}
```

函数名: `ctrlbrk`

功 能: 设置 Ctrl-Break 处理程序

用 法: `void ctrlbrk(*fptr)(void);`

程序例:

```
#include <stdio.h>
#include <dos.h>

#define ABORT 0

int c_break(void)
{
    printf("Control-Break pressed.  Program aborting ...\n");
    return (ABORT);
}

int main(void)
{
    ctrlbrk(c_break);
    for(;;)
    {
        printf("Looping... Press <Ctrl-Break> to quit:\n");
    }
    return 0;
}

*****
```

函数名: `delay`

功 能: 将程序的执行暂停一段时间(毫秒)

用 法: `void delay(unsigned milliseconds);`

程序例:

```
/* Emits a 440-Hz tone for 500 milliseconds */
#include <dos.h>
```

```
int main(void)
{
    sound(440);
    delay(500);
    nosound();

    return 0;
}
```

函数名: `delline`

功 能: 在文本窗口中删去一行

用 法: `void delline(void);`

程序例:

```
#include <conio.h>
```

```
int main(void)
{
    clrscr();
    cprintf("The function DELLINE deletes \
the line containing the\r\n");
    cprintf("cursor and moves all lines \
below it one line up.\r\n");
    cprintf("DELLINE operates within the \
currently active text\r\n");
    cprintf("window. Press any key to \
continue . . .");
    gotoxy(1,2); /* Move the cursor to the
second line and first column */
    getch();

    delline();
    getch();

    return 0;
}
```

函数名: `detectgraph`

功 能: 通过检测硬件确定图形驱动程序和模式

用 法: `void far detectgraph(int far *graphdriver, int far *graphmode);`

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

/* names of the various cards supported */
char *dname[] = { "requests detection",
    "a CGA",
    "an MCGA",
    "an EGA",
    "a 64K EGA",
    "a monochrome EGA",
    "an IBM 8514",
    "a Hercules monochrome",
    "an AT&T 6300 PC",
    "a VGA",
    "an IBM 3270 PC"
};

int main(void)
{
    /* returns detected hardware info. */
    int gdriver, gmode, errorcode;

    /* detect graphics hardware available */
    detectgraph(&gdriver, &gmode);

    /* read result of detectgraph call */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error
        occurred */
    {
        printf("Graphics error: %s\n", \
            grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error
            code */
    }

    /* display the information detected */
    clrscr();
```

```
printf("You have %s video display \\  
card.\n", dname[gdriver]);  
printf("Press any key to halt:");  
getch();  
return 0;  
}
```

函数名: difftime

功 能: 计算两个时刻之间的时间差

用 法: double difftime(time_t time2, time_t time1);

程序例:

```
#include <time.h>  
#include <stdio.h>  
#include <dos.h>  
#include <conio.h>  
  
int main(void)  
{  
    time_t first, second;  
  
    clrscr();  
    first = time(NULL); /* Gets system  
        time */  
    delay(2000); /* Waits 2 secs */  
    second = time(NULL); /* Gets system time  
        again */  
  
    printf("The difference is: %f \\  
seconds\n", difftime(second, first));  
    getch();  
  
    return 0;  
}
```

函数名: disable

功 能: 屏蔽中断

用 法: void disable(void);

程序例:

```
/**NOTE: This is an interrupt service  
routine. You cannot compile this program  
with Test Stack Overflow turned on and  
get an executable file that operates  
correctly. */
```

```
#include <stdio.h>  
#include <dos.h>  
#include <conio.h>
```

```
#define INTR 0X1C    /* The clock tick  
interrupt */
```

```
void interrupt ( *oldhandler)(void);
```

```
int count=0;
```

```
void interrupt handler(void)  
{  
/* disable interrupts during the handling of  
the interrupt */  
disable();  
/* increase the global counter */  
count++;  
/* reenale interrupts at the end of the  
handler */  
enable();  
/* call the old routine */  
oldhandler();  
}
```

```
int main(void)  
{  
/* save the old interrupt vector */  
oldhandler = getvect(INTR);  
  
/* install the new interrupt handler */  
setvect(INTR, handler);  
  
/* loop until the counter exceeds 20 */  
while (count < 20)  
printf("count is %d\n",count);
```

```
/* reset the old interrupt handler */
    setvect(INTR, oldhandler);

    return 0;
}
```

函数名: div
功 能: 将两个整数相除, 返回商和余数
用 法: div_t (int number, int denom);
程序例:

```
#include <stdlib.h>
#include <stdio.h>

div_t x;

int main(void)
{
    x = div(10,3);
    printf("10 div 3 = %d remainder %d\n", x.quot, x.rem);

    return 0;
}
```

函数名: doserror
功 能: 获取扩展 DOS 错误信息
用 法: int doserror(struct DOSERR *dblkp);
程序例:

```
#include <stdio.h>
#include <dos.h>

int main(void)
{
    FILE *fp;
    struct DOSERROR info;

    fp = fopen("perror.dat", "r");
    if (!fp) perror("Unable to open file for
        reading");
    doserror(&info);
```

```
printf("Extended DOS error \
information:\n");
printf("    Extended error: \
%d\n",info.exterror);
printf("        Class: \
%x\n",info.class);
printf("        Action: \
%x\n",info.action);
printf("    Error Locus: \
%x\n",info.locus);

return 0;
}
```

函数名: dostounix

功 能: 转换日期和时间为 UNIX 时间格式

用 法: long dostounix(struct date *dateptr, struct time *timeptr);

程序例:

```
#include <time.h>
#include <stddef.h>
#include <dos.h>
#include <stdio.h>

int main(void)
{
    time_t t;
    struct time d_time;
    struct date d_date;
    struct tm *local;

    getdate(&d_date);
    gettime(&d_time);

    t = dostounix(&d_date, &d_time);
    local = localtime(&t);
    printf("Time and Date: %s\n", \
    asctime(local));

    return 0;
}
```


函数名: drawpoly

功 能: 画多边形

用 法: void far drawpoly(int numpoints, int far *polypoints);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int maxx, maxy;

    /* our polygon array */
    int poly[10];

    /* initialize graphics and local
       variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)
        /* an error occurred */
        {
            printf("Graphics error: %s\n", \
grapherrormsg(errorcode));
            printf("Press any key to halt:");
            getch();
            /* terminate with an error code */
            exit(1);
        }

    maxx = getmaxx();
    maxy = getmaxy();

    poly[0] = 20;          /* 1st vertex */
    poly[1] = maxy / 2;
```

```
poly[2] = maxx - 20; /* 2nd */
poly[3] = 20;

poly[4] = maxx - 50; /* 3rd */
poly[5] = maxy - 20;

poly[6] = maxx / 2; /* 4th */
poly[7] = maxy / 2;
/*
drawpoly doesn't automatically close
the polygon, so we close it.
*/
poly[8] = poly[0];
poly[9] = poly[1];

/* draw the polygon */
drawpoly(5, poly);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: dup

功 能: 复制一个文件句柄

用 法: int dup(int handle);

程序例:

```
#include <string.h>
#include <stdio.h>
#include <conio.h>
#include <io.h>
```

```
void flush(FILE *stream);
```

```
int main(void)
{
    FILE *fp;
    char msg[] = "This is a test";

    /* create a file */
```

```
fp = fopen("DUMMY.FIL", "w");

/* write some data to the file */
fwrite(msg, strlen(msg), 1, fp);

clrscr();
printf("Press any key to flush \
DUMMY.FIL:");
getch();

/* flush the data to DUMMY.FIL without
   closing it */
flush(fp);

printf("\nFile was flushed, Press any \
key to quit:");
getch();
return 0;
}

void flush(FILE *stream)
{
    int duphandle;

    /* flush TC's internal buffer */
    fflush(stream);

    /* make a duplicate file handle */
    duphandle = dup(fileno(stream));

    /* close the duplicate handle to flush the
       DOS buffer */
    close(duphandle);
}
```

函数名: dup2

功 能: 复制文件句柄

用 法: int dup2(int oldhandle, int newhandle);

程序例:

```
#include <sys\stat.h>
#include <string.h>
```

```
#include <fcntl.h>
#include <io.h>

int main(void)
{
    #define STDOUT 1

    int nul, oldstdout;
    char msg[] = "This is a test";

    /* create a file */
    nul = open("DUMMY.FIL", O_CREAT | O_RDWR,
        S_IREAD | S_IWRITE);

    /* create a duplicate handle for standard
       output */
    oldstdout = dup(STDOUT);
    /*
       redirect standard output to DUMMY.FIL
       by duplicating the file handle onto the
       file handle for standard output.
    */
    dup2(nul, STDOUT);

    /* close the handle for DUMMY.FIL */
    close(nul);

    /* will be redirected into DUMMY.FIL */
    write(STDOUT, msg, strlen(msg));

    /* restore original standard output
       handle */
    dup2(oldstdout, STDOUT);

    /* close duplicate handle for STDOUT */
    close(oldstdout);

    return 0;
}
*****
```

函数名: `ecvt`

功 能: 把一个浮点数转换为字符串

用 法: `char ecvt(double value, int ndigit, int *decpt, int *sign);`

程序例:

```
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    char *string;
    double value;
    int dec, sign;
    int ndig = 10;

    clrscr();
    value = 9.876;
    string = ecvt(value, ndig, &dec, &sign);
    printf("string = %s      dec = %d \
sign = %d\n", string, dec, sign);

    value = -123.45;
    ndig = 15;
    string = ecvt(value, ndig, &dec, &sign);
    printf("string = %s dec = %d sign = %d\n",
    string, dec, sign);

    value = 0.6789e5; /* scientific
notation */
    ndig = 5;
    string = ecvt(value, ndig, &dec, &sign);
    printf("string = %s      dec = %d \
sign = %d\n", string, dec, sign);

    return 0;
}
```

函数名: ellipse

功 能: 画一椭圆

用 法: void far ellipse(int x, int y, int stangle, int endangle,
int xradius, int yradius);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
```

```
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy;
    int stangle = 0, endangle = 360;
    int xradius = 100, yradius = 50;

    /* initialize graphics, local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)
        /* an error occurred */
        {
            printf("Graphics error: %s\n",
                grapherrormsg(errorcode));
            printf("Press any key to halt:");
            getch();
            exit(1);
        }
    /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;
    setcolor(getmaxcolor());

    /* draw ellipse */
    ellipse(midx, midy, stangle, endangle,
        xradius, yradius);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

函数名: enable

功 能: 开放硬件中断

用 法: void enable(void);

程序例:

```
/* ** NOTE:
This is an interrupt service routine. You can NOT compile this program
with Test Stack Overflow turned on and get an executable file which will
operate correctly.
*/

#include <stdio.h>
#include <dos.h>
#include <conio.h>

/* The clock tick interrupt */
#define INTR 0X1C

void interrupt ( *oldhandler)(void);

int count=0;

void interrupt handler(void)
{
/*
    disable interrupts during the handling of the interrupt
*/
    disable();
/* increase the global counter */
    count++;
/*
    re enable interrupts at the end of the handler
*/
    enable();
/* call the old routine */
    oldhandler();
}

int main(void)
{
/* save the old interrupt vector */
    oldhandler = getvect(INTR);

/* install the new interrupt handler */
    setvect(INTR, handler);
```

```
/* loop until the counter exceeds 20 */
while (count < 20)
    printf("count is %d\n",count);

/* reset the old interrupt handler */
setvect(INTR, oldhandler);

return 0;
}
```

函数名: eof

功 能: 检测文件结束

用 法: int eof(int *handle);

程序例:

```
#include <sys\stat.h>
#include <string.h>
#include <stdio.h>
#include <fcntl.h>
#include <io.h>

int main(void)
{
    int handle;
    char msg[] = "This is a test";
    char ch;

    /* create a file */
    handle = open("DUMMY.FIL",
        O_CREAT | O_RDWR,
        S_IREAD | S_IWRITE);

    /* write some data to the file */
    write(handle, msg, strlen(msg));

    /* seek to the beginning of the file */
    lseek(handle, 0L, SEEK_SET);

    /*
        reads chars from the file until hit EOF
    */
}
```



```
do
{
    read(handle, &ch, 1);
    printf("%c", ch);
} while (!eof(handle));

close(handle);
return 0;
}
```

函数名: exec...

功 能: 装入并运行其它程序的函数

用 法: int exec(*char *pathname*, *char *arg0*, *arg1*, ..., *argn*, NULL);

int execl(*char *pathname*, *char *arg0*, *arg1*, ..., *argn*, NULL,
 *char *envp*[]);

int execlp(*char *pathname*, *char *arg0*, *arg1*, ..., NULL);

int execlpe(*char *pathname*, *char *arg0*, *arg1*, ..., NULL,
 *char *envp*[]);

int execv(*char *pathname*, *char *argv*[]);

int execve(*char *pathname*, *char *argv*[], *char *envp*[]);

int execvp(*char *pathname*, *char *argv*[]);

int execvpe(*char *pathname*, *char *argv*[], *char *envp*[]);

程序例:

```
/* execv example */
```

```
#include <process.h>
```

```
#include <stdio.h>
```

```
#include <errno.h>
```

```
void main(int argc, char *argv[])
```

```
{
```

```
    int i;
```

```
    printf("Command line arguments:\n");
```

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

```
        printf("[%2d] : %s\n", i, argv[i]);
```

```
    printf("About to exec child with arg1 arg2 ... \n");
```

```
    execv("CHILD.EXE", argv);
```

```
    perror("exec error");
```

```
    exit(1);  
}
```

函数名: exit

功 能: 终止程序

用 法: void exit(int status);

程序例:

```
#include <stdlib.h>  
#include <conio.h>  
#include <stdio.h>
```

```
int main(void)  
{  
    int status;  
  
    printf("Enter either 1 or 2\n");  
    status = getch();  
    /* Sets DOS errorlevel */  
    exit(status - '0');  
  
    /* Note: this line is never reached */  
    return 0;  
}
```

函数名: exp

功 能: 指数函数

用 法: double exp(double x);

程序例:

```
#include <stdio.h>  
#include <math.h>
```

```
int main(void)  
{  
    double result;  
    double x = 4.0;  
  
    result = exp(x);  
    printf("'e' raised to the power \
```

```
    of %lf (e ^ %lf) = %lf\n",
    x, x, result);

    return 0;
}
*****
```

函数名: fabs

功 能: 返回浮点数的绝对值

用 法: double fabs(double x);

程序例:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    float  number = -1234.0;

    printf("number: %f  absolute value: %f\n",
    number, fabs(number));
    return 0;
}
```

函数名: farcalloc

功 能: 从远堆栈中申请空间

用 法: void far *farcalloc(unsigned long units, unsigned long unitsz);

程序例:

```
#include <stdio.h>
#include <alloc.h>
#include <string.h>
#include <dos.h>

int main(void)
{
    char far *fptr;
    char *str = "Hello";

    /* allocate memory for the far pointer */
    fptr = farcalloc(10, sizeof(char));

    /* copy "Hello" into allocated memory */
```

```
/*
    Note: movedata is used because you
    might be in a small data model, in
    which case a normal string copy routine
    can not be used since it assumes the
    pointer size is near.
*/
movedata(FP_SEG(str), FP_OFF(str),
        FP_SEG(fptr), FP_OFF(fptr),
        strlen(str));

/* display string (note the F modifier) */
printf("Far string is: %Fs\n", fptr);

/* free the memory */
farfree(fptr);

return 0;
}
```

函数名: farcoreleft

功 能: 返回远堆中未作用存储区大小

用 法: long farcoreleft(void);

程序例:

```
#include <stdio.h>
#include <alloc.h>

int main(void)
{
    printf("The difference between the\
    highest allocated block in the\
    far\n");
    printf("heap and the top of the far heap\
    is: %lu bytes\n", farcoreleft());

    return 0;
}
```

函数名: farfree

功 能: 从远堆中释放一块

用 法: void farfree(void);

程序例:

```
#include <stdio.h>
#include <alloc.h>
#include <string.h>
#include <dos.h>

int main(void)
{
    char far *fptr;
    char *str = "Hello";

    /* allocate memory for the far pointer */
    fptr = farcalloc(10, sizeof(char));

    /* copy "Hello" into allocated memory */
    /*
       Note: movedata is used because you might be in a small data model,
       in which case a normal string copy routine can't be used since it
       assumes the pointer size is near.
    */
    movedata(FP_SEG(str), FP_OFF(str),
             FP_SEG(fptr), FP_OFF(fptr),
             strlen(str));

    /* display string (note the F modifier) */
    printf("Far string is: %Fs\n", fptr);

    /* free the memory */
    farfree(fptr);

    return 0;
}
```

函数名: farmalloc

功 能: 从远堆中分配存储块

用 法: void far *farmalloc(unsigned long size);

程序例:

```
#include <stdio.h>
#include <alloc.h>
#include <string.h>
#include <dos.h>

int main(void)
{
    char far *fptr;
    char *str = "Hello";

    /* allocate memory for the far pointer */
    fptr = farmalloc(10);

    /* copy "Hello" into allocated memory */
    /*
       Note: movedata is used because we might
       be in a small data model, in which case
       a normal string copy routine can not be
       used since it assumes the pointer size
       is near.
    */
    movedata(FP_SEG(str), FP_OFF(str),
             FP_SEG(fptr), FP_OFF(fptr),
             strlen(str));

    /* display string (note the F modifier) */
    printf("Far string is: %Fs\n", fptr);

    /* free the memory */
    farfree(fptr);

    return 0;
}
```

函数名: farrealloc

功 能: 调整远堆中的分配块

用 法: void far *farrealloc(void far *block, unsigned long newsize);

程序例:

```
#include <stdio.h>
#include <alloc.h>

int main(void)
{
    char far *fptr;

    fptr = farmalloc(10);
    printf("First address: %Fp\n", fptr);
    fptr = farrealloc(fptr,20);
    printf("New address   : %Fp\n", fptr);
    farfree(fptr);
    return 0;
}
```

函数名: fclose

功 能: 关闭一个流

用 法: int fclose(FILE *stream);

程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    FILE *fp;
    char buf[11] = "0123456789";

    /* create a file containing 10 bytes */
    fp = fopen("DUMMY.FIL", "w");
    fwrite(&buf, strlen(buf), 1, fp);

    /* close the file */
    fclose(fp);
    return 0;
}
```

函数名: fcloseall

功 能: 关闭打开流

用 法: int fcloseall(void);

程序例:

```
#include <stdio.h>

int main(void)
{
    int streams_closed;

    /* open two streams */
    fopen("DUMMY.ONE", "w");
    fopen("DUMMY.TWO", "w");

    /* close the open streams */
    streams_closed = fcloseall();

    if (streams_closed == EOF)
        /* issue an error message */
        perror("Error");
    else
        /* print result of fcloseall() function */
        printf("%d streams were closed.\n", streams_closed);

    return 0;
}
```

函数名: fcvt

功 能: 把一个浮点数转换为字符串

用 法: char *fcvt(double value, int ndigit, int *decpt, int *sign);

程序例:

```
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    char *string;
    double value;
    int dec, sign;
    int ndig = 10;
```



```
clrscr();
value = 9.876;
string = ecvt(value, ndig, &dec, &sign);
printf("string = %s      dec = %d \
      sign = %d\n", string, dec, sign);

value = -123.45;
ndig= 15;
string = ecvt(value, ndig, &dec, &sign);
printf("string = %s dec = %d sign = %d\n",
      string, dec, sign);

value = 0.6789e5; /* scientific
                  notation */
ndig = 5;
string = ecvt(value, ndig, &dec, &sign);
printf("string = %s      dec = %d \
      sign = %d\n", string, dec, sign);

return 0;
}
```

函数名: fdopen

功 能: 把流与一个文件句柄相接

用 法: FILE *fdopen(int handle, char *type);

程序例:

```
#include <sys\stat.h>
#include <stdio.h>
#include <fcntl.h>
#include <io.h>

int main(void)
{
    int handle;
    FILE *stream;

    /* open a file */
    handle = open("DUMMY.FIL", O_CREAT,
        S_IREAD | S_IWRITE);
```

```
/* now turn the handle into a stream */
stream = fdopen(handle, "w");

if (stream == NULL)
    printf("fdopen failed\n");
else
{
    fprintf(stream, "Hello world\n");
    fclose(stream);
}
return 0;
}
```

函数名: feof

功 能: 检测流上的文件结束符

用 法: int feof(FILE *stream);

程序例:

```
#include <stdio.h>

int main(void)
{
    FILE *stream;

    /* open a file for reading */
    stream = fopen("DUMMY.FIL", "r");

    /* read a character from the file */
    fgetc(stream);

    /* check for EOF */
    if (feof(stream))
        printf("We have reached end-of-file\n");

    /* close the file */
    fclose(stream);
    return 0;
}
```

函数名: `ferror`

功 能: 检测流上的错误

用 法: `int ferror(FILE *stream);`

程序例:

```
#include <stdio.h>

int main(void)
{
    FILE *stream;

    /* open a file for writing */
    stream = fopen("DUMMY.FIL", "w");

    /* force an error condition by attempting to read */
    (void) getc(stream);

    if (ferror(stream)) /* test for an error on the stream */
    {
        /* display an error message */
        printf("Error reading from DUMMY.FIL\n");

        /* reset the error and EOF indicators */
        clearerr(stream);
    }

    fclose(stream);
    return 0;
}
```

函数名: `fflush`

功 能: 清除一个流

用 法: `int fflush(FILE *stream);`

程序例:

```
#include <string.h>
#include <stdio.h>
#include <conio.h>
#include <io.h>

void flush(FILE *stream);
```

```
int main(void)
{
    FILE *stream;
    char msg[] = "This is a test";

    /* create a file */
    stream = fopen("DUMMY.FIL", "w");

    /* write some data to the file */
    fwrite(msg, strlen(msg), 1, stream);

    clrscr();
    printf("Press any key to flush\
DUMMY.FIL:");
    getch();

    /* flush the data to DUMMY.FIL without\
closing it */
    flush(stream);

    printf("\nFile was flushed, Press any key\
to quit:");
    getch();
    return 0;
}

void flush(FILE *stream)
{
    int duphandle;

    /* flush the stream's internal buffer */
    fflush(stream);

    /* make a duplicate file handle */
    duphandle = dup(fileno(stream));

    /* close the duplicate handle to flush\
the DOS buffer */
    close(duphandle);
}
```

函数名: fgetc

功 能: 从流中读取字符

用 法: int fgetc(FILE *stream);

程序例:

```
#include <string.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    FILE *stream;
    char string[] = "This is a test";
    char ch;

    /* open a file for update */
    stream = fopen("DUMMY.FIL", "w+");

    /* write a string into the file */
    fwrite(string, strlen(string), 1, stream);

    /* seek to the beginning of the file */
    fseek(stream, 0, SEEK_SET);

    do
    {
        /* read a char from the file */
        ch = fgetc(stream);

        /* display the character */
        putchar(ch);
    } while (ch != EOF);

    fclose(stream);
    return 0;
}
```

函数名: fgetchar

功 能: 从流中读取字符

用 法: int fgetchar(void);

程序例:

```
#include <stdio.h>

int main(void)
{
    char ch;

    /* prompt the user for input */
    printf("Enter a character followed by \
    <Enter>: ");

    /* read the character from stdin */
    ch = fgetchar();

    /* display what was read */
    printf("The character read is: '%c'\n",
           ch);
    return 0;
}
```

函数名: fgetpos

功 能: 取得当前文件的句柄

用 法: int fgetpos(FILE *stream);

程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    FILE *stream;
    char string[] = "This is a test";
    fpos_t filepos;

    /* open a file for update */
    stream = fopen("DUMMY.FIL", "w+");

    /* write a string into the file */
    fwrite(string, strlen(string), 1, stream);
```

```
/* report the file pointer position */
fgetpos(stream, &filepos);
printf("The file pointer is at byte\
      %ld\n", filepos);

fclose(stream);
return 0;
}
```

函数名: fgets

功 能: 从流中读取一字符串

用 法: char *fgets(char *string, int n, FILE *stream);

程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    FILE *stream;
    char string[] = "This is a test";
    char msg[20];

    /* open a file for update */
    stream = fopen("DUMMY.FIL", "w+");

    /* write a string into the file */
    fwrite(string, strlen(string), 1, stream);

    /* seek to the start of the file */
    fseek(stream, 0, SEEK_SET);

    /* read a string from the file */
    fgets(msg, strlen(string)+1, stream);

    /* display the string */
    printf("%s", msg);

    fclose(stream);
    return 0;
}
```

函数名: filelength

功 能: 取文件长度字节数

用 法: long filelength(int handle);

程序例:

```
#include <string.h>
#include <stdio.h>
#include <fcntl.h>
#include <io.h>

int main(void)
{
    int handle;
    char buf[11] = "0123456789";

    /* create a file containing 10 bytes */
    handle = open("DUMMY.FIL", O_CREAT);
    write(handle, buf, strlen(buf));

    /* display the size of the file */
    printf("file length in bytes: %ld\n",
        filelength(handle));

    /* close the file */
    close(handle);
    return 0;
}
```

函数名: fillellipse

功 能: 画出并填充一椭圆

用 法: void far fillellipse(int x, int y, int xradius, int yradius);

程序例:

```
#include <graphics.h>
#include <conio.h>
```

```
int main(void)
{
```



```
int gdriver = DETECT, gmode;  
int xcenter, ycenter, i;  
  
initgraph(&gdriver, &gmode, "");  
xcenter = getmaxx() / 2;  
ycenter = getmaxy() / 2;  
  
for (i=0; i<13; i++)  
{  
    setfillstyle(i, WHITE);  
    fillellipse(xcenter, ycenter, 100, 50);  
    getch();  
}  
  
closegraph();  
return 0;  
}
```

函数名: fillpoly

功 能: 画并填充一个多边形

用 法: void far fillpoly(int numpoints, int far *polypoints);

程序例:

```
#include <graphics.h>  
#include <stdlib.h>  
#include <stdio.h>  
#include <conio.h>  
  
int main(void)  
{  
    /* request auto detection */  
    int gdriver = DETECT, gmode, errorcode;  
    int i, maxx, maxy;  
  
    /* our polygon array */  
    int poly[8];  
  
    /* initialize graphics, local variables */  
    initgraph(&gdriver, &gmode, "");  
  
    /* read result of initialization */
```

```
errorcode = graphresult();
if (errorcode != grOk)
/* an error occurred */
{
    printf("Graphics error: %s\n",
           grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1);
    /* terminate with an error code */
}

maxx = getmaxx();
maxy = getmaxy();

poly[0] = 20;          /* 1st vertex */
poly[1] = maxy / 2;

poly[2] = maxx - 20; /* 2nd */
poly[3] = 20;

poly[4] = maxx - 50; /* 3rd */
poly[5] = maxy - 20;

/*
    4th vertex. fillpoly automatically
    closes the polygon.
*/
poly[6] = maxx / 2;
poly[7] = maxy / 2;

/* loop through the fill patterns */
for (i=EMPTY_FILL; i<USER_FILL; i++)
{
    /* set fill pattern */
    setfillstyle(i, getmaxcolor());

    /* draw a filled polygon */
    fillpoly(4, poly);

    getch();
}

/* clean up */
```

```
    closegraph();
    return 0;
}
```

函数名: findfirst, findnext

功 能: 搜索磁盘目录; 取得下一个匹配的 findfirst 模式的文件

用 法: int findfirst(char *pathname, struct fblk *fblk, int attrib);
int findnext(struct fblk *fblk);

程序例:

```
/* findnext example */
```

```
#include <stdio.h>
```

```
#include <dir.h>
```

```
int main(void)
```

```
{
    struct fblk fblk;
    int done;
    printf("Directory listing of *. *\n");
    done = findfirst("*. *",&fblk,0);
    while (!done)
    {
        printf("  %s\n", fblk.ff_name);
        done = findnext(&fblk);
    }

    return 0;
}
```

函数名: floodfill

功 能: 填充一个有界区域

用 法: void far floodfill(int x, int y, int border);

程序例:

```
#include <graphics.h>
```

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int maxx, maxy;

    /* initialize graphics, local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)
        /* an error occurred */
        {
            printf("Graphics error: %s\n",
                grapherrormsg(errorcode));
            printf("Press any key to halt:");
            getch();
            exit(1);
            /* terminate with an error code */
        }

    maxx = getmaxx();
    maxy = getmaxy();

    /* select drawing color */
    setcolor(getmaxcolor());

    /* select fill color */
    setfillstyle(SOLID_FILL, getmaxcolor());

    /* draw a border around the screen */
    rectangle(0, 0, maxx, maxy);

    /* draw some circles */
    circle(maxx / 3, maxy / 2, 50);
    circle(maxx / 2, 20, 100);
    circle(maxx-20, maxy-50, 75);
    circle(20, maxy-20, 25);

    /* wait for a key */
    getch();
}
```

```
/* fill in bounded region */
floodfill(2, 2, getmaxcolor());

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: floor

功 能: 向下舍入

用 法: double floor(double x);

程序例:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    double number = 123.54;
    double down, up;

    down = floor(number);
    up = ceil(number);

    printf("original number      %10.2lf\n",
           number);
    printf("number rounded down %10.2lf\n",
           down);
    printf("number rounded up   %10.2lf\n",
           up);

    return 0;
}
```

函数名: flushall

功 能: 清除所有缓冲区

用 法: int fflush(void);

程序例:

```
#include <stdio.h>

int main(void)
{
    FILE *stream;

    /* create a file */
    stream = fopen("DUMMY.FIL", "w");

    /* flush all open streams */
    printf("%d streams were flushed.\n",
        fflush());

    /* close the file */
    fclose(stream);
    return 0;
}
```

函数名: fmod

功 能: 计算 x 对 y 的模, 即 x/y 的余数

用 法: double fmod(double x, double y);

程序例:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    double x = 5.0, y = 2.0;
    double result;

    result = fmod(x,y);
    printf("The remainder of (%lf / %lf) is \
        %lf\n", x, y, result);
    return 0;
}
```

函数名: fnmerge

功 能: 建立新文件名

用 法: void fnmerge(char *path, char *drive, char *dir);

程序例:

```
#include <string.h>
#include <stdio.h>
#include <dir.h>

int main(void)
{
    char s[MAXPATH];
    char drive[MAXDRIVE];
    char dir[MAXDIR];
    char file[MAXFILE];
    char ext[MAXEXT];

    getcwd(s,MAXPATH);           /* get the current working directory */
    strcat(s,"\\");              /* append on a trailing \ character */
    fnsplit(s,drive,dir,file,ext); /* split the string to separate elems */
    strcpy(file,"DATA");
    strcpy(ext,".TXT");
    fnmerge(s,drive,dir,file,ext); /* merge everything into one string */
    puts(s);                    /* display resulting string */

    return 0;
}
```

函数名: fopen

功 能: 打开一个流

用 法: FILE *fopen(char *filename, char *type);

程序例:

```
#include <stdlib.h>
#include <stdio.h>
#include <dir.h>
```

```
int main(void)
```

```

{
    char *s;
    char drive[MAXDRIVE];
    char dir[MAXDIR];
    char file[MAXFILE];
    char ext[MAXEXT];
    int flags;

    s=getenv("COMSPEC"); /* get the comspec environment parameter */
    flags=fnsplit(s,drive,dir,file,ext);

    printf("Command processor info:\n");
    if(flags & DRIVE)
        printf("\tdrive: %s\n",drive);
    if(flags & DIRECTORY)
        printf("\tdirectory: %s\n",dir);
    if(flags & FILENAME)
        printf("\tfile: %s\n",file);
    if(flags & EXTENSION)
        printf("\textension: %s\n",ext);

    return 0;
}

```

函数名: fprintf

功 能: 传送格式化输出到一个流中

用 法: int fprintf(FILE *stream, char *format[, argument,...]);

程序例:

```

/* Program to create backup of the
   AUTOEXEC.BAT file */

#include <stdio.h>

int main(void)
{
    FILE *in, *out;

    if ((in = fopen("\\AUTOEXEC.BAT", "rt"))
        == NULL)
    {
        fprintf(stderr, "Cannot open input \
file.\n");
    }
}

```



```
        return 1;
    }

    if ((out = fopen("\\AUTOEXEC.BAK", "wt"))
        == NULL)
    {
        fprintf(stderr, "Cannot open output \
        file.\n");
        return 1;
    }

    while (!feof(in))
        fputc(fgetc(in), out);

    fclose(in);
    fclose(out);
    return 0;
}
```

函数名: FP_OFF

功 能: 获取远地址偏移量

用 法: unsigned FP_OFF(void far *farptr);

程序例:

```
/* FP_OFF */

#include <dos.h>
#include <stdio.h>

int main(void)
{
    char *str = "fpoff.c";

    printf("The offset of this file in memory\
    is: %Fp\n", FP_OFF(str));

    return 0;
}
```

函数名: FP_SEG

功 能: 获取远地址段值

用 法: unsigned FP_SEG(void far *farptr);

程序例:

```
/* FP_SEG */

#include <dos.h>
#include <stdio.h>

int main(void)
{
    char *filename = "fpseg.c";

    printf("The offset of this file in memory\
is: %Fp\n", FP_SEG(filename));

    return(0);
}
```

函数名: fputc

功 能: 送一个字符到一个流中

用 法: int fputc(int ch, FILE *stream);

程序例:

```
#include <stdio.h>

int main(void)
{
    char msg[] = "Hello world";
    int i = 0;

    while (msg[i])
    {
        fputc(msg[i], stdout);
        i++;
    }
    return 0;
}
```

函数名: fputc

功 能: 送一个字符到标准输出流(stdout)中

用 法: int fputc(char ch);

程序例:

```
#include <stdio.h>
```

```
int main(void)
```

```
{
    char msg[] = "This is a test";
    int i = 0;

    while (msg[i])
    {
        fputc(msg[i]);
        i++;
    }
    return 0;
}
```

函数名: fputs

功 能: 送一个字符到一个流中

用 法: int fputs(char *string, FILE *stream);

程序例:

```
#include <stdio.h>
```

```
int main(void)
```

```
{
    /* write a string to standard output */
    fputs("Hello world\n", stdout);

    return 0;
}
```

函数名: fread

功 能: 从一个流中读数据

用 法: `int fread(void *ptr, int size, int nitems, FILE *stream);`

程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    FILE *stream;
    char msg[] = "this is a test";
    char buf[20];

    if ((stream = fopen("DUMMY.FIL", "w+"))
        == NULL)
    {
        fprintf(stderr,
                "Cannot open output file.\n");
        return 1;
    }

    /* write some data to the file */
    fwrite(msg, strlen(msg)+1, 1, stream);

    /* seek to the beginning of the file */
    fseek(stream, SEEK_SET, 0);

    /* read the data and display it */
    fread(buf, strlen(msg)+1, 1, stream);
    printf("%s\n", buf);

    fclose(stream);
    return 0;
}
```

函数名: `free`

功 能: 释放已分配的块

用 法: `void free(void *ptr);`

程序例:

```
#include <string.h>
#include <stdio.h>
```

```
#include <alloc.h>

int main(void)
{
    char *str;

    /* allocate memory for string */
    str = malloc(10);

    /* copy "Hello" to string */
    strcpy(str, "Hello");

    /* display string */
    printf("String is %s\n", str);

    /* free memory */
    free(str);

    return 0;
}
```

函数名: freemem

功 能: 释放先前分配的 DOS 内存块

用 法: int freemem(unsigned seg);

程序例:

```
#include <dos.h>
#include <alloc.h>
#include <stdio.h>

int main(void)
{
    unsigned int size, segp;
    int stat;

    size = 64; /* (64 x 16) = 1024 bytes */
    stat = allocmem(size, &segp);
    if (stat < 0)
        printf("Allocated memory at segment:\n\n", segp);
    else
        printf("Failed: maximum number of\n\n", paragraphs available is %u\n",
```

```
    stat);
    freemem(seg);

    return 0;
}
```

函数名: freopen

功 能: 替换一个流

用 法: FILE *freopen(char *filename, char *type, FILE *stream);

程序例:

```
#include <stdio.h>

int main(void)
{
    /* redirect standard output to a file */
    if (freopen("OUTPUT.FIL", "w", stdout)
        == NULL)
        fprintf(stderr, "error redirecting\
            stdout\n");

    /* this output will go to a file */
    printf("This will go into a file.");

    /* close the standard output stream */
    fclose(stdout);

    return 0;
}
```

函数名: frexp

功 能: 把一个双精度数分解为尾数的指数

用 法: double frexp(double value, int *eptr);

程序例:

```
#include <math.h>
#include <stdio.h>
```

```
int main(void)
```

```
{
    double mantissa, number;
    int exponent;

    number = 8.0;
    mantissa = frexp(number, &exponent);

    printf("The number %lf is ", number);
    printf("%lf times two to the ", mantissa);
    printf("power of %d\n", exponent);

    return 0;
}
```

函数名: fscanf

功 能: 从一个流中执行格式化输入

用 法: int fscanf(FILE *stream, char *format[,argument...]);

程序例:

```
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
    int i;

    printf("Input an integer: ");

    /* read an integer from the
       standard input stream */
    if (fscanf(stdin, "%d", &i))
        printf("The integer read was: %i\n",
               i);
    else
    {
        fprintf(stderr, "Error reading an \
integer from stdin.\n");
        exit(1);
    }
    return 0;
}
```

函数名: fseek

功 能: 重定位流上的文件指针

用 法: int fseek(FILE *stream, long offset, int fromwhere);

程序例:

```
#include <stdio.h>
```

```
long filesize(FILE *stream);
```

```
int main(void)
```

```
{  
    FILE *stream;  
  
    stream = fopen("MYFILE.TXT", "w+");  
    fprintf(stream, "This is a test");  
    printf("Filesize of MYFILE.TXT is %ld bytes\n", filesize(stream));  
    fclose(stream);  
    return 0;  
}
```

```
long filesize(FILE *stream)
```

```
{  
    long curpos, length;  
  
    curpos = ftell(stream);  
    fseek(stream, 0L, SEEK_END);  
    length = ftell(stream);  
    fseek(stream, curpos, SEEK_SET);  
    return length;  
}
```

函数名: fsetpos

功 能: 定位流上的文件指针

用 法: int fsetpos(FILE *stream, const fpos_t *pos);

程序例:


```
#include <stdlib.h>
#include <stdio.h>

void showpos(FILE *stream);

int main(void)
{
    FILE *stream;
    fpos_t filepos;

    /* open a file for update */
    stream = fopen("DUMMY.FIL", "w+");

    /* save the file pointer position */
    fgetpos(stream, &filepos);

    /* write some data to the file */
    fprintf(stream, "This is a test");

    /* show the current file position */
    showpos(stream);

    /* set a new file position, display it */
    if (fsetpos(stream, &filepos) == 0)
        showpos(stream);
    else
    {
        fprintf(stderr, "Error setting file \
        pointer.\n");
        exit(1);
    }

    /* close the file */
    fclose(stream);
    return 0;
}

void showpos(FILE *stream)
{
    fpos_t pos;

    /* display the current file pointer
    position of a stream */
    fgetpos(stream, &pos);
```

```
    printf("File position: %ld\n", pos);  
}
```

函数名: fstat

功 能: 获取打开文件信息

用 法: int fstat(char *handle, struct stat *buff);

程序例:

```
#include <sys\stat.h>  
#include <stdio.h>  
#include <time.h>
```

```
int main(void)  
{  
    struct stat statbuf;  
    FILE *stream;  
  
    /* open a file for update */  
    if ((stream = fopen("DUMMY.FIL", "w+"))  
        == NULL)  
    {  
        fprintf(stderr, "Cannot open output \\  
            file.\n");  
        return(1);  
    }  
    fprintf(stream, "This is a test");  
    fflush(stream);  
  
    /* get information about the file */  
    fstat(fileno(stream), &statbuf);  
    fclose(stream);  
  
    /* display the information returned */  
    if (statbuf.st_mode & S_IFCHR)  
        printf("Handle refers to a device.\n");  
    if (statbuf.st_mode & S_IFREG)  
        printf("Handle refers to an ordinary \\  
            file.\n");  
    if (statbuf.st_mode & S_IREAD)  
        printf("User has read permission on \\  
            file.\n");  
    if (statbuf.st_mode & S_IWRITE)  
        printf("User has write permission on \\  
            file.\n");  
}
```

```
        file.\n");

printf("Drive letter of file: %c\n",
'A'+statbuf.st_dev);
printf("Size of file in bytes: %ld\n",
statbuf.st_size);
printf("Time file last opened: %s\n",
ctime(&statbuf.st_ctime));
return 0;
}
```

函数名: ftell

功 能: 返回当前文件指针

用 法: long ftell(FILE *stream);

程序例:

```
#include <stdio.h>

int main(void)
{
    FILE *stream;

    stream = fopen("MYFILE.TXT", "w+");
    fprintf(stream, "This is a test");
    printf("The file pointer is at byte \
        %ld\n", ftell(stream));
    fclose(stream);
    return 0;
}
```

函数名: fwrite

功 能: 写内容到流中

用 法: int fwrite(void *ptr, int size, int nitems, FILE *stream);

程序例:

```
#include <stdio.h>
```

```
struct mystruct
```

```
{
    int i;
    char ch;
};

int main(void)
{
    FILE *stream;
    struct mystruct s;

    if ((stream = fopen("TEST.$$$", "wb")) == NULL) /* open file TEST.$$$ */
    {
        fprintf(stderr, "Cannot open output file.\n");
        return 1;
    }
    s.i = 0;
    s.ch = 'A';
    fwrite(&s, sizeof(s), 1, stream); /* write struct s to file */
    fclose(stream); /* close file */
    return 0;
}
*****
```

函数名: gcvt

功 能: 把浮点数转换成字符串

用 法: char *gcvt(double value, int ndigit, char *buf);

程序例:

```
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
    char str[25];
    double num;
    int sig = 5; /* significant digits */

    /* a regular number */
    num = 9.876;
    gcvt(num, sig, str);
    printf("string = %s\n", str);

    /* a negative number */
    num = -123.4567;
    gcvt(num, sig, str);
```

```
printf("string = %s\n", str);

/* scientific notation */
num = 0.678e5;
gcvt(num, sig, str);
printf("string = %s\n", str);

return(0);
}
```

函数名: geninterrupt

功 能: 产生一个软中断

用 法: void geninterrupt(int intr_num);

程序例:

```
#include <conio.h>
#include <dos.h>

/* function prototype */
void writechar(char ch);

int main(void)
{
    clrscr();
    gotoxy(80,25);
    writechar('*');
    getch();
    return 0;
}

/*
    outputs a character at the current cursor
    position using the video BIOS to avoid the
    scrolling of the screen when writing to
    location (80,25).
*/

void writechar(char ch)
{
    struct text_info ti;
    /* grab current text settings */
```

```

    gettextinfo(&ti);
    /* interrupt 0x10 sub-function 9 */
    _AH = 9;
    /* character to be output */
    _AL = ch;
    _BH = 0;                /* video page */
    _BL = ti.attribute;    /* video attribute */
    _CX = 1;                /* repetition factor */
    geninterrupt(0x10);    /* output the char */
}

```

函数名: getarccoords

功 能: 取得最后一次调用 arc 的坐标

用 法: void far getarccoords(struct arccoordstype far *arccoords);

程序例:

```

#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    struct arccoordstype arcinfo;
    int midx, midy;
    int stangle = 45, endangle = 270;
    charsstr[80], estr[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    /* an error occurred */
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n",
               grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
    }
}

```

```
/* terminate with an error code */
    exit(1);
}

midx = getmaxx() / 2;
midy = getmaxy() / 2;

/* draw arc and get coordinates */
setcolor(getmaxcolor());
arc(midx, midy, stangle, endangle, 100);
getarccoords(&arcinfo);

/* convert arc information into strings */
sprintf(sstr, "%d %d",
        arcinfo.xstart, arcinfo.ystart);
sprintf(estr, "%d %d",
        arcinfo.xend, arcinfo.yend);

/* output the arc information */
outtextxy(arcinfo.xstart,
          arcinfo.ystart, sstr);
outtextxy(arcinfo.xend,
          arcinfo.yend, estr);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: getspectratio

功 能: 返回当前图形模式的纵横比

用 法: void far getspectratio(int far *xasp, int far *yasp);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
```

```
int main(void)
```

```
{
/* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int xasp, yasp, midx, midy;

/* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

/* read result of initialization */
    errorcode = graphresult();
/* an error occurred */
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n",
            grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
/* terminate with an error code */
        exit(1);
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;
    setcolor(getmaxcolor());

/* get current aspect ratio settings */
    getaspectratio(&xasp, &yasp);

/* draw normal circle */
    circle(midx, midy, 100);
    getch();

/* draw wide circle */
    cleardevice();
    setaspectratio(xasp/2, yasp);
    circle(midx, midy, 100);
    getch();

/* draw narrow circle */
    cleardevice();
    setaspectratio(xasp, yasp/2);
    circle(midx, midy, 100);

/* clean up */
```



```
    getch();
    closegraph();
    return 0;
}
```

函数名: getbkcolor

功 能: 返回当前背景颜色

用 法: int far getbkcolor(void);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int bkcolor, midx, midy;
    char bkname[35];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    /* an error occurred */
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n",
               grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
    }
    /* terminate with an error code */
    exit(1);
}

midx = getmaxx() / 2;
midy = getmaxy() / 2;
```

```
    setcolor(getmaxcolor());

/* for centering text on the display */
    setttextjustify(CENTER_TEXT, CENTER_TEXT);

/* get the current background color */
    bkcolor = getbkcolor();

/* convert color value into a string */
    itoa(bkcolor, bkname, 10);
    strcat(bkname,
        " is the current background color.");

/* display a message */
    outtextxy(midx, midy, bkname);

/* clean up */
    getch();
    closegraph();
    return 0;
}
```

函数名: `getc`

功 能: 从流中取字符

用 法: `int getc(FILE *stream);`

程序例:

```
#include <stdio.h>

int main(void)
{
    char ch;

    printf("Input a character:");
/* read a character from the
   standard input stream */
    ch = getc(stdin);
    printf("The character input was: '%c'\n",
        ch);
    return 0;
}
```

函数名: `getcbrk`

功 能: 获取 `Control_break` 设置

用 法: `int getcbrk(void);`

程序例:

```
#include <stdio.h>
#include <dos.h>

int main(void)
{
    if (getcbrk())
        printf("Cntrl-brk flag is on\n");
    else
        printf("Cntrl-brk flag is off\n");

    return 0;
}
```

函数名: `getch`

功 能: 从控制台无回显地取一个字符

用 法: `int getch(void);`

程序例:

```
#include <stdio.h>
#include <conio.h>

int main(void)
{
    char ch;

    printf("Input a character:");
    ch = getch();
    printf("\nYou input a '%c'\n", ch);
    return 0;
}
```

函数名: `getchar`

功 能: 从 `stdin` 流中读字符

用 法: `int getchar(void);`

程序例:

```
#include <stdio.h>

int main(void)
{
    int c;

    /* Note that getchar reads from stdin and
       is line buffered; this means it will
       not return until you press ENTER. */

    while ((c = getchar()) != '\n')
        printf("%c", c);

    return 0;
}
```

函数名: `getche`

功 能: 从控制台取字符(带回显)

用 法: `int getche(void);`

程序例:

```
#include <stdio.h>
#include <conio.h>

int main(void)
{
    char ch;

    printf("Input a character:");
    ch = getche();
    printf("\nYou input a '%c'\n", ch);
    return 0;
}
```

函数名: `getcolor`

功 能: 返回当前画线颜色

用 法: int far getcolor(void);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int color, midx, midy;
    char colname[35];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    /* an error occurred */
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n",
            grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
    }
    /* terminate with an error code */
    exit(1);
}

midx = getmaxx() / 2;
midy = getmaxy() / 2;
setcolor(getmaxcolor());

/* for centering text on the display */
settextjustify(CENTER_TEXT, CENTER_TEXT);

/* get the current drawing color */
color = getcolor();

/* convert color value into a string */
```

```
    itoa(color, colname, 10);
    strtat(colname,
    " is the current drawing color.");

/* display a message */
    outtextxy(midx, midy, colname);

/* clean up */
    getch();
    closegraph();
    return 0;
}
```

函数名: getcurdir

功 能: 取指定驱动器的当前目录

用 法: int getcurdir(int drive, char *direc);

程序例:

```
#include <dir.h>
#include <stdio.h>
#include <string.h>

char *current_directory(char *path)
{
    strcpy(path, "X:\\"); /* fill string with form of response: X:\\ */
    path[0] = 'A' + getdisk(); /* replace X with current drive letter */
    getcurdir(0, path+3); /* fill rest of string with current directory */
    return(path);
}

int main(void)
{
    char curdir[MAXPATH];

    current_directory(curdir);
    printf("The current directory is %s\n", curdir);

    return 0;
}
```

函数名: getcwd

功 能: 取当前工作目录

用 法: char *getcwd(char *buf, int n);

程序例:

```
#include <stdio.h>
#include <dir.h>

int main(void)
{
    char buffer[MAXPATH];

    getcwd(buffer, MAXPATH);
    printf("The current directory is: %s\n", buffer);
    return 0;
}
```

函数名: getdate

功 能: 取 DOS 日期

用 法: void getdate(struct *dateblk);

程序例:

```
#include <dos.h>
#include <stdio.h>

int main(void)
{
    struct date d;

    getdate(&d);
    printf("The current year is: %d\n",
        d.da_year);
    printf("The current day is: %d\n",
        d.da_day);
    printf("The current month is: %d\n",
        d.da_mon);
    return 0;
}
```

函数名: getdefaultpalette

功 能: 返回调色板定义结构

用 法: struct palettetype *far getdefaultpalette(void);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int i;

    /* structure for returning palette copy */
    struct palettetype far *pal=(void *) 0;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    /* an error occurred */
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n",
               grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
    }
    /* terminate with an error code */
    exit(1);
}

setcolor(getmaxcolor());

/* return a pointer to the default palette */
pal = getdefaultpalette();

for (i=0; i<16; i++)
{
    printf("colors[%d]= %d\n", i,
           pal->colors[i]);
    getch();
}
```



```
    }

/* clean up */
    getch();
    closegraph();
    return 0;
}
```

函数名: getdisk

功 能: 取当前磁盘驱动器号

用 法: int getdisk(void);

程序例:

```
#include <stdio.h>
#include <dir.h>

int main(void)
{
    int disk;

    disk = getdisk() + 'A';
    printf("The current drive is: %c\n",
        disk);
    return 0;
}
```

函数名: getdrivename

功 能: 返回指向包含当前图形驱动程序名字的字符串指针

用 法: char *getdrivename(void);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
```

```
int gdriver = DETECT, gmode, errorcode;

/* stores the device driver name */
char *drivename;

/* initialize graphics and local variables */
initgraph(&gdriver, &gmode, "");

/* read result of initialization */
errorcode = graphresult();
/* an error occurred */
if (errorcode != grOk)
{
    printf("Graphics error: %s\n",
           grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
/* terminate with an error code */
    exit(1);
}

setcolor(getmaxcolor());

/* get name of the device driver in use */
drivename = getdrivename();

/* for centering text on the screen */
settextjustify(CENTER_TEXT, CENTER_TEXT);

/* output the name of the driver */
outtextxy(getmaxx() / 2, getmaxy() / 2,
          drivename);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: getdta

功 能: 取磁盘传输地址

用 法: char far *getdta(void);

程序例:

```
#include <dos.h>
#include <stdio.h>

int main(void)
{
    char far *dta;

    dta = getdta();
    printf("The current disk transfer \
address is: %Fp\n", dta);
    return 0;
}
```

函数名: getenv

功 能: 从环境中取字符串

用 法: char *getenv(char *envvar);

程序例:

```
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
    char *s;

    s=getenv("COMSPEC");          /* get the comspec environment parameter */
    printf("Command processor: %s\n",s);  /* display comspec parameter */

    return 0;
}
```

函数名: getfat, getfatd

功 能: 取文件分配表信息

用 法: void getfat(int drive, struct fatinfo *fatblkp);

程序例:

```
#include <stdio.h>
#include <dos.h>

int main(void)
{
    struct fatinfo diskinfo;
    int flag = 0;

    printf("Please insert disk in drive A\n");
    getchar();

    getfat(1, &diskinfo);
    /* get drive information */

    printf("\nDrive A: is ");
    switch((unsigned char) diskinfo.fi_fatid)
    {
        case 0xFD:
            printf("360K low density\n");
            break;

        case 0xF9:
            printf("1.2 Meg high density\n");
            break;

        default:
            printf("unformatted\n");
            flag = 1;
    }

    if (!flag)
    {
        printf("    sectors per cluster %5d\n",
            diskinfo.fi_sclus);
        printf("    number of clusters %5d\n",
            diskinfo.fi_nclus);
        printf("    bytes per sector %5d\n",
            diskinfo.fi_bysec);
    }

    return 0;
}
```

函数名: getfillpattern

功 能: 将用户定义的填充模式拷贝到内存中

用 法: void far getfillpattern(char far *upattern);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int maxx, maxy;
    char pattern[8] = {0x00, 0x70, 0x20, 0x27, 0x25, 0x27, 0x04, 0x04};

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    maxx = getmaxx();
    maxy = getmaxy();
    setcolor(getmaxcolor());

    /* select a user defined fill pattern */
    setfillpattern(pattern, getmaxcolor());

    /* fill the screen with the pattern */
    bar(0, 0, maxx, maxy);

    getch();

    /* get the current user defined fill pattern */
```

```
getfillpattern(pattern);

/* alter the pattern we grabbed */
pattern[4] -= 1;
pattern[5] -= 3;
pattern[6] += 3;
pattern[7] -= 4;

/* select our new pattern */
setfillpattern(pattern, getmaxcolor());

/* fill the screen with the new pattern */
bar(0, 0, maxx, maxy);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: getfillsettings

功 能: 取得有关当前填充模式和填充颜色的信息

用 法: void far getfillsettings(struct fillsettingstype far *fillinfo);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

/ the names of the fill styles supported */
char *fname[] = { "EMPTY_FILL",
                  "SOLID_FILL",
                  "LINE_FILL",
                  "LTSLASH_FILL",
                  "SLASH_FILL",
                  "BKSLASH_FILL",
                  "LTBKSLASH_FILL",
                  "HATCH_FILL",
                  "XHATCH_FILL",
                  "INTERLEAVE_FILL",
                  "WIDE_DOT_FILL",
```

```
        "CLOSE_DOT_FILL",
        "USER_FILL"
    };

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    struct fillsettingstype fillinfo;
    int midx, midy;
    char patstr[40], colstr[40];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* get information about current fill pattern and color */
    getfillsettings(&fillinfo);

    /* convert fill information into strings */
    sprintf(patstr, "%s is the fill style.", fname[fillinfo.pattern]);
    sprintf(colstr, "%d is the fill color.", fillinfo.color);

    /* display the information */
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    outtextxy(midx, midy, patstr);
    outtextxy(midx, midy+2*textheight("W"), colstr);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

```
}
```

函数名: `getftime`

功 能: 取文件日期和时间

用 法: `int getftime(int handle, struct ftime *ftimep);`

程序例:

```
#include <stdio.h>
```

```
#include <io.h>
```

```
int main(void)
```

```
{
```

```
    FILE *stream;
```

```
    struct ftime ft;
```

```
    if ((stream = fopen("TEST.$$$",  
        "wt")) == NULL)
```

```
    {
```

```
        fprintf(stderr,
```

```
            "Cannot open output file.\n");
```

```
        return 1;
```

```
    }
```

```
    getftime(fileno(stream), &ft);
```

```
    printf("File time: %u:%u:%u\n",
```

```
        ft.ft_hour, ft.ft_min,
```

```
        ft.ft_tsec * 2);
```

```
    printf("File date: %u/%u/%u\n",
```

```
        ft.ft_month, ft.ft_day,
```

```
        ft.ft_year+1980);
```

```
    fclose(stream);
```

```
    return 0;
```

```
}
```

函数名: `getgraphmode`

功 能: 返回当前图形模式

用 法: `int far getgraphmode(void);`

程序例:


```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy, mode;
    char numname[80], modename[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    /* an error occurred */
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n",
               grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
    }
    /* terminate with an error code */
    exit(1);
}

midx = getmaxx() / 2;
midy = getmaxy() / 2;

/* get mode number and name strings */
mode = getgraphmode();
sprintf(numname,
        "%d is the current mode number.",
        mode);
sprintf(modename,
        "%s is the current graphics mode",
        getmodename(mode));

/* display the information */
settextjustify(CENTER_TEXT, CENTER_TEXT);
outtextxy(midx, midy, numname);
outtextxy(midx, midy+2*textheight("W"),
```

```
        modename);

/* clean up */
    getch();
    closegraph();
    return 0;
}
```

函数名: getftime

功 能: 取文件日期和时间

用 法: int getftime(int handle, struct ftime *ftimep);

程序例:

```
#include <stdio.h>
#include <io.h>

int main(void)
{
    FILE *stream;
    struct ftime ft;

    if ((stream = fopen("TEST.$$$",
        "wt")) == NULL)
    {
        fprintf(stderr,
            "Cannot open output file.\n");
        return 1;
    }
    getftime(fileno(stream), &ft);
    printf("File time: %u:%u:%u\n",
        ft.ft_hour, ft.ft_min,
        ft.ft_tsec * 2);
    printf("File date: %u/%u/%u\n",
        ft.ft_month, ft.ft_day,
        ft.ft_year+1980);
    fclose(stream);
    return 0;
}
```

函数名: getgraphmode

功 能: 返回当前图形模式

用 法: int far getgraphmode(void);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy, mode;
    char numname[80], modename[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    /* an error occurred */
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n",
            grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
    }
    /* terminate with an error code */
    exit(1);
}

midx = getmaxx() / 2;
midy = getmaxy() / 2;

/* get mode number and name strings */
mode = getgraphmode();
sprintf(numname,
    "%d is the current mode number.",
    mode);
sprintf(modename,
    "%s is the current graphics mode",
    getmodename(mode));
```

```
/* display the information */
    settxtjustify(CENTER_TEXT, CENTER_TEXT);
    outtextxy(midx, midy, numname);
    outtextxy(midx, midy+2*textheight("W"),
              modename);

/* clean up */
    getch();
    closegraph();
    return 0;
}
```

函数名: getimage

功 能: 将指定区域的一个位图存到主存中

用 法: void far getimage(int left, int top, int right, int bottom,
void far *bitmap);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <alloc.h>
```

```
void save_screen(void far *buf[4]);
void restore_screen(void far *buf[4]);
```

```
int maxx, maxy;
```

```
int main(void)
{
    int gdriver=DETECT, gmode, errorcode;
    void far *ptr[4];

    /* auto-detect the graphics driver and mode */
    initgraph(&gdriver, &gmode, "");
    errorcode = graphresult(); /* check for any errors */
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
    }
}
```

```
    getch();
    exit(1);
}
maxx = getmaxx();
maxy = getmaxy();

/* draw an image on the screen */
rectangle(0, 0, maxx, maxy);
line(0, 0, maxx, maxy);
line(0, maxy, maxx, 0);

save_screen(ptr);    /* save the current screen */
getch();             /* pause screen */
cleardevice();        /* clear screen */
restore_screen(ptr); /* restore the screen */
getch();             /* pause screen */

closegraph();
return 0;
}

void save_screen(void far *buf[4])
{
    unsigned size;
    int ystart=0, yend, yincr, block;

    yincr = (maxy+1) / 4;
    yend = yincr;
    size = imagesize(0, ystart, maxx, yend); /* get byte size of image */

    for (block=0; block<=3; block++)
    {
        if ((buf[block] = farmalloc(size)) == NULL)
        {
            closegraph();
            printf("Error: not enough heap space in save_screen().\n");
            exit(1);
        }

        getimage(0, ystart, maxx, yend, buf[block]);
        ystart = yend + 1;
        yend += yincr + 1;
    }
}
```

```
void save_screen(void far *buf[4])
{
    unsigned size;
    int ystart=0, yend, yincr, block;

    yincr = (maxy+1) / 4;
    yend = yincr;
    size = imagesize(0, ystart, maxx, yend); /* get byte size of image */

    for (block=0; block<=3; block++)
    {
        if ((buf[block] = farmalloc(size)) == NULL)
        {
            closegraph();
            printf("Error: not enough heap space in save_screen().\n");
            exit(1);
        }

        getimage(0, ystart, maxx, yend, buf[block]);
        ystart = yend + 1;
        yend += yincr + 1;
    }
}

void restore_screen(void far *buf[4])
{
    int ystart=0, yend, yincr, block;

    yincr = (maxy+1) / 4;
    yend = yincr;

    for (block=0; block<=3; block++)
    {
        putimage(0, ystart, buf[block], COPY_PUT);
        farfree(buf[block]);
        ystart = yend + 1;
        yend += yincr + 1;
    }
}
```

函数名: getlinesettings

功 能: 取当前线型、模式和宽度

用 法: void far getlinesettings(struct linesettingstype far *lininfo):

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

/* the names of the line styles supported */
char *lname[] = { "SOLID_LINE",
                  "DOTTED_LINE",
                  "CENTER_LINE",
                  "DASHED_LINE",
                  "USERBIT_LINE"
                  };

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    struct linesettingstype lineinfo;
    int midx, midy;
    char lstyle[80], lpattern[80], lwidth[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* get information about current line settings */
    getlinesettings(&lineinfo);
```

```
/* convert line information into strings */
sprintf(lstyle, "%s is the line style.",
        lname[lineinfo.linestyle]);
sprintf(lpattern, "0x%X is the user-defined line pattern.",
        lineinfo.upattern);
sprintf(lwidth, "%d is the line thickness.",
        lineinfo.thickness);

/* display the information */
settextjustify(CENTER_TEXT, CENTER_TEXT);
outtextxy(midx, midy, lstyle);
outtextxy(midx, midy+2*textheight("W"), lpattern);
outtextxy(midx, midy+4*textheight("W"), lwidth);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: getmaxcolor

功 能: 返回可以传给函数 setcolor 的最大颜色值

用 法: int far getmaxcolor(void);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy;
    char colstr[80];

    /* initialize graphics and local variables
    */ initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
```



```
if (errorcode != grOk) /* an error occurred */
{
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

midx = getmaxx() / 2;
midy = getmaxy() / 2;

/* grab the color info. and convert it to a string */
sprintf(colstr, "This mode supports colors 0..%d", getmaxcolor());

/* display the information */
settextjustify(CENTER_TEXT, CENTER_TEXT);
outtextxy(midx, midy, colstr);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: getmaxx

功 能: 返回屏幕的最大 x 坐标

用 法: int far getmaxx(void);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
```

```
int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy;
    char xrange[80], yrange[80];
```

```
/* initialize graphics and local variables */
initgraph(&gdriver, &gmode, "");

/* read result of initialization */
errorcode = graphresult();
if (errorcode != grOk) /* an error occurred */
{
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

midx = getmaxx() / 2;
midy = getmaxy() / 2;

/* convert max resolution values into strings */
sprintf(xrange, "X values range from 0..%d", getmaxx());
sprintf(yrange, "Y values range from 0..%d", getmaxy());

/* display the information */
settextjustify(CENTER_TEXT, CENTER_TEXT);
outtextxy(midx, midy, xrange);
outtextxy(midx, midy+textheight("W"), yrange);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: getmaxy

功 能: 返回屏幕的最大 y 坐标

用 法: int far getmaxy(void);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
```

```
int main(void)
```

```

{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy;
    char xrange[80], yrange[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* convert max resolution values into strings */
    sprintf(xrange, "X values range from 0..%d", getmaxx());
    sprintf(yrange, "Y values range from 0..%d", getmaxy());

    /* display the information */
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    outtextxy(midx, midy, xrange);
    outtextxy(midx, midy+textheight("W"), yrange);

    /* clean up */
    getch();
    closegraph();
    return 0;
}

```

函数名: getmodename

功 能: 返回含有指定图形模式名的字符串指针

用 法: char *far getmodename(int mode_name);

程序例:

```
#include <graphics.h>
```

```
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request autodetection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy, mode;
    char numname[80], modename[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* get mode number and name strings */
    mode = getgraphmode();
    sprintf(numname, "%d is the current mode number.", mode);
    sprintf(modename, "%s is the current graphics mode.", getmodename(mode));

    /* display the information */
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    outtextxy(midx, midy, numname);
    outtextxy(midx, midy+2*textheight("W"), modename);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

函数名: getmoderange

功 能: 取给定图形驱动程序的模式范围

用 法: void far getmoderange(int graphdriver, int far *lomode,
int far *himode);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy;
    int low, high;
    char mrange[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* get the mode range for this driver */
    getmoderange(gdriver, &low, &high);

    /* convert mode range info. into strings */
    sprintf(mrange, "This driver supports modes %d..%d", low, high);

    /* display the information */
    settxtjust(CENTER_TEXT, CENTER_TEXT);
```

```
    outtextxy(midx, midy, mrange);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

函数名: getpalette

功 能: 返回有关当前调色板的信息

用 法: void far getpalette(struct palettetype far *palette);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    struct palettetype pal;
    char psize[80], pval[20];
    int i, ht;
    int y = 10;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    /* an error occurred */
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n",
               grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
    }
    /* terminate with an error code */
    exit(1);
}
```

```
/* grab a copy of the palette */
getpalette(&pal);

/* convert palette info. into strings */
sprintf(psize, "The palette has %d \
    modifiable entries.", pal.size);

/* display the information */
outtextxy(0, y, psize);
if (pal.size != 0)
{
    ht = textheight("W");
    y += 2*ht;
    outtextxy(0, y, "Here are the current \
values:");
    y += 2*ht;
    for (i=0; i<pal.size; i++, y+=ht)
    {
        sprintf(pval,
            "palette[%02d]: 0x%02X", i,
            pal.colors[i]);
        outtextxy(0, y, pval);
    }
}

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: getpass

功 能: 读一个口令

用 法: char *getpass(char *prompt);

程序例:

```
#include <conio.h>
```

```
int main(void)
{
    char *password;
```

```
password = getpass("Input a password:");
cprintf("The password is: %s\r\n",
    password);
return 0;
}
```

函数名: `getpixel`

功 能: 取得指定像素的颜色

用 法: `int far getpixel(int x, int y);`

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <dos.h>

#define PIXEL_COUNT 1000
#define DELAY_TIME 100 /* in milliseconds */

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int i, x, y, color, maxx, maxy,
        maxcolor, seed;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    /* an error occurred */
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n",
            grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
    }
    /* terminate with an error code */
    exit(1);
}
```



```

    }

    maxx = getmaxx() + 1;
    maxy = getmaxy() + 1;
    maxcolor = getmaxcolor() + 1;

    while (!kbhit())
    {
/* seed the random number generator */
        seed = random(32767);
        srand(seed);
        for (i=0; i<PIXEL_COUNT; i++)
        {
            x = random(maxx);
            y = random(maxy);
            color = random(maxcolor);
            putpixel(x, y, color);
        }

        delay(DELAY_TIME);
        srand(seed);
        for (i=0; i<PIXEL_COUNT; i++)
        {
            x = random(maxx);
            y = random(maxy);
            color = random(maxcolor);
            if (color == getpixel(x, y))
                continue;
            putpixel(x, y, color);
        }
    }
}

```

函数名: gets

功 能: 从流中取一字符串

用 法: char *gets(char *string);

程序例:

```
#include <stdio.h>
```

```
int main(void)
```

```
{
    char string[80];
```

```
printf("Input a string:");
gets(string);
printf("The string input was: %s\n",
string);
return 0;
}
```

函数名: gettext

功 能: 将文本方式屏幕上的文本拷贝到存储区

用 法: int gettext(int left, int top, int right, int bottom, void *destin);

程序例:

```
#include <conio.h>
```

```
char buffer[4096];
```

```
int main(void)
{
    int i;
    clrscr();
    for (i = 0; i <= 20; i++)
        cprintf("Line #%d\r\n", i);
    gettext(1, 1, 80, 25, buffer);
    gotoxy(1, 25);
    cprintf("Press any key to clear screen...");
    getch();
    clrscr();
    gotoxy(1, 25);
    cprintf("Press any key to restore screen...");
    getch();
    puttext(1, 1, 80, 25, buffer);
    gotoxy(1, 25);
    cprintf("Press any key to quit...");
    getch();
    return 0;
}
```

函数名: gettextinfo

功 能: 取得文本模式的显示信息

用 法: void gettextinfo(struct text_info *info);

程序例:

```
#include <conio.h>

int main(void)
{
    struct text_info ti;
    gettextinfo(&ti);
    cprintf("window left      %2d\r\n",ti.winleft);
    cprintf("window top       %2d\r\n",ti.wintop);
    cprintf("window right     %2d\r\n",ti.winright);
    cprintf("window bottom    %2d\r\n",ti.winbottom);
    cprintf("attribute        %2d\r\n",ti.attribute);
    cprintf("normal attribute %2d\r\n",ti.normattr);
    cprintf("current mode      %2d\r\n",ti.currmode);
    cprintf("screen height     %2d\r\n",ti.screenheight);
    cprintf("screen width      %2d\r\n",ti.screenwidth);
    cprintf("current x         %2d\r\n",ti.curx);
    cprintf("current y         %2d\r\n",ti.cury);
    return 0;
}
```

函数名: gettextsettings

功 能: 返回有关当前图形文本字体的信息

用 法: void far gettextsettings(struct textsettingstype far *textinfo);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

/* the names of the fonts supported */
char *font[] = { "DEFAULT_FONT",
                 "TRIPLEX_FONT",
                 "SMALL_FONT",
                 "SANS_SERIF_FONT",
                 "GOTHIC_FONT"
               };

/* the names of the text directions supported */
```

```
char *dir[] = { "HORIZ_DIR", "VERT_DIR" };

/* horizontal text justifications supported */
char *hjust[] = { "LEFT_TEXT", "CENTER_TEXT", "RIGHT_TEXT" };

/* vertical text justifications supported */
char *vjust[] = { "BOTTOM_TEXT", "CENTER_TEXT", "TOP_TEXT" };

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    struct textsettingtype textinfo;
    int midx, midy, ht;
    char fontstr[80], dirstr[80], sizestr[80];
    char hjuststr[80], vjuststr[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* get information about current text settings */
    gettextsettings(&textinfo);

    /* convert text information into strings */
    sprintf(fontstr, "%s is the text style.", font[textinfo.font]);
    sprintf(dirstr, "%s is the text direction.", dir[textinfo.direction]);
    sprintf(sizestr, "%d is the text size.", textinfo.charsize);
    sprintf(hjuststr, "%s is the horizontal justification.",
            hjust[textinfo.horiz]);
    sprintf(vjuststr, "%s is the vertical justification.",
            vjust[textinfo.vert]);
```

```
/* display the information */
ht = textheight("W");
settextjustify(CENTER_TEXT, CENTER_TEXT);
outtextxy(midx, midy, fontstr);
outtextxy(midx, midy+2*ht, dirstr);
outtextxy(midx, midy+4*ht, sizestr);
outtextxy(midx, midy+6*ht, hjuststr);
outtextxy(midx, midy+8*ht, vjuststr);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: `gettime`

功 能: 取得系统时间

用 法: `void gettime(struct time *timep);`

程序例:

```
#include <stdio.h>
#include <dos.h>

int main(void)
{
    struct time t;

    gettime(&t);
    printf("The current time is: %2d:%02d:%02d.%02d\n",
           t.ti_hour, t.ti_min, t.ti_sec, t.ti_hund);
    return 0;
}
```

函数名: `getvect`

功 能: 取得中断向量入口

用 法: `void interrupt(*getvect(int intr_num));`

程序例:

```
#include <stdio.h>
```

```
#include <dos.h>

void interrupt get_out(); /* interrupt prototype */

void interrupt (*oldfunc)(); /* interrupt function pointer */
int looping = 1;

int main(void)
{
    puts("Press <Shift><Prt Sc> to terminate");

    /* save the old interrupt */
    oldfunc = getvect(5);

    /* install interrupt handler */
    setvect(5,get_out);

    /* do nothing */
    while (looping);

    /* restore to original interrupt routine */
    setvect(5,oldfunc);

    puts("Success");
    return 0;
}
void interrupt get_out()
{
    looping = 0; /* change global variable to get out of loop */
}
```

函数名: getverify

功 能: 返回 DOS 校验标志状态

用 法: int getverify(void);

程序例:

```
#include <stdio.h>
#include <dos.h>

int main(void)
{
    if (getverify())
```

```
    printf("DOS verify flag is on\n");
else
    printf("DOS verify flag is off\n");
return 0;
}
```

函数名: `getviewsetting`

功 能: 返回有关当前视区的信息

用 法: `void far getviewsettings(struct viewporttype far *viewport);`

程序例:

```
#include <graphics.h>
```

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
char *clip[] = { "OFF", "ON" };
```

```
int main(void)
```

```
{
```

```
    /* request auto detection */
```

```
    int gdriver = DETECT, gmode, errorcode;
```

```
    struct viewporttype viewinfo;
```

```
    int midx, midy, ht;
```

```
    char topstr[80], botstr[80], clipstr[80];
```

```
    /* initialize graphics and local variables */
```

```
    initgraph(&gdriver, &gmode, "");
```

```
    /* read result of initialization */
```

```
    errorcode = graphresult();
```

```
    if (errorcode != grOk) /* an error occurred */
```

```
    {
```

```
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
```

```
        printf("Press any key to halt:");
```

```
        getch();
```

```
        exit(1); /* terminate with an error code */
```

```
    }
```

```
    midx = getmaxx() / 2;
```

```
    midy = getmaxy() / 2;
```

```
    /* get information about current viewport */
```

```
getviewsettings(&viewinfo);

/* convert text information into strings */
sprintf(topstr, "(%d, %d) is the upper left viewport corner.",
        viewinfo.left, viewinfo.top);
sprintf(botstr, "(%d, %d) is the lower right viewport corner.",
        viewinfo.right, viewinfo.bottom);
sprintf(clipstr, "Clipping is turned %s.", clip[viewinfo.clip]);

/* display the information */
setttextjustify(CENTER_TEXT, CENTER_TEXT);
ht = textheight("W");
outtextxy(midx, midy, topstr);
outtextxy(midx, midy+2*ht, botstr);
outtextxy(midx, midy+4*ht, clipstr);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: `getw`

功 能: 从流中取一整数

用 法: `int getw(FILE *stream);`

程序例:

```
#include <stdio.h>
#include <stdlib.h>

#define FNAME "test.$$$"

int main(void)
{
    FILE *fp;
    int word;

    /* place the word in a file */
    fp = fopen(FNAME, "wb");
    if (fp == NULL)
    {
        printf("Error opening file %s\n", FNAME);
    }
}
```



```
        exit(1);
    }

    word = 94;
    putw(word,fp);
    if (ferror(fp))
        printf("Error writing to file\n");
    else
        printf("Successful write\n");
    fclose(fp);

    /* reopen the file */
    fp = fopen(FNAME, "rb");
    if (fp == NULL)
    {
        printf("Error opening file %s\n", FNAME);
        exit(1);
    }

    /* extract the word */
    word = getw(fp);
    if (ferror(fp))
        printf("Error reading file\n");
    else
        printf("Successful read: word = %d\n", word);

    /* clean up */
    fclose(fp);
    unlink(FNAME);

    return 0;
}
```

函数名: `getx`

功 能: 返回当前图形位置的 x 坐标

用 法: `int far getx(void);`

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
```

```
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    char msg[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    /* move to the screen center point */
    moveto(getmaxx() / 2, getmaxy() / 2);

    /* create a message string */
    sprintf(msg, "<-(%d,%d) is the here.", getx(), gety());

    /* display the message */
    outtext(msg);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

函数名: gety

功 能: 返回当前图形位置的 y 坐标

用 法: int far gety(void);

程序例:

```
#include <graphics.h>
```

```
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    char msg[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    /* move to the screen center point */
    moveto(getmaxx() / 2, getmaxy() / 2);

    /* create a message string */
    sprintf(msg, "<-(%d,%d) is the here.", getx(), gety());

    /* display the message */
    outtext(msg);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

函数名: gmtime

功 能: 把日期和时间转换为格林尼治标准时间(GMT)

用 法: struct tm *gmtime(long *clock);

程序例:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <dos.h>

/* Pacific Standard Time & Daylight Savings */
char *tzstr = "TZ=PST8PDT";

int main(void)
{
    time_t t;
    struct tm *gmt, *area;

    putenv(tzstr);
    tzset();

    t = time(NULL);
    area = localtime(&t);
    printf("Local time is: %s", asctime(area));
    gmt = gmtime(&t);
    printf("GMT is:          %s", asctime(gmt));
    return 0;
}
```

函数名: gotoxy

功 能: 在文本窗口中设置光标

用 法: void gotoxy(int x, int y);

程序例:

```
#include <conio.h>

int main(void)
{
    clrscr();
    gotoxy(35, 12);
    cprintf("Hello world");
    getch();
    return 0;
}
```

函数名: gotoxy

功 能: 在文本窗口中设置光标

用 法: void gotoxy(int x, int y);

程序例:

```
#include <conio.h>
```

```
int main(void)
```

```
{
    clrscr();
    gotoxy(35, 12);
    cprintf("Hello world");
    getch();
    return 0;
}
```

函数名: graphdefaults

功 能: 将所有图形设置复位为它们的缺省值

用 法: void far graphdefaults(void);

程序例:

```
#include <graphics.h>
```

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main(void)
```

```
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int maxx, maxy;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "c:\\bor\\Borland\\bgi");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
    }
}
```

```
    getch();
    exit(1); /* terminate with an error code */
}

maxx = getmaxx();
maxy = getmaxy();

/* output line with non-default settings */
setlinestyle(DOTTED_LINE, 0, 3);
line(0, 0, maxx, maxy);
outtextxy(maxx/2, maxy/3, "Before default values are restored.");
getch();

/* restore default values for everything */
graphdefaults();

/* clear the screen */
cleardevice();

/* output line with default settings */
line(0, 0, maxx, maxy);
outtextxy(maxx/2, maxy/3, "After restoring default values.");

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: grapherrormsg

功 能: 返回一个错误信息串的指针

用 法: char *far grapherrormsg(int errorcode);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

#define NONSENSE -50

int main(void)
```

```

{
    /* FORCE AN ERROR TO OCCUR */
    int gdriver = NONSENSE, gmode, errorcode;

    /* initialize graphics mode */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();

    /* if an error occurred, then output a */
    /* descriptive error message.          */
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    /* draw a line */
    line(0, 0, getmaxx(), getmaxy());

    /* clean up */
    getch();
    closegraph();
    return 0;
}

```

函数名: graphresult

功 能: 返回最后一次不成功的图形操作的错误代码

用 法: int far graphresult(void);

程序例:

```

#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

```

```

int main(void)
{
    /* request auto detection */

```

```

int gdriver = DETECT, gmode, errorcode;

/* initialize graphics and local variables */
initgraph(&gdriver, &gmode, "");

/* read result of initialization */
errorcode = graphresult();

if (errorcode != grOk) /* an error occurred */
{
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

/* draw a line */
line(0, 0, getmaxx(), getmaxy());

/* clean up */
getch();
closegraph();
return 0;
}

```

函数名: `_graphfreemem`

功 能: 用户可修改的图形存储区释放函数

用 法: `void far _graphfreemem(void far *ptr, unsigned size);`

程序例:

```

#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <alloc.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy;

    /* clear the text screen */

```



```
    clrscr();
    printf("Press any key to initialize graphics mode:");
    getch();
    clrscr();

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* display a message */
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    outtextxy(midx, midy, "Press any key to exit graphics mode:");

    /* clean up */
    getch();
    closegraph();
    return 0;
}

/* called by the graphics kernel to allocate memory */
void far * far _graphgetmem(unsigned size)
{
    printf("_graphgetmem called to allocate %d bytes.\n", size);
    printf("hit any key:");
    getch();
    printf("\n");

    /* allocate memory from far heap */
    return farmalloc(size);
}

/* called by the graphics kernel to free memory */
```

```
void far _graphfreemem(void far *ptr, unsigned size)
{
    printf("_graphfreemem called to free %d bytes.\n", size);
    printf("hit any key:");
    getch();
    printf("\n");

    /* free ptr from far heap */
    farfree(ptr);
}
```

函数名: _graphgetmem

功 能: 用户可修改的图形存储区分配函数

用 法: void far *far _graphgetmem(unsigned size);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <alloc.h>

int main(void)
{
    /* request autodetection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy;

    /* clear the text screen */
    clrscr();
    printf("Press any key to initialize graphics mode:");
    getch();
    clrscr();

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)          /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
    }
```

```

    getch();
    exit(1);          /* terminate with an error code */
}

midx = getmaxx() / 2;
midy = getmaxy() / 2;

/* display a message */
setttextjustify(CENTER_TEXT, CENTER_TEXT);
outtextxy(midx, midy, "Press any key to exit graphics mode.");

/* clean up */
getch();
closegraph();
return 0;
}

/* called by the graphics kernel to allocate memory */
void far * far _graphgetmem(unsigned size)
{
    printf("_graphgetmem called to allocate %d bytes.\n", size);
    printf("hit any key:");
    getch();
    printf("\n");

    /* allocate memory from far heap */
    return farmalloc(size);
}

/* called by the graphics kernel to free memory */
void far _graphfreemem(void far *ptr, unsigned size)
{
    printf("_graphfreemem called to free %d bytes.\n", size);
    printf("hit any key:");
    getch();
    printf("\n");

    /* free ptr from far heap */
    farfree(ptr);
}
*****
函数名: harderr
功 能: 建立一个硬件错误处理程序
用 法: void harderr(int (*fptr)());

```

程序例:

```
/*This program will trap disk errors and prompt
the user for action. Try running it with no
disk in drive A: to invoke its functions.*/

#include <stdio.h>
#include <conio.h>
#include <dos.h>
#define IGNORE 0
#define RETRY 1
#define ABORT 2
int buf[500];
/*define the error messages for trapping disk problems*/
static char *err_msg[] = {
    "write protect",
    "unknown unit",
    "drive not ready",
    "unknown command",
    "data error (CRC)",
    "bad request",
    "seek error",
    "unknown media type",
    "sector not found",
    "printer out of paper",
    "write fault",
    "read fault",
    "general failure",
    "reserved",
    "reserved",
    "invalid disk change"
};

error_win(char *msg)
{
    int retval;

    cputs(msg);

    /*prompt for user to press a key to abort, retry, ignore*/
    while(1)
    {
        retval= getch();
        if (retval == 'a' || retval == 'A')
        {
```

```
        retval = ABORT;
        break;
    }
    if (retval == 'r' || retval == 'R')
    {
        retval = RETRY;
        break;
    }
    if (retval == 't' || retval == 'T')
    {
        retval = IGNORE;
        break;
    }
}

return(retval);
}

/*pragma warn -par reduces warnings which occur
due to the non use of the parameters errval,
bp and si to the handler.*/
#pragma warn -par

int handler(int errval,int ax,int bp,int si)
{
    static char msg[80];
    unsigned di;
    int drive;
    int errorno;
    di= _DI;
    /*if this is not a disk error then it was
another device having trouble*/

    if (ax < 0)
    {
        /* report the error */
        error_win("Device error");
        /* and return to the program directly requesting abort */
        hardretn(ABORT);
    }
    /* otherwise it was a disk error */
    drive = ax & 0x00FF;
    errorno = di & 0x00FF;
    /* report which error it was */
```

```

        sprintf(msg, "Error: %s on drive %c\r\nA)bort, R)etry, I)gnore: ",
            err_msg[erromo], 'A' + drive);
/*
return to the program via dos interrupt 0x23 with abort, retry,
or ignore as input by the user.
*/
        hardresume(error_win(msg));
        return ABORT;
    }
#pragma warn +par

int main(void)
{
/*
install our handler on the hardware problem interrupt
*/
        harderr(handler);
        clrscr();
        printf("Make sure there is no disk in drive A:\n");
        printf("Press any key ....\n");
        getch();
        printf("Trying to access drive A:\n");
        printf("fopen returned %p\n", fopen("A:temp.dat", "w"));
        return 0;
}

```

函数名: hardresume

功 能: 硬件错误处理函数

用 法: void hardresume(int rescode);

程序例:

```

/* This program will trap disk errors and prompt the user for action. */
/* Try running it with no disk in drive A: to invoke its functions      */

#include <stdio.h>
#include <conio.h>
#include <dos.h>

#define IGNORE 0
#define RETRY 1
#define ABORT 2

```

```
int buf[500];

/* define the error messages for trapping disk problems */
static char *err_msg[] = {
    "write protect",
    "unknown unit",
    "drive not ready",
    "unknown command",
    "data error (CRC)",
    "bad request",
    "seek error",
    "unknown media type",
    "sector not found",
    "printer out of paper",
    "write fault",
    "read fault",
    "general failure",
    "reserved",
    "reserved",
    "invalid disk change"
};

error_win(char *msg)
{
    int retval;

    cputs(msg);

    /* prompt for user to press a key to abort, retry, ignore */
    while(1)
    {
        retval= getch();
        if (retval == 'a' || retval == 'A')
        {
            retval = ABORT;
            break;
        }
        if (retval == 'r' || retval == 'R')
        {
            retval = RETRY;
            break;
        }
        if (retval == 'i' || retval == 'I')
```

```
        {
            retval = IGNORE;
            break;
        }
    }

    return(retval);
}

/* pragma warn -par reduces warnings which occur due to the non use */
/* of the parameters errval, bp and si to the handler.                */
#pragma warn -par

int handler(int errval,int ax,int bp,int si)
{
    static char msg[80];
    unsigned di;
    int drive;
    int errorno;

    di = _DI;
    /* if this is not a disk error then it was another device having trouble */

    if (ax < 0)
    {
        /* report the error */
        error_win("Device error");
        /* and return to the program directly
        requesting abort */
        hardretn(ABORT);
    }
    /* otherwise it was a disk error */
    drive = ax & 0x00FF;
    errorno = di & 0x00FF;
    /* report which error it was */
    sprintf(msg, "Error: %s on drive %c\r\nA)bort, R)etry, I)gnore: ",
            err_msg[errorno], 'A' + drive);
    /* return to the program via dos interrupt 0x23 with abort, retry */
    /* or ignore as input by the user. */
    hardresume(error_win(msg));
    return ABORT;
}
#pragma warn +par
```



```
int main(void)
{
/* install our handler on the hardware problem interrupt */
    harderr(handler);
    clrscr();
    printf("Make sure there is no disk in drive A:\n");
    printf("Press any key ....\n");
    getch();
    printf("Trying to access drive A:\n");
    printf("fopen returned %p\n",fopen("A:temp.dat", "w"));
    return 0;
}
```

函数名: highvideo

功 能: 选择高亮度文本字符

用 法: void highvideo(void);

程序例:

```
#include <conio.h>
```

```
int main(void)
{
    clrscr();

    lowvideo();
    cprintf("Low Intensity text\r\n");
    highvideo();
    gotoxy(1,2);
    cprintf("High Intensity Text\r\n");

    return 0;
}
```

函数名: hypot

功 能: 计算直角三角形的斜边长

用 法: double hypot(double x, double y);

程序例:

```
#include <stdio.h>
```

```
#include <math.h>
```

```
int main(void)
{
    double result;
    double x = 3.0;
    double y = 4.0;

    result = hypot(x, y);
    printf("The hypotenuse is: %lf\n", result);

    return 0;
}
```

函数名: imagesize

功 能: 返回保存位图像所需的字节数

用 法: unsigned far imagesize(int left, int top, int right, int bottom);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
```

```
#define ARROW_SIZE 10
```

```
void draw_arrow(int x, int y);
```

```
int main(void)
{
    /* request autodetection */
    int gdriver = DETECT, gmode, errorcode;
    void *arrow;
    int x, y, maxx;
    unsigned int size;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
    }
}
```

```
    getch();
    exit(1); /* terminate with an error code */
}

maxx = getmaxx();
x = 0;
y = getmaxy() / 2;

/* draw the image to be grabbed */
draw_arrow(x,y);

/* calculate the size of the image */
size = imagesize(x, y-ARROW_SIZE, x+(4*ARROW_SIZE), y+ARROW_SIZE);

/* allocate memory to hold the image */
arrow = malloc(size);

/* grab the image */
getimage(x, y-ARROW_SIZE, x+(4*ARROW_SIZE), y+ARROW_SIZE, arrow);

/* repeat until a key is pressed */
while (!kbhit())
{
    /* erase old image */
    putimage(x, y-ARROW_SIZE, arrow, XOR_PUT);

    x += ARROW_SIZE;
    if (x >= maxx)
        x = 0;

    /* plot new image */
    putimage(x, y-ARROW_SIZE, arrow, XOR_PUT);
}

/* clean up */
free(arrow);
closegraph();
return 0;
}

void draw_arrow(int x, int y)
{
    /* draw an arrow on the screen */
    moveto(x, y);
```

```
    linerel(4*ARROW_SIZE, 0);
    linerel(-2*ARROW_SIZE, -1*ARROW_SIZE);
    linerel(0, 2*ARROW_SIZE);
    linerel(2*ARROW_SIZE, -1*ARROW_SIZE);
}
```

函数名: initgraph

功 能: 初始化图形系统

用 法: void far initgraph(int far *graphdriver, int far *graphmode,
char far *pathdriver);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;

    /* initialize graphics mode */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();

    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* return with error code */
    }

    /* draw a line */
    line(0, 0, getmaxx(), getmaxy());

    /* clean up */
    getch();
}
```

```
closegraph();
return 0;
}
```

函数名: inport

功 能: 从硬件端口中输入

用 法: int inp(int protid);

程序例:

```
#include <stdio.h>
#include <dos.h>

int main(void)
{
    int result;
    int port = 0; /* serial port 0 */

    result = inport(port);
    printf("Word read from port %d = 0x%X\n", port, result);
    return 0;
}
```

函数名: inline

功 能: 在文本窗口中插入一个空行

用 法: void inline(void);

程序例:

```
#include <conio.h>

int main(void)
{
    clrscr();
    cprintf("INLINE inserts an empty line in the text window\r\n");
    cprintf("at the cursor position using the current text\r\n");
    cprintf("background color. All lines below the empty one\r\n");
    cprintf("move down one line and the bottom line scrolls\r\n");
    cprintf("off the bottom of the window.\r\n");
    cprintf("\r\nPress any key to continue:");
    gotoxy(1, 3);
    getch();
}
```

```
    insline();  
    getch();  
    return 0;  
}
```

函数名: installuserdriver

功 能: 安装设备驱动程序到 BGI 设备驱动程序表中

用 法: int far installuserdriver(char far *name, int (*detect)(void));

程序例:

```
#include <graphics.h>  
#include <stdlib.h>  
#include <stdio.h>  
#include <conio.h>  
  
/* function prototypes */  
int huge detectEGA(void);  
void checkerrors(void);  
  
int main(void)  
{  
    int gdriver, gmode;  
  
    /* install a user written device driver */  
    gdriver = installuserdriver("EGA", detectEGA);  
  
    /* must force use of detection routine */  
    gdriver = DETECT;  
  
    /* check for any installation errors */  
    checkerrors();  
  
    /* initialize graphics and local variables */  
    initgraph(&gdriver, &gmode, "");  
  
    /* check for any initialization errors */  
    checkerrors();  
  
    /* draw a line */  
    line(0, 0, getmaxx(), getmaxy());  
}
```

```

    /* clean up */
    getch();
    closegraph();
    return 0;
}

/* detects EGA or VGA cards */
int huge detectEGA(void)
{
    int driver, mode, sugmode = 0;

    detectgraph(&driver, &mode);
    if ((driver == EGA) || (driver == VGA))
        /* return suggested video mode number */
        return sugmode;
    else
        /* return an error code */
        return grError;
}

/* check for and report any graphics errors */
void checkerrors(void)
{
    int errorcode;

    /* read result of last graphics operation */
    errorcode = graphresult();
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1);
    }
}

```

函数名: installuserfont

功 能: 安装未嵌入 BGI 系统的字体文件(CHR)

用 法: int far installuserfont(char far *name);

程序例:

```

#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>

```

```
#include <conio.h>

/* function prototype */
void checkerrors(void);

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode;
    int userfont;
    int midx, midy;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* check for any initialization errors */
    checkerrors();

    /* install a user defined font file */
    userfont = installuserfont("USER.CHR");

    /* check for any installation errors */
    checkerrors();

    /* select the user font */
    settextstyle(userfont, HORIZ_DIR, 4);

    /* output some text */
    outtextxy(midx, midy, "Testing!");

    /* clean up */
    getch();
    closegraph();
    return 0;
}

/* check for and report any graphics errors */
void checkerrors(void)
{
    int errorcode;
```



```
/* read result of last graphics operation */
errorcode = graphresult();
if (errorcode != grOk)
{
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1);
}
}
```

函数名: int86

功 能: 通用 8086 软中断接口

用 法: int int86(int intr_num, union REGS *inregs, union REGS *outregs);

程序例:

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>
```

```
#define VIDEO 0x10
```

```
void movetoxy(int x, int y)
{
    union REGS regs;

    regs.h.ah = 2; /* set cursor position */
    regs.h.dh = y;
    regs.h.dl = x;
    regs.h.bh = 0; /* video page 0 */
    int86(VIDEO, &regs, &regs);
}
```

```
int main(void)
{
    clrscr();
    movetoxy(35, 10);
    printf("Hello\n");
    return 0;
}
```

函数名: int86x

功 能: 通用 8086 软中断接口

用 法: int int86x(int intr_num, union REGS *inregs, union REGS *outregs,
struct SREGS *segregs);

程序例:

```
#include <dos.h>
#include <process.h>
#include <stdio.h>

int main(void)
{
    char filename[80];
    union REGS inregs, outregs;
    struct SREGS segregs;

    printf("Enter filename: ");
    gets(filename);
    inregs.h.ah = 0x43;
    inregs.h.al = 0x21;
    inregs.x.dx = FP_OFF(filename);
    segregs.ds = FP_SEG(filename);
    int86x(0x21, &inregs, &outregs, &segregs);
    printf("File attribute: %X\n", outregs.x.cx);
    return 0;
}
```

函数名: intdos

功 能: 通用 DOS 接口

用 法: int intdos(union REGS *inregs, union REGS *outregs);

程序例:

```
#include <stdio.h>
#include <dos.h>

/* deletes file name; returns 0 on success, nonzero on failure */
int delete_file(char near *filename)
{
    union REGS regs;
```

```
int ret;
regs.h.ah = 0x41;                      /* delete file */
regs.x.dx = (unsigned) filename;
ret = intdos(&regs, &regs);

/* if carry flag is set, there was an error */
return(regs.x.cflag ? ret : 0);
}

int main(void)
{
    int err;
    err = delete_file("NOTEXIST.$$$");
    if (!err)
        printf("Able to delete NOTEXIST.$$$\n");
    else
        printf("Not Able to delete NOTEXIST.$$$\n");
    return 0;
}
```

函数名: intdosx

功 能: 通用 DOS 中断接口

用 法: int intdosx(union REGS *inregs, union REGS *outregs,
struct SREGS *segregs);

程序例:

```
#include <stdio.h>
#include <dos.h>

/* deletes file name; returns 0 on success, nonzero on failure */
int delete_file(char far *filename)
{
    union REGS regs; struct SREGS sregs;
    int ret;
    regs.h.ah = 0x41;                      /* delete file */
    regs.x.dx = FP_OFF(filename);
    sregs.ds = FP_SEG(filename);
    ret = intdosx(&regs, &regs, &sregs);

    /* if carry flag is set, there was an error */
    return(regs.x.cflag ? ret : 0);
}
```

```
}

int main(void)
{
    int err;
    err = delete_file("NOTEXIST.$$$");
    if (!err)
        printf("Able to delete NOTEXIST.$$$\n");
    else
        printf("Not Able to delete NOTEXIST.$$$\n");
    return 0;
}
```

函数名: intr

功 能: 改变软中断接口

用 法: void intr(int intr_num, struct REGPACK *preg);

程序例:

```
#include <stdio.h>
#include <string.h>
#include <dir.h>
#include <dos.h>

#define CF 1 /* Carry flag */

int main(void)
{
    char directory[80];
    struct REGPACK reg;

    printf("Enter directory to change to: ");
    gets(directory);
    reg.r_ax = 0x3B << 8; /* shift 3Bh into AH */
    reg.r_dx = FP_OFF(directory);
    reg.r_ds = FP_SEG(directory);
    intr(0x21, &reg);
    if (reg.r_flags & CF)
        printf("Directory change failed\n");
    getcwd(directory, 80);
    printf("The current directory is: %s\n", directory);
    return 0;
}
```

函数名: ioctl

功 能: 控制 I/O 设备

用 法: int ioctl(int handle, int cmd[,int *argdx, int argcx]);

程序例:

```
#include <stdio.h>
#include <dir.h>
#include <io.h>

int main(void)
{
    int stat;

    /* use func 8 to determine if the default drive is removable */
    stat = ioctl(0, 8, 0, 0);
    if (!stat)
        printf("Drive %c is removable.\n", getdisk() + 'A');
    else
        printf("Drive %c is not removable.\n", getdisk() + 'A');
    return 0;
}
```

函数名: isatty

功 能: 检查设备类型

用 法: int isatty(int handle);

程序例:

```
#include <stdio.h>
#include <io.h>

int main(void)
{
    int handle;

    handle = fileno(stdprn);
    if (isatty(handle))
        printf("Handle %d is a device type\n", handle);
    else
```

```
    printf("Handle %d isn't a device type\n", handle);
    return 0;
}
```

函数名: itoa

功 能: 把一整数转换为字符串

用 法: char *itoa(int value, char *string, int radix);

程序例:

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int number = 12345;
```

```
    char string[25];
```

```
    itoa(number, string, 10);
```

```
    printf("integer = %d string = %s\n", number, string);
```

```
    return 0;
```

```
}
```

```
*****
```

函数名: kbhit

功 能: 检查当前按下的键

用 法: int kbhit(void);

程序例:

```
#include <conio.h>
```

```
int main(void)
```

```
{
```

```
    cprintf("Press any key to continue:");
```

```
    while (!kbhit()) /* do nothing */ ;
```

```
    cprintf("\r\nA key was pressed...\r\n");
```

```
    return 0;
```

```
}
```

函数名: keep

功 能: 退出并继续驻留

用 法: void keep(int status, int size);

程序例:

```
/**NOTE:
    This is an interrupt service routine.  You
    can NOT compile this program with Test
    Stack Overflow turned on and get an
    executable file which will operate
    correctly.  Due to the nature of this
    function the formula used to compute
    the number of paragraphs may not
    necessarily work in all cases.  Use with
    care!  Terminate Stay Resident (TSR)
    programs are complex and no other support
    for them is provided.  Refer to the
    MS-DOS technical documentation
    for more information.  */
#include <dos.h>
/* The clock tick interrupt */
#define INTR 0x1C
/* Screen attribute (blue on grey) */
#define ATTR 0x7900

/* reduce heaplength and stacklength
to make a smaller program in memory */
extern unsigned _heaplen = 1024;
extern unsigned _stklen = 512;

void interrupt ( *oldhandler)(void);

void interrupt handler(void)
{
    unsigned int (far *screen)[80];
    static int count;

/* For a color screen the video memory
   is at B800:0000.  For a monochrome
   system use B000:000 */
    screen = MK_FP(0xB800,0);

/* increase the counter and keep it
   within 0 to 9 */
    count++;
```

```
count %= 10;

/* put the number on the screen */
screen[0][79] = count + '0' + ATTR;

/* call the old interrupt handler */
oldhandler();
}

int main(void)
{

/* get the address of the current clock
tick interrupt */
oldhandler = getvect(INTR);

/* install the new interrupt handler */
setvect(INTR, handler);

/* _psp is the starting address of the
program in memory. The top of the stack
is the end of the program. Using _SS and
_SP together we can get the end of the
stack. You may want to allow a bit of
saftey space to insure that enough room
is being allocated ie :
(_SS + ((_SP + safety space)/16) - _psp)
*/
keep(0, (_SS + (_SP/16) - _psp));
return 0;
}
*****
```

函数名: labs

功 能: 取长整型绝对值

用 法: long labs(long n);

程序例:

```
#include <stdio.h>
#include <math.h>
```

```
int main(void)
{
    long result;
    long x = -12345678L;
```



```
    result= labs(x);
    printf("number: %ld abs value: %ld\n",
           x, result);

    return 0;
}
```

函数名: ldexp

功 能: 计算 value*2 的幂

用 法: double ldexp(double value, int exp);

程序例:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    double value;
    double x = 2;

    /* ldexp raises 2 by a power of 3
       then multiplies the result by 2    */
    value = ldexp(x,3);
    printf("The ldexp value is: %lf\n",
           value);

    return 0;
}
```

函数名: ldiv

功 能: 两个长整型数相除, 返回商和余数

用 法: ldiv_t ldiv(long lnumer, long ldenom);

程序例:

```
/* ldiv example */

#include <stdlib.h>
#include <stdio.h>
```

```
int main(void)
{
    ldiv_t lx;

    lx = ldiv(100000L, 30000L);
    printf("100000 div 30000 = %ld remainder %ld\n", lx.quot, lx.rem);
    return 0;
}
```

函数名: lfind

功 能: 执行线性搜索

用 法: void *lfind(void *key, void *base, int *nelem, int width,
int (*fcmp)());

程序例:

```
#include <stdio.h>
#include <stdlib.h>
```

```
int compare(int *x, int *y)
{
    return( *x - *y );
}
```

```
int main(void)
{
    int array[5] = {35, 87, 46, 99, 12};
    size_t nelem = 5;
    int key;
    int *result;

    key = 99;
    result = lfind(&key, array, &nelem,
        sizeof(int), (int (*)(const void *,const void *))compare);
    if (result)
        printf("Number %d found\n",key);
    else
        printf("Number %d not found\n",key);

    return 0;
}
```

函数名: line

功 能: 在指定两点间画一直线

用 法: void far line(int x0, int y0, int x1, int y1);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int xmax, ymax;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    /* an error occurred */
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n",
               grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1);
    }

    setcolor(getmaxcolor());
    xmax = getmaxx();
    ymax = getmaxy();

    /* draw a diagonal line */
    line(0, 0, xmax, ymax);

    /* clean up */
    getch();
    closegraph();
}
```

```
    return 0;
}
```

函数名: linerel

功 能: 从当前位置点(CP)到与 CP 有一给定相对距离的点画一直线

用 法: void far linerel(int dx, int dy);

程序例:

```
#include <graphics.h>
```

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main(void)
```

```
{
```

```
    /* request auto detection */
```

```
    int gdriver = DETECT, gmode, errorcode;
```

```
    char msg[80];
```

```
    /* initialize graphics and local variables */
```

```
    initgraph(&gdriver, &gmode, "");
```

```
    /* read result of initialization */
```

```
    errorcode = graphresult();
```

```
    if (errorcode != grOk)
```

```
    {
```

```
        printf("Graphics error: %s\n",
```

```
grapherrormsg(errorcode));
```

```
        printf("Press any key to halt:");
```

```
        getch();
```

```
        exit(1);
```

```
    }
```

```
    /* move the C.P. to location (20, 30) */
```

```
    moveto(20, 30);
```

```
    /* create and output a
```

```
        message at (20, 30) */
```

```
    sprintf(msg, " (%d,%d)", getx(), gety());
```

```
    outtextxy(20, 30, msg);
```

```
    /* draw a line to a point a relative
```

```
        distance away from the current
        value of C.P.    */
    linerel(100, 100);

    /* create and output a message at C.P. */
    sprintf(msg, " (%d,%d)", getx(), gety());
    outtext(msg);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

函数名: localtime

功 能: 把日期和时间转变为结构

用 法: struct tm *localtime(long *clock);

程序例:

```
#include <time.h>
#include <stdio.h>
#include <dos.h>

int main(void)
{
    time_t timer;
    struct tm *tblock;

    /* gets time of day */
    timer = time(NULL);

    /* converts date/time to a structure */
    tblock = localtime(&timer);

    printf("Local time is: %s", asctime(tblock));

    return 0;
}
```

函数名: lock

功 能: 设置文件共享锁

用 法: int lock(int handle, long offset, long length);

程序例:

```
#include <io.h>
#include <fcntl.h>
#include <sys\stat.h>
#include <process.h>
#include <share.h>
#include <stdio.h>

int main(void)
{
    int handle, status;
    long length;

    /* Must have DOS Share.exe loaded for */
    /* file locking to function properly */

    handle = sopen("c:\\autoexec.bat",
        O_RDONLY,SH_DENYNO,S_IREAD);

    if (handle < 0)
    {
        printf("sopen failed\n");
        exit(1);
    }

    length = filelength(handle);
    status = lock(handle,0L,length/2);

    if (status == 0)
        printf("lock succeeded\n");
    else
        printf("lock failed\n");

    status = unlock(handle,0L,length/2);

    if (status == 0)
        printf("unlock succeeded\n");
    else
        printf("unlock failed\n");

    close(handle);
}
```

```
    return 0;  
}
```

函数名: `log`

功 能: 对数函数 $\ln(x)$

用 法: `double log(double x);`

程序例:

```
#include <math.h>  
#include <stdio.h>  
  
int main(void)  
{  
    double result;  
    double x = 8.6872;  
  
    result = log(x);  
    printf("The natural log of %lf is %lf\n", x, result);  
  
    return 0;  
}
```

函数名: `log10`

功 能: 对数函数 \log

用 法: `double log10(double x);`

程序例:

```
#include <math.h>  
#include <stdio.h>  
  
int main(void)  
{  
    double result;  
    double x = 800.6872;  
  
    result = log10(x);  
    printf("The common log of %lf is %lf\n", x, result);  
  
    return 0;  
}
```

函数名: longjump

功 能: 执行非局部转移

用 法: void longjump(jmp_buf env, int val);

程序例:

```
#include <stdio.h>
#include <setjmp.h>
#include <stdlib.h>
```

```
void subroutine(jmp_buf);
```

```
int main(void)
{
    int value;
    jmp_buf jumper;

    value = setjmp(jumper);
    if (value != 0)
    {
        printf("Longjmp with value %d\n", value);
        exit(value);
    }
    printf("About to call subroutine ... \n");
    subroutine(jumper);

    return 0;
}
```

```
void subroutine(jmp_buf jumper)
{
    longjmp(jumper,1);
}
```

函数名: lowvideo

功 能: 选择低亮度字符

用 法: void lowvideo(void);

程序例:

```
#include <conio.h>

int main(void)
{
    clrscr();

    highvideo();
    cprintf("High Intesity Text\r\n");
    lowvideo();
    gotoxy(1,2);
    cprintf("Low Intensity Text\r\n");

    return 0;
}
```

函数名: `lrotl`, `_lrotl`

功 能: 将无符号长整型数向左循环移位

用 法: `unsigned long lrotl(unsigned long lvalue, int count);`

`unsigned long _lrotl(unsigned long lvalue, int count);`

程序例:

```
/* lrotl example */
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
    unsigned long result;
    unsigned long value = 100;

    result = _lrotl(value,1);
    printf("The value %lu rotated left one bit is: %lu\n", value, result);

    return 0;
}
```

函数名: `lsearch`

功 能: 线性搜索

用 法: void *lsearch(const void *key, void *base, size_t *nelem,
size_t width, int (*fcmp)(const void *, const void *));

程序例:

```
#include <stdio.h>
#include <stdlib.h>

int compare(int *x, int *y)
{
    return( *x - *y );
}

int main(void)
{
    int array[5] = {35, 87, 46, 99, 12};
    size_t nelem = 5;
    int key;
    int *result;

    key = 99;
    result = lfind(&key, array, &nelem,
        sizeof(int), (int (*)(const void *,const void *))compare);
    if (result)
        printf("Number %d found\n",key);
    else
        printf("Number %d not found\n",key);

    return 0;
}
```

函数名: lseek

功 能: 移动文件读/写指针

用 法: long lseek(int handle, long offset, int fromwhere);

程序例:

```
#include <sys/stat.h>
#include <string.h>
#include <stdio.h>
#include <fcntl.h>
#include <io.h>
```

```

int main(void)
{
    int handle;
    char msg[] = "This is a test";
    char ch;

    /* create a file */
    handle = open("TEST.$$$", O_CREAT | O_RDWR, S_IRREAD | S_IWRITE);

    /* write some data to the file */
    write(handle, msg, strlen(msg));

    /* seek to the beginning of the file */
    lseek(handle, 0L, SEEK_SET);

    /* reads chars from the file until we hit EOF */
    do
    {
        read(handle, &ch, 1);
        printf("%c", ch);
    } while (!eof(handle));

    close(handle);
    return 0;
}
*****

```

main()主函数

每一 C 程序都必须有一 main() 函数, 可以根据自己的爱好把它放在程序的某个地方。有些程序员把它放在最前面, 而另一些程序员把它放在最后面, 无论放在哪个地方, 以下几点说明都是适合的。

1. main() 参数

在 Turbo C2.0 启动过程中, 传递 main() 函数三个参数: argc, argv 和 env。

* argc: 整数, 为传给 main() 的命令行参数个数。

* argv: 字符串数组。

在 DOS 3.X 版本中, argv[0] 为程序运行的全路径名; 对 DOS 3.0

以下的版本, argv[0] 为空串("")。

argv[1] 为在 DOS 命令行中执行程序名后的第一个字符串;

argv[2] 为执行程序名后的第二个字符串;

...

argv[argc] 为 NULL。

* env: 字符串数组。env[] 的每一个元素都包含 ENVVAR=value 形式的字符串。其中 ENVVAR 为环境变量如 PATH 或 87。value 为 ENVVAR 的对应值如 C:\DOS, C:\TURBOC(对于 PATH) 或 YES(对于 87)。

Turbo C2.0 启动时总是把这三个参数传递给 `main()` 函数, 可以在用户程序中说明(或不说明)它们, 如果说明了部分(或全部)参数, 它们就成为 `main()` 子程序的局部变量。

请注意: 一旦想说明这些参数, 则必须按 `argc`, `argv`, `env` 的顺序, 如以下的例子:

```
main()
main(int argc)
main(int argc, char *argv[])
main(int argc, char *argv[], char *env[])
```

其中第二种情况是合法的, 但不常见, 因为在程序中很少有只用 `argc`, 而不用 `argv[]` 的情况。

以下提供一样例程序 `EXAMPLE.EXE`, 演示如何在 `main()` 函数中使用三个参数:

```
/*program name EXAMPLE.EXE*/
#include <stdio.h>
#include <stdlib.h>
main(int argc, char *argv[], char *env[])
{
    int i;
    printf("These are the %d command- line arguments passed to
           main:\n\n", argc);
    for(i=0; i<=argc; i++)
        printf("argv[%d]:%s\n", i, argv[i]);
    printf("\nThe environment string(s) on this system are:\n\n");
    for(i=0; env[i]!=NULL; i++)
        printf(" env[%d]:%s\n", i, env[i]);
}
```

如果在 DOS 提示符下, 按以下方式运行 `EXAMPLE.EXE`:

`C:\example first_argument "argument with blanks" 3 4 "last but one" stop!`

注意: 可以用双引号括起内含空格的参数, 如本例中的: `" argument with blanks"`和`"Last but one"`。

结果是这样的:

```
The value of argc is 7
These are the 7 command-line arguments passed to main:
argv[0]:C:\TURBO\EXAMPLE.EXE
argv[1]:first_argument
argv[2]:argument with blanks
argv[3]:3
argv[4]:4
argv[5]:last but one
argv[6]:stop!
argv[7]:(NULL)
The environment string(s) on this system are:
env[0]: COMSPEC=C:\COMMAND.COM
```

```
env[1]: PROMPT=$P$G          /*视具体设置而定*/
env[2]: PATH=C:\DOS;C:\TC     /*视具体设置而定*/
```

应该提醒的是: 传送 main() 函数的命令行参数的最大长度为 128 个字符 (包括参数间的空格), 这是由 DOS 限制的。

函数名: matherr

功 能: 用户可修改的数学错误处理程序

用 法: int matherr(struct exception *e);

程序例:

```
/* This is a user-defined matherr function that prevents
   any error messages from being printed. */
```

```
#include<math.h>
```

```
int matherr(struct exception *a)
{
    return 1;
}
```

函数名: memcpy

功 能: 从源 source 中拷贝 n 个字节到目标 dest 中

用 法: void *memcpy(void *dest, void *source, unsigned char ch,
 unsigned n);

程序例:

```
#include <string.h>
```

```
#include <stdio.h>
```

```
int main(void)
{
    char *src = "This is the source string";
    char dest[50];
    char *ptr;

    ptr = memcpy(dest, src, 'c', strlen(src));

    if (ptr)
    {
```

```
    *ptr = '\0';
    printf("The character was found:  %s\n", dest);
}
else
    printf("The character wasn't found\n");
return 0;
}
```

函数名: malloc

功 能: 内存分配函数

用 法: void *malloc(unsigned size);

程序例:

```
#include <stdio.h>
#include <string.h>
#include <alloc.h>
#include <process.h>

int main(void)
{
    char *str;

    /* allocate memory for string */
    /* This will generate an error when compiling */
    /* with C++, use the new operator instead. */
    if ((str = malloc(10)) == NULL)
    {
        printf("Not enough memory to allocate buffer\n");
        exit(1); /* terminate program if out of memory */
    }

    /* copy "Hello" into string */
    strcpy(str, "Hello");

    /* display string */
    printf("String is %s\n", str);

    /* free memory */
    free(str);

    return 0;
}
```

函数名: memchr

功 能: 在数组的前 n 个字节中搜索字符

用 法: void *memchr(void *s, char ch, unsigned n);

程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    char str[17];
    char *ptr;

    strcpy(str, "This is a string");
    ptr = memchr(str, 'r', strlen(str));
    if (ptr)
        printf("The character 'r' is at position: %d\n", ptr - str);
    else
        printf("The character was not found\n");
    return 0;
}
```

函数名: memcpy

功 能: 从源 source 中拷贝 n 个字节到目标 destin 中

用 法: void *memcpy(void *destin, void *source, unsigned n);

程序例:

```
#include <stdio.h>
#include <string.h>
int main(void)
{
    char src[] = "*****";
    char dest[] = "abcdefghijklmnopqrstuvwxy0123456709";
    char *ptr;
    printf("destination before memcpy: %s\n", dest);
    ptr = memcpy(dest, src, strlen(src));
    if (ptr)
        printf("destination after memcpy:  %s\n", dest);
    else
```

```
    printf("memcpy failed\n");
    return 0;
}
```

函数名: memicmp

功 能: 比较两个串 s1 和 s2 的前 n 个字节, 忽略大小写

用 法: int memicmp(void *s1, void *s2, unsigned n);

程序例:

```
#include <stdio.h>
#include <string.h>
```

```
int main(void)
{
    char *buf1 = "ABCDE123";
    char *buf2 = "abcde456";
    int stat;
    stat = memicmp(buf1, buf2, 5);
    printf("The strings to position 5 are ");
    if (stat)
        printf("not ");
    printf("the same\n");
    return 0;
}
```

函数名: memmove

功 能: 移动一块字节

用 法: void *memmove(void *destin, void *source, unsigned n);

程序例:

```
#include <string.h>
#include <stdio.h>
```

```
int main(void)
{
    char *dest = "abcdefghijklmnopqrstuvwxy0123456789";
    char *src = "*****";
    printf("destination prior to memmove: %s\n", dest);
    memmove(dest, src, 26);
    printf("destination after memmove:    %s\n", dest);
}
```



```
    return 0;
}
```

函数名: `memset`

功 能: 设置 `s` 中的所有字节为 `ch`, `s` 数组的大小由 `n` 给定

用 法: `void *memset(void *s, char ch, unsigned n);`

程序例:

```
#include <string.h>
#include <stdio.h>
#include <mem.h>
```

```
int main(void)
{
    char buffer[] = "Hello world\n";

    printf("Buffer before memset: %s\n", buffer);
    memset(buffer, '*', strlen(buffer) - 1);
    printf("Buffer after memset:  %s\n", buffer);
    return 0;
}
```

函数名: `mkdir`

功 能: 建立一个目录

用 法: `int mkdir(char *pathname);`

程序例:

```
#include <stdio.h>
#include <conio.h>
#include <process.h>
#include <dir.h>
```

```
int main(void)
{
    int status;

    clrscr();
    status = mkdir("asdfjklm");
    (!status) ? (printf("Directory created\n")) :
```

```
        (printf("Unable to create directory\n"));

    getch();
    system("dir");
    getch();

    status = rmdir("asdfjklm");
    (!status) ? (printf("Directory deleted\n")) :
    (perror("Unable to delete directory"));

    return 0;
}
```

函数名: mktemp

功 能: 建立唯一的文件名

用 法: char *mktemp(char *template);

程序例:

```
#include <dir.h>
#include <stdio.h>

int main(void)
{
    /* fname defines the template for the
       temporary file. */

    char *fname = "TXXXXXX", *ptr;

    ptr = mktemp(fname);
    printf("%s\n", ptr);
    return 0;
}
```

函数名: MK_FP

功 能: 设置一个远指针

用 法: void far *MK_FP(unsigned seg, unsigned off);

程序例:

```
#include <dos.h>
```

```
#include <graphics.h>

int main(void)
{
    int gd, gm, i;
    unsigned int far *screen;

    detectgraph(&gd, &gm);
    if (gd == HERCMONO)
        screen = MK_FP(0xB000, 0);
    else
        screen = MK_FP(0xB800, 0);
    for (i=0; i<26; i++)
        screen[i] = 0x0700 + ('a' + i);
    return 0;
}
```

函数名: modf

功 能: 把数分为指数和尾数

用 法: double modf(double value, double *iptr);

程序例:

```
#include <math.h>
#include <stdio.h>

int main(void)
{
    double fraction, integer;
    double number = 100000.567;

    fraction = modf(number, &integer);
    printf("The whole and fractional parts of %lf are %lf and %lf\n",
        number, integer, fraction);
    return 0;
}
```

函数名: movedata

功 能: 拷贝字节

用 法: void movedata(int segsrc, int offsrc, int segdest,
int offdest, unsigned numbytes);

程序例:

```
#include <mem.h>

#define MONO_BASE 0xB000

/* saves the contents of the monochrome screen in buffer */
void save_mono_screen(char near *buffer)
{
    movedata(MONO_BASE, 0, _DS, (unsigned)buffer, 80*25*2);
}

int main(void)
{
    char buf[80*25*2];
    save_mono_screen(buf);
}
```

函数名: moverel

功 能: 将当前位置(CP)移动一相对距离

用 法: void far moverel(int dx, int dy);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    char msg[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
```

```
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

/* move the C.P. to location (20, 30) */
moveto(20, 30);

/* plot a pixel at the C.P. */
putpixel(getx(), gety(), getmaxcolor());

/* create and output a message at (20, 30) */
sprintf(msg, " (%d,%d)", getx(), gety());
outtextxy(20, 30, msg);

/* move to a point a relative distance */
/* away from the current value of C.P. */
moverel(100, 100);

/* plot a pixel at the C.P. */
putpixel(getx(), gety(), getmaxcolor());

/* create and output a message at C.P. */
sprintf(msg, " (%d,%d)", getx(), gety());
outtext(msg);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: movetext

功 能: 将屏幕文本从一个矩形区域拷贝到另一个矩形区域

用 法: int movetext(int left, int top, int right, int bottom,
int newleft, int newtop);

程序例:

```
#include <conio.h>
#include <string.h>
```

```
int main(void)
{
```

```
char *str = "This is a test string";

clrscr();
cputs(str);
getch();

movetext(1, 1, strlen(str), 2, 10, 10);
getch();

return 0;
}
```

函数名: moveto

功 能: 将 CP 移到(x, y)

用 法: void far moveto(int x, int y);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    char msg[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    /* move the C.P. to location (20, 30) */
```

```
moveto(20, 30);

/* plot a pixel at the C.P. */
putpixel(getx(), gety(), getmaxcolor());

/* create and output a message at (20, 30) */
sprintf(msg, " (%d,%d)", getx(), gety());
outtextxy(20, 30, msg);

/* move to (100, 100) */
moveto(100, 100);

/* plot a pixel at the C.P. */
putpixel(getx(), gety(), getmaxcolor());

/* create and output a message at C.P. */
sprintf(msg, " (%d,%d)", getx(), gety());
outtext(msg);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: movemem

功 能: 移动一块字节

用 法: void movemem(void *source, void *destin, unsigned len);

程序例:

```
#include <mem.h>
#include <alloc.h>
#include <stdio.h>
#include <string.h>
```

```
int main(void)
{
    char *source = "Borland International";
    char *destination;
    int length;

    length = strlen(source);
```

```
destination = malloc(length + 1);
movmem(source,destination,length);
printf("%s\n",destination);

return 0;
}
```

函数名: normvideo

功 能: 选择正常亮度字符

用 法: void normvideo(void);

程序例:

```
#include <conio.h>
```

```
int main(void)
{
    normvideo();
    cprintf("NORMAL Intensity Text\r\n");
    return 0;
}
```

函数名: nosound

功 能: 关闭 PC 扬声器

用 法: void nosound(void);

程序例:

```
/* Emits a 7-Hz tone for 10 seconds.
```

```
    True story: 7 Hz is the resonant frequency of a chicken's skull cavity.
    This was determined empirically in Australia, where a new factory
    generating 7-Hz tones was located too close to a chicken ranch:
    When the factory started up, all the chickens died.
```

```
    Your PC may not be able to emit a 7-Hz tone.
```

```
*/
```

```
int main(void)
{
    sound(7);
```



```
    delay(10000);
    nosound();
}
```

函数名: open

功 能: 打开一个文件用于读或写

用 法: int open(char *pathname, int access[, int permiss]);

程序例:

```
#include <string.h>
#include <stdio.h>
#include <fcntl.h>
#include <io.h>
```

```
int main(void)
{
    int handle;
    char msg[] = "Hello world";

    if ((handle = open("TEST.$$$", O_CREAT | O_TEXT)) == -1)
    {
        perror("Error:");
        return 1;
    }
    write(handle, msg, strlen(msg));
    close(handle);
    return 0;
}
```

函数名: outport

功 能: 输出整数到硬件端口中

用 法: void outport(int port, int value);

程序例:

```
#include <stdio.h>
#include <dos.h>
```

```
int main(void)
{
    int value = 64;
    int port = 0;
```

```
    outportb(port, value);
    printf("Value %d sent to port number %d\n", value, port);
    return 0;
}
```

函数名: outportb

功 能: 输出字节到硬件端口中

用 法: void outportb(int port, char byte);

程序例:

```
#include <stdio.h>
#include <dos.h>

int main(void)
{
    int value = 64;
    int port = 0;

    outportb(port, value);
    printf("Value %d sent to port number %d\n", value, port);
    return 0;
}
```

函数名: outtext

功 能: 在视区显示一个字符串

用 法: void far outtext(char far *textstring);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy;

    /* initialize graphics and local variables */
```

```
initgraph(&gdriver, &gmode, "");

/* read result of initialization */
errorcode = graphresult();
if (errorcode != grOk) /* an error occurred */
{
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

midx = getmaxx() / 2;
midy = getmaxy() / 2;

/* move the C.P. to the center of the screen */
moveto(midx, midy);

/* output text starting at the C.P. */
outtext("This ");
outtext("is ");
outtext("a ");
outtext("test.");

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: outtextxy

功 能: 在指定位置显示一字符串

用 法: void far outtextxy(int x, int y, char *textstring);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
```

```
int main(void)
{
```

```

/* request auto detection */
int gdriver = DETECT, gmode, errorcode;
int midx, midy;

/* initialize graphics and local variables */
initgraph( &gdriver, &gmode, "");

/* read result of initialization */
errorcode = graphresult();
if (errorcode != grOk) /* an error occurred */
{
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

midx = getmaxx() / 2;
midy = getmaxy() / 2;

/* output text at the center of the screen*/
/* Note: the C.P. doesn't get changed.*/
outtextxy(midx, midy, "This is a test.");

/* clean up */
getch();
closegraph();
return 0;
}
*****

```

函数名: parsfnm

功 能: 分析文件名

用 法: char *parsfnm (char *cmdline, struct fcb *fcbptr, int option);

程序例:

```

#include <process.h>
#include <string.h>
#include <stdio.h>
#include <dos.h>

```

```
int main(void)
```

```

{
    char line[80];
    struct fcb blk;

```

```
/* get file name */
printf("Enter drive and file name (no path - ie. a:file.dat)\n");
gets(line);

/* put file name in fcb */
if (parsfnm(line, &blk, 1) == NULL)
    printf("Error in parsfnm call\n");
else
    printf("Drive # %d   Name: %11s\n", blk.fcb_drive, blk.fcb_name);

return 0;
}
```

函数名: peek

功 能: 检查存储单元

用 法: int peek(int segment, unsigned offset);

程序例:

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>

int main(void)
{
    int value = 0;

    printf("The current status of your keyboard is:\n");
    value = peek(0x0040, 0x0017);
    if (value & 1)
        printf("Right shift on\n");
    else
        printf("Right shift off\n");

    if (value & 2)
        printf("Left shift on\n");
    else
        printf("Left shift off\n");

    if (value & 4)
        printf("Control key on\n");
    else
```

```
    printf("Control key off\n");

    if (value & 8)
        printf("Alt key on\n");
    else
        printf("Alt key off\n");

    if (value & 16)
        printf("Scroll lock on\n");
    else
        printf("Scroll lock off\n");

    if (value & 32)
        printf("Num lock on\n");
    else
        printf("Num lock off\n");

    if (value & 64)
        printf("Caps lock on\n");
    else
        printf("Caps lock off\n");

    return 0;
}
```

函数名: peekb

功 能: 检查存储单元

用 法: char peekb (int segment, unsigned offset);

程序例:

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>

int main(void)
{
    int value = 0;

    printf("The current status of your keyboard is:\n");
    value = peekb(0x0040, 0x0017);
    if (value & 1)
        printf("Right shift on\n");
```

```
    else
        printf("Right shift off\n");

    if (value & 2)
        printf("Left shift on\n");
    else
        printf("Left shift off\n");

    if (value & 4)
        printf("Control key on\n");
    else
        printf("Control key off\n");

    if (value & 8)
        printf("Alt key on\n");
    else
        printf("Alt key off\n");

    if (value & 16)
        printf("Scroll lock on\n");
    else
        printf("Scroll lock off\n");

    if (value & 32)
        printf("Num lock on\n");
    else
        printf("Num lock off\n");

    if (value & 64)
        printf("Caps lock on\n");
    else
        printf("Caps lock off\n");

    return 0;
}
```

函数名: perror

功 能: 系统错误信息

用 法: void perror(char *string);

程序例:

```
#include <stdio.h>
```

```
int main(void)
{
    FILE *fp;

    fp = fopen("perror.dat", "r");
    if (!fp)
        perror("Unable to open file for reading");
    return 0;
}
```

函数名: pieslice

功 能: 绘制并填充一个扇形

用 法: void far pieslice(int x, int stanle, int endangle, int radius);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy;
    int stangle = 45, endangle = 135, radius = 100;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
```



```
midy = getmaxy() / 2;

/* set fill style and draw a pie slice */
setfillstyle(EMPTY_FILL, getmaxcolor());
pieslice(midx, midy, stangle, endangle, radius);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: poke

功 能: 存值到一个给定存储单元

用 法: void poke(int segment, int offset, int value);

程序例:

```
#include <dos.h>
#include <conio.h>

int main(void)
{
    clrscr();
    cprintf("Make sure the scroll lock key is off and press any key\r\n");
    getch();
    poke(0x0000, 0x0417, 16);
    cprintf("The scroll lock is now on\r\n");
    return 0;
}
```

函数名: pokeb

功 能: 存值到一个给定存储单元

用 法: void pokeb(int segment, int offset, char value);

程序例:

```
#include <dos.h>
#include <conio.h>
```

```
int main(void)
{
```

```
clrscr();
cprintf("Make sure the scroll lock key is off and press any key\r\n");
getch();
pokeb(0x0000,0x0417,16);
cprintf("The scroll lock is now on\r\n");
return 0;
}
```

函数名: poly

功 能: 根据参数产生一个多项式

用 法: double poly(double x, int n, double c[]);

程序例:

```
#include <stdio.h>
#include <math.h>

/* polynomial:  x**3 - 2x**2 + 5x - 1 */

int main(void)
{
    double array[] = { -1.0, 5.0, -2.0, 1.0 };
    double result;

    result = poly(2.0, 3, array);
    printf("The polynomial: x**3 - 2.0x**2 + 5x - 1 at 2.0 is %lf\n",
           result);
    return 0;
}
```

函数名: pow

功 能: 指数函数(x 的 y 次方)

用 法: double pow(double x, double y);

程序例:

```
#include <math.h>
#include <stdio.h>

int main(void)
{
    double x = 2.0, y = 3.0;
```

```
    printf("%lf raised to %lf is %lf\n", x, y, pow(x, y));  
    return 0;  
}
```

函数名: pow10

功 能: 指数函数(10 的 p 次方)

用 法: double pow10(int p);

程序例:

```
#include <math.h>  
#include <stdio.h>  
  
int main(void)  
{  
    double p = 3.0;  
  
    printf("Ten raised to %lf is %lf\n", p, pow10(p));  
    return 0;  
}
```

函数名: printf

功 能: 产生格式化输出的函数

用 法: int printf(char *format...);

程序例:

```
#include <stdio.h>  
#include <string.h>  
  
#define I 555  
#define R 5.5  
  
int main(void)  
{  
    int i,j,k,l;  
    char buf[7];  
    char *prefix = buf;  
    char tp[20];  
    printf("prefix 6d      6o      8x      10.2e      "  
          "10.2f\n");  
    strcpy(prefix,"%");
```

```

for (i = 0; i < 2; i++)
{
    for (j = 0; j < 2; j++)
        for (k = 0; k < 2; k++)
            for (l = 0; l < 2; l++)
                {
                    if (i==0) strcat(prefix,"-");
                    if (j==0) strcat(prefix,"+");
                    if (k==0) strcat(prefix,"#");
                    if (l==0) strcat(prefix,"0");
                    printf("%5s |",prefix);
                    strcpy(tp,prefix);
                    strcat(tp,"6d |");
                    printf(tp,l);
                    strcpy(tp,"");
                    strcpy(tp,prefix);
                    strcat(tp,"6o |");
                    printf(tp,l);
                    strcpy(tp,"");
                    strcpy(tp,prefix);
                    strcat(tp,"8x |");
                    printf(tp,l);
                    strcpy(tp,"");
                    strcpy(tp,prefix);
                    strcat(tp,"10.2e |");
                    printf(tp,R);
                    strcpy(tp,prefix);
                    strcat(tp,"10.2f |");
                    printf(tp,R);
                    printf(" \n");
                    strcpy(prefix,"%");
                }
            }
    return 0;
}

```

函数名: `putc`

功 能: 输出一字符到指定流中

用 法: `int putc(int ch, FILE *stream);`

程序例:

```
#include <stdio.h>
```

```
int main(void)
{
    char msg[] = "Hello world\n";
    int i = 0;

    while (msg[i])
        putc(msg[i++], stdout);
    return 0;
}
```

函数名: `putc`

功 能: 输出字符到控制台

用 法: `int putc(int ch);`

程序例:

```
#include <stdio.h>
#include <conio.h>
```

```
int main(void)
{
    char ch = 0;

    printf("Input a string:");
    while ((ch != '\r'))
    {
        ch = getch();
        putc(ch);
    }
    return 0;
}
```

函数名: `putchar`

功 能: 在 `stdout` 上输出字符

用 法: `int putchar(int ch);`

程序例:

```
#include <stdio.h>
```

```
/* define some box-drawing characters */
```

```
#define LEFT_TOP  0xDA
#define RIGHT_TOP 0xBF
#define HORIZ     0xC4
#define VERT      0xB3
#define LEFT_BOT  0xC0
#define RIGHT_BOT 0xD9

int main(void)
{
    char i, j;

    /* draw the top of the box */
    putchar(LEFT_TOP);
    for (i=0; i<10; i++)
        putchar(HORIZ);
    putchar(RIGHT_TOP);
    putchar('\n');

    /* draw the middle */
    for (i=0; i<4; i++)
    {
        putchar(VERT);
        for (j=0; j<10; j++)
            putchar(' ');
        putchar(VERT);
        putchar('\n');
    }

    /* draw the bottom */
    putchar(LEFT_BOT);
    for (i=0; i<10; i++)
        putchar(HORIZ);
    putchar(RIGHT_BOT);
    putchar('\n');

    return 0;
}
```

函数名: putenv

功 能: 把字符串加到当前环境中

用 法: int putenv(char *envvar);

程序例:

```
#include <stdio.h>
#include <stdlib.h>
#include <alloc.h>
#include <string.h>
#include <dos.h>

int main(void)
{
    char *path, *ptr;
    int i = 0;

    /* get the current path environment */
    ptr = getenv("PATH");

    /* set up new path */
    path = malloc(strlen(ptr)+15);
    strcpy(path,"PATH=");
    strcat(path,ptr);
    strcat(path,";c:\\temp");

    /* replace the current path and display current environment */
    putenv(path);
    while (environ[i])
        printf("%s\n",environ[i++]);

    return 0;
}
```

函数名: putimage

功 能: 在屏幕上输出一个位图

用 法: void far putimage(int x, int y, void far *bitmap, int op);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

#define ARROW_SIZE 10

void draw_arrow(int x, int y);
```

```
int main(void)
{
    /* request autodetection */
    int gdriver = DETECT, gmode, errorcode;
    void *arrow;
    int x, y, maxx;
    unsigned int size;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    maxx = getmaxx();
    x = 0;
    y = getmaxy() / 2;

    /* draw the image to be grabbed */
    draw_arrow(x, y);

    /* calculate the size of the image */
    size = imagesize(x, y-ARROW_SIZE, x+(4*ARROW_SIZE), y+ARROW_SIZE);

    /* allocate memory to hold the image */
    arrow = malloc(size);

    /* grab the image */
    getimage(x, y-ARROW_SIZE, x+(4*ARROW_SIZE), y+ARROW_SIZE, arrow);

    /* repeat until a key is pressed */
    while (!kbhit())
    {
        /* erase old image */
        putimage(x, y-ARROW_SIZE, arrow, XOR_PUT);
    }
}
```



```
    x += ARROW_SIZE;
    if (x >= maxx)
        x = 0;

    /* plot new image */
    putimage(x, y-ARROW_SIZE, arrow, XOR_PUT);
}

/* clean up */
free(arrow);
closegraph();
return 0;
}

void draw_arrow(int x, int y)
{
    /* draw an arrow on the screen */
    moveto(x, y);
    linerel(4*ARROW_SIZE, 0);
    linerel(-2*ARROW_SIZE, -1*ARROW_SIZE);
    linerel(0, 2*ARROW_SIZE);
    linerel(2*ARROW_SIZE, -1*ARROW_SIZE);
}
```

函数名: putpixel

功 能: 在指定位置画一像素

用 法: void far putpixel (int x, int y, int pixelcolor);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <dos.h>

#define PIXEL_COUNT 1000
#define DELAY_TIME 100 /* in milliseconds */

int main(void)
{
    /* request autodetection */
    int gdriver = DETECT, gmode, errorcode;
```

```
int i, x, y, color, maxx, maxy, maxcolor, seed;

/* initialize graphics and local variables */
initgraph(&gdriver, &gmode, "");

/* read result of initialization */
errorcode = graphresult();
if (errorcode != grOk) /* an error occurred */
{
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

maxx = getmaxx() + 1;
maxy = getmaxy() + 1;
maxcolor = getmaxcolor() + 1;

while (!kbhit())
{
    /* seed the random number generator */
    seed = random(32767);
    srand(seed);
    for (i=0; i<PIXEL_COUNT; i++)
    {
x = random(maxx);
        y = random(maxy);
        color = random(maxcolor);
        putpixel(x, y, color);
    }

    delay(DELAY_TIME);
    srand(seed);
    for (i=0; i<PIXEL_COUNT; i++)
    {
x = random(maxx);
y = random(maxy);
color = random(maxcolor);
if (color == getpixel(x, y))
    putpixel(x, y, 0);
    }
}
```

```
/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: puts

功 能: 送一字符串到流中

用 法: int puts(char *string);

程序例:

```
#include <stdio.h>
int main(void)
{
    char string[] = "This is an example output string\n";

    puts(string);
    return 0;
}
```

函数名: puttext

功 能: 将文本从存储区拷贝到屏幕

用 法: int puttext(int left, int top, int right, int bottom, void *source);

程序例:

```
#include <conio.h>
int main(void)
{
    char buffer[512];

    /* put some text to the console */
    clrscr();
    gotoxy(20, 12);
    cprintf("This is a test. Press any key to continue ...");
    getch();

    /* grab screen contents */
    gettext(20, 12, 36, 21,buffer);
    clrscr();
}
```

```
/* put selected characters back to the screen */
gotoxy(20, 12);
puttext(20, 12, 36, 21, buffer);
getch();

return 0;
}
```

函数名: putw

功 能: 把一字符或字送到流中

用 法: int putw(int w, FILE *stream);

程序例:

```
#include <stdio.h>
#include <stdlib.h>

#define FNAME "test.$$$"

int main(void)
{
    FILE *fp;
    int word;

    /* place the word in a file */
    fp = fopen(FNAME, "wb");
    if (fp == NULL)
    {
        printf("Error opening file %s\n", FNAME);
        exit(1);
    }

    word = 94;
    putw(word, fp);
    if (ferror(fp))
        printf("Error writing to file\n");
    else
        printf("Successful write\n");
    fclose(fp);

    /* reopen the file */
    fp = fopen(FNAME, "rb");
    if (fp == NULL)
```

```

    {
        printf("Error opening file %s\n", FNAME);
        exit(1);
    }

    /* extract the word */
    word = getw(fp);
    if (ferror(fp))
        printf("Error reading file\n");
    else
        printf("Successful read: word = %d\n", word);

    /* clean up */
    fclose(fp);
    unlink(FNAME);

    return 0;
}
*****

```

函数名: qsort

功 能: 使用快速排序例程进行排序

用 法: void qsort(void *base, int nelem, int width, int (*fcmp)());

程序例:

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

```

```
int sort_function( const void *a, const void *b);
```

```
char list[5][4] = { "cat", "car", "cab", "cap", "can" };
```

```

int main(void)
{
    int x;

    qsort((void *)list, 5, sizeof(list[0]), sort_function);
    for (x = 0; x < 5; x++)
        printf("%s\n", list[x]);
    return 0;
}

```

```
int sort_function( const void *a, const void *b)
```

```
{  
    return( strcmp(a,b));  
}
```

函数名: raise

功 能: 向正在执行的程序发送一个信号

用 法: int raise(int sig);

程序例:

```
#include <signal.h>
```

```
int main(void)  
{  
    int a, b;  
  
    a = 10;  
    b = 0;  
    if (b == 0)  
        /* preempt divide by zero error */  
        raise(SIGFPE);  
    a = a / b;  
    return 0;  
}
```

函数名: rand

功 能: 随机数发生器

用 法: void rand(void);

程序例:

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
int main(void)  
{  
    int i;  
  
    printf("Ten random numbers from 0 to 99\n\n");  
    for(i=0; i<10; i++)  
        printf("%d\n", rand() % 100);  
    return 0;  
}
```

函数名: randbrd

功 能: 随机块读

用 法: int randbrd(struct fcb *fcbptr, int reccnt);

程序例:

```
#include <process.h>
#include <string.h>
#include <stdio.h>
#include <dos.h>

int main(void)
{
    char far *save_dta;
    char line[80], buffer[256];
    struct fcb blk;
    int i, result;

    /* get user input file name for dta */
    printf("Enter drive and file name (no path - i.e. a:file.dat)\n");
    gets(line);

    /* put file name in fcb */
    if (!parsfnm(line, &blk, 1))
    {
        printf("Error in call to parsfnm\n");
        exit(1);
    }
    printf("Drive #%%d   File: %%s\n\n", blk.fcb_drive, blk.fcb_name);

    /* open file with DOS FCB open file */
    bdosptr(0x0F, &blk, 0);

    /* save old dta, and set new one */
    save_dta = getdta();
    setdta(buffer);

    /* set up info for the new dta */
    blk.fcb_recsz = 128;
    blk.fcb_random = 0L;
    result = randbrd(&blk, 1);

    /* check results from randbrd */
```

```
if (!result)
    printf("Read OK\n\n");
else
{
    perror("Error during read");
    exit(1);
}

/* read in data from the new dta */
printf("The first 128 characters are:\n");
for (i=0; i<128; i++)
    putchar(buffer[i]);

/* restore previous dta */
setdta(save_dta);

return 0;
}
```

函数名: randbwr

功 能: 随机块写

用 法: int randbwr(struct fcb *fcbptr, int recnt);

程序例:

```
#include <process.h>
#include <string.h>
#include <stdio.h>
#include <dos.h>

int main(void)
{
    char far *save_dta;
    char line[80];
    char buffer[256] = "RANDBWR test!";
    struct fcb blk;
    int result;

    /* get new file name from user */
    printf("Enter a file name to create (no path - ie. a:file.dat\n");
    gets(line);

    /* parse the new file name to the dta */
    parsfnm(line, &blk, 1);
}
```



```
printf("Drive #%%d   File: %s\\n", blk.fcb_drive, blk.fcb_name);

/* request DOS services to create file */
if (bdosptr(0x16, &blk, 0) == -1)
{
    perror("Error creating file");
    exit(1);
}

/* save old dta and set new dta */
save_dta = getdta();
setdta(buffer);

/* write new records */
blk.fcb_recsz = 256;
blk.fcb_random = 0L;
result = randbwr(&blk, 1);

if (!result)
    printf("Write OK\\n");
else
{
    perror("Disk error");
    exit(1);
}

/* request DOS services to close the file */
if (bdosptr(0x10, &blk, 0) == -1)
{
    perror("Error closing file");
    exit(1);
}

/* reset the old dta */
setdta(save_dta);

return 0;
}
```

函数名: random

功 能: 随机数发生器

用 法: int random(int num);

程序例:

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>

/* prints a random number in the range 0 to 99 */
int main(void)
{
    randomize();
    printf("Random number in the 0-99 range: %d\n", random(100));
    return 0;
}
```

函数名: randomize

功 能: 初始化随机数发生器

用 法: void randomize(void);

程序例:

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>

int main(void)
{
    int i;

    randomize();
    printf("Ten random numbers from 0 to 99\n\n");
    for(i=0; i<10; i++)
        printf("%d\n", rand() % 100);
    return 0;
}
```

函数名: read

功 能: 从文件中读

用 法: int read(int handle, void *buf, int nbyte);

程序例:

```
#include <stdio.h>
```

```
#include <io.h>
#include <alloc.h>
#include <fcntl.h>
#include <process.h>
#include <sys\stat.h>

int main(void)
{
    void *buf;
    int handle, bytes;

    buf = malloc(10);

    /*
     Looks for a file in the current directory named TEST.$$$ and attempts
     to read 10 bytes from it. To use this example you should create the
     file TEST.$$$
    */
    if ((handle =
        open("TEST.$$$", O_RDONLY | O_BINARY, S_IWRITE | S_IREAD)) == -1)
    {
        printf("Error Opening File\n");
        exit(1);
    }

    if ((bytes = read(handle, buf, 10)) == -1) {
        printf("Read Failed.\n");
        exit(1);
    }
    else {
        printf("Read: %d bytes read.\n", bytes);
    }
    return 0;
}
```

函数名: realloc

功 能: 重新分配主存

用 法: void *realloc(void *ptr, unsigned newsize);

程序例:

```
#include <stdio.h>
#include <alloc.h>
```

```
#include <string.h>

int main(void)
{
    char *str;

    /* allocate memory for string */
    str = malloc(10);

    /* copy "Hello" into string */
    strcpy(str, "Hello");

    printf("String is %s\n  Address is %p\n", str, str);
    str = realloc(str, 20);
    printf("String is %s\n  New address is %p\n", str, str);

    /* free memory */
    free(str);

    return 0;
}
```

函数名: rectangle

功 能: 画一个矩形

用 法: void far rectangle(int left, int top, int right, int bottom);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int left, top, right, bottom;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
```

```
errorcode = graphresult();
if (errorcode != grOk) /* an error occurred */
{
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

left = getmaxx() / 2 - 50;
top = getmaxy() / 2 - 50;
right = getmaxx() / 2 + 50;
bottom = getmaxy() / 2 + 50;

/* draw a rectangle */
rectangle(left,top,right,bottom);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: registerbgidriver

功 能: 登录已连接进来的图形驱动程序代码

用 法: int registerbgidriver(void(*driver)(void));

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;

    /* register a driver that was added into graphics.lib */
    errorcode = registerbgidriver(EGAVGA_driver);

    /* report any registration errors */
```

```
if (errorcode < 0)
{
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

/* initialize graphics and local variables */
initgraph(&gdriver, &gmode, "");

/* read result of initialization */
errorcode = graphresult();
if (errorcode != grOk) /* an error occurred */
{
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

/* draw a line */
line(0, 0, getmaxx(), getmaxy());

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: remove

功 能: 删除一个文件

用 法: int remove(char *filename);

程序例:

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    char file[80];
```

```
    /* prompt for file name to delete */
```

```
printf("File to delete: ");
gets(file);

/* delete the file */
if (remove(file) == 0)
    printf("Removed %s.\n", file);
else
    perror("remove");

return 0;
}
```

函数名: rename

功 能: 重命名文件

用 法: int rename(char *oldname, char *newname);

程序例:

```
#include <stdio.h>

int main(void)
{
    char oldname[80], newname[80];

    /* prompt for file to rename and new name */
    printf("File to rename: ");
    gets(oldname);
    printf("New name: ");
    gets(newname);

    /* Rename the file */
    if (rename(oldname, newname) == 0)
        printf("Renamed %s to %s.\n", oldname, newname);
    else
        perror("rename");

    return 0;
}
```

函数名: restorecrtmode

功 能: 将屏幕模式恢复为先前的 imitgraph 设置

用 法: void far restorecrtmode(void);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int x, y;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    x = getmaxx() / 2;
    y = getmaxy() / 2;

    /* output a message */
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    outtextxy(x, y, "Press any key to exit graphics:");
    getch();

    /* restore system to text mode */
    restorecrtmode();
    printf("We're now in text mode.\n");
    printf("Press any key to return to graphics mode:");
    getch();

    /* return to graphics mode */
    setgraphmode(getgraphmode());
}
```



```
/* output a message */
settextjustify(CENTER_TEXT, CENTER_TEXT);
outtextxy(x, y, "We're back in graphics mode.");
outtextxy(x, y+textheight("W"), "Press any key to halt:");

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: rewind

功 能: 将文件指针重新指向一个流的开头

用 法: int rewind(FILE *stream);

程序例:

```
#include <stdio.h>
#include <dir.h>

int main(void)
{
    FILE *fp;
    char *fname = "TXXXXXX", *newname, first;

    newname = mktemp(fname);
    fp = fopen(newname, "w+");
    fprintf(fp, "abcdefghijklmnopqrstuvwxyz");
    rewind(fp);
    fscanf(fp, "%c", &first);
    printf("The first character is: %c\n", first);
    fclose(fp);
    remove(newname);

    return 0;
}
```

函数名: rmdir

功 能: 删除 DOS 文件目录

用 法: int rmdir(char *stream);

程序例:

```
#include <stdio.h>
#include <conio.h>
#include <process.h>
#include <dir.h>

#define DIRNAME "testdir.$$$"

int main(void)
{
    int stat;

    stat = mkdir(DIRNAME);
    if (!stat)
        printf("Directory created\n");
    else
    {
        printf("Unable to create directory\n");
        exit(1);
    }

    getch();
    system("dir/p");
    getch();

    stat = rmdir(DIRNAME);
    if (!stat)
        printf("\nDirectory deleted\n");
    else
    {
        perror("\nUnable to delete directory\n");
        exit(1);
    }

    return 0;
}
*****
```

函数名: sbrk

功 能: 改变数据段空间位置

用 法: char *sbrk(int incr);

程序例:

```
#include <stdio.h>
#include <alloc.h>
```

```
int main(void)
{
    printf("Changing allocation with sbrk()\n");
    printf("Before sbrk() call: %lu bytes free\n",
        (unsigned long) coreleft());
    sbrk(1000);
    printf(" After sbrk() call: %lu bytes free\n",
        (unsigned long) coreleft());
    return 0;
}
```

函数名: scanf

功 能: 执行格式化输入

用 法: int scanf(char *format[,argument,...]);

程序例:

```
#include <stdio.h>
#include <conio.h>
```

```
int main(void)
{
    char label[20];
    char name[20];
    int entries = 0;
    int loop, age;
    double salary;

    struct Entry_struct
    {
        char  name[20];
        int   age;
        float salary;
    } entry[20];

    /* Input a label as a string of characters restricting to 20 characters */
    printf("\n\nPlease enter a label for the chart: ");
    scanf("%20s", label);
    fflush(stdin); /* flush the input stream in case of bad input */

    /* Input number of entries as an integer */
    printf("How many entries will there be? (less than 20) ");
```

```
scanf("%d", &entries);
fflush(stdin); /* flush the input stream in case of bad input */

/* input a name restricting input to only letters upper or lower case */
for (loop=0;loop<entries;++loop)
{
    printf("Entry %d\n", loop);
    printf("  Name   : ");
    scanf("%[A-Za-z]", entry[loop].name);
    fflush(stdin); /* flush the input stream in case of bad input */

/* input an age as an integer */
    printf("  Age    : ");
    scanf("%d", &entry[loop].age);
    fflush(stdin); /* flush the input stream in case of bad input */

/* input a salary as a float */
    printf("  Salary : ");
    scanf("%f", &entry[loop].salary);
    fflush(stdin); /* flush the input stream in case of bad input */
}

/* Input a name, age and salary as a string, integer, and double */
printf("\nPlease enter your name, age and salary\n");
scanf("%20s %d %lf", name, &age, &salary);

/* Print out the data that was input */
printf("\n\nTable %s\n",label);
printf("Compiled by %s  age %d  $%15.2lf\n", name, age, salary);
printf("-----\n");
for (loop=0;loop<entries;++loop)
    printf("%4d | %-20s | %5d | %15.2lf\n",
        loop + 1,
        entry[loop].name,
        entry[loop].age,
        entry[loop].salary);
printf("-----\n");
return 0;
}
```

函数名: searchpath

功 能: 搜索 DOS 路径

用 法: char *searchpath(char *filename);

程序例:

```
#include <stdio.h>
#include <dir.h>

int main(void)
{
    char *p;

    /* Looks for TLINK and returns a pointer
       to the path */
    p = searchpath("TLINK.EXE");
    printf("Search for TLINK.EXE : %s\n", p);

    /* Looks for non-existent file */
    p = searchpath("NOTEXIST.FIL");
    printf("Search for NOTEXIST.FIL : %s\n", p);

    return 0;
}
```

函数名: sector

功 能: 画并填充椭圆扇区

用 法: void far sector(int x, int y, int stangle, int endangle);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy, i;
    int stangle = 45, endangle = 135;
    int xrad = 100, yrad = 50;

    /* initialize graphics and local variables */
```

```
initgraph(&gdriver, &gmode, "");

/* read result of initialization */
errorcode = graphresult();
if (errorcode != grOk) /* an error occurred */
{
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

midx = getmaxx() / 2;
midy = getmaxy() / 2;

/* loop through the fill patterns */
for (i=EMPTY_FILL; i<USER_FILL; i++)
{
    /* set the fill style */
    setfillstyle(i, getmaxcolor());

    /* draw the sector slice */
    sector(midx, midy, stangle, endangle, xrad, yrad);

    getch();
}

/* clean up */
closegraph();
return 0;
}
```

函数名: segread

功 能: 读段寄存器值

用 法: void segread(struct SREGS *segtbl);

程序例:

```
#include <stdio.h>
```

```
#include <dos.h>
```

```
int main(void)
```

```
{
    struct SREGS segs;
```

```
    segread(&segs);
    printf("Current segment register settings\n\n");
    printf("CS: %X    DS: %X\n", segs.cs, segs.ds);
    printf("ES: %X    SS: %X\n", segs.es, segs.ss);

    return 0;
}
```

函数名: setactivepage

功 能: 设置图形输出活动页

用 法: void far setactivepage(int pagenum);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* select a driver and mode that supports */
    /* multiple pages.                        */
    int gdriver = EGA, gmode = EGAHI, errorcode;
    int x, y, ht;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    x = getmaxx() / 2;
    y = getmaxy() / 2;
    ht = textheight("W");
```

```
/* select the offscreen page for drawing */
setactivepage(1);

/* draw a line on page #1 */
line(0, 0, getmaxx(), getmaxy());

/* output a message on page #1 */
settextjustify(CENTER_TEXT, CENTER_TEXT);
outtextxy(x, y, "This is page #1.");
outtextxy(x, y+ht, "Press any key to halt.");

/* select drawing to page #0 */
setactivepage(0);

/* output a message on page #0 */
outtextxy(x, y, "This is page #0.");
outtextxy(x, y+ht, "Press any key to view page #1.");
getch();

/* select page #1 as the visible page */
setvisualpage(1);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: setallpalette

功 能: 按指定方式改变所有的调色板颜色

用 法: void far setallpalette(struct palette, far *palette);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
```

```
int main(void)
{
    /* request auto detection */
```



```
int gdriver = DETECT, gmode, errorcode;
struct palettetype pal;
int color, maxcolor, ht;
int y = 10;
char msg[80];

/* initialize graphics and local variables */
initgraph(&gdriver, &gmode, "");

/* read result of initialization */
errorcode = graphresult();
if (errorcode != grOk) /* an error occurred */
{
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

maxcolor = getmaxcolor();
ht = 2 * textheight("W");

/* grab a copy of the palette */
getpalette(&pal);

/* display the default palette colors */
for (color=1; color<=maxcolor; color++)
{
    setcolor(color);
    sprintf(msg, "Color: %d", color);
    outtextxy(1, y, msg);
    y += ht;
}

/* wait for a key */
getch();

/* black out the colors one by one */
for (color=1; color<=maxcolor; color++)
{
    setpalette(color, BLACK);
    getch();
}
```

```
/* restore the palette colors */
setallpalette(&pal);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: setaspectratio

功 能: 设置图形纵横比

用 法: void far setaspectratio(int xasp, int yasp);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int xasp, yasp, midx, midy;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;
    setcolor(getmaxcolor());
```

```
/* get current aspect ratio settings */
getaspectratio(&xasp, &yasp);

/* draw normal circle */
circle(midx, midy, 100);
getch();

/* clear the screen */
cleardevice();

/* adjust the aspect for a wide circle */
setaspectratio(xasp/2, yasp);
circle(midx, midy, 100);
getch();

/* adjust the aspect for a narrow circle */
cleardevice();
setaspectratio(xasp, yasp/2);
circle(midx, midy, 100);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: setbkcolor

功 能: 用调色板设置当前背景颜色

用 法: void far setbkcolor(int color);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* select a driver and mode that supports */
    /* multiple background colors.          */
    int gdriver = EGA, gmode = EGAHI, errorcode;
    int bkcol, maxcolor, x, y;
```

```
char msg[80];

/* initialize graphics and local variables */
initgraph(&gdriver, &gmode, "");

/* read result of initialization */
errorcode = graphresult();
if (errorcode != grOk) /* an error occurred */
{
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

/* maximum color index supported */
maxcolor = getmaxcolor();

/* for centering text messages */
settextjustify(CENTER_TEXT, CENTER_TEXT);
x = getmaxx() / 2;
y = getmaxy() / 2;

/* loop through the available colors */
for (bkcol=0; bkcol<=maxcolor; bkcol++)
{
    /* clear the screen */
    cleardevice();

    /* select a new background color */
    setbkcolor(bkcol);

    /* output a message */
    if (bkcol == WHITE)
        setcolor(EGA_BLUE);
    sprintf(msg, "Background color: %d", bkcol);
    outtextxy(x, y, msg);
    getch();
}

/* clean up */
closegraph();
return 0;
}
```

函数名: `setblock`

功 能: 修改先前已分配的 DOS 存储段大小

用 法: `int setblock(int seg, int newsize);`

程序例:

```
#include <dos.h>
#include <alloc.h>
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    unsigned int size, segp;
    int stat;

    size = 64; /* (64 x 16) = 1024 bytes */
    stat = allocmem(size, &segp);
    if (stat == -1)
        printf("Allocated memory at segment: %X\n", segp);
    else
    {
        printf("Failed: maximum number of paragraphs available is %d\n",
            stat);
        exit(1);
    }

    stat = setblock(seg, size * 2);
    if (stat == -1)
        printf("Expanded memory block at segment: %X\n", segp);
    else
        printf("Failed: maximum number of paragraphs available is %d\n",
            stat);

    freemem(seg);

    return 0;
}
```

函数名: `setbuf`

功 能: 把缓冲区与流相联

用 法: void setbuf(FILE *stream, char *buf);

程序例:

```
#include <stdio.h>

/* BUFSIZ is defined in stdio.h */
char outbuf[BUFSIZ];

int main(void)
{
    /* attach a buffer to the standard output stream */
    setbuf(stdout, outbuf);

    /* put some characters into the buffer */
    puts("This is a test of buffered output.\n\n");
    puts("This output will go into outbuf\n");
    puts("and won't appear until the buffer\n");
    puts("fills up or we flush the stream.\n");

    /* flush the output buffer */
    fflush(stdout);

    return 0;
}
```

函数名: setcbkr

功 能: 设置 Control-break

用 法: int setcbkr(int value);

程序例:

```
#include <dos.h>
#include <conio.h>
#include <stdio.h>

int main(void)
{
    int break_flag;

    printf("Enter 0 to turn control break off\n");
    printf("Enter 1 to turn control break on\n");
```

```
break_flag = getch() - 0;

setcbreak(break_flag);

if (getcbreak())
    printf("Cntrl-brk flag is on\n");
else
    printf("Cntrl-brk flag is off\n");
return 0;
}
```

函数名: setcolor

功 能: 设置当前画线颜色

用 法: void far setcolor(int color);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* select a driver and mode that supports */
    /* multiple drawing colors.                */
    int gdriver = EGA, gmode = EGAHI, errorcode;
    int color, maxcolor, x, y;
    char msg[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }
}
```

```
/* maximum color index supported */
maxcolor = getmaxcolor();

/* for centering text messages */
settextjustify(CENTER_TEXT, CENTER_TEXT);
x = getmaxx() / 2;
y = getmaxy() / 2;

/* loop through the available colors */
for (color=1; color<=maxcolor; color++)
{
    /* clear the screen */
    cleardevice();

    /* select a new background color */
    setcolor(color);

    /* output a message */
    sprintf(msg, "Color: %d", color);
    outtextxy(x, y, msg);
    getch();
}

/* clean up */
closegraph();
return 0;
}
```

函数名: setdate

功 能: 设置 DOS 日期

用 法: void setdate(struct date *dateblk);

程序例:

```
#include <stdio.h>
#include <process.h>
#include <dos.h>

int main(void)
{
    struct date reset;
    struct date save_date;
```



```
    getdate(&save_date);
    printf("Original date:\n");
    system("date");

    reset.da_year = 2001;
    reset.da_day = 1;
    reset.da_mon = 1;
    setdate(&reset);

    printf("Date after setting:\n");
    system("date");

    setdate(&save_date);
    printf("Back to original date:\n");
    system("date");

    return 0;
}
```

函数名: setdisk

功 能: 设置当前磁盘驱动器

用 法: int setdisk(int drive);

程序例:

```
#include <stdio.h>
#include <dir.h>

int main(void)
{
    int save, disk, disks;

    /* save original drive */
    save = getdisk();

    /* print number of logic drives */
    disks = setdisk(save);
    printf("%d logical drives on the system\n\n", disks);

    /* print the drive letters available */
    printf("Available drives:\n");
    for (disk = 0; disk < 26; ++disk)
```

```
{
    setdisk(disk);
    if (disk == getdisk())
        printf("%c: drive is available\n", disk + 'a');
}
setdisk(save);

return 0;
}
```

函数名: setdta

功 能: 设置磁盘传输区地址

用 法: void setdta(char far *dta);

程序例:

```
#include <process.h>
#include <string.h>
#include <stdio.h>
#include <dos.h>

int main(void)
{
    char line[80], far *save_dta;
    char buffer[256] = "SETDTA test!";
    struct fcb blk;
    int result;

    /* get new file name from user */
    printf("Enter a file name to create:");
    gets(line);

    /* parse the new file name to the dta */
    parsfnm(line, &blk, 1);
    printf("%d %s\n", blk.fcb_drive, blk.fcb_name);

    /* request DOS services to create file */
    if (bdosptr(0x16, &blk, 0) == -1)
    {
        perror("Error creating file");
        exit(1);
    }
}
```

```
/* save old dta and set new dta */
save_dta = getdta();
setdta(buffer);

/* write new records */
blk.fcb_recsz = 256;
blk.fcb_random = 0L;
result = randbwr(&blk, 1);
printf("result = %d\n", result);

if (!result)
    printf("Write OK\n");
else
{
    perror("Disk error");
    exit(1);
}

/* request DOS services to close the file */
if (bdosptr(0x10, &blk, 0) == -1)
{
    perror("Error closing file");
    exit(1);
}

/* reset the old dta */
setdta(save_dta);
return 0;
}
```

函数名: setfillpattern

功 能: 选择用户定义的填充模式

用 法: void far setfillpattern(char far *upattern, int color);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
```

```
int main(void)
{
```

```
/* request auto detection */
int gdriver = DETECT, gmode, errorcode;
int maxx, maxy;

/* a user defined fill pattern */
char pattern[8] = {0x00, 0x70, 0x20, 0x27, 0x24, 0x24, 0x07, 0x00};

/* initialize graphics and local variables */
initgraph(&gdriver, &gmode, "");

/* read result of initialization */
errorcode = graphresult();
if (errorcode != grOk) /* an error occurred */
{
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

maxx = getmaxx();
maxy = getmaxy();
setcolor(getmaxcolor());

/* select a user defined fill pattern */
setfillpattern(pattern, getmaxcolor());

/* fill the screen with the pattern */
bar(0, 0, maxx, maxy);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: setfillstyle

功 能: 设置填充模式和颜色

用 法: void far setfillstyle(int pattern, int color);

程序例:

```
#include <graphics.h>
```

```
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <conio.h>

/* the names of the fill styles supported */
char *fname[] = { "EMPTY_FILL",
                  "SOLID_FILL",
                  "LINE_FILL",
                  "LTSLASH_FILL",
                  "SLASH_FILL",
                  "BKSLASH_FILL",
                  "LTBKSLASH_FILL",
                  "HATCH_FILL",
                  "XHATCH_FILL",
                  "INTERLEAVE_FILL",
                  "WIDE_DOT_FILL",
                  "CLOSE_DOT_FILL",
                  "USER_FILL"
                };

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int style, midx, midy;
    char stylestr[40];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;
```

```
for (style = EMPTY_FILL; style < USER_FILL; style++)
{
    /* select the fill style */
    setfillstyle(style, getmaxcolor());

    /* convert style into a string */
    strcpy(stylestr, fname[style]);

    /* fill a bar */
    bar3d(0, 0, midx-10, midy, 0, 0);

    /* output a message */
    outtextxy(midx, midy, stylestr);

    /* wait for a key */
    getch();
    cleardevice();
}

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: setftime

功 能: 设置文件日期和时间

用 法: int setftime(int handle, struct ftime *ftimep);

程序例:

```
#include <stdio.h>
#include <process.h>
#include <fcntl.h>
#include <io.h>
```

```
int main(void)
{
    struct ftime filet;
    FILE *fp;

    if ((fp = fopen("TEST.$$$", "w")) == NULL)
    {
```

```
        perror("Error:");
        exit(1);
    }

    fprintf(fp, "testing...\n");

    /* load ftime structure with new time and date */
    filet.ft_tsec = 1;
    filet.ft_min = 1;
    filet.ft_hour = 1;
    filet.ft_day = 1;
    filet.ft_month = 1;
    filet.ft_year = 21;

    /* show current directory for time and date */
    system("dir TEST.$$$");

    /* change the time and date stamp*/
    setftime(fileno(fp), &filet);

    /* close and remove the temporary file */
    fclose(fp);

    system("dir TEST.$$$");

    unlink("TEST.$$$");
    return 0;
}
```

函数名: setgraphbufsize

功 能: 改变内部图形缓冲区的大小

用 法: unsigned far setgraphbufsize(unsigned bufsize);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

#define BUFSIZE 1000 /* internal graphics buffer size */

int main(void)
```

```
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int x, y, oldsize;
    char msg[80];

    /* set the size of the internal graphics buffer */
    /* before making a call to initgraph.          */
    oldsize = setgraphbufsize(BUFSIZE);

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    x = getmaxx() / 2;
    y = getmaxy() / 2;

    /* output some messages */
    sprintf(msg, "Graphics buffer size: %d", BUFSIZE);
    setttextjustify(CENTER_TEXT, CENTER_TEXT);
    outtextxy(x, y, msg);
    sprintf(msg, "Old graphics buffer size: %d", oldsize);
    outtextxy(x, y+textheight("W"), msg);

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

函数名: setgraphmode

功 能: 将系统设置成图形模式且清屏

用法: void far setgraphmode(int mode);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int x, y;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    x = getmaxx() / 2;
    y = getmaxy() / 2;

    /* output a message */
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    outtextxy(x, y, "Press any key to exit graphics:");
    getch();

    /* restore system to text mode */
    restorecrtmode();
    printf("We're now in text mode.\n");
    printf("Press any key to return to graphics mode:");
    getch();

    /* return to graphics mode */
    setgraphmode(getgraphmode());
}
```

```
/* output a message */
setttextjustify(CENTER_TEXT, CENTER_TEXT);
outtextxy(x, y, "We're back in graphics mode.");
outtextxy(x, y+textheight("W"), "Press any key to halt:");

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: setjmp

功 能: 非局部转移

用 法: int setjmp(jmp_buf env);

程序例:

```
#include <stdio.h>
#include <process.h>
#include <setjmp.h>
```

```
void subroutine(void);
```

```
jmp_buf jumper;
```

```
int main(void)
{
    int value;

    value = setjmp(jumper);
    if (value != 0)
    {
        printf("Longjmp with value %d\n", value);
        exit(value);
    }
    printf("About to call subroutine ... \n");
    subroutine();
    return 0;
}
```

```
void subroutine(void)
{
```

```
    longjmp(jumper,1);  
}
```

函数名: setlinestyle

功 能: 设置当前画线宽度和类型

用 法: void far setlinestyle(int linestyle, unsigned upattern);

程序例:

```
#include <graphics.h>  
#include <stdlib.h>  
#include <string.h>  
#include <stdio.h>  
#include <conio.h>  
  
/* the names of the line styles supported */  
char *lname[] = {  
    "SOLID_LINE",  
    "DOTTED_LINE",  
    "CENTER_LINE",  
    "DASHED_LINE",  
    "USERBIT_LINE"  
};  
  
int main(void)  
{  
    /* request auto detection */  
    int gdriver = DETECT, gmode, errorcode;  
  
    int style, midx, midy, userpat;  
    char stylestr[40];  
  
    /* initialize graphics and local variables */  
    initgraph(&gdriver, &gmode, "");  
  
    /* read result of initialization */  
    errorcode = graphresult();  
    if (errorcode != grOk) /* an error occurred */  
    {  
        printf("Graphics error: %s\n", grapherrormsg(errorcode));  
        printf("Press any key to halt:");  
        getch();  
        exit(1); /* terminate with an error code */  
    }  
}
```

```
}

midx = getmaxx() / 2;
midy = getmaxy() / 2;

/* a user defined line pattern */
/* binary: "0000000000000001" */
userpat = 1;

for (style=SOLID_LINE; style<=USERBIT_LINE; style++)
{
    /* select the line style */
    setlinestyle(style, userpat, 1);

    /* convert style into a string */
    strcpy(stylestr, lname[style]);

    /* draw a line */
    line(0, 0, midx-10, midy);

    /* draw a rectangle */
    rectangle(0, 0, getmaxx(), getmaxy());

    /* output a message */
    outtextxy(midx, midy, stylestr);

    /* wait for a key */
    getch();
    cleardevice();
}

/* clean up */
closegraph();
return 0;
}
```

函数名: setmem

功 能: 存值到存储区

用 法: void setmem(void *addr, int len, char value);

程序例:

```
#include <stdio.h>
#include <alloc.h>
#include <mem.h>

int main(void)
{
    char *dest;

    dest = calloc(21, sizeof(char));
    setmem(dest, 20, 'c');
    printf("%s\n", dest);

    return 0;
}
```

函数名: setmode

功 能: 设置打开文件方式

用 法: int setmode(int handle, unsigned mode);

程序例:

```
#include <stdio.h>
#include <fcntl.h>
#include <io.h>

int main(void)
{
    int result;

    result = setmode(fileno(stdprn), O_TEXT);
    if (result == -1)
        perror("Mode not available\n");
    else
        printf("Mode successfully switched\n");
    return 0;
}
```

函数名: setpalette

功 能: 改变调色板的颜色

用法: void far setpalette(int index, int actual_color);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int color, maxcolor, ht;
    int y = 10;
    char msg[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    maxcolor = getmaxcolor();
    ht = 2 * textheight("W");

    /* display the default colors */
    for (color=1; color<=maxcolor; color++)
    {
        setcolor(color);
        sprintf(msg, "Color: %d", color);
        outtextxy(1, y, msg);
        y += ht;
    }

    /* wait for a key */
    getch();
}
```

```
/* black out the colors one by one */
for (color=1; color<=maxcolor; color++)
{
    setpalette(color, BLACK);
    getch();
}

/* clean up */
closegraph();
return 0;
}
```

函数名: setrgbpalette

功 能: 定义 IBM8514 图形卡的颜色

用 法: void far setrgbpalette(int colorm, int red, int green, int blue);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* select a driver and mode that supports the use */
    /* of the setrgbpalette function. */
    int gdriver = VGA, gmode = VGAHI, errorcode;
    struct palettetype pal;
    int i, ht, y, xmax;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }
}
```

```
/* grab a copy of the palette */
getpalette(&pal);

/* create gray scale */
for (i=0; i<pal.size; i++)
    setrgbpalette(pal.colors[i], i*4, i*4, i*4);

/* display the gray scale */
ht = getmaxy() / 16;
xmax = getmaxx();
y = 0;
for (i=0; i<pal.size; i++)
{
    setfillstyle(SOLID_FILL, i);
    bar(0, y, xmax, y+ht);
    y += ht;
}

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: `settextjustify`

功 能: 为图形函数设置文本的对齐方式

用 法: `void far settextjustify(int horiz, int vert);`

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

/* function prototype */
void xat(int x, int y);

/* horizontal text justification settings */
char *hjust[] = { "LEFT_TEXT",
                  "CENTER_TEXT",
```



```
        "RIGHT_TEXT"
    };

/* vertical text justification settings */
char *vjust[] = { "LEFT_TEXT",
    "CENTER_TEXT",
    "RIGHT_TEXT"
};

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy, hj, vj;
    char msg[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    /* loop through text justifications */
    for (hj=LEFT_TEXT; hj<=RIGHT_TEXT; hj++)
        for (vj=LEFT_TEXT; vj<=RIGHT_TEXT; vj++)
        {
            cleardevice();
            /* set the text justification */
            settextjustify(hj, vj);

            /* create a message string */
            sprintf(msg, "%s  %s", hjust[hj], vjust[vj]);

            /* create cross hairs on the screen */
```

```
xat(midx, midy);

    /* output the message */
    outtextxy(midx, midy, msg);
    getch();
}

/* clean up */
closegraph();
return 0;
}

/* draw an "x" at (x, y) */
void xat(int x, int y)
{
    line(x-4, y, x+4, y);
    line(x, y-4, x, y+4);
}
```

函数名: settextstyle

功 能: 为图形输出设置当前的文本属性

用 法: void far settextstyle (int font, int direction, char size);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

/* the names of the text styles supported */
char *fname[] = { "DEFAULT font",
                  "TRIPLEX font",
                  "SMALL font",
                  "SANS SERIF font",
                  "GOTHIC font"
                };

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int style, midx, midy;
```

```
int size = 1;

/* initialize graphics and local variables */
initgraph(&gdriver, &gmode, "");

/* read result of initialization */
errorcode = graphresult();
if (errorcode != grOk) /* an error occurred */
{
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

midx = getmaxx() / 2;
midy = getmaxy() / 2;

settextjustify(CENTER_TEXT, CENTER_TEXT);

/* loop through the available text styles */
for (style=DEFAULT_FONT; style<=GOTHIC_FONT; style++)
{
    cleardevice();
    if (style == TRIPLEX_FONT)
        size = 4;

    /* select the text style */
    settextstyle(style, HORIZ_DIR, size);

    /* output a message */
    outtextxy(midx, midy, fname[style]);
    getch();
}

/* clean up */
closegraph();
return 0;
}
```

函数名: settextstyle

功 能: 为图形输出设置当前的文本属性

用法: void far settextstyle (int font, int direction, char size);
程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

/* the names of the text styles supported */
char *fname[] = { "DEFAULT font",
                  "TRIPLEX font",
                  "SMALL font",
                  "SANS SERIF font",
                  "GOTHIC font"
                };

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int style, midx, midy;
    int size = 1;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    midx = getmaxx() / 2;
    midy = getmaxy() / 2;

    settextjustify(CENTER_TEXT, CENTER_TEXT);

    /* loop through the available text styles */
    for (style=DEFAULT_FONT; style<=GOTHIC_FONT; style++)
    {
```

```
cleardevice();
if (style == TRIPLEX_FONT)
    size = 4;

/* select the text style */
settextstyle(style, HORIZ_DIR, size);

/* output a message */
outtextxy(midx, midy, fname[style]);
getch();
}

/* clean up */
closegraph();
return 0;
}
```

函数名: settime

功 能: 设置系统时间

用 法: void settime(struct time *timep);

程序例:

```
#include <stdio.h>
#include <dos.h>

int main(void)
{
    struct time t;

    gettime(&t);
    printf("The current minute is: %d\n", t.ti_min);
    printf("The current hour is: %d\n", t.ti_hour);
    printf("The current hundredth of a second is: %d\n", t.ti_hund);
    printf("The current second is: %d\n", t.ti_sec);

    /* Add one to the minutes struct element and then call settime */
    t.ti_min++;
    settime(&t);

    return 0;
}
```

函数名: setusercharsize

功 能: 为矢量字体改变字符宽度和高度

用 法: void far setusercharsize(int multx, int dirx, int multy, int diry);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request autodetection */
    int gdriver = DETECT, gmode, errorcode;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk)      /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1);               /* terminate with an error code */
    }

    /* select a text style */
    settextstyle(TRIPLEX_FONT, HORIZ_DIR, 4);

    /* move to the text starting position */
    moveto(0, getmaxy() / 2);

    /* output some normal text */
    outtext("Norm ");

    /* make the text 1/3 the normal width */
    setusercharsize(1, 3, 1, 1);
    outtext("Short ");

    /* make the text 3 times normal width */
```

```
setusercharsize(3, 1, 1, 1);
outtext("Wide");

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: setvbuf

功 能: 把缓冲区与流相关

用 法: int setvbuf(FILE *stream, char *buf, int type, unsigned size);

程序例:

```
#include <stdio.h>

int main(void)
{
    FILE *input, *output;
    char bufi[512];

    input = fopen("file.in", "r+b");
    output = fopen("file.out", "w");

    /* set up input stream for minimal disk access,
       using our own character buffer */
    if (setvbuf(input, bufi, _IOFBF, 512) != 0)
        printf("failed to set up buffer for input file\n");
    else
        printf("buffer set up for input file\n");

    /* set up output stream for line buffering using space that
       will be obtained through an indirect call to malloc */
    if (setvbuf(output, NULL, _IOLBF, 132) != 0)
        printf("failed to set up buffer for output file\n");
    else
        printf("buffer set up for output file\n");

    /* perform file I/O here */

    /* close files */
    fclose(input);
    fclose(output);
}
```

```
    return 0;
}
```

函数名: setvect

功 能: 设置中断矢量入口

用 法: void setvect(int intr_num, void interrupt(*isr)());

程序例:

/**NOTE:

This is an interrupt service routine. You can NOT compile this program with Test Stack Overflow turned on and get an executable file which will operate correctly. */

```
#include <stdio.h>
```

```
#include <dos.h>
```

```
#include <conio.h>
```

```
#define INTR 0X1C    /* The clock tick interrupt */
```

```
void interrupt ( *oldhandler)(void);
```

```
int count=0;
```

```
void interrupt handler(void)
```

```
{
```

```
/* increase the global counter */
```

```
    count++;
```

```
/* call the old routine */
```

```
    oldhandler();
```

```
}
```

```
int main(void)
```

```
{
```

```
/* save the old interrupt vector */
```

```
    oldhandler = getvect(INTR);
```

```
/* install the new interrupt handler */
```

```
    setvect(INTR, handler);
```

```
/* loop until the counter exceeds 20 */
```



```
while (count < 20)
    printf("count is %d\n",count);

/* reset the old interrupt handler */
setvect(INTR, oldhandler);

return 0;
}
```

函数名: setverify

功 能: 设置验证状态

用 法: void setverify(int value);

程序例:

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>

int main(void)
{
    int verify_flag;

    printf("Enter 0 to set verify flag off\n");
    printf("Enter 1 to set verify flag on\n");

    verify_flag = getch() - 0;

    setverify(verify_flag);

    if (getverify())
        printf("DOS verify flag is on\n");
    else
        printf("DOS verify flag is off\n");

    return 0;
}
```

函数名: setviewport

功 能: 为图形输出设置当前视口

用 法: void far setviewport(int left, int top, int right,

```
int bottom, int clipflag);
```

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

#define CLIP_ON 1    /* activates clipping in viewport */

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    setcolor(getmaxcolor());

    /* message in default full-screen viewport */
    outtextxy(0, 0, "* <-- (0, 0) in default viewport");

    /* create a smaller viewport */
    setviewport(50, 50, getmaxx()-50, getmaxy()-50, CLIP_ON);

    /* display some text */
    outtextxy(0, 0, "* <-- (0, 0) in smaller viewport");

    /* clean up */
    getch();
    closegraph();
    return 0;
}
```

函数名: setvisualpage

功 能: 设置可见图形页号

用 法: void far setvisualpage(int pagenum);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* select a driver and mode that supports */
    /* multiple pages. */
    int gdriver = EGA, gmode = EGAHI, errorcode;
    int x, y, ht;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    x = getmaxx() / 2;
    y = getmaxy() / 2;
    ht = textheight("W");

    /* select the offscreen page for drawing */
    setactivepage(1);

    /* draw a line on page #1 */
    line(0, 0, getmaxx(), getmaxy());

    /* output a message on page #1 */
```

```
settextjustify(CENTER_TEXT, CENTER_TEXT);
outtextxy(x, y, "This is page #1:");
outtextxy(x, y+ht, "Press any key to halt:");

/* select drawing to page #0 */
setactivepage(0);

/* output a message on page #0 */
outtextxy(x, y, "This is page #0.");
outtextxy(x, y+ht, "Press any key to view page #1:");
getch();

/* select page #1 as the visible page */
setvisualpage(1);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: setwritemode

功 能: 设置图形方式下画线的输出模式

用 法: void far setwritemode(int mode);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main()
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int xmax, ymax;

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
```

```
if (errorcode != grOk) /* an error occurred */
{
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
}

xmax = getmaxx();
ymax = getmaxy();

/* select XOR drawing mode */
setwritemode(XOR_PUT);

/* draw a line */
line(0, 0, xmax, ymax);
getch();

/* erase the line by drawing over it */
line(0, 0, xmax, ymax);
getch();

/* select overwrite drawing mode */
setwritemode(COPY_PUT);

/* draw a line */
line(0, 0, xmax, ymax);

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: signal

功 能: 设置某一信号的对应动作

用 法: int signal(int sig, sigfun fname);

程序例:

```
/* This example installs a signal handler routine for SIGFPE,
   catches an integer overflow condition, makes an adjustment
   to AX register, and returns. This example program MAY cause
```

```
your computer to crash, and will produce runtime errors
depending on which memory model is used.
*/

#pragma inline
#include <stdio.h>
#include <signal.h>

void Catcher(int sig, int type, int *reglist)
{
    printf("Caught it!\n");
    *(reglist + 8) = 3;          /* make return AX = 3 */
}

int main(void)
{
    signal(SIGFPE, Catcher);
    asm    mov    ax,0FFFFH    /* AX = 32767 */
    asm    inc    ax           /* cause overflow */
    asm    into                    /* activate handler */

    /* The handler set AX to 3 on return. If that hadn't happened,
       there would have been another exception when the next 'into'
       was executed after the 'dec' instruction. */
    asm    dec    ax           /* no overflow now */
    asm    into                    /* doesn't activate */
    return 0;
}
```

函数名: sin
功 能: 正弦函数
用 法: double sin(double x);
程序例:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    double result, x = 0.5;
```

```
    result = sin(x);
    printf("The sin() of %lf is %lf\n", x, result);
    return 0;
}
```

函数名: sinh

功 能: 双曲正弦函数

用 法: double sinh(double x);

程序例:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    double result, x = 0.5;

    result = sinh(x);
    printf("The hyperbolic sin() of %lf is %lf\n", x, result);
    return 0;
}
```

函数名: sleep

功 能: 执行挂起一段时间

用 法: unsigned sleep(unsigned seconds);

程序例:

```
#include <dos.h>
#include <stdio.h>

int main(void)
{
    int i;

    for (i=1; i<5; i++)
    {
        printf("Sleeping for %d seconds\n", i);
        sleep(i);
    }
}
```

```
    return 0;
}
```

函数名: sopen

功 能: 打开一共享文件

用 法: int sopen(char *pathname, int access, int shflag, int permiss);

程序例:

```
#include <io.h>
#include <fcntl.h>
#include <sys\stat.h>
#include <process.h>
#include <share.h>
#include <stdio.h>

int main(void)
{
    int handle;
    int status;

    handle = sopen("c:\\autoexec.bat", O_RDONLY, SH_DENYNO, S_IREAD);

    if (!handle)
    {
        printf("sopen failed\n");
        exit(1);
    }

    status = access("c:\\autoexec.bat", 6);
    if (status == 0)
        printf("read/write access allowed\n");
    else
        printf("read/write access not allowed\n");

    close(handle);
    return 0;
}
```


函数名: sound

功 能: 以指定频率打开 PC 扬声器

用 法: void sound(unsigned frequency);

程序例:

```
/* Emits a 7-Hz tone for 10 seconds.
   Your PC may not be able to emit a 7-Hz tone. */
#include <dos.h>

int main(void)
{
    sound(7);
    delay(10000);
    nosound();
    return 0;
}
```

函数名: spawnl

功 能: 创建并运行子程序

用 法: int spawnl(int mode, char *pathname, char *arg0,
 arg1, ... argn, NULL);

程序例:

```
#include <process.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    int result;

    clrscr();
    result = spawnl(P_WAIT, "tcc.exe", NULL);
    if (result == -1)
    {
        perror("Error from spawnl");
        exit(1);
    }
    return 0;
}
```

函数名: spawnle

功 能: 创建并运行子程序

用 法: int spawnle(int mode, char *pathname, char *arg0,
arg1,..., argn, NULL);

程序例:

```
/* spawnle() example */
```

```
#include <process.h>
```

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main(void)
```

```
{
```

```
    int result;
```

```
    clrscr();
```

```
    result = spawnle(P_WAIT, "tcc.exe", NULL, NULL);
```

```
    if (result == -1)
```

```
    {
```

```
        perror("Error from spawnle");
```

```
        exit(1);
```

```
    }
```

```
    return 0;
```

```
}
```

函数名: sprintf

功 能: 送格式化输出到字符串中

用 法: int sprintf(char *string, char *format [,argument,...]);

程序例:

```
#include <stdio.h>
```

```
#include <math.h>
```

```
int main(void)
```

```
{
```

```
    char buffer[80];
```

```
    sprintf(buffer, "An approximation of Pi is %f\n", M_PI);
```

```
    puts(buffer);  
    return 0;  
}
```

函数名: sqrt

功 能: 计算平方根

用 法: double sqrt(double x);

程序例:

```
#include <math.h>  
#include <stdio.h>  
  
int main(void)  
{  
    double x = 4.0, result;  
  
    result = sqrt(x);  
    printf("The square root of %lf is %lf\n", x, result);  
    return 0;  
}
```

函数名: srand

功 能: 初始化随机数发生器

用 法: void srand(unsigned seed);

程序例:

```
#include <stdlib.h>  
#include <stdio.h>  
#include <time.h>  
  
int main(void)  
{  
    int i;  
    time_t t;  
  
    srand((unsigned) time(&t));  
    printf("Ten random numbers from 0 to 99\n\n");  
    for(i=0; i<10; i++)  
        printf("%d\n", rand() % 100);  
    return 0;  
}
```

函数名: sscanf

功 能: 执行从字符串中的格式化输入

用 法: int sscanf(char *string, char *format[,argument,...]);

程序例:

```
#include <stdio.h>
#include <conio.h>

int main(void)
{
    char label[20];
    char name[20];
    int entries = 0;
    int loop, age;
    double salary;

    struct Entry_struct
    {
        char   name[20];
        int    age;
        float  salary;
    } entry[20];

    /* Input a label as a string of characters restricting to 20 characters */
    printf("\n\nPlease enter a label for the chart: ");
    scanf("%20s", label);
    fflush(stdin); /* flush the input stream in case of bad input */

    /* Input number of entries as an integer */
    printf("How many entries will there be? (less than 20) ");
    scanf("%d", &entries);
    fflush(stdin); /* flush the input stream in case of bad input */

    /* input a name restricting input to only letters upper or lower case */
    for (loop=0;loop<entries;++loop)
    {
        printf("Entry %d\n", loop);
        printf("  Name    : ");
        scanf("%[A-Za-z]", entry[loop].name);
        fflush(stdin); /* flush the input stream in case of bad input */
    }
}
```

```
/* input an age as an integer */
printf(" Age : ");
scanf("%d", &entry[loop].age);
fflush(stdin); /* flush the input stream in case of bad input */

/* input a salary as a float */
printf(" Salary : ");
scanf("%f", &entry[loop].salary);
fflush(stdin); /* flush the input stream in case of bad input */
}

/* Input a name, age and salary as a string, integer, and double */
printf("\nPlease enter your name, age and salary\n");
scanf("%20s %d %lf", name, &age, &salary);

/* Print out the data that was input */
printf("\n\nTable %s\n",label);
printf("Compiled by %s age %d $%15.2lf\n", name, age, salary);
printf("-----\n");
for (loop=0;loop<entries;++loop)
    printf("%4d | %-20s | %5d | %15.2lf\n",
        loop + 1,
        entry[loop].name,
        entry[loop].age,
        entry[loop].salary);
printf("-----\n");
return 0;
}
```

函数名: stat

功 能: 读取打开文件信息

用 法: int stat(char *pathname, struct stat *buff);

程序例:

```
#include <sys/stat.h>
#include <stdio.h>
#include <time.h>

#define FILENAME "TEST.$$$"

int main(void)
```

```
{
    struct stat statbuf;
    FILE *stream;

    /* open a file for update */
    if ((stream = fopen(FILENAME, "w+")) == NULL)
    {
        fprintf(stderr, "Cannot open output file.\n");
        return(1);
    }

    /* get information about the file */
    stat(FILENAME, &statbuf);

    fclose(stream);

    /* display the information returned */
    if (statbuf.st_mode & S_IFCHR)
        printf("Handle refers to a device.\n");
    if (statbuf.st_mode & S_IFREG)
        printf("Handle refers to an ordinary file.\n");
    if (statbuf.st_mode & S_IRREAD)
        printf("User has read permission on file.\n");
    if (statbuf.st_mode & S_IWRITE)
        printf("User has write permission on file.\n");

    printf("Drive letter of file: %c\n", 'A'+statbuf.st_dev);
    printf("Size of file in bytes: %ld\n", statbuf.st_size);
    printf("Time file last opened: %s\n", ctime(&statbuf.st_ctime));
    return 0;
}
```

函数名: _status87

功 能: 取浮点状态

用 法: unsigned int _status87(void);

程序例:

```
#include <stdio.h>
```

```
#include <float.h>
```

```
int main(void)
```

```
{
    float x;
    double y = 1.5e-100;

    printf("Status 87 before error: %x\n", _status87());

    x = y;  /* <-- force an error to occur */
    y = x;

    printf("Status 87 after error : %x\n", _status87());
    return 0;
}
```

函数名: stime

功 能: 设置时间

用 法: int stime(long *tp);

程序例:

```
#include <stdio.h>
#include <time.h>
#include <dos.h>

int main(void)
{
    time_t t;
    struct tm *area;

    t = time(NULL);
    area = localtime(&t);
    printf("Number of seconds since 1/1/1970 is: %ld\n", t);
    printf("Local time is: %s", asctime(area));

    t++;
    area = localtime(&t);
    printf("Add a second:  %s", asctime(area));

    t += 60;
    area = localtime(&t);
    printf("Add a minute:  %s", asctime(area));

    t += 3600;
    area = localtime(&t);
```

```
printf("Add an hour:   %s", asctime(area));

t += 86400L;
area = localtime(&t);
printf("Add a day:     %s", asctime(area));

t += 2592000L;
area = localtime(&t);
printf("Add a month:   %s", asctime(area));

t += 31536000L;
area = localtime(&t);
printf("Add a year:    %s", asctime(area));
return 0;
}
```

函数名: stpcpy

功 能: 拷贝一个字符串到另一个

用 法: char *stpcpy(char *destin, char *source);

程序例:

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char string[10];
    char *str1 = "abcdefghi";

    stpcpy(string, str1);
    printf("%s\n", string);
    return 0;
}
```

函数名: strcat

功 能: 字符串拼接函数

用 法: char *strcat(char *destin, char *source);

程序例:


```
#include <string.h>
#include <stdio.h>

int main(void)
{
    char destination[25];
    char *blank = " ", *c = "C++", *Borland = "Borland";

    strcpy(destination, Borland);
    strcat(destination, blank);
    strcat(destination, c);

    printf("%s\n", destination);
    return 0;
}
```

函数名: strchr

功 能: 在一个串中查找给定字符的第一个匹配之处\

用 法: char *strchr(char *str, char c);

程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    char string[15];
    char *ptr, c = 'r';

    strcpy(string, "This is a string");
    ptr = strchr(string, c);
    if (ptr)
        printf("The character %c is at position: %d\n", c, ptr-string);
    else
        printf("The character was not found\n");
    return 0;
}
```

函数名: strcmp

功 能: 串比较

用 法: int strcmp(char *str1, char *str2);

程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    char *buf1 = "aaa", *buf2 = "bbb", *buf3 = "ccc";
    int ptr;

    ptr = strcmp(buf2, buf1);
    if (ptr > 0)
        printf("buffer 2 is greater than buffer 1\n");
    else
        printf("buffer 2 is less than buffer 1\n");

    ptr = strcmp(buf2, buf3);
    if (ptr > 0)
        printf("buffer 2 is greater than buffer 3\n");
    else
        printf("buffer 2 is less than buffer 3\n");

    return 0;
}
```

函数名: strncmpi

功 能: 将一个串中的一部分与另一个串比较, 不管大小写

用 法: int strncmpi(char *str1, char *str2, unsigned maxlen);

程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    char *buf1 = "BBB", *buf2 = "bbb";
    int ptr;
```

```
ptr = strcmpi(buf2, buf1);

if (ptr > 0)
    printf("buffer 2 is greater than buffer 1\n");

if (ptr < 0)
    printf("buffer 2 is less than buffer 1\n");

if (ptr == 0)
    printf("buffer 2 equals buffer 1\n");

return 0;
}
```

函数名: strcpy

功 能: 串拷贝

用 法: char *strcpy(char *str1, char *str2);

程序例:

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char string[10];
    char *str1 = "abcdefghi";

    strcpy(string, str1);
    printf("%s\n", string);
    return 0;
}
```

函数名: strcspn

功 能: 在串中查找第一个给定字符集内容的段

用 法: int strcspn(char *str1, char *str2);

程序例:

```
#include <stdio.h>
#include <string.h>
#include <alloc.h>

int main(void)
{
    char *string1 = "1234567890";
    char *string2 = "747DC8";
    int length;

    length = strcspn(string1, string2);
    printf("Character where strings intersect is at position %d\n", length);

    return 0;
}
```

函数名: strdup

功 能: 将串拷贝到新建的位置处

用 法: char *strdup(char *str);

程序例:

```
#include <stdio.h>
#include <string.h>
#include <alloc.h>

int main(void)
{
    char *dup_str, *string = "abcde";

    dup_str = strdup(string);
    printf("%s\n", dup_str);
    free(dup_str);

    return 0;
}
```

函数名: stricmp

功 能: 以大小写不敏感方式比较两个串

用 法: `int strcmp(char *str1, char *str2);`

程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    char *buf1 = "BBB", *buf2 = "bbb";
    int ptr;

    ptr = strcmp(buf2, buf1);

    if (ptr > 0)
        printf("buffer 2 is greater than buffer 1\n");

    if (ptr < 0)
        printf("buffer 2 is less than buffer 1\n");

    if (ptr == 0)
        printf("buffer 2 equals buffer 1\n");

    return 0;
}
```

函数名: `strerror`

功 能: 返回指向错误信息字符串的指针

用 法: `char *strerror(int errnum);`

程序例:

```
#include <stdio.h>
#include <errno.h>

int main(void)
{
    char *buffer;
    buffer = strerror(errno);
    printf("Error: %s\n", buffer);
    return 0;
}
```

函数名: strcmpi

功 能: 将一个串与另一个比较, 不管大小写

用 法: int strcmpi(char *str1, char *str2);

程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    char *buf1 = "BBB", *buf2 = "bbb";
    int ptr;

    ptr = strcmpi(buf2, buf1);

    if (ptr > 0)
        printf("buffer 2 is greater than buffer 1\n");

    if (ptr < 0)
        printf("buffer 2 is less than buffer 1\n");

    if (ptr == 0)
        printf("buffer 2 equals buffer 1\n");

    return 0;
}
```

函数名: strncmp

功 能: 串比较

用 法: int strncmp(char *str1, char *str2, int maxlen);

程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    char *buf1 = "aaabbb", *buf2 = "bbbcc", *buf3 = "ccc";
```

```
int ptr;

ptr = strcmp(buf2,buf1,3);
if (ptr > 0)
    printf("buffer 2 is greater than buffer 1\n");
else
    printf("buffer 2 is less than buffer 1\n");

ptr = strcmp(buf2,buf3,3);
if (ptr > 0)
    printf("buffer 2 is greater than buffer 3\n");
else
    printf("buffer 2 is less than buffer 3\n");

return(0);
}
```

函数名: strcmpi

功 能: 把串中的一部分与另一串中的一部分比较, 不管大小写

用 法: int strcmpi(char *str1, char *str2);

程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    char *buf1 = "BBBccc", *buf2 = "bbbccc";
    int ptr;

    ptr = strcmpi(buf2,buf1,3);

    if (ptr > 0)
        printf("buffer 2 is greater than buffer 1\n");

    if (ptr < 0)
        printf("buffer 2 is less than buffer 1\n");

    if (ptr == 0)
        printf("buffer 2 equals buffer 1\n");

    return 0;
}
```

```
}
```

函数名: stncpy

功 能: 串拷贝

用 法: char *stncpy(char *destin, char *source, int maxlen);

程序例:

```
#include <stdio.h>
#include <string.h>
```

```
int main(void)
{
    char string[10];
    char *str1 = "abcdefghi";

    stncpy(string, str1, 3);
    string[3] = '\0';
    printf("%s\n", string);
    return 0;
}
```

函数名: stnicmp

功 能: 不注重大小写地比较两个串

用 法: int stnicmp(char *str1, char *str2, unsigned maxlen);

程序例:

```
#include <string.h>
#include <stdio.h>
```

```
int main(void)
{
    char *buf1 = "BBBccc", *buf2 = "bbbccc";
    int ptr;

    ptr = stnicmp(buf2, buf1, 3);

    if (ptr > 0)
        printf("buffer 2 is greater than buffer 1\n");

    if (ptr < 0)
```



```
printf("buffer 2 is less than buffer 1\n");

if (ptr == 0)
    printf("buffer 2 equals buffer 1\n");

return 0;
}
```

函数名: `strset`

功 能: 将一个串中的所有字符都设为指定字符

用 法: `char *strset(char *str, char ch, unsigned n);`

程序例:

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char *string = "abcdefghijklmnopqrstuvwxyz";
    char letter = 'x';

    printf("string before strset: %s\n", string);
    strset(string, letter, 13);
    printf("string after strset: %s\n", string);

    return 0;
}
```

函数名: `strpbrk`

功 能: 在串中查找给定字符集中的字符

用 法: `char *strpbrk(char *str1, char *str2);`

程序例:

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char *string1 = "abcdefghijklmnopqrstuvwxyz";
```

```
char *string2 = "onm";
char *ptr;

ptr = strpbrk(string1, string2);

if (ptr)
    printf("strpbrk found first character: %c\n", *ptr);
else
    printf("strpbrk didn't find character in set\n");

return 0;
}
```

函数名: strchr

功 能: 在串中查找指定字符的最后一个出现

用 法: char *strchr(char *str, char c);

程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    char string[15];
    char *ptr, c = 'r';

    strcpy(string, "This is a string");
    ptr = strchr(string, c);
    if (ptr)
        printf("The character %c is at position: %d\n", c, ptr-string);
    else
        printf("The character was not found\n");
    return 0;
}
```

函数名: strev

功 能: 串倒转

用 法: char *strev(char *str);

程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    char *forward = "string";

    printf("Before strrev(): %s\n", forward);
    strrev(forward);
    printf("After strrev():  %s\n", forward);
    return 0;
}
```

函数名: `strset`

功 能: 将一个串中的所有字符都设为指定字符

用 法: `char *strset(char *str, char c);`

程序例:

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char string[10] = "123456789";
    char symbol = 'c';

    printf("Before strset(): %s\n", string);
    strset(string, symbol);
    printf("After strset():  %s\n", string);
    return 0;
}
```

函数名: `strspn`

功 能: 在串中查找指定字符集的子集的第一次出现

用 法: `int strspn(char *str1, char *str2);`

程序例:

```
#include <stdio.h>
```

```
#include <string.h>
#include <alloc.h>

int main(void)
{
    char *string1 = "1234567890";
    char *string2 = "123DC8";
    int length;

    length = strspn(string1, string2);
    printf("Character where strings differ is at position %d\n", length);
    return 0;
}
```

函数名: strstr

功 能: 在串中查找指定字符串的第一次出现

用 法: char *strstr(char *str1, char *str2);

程序例:

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char *str1 = "Borland International", *str2 = "nation", *ptr;

    ptr = strstr(str1, str2);
    printf("The substring is: %s\n", ptr);
    return 0;
}
```

函数名: strtod

功 能: 将字符串转换为 double 型值

用 法: double strtod(char *str, char **endptr);

程序例:

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
```

```
{
    char input[80], *endptr;
    double value;

    printf("Enter a floating point number:");
    gets(input);
    value = strtod(input, &endptr);
    printf("The string is %s the number is %lf\n", input, value);
    return 0;
}
```

函数名: strtok

功 能: 查找由在第二个串中指定的分界符分隔开的单词

用 法: char *strtok(char *str1, char *str2);

程序例:

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    char input[16] = "abc,d";
    char *p;

    /* strtok places a NULL terminator
    in front of the token, if found */
    p = strtok(input, ",");
    if (p)    printf("%s\n", p);

    /* A second call to strtok using a NULL
    as the first parameter returns a pointer
    to the character following the token */
    p = strtok(NULL, ",");
    if (p)    printf("%s\n", p);
    return 0;
}
```

函数名: strtol

功 能: 将串转换为长整数

用 法: long strtol(char *str, char **endptr, int base);

程序例:

```
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
    char *string = "87654321", *endptr;
    long lnumber;

    /* strtol converts string to long integer */
    lnumber = strtol(string, &endptr, 10);
    printf("string = %s    long = %ld\n", string, lnumber);

    return 0;
}
```

函数名:strupr

功 能: 将串中的小写字母转换为大写字母

用 法: char *strupr(char *str);

程序例:

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char *string = "abcdefghijklmnopqrstuvwxyz", *ptr;

    /* converts string to upper case characters */
    ptr = strupr(string);
    printf("%s\n", ptr);
    return 0;
}
```

函数名:swab

功 能: 交换字节

用 法: void swab (char *from, char *to, int nbytes);

程序例:

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

char source[15] = "rFna koBlrna d";
char target[15];

int main(void)
{
    swab(source, target, strlen(source));
    printf("This is target: %s\n", target);
    return 0;
}
```

函数名: system

功 能: 发出一个 DOS 命令

用 法: int system(char *command);

程序例:

```
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
    printf("About to spawn command.com and run a DOS command\n");
    system("dir");
    return 0;
}

*****
```

函数名: tan

功 能: 正切函数

用 法: double tan(double x);

程序例:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
```

```
double result, x;

x = 0.5;
result = tan(x);
printf("The tan of %lf is %lf\n", x, result);
return 0;
}
```

函数名: **tanh**

功 能: 双曲正切函数

用 法: `double tanh(double x);`

程序例:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    double result, x;

    x = 0.5;
    result = tanh(x);
    printf("The hyperbolic tangent of %lf is %lf\n", x, result);
    return 0;
}
```

函数名: **tell**

功 能: 取文件指针的当前位置

用 法: `long tell(int handle);`

程序例:

```
#include <string.h>
#include <stdio.h>
#include <fcntl.h>
#include <io.h>
```

```
int main(void)
```



```
{
    int handle;
    char msg[] = "Hello world";

    if ((handle = open("TEST.$$$", O_CREAT | O_TEXT | O_APPEND)) == -1)
    {
        perror("Error:");
        return 1;
    }
    write(handle, msg, strlen(msg));
    printf("The file pointer is at byte %ld\n", tell(handle));
    close(handle);
    return 0;
}
```

函数名: textattr

功 能: 设置文本属性

用 法: void textattr(int attribute);

程序例:

```
#include <conio.h>
```

```
int main(void)
{
    int i;

    clrscr();
    for (i=0; i<9; i++)
    {
        textattr(i + ((i+1) << 4));
        cprintf("This is a test\r\n");
    }

    return 0;
}
```

函数名: textbackground

功 能: 选择新的文本背景颜色

用 法: void textbackground(int color);

程序例:

```
#include <conio.h>

int main(void)
{
    int i, j;

    clrscr();
    for (i=0; i<9; i++)
    {
        for (j=0; j<80; j++)
            cprintf("C");
        cprintf("\r\n");
        textcolor(i+1);
        textbackground(i);
    }

    return 0;
}
```

函数名: textcolor

功 能: 在文本模式中选择新的字符颜色

用 法: void textcolor(int color);

程序例:

```
#include <conio.h>

int main(void)
{
    int i;

    for (i=0; i<15; i++)
    {
        textcolor(i);
        cprintf("Foreground Color\r\n");
    }

    return 0;
}
```

函数名: `textheight`

功 能: 返回以像素为单位的字符串高度

用 法: `int far textheight(char far *textstring);`

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int y = 0;
    int i;
    char msg[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    /* draw some text on the screen */
    for (i=1; i<11; i++)
    {
        /* select the text style, direction, and size */
        settextstyle(TRIPLEX_FONT, HORIZ_DIR, i);

        /* create a message string */
        sprintf(msg, "Size: %d", i);
```

```
    /* output the message */
    outtextxy(1, y, msg);

    /* advance to the next text line */
    y += textheight(msg);
}

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: textmode

功 能: 将屏幕设置成文本模式

用 法: void textmode(int mode);

程序例:

```
#include <conio.h>
```

```
int main(void)
{
    textmode(BW40);
    cprintf("ABC");
    getch();

    textmode(C40);
    cprintf("ABC");
    getch();

    textmode(BW80);
    cprintf("ABC");
    getch();

    textmode(C80);
    cprintf("ABC");
    getch();

    textmode(MONO);
    cprintf("ABC");
    getch();
}
```

```
    return 0;
}
```

函数名: textwidth

功 能: 返回以像素为单位的字符串宽度

用 法: int far textwidth(char far *textstring);

程序例:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int x = 0, y = 0;
    int i;
    char msg[80];

    /* initialize graphics and local variables */
    initgraph(&gdriver, &gmode, "");

    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1); /* terminate with an error code */
    }

    y = getmaxy() / 2;

    settextjustify(LEFT_TEXT, CENTER_TEXT);
    for (i=1; i<11; i++)
    {
        /* select the text style, direction, and size */
        settextstyle(TRIPLEX_FONT, HORIZ_DIR, i);
```

```
/* create a message string */
sprintf(msg, "Size: %d", i);

/* output the message */
outtextxy(x, y, msg);

/* advance to the end of the text */
x += textwidth(msg);
}

/* clean up */
getch();
closegraph();
return 0;
}
```

函数名: time

功 能: 取一天的时间

用 法: long time(long *tloc);

程序例:

```
#include <time.h>
#include <stdio.h>
#include <dos.h>

int main(void)
{
    time_t t;

    t = time(NULL);
    printf("The number of seconds since January 1, 1970 is %ld",t);
    return 0;
}
```

函数名: tmpfile

功 能: 以二进制方式打开暂存文件

用 法: FILE *tmpfile(void);

程序例:

```
#include <stdio.h>
#include <process.h>

int main(void)
{
    FILE *tempfp;

    tempfp = tmpfile();
    if (tempfp)
        printf("Temporary file created\n");
    else
    {
        printf("Unable to create temporary file\n");
        exit(1);
    }

    return 0;
}
```

函数名: tmpnam

功 能: 创建一个唯一的文件名

用 法: char *tmpnam(char *sptr);

程序例:

```
#include <stdio.h>

int main(void)
{
    char name[13];

    tmpnam(name);
    printf("Temporary name: %s\n", name);
    return 0;
}
```

函数名: tolower

功 能: 把字符转换成小写字母

用 法: int tolower(int c);

程序例:

```
#include <string.h>
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int length, i;
    char *string = "THIS IS A STRING";

    length = strlen(string);
    for (i=0; i<length; i++)
    {
        string[i] = tolower(string[i]);
    }
    printf("%s\n",string);

    return 0;
}
```

函数名: toupper

功 能: 把字符转换成大写字母

用 法: int toupper(int c);

程序例:

```
#include <string.h>
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int length, i;
    char *string = "this is a string";

    length = strlen(string);
    for (i=0; i<length; i++)
    {
        string[i] = toupper(string[i]);
    }
}
```



```
printf("%s\n",string);

return 0;
}
```

函数名: tzset

功 能: UNIX 时间兼容函数

用 法: void tzset(void);

程序例:

```
#include <time.h>
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
    time_t td;

    putenv("TZ=PST8PDT");
    tzset();
    time(&td);
    printf("Current time = %s\n", asctime(localtime(&td)));
    return 0;
}
*****
```

函数名: ultoa

功 能: 转换一个无符号长整型数为字符串

用 法: char *ultoa(unsigned long value, char *string, int radix);

程序例:

```
#include <stdlib.h>
#include <stdio.h>

int main( void )
{
    unsigned long lnumber = 3123456789L;
    char string[25];

    ultoa(lnumber,string,10);
    printf("string = %s   unsigned long = %lu\n",string,lnumber);

    return 0;
}
```

```
}
```

函数名: `ungetc`

功 能: 把一个字符退回到输入流中

用 法: `int ungetc(char c, FILE *stream);`

程序例:

```
#include <stdio.h>
#include <ctype.h>

int main( void )
{
    int i=0;
    char ch;

    puts("Input an integer followed by a char:");

    /* read chars until non digit or EOF */
    while((ch = getchar()) != EOF && isdigit(ch))
        i = 10 * i + ch - 48; /* convert ASCII into int value */

    /* if non digit char was read, push it back into input buffer */
    if (ch != EOF)
        ungetc(ch, stdin);

    printf("i = %d, next char in buffer = %c\n", i, getchar());
    return 0;
}
```

函数名: `ungetch`

功 能: 把一个字符退回到键盘缓冲区中

用 法: `int ungetch(int c);`

程序例:

```
#include <stdio.h>
#include <ctype.h>
#include <conio.h>
```

```
int main( void )
{
    int i=0;
    char ch;

    puts("Input an integer followed by a char:");

    /* read chars until non digit or EOF */
    while((ch = getche()) != EOF && isdigit(ch))
        i = 10 * i + ch - 48; /* convert ASCII into int value */

    /* if non digit char was read, push it back into input buffer */
    if (ch != EOF)
        ungetch(ch);

    printf("\n\ni = %d, next char in buffer = %c\n", i, getch());
    return 0;
}
```

函数名: unixtodos

功 能: 把日期和时间转换成 DOS 格式

用 法: void unixtodos(long utime, struct date *dateptr,
struct time *timeptr);

程序例:

```
#include <stdio.h>
#include <dos.h>

char *month[] = {"---", "Jan", "Feb", "Mar", "Apr", "May", "Jun",
                 "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"};

#define SECONDS_PER_DAY 86400L /* the number of seconds in one day */

struct date dt;
struct time tm;

int main(void)
{
    unsigned long val;

    /* get today's date and time */
```

```
    getdate(&dt);
    gettime(&tm);
    printf("today is %d %s %d\n", dt.da_day, month[dt.da_mon], dt.da_year);

/* convert date and time to unix format (number of seconds since Jan 1, 1970 */
    val = dostounix(&dt, &tm);
/* subtract 42 days worth of seconds */
    val -= (SECONDS_PER_DAY * 42);

/* convert back to dos time and date */
    unixtodos(val, &dt, &tm);
    printf("42 days ago it was %d %s %d\n",
           dt.da_day, month[dt.da_mon], dt.da_year);
    return 0;
}
```

函数名: unlink

功 能: 删掉一个文件

用 法: int unlink(char *filename);

程序例:

```
#include <stdio.h>
#include <io.h>

int main(void)
{
    FILE *fp = fopen("junk.jnk", "w");
    int status;

    fprintf(fp, "junk");

    status = access("junk.jnk", 0);
    if (status == 0)
        printf("File exists\n");
    else
        printf("File doesn't exist\n");

    fclose(fp);
    unlink("junk.jnk");
    status = access("junk.jnk", 0);
    if (status == 0)
```

```
    printf("File exists\n");
else
    printf("File doesn't exist\n");

return 0;
}
```

函数名: unlock

功 能: 解除文件共享锁

用 法: int unlock(int handle, long offset, long length);

程序例:

```
#include <io.h>
#include <fcntl.h>
#include <sys\stat.h>
#include <process.h>
#include <share.h>
#include <stdio.h>

int main(void)
{
    int handle, status;
    long length;

    handle = sopen("c:\\autoexec.bat",O_RDONLY,SH_DENYNO,S_IREAD);

    if (handle < 0)
    {
        printf("sopen failed\n");
        exit(1);
    }

    length = filelength(handle);
    status = lock(handle,0L,length/2);

    if (status == 0)
        printf("lock succeeded\n");
    else
        printf("lock failed\n");
}
```

```
status = unlock(handle, 0L, length/2);

if (status == 0)
    printf("unlock succeeded\n");
else
    printf("unlock failed\n");

close(handle);
return 0;
}
*****
```

函数名: vfprintf

功 能: 送格式化输出到一流中

用 法: int vfprintf(FILE *stream, char *format, va_list param);

程序例:

```
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
```

```
FILE *fp;
```

```
int vfprf(char *fmt, ...)
```

```
{
    va_list argptr;
    int cnt;

    va_start(argptr, fmt);
    cnt = vfprintf(fp, fmt, argptr);
    va_end(argptr);

    return(cnt);
}
```

```
int main(void)
```

```
{
    int inumber = 30;
    float fnumber = 90.0;
    char string[4] = "abc";

    fp = tmpfile();
    if (fp == NULL)
    {
        perror("tmpfile() call");
    }
}
```

```
        exit(1);
    }

    vfprintf(fp, "%d %f %s", inumber, fnumber, string);
    rewind(fp);
    fscanf(fp, "%d %f %s", &inumber, &fnumber, string);
    printf("%d %f %s\n", inumber, fnumber, string);
    fclose(fp);

    return 0;
}
```

函数名: vfscanf

功 能: 从流中执行格式化输入

用 法: int vfscanf(FILE *stream, char *format, va_list param);

程序例:

```
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
```

```
FILE *fp;
```

```
int vfprintf(char *fmt, ...)
```

```
{
    va_list argptr;
    int cnt;

    va_start(argptr, fmt);
    cnt = vfscanf(fp, fmt, argptr);
    va_end(argptr);

    return(cnt);
}
```

```
int main(void)
```

```
{
    int inumber = 30;
    float fnumber = 90.0;
    char string[4] = "abc";
```

```
fp = tmpfile();
if (fp == NULL)
{
    perror("tmpfile() call");
    exit(1);
}
fprintf(fp,"%d %f %s\n",inumber,fnumber,string);
rewind(fp);

vfprintf(fp,"%d %f %s",&inumber,&fnumber,string);
printf("%d %f %s\n",inumber,fnumber,string);
fclose(fp);

return 0;
}
```

函数名: vprintf

功 能: 送格式化输出到 stdout 中

用 法: int vprintf(char *format, va_list param);

程序例:

```
#include <stdio.h>
#include <stdarg.h>

int vprintf(char *fmt, ...)
{
    va_list argptr;
    int cnt;

    va_start(argptr, format);
    cnt = vprintf(fmt, argptr);
    va_end(argptr);

    return(cnt);
}

int main(void)
{
    int inumber = 30;
    float fnumber = 90.0;
    char *string = "abc";
```



```
    vpf("%d %f %s\n",inumber,fnumber,string);

    return 0;
}
```

函数名: vscanf

功 能: 从 stdin 中执行格式化输入

用 法: int vscanf(char *format, va_list param);

程序例:

```
#include <stdio.h>
#include <conio.h>
#include <stdarg.h>

int vsconf(char *fmt, ...)
{
    va_list argptr;
    int cnt;

    printf("Enter an integer, a float, and a string (e.g. i,f,s.)\n");
    va_start(argptr, fmt);
    cnt = vsconf(fmt, argptr);
    va_end(argptr);

    return(cnt);
}

int main(void)
{
    int inumber;
    float fnumber;
    char string[80];

    vsconf("%d, %f, %s", &inumber, &fnumber, string);
    printf("%d %f %s\n", inumber, fnumber, string);

    return 0;
}
```

函数名: vsprintf

功 能: 送格式化输出到串中

用 法: int vsprintf(char *string, char *format, va_list param);

程序例:

```
#include <stdio.h>
#include <conio.h>
#include <stdarg.h>
```

```
char buffer[80];
```

```
int vspvf(char *fmt, ...)
{
    va_list argptr;
    int cnt;

    va_start(argptr, fmt);
    cnt = vsprintf(buffer, fmt, argptr);
    va_end(argptr);

    return(cnt);
}
```

```
int main(void)
{
    int inumber = 30;
    float fnumber = 90.0;
    char string[4] = "abc";

    vspvf("%d %f%s", inumber, fnumber, string);
    printf("%s\n", buffer);
    return 0;
}
```

函数名: vscanf

功 能: 从流中执行格式化输入

用 法: int vscanf(char *s, char *format, va_list param);

程序例:

```
#include <stdio.h>
#include <conio.h>
```

```
#include <stdarg.h>
```

```
char buffer[80] = "30 90.0 abc";
```

```
int vssf(char *fmt, ...)
```

```
{
```

```
    va_list argptr;
```

```
    int cnt;
```

```
    fflush(stdin);
```

```
    va_start(argptr, fmt);
```

```
    cnt = vsscanf(buffer, fmt, argptr);
```

```
    va_end(argptr);
```

```
    return(cnt);
```

```
}
```

```
int main(void)
```

```
{
```

```
    int inumber;
```

```
    float fnumber;
```

```
    char string[80];
```

```
    vssf("%d %f %s", &inumber, &fnumber, string);
```

```
    printf("%d %f %s\n", inumber, fnumber, string);
```

```
    return 0;
```

```
}
```

```
*****
```

函数名: wherex

功 能: 返回窗口内水平光标位置

用 法: int wherex(void);

程序例:

```
#include <conio.h>
```

```
int main(void)
```

```
{
```

```
    clrscr();
```

```
    gotoxy(10,10);
```

```
    cprintf("Current location is X: %d   Y: %d\r\n", wherex(), wherey());
```

```
    getch();
```

```
    return 0;
```

```
}
```

函数名: wherey

功 能: 返回窗口内垂直光标位置

用 法: int wherey(void);

程序例:

```
#include <conio.h>
```

```
int main(void)
{
    clrscr();
    gotoxy(10,10);
    printf("Current location is X: %d   Y: %d\r\n", wherex(), wherey());
    getch();

    return 0;
}
```

函数名: window

功 能: 定义活动文本模式窗口

用 法: void window(int left, int top, int right, int bottom);

程序例:

```
#include <conio.h>
```

```
int main(void)
{

    window(10,10,40,11);
    textcolor(BLACK);
    textbackground(WHITE);
    printf("This is a test\r\n");

    return 0;
}
```

函数名: write

功 能: 写到一文件中

用 法: int write(int handel, void *buf, int nbyte);

程序例:

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys\stat.h>
#include <io.h>
#include <string.h>

int main(void)
{
    int handle;
    char string[40];
    int length, res;

    /*
     Create a file named "TEST.$$$" in the current directory and write
     a string to it.  If "TEST.$$$" already exists, it will be overwritten.
    */

    if ((handle = open("TEST.$$$", O_WRONLY | O_CREAT | O_TRUNC,
                      S_IRREAD | S_IWRITE)) == -1)
    {
        printf("Error opening file.\n");
        exit(1);
    }

    strcpy(string, "Hello, world!\n");
    length = strlen(string);

    if ((res = write(handle, string, length)) != length)
    {
        printf("Error writing to the file.\n");
        exit(1);
    }
    printf("Wrote %d bytes to the file.\n", res);

    close(handle);
    return 0;
}
```

```
}
```

turbo c2.0 字符屏幕函数

Turbo C2.0 的字符屏幕函数主要包括文本窗口大小的设定、窗口颜色的设置、窗口文本的清除和输入输出等函数。

1. 文本窗口的定义

Turbo C2.0 默认定义的文本窗口为整个屏幕，共有 80 列(或 40 列)25 行的文本单元，每个单元包括一个字符和一个属性，字符即 ASCII 码字符，属性规定该字符的颜色和强度。

Turbo C2.0 可以定义屏幕上的一个矩形域作为窗口，使用 `window()` 函数定义。窗口定义之后，用有关窗口的输入输出函数就可以只在此窗口内进行操作而不超出窗口的边界。

`window()` 函数的调用格式为：

```
void window(int left, int top, int right, int bottom);
```

该函数的原型在 `conio.h` 中（关于文本窗口的所有函数其头文件均为 `conio.h`，后面不再说明）。函数中形式参数(`int left, int top`)是窗口左上角的坐标，(`int right, int bottom`)是窗口的右下角坐标，其中(`left, top`)和(`right, bottom`)是相对于整个屏幕而言的。Turbo C 2.0 规定整个屏幕的左上角坐标为(1, 1)，右下角坐标为(80, 25)。并规定沿水平方向为 X 轴，方向朝右；沿垂直方向为 Y 轴，方向朝下。若 `window()` 函数中的坐标超过了屏幕坐标的界限，则窗口的定义就失去了意义，也就是说定义将不起作用，但程序编译链接时并不出错。

另外，一个屏幕可以定义多个窗口，但现行窗口只能有一个(因为 DOS 为单任务操作系统)，当需要用另一窗口时，可将定义该窗口的 `window()` 函数再调用一次，此时该窗口便成为现行窗口了。

如要定义一个窗口左上角在屏幕(20, 5)处，大小为 30 列 15 行的窗口可写成：

```
window(20, 5, 50, 25);
```

2. 文本窗口颜色的设置

文本窗口颜色的设置包括背景颜色的设置和字符颜色的设置，使用的函数及其调用格式为：

设置背景颜色：`void textbackground(int color);`

设置字符颜色：`void textcolor(int color);`

有关颜色的定义见表 1。

表 1. 有关颜色的定义

符号常数	数值	含义	字符或背景
------	----	----	-------

BLACK	0	黑	两者均可
BLUE	1	兰	两者均可
GREEN	2	绿	两者均可
CYAN	3	青	两者均可
RED	4	红	两者均可
MAGENTA	5	洋红	两者均可
BROWN	6	棕	两者均可
LIGHTGRAY	7	淡灰	两者均可
DARKGRAY	8	深灰	只用于字符
LIGHTBLUE	9	淡兰	只用于字符
LIGHTGREEN	10	淡绿	只用于字符
LIGHTCYAN	11	淡青	只用于字符
LIGHTRED	12	淡红	只用于字符
LIGHTMAGENTA	13	淡洋红	只用于字符
YELLOW	14	黄	只用于字符
WHITE	15	白	只用于字符
BLINK	128	闪烁	只用于字符

上表中的符号常数与相应的数值等价,二者可以互换。例如设定兰色背景可以使用 `textbackground(1)`, 也可以使用 `textbackground(BLUE)`, 两者没有任何区别, 只不过后者比较容易记忆, 一看就知道是兰色。

Turbo C 另外还提供了一个函数, 可以同时设置文本的字符和背景颜色, 这个函数的调用格式为:

```
void textattr(int attr);
```

其中: `attr` 的值表示颜色形式编码的信息, 每一位代表的含义如下:

位	7	6	5	4	3	2	1	0
	B	b	b	b	c	c	c	c
	↓	└────────┘			└────────┘			
	闪烁	背景颜色			字符颜色			

字节低四位 `cccc` 设置字符颜色(0 到 15), 4~6 三位 `bbb` 设置背景颜色(0 到 7), 第 7 位 `B` 设置字符是否闪烁。假如要设置一个兰底黄字, 定义方法如下:

```
textattr(YELLOW+(BLUE<<4));
```

若再要求字符闪烁, 则定义变为:

```
textattr(128+YELLOW+(BLUE<<4));
```

注意:

(1) 对于背景只有 0 到 7 共八种颜色, 若取大于 7 小于 15 的数, 则代表的颜色与减 7 后的值对应的颜色相同。

(2) 用 `textbackground()` 和 `textcolor()` 函数设置了窗口的背景与字符颜色后, 在没有用 `clrscr()` 函数清除窗口之前, 颜色不会改变, 直到使用了函数 `clrscr()`, 整个窗口和随后输出到窗口中的文本字符才会变成新颜色。

(3) 用 `textattr()` 函数时背景颜色应左移 4 位, 才能使 3 位背景颜色移到正确位置。

下面这个程序使用了关于窗口大小的定义、颜色的设置等函数, 在一个屏幕

上不同位置定义了 7 个窗口，其背景色分别使用了 7 种不同的颜色。

例 1.:

```
#include <stdio.h>
#include <conio.h>
main()
{
    int i;
    textbackground(0);          /* 设置屏幕背景色 */
    clrscr();                   /* 清除文本屏幕 */
    for(i=1; i<8; i++)
    {
        window(10+i*5, 5+i, 30+i*5, 15+i); /* 定义文本窗口 */
        textbackground(i);        /* 定义窗口背景色 */
        clrscr();                 /* 清除窗口 */
    }
    getch();
}
```

3. 窗口内文本的输入输出函数

一、窗口内文本的输出函数

```
int cprintf("<格式化字符串>", <变量表>);
int cputs(char *string);
int putchar(int ch);
```

cprintf()函数输出一个格式化的字符串或数值到窗口中。它与 **printf()** 函数的用法完全一样，区别在于 **cprintf()**函数的输出受窗口限制，而 **printf()** 函数的输出为整个屏幕。

cputs()函数输出一个字符串到屏幕上，它与 **puts()**函数用法完全一样，只是受窗口大小的限制。

putchar()函数输出一个字符到窗口内。

注意:

(1) 使用以上几种函数，当输出超出窗口的右边界时会自动转到下一行的开始处继续输出。当窗口内填满内容仍没有结束输出时，窗口屏幕将会自动逐行上卷直到输出结束为止。

二、窗口内文本的输入函数

```
int getche(void);
```

该函数在前面已经讲过，需要说明的是，**getche()**函数从键盘上获得一个字符，在屏幕上显示的时候，如果字符超过了窗口右边界，则会被自动转移到下一行的开始位置。

下面这个程序给例 1.中加入了一些文本的输出函数。

例 2.:

```
#include <stdio.h>
#include <conio.h>
```



```

int main()
{
    int i;
    char *c[]={"BLACK", "BLUE", "GREEN", "CYAN", "RED",
               "MAGENTA", "BROWN", "LIGHTGRAY"};
    textbackground(0);          /* 设置屏幕背景色 */
    clrscr();                   /* 清除文本屏幕 */
    for(i=1; i<8; i++)
    {
        window(10+i*5, 5+i, 30+i*5, 15+i); /* 定义文本窗口 */
        textbackground(i);          /* 定义窗口背景色 */
        clrscr();                   /* 清除窗口 */
    }
    getch();
    return 0;
}

```

4. 有关屏幕操作的函数

void clrscr(void); 清除当前窗口中的文本内容，并把光标定位在窗口的左上角(1, 1)处。

void clreol(void); 清除当前窗口中从光标位置到行尾的所有字符，光标位置不变。

void gotoxy(x, y); 该函数很有用，它用来定位光标在当前窗口中的位置。这里 x, y 是指光标要定位处的坐标(相对于窗口而言)，当 x, y 超出了窗口的大小时，该函数就不起作用了。

```

int gettext(int x1, int y1, int x2, int y2, void *buffer);
int puttext(int x1, int y1, int x2, int y2, void *buffer);

```

gettext()函数是将屏幕上指定的矩形区域内文本内容存入 **buffer** 指针指向的一个内存空间。内存的大小用下式计算：

所用字节大小=行数*列数*2

其中：

行数=y2-y1+1 列数=x2-x1+1

puttext()函数则是将 **gettext()**函数存入内存 **buffer** 中的文字内容拷贝到屏幕上指定的位置。

```

int movetext(int x1, int x2, int y2, int x3, int y3);

```

movetext()函数将屏幕上左上角为(x1, y1)，右下角为(x2, y2)的一矩形窗口内的文本内容拷贝到左上角为(x3, y3)的新的位置。该函数的坐标也是相对于整个屏幕而言的。

注意：

1. **gettext()**函数和 **puttext()**函数中的坐标是对整个屏幕而言的，即是屏幕的绝对坐标，而不是相对窗口的坐标。

2. `movetext()`函数是拷贝而不是移动窗口区域内容，即使用该函数后，原位置区域的文本内容仍然存在。

例 3.

```
include <conio.h>
main()
{
    int i;
    char *f[]={ "Load    F3", "Pick  Alt-F3", "New    ",
                "Save    F2", "Write to  ", "Directory",
                "Change dir", "Os shee   ", "Quit Alt-X"};
    char buf[9*14*2];
    clrscr();
    textcolor(YELLOW);
    textbackground(BLUE);
    clrscr();
    gettext(10, 2, 24, 11, buf);
    window(10, 2, 24, 11);
    textbackground(RED);
    textcolor(YELLOW);
    clrscr();
    for(i=0; i<9; i++)
    {
        gotoxy(1, i++);
        cprintf("%s", f[i]);
    }
    getch();
    movetext(10, 2, 24, 11, 40, 10);
    puttext(10, 2, 24, 11, buf);
    getch();
}
```

下面再介绍一些函数:

<code>void highvideo(void);</code>	设置显示器高亮度显示字符。
<code>void lowvideo(void);</code>	设置显示器低亮度显示字符。
<code>void normvideo(void);</code>	使显示器返回到程序运行前的显示方式。
<code>int wherex(void);</code>	这两个函数返回当前窗口下光标的 x, y 坐标。
<code>int wherey(void);</code>	

turbo c2.0 图形处理

Turbo C 提供了非常丰富的图形函数, 所有图形函数的原型均在 `graphics.h` 中, 本节主要介绍图形模式的初始化、独立图形程序的建立、基本图形功能、图形窗口以及图形模式下的文本输出等函数。另外, 使用图形函数时要确保有显示器图形驱动程序*BGI, 同时将集成开发环境 Options/Linker 中的 Graphics lib 选为 on, 只有这样才能保证正确使用图形函数。

1. 图形模式的初始化

不同的显示器适配器有不同的图形分辨率。即是同一显示器适配器, 在不同模式下也有不同分辨率。因此, 在屏幕作图之前, 必须根据显示器适配器种类将显示器设置成为某种图形模式, 在未设置图形模式之前, 微机系统默认屏幕为文本模式(80 列, 25 行字符模式), 此时所有图形函数均不能工作。设置屏幕为图形模式, 可用下列图形初始化函数:

```
void far initgraph(int far *gdriver, int far *gmode, char *path);
```

其中 `gdriver` 和 `gmode` 分别表示图形驱动器和模式, `path` 是指图形驱动程序所在的目录路径。有关图形驱动器、图形模式的符号常数及对应的分辨率见表 2。

图形驱动程序由 Turbo C 出版商提供, 文件扩展名为 .BGI。根据不同的图形适配器有不同的图形驱动程序。例如对于 EGA、VGA 图形适配器就调用驱动程序 EGAVGA.BGI。

表 2. 图形驱动器、模式的符号常数及数值

图形驱动器(gdriver)		图形模式(gmode)		色调	分辨率
符号常数	数值	符号常数	数值		
CGA	1	CGAC0	0	C0	320*200
		CGAC1	1	C1	320*200
		CGAC2	2	C2	320*200
		CGAC3	3	C3	320*200
		CGAHI	4	2 色	640*200
MCGA	2	MCGAC0	0	C0	320*200
		MCGAC1	1	C1	320*200
		MCGAC2	2	C2	320*200
		MCGAC3	3	C3	320*200
		MCGAMED	4	2 色	640*200
		MCGAHI	5	2 色	640*480
EGA	3	EGALO	0	16 色	640*200
		EGAHI	1	16 色	640*350

EGA64	4	EGA64LO EGA64HI	0 1	16 色 640*200 4 色 640*350
EGAMON	5	EGAMONHI	0	2 色 640*350
IBM8514	6	IBM8514LO IBM8514HI	0 1	256 色 640*480 256 色 1024*768
HERC	7	HERCMONHI	0	2 色 720*348
ATT400	8	ATT400C0 ATT400C1 ATT400C2 ATT400C3 ATT400MED ATT400HI	0 1 2 3 4 5	C0 320*200 C1 320*200 C2 320*200 C3 320*200 2 色 320*200 2 色 320*200
VGA	9	VGALO VGAMED VGAHI	0 1 2	16 色 640*200 16 色 640*350 16 色 640*480
PC3270	10	PC3270HI	0	2 色 720*350
DETECT	0	用于硬件测试		

例 4. 使用图形初始化函数设置 VGA 高分辨率图形模式

```
#include <graphics.h>
int main()
{
    int gdriver, gmode;
    gdriver=VGA;
    gmode=VGAHI;
    initgraph(&gdriver, &gmode, "c:\\tc");
    bar3d(100, 100, 300, 250, 50, 1);    /*画一长方体*/
    getch();
    closegraph();
    return 0;
}
```

有时编程者并不知道所用的图形显示器适配器种类, 或者需要将编写的程序用于不同图形驱动器, Turbo C 提供了一个自动检测显示器硬件的函数, 其调用格式为:

```
void far detectgraph(int *gdriver, *gmode);
```

其中 gdriver 和 gmode 的意义与上面相同。

例 5. 自动进行硬件测试后进行图形初始化

```
#include <graphics.h>
int main()
{
    int gdriver, gmode;
    detectgraph(&gdriver, &gmode);    /*自动测试硬件*/
    printf("the graphics driver is %d, mode is %d\n", gdriver,
           gmode);    /*输出测试结果*/
    getch();
    initgraph(&gdriver, &gmode, "c:\\tc");
           /* 根据测试结果初始化图形*/
    bar3d(10, 10, 130, 250, 20, 1);
    getch();
    closegraph();
    return 0;
}
```

上例程序中先对图形显示器自动检测, 然后再用图形初始化函数进行初始化设置, 但 Turbo C 提供了一种更简单的方法, 即用 gdriver= DETECT 语句后再跟 initgraph()函数就行了。采用这种方法后, 上例可改为:

例 6.

```
#include <graphics.h>
int main()
{
    int gdriver=DETECT, gmode;
    initgraph(&gdriver, &gmode, "c:\\tc");
    bar3d(50, 50, 150, 30, 1);
    getch();
    closegraph();
    return 0;
}
```

另外, Turbo C 提供了退出图形状态的函数 closegraph(), 其调用格式为:

```
void far closegraph(void);
```

调用该函数后可退出图形状态而进入文本方式(Turbo C 默认方式), 并释放用于保存图形驱动程序和字体的系统内存。

2. 独立图形运行程序的建立

Turbo C 对于用 initgraph()函数直接进行的图形初始化程序, 在编译和链接时并没有将相应的驱动程序(*.BGI)装入到执行程序, 当程序进行到 intitgraph()语句时, 再从该函数中第三个形式参数 char *path 中所规定的路径中去找相应的

驱动程序。若没有驱动程序,则在 C:\TC 中去找,如 C:\TC 中仍没有或 TC 不存在,将会出现错误:

BGI Error: Graphics not initialized (use 'initgraph')

因此,为了使用方便,应该建立一个不需要驱动程序就能独立运行的可执行图形程序,Turbo C 中规定用下述步骤(这里以 EGA、VGA 显示器为例):

- 1. 在 C:\TC 子目录下输入命令:BGI\OBJ EGAVGA
此命令将驱动程序 EGAVGA.BGI 转换成 EGAVGA.OBJ 的目标文件。
- 2. 在 C:\TC 子目录下输入命令:TLIB LIB\GRAPHICS.LIB+EGAVGA
此命令的意思是将 EGAVGA.OBJ 的目标模块装到 GRAPHICS.LIB 库文件中。
- 3. 在程序中 initgraph()函数调用之前加上一句:
registerbgidriver(EGAVGA_driver);

该函数告诉连接程序在连接时把 EGAVGA 的驱动程序装入到用户的执行程序中。
经过上面处理,编译链接后的执行程序可在任何目录或其它兼容机上运行。
假设已作了前两个步骤,若再向例 6 中加 registerbgidriver()函数则变成:

```
例 7:
#include<stdio.h>
#include<graphics.h>
int main()
{
    int gdriver=DETECT,gmode;
    registerbgidriver(EGAVGA_driver); /*建立独立图形运行程序 */
    initgraph( gdriver, gmode,"c:\\tc");
    bar3d(50,50,250,150,20,1);
    getch();
    closegraph();
    return 0;
}
```

上例编译链接后产生的执行程序可独立运行。
如不初始化成 EGA 或 CGA 分辨率,而想初始化为 CGA 分辨率, 则只需要将上述步骤中有 EGAVGA 的地方用 CGA 代替即可。

3.屏幕颜色的设置和清屏函数

对于图形模式的屏幕颜色设置,同样分为背景色的设置和前景色的设置。在 Turbo C 中分别用下面两个函数。

- 设置背景色: void far setbkcolor(int color);
- 设置作图色: void far setcolor(int color);

其中 color 为图形方式下颜色的规定数值,对 EGA, VGA 显示器适配器,有关颜色的符号常数及数值见下表所示。

表 3 有关屏幕颜色的符号常数表

符号常数	数值	含义	符号常数	数值	含义
BLACK	0	黑色	DARKGRAY	8	深灰
BLUE	1	兰色	LIGHTBLUE	9	深兰

GREEN	2	绿色	LIGHTGREEN	10	淡绿
CYAN	3	青色	LIGHTCYAN	11	淡青
RED	4	红色	LIGHTRED	12	淡红
MAGENTA	5	洋红	LIGHTMAGENTA	13	淡洋红
BROWN	6	棕色	YELLOW	14	黄色
LIGHTGRAY	7	淡灰	WHITE	15	白色

对于 CGA 适配器, 背景色可以为表 3 中 16 种颜色的一种, 但前景色依赖于不同的调色板。共有四种调色板, 每种调色板上有四种颜色可供选择。不同调色板所对应的原色见表 4。

表 4 CGA 调色板与颜色值表

调色板		颜色值			
符号常数	数值	0	1	2	3
C0	0	背景	绿	红	黄
C1	1	背景	青	洋红	白
C2	2	背景	淡绿	淡红	黄
C3	3	背景	淡青	淡洋红	白

清除图形屏幕内容使用清屏函数, 其调用格式如下:

```
void far cleardevice(void);
```

有关颜色设置、清屏函数的使用请看例 8。

例 8:

```
#include<stdio.h>
#include<graphics.h>
int main()
{
    int gdriver, gmode, i;
    gdriver=DETECT;
    registerbgidriver(EGAVGA_DRIVER);/*建立独立图形运行程序*/
    initgraph(&gdriver, &gmode, "");/*图形初始化*/
    setbkcolor(0);                /*设置图形背景*/
    cleardevice();
    for(i=0; i<=15; i++)
    {
        setcolor(i);              /*设置不同作图色*/
        circle(320, 240, 20+i*10); /*画半径不同的圆*/
        delay(100);               /*延迟 100 毫秒*/
    }
    for(i=0; i<=15; i++)
    {
```

```

        setbkcolor(i);           /*设置不同背景色*/
        cleardevice();
        circle(320, 240, 20+i*10);
        delay(100);
    }
    closegraph();
    return 0;
}

```

另外, TURBO C 也提供了几个获得现行颜色设置情况的函数。

```

int far getbkcolor(void);      返回现行背景颜色值。
int far getcolor(void);       返回现行作图颜色值。
int far getmaxcolor(void);    返回最高可用的颜色值。

```

4. 基本图形函数

基本图形函数包括画点、线以及其它一些基本图形的函数。本节对这些函数作一全面的介绍。

一、画点

1. 画点函数

```
void far putpixel(int x, int y, int color);
```

该函数表示有指定的象元画一个按 **color** 所确定颜色的点。对于颜色 **color** 的值可从表 3 中获得而对 **x, y** 是指图形象元的坐标。

在图形模式下, 是按象元来定义坐标的。对 VGA 适配器, 它的最高分辨率为 640x480, 其中 640 为整个屏幕从左到右所有象元的个数, 480 为整个屏幕从上到下所有象元的个数。屏幕的左上角坐标为(0, 0), 右下角坐标为(639, 479), 水平方向从左到右为 x 轴正向, 垂直方向从上到下为 y 轴正向。TURBO C 的图形函数都是相对于图形屏幕坐标, 即象元来说的。

关于点的另外一个函数是:

```
int far getpixel(int x, int y);
```

它获得当前点(x, y)的颜色值。

2. 有关坐标位置的函数

```
int far getmaxx(void);
```

返回 x 轴的最大值。

```
int far getmaxy(void);
```

返回 y 轴的最大值。

```
int far getx(void);
```

返回游标在 x 轴的位置。

```
void far gety(void);
```


返回游标有 y 轴的位置。

```
void far moveto(int x, int y);
```

移动游标到(x, y)点, 不是画点, 在移动过程中亦画点。

```
void far moverel(int dx, int dy);
```

移动游标从现行位置(x, y)移动到(x+dx, y+dy)的位置, 移动过程中不画点。

二、画线

1. 画线函数

TURBO C 提供了一系列画线函数, 下面分别叙述:

```
void far line(int x0, int y0, int x1, int y1);
```

画一条从点(x0, y0)到(x1, y1)的直线。

```
void far lineto(int x, int y);
```

画一作从现行游标到点(x, y)的直线。

```
void far linerel(int dx, int dy);
```

画一条从现行游标(x, y)到按相对增量确定的点(x+dx, y+dy)的直线。

```
void far circle(int x, int y, int radius);
```

以(x, y)为圆心, radius 为半径, 画一个圆。

```
void far arc(int x, int y, int stangle, int endangle, int radius);
```

以(x, y)为圆心, radius 为半径, 从 stangle 开始到 endangle 结束(用度表示)

画一段圆弧线。在 TURBO C 中规定 x 轴正向为 0 度, 逆时针方向旋转一周, 依次为 90, 180, 270 和 360 度(其它有关函数也按此规定, 不再重述)。

```
void ellipse(int x, int y, int stangle, int endangle, int xradius,
int yradius);
```

以(x, y)为中心, xradius, yradius 为 x 轴和 y 轴半径, 从角 stangle 开始到 endangle 结束画一段椭圆线, 当 stangle=0, endangle=360 时, 画出一个完整的椭圆。

```
void far rectangle(int x1, int y1, int x2, int y2);
```

以(x1, y1)为左上角, (x2, y2)为右下角画一个矩形框。

```
void far drawpoly(int numpoints, int far *polypoints);
```

画一个顶点数为 numpoints, 各顶点坐标由 polypoints 给出的多边形。polypoints 整型数组必须至少有 2 倍顶点数个元素。每一个顶点的坐标都定义为 x, y, 并且 x 在前。值得注意的是当画一个封闭的多边形时, numpoints 的值取实际多边形的顶点数加一, 并且数组 polypoints 中第一个和最后一个点的坐标相同。

下面举一个用 drawpoly() 函数画箭头的例子。

例 9:

```
#include<stdlib.h>
#include<graphics.h>
int main()
{
    int gdriver, gmode, i;
    int arw[16]={200, 102, 300, 102, 300, 107, 330,
                100, 300, 93, 300, 98, 200, 98, 200, 102};
    gdriver=DETECT;
    registerbgidriver(EGAVGA_driver);
    initgraph(&gdriver, &gmode, "");
    setbkcolor(BLUE);
    cleardevice();
    setcolor(12);      /*设置作图颜色*/
    drawpoly(8, arw);  /*画一箭头*/
    getch();
    closegraph();
    return 0;
}
```

2. 设定线型函数

在没有对线的特性进行设定之前, TURBO C 用其默认值, 即一点宽的实线, 但 TURBO C 也提供了可以改变线型的函数。线型包括:宽度和形状。其中宽度只有两种选择: 一点宽和三点宽。而线的形状则有五种。下面介绍有关线型的设置函数。

```
void far setlinestyle(int linestyle, unsigned upattern, int
thickness);
```

该函数用来设置线的有关信息, 其中 `linestyle` 是线形状的规定, 见表 5。

表 5. 有关线的形状(linestyle)

符号常数	数值	含义
SOLID_LINE	0	实线
DOTTED_LINE	1	点线
CENTER_LINE	2	中心线
DASHED_LINE	3	点画线
USERBIT_LINE	4	用户定义线

`thickness` 是线的宽度, 见表 6。

表 6. 有关线宽(thickness)

符号常数	数值	含义
------	----	----

NORM_WIDTH	1	一点宽
THIC_WIDTH	3	三点宽

对于 upattern, 只有 linestyle 选 USERBIT_LINE 时才有意义(选其它线型, upattern 取 0 即可)。此进 upattern 的 16 位二进制数的每一位代表一个象元, 如果那位为 1, 则该象元打开, 否则该象元关闭。

```
void far getlinesettings(struct linesettingstype far *lineinfo);
```

该函数将有关线的信息存放由 lineinfo 指向的结构中, 表中 linesettingstype 的结构如下:

```
struct linesettingstype{
    int linestyle;
    unsigned upattern;
    int thickness;
}
```

例如下面两句程序可以读出当前线的特性

```
struct linesettingstype *info;
getlinesettings(info);
```

```
void far setwritemode(int mode);
```

该函数规定画线的方式。如果 mode=0, 则表示画线时将所画位置的原来信息覆盖了(这是 TURBO C 的默认方式)。如果 mode=1, 则表示画线时用现在特性的线与所画之处原有的线进行异或(XOR)操作, 实际上画出的线是原有线与现在规定的线进行异或后的结果。因此, 当线的特性不变, 进行两次画线操作相当于没有画线。

有关线型设定和画线函数的例子如下所示。

例 10.

```
#include<stdlib.h>
#include<graphics.h>
int main()
{
    int gdriver, gmode, i;
    gdriver=DETECT;
    registerbgidriver(EGAVGA_driver);
    initgraph(&gdriver, &gmode, "");
    setbkcolor(BLUE);
    cleardevice();
    setcolor(GREEN);
    circle(320, 240, 98);
    setlinestyle(0, 0, 3); /*设置三点宽实线*/
    setcolor(2);
    rectangle(220, 140, 420, 340);
    setcolor(WHITE);
```

```

        setlinestyle(4, 0xaaaa, 1);    /*设置一点宽用户定义线*/
        line(220, 240, 420, 240);
        line(320, 140, 320, 340);
        getch();
        closegraph();
        return 0;
    }

```

5. 封闭图形的填充

填充就是用规定的颜色和图模填满一个封闭图形。

一、先画轮廓再填充

TURBO C 提供了一些先画出基本图形轮廓，再按规定图模和颜色填充整个封闭图形的函数。在没有改变填充方式时，TURBO C 以默认方式填充。下面介绍这些函数。

```
void far bar(int x1, int y1, int x2, int y2);
```

确定一个以(x1, y1)为左上角, (x2, y2)为右下角的矩形窗口, 再按规定图模和颜色填充。

说明: 此函数不画出边框, 所以填充色为边框。

```
void far bar3d(int x1, int y1, int x2, int y2, int depth, int
topflag);
```

当 topflag 为非 0 时, 画出一个三维的长方体。当 topflag 为 0 时, 三维图形不封顶, 实际上很少这样使用。

说明: bar3d() 函数中, 长方体第三维的方向不随任何参数而变, 即始终为 45 度的方向。

```
void far pieslice(int x, int y, int stangle, int endangle, int
radius);
```

画一个以(x, y)为圆心, radius 为半径, stangle 为起始角度, endangle 为终止角度的扇形, 再按规定方式填充。当 stangle=0, endangle=360 时变成一个实心圆, 并在圆内从圆点沿 X 轴正向画一条半径。

```
void far sector(int x, int y, int stanle, int endangle, int
xradius, int yradius);
```

画一个以(x, y)为圆心分别以 xradius, yradius 为 x 轴和 y 轴半径, stangle 为起始角, endangle 为终止角的椭圆扇形, 再按规定方式填充。

二、设定填充方式

TURBO C 有四个与填充方式有关的函数。下面分别介绍:

```
void far setfillstyle(int pattern, int color);
```

color 的值是当前屏幕图形模式时颜色的有效值。pattern 的值及与其等价的

符号常数如表 7 所示。

表 7. 关于填充式样 pattern 的规定

符号常数	数值	含义
EMPTY_FILL	0	以背景颜色填充
SOLID_FILL	1	以实填充
LINE_FILL	2	以直线填充
LTSLASH_FILL	3	以斜线填充(阴影线)
SLASH_FILL	4	以粗斜线填充(粗阴影线)
BKSLASH_FILL	5	以粗反斜线填充(粗阴影线)
LTBKSLASH_FILL	6	以反斜线填充(阴影线)
HATCH_FILL	7	以直方网格填充
XHATCH_FILL	8	以斜网格填充
INTTERLEAVE_FILL	9	以间隔点填充
WIDE_DOT_FILL	10	以稀疏点填充
CLOSE_DOS_FILL	11	以密集点填充
USER_FILL	12	以用户定义式样填充

除 USER_FILL(用户定义填充式样)以外, 其它填充式样均可由 `setfillstyle()` 函数设置。当选用 USER_FILL 时, 该函数对填充图模和颜色不作任何改变。之所以定义 USER_FILL 主要因为在获得有关填充信息时用到此项。

```
void far setfillpattern(char * upattern,int color);
```

设置用户定义的填充图模的颜色以供对封闭图形填充。

其中 `upattern` 是一个指向 8 个字节的指针。这 8 个字节定义了 8x8 点阵的图形。每个字节的 8 位二进制数表示水平 8 点, 8 个字节表示 8 行, 然后以此为模型向个封闭区域填充。

```
void far getfillpattern(char * upattern);
```

该函数将用户定义的填充图模存入 `upattern` 指针指向的内存区域。

```
void far getfillsetings(struct fillsettingstype far * fillinfo);
```

获得现行图模的颜色并将存入结构指针变量 `fillinfo` 中。其中 `fillsettingstype` 结构定义如下:

```
struct fillsettingstype{
    int pattern; /* 现行填充模式 */
    int color;   /* 现行填充模式 */
};
```

有关图形填充图模的颜色的选择, 请看下面例程。

例 11:

```
#include<graphics.h>
main(){
    char str[8]={10,20,30,40,50,60,70,80}; /*用户定义图模*/
    int gdriver,gmode,i;
    struct fillsettingstype save; /*定义一个用来存储填充信息的结构变量*/
```

```

gdriver=DETECT;
initgraph(&gdriver,&gmode,"c:\\tc");
setbkcolor(BLUE);
cleardevice();
for(i=0;i<13;i++)
{
    setcolor(i+3);
    setfillstyle(i,2+i);      /* 设置填充类型 */
    bar(100,150,200,50);      /*画矩形并填充*/
    bar3d(300,100,500,200,70,1); /* 画长方体并填充*/
    pieslice(200,300,90,180,90);/*画扇形并填充*/
    sector(500,300,180,270,200,100);/*画椭圆扇形并填充*/
    delay(1000);              /*延时 1 秒*/
}
cleardevice();
setcolor(14);
setfillpattern(str, RED);
bar(100,150,200,50);
bar3d(300,100,500,200,70,0);
pieslice(200,300,0,360,90);
sector(500,300,0,360,100,50);
getch();
getfillsettings(&save);      /*获得用户定义的填充模式信息*/
closegraph();
clrscr();
printf("The pattern is %d, The color of filling is %d",
       save.pattern, save.color); /*输出目前填充图模和颜色值*/
getch();
}

```

以上程序运行结束后，在屏幕上显示出当前填充图模和颜色的常数值。

三、任意封闭图形的填充

截止目前为止，我们只能对一些特定形状的封闭图形进行填充，但还不能对任意封闭图形进行填充。为此，TURBO C 提供了一个可对任意封闭图形填充的函数，其调用格式如下：

```
void far floodfill(int x, int y, int border);
```

其中：x, y 为封闭图形内的任意一点。border 为边界的颜色，也就是封闭图形轮廓的颜色。调用了该函数后，将用规定的颜色和图模填满整个封闭图形。

注意：

1. 如果 x 或 y 取在边界上，则不进行填充。
2. 如果不是封闭图形则填充会从没有封闭的地方溢出去，填满其它地方。
3. 如果 x 或 y 在图形外面，则填充封闭图形外的屏幕区域。
4. 由 border 指定的颜色值必须与图形轮廓的颜色值相同，但填充色可选任意颜色。下例是有关 floodfill() 函数的用法，该程序填充了 bar3d() 所画长方体

中其它两个未填充的面。

例 12:

```
#include<stdlib.h>
#include<graphics.h>
main()
{
    int gdriver, gmode;
    struct fillsettingstype save;
    gdriver=DETECT;
    initgraph(&gdriver, &gmode, "");
    setbkcolor(BLUE);
    cleardevice();
    setcolor(LIGHTRED);
    setlinestyle(0,0,3);
    setfillstyle(1,14);          /*设置填充方式*/
    bar3d(100,200,400,350,200,1); /*画长方体并填充*/
    floodfill(450,300,LIGHTRED); /*填充长方体另外两个面*/
    floodfill(250,150, LIGHTRED);
    rectanle(450,400,500,450);   /*画一矩形*/
    floodfill(470,420, LIGHTRED); /*填充矩形*/
    getch();
    closegraph();
}
```

6. 有关图形窗口和图形屏幕操作函数

一、图形窗口操作

象文本方式下可以设定屏幕窗口一样，图形方式下也可以在屏幕上某一区域设定窗口，只是设定的为图形窗口而已，其后的有关图形操作都将以这个窗口的左上角(0,0)作为坐标原点，而且可为通过设置使窗口之外的区域为不可接触。这样，所有的图形操作就被限定在窗口内进行。

```
void far setviewport(int x1,int y1,int x2, int y2,int clipflag);
```

设定一个以(x1,y1)象元点为左上角, (x2,y2)象元为右下角的图形窗口，其中 x1,y1,x2,y2 是相对于整个屏幕的坐标。若 clipflag 为非 0，则设定的图形以外部分不可接触，若 clipflag 为 0，则图形窗口以外可以接触。

```
void far clearviewport(void);
```

清除现行图形窗口的内容。

```
void far getviewsettings(struct viewporttype far * viewport);
```

获得关于现行窗口的信息,并将其存于 viewporttype 定义的结构变量 viewport 中，其中 viewporttype 的结构说明如下:

```
struct viewporttype{
    int left, top, right, bottom;
    int cliplag;
};
```

注明:

1. 窗口颜色的设置与前面讲过的屏幕颜色设置相同, 但屏幕背景色和窗口背景色只能是一种颜色, 如果窗口背景色改变, 整个屏幕的背景色也将改变这与文本窗口不同。

2. 可以在同一个屏幕上设置多个窗口, 但只能有一个现行窗口工作, 要对其它窗口操作, 通过将定义那个窗口的 `setviewport()` 函数再用一次即可。

3. 前面讲过图形屏幕操作的函数均适合于对窗口的操作。

二、屏幕操作

除了清屏函数以外, 关于屏幕操作还有以下函数:

```
void far setactivepage(int pagenum);
```

```
void far setvisualpage(int pagenum);
```

这两个函数只用于 EGA,VGA 以及 HERCULES 图形适配器。`setactivepage()` 函数是为图形输出选择激活页。所谓激活页是指后续图形的输出被写到函数选定的 `pagenum` 页面, 该页面并不一定可见。`setvisualpage()` 函数才使 `pagenum` 所指定的页面变成可见页。页面从 0 开始(Turbo C 默认页)。如果先用 `setactivepage()` 函数在不同页面上画出一幅幅图像,再用 `setvisualpage()` 函数交替显示, 就可以实现一些动画的效果。

```
void far getimage(int x1,int y1, int x2,int y2, void far *mapbuf);
```

```
void far putimage(int x,int,y,void * mapbuf, int op);
```

```
unsigned far imagesize(int x1,int y1,int x2,int y2);
```

这三个函数用于将屏幕上的图像复制到内存,然后再将内存中的图像送回到屏幕上。首先通过函数 `imagesize()` 测试要保存左上角为(x1,y1), 右上角为(x2, y2)的图形屏幕区域内的全部内容需多少个字节, 然后再给 `mapbuf` 分配一个所测数字字节内存空间的指针。通过调用 `getimage()` 函数就可将该区域内的图像保存在内存中, 需要时可用 `putimage()` 函数将该图像输出到左上角为点(x, y)的位置上, 其中 `getimage()` 函数中的参数 `op` 规定如何释放内存中图像。

关于这个参数的定义参见表 8。

表 8. `putimage()` 函数中的 `op` 值

符号常数	数值	含 义
<code>COPY_PUT</code>	0	复制
<code>XOR_PUT</code>	1	与屏幕图像异或的复制
<code>OR_PUT</code>	2	与屏幕图像或后复制
<code>AND_PUT</code>	3	与屏幕图像与后复制
<code>NOT_PUT</code>	4	复制反像的图形

对于 `imagesize()` 函数, 只能返回字节数小于 64K 字节的图像区域, 否则将会出错, 出错时返回-1。

本节介绍的函数在图像动画处理、菜单设计技巧中非常有用。

例 13: 下面程序模拟两个小球动态碰撞过程。

```
#include<stdio.h>
#include<graphics.h>
int main()
{
    int i, gdriver, gmode, size;
    void *buf;
    gdriver=DETECT;
    initgraph(&gdriver, &gmode, "");
    setbkcolor(BLUE);
    cleardevice();
    setcolor(LIGHTRED);
    setlinestyle(0,0,1);
    setfillstyle(1, 10);
    circle(100, 200, 30);
    floodfill(100, 200, 12);
    size=imagesize(69, 169, 131, 231);
    buf=malloc(size);
    getimage(69, 169, 131, 231,buf);
    putimage(500, 269, buf, COPY_PUT);
    for(i=0; i<185; i++){
        putimage(70+i, 170, buf, COPY_PUT);
        putimage(500-i, 170, buf, COPY_PUT);
    }
    for(i=0;i<185; i++){
        putimage(255-i, 170, buf, COPY_PUT);
        putimage(315+i, 170, buf, COPY_PUT);
    }
    getch();
    closegraph();
}
```

7. 图形模式下的文本输出

在图形模式下, 只能用标准输出函数, 如 `printf()`, `puts()`, `putchar()` 函数输出文本到屏幕。除此之外, 其它输出函数(如窗口输出函数)不能使用, 即是可以输出的标准函数, 也只以前景色为白色, 按 80 列, 25 行的文本方式输出。

Turbo C2.0 也提供了一些专门用于在图形显示模式下的文本输出函数。下面将分别进行介绍。

一、文本输出函数

```
void far outtext(char far *textstring);
```

该函数输出字符串指针 `textstring` 所指的文本在现行位置。

```
void far outtextxy(int x, int y, char far *textstring);
```

该函数输出字符串指针 `textstring` 所指的文本在规定的(x, y)位置。其中 x

和 y 为象元坐标。

说明:

这两个函数都是输出字符串, 但经常会遇到输出数值或其它类型的数据, 此时就必须使用格式化输出函数 `sprintf()`。

`sprintf()`函数的调用格式为:

```
int sprintf(char *str, char *format, variable-list);
```

它与 `printf()`函数不同之处是将按格式化规定的内容写入 `str` 指向的字符串中, 返回值等于写入的字符个数。

例如:

```
sprintf(s, "your TOEFL score is %d", mark);
```

这里 `s` 应是字符串指针或数组, `mark` 为整型变量。

二、有关文本字体、字型 and 输出方式的设置

有关图形方式下的文本输出函数, 可以通过 `setcolor()`函数设置输出文本的颜色。另外, 也可以改变文本字体大小以及选择是水平方向输出还是垂直方向输出。

```
void far settexjustify(int horiz, int vert);
```

该函数用于定位输出字符串。

对使用 `outtextxy(int x, int y, char far *str textstring)` 函数所输出的字符串, 其中哪个点对应于定位坐标(x, y)在 Turbo C2.0 中是有规定的。如果把一个字符串看成一个长方形的图形, 在水平方向显示时, 字符串长方形按垂直方向可分为顶部、中部和底部三个位置, 水平方向可分为左、中、右三个位置, 两者结合就有 9 个位置。

`settexjustify()`函数的第一个参数 `horiz` 指出水平方向三个位置中的一个, 第二个参数 `vert` 指出垂直方向三个位置中的一个, 二者就确定了其中一个位置。当规定了这个位置后, 用 `outtextxy()`函数输出字符串时, 字符串长方形的这个规定位置就对准函数中的(x, y)位置。而对用 `outtext()`函数输出字符串时, 这个规定的位置就位于现行游标的位置。有关参数 `horiz` 和 `vert` 的取值参见表 9。

表 9. 参数 `horiz` 和 `vert` 的取值

符号常数	数值	用于
LEFT_TEXT	0	水平
RIGHT_TEXT	2	水平
BOTTOM_TEXT	0	垂直
TOP_TEXT	2	垂直
CENTER_TEXT	1	水平或垂直

```
void far settextstyle(int font, int direction, int charsize);
```

该函数用来设置输出字符的字形(由 `font` 确定)、输出方向(由 `direction` 确定)和字符大小(由 `charsize` 确定)等特性。Turbo C2.0 对函数中各个参数的规定见下

列各表所示:

表 10. font 的取值

符号常数	数值	含义
DEFAULT_FONT	0	8*8 点阵字(缺省值)
TRIPLEX_FONT	1	三倍笔划字体
SMALL_FONT	2	小号笔划字体
SANSERIF_FONT	3	无衬线笔划字体
GOTHIC_FONT	4	黑体笔划字

表 11. direction 的取值

符号常数	数值	含义
HORIZ_DIR	0	从左到右
VERT_DIR	1	从底到顶

表 12. charsize 的取值

符号常数或数值	含义
1	8*8 点阵
2	16*16 点阵
3	24*24 点阵
4	32*32 点阵
5	40*40 点阵
6	48*48 点阵
7	56*56 点阵
8	64*64 点阵
9	72*72 点阵
10	80*80 点阵
USER_CHAR_SIZE=0	用户定义的字符大小

有关图形屏幕下文本输出和字体字型设置函数的用法请看下例:

例 14:

```
#include<graphics.h>
#include<stdio.h>
int main()
```

```

{
    int i, gdriver, gmode;
    char s[30];
    gdriver=DETECT;
    initgraph(&gdriver, &gmode, "");
    setbkcolor(BLUE);
    cleardevice();
    setviewport(100, 100, 540, 380, 1); /*定义一个图形窗口*/
    setfillstyle(1, 2);      /*绿色以实填充*/
    setcolor(YELLOW);
    rectangle(0, 0, 439, 279);
    floodfill(50, 50, 14);
    setcolor(12);
    settextstyle(1, 0, 8);    /*三重笔划字体, 水平放大 8 倍*/
    outtextxy(20, 20, "Good Better");
    setcolor(15);
    settextstyle(3, 0, 5);   /*无衬笔划字体, 水平放大 5 倍*/
    outtextxy(120, 120, "Good Better");
    setcolor(14);
    settextstyle(2, 0, 8);
    i=620;
    sprintf(s, "Your score is %d", i); /*将数字转化为字符串*/
    outtextxy(30, 200, s);      /*指定位置输出字符串*/
    setcolor(1);
    settextstyle(4, 0, 3);
    outtextxy(70, 240, s);
    getch();
    closegraph();
    return 0;
}

```

三、用户对文本字符大小的设置

前面介绍的 `settextstyle()` 函数, 可以设定图形方式下输出文本字符这字体和大小但对于笔划型字体(除 8*8 点阵字以个的字体), 只能在水平和垂直方向以相同的放大倍数放大。为此 Turbo C2.0 又提供了另外一个 `setusercharsize()` 函数, 对笔划字体可以分别设置水平和垂直方向的放大倍数。该函数的调用格式为:

```
void far setusercharsize(int mulx, int divx, int muly, int divy);
```

该函数用来设置笔划型字和放大系数, 它只有在 `settextstyle()` 函数中的 `charsize` 为 0(或 `USER_CHAR_SIZE`)时才起作用, 并且字体为函数 `settextstyle()` 规定的字体。调用函数 `setusercharsize()` 后, 每个显示在屏幕上的字符都以其缺省大小乘以 `mulx/divx` 为输出字符宽, 乘以 `muly/divy` 为输出字符高。该函数的用法见下例。

例 15:

```

#include<stdio.h>
#include<graphics.h>
int main()
{
    int gdriver, gmode;
    gdriver=DETECT;
    initgraph(&gdriver, &gmode, "");
    setbkcolor(BLUE);
    cleardevice();
    setfillstyle(1, 2); /*设置填充方式*/
    setcolor(WHITE);    /*设置白色作图*/
    rectangle(100, 100, 330, 380);
    floodfill(50, 50, 14); /*填充方框以外的区域*/
    setcolor(12);        /*作图色为淡红*/
    settextstyle(1, 0, 8);/*三重笔划字体, 放大 8 倍*/
    outtextxy(120, 120, "Very Good");
    setusercharsize(2, 1, 4, 1);/*水平放大 2 倍, 垂直放大 4 倍*/
    setcolor(15);
    settextstyle(3, 0, 5); /*无衬字笔划, 放大 5 倍*/
    outtextxy(220, 220, "Very Good");
    setusercharsize(4, 1, 1, 1);
    settextstyle(3, 0, 0);
    outtextxy(180, 320, "Good");
    getch();
    closegraph();
    return 0;
}

```

C 语言程序 100 例

【程序 1】

题目：有 1、2、3、4 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？

1. 程序分析：可填在百位、十位、个位的数字都是 1、2、3、4。组成所有的排列后再去掉不满足条件的排列。

2. 程序源代码：

```

main()
{
    int i, j, k;
    printf("\n");
    for (i=1; i<5; i++)        /*以下为三重循环*/
        for (j=1; j<5; j++)
            for (k=1; k<5; k++)
                {

```

```

        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);
    }
}

```

【程序 2】

题目：企业发放的奖金根据利润提成。利润(I)低于或等于 10 万元时，奖金可提 10%；利润高于 10 万元，低于 20 万元时，低于 10 万元的部分按 10%提成，高于 10 万元的部分，可提成 7.5%；20 万到 40 万之间时，高于 20 万元的部分，可提成 5%；40 万到 60 万之间时高于 40 万元的部分，可提成 3%；60 万到 100 万之间时，高于 60 万元的部分，可提成 1.5%，高于 100 万元时，超过 100 万元的部分按 1%提成，从键盘输入当月利润 I，求应发放奖金总数？

1. 程序分析：请利用数轴来分界，定位。注意定义时需把奖金定义成长整型。

2. 程序源代码：

```

main()
{
    long int i;
    int bonus1, bonus2, bonus4, bonus6, bonus10, bonus;
    scanf("%ld", &i);
    bonus1=100000*0.1; bonus2=bonus1+100000*0.75;
    bonus4=bonus2+200000*0.5;
    bonus6=bonus4+200000*0.3;
    bonus10=bonus6+400000*0.15;
    if(i<=100000)
        bonus=i*0.1;
    else if(i<=200000)
        bonus=bonus1+(i-100000)*0.075;
    else if(i<=400000)
        bonus=bonus2+(i-200000)*0.05;
    else if(i<=600000)
        bonus=bonus4+(i-400000)*0.03;
    else if(i<=1000000)
        bonus=bonus6+(i-600000)*0.015;
    else
        bonus=bonus10+(i-1000000)*0.01;
    printf("bonus=%d", bonus);
}

```

【程序 3】

题目：一个整数，它加上 100 后是一个完全平方数，再加上 168 又是一个完全平方数，请问该数是多少？

1. 程序分析：在 10 万以内判断，先将该数加上 100 后再开方，再将该数加上 268 后再开方，如果开方后的结果满足如下条件，即是结果。请看具体分析：

2. 程序源代码:

```
#include "math.h"
main()
{
    long int i, x, y, z;
    for (i=1; i<100000; i++)
    {
        x=sqrt(i+100);    /*x 为加上 100 后开方后的结果*/
        y=sqrt(i+268);    /*y 为再加上 168 后开方后的结果*/
        if(x*x==i+100&& y*y==i+268) /*如果一个数的平方根的平方等于该数, 这说明此数是
        完全平方数*/
            printf("\n%ld\n", i);
    }
}
```

【程序 4】

题目: 输入某年某月某日, 判断这一天是这一年的第几天?

1. 程序分析: 以 3 月 5 日为例, 应该先把前两个月的加起来, 然后再加上 5 天即本年的第几天, 特殊情况, 闰年且输入月份大于 3 时需考虑多加一天。

2. 程序源代码:

```
main()
{
    int day, month, year, sum, leap;
    printf("\nplease input year, month, day\n");
    scanf("%d, %d, %d", &year, &month, &day);
    switch(month) /*先计算某月以前月份的总天数*/
    {
        case 1: sum=0; break;
        case 2: sum=31; break;
        case 3: sum=59; break;
        case 4: sum=90; break;
        case 5: sum=120; break;
        case 6: sum=151; break;
        case 7: sum=181; break;
        case 8: sum=212; break;
        case 9: sum=243; break;
        case 10: sum=273; break;
        case 11: sum=304; break;
        case 12: sum=334; break;
        default: printf("data error"); break;
    }
    sum=sum+day;    /*再加上某天的天数*/
    if(year%400==0 || (year%4==0 && year%100!=0)) /*判断是不是闰年*/
        leap=1;
    else
```

```

    leap=0;
if (leap==1&&month>2)/*如果是闰年且月份大于 2, 总天数应该加一天*/
sum++;
printf("It is the %dth day.",sum);}
=====

```

【程序 5】

题目：输入三个整数 x, y, z，请把这三个数由小到大输出。

1. 程序分析：我们想办法把最小的数放到 x 上，先将 x 与 y 进行比较，如果 x>y 则将 x 与 y 的值进行交换，然后再用 x 与 z 进行比较，如果 x>z 则将 x 与 z 的值进行交换，这样能使 x 最小。

2. 程序源代码：

```

main()
{
int x, y, z, t;
scanf("%d%d%d", &x, &y, &z);
if (x>y)
{t=x;x=y;y=t;} /*交换 x, y 的值*/
if(x>z)
{t=z;z=x;x=t;} /*交换 x, z 的值*/
if(y>z)
{t=y;y=z;z=t;} /*交换 z, y 的值*/
printf("small to big: %d %d %d\n", x, y, z);
}
=====

```

【程序 6】

题目：用*号输出字母 C 的图案。

1. 程序分析：可先用 '*' 号在纸上写出字母 C，再分行输出。

2. 程序源代码：

```

#include "stdio.h"
main()
{
printf("Hello C-world!\n");
printf(" ****\n");
printf(" *\n");
printf(" * \n");
printf(" ****\n");
}
=====

```

【程序 7】

题目：输出特殊图案，请在 c 环境中运行，看一看，Very Beautiful!

1. 程序分析：字符共有 256 个。不同字符，图形不一样。

2. 程序源代码：

```

#include "stdio.h"
main()

```



```
{
char a=176,b=219;
printf("%c%c%c%c%c\n",b,a,a,a,b);
printf("%c%c%c%c%c\n",a,b,a,b,a);
printf("%c%c%c%c%c\n",a,a,b,a,a);
printf("%c%c%c%c%c\n",a,b,a,b,a);
printf("%c%c%c%c%c\n",b,a,a,a,b);}
```

【程序 8】

题目：输出 9*9 口诀。

1. 程序分析：分行与列考虑，共 9 行 9 列，i 控制行，j 控制列。

2. 程序源代码：

```
#include "stdio.h"
main()
{
    int i,j,result;
    printf("\n");
    for (i=1;i<10;i++)
        { for(j=1;j<10;j++)
            {
                result=i*j;
                printf("%d*d=%-3d",i,j,result);/*-3d 表示左对齐，占 3 位*/
            }
            printf("\n");/*每一行后换行*/
        }
}
```

【程序 9】

题目：要求输出国际象棋棋盘。

1. 程序分析：用 i 控制行，j 来控制列，根据 i+j 的和的变化来控制输出黑方格，还是白方格。

2. 程序源代码：

```
#include "stdio.h"
main()
{
    int i,j;
    for(i=0;i<8;i++)
    {
        for(j=0;j<8;j++)
            if((i+j)%2==0)
                printf("%c%c",219,219);
            else
                printf(" ");
            printf("\n");
        }
}
```

```

    }
}
=====

```

【程序 10】

题目：打印楼梯，同时在楼梯上方打印两个笑脸。

1. 程序分析：用 i 控制行，j 来控制列，j 根据 i 的变化来控制输出黑方格的个数。

2. 程序源代码：

```

#include "stdio.h"
main()
{
    int i, j;
    printf("\1\1\n"); /*输出两个笑脸*/
    for(i=1; i<11; i++)
    {
        for(j=1; j<=i; j++)
            printf("%c%c", 219, 219);
        printf("\n");
    }
}

```

【程序 11】

题目：古典问题：有一对兔子，从出生后第 3 个月起每个月都生一对兔子，小兔子长到第三个月后每个月又生一对兔子，假如兔子都不死，问每个月的兔子总数为多少？

1. 程序分析：兔子的规律为数列 1, 1, 2, 3, 5, 8, 13, 21...

2. 程序源代码：

```

main()
{
    long f1, f2;
    int i;
    f1=f2=1;
    for(i=1; i<=20; i++)
    { printf("%12ld %12ld", f1, f2);
      if(i%2==0) printf("\n"); /*控制输出，每行四个*/
      f1=f1+f2; /*前两个月加起来赋值给第三个月*/
      f2=f1+f2; /*前两个月加起来赋值给第三个月*/
    }
}
=====

```

【程序 12】

题目：判断 101-200 之间有多少个素数，并输出所有素数。

1. 程序分析：判断素数的方法：用一个数分别去除 2 到 sqrt(这个数)，如果能被整除，则表明此数不是素数，反之是素数。

2. 程序源代码：

```

#include "math.h"
main()

```

```

{
    int m, i, k, h=0, leap=1;
    printf("\n");
    for(m=101; m<=200; m++)
    {
        k=sqrt(m+1);
        for(i=2; i<=k; i++)
            if(m%i==0)
                {leap=0; break;}
        if(leap) {printf("%-4d", m); h++;
                    if(h%10==0)
                        printf("\n");
                }
        leap=1;
    }
    printf("\nThe total is %d", h);
}

```

【程序 13】

题目：打印出所有的“水仙花数”，所谓“水仙花数”是指一个三位数，其各位数字立方和等于该数本身。例如：153 是一个“水仙花数”，因为 $153=1$ 的三次方+ 5 的三次方+ 3 的三次方。

1. 程序分析：利用 for 循环控制 100-999 个数，每个数分解出个位，十位，百位。

2. 程序源代码：

```

main()
{
    int i, j, k, n;
    printf("'water flower' number is:");
    for(n=100; n<1000; n++)
    {
        i=n/100; /*分解出百位*/
        j=n/10%10; /*分解出十位*/
        k=n%10; /*分解出个位*/
        if(i*100+j*10+k==i*i*i+j*j*j+k*k*k)
        {
            printf("%-5d", n);
        }
    }
    printf("\n");
}

```

【程序 14】

题目：将一个正整数分解质因数。例如：输入 90，打印出 $90=2*3*3*5$ 。

程序分析：对 n 进行分解质因数，应先找到一个最小的质数 k，然后按下述步骤完成：

- (1) 如果这个质数恰等于 n ，则说明分解质因数的过程已经结束，打印出即可。
- (2) 如果 $n < k$ ，但 n 能被 k 整除，则应打印出 k 的值，并用 n 除以 k 的商，作为新的正整数 n ，重复执行第一步。
- (3) 如果 n 不能被 k 整除，则用 $k+1$ 作为 k 的值，重复执行第一步。

2. 程序源代码:

```
/* zheng int is divided yinshu*/
main()
{
    int n, i;
    printf("\nplease input a number:\n");
    scanf("%d", &n);
    printf("%d=", n);
    for(i=2; i<=n; i++)
    {
        while(n!=i)
        {
            if(n%i==0)
            { printf("%d*", i);
              n=n/i;
            }
            else
                break;
        }
    }
    printf("%d", n);}
=====
```

【程序 15】

题目：利用条件运算符的嵌套来完成此题：学习成绩 ≥ 90 分的同学用 A 表示，60-89 分之间的用 B 表示，60 分以下的用 C 表示。

1. 程序分析： $(a > b) ? a : b$ 这是条件运算符的基本例子。

2. 程序源代码:

```
main()
{
    int score;
    char grade;
    printf("please input a score\n");
    scanf("%d", &score);
    grade=score>=90?'A':(score>=60?'B':'C');
    printf("%d belongs to %c", score, grade);
}
=====
```

【程序 16】

题目：输入两个正整数 m 和 n ，求其最大公约数和最小公倍数。

1. 程序分析：利用辗除法。

2. 程序源代码：

```
main()
{
    int a,b,num1,num2,temp;
    printf("please input two numbers:\n");
    scanf("%d,%d",&num1,&num2);
    if(num1 > num2) { temp=num1;
        num1=num2;
        num2=temp;
    }
    a=num1;b=num2;
    while(b!=0)/*利用辗除法，直到b为0为止*/
    {
        temp=a%b;
        a=b;
        b=temp;
    }
    printf("gongyueshu:%d\n",a);
    printf("gongbeishu:%d\n",num1*num2/a);
}
```

【程序 17】

题目：输入一行字符，分别统计出其中英文字母、空格、数字和其它字符的个数。

1. 程序分析：利用 while 语句, 条件为输入的字符不为 '\n'。

2. 程序源代码：

```
#include "stdio.h"
main()
{char c;
    int letters=0,space=0,digit=0,others=0;
    printf("please input some characters\n");
    while((c=getchar())!='\n')
    {
        if(c>='a' && c<='z' || c>='A' && c<='Z')
            letters++;
        else if(c==' ')
            space++;
        else if(c>='0' && c<='9')
            digit++;
        else
            others++;
    }
    printf("all in all:char=%d space=%d digit=%d others=%d\n",letters,
        space,digit,others);
}
```

}

【程序 18】

题目：求 $s=a+aa+aaa+aaaa+aa\dots a$ 的值，其中 a 是一个数字。例如 $2+22+222+2222+22222$ (此时共有 5 个数相加)，几个数相加有键盘控制。

1. 程序分析：关键是计算出每一项的值。

2. 程序源代码：

```
main()
{
    int a,n,count=1;
    long int sn=0,tn=0;
    printf("please input a and n\n");
    scanf("%d,%d",&a,&n);
    printf("a=%d,n=%d\n",a,n);
    while(count<=n)
    {
        tn=tn+a;
        sn=sn+tn;
        a=a*10;
        ++count;
    }
    printf("a+aa+...=%ld\n",sn);
}
```

【程序 19】

题目：一个数如果恰好等于它的因子之和，这个数就称为“完数”。例如 $6=1+2+3$ 。编程找出 1000 以内的所有完数。

1. 程序分析：请参照程序<--上页程序 14。

2. 程序源代码：

```
main()
{
    static int k[10];
    int i,j,n,s;
    for(j=2;j<1000;j++)
    {
        n=-1;
        s=j;
        for(i=1;i<j;i++)
        {
            if((j%i)==0)
            {
                n++;
                s=s-i;
                k[n]=i;
            }
        }
    }
}
```

```

    if(s==0)
    {
        printf("%d is a wanshu",j);
        for(i=0;i<j;i++) printf("%d",k[i]);
        printf("%d\n",k[n]);
    }
}
}

```

【程序 20】

题目：一球从 100 米高度自由落下，每次落地后反跳回原高度的一半；再落下，求它在第 10 次落地时，共经过多少米？第 10 次反弹多高？

1. 程序分析：见下面注释

2. 程序源代码：

```

main()
{
    float sn=100.0,hn=sn/2;
    int n;
    for(n=2;n<=10;n++)
    {
        sn=sn+2*hn;/*第 n 次落地时共经过的米数*/
        hn=hn/2; /*第 n 次反跳高度*/
    }
    printf("the total of road is %f\n",sn);
    printf("the tenth is %f meter\n",hn);
}

```

【程序 21】

题目：猴子吃桃问题：猴子第一天摘下若干个桃子，当即吃了一半，还不瘾，又多吃了一个，第二天早上又将剩下的桃子吃掉一半，又多吃了一个。以后每天早上都吃了前一天剩下的一半零一个。到第 10 天早上想再吃时，见只剩下一个桃子了。求第一天共摘了多少。

1. 程序分析：采取逆向思维的方法，从后往前推断。

2. 程序源代码：

```

main()
{
    int day,x1,x2;
    day=9;
    x2=1;
    while(day>0)
    {
        x1=(x2+1)*2;/*第一天的桃子数是第 2 天桃子数加 1 后的 2 倍*/
        x2=x1;
        day--;
    }
    printf("the total is %d\n",x1);
}

```

【程序 22】

题目：两个乒乓球队进行比赛，各出三人。甲队为 a, b, c 三人，乙队为 x, y, z 三人。已抽签决定比赛名单。有人向队员打听比赛的名单。a 说他不和 x 比，c 说他不和 x, z 比，请编程序找出三队赛手的名单。

1. 程序分析：判断素数的方法：用一个数分别去除 2 到 sqrt(这个数)，如果能被整除，则表明此数不是素数，反之是素数。

2. 程序源代码：

```
main()
{
char i, j, k; /*i 是 a 的对手, j 是 b 的对手, k 是 c 的对手*/
for(i='x'; i<='z'; i++)
    for(j='x'; j<='z'; j++)
    {
        if(i!=j)
            for(k='x'; k<='z'; k++)
            { if(i!=k&&j!=k)
                { if(i!='x' &&k!='x' &&k!='z')
                    printf("order is a--%c\tb--%c\tc--%c\n", i, j, k);
                }
            }
    }
}
```

【程序 23】

题目：打印出如下图案（菱形）

```
*
***
*****
*****
*****
***
*
```

1. 程序分析：先把图形分成两部分来看待，前四行一个规律，后三行一个规律，利用双重 for 循环，第一层控制行，第二层控制列。

2. 程序源代码：

```
main()
{
int i, j, k;
for(i=0; i<=3; i++)
{
    for(j=0; j<=2-i; j++)
        printf(" ");
}
```



```

    for(k=0;k<=2*i;k++)
        printf("*");
    printf("\n");
}
for(i=0;i<=2;i++)
{
    for(j=0;j<=i;j++)
        printf(" ");
    for(k=0;k<=4-2*i;k++)
        printf("*");
    printf("\n");
}
}

```

【程序 24】

题目：有一分数序列：2/1, 3/2, 5/3, 8/5, 13/8, 21/13... 求出这个数列的前 20 项之和。

1. 程序分析：请抓住分子与分母的变化规律。

2. 程序源代码：

```

main()
{
    int n, t, number=20;
    float a=2, b=1, s=0;
    for(n=1;n<=number;n++)
    {
        s=s+a/b;
        t=a;a=a+b;b=t; /*这部分是程序的关键，请读者猜猜 t 的作用*/
    }
    printf("sum is %9.6f\n", s);
}

```

【程序 25】

题目：求 $1+2!+3!+\dots+20!$ 的和

1. 程序分析：此程序只是把累加变成了累乘。

2. 程序源代码：

```

main()
{
    float n, s=0, t=1;
    for(n=1;n<=20;n++)
    {
        t*=n;
        s+=t;
    }
    printf("1+2!+3!...+20!=%e\n", s);
}

```

=====

【程序 26】

题目：利用递归方法求 5!。

1. 程序分析：递归公式：fn=fn_1*4!

2. 程序源代码：

```
#include "stdio.h"
main()
{
    int i;
    int fact();
    for(i=0;i<5;i++)
        printf("\40:%d!=%d\n", i, fact(i));
}
int fact(j)
int j;
{
    int sum;
    if(j==0)
        sum=1;
    else
        sum=j*fact(j-1);
    return sum;
}
```

=====

【程序 27】

题目：利用递归函数调用方式，将所输入的 5 个字符，以相反顺序打印出来。

1. 程序分析：

2. 程序源代码：

```
#include "stdio.h"
main()
{
    int i=5;
    void palin(int n);
    printf("\40:");
    palin(i);
    printf("\n");
}
void palin(n)
int n;
{
    char next;
    if(n<=1)
    {
        next=getchar();
    }
}
```

```

    printf("\n0:");
    putchar(next);
}
else
{
    next=getchar();
    palin(n-1);
    putchar(next);
}
}

```

【程序 28】

题目：有 5 个人坐在一起，问第五个人多少岁？他说比第 4 个人大 2 岁。问第 4 个人岁数，他说比第 3 个人大 2 岁。问第三个人，又说比第 2 人大两岁。问第 2 个人，说比第一个人大两岁。最后问第一个人， he 说是 10 岁。请问第五个人多大？

1. 程序分析：利用递归的方法，递归分为回推和递推两个阶段。要想知道第五个人岁数，需知道第四人的岁数，依次类推，推到第一人（10 岁），再往回推。

2. 程序源代码：

```

age(n)
int n;
{
    int c;
    if(n==1) c=10;
    else c=age(n-1)+2;
    return(c);
}
main()
{ printf("%d",age(5));
}

```

【程序 29】

题目：给一个不多于 5 位的正整数，要求：一、求它是几位数，二、逆序打印出各位数字。

1. 程序分析：学会分解出每一位数，如下解释：

2. 程序源代码：

```

main( )
{
    long a,b,c,d,e,x;
    scanf("%ld",&x);
    a=x/10000; /*分解出万位*/
    b=x%10000/1000; /*分解出千位*/
    c=x%1000/100; /*分解出百位*/
    d=x%100/10; /*分解出十位*/
    e=x%10; /*分解出个位*/
    if (a!=0) printf("there are 5, %ld %ld %ld %ld %ld\n",e,d,c,b,a);
}

```

```

else if (b!=0) printf("there are 4, %ld %ld %ld %ld\n",e,d,c,b);
    else if (c!=0) printf(" there are 3,%ld %ld %ld\n",e,d,c);
        else if (d!=0) printf("there are 2, %ld %ld\n",e,d);
            else if (e!=0) printf(" there are 1,%ld\n",e);
}

```

【程序 30】

题目：一个 5 位数，判断它是不是回文数。即 12321 是回文数，个位与万位相同，十位与千位相同。

1. 程序分析：同 29 例

2. 程序源代码：

```

main( )
{
long ge,shi,qian,wan,x;
scanf("%ld",&x);
wan=x/10000;
qian=x%10000/1000;
shi=x%100/10;
ge=x%10;
if (ge==wan&&shi==qian)/*个位等于万位并且十位等于千位*/
    printf("this number is a huiwen\n");
else
    printf("this number is not a huiwen\n");
}

```

【程序 31】

题目：请输入星期几的第一个字母来判断一下是星期几，如果第一个字母一样，则继续判断第二个字母。

1. 程序分析：用情况语句比较好，如果第一个字母一样，则判断用情况语句或 if 语句判断第二个字母。

2. 程序源代码：

```

#include <stdio.h>
void main()
{
char letter;
printf("please input the first letter of someday\n");
while ((letter=getch())!='Y')/*当所按字母为 Y 时才结束*/
{ switch (letter)
{case 'S':printf("please input second letter\n");
    if((letter=getch())=='a')
        printf("saturday\n");
    else if ((letter=getch())=='u')
        printf("sunday\n");
    else printf("data error\n");
    break;
}
}
}

```

```

case 'F':printf("friday\n");break;
case 'M':printf("monday\n");break;
case 'T':printf("please input second letter\n");
        if((letter=getch())=='u')
            printf("tuesday\n");
        else if ((letter=getch())=='h')
            printf("thursday\n");
        else printf("data error\n");
        break;
case 'W':printf("wednesday\n");break;
default: printf("data error\n");
    }
}
}

```

【程序 32】

题目：Press any key to change color, do you want to try it. Please hurry up!

1. 程序分析：

2. 程序源代码：

```

#include <conio.h>
void main(void)
{
    int color;
    for (color = 0; color < 8; color++)
    {
        textbackground(color);/*设置文本的背景颜色*/
        cprintf("This is color %d\r\n", color);
        cprintf("Press any key to continue\r\n");
        getch();/*输入字符看不见*/
    }
}

```

【程序 33】

题目：学习 gotoxy() 与 clrscr() 函数

1. 程序分析：

2. 程序源代码：

```

#include <conio.h>
void main(void)
{
    clrscr();/*清屏函数*/
    textbackground(2);
    gotoxy(1, 5);/*定位函数*/
    cprintf("Output at row 5 column 1\n");
    textbackground(3);
}

```

```
gotoxy(20, 10);
cprintf("Output at row 10 column 20\n");
}
```

【程序 34】

题目：练习函数调用

1. 程序分析：

2. 程序源代码：

```
#include <stdio.h>
void hello_world(void)
{
printf("Hello, world!\n");
}
void three_hellos(void)
{
int counter;
for (counter = 1; counter <= 3; counter++)
hello_world();/*调用此函数*/
}
void main(void)
{
three_hellos();/*调用此函数*/
}
```

【程序 35】

题目：文本颜色设置

1. 程序分析：

2. 程序源代码：

```
#include <conio.h>
void main(void)
{
int color;
for (color = 1; color < 16; color++)
{
textcolor(color);/*设置文本颜色*/
cprintf("This is color %d\r\n", color);
}
textcolor(128 + 15);
cprintf("This is blinking\r\n");
}
```

【程序 36】

题目：求 100 之内的素数

1. 程序分析：

2. 程序源代码:

```

#include <stdio.h>
#include "math.h"
#define N 101
main()
{
    int i, j, line, a[N];
    for(i=2; i<N; i++) a[i]=i;
    for(i=2; i<sqrt(N); i++)
        for(j=i+1; j<N; j++)
        {
            if(a[i]!=0&& a[j]!=0)
                if(a[j]%a[i]==0)
                    a[j]=0;
        }
    printf("\n");
    for(i=2, line=0; i<N; i++)
    {
        if(a[i]!=0)
            {printf("%5d", a[i]);
              line++;}
        if(line==10)
            {printf("\n");
              line=0;}
    }
}

```

=====

【程序 37】

题目：对 10 个数进行排序

1. 程序分析：可以利用选择法，即从后 9 个比较过程中，选择一个最小的与第一个元素交换，下次类推，即用第二个元素与后 8 个进行比较，并进行交换。

2. 程序源代码：

```

#define N 10
main()
{int i, j, min, tem, a[N];
  /*input data*/
  printf("please input ten num:\n");
  for(i=0; i<N; i++)
  {
    printf("a[%d]=", i);
    scanf("%d", &a[i]);
  }
  printf("\n");
  for(i=0; i<N; i++)
    printf("%5d", a[i]);
  printf("\n");
}

```

```

/*sort ten num*/
for(i=0;i<N-1;i++)
{min=i;
for(j=i+1;j<N;j++)
if(a[min]>a[j]) min=j;
tem=a[i];
a[i]=a[min];
a[min]=tem;
}

/*output data*/
printf("After sorted \n");
for(i=0;i<N;i++)
printf("%5d",a[i]);
}

```

【程序 38】

题目：求一个 3*3 矩阵对角线元素之和

1. 程序分析：利用双重 for 循环控制输入二维数组，再将 a[i][i] 累加后输出。

2. 程序源代码：

```

main()
{
float a[3][3],sum=0;
int i,j;
printf("please input rectangle element:\n");
for(i=0;i<3;i++)
for(j=0;j<3;j++)
scanf("%f",&a[i][j]);
for(i=0;i<3;i++)
sum=sum+a[i][i];
printf("duijiaoxian he is %6.2f",sum);
}

```

【程序 39】

题目：有一个已经排好序的数组。现输入一个数，要求按原来的规律将它插入数组中。

1. 程序分析：首先判断此数是否大于最后一个数，然后再考虑插入中间的数的情况，插入后此元素之后的数，依次后移一个位置。

2. 程序源代码：

```

main()
{
int a[11]={1,4,6,9,13,16,19,28,40,100};
int temp1,temp2,number,end,i,j;
printf("original array is:\n");
for(i=0;i<10;i++)
printf("%5d",a[i]);

```



```

printf("\n");
printf("insert a new number:");
scanf("%d",&number);
end=a[9];
if (number>end)
    a[10]=number;
else
    {for(i=0;i<10;i++)
      { if(a[i]>number)
        {temp1=a[i];
          a[i]=number;
          for(j=i+1;j<11;j++)
            {temp2=a[j];
              a[j]=temp1;
              temp1=temp2;
            }
          break;
        }
      }
    }
}
for(i=0;i<11;i++)
    printf("%6d",a[i]);
}
=====

```

【程序 40】

题目：将一个数组逆序输出。

1. 程序分析：用第一个与最后一个交换。

2. 程序源代码：

```

#define N 5
main()
{ int a[N]={9,6,5,4,1},i,temp;
  printf("\n original array:\n");
  for(i=0;i<N;i++)
    printf("%4d",a[i]);
  for(i=0;i<N/2;i++)
    {temp=a[i];
      a[i]=a[N-i-1];
      a[N-i-1]=temp;
    }
  printf("\n sorted array:\n");
  for(i=0;i<N;i++)
    printf("%4d",a[i]);
}

```

【程序 41】

题目：学习 static 定义静态变量的用法

1. 程序分析：

2. 程序源代码：

```
#include "stdio.h"
varfunc()
{
    int var=0;
    static int static_var=0;
    printf("\40:var equal %d \n",var);
    printf("\40:static var equal %d \n",static_var);
    printf("\n");
    var++;
    static_var++;
}

void main()
{int i;
  for(i=0;i<3;i++)
    varfunc();
}
```

=====

【程序 42】

题目：学习使用 auto 定义变量的用法

1. 程序分析：

2. 程序源代码：

```
#include "stdio.h"
main()
{int i,num;
 num=2;
  for (i=0;i<3;i++)
  { printf("\40: The num equal %d \n",num);
    num++;
    {
      auto int num=1;
      printf("\40: The internal block num equal %d \n",num);
      num++;
    }
  }
}
```

=====

【程序 43】

题目：学习使用 static 的另一用法。

1. 程序分析：

2. 程序源代码：

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

```
main()
{
    int i,num;
    num=2;
    for(i=0;i<3;i++)
    {
        printf("\40: The num equal %d \n",num);
        num++;
    }
    static int num=1;
    printf("\40:The internal block num equal %d\n",num);
    num++;
}
}
```

【程序 44】

题目：学习使用 external 的用法。

1. 程序分析：

2. 程序源代码：

```
#include "stdio.h"
int a,b,c;
void add()
{ int a;
  a=3;
  c=a+b;
}
void main()
{ a=b=4;
  add();
  printf("The value of c is equal to %d\n",c);
}
```

【程序 45】

题目：学习使用 register 定义变量的方法。

1. 程序分析：

2. 程序源代码：

```
void main()
{
    register int i;
    int tmp=0;
    for(i=1;i<=100;i++)
        tmp+=i;
    printf("The sum is %d\n",tmp);
}
```

}

【程序 46】

题目：宏#define 命令练习(1)

1. 程序分析：

2. 程序源代码：

```
#include "stdio.h"
#define TRUE 1
#define FALSE 0
#define SQ(x) (x)*(x)
void main()
{
    int num;
    int again=1;
    printf("\40: Program will stop if input value less than 50.\n");
    while(again)
    {
        printf("\40:Please input number==>");
        scanf("%d",&num);
        printf("\40:The square for this number is %d \n",SQ(num));
        if(num>=50)
            again=TRUE;
        else
            again=FALSE;
    }
}
```

【程序 47】

题目：宏#define 命令练习(2)

1. 程序分析：

2. 程序源代码：

```
#include "stdio.h"
#define exchange(a,b) { \ /*宏定义中允许包含两道衣裳命令的情形，此时必须在最右边加上"\*/
                                int t;\
                                t=a;\
                                a=b;\
                                b=t;\
                                }

void main(void)
{
    int x=10;
    int y=20;
    printf("x=%d; y=%d\n",x,y);
```

```
exchange(x, y);
printf("x=%d; y=%d\n", x, y);
}
```

【程序 48】

题目：宏#define 命令练习(3)

1. 程序分析：

2. 程序源代码：

```
#define LAG >
#define SMA <
#define EQ ==
#include "stdio.h"
void main()
{ int i=10;
  int j=20;
  if(i LAG j)
    printf("\40: %d larger than %d \n", i, j);
  else if(i EQ j)
    printf("\40: %d equal to %d \n", i, j);
  else if(i SMA j)
    printf("\40:%d smaller than %d \n", i, j);
  else
    printf("\40: No such value.\n");
}
```

【程序 49】

题目：#if #ifdef 和#endif 的综合应用。

1. 程序分析：

2. 程序源代码：

```
#include "stdio.h"
#define MAX
#define MAXIMUM(x, y) (x>y)?x:y
#define MINIMUM(x, y) (x>y)?y:x
void main()
{ int a=10, b=20;
#ifdef MAX
  printf("\40: The larger one is %d\n", MAXIMUM(a, b));
#else
  printf("\40: The lower one is %d\n", MINIMUM(a, b));
#endif
#ifndef MIN
  printf("\40: The lower one is %d\n", MINIMUM(a, b));
#else
  printf("\40: The larger one is %d\n", MAXIMUM(a, b));
}
```

```
#endif
#undef MAX
#ifdef MAX
printf("\40: The larger one is %d\n", MAXIMUM(a, b));
#else
printf("\40: The lower one is %d\n", MINIMUM(a, b));
#endif
#define MIN
#ifndef MIN
printf("\40: The lower one is %d\n", MINIMUM(a, b));
#else
printf("\40: The larger one is %d\n", MAXIMUM(a, b));
#endif
}
```

=====

【程序 50】

题目：#include 的应用练习

1. 程序分析：

2. 程序源代码：

test.h 文件如下：

```
#define LAG >
#define SMA <
#define EQ ==
#include "test.h" /*一个新文件 50.c, 包含 test.h*/
#include "stdio.h"
void main()
{ int i=10;
  int j=20;
  if(i LAG j)
    printf("\40: %d larger than %d \n", i, j);
  else if(i EQ j)
    printf("\40: %d equal to %d \n", i, j);
  else if(i SMA j)
    printf("\40:%d smaller than %d \n", i, j);
  else
    printf("\40: No such value.\n");
}
```

【程序 51】

题目：学习使用按位与 & 。

1. 程序分析：0&0=0; 0&1=0; 1&0=0; 1&1=1

2. 程序源代码：

```
#include "stdio.h"
main()
{
```

```

int a,b;
a=077;
b=a&3;
printf("\40: The a & b(decimal) is %d \n",b);
b&=7;
printf("\40: The a & b(decimal) is %d \n",b);
}

```

【程序 52】

题目：学习使用按位或 $|$ 。

1. 程序分析： $0|0=0$ ； $0|1=1$ ； $1|0=1$ ； $1|1=1$

2. 程序源代码：

```

#include "stdio.h"
main()
{
    int a,b;
    a=077;
    b=a|3;
    printf("\40: The a & b(decimal) is %d \n",b);
    b|=7;
    printf("\40: The a & b(decimal) is %d \n",b);
}

```

【程序 53】

题目：学习使用按位异或 \wedge 。

1. 程序分析： $0\wedge0=0$ ； $0\wedge1=1$ ； $1\wedge0=1$ ； $1\wedge1=0$

2. 程序源代码：

```

#include "stdio.h"
main()
{
    int a,b;
    a=077;
    b=a^3;
    printf("\40: The a & b(decimal) is %d \n",b);
    b^=7;
    printf("\40: The a & b(decimal) is %d \n",b);
}

```

【程序 54】

题目：取一个整数 a 从右端开始的 4~7 位。

程序分析：可以这样考虑：

- (1) 先使 a 右移 4 位。
- (2) 设置一个低 4 位全为 1, 其余全为 0 的数。可用 $\sim(\sim0<<4)$
- (3) 将上面二者进行 $\&$ 运算。

2. 程序源代码:

```
main()
{
    unsigned a, b, c, d;
    scanf("%o", &a);
    b=a>>4;
    c=~(0<<4);
    d=b&c;
    printf("%o\n%o\n", a, d);
}
```

=====

【程序 55】

题目：学习使用按位取反~。

1. 程序分析：~0=1; ~1=0;

2. 程序源代码:

```
#include "stdio.h"
main()
{
    int a, b;
    a=234;
    b=~a;
    printf("\40: The a's 1 complement(decimal) is %d \n", b);
    a=~a;
    printf("\40: The a's 1 complement(hexidecimal) is %x \n", a);
}
```

=====

【程序 56】

题目：画图，学用 circle 画圆形。

1. 程序分析:

2. 程序源代码:

```
/*circle*/
#include "graphics.h"
main()
{int driver,mode,i;
float j=1,k=1;
driver=VGA;mode=VGAHI;
initgraph(&driver,&mode,"");
setbkcolor(YELLOW);
for(i=0;i<=25;i++)
{
    setcolor(8);
    circle(310,250,k);
    k=k+j;
    j=j+0.3;
}
```



```

}
}

```

【程序 57】

题目：画图，学用 line 画直线。

1. 程序分析：

2. 程序源代码：

```

#include "graphics.h"
main()
{
    int driver, mode, i;
    float x0, y0, y1, x1;
    float j=12, k;
    driver=VGA; mode=VGAHI;
    initgraph(&driver, &mode, "");
    setbkcolor(GREEN);
    x0=263; y0=263; y1=275; x1=275;
    for(i=0; i<=18; i++)
    {
        setcolor(5);
        line(x0, y0, x0, y1);
        x0=x0-5;
        y0=y0-5;
        x1=x1+5;
        y1=y1+5;
        j=j+10;
    }
    x0=263; y1=275; y0=263;
    for(i=0; i<=20; i++)
    {
        setcolor(5);
        line(x0, y0, x0, y1);
        x0=x0+5;
        y0=y0+5;
        y1=y1-5;
    }
}

```

【程序 58】

题目：画图，学用 rectangle 画方形。

1. 程序分析：利用 for 循环控制 100-999 个数，每个数分解出个位，十位，百位。

2. 程序源代码：

```

#include "graphics.h"
main()
{
    int x0, y0, y1, x1, driver, mode, i;

```

```

driver=VGA;mode=VGAHI;
initgraph(&driver,&mode,"");
setbkcolor(YELLOW);
x0=263;y0=263;y1=275;x1=275;
for(i=0;i<=18;i++)
{
setcolor(1);
rectangle(x0,y0,x1,y1);
x0=x0-5;
y0=y0-5;
x1=x1+5;
y1=y1+5;
}
settextstyle(DEFAULT_FONT,HORIZ_DIR,2);
outtextxy(150,40,"How beautiful it is!");
line(130,60,480,60);
setcolor(2);
circle(269,269,137);
}
=====

```

【程序 59】

题目：画图，综合例子。

1. 程序分析：

2. 程序源代码：

```

# define PAI 3.1415926
# define B 0.809
# include "graphics.h"
#include "math.h"
main()
{
int i,j,k,x0,y0,x,y,driver,mode;
float a;
driver=CGA;mode=CGACO;
initgraph(&driver,&mode,"");
setcolor(3);
setbkcolor(GREEN);
x0=150;y0=100;
circle(x0,y0,10);
circle(x0,y0,20);
circle(x0,y0,50);
for(i=0;i<16;i++)
{
a=(2*PAI/16)*i;
x=ceil(x0+48*cos(a));

```

```

    y=ceil(y0+48*sin(a)*B);
    setcolor(2); line(x0,y0,x,y);}
setcolor(3);circle(x0,y0,60);
/* Make 0 time normal size letters */
settextstyle(DEFAULT_FONT,HORIZ_DIR,0);
outtextxy(10,170,"press a key");
getch();
setfillstyle(HATCH_FILL,YELLOW);
floodfill(202,100,WHITE);
getch();
for(k=0;k<=500;k++)
{
    setcolor(3);
    for(i=0;i<=16;i++)
    {
        a=(2*PAI/16)*i+(2*PAI/180)*k;
        x=ceil(x0+48*cos(a));
        y=ceil(y0+48*sin(a)*B);
        setcolor(2); line(x0,y0,x,y);
    }
    for(j=1;j<=50;j++)
    {
        a=(2*PAI/16)*i+(2*PAI/180)*k-1;
        x=ceil(x0+48*cos(a));
        y=ceil(y0+48*sin(a)*B);
        line(x0,y0,x,y);
    }
}
restorecrtmode();
}

```

【程序 60】

题目：画图，综合例子。

1. 程序分析：

2. 程序源代码：

```

#include "graphics.h"
#define LEFT 0
#define TOP 0
#define RIGHT 639
#define BOTTOM 479
#define LINES 400
#define MAXCOLOR 15
main()
{

```

```

int driver, mode, error;
int x1, y1;
int x2, y2;
int dx1, dy1, dx2, dy2, i=1;
int count=0;
int color=0;
driver=VGA;
mode=VGAHI;
initgraph(&driver, &mode, "");
x1=x2=y1=y2=10;
dx1=dy1=2;
dx2=dy2=3;
while(!kbhit())
{
    line(x1, y1, x2, y2);
    x1+=dx1; y1+=dy1;
    x2+=dx2; y2+=dy2;
    if(x1<=LEFT || x1>=RIGHT)
        dx1=-dx1;
    if(y1<=TOP || y1>=BOTTOM)
        dy1=-dy1;
    if(x2<=LEFT || x2>=RIGHT)
        dx2=-dx2;
    if(y2<=TOP || y2>=BOTTOM)
        dy2=-dy2;
    if(++count>LINES)
    {
        setcolor(color);
        color=(color>=MAXCOLOR)?0:++color;
    }
}
closegraph();
}

```

【程序 61】

题目：打印出杨辉三角形（要求打印出 10 行如下图）

1. 程序分析：

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1

```

2. 程序源代码：

```
main()
```

```
{int i, j;
int a[10][10];
printf("\n");
for(i=0;i<10;i++)
    {a[i][0]=1;
    a[i][i]=1;}
for(i=2;i<10;i++)
    for(j=1;j<i;j++)
        a[i][j]=a[i-1][j-1]+a[i-1][j];
for(i=0;i<10;i++)
    {for(j=0;j<=i;j++)
    printf("%5d", a[i][j]);
    printf("\n");
    }
}
```

【程序 62】

题目：学习 putpixel 画点。

1. 程序分析：

2. 程序源代码：

```
#include "stdio.h"
#include "graphics.h"
main()
{
    int i, j, driver=VGA, mode=VGAHI;
    initgraph(&driver, &mode, "");
    setbkcolor(YELLOW);
    for(i=50;i<=230;i+=20)
        for(j=50;j<=230;j++)
            putpixel(i, j, 1);
    for(j=50;j<=230;j+=20)
        for(i=50;i<=230;i++)
            putpixel(i, j, 1);
}
```

【程序 63】

题目：画椭圆 ellipse

1. 程序分析：

2. 程序源代码：

```
#include "stdio.h"
#include "graphics.h"
#include "conio.h"
main()
{
```

```
int x=360, y=160, driver=VGA, mode=VGAHI;
int num=20, i;
int top, bottom;
initgraph(&driver, &mode, "");
top=y-30;
bottom=y-30;
for(i=0; i<num; i++)
{
    ellipse(250, 250, 0, 360, top, bottom);
    top-=5;
    bottom+=5;
}
getch();
}
```

=====

【程序 64】

题目：利用 ellipse and rectangle 画图。

1. 程序分析：

2. 程序源代码：

```
#include "stdio.h"
#include "graphics.h"
#include "conio.h"
main()
{
    int driver=VGA, mode=VGAHI;
    int i, num=15, top=50;
    int left=20, right=50;
    initgraph(&driver, &mode, "");
    for(i=0; i<num; i++)
    {
        ellipse(250, 250, 0, 360, right, left);
        ellipse(250, 250, 0, 360, 20, top);
        rectangle(20-2*i, 20-2*i, 10*(i+2), 10*(i+2));
        right+=5;
        left+=5;
        top+=10;
    }
    getch();
}
```

=====

【程序 65】

题目：一个最优美的图案。

1. 程序分析：

2. 程序源代码：

```
#include "graphics.h"
#include "math.h"
#include "dos.h"
#include "conio.h"
#include "stdlib.h"
#include "stdio.h"
#include "stdarg.h"
#define MAXPTS 15
#define PI 3.1415926
struct PTS {
    int x,y;
};
double AspectRatio=0.85;
void LineToDemo(void)
{
    struct viewporttype vp;
    struct PTS points[MAXPTS];
    int i, j, h, w, xcenter, ycenter;
    int radius, angle, step;
    double rads;
    printf(" MoveTo / LineTo Demonstration" );
    getviewsettings( &vp );
    h = vp.bottom - vp.top;
    w = vp.right - vp.left;
    xcenter = w / 2; /* Determine the center of circle */
    ycenter = h / 2;
    radius = (h - 30) / (AspectRatio * 2);
    step = 360 / MAXPTS; /* Determine # of increments */
    angle = 0; /* Begin at zero degrees */
    for( i=0 ; i<MAXPTS ; ++i ){ /* Determine circle intercepts */
        rads = (double)angle * PI / 180.0; /* Convert angle to radians */
        points[i].x = xcenter + (int)( cos(rads) * radius );
        points[i].y = ycenter - (int)( sin(rads) * radius * AspectRatio );
        angle += step; /* Move to next increment */
    }
    circle( xcenter, ycenter, radius ); /* Draw bounding circle */
    for( i=0 ; i<MAXPTS ; ++i ){ /* Draw the cords to the circle */
        for( j=i ; j<MAXPTS ; ++j ){ /* For each remaining intersect */
            moveto(points[i].x, points[i].y); /* Move to beginning of cord */
            lineto(points[j].x, points[j].y); /* Draw the cord */
        } }
}
main()
{int driver,mode;
driver=CGA;mode=CGACO;
```

```

initgraph(&driver,&mode,"");
setcolor(3);
setbkcolor(GREEN);
LineToDemo();}
=====

```

【程序 66】

题目：输入 3 个数 a, b, c，按大小顺序输出。

1. 程序分析：利用指针方法。

2. 程序源代码：

```

/*pointer*/
main()
{
    int n1,n2,n3;
    int *pointer1,*pointer2,*pointer3;
    printf("please input 3 number:n1,n2,n3:");
    scanf("%d,%d,%d",&n1,&n2,&n3);
    pointer1=&n1;
    pointer2=&n2;
    pointer3=&n3;
    if(n1>n2) swap(pointer1,pointer2);
    if(n1>n3) swap(pointer1,pointer3);
    if(n2>n3) swap(pointer2,pointer3);
    printf("the sorted numbers are:%d,%d,%d\n",n1,n2,n3);
}

swap(p1,p2)
int *p1,*p2;
{int p;
p=*p1;*p1=*p2;*p2=p;
}
=====

```

【程序 67】

题目：输入数组，最大的与第一个元素交换，最小的与最后一个元素交换，输出数组。

1. 程序分析：谭浩强的书中答案有问题。

2. 程序源代码：

```

main()
{
    int number[10];
    input(number);
    max_min(number);
    output(number);
}

input(number)
int number[10];
{int i;

```



```

for(i=0;i<9;i++)
    scanf("%d",&number[i]);
scanf("%d",&number[9]);
}
max_min(array)
int array[10];
{int *max,*min,k,l;
int *p,*arr_end;
arr_end=array+10;
max=min=array;
for(p=array+1;p<arr_end;p++)
    if(*p>*max) max=p;
    else if(*p<*min) min=p;
    k=*max;
    l=*min;
    *p=array[0];array[0]=l;l=*p;
    *p=array[9];array[9]=k;k=*p;
    return;
}
output(array)
int array[10];
{ int *p;
for(p=array;p<array+9;p++)
    printf("%d",&*p);
printf("%d\n",array[9]);
}

```

【程序 68】

题目：有 n 个整数，使其前面各数顺序向后移 m 个位置，最后 m 个数变成最前面的 m 个数

1. 程序分析：

2. 程序源代码：

```

main()
{
int number[20],n,m,i;
printf("the total numbers is:");
scanf("%d",&n);
printf("back m:");
scanf("%d",&m);
for(i=0;i<n-1;i++)
    scanf("%d",&number[i]);
scanf("%d",&number[n-1]);
move(number,n,m);
for(i=0;i<n-1;i++)
    printf("%d",&number[i]);
}

```

```

printf("%d", number[n-1]);
}
move(array, n, m)
int n, m, array[20];
{
    int *p, array_end;
    array_end=(array+n-1);
    for(p=array+n-1; p>array; p--)
        *p=*(p-1);
        *array=array_end;
        m--;
        if(m>0) move(array, n, m);
}
=====

```

【程序 69】

题目：有 n 个人围成一圈，顺序排号。从第一个人开始报数（从 1 到 3 报数），凡报到 3 的人退出圈子，问最后留下的是原来第几号的那位。

1. 程序分析：

2. 程序源代码：

```

#define nmax 50
main()
{
    int i, k, m, n, num[nmax], *p;
    printf("please input the total of numbers:");
    scanf("%d", &n);
    p=num;
    for(i=0; i<n; i++)
        *(p+i)=i+1;
        i=0;
        k=0;
        m=0;
        while(m<n-1)
        {
            if(*(p+i)!=0) k++;
            if(k==3)
            { *(p+i)=0;
              k=0;
              m++;
            }
            i++;
            if(i==n) i=0;
        }
        while(*p==0) p++;
        printf("%d is left\n", *p);
}

```

```
}
```

```
=====
```

【程序 70】

题目：写一个函数，求一个字符串的长度，在 main 函数中输入字符串，并输出其长度。

1. 程序分析：

2. 程序源代码：

```
main()
{
    int len;
    char *str[20];
    printf("please input a string:\n");
    scanf("%s", str);
    len=length(str);
    printf("the string has %d characters.", len);
}

length(p)
char *p;
{
    int n;
    n=0;
    while(*p!='\0')
    {
        n++;
        p++;
    }
    return n;
}
```

【程序 71】

题目：编写 input() 和 output() 函数输入，输出 5 个学生的数据记录。

1. 程序分析：

2. 程序源代码：

```
#define N 5
struct student
{ char num[6];
  char name[8];
  int score[4];
} stu[N];
input(stu)
struct student stu[];
{ int i, j;
  for(i=0; i<N; i++)
  { printf("\n please input %d of %d\n", i+1, N);
    printf("num: ");
    scanf("%s", stu[i].num);
```

```

        printf("name: ");
        scanf("%s", stu[i].name);
        for(j=0;j<3;j++)
        { printf("score %d.", j+1);
          scanf("%d",&stu[i].score[j]);
        }
        printf("\n");
    }
}

print(stu)
struct student stu[];
{ int i,j;
printf("\nNo. Name Sco1 Sco2 Sco3\n");
for(i=0;i<N;i++)
{ printf("%-6s%-10s", stu[i].num, stu[i].name);
  for(j=0;j<3;j++)
    printf("%-8d", stu[i].score[j]);
  printf("\n");
}
}

main()
{
    input();
    print();
}

```

【程序 72】

题目：创建一个链表。

1. 程序分析：

2. 程序源代码：

```

/*creat a list*/
#include "stdlib.h"
#include "stdio.h"
struct list
{ int data;
  struct list *next;
};

typedef struct list node;
typedef node *link;
void main()
{ link ptr, head;
  int num, i;
  ptr=(link)malloc(sizeof(node));
  ptr=head;

```

```

printf("please input 5 numbers==>\n");
for(i=0;i<=4;i++)
{
    scanf("%d",&num);
    ptr->data=num;
    ptr->next=(link)malloc(sizeof(node));
    if(i==4) ptr->next=NULL;
    else ptr=ptr->next;
}
ptr=head;
while(ptr!=NULL)
{ printf("The value is ==>%d\n",ptr->data);
  ptr=ptr->next;
}
}

```

【程序 73】

题目：反向输出一个链表。

1. 程序分析：

2. 程序源代码：

```

/*reverse output a list*/
#include "stdlib.h"
#include "stdio.h"
struct list
{ int data;
  struct list *next;
};
typedef struct list node;
typedef node *link;
void main()
{ link ptr,head,tail;
  int num,i;
  tail=(link)malloc(sizeof(node));
  tail->next=NULL;
  ptr=tail;
  printf("\nplease input 5 data==>\n");
  for(i=0;i<=4;i++)
  {
      scanf("%d",&num);
      ptr->data=num;
      head=(link)malloc(sizeof(node));
      head->next=ptr;
      ptr=head;
  }
}

```

```

ptr=ptr->next;
while(ptr!=NULL)
{ printf("The value is ==>%d\n",ptr->data);
  ptr=ptr->next;
}

```

【程序 74】

题目：连接两个链表。

1. 程序分析：

2. 程序源代码：

```

#include "stdlib.h"
#include "stdio.h"
struct list
{ int data;
  struct list *next;
};
typedef struct list node;
typedef node *link;
link delete_node(link pointer,link tmp)
{if (tmp==NULL) /*delete first node*/
  return pointer->next;
else
{ if(tmp->next->next==NULL)/*delete last node*/
  tmp->next=NULL;
  else /*delete the other node*/
  tmp->next=tmp->next->next;
  return pointer;
}
}

void selection_sort(link pointer,int num)
{ link tmp,btmp;
  int i,min;
  for(i=0;i<num;i++)
  {
    tmp=pointer;
    min=tmp->data;
    btmp=NULL;
    while(tmp->next)
    { if(min>tmp->next->data)
      {min=tmp->next->data;
       btmp=tmp;
      }
    }
    tmp=tmp->next;
  }
}

```

```

printf("\40: %d\n",min);
pointer=delete_node(pointer,btmp);
}
}
link create_list(int array[],int num)
{ link tmp1,tmp2,pointer;
int i;
pointer=(link)malloc(sizeof(node));
pointer->data=array[0];
tmp1=pointer;
for(i=1;i<num;i++)
{ tmp2=(link)malloc(sizeof(node));
  tmp2->next=NULL;
  tmp2->data=array[i];
  tmp1->next=tmp2;
  tmp1=tmp1->next;
}
return pointer;
}
link concatenate(link pointer1,link pointer2)
{ link tmp;
tmp=pointer1;
while(tmp->next)
  tmp=tmp->next;
tmp->next=pointer2;
return pointer1;
}
void main(void)
{ int arr1[]={3,12,8,9,11};
  link ptr;
  ptr=create_list(arr1,5);
  selection_sort(ptr,5);
}

```

【程序 75】

题目：放松一下，算一道简单的题目。

1. 程序分析：

2. 程序源代码：

```

main()
{
int i,n;
for(i=1;i<5;i++)
{ n=0;
  if(i!=1)

```

```

    n=n+1;
    if(i==3)
    n=n+1;
    if(i==4)
    n=n+1;
    if(i!=4)
    n=n+1;
    if(n==3)
        printf("zhu hao shi de shi:%c",64+i);
    }
}

```

【程序 76】

题目：编写一个函数，输入 n 为偶数时，调用函数求 $1/2+1/4+\dots+1/n$ ，当输入 n 为奇数时，调用函数 $1/1+1/3+\dots+1/n$ (利用指针函数)

1. 程序分析：

2. 程序源代码：

```

main()
#include "stdio.h"
main()
{
float peven(),podd(),dcall();
float sum;
int n;
while (1)
{
scanf("%d",&n);
if(n>1)
break;
}
if(n%2==0)
{
printf("Even=");
sum=dcall(peven,n);
}
else
{
printf("Odd=");
sum=dcall(podd,n);
}
printf("%f",sum);
}
float peven(int n)
{

```



```
float s;
int i;
s=1;
for(i=2;i<=n;i+=2)
    s+=1/(float)i;
return(s);
}

float podd(n)
int n;
{
float s;
int i;
s=0;
for(i=1;i<=n;i+=2)
    s+=1/(float)i;
return(s);
}

float dcall(fp,n)
float (*fp)();
int n;
{
float s;
s=(*fp)(n);
return(s);
}
```

【程序 77】

题目：填空练习（指向指针的指针）

1. 程序分析：

2. 程序源代码：

```
main()
{ char *s[]={"man","woman","girl","boy","sister"};
char **q;
int k;
for(k=0;k<5;k++)
{
    ;/*这里填写什么语句*/
    printf("%s\n",*q);
}
}
```

【程序 78】

题目：找到年龄最大的人，并输出。请找出程序中有什么问题。

1. 程序分析：

2. 程序源代码：

```

#define N 4
#include "stdio.h"
static struct man
{ char name[20];
  int age;
} person[N]={ "li", 18, "wang", 19, "zhang", 20, "sun", 22 };
main()
{ struct man *q, *p;
  int i, m=0;
  p=person;
  for (i=0; i<N; i++)
  { if (m<p->age)
    { q=p++;
      m=q->age; }
    printf("%s, %d", (*q).name, (*q).age);
  }
}

```

【程序 79】

题目：字符串排序。

1. 程序分析：
2. 程序源代码：

```

main()
{
  char *str1[20], *str2[20], *str3[20];
  char swap();
  printf("please input three strings\n");
  scanf("%s", str1);
  scanf("%s", str2);
  scanf("%s", str3);
  if(strcmp(str1, str2)>0) swap(str1, str2);
  if(strcmp(str1, str3)>0) swap(str1, str3);
  if(strcmp(str2, str3)>0) swap(str2, str3);
  printf("after being sorted\n");
  printf("%s\n%s\n%s\n", str1, str2, str3);
}

char swap(p1, p2)
char *p1, *p2;
{
  char *p[20];
  strcpy(p, p1); strcpy(p1, p2); strcpy(p2, p);
}

```

【程序 80】

题目：海滩上有一堆桃子，五只猴子来分。第一只猴子把这堆桃子凭据分为五份，多了一个，

这只猴子把多的一个扔入海中，拿走了一份。第二只猴子把剩下的桃子又平均分成五份，又多了一个，它同样把多的一个扔入海中，拿走了一份，第三、第四、第五只猴子都是这样做的，问海滩上原来最少有多少个桃子？

1. 程序分析：

2. 程序源代码：

```
main()
{int i,m,j,k,count;
for(i=4;i<10000;i+=4)
{ count=0;
m=i;
for(k=0;k<5;k++)
{
j=i/4*5+1;
i=j;
if(j%4==0)
count++;
else
break;
}
i=m;
if(count==4)
{printf("%d\n",count);
break;}
}
}
```

【程序 81】

题目：809*??=800*??+9*??+1 其中??代表的两位数，8*??的结果为两位数，9*??的结果为 3 位数。求??代表的两位数，及 809*??后的结果。

1. 程序分析：

2. 程序源代码：

```
output(long b,long i)
{ printf("\n%d/%d=809*%ld+%ld",b,i,i,b%i);
}

main()
{long int a,b,i;
a=809;
for(i=10;i<100;i++)
{b=i*a+1;
if(b>=1000&&b<=10000&&8*i<100&&9*i>=100)
output(b,i); }
}
```

【程序 82】

题目：八进制转换为十进制

1. 程序分析:

2. 程序源代码:

```
main()
{ char *p, s[6]; int n;
  p=s;
  gets(p);
  n=0;
  while(*(p)!='\0')
    {n=n*8+*p-'0';
    p++;}
  printf("%d", n);
}
```

=====

【程序 83】

题目：求 0—7 所能组成的奇数个数。

1. 程序分析:

2. 程序源代码:

```
main()
{
  long sum=4, s=4;
  int j;
  for(j=2; j<=8; j++)/*j is place of number*/
    { printf("\n%d", sum);
      if(j<=2)
        s*=7;
      else
        s*=8;
      sum+=s;}
  printf("\nsum=%ld", sum);
}
```

=====

【程序 84】

题目：一个偶数总能表示为两个素数之和。

1. 程序分析:

2. 程序源代码:

```
#include "stdio.h"
#include "math.h"
main()
{ int a, b, c, d;
  scanf("%d", &a);
  for(b=3; b<=a/2; b+=2)
    { for(c=2; c<=sqrt(b); c++)
      if(b%c==0) break;
      if(c>sqrt(b))
```

```

d=a-b;
else
break;
for (c=2;c<=sqrt(d);c++)
if (d%c==0) break;
if (c>sqrt(d))
printf("%d=%d+%d\n", a, b, d);
}
}

```

【程序 85】

题目：判断一个素数能被几个 9 整除

1. 程序分析：

2. 程序源代码：

```

main()
{ long int m9=9, sum=9;
int zi, n1=1, c9=1;
scanf("%d", &zi);
while (n1!=0)
{ if (!(sum%zi))
n1=0;
else
{m9=m9*10;
sum=sum+m9;
c9++;
}
}
printf("%ld, can be divided by %d \"9\"", sum, c9);
}

```

【程序 86】

题目：两个字符串连接程序

1. 程序分析：

2. 程序源代码：

```

#include "stdio.h"
main()
{char a[]="acegikm";
char b[]="bdfhjlnpq";
char c[80], *p;
int i=0, j=0, k=0;
while (a[i]!='\0' && b[j]!='\0')
{if (a[i] { c[k]=a[i]; i++;}
else
c[k]=b[j++];

```

```

k++;
}
c[k]=' \0' ;
if(a[i]!=' \0')
p=b+j;
else
p=a+i;
strcat(c,p);
puts(c);
}

```

【程序 87】

题目：回答结果（结构体变量传递）

1. 程序分析：

2. 程序源代码：

```

#include "stdio.h"
struct student
{ int x;
char c;
} a;
main()
{a.x=3;
a.c=' a' ;
f(a);
printf("%d,%c",a.x,a.c);
}
f(struct student b)
{
b.x=20;
b.c=' y' ;
}

```

【程序 88】

题目：读取 7 个数（1—50）的整数值，每读取一个值，程序打印出该值个数的*。

1. 程序分析：

2. 程序源代码：

```

main()
{int i,a,n=1;
while(n<=7)
{ do {
scanf("%d",&a);
}while(a<1||a>50);
for(i=1;i<=a;i++)
printf("*");
}
}

```

```
printf("\n");
n++;}
getch();
}
```

【程序 89】

题目：某个公司采用公用电话传递数据，数据是四位的整数，在传递过程中是加密的，加密规则如下：每位数字都加上 5, 然后用和除以 10 的余数代替该数字，再将第一位和第四位交换，第二位和第三位交换。

1. 程序分析：

2. 程序源代码：

```
main()
{int a,i,aa[4],t;
scanf("%d",&a);
aa[0]=a%10;
aa[1]=a%100/10;
aa[2]=a%1000/100;
aa[3]=a/1000;
for(i=0;i<=3;i++)
{aa[i]+=5;
aa[i]%=10;
}
for(i=0;i<=3/2;i++)
{t=aa[i];
aa[i]=aa[3-i];
aa[3-i]=t;
}
for(i=3;i>=0;i--)
printf("%d",aa[i]);
}
```

【程序 90】

题目：专升本一题，读结果。

1. 程序分析：

2. 程序源代码：

```
#include "stdio.h"
#define M 5
main()
{int a[M]={1,2,3,4,5};
int i,j,t;
i=0;j=M-1;
while(i {t=(a+i);
*(a+i)=*(a+j);
*(a+j)=t;
```

```
i++;j--;  
}  
for(i=0;i printf("%d",*(a+i));  
}
```

【程序 91】

题目：时间函数举例 1

1. 程序分析：

2. 程序源代码：

```
#include "stdio.h"  
#include "time.h"  
void main()  
{ time_t lt; /*define a longint time variable*/  
lt=time(NULL);/*system time and date*/  
printf(ctime(<)); /*english format output*/  
printf(asctime(localtime(<)));/*transfer to tm*/  
printf(asctime(gmtime(<))); /*transfer to Greenwich time*/  
}
```

=====

【程序 92】

题目：时间函数举例 2

1. 程序分析：

2. 程序源代码：

```
/*calculate time*/  
#include "time.h"  
#include "stdio.h"  
main()  
{ time_t start,end;  
int i;  
start=time(NULL);  
for(i=0;i<3000;i++)  
{ printf("\1\1\1\1\1\1\1\1\1\1\n");}  
end=time(NULL);  
printf("\1: The different is %6.3f\n",difftime(end,start));  
}
```

=====

【程序 93】

题目：时间函数举例 3

1. 程序分析：

2. 程序源代码：

```
/*calculate time*/  
#include "time.h"  
#include "stdio.h"  
main()  
{ clock_t start,end;
```



```

int i;
double var;
start=clock();
for(i=0;i<10000;i++)
{ printf("\1\1\1\1\1\1\1\1\1\1\n");}
end=clock();
printf("\1: The different is %6.3f\n", (double)(end-start));
}

```

【程序 94】

题目：时间函数举例 4, 一个猜数游戏，判断一个人反应快慢。

1. 程序分析：

2. 程序源代码：

```

#include "time.h"
#include "stdlib.h"
#include "stdio.h"
main()
{char c;
clock_t start,end;
time_t a,b;
double var;
int i,guess;
srand(time(NULL));
printf("do you want to play it.('y' or 'n') \n");
loop:
while((c=getchar())=='y')
{
i=rand()%100;
printf("\nplease input number you guess:\n");
start=clock();
a=time(NULL);
scanf("%d",&guess);
while(guess!=i)
{if(guess>i)
{printf("please input a little smaller.\n");
scanf("%d",&guess);}
else
{printf("please input a little bigger.\n");
scanf("%d",&guess);}
}
end=clock();
b=time(NULL);
printf("\1: It took you %6.3f seconds\n", var=(double)(end-start)/18.2);
printf("\1: it took you %6.3f seconds\n\n", difftime(b,a));
}
}

```

```

if(var<15)
printf("\1\1 You are very clever! \1\1\n\n");
else if(var<25)
printf("\1\1 you are normal! \1\1\n\n");
else
printf("\1\1 you are stupid! \1\1\n\n");
printf("\1\1 Congradulations \1\1\n\n");
printf("The number you guess is %d",i);
}

printf("\ndo you want to try it again?(\\"yy\\".or.\\n\\")\n");
if((c=getch())=='y')
goto loop;
}
=====

```

【程序 95】

题目：家庭财务管理小程序

1. 程序分析：

2. 程序源代码：

```

/*money management system*/
#include "stdio.h"
#include "dos.h"
main()
{
FILE *fp;
struct date d;
float sum, chm=0.0;
int len, i, j=0;
int c;
char ch[4]="", chl[16]="", chtime[12]="", chshop[16], chmoney[8];
pp: clrscr();
sum=0.0;
gotoxy(1,1);printf(" |-----|
-----|");
gotoxy(1,2);printf(" | money management system(C1.0) 2000.03 |");
gotoxy(1,3);printf(" |-----|
-----|");
gotoxy(1,4);printf(" | -- money records -- | -- today cost list -- |");
gotoxy(1,5);printf(" |-----|
|-----|");
gotoxy(1,6);printf(" | date: ----- | |");
gotoxy(1,7);printf(" | | | |");
gotoxy(1,8);printf(" | ----- | |");
gotoxy(1,9);printf(" | thgs: ----- | |");
gotoxy(1,10);printf(" | | | |");

```

```

gotoxy(1,11);printf(" | ----- | |");
gotoxy(1,12);printf(" | cost: ----- | |");
gotoxy(1,13);printf(" | | | |");
gotoxy(1,14);printf(" | ----- | |");
gotoxy(1,15);printf(" | | |");
gotoxy(1,16);printf(" | | |");
gotoxy(1,17);printf(" | | |");
gotoxy(1,18);printf(" | | |");
gotoxy(1,19);printf(" | | |");
gotoxy(1,20);printf(" | | |");
gotoxy(1,21);printf(" | | |");
gotoxy(1,22);printf(" | | |");
gotoxy(1,23);printf(" |-----|");
gotoxy(1,24);printf(" |");

i=0;
getdate(&d);
sprintf(chtime,"%4d.%02d.%02d",d.da_year,d.da_mon,d.da_day);
for(;;)
{
gotoxy(3,24);printf(" Tab __browse cost list Esc __quit");
gotoxy(13,10);printf(" ");
gotoxy(13,13);printf(" ");
gotoxy(13,7);printf("%s",ctime);
j=18;
ch[0]=getch();
if(ch[0]==27)
break;
strcpy(chshop,"");
strcpy(chmoney,"");
if(ch[0]==9)
{
mm:i=0;
fp=fopen("home.dat","r+");
gotoxy(3,24);printf(" ");
gotoxy(6,4);printf(" list records ");
gotoxy(1,5);printf(" |-----|");
gotoxy(41,4);printf(" ");
gotoxy(41,5);printf(" |");
while(fscanf(fp,"%10s%14s%f\n",ctime,chshop,&chm)!=EOF)
{ if(i==36)
{ getch();
i=0;}
if ((i%36)<17)
{ gotoxy(4,6+i);

```

```
printf(" ");
gotoxy(4, 6+i);}
else
if((i%36)>16)
{ gotoxy(41, 4+i-17);
printf(" ");
gotoxy(42, 4+i-17);}
i++;
sum=sum+chm;
printf("%10s %-14s %6.1f\n", chtime, chshop, chm);}
gotoxy(1, 23);printf(" |-----
-----|");
gotoxy(1, 24);printf(" | |");
gotoxy(1, 25);printf(" |-----
-----|");
gotoxy(10, 24);printf("total is %8.1f$", sum);
fclose(fp);
gotoxy(49, 24);printf("press any key to.....");getch();goto pp;
}
else
{
while(ch[0]!='\r')
{ if(j<10)
{ strncat(chtime, ch, 1);
j++;}
if(ch[0]==8)
{
len=strlen(chtime)-1;
if(j>15)
{ len=len+1; j=11;}
strcpy(ch1, "");
j=j-2;
strncat(ch1, chtime, len);
strcpy(chtime, "");
strncat(chtime, ch1, len-1);
gotoxy(13, 7);printf(" ");}
gotoxy(13, 7);printf("%s", chtime);ch[0]=getch();
if(ch[0]==9)
goto mm;
if(ch[0]==27)
exit(1);
}
gotoxy(3, 24);printf(" ");
gotoxy(13, 10);
```

```

j=0;
ch[0]=getch();
while(ch[0]!='\r')
{ if (j<14)
{ strncat(chshop, ch, 1);
j++;}
if(ch[0]==8)
{ len=strlen(chshop)-1;
strcpy(ch1, "");
j=j-2;
strncat(ch1, chshop, len);
strcpy(chshop, "");
strncat(chshop, ch1, len-1);
gotoxy(13, 10);printf(" ");}
gotoxy(13, 10);printf("%s", chshop);ch[0]=getch();}
gotoxy(13, 13);
j=0;
ch[0]=getch();
while(ch[0]!='\r')
{ if (j<6)
{ strncat(chmoney, ch, 1);
j++;}
if(ch[0]==8)
{ len=strlen(chmoney)-1;
strcpy(ch1, "");
j=j-2;
strncat(ch1, chmoney, len);
strcpy(chmoney, "");
strncat(chmoney, ch1, len-1);
gotoxy(13, 13);printf(" ");}
gotoxy(13, 13);printf("%s", chmoney);ch[0]=getch();}
if((strlen(chshop)==0) || (strlen(chmoney)==0))
continue;
if((fp=fopen("home.dat", "a+"))!=NULL);
fprintf(fp, "%10s%14s%6s", chtime, chshop, chmoney);
fputc('\n', fp);
fclose(fp);
i++;
gotoxy(41, 5+i);
printf("%10s %-14s %-6s", chtime, chshop, chmoney);
}}}

```

【程序 96】

题目：计算字符串中子串出现的次数

1. 程序分析:

2. 程序源代码:

```
#include "string.h"
#include "stdio.h"
main()
{ char str1[20],str2[20],*p1,*p2;
  int sum=0;
  printf("please input two strings\n");
  scanf("%s%s",str1,str2);
  p1=str1;p2=str2;
  while(*p1!='\0')
  {
    if(*p1==*p2)
    {while(*p1==*p2&&*p2!='\0')
      {p1++;
       p2++;}
    }
    else
    p1++;
    if(*p2=='\0')
    sum++;
    p2=str2;
  }
  printf("%d",sum);
  getch();}
```

【程序 97】

题目：从键盘输入一些字符，逐个把它们送到磁盘上去，直到输入一个#为止。

1. 程序分析:

2. 程序源代码:

```
#include "stdio.h"
main()
{ FILE *fp;
  char ch,filename[10];
  scanf("%s",filename);
  if((fp=fopen(filename,"w"))==NULL)
  {printf("cannot open file\n");
   exit(0);}
  ch=getchar();
  ch=getchar();
  while(ch!='#')
  {fputc(ch,fp);putchar(ch);
   ch=getchar();
  }
```

```
fclose(fp);
}
```

【程序 98】

题目：从键盘输入一个字符串，将小写字母全部转换成大写字母，然后输出到一个磁盘文件“test”中保存。输入的字符串以！结束。

1. 程序分析：

2. 程序源代码：

```
#include "stdio.h"
main()
{FILE *fp;
char str[100], filename[10];
int i=0;
if((fp=fopen("test", "w"))==NULL)
{ printf("cannot open the file\n");
exit(0);}
printf("please input a string:\n");
gets(str);
while(str[i]!='!')
{ if(str[i]>='a' && str[i]<='z')
str[i]=str[i]-32;
fputc(str[i], fp);
i++;}
fclose(fp);
fp=fopen("test", "r");
fgets(str, strlen(str)+1, fp);
printf("%s\n", str);
fclose(fp);
}
```

【程序 99】

题目：有两个磁盘文件 A 和 B，各存放一行字母，要求把这两个文件中的信息合并（按字母顺序排列），输出到一个新文件 C 中。

1. 程序分析：

2. 程序源代码：

```
#include "stdio.h"
main()
{ FILE *fp;
int i, j, n, ni;
char c[160], t, ch;
if((fp=fopen("A", "r"))==NULL)
{printf("file A cannot be opened\n");
exit(0);}
printf("\n A contents are :\n");
```

```

for(i=0;(ch=fgetc(fp))!=EOF;i++)
{c[i]=ch;
putchar(c[i]);
}
fclose(fp);
ni=i;
if((fp=fopen("B","r"))==NULL)
{printf("file B cannot be opened\n");
exit(0);}
printf("\n B contents are :\n");
for(i=0;(ch=fgetc(fp))!=EOF;i++)
{c[i]=ch;
putchar(c[i]);
}
fclose(fp);
n=i;
for(i=0;i<n;i++)
for(j=i+1;j<n;j++)
if(c[i]>c[j])
{t=c[i];c[i]=c[j];c[j]=t;}
printf("\n C file is:\n");
fp=fopen("C","w");
for(i=0;i<n;i++)
{ putc(c[i],fp);
putchar(c[i]);
}
fclose(fp);
}

```

【程序 100】

题目：有五个学生，每个学生有 3 门课的成绩，从键盘输入以上数据（包括学生号，姓名，三门课成绩），计算出平均成绩，将原有的数据和计算出的平均分数存放在磁盘文件“stud”中。

1. 程序分析：

2. 程序源代码：

```

#include "stdio.h"
struct student
{ char num[6];
char name[8];
int score[3];
float avr;
} stu[5];
main()
{int i,j,sum;

```


三级 C 语言上机 100 题

☆题目 1

请编写一个函数 jsValue(int m,int k,int xx[]), 该函数的功能是: 将大于整数 m 且紧靠 m 的 k 个素数存入数组 xx 传回。

最后调用函数 writeDat()读取 10 组数据, 分别得出结果且把结果输出到文件 out.dat 中。

部分源程序存在文件 prog1.c 中。

例如: 若输入 17 5 则应输出: 19, 23, 29, 31, 37。

请勿改动主函数 main()和写函数 writeDat()的内容。

```
#include <conio.h>
#include <stdio.h>
void readwriteDat();

int isP(int m)
{
    int i;
    for(i=2;i<m;i++)
        if(m % i==0)return 0;
    return 1;
}

void num(int m,int k,int xx[])
{
    int s=0;
    for(m=m+1;k>0;m++)
        if(isP(m)) { xx[s++]=m; k--;}
}

main()
{
    int m,n,xx[1000];
    clrscr();
    printf("\nPlease enter two integers:");
    scanf("%d%d",&m,&n);
    num(m,n,xx);
    for(m=0;m<n;m++)
        printf("%d ",xx[m]);
    printf("\n");
    readwriteDat();
}

void readwriteDat()
{
```

```

int m,n,xx[1000],i;
FILE *rf,*wf;

rf=fopen("in.dat","r");
wf=fopen("out.dat","w");
for(i=0;i<10;i++){
    fscanf(rf,"%d %d",&m,&n);
    num(m,n,xx);
    for(m=0;m<n;m++)fprintf(wf,"%d ",xx[m]);
    fprintf(wf,"n");
}
fclose(rf);
fclose(wf);
}

```

题目 2

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat()把这些数存入数组 a 中，请考生编制一函数 jsVal()，其功能是：如果四位数各位上的数字均是 0 或 2 或 4 或 6 或 8，则统计出满足此条件的个数 cnt，并把这些四位数按从大到小的顺序存入数组 b 中。最后 main() 函数调用写函数 writeDat()把结果 cnt 以及数组 b 中符合条件的四位数输出到 OUT.DAT 文件中。

注意：部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[200]，b[200]，已定义变量：cnt

请勿改动数据文件 IN.DAT 中的任何数据、主函数 main()、读函数 readDat()和写函数 writeDat()的内容。

```

#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0 ;

void jsVal()
{ int i,j,qw,bw,sw,gw;
  for(i=0;i<MAX;i++)
  { qw=a[i]/1000; bw=a[i]/100%10;
    sw=a[i]%100/10; gw=a[i]%10;
    if(qw&&qw%2==0&&bw%2==0&&sw%2==0&&gw%2==0) b[cnt++]=a[i];
  }
  for(i=0;i<cnt-1;i++)
    for(j=i+1;j<cnt;j++)
      if(b[i]<b[j]) { qw=b[i]; b[i]=b[j]; b[j]=qw;}
}

void readDat()

```

```

{
    int i ;
    FILE *fp ;
    fp = fopen("in.dat", "r") ;
    for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
    fclose(fp) ;
}

```

```

void main()
{
    int i ;
    readDat() ;
    jsVal() ;
    printf("满足条件的数=%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
    printf("\n") ;
    writeDat() ;
}

```

```

writeDat()
{
    FILE *fp ;
    int i ;
    fp = fopen("out.dat", "w") ;
    fprintf(fp, "%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) fprintf(fp, "%d\n", b[i]) ;
    fclose(fp) ;
}

```

★题目 3

函数 ReadDat()实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 xx 中；请编制函数 StrOR()，其函数的功能是：以行为单位依次把字符串中所有小写字母 o 左边的字符串内容移到该串的右边存放，然后把小写字母 o 删除，余下的字符串内容移到已处理字符串的左边存放，之后把已处理的字符串仍按行重新存入字符串数组 xx 中。最后 main()函数调用函数 WriteDat()把结果 xx 输出到文件 OUT5.DAT 中。

原始数据文件存放的格式是：每行的宽度均小于 80 个字符，含标点符号和空格。

注意：部分源程序存放在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include <stdio.h>
#include <string.h>
#include <conio.h>

```

```
char xx[50][80];
int maxline = 0; /* 文章的总行数 */

int ReadDat(void);
void WriteDat(void);

void StrOR(void)
{int i, righto, j, s, k;
 char tem[80];
 for(i=0; i<maxline; i++)
 for(j=strlen(xx[i])-1; j>=0; j--)
 { k=0;
  memset(tem, 0, 80);
  if(xx[i][j]!='o')
   {righto=j;
    for(s=righto+1; s<strlen(xx[i]); s++)
     tem[k++]=xx[i][s];
    for(s=0; s<righto; s++)
     if(xx[i][s]!='o')
      tem[k++]=xx[i][s];
    strcpy(xx[i], tem);
   }
   else continue;
 }
}

void main()
{
 clrscr();
 if(ReadDat()) {
  printf("数据文件 IN.DAT 不能打开!\n\007");
  return;
 }
 StrOR();
 WriteDat();
}

int ReadDat(void)
{
 FILE *fp;
 int i = 0;
 char *p;
```

```

if((fp = fopen("IN.DAT", "r")) == NULL) return 1 ;
while(fgets(xx[i], 80, fp) != NULL) {
    p = strchr(xx[i], '\n') ;
    if(p) *p = 0 ;
    i++ ;
}
maxline = i ;
fclose(fp) ;
return 0 ;
}

```

```

void WriteDat(void)
{
    FILE *fp ;
    int i ;

    clrscr() ;
    fp = fopen("OUT5.DAT", "w") ;
    for(i = 0 ; i < maxline ; i++) {
        printf("%s\n", xx[i]) ;
        fprintf(fp, "%s\n", xx[i]) ;
    }
    fclose(fp) ;
}

```

★题目 4

函数 ReadDat()实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 xx 中，请编制函数 StrOL()，其函数的功能是：以行为单位对行中以空格或标点符号为分隔的所有单词进行倒排。最后把已处理的字符串（应不含标点符号）仍按行重新存入字符串数组 xx 中，最后调用函数 writeDat()把结果 xx 输出到文件 OUT6.DAT 中。

例如：原文： You He Me
 I am a student.
 结果： Me He You
 student a am I

原始数据文件存放的格式是：每行的宽度均小于 80 个字符，含标点符号和空格。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 writeDat()的内容。

```

#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <ctype.h>

```

```
char xx[50][80];
int maxline=0;/*文章的总行数*/

int ReadDat(void);
void WriteDat(void);

void StrOL(void)
{ int i,j,k,s,m,trl;
  char str[80];
  for(i=0;i<maxline;i++)
    { trl=strlen(xx[i]);
  memset(str,0,80);
  s=k=0;
  for(j=trl-1;j>=0;j--)
    { if(isalpha(xx[i][j])) k++;
      else { for(m=1;m<=k;m++)
        str[s++]=xx[i][j+m];
          k=0;
        }
      if(!isalpha(xx[i][j]))
        str[s++]=' ';
      }
  for(m=1;m<=k;m++)
    str[s++]=xx[i][j+m];
  str[s]='\0';
  strcpy(xx[i],str);  }
}

void main()
{
  clrscr();
  if(ReadDat()){
    printf("数据文件 IN.DAT 不能打开! \n\007");
    return;
  }
  StrOL();
  WriteDat();
}

int ReadDat(void)
{
  FILE *fp;
  int i=0;
  char *p;
```

```

if((fp=fopen("IN.DAT","r"))==NULL) return 1;
while(fgets(xx[i],80,fp)!=NULL){
    p=strchr(xx[i],'\n');
    if(p)*p=0;
    i++;
}
maxline=i;
fclose(fp);
return 0;
}

```

```

void WriteDat(void)
{
    FILE *fp;
    int i;

    clrscr();
    fp=fopen("OUT6.DAT","w");
    for(i=0;i<maxline;i++){
        printf("%s\n",xx[i]);
        fprintf(fp,"%s\n",xx[i]);
    }
    fclose(fp);
}

```

☆题目 5（整数排序题）

在文件 in.dat 中有 200 个正整数，且每个数均在 1000 至 9999 之间。函数 ReadDat()读取这 200 个数存放到数组 aa 中。请编制函数 jsSort()，其函数的功能是：要求按每个数的后三位的大小进行升序排列，然后取出满足此条件的前 10 个数依次存入数组 bb 中，如果后三位的数值相等，则按原先的数值进行降序排列。最后调用函数 WriteDat()把结果 bb 输出到文件 out.dat 中。

例：处理前 6012 5099 9012 7025 8088

处理后 9012 6012 7025 8088 5099

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include <stdio.h>
#include <string.h>
#include <conio.h>

```

```

int aa[200],bb[10];

```



```

void jsSort()
{
    int i,j,data;
    for(i=0;i<199;i++)
        for(j=i+1;j<200;j++)
            if(aa[i]%1000>aa[j]%1000||aa[i]%1000==aa[j]%1000&&aa[i]<aa[j])
                {data=aa[i];aa[i]=aa[j];aa[j]=data;}
    for(i=0;i<10;i++)
        bb[i]=aa[i];
}

void main()
{
    readDat();
    jsSort();
    writeDat();
}

readDat()
{
    FILE *in;
    int i;

    in=fopen("in.dat","r");
    for(i=0; i<200; i++) fscanf(in,"%d",&aa[i]);
    fclose(in);
}

writeDat()
{
    FILE *out;
    int i;

    clrscr();
    out=fopen("out.dat","w");
    for(i=0; i<10; i++){
        printf(" %d",bb[i]);
        fprintf(out,"%d\n",bb[i]);
    }
    fclose(out);
}

```

☆题目 6 正整数排序

在文件 in.dat 中有 200 个正整数，且每个数均在 1000 至 9999 之间。函数 ReadDat()读取这

200 个数存放到数组 `aa` 中。请编制函数 `jsSort()`，其函数的功能是：要求按每个数的后三位的大小进行降序排列，然后取出满足此条件的前 10 个数依次存入数组 `b` 中，如果后三位的数值相等，则按原先的数值进行升序排列。最后调用函数 `WriteDat()` 把结果 `bb` 输出到文件 `out.dat` 中。

例：处理前 9012 5099 6012 7025 8088

处理后 5099 8088 7025 6012 9012

注意：部分源程序已给出。

请勿改动主函数 `main()`、读数据函数 `ReadDat()` 和输出数据函数 `WriteDat()` 的内容。

```
#include<stdio.h>
#include<string.h>
#include<conio.h>

int aa[200],bb[10];

void jsSort()
{
    int i,j,data;
    for(i=0;i<199;i++)
        for(j=i+1;j<200;j++)
            if(aa[i]%1000<aa[j]%1000||aa[i]%1000==aa[j]%1000&&aa[i]>aa[j])
                {data=aa[i];aa[i]=aa[j];aa[j]=data;}
    for(i=0;i<10;i++)
        bb[i]=aa[i];
}

void main()
{
    readDat();
    jsSort();
    writeDat();
    system("pause");
}

readDat()
{
    FILE *in;
    int i;

    in=fopen("in.dat","r");
    for(i=0; i<200; i++) fscanf(in,"%d",&aa[i]);

    fclose(in);
}
```

```

writeDat()
{
FILE *out;
int i;
clrscr();
out=fopen("out.dat","w");
for(i=0; i<10; i++){
printf("i=%d,%d\n",i+1,bb[i]);
fprintf(out,"%d\n",bb[i]);
}
fclose(out);
}

```

☆题目 7（结构体操作题）

已知在文件 IN.DAT 中存有 100 个产品销售记录，每个产品销售记录由产品代码 dm(字符型 4 位)，产品名称 mc(字符型 10 位)，单价 dj(整型)，数量 sl(整型)，金额 je(长整型)五部分组成。其中：金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat()，其功能要求：按产品代码从大到小进行排列，若产品代码相同，则按金额从大到小进行排列，最终排列结果仍存入结构数组 sell 中，最后调用函数 WriteDat()把结果输出到文件 OUT6.DAT 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include <stdio.h>
#include <mem.h>
#include <string.h>
#include <conio.h>
#include <stdlib.h>

#define MAX 100
typedef struct{
    char dm[5]; /*产品代码*/
    char mc[11]; /*产品名称*/
    int dj; /*单价*/
    int sl; /*数量*/
    long je; /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();

void SortDat()
{int i,j;
    PRO xy;

```

```
for(i=0;i<99;i++)
for(j=i+1;j<100;j++)
    if(strcmp(sell[i].dm,sell[j].dm)<0||strcmp(sell[i].dm,sell[j].dm)==0&&sell[i].je<sell[j].je)
        {xy=sell[i]; sell [i]=sell[j]; sell[j]=xy;}
}
```

```
void main()
{
    memset(sell,0,sizeof(sell));
    ReadDat();
    SortDat();
    WriteDat();
}
```

```
void ReadDat()
{
    FILE *fp;
    char str[80],ch[11];
    int i;

    fp=fopen("IN.DAT","r");
    for(i=0;i<100;i++){
        fgets(str,80,fp);
        memcpy(sell[i].dm,str,4);
        memcpy(sell[i].mc,str+4,10);
        memcpy(ch,str+14,4);ch[4]=0;
        sell[i].dj=atoi(ch);
        memcpy(ch,str+18,5);ch[5]=0;
        sell[i].sl=atoi(ch);
        sell[i].je=(long)sell[i].dj*sell[i].sl;
    }
    fclose(fp);
}
```

```
void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("OUT6.DAT","w");
    for(i=0;i<100;i++){
        printf("%s %s %4d %5d %5d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
        fprintf(fp,"%s %s %4d %5d %5d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
    }
}
```

```
fclose(fp);
}
```

★题目 8（字符替换题）

函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS1.DAT 中。

替代关系： $f(p)=p*11 \bmod 256$ （p 是数组中某一个字符的 ASCII 值，f(p)是计算后新字符的 ASCII 值），如果计算后 f(p)值小于等于 32 或大于 130，则该字符不变，否则将 f(p)所对应的字符进行替代。（注意中间变量用无符号整型），部分源程序存在文件 prog1.c 中。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <ctype.h>

unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/

int ReadDat(void);
void WriteDat(void);

void encryptChar()
{ int i,j;
  for(i=0;i<maxline;i++)
    for(j=0;j<strlen(xx[i]);j++)
      if(xx[i][j]*11%256<=32||xx[i][j]*11%256>130) continue;
      else xx[i][j]=xx[i][j]*11%256;
}

void main()
{
  clrscr();
  if(ReadDat()){
    printf("数据文件 ENG.IN 不能打开! \n\007");
    return;
  }
  encryptChar();
  WriteDat();
}
```

```

int ReadDat(void)
{
    FILE *fp;
    int i=0;
    unsigned char *p;

    if((fp=fopen("eng.in","r"))==NULL) return 1;
    while(fgets(xx[i],80,fp)!=NULL){
        p=strchr(xx[i],'\n');
        if(p)*p=0;
        i++;
    }
    maxline=i;
    fclose(fp);
    return 0;
}

```

```

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("ps1.dat","w");
    for(i=0;i<maxline;i++){
        printf("%s\n",xx[i]);
        fprintf(fp,"%s\n",xx[i]);
    }
    fclose(fp);
}

```

★☆☆题目 9（字符串排序题）

函数 ReadDat()实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 xx 中，请编制函数 SortCharD()，其函数的功能是：以行为单位对字符按从大到小的顺序进行排序，排序后的结果仍按行重新存入字符串数组 xx 中，最后调用函数 writeDat()把结果 xx 输出到文件 OUT2.DAT 中。

例：原文：dAe,BfC.

CCbbAA

结果：fedCBA.

bbCCAA

原始数据文件存放的格式是：每行的宽度均小于 80 个字符，含标点符号和空格。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 writeDat()的内容。

```
#include <stdio.h>
```

```
#include <string.h>
#include <conio.h>

char xx[50][80];
int maxline=0;/*文章的总行数*/

int ReadDat(void);
void WriteDat(void);

void SortCharD(void)
{int i,j,k,sl;
 char ch;
 for(i=0;i<maxline;i++)
 {sl=strlen(xx[i]);
  for(j=0;j<sl-1;j++)
   for(k=j+1;k<sl;k++)
    if(xx[i][j]<xx[i][k])
     {ch=xx[i][j]; xx[i][j]=xx[i][k]; xx[i][k]=ch;}
 }
}

void main()
{
 clrscr();
 if(ReadDat()){
  printf("数据文件 IN.DAT 不能打开! \n\007");
  return;
 }
 SortCharD();
 WriteDat();
}

int ReadDat(void)
{
 FILE *fp;
 int i=0;
 char *p;

 if((fp=fopen("IN.DAT","r"))==NULL) return 1;
 while(fgets(xx[i],80,fp)!=NULL){
  p=strchr(xx[i],'\n');
  if(p)*p=0;
  i++;
 }
```

```

}
maxline=i;
fclose(fp);
return 0;
}

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("OUT2.DAT","w");
    for(i=0;i<maxline;i++){
        printf("%s\n",xx[i]);
        fprintf(fp,"%s\n",xx[i]);
    }
    fclose(fp);
}

```

★☆题目 10（替换字符题）

函数 ReadDat()实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 xx 中，请编制函数 ConvertCharA()，其函数的功能是：以行为单位把字符串中的所有小写字母改写成该字母的下一个字母，如果是字母 z，则改写成字母 a。大写字母仍为大写字母，小写字母仍为小写字母，其他字符不变。把已处理的字符串仍按行重新存入字符串数组 xx 中，最后调用函数 writeDat()把结果 xx 输出到文件 OUT1.DAT 中。

例：原文： Adb.Bcdza

abck.LLhj

结果： Aec.Bdeab

bcdl.LLik

原始数据文件存放的格式是：每行的宽度均小于 80 个字符，含标点符号和空格。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 writeDat()的内容。

```

#include <stdio.h>
#include <string.h>
#include <conio.h>

char xx[50][80];
int maxline=0;/*文章的总行数*/

int ReadDat(void);
void WriteDat(void);

void ConvertCharA(void)

```



```
{ int i,j;
  for(i=0;i<maxline;i++)
    { for(j=0;j<strlen(xx[i]);j++)
      if(xx[i][j]!='z') xx[i][j]='a';
      else if(xx[i][j]>='a'&&xx[i][j]<='y') xx[i][j]+=1;
    }
}
```

```
void main()
{
  clrscr();
  if(ReadDat()){
    printf("数据文件 IN.DAT 不能打开! \n\007");
    return;
  }
  ConvertCharA();
  WriteDat();
}
```

```
int ReadDat(void)
{
  FILE *fp;
  int i=0;
  char *p;

  if((fp=fopen("IN.DAT","r"))==NULL) return 1;
  while(fgets(xx[i],80,fp)!=NULL){
    p=strchr(xx[i],'\n');
    if(p)*p=0;
    i++;
  }
  maxline=i;
  fclose(fp);
  return 0;
}
```

```
void WriteDat(void)
{
  FILE *fp;
  int i;

  clrscr();
  fp=fopen("OUT1.DAT","w");
  for(i=0;i<maxline;i++){
```

```

    printf("%s\n",xx[i]);
    fprintf(fp,"%s\n",xx[i]);
}
fclose(fp);
}

```

☆题目 11（字符串字母移位题）

程序 prog1.c 的功能是：把 s 字符串中的所有字母改写成该字母的下一个字母，字母 z 改写成字母 a。要求大写字母仍为大写字母，小写字母仍为小写字母，其它字符不做改变。

请考生编写函数 chg(char *s)实现程序要求，最后调用函数 readwriteDAT()把结果输出到文件 bc1.out 中。

例如：s 字符串中原有的内容为：Mn.123Zxy，则调用该函数后，结果为：No.123Ayz。

注意：部分源程序存在文件 prog1.c 文件中。

请勿改动主函数 main()和输出数据函数 readwriteDAT()的内容。

```

#include <conio.h>
#include <string.h>
#include <stdio.h>
#include <ctype.h>
#define N 81
void readwriteDAT();

void chg(char *s)
{int i;
for(I=0;I<strlen(s);I++)
if(s[i]=='z'||s[i]=='Z') s[i]-=25;
else if(s[i]>='a'&&s[i]<='y'||s[i]>='A'&&s[i]<='Y') s[i]+=1;
}

main( )
{
    char a[N];
    clrscr();
    printf("Enter a string : "); gets(a);
    printf("The original string is : "); puts(a);
    chg(a);
    printf("The string after modified : ");
    puts (a);
    readwriteDAT();
}

void readwriteDAT()
{
    int i ;

```

```

char a[N] ;
FILE *rf, *wf;

rf = fopen("bc1.in", "r") ;
wf = fopen("bc1.out", "w") ;
for(i = 0 ; i < 50 ; i++) {
    fscanf(rf, "%s", a);
    chg(a);
    fprintf(wf, "%s\n", a);
}
fclose(rf);
fclose(wf);
}

```

★☆题目 12 (结构体运算题)

已知在文件 IN.DAT 中存有 100 个产品销售记录, 每个产品销售记录由产品代码 dm(字符型 4 位), 产品名称 mc(字符型 10 位), 单价 dj(整型), 数量 sl(整型), 金额 je(长整型)五部分组成。其中: 金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat(), 其功能要求:

按产品名称从小到大进行排列, 若产品名称相等, 则按金额从小到大进行排列, 最终排列结果仍存入结构数组 sell 中, 最后调用函数 WriteDat()把结果输出到文件 OUT5.DAT 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include <stdio.h>
#include <mem.h>
#include <string.h>
#include <conio.h>
#include <stdlib.h>

#define MAX 100
typedef struct{
    char dm[5]; /*产品代码*/
    char mc[11]; /*产品名称*/
    int dj; /*单价*/
    int sl; /*数量*/
    long je; /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();

void SortDat()
{int i,j;

```

```
PRO xy;
for(i=0;i<99;i++)
    for(j=i+1;j<100;j++)
        if(strcmp(sell[i].mc,sell[j].mc)>0||strcmp(sell[i].mc,sell[j].mc)==0&&sell[i].je>sell[j].je)
            {xy=sell[i];sell[i]=sell[j];sell[j]=xy;}
}
```

```
void main()
{
    memset(sell,0,sizeof(sell));
    ReadDat();
    SortDat();
    WriteDat();
}
```

```
void ReadDat()
{
    FILE *fp;
    char str[80],ch[11];
    int i;

    fp=fopen("IN.DAT","r");
    for(i=0;i<100;i++){
        fgets(str,80,fp);
        memcpy(sell[i].dm,str,4);
        memcpy(sell[i].mc,str+4,10);
        memcpy(ch,str+14,4);ch[4]=0;
        sell[i].dj=atoi(ch);
        memcpy(ch,str+18,5);ch[5]=0;
        sell[i].sl=atoi(ch);
        sell[i].je=(long)sell[i].dj*sell[i].sl;
    }
    fclose(fp);
}
```

```
void WriteDat()
{
    FILE *fp;
    int i;

    fp=fopen("OUT5.DAT","w");
    for(i=0;i<100;i++){
        printf("%s %s %4d %5d %5d\n",sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
        fprintf(fp,"%s %s %4d %5d %5d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
    }
}
```

```

}
fclose(fp);
}

```

题目 13 (结构体运算题)

已知在文件 IN.DAT 中存有 100 个产品销售记录, 每个产品销售记录由产品代码 dm(字符型 4 位), 产品名称 mc(字符型 10 位), 单价 dj(整型), 数量 sl(整型), 金额 je(长整型)五部分组成。其中: 金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat(), 其功能

要求: 按产品代码从小到大进行排列, 若产品代码相等, 则按金额从大到小进行排列, 最终排列结果仍存入结构数组 sell 中, 最后调用函数 WriteDat()把结果输出到文件 OUT9.DAT 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include <stdio.h>
#include <mem.h>
#include <string.h>
#include <conio.h>
#include <stdlib.h>

#define MAX 100
typedef struct{
    char dm[5]; /*产品代码*/
    char mc[11]; /*产品名称*/
    int dj; /*单价*/
    int sl; /*数量*/
    long je; /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();

void SortDat()
{int i,j;
  PRO xy;
  for(i=0;i<99;i++)
    for(j=i+1;j<100;j++)
      if(strcmp(sell[i].dm,sell[j].dm)>0||strcmp(sell[i].dm,sell[j].dm)==0&&sell[i].je<sell[j].je)
        {xy=sell[i];sell[i]=sell[j];sell[j]=xy;}
}

void main()
{
  memset(sell,0,sizeof(sell));

```

```

ReadDat();
SortDat();
WriteDat();
}

void ReadDat()
{
    FILE *fp;
    char str[80],ch[11];
    int i;

    fp=fopen("IN.DAT","r");
    for(i=0;i<100;i++){
        fgets(str,80,fp);
        memcpy(sell[i].dm,str,4);
        memcpy(sell[i].mc,str+4,10);
        memcpy(ch,str+14,4);ch[4]=0;
        sell[i].dj=atoi(ch);
        memcpy(ch,str+18,5);ch[5]=0;
        sell[i].sl=atoi(ch);
        sell[i].je=(long)sell[i].dj*sell[i].sl;
    }
    fclose(fp);
}

void WriteDat()
{
    FILE *fp;
    int i;

    fp=fopen("OUT9.DAT","w");
    for(i=0;i<100;i++){
        printf("%s %s %4d %5d %5d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
        fprintf(fp,"%s %s %4d %5d %5d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
    }
    fclose(fp);
}

```

★☆☆题目 14（整数统计排序题）

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat()把这些数存入数组 a 中，请考生编制一函数 jsVal()，其功能是：依次从数组 a 中取出一个四位数，如果该四位数连续大于该四位数以后的五个数且该数是奇数（该四位数以后不满五个数，则不统计），则统计出满足此条件的个数 cnt 并把这些四位数按从小到大的顺序存入数组 b 中，最后调用写

函数 writeDat()把结果 cnt 以及数组 b 中符合条件的四位数输出到 OUT.DAT 文件中。

注意：部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[200]，b[200]，已定义变量：cnt

请勿改动数据文件 IN.DAT 中的任何数据、主函数 main()、读函数 readDat()和写函数 writeDat()的内容。

```
#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0 ;

void jsVal()
{int i,j,flag;
 for(i=0;i<MAX-5;i++)
  {for(j=i+1;j<=i+5;j++)
   if(a[i]>a[j]&& a[i]%2) flag=1;
   else {flag=0;break;}
   if(flag==1) b[cnt++]=a[i];
  }
 for(i=0;i<cnt-1;i++)
  for(j=i+1;j<cnt;j++)
   if(b[i]>b[j]) {flag=b[i];b[i]=b[j];b[j]=flag;}
}

void readDat()
{
 int i ;
 FILE *fp ;
 fp = fopen("in.dat", "r") ;
 for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
 fclose(fp) ;
}

void main()
{
 int i ;
 readDat() ;
 jsVal() ;
 printf("满足条件的数=%d\n", cnt) ;
 for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
 printf("\n") ;
 writeDat() ;
}

writeDat()
{
```

```

FILE *fp;
int i;
fp = fopen("out.dat", "w");
fprintf(fp, "%d\n", cnt);
for(i = 0; i < cnt; i++) fprintf(fp, "%d\n", b[i]);
fclose(fp);
}

```

★题目 15 (整数各位数字运算排序题)

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat()把这些数存入数组 a 中，请考生编制一函数 jsVal()，其功能是：若一个四位数的千位数字上的值小于等于百位数字上的值，百位数字上的值小于等于十位数字上的值，以及十位数字上的值小于等于个位数字上的值，并且原四位数是偶数，则统计出满足此条件的个数 cnt 并把这些四位数按从小到大的顺序存入数组 b 中，最后调用写函数 writeDat()把结果 cnt 以及数组 b 中符合条件的四位数输出到 OUT.DAT 文件中。

注意：部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[200]，b[200]，已定义变量：cnt

请勿改动数据文件 IN.DAT 中的任何数据、主函数 main()、读函数 readDat()和写函数 writeDat()的内容。

```

#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt=0;

void jsVal()
{int i,j;
 int qw,bw,sw,gw;
 for(i=0;i<MAX;i++)
 {qw=a[i]/1000;   bw=a[i]%1000/100;
  sw=a[i]%100/10;  gw=a[i]%10;
  if((qw<=bw)&&(bw<=sw)&&(sw<=gw)&&(a[i]%2==0))  b[cnt++]=a[i];
 }
 for(i=0;i<cnt-1;i++)
 for(j=i+1;j<cnt;j++)
 if (b[i]>b[j]) {qw=b[i];b[i]=b[j];b[j]=qw;}
}

void readDat()
{
 int i;
 FILE *fp;
 fp = fopen("in.dat", "r");

```



```

    for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
    fclose(fp) ;
}

```

```

void main()
{
    int i ;
    readDat() ;
    jsVal() ;
    printf("满足条件的数=%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
    printf("\n") ;
    writeDat() ;
}

```

```

writeDat()
{
    FILE *fp ;
    int i ;
    fp = fopen("out.dat", "w") ;
    fprintf(fp, "%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) fprintf(fp, "%d\n", b[i]) ;
    fclose(fp) ;
}

```

★☆题目 16 (字符替换题)

函数 ReadDat()实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 xx 中，请编制函数 StrCharJR()，其函数的功能是：以行为单位把字符串中所有字符的 ASCII 值右移 4 位，然后把右移后的字符 ASCII 值再加上原字符的 ASCII 值，得到新的字符仍存入原字符串对应的位置上。最后把已处理的字符串仍按行重新存入字符串数组 xx 中，最后调用函数 writeDat()把结果 xx 输出到文件 OUT8.DAT 中。

原始数据文件存放的格式是：每行的宽度均小于 80 个字符，含标点符号和空格。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 writeDat()的内容。

```

#include <stdio.h>
#include <string.h>
#include <conio.h>

```

```

char xx[50][80];
int maxline=0;/*文章的总行数*/

```

```

int ReadDat(void);
void WriteDat(void);

```

```
void StrCharJR()
{int i,j;
 for(i=0;i<maxline;i++)
  for(j=0;j<strlen(xx[i]);j++)
   xx[i][j]+=(xx[i][j]>>4);
}

void main()
{
 clrscr();
 if(ReadDat()){
  printf("数据文件 IN.DAT 不能打开! \n\007");
  return;
 }
 StrCharJR();
 WriteDat();
}

int ReadDat(void)
{
 FILE *fp;
 int i=0;
 char *p;

 if((fp=fopen("IN.DAT","r"))==NULL) return 1;
 while(fgets(xx[i],80,fp)!=NULL){
  p=strchr(xx[i],'\n');
  if(p)*p=0;
  i++;
 }
 maxline=i;
 fclose(fp);
 return 0;
}

void WriteDat(void)
{
 FILE *fp;
 int i;
 clrscr();
 fp=fopen("OUT8.DAT","w");
 for(i=0;i<maxline;i++){
  printf("%s\n",xx[i]);
 }
```

```

    fprintf(fp,"%s\n",xx[i]);
}
fclose(fp);
}

```

☆题目 17 （亲朋字符替换题）

函数 READDAT（）实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 XX 中;请编制函数 CHA(),其函数功能是:以行为单位把字符串中的第一个字符的 ASCII 值加第二个字符的 ASCII 值,得到第一个亲朋字符,第二个字符的 ASCII 值加第三个字符的 ASCII 值,得到第二个新字符,依此类推一直处理到最后第二个字符,最后一个字符的 ASCII 值加原第一个字符的 ASCII 值,得到最后一个新的字符,得到的新字符分别存放在原字符串对应的位置上。最后把已处理的字符串逆转后按行重新存入字符串数组 XX 中,最后调用函数 WRITEDAT()把结果 XX 输出到文件 OUT9.DAT 中.原始数据文件存放的格式是:每行的宽度均小于 80 个字符,含标点符号和空格。

注意：部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 writeDat()的内容。

```

#include <stdio.h>
#include <string.h>
#include <conio.h>

char xx[50][80];
int maxline = 0; /* 文章的总行数 */

int ReadDat(void);
void WriteDat(void);

void ChA(void)
{ int i,j;
  char ch;
  for(i=0;i<maxline;i++)
  { ch=xx[i][0];
    for(j=0;j<strlen(xx[i])-1;j++)
      xx[i][j]+=xx[i][j+1];
    xx[i][strlen(xx[i])-1]+=ch;
    strev(xx[i]);
  }
}

void main()
{
  clrscr();
  if(ReadDat()) {
    printf("数据文件 IN.DAT 不能打开!\n\007");

```

```

    return ;
}
ChA() ;
WriteDat() ;
}

int ReadDat(void)
{
    FILE *fp ;
    int i = 0 ;
    char *p ;

    if((fp = fopen("IN.DAT", "r")) == NULL) return 1 ;
    while(fgets(xx[i], 80, fp) != NULL) {
        p = strchr(xx[i], '\n') ;
        if(p) *p = 0 ;
        i++ ;
    }
    maxline = i ;
    fclose(fp) ;
    return 0 ;
}

void WriteDat(void)
{
    FILE *fp ;
    int i ;

    clrscr() ;
    fp = fopen("OUT9.DAT", "w") ;
    for(i = 0 ; i < maxline ; i++) {
        printf("%s\n", xx[i]) ;
        fprintf(fp, "%s\n", xx[i]) ;
    }
    fclose(fp) ;
}

```

☆题目 18

函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS10.DAT 中。

替代关系： $f(p)=p*11 \bmod 256$ (p 是数组中某一个字符的 ASCII 值， $f(p)$ 是计算后新字符的 ASCII 值)，如果原字符的 ASCII 值是偶数或计算后 $f(p)$ 值小于等于 32，则该字符不变，

否则将 $f(p)$ 所对应的字符进行替代。

部分源程序存在文件 `prog1.c` 中。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 `main()`、读数据函数 `ReadDat()` 和输出数据函数 `WriteDat()` 的内容。

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <ctype.h>

unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/

int ReadDat(void);
void WriteDat(void);

void encryptChar()
{ int i,j;
  for(i=0;i<maxline;i++)
    for(j=0;j<strlen(xx[i]);j++)
      if(xx[i][j]*11%256<=32||xx[i][j]%2==0) continue;
      else xx[i][j]=xx[i][j]*11%256;
}

void main()
{
  clrscr();
  if(ReadDat()){
    printf("数据文件 ENG.IN 不能打开! \n\007");
    return;
  }
  encryptChar();
  WriteDat();
}

int ReadDat(void)
{
  FILE *fp;
  int i=0;
  unsigned char *p;

  if((fp=fopen("eng.in","r"))==NULL) return 1;
  while(fgets(xx[i],80,fp)!=NULL){
    p=strchr(xx[i],'\n');
    if(p)*p=0;
  }
}
```

```

        i++;
    }
    maxline=i;
    fclose(fp);
    return 0;
}

void WriteDat(void)
{
    FILE *fp;
    int i;
    fp=fopen("ps10.dat","w");
    for(i=0;i<maxline;i++){
        printf("%s\n",xx[i]);
        fprintf(fp,"%s\n",xx[i]);
    }
    fclose(fp);
}

```

☆题目 19 (单词个数统计题)

编写一个函数 findStr(char *str,char *substr)，该函数统计一个长度为 2 的子字符串在另一个字符串中出现的次数。例如，假定输入的字符串为"asd asasdfg asd as zx67 asd mklo"，子字符串为"as"，函数返回值是 6。

函数 ReadWrite()实现从文件 in.dat 中读取两个字符串，并调用函数 findStr()，最后把结果输出到文件 out.dat 中。

注意：部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()和其它函数中的任何内容，仅在函数 findStr()的花括号中填入你编写的若干语句。

```

#include <stdio.h>
#include <string.h>
#include <conio.h>

```

```

int findStr(char *str,char *substr)
{ int i,j,len1,len2,cnt=0,flag;
  len1=strlen(str);
  len2=strlen(substr);
  for(i=0;i<len1;i++)
  { for(j=0;j<len2;j++)
    { if(str[i+j]==substr[j])    flag=1;
      else {flag=0;break;}
    }
    if(flag==1)  cnt++;
  }
  return cnt;
}

```

```

}

main()
{
    char str[81], substr[3];
    int n;

    clrscr();
    printf("输入原字符串:");
    gets(str);
    printf("输入子字符串:");
    gets(substr);
    puts(str);
    puts(substr);
    n=findStr(str, substr);
    printf("n=%d\n", n);
    ReadWrite();
}

ReadWrite()
{
    char str[81], substr[3], ch;
    int n, len, i = 0;
    FILE *rf, *wf;

    rf = fopen("in.dat", "r");
    wf = fopen("out.dat", "w");
    while(i < 25) {
        fgets(str, 80, rf);
        fgets(substr, 10, rf);
        len = strlen(substr) - 1;
        ch = substr[len];
        if(ch == '\n' || ch == 0x1a) substr[len] = 0;
        n=findStr(str, substr);
        fprintf(wf, "%d\n", n);
        i++;
    }
    fclose(rf);
    fclose(wf);
}

```

★题目 20 (Fibonacci 数列题)

编写函数 jsValue，它的功能是：求 Fibonacci 数列中大于 t 的最小的一个数，结果由函数返

回。其中 Fibonacci 数列 $F(n)$ 的定义为：

$F(0)=0, F(1)=1$

$F(n)=F(n-1)+F(n-2)$

最后调用函数 `writeDat()` 读取 50 个数据 `t`, 分别得出结果且把结果输出到文件 `out.dat` 中。

例如：当 `t=1000` 时，函数值为：1597。

部分源程序存在文件 `prog1.c` 中。

请勿改动主函数 `main()` 和写函数 `writeDat()` 的内容。

```
#include <stdio.h>
```

```
int jsValue(int t)
```

```
{ int f0=0,f1=1,fn;
```

```
    fn=f0+f1;
```

```
    while(fn<=t)
```

```
    { f0=f1;
```

```
      f1=fn;
```

```
      fn=f0+f1;
```

```
    }
```

```
    return fn;
```

```
}
```

```
main()
```

```
{
```

```
    int n;
```

```
    n=1000;
```

```
    printf("n=%d,f=%d\n",n,jsValue(n));
```

```
    writeDat();
```

```
}
```

```
writeDat()
```

```
{
```

```
    FILE *in,*out;
```

```
    int i,n,s;
```

```
    in=fopen("in.dat","r");
```

```
    out=fopen("out.dat","w");
```

```
    for(i=0;i<50;i++){
```

```
        fscanf(in,"%d",&n);
```

```
        s=jsValue(n);
```

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

```
        fprintf(out,"%d\n",s);
```

```
    }
```

```
    fclose(in);
```

```
    fclose(out);
```

```
}
```

☆题目 21 (迭代方法求方程题)

下列程序 prog1.c 的功能是：利用以下所示的简单迭代方法求方程： $\cos(x)-x=0$ 的一个实根。

$$X_{n+1}=\cos(X_n)$$

迭代步骤如下：

- (1)取 X_1 初值为 0.0;
- (2) $X_0=X_1$, 把 X_1 的值赋给 X_0 ;
- (3) $X_1=\cos(X_0)$, 求出一个新的 X_1 ;
- (4)若 X_0-X_1 的绝对值小于 0.000001, 执行步骤(5), 否则执行步骤(2);
- (5)所求 X_1 就是方程 $\cos(X)-X=0$ 的一个实根, 作为函数值返回。

请编写函数 countValue()实现程序的要求, 最后调用函数 writeDat()把结果输出到文件 OUT17.DAT 中。

部分源程序已给出。

请勿改动主函数 main()和输出数据函数 writeDat()的内容。

```
#include <conio.h>
#include <math.h>
#include <stdio.h>

float countValue()
{ float x0,x1;
  x1=0.0;
  do{ x0=x1;
    x1=cos(x0);
  }while(fabs(x0-x1)>=0.000001);
  return x1;
}

main()
{
  clrscr();
  printf("实根=%f\n",countValue());
  printf("%.f\n",cos(countValue())-countValue());
  writeDat();
}

writeDat()
{
  FILE *wf;

  wf=fopen("OUT17.DAT","w");
  fprintf(wf,"%f\n",countValue());
  fclose(wf);
}
```

注：本题在无忧或捷成模拟盘中，需定义 `double x0,x1`;才能通过，据实际上机抽到该题的考生反映，实际上机需按上面解法的定义方能通过，特此说明，供参考。

★题目 22（平方根问题）

请编写函数 `countValue()`，它的功能是：求 `n` 以内（不包括 `n`）同时能被 3 与 7 整除的所有自然数之和的平方根 `s`，并作为函数值返回，最后结果 `s` 输出到文件 `out.dat` 中。

例如若 `n` 为 1000 时，函数值应为：s=153.909064。

部分源程序存在文件 `prog1.c` 中。

请勿改动主函数 `main()` 和输入输出数据函数 `progReadWrite()` 的内容。

```
#include <conio.h>
#include <math.h>
#include <stdio.h>

double countValue(int n)
{ int i;
  double s=0.0;
  for(i=1;i<n;i++)
    if(i%21==0) s+=i;
  return sqrt(s);
}

main()
{
  clrscr();
  printf("自然数之和的平方根=%f\n",countValue(1000));
  progReadWrite();
}

progReadWrite()
{
  FILE *fp,*wf;
  int i,n;
  float s;

  fp=fopen("in.dat","r");
  if(fp==NULL){
    printf("数据文件 in.dat 不存在!");
    return;
  }
  wf=fopen("out.dat","w");
  for(i=0;i<10;i++){
```

```

        fscanf(fp,"%d\n",&n);
        s=countValue(n);
        fprintf(wf,"%f\n",s);
    }
    fclose(fp);
    fclose(wf);
}

```

题目 23 (实数运算题)

已知在文件 in.dat 中存有 N 个 (个数<200) 实数, 函数 readdat() 读取这 N 个实数并存入数组 xx 中。请编制函数 calvalue(), 其功能要求:

- 1、求出这 N 个实数的平均值 aver;
- 2、分别求出这 N 个实数的整数部分之和 sumint 以及小数部分之和 sumdec, 最后调用函数 writedat() 把所求的结果输出到文件 out.dat 中。

注意: 部分源程序已给出。

请勿改动主函数 main()、读数据函数 readdat() 和输出数据函数 writedat() 的内容。

```

#include <stdio.h>
#include <conio.h>
#define MAXNUM 200

float xx[MAXNUM];
int N= 0; /* 文件 IN.DAT 中共有多少个实数 */
double aver=0.0; /*平均值*/
double sumint=0.0; /*整数部分之和*/
double sumdec=0.0; /*小数部分之和*/

int ReadDat(void) ;
void WriteDat(void) ;

void CalValue(void)
{int i;
 for (i=0;i<N;i++)
 {sumint+=(long)xx[i];
  sumdec+=xx[i]-(long)xx[i];
  aver+=xx[i];
 }
 aver/=N;
}

void main()
{ int i ;

```

```

clrscr() ;
for(i = 0 ; i < MAXNUM ; i++) xx[i] = 0 ;
if(ReadDat()) {
    printf("数据文件 IN.DAT 不能打开!\007\n") ;
    return ;
}
Calvalue() ;
printf("文件 IN.DAT 中共有实数=%d 个\n", N) ;
printf("平均值=%.2lf\n", aver) ;
printf("整数部分之和=%.2lf\n", sumint) ;
printf("小数部分之和=%.2lf\n", sumdec) ;
WriteDat() ;
system("pause");
}

int ReadDat(void)
{
    FILE *fp ;
    int i = 0 ;

    if((fp = fopen("in.dat", "r")) == NULL) return 1 ;
    while(!feof(fp)) {
        fscanf(fp, "%d,", &xx[i++]) ;
    }
    fclose(fp) ;
    return 0 ;
}

void WriteDat(void)
{
    FILE *fp ;

    fp = fopen("OUT.DAT", "w") ;
    fprintf(fp, "%.2lf\n%.2lf\n%.2lf\n", aver, sumint, sumdec) ;
    fclose(fp) ;
}

```

题目 24 (完全平方数问题)

下列程序 prog1.c 的功能是：在三位整数（100 至 999）中寻找符合条件的整数并依次从小到大存入数组中；它既是完全平方数，又是两位数字相同，例如 144、676 等。

请编制函数实现此功能，满足该条件的整数的个数通过所编制的函数返回。

最后调用函数 writeDat()把结果输出到文件 out.dat 中。

请勿改动主函数 main()和写函数 writeDat()的内容。

```

#include <stdio.h>

int jsValue(int bb[])
{int i,j,k=0;
  int hun,ten,data;
  for(i=10;i<32;i++)
    {j=i*1;
    hum=j/100;
    ten=j%100/10;
    data=j%10;
    if(hum==ten||hum==data||ten==data)
    bb[k++]=j;}
  return k;
}

main()
{
  int b[20],num;
  num=jsValue(b);
  writeDat(num,b);
}

writeDat(int num,int b[])
{
  FILE *out;
  int i;

  out=fopen("out.dat","w");
  printf("%d\n",num);
  fprintf(out,"%d\n",num);
  for(i=0;i<num;i++){printf("%d\n",b[i]);fprintf(out,"%d\n",b[i]);}
  fclose(out);
}

```

★题目 25 （回文数问题）

下列程序的功能是：寻找并输出 11 至 999 之间的数 m，它满足 m,m2 和 m3 均为回文数。所谓回文数是指其各位数字左右对称的整数，例如 121，676，94249 等。满足上述条件的数如 m=11,m2=121,m3=1331 皆为回文数。请编制函数 int svalue(long m)实现此功能，如果是回文数，则函数返回 1，反之则返回 0。最后把结果输出到文件 out.dat 中。

注意：部分源程序已给出。

请勿改动主函数 main() 的内容。

```

#include <stdio.h>
int jsValue(long n)
{int i, strl, half;
 char xy[20];
 ltoa(n, xy, 10); /*注意这里不能使用 itoa()函数, 因为 n 是 long 型的*/
 strl = strlen(xy);
 half = strl/2;
 for(i=0; i<half; i++)
  if(xy[i] != xy[--strl]) break;
  if(i >= half) return 1;
  else return 0;
}

main()
{long m;
 FILE *out;
 out = fopen("out.dat", "w");
 for(m=11; m<1000; m++)
  { if(jsValue(m)&&jsValue(m*m)&&jsValue(m*m*m))
   { printf("m=%4ld, m*m=%6ld, m*m*m=%8ld \n", m, m*m, m*m*m);
     fprintf(out, "m=%4ld, m*m=%6ld, m*m*m=%8ld \n", m, m*m, m*m*m);
   }
 }
 fclose(out);
 system("pause");
}

```

或者下面的解法:

```

int jsValue(long n)
{long int s=0, k;
 k=n;
 while(k)
  { s=s*10+k%10;
    k/=10;
  }
 if(s==n) return 1;
 if(s!=n) return 0;
}

```

输出结果为:

```

m=  11, m*m=  121, m*m*m=  1331
m= 101, m*m= 10201, m*m*m= 1030301
m= 111, m*m= 12321, m*m*m= 1367631

```

★☆题目 26（整数统计运算题）

已知在文件 IN.DAT 中存有若干个（个数<200）四位数字的正整数，函数 ReadDat()读取这若干个正整数并存入数组 xx 中。请编制函数 CalValue()，其功能要求：1、求出这文件中共有多少个正整数 totNum；2、求这些数右移 1 位后，产生的新数是偶数的数的个数 totCnt，以及满足此条件的这些数(右移前的值)的算术平均值 totPjz，最后调用函数 writeDat()把所求的结果输出到文件 OUT.DAT 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读函数 ReadDat()和写函数 writeDat()的内容。

```
#include <stdio.h>
#include <conio.h>
#define MAXNUM 200

int xx[MAXNUM];
int totNum=0; /*文件 IN.DAT 中共有多少个正整数*/
int totCnt=0; /*符合条件的正整数的个数*/
double totPjz=0.0; /*平均值*/

int ReadDat(void);
void writeDat(void);

void CalValue(void)
{int i,data;
for(i=0;i<MAXNUM;i++)
    if(xx[i]>0)
    { totNum++;
        data=xx[i]>>1;
        if(data%2==0){totCnt++;totPjz+=xx[i];}
    }
    if(totCnt==0) totPjz=0;
    else totPjz/=totCnt;
}

void main()
{
    int i;
    clrscr();
    for(i=0;i<MAXNUM;i++)xx[i]=0;
    if(ReadDat()){
        printf("数据文件 IN.DAT 不能打开!\007\n");
        return;
    }
    CalValue();
```

```

printf("文件 IN.DAT 中共有正整数=%d 个\n",totNum);
printf("符合条件的正整数的个数=%d 个\n",totCnt);
printf("平均值=%.2f\n",totPjz);
writeDat();
}

```

```

int ReadDat(void)

```

```

{
    FILE *fp;
    int i=0;

    if((fp=fopen("IN.DAT","r"))==NULL) return 1;
    while(!feof(fp)){
        fscanf(fp,"%d",&xx[i++]);
    }
    fclose(fp);
    return 0;
}

```

```

void writeDat(void)

```

```

{
    FILE *fp;
    fp=fopen("OUT.DAT","w");
    fprintf(fp,"%d\n%d\n%.2f\n",totNum,totCnt,totPjz);
    fclose(fp);
}

```

题目 27（整数各位数字运算题）

已知数据文件 in.dat 中存有 300 个四位数，并已调用读函数 ReadDat()把这些数存入数组 a 中，请编制一函数 jsValue()，其功能是：求出千位数上的数减百位数上的数减十位数上的数减个位数上的数大于零的个数 cnt，再把所有满足此条件的四位数依次存入数组 b 中，然后对数组 b 的四位数按从小到大的顺序进行排序，最后调用写函数 writeDat()把结果输出到 out.dat 文件。

例如：9123，9-1-2-3>0，则该数满足条件存入数组 b 中，且个数 cnt=cnt+1。

9812，9-8-1-2>0，则该数不满足条件忽略。

部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[300]，b[300]，已定义变量：cnt

请勿改动主函数 main()、读函数 ReadDat()和写函数 writeDat()的内容。

```

#include <stdio.h>
int a[300],b[300],cnt=0;

```

```

jsValue()

```

```

{ int i,j,qw,bw,sw,gw;

```



```

for(i=0;i<300;i++)
{ qw=a[i]/1000;   bw=a[i]/100%10;
  sw=a[i]%100/10; gw=a[i]%10;
  if(qw-bw-sw-gw>0) b[cnt++]=a[i];
}
for(i=0;i<cnt-1;i++)
for(j=i+1;j<cnt;j++)
    if( b[i]>b[j]) { qw=b[i]; b[i]=b[j]; b[j]=qw;}
}

main()
{
    int i;

    readDat();
    jsValue();
    writeDat();
    printf("cnt=%d\n",cnt);
    for(i=0;i<cnt;i++) printf("b[%d]=%d\n",i,b[i]);
}

readDat()
{
    FILE *fp;
    int i;
    fp=fopen("in.dat","r");
    for(i=0;i<300;i++)fscanf(fp,"%d",&a[i]);
    fclose(fp);
}

writeDat()
{
    FILE *fp;
    int i;
    fp=fopen("out.dat","w");
    fprintf(fp,"%d\n",cnt);
    for(i=0;i<cnt;i++)fprintf(fp,"%d\n",b[i]);
    fclose(fp);
}

```

★☆☆题目 28（整数各位打散组合运算题）

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat()把这些数存入数组 a

中，请考生编制一函数 `jsVal()`，其功能是：把千位数字和十位数字重新组成一个新的十位数 `ab`（新十位数的十位数字是原四位数的千位数字，新十位数的个位数字是原四位数的十位数字），以及把个位数字和百位数字组成另一个新的十位数 `cd`（新十位数的十位数字是原四位数的个位数字，新十位数的个位数字是原四位数的百位数字），如果新组成的两个十位数 `ab-cd` ≥ 0 且 `ab-cd` ≤ 10 且两个数均是奇数，同时两个新数的十位数字均不为零，则将满足此条件的四位数按从大到小的顺序存入数组 `b` 中，并要计算满足上述条件的四位数的个数 `cnt`。最后 `main()` 函数调用写函数 `writeDat()` 把结果 `cnt` 以及数组 `b` 中符合条件的四位数输出到 `OUT.DAT` 文件中。

注意：部分源程序存在文件 `prog1.c` 中。

程序中已定义数组：`a[200]`，`b[200]`，已定义变量：`cnt`

请勿改动数据文件 `IN.DAT` 中的任何数据、主函数 `main()`、读函数 `readDat()` 和写函数 `writeDat()` 的内容。

```
#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0 ;

void jsVal()
{int i,j,thou,hun,ten,data,ab,cd;
 for(i=0;i<200;i++)
 {thou=a[i]/1000; hun=a[i]%1000/100;
 ten=a[i]%100/10; data=a[i]%10;
 ab=10*thou+ten; cd=10*data+hun;
 if((ab-cd)>=0&&(ab-cd)<=10&&ab%2==1&&cd%2==1&&ab>=10&&cd>=10)
 b[cnt++]=a[i];
 }
 for(i=0;i<cnt-1;i++)
 for(j=i+1;j<cnt;j++)
 if(b[i]<b[j]) {thou=b[i]; b[i]=b[j]; b[j]=thou;}
}

void readDat()
{
 int i ;
 FILE *fp ;
 fp = fopen("in.dat", "r") ;
 for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
 fclose(fp) ;
}

void main()
{
 int i ;
 readDat() ;
 jsVal() ;
}
```

```

printf("满足条件的数=%d\n", cnt);
for(i = 0; i < cnt; i++) printf("%d ", b[i]);
printf("\n");
writeDat();
}

writeDat()
{
    FILE *fp;
    int i;
    fp = fopen("out.dat", "w");
    fprintf(fp, "%d\n", cnt);
    for(i = 0; i < cnt; i++) fprintf(fp, "%d\n", b[i]);
    fclose(fp);
}

```

☆题目 29 (整数各位打散组合运算题)

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat() 把这些数存入数组 a 中，请考生编制一函数 jsVal()，其功能是：把千位数字和十位数字重新组成一个新的十位数 ab (新十位数的十位数字是原四位数的千位数字，新十位数的个位数字是原四位数的十位数字)，以及把个位数字和百位数字组成另一个新的十位数 cd (新十位数的十位数字是原四位数的个位数字，新十位数的个位数字是原四位数的百位数字)，如果新组成的两个十位数 $ab - cd \geq 10$ 且 $ab - cd \leq 20$ 且两个数均是偶数，同时两个新数的十位数字均不为零，则将满足此条件的四位数按从大到小的顺序存入数组 b 中，并要计算满足上述条件的四位数的个数 cnt。

最后 main() 函数调用写函数 writeDat() 把结果 cnt 以及数组 b 中符合条件的四位数输出到 OUT.DAT 文件中。

注意：部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[200]，b[200]，已定义变量：cnt

请勿改动数据文件 IN.DAT 中的任何数据、主函数 main()、读函数 readDat() 和写函数 writeDat() 的内容。

```

#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0;

void jsVal()
{
    int i, j, qw, bw, sw, gw, ab, cd;
    for(i=0; i<MAX; i++)
    {
        qw=a[i]/1000;    bw=a[i]/100%10;
        sw=a[i]%100/10; gw=a[i]%10;
        ab=qw*10+sw;    cd=gw*10+bw;
        if(ab-cd>=10&&ab-cd<=20&&ab%2==0&&cd%2==0&&qw!=0&&gw!=0)
            b[cnt++]=a[i];
    }
}

```

```

    }
    for(i=0;i<cnt-1;i++)
        for(j=i+1;j<cnt;j++)
            if(b[i]<b[j]) {qw=b[i];b[i]=b[j];b[j]=qw;}
}
void readDat()
{
    int i ;
    FILE *fp ;
    fp = fopen("in.dat", "r") ;
    for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
    fclose(fp) ;
}

void main()
{
    int i ;
    readDat() ;
    jsVal() ;
    printf("满足条件的数=%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
    printf("\n") ;
    writeDat() ;
}

writeDat()
{
    FILE *fp ;
    int i ;
    fp = fopen("out.dat", "w") ;
    fprintf(fp, "%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) fprintf(fp, "%d\n", b[i]) ;
    fclose(fp) ;
}

```

☆题目 30（整数统计排序题）

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat() 把这些数存入数组 a 中，请考生编制一函数 jsVal()，其功能是：依次从数组 a 中取出一个四位数，如果该四位数连续小于该四位数以后的五个数且该数是偶数（该四位数以后不满五个数，则不统计），则统计出满足此条件的个数 cnt 并把这些四位数按从小到大的顺序存入数组 b 中，最后调用写函数 writeDat() 把结果 cnt 以及数组 b 中符合条件的四位数输出到 OUT.DAT 文件中。

注意：部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[200]，b[200]，已定义变量：cnt

请勿改动数据文件 IN.DAT 中的任何数据、主函数 main()、读函数 readDat()和写函数 writeDat()的内容。

```
#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0 ;

void jsVal()
{int i,j,flag;
 for(i=0;i<MAX-5;i++)
 {for(j=i+1;j<=i+5;j++)
  if(a[i]<a[j]&& a[i]%2==0) flag=1;
  else { flag=0; break;}
  if(flag==1) b[cnt++]=a[i];
 }
 for(i=0;i<cnt-1;i++)
 for(j=i+1;j<cnt;j++)
  if(b[i]>b[j]) {flag=b[i];b[i]=b[j];b[j]=flag;}
}

void readDat()
{
 int i ;
 FILE *fp ;
 fp = fopen("in.dat", "r") ;
 for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
 fclose(fp) ;
}

void main()
{
 int i ;
 readDat() ;
 jsVal() ;
 printf("满足条件的数=%d\n", cnt) ;
 for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
 printf("\n") ;
 writeDat() ;
}

writeDat()
{
 FILE *fp ;
```

```

int i ;
fp = fopen("out.dat", "w") ;
fprintf(fp, "%d\n", cnt);
for(i = 0 ; i < cnt ; i++) fprintf(fp, "%d\n", b[i]) ;
fclose(fp) ;
}

```

★☆题目 31（结构体运算题）

已知在文件 IN.DAT 中存有 100 个产品销售记录，每个产品销售记录由产品代码 dm(字符型 4 位)，产品名称 mc(字符型 10 位)，单价 dj(整型)，数量 sl(整型)，金额 je(长整型)五部分组成。其中：金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat()，其功能要求：按金额从大到小进行排列，若金额相同，则按产品代码从大到小进行排列，最终排列结果仍存入结构数组 sell 中，最后调用函数 WriteDat() 把结果输出到文件 OUT4.DAT 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include <stdio.h>
#include <mem.h>
#include <string.h>
#include <conio.h>
#include <stdlib.h>

#define MAX 100
typedef struct{
    char dm[5]; /*产品代码*/
    char mc[11]; /*产品名称*/
    int dj; /*单价*/
    int sl; /*数量*/
    long je; /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();

void SortDat()
{int i,j;
  PRO xy;
  for(i=0;i<99;i++)
    for(j=i+1;j<100;j++)
      if(sell[i].je<sell[j].je||sell[i].je==sell[j].je&&strcmp(sell[i].dm,sell[j].dm)<0)
        {xy=sell[i];sell[i]=sell[j];sell[j]=xy;}
}

```

```
void main()
{
    memset(sell,0,sizeof(sell));
    ReadDat();
    SortDat();
    WriteDat();
}

void ReadDat()
{
    FILE *fp;
    char str[80],ch[11];
    int i;

    fp=fopen("IN.DAT","r");
    for(i=0;i<MAX;i++){
        fgets(str,80,fp);
        memcpy(sell[i].dm,str,4);
        memcpy(sell[i].mc,str+4,10);
        memcpy(ch,str+14,4);ch[4]=0;
        sell[i].dj=atoi(ch);
        memcpy(ch,str+18,5);ch[5]=0;
        sell[i].sl=atoi(ch);
        sell[i].je=(long)sell[i].dj*sell[i].sl;
    }
    fclose(fp);
}

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("OUT4.DAT","w");
    for(i=0;i<MAX;i++){
        printf("%s %s %4d %5d %5d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
        fprintf(fp,"%s %s %4d %5d %5d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
    }
    fclose(fp);
}
```

☆题目 32 (结构体运算题)

已知在文件 IN.DAT 中存有 100 个产品销售记录, 每个产品销售记录由产品代码 dm(字符型 4 位), 产品名称 mc(字符型 10 位), 单价 dj(整型), 数量 sl(整型), 金额 je(长整型)五部分组成。其中: 金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat(), 其功能要求: 按产品名称从大到小进行排列, 若产品名称相等, 则按金额从小到大进行排列, 最终排列结果仍存入结构数组 sell 中

, 最后调用函数 WriteDat()把结果输出到文件 OUT7.DAT 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include <stdio.h>
#include <mem.h>
#include <string.h>
#include <conio.h>
#include <stdlib.h>

#define MAX 100
typedef struct{
    char dm[5]; /*产品代码*/
    char mc[11]; /*产品名称*/
    int dj; /*单价*/
    int sl; /*数量*/
    long je; /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();

void SortDat()
{int i,j;
    PRO xy;
    for(i=0;i<99;i++)
        for(j=i+1;j<100;j++)
            if(strcmp(sell[i].mc,sell[j].mc)<0||strcmp(sell[i].mc,sell[j].mc)==0&&sell[i].je>sell[j].je)
                {xy=sell[i];sell[i]=sell[j];sell[j]=xy;}
}

void main()
{
    memset(sell,0,sizeof(sell));
    ReadDat();
    SortDat();
    WriteDat();
}
```



```

void ReadDat()
{
    FILE *fp;
    char str[80],ch[11];
    int i;

    fp=fopen("IN.DAT","r");
    for(i=0;i<100;i++){
        fgets(str,80,fp);
        memcpy(sell[i].dm,str,4);
        memcpy(sell[i].mc,str+4,10);
        memcpy(ch,str+14,4);ch[4]=0;
        sell[i].dj=atoi(ch);
        memcpy(ch,str+18,5);ch[5]=0;
        sell[i].sl=atoi(ch);
        sell[i].je=(long)sell[i].dj*sell[i].sl;
    }
    fclose(fp);
}

void WriteDat()
{
    FILE *fp;
    int i;

    fp=fopen("OUT7.DAT","w");
    for(i=0;i<100;i++){
        printf("%s %s %4d %5d %5d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
        fprintf(fp,"%s %s %4d %5d %5d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
    }
    fclose(fp);
}

```

★☆题目 33（方差运算题）

请编制函数 ReadDat()实现从文件 IN.DAT 中读取 1000 个十进制整数到数组 xx 中；请编制函数 Compute()分别计算出 xx 中偶数的个数 even，奇数的平均值 ave1，偶数的平均值 ave2 以及方差 totfc 的值，最后调用函数 WriteDat()把结果输出到 OUT.DAT 文件中。

计算方差的公式如下：

$$\text{totfc} = \frac{1}{N} \sum_{i=1}^N (\text{xx}[i] - \text{ave2})^2$$

设 N 为偶数的个数，xx[i]为偶数，ave2 为偶数的平均值。

原始数据文件存放的格式是：每行存放 10 个数，并用逗号隔开。（每个数均大于 0 且小

于等于 2000)

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main() 和输出数据函数 writeDat() 的内容。

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 1000

int xx[MAX], odd=0, even=0;
double ave1=0.0, ave2=0.0, totfc=0.0;
void WriteDat(void);

int ReadDat(void)
{int i;
 FILE *fp;

 if((fp=fopen("IN.DAT", "r"))==NULL) return 1;
/*****编制函数 ReadDat()的部分*****/
 for(i=0; i<MAX; i++)
 { fscanf(fp, "%d", &xx[i]);
  if((i+1)%10==0)
   fscanf(fp, "\n"); }
/*****/

 fclose(fp);
 return 0;
}

void Compute(void)
{ int i, yy[MAX];
 for(i=0; i<MAX; i++)
  yy[i]=0;
 for(i=0; i<MAX; i++)
  if(xx[i]%2==0) { yy[even++]=xx[i]; ave2+=xx[i]; }
  else { odd++; ave1+=xx[i]; }
 if(odd==0) ave1=0;
 else ave1/=odd;
 if(even==0) ave2=0;
 else ave2/=even;
 for(i=0; i<even; i++)
  totfc+=(yy[i]-ave2)*(yy[i]-ave2)/even;
}

void main()
```

```

{
    int i;

    for(i=0;i<MAX;i++)xx[i]=0;
    if(ReadDat()){
        printf("数据文件 IN.DAT 不能打开!\n");
        return;
    }
    Compute();
    printf("OVEN=%d\nAVE1=%f\nAVER2=%f\nTOTFC=%f\n",even,ave1,ave2,totfc);
    WriteDat();
}

```

```

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("OUT.DAT","w");
    fprintf(fp,"%d\n%f\n%f\n%f\n",even,ave1,ave2,totfc);
    fclose(fp);
}

```

题目 34 (素数题)

无忧 id 102 题提供了求素数 isPrime() 函数

程序 prog1.c 的功能是：选出 100 以上 1000 之内所有个位数字与十位数字之和被 10 除所得余数恰是百位数字的素数（如 293）。计算并输出上述这些素数的个数 cnt 以及这些素数值的和 sum。请考生编写函数 countValue() 实现程序要求，最后调用函数 writeDAT() 把结果 cnt 和 sum 输出到文件 bc10.out 中。

注意：部分源程序存放在文件 prog1.c 中。

请勿改动主函数 main() 和输出数据函数 writeDAT() 的内容。

```
#include <stdio.h>
```

```
int cnt, sum;
```

```
int isPrime(int m)
```

```

{
    int i;
    for(i=2;i<m;i++)
        if(m % i==0)return 0;
    return 1;
}

```

```
void countValue()
```

```

{ int i,j,bw,sw,gw;
  for(i=100;i<1000;i++)
    { bw=i/100;   sw=i%100/10; gw=i%10;
      if(isPrime(i)&&(gw+sw)%10==bw)  { cnt++; sum+=i;}
    }
}

```

```

void main()
{
  cnt=sum=0;

  countValue() ;
  printf("素数的个数=%d\n", cnt);
  printf("满足条件素数值的和=%d", sum);
  writeDAT();
}

```

```

writeDAT()
{
  FILE *fp;

  fp = fopen("bc10.out", "w");
  fprintf(fp, "%d\n%d\n", cnt, sum);
  fclose(fp);
}

```

★☆题目 35（级数运算题）

某级数的前两项 $A_1=1$ ， $A_2=1$ ，以后各项具有如下关系：

$A_n=A_{n-2}+2A_{n-1}$ （注： $n-2$ 与 $n-1$ 为下标）

下列程序 prog1.c 的功能是：要求依次对于整数 $M=100$ ， 1000 和 10000 求出对应的 n 值，使其满足： $S_n < M$ 且 $S_{n+1} \geq M$ ，这里 $S_n=A_1+A_2+\dots+A_n$ ，并依次把 n 值存入数组单元 $b[0]$ ， $b[1]$ 和 $b[2]$ 中，请编制 jsValue() 函数来实现此功能，最后调用函数 writeDat() 把数组 $b[]$ 中的值输出到 out.dat 文件中。

请勿改动主函数 main() 和写函数 writeDat() 的内容。

```

#include <stdio.h>
int b[3];

jsValue()
{int a1=1,a2=1,a12,sn,k=2;
  sn=a1+a2;
  while(1)
    {a12=a1+2*a2;

```

```

    if(sn<100&&sn+a12>=100) b[0]=k;
    if(sn<1000&&sn+a12>=1000) b[1]=k;
    if(sn<10000&&sn+a12>=10000) {b[2]=k;break;}
    sn=sn+a12;
    a1=a2;a2=a12;
    k++;
}
}

main()
{
    jsValue();
    printf("M=100,n=%d\nM=1000,n=%d\nM=10000,n=%d\n",b[0],b[1],b[2]);
    writeDat();
}

writeDat()
{
    FILE *fp;

    fp=fopen("out.dat","w");
    printf("%d\n%d\n%d\n",b[0],b[1],b[2]);
    fprintf(fp,"%d\n%d\n%d\n",b[0],b[1],b[2]);
    fclose(fp);
}

```

运行结果为:

M=100,n=6

M=1000,n=9

M=10000,n=11

★题目 36 (字符替换题)

函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 pS6.DAT 中。

替代关系： $f(p)=p*11 \bmod 256$ (p 是数组中某一个字符的 ASCII 值， $f(p)$ 是计算后新字符的 ASCII 值)，如果计算后 $f(p)$ 值小于等于 32 或 $f(p)$ 对应的字符是数字 0 至 9，则该字符不变，否则将 $f(p)$ 所对应的字符进行替代。

部分源程序存在文件 prog1.c 中。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include <stdio.h>
```

```
#include <string.h>
#include <conio.h>
#include <ctype.h>

unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/

int ReadDat(void);
void WriteDat(void);

void encryptChar()
{ int i,j;
  for(i=0;i<maxline;i++)
    for(j=0;j<strlen(xx[i]);j++)
      if(xx[i][j]*11%256<=32||xx[i][j]*11%256>='0'&&xx[i][j]*11%256<='9') continue;
      else xx[i][j]=xx[i][j]*11%256;
}

void main()
{
  clrscr();
  if(ReadDat()){
    printf("数据文件 ENG.IN 不能打开! \n\007");
    return;
  }
  encryptChar();
  WriteDat();
}

int ReadDat(void)
{
  FILE *fp;
  int i=0;
  unsigned char *p;

  if((fp=fopen("eng.in","r"))==NULL) return 1;
  while(fgets(xx[i],80,fp)!=NULL){
    p=strchr(xx[i],'\n');
    if(p)*p=0;
    i++;
  }
  maxline=i;
  fclose(fp);
  return 0;
}
```

```

}

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("ps6.dat","w");
    for(i=0;i<maxline;i++){
        printf("%s\n",xx[i]);
        fprintf(fp,"%s\n",xx[i]);
    }
    fclose(fp);
}

```

题目 37（字符替换题）

函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS7.DAT 中。

替代关系： $f(p)=p*11 \bmod 256$ （p 是数组中某一个字符的 ASCII 值，f(p)是计算后新字符的 ASCII 值），如果原字符是大写字母或计算后 f(p)值小于等于 32，则该字符不变，否则将 f(p)所对应的字符进行替代。

部分源程序存在文件 prog1.c 中。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <ctype.h>

unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/

int ReadDat(void);
void WriteDat(void);

void encryptChar()
{ int i,j;
  for(i=0;i<maxline;i++)
    for(j=0;j<strlen(xx[i]);j++)
      if(xx[i][j]*11%256<=32||xx[i][j]>='A'&&xx[i][j]<='Z') continue;
      else xx[i][j]=xx[i][j]*11%256;
}

```

```
}

void main()
{
    clrscr();
    if(ReadDat()){
        printf("数据文件 ENG.IN 不能打开! \n\007");
        return;
    }
    encryptChar();
    WriteDat();
}

int ReadDat(void)
{
    FILE *fp;
    int i=0;
    unsigned char *p;

    if((fp=fopen("eng.in", "r"))==NULL) return 1;
    while(fgets(xx[i],80,fp)!=NULL){
        p=strchr(xx[i], '\n');
        if(p)*p=0;
        i++;
    }
    maxline=i;
    fclose(fp);
    return 0;
}

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("ps7.dat", "w");
    for(i=0; i<maxline; i++){
        printf("%s\n", xx[i]);
        fprintf(fp, "%s\n", xx[i]);
    }
    fclose(fp);
}
```

题目 38（结构体运算题）

已知在文件 IN.DAT 中存有 100 个产品销售记录，每个产品销售记录由产品代码 dm(字符型 4 位)，产品名称 mc(字符型 10 位)，单价 dj(整型)，数量 sl(整型)，金额 je(长整型)五部分组成。其中：金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat()，其功能要求：按产品代码从小到大进行排列，若产品代码相同，则按金额从小到大进行排列，最终排列结果仍存入结构数组 sell 中，最后调用函数 WriteDat()把结果输出到文件 OUT6.DAT 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include <stdio.h>
#include <mem.h>
#include <string.h>
#include <conio.h>
#include <stdlib.h>

#define MAX 100
typedef struct{
    char dm[5]; /*产品代码*/
    char mc[11]; /*产品名称*/
    int dj;      /*单价*/
    int sl;      /*数量*/
    long je;     /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();

void SortDat()
{int i,j;
 PRO xy;
 for(i=0;i<99;i++)
    for(j=i+1;j<100;j++)
        if(strcmp(sell[i].dm,sell[j].dm)>0||strcmp(sell[i].dm,sell[j].dm)==0&&sell[i].je>sell[j].je)
            {xy=sell[i];sell[i]=sell[j];sell[j]=xy;}
}

void main()
{
    memset(sell,0,sizeof(sell));
    ReadDat();
    SortDat();
    WriteDat();
}
```

```

void ReadDat()
{
    FILE *fp;
    char str[80],ch[11];
    int i;

    fp=fopen("IN.DAT","r");
    for(i=0;i<100;i++){
        fgets(str,80,fp);
        memcpy(sell[i].dm,str,4);
        memcpy(sell[i].mc,str+4,10);
        memcpy(ch,str+14,4);ch[4]=0;
        sell[i].dj=atoi(ch);
        memcpy(ch,str+18,5);ch[5]=0;
        sell[i].sl=atoi(ch);
        sell[i].je=(long)sell[i].dj*sell[i].sl;
    }
    fclose(fp);
}

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("OUT6.DAT","w");
    for(i=0;i<100;i++){
        printf("%s %s %4d %5d %5d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
        fprintf(fp,"%s %s %4d %5d %5d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
    }
    fclose(fp);
}

```

★☆☆题目 39（选票问题）

现有一个 10 个人 100 行的选票数据文件 IN.DAT，其数据存放的格式是每条记录的长度均为 10 位，第一位表示第一个人的选中情况，第二位表示第二个人的选中情况，依此类推：内容均为字符 0 和 1，1 表示此人被选中，0 表示此人未被选中，全选或不选均为无效的选票。给定函数 ReadDat()的功能是把选票数据读入到字符串数组 xx 中。请编制函数 CountRs()来统计每个人的选票数并把得票数依次存入 yy[0]到 yy[9]中。把结果 yy 输出到文件 OUT.DAT 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 writeDat()的内容。

```
#include <stdio.h>

char xx[100][11];
int yy[10];
int ReadDat(void);
void WriteDat(void);

void CountRs(void)
{ int i,j,count;
  for(i=0;i<100;i++)
  { count=0;
    for(j=0;j<10;j++)
      if(xx[i][j]=='1') count++;
    if(count==0||count==10) continue;
    for(j=0;j<10;j++)
      if(xx[i][j]=='1') yy[j]++;
  }
}

void main()
{
  int i;

  for(i=0;i<10;i++)yy[i]=0;
  if(ReadDat()){
    printf("选票数据文件 IN.DAT 不能打开! \n\007");
    return;
  }
  CountRs();
  WriteDat();
}

int ReadDat(void)
{
  FILE *fp;
  int i;

  if((fp=fopen("IN.DAT","r")) == NULL) return 1;
  for(i=0;i<100;i++){
    if(fgets(xx[i],11,fp)==NULL)return 1;
    xx[i][10]='\0';
  }
}
```

```

fclose(fp);
return 0;
}

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("OUT.DAT","w");
    for(i=0;i<10;i++){
        fprintf(fp,"%d\n",yy[i]);
        printf("第%d 个人的选票数=%d\n",i+1,yy[i]);
    }
    fclose(fp);
}

```

☆题目 40（整数统计运算题）

已知在文件 IN.DAT 中存有若干个（个数<200）四位数字的正整数，函数 ReadDat()读取这若干个正整数并存入数组 xx 中。请编制函数 CalValue()，其功能要求：

1、求出这文件中共有多少个正整数 totNum；

2、求这些数中的各位数字之和是偶数的数的个数 totCnt，

以及满足此条件的这些数的算术平均值 totPjz，最后调用函数 writeDat()把所求的结果输出到文件 OUT.DAT 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读函数 ReadDat()和写函数 writeDat()的内容。

```

#include <stdio.h>
#include <conio.h>
#define MAXNUM 200

int xx[MAXNUM];
int totNum=0; /*文件 IN.DAT 中共有多少个正整数*/
int totCnt=0; /*符合条件的正整数的个数*/
double totPjz=0.0; /*平均值*/

int ReadDat(void);
void writeDat(void);

void CalValue(void)
{ int i,qw,bw,sw,gw;

```

```
    for(i=0;i<MAXNUM;i++)
if(xx[i]>0)
{ totNum++;
    qw=xx[i]/1000;
    bw=xx[i]/100%10;
    sw=xx[i]%100/10;
    gw=xx[i]%10;
    if((qw+bw+sw+gw)%2==0) { totCnt++; totPjz+=xx[i];}
}
if(totCnt==0) totPjz=0;
else totPjz/=totCnt;
}
```

```
void main()
{
    int i;
    clrscr();
    for(i=0;i<MAXNUM;i++)xx[i]=0;
    if(ReadDat()){
        printf("数据文件 IN.DAT 不能打开!\007\n");
        return;
    }
    CalValue();
    printf("文件 IN.DAT 中共有正整数=%d 个\n",totNum);
    printf("符合条件的正整数的个数=%d 个\n",totCnt);
    printf("平均值=%.2f\n",totPjz);
    writeDat();
}
```

```
int ReadDat(void)
{
    FILE *fp;
    int i=0;

    if((fp=fopen("IN.DAT","r"))==NULL) return 1;
    while(!feof(fp)){
        fscanf(fp,"%d",&xx[i++]);
    }
    fclose(fp);
    return 0;
}
```

```
void writeDat(void)
```

```

{
    FILE *fp;
    fp=fopen("OUT.DAT","w");
    fprintf(fp,"%d\n%d\n%.2f\n",totNum,totCnt,totPjz);
    fclose(fp);
}

```

☆题目 41 (SIX/NINE 问题)

下列程序 prog1.c 的功能是：计算出自然数 SIX 和 NINE，它们满足的条件是 $SIX+SIX+SIX=NINE+NINE$ 的个数 cnt 以及满足此条件所有的 SIX 与 NINE 的和 SUM。请编写函数 countValue()实现程序的要求，最后调用函数 writeDat()把结果 cnt 和 sum，输出到文件 OUT15.DAT 中。

其中的 S, I, X, N, E 各代表一个十进制数字。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()和输出数据函数 writeDat()的内容。

```
#include <stdio.h>
```

```
int cnt,sum;
```

```
void countValue()
```

```
{ int s,i,x,n,e,six,nine;
```

```
    for(s=1;s<10;s++)
```

```
        for(i=0;i<10;i++)
```

```
            for(x=0;x<10;x++)
```

```
                for(n=1;n<10;n++)
```

```
                    for(e=0;e<10;e++)
```

```
                        { six=s*100+i*10+x;
```

```
                            nine=n*1000+i*100+n*10+e;
```

```
                            if(3*six==2*nine)
```

```
                                { cnt++;sum+=six+nine;}
```

```
                        }
```

```
    }
```

```
void main()
```

```
{
```

```
    cnt=sum=0;
```

```
    countValue();
```

```
    printf("满足条件的个数=%d\n",cnt);
```

```
    printf("满足条件所有的 SIX 与 NINE 的和=%d\n",sum);
```

```
    writeDat();
```

```
}
```

```
writeDat()
```

```

{
FILE *fp;

fp=fopen("OUT15.DAT","w");
fprintf(fp,"%d\n%d\n",cnt,sum);
fclose(fp);
}

```

题目 42 结构体运算题

已知在文件 IN.DAT 中存有 100 个产品销售记录，每个产品销售记录由产品代码 dm(字符型 4 位)，产品名称 mc(字符型 10 位)，单价 dj(整型)，数量 sl(整型)，金额 je(长整型)五部分组成。其中：金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat()，其功能要求：按产品名称从大到小进行排列，若产品名称相等，则按金额从大到小进行排列，最终排列结果仍存入结构数组 sell 中，最后调用函数 WriteDat()把结果输出到文件 OUT5.DAT 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include <stdio.h>
#include <mem.h>
#include <string.h>
#include <conio.h>
#include <stdlib.h>

#define MAX 100
typedef struct{
    char dm[5]; /*产品代码*/
    char mc[11]; /*产品名称*/
    int dj;      /*单价*/
    int sl;      /*数量*/
    long je;     /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();

void SortDat()
{int i,j;
  PRO xy;
  for(i=0;i<99;i++)
    for(j=i+1;j<100;j++)
      if(strcmp(sell[i].mc,sell[j].mc)<0||strcmp(sell[i].mc,sell[j].mc)==0&&sell[i].je<sell[j].je)
        {xy=sell[i];sell[i]=sell[j];sell[j]=xy;}
}

```

```
void main()
{
    memset(sell,0,sizeof(sell));
    ReadDat();
    SortDat();
    WriteDat();
}

void ReadDat()
{
    FILE *fp;
    char str[80],ch[11];
    int i;

    fp=fopen("IN.DAT","r");
    for(i=0;i<100;i++){
        fgets(str,80,fp);
        memcpy(sell[i].dm,str,4);
        memcpy(sell[i].mc,str+4,10);
        memcpy(ch,str+14,4);ch[4]=0;
        sell[i].dj=atoi(ch);
        memcpy(ch,str+18,5);ch[5]=0;
        sell[i].sl=atoi(ch);
        sell[i].je=(long)sell[i].dj*sell[i].sl;
    }
    fclose(fp);
}

void WriteDat()
{
    FILE *fp;
    int i;

    fp=fopen("OUT5.DAT","w");
    for(i=0;i<100;i++){
        printf("%s %s %4d %5d %5d\n",sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
        fprintf(fp,"%s %s %4d %5d %5d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
    }
    fclose(fp);
}
```

★题目 43（字符排序题）

无忧 id 82 题(只是将结果按“从大到小”排序)

函数 ReadDat()实现从文件 in.dat 中读取 20 行数据存放到字符串数组 xx 中(每行字符串长度均小于 80)。请编制函数 jsSort(), 其函数的功能是: 以行为单位对字符串变量的下标为奇数的字符按其 ASCII 值从小到大的顺序进行排序, 排序后的结果仍按行重新存入字符串数组 xx 中, 最后调用函数 WriteDat()把结果 xx 输出到文件 out.dat 中。

例如: 位置 0 1 2 3 4 5 6 7

源字符串 a b c d e f g h

则处理后字符串 a h c f e d g b

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
```

```
char xx[20][80];
```

```
void jsSort()
{int i,j,k,trl;
 char ch;
 for(i=0;i<20;i++)
 { trl=strlen(xx[i]);
  for(j=1;j<trl-2;j=j+2)
   for(k=j+2;k<trl;k=k+2)
    if(xx[i][j]>xx[i][k]) { ch=xx[i][j];xx[i][j]=xx[i][k];xx[i][k]=ch;}
 }
}
```

```
void main()
{
 readDat();
 jsSort();
 writeDat();
}
```

```
readDat()
{
 FILE *in;
 int i=0;
 char *p;

 in=fopen("in.dat","r");
 while(i<20&&fgets(xx[i],80,in)!=NULL){
  p=strchr(xx[i],'\n');
```

```

        if(p)*p=0;
        i++;
    }
    fclose(in);
}

writeDat()
{
    FILE *out;
    int i;

    out=fopen("out.dat","w");
    clrscr();
    for(i=0;i<20;i++){
        printf("%s\n",xx[i]);
        fprintf(out,"%s\n",xx[i]);
    }
    fclose(out);
}

```

★☆题目 44（字符串左右排序交换题）

函数 ReadDat()实现从文件 in.dat 中读取 20 行数据存放到字符串数组 xx 中（每行字符串长度均小于 80）。请编制函数 jsSort()，其函数的功能是：以行为单位对字符串按给定的条件进行排序，排序后的结果仍按行重新存入字符串数组 xx 中，最后调用函数 WriteDat()把结果 xx 输出到文件 out.dat 中。

条件：从字符串中间一分为二，左边部分按字符的 ASCII 值降序排序，排序后左边部分与右边部分进行交换。如果原字符串长度为奇数，则最中间的字符不参加处理，字符仍放在原位置上。

```

例如：位置      0  1  2  3  4  5  6  7  8
源字符串  a  b  c  d  h  g  f  e
          1  2  3  4  9  8  7  6  5
则处理后字符串 h  g  f  e  d  c  b  a
          8  7  6  5  9  4  3  2  1

```

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include <stdio.h>
#include <string.h>
#include <conio.h>

char xx[20][80];

void jsSort()
{ int i,j,k,slr1,half;

```

```
char temp;
for(i=0;i<20;i++)
{ strl=strlen(xx[i]);
  half=strl/2;
  for(j=0;j<half-1;j++)
    for(k=j+1;k<half;k++)
      if(xx[i][j]<xx[i][k])
        { temp=xx[i][j]; xx[i][j]=xx[i][k]; xx[i][k]=temp;}
  for(j=half-1,k=strl-1;j>=0;j--,k--)
    { temp=xx[i][j]; xx[i][j]=xx[i][k]; xx[i][k]=temp;}
}
```

```
void main()
{
  readDat();
  jsSort();
  writeDat();
}
```

```
readDat()
{
  FILE *in;
  int i=0;
  char *p;

  in=fopen("in.dat","r");
  while(i<20&&fgets(xx[i],80,in)!=NULL){
    p=strchr(xx[i],'\n');
    if(p)*p=0;
    i++;
  }
  fclose(in);
}
```

```
writeDat()
{
  FILE *out;
  int i;

  clrscr();
  out=fopen("out.dat","w");
  for(i=0;i<20;i++){
    printf("%s\n",xx[i]);
```

```

    fprintf(out,"%s\n",xx[i]);
}
fclose(out);
}

```

★题目 45（数字统计排序题）

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat()把这些数存入数组 a 中，请考生编制一函数 jsVal()，其功能是：依次从数组 a 中取出一个四位数，如果该四位数连续大于该四位数以前的五个数且该数是偶数（该四位数以前不满五个数，则不统计），则统计出满足此条件的个数 cnt 并把这些四位数按从大到小的顺序存入数组 b 中，最后调用写函数 writeDat()把结果 cnt 以及数组 b 中符合条件的四位数输出到 OUT.DAT 文件中。

注意：部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[200]，b[200]，已定义变量：cnt

请勿改动数据文件 IN.DAT 中的任何数据、主函数 main()、读函数 readDat()和写函数 writeDat()的内容。

```

#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0 ;

void jsVal()
{int i,j,flag;
 for(i=5;i<MAX;i++)
 {for(j=i-5;j<i;j++)
  if(a[i]>a[j]&& a[i]%2==0) flag=1;
  else { flag=0; break;}
  if(flag==1) b[cnt++]=a[i];
 }
 for(i=0;i<cnt-1;i++)
 for(j=i+1;j<cnt;j++)
  if(b[i]<b[j]) {flag=b[i];b[i]=b[j];b[j]=flag;}
}

void readDat()
{
 int i ;
 FILE *fp ;
 fp = fopen("in.dat", "r") ;
 for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
 fclose(fp) ;
}

```

```

void main()
{
    int i ;
    readDat() ;
    jsVal() ;
    printf("满足条件的数=%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
    printf("\n") ;
    writeDat() ;
}

writeDat()
{
    FILE *fp ;
    int i ;
    fp = fopen("out.dat", "w") ;
    fprintf(fp, "%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) fprintf(fp, "%d\n", b[i]) ;
    fclose(fp) ;
}

```

☆题目 46（数字统计排序题）

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat() 把这些数存入数组 a 中，请考生编制一函数 jsVal()，其功能是：依次从数组 a 中取出一个四位数，如果该四位数连续大于该四位数以前的五个数且该数是奇数，且该数必须能被 7 整除（该四位数以前不满五个数，则不统计），则统计出满足此条件的个数 cnt 并把这些四位数按从大到小的顺序存入数组 b 中，最后调用写函数 writeDat() 把结果 cnt 以及数组 b 中符合条件的四位数输出到 OUT.DAT 文件中。

注意：部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[200]，b[200]，已定义变量：cnt

请勿改动数据文件 IN.DAT 中的任何数据、主函数 main()、读函数 readDat() 和写函数 writeDat() 的内容。

```

#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0 ;

void jsVal()
{ int i,j,flag;
  for(i=5;i<MAX;i++)
  { for(j=i-5;j<i;j++)
    if(a[i]>a[j]&& a[i]%2&& a[i]%7==0) flag=1;
    else { flag=0;break;}
    if(flag==1) b[cnt++]=a[i];
  }
}

```

```

    }
    for(i=0;i<cnt-1;i++)
        for(j=i+1;j<cnt;j++)
            if(b[i]<b[j]) { flag=b[i]; b[i]=b[j]; b[j]=flag;}
}

void readDat()
{
    int i ;
    FILE *fp ;
    fp = fopen("in.dat", "r") ;
    for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
    fclose(fp) ;
}

void main()
{
    int i ;
    readDat() ;
    jsVal() ;
    printf("满足条件的数=%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
    printf("\n") ;
    writeDat() ;
}

writeDat()
{
    FILE *fp ;
    int i ;
    fp = fopen("out.dat", "w") ;
    fprintf(fp, "%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) fprintf(fp, "%d\n", b[i]) ;
    fclose(fp) ;
}

```

题目 47（结构体运算题）

已知在文件 IN.DAT 中存有 100 个产品销售记录，每个产品销售记录由产品代码 dm(字符型 4 位)，产品名称 mc(字符型 10 位)，单价 dj(整型)，数量 sl(整型)，金额 je(长整型)五部分组成。其中：金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat()，其功能要求：按金额从大到小进行排列，若金额相等，则按产品代码从小到大进行排列，最终排列结果仍存入结构数组 sell 中，最后调用函数 WriteDat()把结果输出到文件 OUT3.DAT 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include <stdio.h>
#include <mem.h>
#include <string.h>
#include <conio.h>
#include <stdlib.h>

#define MAX 100
typedef struct{
    char dm[5]; /*产品代码*/
    char mc[11]; /*产品名称*/
    int dj; /*单价*/
    int sl; /*数量*/
    long je; /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();

void SortDat()
{int i,j;
    PRO xy;
    for(i=0;i<99;i++)
        for(j=i+1;j<100;j++)
            if(sell[i].je<sell[j].je||sell[i].je==sell[j].je&&strcmp(sell[i].dm,sell[j].dm)>0)
                {xy=sell[i];sell[i]=sell[j];sell[j]=xy;}
}

void main()
{
    memset(sell,0,sizeof(sell));
    ReadDat();
    SortDat();
    WriteDat();
}

void ReadDat()
{
    FILE *fp;
    char str[80],ch[11];
    int i;

    fp=fopen("IN.DAT","r");
```

```

for(i=0;i<100;i++){
    fgets(str,80,fp);
    memcpy(sell[i].dm,str,4);
    memcpy(sell[i].mc,str+4,10);
    memcpy(ch,str+14,4);ch[4]=0;
    sell[i].dj=atoi(ch);
    memcpy(ch,str+18,5);ch[5]=0;
    sell[i].sl=atoi(ch);
    sell[i].je=(long)sell[i].dj*sell[i].sl;
}
fclose(fp);
}

void WriteDat()
{
    FILE *fp;
    int i;

    fp=fopen("OUT3.DAT","w");
    for(i=0;i<100;i++){
        printf("%s %s %4d %5d %5d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
        fprintf(fp,"%s %s %4d %5d %5d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
    }
    fclose(fp);
}

```

☆题目 48（整数各位数字运算题）

已知数据文件 in.dat 中存有 300 个四位数，并已调用读函数 ReadDat()把这些数存入数组 a 中，请编制一函数 jsValue()，其功能是：求出千位数上的数加个位数上的数等于百位数上的数加十位数上的数的个数 cnt，再把所有满足此条件的四位数依次存入数组 b 中，然后对数组 b 的四位数按从小到大的顺序进行排序，最后调用写函数 writeDat()把结果输出到 out.dat 文件。

例如：6712， $6+2=7+1$ ，则该数满足条件存入数组 b 中，且个数 $cnt=cnt+1$ 。

8129， $8+9 \neq 1+2$ ，则该数不满足条件忽略。

部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[300]，b[300]，已定义变量：cnt

请勿改动主函数 main()、读函数 ReadDat()和写函数 writeDat()的内容。

```

#include <stdio.h>
int a[300],b[300],cnt=0;

jsValue()
{int i,qw,bw,sw,gw,j;

```



```

for(i=0;i<300;i++)
{qw=a[i]/1000;    bw=a[i]%1000/100;
 sw=a[i]%100/10;  gw=a[i]%10;
 if(qw+gw==bw+sw) b[cnt++]=a[i];
}
for(i=0;i<cnt-1;i++)
for(j=i+1;j<cnt;j++)
    if(b[i]>b[j]) {qw=b[i]; b[i]=b[j]; b[j]=qw;}
}

```

```

main()
{
    int i;

    readDat();
    jsValue();
    writeDat();
    printf("cnt=%d\n",cnt);
    for(i=0;i<cnt;i++) printf("b[%d]=%d\n",i,b[i]);
}

```

```

readDat()
{
    FILE *fp;
    int i;
    fp=fopen("in.dat","r");
    for(i=0;i<300;i++)fscanf(fp,"%d",&a[i]);
    fclose(fp);
}

```

```

writeDat()
{
    FILE *fp;
    int i;
    fp=fopen("out.dat","w");
    fprintf(fp,"%d\n",cnt);
    for(i=0;i<cnt;i++)fprintf(fp,"%d\n",b[i]);
    fclose(fp);
}

```

题目 49（整数各位数字运算题）

已知数据文件 in.dat 中存有 300 个四位数，并已调用读函数 ReadDat()把这些数存入数组 a 中，请编制一函数 jsValue()，其功能是：求出千位数上的数减百位数减十位数上的数减个位数上的数大于零的个数 cnt，再求出所有满足此条件的四位数平均值 pjz1，以及不满足此条件的四位数平均值 pjz2，最后调用写函数 writeDat()把结果输出到 out.dat 文件。

例如：9123,9-1-2-3>0,则该数满足条件计算平均值 pjz1，且个数 cnt=cnt+1。9812,9-8-1-2<0,则该数不满足条件计算平均值 pjz2。

部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[300]，已定义变量：cnt,pjz1,pjz2。

请勿改动主函数 main()、读函数 ReadDat()和写函数 writeDat()的内容。

```
#include <stdio.h>
int a[300],cnt=0;
double pjz1=0.0,pjz2=0.0;

jsValue()
{int i,qw,bw,sw,gw,n=0;
for(i=0;i<300;i++)
    {qw=a[i]/1000;    bw=a[i]%1000/100;
    sw=a[i]%100/10;  gw=a[i]%10;
    if(qw-bw-sw-gw>0) {cnt++;pjz1+=a[i];}
    else {n++;pjz2+=a[i];}
    }
if(cnt==0) pjz1=0;
else pjz1/=cnt;
if(n==0) pjz2=0;
else pjz2/=n;
}

main()
{
    int i;

    readDat();
    jsValue();
    writeDat();
    printf("cnt=%d\n 满足条件的平均值 pzj1=%7.2f\n 不满足条件的平均值
    pzj2=%7.2f\n",cnt,pjz1,pjz2);
}

readDat()
{
    FILE *fp;
    int i;
    fp=fopen("in.dat","r");
    for(i=0;i<300;i++)fscanf(fp,"%d",&a[i]);
}
```

```

fclose(fp);
}

writeDat()
{
    FILE *fp;
    int i;
    fp=fopen("out.dat","w");
    fprintf(fp,"%d\n%7.2f\n%7.2f\n",cnt,pjz1,pjz2);
    fclose(fp);
}

```

题目 50（字符串左右排序交换题）

函数 ReadDat()实现从文件 in.dat 中读取 20 行数据存放到字符串数组 xx 中（每行字符串长度均小于 80）。请编制函数 jsSort()，其函数的功能是：以行为单位对字符串按给定的条件进行排序，排序后的结果仍按行重新存入字符串数组 xx 中，最后调用函数 WriteDat()把结果 xx 输出到文件 out.dat 中。

条件：从字符串中间一分为二，左边部分按字符的 ASCII 值升序排序，排序后左边部分与右边部分进行交换。如果原字符串长度为奇数，则最中间的字符不参加处理，字符仍放在原位置上。

例如：位置	0	1	2	3	4	5	6	7	8
源字符串	d	c	b	a	h	g	f	e	
	4	3	2	1	9	8	7	6	5
则处理后字符串	h	g	f	e	a	b	c	d	
	8	7	6	5	9	1	2	3	4

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读函数 ReadDat()和写函数 WriteDat()的内容。

```

#include <stdio.h>
#include <string.h>
#include <conio.h>

```

```
char xx[20][80];
```

```

void jsSort()
{ int i,j,k,slr,half;
  char temp;
  for(i=0;i<20;i++)
  { slr=strlen(xx[i]);
    half=slr/2;
    for(j=0;j<half-1;j++)
      for(k=j+1;k<half;k++)
        if(xx[i][j]>xx[i][k])
          { temp=xx[i][j]; xx[i][j]=xx[i][k]; xx[i][k]=temp;}
  }
}

```

```
        for(j=half-1,k=strl-1;j>=0;j--,k--)
            { temp=xx[i][j]; xx[i][j]=xx[i][k]; xx[i][k]=temp;}
    }
}
```

```
void main()
{
    readDat();
    jsSort();
    writeDat();
}
```

```
readDat()
{
    FILE *in;
    int i=0;
    char *p;

    in=fopen("in.dat","r");
    while(i<20 && fgets(xx[i],80,in)!=NULL){
        p=strchr(xx[i],'\n');
        if(p) *p=0;
        i++;
    }
    fclose(in);
}
```

```
writeDat()
{
    FILE *out;
    int i;
    clrscr();
    out=fopen("out.dat","w");
    for(i=0;i<20;i++){
        printf("%s\n",xx[i]);
        fprintf(out,"%s\n",xx[i]);
    }
    fclose(out);
}
```

★☆☆题目 51（整数各位打散组合运算题）

没有提供求素数函数 isprime() ,可自己编制

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat()把这些数存入数组 a 中，请考生编制一函数 jsVal()，其功能是：把千位数字和十位数字重新组成一个新的十位数（新十位数的十位数字是原四位数的千位数字，新十位数的个位数字是原四位数的十位数字），以及把个位数字和百位数字组成另一个新的十位数（新十位数的十位数字是原四位数的个位数字，新十位数的个位数字是原四位数的百位数字），如果新组成的两个十位数均是素数且新数的十位数字均不为零，则将满足此条件的四位数按从大到小的顺序存入数组 b 中，并要计算满足上述条件的四位数的个数 cnt。最后 main()函数调用写函数 writeDat()把结果 cnt 以及数组 b 中符合条件的四位数输出到 OUT.DAT 文件中。

注意：部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[200]，b[200]，已定义变量：cnt

请勿改动数据文件 IN.DAT 中的任何数据、主函数 main()、读函数 readDat()和写函数 writeDat()的内容。

```
#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0 ;

int isprime(int m)
{int i;
 for(i=2;i<=m/2;i++)
  if(m%i==0) return 0;
 return 1;
}

void jsVal()
{int i,j,qw,bw,sw,gw;
 int ab,cd;
 for(i=0;i<200;i++)
 {qw=a[i]/1000;    bw=a[i]%1000/100;
  sw=a[i]%100/10;  gw=a[i]%10;
  ab=10*qw+sw;    cd=10*gw+bw;
  if(isprime(ab)&&isprime(cd)&&ab>=10&&cd>=10) b[cnt++]=a[i];
 }
 for(i=0;i<cnt-1;i++)
  for(j=i+1;j<cnt;j++)
   if(b[i]<b[j]) {qw=b[i];b[i]=b[j];b[j]=qw;}
}

void readDat()
{
 int i ;
 FILE *fp ;
 fp = fopen("in.dat", "r") ;
```

```

    for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
    fclose(fp) ;
}

```

```

void main()
{
    int i ;
    readDat() ;
    jsVal() ;
    printf("满足条件的数=%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
    printf("\n") ;
    writeDat() ;
}

```

```

writeDat()
{
    FILE *fp ;
    int i ;
    fp = fopen("out.dat", "w") ;
    fprintf(fp, "%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) fprintf(fp, "%d\n", b[i]) ;
    fclose(fp) ;
}

```

题目 52（整数各位打散组合运算题）

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat() 把这些数存入数组 a 中，请考生编制一函数 jsVal()，其功能是：把千位数字和个位数字重新组成一个新的十位数（新十位数的十位数字是原四位数的千位数字，新十位数的个位数字是原四位数的个位数字），以及把百位数字和十位数字组成另一个新的十位数（新十位数的十位数字是原四位数的百位数字，新十位数的个位数字是原四位数的十位数字），如果新组成的两个十位数均是奇数并且两个十位数中至少有一个数能被 5 整除，同时两个新数的十位数字均不为零，则将满足此条件的四位数按从大到小的顺序存入数组 b 中，并要计算满足上述条件的四位数的个数 cnt。最后 main() 函数调用写函数 writeDat() 把结果 cnt 以及数组 b 中符合条件的四位数输出到 OUT.DAT 文件中。

注意：部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[200]，b[200]，已定义变量：cnt

请勿改动数据文件 IN.DAT 中的任何数据、主函数 main()、读函数 readDat() 和写函数 writeDat() 的内容。

```

#include <stdio.h>
#define MAX  200
int a[MAX], b[MAX], cnt = 0 ;

```

```
void jsVal()
{int i,thou,hun,ten,data,j;
 int ab,cd;
 for(i=0;i<MAX;i++)
 {thou=a[i]/1000;    hun=a[i]%1000/100;
  ten=a[i]%100/10;   data=a[i]%10;
  ab=10*thou+data;   cd=10*hun+ten;
  if(ab%2&&cd%2&&(ab%5==0||cd%5==0)&&ab>=10&&cd>=10) b[cnt++]=a[i];
 }
 for(i=0;i<cnt-1;i++)
  for(j=i+1;j<cnt;j++)
   if(b[i]<b[j]) {data=b[i];b[i]=b[j];b[j]=data;}
}
```

```
void readDat()
{
 int i ;
 FILE *fp ;
 fp = fopen("in.dat", "r") ;
 for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
 fclose(fp) ;
}
```

```
void main()
{
 int i ;
 readDat() ;
 jsVal() ;
 printf("满足条件的数=%d\n", cnt) ;
 for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
 printf("\n") ;
 writeDat() ;
}
```

```
writeDat()
{
 FILE *fp ;
 int i ;
 fp = fopen("out.dat", "w") ;
 fprintf(fp, "%d\n", cnt) ;
 for(i = 0 ; i < cnt ; i++) fprintf(fp, "%d\n", b[i]) ;
 fclose(fp) ;
}
```

☆题目 53（整数各位数字运算题）

已知数据文件 in.dat 中存有 300 个四位数，并已调用读函数 ReadDat()把这些数存入数组 a 中，请编制一函数 jsValue()，其功能是：求出千位数上的数减百位数上的数减十位数上的数减个位数上的数大于零的个数 cnt，再把所有满足此条件的四位数依次存入数组 b 中，然后对数组 b 的四位数按从小到大的顺序进行排序，最后调用写函数 writeDat()把结果输出到 out.dat 文件。

例如：9123，9-1-2-3>0，则该数满足条件存入数组 b 中，且个数 cnt=cnt+1。

9812，9-8-1-2>0，则该数不满足条件忽略。

部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[300]，b[300]，已定义变量：cnt

请勿改动主函数 main()、读函数 ReadDat()和写函数 writeDat()的内容。

```
#include <stdio.h>
int a[300],b[300],cnt=0;
jsValue()
{ int i,j,qw,bw,sw,gw;
  for(i=0;i<300;i++)
  { qw=a[i]/1000; bw=a[i]/100%10;
    sw=a[i]%100/10; gw=a[i]%10;
    if((qw-bw-sw-gw)>0) b[cnt++]=a[i];
  }
  for(i=0;i<cnt-1;i++)
  for(j=i+1;j<cnt;j++)
    if(b[i]>b[j]) { qw=b[i];b[i]=b[j];b[j]=qw;}
}

main()
{
  int i;

  readDat();
  jsValue();
  writeDat();
  printf("cnt=%d\n",cnt);
  for(i=0;i<cnt;i++) printf("b[%d]=%d\n",i,b[i]);
}

readDat()
{
  FILE *fp;
  int i;
  fp=fopen("in.dat","r");
```



```

for(i=0;i<300;i++)fscanf(fp,"%d",&a[i]);
fclose(fp);
}

```

```

writeDat()
{
    FILE *fp;
    int i;
    fp=fopen("out.dat","w");
    fprintf(fp,"%d\n",cnt);
    for(i=0;i<cnt;i++)fprintf(fp,"%d\n",b[i]);
    fclose(fp);
}

```

题目 54（整数各位数运算排序题）

已知数据文件 in.dat 中存有 300 个四位数，并已调用读函数 ReadDat()把这些数存入数组 a 中，请编制一函数 jsValue()，其功能是：求出千位数上的数加百位数上的数等于十位数上的数加个位数上的数的个数 cnt，再把所有满足此条件的四位数依次存入数组 b 中，然后对数组 b 的四位数按从大到小的顺序进行排序，最后调用写函数 writeDat()把结果输出到 out.dat 文件。

例如：7153，7+1=5+3，则该数满足条件存入数组 b 中，且个数 cnt=cnt+1。

8129，8+1<>2+9，则该数不满足条件忽略。

部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[300]，b[300]，已定义变量：cnt

请勿改动主函数 main()、读函数 ReadDat()和写函数 writeDat()的内容。

```
#include <stdio.h>
```

```
int a[300],b[300],cnt=0;
```

```

jsValue()
{ int i,j,qw,bw,sw,gw;
  for(i=0;i<300;i++)
  { qw=a[i]/1000;   bw=a[i]/100%10;
    sw=a[i]%100/10; gw=a[i]%10;
    if(qw+bw==sw+gw) b[cnt++]=a[i];
  }
  for(i=0;i<cnt-1;i++)
  for(j=i+1;j<cnt;j++)
    if(b[i]<b[j]) { qw=b[i];b[i]=b[j];b[j]=qw;}
}

```

```
main()
```

```

{
    int i;

```

```

readDat();
jsValue();
writeDat();
printf("cnt=%d\n",cnt);
for(i=0;i<cnt;i++) printf("b[%d]=%d\n",i,b[i]);
}

```

```

readDat()
{
    FILE *fp;
    int i;
    fp=fopen("in.dat","r");
    for(i=0;i<300;i++)fscanf(fp,"%d",&a[i]);
    fclose(fp);
}

```

```

writeDat()
{
    FILE *fp;
    int i;
    fp=fopen("out.dat","w");
    fprintf(fp,"%d\n",cnt);
    for(i=0;i<cnt;i++)fprintf(fp,"%d\n",b[i]);
    fclose(fp);
}

```

题目 55（素数统计排序题）

已知数据文件 in.dat 中存有 300 个四位数，并已调用读函数 ReadDat()把这些数存入数组 a 中，请编制一函数 jsValue()，其功能是：求出所有这些四位数是素数的个数 cnt，再把所有满足此条件的四位数依次存入数组 b 中，然后对数组 b 的四位数按从小到大的顺序进行排序，最后调用写函数 writeDat()把结果输出到 out.dat 文件。

例如：5591 是素数，则该数满足条件存入数组 b 中，且个数 cnt=cnt+1。9812 是非素数，则该数不满足条件忽略。

部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[300]，b[300]，已定义变量：cnt

请勿改动主函数 main()、读函数 ReadDat()和写函数 writeDat()的内容。

```

#include <stdio.h>
int a[300],b[300],cnt=0;
int isP(int m){
    int i;
    for(i=2;i<m;i++)

```

```
        if(m%i==0)return 0;
        return 1;
    }

jsValue()
{int i,j,value;
 for(i=0;i<300;i++)
    if(isP(a[i])) b[cnt++]=a[i];
 for(i=0;i<cnt-1;i++)
    for(j=i+1;j<cnt;j++)
        if(b[i]>b[j]) {value=b[i]; b[i]=b[j]; b[j]=value;}
}
```

```
main()
{
    int i;

    readDat();
    jsValue();
    writeDat();
    printf("cnt=%d\n",cnt);
    for(i=0;i<cnt;i++) printf("b[%d]=%d\n",i,b[i]);
}
```

```
readDat()
{
    FILE *fp;
    int i;
    fp=fopen("in.dat","r");
    for(i=0;i<300;i++)fscanf(fp,"%d",&a[i]);
    fclose(fp);
}
```

```
writeDat()
{
    FILE *fp;
    int i;
    fp=fopen("out.dat","w");
    fprintf(fp,"%d\n",cnt);
    for(i=0;i<cnt;i++)fprintf(fp,"%d\n",b[i]);
    fclose(fp);
}
```

题目 56 (字符替换题)

函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS5.DAT 中。

替代关系： $f(p)=p*11\text{mod } 256$ (p 是数组中某一个字符的 ASCII 值， $f(p)$ 是计算后新字符的 ASCII 值)，如果计算后 $f(p)$ 值小于等于 32 或 $f(p)$ 对应的字符是小写字母，则该字符不变，否则将 $f(p)$ 所对应的字符进行替代。

部分源程序存在文件 prog1.c 中。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <conio.h>
```

```
#include <ctype.h>
```

```
unsigned char xx[50][80];
```

```
int maxline=0;/*文章的总行数*/
```

```
int ReadDat(void);
```

```
void WriteDat(void);
```

```
void encryptChar()
```

```
{ int i,j;
```

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

```
for(j=0;j<strlen(xx[i]);j++)
```

```
if(xx[i][j]*11%256<=32||xx[i][j]*11%256>='a'&&xx[i][j]*11%256<='z') continue;
```

```
else xx[i][j]=xx[i][j]*11%256;
```

```
}
```

```
void main()
```

```
{
```

```
clrscr();
```

```
if(ReadDat()){
```

```
printf("数据文件 ENG.IN 不能打开! \n\007");
```

```
return;
```

```
}
```

```
encryptChar();
```

```
WriteDat();
```

```
}
```

```
int ReadDat(void)
```

```
{
```

```
FILE *fp;
```

```

int i=0;
unsigned char *p;

if((fp=fopen("eng.in", "r"))==NULL) return 1;
while(fgets(xx[i], 80, fp)!=NULL){
    p=strchr(xx[i], '\n');
    if(p)*p=0;
    i++;
}
maxline=i;
fclose(fp);
return 0;
}

```

```

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("ps5.dat", "w");
    for(i=0; i<maxline; i++){
        printf("%s\n", xx[i]);
        fprintf(fp, "%s\n", xx[i]);
    }
    fclose(fp);
}

```

★题目 57（出圈题）

(本题采用实际上机题，05 年 9 月和 04 年 9 月实际上机题与下面相同)

设有 n 个人围坐一圈并按顺时针方向从 1 到 n 编号,从第 s 个人开始进行 1 到 m 的报数,报数到第 m 个人,此人出圈,再从他的下一个人重新开始 1 到 m 的报数,如此进行下去直到所有的人都出圈为止。现要求按出圈次序,给出这 n 个人的顺序表 p 。请考生编制函数 Josegh() 实现此功能并调用函数 WriteDat()把编号按照出圈的顺序输出到 OUT.DAT 文件中。
注意：第 1 个出圈的编号存放在 $p[0]$ 中,第 2 个出圈的编号存放在 $p[1]$ 中,直至第 n 个出圈的编号存放在 $p[n-1]$ 中。

设 $n = 100, s = 1, m = 10$ 进行编程。

注意：部分源程序存放在 PROG1.C 中。

请勿改动主函数 main() 和输出数据函数 WriteDat()的内容。

```

#include <stdio.h>
#define N 100
#define S 1
#define M 10

```

```

int p[100], n, s, m ;
void WriteDat(void) ;
void Josegh(void)
{ int i,j,s1,w,q[100];
  for(i=0;i<n;i++) q[i]=0;
  s1=s;
  for(i=1; i<=n; i++) p[i-1]=i;
  for(i=n; i>=2; i--)
  { s1=(s1+m-1)%i;
    if(s1==0) s1=i;
    w=p[s1-1];
    for(j=s1; j<i; j++) p[j-1]=p[j];
    p[i-1]=w;
  }
  for(i=0,j=n-1;i<n,j>=0;i++,j--) q[j]=p[i];
  for(i=0;i<n;i++)
    p[i]=q[i];
}
void main()
{
  m = M ;
  n = N ;
  s = S ;
  Josegh() ;
  WriteDat() ;
}
void WriteDat(void) {
  int i, j = 0 ;
  FILE *fp ;
  fp = fopen("out.dat", "w") ;
  for(i = 0 ; i <= N - 1 ; i++) {
    printf("%4d ", p[i]) ;
    fprintf(fp, "%4d", p[i]) ;
    j++ ;
    if(j % 10 == 0) {
      printf("\n") ;
      fprintf(fp, "\n") ;
    }
  }
  fclose(fp) ;
}

```

题目 58（选票问题）

现有一个 10 个人 100 行的选票数据文件 IN.DAT，其数据存放的格式是每条记录的长度均为 10 位，第一位表示第一个人的选中情况，第二位表示第二个人的选中情况，依此类推：内容均为字符 0 和 1，1 表示此人被选中，0 表示此人未被选中，若一张选票人数小于等于 5 个人时被认为无效的选票。给定函数 ReadDat()的功能是把选票数据读入到字符串数组 xx 中。请编制函数 CountRs()来统计每个人的选票数并把得票数依次存入 yy[0]到 yy[9]中。把结果 yy 输出到文件 OUT.DAT 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 writeDat()的内容。

```
#include <stdio.h>
```

```
char xx[100][11];
int yy[10];
int ReadDat(void);
void WriteDat(void);
```

```
void CountRs(void)
{ int i,j,count;
  for(i=0;i<100;i++)
  { count=0;
    for(j=0;j<10;j++)
      if(xx[i][j]=='1') count++;
    if(count<=5) continue;
    for(j=0;j<10;j++)
      if(xx[i][j]=='1') yy[j]++;
  }
}
```

```
void main()
{
  int i;

  for(i=0;i<10;i++)yy[i]=0;
  if(ReadDat()){
    printf("选票数据文件 IN.DAT 不能打开! \n\007");
    return;
  }
  CountRs();
  WriteDat();
}
```

```
int ReadDat(void)
{
  FILE *fp;
  int i;
```

```

if((fp=fopen("IN.DAT","r")) == NULL) return 1;
for(i=0;i<100;i++){
    if(fgets(xx[i],12,fp)==NULL)return 1;

    xx[i][10]='\0';
}

fclose(fp);
return 0;
}

```

```

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("OUT.DAT","w");
    for(i=0;i<10;i++){
        fprintf(fp,"%d\n",yy[i]);
        printf("第%d 个人的选票数=%d\n",i+1,yy[i]);
    }
    fclose(fp);
}

```

★题目 59（整数各位打散组合运算题）

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat()把这些数存入数组 a 中，请考生编制一函数 jsVal()，其功能是：把千位数字和十位数字重新组成一个新的十位数 ab（新十位数的十位数字是原四位数的千位数字，新十位数的个位数字是原四位数的十位数字），以及把个位数字和百位数字组成另一个新的十位数 cd（新十位数的十位数字是原四位数的个位数字，新十位数的个位数字是原四位数的百位数字），如果新组成的两个十位数 ab<cd，ab 必须是奇数且不能被 5 整除，cd 必须是偶数，同时两个新数的十位数字均不为零，则将满足此条件的四位数按从大到小的顺序存入数组 b 中，并要计算满足上述条件的四位数的个数 cnt。最后 main()函数调用写函数 writeDat()把结果 cnt 以及数组 b 中符合条件的四位数输出到 OUT.DAT 文件中。

注意：部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[200]，b[200]，已定义变量：cnt

请勿改动数据文件 IN.DAT 中的任何数据、主函数 main()、读函数 readDat()和写函数 writeDat()的内容。

```

#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0 ;

```



```
void jsVal()
{int i,thou,hun,ten,data,j;
 int ab,cd;
 for(i=0;i<MAX;i++)
 {thou=a[i]/1000; hun=a[i]/100%10;
 ten=a[i]%100/10; data=a[i]%10;
 ab=10*thou+ten; cd=10*data+hun;
 if(((ab-cd)<0&&ab%2==1&&ab%5!=0)&&cd%2!=1&&ab>=10&&cd>=10)
 {b[cnt]=a[i];cnt++;}
 }
 for(i=0;i<cnt-1;i++)
 for(j=i+1;j<cnt;j++)
 if(b[i]<b[j]) {data=b[i];b[i]=b[j];b[j]=data;}
}
```

```
void readDat()
{
 int i ;
 FILE *fp ;
 fp = fopen("in.dat", "r") ;
 for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
 fclose(fp) ;
}
```

```
void main()
{
 int i ;
 readDat() ;
 jsVal() ;
 printf("满足条件的数=%d\n", cnt) ;
 for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
 printf("\n") ;
 writeDat() ;
}
```

```
writeDat()
{
 FILE *fp ;
 int i ;
 fp = fopen("out.dat", "w") ;
 fprintf(fp, "%d\n", cnt) ;
 for(i = 0 ; i < cnt ; i++) fprintf(fp, "%d\n", b[i]) ;
 fclose(fp) ;
}
```

```
}

```

```
*****

```

题目 60（整数运算排序题）

在文件 in.dat 中有 200 组数据，每组有 3 个数，每个数均是三位数。函数 ReadDat() 读取这 200 组数据存放到结构数组 aa 中，请编制函数 jsSort()，其函数的功能是：要求在 200 组数据中找出条件为每组中的第二个数大于第一个数加第三个数的之和，其中满足条件的组数作为函数 jsSort() 的返回值，同时把满足条件的数据存入结构数组 bb 中，再对 bb 中的数据按照每组数据的第二个数加第三个之和的大小进行降序排列（第二个数加第三个数的和均不相等），排序后的结果仍重新存入结构数组 bb 中，最后调用函数 WriteDat() 把结果 bb 输出到文件 out.dat 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat() 和输出数据函数 WriteDat() 的内容。

```
#include <stdio.h>
#include <string.h>
#include <conio.h>

typedef struct{
    int x1,x2,x3;
}data;

data aa[200],bb[200];

int jsSort()
{int i, j, cnt=0;
 data ch;
 for(i=0;i<200;i++)
     if(aa[i].x2>aa[i].x1+aa[i].x3) bb[cnt++]=aa[i];
 for(i=0;i<cnt-1;i++)
     for(j=i+1;j<cnt;j++)
         if(bb[i].x2+bb[i].x3<bb[j].x2+bb[j].x3) { ch=bb[i];bb[i]=bb[j];bb[j]=ch;}
 return cnt;
}

void main()
{
    int count;

    readDat();
    count=jsSort(); /*返回满足条件的个数*/
    writeDat(count);
}

readDat()
```

```

{
    FILE *in;
    int i;

    in=fopen("in.dat","r");
    for(i=0; i<200; i++)
        fscanf(in,"%d %d %d",&aa[i].x1,&aa[i].x2,&aa[i].x3);
    fclose(in);
}

writeDat(int count)
{
    FILE *out;
    int i;

    out=fopen("out.dat","w");
    clrscr();
    for(i=0; i<count; i++){
        printf("%d,%d,%d 第二个数+第三个数=%d\n",bb[i].x1,bb[i].x2,bb[i].x3,bb[i].x2+bb[i].x3);
        fprintf(out,"%d %d %d\n",bb[i].x1,bb[i].x2,bb[i].x3);
    }
    fclose(out);
}

```

★题目 61（结构体运算题）

已知在文件 IN.DAT 中存有 100 个产品销售记录，每个产品销售记录由产品代码 dm(字符型 4 位)，产品名称 mc(字符型 10 位)，单价 dj(整型)，数量 sl(整型)，金额 je(长整型)五部分组成。其中：金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat()，其功能要求：按金额从小到大进行排列，若金额相同，则按产品代码从大到小进行排列，最终排列结果仍存入结构数组 sell 中，最后调用函数 WriteDat()把结果输出到文件 OUT2.DAT 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include <stdio.h>
#include <mem.h>
#include <string.h>
#include <conio.h>
#include <stdlib.h>

#define MAX 100
typedef struct{
    char dm[5]; /*产品代码*/

```

```
char mc[11]; /*产品名称*/
int dj;      /*单价*/
int sl;      /*数量*/
long je;     /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();

void SortDat()
{int i,j;
 PRO xy;
for(i=0;i<99;i++)
for(j=i+1;j<100;j++)
if(sell[i].je>sell[j].je||sell[i].je==sell[j].je&&strcmp(sell[i].dm,sell[j].dm)<0)
{xy=sell[i]; sell[i]=sell[j]; sell[j]=xy;}
}

void main()
{
memset(sell,0,sizeof(sell));
ReadDat();
SortDat();
WriteDat();
}

void ReadDat()
{
FILE *fp;
char str[80],ch[11];
int i;

fp=fopen("IN.DAT","r");
for(i=0;i<100;i++){
fgets(str,80,fp);
memcpy(sell[i].dm,str,4);
memcpy(sell[i].mc,str+4,10);
memcpy(ch,str+14,4);ch[4]=0;
sell[i].dj=atoi(ch);
memcpy(ch,str+18,5);ch[5]=0;
sell[i].sl=atoi(ch);
sell[i].je=(long)sell[i].dj*sell[i].sl;
}
fclose(fp);
```

```

}

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("OUT2.DAT","w");
    for(i=0;i<100;i++){
        printf("%s %s %4d %5d %5d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
        fprintf(fp,"%s %s %4d %5d %5d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
    }
    fclose(fp);
}

```

☆题目 62（字符替换题）

无忧 id 9 题条件微有不同

函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS2.DAT 中。

替代关系： $f(p)=p*13 \bmod 256$ （p 是数组中某一个字符的 ASCII 值，f(p)是计算后新字符的 ASCII 值），如果计算后 f(p)值小于等于 32 或其 ASCII 值是偶数，则该字符不变，否则将 f(p)所对应的字符进行替代。

部分源程序存在文件 prog1.c 中。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <ctype.h>

unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/

int ReadDat(void);
void WriteDat(void);

void encryptChar()
{ int i,j;
  for(i=0;i<maxline;i++)
    for(j=0;j<strlen(xx[i]);j++)
      if(xx[i][j]*13%256<=32||(xx[i][j]*13%256)%2==0) continue;
}

```

```
        else xx[i][j]=xx[i][j]*13%256;
    }

void main()
{
    clrscr();
    if(ReadDat()){
        printf("数据文件 ENG.IN 不能打开! \n\007");
        return;
    }
    encryptChar();
    WriteDat();
}

int ReadDat(void)
{
    FILE *fp;
    int i=0;
    unsigned char *p;

    if((fp=fopen("eng.in","r"))==NULL) return 1;
    while(fgets(xx[i],80,fp)!=NULL){
        p=strchr(xx[i],'\n');
        if(p)*p=0;
        i++;
    }
    maxline=i;
    fclose(fp);
    return 0;
}

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("ps2.dat","w");
    for(i=0;i<maxline;i++){
        printf("%s\n",xx[i]);
        fprintf(fp,"%s\n",xx[i]);
    }
    fclose(fp);
}
```

★题目 63（字符串移位题）

程序 prog1.c 的功能是：把 s 字符串中的所有字符左移一个位置，串中的第一个字符移到最后。请考生编写函数 chg(char *s)实现程序要求，最后调用函数 readwriteDAT()把结果输出到 bc2.out 文件中。

例如：s 字符串中原有内容为：Mn.123xyZ，则调用该函数后，结果为：n.123xyZM。

注意：部分源程序存在文件 prog1.c 文件中。

请勿改动主函数 main()和输出数据函数 readwriteDAT()的内容。

```
#include <conio.h>
#include <stdio.h>
#define N 81
void readwriteDAT();

void chg(char *s)
{ int i;
  char ch=*s;
  for(i=0;i<strlen(s)-1;i++)
    s[i]=s[i+1];
  s[strlen(s)-1]=ch;
}

main()
{
  char a[ N ] ;
  clrscr();
  printf("Enter a string : "); gets(a);
  printf("The original string is : "); puts(a);
  chg(a);
  printf("The string after modified : ");
  puts(a);
  readwriteDAT();
}

void readwriteDAT()
{
  int i ;
  char a[N] ;
  FILE *rf, *wf;

  rf= fopen("bc2.in", "r") ;
  wf= fopen("bc2.out", "w") ;
  for(i = 0 ; i < 50 ; i++) {
    fscanf(rf, "%s", a);
    chg(a);
```

```

    fprintf(wf, "%s\n", a);
}
fclose(rf);
fclose(wf);
}

```

★☆题目 64（素数题）

下列程序的功能是：找出所有 100 以内（含 100）满足 I, I+4, I+10 都是素数的整数 I（I+10 也在 100 以内）的个数 cnt 以及这些 I 之和 sum。请编写函数 countValue()实现程序要求的功能，最后调用函数 writeDat()把结果 cnt 和 sum 输出到文件 out.dat 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()和输出数据函数 writeDat()的内容。

```

#include <stdio.h>
int cnt,sum;
int isPrime(int m)
{
    int i;
    for(i=2; i<m; i++)
        if(m % i ==0) return 0 ;
    return 1;
}

void countValue()
{int i;
 for(i=2;i<=90;i++)
    if(isPrime(i)&&isPrime(i+4)&&isPrime(i+10))
        {cnt++;sum+=i;}
}

void main()
{
    cnt=sum=0;
    countValue();
    printf("满足条件的整数的个数=%d\n",cnt);
    printf("满足条件的整数的和值=%d\n",sum);
    writeDat();
}

writeDat()
{
    FILE *fp;

```



```

fp=fopen("out.dat","w");
fprintf(fp,"%d\n%d\n",cnt,sum);
fclose(fp);
}

```

运行结果为：

满足条件的整数的个数=7

满足条件的整数的和值=201

☆题目 65（整数统计运算题）

已知在文件 IN.DAT 中存有若干个（个数<200）四位数字的正整数，函数 ReadDat()读取这若干个正整数并存入数组 xx 中。请编制函数 CalValue()，其功能要求：1、求出这文件中共有多少个正整数 totNum；2、求这些数中的各位数字之和是奇数的数的个数 totCnt，以及满足此条件的这些数的算术平均值 totPjz，最后调用函数 writeDat()把所求的结果输出到文件 OUT.DAT 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读函数 ReadDat()和写函数 writeDat()的内容。

```

#include <stdio.h>
#include <conio.h>
#define MAXNUM 200

int xx[MAXNUM];
int totNum=0; /*文件 IN.DAT 中共有多少个正整数*/
int totCnt=0; /*符合条件的正整数的个数*/
double totPjz=0.0; /*平均值*/

int ReadDat(void);
void writeDat(void);

void CalValue(void)
{ int i,qw,bw,sw,gw;
  for(i=0;i<MAXNUM;i++)
  if(xx[i]>0)
  { totNum++;
    qw=xx[i]/1000;   bw=xx[i]/100%10;
    sw=xx[i]%100/10; gw=xx[i]%10;
    if((qw+bw+sw+gw)%2) { totCnt++; totPjz+=xx[i];}
  }
  if(totCnt==0) totPjz=0;
  else totPjz/=totCnt;
}

void main()

```

```

{
    int i;
    clrscr();
    for(i=0;i<MAXNUM;i++)xx[i]=0;
    if(ReadDat()){
        printf("数据文件 IN.DAT 不能打开!\007\n");
        return;
    }
    CalValue();
    printf("文件 IN.DAT 中共有正整数=%d 个\n",totNum);
    printf("符合条件的正整数的个数=%d 个\n",totCnt);
    printf("平均值=%.2f\n",totPjz);
    writeDat();
}

```

```

int ReadDat(void)
{
    FILE *fp;
    int i=0;

    if((fp=fopen("IN.DAT","r"))==NULL) return 1;
    while(!feof(fp)){
        fscanf(fp,"%d",&xx[i++]);
    }
    fclose(fp);
    return 0;
}

```

```

void writeDat(void)
{
    FILE *fp;
    fp=fopen("OUT.DAT","w");
    fprintf(fp,"%d\n%d\n%.2f\n",totNum,totCnt,totPjz);
    fclose(fp);
}

```

★题目 66 (素数题)

下列程序 prog1.c 的功能是：计算 500-800 区间内素数的个数 cnt，并按所求素数的值从大到小的顺序，再计算其间隔加、减之和，即第 1 个素数-第 2 个素数+第 3 个素数-第 4 个素数+第 5 个素数...的值 sum。请编写函数 countValue()实现程序的要求，最后调用函数 writeDat()把结果 cnt 和 sum，输出到文件 OUT11.DAT 中。

部分源程序已给出。

请勿改动主函数 main()和输出数据函数 writeDat()的内容。

```
#include <stdio.h>
int cnt,sum;

int isP(int m)
{ int i;

  for(i=2;i<m;i++)
    if(m%i==0)return 0;
  return 1;
}

void countValue()
{int i,j,k=1;
  for(i=800;i>500;i--)
  if(isP(i))
  {
    cnt++;
    sum=sum+k*i;
    k=k*(-1); }
  }

void main()
{
  cnt=sum=0;

  countValue();
  printf("素数的个数=%d\n",cnt);
  printf("按要求计算的值=%d\n",sum);
  writeDat();
}

writeDat()
{
  FILE *fp;

  fp=fopen("OUT11.DAT","w");
  fprintf(fp,"%d\n%d\n",cnt,sum);
  fclose(fp);
}
```

题目 67（整数各位数字运算题）

下列程序的功能是：选出 5000 以下符合条件的自然数。条件是：千位数字与百位数字之和等于十位数字与个位数字之和，且千位数字与百位数字之和等于个位数字与千位数字之差的 10 倍。计算并输出这些四位自然数的个数 cnt 以及这些数的和 sum。请编写函数 countValue() 实现程序的要求，最后调用函数 writeDat() 把结果 cnt 和 sum，输出到文件 OUT13.DAT 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main() 和输出数据函数 writeDat() 的内容。

```
#include <stdio.h>

int cnt,sum;

void countValue()
{int i,thou,hun,ten,data;
 for(i=1000;i<5000;i++)
 { thou=i/1000;    hun=i%1000/100;
   ten=i%100/10;   data=i%10;
   if((thou+hun==ten+data&&thou+hun==(data-thou)*10) {cnt++;sum+=i;}}
}

void main()
{
 cnt=sum=0;

 countValue();
 printf("满足条件的自然数的个数=%d\n",cnt);
 printf("满足条件的自然数的个数之和=%d\n",sum);
 writeDat();
}

writeDat()
{
 FILE *fp;

 fp=fopen("OUT13.DAT","w");
 fprintf(fp,"%d\n%d\n",cnt,sum);
 fclose(fp);
}
```

☆题目 68（整数各位数字统计运算题）

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat() 把这些数存入数组 a 中，请考生编制一函数 jsVal()，其功能是：如果四位数各位上的数字均是奇数，则统计出满足此条件的个数 cnt 并把这些四位数按从大到小的顺序存入数组 b 中。最后 main() 函数调用写函数 writeDat() 把结果 cnt 以及数组 b 中符合条件的四位数输出到 OUT.DAT 文件中。

注意：部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[200]，b[200]，已定义变量：cnt

请勿改动数据文件 IN.DAT 中的任何数据、主函数 main()、读函数 readDat()和写函数 writeDat()的内容。

```
#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0 ;

void jsVal()
{int i,j,thou,hun,ten,data;
 for(i=0;i<200;i++)
  {thou=a[i]/1000; hun=a[i]%1000/100;
   ten=a[i]%100/10; data=a[i]%10;
   if((thou&&thou%2&&hun%2&&ten%2&&data%2) b[cnt++]=a[i];
  }
 for(i=0;i<cnt-1;i++)
  for(j=i+1;j<cnt;j++)
   if(b[i]<b[j]) {thou=b[i]; b[i]=b[j];b[j]=thou;}
}

void readDat()
{
 int i ;
 FILE *fp ;
 fp = fopen("in.dat", "r") ;
 for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
 fclose(fp) ;
}

void main()
{
 int i ;
 readDat() ;
 jsVal() ;
 printf("满足条件的数=%d\n", cnt) ;
 for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
 printf("\n") ;
 writeDat() ;
}

writeDat()
{
 FILE *fp ;
 int i ;
```

```

fp = fopen("out.dat", "w");
fprintf(fp, "%d\n", cnt);
for(i = 0; i < cnt; i++) fprintf(fp, "%d\n", b[i]);
fclose(fp);
}

```

题目 69（整数统计运算题）

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat()把这些数存入数组 a 中，请考生编制一函数 jsVal()，其功能是：把一个四位数的千位数字上的值减去百位数字上的值再减去十位数字上的值最后减去个位数字上的值，如果得出的值大于等于零且原四位数是奇数，则统计出满足此条件的个数 cnt 并把这些四位数按从小到大的顺序存入数组 b 中，最后调用写函数 writeDat()把结果 cnt 以及数组 b 中符合条件的四位数输出到 OUT.DAT 文件中。

注意：部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[200]，b[200]，已定义变量：cnt

请勿改动数据文件 IN.DAT 中的任何数据、主函数 main()、读函数 readDat()和写函数 writeDat()的内容。

```

#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0;

void jsVal()
{
    int i, j, thou, hun, ten, data;
    for(i=0; i<200; i++)
    {
        thou=a[i]/1000; hun=a[i]%1000/100;
        ten=a[i]%100/10; data=a[i]%10;
        if(thou-hun-ten-data>=0&& a[i]%2) b[cnt++]=a[i];
    }
    for(i=0; i<cnt-1; i++)
        for(j=i+1; j<cnt; j++)
            if(b[i]>b[j]) {thou=b[i]; b[i]=b[j]; b[j]=thou;}
}

void readDat()
{
    int i;
    FILE *fp;
    fp = fopen("in.dat", "r");
    for(i = 0; i < MAX; i++) fscanf(fp, "%d", &a[i]);
    fclose(fp);
}

```

```

void main()
{
    int i ;
    readDat() ;
    jsVal() ;
    printf("满足条件的数=%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
    printf("\n") ;
    writeDat() ;
}

writeDat()
{
    FILE *fp ;
    int i ;
    fp = fopen("out.dat", "w") ;
    fprintf(fp, "%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) fprintf(fp, "%d\n", b[i]) ;
    fclose(fp) ;
}

```

题目 70（整数统计运算题）

已知在文件 IN.DAT 中存有若干个（个数<200）四位数字的正整数，函数 ReadDat()读取这若干个正整数并存入数组 xx 中。请编制函数 CalValue()，其功能要求：1、求出这文件中共有多少个正整数 totNum；2、求这些数右移 1 位后，产生的新数是奇数的数的个数 totCnt，以及满足此条件的这些数（右移前的值）的算术平均值 totPjz，最后调用函数 writeDat()把所求的结果输出到文件 OUT.DAT 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读函数 ReadDat()和写函数 writeDat()的内容。

```

#include <stdio.h>
#include <conio.h>
#define MAXNUM 200

int xx[MAXNUM];
int totNum=0; /*文件 IN.DAT 中共有多少个正整数*/
int totCnt=0; /*符合条件的正整数的个数*/
double totPjz=0.0; /*平均值*/

int ReadDat(void);
void writeDat(void);

void CalValue(void)

```

```
{int i,data;
for(i=0;i<MAXNUM;i++)
    if(xx[i]>0)
    { totNum++;
      data=xx[i]>>1;
      if(data%2){totCnt++;totPjz+=xx[i];}
    }
if(totCnt==0) totPjz=0;
else totPjz/=totCnt;
}

void main()
{
    int i;
    clrscr();
    for(i=0;i<MAXNUM;i++)xx[i]=0;
    if(ReadDat()){
        printf("数据文件 IN.DAT 不能打开!\007\n");
        return;
    }
    CalValue();
    printf("文件 IN.DAT 中共有正整数=%d 个\n",totNum);
    printf("符合条件的正整数的个数=%d 个\n",totCnt);
    printf("平均值=%.2f\n",totPjz);
    writeDat();
}

int ReadDat(void)
{
    FILE *fp;
    int i=0;

    if((fp=fopen("IN.DAT","r"))==NULL) return 1;
    while(!feof(fp)){
        fscanf(fp,"%d",&xx[i++]);
    }
    fclose(fp);
    return 0;
}

void writeDat(void)
{
    FILE *fp;
```



```

fp=fopen("OUT.DAT","w");
fprintf(fp,"%d\n%d\n%.2f\n",totNum,totCnt,totPjz);
fclose(fp);
}

```

★题目 71（字符串左右排序题）

函数 ReadDat()实现从文件 in.dat 中读取 20 行数据存放到字符串数组 xx 中（每行字符串长度均小于 80）。请编制函数 jsSort(), 其函数的功能是：以行为单位对字符串按给定的条件进行排序，排序后的结果仍按行重新存入字符串数组 xx 中，最后调用函数 WriteDat()把结果 xx 输出到文件 out.dat 中。

条件：从字符串中间一分为二，左边部分按字符的 ASCII 值降序排序，右边部分按字符的 ASCII 值升序排序。如果原字符串长度为奇数，则最中间的字符不参加排序，字符仍放在原位置上。

```

例如：位置      0  1  2  3  4  5  6  7  8
源字符串    a  b  c  d  h  g  f  e
              1  2  3  4  9  8  7  6  5
则处理后字符串 d  c  b  a  e  f  g  h
              4  3  2  1  9  5  6  7  8

```

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include <stdio.h>
#include <string.h>
#include <conio.h>

char xx[20][80];

void jsSort()
{int i,j,k,trl,half;
 char ch;
 for(i=0;i<20;i++)
 { trl=strlen(xx[i]);
  half=trl/2;
  for(j=0;j<half-1;j++)
  for(k=j+1;k<half;k++)
    if(xx[i][j]<xx[i][k]) { ch=xx[i][j];xx[i][j]=xx[i][k];xx[i][k]=ch;}
  if(trl%2) half++;
  for(j=half;j<trl-1;j++)
  for(k=j+1;k<trl;k++)
    if(xx[i][j]>xx[i][k]) { ch=xx[i][j];xx[i][j]=xx[i][k];xx[i][k]=ch;}
 }
}

```

```

void main()
{
    readDat();
    jsSort();
    writeDat();
}

readDat()
{
    FILE *in;
    int i=0;
    char *p;

    in=fopen("in.dat","r");
    while(i<20&&fgets(xx[i],80,in)!=NULL){
        p=strchr(xx[i],'\n');
        if(p)*p=0;
        i++;
    }
    fclose(in);
}

writeDat()
{
    FILE *out;
    int i;

    clrscr();
    out=fopen("out.dat","w");
    for(i=0;i<20;i++){
        printf("%s\n",xx[i]);
        fprintf(out,"%s\n",xx[i]);
    }
    fclose(out);
}

```

题目 72（整数各位打散组合运算题）

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat()把这些数存入数组 a 中，请考生编制一函数 jsVal()，其功能是：把个位数字和千位数字重新组成一个新的十位数（新十位数的十位数字是原四位数的个位数字，新十位数的个位数字是原四位数的千位数字），以及把百位数字和十位数字组成另一个新的十位数（新十位数的十位数字是原四位数的百位数字，新十位数的个位数字是原四位数的十位数字），如果新组成的两个十位数均是

偶数并且两个十位数中至少有一个数能被 9 整除，同时两个新数的十位数字均不为零，则将满足此条件的四位数按从大到小的顺序存入数组 b 中，并要计算满足上述条件的四位数的个数 cnt。最后 main() 函数调用写函数 writeDat() 把结果 cnt 以及数组 b 中符合条件的四位数输出到 OUT.DAT 文件中。

注意：部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[200]，b[200]，已定义变量：cnt

请勿改动数据文件 IN.DAT 中的任何数据、主函数 main()、读函数 readDat() 和写函数 writeDat() 的内容。

```
#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0 ;

void jsVal()
{int i,j,thou,hun,ten,data,ab,cd;
for(i=0;i<200;i++)
{thou=a[i]/1000;    hun=a[i]/100%10;
ten=a[i]%100/10;   data=a[i]%10;
ab=10*data+thou;   cd=10*hun+ten;
if((ab%9==0||cd%9==0)&&ab%2==0&&cd%2==0&&hun!=0&&data!=0)
{b[cnt]=a[i];cnt++;}
}
for(i=0;i<cnt-1;i++)
for(j=i+1;j<cnt;j++)
if(b[i]<b[j]) {data=b[i];b[i]=b[j];b[j]=data;}
}

void readDat()
{
int i ;
FILE *fp ;
fp = fopen("in.dat", "r") ;
for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
fclose(fp) ;
}

void main()
{
int i ;
readDat() ;
jsVal() ;
printf("满足条件的数=%d\n", cnt) ;
for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
printf("\n") ;
writeDat() ;
}
```

```

}

writeDat()
{
    FILE *fp;
    int i;
    fp = fopen("out.dat", "w");
    fprintf(fp, "%d\n", cnt);
    for(i = 0; i < cnt; i++) fprintf(fp, "%d\n", b[i]);
    fclose(fp);
}

```

★题目 73 (字符替换题)

函数 ReadDat()实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 xx 中,请编制函数 ConverCharD(), 其函数的功能是: 以行为单位把字符串中的所有小写字母改写成该字母的上一个字母, 如果是字母 a, 则改写成字母 z。大写字母仍为大写字母, 小写字母仍为小写字母, 其它字符不变。最后把已处理的字符串仍按行重新存入字符串数组 xx 中, 最后调用函数 writeDat()把结果 xx 输出到文件 OUT4.DAT 中。

例: 原文: Adb.Bcdza

abck.LLhj

结果: Aca.Bbcyz

zabj.LLgi

原始数据文件存放的格式是: 每行的宽度均小于 80 个字符, 含标点符号和空格。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 writeDat()的内容。

```

#include <stdio.h>
#include <string.h>
#include <conio.h>

char xx[50][80];
int maxline=0;/*文章的总行数*/

int ReadDat(void);
void WriteDat(void);

void ConvertCharD(void)
{ int i,j;
  for(i=0;i<maxline;i++)
  { for(j=0;j<strlen(xx[i]);j++)
    if(xx[i][j]=='a') xx[i][j]='z';
    else if(xx[i][j]>='b' && xx[i][j]<='z') xx[i][j]-=1;
  }
}

```

```
void main()
{
    clrscr();
    if(ReadDat()){
        printf("数据文件 IN.DAT 不能打开! \n\007");
        return;
    }
    ConvertCharD();
    WriteDat();
}
```

```
int ReadDat(void)
{
    FILE *fp;
    int i=0;
    char *p;

    if((fp=fopen("IN.DAT","r"))==NULL) return 1;
    while(fgets(xx[i],80,fp)!=NULL){
        p=strchr(xx[i],'\n');
        if(p)*p=0;
        i++;
    }
    maxline=i;
    fclose(fp);
    return 0;
}
```

```
void WriteDat(void)
{
    FILE *fp;
    int i;

    clrscr();
    fp=fopen("OUT4.DAT","w");
    for(i=0;i<maxline;i++){
        printf("%s\n",xx[i]);
        fprintf(fp,"%s\n",xx[i]);
    }
    fclose(fp);
}
```

★题目 74（字符 ASCII 值移位替换题）

函数 ReadDat()实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 xx 中，请编制函数 StrCharJL()，其函数的功能是：以行为单位把字符串中的所有字符的 ASCII 值左移 4 位，低位补 0，如果左移后，其字符的 ASCII 值小于等于 32 或大于 100，则原字符保持不变，否则就把左移后的字符 ASCII 值再加上原字符 ASCII 值，得到新的字符仍存入原字符串对应的位置上。最后把已处理的字符串仍按行重新存入字符串数组 xx 中，最后调用函数 writeDat()把结果 xx 输出到文件 OUT7.DAT 中。

原始数据文件存放的格式是：每行的宽度均小于 80 个字符，含标点符号和空格。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 writeDat()的内容。

```
#include <stdio.h>
#include <string.h>
#include <conio.h>

char xx[50][80];
int maxline=0;/*文章的总行数*/

int ReadDat(void);
void WriteDat(void);

void StrCharJL(void)
{ int i,j;
  char c;
  for(i=0;i<maxline;i++)
    for(j=0;j<strlen(xx[i]);j++)
      { c=xx[i][j]<<4;
        if(!(c<=32||c>100)) xx[i][j]+=c;
      }
}

void main()
{
  clrscr();
  if(ReadDat()){
    printf("数据文件 IN.DAT 不能打开! \n\007");
    return;
  }
  StrCharJL();
  WriteDat();
}

int ReadDat(void)
{
  FILE *fp;
```

```

int i=0;
char *p;

if((fp=fopen("IN.DAT","r"))==NULL) return 1;
while(fgets(xx[i],80,fp)!=NULL){
    p=strchr(xx[i],'\n');
    if(p)*p=0;
    i++;
}
maxline=i;
fclose(fp);
return 0;
}

```

```

void WriteDat(void)
{
    FILE *fp;
    int i;

    clrscr();
    fp=fopen("OUT7.DAT","w");
    for(i=0;i<maxline;i++){
        printf("%s\n",xx[i]);
        fprintf(fp,"%s\n",xx[i]);
    }
    fclose(fp);
}

```

★☆题目 75 （整数各位数运算及排序题）

在文件 in.dat 中有 200 组数据，每组有 3 个数，每个数均是三位数。函数 ReadDat() 读取这 200 组数据存放在结构数组 aa 中，请编制函数 jsSort()，其函数的功能是：要求在 200 组数据中找出条件为每组中的第一个数大于第二个数加第三个数的和，其中满足条件的组数作为函数 jsSort() 的返回值，同时把满足条件的数据存入结构数组 bb 中，再对 bb 中的数据按照每组数据的第一个数加第三个之和的大小进行升序排列（第一个数加第三个数的和均不相等），排序后的结果仍重新存入结构数组 bb 中，最后调用函数 writeDat() 把结果 bb 输出到文件 out.dat 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat() 和输出数据函数 WriteDat() 的内容。

```

#include <stdio.h>
#include <string.h>
#include <conio.h>

```

```

typedef struct{
    int x1,x2,x3;

```

```
} data;

data aa[200],bb[200];

int jsSort()
{ int i,j,cnt=0;
  data xy;
  for(i=0;i<200;i++)
    if(aa[i].x1>(aa[i].x2+aa[i].x3)) bb[cnt++]=aa[i];
  for(i=0;i<cnt-1;i++)
    for(j=i+1;j<cnt;j++)
      if(bb[i].x1+bb[i].x3>bb[j].x1+bb[j].x3) { xy=bb[i]; bb[i]=bb[j]; bb[j]=xy;}
  return cnt;
}

void main()
{
  int count;

  readDat();
  count=jsSort(); /*返回满足条件的个数*/
  writeDat(count);
}

readDat()
{
  FILE *in;
  int i;
  in=fopen("in.dat","r");
  for(i=0; i<200; i++)
    fscanf(in,"%d %d %d",&aa[i].x1,&aa[i].x2,&aa[i].x3);
  fclose(in);
}

writeDat(int count)
{
  FILE *out;
  int i;

  clrscr();
  out=fopen("out.dat","w");
  for(i=0; i<count; i++){
    printf("%d,%d,%d 第一个数+第三个数=%d\n",bb[i].x1,bb[i].x2,bb[i].x3,bb[i].x1+bb[i].x3);
    fprintf(out,"%d %d %d\n",bb[i].x1,bb[i].x2,bb[i].x3);
  }
}
```



```

}
fclose(out);
}

```

★题目 76 (整数各位打散组合运算题)

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat()把这些数存入数组 a 中，请考生编制一函数 jsVal()，其功能是：把个位数字和千位数字重新组成一个新的十位数（新十位数的十位数字是原四位数的个位数字，新十位数的个位数字是原四位数的千位数字），以及把百位数字和十位数字组成另一个新的十位数（新十位数的十位数字是原四位数的百位数字，新十位数的个位数字是原四位数的十位数字），如果新组成的两个十位数必须是一个奇数，另一个为偶数并且两个十位数中至少有一个数能被 17 整除，同时两个新数的十位数字均不为零，则将满足此条件的四位数按从大到小的顺序存入数组 b 中，并要计算满足上述条件的四位数的个数 cnt。最后 main() 函数调用写函数 writeDat()把结果 cnt 以及数组 b 中符合条件的四位数输出到

OUT.DAT 文件中。

注意：部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[200]，b[200]，已定义变量：cnt

请勿改动数据文件 IN.DAT 中的任何数据、主函数 main()、读函数 readDat()和写函数 writeDat()的内容。

```

#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0 ;

void jsVal()
{int i,j,thou,hun,ten,data,ab,cd;
 for(i=0;i<MAX;i++)
 {thou=a[i]/1000;   hun=a[i]/100%10;
  ten=a[i]%100/10;  data=a[i]%10;
  ab=10*data+thou;  cd=10*hun+ten;
  if((ab%2!=cd%2)&&((ab%17==0||cd%17==0)&&ab>=10&&cd>=10))
    b[cnt++]=a[i];
 }
 for(i=0;i<cnt-1;i++)
  for(j=i+1;j<cnt;j++)
   if(b[i]<b[j]) {data=b[i];b[i]=b[j];b[j]=data;}
}

void readDat()
{
 int i ;
 FILE *fp ;
 fp = fopen("in.dat", "r") ;

```

```

    for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
    fclose(fp) ;
}

```

```

void main()
{
    int i ;
    readDat() ;
    jsVal() ;
    printf("满足条件的数=%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
    printf("\n") ;
    writeDat() ;
}

```

```

writeDat()
{
    FILE *fp ;
    int i ;
    fp = fopen("out.dat", "w") ;
    fprintf(fp, "%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) fprintf(fp, "%d\n", b[i]) ;
    fclose(fp) ;
}

```

★☆☆题目 77（整数各位运算题）

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat() 把这些数存入数组 a 中，请考生编制一函数 jsVal()，其功能是：把一个四位数的千位数字上的值加上十位数字上的值恰好等于百位数字上的值加上个位数字上的值，并且原四位数是偶数，则统计出满足此条件的个数 cnt 并把这些四位数按从小到大的顺序存入数组 b 中，最后调用写函数 writeDat() 把结果 cnt 以及数组 b 中符合条件的四位数输出到 OUT.DAT 文件中。

注意：部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[200]，b[200]，已定义变量：cnt

请勿改动数据文件 IN.DAT 中的任何数据、主函数 main()、读函数 readDat() 和写函数 writeDat() 的内容。

```

#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0 ;

```

```

void jsVal()
{int i,j,thou,hun,ten,data;

```

```

for(i=0;i<MAX;i++)
{
    thou=a[i]/1000;    hun=a[i]/100%10;
    ten=a[i]%100/10;   data=a[i]%10;
    if((thou+ten==hun+data)&& a[i]%2==0) b[cnt++]=a[i];
}
for(i=0;i<cnt-1;i++)
    for(j=i+1;j<cnt;j++)
        if(b[i]>b[j]) {thou=b[i];b[i]=b[j];b[j]=thou;}
}

```

```

void readDat()
{
    int i ;
    FILE *fp ;
    fp = fopen("in.dat", "r") ;
    for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
    fclose(fp) ;
}

```

```

void main()
{
    int i ;
    readDat() ;
    jsVal() ;
    printf("满足条件的数=%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
    printf("\n") ;
    writeDat() ;
}

```

```

writeDat()
{
    FILE *fp ;
    int i ;
    fp = fopen("out.dat", "w") ;
    fprintf(fp, "%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) fprintf(fp, "%d\n", b[i]) ;
    fclose(fp) ;
}

```

★题目 78（整数各位统计排序题）

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat()把这些数存入数组 a

中，请考生编制一函数 `jsVal()`，其功能是：若一个四位数的千位数字上的值大于等于百位数字上的值，百位数字上的值大于等于十位数字上的值，以及十位数字上的值大于等于个位数字上的值，并且原四位数是奇数，则统计出满足此条件的个数 `cnt` 并把这些四位数按从小到大的顺序存入数组 `b` 中，最后调用写函数 `writeDat()` 把结果 `cnt` 以及数组 `b` 中符合条件的数输出到 `OUT.DAT` 文件中。

注意：部分源程序存在文件 `prog1.c` 中。

程序中已定义数组：`a[200]`，`b[200]`，已定义变量：`cnt`

请勿改动数据文件 `IN.DAT` 中的任何数据、主函数 `main()`、读函数 `readDat()` 和写函数 `writeDat()` 的内容。

```
#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0 ;

void jsVal()
{int i,j,thou,hun,ten,data;
 for(i=0;i<MAX;i++)
 {thou=a[i]/1000;   hun=a[i]/100%10;
  ten=a[i]%100/10;  data=a[i]%10;
  if((thou>=hun&&hun>=ten&&ten>=data&&a[i]%2)) b[cnt++]=a[i];
 }
 for(i=0;i<cnt-1;i++)
  for(j=i+1;j<cnt;j++)
   if(b[i]>b[j]) {thou=b[i];b[i]=b[j];b[j]=thou;}
}

void readDat()
{
 int i ;
 FILE *fp ;
 fp = fopen("in.dat", "r") ;
 for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
 fclose(fp) ;
}

void main()
{
 int i ;
 readDat() ;
 jsVal() ;
 printf("满足条件的数=%d\n", cnt) ;
 for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
 printf("\n") ;
 writeDat() ;
}
```

```

writeDat()
{
    FILE *fp;
    int i;
    fp = fopen("out.dat", "w");
    fprintf(fp, "%d\n", cnt);
    for(i = 0; i < cnt; i++) fprintf(fp, "%d\n", b[i]);
    fclose(fp);
}

```

★题目 79（数字运算题）

请编写函数 `void countValue(int *a,int *n)`，它的功能是：求出 1 到 1000 之内能被 7 或 11 整除但不能同时被 7 或 11 整除的所有整数放在数组 `a` 中，并通过 `n` 返回这些数的个数。

部分源程序存在文件 `prog1.c` 中。

请勿改动主函数 `main()` 和输出数据函数 `writeDAT()` 的内容。

```

#include <conio.h>
#include <stdio.h>

```

```

void countValue(int *a,int *n)
{ int i;
  *n=0;
  for(i=1;i<=1000;i++)
    if(i%7==0&& i%11||i%7&&i%11==0)
      a((*n)++)=i;
}

```

```

main()
{
    int aa[1000],n,k;

    clrscr();
    countValue(aa,&n);
    for(k=0;k<n;k++)
        if((k+1)%10==0)printf("\n");
        else printf("%5d",aa[k]);
    writeDAT();
}

```

```

writeDAT()
{
    int aa[1000],n,k;

```

```

FILE *fp;

fp=fopen("out19.dat","w");
countValue(aa,&n);
for(k=0;k<n;k++)
    if((k+1)%10==0) fprintf(fp,"\n");
    else fprintf(fp,"%5d",aa[k]);
fprintf(fp,"\n");
fclose(fp);
}

```

题目 80（整数各位数运算题）

已知数据文件 in.dat 中存有 300 个四位数，并已调用读函数 ReadDat()把这些数存入数组 a 中，请编制一函数 jsValue()，其功能是：求出个位数上的数减千位数上的数减百位数上的数减十位数上的数大于零的个数 cnt，再求出所有满足此条件的四位数平均值 pjz1，以及不满足此条件的四位数平均值 pjz2，最后调用写函数 writeDat()把结果输出到 out.dat 文件。

例如：1239，9-1-2-3>0，则该数满足条件计算平均值 pjz1，且个数 cnt=cnt+1。

8129，9-8-1-2<0，则该数不满足条件计算平均值 pjz2。

部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[300]，已定义变量：cnt,pjz1,pjz2

请勿改动主函数 main()、读函数 ReadDat()和写函数 writeDat()的内容。

```

#include <stdio.h>
int a[300],cnt=0;
double pjz1=0.0,pjz2=0.0;

jsValue()
{int i,j,thou,hun,ten,data,n=0;
for(i=0;i<300;i++)
    {thou=a[i]/1000;    hun=a[i]/100%10;
ten=a[i]%100/10;    data=a[i]%10;
    if(data-thou-hun-ten>0) { cnt++; pjz1+=a[i];}
    else { n++; pjz2+=a[i];}
    }
if(cnt==0) pjz1=0;
else pjz1/=cnt;
if(n==0) pjz2=0;
else pjz2/=n;
}

main()
{
int i;

```

```

readDat();
jsValue();
writeDat();
printf("cnt=%d\n 满足条件的平均值 pjz1=%7.2f\n 不满足条件的平均值
pjz2=%7.2f\n",cnt,pjz1,pjz2);
}

```

```

readDat()
{
    FILE *fp;
    int i;
    fp=fopen("in.dat","r");
    for(i=0;i<300;i++)fscanf(fp,"%d",&a[i]);
    fclose(fp);
}

```

```

writeDat()
{
    FILE *fp;
    int i;
    fp=fopen("out.dat","w");
    fprintf(fp,"%d\n%7.2f\n%7.2f\n",cnt,pjz1,pjz2);
    fclose(fp);
}

```

☆题目 81（整数各位数运算题）

已知数据文件 in.dat 中存有 300 个四位数，并已调用读函数 ReadDat()把这些数存入数组 a 中，请编制一函数 jsValue()，其功能是：求出千位数上的数加百位数等于十位数上的数加个位数上的数的个数 cnt，再求出所有满足此条件的四位数平均值 pjz1，以及不满足此条件的四位数平均值 pjz2，最后调用写函数 writeDat()把结果输出到 out.dat 文件。

例如：7153,7+1=5+3,则该数满足条件计算平均值 pjz1,且个数 cnt=cnt+1。8129,8+1<2+9,则该数不满足条件计算平均值 pjz2。

部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[300]，已定义变量：cnt,pjz1,pjz2。

请勿改动主函数 main()、读函数 ReadDat()和写函数 writeDat()的内容。

```

#include <stdio.h>
int a[300],cnt=0;
double pjz1=0.0,pjz2=0.0;

jsValue()
{int i,thou,hun,ten,data,n=0;

```

```

for(i=0;i<300;i++)
{thou=a[i]/1000;  hun=a[i]%1000/100;
 ten=a[i]%100/10;  data=a[i]%10;
 if(thou+hun==ten+data) {cnt++;pjz1+=a[i];}
 else {n++;pjz2+=a[i];}
}
if(cnt==0) pjz1=0;
else pjz1/=cnt;
if(n==0) pjz2=0;
else pjz2/=n;
}

main()
{
    int i;

    readDat();
    jsValue();
    writeDat();
    printf("cnt=%d\n 满足条件的平均值 pjz1=%7.2f\n 不满足条件的平均值
    pjz2=%7.2f\n",cnt,pjz1,pjz2);
}

readDat()
{
    FILE *fp;
    int i;
    fp=fopen("in.dat","r");
    for(i=0;i<300;i++)fscanf(fp,"%d",&a[i]);
    fclose(fp);
}

writeDat()
{
    FILE *fp;
    int i;
    fp=fopen("out.dat","w");
    fprintf(fp,"%d\n%7.2f\n%7.2f\n",cnt,pjz1,pjz2);
    fclose(fp);
}

```

★题目 82（素数题）

已知数据文件 in.dat 中存有 300 个四位数，并已调用读函数 ReadDat()把这些数存入数组 a

中，请编制一函数 `jsValue()`，其功能是：求出所有这些四位数是素数的个数 `cnt`，再求出所有满足此条件的四位数平均值 `pjz1`，以及不满足此条件的四位数平均值 `pjz2`，最后调用写函数 `writeDat()` 把结果 `cnt,pjz1,pjz2` 输出到 `out.dat` 文件。

例如：5591 是素数，则该数满足条件计算平均值 `pjz1`，且个数 `cnt=cnt+1`。

9812 是非素数，则该数不满足条件计算平均值 `pjz2`。

部分源程序存在文件 `prog1.c` 中。

程序中已定义数组：`a[300]`，已定义变量：`cnt,pjz1,pjz2`

请勿改动主函数 `main()`、读函数 `ReadDat()` 和写函数 `writeDat()` 的内容。

```
#include <stdio.h>
int a[300],cnt=0;
double pjz1=0.0,pjz2=0.0;

int isP(int m)
{int i;

    for(i=2;i<m;i++)
        if(m%i==0)return 0;
    return 1;
}

jsValue()
{int i;
    for(i=0;i<300;i++)
        if(isP(a[i])) {pjz1+=a[i];cnt++;}
        else pjz2+=a[i];
    if(cnt==0) pjz1=0;
    else pjz1/=cnt;
    if(300-cnt==0) pjz2=0;
    else pjz2/=(300-cnt);
}

main()
{
    int i;

    readDat();
    jsValue();
    writeDat();
    printf("cnt=%d\n 满足条件的平均值 pjz1=%7.2f\n 不满足条件的平均值 pjz2=%7.2f\n",cnt,pjz1,pjz2);
}

readDat()
```

```

{
    FILE *fp;
    int i;
    fp=fopen("in.dat","r");
    for(i=0;i<300;i++)fscanf(fp,"%d",&a[i]);
    fclose(fp);
}

writeDat()
{
    FILE *fp;
    int i;
    fp=fopen("out.dat","w");
    fprintf(fp,"%d\n%7.2f\n%7.2f\n",cnt,pjz1,pjz2);
    fclose(fp);
}

```

☆题目 83（方差运算题）

请编制函数 ReadDat()实现从文件 IN.DAT 中读取 1000 个十进制整数到数组 xx 中；请编制函数 Compute()分别计算出 xx 中奇数的个数 odd，奇数的平均值 ave1，偶数的平均值 ave2 以及所有奇数的方差 totfc 的值，最后调用函数 WriteDat()把结果输出到 OUT.DAT 文件中。

计算方差的公式如下：

$$\text{totfc} = 1/N \sum_{i=1}^N (\text{xx}[i] - \text{ave1})^2$$

设 N 为奇数的个数，xx[i]为奇数，ave1 为奇数的平均值。

原始数据文件存放的格式是：每行存放 10 个数，并用逗号隔开。（每个数均大于 0 且小于等于 2000）

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()和输出数据函数 writeDat()的内容。

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 1000

int xx[MAX],odd=0,even=0;
double ave1=0.0,ave2=0.0,totfc=0.0;
void WriteDat(void);

int ReadDat(void)
{
    int i;

```

```
FILE *fp;

if((fp=fopen("IN.DAT","r"))==NULL) return 1;

/*****编制函数 ReadDat()的部分*****/
for(i=0;i<MAX;i++)
{fscanf(fp,"%d",&xx[i]);
 if((i+1)%10==0)
  fscanf(fp,"\n");
}
/*****/

fclose(fp);
return 0;
}

void Compute(void)
{ int i,yy[MAX];
  for(i=0;i<MAX;i++)
    yy[i]=0;
  for(i=0;i<MAX;i++)
    if(xx[i]%2) { yy[odd++]=xx[i]; ave1+=xx[i];}
    else { even++; ave2+=xx[i];}
  if(odd==0) ave1=0;
  else ave1/=odd;
  if(even==0) ave2=0;
  else ave2/=even;
  for(i=0;i<odd;i++)
    totfc+=(yy[i]-ave1)*(yy[i]-ave1)/odd;
}

void main()
{
  int i;

  for(i=0;i<MAX;i++)xx[i]=0;
  if(ReadDat()){
    printf("数据文件 IN.DAT 不能打开!\007\n");
    return;
  }
  Compute();
  printf("ODD=%d\nAVE1=%f\nAVE2=%f\nTOTFC=%f\n",odd,ave1,ave2,totfc);
  WriteDat();
}
```

```

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("OUT.DAT","w");
    fprintf(fp,"%d\n%f\n%f\n%f\n",odd,ave1,ave2,totfc);
    fclose(fp);
}

```

☆题目 84（选票问题）

现有一个 10 个人 100 行的选票数据文件 IN.DAT，其数据存放的格式是每条记录的长度均为 10 位，第一位表示第一个人的选中情况，第二位表示第二个人的选中情况，依此类推：内容均为字符 0 和 1，1 表示此人被选中，0 表示此人未被选中，若一张选票人数大于 5 个人时被认为无效的选票。给定函数 ReadDat()的功能是把选票数据读入到字符串数组 xx 中。请编制函数 CountRs()来统计每个人的选票数并把得票数依次存入 yy[0]到 yy[9]中。把结果 yy 输出到文件 OUT.DAT 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 writeDat()的内容。

```
#include <stdio.h>
```

```

char xx[100][11];
int yy[10];
int ReadDat(void);
void WriteDat(void);

```

```

void CountRs(void)
{ int i,j,count;
  for(i=0;i<100;i++)
  { count=0;
    for(j=0;j<10;j++)
      if(xx[i][j]=='1') count++;
    if(count>5) continue;
    for(j=0;j<10;j++)
      if(xx[i][j]=='1') yy[j]++;
  }
}

```

```

void main()
{
    int i;

```

```

    for(i=0;i<10;i++)yy[i]=0;
    if(ReadDat()){
        printf("选票数据文件 IN.DAT 不能打开! \n\007");
        return;
    }
    CountRs();
    WriteDat();
}

int ReadDat(void)
{
    FILE *fp;
    int i;

    if((fp=fopen("IN.DAT","r")) == NULL) return 1;
    for(i=0;i<100;i++){
        if(fgets(xx[i],12,fp)==NULL)return 1;
        xx[i][10]='\0';
    }

    fclose(fp);
    return 0;
}

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("OUT.DAT","w");
    for(i=0;i<10;i++){
        fprintf(fp,"%d\n",yy[i]);
        printf("第%d 个人的选票数=%d\n",i+1,yy[i]);
    }
    fclose(fp);
}

```

☆题目 85（整数各位数运算题）

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat()把这些数存入数组 a 中，请考生编制一函数 jsVal()，其功能是：把一个四位数的个位数字上的值减去千位数字上的值再减去百位数字上的值最后减去十位数字上的值，如果得出的值大于等于零且原四位数

是偶数，则统计出满足此条件的个数 `cnt` 并把这些四位数按从小到大的顺序存入数组 `b` 中，最后调用写函数 `writeDat()` 把结果 `cnt` 以及数组 `b` 中符合条件的四位数输出到 `OUT.DAT` 文件中。

注意：部分源程序存在文件 `prog1.c` 中。

程序中已定义数组：`a[200]`，`b[200]`，已定义变量：`cnt`

请勿改动数据文件 `IN.DAT` 中的任何数据、主函数 `main()`、读函数 `readDat()` 和写函数 `writeDat()` 的内容。

```
#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0 ;

void jsVal()
{int i,j,thou,hun,ten,data;
 for(i=0;i<MAX;i++)
  {thou=a[i]/1000;   hun=a[i]/100%10;
   ten=a[i]%100/10;  data=a[i]%10;
   if(data-thou-hun-ten>=0&& a[i]%2==0)  b[cnt++]=a[i];
  }
 for(i=0;i<cnt-1;i++)
  for(j=i+1;j<cnt;j++)
   if(b[i]>b[j]) {thou=b[i];b[i]=b[j];b[j]=thou;}
}

void readDat()
{
 int i ;
 FILE *fp ;
 fp = fopen("in.dat", "r") ;
 for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
 fclose(fp) ;
}

void main()
{
 int i ;
 readDat() ;
 jsVal() ;
 printf("满足条件的数=%d\n", cnt) ;
 for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
 printf("\n") ;
 writeDat() ;
}
```

```

writeDat()
{
    FILE *fp;
    int i;
    fp = fopen("out.dat", "w");
    fprintf(fp, "%d\n", cnt);
    for(i = 0; i < cnt; i++) fprintf(fp, "%d\n", b[i]);
    fclose(fp);
}

```

题目 86 (整数各位打散组合运算题)

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat() 把这些数存入数组 a 中，请考生编制一函数 jsVal()，其功能是：把千位数字和十位数字重新组成一个新的十位数 ab (新十位数的十位数字是原四位数的千位数字，新十位数的个位数字是原四位数的十位数字)，以及把个位数字和百位数字组成另一个新的十位数 cd (新十位数的十位数字是原四位数的个位数字，新十位数的个位数字是原四位数的百位数字)，如果新组成的两个十位数 ab>cd，ab 必须是偶数且能被 5 整除，cd 必须是奇数，同时两个新数的十位数字均不为零，则将满足此条件的四位数按从大到小的顺序存入数组 b 中，并要计算满足上述条件的四位数的个数 cnt。最后 main() 函数调用写函数 writeDat() 把结果 cnt 以及数组 b 中符合条件的四位数输出到 OUT.DAT 文件中。

注意：部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[200]，b[200]，已定义变量：cnt

请勿改动数据文件 IN.DAT 中的任何数据、主函数 main()、读函数 readDat() 和写函数 writeDat() 的内容。

```

#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0;

void jsVal()
{
    int i, j, thou, hun, ten, data, ab, cd;
    for(i=0; i<MAX; i++)
    {
        thou=a[i]/1000;    hun=a[i]%1000/100;
        ten=a[i]%100/10;   data=a[i]%10;
        ab=10*thou+ten;    cd=10*data+hun;
        if((ab-cd)>0&&(ab%2!=1&&ab%5==0)&&cd%2==1&&ab>=10&&cd>=10)
            {b[cnt]=a[i]; cnt++;}
    }
    for(i=0; i<cnt-1; i++)
        for(j=i+1; j<cnt; j++)
            if(b[i]<b[j]) {data=b[i]; b[i]=b[j]; b[j]=data;}
}

```

```

void readDat()
{
    int i ;
    FILE *fp ;
    fp = fopen("in.dat", "r") ;
    for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
    fclose(fp) ;
}

```

```

void main()
{
    int i ;
    readDat() ;
    jsVal() ;
    printf("满足条件的数=%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
    printf("\n") ;
    writeDat() ;
}

```

```

writeDat()
{
    FILE *fp ;
    int i ;
    fp = fopen("out.dat", "w") ;
    fprintf(fp, "%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) fprintf(fp, "%d\n", b[i]) ;
    fclose(fp) ;
}

```

☆题目 87（整数各位运算统计题）

已知数据文件 in.dat 中存有 300 个四位数，并已调用读函数 ReadDat()把这些数存入数组 a 中，请编制一函数 jsValue()，其功能是：求出千位数上的数加个位数等于百位数上的数加十位数上的数的个数 cnt，再求出所有满足此条件的四位数平均值 pjz1，以及不满足此条件的四位数平均值 pjz2，最后调用写函数把结果输出到 out.dat 文件。

例如：6712,6+2=7+1,则该数满足条件计算平均值 pjz1,且个数 cnt=cnt+1。8129,8+9<>1+2,则该数不满足条件计算平均值 pjz2。

部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[300]，已定义变量：cnt,pjz1,pjz2。

请勿改动主函数 main()、读函数 ReadDat()和写函数 writeDat()的内容。

```
#include <stdio.h>
```

```
int a[300],cnt=0;
```



```
double pjz1=0.0,pjz2=0.0;
```

```
jsValue()
```

```
{ int i,gw,sw,bw,qw,cnt2=0;
  for(i=0;i<300;i++)
  { gw=a[i]%10;      sw=a[i]/10%10;
    bw=a[i]/100%10;  qw=a[i]/1000;
    if((qw+gw)==(sw+bw)) { cnt++;pjz1+=a[i]; }
    else { cnt2++;pjz2+=a[i]; }
  }
  if(cnt==0) pjz1=0;
  else pjz1/=cnt;
  if(cnt2==0) pjz2=0;
  else pjz2/=cnt2;
}
```

```
main()
```

```
{
  int i;

  readDat();
  jsValue();
  writeDat();
  printf("cnt=%d\n 满足条件的平均值 pzj1=%7.2f\n 不满足条件的平均值
pzj2=%7.2f\n",cnt,pjz1,pjz2);
}
```

```
readDat()
```

```
{
  FILE *fp;
  int i;
  fp=fopen("in.dat","r");
  for(i=0;i<300;i++)fscanf(fp,"%d",&a[i]);
  fclose(fp);
}
```

```
writeDat()
```

```
{
  FILE *fp;
  int i;
  fp=fopen("out.dat","w");
  fprintf(fp,"%d\n%7.2f\n%7.2f\n",cnt,pjz1,pjz2);
  fclose(fp);
}
```

```
}

```

```
*****

```

题目 88（字符串排序题）

函数 ReadDat()实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 xx 中，请编制函数 SortCharA()，其函数的功能是：以行为单位对字符按从小到大的顺序进行排序，排序后的结果仍按行重新存入字符串数组 xx 中，最后调用函数 writeDat()把结果 xx 输出到文件 OUT1.DAT 中。

例：原文： dAe,BfC.

CCbbAA

结果： ,.ABCdef

AACCbb

原始数据文件存放的格式是：每行的宽度均小于 80 个字符，含标点符号和空格。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 writeDat()的内容。

```
#include <stdio.h>

```

```
#include <string.h>

```

```
#include <conio.h>

```

```
char xx[50][80];

```

```
int maxline=0;/*文章的总行数*/

```

```
int ReadDat(void);

```

```
void WriteDat(void);

```

```
void SortCharA(void)

```

```
{int i,j,k,strlen;

```

```
char ch;

```

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

```

```
{strlen=strlen(xx[i]);

```

```
for(j=0;j<strlen-1;j++)

```

```
for(k=j+1;k<strlen;k++)

```

```
if(xx[i][j]>xx[i][k])

```

```
{ch=xx[i][j]; xx[i][j]=xx[i][k]; xx[i][k]=ch;}

```

```
}

```

```
}

```

```
void main()

```

```
{

```

```
clrscr();

```

```
if(ReadDat()){

```

```
printf("数据文件 IN.DAT 不能打开！\n\007");

```

```
return;

```

```
}
SortCharA();
WriteDat();
}

int ReadDat(void)
{
    FILE *fp;
    int i=0;
    char *p;

    if((fp=fopen("IN.DAT","r"))==NULL) return 1;
    while(fgets(xx[i],80,fp)!=NULL){
        p=strchr(xx[i],'\n');
        if(p)*p=0;
        i++;
    }
    maxline=i;
    fclose(fp);
    return 0;
}

void WriteDat(void)
{
    FILE *fp;
    int i;

    clrscr();
    fp=fopen("OUT1.DAT","w");
    for(i=0;i<maxline;i++){
        printf("%s\n",xx[i]);
        fprintf(fp,"%s\n",xx[i]);
    }
    fclose(fp);
}
```

题目 89（整数统计排序题）

已知数据文件 IN.DAT 中存有 200 个四位数，并已调用读函数 readDat()把这些数存入数组 a 中，请考生编制一函数 jsVal()，其功能是：把一个四位数的千位数字上的值加上个位数字上的值恰好等于百位数字上的值加上十位数字上的值，并且原四位数是奇数，则统计出满足此条件的个数 cnt 并把这些四位数按从小到大的顺序存入数组 b 中，最后调用写函数 writeDat() 把结果 cnt 以及数组 b 中符合条件的四位数输出到 OUT.DAT 文件中。

注意：部分源程序存在文件 prog1.c 中。

程序中已定义数组：a[200]，b[200]，已定义变量：cnt

请勿改动数据文件 IN.DAT 中的任何数据、主函数 main()、读函数 readDat()和写函数 writeDat()的内容。

```
#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0 ;

void jsVal()
{
    int i,j,gw,sw,bw,qw;
    for(i=0;i<MAX;i++)
        {gw=a[i]%10;      sw=a[i]/10%10;
          bw=a[i]/100%10;  qw=a[i]/1000;
          if((qw+gw)==(sw+bw)&&a[i]%2)  b[cnt++]=a[i];
        }
    for(i=0;i<cnt-1;i++)
        for(j=i+1;j<cnt;j++)
            if(b[i]>b[j]) { qw=b[i]; b[i]=b[j]; b[j]=qw;}
}

void readDat()
{
    int i ;
    FILE *fp ;
    fp = fopen("in.dat", "r") ;
    for(i = 0 ; i < MAX ; i++) fscanf(fp, "%d", &a[i]) ;
    fclose(fp) ;
}

void main()
{
    int i ;
    readDat() ;
    jsVal() ;
    printf("满足条件的数=%d\n", cnt) ;
    for(i = 0 ; i < cnt ; i++) printf("%d ", b[i]) ;
    printf("\n") ;
    writeDat() ;
}

writeDat()
{
    FILE *fp ;
    int i ;
```

```

fp = fopen("out.dat", "w");
fprintf(fp, "%d\n", cnt);
for(i = 0; i < cnt; i++) fprintf(fp, "%d\n", b[i]);
fclose(fp);
}

```

★题目 90（整数排序题）

下列程序的功能是：将一正整数序列 {K1,K2,...,K9} 重新排列成一个新的序列。新序列中，比 K1 小的数都在 K1 的前面（左面），比 K1 大的数都在 K1 的后面（右面）。要求编写函数 jsValue()实现此功能，最后调用 writeDat()函数将新序列输出到文件 out.dat 中。

说明：在程序中已给出了 10 个序列，每个序列有 9 个正整数，并存入数组 a[10][9]中，分别求出这 10 个新序列。

例：序列 {6,8,9,1,2,5,4,7,3}

经重排后成为 {3,4,5,2,1,6,8,9,7}

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()和写函数 writeDat()的内容。

```
#include <stdio.h>
```

```

jsValue(int a[10][9])
{ int i,j,k,val,num;
  for(i=0;i<10;i++)
  { val=a[i][0];
    for(j=0;j<9;j++)
      if(a[i][j]<val)
      { num=a[i][j];
        for(k=j;k>0;k--)
          a[i][k]=a[i][k-1];
        a[i][0]=num;
      }
  }
}

```

```

main()
{
  int a[10][9]={ {6,8,9,1,2,5,4,7,3},
                  {3,5,8,9,1,2,6,4,7},
                  {8,2,1,9,3,5,4,6,7},
                  {3,5,1,2,9,8,6,7,4},
                  {4,7,8,9,1,2,5,3,6},
                  {4,7,3,5,1,2,6,8,9},
                  {9,1,3,5,8,6,2,4,7},
                  {2,6,1,9,8,3,5,7,4},
                  {5,3,7,9,1,8,2,6,4},

```

```

        {7,1,3,2,5,8,9,4,6},
    };

    int i,j;
    jsValue(a);
    for(i=0;i<10;i++){
        for(j=0;j<9;j++) {
            printf("%d",a[i][j]);
            if(j<=7)printf(",");
        }
        printf("\n");
    }
    writeDat(a);
}

```

```

writeDat(int a[10][9])
{
    FILE *fp;
    int i,j;

    fp=fopen("out.dat","w");
    for(i=0;i<10;i++){
        for(j=0;j<9;j++){
            printf("%d",a[i][j]);
            fprintf(fp,"%d",a[i][j]);
            if(j<=7) fprintf(fp,",");
        }
        printf("\n");
        fprintf(fp,"\n");
    }
    fclose(fp);
}

```

题目 91（整数统计运算题）

请编制程序 prog1.c，从文件 in.dat 中读取 200 个整数至数组 xx 中，求出奇数的个数 cnt1 和偶数的个数 cnt2 以及数组 xx 下标为偶数的元素值的算术平均值 pj（保留 2 位小数）。

结果 cnt1,cnt2,pj 输出到 out.dat 中。

部分程序、读数据函数 read_dat(int xx[200])及输出格式已给出。

```

#include <conio.h>
#include <stdio.h>
#define N 200

void read_dat(int xx[N])
{

```

```

int i,j;
FILE *fp;

fp=fopen("in.dat","r");
for(i=0;i<20;i++){
    for(j=0;j<10;j++){
        fscanf(fp,"%d",&xx[i*10+j]);
        printf("%d ",xx[i*10+j]);
    }
    printf("\n");
}
fclose(fp);
}

void main()
{
    int i,j,sum;
    int cnt1,cnt2,xx[N];
    float pj;
    FILE *fw;

    clrscr();
    fw=fopen("out.dat","w");
    read_dat(xx);
    /*****/
    sum=0;  pj=0.0;  cnt1=cnt2=0;
    for(i=0;i<N;i++)
    { if(xx[i]%2) cnt1++;
      else cnt2++;
      if(i%2==0) {pj+=xx[i];sum++;}
    }
    pj/=sum;

    /*****/

    printf("\n\ncnt1=%d,cnt2=%d,pj=%6.2f\n",cnt1,cnt2,pj);
    fprintf(fw,"%d\n%d\n%6.2f\n",cnt1,cnt2,pj);
    fclose(fw);
}

```

☆题目 92（整数统计运算题）

请编制程序 prog1.c，从文件 IN.DAT 中读取 200 个整数至数组 xx 中，求出最大数 max 及最

大数的个数 cnt 和数组 xx 中能被 3 整除或能被 7 整除的算术平均值 pj（保留 2 位小数）。

结果 max,cnt,pj 输出到 OUT.DAT 中。

部分程序、读数据函数 read_dat(int xx[200])及输出格式已给出。

```
#include <conio.h>
#include <stdio.h>
#define N 200

void read_dat(int xx[N])
{
    int i,j;
    FILE *fp;

    fp=fopen("in.dat","r");
    for(i=0;i<20;i++){
        for(j=0;j<10;j++){
            fscanf(fp,"%d",&xx[i*10+j]);
            printf("%d",xx[i*10+j]);
        }
        printf("\n");
    }
    fclose(fp);
}

void main()
{
    int m,temp,n,sum;
    int cnt,xx[N],max ;
    float pj;
    FILE *fw;

    clrscr();
    fw=fopen("out.dat","w");
    read_dat(xx);
    /***/
    cnt=0;  max=xx[0];  pj=0.0;  n=0;
    for(m=0;m<N;m++)
        if(max<xx[m]) max=xx[m];
    for(m=0;m<N;m++)
        {if(xx[m]==max) cnt++;
         if(xx[m]%3==0||xx[m]%7==0)
             { pj+=xx[m];  n++; }
        }
    pj/=n;
```



```

/*****

```

```

printf("\n\nmax=%d,cnt=%d,pj=%6.2f\n",max,cnt,pj);
fprintf(fw,"%d\n%d\n%6.2f\n",max,cnt,pj);
fclose(fw);
}

```

```

*****

```

☆题目 93（方差运算题）

请编制函数 ReadDat()实现从文件 IN.DAT 中读取 1000 个十进制整数到数组 xx 中；请编制函数 Compute()，分别计算出 xx 中奇数的个数 odd，偶数的个数 even，平均值 aver 以及方差 totfc 的值，最后调用函数 WriteDat()把结果输出到 OUT.DAT 文件中。

计算方差的公式如下：

$$\text{totfc} = 1/N \sum_{i=1}^N (\text{xx}[i] - \text{aver})^2$$

原始数据文件存放的格式是：每行存放 10 个数，并用逗号隔开。（每个数均大于 0 且小于等于 2000）

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()和输出数据函数 writeDat()的内容。

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 1000

int xx[MAX],odd=0,even=0;
double aver=0.0,totfc=0.0;
void WriteDat(void);

int ReadDat(void)
{
    int i;
    FILE *fp;

    if((fp=fopen("IN.DAT","r"))==NULL) return 1;
    /*****编制函数 ReadDat()*****/
    for(i=0;i<MAX;i++)
    { fscanf(fp,"%d",&xx[i]);
      if((i+1)%10==0)
        fscanf(fp,"\n");
    }
    /*****/
    fclose(fp);
    return 0;
}

```

```

}

void Compute(void)
{
    int i;
    for(i=0;i<MAX;i++)
    {
        if(xx[i]%2)    odd++;
        else    even++;
        aver+=xx[i];
    }
    aver/=MAX;
    for(i=0;i<MAX;i++)
        totfc+=(xx[i]-aver)*(xx[i]-aver);
    totfc/=MAX;
}

void main()
{
    int i;

    for(i=0;i<MAX;i++)xx[i]=0;
    if(ReadDat()){
        printf("数据文件 IN.DAT 不能打开!\007\n");
        return;
    }
    Compute();
    printf("ODD=%d\nOVEN=%d\nAVER=%f\nTOTFC=%f\n",odd,even,aver,totfc);
    WriteDat();
}

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("OUT.DAT","w");
    fprintf(fp,"%d\n%d\n%f\n%f\n",odd,even,aver,totfc);
    fclose(fp);
}

```

题目 94（整数统计运算题）

请编制程序 prog1.c，从文件 in.dat 中读取 200 个整数至数组 xx 中，求出奇数的个数 cnt1 和

偶数的个数 cnt2 以及数组 xx 中值为偶数的算术平均值 pj（保留 2 位小数）。

结果 cnt1,cnt2,pj 输出到 out.dat 中。

部分程序、读数据函数 read_dat(int xx[200])及输出格式已给出。

```
#include <conio.h>
#include <stdio.h>
#define N 200

void read_dat(int xx[N])
{
    int i,j;
    FILE *fp;

    fp=fopen("in.dat","r");
    for(i=0;i<20;i++){
        for(j=0;j<10;j++){
            fscanf(fp,"%d",&xx[i*10+j]);
            printf("%d ",xx[i*10+j]);
        }
        printf("\n");
    }
    fclose(fp);
}

void main()
{
    int m,sum;
    int cnt1,cnt2,xx[N];
    float pj;
    FILE *fw;

    fw=fopen("out.dat","w");
    clrscr();
    read_dat(xx);
    /***/
    cnt1=0; cnt2=0; pj=0.0;
    for(m=0;m<N;m++)
        if(xx[m]%2) cnt1++;
        else { cnt2++; pj+=xx[m];}
    if(cnt2==0) pj=0;
    else pj/=cnt2;
    /***/
    printf("\n\ncnt1=%d,cnt2=%d,pj=%6.2f\n",cnt1,cnt2,pj);
    fprintf(fw,"%d\n%d\n%6.2f\n",cnt1,cnt2,pj);
```

```
fclose(fw);
}
```

☆题目 95（字符替换题）

函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS4.DA 中。

替代关系： $f(p)=p*11 \bmod 256$ （p 是数组中某一个字符的 ASCII 值，f(p)是计算后新字符的 ASCII 值），如果计算后 f(p)值小于等于 32 或 f(p)对应的字符是大写字母，则该字符不变，否则将 f(p)所对应的字符进行替代。

部分源程序存在文件 prog1.c 中。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <ctype.h>
```

```
unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/
```

```
int ReadDat(void);
void WriteDat(void);
```

```
void encryptChar()
{ int i,j;
  for(i=0;i<maxline;i++)
    for(j=0;j<strlen(xx[i]);j++)
      if(xx[i][j]*11%256<=32||xx[i][j]*11%256>='A'&&xx[i][j]*11%256<='Z') continue;
      else xx[i][j]=xx[i][j]*11%256;
}
```

```
void main()
{
  clrscr();
  if(ReadDat()){
    printf("数据文件 ENG.IN 不能打开! \n\007");
    return;
  }
  encryptChar();
  WriteDat();
}
```

```

int ReadDat(void)
{
    FILE *fp;
    int i=0;
    unsigned char *p;

    if((fp=fopen("eng.in", "r"))==NULL) return 1;
    while(fgets(xx[i],80,fp)!=NULL){
        p=strchr(xx[i],'\n');
        if(p)*p=0;
        i++;
    }
    maxline=i;
    fclose(fp);
    return 0;
}

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("ps4.dat", "w");
    for(i=0; i<maxline; i++){
        printf("%s\n", xx[i]);
        fprintf(fp, "%s\n", xx[i]);
    }
    fclose(fp);
}

```

☆题目 96（字符替换题）

函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS5.DAT 中。

替代关系： $f(p)=p*11\text{mod } 256$ （p 是数组中某一个字符的 ASCII 值，f(p)是计算后新字符的 ASCII 值），如果原字符是小写字母或计算后 f(p)值小于等于 32，则该字符不变，否则将 f(p)所对应的字符进行替代。

部分源程序存在文件 prog1.c 中。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include <stdio.h>
```

```
#include <string.h>
#include <conio.h>
#include <ctype.h>

unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/

int ReadDat(void);
void WriteDat(void);

void encryptChar()
{ int i,j;
  for(i=0;i<maxline;i++)
    for(j=0;j<strlen(xx[i]);j++)
      if(xx[i][j]*11%256<=32||xx[i][j]>='a'&&xx[i][j]<='z') continue;
      else xx[i][j]=xx[i][j]*11%256;
}

void main()
{
  clrscr();
  if(ReadDat()){
    printf("数据文件 ENG.IN 不能打开! \n\007");
    return;
  }
  encryptChar();
  WriteDat();
}

int ReadDat(void)
{
  FILE *fp;
  int i=0;
  unsigned char *p;

  if((fp=fopen("eng.in","r"))==NULL) return 1;
  while(fgets(xx[i],80,fp)!=NULL){
    p=strchr(xx[i],'\n');
    if(p)*p=0;
    i++;
  }
  maxline=i;
  fclose(fp);
  return 0;
}
```

```

}

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("ps5.dat","w");
    for(i=0;i<maxline;i++){
        printf("%s\n",xx[i]);
        fprintf(fp,"%s\n",xx[i]);
    }
    fclose(fp);
}

```

★题目 97（字符替换题）

函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS9.DAT 中。

替代关系： $f(p)=p*11 \bmod 256$ （p 是数组中某一个字符的 ASCII 值，f(p)是计算后新字符的 ASCII 值），如果原字符是数字字符 0 至 9 或计算后 f(p)值小于等于 32，则该字符不变，否则将 f(p)所对应的字符进行替代。

部分源程序存在文件 prog1.c 中。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <ctype.h>

unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/

int ReadDat(void);
void WriteDat(void);

void encryptChar()
{ int i,j;
  for(i=0;i<maxline;i++)
    for(j=0;j<strlen(xx[i]);j++)
      if(xx[i][j]*11%256<=32||xx[i][j]>='0'&&xx[i][j]<='9') continue;
      else xx[i][j]=xx[i][j]*11%256;
}

```

```
}

void main()
{
    clrscr();
    if(ReadDat()){
        printf("数据文件 ENG.IN 不能打开! \n\007");
        return;
    }
    encryptChar();
    WriteDat();
}

int ReadDat(void)
{
    FILE *fp;
    int i=0;
    unsigned char *p;

    if((fp=fopen("eng.in", "r"))==NULL) return 1;
    while(fgets(xx[i], 80, fp)!=NULL){
        p=strchr(xx[i], '\n');
        if(p)*p=0;
        i++;
    }
    maxline=i;
    fclose(fp);
    return 0;
}

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("ps9.dat", "w");
    for(i=0; i<maxline; i++){
        printf("%s\n", xx[i]);
        fprintf(fp, "%s\n", xx[i]);
    }
    fclose(fp);
}
```

★题目 98（字符替题）

函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS3.DAT 中。

替代关系： $f(p)=p*17 \bmod 256$ （p 是数组中某一个字符的 ASCII 值，f(p)是计算后新字符的 ASCII 值），如果计算后 f(p)值小于等于 32 或其 ASCII 值是奇数，则该字符不变，否则将 f(p)所对应的字符进行替代。

部分源程序存在文件 prog1.c 中。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <ctype.h>

unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/

int ReadDat(void);
void WriteDat(void);

void encryptChar()
{ int i,j;
  for(i=0;i<maxline;i++)
    for(j=0;j<strlen(xx[i]);j++)
      if(xx[i][j]*17%256<=32||((xx[i][j]*17%256)%2!=0) continue;
      else xx[i][j]=xx[i][j]*17%256;
}

void main()
{
  clrscr();
  if(ReadDat()){
    printf("数据文件 ENG.IN 不能打开! \n\007");
    return;
  }
  encryptChar();
  WriteDat();
}

int ReadDat(void)
{
  FILE *fp;
```

```

int i=0;
unsigned char *p;

if((fp=fopen("eng.in", "r"))==NULL) return 1;
while(fgets(xx[i], 80, fp)!=NULL){
    p=strchr(xx[i], '\n');
    if(p)*p=0;
    i++;
}
maxline=i;
fclose(fp);
return 0;
}

```

```

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("ps3.dat", "w");
    for(i=0; i<maxline; i++){
        printf("%s\n", xx[i]);
        fprintf(fp, "%s\n", xx[i]);
    }
    fclose(fp);
}

```

题目 99（结构体排列题）

已知在文件 IN.DAT 中存有 100 个产品销售记录，每个产品销售记录由产品代码 dm(字符型 4 位)，产品名称 mc(字符型 10 位)，单价 dj(整型)，数量 sl(整型)，金额 je(长整型)五部分组成。其中：金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat()，其功能要求：按金额从小到大进行排列，若金额相等，则按产品代码从小到大进行排列，最终排列结果仍存入结构数组 sell 中，最后调用函数 WriteDat()把结果输出到文件 OUT1.DAT 中。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include <stdio.h>
#include <mem.h>
#include <string.h>
#include <conio.h>
#include <stdlib.h>

```

```

#define MAX 100

```

```
typedef struct{
    char dm[5]; /*产品代码*/
    char mc[11]; /*产品名称*/
    int dj; /*单价*/
    int sl; /*数量*/
    long je; /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();

void SortDat()
{int i,j;
 PRO xy;
 for(i=0;i<99;i++)
  for(j=i+1;j<100;j++)
   if(sell[i].je>sell[j].je||sell[i].je==sell[j].je&&strcmp(sell[i].dm,sell[j].dm)>0)
    {xy=sell[i];sell[i]=sell[j];sell[j]=xy;}
}

void main()
{
    memset(sell,0,sizeof(sell));
    ReadDat();
    SortDat();
    WriteDat();
}

void ReadDat()
{
    FILE *fp;
    char str[80],ch[11];
    int i;

    fp=fopen("IN.DAT","r");
    for(i=0;i<100;i++){
        fgets(str,80,fp);
        memcpy(sell[i].dm,str,4);
        memcpy(sell[i].mc,str+4,10);
        memcpy(ch,str+14,4);ch[4]=0;
        sell[i].dj=atoi(ch);
        memcpy(ch,str+18,5);ch[5]=0;
        sell[i].sl=atoi(ch);
        sell[i].je=(long)sell[i].dj*sell[i].sl;
    }
```

```

    }
    fclose(fp);
}

void WriteDat()
{
    FILE *fp;
    int i;

    fp=fopen("OUT1.DAT","w");
    for(i=0;i<100;i++){
        printf("%s %s %4d %5d %5d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
        fprintf(fp,"%s %s %4d %5d %5d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
    }
    fclose(fp);
}

```

★☆题目 100（字符替换题）

函数 ReadDat()实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 xx 中，请编制函数 CharConvA()，其函数的功能是：以行为单位把字符串中的最后一个字符的 ASCII 值右移 4 位，高位补 0 后加最后第二个字符的 ASCII 值，得到最后一个新的字符，最后第二个字符的 ASCII 值右移 4 位，高位补 0 后加最后第三个字符的 ASCII 值，得到最后第二个新的字符，依此类推一直处理到第二个字符，第一个字符的 ASCII 值加原最后一个字符的 ASCII 值，得到第一个新的字符，得到的新字符分别存放在原字符串对应的位置上。最后已处理的字符串仍按行重新存入字符串数组 xx 中，最后调用函数 writeDat()把结果 xx 输出到文件 OUT10.DAT 中。

原始数据文件存放的格式是：每行的宽度均小于 80 个字符，含标点符号和空格。

部分源程序存在文件 prog1.c 中。

请勿改动主函数 main()和写函数 writeDat()的内容。

```

#include <stdio.h>
#include <string.h>
#include <conio.h>

char xx[50][80];
int maxline=0;/*文章的总行数*/

int ReadDat(void);
void WriteDat(void);

void CharConvA()
{ int i,j;
  char p,c;
  for(i=0;i<maxline;i++)

```

```
        { p=xx[i][strlen(xx[i])-1];
          c=xx[i][0];
          for(j=strlen(xx[i])-1;j>0;j--)
              xx[i][j]=(xx[i][j]>>4)+xx[i][j-1];
          xx[i][0]=p+c;
        }
    }

void main()
{
    clrscr();
    if(ReadDat()){
        printf("数据文件 IN.DAT 不能打开! \n\007");
        return;
    }
    CharConvA();
    WriteDat();
}

int ReadDat(void)
{
    FILE *fp;
    int i=0;
    char *p;

    if((fp=fopen("IN.DAT","r"))==NULL) return 1;
    while(fgets(xx[i],80,fp)!=NULL){
        p=strchr(xx[i],'\n');
        if(p)*p=0;
        i++;
    }
    maxline=i;
    fclose(fp);
    return 0;
}

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("OUT10.DAT","w");
    for(i=0;i<maxline;i++){
        printf("%s\n",xx[i]);
    }
}
```

```
fprintf(fp,"%s\n",xx[i]);  
}  
fclose(fp);  
}
```

南开 100 题二级 C

题目 1

编一个函数 fun(char *s),函数的功能是把字符串中的内容逆置。

例子如：字符串中原有的内容为：abcdefg，则调用该函数后，串中的内容为：gfedcba

。

试题程序：

```
#include <string.h>
#include <conio.h>
#include <stdio.h>
#define N 81
/*注：该题的算法是先分别找出字符串的两头，然后同时逐一往中间移动，每移动一次
都进行两字符的位置对换，直到中间字符（用 s+i<s+n-1-i 来控制）。由于 s+i 中一个
地址，因此要注意把它的内容取出再进行换位。即先进行取内容运算*） */
fun(char *s)
{ int i=0,t,n=strlen(s);
  for(;s+i<s+n-1-i;i++)
  {t=*(s+i);*(s+i)=*(s+n-1-i);*(s+n-1-i)=t;}
}
main()
{ char a[N];
  clrscr();
  printf("Enter a string:"); gets(a);
  printf("The original string is:");puts(a);
  fun(a);
  printf("\n");
  printf("The string after modified:");
  puts(a);
}
```

题目 2

写程序，实现矩阵（3 行 3 列）的转置（即行列互换）。

例如，输入一面的矩阵：

程序输出：

试题程序：

```
#include <stdio.h>
#include <conio.h>
/*这题的关键在于进行行列下标转换的算法，由矩阵的对称性我们不难看出在进行行列
互换时 a[j] 在好是与 a[j][i] 互换，因而只要我位让程序走完矩阵的左上角即可（用
for(i=0;i<2;i++)再套 for(j=i+1;j<3;j++)来完成左上角的走动。*/
int fun(int array[3][3])
{ int i,j,t;
  for(i=0;i<2;i++)
```

```

for(j=i+1;j<3;j++)
{t=array[i][j];array[i][j]=array[j][i];array[j][i]=t;}
}
main()
{ int i,j;
int array[3][3]={ {100,200,300},{400,500,600},{700,800,900} };
clrscr();
for(i=0;i<3;i++)
{for(j=0;j<3;j++)
printf("%7d",array[i][j]);
printf("\n");
}
fun(array);
printf("Converted array:\n");
for(i=0;i<3;i++)
{for(j=0;j<3;j++)
printf("%7d",array[i][j]);
printf("\n");
}
}

```

题目 3

请编一个函数 fun(int *a,int n,int *odd,int *even)，函数的功能是分别求出数组中所有奇数之和以及所有偶数之和。形参 n 给了数组中数据的个数：利用指针 odd 返回奇数之和，利用指针 even 返回偶数之和。

例如：数组中的值依次为：1，8，2，3，11，6；则利用指针 odd 返回奇数之和 24；利用指针 even 返回偶数之和 8。

试题程序。

```

#include <stdio.h>
#include <conio.h>
#define N 20
/*注：该题的算法是：用 for() 循环一步一步地找元素，用 if(!a[i]%2) 来判断是否是奇数，% 运算是求余运算，当对 2 求余为 1 时表示原数为奇数，否则为偶数。*/
fun(int *a,int n,int *odd,int *even)
{ int i; *even=0;*odd=0;
for(i=0;i<n;i++)
if(!(a[i]%2))
*even+=a[i];
else
*odd+=a[i];
}

main()
{ int a[N]={1,9,2,3,11,6};i,n=6,odd,even;
clrscr();

```



```
printf("The original data is:\n");
for(i=0;i<n;i++) printf("%5d",*(a+i));
printf("\n\n");
fun(a,n,&odd,&even);
printf("The sum of odd numbers:%d\n",odd);
printf("The sum of even number:%d\n",even);
}
```

题目 4

要求程序的功能是：把 20 个随机数存入一个数组，然后输出该数组中的最小值。其中确定最小值的下标的操作在 fun 函数中实现，请给出该函数的定义。

试题程序。

```
#include <stdio.h>
#include <conio.h>
#define VSIZE 20
int vector[VSIZE];
/*注：该题的算法是用一个变量（j）来存储最小值元素的下标，在循环过程中让每个元素都与原最小值元素进行大小比较（if(list[i]<list[j])），如发现更小的则让 j 重新拥有最小值元素的下标（j=i）。*/
int fun(int list[],int size)
{ int i,j=0;
for(i=1;i<size;i++)
if(list[i]<list[j])
j=i;
return j;
}
main()
{
int i;
clrscr();
for(i=0;i<VSIZE;i++)
{
vector[i]=rand();
printf("Vector[%d]=%6d\n",i,vector[i]);
}
i=fun(vector,VSIZE);
printf("\nMininum:vector[%d]=%6d\n",i,vector[i]);
}
```

题目 5

请编一个函数 float fun(double h)，函数的功能是对变量 h 中的值保留 2 位小数，并对第三位进行四舍五入（规定 h 中的值为正数）。

例如：h 值为 8.32433，则函数返回 8.32；

h 值为 8.32533，则函数返回 8.33。

试题程序。

```
#include <stdio.h>
```

```
#include <conio.h>
```

/*注：h 乘以 1000 后正好是原小数点后第三位做了新数的个位数，然后再进行加 5 运算时。如原小数点后第三位为 4 及以下则加 5 后还是不能进一位（即四舍），如是 5 及以上则加 5 后该位就要向前进一位数（即五入）。进行加 5 运算后除 10 再赋给一个整型变量此时就只有原小数点第二位及以前各位保留在整型变量中，最后再对整型变量除 100，这样又出现了两位小数。该题中，进行四舍五入后一定要赋给一个整型变量才能将不用部分彻底变成 0。*/

```
float fun(float h)
{ long t;
h=h*1000;
t=(h+5)/10;
return (float)t/100;
}
main()
{
float a;
clrscr();
printf("Enter a:"); scanf("%f",&a);
printf("The original data is:");
printf("%f\n\n",a);
printf("The result :%6.2f\n",fun(a));
}
```

题目 6

编写函数 fun，函数的功能是：从字符串中删除指定的字符。同一字母的大、小写按不同字符处理。

若程序执行时，输入字符串为：turbo c and Borland c++

从键盘上输入字符：n，则输出后变为：turbo c ad borlad c++

如果输入的字符串不存在，则字符串照原样输出。

试题程序。

```
#include <stdio.h>
```

```
#include <conio.h>
```

/*注：该题的算法是让 i 控制一个一个字符往后走，在移动过程中如果 s[i]不是要删的字符，则将其按顺序放到新串中（新串亦是用 s 来做，只是用 k 来控制新串的下标，由于要删除一些元素，因此新串的下标总是比原下标 i 要慢。因而可用此法即同一字符串变量的方法。*/

```
int fun(char s[],int c)
{int i,k=0;
for(i=0;s[i];i++)
if(s[i]!=c) s[k++]=s[i];
s[k]='\0';
}
main()
{ static char str[]="turbo c and borland c++";
char ch;
```

```

clrscr();
printf(" :%s\n",str);
printf(" :");
scanf("%c",&ch);
fun(str,ch);
printf("str[]=%s\n",str);
}

```

题目 7

请编写一个 unsigned fun(unsigned w)，w 是一个大于 10 的无符号整数，若 w 是 n (n >=2) 位的整数，函数求出 w 的后 n-1 位的数作为函数值返回。

例如：w 值为 5923，则函数返回 923；w 值为 923 则函数返回 23。

试题程序。

```

#include <conio.h>
#include <stdio.h>
/* 注：由于 unsigned 型整数在 0-65535 这间，只要它大于 10000 则对 10000 求余即得出后面 4 位，否则如果大于 1000 则对 1000 求余得出后 3 位数，这样一层一层往小的判断。由于 return 的作用除了返回值以外，还有当执行到 return 时就跳出该程序，所以可以连续的用 if() 语句。 */
unsigned fun( unsigned w )
{ if(w>=10000) return w%10000;
  if(w>=1000) return w%1000;
  if(w>=100) return w%100;
  return w%10;
}
main()
{ unsigned x;
  printf("enter a unsigned integer number :");
  scanf("%u",&x);
  if(x<10) printf("data error!");
  else printf("the result :%u\n", fun(x));
}

```

题目 8

编写函数 int fun(int lim,int aa[MAX])，该函数的功能是求出小于 lim 的所有素数并放在 aa 数组中，该函数返回所求出素数的个数。

试题程序。

```

#include <stdio.h>
#include <conio.h>
#define MAX 100
/*注：循环 for(i=2;i<k;i++)用于判断 k 是否为素数，原理是当用 2 到 k-1 这间的数去对 k 求余，如余数为 0（即被整除）则表示 k 不是一个素数。语句 if(i>=k)用于判断在上一个 for() 循环中 i 能否走到 k，如果能则表示在 2 到 k-1 的数都不能整除 k，即 k 为素数。*/
int fun( int lim, int aa[MAX])
{ int i,j=0,k;

```

```

for( k=2; k<lim; k++)
{ for( i=2; i<k; i++)
if( !(k%i)) break;
if( i>=k) aa[j++]=k;
}
return j;
}
main()
{ int limit,i,sum;
int aa[MAX];
printf("\n input a integer number:");
scanf("%d",&limit);
sum=fun(limit,aa);
for(i=0; i<sum; i++)
{ if(i%10==0&&i!=0)
printf("\n");
printf("%5d", aa[i]);
}
}

```

题目 9

请编写函数 fun，函数的功能是求出二维数组周边元素之和，作为函数值返回。二维数组中的值在主函数中赋予。

例如：二维数组中的值为

则函数值为 61。

试题程序。

```

#include <conio.h>
#include <stdio.h>
#define M 4
#define N 5
/*注：该题的第一个 for()循环是计算矩阵的最上一行和最下一行的总和，第二个 for()
是计算除两头元素以外的最左一列和最右一列的元素的和，最后 sun 就是周边元素的和。
*/
int fun(int a[M][N])
{int sum=0,i;
for(i=0;i<N;i++)
sum+=a[0][i]+a[M-1][i];
for(i=1;i<M-1;i++);
sum+=a[i][0]+a[i][N-1];
return sum ;
}
main()
{ int aa[M][N]={ {1,3,5,7,9},
{2,9,9,9,4},

```

```

{6,9,9,9,8},
{1,3,5,7,0}};
int i,j,y;
clrscr();
printf("The original data is :\n");
for(i=0;i<M;i++)
{for(j=0;j<N;j++) printf("%6d",aa[i][j]);
printf("\n");
}
y=fun(aa);
printf("\nThe sum: %d\n",y);
printf("\n");
}

```

题目 10

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 个字符按降序排列。例如，原来的字符串为 CEAedca，排序后输出为 CedcEAa。

试题程序。

```

#include <string.h>
#include <conio.h>
#include <stdio.h>
/*该题采用的排序法是选择法进行降序排序，算法是用外 for() 循环从字符串的前端往后
端走动，每走动一个字符都用内嵌的 for() 循环在该字符后找出最小的字符与该字符进
行换位。直到外 for() 循环走到最后一个字符。此外，此题还要注意把首尾字符除开，即
在最外层 for() 循环中从 1 开始，只到 num-2 即可。*/
int fun(char *s,int num)
{ int i,j,t;
for(i=1;i<num-2;i++)
for(j=i+1;j<num-1;j++)
if(s[i]<s[j])
{ t=s[i];
s[i]=s[j];
s[j]=t;
}
}
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);
}

```

题目 11

请编一函数 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小元素，并依次放入 pp 所指一维数组中，二维数组中的数已在主函数中赋予。

试题程序。

```
#include "conio.h"
```

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

```
#define M 3
```

```
#define N 4
```

/*注：该题用 for(i=0;i<N;i++)来控制一列一列地找，而内嵌循环 for(j=0;j<M;j++)用于控制同列内元素的比较。多重循环的嵌套总是最里层循环变化最快，即外层循环改变一个值，内层循环就要循环完一次，对于多重循环一定要好好去体会和理解，在多数题目中都要用到多重循环（一般为二重）。*/

```
void fun(int tt[M][N],int pp[N])
```

```
{ int i, j;
```

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

```
{ pp=tt[0][i];
```

```
for(j=0;j<M;j++)
```

```
if(tt[j][i]<pp[i]) pp[i]=tt[j][i];
```

```
}
```

```
}
```

```
main()
```

```
{ int t[M][N]={ {22,45,56,30},
```

```
{19,33,45,38},
```

```
{20,22,66,40}};
```

```
int p[N],i,j,k;
```

```
clrscr();
```

```
printf("the original 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");
```

```
}
```

题目 12

写一个函数，从传入的 num 个字符串中找出最长的一个字符串，并通过形参指针 max 传回该串地址。（注意：用****作为结束输入的标志。）

试题程序。

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

```
#include "string.h"
```

```
#include "conio.h"
```

/* 函数 strlen()用于求出字符串的长度，这个题中主要是*max=p;不能换成 max=&p
; 如果用 maz=&p;则只改变了 max 的指向，它不能传回给实参。因此我们要改变 max 指向地址中的内容，这样才能使得实参 ps 有正确的值。*/

```
fun(char (*a)[81], int num, char **max)
{ char *p=a[0];int i;
for(i=1;i<num;i++)
if(strlen(a[i])>strlen(p))
p=a[i];
*max=p;
}
main()
{ char ss[10][81],*ps;
int n,i=0;
clrscr();
printf("enter string:\n");
gets(ss[i]);
puts(ss[i]);
while(!strcmp(ss[i], "****")==0)
{
i++;
gets(ss[i]);
puts(ss[i]);
}
n=i;
fun(ss,n,&ps);
printf("\nmax=%s\n",ps);
}
```

题目 13

请编一个函数 fun，其中 n 所指存储单元中存放了数组中元素的个数。函数的功能是：删除所有值为 y 的元素。数组元素中的值和 y 的值由主函数通过键盘读入。试题程序。

```
#include <conio.h>
#include <stdio.h>
#define M 20
/*注：该题的算法是，用 for()循环控制元素逐一判断数组元素是否等于 y，若不等则赋给新数 bb，由于删除的关系 j 总是小于或等于 i，故可用 bb 作为新数组，与第 6 题相似。还要注意*n=j；如果没有这个语句则不能传回新数组的个数，另外也不能换成 n=&j;这点是用指针时常犯的错误，切记！*/
void fun (int bb[], int *n,int y)
{ int i, j=0;
for( i=0; i<*n; i++)
if(bb[i]!=y) bb[j++]=bb[i];
*n=j;
}
```

```

main()
{ int aa[M],n,y,k;
printf("\nplease enter n:");
scanf("%d",&n);
printf("\nenter %d positive number\n", n);
for( k=0; k<n; k++) scanf( "%d",&aa[k]);
printf("the original data is: \n");
for( k=0; k<n; k++) printf(" %5d", aa[k]);
printf("\nenter a number to deleted:");
scanf("%d",&y);
fun(aa,&n,y);
printf("the data after deleted %d:\n", y);
for(k=0; k<n; k++) printf("%4d",aa[k]);
printf("\n");
}

```

题目 14

编写一个函数，该函数可以统计一个长度为 2 的字符串在另一个字符串中出现的次数。例如，假定输入的字符串为：asd asasdfg asd as zx67 asd mklo，子字符串为：as，则应输出 6。

试题程序。

```

#include "stdio.h"
#include "string.h"
#include "conio.h"
/*注：由于小串中只有 2 个字符所以可用 str[i]==substr[0]&&str[i+1]==subs
tr[1]来判断小串是否与长串当前位置（str[i]）相同（即出现一次）。因而只要让长串
当前位置逐一向后移即可（用 for()循环来完成）。*/
int fun(char *str,char *substr)
{ int i,n=0,s=strlen(str);
for(i=0;i<s;i++)
if((str[i]==substr[0])&&(str[i+1]==substr[1]))
n++;
return n;
}
main()
{
char str[81],substr[3];
int n; clrscr();
printf("enter 1:");
gets(str);
printf("enter 2:");
gets(substr);
puts (str);
puts(substr);
n=fun(str,substr);

```



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

题目 15

请编写一个函数 `int fun(int x)`，它的功能是：判断整数 `x` 是否是同构数。若是同构数，函数返回 1；否则返回 0。

所谓“同构数”是指这样的数，它出现在它的平方数的右边。

例如：输入整数 5，5 的平方数是 25，5 是 25 中右侧的数，所以 5 是同构数。

`x` 的值由主函数从键盘读入，要求不大于 100。

试题程序。

```
#include "conio.h"
#include "stdio.h"
/*注：由“同构数”的定义可得只要求出平方数或平方数的最后一位或平方数的最后两位然后与 x 进行判断即可，分别求余后即得出平方数的右边。*/
int fun (int x)
{ int xx=x * x;
  if(xx==x/xx%10==x/xx%100==x) return 1;
  else return 0;
}
main()
{ int x,y;
  clrscr();
  printf("\n please enter a integer numbers:");
  scanf("%d",&x);
  if(x>100){printf("data eror!\n");exit(0);}
  y=fun(x);
  if(y) printf("%d yes\n",x);
  else printf ("%d no!\n",x);
}
```

题目 16

请编写函数 `fun`，函数的功能是：在字符串中所有数字字符前加一个 '\$' 字符。

例如，输入：A1B23CD45，则输出为：A\$1B\$2\$3CD\$4\$5。

注意：部分源程序给出如下。

```
#include <stdio.h>
/*注：该题用 while() 循环来控制原字符串从头走到尾，在走动过程中判断是当前字符是否是数字，若是则在新串中先连一个 '$' 然后再连原字符，否则直接连原字符。一定要注意指针和下标的变化。最后要把新串拷贝到 s 所指的地址中，注意不能用 s=a；若用了，则实参数组还是原字符串。*/
void fun( char *s)
{char a[100];
  int i=0;
  while(*s)
  if(*s>='0'&&*s<='9') {a[i++]='$';a[i++]=*s++;}
  else a[i++]=*s++;
  a='\0';
```

```
strcpy(s,a);
}
main()
{ char s[80];
printf("enter a string:");
scanf("%s", s);
fun(s);
printf("the result: %s\n", s);
}
```

题目 17

请编一个函数 fun(char *s)，函数的功能是把字符串中所有的字符前移一个位置，串中的第一个字符移到最后。

例如：原有的字符串为：Mn.123xyZ，则调用该函数后，串中的内容为：n.123xyZM。

试题程序。

```
#include "conio.h"
#include "stdio.h"
#define N 81
/* 注：该题要先将字符串的头元素存到某一变量（用 c=*s）中，然后后面的字符依向前移（用 for() 循环），要记得在串的末尾加一个结束符。*/
fun(char *s)
{ char c=*s;
for(;*(s+1);s++)
*s=*(s+1);
*s=c;
*(s+1)='\0';
}
main()
{ char a[N];
clrscr();
printf("enter a string:");
gets(a);
printf("the original string is:");
puts(a);
fun(a);
printf("the string after modified:");
puts(a);
}
```

题目 18

请编写函数 fun，函数的功能是：将所有大于 1 小于整数 m 的非素数存入 xx 所指数组中，非素数的个数通过 k 传回。

例如，若输入：17，则应输出：9 和 4 6 8 9 10 12 14 15 16。

试题程序。

```
#include <conio.h>
#include <stdio.h>
```

/* 注：内嵌的 for() 循环用于判断是否是素数，在 $j < i$ 的情况下，只要 j 对 i 求余，余数为 0 则表示 i 不是素数，则将 i 存入 xx 数组中。`break;` 语句只是让它提前结束循环。不用亦可。*/

```
void fun( int m, int *k, int xx[] )
{ int i,j;
  *k=0;
  for( i=2; i<m; i++)
  for( j=2; j<i; j++)
  if( i%j==0)
  { xx[( *k)++] = i; break; }
}

main()
{ int m, n, zz[100];
  printf("\n please enter an integer number between 10 and 100: "
);
  scanf( "%d",&n);
  fun(n,&m,zz);
  printf("\n\n there are %d non-prime numbers less than %d: ", m,
n);
  for(n=0; n<m; n++)
  printf("\n %4d",zz[n]);
}
```

题目 19

请编一个函数 `fun(char *s)`，函数的功能是把字符串中所有的字母改写成该字母的下一个字母，最后一个字母 `z` 改写成字母 `a`。大写字母仍为大写字母，小写字母仍为小写字母，其他的字符不变。

例如：原有的字符串为：Mn.123zyZ，则调用该函数后，串中的内容为：No.123yzA。

试题程序。

```
#include "conio.h"
#include "string.h"
#include "stdio.h"
#include "ctype.h"

#define N 81

/* 该题的思路较为明显，用 for() 循环一个字符一个字符地走动，移动过程中判断是否在 a 到 y 之间，若是则加 1（即下移一个字母），若是 z 则换成对应的 a */
fun(char *s)
{ int i,n=strlen(s);
  for(i=0;i<n;i++)
  { if((s[i]>='A'&& s[i]<'Z') || (s[i]>='a'&& s[i]&lt;'z'))
    s[i]=s[i]+1;
    else if(s[i]=='z') s[i]='a';
    else if(s[i]=='Z') s[i]='A';
  }
}
```

```

}
main()
{ char a[N];
clrscr();
printf("enter a string:");gets(a);
printf("the original string is:");puts(a);
fun(a);
printf("the string after modified:");
puts(a);
}

```

题目 20

请编写函数 fun，函数的功能是：将 s 所指字符串中、下标为奇数位置上的大写字母转换为字母序列中的下一个小写字母；若该位置上是小写字母，则不转换；若该位置上是小写字母 Z，则仅转换成小写字母 z。

例如，输入：ABCdxZZZ，则输出为：AcCdXzZz。

试题程序。

```

#include "stdio.h"
#define N 80
/* 该题的算法与上一题相似，注意同一个字母小写要比大写在 ASCII 值上大 32。*/
void fun(char *s)
{
int i;
for(i=1;i<N;i=i+2)
if (s[i]>='A'&& s[i]<'Z')
s[i]+=32+1;
else if(s[i]=='Z') s[i]+=32;
}
main()
{
char s[N];
printf("\nenter a string:");scanf("%s",s);
fun(s);
printf("\nthe result:%s\n",s);
}

```

题目 21

程序定义了 N*N 的二维数组，并在主函数中赋值。请编写函数 fun，函数的功能是：

给数组周边元素置 0 值。

例如：a 数组中的值为

则返回主程序后 a 数组中的值应为

注意：部分源程序给出如下。

请勿改动主函数 main 和其它函数中的任何去何内容，仅在函数 fun 的花括号中填入你编写的若干语句。

试题程序。

```

#include <stdio.h>

```

```

#include <conio.h>
#include <stdlib.h>
#define N 5
/* 由于是赋值 0，因而只要能表示出周边元素即好办，它与求和不同在于求和不能重复
对某个元素进行运算，而赋值则可重复赋值。所以这里只用了一个 for() 循环来找出所有
的周边元素。*/
fun(int w[][N])
{
    int i;
    for(i=0;i<N;i++)
    {
        w[i][0]=0;
        w[i][N-1]=0;
        w[0][i]=0;
        w[N-1][i]=0;
    }
}
main()
{
    int a[N][N],i,j;
    clrscr();
    printf("***** The array *****\n");
    for(i=0;i<N;i++)
    {
        for(j=0;j<N;j++)
        {
            a[i][j]=rand()%10;
            printf("%4d",a[i][j]);
        }
        printf("\n");
    }
    fun(a);
    printf("***** The result *****\n");
    for(i=0;i<N;i++)
    {
        for(j=0;j<N;j++)
        {
            printf("%4d",a[i][j]);
        }
        printf("\n");
    }
}

```

题目 22

请编写函数 fun，它的功能是：求出 ss 所指字符串中、指定字符的个数，并返回此值。

例如：若输入字符串：123412132，输入字符为：1，则输出：3。

试题程序。

```

#include<conio.h>
#include<stdio.h>
#define M 81
/* 注：本题用 while() 循环来控制字符的移动，每移动一个字符都要进行判断（if(*ss
==c)）是否为指定的字母，若是则个数加 1。这里要注意如何让 ss 指针向下走动（ss++）

```

```

。*/
int fun(char *ss,char c)
{int num=0;
while(*ss!='\0')
{if(*ss==c) num++;
ss++;
}
return(num);
}
main()
{char a[M],ch;
clrscr();
printf("\nPlease enter a string:");gets(a);
printf("\nPlease enter a char:");ch=getchar();
printf("\nThe number of the char is:%d\n",fun(a,ch));
}

```

题目 23

程序定义了 $N*N$ 的二维数组，并在主函数中赋值。请编写函数 `fun`，函数的功能是：求出数组周边元素的平均值并作为函数值返回给主函数中的 `s`。

例如：a 数组中的值为

则返回主程序后 `s` 的值应为：3.375。

试题程序。

```

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define N 5
/* 注意不能重复计算，它与题 21 不同，第一个 for() 循环求出第一列与最后一列的和，
第二个 for() 循环求出第一行与最后一行（已除去两头元素），要注意第二个 for() 中起
始为 1，最后为 N-2。*/
double fun (int w[][N])
{ int i,t=0;
double s=0;
for(i=0;i<N;i++)
{s+=w[i][0]+w[i][N-1]; t+=2;}
for(i=1;i<N-1;i++)
{s+=w[0][i]+w[N-1][i]; t+=2;}
s=s/t;
return s;
}
main()
{ int a[N][N]={0,1,2,7,9,1,9,7,4,5,2,3,8,3,1,4,5,6,8,2,5,9,1,4,1};
int i,j;
double s;
clrscr();

```

```

printf("***** The array *****\n");
for(i=0;i<N;i++)
{for(j=0;j<N;j++)
printf("%4d",a[i][j]);
printf("\n");
}
s=fun(a);
printf("***** The result *****\n");
printf("The sum is %lf\n",s);
}

```

题目 24

编写一个函数 fun，它的功能是：实现两个字符串的连接（不使用库函数 strcat）

。

例如，分别输入下面两个字符串：

FirstString—

SecondString

程序输出：

FirstString—SecondString

注意：部分源程序给出如下。

请勿改动主函数 main 和其它函数中的任何内容，仅在函数 fun 的花括号中填入你编写的若干语句。

试题程序。

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
/* 注：第一个 for() 循环用于让 i 成为第一个字符串的结束符的下标，要注意在这个 for() 后要有个分号否则出错。第二个 for() 的作用是将第二个字符串一个一个字符依次放到第一个字符串的末尾。此外，还要注意最后一定要有 p1[i]='\0'; 来给字符串加上结束符。*/
```

```
void fun(char p1[],char p2[])
```

```
{int i,j;
```

```
for(i=0;p1[i];i++);
```

```
for(j=0;p2[j];j++)
```

```
p1[i++]=p2[j];
```

```
p1[i]='\0';
```

```
}
```

```
main()
```

```
{ char s1[80],s2[80];
```

```
clrscr();
```

```
printf("Enter s1 and s2:\n");
```

```
scanf("%s%s",s1,s2);
```

```
printf("s1=%s\n",s1);
```

```
printf("s2=%s\n",s2);
```

```
printf("Invoke fun(s1,s2):\n");
```

```
fun(s1,s2);
```

```
printf("After invoking:\n");
printf("%s\n",s1);
}
```

题目 25

程序定义了 $N \times N$ 的二维数组，并在主函数中自动赋值。请编写函数 `fun(int a[][N])`，函数的功能是：使数组第一列元素中的值与最后一列元素中的值对调、第二列元素的值与倒数第二列中的值对调、……、其他依次类推。

例如：a 数组中的值为

则返回主程序后 a 数组中的值应为

注意：部分源程序给出如下。

请勿改动主函数 `main` 和其它函数中的任何内容，仅在函数 `fun` 的花括号中填入你编写的若干语句。

试题程序。

```
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#define N 5
/* 本题的第一个 for() 循环用于控制行，而内嵌的 for() 用于在同一行中将列与列对调，
当 i 为 0 时 a[m][i] 为第 m 行的第一列元素，a[m][N-1-i] 为第 m 行的最后一列的元素，正好
符合调换的要求。依次类推。*/
int fun(int a[][N])
{ int i ,m,t;
for(m=0;m<N;m++)
for(i=0;i<N/2;i++)
{t=a[m][i];
a[m][i]=a[m][N-1-i];
a[m][N-1-i]=t;
}
}
main()
{ int a[N][N],i,j;
clrscr();
printf("***** The array *****\n");
for(i=0;i<N;i++)
{for(j=0;j<N;j++)
{a[i][j]=rand()%30;printf("%4d",a[i][j]);}
printf("\n");
}
fun(a);
printf("***** The result *****\n");
for(i=0;i<N;i++)
{for(j=0;j<N;j++)
printf("%4d",a[i][j]);
printf("\n");
}
```



```

}
}

```

题目 26

请编写函数 fun，函数的功能是：实现 $B=A+A'$ ，即把矩阵 A 加上 A 的转置，存放在矩阵 B 中。计算结果在 main 函数中输出。

例如：输入下面的矩阵： 其转置矩阵为：

程序输出：

注意：部分源程序给出如下。

请勿改动主函数 main 和其它函数中的任何内容，仅在函数 fun 的花括号中填入你编写的若干语句。

试题程序。

```
#include<conio.h>
```

```
#include<stdio.h>
```

```
/* 26 题：本题没有先求出 A 的转置，而是直接利用转置的性质（即第一行作为转置后的第一列，第二行作为转置后的第二列，的行列互换的性质）。而且最后要求出两矩阵的相加后得出的新矩阵，由转置性质可知原 a[j][i] 转置后位置为 a[i][j]，在表达式：b[i][j]=a[i][j]+a[j][i] 第一个 a 为 A 的元素，第二个 a 为 A' 的元素。*/
```

```
void fun(int a[3][3],int b[3][3])
```

```
{int i,j;
```

```
for(i=0;i<3;i++)
```

```
for(j=0;j<3;j++)
```

```
b[i][j]=a[i][j]+a[j][i];
```

```
}
```

```
main()
```

```
{int a[3][3]={ {1,2,3},{4,5,6},{7,8,9}},t[3][3];
```

```
int i,j;
```

```
clrscr();
```

```
fun(a,t);
```

```
for(i=0;i<3;i++)
```

```
{for(j=0;j<3;j++)
```

```
printf("%7d",t[i][j]);
```

```
printf("\n");
```

```
}
```

```
}
```

题目 27

序定义了 N*N 的二维数组，并在主函数中自动赋值。请编写函数 fun(int a[][N])，

函数的功能是：使数组第一行元素中的值与最后一行元素中的值对调、第二行元素的值与倒数第二行中的值对调、……、其他依次类推。

例如：a 数组中的值为

则返回主程序后 a 数组中的值应为

注意：部分源程序给出如下。

请勿改动主函数 main 和其它函数中的任何内容，仅在函数 fun 的花括号中填入你编写的若干语句。

试题程序。

```

#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#define N 5
/* 27 题：该题与 25 题相似，外层 for() 循环用于控制一列一列地移动，内嵌的 forr() 循环用于控制在同一列内对应行元素的对调。a[i][m] 为第 m 列第 i 行元素，而 a[N-1-i][m] 为第 m 列倒数第 i 行的元素，正好需要对调。*/
fun(int a[][N])
{ int i,m,t;
  for(m=0;m<N;m++)
  for(i=0;i<N/2;i++)
  {t=a[i][m];
   a[i][m]=a[N-1-i][m];
   a[N-1-i][m]=t;
  }
}

main()
{ int a[N][N], i,j;
  clrscr();
  printf("***** The array *****\n");
  for(i=0;i<N;i++)
  {for(j=0;j<N;j++)
   {a[i][j]=rand()%30; printf("%4d",a[i][j]);}
   printf("\n");
  }
  fun(a);
  printf("***** The result *****\n");
  for(i=0;i<N;i++)
  {for(j=0;j<N;j++)
   printf("%4d",a[i][j]);
   printf("\n");
  }
}

```

题目 28

m 个人的成绩存放在 score 数组中，请编写函数 fun，它的功能是：将低于平均分的人数作为函数值返回。

例如，当 score 数组中的数据为：10、20、30、40、50、60、70、80、90、时，函数返回的人数应该是 4，below 中的数据应为：10、20、30、40。

注意：部分源程序给出如下。

请勿改动主函数 main 和其它函数中的任何去何内容，仅在函数 fun 的花括号中填入你编写的若干语句。

试题程序。

```

#include <string.h>
#include <conio.h>

```

```

#include <stdio.h>
/* 28 题：第一个 for() 循环用来计算 score 数组中分数的总和，然后用 aver/=m 求出平均值，第二个循环用来找出小于平均分的元素，并放到数组 below 中，这里要注意 j 的递增方式。*/
int fun(int score[], int m, int below[])
{
    int i, j = 0, aver = 0;
    for(i = 0; i < m; i++)
        aver += score[i];
    aver /= m;
    for(i = 0; i < m; i++)
        if(score[i] < aver)
            below[j++] = score[i];
    return j;
}

main()
{
    int i, n, below[9];
    int score[9] = {10, 20, 30, 40, 50, 60, 70, 80, 90};
    clrscr();
    n = fun(score, 9, below);
    printf("\nBelow the average score are :");
    for(i = 0; i < n; i++) printf("%4d", below[i]);
}

```

题目 29

程序定义了 N*N 的二维数组，并在主函数中自动赋值。请编写函数 fun(int a[][N])，函数的功能是：使数组左下半三角元素中的值全部置成 0。

例如：a 数组中的值为

，则返回主程序后 a 数组中的值应为

注意：部分源程序给出如下。

请勿改动主函数 main 和其它函数中的任何内容，仅在函数 fun 的花括号中填入你编写的若干语句。

试题程序。

```

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define N 5
/* 29 题：该题的关键是如何表示出左下半角的元素，当从上往下到第 i 行时只有左边的 i 个元素要置为 0（每行总是如此）。用外层 for() 来控制行而内嵌的 for() 来控制每行的左边元素，故而在内嵌 for() 中 j 最大只能循环到 i（即表示出第 i 行左边的 i 个元素）。*/
/
int fun(int a[][N])
{
    int i, j;
    for(i = 0; i < N; i++)
        for(j = 0; j <= i; j++)
            a[i][j] = 0;
}

```

```

}
main()
{int a[N][N],i,j;
clrscr();
printf("*** The array ****\n");
for(i=0;i<N;i++)
{for(j=0;j<N;j++)
{a[i][j]=rand()%10;printf("%4d",a[i][j]);}
printf("\n");
}
fun(a);
printf("The result\n");
for(i=0;i<N;i++)
{for(j=0;j<N;j++)
printf("%4d",a[i][j]);
printf("\n");
}
}

```

题目 30

请编写函数 `fun`，经的功能是：求出 1 到 1000 之内能被 7 或 11 整除、但不能同时被 7 和 11 整除的所有整数并将它们放在 `a` 所指的数组中，通过 `n` 返回这些数的个数。

注意：部分源程序给出如下。

请勿改动主函数 `main` 和其它函数中的任何去何内容，仅在函数 `fun` 的花括号中填入你编写的若干语句。

试题程序。

```

#include <conio.h>
#include <stdio.h>

/* 30 题：该题关键就是如何表示能被 7 或 11 整除，但不能同时被 7 和 11 整除。用  $(i\%7==0)/(i\%11==0)$  来表示能被 7 或 11 整除，用  $!((i\%7==0)\&\&(i\%11==0))$  来表示不能同时被 7 和 11 整除。 */
void fun(int *a,int *n)
{ int i,m=0;
for(i=1;i<1000;i++)
if(((i%7==0)/(i%11==0))&&!((i%7==0)&&(i%11==0)))
{a[m]=i;m+=1;}
*n=m;
}

main()
{ int aa[1000],n,k;
clrscr();
fun(aa,&n);
for(k=0;k<n;k++)
if((k+1)%10==0) printf("\n");
else printf("%d,",aa[k]);
}

```

}

题目 31

程序定义了 $N \times N$ 的二维数组，并在主函数中自动赋值。请编写函数 `fun(int a[][N])`

函数的功能是：使数组右上半三角元素中的值全部置成 0。

例如：a 数组中的值为

，则返回主程序后 a 数组中的值应为

注意：部分源程序给出如下。

请勿改动主函数 `main` 和其它函数中的任何内容，仅在函数 `fun` 的花括号中填入你编写的若干语句。

试题程序。

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
#include <stdlib.h>
```

```
#define N 5
```

/* 31 题：该题的关键亦是在如何表示右上半角，当从上往下到第 i 行时要从第 i 列（所以内嵌 `for()` 的 j 从 i 开始循环）开始替换直到最后一列，本题用外层 `for()` 来控制行，而用内嵌的 `for()` 来控制每一行内右边元素置 0。

```
int fun(int a[][N])
```

```
{ int i,j;
```

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

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

```
a[i][j]=0;
```

```
}
```

```
main()
```

```
{ int a[N][N],i,j;
```

```
clrscr();
```

```
printf("***** The array *****\n");
```

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

```
{for(j=0;j<N;j++)
```

```
{a[i][j]=rand()%20;printf("%4d",a[i][j]);}
```

```
printf("\n");
```

```
}
```

```
fun(a);
```

```
printf("THE RESULT\n");
```

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

```
{for(j=0;j<N;j++) printf("%4d",a[i][j]);
```

```
printf("\n");
```

```
}
```

```
}
```

题目 32

编写函数 `void fun(int x,int pp[],int *n)`，它的功能是：求出能整除 x 且不是偶数的各整数，并放在 `pp` 所指的数组中，这些除数的个数通过形参 n 返回。

例如，若 x 中的值为：30，则有 4 个数符合要求，它们是

1，3，5，15。

注意：部分源程序给出如下。

请勿改动主函数 main 和其它函数中的任何去何内容，仅在函数 fun 的花括号中填入你编写的若干语句。

试题程序。

```
#include <conio.h>
#include <stdio.h>

/* 32 题：由于要求的数不能是偶数故编程时从 1 开始循环而步长为 2 这样正好保正 i 永远
的奇数。这里也要注意存放结果的数组 pp 的下标变化方式。*/

void fun(int x, int pp[], int *n)
{
    int i,j=0;
    for( i=1;i<=x; i=i+2)
        if( x%i!=0)
            pp[j++]=i;
    *n=j;
}

main()
{
    int x, aa[1000], n, i;
    printf("\n please enter an integer number:\n"); scanf("%d", &x);
    fun(x, aa, &n);
    for(i=0; i<n; i++)
        printf("%d ", aa[i]);
    printf("\n");
}
```

题目 33

序定义了 N*N 的二维数组，并在主函数中自动赋值。请编写函数 fun(int a[][N], int n)，函数的功能是：使数组右上半三角元素中的值乘以 m。例如：若 m 的值为 2，a 数组中的值为

，则返回主程序后 a 数组中的值应为

注意：部分源程序给出如下。

请勿改动主函数 main 和其它函数中的任何去何内容，仅在函数 fun 的花括号中填入你编写的若干语句。

试题程序。

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define N 5

/* 33 题：该题的关键也是在如何表示右上半角（可以与 31 题相同），本题采用的是在一
行内从最后一列的元素开始往前乘，只有 j>=i 时才改变（原理与 31 题相同）。*/

int fun( int a[][N], int m)
{
    int i,j;
    for(i=0; i<N; i++)
        for(j=N-1; j>=i; j--)
```

```

a[i][j]*=m;
}
main()
{ int a[N][N],m,i,j;
printf("**** the array ****\n");
for( i=0; i<N;i++)
{ for( j=0; j<N; j++)
{a[i][j]=rand()%20; printf("%4d", a[i][j]); }
printf("\n");
}
do m=rand()%10; while( m>=3);
printf("m=%4d\n", m);
fun( a,m);
printf(" the result \n");
for(i=0; i<N; i++)
{for(j=0; j<N; j++) printf("%4d", a[i][j]);
printf("\n");
}
}

```

题目 34

编写一个函数 void fun(char *tt,int pp[]), 统计在 tt 字符串中 'a' 到 'z' 26 个字母各自出现的次数, 并依次放在 pp 所指数组中。

例如, 当输入字符串: abcdefgabcdeabc 后, 程序的输出结果应该是:

3 3 3 2 2 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

注意: 部分源程序给出如下。

请勿改动主函数 main 和其它函数中的任何去何内容, 仅在函数 fun 的花括号中填入你编写的若干语句。

试题程序。

```
#include<conio.h>
```

```
#include<stdio.h>
```

/* 34 题: 本题采用的是字母的 ASCII 码值与数组 pp 下标的对应转换关系分别求出对应字母的个数。第一个 for() 是给 pp 赋初值 0, 第二个 for() 用于控制在字符串内从头到尾移动。由于字母'a'的 ASCII 码值为 97 而它的个数要放在 pp[0]中, 而'a'-97 的值正好为 0, 其它的依此类推。*/

```
void fun(char *tt,int pp[])
```

```
{ int i;
```

```
for(i=0;i<26;i++)
```

```
pp[i]=0;
```

```
for(;*tt;tt++)
```

```
if(*tt<='z'&&*tt>='a')
```

```
pp[*tt-97]++;
```

```
}
```

```
main()
```

```
{ char aa[1000];
```

```

int bb[26],k;
clrscr();
printf("\nPlease enter a char string:");scanf("%s",aa)
;
fun(aa,bb);
for(k=0;k<26;k++) printf("%d",bb[k]);
printf("\n");
}

```

题目 35

序定义了 $N \times N$ 的二维数组，并在主函数中自动赋值。请编写函数 `fun(int a[][N],int n)`，函数的功能是：使数组左下半三角元素中的值乘以 n 。例如：若 n 的值为 2， a 数组中的值为

，则返回主程序后 a 数组中的值应为

注意：部分源程序给出如下。

请勿改动主函数 `main` 和其它函数中的任何去何内容，仅在函数 `fun` 的花括号中填入你编写的若干语句。

试题程序。

```

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define N 5
/* 35 题：本题与 29 题相似，只要能表示出左下半角的元素即可。*/
int fun( int a[][N], int n)
{ int i,j;
for(i=0; i<N; i++)
for(j=0; j<=i; j++)
a[i][j]*=n;
}
main()
{ int a[N][N],n,i,j;
printf("**** the array ****\n");
for( i=0; i<N; i++)
{for( j=0; j<N; j++)
{a[i][j]=rand()%10; printf("%4d", a[i][j]); }
printf("\n");
}
do n=rand()%10 ; while(n>=3);
printf("n=%4d \n", n);
fun( a,n);
printf( "**** the result ****\n");
for(i=0;i<N; i++)
{for(j=0; j<N; j++) printf("%4d", a[i][j]);
printf("\n");
}
}

```


}

题目 36

数 fun 的功能是：将两个两位数的整数 a、b 合并形成一个整数放在 c 中。合并的方式是：将 a 的十位和个位数依次放在 c 数的千位和十位上，b 数的十位和个位数依次放在 c 数的个位和百位上。

例如：当 a=45，b=12。调用该函数后，c=4251

注意：部分源程序给出如下。

请勿改动主函数 main 和其它函数中的任何去何内容，仅在函数 fun 的花括号中填入你编写的若干语句。

试题程序。

```
#include <conio.h>
```

```
#include <stdio.h>
```

```
/* 36 题：本题的关键在于如何表示出个、十、百、千位数。对于一个两位的整数，用 10 对它求余得到个位数上的数，将它除 10 得到十位数上的数。*/
```

```
void fun(int a, int b, long *c)
```

```
{ int i,j,k,n;
```

```
i=a%10; j=a/10; k=b%10; n=b/10;
```

```
*c=j*1000+k*100+i*10+n;
```

```
}
```

```
main()
```

```
{ int a,b; long c;
```

```
printf(" input a, b: ");
```

```
scanf("%d%d", &a,&b);
```

```
fun(a,b,&c);
```

```
printf(" the result is :%ld\n", c);
```

```
}
```

题目 37

请编写一个函数 void fun(int m,int k,int xx[])，该函数的功能是：将大于整数 m 且紧靠 m 的 k 个素数存入 xx 所指的数组中。

例如，若输入 17，5，则应输出：19，23，29，31，37。

注意：部分源程序给出如下。

请勿改动主函数 main 和其它函数中的任何去何内容，仅在函数 fun 的花括号中填入你编写的若干语句。

试题程序。

```
#include <conio.h>
```

```
#include <stdio.h>
```

```
/* 37 题：本题中的 for() 循环用来判断一个数是否为素数，由于个数只能是 k 个所以用 do-while 循环来控制。*/
```

```
void fun(int m, int k, int xx[])
```

```
{int i,j,num=0;
```

```
for(i=m+1;num<k;i++)
```

```
{
```

```
for(j=2;j<i;j++)
```

```
if(i%j==0)
```

```

break;
if(j>=i)
xx[num++]=i;}
}
main()
{ int m,n,zz[1000];
printf("\n please enter two integers: ");
scanf("%d%d",&m,&n);
fun( m,n,zz);
for(m=0; m<n; m++)
printf("%d ", zz[m]);
printf("\n");
}

```

题目 38

已知学生的记录由学号和学习成绩构成，N 名学生的数据已存入 a 结构体数组中。请编写函数 fun，函数的功能是：找出成绩最低的学生记录，通过形参返回主函数（规定只有一个最低分）。已给出函数的首部，请完成该函数。

注意：部分源程序给出如下。

请勿改动主函数 main 和其它函数中的任何去何内容，仅在函数 fun 的花括号中填入你编写的若干语句。

试题程序。

```

#include <stdio.h>
#include <string.h>
#include <conio.h>
#define N 10
typedef struct ss
{ char num[10]; int s;} STU;
/* 38 题：这个题目中有一结构组数，要找的是数组元素中整型成员的值最小的元素。要注意本题中的赋值方式（分两部分，对于字符型成员只能用 strcpy() 函数，而不能用等号“=”进行赋（即不能用(*s).num=a[k].num），这与字符串的赋值相同（切记）。*/
fun( STU a[], STU *s)
{ int k,i;
(*s).s=a[0].s;
for( i=0;i<N; i++)
if (a[i].s<(*s).s)
{(*s).s=a[i].s; k=i; }
strcpy ( (*s).num, a[k].num);
}
main()
{ STU a[N]={ {"A01",81},{ "A02",89},{ "A03",66}
, {"A04",87},{ "A05",77},
{"A06",90},{ "A07",79},{ "A08",61},{ "
;A09",80},{ "A10",71} }, m;
int i;

```

```
printf("**** the original data ****\n");
for(i=0;i<N; i++) printf("N0=%s Mark=%d\n", a[i].num,a[i].s);
);
fun( a,&m);
printf("**** the result****\n");
printf(" the lowest: %s ,%d\n", m.num,m.s);
}
```

题目 39

请编写一个函数 `void fun(char a[],int k,int n)`，其功能是：删除字符串中指定下标开始的 `n` 个字符。其中，`a` 指向字符串，`k` 中存放指定的下标。

例如，字符串内容为：Hello World!，`k` 中值为：5，`n` 中的值为：3，则调用该函数的结果为：Hello World!。

注意：部分源程序给出如下。

请勿改动主函数 `main` 和其它函数中的任何去何内容，仅在函数 `fun` 的花括号中填入你编写的若干语句。

试题程序。

```
#include <stdio.h>
#define N 80
/* 39 题：本题要删除从下标 k 开始的后 n 个字符。所以所用的算法是将下标为 k+n 及以后的所有字符依次往前移 n 个字符，这样就把要删除的字符覆盖了（即删除了）。记得最后要在新串尾加一个结束符'\0'。*/
void fun( char *a, int k, int n)
{ int m=strlen(a),i;
for(i=k+n; i<m; i++) a[k++]=a[i];
a[k]='\0';
}
main()
{ char s[N]="Hellollo World!";
int k,n;
printf("\n the original string: %s\n", s);
printf(" enter index-----k: "); scanf("%d"
, &k);
printf(" enter number to delete----n: "); scanf("%d"
, &n);
fun(s,k,n);
printf("\n the string after deleted: %s\n", s);
}
```

题目 40

已知学生的记录由学号和学习成绩构成，`N` 名学生的数据已存入 `a` 结构体数组中。请编写函数 `fun`，函数的功能是：找出成绩最高的学生记录，通过形参返回主函数（规定只有一个最高分）。已给出函数的首部，请完成该函数。

注意：部分源程序给出如下。

请勿改动主函数 `main` 和其它函数中的任何去何内容，仅在函数 `fun` 的花括号中填入你编写的若干语句。

试题程序。

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
#define N 10
typedef struct ss
{ char num[10]; int s;} STU;
/* 40 题：本题与 38 题相似，只是采用的赋值方式不同。采用的是结构体元素对结构体元素赋值。*/
fun( STU a[], STU *s)
{ int i;
*s=a[0];
for( i=1;i<N; i++)
if (a[i].s>( *s).s)
*s=a[i];
}
main()
{ STU a[N]={ {"A01",81},{ "A02",89},{ "A03",66}
, {"A04",87},{ "A05",77},
{"A06",90},{ "A07",79},{ "A08",61},{
:A09",80},{ "A10",71} }, m;
int i;
printf("**** the original data ****\n");
for(i=0;i<N; i++) printf("N0=%s Mark=%d\n", a[i].num,a[i].s
);
fun( a,&m);
printf("**** the result****\n");
printf(" the top : %s ,%d\n", m.num,m.s);
}
```

1 题 函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS10.DAT 中。

替代关系： $f(p)=p*11 \bmod 256$ (p 是数组中某一个字符的 ASCII 值，f(p)是计算后新字符的 ASCII 值)，如果原字符的 ASCII 值是偶数或计算后 f(p)值小于等于 32，则该字符不变，否则将 f(p)所对应的字符进行替代。

部分源程序已给出，原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include
#include
#include
```

```
#include
unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/
int ReadDat(void)
void WriteDat(void)
void encryptChar()
{
}
void main()
{
clrscr();
if(ReadDat()){
printf("数据文件 ENG.IN 不能打开! \n\\007\\");
return;
}
encryptChar();
WriteDat();
}
int ReadDat(void)
{
FILE *fp;
int i=0;
unsigned char *p;
if((fp=fopen("eng.in","r"))==NULL) return 1;
while(fgets(xx[i],80,fp)!=NULL){
p=strchr(xx[i],'\n');
if(p)*p=0;
i++;
}
maxline=i;
fclose(fp);
return 0;
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen("ps10.dat","w");
for(i=0;i<MAXLINE;i++){
printf("%s\\n",xx[i]);
fprintf(fp,"%s\\n",xx[i]);
}
fclose(fp);
}
```

注：在 ReadDat()函数中由于 fgets()函数读入数据时没有读入字符串结束符'\0'，因而用 while()循环在 xx 数组每一行末尾将换行符'\n'替换成结束符'\0'。

编写的函数如下：该函数的基本算法是——让字符指针 pf 指向每一行的开头然后逐一往后移动，在移动过程中按要求进行转换。 $*pf=*pf/2*2$ 用于判断是否为偶数。if()条件语句用于控制不替代字符。

```
void encryptChar()
{
    int i;
    char *pf;
    for(i=0;i<MAXLINE;I++)
    {pf=xx[i];
    while(*pf!=0)
    {if(*pf==*pf/2*2||*pf*11%256<32)
    {pf++;continue;}
    *pf=*pf*11%256;
    pf++;
    }
    }
}

void encryptChar()
{
    int i,j,t;
    for(i=0;i<MAXLINE;I++)
    {
    for(j=0;j<STRLEN(XX[I]);J++)
    {
    t=xx[i][j]*11%256;
    if(t<=32 || xx[i][j]%2==0) continue;
    xx[i][j]=t;
    }
    }
}
```

2 题 函数 ReadDat()实现从文件 in.dat 中读取 20 行数据存放到字符串数组 xx 中(第行字符串长度均小于 80)。请编制函数 jsSort()，其函数的功能是：以行为单位对字符串按给定的条件进行排序，排序后的结果仍按行重新存入字符串数组 xx 中，最后调用函数 WriteDat()把结果 xx 输出到文件 out.dat 中。

条件：从字符串中间一分为二，左边部分按字符的 ASCII 值升序排序，排序后左边部分与右边部分进行交换。如果原字符串长度为奇数，则最中间的字符不参加处理，字符仍放在原位置上。

例如：位置 0 1 2 3 4 5 6 7 8

源字符串 d c b a h g f e

4 3 2 1 9 8 7 6 5

则处理后字符串 h g f e a b c d

8 7 6 5 9 1 2 3 4

部分源程序已给出。

请勿改动主函数 main()、读函数 ReadDat()和写函数 WriteDat()的内容。

```
#include
#include
#include
char xx[20][80];
void jsSort()
{
}
void main()
{
    readDat();
    jsSort();
    writeDat();
}
readDat()
{
    FILE *in;
    int i=0;
    char *p;
    in=fopen("in.dat","r");
    while(i<20&&fgets(xx[i],80,in)!=NULL){
        p=strchr(xx[i],'\n');
        if(p)*p=0;
        i++;
    }
    fclose(in);
}
writeDat()
{
    FILE *out();
    int i;
    clrscr();
    out=fopen("out.dat","w");
    for(i=0;i<20;i++){
        printf("%s\n",xx[i]);
        fprintf(out,"%s\n",xx[i]);
    }
    fclose(out);
}
```

注：先采用冒泡法对左边部分进行升序排序，然后将排序后的左半与右半按对应位进行调换。

```
void jsSort()
{
    int i, strl, half, j, k;
    char ch;
    for(i=0; i<20; i++)
    {strl=strlen(xx[i]);
    half=strl/2;
    for(j=0; j<HALF-1; j++)
    for(k=j+1; k<HALF; k++)
    if(xx[i][j]>xx[i][k])
    {ch=xx[i][j];
    xx[i][j]=xx[i][k];
    xx[i][k]=ch;
    }
    for(j=half-1; k=strl-1; j>=0; j--, k--)
    {ch=xx[i][j];
    xx[i][j]=xx[i][k];
    xx[i][k]=ch;
    }
    }
}

void jsSort()
{
    int i, j, k, strl;
    char ch;
    for(i=0; i<20; i++)
    {
        strl=strlen(xx[i]);
        for(j=0; j
        for(k=j+1; k
        if(xx[i][j]>xx[i][k])
        {
            ch=xx[i][j];
            xx[i][j]=xx[i][k];
            xx[i][k]=ch;
        }
        for(j=0; j
        {
            ch=xx[i][j];
            xx[i][j]=xx[i][(strl+1)/2+j];
            xx[i][(strl+1)/2+j]=ch;
        }
    }
}
```


3 题 函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS8.DAT 中。

替代关系： $f(p)=p*11 \bmod 256$ (p 是数组中某一个字符的 ASCII 值，f(p)是计算后新字符的 ASCII 值)，如果原字符是小写字母或计算后 f(p)值小于等于 32，则该字符不变，否则将 f(p)所对应的字符进行替代。

部分源程序已给出。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include
#include
#include
#include
unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/
int ReadDat(void)
void WriteDat(void)
void encryptChar()
{
}
void main()
{
clrscr();
if(ReadDat()){
printf("数据文件 ENG.IN 不能打开! \n\\007\\");
return;
}
encryptChar();
WriteDat();
}
int ReadDat(void)
{
FILE *fp;
int i=0;
unsigned char *p;
if((fp=fopen("eng.in","r"))==NULL) return 1;
while(fgets(xx[i],80,fp)!=NULL){
p=strchr(xx[i],'\n');
if(p)*p=0;
i++;
}
maxline=i;
fclose(fp);
return 0;
}
```

```

void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen("ps8.dat","w");
for(i=0;i<MAXLINE;I++){
printf("%s\n",xx[i]);
fprintf(fp,"%s\n",xx[i]);
}
fclose(fp);
}

```

注：与题 1 相似。

```

void encryptchar()
{
int i;
char *pf;
for(i=0;i<MAXLINE;I++)
{pf=xx[i];
while(*pf!=0)
{if(*pf>='a' && *pf<='z' || *pf*11%256<32)
{pf++;continue;}
*pf=*pf*11%256;
pf++;
}
}
}

void encryptChar()
{
int i,j,t;
for(i=0;i<MAXLINE;I++)
{
for(j=0;j<STRLEN(XX[I]);J++)
{
t=xx[i][j]*11%256;
if(t<=32 || xx[i][j]>='a' && xx[i][j]<='z') continue;
xx[i][j]=t;
}
}
}

```

4 题 函数 ReadDat()实现从文件 in.dat 中读取 20 行数据存放到字符串数组 xx 中(第行字符串长度均小于 80)。请编制函数 jsSort(), 其函数的功能是：以行为单位对字符串按给定的条件进行排序, 排序后的结果仍按行重新存入字符串数组 xx 中, 最后调用函数 WriteDat()把结果 xx 输出到文件 out.dat 中。

条件：从字符串中间一分为二，左边部分按字符的 ASCII 值降序排序，排序后左边部分与右边部分进行交换。如果原字符串长度为奇数，则最中间的字符不参加处理，字符仍放在原位置上。

例如：位置 0 1 2 3 4 5 6 7 8

源字符串 a b c d h g f e

1 2 3 4 9 8 7 6 5

则处理后字符串 h g f e d c b a

8 7 6 5 9 4 3 2 1

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include
#include
#include
char xx[20][80];
void jsSort()
{
}
void main()
{
    readDat();
    jsSort();
    writeDat();
}
readDat()
{
    FILE *in;
    int i=0;
    char *p;
    in=fopen("in.dat","r");
    while(i<20&&fgets(xx[i],80,in)!=NULL){
        p=strchr(xx[i],'\n');
        if(p)*p=0;
        i++;
    }
    fclose(in);
}
writeDat()
{
    FILE *out();
    int i;
    clrscr();
    out=fopen("out.dat","w");
    for(i=0;i<20;i++){
        printf("%s\n",xx[i]);
```

```

fprintf(out,"%s\\n",xx[i]);
}
fclose(out);
}

```

注：该题采用插入法实行降序排序，其它与题 2 相似。

```

void jsSort()
{
    int i, strl, half, j, k;
    char ch;
    for(i=0; i<20; i++)
    {
        strl=strlen(xx[i]);
        half=strl/2;
        for(j=1; j<HALF; j++)
        {
            ch=xx[i][j];
            k=j-1;
            while((k>=0)&&(ch>xx[i][k]))
            {
                xx[i][k+1]=xx[i][k];
                k--;
            }
            xx[i][k+1]=ch;
        }
        for(j=half-1, k=strl-1; j>=0; j--, k--)
        {
            ch=xx[i][j];
            xx[i][j]=xx[i][k];
            xx[i][k]=ch;
        }
    }
}

void jsSort()
{
    int i, j, k, strl;
    char ch;
    for(i=0; i<20; i++)
    {
        strl=strlen(xx[i]);
        for(j=0; j
        for(k=j+1; k
        if(xx[i][j]<xx[i][k])
        {
            ch=xx[i][j];
            xx[i][j]=xx[i][k];
            xx[i][k]=ch;
        }
    }
}

```

```

for(j=0;j
{
ch=xx[i][j];
xx[i][j]=xx[i][(strl+1)/2+j];
xx[i][(strl+1)/2+j]=ch;
}
}
}

```

5 题 函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS6.DAT 中。

替代关系： $f(p)=p*11 \bmod 256$ (p 是数组中某一个字符的 ASCII 值，f(p)是计算后新字符的 ASCII 值)，如果计算后 f(p)值小于等于 32 或 f(p)对应的字符是数字 0 至 9，则该字符不变，否则将 f(p)所对应的字符进行替代。

部分源程序已给出。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
#include
unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/
int ReadDat(void)
void WriteDat(void)
void encryptChar()
{
}
void main()
{
clrscr();
if(ReadDat()){
printf("数据文件 ENG.IN 不能打开! \n\n007\n");
return;
}
encryptChar();
WriteDat();
}
int ReadDat(void)
{
FILE *fp;
int i=0;
unsigned char *p;
if((fp=fopen("eng.in","r"))==NULL) return 1;
while(fgets(xx[i],80,fp)!=NULL){

```

```
p=strchr(xx[i],'\n');
if(p)*p=0;
i++;
}
maxline=i;
fclose(fp);
return 0;
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen("ps6.dat","w");
for(i=0;i<MAXLINE;i++){
printf("%s\n",xx[i]);
fprintf(fp,"%s\n",xx[i]);
}
fclose(fp);
}
```

注：与题 1 相似。只是它要求对 f(p)进行判断。

```
void encryptchar()
{
int i;
char *pf;
for(i=0;i<MAXLINE;i++)
{pf=xx[i];
while(*pf!=0)
{if(*pf*11%256>='0'&&*pf*11%256<='9'||*pf*11%256<32)
{pf++;continue;}
*pf=*pf*11%256;
pf++;
}
}
}
```

本人只是为了上机能通过，不追求什么算法速度,答案仅供大家参考，我 9 号长春上机。
(tonggu)

```
void encryptChar()
{
int i,j,stri;
int t;
for (i=0;i<MAXLINE;i++)
{
```

```

strl=strlen(xx[i]);
for(j=0;j<STRL;J++)
{
t=xx[i][j]*11%256;
if (t<=32 || t>='0' && t<='9')
continue;
xx[i][j]=t;
}
}
}

```

6 题 函数 ReadDat()实现从文件 in.dat 中读取 20 行数据存放到字符串数组 xx 中(第行字符串长度均小于 80)。请编制函数 jsSort(), 其函数的功能是: 以行为单位对字符串按给定的条件进行排序, 排序后的结果仍按行重新存入字符串数组 xx 中, 最后调用函数 WriteDat()把结果 xx 输出到文件 out.dat 中。

条件: 从字符串中间一分为二, 左边部分按字符的 ASCII 值降序排序, 右边部分按字符的 ASCII 值升序排序。如果原字符串长度为奇数, 则最中间的字符不参加排序, 字符仍放在原位置上。

例如: 位置 0 1 2 3 4 5 6 7 8

源字符串 a b c d h g f e

1 2 3 4 9 8 7 6 5

则处理后字符串 d c b a e f g h

4 3 2 1 9 5 6 7 8

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
char xx[20][80];
void jsSort()
{
}
void main()
{
readDat();
jsSort();
writeDat();
}
readDat()
{
FILE *in;
int i=0;
char *p;
in=fopen("in.dat","r");
while(i<20&&fgets(xx[i],80,in)!=NULL){

```

```

p=strchr(xx[i],'\n');
if(p)*p=0;
i++;
}
fclose(in);
}
writeDat()
{
FILE *out();
int i;
clrscr();
out=fopen("out.dat","w");
for(i=0;i<20;i++){
printf("%s\n",xx[i]);
fprintf(out,"%s\n",xx[i]);
}
fclose(out);
}

```

注：要注意当要进行右半部分排序时，一定要判断原字符串个数是否为奇数，若是则要 half 加 1，本题对右半部分采用选择法对其进行升序排序。

```

void jsSort()
{
int i, strl, half, j, k, p;
char ch;
for(i=0; i<20; i++)
{strl=strlen(xx[i]);
half=strl/2;
for(j=0; j<HALF-1; j++)
for(k=j+1; k<HALF; k++)
if(xx[i][j]<xx[i][k])
{ch=xx[i][j];
xx[i][j]=xx[i][k];
xx[i][k]=ch;
}
if(strl%2) half++; /* Must to note the order line! */
for(j=half; j<STRL; j++)
{p=j;
for(k=j; k<STR; k++)
if(xx[i][k]<xx[i][p])
p=k;
ch=xx[i][j]; xx[i][j]=xx[i][p]; xx[i][p]=ch;
}
}
}

```



```

void jsSort()
{
int i,j,k,strl;
char ch;
for(i=0;i<20;i++)
{
strl=strlen(xx[i]);
for(j=0;j
for(k=j+1;k
if(xx[i][j]<XX[I][K])
{
ch=xx[i][j];
xx[i][j]=xx[i][k];
xx[i][k]=ch;
}
for(j=(strl+1)/2;j<STRL;J++)
for(k=j+1;k<STRL;K++)
if(xx[i][j]>xx[i][k])
{
ch=xx[i][j];
xx[i][j]=xx[i][k];
xx[i][k]=ch;
}
}
}
}

```

7 题 函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS4.DAT 中。

替代关系： $f(p)=p*11 \bmod 256$ (p 是数组中某一个字符的 ASCII 值，f(p)是计算后新字符的 ASCII 值)，如果计算后 f(p)值小于等于 32 或 f(p)对应的字符是大写字母，则该字符不变，否则将 f(p)所对应的字符进行替代。

部分源程序已给出。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
#include
unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/
int ReadDat(void)
void WriteDat(void)
void encryptChar()
{
}

```

```
void main()
{
clrscr();
if(ReadDat()){
printf("\n数据文件 ENG.IN 不能打开! \n\\007\\");
return;
}
encryptChar();
WriteDat();
}
int ReadDat(void)
{
FILE *fp;
int i=0;
unsigned char *p;
if((fp=fopen("eng.in","r"))==NULL) return 1;
while(fgets(xx[i],80,fp)!=NULL){
p=strchr(xx[i],'\n');
if(p)*p=0;
i++;
}
maxline=i;
fclose(fp);
return 0;
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen("ps4.dat","w");
for(i=0;i<MAXLINE;i++){
printf("%s\n",xx[i]);
fprintf(fp,"%s\n",xx[i]);
}
fclose(fp);
}
```

注：下题 1 相似，只是它要求对 f(p)进行判断。

```
void encryptchar()
{
int i;
char *pf;
for(i=0;i<MAXLINE;i++)
{pf=xx[i];
```

```

while(*pf!=0)
{if(*pf*11%256>='A'&&*pf*11%256<='Z'||*pf*11%256<32)
{pf++;continue;}
*pf=*pf*11%256;
pf++;
}
}
}
void encryptChar()
{
int i,j,t;
for(i=0;i<MAXLINE;I++)
{
for(j=0;j<STRLEN(XX[I]);J++)
{
t=xx[i][j]*11%256;
if(t<=32 || t>='A' && t<='Z') continue;
xx[i][j]=t;
}
}
}

```

8 题 在文件中有 200 个正整数,且每个数均在 1000 至 9999 之间。函数 ReadDat()读取这 200 个数存放到数组 aa 中。请编制函数 jsSort(), 其函数的功能是: 要求按每个数的后三位的大小进行升序排列, 然后取出满足此条件的前 10 个数依次存入数组 b 中, 如果后三位的数值相等, 则按原先的数值进行降序排列。最后调用函数 WriteDat()把结果 bb 输出到文件 out.dat 中。

例: 处理前 6012 5099 9012 7025 8088

处理后 9012 6012 7025 8088 5099

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
int aa[200],bb[10];
void jsSort()
{
}
void main()
{
readDat();
jsSort();
writeDat();
}
readDat()

```

```

{
FILE *in;
int i;
in=fopen("in.dat","r");
for(i=0; i<200; i++) fscanf(in,"%d",&aa[i]);
fclose(in);
}
writeDat()
{
FILE *out;
int i;
clrscr();
out=fopen("out.dat","w");
for(i=0; i<10; i++){
printf("i=%d,%d\\n",i+1,bb[i]);
fprintf(out,"%d\\n",bb[i]);
}
fclose(out);
}

```

注：每个元素对 1000 求余得到后三位的数，然后用冒泡法进行排序。

```

void jsSort()
{
int i,j,data;
for(i=0;i<199;i++)
for(j=i+1;j<200;j++)
{if(aa[i]%1000>aa[j]%1000)
{data=aa[i];
aa[i]=aa[j];
aa[j]=data;
}
else
if(aa[i]%1000==aa[j]%1000)
if(aa[i]<aa[j])
{data=aa[i];
aa[i]=aa[j];
aa[j]=data;
}
}
for(i=0;i<10;i++)
bb[i]=aa[i];
}

```

9 题 函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组

xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS2.DAT 中。

替代关系： $f(p)=p*13 \bmod 256$ (p 是数组中某一个字符的 ASCII 值，f(p)是计算后新字符的 ASCII 值)，如果计算后 f(p)值小于等于 32 或其 ASCII 值是偶数，则该字符不变，否则将 f(p)所对应的字符进行替代。

部分源程序已给出。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include
#include
#include
#include
unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/
int ReadDat(void)
void WriteDat(void)
void encryptChar()
{
}
void main()
{
clrscr();
if(ReadDat()){
printf("数据文件 ENG.IN 不能打开! \n\\007\\");
return;
}
encryptChar();
WriteDat();
}
int ReadDat(void)
{
FILE *fp;
int i=0;
unsigned char *p;
if((fp=fopen("eng.in","r"))==NULL) return 1;
while(fgets(xx[i],80,fp)!=NULL){
p=strchr(xx[i],'\n');
if(p)*p=0;
i++;
}
maxline=i;
fclose(fp);
return 0;
}
void WriteDat(void)
{
}
```

```

FILE *fp;
int i;
fp=fopen("ps2.dat","w");
for(i=0;i<MAXLINE;i++){
printf("%s\\n",xx[i]);
fprintf(fp,"%s\\n",xx[i]);
}
fclose(fp);
}

```

注：与题 1 相似。

```

void encryptchar()
{
int i;
char *pf;
for(i=0;i<MAXLINE;i++)
{pf=xx[i];
while(*pf!=0)
{if(*pf%2==0||*pf*13%256<32)
{pf++;continue;}
*pf=*pf*11%256;
pf++;
}
}
}
void encryptChar()
{
int i,j,t;
for(i=0;i<MAXLINE;i++)
{
for(j=0;j<STRLEN(XX[I]);j++)
{
t=xx[i][j]*13%256;
if(t<=32 || t%2==0) continue;
xx[i][j]=t;
}
}
}

```

10 题 在文件 in.dat 中有 200 组数据，每组有 3 个数，每个数均是三位数。函数 ReadDat() 读取这 200 组数据存放到结构数组 aa 中，请编制函数 jsSort()，其函数的功能是：要求在 200 组数据中找出条件为每组中的第一个数大于第二个数加第三个数的之和，其中满足条件的个数作为函数 jsSort() 的返回值，同时把满足条件的数据存入结构数组 bb 中，再对 bb 中的数据按照每组数据的第一个数加第三个之和的大小进行升序排列(第一个数加第三个数的和均不相等)，排序后的结果仍重新存入结构数组 bb 中，最后调用函数 WriteDat()把结果 bb 输出

到文件 out.dat 中。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include
#include
#include
typedef struct{
int x1,x2,x3;
}data;
data aa[200],bb[200];
int jsSort()
{
}
void main()
{
int count;
readDat();
count=jsSort();/*返回满足条件的个数*/
writeDat(count);
}
readDat(int count)
{
FILE *in;
int i;
in=fopen("in.dat","r");
for(i=0; i<200; i++)
fscanf(in,"%d%d%d",&aa[i].x1,&aa[i].x2,&aa[i].x3);
fclose(in);
}
writeDat()
{
FILE *out;
int i;
clrscr();
out=fopen("out.dat","w");
for(i=0; i<10; i++){
printf("%d,%d,%d 第一个数+第三个数=%d\n",bb[i].x1,bb[i].x2,bb[i].x3,bb[i].x1+bb[i].x3);
fprintf(out,"%d%d%d\n",bb[i].x1,bb[i].x2,bb[i].x3);
}
fclose(out);
}
```

注：最后排序采用冒泡法。

```
int jsSort()
```

```

{
int i,j,k=0;
DATA swap;
for(i=0;i<200;i++)
if(aa[i].x1>(aa[i].x2+aa[i].x3))
bb[k++]=aa[i];
for(i=0;i<K-1;i++)
for(j=i+1;j<K;j++)
if((bb[i].x1+bb[i].x3)<(bb[j].x1+bb[j].x3))
{
swap=bb[i];
bb[i]=bb[j];
bb[j]=swap;
}
return k;
}

```

11 题 已知在文件 IN.DAT 中存有 100 个产品销售记录,每个产品销售记录由产品代码 dm(字符型 4 位), 产品名称 mc(字符型 10 位), 单价 dj(整型), 数量 sl(整型), 金额 je(长整型)四部分组成。其中: 金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat(), 其功能要求: 按产品代码从大到小进行排列, 若产品代码相同, 则按金额从大到小进行排列, 最终排列结果仍存入结构数组 sell 中, 最后调用函数 WriteDat()把结果输出到文件 OUT10.DAT 中。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
#include
#include
#define MAX 100
typedef struct{
char dm[5]; /*产品代码*/
char mc[11]; /*产品名称*/
int dj; /*单价*/
int sl; /*数量*/
long je; /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();
void SortDat()
{
}
void main()

```



```

{
memset(sell,0,sizeof(sell));
ReadDat();
SortDat();
WriteDat();
}
void ReadDat()
{
FILE *fp;
char str[80],ch[11];
int i;
fp=fopen("IN.DAT","r");
for(i=0;i<100;i++){
fgets(str,80,fp);
memcpy(sell[i].dm,str,4);
memcpy(sell[i].mc,str+4,10);
memcpy(ch,str+14,4);ch[4]=0;
sell[i].dj=atoi(ch);
memcpy(ch,str+18,5);ch[5]=0;
sell[i].sl=atoi(ch);
sell[i].je=(long)sell[i].dj*sell[i].sl;
}
fclose(fp);
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen("OUT10.DAT","w");
for(i=0;i<100;i++){
fprintf(fp,"%s %s %4d %5d %101d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
}
fclose(fp);
}
}

```

注：这时也是采用冒泡法进行排序。与前面的冒泡法在写法上有所不同请注意区分。

```

void SortDat()
{
int i,j;
PRO swap;
for(i=0;i<MAX-1;i++)
for(j=0;j<MAX-1-i;j++)
{ if(strcmp(sell[j].dm,sell[j+1].dm)<0)
{swap=sell[j];

```

```

sell[j]=sell[j+1];
sell[j+1]=swap;
}
if(strcmp(sell[j].dm,sell[j+1].dm)==0&&sell[j].je<SELL[J+1]
.je)
{
swap=sell[j];
sell[j]=sell[j+1];
sell[j+1]=swap;
}
}
}

```

12 题 函数 ReadDat()实现从文件 in.dat 中读取 20 行数据存放到字符串数组 xx 中(第行字符串长度均小于 80)。请编制函数 jsSort(), 其函数的功能是: 以行为单位对字符串变量的下标为奇数的字符按其 ASCII 值从小到大的顺序进行排序, 排序后的结果仍按行重新存入字符串数组 xx 中, 最后调用函数 WriteDat()把结果 xx 输出到文件 out.dat 中。

例如: 位置 0 1 2 3 4 5 6 7
源字符串 h g f e d c b a

则处理后字符串 h a f c d e b g

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
char xx[20][80];
void jsSort()
{
}
void main()
{
readDat();
jsSort();
writeDat();
}
readDat()
{
FILE *in;
int i=0;
char *p;
in=fopen("in.dat","r");
while(i<20&&fgets(xx[i],80,in)!=NULL){
p=strchr(xx[i],'\n');
if(p)*p=0;
i++;
}
}

```

```
}
fclose(in);
}
writeDat()
{
FILE *out();
int i;
out=fopen("out.dat","w");
clrscr();
for(i=0;i<20;i++){
printf("%s\\n",xx[i]);
fprintf(out,"%s\\n",xx[i]);
}
fclose(out);
}
```

注：该题亦是采用冒泡法，但在写法上与 11 题不同。

```
void jsSort()
{
int i,j,k,trl;
char temp;
for(i=0;i<20;i++)
{
trl=strlen(xx[i]);
for(k=1;k
for(j=k+2;j
if(xx[i][k]>xx[i][j])
{
temp=xx[i][k];
xx[i][k]=xx[i][j];
xx[i][j]=temp;
}
}
}
void jsSort()
{
int i,j,k,trl;
char ch;
for(i=0;i<20;i++)
{
trl=strlen(xx[i]);
for(j=1;j
for(k=j+2;k
if(xx[i][j]>xx[i][k])
```

```

{
ch=xx[i][j];
xx[i][j]=xx[i][k];
xx[i][k]=ch;
}
}
}

```

13 题

已知在文件 IN.DAT 中存有 100 个产品销售记录，每个产品销售记录由产品代码 dm(字符型 4 位)，产品名称 mc(字符型 10 位)，单价 dj(整型)，数量 sl(整型)，金额 je(长整型)四部分组成。其中：金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat()，其功能要求：按产品代码从大到小进行排列，若产品代码相同，则按金额从大到小进行排列，最终排列结果仍存入结构数组 sell 中，最后调用函数 WriteDat()把结果输出到文件 OUT8.DAT 中。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
#include
#include
#define MAX 100
typedef struct{
char dm[5]; /*产品代码*/
char mc[11]; /*产品名称*/
int dj; /*单价*/
int sl; /*数量*/
long je; /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();
void SortDat()
{
}
void main()
{
memset(sell,0,sizeof(sell));
ReadDat();
SortDat();
WriteDat();
}
void ReadDat()
{

```

```

FILE *fp;
char str[80],ch[11];
int i;
fp=fopen("IN.DAT","r");
for(i=0;i<100;i++){
fgets(str,80,fp);
memcpy(sell[i].dm,str,4);
memcpy(sell[i].mc,str+4,10);
memcpy(ch,str+14,4);ch[4]=0;
sell[i].dj=atoi(ch);
memcpy(ch,str+18,5);ch[5]=0;
sell[i].sl=atoi(ch);
sell[i].je=(long)sell[i].dj*sell[i].sl;
}
fclose(fp);
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen("OUT8.DAT","w");
for(i=0;i<100;i++){
fprintf(fp,"%s %s %4d %5d %101d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
}
fclose(fp);
}

```

注：该题的冒泡法与 11 题不同。

```

void SortDat()
{
int i,j;
PRO swap;
for(i=0;i<MAX-1;i++)
for(j=i+1;j<MAX;j++)
{
if(strcmp(sell[i].mc,sell[j].mc)<0)
{
swap=sell[i];
sell[i]=sell[j];
sell[j]=swap;
}
if(strcmp(sell[i].mc,sell[j].mc)==0&&sell[i].je<SELL[J].JE)
{
swap=sell[i];

```

```

sell[i]=sell[j];
sell[j]=swap;
}
}
}

```

14 题

请编写一个函数 `jsValue(int m,int k,int xx[])`，该函数的功能是：将大于整数 `m` 且紧靠 `m` 的 `k` 个素数存入数组 `xx` 传回。

最后调用函数 `writeDat()` 读取 10 组数据，分别得出结果且把结果输出到文件 `out.dat` 中。

例如：若输入 17，5，则应输出：19，23，29，31，37。

请勿改动主函数 `main()` 和写函数 `writeDat()` 的内容。

```

#include
void jsValue(int m,int k,int xx[])
{
}
main()
{
int m,n,zz[100];
printf("\n 请输入两个整数: \n");
scanf("%d%d",&m,&n);
jsValue(m,n,zz);
for(m=0;m<N;M++)PRINTF("%D",ZZ[M]);
printf("\n\n");
writeDat();
}
writeDat()
{
int m,n,zz[100],i;
FILE *.in,*out;
in=fopen("in.dat","r");
out=fopen("out.dat","w");
for(i=0;i<10;i++){
fscanf(in,"%d%d",&m,&n);
jsValue(m,n,zz);
for(m=0;m<N;M++)FPRINTF(OUT,"%D",ZZ[M]);
fprintf(out,"\n\n");
}
fclose(in);
fclose(out);
}

```

注：该题主要是对要素数方法的理解。

```

void jsvalue(int m,int k,int xx[])
{

```

```

int i,j,s=0;
for(i=m+1;k>0;i++)
{for(j=2;j<1;J++)
if(i%j==0) break;
if(i==j)
{xx[s++]=i;k--;}
}
}
}

```

15 题

已知在文件 IN.DAT 中存有 100 个产品销售记录，每个产品销售记录由产品代码 dm(字符型 4 位)，产品名称 mc(字符型 10 位)，单价 dj(整型)，数量 sl(整型)，金额 je(长整型)四部分组成。其中：金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat()，其功能要求：按产品代码从大到小进行排列，若产品代码相同，则按金额从大到小进行排列，最终排列结果仍存入结构数组 sell 中，最后调用函数 WriteDat()把结果输出到文件 OUT6.DAT 中。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
#include
#include
#define MAX 100
typedef struct{
char dm[5]; /*产品代码*/
char mc[11]; /*产品名称*/
int dj; /*单价*/
int sl; /*数量*/
long je; /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();
void SortDat()
{
}
void main()
{
memset(sell,0,sizeof(sell));
ReadDat();
SortDat();
WriteDat();
}
void ReadDat()

```

```

{
FILE *fp;
char str[80],ch[11];
int i;
fp=fopen("IN.DAT","r");
for(i=0;i<100;i++){
fgets(str,80,fp);
memcpy(sell[i].dm,str,4);
memcpy(sell[i].mc,str+4,10);
memcpy(ch,str+14,4);ch[4]=0;
sell[i].dj=atoi(ch);
memcpy(ch,str+18,5);ch[5]=0;
sell[i].sl=atoi(ch);
sell[i].je=(long)sell[i].dj*sell[i].sl;
}
fclose(fp);
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen("OUT6.DAT","w");
for(i=0;i<100;i++){
fprintf(fp,"%s %s %4d %5d %101d\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
}
fclose(fp);
}

```

注：也是考对排序法的运用。

```

void SortDat()
{
int i,j;
PRO swap;
for(i=0;i<MAX-1;i++)
for(j=i+1;j<MAX;j++)
{
if(strcmp(sell[i].mc,sell[j].mc)>0)
{
swap=sell[i];
sell[i]=sell[j];
sell[j]=swap;
}
if(strcmp(sell[i].mc,sell[j].mc)==0&&sell[i].je<SELL[J].JE)
{

```



```

swap=sell[i];
sell[i]=sell[j];
sell[j]=swap;
}
}
}

```

16 题

下列程序的功能是：在三位整数(100 至 999)中寻找符合条件的整数并依次从小到大存入数组中；它既是完全平方数，又是两位数字相同，例如 144、676 等。

请编制函数实现此功能，满足该条件的整数的个数通过所编制的函数返回。

最后调用函数 writeDat()把结果输出到文件 out.dat 中。

请勿改动主函数 main()和写函数 writeDat()的内容。

```

#include
int jsValue(int bb[])
{
}
main
{
int b[20],num;
num=jsValue(b);
writeDat(num,b);
}
writeDat(int num,int b[])
{
FILE *out;
int i;
out=fopen("out.dat","w");
fprintf(out,"%d\n",num);
for(i=0;i<NUM;i++)PRINTF(OUT,"%D\\N",B[I]);
fclose(out);
}

```

注：注意在 $i=(\text{int})\sqrt{i}*(\text{int})\sqrt{i}$ 中只有当 i 是完全平方数时开平方后再取整才不会丢失任何数据。

```

int jsvalue(int bb[])
{
int i,j,k=0,g,s,b;
for(i=100;i<=999;i++)
{
g=i%10;
s=i/10%10;
b=i/100;
if((i==(int)sqrt(i)*(int)sqrt(i))&&(g==s//s==b//b==g))
bb[k++]=i;
}
}

```

```

}
return k;
}

```

17 题

已知在文件 IN.DAT 中存有 100 个产品销售记录，每个产品销售记录由产品代码 dm(字符型 4 位)，产品名称 mc(字符型 10 位)，单价 dj(整型)，数量 sl(整型)，金额 je(长整型)四部分组成。其中：金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat()，其功能要求：按金额从大到小进行排列，若金额相同，则按产品代码从大到小进行排列，最终排列结果仍存入结构数组 sell 中，最后调用函数 WriteDat()把结果输出到文件 OUT4.DAT 中。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
#include
#include
#define MAX 100
typedef struct{
char dm[5]; /*产品代码*/
char mc[11]; /*产品名称*/
int dj; /*单价*/
int sl; /*数量*/
long je; /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();
void SortDat()
{
}
void main()
{
memset(sell,0,sizeof(sell));
ReadDat();
SortDat();
WriteDat();
}
void ReadDat()
{
FILE *fp;
char str[80],ch[11];
int i;
fp=fopen("IN.DAT","r");

```

```

for(i=0;i<100;i++){
fgets(str,80,fp);
memcpy(sell[i].dm,str,4);
memcpy(sell[i].mc,str+4,10);
memcpy(ch,str+14,4);ch[4]=0;
sell[i].dj=atoi(ch);
memcpy(ch,str+18,5);ch[5]=0;
sell[i].sl=atoi(ch);
sell[i].je=(long)sell[i].dj*sell[i].sl;
}
fclose(fp);
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen("OUT4.DAT","w");
for(i=0;i<100;i++){
fprintf(fp,"%s %s %4d %5d %101d\\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
}
fclose(fp);
}

```

注：

```

void SortDat()
{
int i,j;
PRO swap;
for(i=0;i<MAX-1;i++)
for(j=i+1;j<MAX;j++)
{
if(sell[i].je<SELL[j].JE)
{
swap=sell[i];
sell[i]=sell[j];
sell[j]=swap;
}
if(strcmp(sell[i].dm,sell[j].dm)<0&&sell[i].je==sell[j].je)
{
swap=sell[i];
sell[i]=sell[j];
sell[j]=swap;
}
}
}

```

}

18 题 下列程序的功能是：将一正整数序列 {K1,K2,...,K9} 重新排列成一个新的序列。新序列中，比 K1 小的数都在 K1 的前面(左面)，比 K1 大的数都在 K1 的后面(右面)。要求编写函数 jsValue() 实现此功能，最后调用 writeDat() 函数所新序列输出到文件 out.dat 中。

说明：在程序中已给出了 10 个序列，每个序列有 9 个正整数，并存入数组 a[10][9] 中，分别求出这 10 个新序列。

例：序列 {6,8,9,1,2,5,4,7,3}

经重排后成为 {3,4,5,2,1,6,8,9,7}

部分源程序已给出。

请勿改动主函数 main() 和写函数 writeDat() 的内容。

```
#include
```

```
jsValue(int a[10][9])
```

```
{
```

```
}
```

```
main()
```

```
{
```

```
int a[10][9]={ {6,8,9,1,2,5,4,7,3}
```

```
{3,5,8,9,1,2,6,4,7}
```

```
{8,2,1,9,3,5,4,6,7}
```

```
{3,5,1,2,9,8,6,7,4}
```

```
{4,7,8,9,1,2,5,3,6}
```

```
{4,7,3,5,1,2,6,8,9}
```

```
{9,1,3,5,8,6,2,4,7}
```

```
{2,6,1,9,8,3,5,7,4}
```

```
{5,3,7,9,1,8,2,6,4}
```

```
{7,1,3,2,5,8,9,4,6}
```

```
};
```

```
int i,j;
```

```
jsValue(a);
```

```
for(i=0;i<10;i++){
```

```
for(j=0;j<9;j++) {
```

```
printf("%d\\",a[i][j]);
```

```
if(j<=7)printf("\\");
```

```
}
```

```
printf("\\n\\n");
```

```
}
```

```
writeDat(a);
```

```
}
```

```
writeDat(int a[10][9])
```

```
{
```

```
FILE *fp;
```

```
int i,j;
```

```
fp=fopen("out.dat","w");
```

```
for(i=0;i<10;i++){
```

```

for(j=0;j<9;j++){
fprintf(fp,"%d\\",a[i][j]);
if(j<=7)fprintf(fp, "\\");
}
fprintf(fp, "\\n\\n");
}
fclose(fp);
}

```

注：该题的算法是：从数列的最后一个数开始往前逐一移动直到完，在移动过程中与 K1 进行比较，如有大小或等于 K1 的则进行调换。调换的方式是，第一次调换的数与原最后一个数换，第二次调换的数与原倒数第二个数换，依次类推。

```

jsvalue(int a[10][9])
{
int value,i,j,k;
for(i=0;i<10;i++)
for(j=8,k=8;j>=0;j--)
if(a[i][j]>=a[i][0])
{value=a[i][k];
a[i][k--]=a[i][j];
a[i][j]=value;
}
}

```

19 题 已知在文件 IN.DAT 中存有 100 个产品销售记录,每个产品销售记录由产品代码 dm(字符型 4 位),产品名称 mc(字符型 10 位),单价 dj(整型),数量 sl(整型),金额 je(长整型)四部分组成。其中:金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat(),其功能要求:按金额从小到大进行排列,若金额相同,则按产品代码从大到小进行排列,最终排列结果仍存入结构数组 sell 中,最后调用函数 WriteDat()把结果输出到文件 OUT2.DAT 中。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
#include
#include
#define MAX 100
typedef struct{
char dm[5]; /*产品代码*/
char mc[11]; /*产品名称*/
int dj; /*单价*/
int sl; /*数量*/
long je; /*金额*/
}PRO;

```

```
PRO sell[MAX];
void ReadDat();
void WriteDat();
void SortDat()
{
}
void main()
{
memset(sell,0,sizeof(sell));
ReadDat();
SortDat();
WriteDat();
}
void ReadDat()
{
FILE *fp;
char str[80],ch[11];
int i;
fp=fopen("IN.DAT","r");
for(i=0;i<100;i++){
fgets(str,80,fp);
memcpy(sell[i].dm,str,4);
memcpy(sell[i].mc,str+4,10);
memcpy(ch,str+14,4);ch[4]=0;
sell[i].dj=atoi(ch);
memcpy(ch,str+18,5);ch[5]=0;
sell[i].sl=atoi(ch);
sell[i].je=(long)sell[i].dj*sell[i].sl;
}
fclose(fp);
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen("OUT2.DAT","w");
for(i=0;i<100;i++){
fprintf(fp,"%s %s %4d %5d %101d\\n", sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
}
fclose(fp);
}
```

注：该题考我们对排序法的应用。

```
void SortDat()
```

```

{
int i,j;
PRO swap;
for(i=0;i<MAX-1;i++)
for(j=i+1;j<MAX;j++)
{
if(sell[i].je>sell[j].je)
{
swap=sell[i];
sell[i]=sell[j];
sell[j]=swap;
}
if(strcmp(sell[i].dm,sell[j].dm)<0&&sell[i].je==sell[j].je)
{
swap=sell[i];
sell[i]=sell[j];
sell[j]=swap;
}
}
}

```

20 题 某级数的前两项 $A_1=1, A_2=1$ ，以后各项具有如下关系：

$$A_n = A_{n-2} + 2A_{n-1}$$

下列程序的功能是：要求依次对于整数 $M=100, 1000$ 和 10000 求出对应的 n 值，使其满足：

$S_n = M$ ，这里

$$S_n = A_1 + A_2 + \dots + A_n,$$

并依次把 n 值存入数组单元 $b[0], b[1]$ 和 $b[2]$ 中，请编制 `jsValue()` 函数来实现此功能，最后调用函数 `writeDat()` 把数组 $b[]$ 中的值输出到 `out.dat` 文件中。

请勿改动主函数 `main()` 和写函数 `writeDat()` 的内容。

```

#include
int b[3]
jsValue()
{
}
main()
{
jsValue();
printf("M=100,n=%d\nM=1000,n=%d\nM=10000,n=%d\n",b[0],b[1],b[2]);
writeDat()
}
writeDat()
{
FILE *fp;
fp=fopen("out.dat","w");
fprintf(fp,"%d\n%d\n%d\n",b[0],b[1],b[2]);
}

```

```
fclose(fp);
}
```

注：在该题中主要就是要正确简捷的表示出各项的关系

```
jsvalue()
{
int a1=1,a2=1,a12,sn,k=2;
sn=a1+a2;
while(1)
{a12=a1+2*a2;
if(sn<100&&sn+a12>=100) b[0]=k;
if(sn<1000&&sn+a12>=1000) b[1]=k;
if(sn<10000&&sn+a12>=10000) {b[2]=k;break;}
sn=sn+a12;
a1=a2;
a2=a12;
k++;
}
}
```

21 题

函数 ReadDat()实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 xx 中，请编制函数 CharConvA()，其函数的功能是：以行为单位把字符串中的最后一个字符的 ASCII 值右移 4 位后加最后第二个字符的 ASCII 值，得到最后一个新的字符，最后第二个字符的 ASCII 值右移 4 位后加最后第三个字符的 ASCII 值，得到最后第二个新的字符，依此类推一直处理到第二个字符，第一个字符的 ASCII 值加原最后一个字符的 ASCII 值，得到第一个新的字符，得到的新字符分别存放在原字符串对应的位置上。最后已处理的字符串仍按行重新存入字符串数组 xx 中，最后调用函数 writeDat()把结果 xx 输出到文件 OUT10.DAT 中。

原始数据文件存放的格式是：每行的宽度均小于 80 个字符，含标点符号和空格。

部分源程序已给出。

请勿改动主函数 main()和写函数 writeDat()的内容。

```
#include
#include
#include
char xx[50][80];
int maxline=0;/*文章的总行数*/
int ReadDat(void)
void WriteDat(void)
void CharConvA()
{
}
void main()
{
clrscr();
if(ReadDat()){
```



```
printf(\ 数据文件 ENG.IN 不能打开! \n\007\);
return;
}
CharConvA();
WriteDat();
}
int ReadDat(void)
{
FILE *fp;
int i=0;
char *p;
if((fp=fopen(\ IN.DAT\,\ r\ ))==NULL) return 1;
while(fgets(xx[i],80,fp)!=NULL){
p=strchr(xx[i],'\n');
if(p)*p=0;
i++;
}
maxline=i;
fclose(fp);
return 0;
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen(\ OUT10.DAT\,\ w\ );
for(i=0;i<MAXLINE;I++){
printf(\ %s\n\,xx[i]);
fprintf(fp,\ %s\n\,xx[i]);
}
fclose(fp);
}
```

注：在该题中主要是控制好每一个字符的走动，由于最后要用到原来的最后一个字符故先将其存入一个字符变量中。

```
void CharConvA(void)
{
int i,j,strlen;
char ch;
for(i=0;i<MAXLINE;I++)
{strlen=strlen(xx[i]);
ch=xx[i][strlen-1];
for(j=strlen-1;j>0;j--)
{xx[i][j]>=>4;
```

```

xx[i][j]+=xx[i][j-1];
}
xx[i][0]+=ch;
}
}

```

22 题 下列程序的功能是：找出所有 100 以内(含 100)满足 I, I+4, I+10 都是素数的整数 I(I+10 也在 100 以内)的个数 cnt 以及这些 I 之和 sum。请编写函数 countValue()实现程序要求，最后调用函数 writeDat()把结果 cnt 和 sum 输出到文件 out.dat 中。

部分源程序已给出。

请勿改动主函数 main() 和输出数据函数 writeDat() 的内容。

```

#include
int cnt,sum;
int isPrime(int number)
{
    int i,tag=1;
    for(i=2;tag && i<=number / 2; i++)
        if(number % i ==0) tag =0;
    return tag;
}
void countValue()
{
}
void main()
{
    cnt=sum=0;
    countValue();
    printf(\ 满足条件的整数的个数=%d\\n\\,cnt);
    printf(\ 满足条件的整数的和值=%d\\n\\,sum);
    writeDat();
}
writeDat()
{
    FILE *fp;
    fp=fopen(\out.dat\\, w\\ );
    fprintf(fp,\\ %d\\n%d\\n\\ ,cnt,sum);
    fclose(fp);
}

```

注：该题是较简单的编程题之一，只要注意在 for() 循环中 i 不能大于 90（由于要求 I+10 也要 100 内）即可。

```

void countvalue()
{
    int i,j;
    for(i=1;i<=90;i++)

```

```

if(isprime(i)&&isprime(i+4)&&isprime(i+10))
{cnt++;
sum+=i;
}
}

```

23 题 函数 ReadDat()实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 xx 中，请编制函数 StrCharJR()，其函数的功能是：以行为单位把字符串中所有字符的 ASCII 值右移 4 位，然后把右移后的字符 ASCII 值再加上原字符的 ASCII 值，得到新的字符仍存入原字符串对应的位置上。最后把已处理的字符串仍按行重新存入字符串数组 xx 中，最后调用函数 writeDat()把结果 xx 输出到文件 OUT8.DAT 中。

原始数据文件存放的格式是：每行的宽度均小于 80 个字符，含标点符号和空格。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 writeDat()的内容。 #include

#include

char xx[50][80];

int maxline=0;/*文章的总行数*/

int ReadDat(void)

void WriteDat(void)

void StrCharJR()

{

}

void main()

{

clrscr();

if(ReadDat()){

printf(数据文件 ENG.IN 不能打开! \n\007\);

return;

}

StrCharJR();

WriteDat();

}

int ReadDat(void)

{

FILE *fp;

int i=0;

char *p;

if((fp=fopen(IN.DAT,\r\))==NULL) return 1;

while(fgets(xx[i],80,fp)!=NULL){

p=strchr(xx[i],'\n');

if(p)*p=0;

i++;

}

maxline=i;

```

fclose(fp);
return 0;
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen("OUT8.DAT","w");
for(i=0;i<MAXLINE;i++){
printf("%s\n",xx[i]);
fprintf(fp,"%s\n",xx[i]);
}
fclose(fp);
}

```

注：这个编程题比二级考得还简单。

```

void StrCharJR(void)
{
int i,j;
for(i=0;i<MAXLINE;i++)
for(j=0;xx[i][j]!='\0';j++)
xx[i][j]+=xx[i][j]>>4;
}

```

24 题 下列程序的功能是：已知学生的记录由学号和学习成绩构成，N 名学生的数据已存入 a 数组中。找出成绩最低的学生记录(假定最低成绩的记录是唯一的)，通过形参返回。请编写函数 mm(STU a[],STU *s)实现程序要求，最后调用函数 reawriteDAT()把结果输出到文件 out.dat 中。

例如： KS01 87

KS09 67

KS11 97

则调用该函数后，输出 The lowest:KS19,67

请勿改动主函数 main()和输出数据函数 readwriteDAT()的内容。 #include

#include

#include

#define N 10

void readwriteDAT();

typedef struct ss

{char num[10]; int s;}STU;

mmm(STU a[],STU *s)

{

}

main()

{

STU a[N]={ {"A01",81},{"A02",89},{"A03",66},{"A04",87},{"A05",77}

```

{\A06\,90}{\A07\,79}{\A08\,61}{\A9\,80}{\A10\,71}},m;
int i;
clrscr();
printf(\ ***** The original data *****\n\ );
for(i=0;i
mmm(a,&m);
printf(\ ***** THE RESULT *****\n\ );
printf(\ The lowest:%s,%d\n\,m.num,m.s);
readwriteDAT();
}
viod readwriteDAT()
{
FILE *rf,*wf;
STU a[N],m;
int i;
rf=fopen(\ in.dat\ ,\ r\ );
wf=fopen(\ out.dat\ ,\ w\ );
for(i=0;i<10;i++)fscanf(rf,\ %s%d\ ,a[i].num,%a[i].s);
mm(a,&m);
fprintf(wf,\ The lowest:%s, %d\n\,m.num,m.s);
fclose(rf);
fclose(wf);
}

```

注：该题亦是较简单，只要注意对结构体变量的使用方式，元素的使用、变量之间互相赋值懂了，则这题就不在话下了，我相信你们。

```

mmm(STU a[],STU *s)
{
int i;
s->s=a[0].s;
for(i=1;i<N;i++)
if(a[i].ss)
*s=a[i];
}

```

25 题 函数 ReadDat()实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 xx 中，请编制函数 StrOL(), 其函数的功能是：以行为单位对行中以空格或标点符号为分隔的所有单词进行倒排。最后把已处理的字符串(应不含标点符号)仍按行重新存入字符串数组 xx 中，最后调用函数 writeDat()把结果 xx 输出到文件 OUT6.DAT 中。

例如：原文： You He Me

I am a student.

结果： Me He You

student a am I

原始数据文件存放的格式是：每行的宽度均小于 80 个字符，含标点符号和空格。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 writeDat()的内容。 #include

#include

#include

char xx[50][80];

int maxline=0;/*文章的总行数*/

int ReadDat(void)

void WriteDat(void)

void StrOL(void)

{

}

void main()

{

clrscr();

if(ReadDat()){

printf(\ 数据文件 ENG.IN 不能打开! \n\007\);

return;

}

StrOL();

WriteDat();

}

int ReadDat(void)

{

FILE *fp;

int i=0;

char *p;

if((fp=fopen(\ IN.DAT\,\ r\))==NULL) return 1;

while(fgets(xx[i],80,fp)!=NULL){

p=strchr(xx[i],'\n');

if(p)*p=0;

i++;

}

maxline=i;

fclose(fp);

return 0;

}

void WriteDat(void)

{

FILE *fp;

int i;

clrscr();

fp=fopen(\ OUT6.DAT\,\ w\);

for(i=0;i<MAXLINE;i++){

printf(\ %s\n\ ,xx[i]);

```

fprintf(fp, \"%s\\n\", xx[i]);
}
fclose(fp);
}

```

注：该题的主要算法是先让两字符指针都指向串尾，然后使一指针（p1）往前移动，当出现不是字母时则表示在 p1+1 与 p2 之间是一个单词，并将该单词存入一变量(t1)，最后将 t1 连接到新串中（t）；接着再往前找第二个单词，依次类推直到字符串头。由此可知新串就是原串中各单词的倒排。

```

void Str0L(void)
{
    int i,k;
    char *p1,*p2;
    char t[80],t1[80];
    for(i=0;i<MAXLINE;i++)
    {p2=p1=strchr(xx[i],'\0')-'\0';
    t[0]=t1[0]='\0';
    k=1;
    while(k)
    {
        while(isalpha(*p1)==0&&p1!=xx[i]) {p1--;p2=p1;}
        while(isalpha(*p1)&&p1>=xx[i]) p1--;
        memcpy(t1,p1+1,p2-p1);
        t1[p2-p1]=0;
        strcat(t,t1);
        strcat(t," ");
        if(p1 )
            strcpy(xx[i],t);
    }
}

```

26 题 下列程序的功能是：求出 ss 字符串中指定字符 c 的个数，并返回此值。请编写函数 int num(*char ss,char c)实现程序要求，最后调用函数 readwriteDat()把结果输出到文件 out.dat 中。

(注：大小写字母有区别)

例如，若输入字符串 ss=\\ 123412132\\,c='1'，则输出 3。

部分源程序已给出。

请勿改动主函数 main()和输出数据函数 writeDat()的内容。 #include

```

#include
#define M 81
void readwriteDAT();
int num(char *ss,char c)
{
}
main()
{

```

```

char a[M],ch;
clrscr();
printf(\\nPlease enter a string:\\ );gets(a);
printf(\\nPlease enter a char:\\ );ch=getchar();
printf(\\n\\nThe number of the char is:%d\\n\\ ,num(a,ch));
readwriteDAT();
}
viod readwriteDAT()
{
int i;
FILE *rf,*wf;
char a[M],b[M],ch;
rf=fopen(\\ in.dat\\ ,\\ r\\ );
wf=fopen(\\ out.dat\\ ,\\ w\\ );
for(i=0;i<10;i++){
fscanf(rf,\\ %s\\ ,a);
fscanf(rf,\\ %s\\ ,b);
ch=*b;
fprintf(wf,\\ %c=%d\\n\\ ,ch,num(a,ch));
}
fclose(rf);
fclose(wf);
}

```

注：太简单了。

```

int num(char *ss,char c)
{
int i=0;
while(*ss!=0)
if(*ss++==c) i++;
return i;
}

```

27 题 函数 ReadDat()实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 xx 中，请编制函数 ConverCharD()，其函数的功能是：以行为单位把字符串中的所有小写字母改写成该字母的上一个字母，如果是字母 a,则改写成字母 z。大写字母仍为大写字母，小写字母仍为小写字母，其他字符不变。最后把已处理的字符串仍按行重新存入字符串数组 xx 中，最后调用函数 writeDat()把结果 xx 输出到文件 OUT4.DAT 中。

例：原文： Adb.Bcdza

abck.LLhj

结果： Aca.Bbcyz

zabj.LLgi

原始数据文件存放的格式是：每行的宽度均小于 80 个字符，含标点符号和空格。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 writeDat()的内容。 #include


```
#include
#include
char xx[50][80];
int maxline=0;/*文章的总行数*/
int ReadDat(void)
void WriteDat(void)
void ConverCharD(void)
{
}
void main()
{
clrscr();
if(ReadDat()){
printf(\ 数据文件 ENG.IN 不能打开! \n\007\);
return;
}
ConverCharD();
WriteDat();
}
int ReadDat(void)
{
FILE *fp;
int i=0;
char *p;
if((fp=fopen(\IN.DAT\,\r\))==NULL) return 1;
while(fgets(xx[i],80,fp)!=NULL){
p=strchr(xx[i],'\n');
if(p)*p=0;
i++;
}
maxline=i;
fclose(fp);
return 0;
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen(\OUT4.DAT\,\w\);
for(i=0;i<MAXLINE;i++){
printf(\ %s\n\,xx[i]);
fprintf(fp,\ %s\n\,xx[i]);
}
fclose(fp);
```

```
}
-----
```

注：*(p-1)=(*(p-1)-'a'+26-1)%26+'a' 该表达式是求出小写字母*(p-1)的前一个字母的算法。如果是求大写字母则将'a'改成'A'即可，如果是求前 4 个字母（a 换成 w 等）则将-1 改成-4，如果是求后面的字母则将减号改成加号即可。在该题中由于 while (*p++)判断完后 p 向后移动一个字符，因而在 if()中要用*(p-1)转换前一个字符。

```
void convertchard(void)
{
    char *p;
    int i;
    for(i=0;i<MAXLINE;i++)
    {p=xx[i];
    while(*p++)
    if(islower(*(p-1))) *(p-1)=(*(p-1)-'a'+26-1)%26+'a';
    }
}
```

28 题 下列程序的功能是：将大于整数 m 且紧靠 m 的 k 个素数存入数组 xx。请编写函数 num(int m,int k,int xx[])实现程序的要求，最后调用函数 readwriteDat()把结果输出到文件 out.dat 中。

例如：若输入 17,5，则应输出：19,23,29,31,37。

部分源程序已给出。

请勿改动主函数 main()和输出数据函数 writeDat()的内容。 #include

```
#include
void readwriteDAT();
int isP(int m)
{
    int i;
    for(i=2;i<M;i++)
    if(m % i==0)return 0;
    return 1;
}
void num(int m,int k,int xx[])
{
}
main()
{
    int m,n,xx[1000];
    clrscr();
    printf("\nPlease enter two integers:\n");
    scanf("%d%d",&m,&n);
    num(m,n,xx);
    for(m=n;m<N;M++)
    printf("%d\n",xx[m]);
    printf("\n\n");
}
```

```

readwriteDAT();
}
viod readwriteDAT()
{
int m,n,xx[1000],i;
FILE *rf,*wf;
rf=fopen("in.dat","r");
wf=fopen("out.dat","w");
for(i=0;i<10;i++){
fscanf(rf,"%d%d",&m,&n);
num(m,n,xx);
for(m=n;m
fprintf(wf,"%d\n");
}
fclose(rf);
fclose(wf);
}

```

注：太简单。

```

void num(int m,int k,int xx[])
{
int i=0;
for(m=m+1;k>0;m++)
if(isp(m))
{xx[i++]=m;
k--;
}
}

```

29 题 函数 ReadDat()实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 xx 中，请编制函数 SortCharD(), 其函数的功能是：以行为单位对字符按从大到小的顺序进行排序，排序后的结果仍按行重新存入字符串数组 xx 中，最后调用函数 writeDat()把结果 xx 输出到文件 OUT2.DAT 中。

例：原文：dAe,BfC.

CCbbAA

结果：fedCBA.

bbCCAA

原始数据文件存放的格式是：每行的宽度均小于 80 个字符，含标点符号和空格。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 writeDat()的内容。 #include

#include

#include

char xx[50][80];

int maxline=0;/*文章的总行数*/

int ReadDat(void)

```
void WriteDat(void)
void SortCharD(void)
{
}
void main()
{
clrscr();
if(ReadDat()){
printf(\ 数据文件 ENG.IN 不能打开! \n\007\);
return;
}
SortCharD();
WriteDat();
}
int ReadDat(void)
{
FILE *fp;
int i=0;
char *p;
if((fp=fopen(\ IN.DAT\ ,\ r\ ))==NULL) return 1;
while(fgets(xx[i],80,fp)!=NULL){
p=strchr(xx[i],'\n');
if(p)*p=0;
i++;
}
maxline=i;
fclose(fp);
return 0;
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen(\ OUT2.DAT\ ,\ w\ );
for(i=0;i<MAXLINE;i++){
printf(\ %s\n\ ,xx[i]);
fprintf(fp,\ %s\n\ ,xx[i]);
}
fclose(fp);
}
```

注：该题采用的是起（冒）泡法进行排序。

```
void sortchard(void)
{
```

```

int i,j,k,strt;
char ch;
for(i=0;i<MAXLINE;I++)
{
    strt=strlen(xx[i]);
    for(j=0;j<STRL-1;J++)
    for(k=0;k<STRL-1-I;K++)
    if(xx[i][k]<XX[I][K+1])
    {
        ch=xx[i][k];
        xx[i][k]=xx[i][k+1];
        xx[i][k+1]=ch;
    }
}
}

```

30 题 下列程序的功能是：把 s 字符串中的所有字母改写成该字母的下一个字符，字母 z 改写成字母 a。要求大写字母仍为大写字母，小写字母仍为小写字母，其他字符不做改变。请编写函数 `chg(char *s)` 实现程序的要求，最后调用函数 `readwriteDat()` 把结果输出到文件 `out.dat` 中。

例如：s 字符串中原有的内容为：Mn.123Zxy，则调用该函数后，结果为 No.123Ayz。

部分源程序已给出。

请勿改动主函数 `main()` 和输出数据函数 `writeDat()` 的内容。 #include

```

#include
#include
#include
#define N 81
void readwriteDAT();
void chg(char *s)
{
}
main()
{
    char a[N];
    clrscr();
    printf("\ Enter a string:\ ");gets(a);
    printf("\ The original string is;\ ");puts(a);
    chg(a);
    printf("\ The string after modified:\ ");
    puts(a);
    readwriteDAT();
}
viod readwriteDAT()
{
    int i;
    char a[N];
    FILE *rf,*wf;

```

```

rf=fopen("in.dat","r");
wf=fopen("out.dat","w");
for(i=0;i<10;i++){
fscanf(rf,"%s",a);
chg(a);
fprintf(wf,"%s\n",a);
}
fclose(rf);
fclose(wf);
}

```

注：该题的思路比较明显，先判断是否是'z'或'Z'，若是则转成相应的'a'或'A'，否则再判断是不是字母（已除'z'或'Z'）。

```

void chg(char *s)
{
int i,j,k;
k=strlen(s);
for(i=0;i<K;i++){
if(s[i]=='z'//s[i]=='Z')
s[i]='a';
else
if(isalpha(s[i]))
s[i]='A';
}
}

```

31 题 已知数据文件 IN.DAT 中存有 300 个四位数，并已调用读函数 ReadDat()把这些数存入数组 a 中，请编制一函数 jsValue()，其功能是：求出千位数上的数加个位数等于百位数上的数加十位数上的数的个数 cnt，再求出所有满足此条件的四位数平均值 pjz1，以及不满足此条件的四位数平均值 pjz2，最后调用写函数把结果输出到 OUT.DAT 文件。

例如：6712,6+2=7+1，则该数满足条件计算平均值 pjz1，且个数 cnt=cnt+1。8129,8+9>1+2，则该数不满足条件计算平均值 pjz2。

部分源程序已给出。

程序中已定义数组：a[300]，已定义变量：cnt,pjz1,pjz2

请勿改动主函数 main()、读函数 ReadDat()和写函数 writeDat()的内容。

```

#include
int a[300],cnt=0;
double pjz1=0.0,pjz2=0.0;
jsValue()
{
}
main()
{
int i;
readDat();

```

```

jsValue();
writeDat();
printf(\ cnt=%d\n 满足条件的平均值 pjz1=%7.21f\n 不满足条件的平均值
pjz2=%7.21f\n\,cnt,pjz1,pjz2);
}
readDat()
{
FILE *fp;
int i;
fp=fopen(\in.dat\,\r\ );
for(i=0;i<300;i++)fscanf(fp,\ %d\,&a[i]);
fclose(fp);
}
writeDat()
{
FILE *fp;
int i;
fp=fopen(\out.dat\,\w\ );
fprintf(fp,\ %d\n%7.21f\n%7.21f\n\,cnt,pjz1,pjz2);
fclose(fp);
}

```

注：该题的关键在于会不会取出一个数的个、十、百、千位上的数。a[i]%10 对 10 求余结果为个位数，a[i]%100/10 先对 100 求余得出后两位数然后再除 10，由于为整数因此得出上一个后两位数的第一位。依此类推。*/

```

jsvalue()
{
int i,g,s,b,q,k=0;
for(i=0;i<300;i++)
{g=a[i]%10;
s=a[i]%100/10;
b=a[i]/100%10;
q=a[i]/1000;
if((q+g)==(s+b)) {cnt++;pjz1+=a[i];}
else {k++;pjz2+=a[i];}
}
pjz1/=cnt;
pjz2/=k;
}

```

32 题 现有一个 10 个人 100 行的选票数据文件 IN.DAT，其数据存放的格式是每条记录的长度均为 10 位，第一位表示第一个人的选中情况，第二位表示第二个人的选中情况，依此类推：内容均为字符 0 和 1，1 表示此人被选中，0 表示此人未被选中，若一张选票人数小于等于 5 个人时被认为无效的选票。给定函数 ReadDat()的功能是把选票数据读入到字符串数组 xx 中。请编制函数 CountRs()来统计每个人的选票数并把得票数依次存入 yy[0]到 yy[9]

中。把结果 yy 输出到文件 OUT.DAT 中。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 writeDat()的内容。 #include

```
char xx[100][11];
int yy[10]
int ReadDat(void)
void WriteDat(void)
void CountRs(void)
{
}
void main()
{
int i;
for(i=0;i<10;i++)yy[i]=0;
if(ReadDat()){
printf(\ 选票数据文件 IN.DAT 不能打开! \n\007\);
return;
}
CountRs();
WriteDat();
}
int ReadDat(void)
{
FILE *fp;
int i;
if((fp=fopen(\ IN.DAT\,\ r\ ))==NULL) return 1;
for(i=0;i<10;i++){
if(fgets(xx[i],11,fp)!=NULL)return 1;
xx[i][10]='\0';
}
fclose(fp);
return 0;
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen(\ OUT.DAT\,\ w\ );
for(i=0;i<10;i++){
fprintf(fp,\ %d\n\,yy[i]);
printf(\ 第%d 个人的选票数=%d\n\,i+1,yy[i]);
}
fclose(fp);
}
```

 /*注：题目要求将那些选了不超过 5 人的选票视为无效票，即一张选票中如果 ‘1’ 不超过 5 个则为无效选票。该题先用 for() 循环对 100 行选票进行循环，在循环内先进行用 for() 循环数出这张选票中选中的人数 (count) 再用 if() 判断，如果选中人数小于等于 5 则 continue 即判断下一张选票，否则用内嵌的 for() 对行内的数据进行判断，注意如果数据为 ‘1’ 时才对对应的人增加 1。*/

```
void countrs(void)
{
    int i,j,count;
    for(i=0;i<300;i++)
    {count=0;
    for(j=0;j<10;j++)
    if(xx[i][j]=='1')
    count++;
    if(count<=5)
    continue;
    for(j=0;j<xx[i][j];j++)
    if(xx[i][j]=='1') yy[j]++;
    }
}
```

33 题 已知数据文件 IN.DAT 中存有 300 个四位数，并已调用读函数 ReadDat() 把这些数存入数组 a 中，请编制一函数 jsValue()，其功能是：求出千位数上的数加百位数等于十位数上的数加个位数上的数的个数 cnt，再求出所有满足此条件的四位数平均值 pjz1，以及不满足此条件的四位数平均值 pjz2，最后调用写函数 writeDat() 把结果输出到 OUT.DAT 文件。

例如：7153, 7+1=5+3，则该数满足条件计算平均值 pjz1，且个数 cnt=cnt+1。8129, 8+1<2+9，则该数不满足条件计算平均值 pjz2。

部分源程序已给出。

程序中已定义数组：a[300]，已定义变量：cnt, pjz1, pjz2

请勿改动主函数 main()、读函数 ReadDat() 和写函数 writeDat() 的内容。 #include

```
int a[300], cnt=0;
double pjz1=0.0, pjz2=0.0;
jsValue()
{
}
main()
{
    int i;
    readDat();
    jsValue();
    writeDat();
    printf("\t cnt=%d\n\t 满足条件的平均值 pjz1=%7.21f\n\t 不满足条件的平均值 pjz2=%7.21f\n\t", cnt, pjz1, pjz2);
}
readDat()
```

```

{
FILE *fp;
int i;
fp=fopen("in.dat", "r");
for(i=0; i<300; i++) fscanf(fp, "%d", &a[i]);
fclose(fp);
}
writeDat()
{
FILE *fp;
int i;
fp=fopen("out.dat", "w");
fprintf(fp, "%d\n%7.21f\n%7.21f\n", cnt, pjz1, pjz2);
fclose(fp);
}

```

/* 注：与题 31 相似。*/

```

jsvalue()
{
int i, g, s, b, q, k=0;
for(i=0; i<300; i++)
{g=a[i]%10;
s=a[i]%100/10;
b=a[i]/100%10;
q=a[i]/1000;
if((q+b)==(s+g)) {cnt++; pjz1+=a[i];}
else {k++; pjz2+=a[i];}
}
pjz1/=cnt;
pjz2/=k;
}

```

34 题 设有 n 个人围坐一圈并按顺时针方向从 1 到 n 编号，从第 s 个人开始进行 1 到 m 的报数，报数到第 m 人，此人出圈，再从他的下一个人重新开始 1 到 m 的报数，如此进行下去直到所有的人都出圈为止。现要求按出圈次序，每 10 人一组，给出这 n 个人的顺序表。请考生编制函数 Josegh() 实现此功能并调用函数 WriteDat() 把结果 p 输出到文件 OUT.DAT 中。

设 $n=100, c=1, m=10$ 。

(1) 将 1 到 n 个人的序号存入一维数组 p 中；

(2) 若第 i 个人报数后出圈，则将 $p[i]$ 置于数组的倒数第 i 个位置上，而原来第 $i+1$ 个至倒数第 i 个元素依次向前移动一个位置；

(3) 重复第(2)步直至圈中只剩下 $p[1]$ 为止。

部分源程序已给出。

请勿改动主函数 main() 和输出数据函数 writeDat() 的内容。 #include

#define N 100

```

#define S 1
#define M 10
int p[100],n,s,m;
void WriteDat(void);
void Josegh(void)
{
}
void main()
{
m=M;
n=N;
s=S;
Josegh();
WriteDat();
}
void WriteDat(void)
{
int i;
FILE *fp;
fp=fopen("out.dat","w");
for(i=N-1;i>=0;i--){
printf("%4d\\",p[i]);
fprintf(fp,"%4d\\",p[i]);
if(i % 10==0){
printf("\\n\\");
fprintf(fp,"\\n\\");
}
}
fclose(fp);
}

```

/* 注：题中第一个 for() 循环是先对数组 p 赋初值。在第二个 for() 中用 i 来控制没出圈的总人数，s1=(s1+m-1)%i 的作用是找出报数后出圈人的下标，其中对 i 求余的作用是使报数按圈进行（即报到尾后又从头报），该算法在很多题目中都用到。由于求余的作用当报数正好到最后一个时 s1 为 0，故而要进行 if(s1==0) 的判断。内嵌的 for() 循环是将出圈以后的人依次往前移。*/

```

void Josegh(void)
{
int i,j,s1,w;
s1=s;
for(i=1;i<=n;i++)
p[i-1]=i;
for(i=n;i>=2;i--)
{s1=(s1+m-1)%i;

```

```

if(s1==0)
s1=i;
w=p[s1-1];
for(j=s1;j<I;J++)
p[j-1]=p[j];
p[i-1]=w;
}
}

```

35 题 已知数据文件 IN.DAT 中存有 300 个四位数，并已调用读函数 ReadDat()把这些数存入数组 a 中，请编制一函数 jsValue()，其功能是：求出千位数上的数减百位数减十位数上的数减个位数上的数大于零的个数 cnt，再求出所有满足此条件的四位数平均值 pjz1，以及不满足此条件的四位数平均值 pjz2，最后调用写函数 writeDat()把结果输出到 OUT.DAT 文件。

例如：9123,9-1-2-3>0，则该数满足条件计算平均值 pjz1，且个数 cnt=cnt+1。9812,9-8-1-2<0，则该数不满足条件计算平均值 pjz2。

部分源程序已给出。

程序中已定义数组：a[300]，已定义变量：cnt,pjz1,pjz2

请勿改动主函数 main()、读函数 ReadDat()和写函数 writeDat()的内容。 #include

```

int a[300],cnt=0;
double pjz1=0.0,pjz2=0.0;
jsValue()
{
}
main()
{
int i;
readDat();
jsValue();
writeDat();
printf("\ cnt=%d\n 满足条件的平均值 pzj1=%7.21f\n 不满足条件的平均值
pjz2=%7.21f\n",cnt,pjz1,pjz2);
}
readDat()
{
FILE *fp;
int i;
fp=fopen("in.dat","r");
for(i=0;i<300;i++)fscanf(fp,"%d",&a[i]);
fclose(fp);
}
writeDat()
{
FILE *fp;
int i;
fp=fopen("out.dat","w");

```

```
fprintf(fp, "%d\\n%7.21f\\n%7.21f\\n", cnt, pjz1, pjz2);
fclose(fp);
}
```

/* 注：参考题 31。*/

```
jsvalue()
{
int i,g,s,b,q,k=0;
for(i=0;i<300;i++)
{g=a[i]%10;
s=a[i]%100/10;
b=a[i]/100%10;
q=a[i]/1000;
if(q-b-s-g>0) {cnt++;pjz1+=a[i];}
else {k++;pjz2+=a[i];}
}
pjz1/=cnt;
pjz2/=k;
}
```

36 题 请编制函数 ReadDat()实现从文件 IN.DAT 中读取 1000 个十进制整数到数组 xx 中；请编制函数 Compute()分别计算出 xx 中奇数的个数 odd，奇数的平均值 ave1，偶数的平均值 ave2 以及所有奇数的方差 totfc 的值，最后调用函数 WriteDat()把结果输出到 OUT.DAT 文件中。

计算方差的公式如下：

$$N^2$$

$$\text{totfc} = 1/N \sum_{i=1}^N (xx[i] - \text{ave1})^2$$

设 N 为奇数的个数，xx[i]为奇数，ave1 为奇数的平均值。

原始数据文件存放的格式是：每行存放 10 个数，并用逗号隔开。(每个数均大于 0 且小于等于 2000)

部分源程序已给出。

请勿改动主函数 main()和输出数据函数 writeDat()的内容。 #include

```
#include
#include
#define MAX 1000
int xx[MAX], odd=0, even=0;
double ave1=0.0, ave2=0.0, totfc=0.0;
void WriteDat(void);
int ReadDat(viod)
{
FILE *fp;
if((fp=fopen("IN.DAT", "r"))==NULL) return 1;
fclose(fp);
return 0;
```

```

}
void Compute(void)
{
}
void main()
{
int i;
for(i=0;i
if(ReadDat()){
printf(\ 数据文件 IN.DAT 不能打开!\007\n\ );
return;
}
Compute()
printf(\ ODD=%d\nOVEN=%d\nAVER=%lf\nTOTFC=%lf\n\ ,odd,ave1,ave2,totfc);
writeDat();
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen(\OUT.DAT\,\w\ );
fprintf(fp,\ %d\n%lf\n%lf\n%lf\n\ ,odd,ave1,ave2,totfc);
fclose(fp);
}

```

/* 注：注意该题还要编制读函数 readdat()，在读函数中格式输入符的一定要加逗号。

在 compute()函数中算方差的方式也要注意哦！*/

```

int ReadDat(void)
{
FILE *fp ;
int i;
if((fp=fopen("in.dat","r"))==NULL) return 1;
for(i=0;i<MAX;i++)
{fscanf(fp,"%d",&xx[i]);
if(feof(fp)) break;
}
fclose(fp);
return 0 ;
}
void Compute(void)
{
int i,yy[1000];
for(i=0;i<MAX;i++)
if(xx[i]%2)

```

```

{odd++;
ave1+=xx[i];
yy[odd-1]=xx[i];
}
else
{even++;
ave2+=xx[i];
}
ave1/=odd;
ave2/=even;
for(i=0;i<ODD;I++)
totfc+=(yy[i]-ave1)*(yy[i]-ave1)/odd;
}

```

37 题 已知数据文件 IN.DAT 中存有 300 个四位数，并已调用读函数 ReadDat()把这些数存入数组 a 中，请编制一函数 jsValue()，其功能是：求出所有这些四位数是素数的个数 cnt，再把所有满足此条件的四位数依次存入数组 b 中，然后对数组 b 的四位数按从小到大的顺序进行排序，最后调用写函数 writeDat()把结果输出到 OUT.DAT 文件。

例如：5591 是素数，则该数满足条件存入数组 b 中，且个数 cnt=cnt+1。

9812 是非素数，则该数不满足条件忽略。

部分源程序已给出。

程序中已定义数组：a[300]，b[300]，已定义变量：cnt

请勿改动主函数 main()、读函数 ReadDat()和写函数 writeDat()的内容。 #include

```

int a[300],b[300],cnt=0;
int isP(int m){
int i;
for(i=2;i<M;I++)
if(m%i==0)return 0;
return 1;
}
jsValue()
{
}
main()
{
int i;
readDat();
jsValue();
writeDat();
printf("\ cnt=%d\\n",cnt);
for(i=0;i
}
readDat()
{
FILE *fp;

```

```

int i;
fp=fopen("in.dat","r");
for(i=0;i<300;i++)fscanf(fp,"%d",&a[i]);
fclose(fp);
}
writeDat()
{
FILE *fp;
int i;
fp=fopen("out.dat","w");
fprintf(fp,"%d\\n",cnt);
for(i=0,i
fclose(fp);
}

```

/*注：由于原题中已给出判断素数的函数 isP()故可直接使用它，本题中对数组 b 的排序采用插入法。*/

```

jsvalue()
{
int i,j,temp;
for(i=0;i<300;i++)
if(isP(a[i])) b[cnt++]=a[i];
for(i=1;i<CNT;i++)
{temp=b[i];
j=i-1;
while(j>=0&&temp<b[j])
b[j+1]=b[j--];
b[j+1]=temp;
}
}

```

38、请编制函数 readdat()实现从文件 in.dat 中读取 100 个十六进制数到字符串数组 xx 中；再编制函数 h16to2(), 将 xx 中的十六进制数转换成二进制数并把已转换的二进制数仍存放在字符串数组 xx, 最后调用函数 writedat()把结果输出到 out.dat 文件中。
原始数据文件存放的格式是：每行存放 10 个数，并用逗号隔开。(每个数均大于 0 且小于等于 2000)
部分源程序已给出。

请勿改动主函数 main()和输出数据函数 writedat()的内容。

```

#include
#include
#include
#include
#define MAX 100
char xx[MAX][20];
void WriteDat(void);

```



```
int ReadDat(void)
{
FILE *fp ;
if((fp=fopen("in.dat","r"))==NULL) return 1;
fclose(fp) ;
return 0 ;
}

void H16to2(void)
{
}

void main()
{
int i ;
for(i=0;i
if(ReadDat())
{printf("Can't open the data file in.dat!\007\n") ;
return;
}
H16to2();
WriteDat();
}

void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen("out.dat", "w") ;
for(i=0;i
fclose(fp) ;
}
```

/*注：注意该题要编写 readdat()函数，其中函数 itoa(data,yy,16)的作用是将整数 dat
a 转换成十六进制型式的字符串存入 yy 中，若要转成二进制型式的字符串则用 itoa(data
,yy,2)。函数 strtol(xx[i],NULL,16)的作用是将字符型转换成整型。*/

```
int ReadDat(void)
{
FILE *fp ;
int i,data;
char yy[20];
if((fp=fopen("in.dat","r"))==NULL) return 1;
for(i=0;i<100;i++)
{fscanf(fp,"%x",&data);
itoa(data,yy,16);
strcpy(xx[i],yy);
}
```

```

fclose(fp);
return 0;
}
void H16to2(void)
{
int i,data;
char yy[20];
for(i=0;i<100;i++)
{data=strol(xx[i],NULL,16);
itoa(data,yy,2);
strcpy(xx[i],yy);
}
}

```

39 题 已知数据文件 IN.DAT 中存有 300 个四位数，并已调用读函数 ReadDat()把这些数存入数组 a 中，请编制一函数 jsValue()，其功能是：求出个位数上的数减千位数上的数减百位数上的数减十位数上的数大于零的个数 cnt，再把所有满足此条件的四位数依次存入数组 b 中，然后对数组 b 的四位数按从大到小的顺序进行排序，最后调用写函数 writeDat()把结果输出到 OUT.DAT 文件。

例如：1239， $9-1-2-3>0$ ，则该数满足条件存入数组 b 中，且个数 $cnt=cnt+1$ 。

8129， $9-8-1-2<0$ ，则该数不满足条件忽略。

部分源程序已给出。

程序中已定义数组：a[300]，b[300]，已定义变量：cnt

请勿改动主函数 main()、读函数 ReadDat()和写函数 writeDat()的内容。 #include

```

int a[300],b[300],cnt=0;
jsValue()
{
}
main()
{
int i;
readDat();
jsValue();
writeDat();
printf("\ cnt=%d\n",cnt);
for(i=0;i
}
readDat()
{
FILE *fp;
int i;
fp=fopen("in.dat","r");
for(i=0;i<300;i++)fscanf(fp,"%d",&a[i]);
fclose(fp);
}

```

```

writeDat()
{
FILE *fp;
int i;
fp=fopen("out.dat","w");
fprintf(fp,"%d\n",cnt);
for(i=0;i
fclose(fp);
}

```

/* 注：该题与题 37 相似。*/

```

jsvalue()
{
int q,b,s,g,i,j,data;
for(i=0;i<300;i++)
{q=a[i]/1000;
b=a[i]/100%10;
s=a[i]/10%10;
g=a[i]%10;
if(g-q-b-s>0)
bb[cnt++]=a[i];
}
for(i=0;i<CNT-1;i++)
for(j=i+1;j<CNT;j++)
if(bb[i]<BB[j])
{data=bb[i];
bb[i]=bb[j];
bb[j]=data;
}
}

```

040 题 请编制函数 ReadDat()实现从文件 IN.DAT 中读取 100 个十六进制整数到数组 xx 中；请编制函数 H16To10()将 xx 中的十六进制数转换成十进制数并把已转换的十进制数仍存放在字符串数组 xx,最后调用函数 WriteDat()把结果输出到 OUT.DAT 文件中。

原始数据文件存放的格式是：每行存放 10 个数，并用逗号隔开。(每个数均大于 0 且小于等于 2000)

部分源程序已给出。

请勿改动主函数 main()和输出数据函数 writeDat()的内容。 #include

#include

#include

#include

#define MAX 100

char xx[MAX][20]

void WriteDat(void)

int ReadDat(viod)

```
{
FILE *fp;
if((fp=fopen(\ IN.DAT\,\r))==NULL) return 1;
fclose(fp);
return 0;
}
void H16To10(void)
{
}
void main()
{
int i;
for(i=0;i<MAX;i++)MEMSET(XX[I],0,20);
if(ReadDat()){
printf(\ 数据文件 IN.DAT 不能打开!\007\n\ );
return;
}
H16To10()
WriteDat()
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen(\OUT.DAT\,\w\ );
for(i=0;i
fclose(fp);
}

-----
/* 注：该题与题 39 相似，只是改变函数 itoa() 的格式。*/
int ReadDat(void)
{
FILE *fp ;
int i,data;
char yy[20];
if((fp=fopen("in.dat","r"))==NULL) return 1;
for(i=0;i<100;i++)
{fscanf(fp,"%x",&data);
itoa(data,yy,16);
strcpy(xx[i],yy);
}
fclose(fp) ;
return 0 ;
}
```

```

void H16to10(void)
{
    int i,data;
    char yy[20];
    for(i=0;i<100;i++)
    {data=atoi(xx[i],16);
    itoa(data,yy,10);
    strcpy(xx[i],yy);
    }
}

```

41、已知在文件 in.dat 中存有若干个（个数<200）四位数字的正整数，函数 readdat()读取这若干个正整数并存入数组 xx 中。请编制函数 calvalue()，其功能要求：1、求出这文件中共有多少个正整数 totnum；2、求这些数右移 1 位后，产生的新数是偶数的数的个数 totcnt，以及满足此条件的这些数（右移前的值）的算术平均值 totpjz，最后调用函数 writedat()把所求的结果输出到文件 out.dat 中。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 readdat()和输出数据函数 writedat()的内容。

```

#include
#include
#define MAXNUM 200
int xx[MAXNUM];
int totnum=0;
int totcnt=0;
double totpjz=0.0;
int readdat(void);
void writedat(void);
void calvalue(void)
{
}
void main()
{
    int i;
    clrscr();
    for(i=0;i if(readdat())
    {printf("Can't open the data file in.dat!\007\n");
    return;
    }
    calvalue();
    printf("totnum=%d\n",totnum);
    printf("totcnt=%d\n",totcnt);
    printf("totpjz=%.2lf\n",totpjz);
    writedat();
}
int readdat(void)

```

```

{
FILE *fp;
int i=0;
if((fp=fopen("in.dat","r"))==NULL) return 1;
while(!feof(fp))
fscanf(fp,"%d",&xx[i++]);
fclose(fp);
return 0;
}
void writedat(void)
{
FILE *fp;
fp=fopen("out.dat","w");
fprintf(fp,"%d\n%d\n%.2lf\n",totnum,totcnt,totpjz);
fclose(fp);
}
/* 注：本题用 if(!xx[i]) break;来判断 xx[i]是否为 0，若是则跳出循环。亦是较简单。*/
void calvalue(void)
{
int i,data;
for(i=0;i<MAXNUM;i++)
{if(!xx[i]) break;
if(xx[i]>0) totnum++;
data=xx[i]>>1;
if(data%2==0)
{totcnt++;
totpjz+=xx[i];
}
}
}
totpjz/=totcnt;
}

```

42、请编制程序，从文件 in.dat 中读取 200 个整数至数组 xx 中，不计数组 xx 中值最大和最小的数（若有重复值，则都不计），再求出最大数 max 及最大数的个数 cnt1、最小数 min 及最小数的个数 cnt2 和所剩下 200-cnt1-cnt2 个数的算求平均值 pj（保留 2 位小数）。结果 max,cnt1,min,cnt2,pj 输出到 out.dat 中。

部分程序、读数据函数 read_dat(int xx[200])及输出格式已给出。

```

#include
#include
#define N 200
void read_dat(int xx[N])
{
int i,j;
FILE *fp;

```

```

fp=fopen("in.dat","r");
for(i=0;i<20;i++)
{for(j=0;j<10;j++)
{fscanf(fp,"%d",&xx[i*10+j]);
printf("%d ",xx[i*10+j]);
}
printf("\n");
}
fclose(fp);
}
void main()
{
int cnt1,cnt2,xx[N],max,min;
float pj;
FILE *fw;
clrscr();
fw=fopen("out.dat","w");
read_dat(xx);
printf("\n\nmax=%d,cnt1=%d,min=%d,cnt2=%d,pj=%6.2f\n",
max,cnt1,min,cnt2,pj);
fprintf(fw,"%d\n%d\n%d\n%d\n%d\n%6.2f\n",max,cnt1,min,cnt2,pj);
fclose(fw);
}
/* 注：该题是要在主函数中加程序行，注意本题要求去除最大和最小数后再求出去掉后
的最大数 max 和最小数 min 及相应的个数。因此程序中出现的 max0 和 min0，用来标识所
有
数中的最大数和最小数，而后求出次大数和次小数。在求 max0 和 min0 时一定要给 max 和
m
in 赋一个与 max0 和 min0 不同的数（该题用 esle max=min=xx[i]），否则会出错。对于一个
浮点数要对其保留 2 位小数时，要将其乘 100 后转换成整数，最后再将其转换成浮点数然
后除 100 即可。*/
void main()
{
int cnt1,cnt2,xx[N],max,min;
float pj;
FILE *fw;
int i,max0,min0;
clrscr();
fw=fopen("out.dat","w");
read_dat(xx);
max0=min0=xx[0];
pj=0;
cnt1=cnt2=0;
for(i=1;i<N;i++)

```

```

{if(max0 else if(min0>xx[i]) min0=xx[i];
else max=min=xx[i];
}
for(i=0;i<N;I++)
{if(max<XX[I]&&XX[I] if(min>xx[i]&&xx[i]>min0) min=xx[i];
}
for(i=0;i<N;I++)
if(xx[i]==max) cnt1++;
else if(xx[i]==min) cnt2++;
else pj+=xx[i];
pj=(float)((int)pj*100/(200-cnt1-cnt2))/100;
printf("\n\nmax=%d,cnt1=%d,min=%d,cnt2=%d,pj=%6.2f\n",
max,cnt1,min,cnt2,pj);
fprintf(fw,"%d\n%d\n%d\n%d\n%6.2f\n",max,cnt1,min,cnt2,pj);
fclose(fw);
}

```

43、请编写函数 countvalue()，它的功能是：求 n 以内（不包括 n）同时能被 3 和 7 整数的所有自然数之和的平方根 s，并作为函数值返回，最后结果 s 输出到文件 out.dat 中。

例如若 n 为 1000 时，函数值应为：s=153.909064。

部分源程序已给出。

请勿改动主函数 main() 和输入输出数据函数 progreawrite() 的内容。

```

#include
#include
#include
double countvalue(int n)
{
}
main()
{
clrscr();
printf("s=%f\n",countvalue(1000));
progreawrite();
}
progreawrite()
{
FILE *fp,*wf;
int i,n;
float s;
fp=fopen("in.dat","r");
if(fp==NULL)
{printf("Can't open the data file in.dat\007\n");
return;
}
wf=fopen("out.dat","w");

```



```

for(i=0;i<10;i++)
{fscanf(fp,"%d",&n);
s=countvalue(n);
fprintf(wf,"%f\n",s);
}
fclose(fp);
fclose(wf);
}
/* 注：由于能同时被 3 和 7 整除的自然数最小的是 21，而它们之间的间隔应是 21（由最小公倍数可得）。因而在 for() 循环可定成如下形式。*/
double countvalue(int n)

```

```

{
int i;
float s=0.0;
for(i=21;i
s+=i;
s=sqrt(s);
return s;
}

```

44、请编制程序，从文件 in.dat 中读取 200 个整数至数组 xx 中，不计数组 xx 绝对值最大的数（若有重复值，则都不计），再求出绝对值最大数 max 及绝对值最大数的个数 cnt 和所剩下 200-cnt 个数的算求平均值 pj（保留 2 位小数）。

结果 max,cnt1,min,cnt2,pj 输出到 out.dat 中。

部分程序、读数据函数 read_dat(int xx[200])及输出格式已给出。

```

#include
#include
#define N 200
void read_dat(int xx[N])
{
int i,j;
FILE *fp;
fp=fopen("in.dat","r");
for(i=0;i<20;i++)
{for(j=0;j<10;j++)
{fscanf(fp,"%d",&xx[i*10+j]);
printf("%d ",xx[i*10+j]);
}
printf("\n");
}
fclose(fp);
}
void main()
{
int cnt,xx[N],max;

```

```

float pj;
FILE *fw;
clrscr();
fw=fopen("out.dat","w");
read_dat(xx);
printf("\n\nmax=%d,cnt=%d,pj=%6.2f\n",max,cnt,pj);
fprintf(fw,"%d\n%d\n%6.2f\n",max,cnt,pj);
fclose(fw);
}
/* 注：该题的编程方法与题 42 相似。*/
void main()
{
int cnt,xx[N],max;
float pj;
FILE *fw;
int i,max0;
clrscr();
fw=fopen("out.dat","w");
read_dat(xx);
max0=abs(xx[0]);
pj=0;
cnt=0;
for(i=1;i<N;i++)
if(max0 else max=abs(xx[i]);
for(i=0;i<N;i++)
if(abs(xx[i])<MAX0&&ABS(XX[I])>max) max=abs(xx[i]);
for(i=0;i<N;i++)
if(abs(xx[i])==max) cnt++;
else pj+=xx[i];
pj=(float)((int)pj*100/(200-cnt))/100;
printf("\n\nmax=%d,cnt=%d,pj=%6.2f\n",max,cnt,pj);
fprintf(fw,"%d\n%d\n%6.2f\n",max,cnt,pj);
fclose(fw);
}

```

45、已知在文件 in.dat 中存有若干个（个数<200）四位数字的正整数，函数 readdat()读取这若干个正整数并存入数组 xx 中。请编制函数 calvalue()，其功能要求：1、求出这文件中共有多少个正整数 totnum；2、求这些数右移 1 位后，产生的新数是奇数的数的个数 totcnt，以及满足此条件的这些数（右移前的值）的算术平均值 totpjz，最后调用函数 writedat()把所求的结果输出到文件 out.dat 中。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 readdat()和输出数据函数 writedat()的内容。

```

#include
#include
#define MAXNUM 200

```

```
int xx[MAXNUM];
int totnum=0;
int totcnt=0;
double totpjz=0.0;
int readdat(void);
void writedat(void);
void calvalue(void)
{
}
totpjz/=totcnt;
}
void main()
{
int i;
clrscr();
for(i=0;i if(readdat())
{printf("Can't open the data file in.dat!\007\n");
return;
}
calvalue();
printf("totnum=%d\n",totnum);
printf("totcnt=%d\n",totcnt);
printf("totpjz=%.2lf\n",totpjz);
writedat();
}
int readdat(void)
{
FILE *fp;
int i=0;
if((fp=fopen("in.dat","r"))==NULL) return 1;
while(!feof(fp))
fscanf(fp,"%d",&xx[i++]);
fclose(fp);
return 0;
}
void writedat(void)
{
FILE *fp;
fp=fopen("out.dat","w");
fprintf(fp,"%d\n%d\n%.2lf\n",totnum,totcnt,totpjz);
fclose(fp);
}
/* 注：该题与题 41 相似。*/
void calvalue(void)
```

```

{
int i,data;
for(i=0;i<MAXNUM;i++)
{if(!xx[i]) break;
if(xx[i]>0) totnum++;
data=xx[i]>>1;
if(data%2)
{totcnt++;
totpjz+=xx[i];
}
}
totpjz/=totcnt;
}

```

46、请编制程序，从文件 in.dat 中读取 200 个整数至数组 xx 中，求出奇数的个数 cnt1 和偶数的个数 cnt2 以及数组 xx 下标为奇数的元素值的算术平均值 pj（保留 2 位小数）。

结果 cnt1,cnt2,pj 输出到 out.dat 中。

部分程序、读数据函数 read_dat(int xx[200])及输出格式已给出。

```

#include
#include
#define N 200
void read_dat(int xx[N])
{
int i,j;
FILE *fp;
fp=fopen("in.dat","r");
for(i=0;i<20;i++)
{for(j=0;j<10;j++)
{fscanf(fp,"%d",&xx[i*10+j]);
printf("%d ",xx[i*10+j]);
}
printf("\n");
}
fclose(fp);
}
void main()
{
int cnt1,cnt2,xx[N];
float pj;
FILE *fw;
clrscr();
fw=fopen("out.dat","w");
read_dat(xx);
printf("\n\ncnt1=%d,cnt2=%d,pj=%6.2f\n",cnt1,cnt2,pj);
fprintf(fw,"%d\n%d\n%6.2f\n",cnt1,cnt2,pj);
}

```

```

fclose(fw);
}
/* 注：
void main()
{
int cnt1,cnt2,xx[N];
float pj;
FILE *fw;
int i,j=0;
clrscr();
fw=fopen("out.dat","w");
read_dat(xx);
pj=0;
cnt1=cnt2=0;
for(i=0;i<N;i++)
{if(xx%2) cnt1++;
else cnt2++;
if(i%2) pj+=xx[i],j++;
}
pj=(float)((int)pj*100/j)/100;
printf("\n\ncnt1=%d,cnt2=%d,pj=%6.2f\n",cnt1,cnt2,pj);
fprintf(fw,"%d\n%d\n%6.2f\n",cnt1,cnt2,pj);
fclose(fw);
}

```

47、已知在文件 in.dat 中存有若干个（个数<200）四位数字的正整数，函数 readdat()读取这若干个正整数并存入数组 xx 中。请编制函数 calvalue()，其功能要求：1、求出这文件中共有多少个正整数 totnum；2、求出这些数中的各位数字之和是偶数的数的个数 totcnt，以及满足此条件的这些数的算术平均值 totpjz，最后调用函数 writedat()把所求的结果输出到文件 out.dat 中。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 readdat()和输出数据函数 writedat()的内容。

```

#include
#include
#define MAXNUM 200
int xx[MAXNUM];
int totnum=0;
int totcnt=0;
double totpjz=0.0;
int readdat(void);
void writedat(void);
void calvalue(void)
{
}
void main()

```

```

{
int i;
clrscr();
for(i=0;i if(readdat())
{printf("Can't open the data file in.dat!\007\n");
return;
}
calvalue();
printf("totnum=%d\n",totnum);
printf("totcnt=%d\n",totcnt);
printf("totpjz=%.2lf\n",totpjz);
writedat();
}
int readdat(void)
{
FILE *fp;
int i=0;
if((fp=fopen("in.dat","r"))==NULL) return 1;
while(!feof(fp))
fscanf(fp,"%d",&xx[i++]);
fclose(fp);
return 0;
}
void writedat(void)
{
FILE *fp;
fp=fopen("out.dat","w");
fprintf(fp,"%d\n%d\n%.2lf\n",totnum,totcnt,totpjz);
fclose(fp);
}
/* 注：该题的关键在于会不会取出一个数的个、十、百、千位上的数。a[i]%10 对 10 求
余结果为个位数，a[i]%100/10 先对 100 求余得出后两位数然后再除 10，由于为整数因此
得出上一个后两位数的第一位。依此类推。*/
void calvalue(void)
{
int i;
for(i=0;i<MAXNUM;i++)
{if(!xx[i]) break;
if(xx[i]>0) totnum++;
if((xx[i]/1000+xx[i]/100%10+xx[i]/10%10+xx[i]%10)/2==0)
{totcnt++;
totpjz+=xx[i];
}
}
}

```

```
totpjz/=totcnt;
}
```

48、请编制程序，从文件 in.dat 中读取 200 个整数至数组 xx 中，求出奇数的个数 cnt1 和偶数的个数 cnt2 以及数组 xx 值为奇数的元素值的算术平均值 pj（保留 2 位小数）。

结果 cnt1,cnt2,pj 输出到 out.dat 中。

部分程序、读数据函数 read_dat(int xx[200])及输出格式已给出。

```
#include
#include
#define N 200
void read_dat(int xx[N])
{
    int i,j;
    FILE *fp;
    fp=fopen("in.dat","r");
    for(i=0;i<20;i++)
    {for(j=0;j<10;j++)
    {fscanf(fp,"%d",&xx[i*10+j]);
    printf("%d ",xx[i*10+j]);
    }
    printf("\n");
    }
    fclose(fp);
}
void main()
{
    int cnt1,cnt2,xx[N];
    float pj;
    FILE *fw;
    clrscr();
    fw=fopen("out.dat","w");
    read_dat(xx);
    printf("\n\ncnt1=%d,cnt2=%d,pj=%6.2f\n",cnt1,cnt2,pj);
    fprintf(fw,"%d\n%d\n%6.2f\n",cnt1,cnt2,pj);
    fclose(fw);
}
/* 注：
void main()
{
    int cnt1,cnt2,xx[N];
    float pj;
    FILE *fw;
    int i;
    clrscr();
    fw=fopen("out.dat","w");
```

```

read_dat(xx);
pj=0;
cnt1=cnt2=0;
for(i=0;i<N;I++)
if(xx[i]%2) {cnt1++;pj+=xx[i];}
else cnt2++;
pj=(float)((int)pj*100/cnt1)/100;
printf("\n\ncnt1=%d,cnt2=%d,pj=%6.2f\n",cnt1,cnt2,pj);
fprintf(fw,"%d\n%d\n%6.2f\n",cnt1,cnt2,pj);
fclose(fw);
}

```

49、已知在文件 in.dat 中存有若干个（个数<200）四位数字的正整数，函数 readdat()读取这若干个正整数并存入数组 xx 中。请编制函数 calvalue()，其功能要求：1、求出这文件中共有多少个正整数 totnum；2、求出这些数中的各位数字之和是奇数的数的个数 totcnt，以及满足此条件的这些数的算术平均值 totpjz，最后调用函数 writedat()把所求的结果输出到文件 out.dat 中。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 readdat()和输出数据函数 writedat()的内容。

```

#include
#include
#define MAXNUM 200
int xx[MAXNUM];
int totnum=0;
int totcnt=0;
double totpjz=0.0;
int readdat(void);
void writedat(void);
void calvalue(void)
{
}
void main()
{
int i;
clrscr();
for(i=0;i if(readdat())
{printf("Can't open the data file in.dat!\007\n");
return;
}
calvalue();
printf("totnum=%d\n",totnum);
printf("totcnt=%d\n",totcnt);
printf("totpjz=%6.2f\n",totpjz);
writedat();
}

```



```

int readdat(void)
{
    FILE *fp;
    int i=0;
    if((fp=fopen("in.dat","r"))==NULL) return 1;
    while(!feof(fp))
        fscanf(fp,"%d",&xx[i++]);
    fclose(fp);
    return 0;
}

void writedat(void)
{
    FILE *fp;
    fp=fopen("out.dat","w");
    fprintf(fp,"%d\n%d\n%.2lf\n",totnum,totcnt,totpjz);
    fclose(fp);
}

/* 注：与 47 题相似。*/

void calvalue(void)
{
    int i;
    for(i=0;i<MAXNUM;i++)
        if(!xx[i]) break;
    if(xx[i]>0) totnum++;
    if((xx[i]/1000+xx[i]/100%10+xx[i]/10%10+xx[i]%10)/2==1)
        {totcnt++;
        totpjz+=xx[i];
        }
}

totpjz/=totcnt;
}

```

50、请编制程序，从文件 in.dat 中读取 200 个整数至数组 xx 中，不计数组 xx 最大的数（若有重复值，则都不计），再求出最大数 max 及最大数的个数 cnt 和所剩下 200-cnt 个数的算术平均值 pj（保留 2 位小数）。

结果 max,cnt1,min,cnt2,pj 输出到 out.dat 中。

部分程序、读数据函数 read_dat(int xx[200])及输出格式已给出。

```

#include
#include
#define N 200
void read_dat(int xx[N])
{
    int i,j;
    FILE *fp;
    fp=fopen("in.dat","r");

```

```

for(i=0;i<20;i++)
{for(j=0;j<10;j++)
{fscanf(fp,"%d",&xx[i*10+j]);
printf("%d ",xx[i*10+j]);
}
printf("\n");
}
fclose(fp);
}
void main()
{
int cnt,xx[N],max;
float pj;
FILE *fw;
int i,max0;
clrscr();
fw=fopen("out.dat","w");
read_dat(xx);
printf("\n\nmax=%d,cnt=%d,pj=%6.2f\n",max,cnt,pj);
fprintf(fw,"%d\n%d\n%6.2f\n",max,cnt,pj);
fclose(fw);
}
/* 注：本题要注意先去掉最大数然后最求出次大数 max 及个数 cnt。要在求 max0 时给 ma
x 任意赋一个不能等于 max0 的数，只有这样才能保证求次大数 max 时不会出错。*/
void main()
{
int cnt,xx[N],max;
float pj;
FILE *fw;
int i,max0;
clrscr();
fw=fopen("out.dat","w");
read_dat(xx);
max0=xx[0];
pj=0;
cnt=0;
for(i=1;i<N;i++)
if(max0 else max=xx[i];
for(i=0;i<N;i++)
if(max0>xx[i]&&max for(i=0;i<N;i++)
if(xx[i]==max) cnt++;
else pj+=xx[i];
pj=(float)((int)pj*100/(200-cnt))/100;
printf("\n\nmax=%d,cnt=%d,pj=%6.2f\n",max,cnt,pj);

```

```
fprintf(fw,"%d\n%d\n%6.2f\n",max,cnt,pj);
fclose(fw);
}
```

51、下列程序的功能是：计算 500-800 区间内素数的个数 cnt，并按所求素数的值从大到小的顺序，再计算其间隔加、减之各，即第 1 个素数-第 2 个素数+第 3 个素数-第 4 个素数+第 5 个素数……的值 sum。请编写函数 countvalue()实现程序的要求，最后调用函数 writ

edat()把结果 cnt 和 sum，输出到文件 out11.dat 中。

部分源程序已给出。

请勿改动主函数 main() 和输出数据函数 writedat()的内容。

```
#include
int cnt,sum;
void countValue()
{
}
void main()
{
cnt=sum=0;
countValue();
printf("cnt=%d\n",cnt);
printf("sum=%d\n",sum);
writeDAT();
}
writeDAT()
{
FILE *fp;
fp=fopen("OUT11.DAT","w");
fprintf(fp,"%d\n%d\n",cnt,sum);
fclose(fp);
}
```

/* 注：本题要求从大到小的顺序进行加减运算，因此这里采用从 800 开始循环到 500 这样就能保证出来的素数就是从大到小，内嵌的 for()再加上 if()是用来判断是否为素数。其中 k 用来控制加减运算。*/

```
void countValue()
{
int i,j,k=1;
for(i=800;i>=500;i--)
{for(j=2;j<I;j++)
if(i%j==0) break;
if(j>=i)
{cnt++;
sum+=k*i;
k=-1*k;
}
}
```

```
}
```

52、请编制程序，从文件 in.dat 中读取 200 个整数至数组 xx 中，不计数组 xx 最小的数（若有重复值，则都不计），再求出最小数 min 及最小数 cnt 和所剩下 200-cnt 个数的算求平均值 pj（保留 2 位小数）。

结果 min,cnt,pj 输出到 out.dat 中。

部分程序、读数据函数 read_dat(int xx[200])及输出格式已给出。

```
#include
#include
#define N 200
void read_dat(int xx[N])
{
    int i,j;
    FILE *fp;
    fp=fopen("in.dat","r");
    for(i=0;i<20;i++)
    {for(j=0;j<10;j++)
    {fscanf(fp,"%d",&xx[i*10+j]);
    printf("%d ",xx[i*10+j]);
    }
    printf("\n");
    }
    fclose(fp);
}
void main()
{
    int cnt,xx[N],min;
    float pj;
    FILE *fw;
    clrscr();
    fw=fopen("out.dat","w");
    read_dat(xx);
    printf("\n\nmin=%d,cnt=%d,pj=%6.2f\n",min,cnt,pj);
    fprintf(fw,"%d\n%d\n%6.2f\n",min,cnt,pj);
    fclose(fw);
}
/* 注：与 50 相似。*/
void main()
{
    int cnt,xx[N],min;
    float pj;
    FILE *fw;
    int i,min0;
    clrscr();
    fw=fopen("out.dat","w");
```

```

read_dat(xx);
min0=xx[0];
pj=0;
cnt=0;
for(i=1;i<N;I++)
if(min0>xx) min0=xx[i];
else min=xx[i];
for(i=0;i<N;I++)
if(min0<XX[I]&&MIN>xx[i]) min=xx[i];
for(i=0;i<N;I++)
if(xx[i]==min) cnt++;
else pj+=xx[i];
pj=(float)((int)pj*100/(200-cnt))/100;
printf("\n\nmin=%d,cnt=%d,pj=%6.2f\n",min,cnt,pj);
fprintf(fw,"%d\n%d\n%6.2f\n",min,cnt,pj);
fclose(fw);
}

```

53、下列程序的功能是：先出 5000 以下符合条件的自然数。条件是：千位数字与百位数字之和等于十位数字与个位数字之和，且千位数字与百位数字之和等于个位数字与千位数字之差的确 10 倍。计算并输出这些四位自然数的个数 cnt 以及这些数的和 sum。请编写函数 countvalue(0 实现程序的要求，最后调用小孩子数 writedat()把结果 cnt 和 sum，输出到文件 out13.dat 中。

```

#include
int cnt,sum;
void countValue()
{
}
void main()
{
cnt=sum=0;
countValue();
printf("cnt=%d\n",cnt);
printf("sum=%d\n",sum);
writeDAT();
}
writeDAT()
{
FILE *fp;
fp=fopen("OUT13.DAT","w");
fprintf(fp,"%d\n%d\n",cnt,sum);
fclose(fp);
}
/* 注：该题的关键亦是取出各位数字。*/
void countValue()

```

```

{
int i;
for(i=1000;i<5000;i++)
if((i/1000+i/100%10)==(i/10%10+i%10))
if((i/1000+i/100%10)==(i%10-i/1000)*10)
{cnt++;
sum+=i;
}
}

```

54、请编制程序，从文件 in.dat 中读取 200 个整数至数组 xx 中，求出奇数的个数 cnt1 和偶数的个数 cnt2 以及数级 xx 值为偶数的元素值的算术平均值 pj（保留 2 位小数）。

结果 cnt1,cnt2,pj 输出到 out.dat 中。

部分程序、读数据函数 read_dat(int xx[200])及输出格式已给出。

```

#include
#include
#define N 200
void read_dat(int xx[N])
{
int i,j;
FILE *fp;
fp=fopen("in.dat","r");
for(i=0;i<20;i++)
{for(j=0;j<10;j++)
{fscanf(fp,"%d",&xx[i*10+j]);
printf("%d ",xx[i*10+j]);
}
printf("\n");
}
fclose(fp);
}
void main()
{
int cnt1,cnt2,xx[N];
float pj;
FILE *fw;
clrscr();
fw=fopen("out.dat","w");
read_dat(xx);
printf("\n\ncnt1=%d,cnt2=%d,pj=%6.2f\n",cnt1,cnt2,pj);
fprintf(fw,"%d\n%d\n%6.2f\n",cnt1,cnt2,pj);
fclose(fw);
}
/* 注：已是很简单了吧！ */
void main()

```

```

{
int cnt1,cnt2,xx[N];
float pj;
FILE *fw;
int i;
clrscr();
fw=fopen("out.dat","w");
read_dat(xx);
cnt1=cnt2=0;
pj=0.0;
for(i=0;i<N;i++)
if(xx[i]%2) cnt1++;
else {cnt2++;pj+=xx[i];}
pj=(float)((long)(pj*100/cnt2))/100;
printf("\n\ncnt1=%d,cnt2=%d,pj=%6.2f\n",cnt1,cnt2,pj);
fprintf(fw,"%d\n%d\n%6.2f\n",cnt1,cnt2,pj);
fclose(fw);
}

```

55、下列程序的功能是：计算出自然数 SIX 和 NINE，它们满足的条件是 SIX+SIX+SIX=NINE+NINE 的个数 cnt，以及满足此条件所有的 SIX 与 NINE 的和 SUM。请编写函数 countvalue() 实现程序的要求，最后调用函数 writedat() 把结果 cnt 和 sum，输出到文件 out15.dat 中

。

其中 S，I，X，N，E 各代表一个十进制数字。

```

#include
long cnt,sum;
void countValue()
{
}
void main()
{
cnt=sum=0;
countValue();
printf("cnt=%d\n",cnt);
printf("sum=%d\n",sum);
writeDAT();
}
writeDAT()
{
FILE *fp;
fp=fopen("OUT11.DAT","w");
fprintf(fp,"%d\n%d\n",cnt,sum);
fclose(fp);
}

```

/* 注：由于有 SIX+SIX+SIX=NINE+NINE 可看出 SIX 的 3 倍必须大于等于 2000（右边是一

个

四位数字的数可知)，因此从 666 开始循环，再由三个数的和是个偶数（右边为一个数的 2 倍可知）可循环时步长为 2。再由 SIX 和 NINE 中有一个共同数字 I 所以得出第一个 if() 判

断，还有 NINE 中有一共同数字 N 所以得出第二个 if() 判断。*/

```
void countValue()
{ int i;
printf("\n");
for(i=666;i<=999;i+=2)
if(i/10%10==(3*i/2)/100%10)
{cnt++;
sum+=i+3*i/2;
}
}
```

本人只是为了上机能通过，不追求什么算法速度,答案仅供大家参考，我 9 号长春上机。

(tonggu)

```
void countValue()
{
int i;
cnt=0;
sum=0;
for(i=1000;i<1500;i++)
if(i/1000==i/10%10 && i/3*3==i)
{
cnt++;
sum+=i*4;
}
}
```

056 题 请编制程序，从文件 IN.DAT 中读取 200 个整数至数组 xx 中，求出奇数的个数 cnt1 和偶数的个数 cnt2 以及数组 xx 下标为偶数的元素值的算术平均值 pj(保留 2 位小数)。

结果 cnt1,cnt2,pj 输出到 OUT.DAT 中。

部分程序、读数据函数 read_dat(int xx[200])及输出格式已给出。 #include

#include

#define N 200

void read_dat(int xx[N])

```
{
int i,j;
FILE *fp;
fp=fopen("in.dat","r");
for(i=0;i<20;i++){
for(j=0;j<20;j++){
fscanf(fp,"%d",&xx[i*10+j]);
printf("%d\\",xx[i*10+j]);
```



```

}
printf(\\n\\);
}
fclose(fp);
}
void main()
{
int cnt1,cnt2,xx[N];
float pj;
FILE *fw;
clrscr();
fw=fopen(\\out.dat\\,\\w\\);
read_dat(xx);
printf(\\n\\ncnt1=%d,cnt2=%d,pj=%6.2f\\n\\,cnt1,cnt2,pj);
fprintf(fw,\\%d\\n\\%d\\n\\%6.2f\\n\\,cnt1,cnt2,pj);
fclose(fw);
}

```

57、下列程序的功能是：利用发下所示的简单迭代方法求方程：

$\cos(x)-x=0$ 的一个实根。

$x_{n+1}=\cos(x_n)$

迭代步骤如下：

- (1) 取 X_1 初值为 0.0;
- (2) $X_0=X_1$ ，把 X_1 的值赋给 X_0 ;
- (3) $X_1=\cos(X_0)$ ，求出一个新的 X_1 ;
- (4) 若 X_0-X_1 绝对值小 0.000001,执行步骤(5),否则执行步骤(2);
- (5)所求 X_1 就是方程 $\cos(X)-X=0$ 的一个实根,作为函数值返回。

请编写函数 countValue () 实现程序的要求，最后调用函数 WRITEDAT () 把结果输出到文件 OUT17.DAT 中。

部分源程序已给出。

请勿改动主函数 main() 和输出数据函数 writeDAT () 的内容。

```

#include
#include
#include
float countValue()
{
}
main()
{
clrscr();
printf("A=%f\n",countValue());
printf("%f\n",cos(countValue())-countValue());
writeDAT();
}
writeDAT()

```

```

{
FILE *wf;
wf=fopen("out17.dat","w");
fprintf(wf,"%f\n",countValue());
fclose(wf);
}
/* 注：该迭代法的基本算法是：反复赋值直到 x0-x1 的绝对值小于 0.00001。故而可用 d
o-while 循环。这是一个典型的算法。*/
float countValue()
{
float x0,x1=0.0;
do
{x0=x1;
x1=cos(x0);
}while(fabs(x0-x1)>=0.000001);
return x1;
}

```

58、请编制程序，从文件 in.dat 中读取 200 个整数至数组 xx 中，不计数组 xx 绝对值最小的数（若有重复值，则都不计），再求出绝对值最小数 max 及绝对值最小数的个数 cnt 和所剩下 200-cnt 个数的算求平均值 pj（保留 2 位小数）。

结果 min,cnt,pj 输出到 out.dat 中。

部分程序、读数据函数 read_dat(int xx[200])及输出格式已给出。

```

#include
#include
#define N 200
void read_dat(int xx[N])
{
int i,j;
FILE *fp;
fp=fopen("in.dat","r");
for(i=0;i<20;i++)
{for(j=0;j<10;j++)
{fscanf(fp,"%d",&xx[i*10+j]);
printf("%d ",xx[i*10+j]);
}
printf("\n");
}
fclose(fp);
}
void main()
{
int cnt,xx[N],min;
float pj;
FILE *fw;

```

```

clrscr();
fw=fopen("out.dat","w");
read_dat(xx);
printf("\n\nmin=%d,cnt=%d,pj=%6.2f\n",min,cnt,pj);
fprintf(fw,"%d\n%d\n%6.2f\n",min,cnt,pj);
fclose(fw);
}
/* 注：记住是求次最小数。*/
void main()
{
int cnt,xx[N],min;
float pj;
FILE *fw;
int i,min0;
clrscr();
fw=fopen("out.dat","w");
read_dat(xx);
min0=abs(xx[0]);
pj=0;
cnt=0;
for(i=1;i<N;i++)
if(min0>abs(xx[i])) min0=abs(xx[i]);
else min=abs(xx[i]);
for(i=0;i<N;i++)
if(abs(xx[i])>min0&&abs(xx[i]) for(i=0;i<N;i++)
if(abs(xx[i])==min) cnt++;
else pj+=xx[i];
pj=(float)((int)pj*100/(200-cnt))/100;
printf("\n\nmin=%d,cnt=%d,pj=%6.2f\n",min,cnt,pj);
fprintf(fw,"%d\n%d\n%6.2f\n",min,cnt,pj);
fclose(fw);
}

```

59、请编写函数 void countvalue(int *a,int *n)，它的功能是：求出 1 到 1000 之内能被 7 或 11 整除但不能同时被 7 和 11 整除的所有整数放在数组 a 中，并通过 n 返回这些数的个数。

```

#include
#include
void countValue(int *a,int *n)
{
}
main()
{
int aa[1000],n,k;

```

```

clrscr();
countValue(aa,&n);
for(k=0;k<N;K++)
if((k+1)%10==0) printf("\n");
else printf("%5d",aa[k]);
writeDAT();
}
writeDAT()
{
int aa[1000],n,k;
FILE *fp;
fp=fopen("out19.dat","w");
countValue(aa,&n);
for (k=0;k<N;K++)
if((k+1)%10==0) fprintf(fp,"\n");
else fprintf(fp,"%5d",aa[k]);
fclose(fp);
}
/* 注：由于 7 和 11 的最小公倍数为 77 故而可用 i%77!=0 来表示不能同时被 7 和 11 整除。
在
a[( *n)++]=i, 中必须有小括号，这是由于*与++的运算优先级别相同，而它们的结合方
向是自右向左，如不用括号则变成 n 所指的地址向前移一位，但本题要的是 n 所指的变量
的值增加 1。*/

```

```

void countValue(int *a,int *n)
{
int i;
*n=0;
for(i=7;i<1000;i++)
if((i%7==0//i%11==0)&& i%77!=0)
a[( *n)++]=i;
}

```

60、请编制程序，从文件 in.dat 中读取 200 个整数至数线 xx 中，求出最大数 max 及最大数的个数 cnt 和数组 xx 中值能被 3 整除或能被 7 整除的算求增均值 pj(保留 2 位小数)。

```

#include
#include
#include
#define N 200
void read_dat(int xx[N])
{
int i,j;
FILE *fp;
fp=fopen("in.dat","r");
for(i=0;i<20;i++)
{for(j=0;j<10;j++)

```

```

{fscanf(fp,"%d",&xx[i*10+j]);
printf("%d ",xx[i*10+j]);
}
printf("\n");
}
fclose(fp);
}
void main()
{
int cnt,xx[N],max;
float pj;
FILE *fw;
clrscr();
fw=fopen("out.dat","w");
read_dat(xx);
printf("\n\nmax=%d,cnt=%d,pj=%6.2f\n",max,cnt,pj);
fprintf(fw,"%d\n%d\n%6.2f\n",max,cnt,pj);
fclose(fw);
}
/* 注：注意本题要直接求出所有数中的最大数，而不象前面题目中的求次最大数。*/
void main()
{
int cnt,xx[N],max;
float pj;
FILE *fw;
int i,k=0;
clrscr();
fw=fopen("out.dat","w");
read_dat(xx);
max=xx[0];
pj=0;
cnt=0;
for(i=1;i<N;i++)
if(max for(i=0;i<N;i++)
{if(xx[i]==max) cnt++;
if(xx[i]%3==0//xx[i]%7==0)
{pj=xx[i];k++;}
}
pj=(float)((int)pj*100/k)/100;
printf("\n\nmax=%d,cnt=%d,pj=%6.2f\n",max,cnt,pj);
fprintf(fw,"%d\n%d\n%6.2f\n",max,cnt,pj);
fclose(fw);
}

```

61、已知数据文件 IN.DAT 中存有 300 个四位数,并已调用读函数 READDAT()把这些数存入

数

组 a 中,请编制一函数 JSVALUE(),其功能是:求出千位数上的数减百位数上的数减十位数上的数减个位数上的数大于零的个数 CNT,再把所有满足此条件的四位数依次存入数组 bb 中,然后对数组 bb 的四位数按小到大的顺序进行排序,最后调用写函数 WRITEDAT()把数组 B

中的数输出到 OUT.DAT 文件.

例如:9123,9-1-2-3>0,则该数满足条件存入数组 bb 中,且个数 CNT=CNT+1.

9812,9-8-1-2<0,则该数不满足条件忽略.

```
#include
int a[300],bb[300],cnt=0;
jsValue()
{
}
main()
{
int i;
readDat();
jsValue();
wrtDat();
for (i=0;i }
readDat()
{
FILE *fp;
int i;
fp=fopen("in.dat","r");
printf("cnt=%d\n",cnt);
for (i=0;i<300;i++) fscanf(fp,"%d",&a[i]);
fclose(fp);
}
wrtDat()
{
FILE *fp;
int i;
fp=fopen("out.dat","w");
fprintf(fp,"%d\n",cnt);
for (i=0;i
fclose(fp);
}
/* 注: 对 bb 的排序本题采用了冒泡法。*/
jsValue()
{
int q,b,s,g,i,j;
for(i=0;i<300;i++)
{q=a[i]/1000;
```

```

b=a[i]/100%10;
s=a[i]/10%10;
g=a[i]%10;
if(q-b-s-g>0)
bb[cnt++]=a[i];
}
for(i=0;i<CNT-1;i++)
for(j=i+1;j<CNT;j++)
if(bb[i]>bb[j])
{q=bb[i];
bb[i]=bb[j];
bb[j]=q;
}
}

```

62、请编制函数 READDAT () 实现从文件 IN.DAT 中读取 100 个十六进制数到字符串数组 xx

中;再编制函数 H16TO8(),将 xx 中的十六进制数转换成八进制数并把已转换的八进制数仍存放在字符串数组 XX 中,最后调用函数 WRITEDAT()把结果输出到 OUT.DAT 文件中.

原始数据文件存放的格式是:每行存放 10 个数,并用逗号隔开(每个数均大于 0 且小于等于 2000).

```

#include
#include
#include
#include
#define MAX 100
char xx[MAX][20];
void WriteDat(void);
int ReadDat(void)
{
FILE *fp;
if((fp=fopen("in.dat","r"))==NULL) return 1;
fclose(fp);
return 0;
}
void H16to8(void)
{
}
void main()
{
int i;
for(i=0;i
if(ReadDat())
{printf("Can't open the data file in.dat!\007\n");
return;
}

```

```

}
H16to8();
WriteDat();
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen("out.dat", "w") ;
for(i=0;i
fclose(fp) ;
}
/* 注： 本题中用到函数 itoa() 来实现从整型变成字符型。*/
int ReadDat(void)
{
FILE *fp ;
int i,data;
char yy[20];
if((fp=fopen("in.dat","r"))==NULL) return 1;
for(i=0;i<100;i++)
{fscanf(fp,"%x",&data);
itoa(data,yy,16);
strcpy(xx[i],yy);
}
fclose(fp) ;
return 0 ;
}
void H16to8(void)
{
int i,data;
char yy[20];
for(i=0;i<100;i++)
{data=atoi(xx[i],NULL,16);
itoa(data,yy,8);
strcpy(xx[i],yy);
}
}

```

63、已知数据文件 IN.DAT 中存有 300 个四位数,并已调用读函数 READDAT()把这些存入数组

a 中,请编制一函数 JSVALUE()其功能是:求出千位数上的数加百位数上的数等于十位数上的数加个位数上的数的个数 CNT,再把所有满足条件的四位数依次存入数组 B 中,然后对数组 bb 的四位数按从大到小的顺序进行排序,最后调用写函数 WRITEDAT()把数组 bb 中的数

输出到 OUT.DAT 文件.

例如:7153,7+1=5+3,则该数满足条件存入数组 bb 中,且个数 CNT=CNT+1.

8129,8+1<2+9,则该数不满足条件忽略.

```
#include
int a[300],bb[300],cnt=0;
jsValue()
{
}
main()
{
int i;
readDat();
jsValue();
wrtDat();
for (i=0;i }
readDat()
{
FILE *fp;
int i;
fp=fopen("in.dat", "r");
printf("cnt=%d\n", cnt);
for (i=0;i<300;i++) fscanf(fp, "%d,", &a[i]);
fclose(fp);
}
wrtDat()
{
FILE *fp;
int i;
fp=fopen("out.dat", "w");
fprintf(fp, "%d\n", cnt);
for (i=0;i
fclose(fp);
}
/* 注:
jsValue()
{
int q,b,s,g,i,j;
for(i=0;i<300;i++)
{q=a[i]/1000;
b=a[i]/100%10;
s=a[i]/10%10;
g=a[i]%10;
if(q+b==s+g)
bb[cnt++]=a[i];
}
```

```

for(i=0;i<CNT-1;i++)
for(j=i+1;j<CNT;j++)
if(bb[i]<BB[j])
{q=bb[i];
bb[i]=bb[j];
bb[j]=q;
}
}

```

64、请编制函数 READDAT () 实现从文件 IN.DAT 中读取 1000 个十进制整数到数组 XX 中;再

编制函数 COMPUTE() 分别计算出 XX 中奇数的个数 ODD,偶数的个数 EVEN,平均值`AVER 以及方

差 TOTFE 的值,最后调用函数 WRITEDAT()把结果输出到 OUT.DAT 文件中.

计算方差的公式如下:

原始数据文件存放的格式是:每行存放 10 个数,并用逗号隔开(每个数均大于 0 且小于等于 2000).

```

#include
#include
#include
#define MAX 1000
int xx[MAX],odd=0,even=0;
double aver=0.0,totfc=0.0;
void WriteDat(void);
int ReadDat(void)
{
FILE *fp;
if((fp=fopen("in.dat","r"))==NULL) return 1;
fclose(fp);
return 0;
}
void Compute(void)
{
}
void main()
{
int i;
for(i=0;i<MAX;i++)
xx[i]=0;
if(ReadDat())
{printf("Can't open the data file in.dat!\007\n");
return;
}
Compute();
printf("ODD=%d\nEVEN=%d\nAVER=%lf\nTOTFC=%lf\n", odd,even,aver,t

```

```

    totfc);
WriteDat();
}
void WriteDat(void)
{
    FILE *fp;
    int i;
    fp=fopen("out.dat", "w") ;
    fprintf(fp, "%d\n%d\n%lf\n%lf\n", odd, even, aver, totfc);
    fclose(fp) ;
}
/* 注：
int ReadDat(void)
{
    FILE *fp ;
    int i;
    if((fp=fopen("in.dat", "r"))==NULL) return 1;
    for(i=0; i<MAX; i++)
    {fscanf(fp, "%d", &xx[i]);
    if(feof(fp)) break;
    }
    fclose(fp) ;
    return 0 ;
}
void Compute(void)
{
    int i, yy[1000];
    for(i=0; i<MAX; i++)
    {aver+=xx[i];
    if(xx[i]%2)
    odd++;
    else
    even++;
    }
    aver/=(odd+even);
    for(i=0; i<MAX; i++)
    totfc+=(xx[i]-aver)*(xx[i]-aver)/(odd+even);
}

```

65、已知数据文件 IN.DAT 中存有 300 个四位数,并已调用读函数 READDAT()把这些数存入数组

组 A 中.请编制一函数 JSVALUE(),其功能是:求出千位数上的数加个位数上的数等于百位数上的数加十位数上的数的个数 CNT,再把所有满足条件的四位数依次存入数组 bb 中,然后对数组 bb 的四位数按从小到大的顺序进行排序,最后调用写函数 WRITEDAT()把数组 bb 中的数

输出到 OUT.DAT 文件.

例如:6712,6+2=7+1,则该数满足条件存入数组 bb 中,且个数 CNT=CNT+1.

8129,8+1<2+9,则该数不满足条件忽略.

```
#include
int a[300],bb[300],cnt=0;
jsValue()
{
}
main()
{
int i;
readDat();
jsValue();
wrtDat();
for (i=0;i }
readDat()
{
FILE *fp;
int i;
fp=fopen("in.dat", "r");
printf("cnt=%d\n",cnt);
for (i=0;i<300;i++) fscanf(fp, "%d",&a[i]);
fclose(fp);
}
wrtDat()
{
FILE *fp;
int i;
fp=fopen("out.dat", "w");
fprintf(fp, "%d\n",cnt);
for (i=0;i
fclose(fp);
}
/* 注:
jsValue()
{
int q,b,s,g,i,j;
for(i=0;i<300;i++)
{q=a[i]/1000;
b=a[i]/100%10;
s=a[i]/10%10;
g=a[i]%10;
if(q+g==b+s)
bb[cnt++]=a[i];
```

```

}
for(i=0;i<CNT-1;i++)
for(j=i+1;j<CNT;j++)
if(bb[i]> bb[j])
{q=bb[i];
bb[i]=bb[j];
bb[j]=q;
}
}

```

66、请编制函数 READDAT () 实现从文件 IN.DAT 中读取 1000 个十进制整数到数组 XX 中;再

编制函数 COMPUTE()分别计算出 XX 中偶数的个数 EVEN,奇数的平均值`AVE1,偶数的平均值

AVE2 以所有偶数的方差 TOTFC 的值,最后调用函数 WRITEDAT()把结果输出到 OUT.DAT 文件中

计算方差的公式如下:

设 N 为偶数的个数,XX[I]为偶数,AVE2 为偶数的平均值.

原始数据文件存放的格式是:每行存放 10 个数,并用逗号隔开(每个数均大于 0 且小于等于 2000).

```

#include
#include
#include
#define MAX 1000
int xx[MAX],odd=0,even=0;
double ave1=0.0,ave2=0.0,totfc=0.0;
void WriteDat(void);
int ReadDat(void)
{
FILE *fp;
if((fp=fopen("in.dat","r"))==NULL) return 1;
fclose(fp);
return 0;
}
void Compute(void)
{
}
void main()
{
int i;
for(i=0;i<MAX;i++)
xx[i]=0;
if(ReadDat())
{printf("Can't open the data file in.dat!\007\n");

```

```
return;
}
Compute();
printf("EVEN=%d\nAVE1=%lf\nAVE2=%lf\nTOTFC=%lf\n", even,ave1,ave
2,totfc);
WriteDat();
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen("out.dat", "w") ;
fprintf(fp, "%d\n%lf\n%lf\n%lf\n",even,ave1,ave2,totfc);
fclose(fp) ;
}
/* 注：
int ReadDat(void)
{
FILE *fp ;
int i;
if((fp=fopen("in.dat", "r"))==NULL) return 1;
for(i=0;i<MAX;I++)
{fscanf(fp, "%d,",&xx[i]);
if(feof(fp)) break;
}
fclose(fp) ;
return 0 ;
}
void Compute(void)
{
int i,yy[1000];
for(i=0;i<MAX;I++)
if(xx[i]%2)
{odd++;
ave1+=xx[i];
}
else
{even++;
ave2+=xx[i];
yy[even-1]=xx[i];
}
ave1/=odd;
ave2/=even;
for(i=0;i<EVEN;I++)
```

```
totfc+=(yy[i]-ave2)*(yy[i]-ave2)/even;
}
```

67、已知数据文件 IN.DAT 中存有 300 个四位数,并已调用读函数 READDAT()把这些数存入数

组 A 中,请编制一函数 JSVALUE(),其功能是:求出个位数上的数减千位数上的数减百位数上的数减十位数上的数大于零的个数 CNT,再把所有满足此条件的四位数平均值 PJZ1,以及不满足此条件的四位数平均值 PJZ2,最后调用写函数 WRITEDAT()把结果 CNT, PJZ, PJZ2 输

出到 OUT.DAT 文件.

例如:1239,9-1-2-3>0,则该数满足条件计算平均值 PJZ1,且个数 CNT=CNT+1.

8129,9-8-1-2<0,则该数不满足条件计算平均值 PJZ2。

```
#include
int a[300],cnt=0;
double pjz1=0.0,pjz2=0.0;
jsValue()
{
}
main()
{
int i;
readDat();
jsValue();
writeDat();
printf("cnt=%d\npjz1=%7.2lf\npjz2=%7.2lf\n",cnt,pjz1,pjz2);
}
readDat()
{
FILE *fp;
int i;
fp=fopen("in.dat", "r");
for(i=0;i<300;i++) fscanf(fp, "%d",&a[i]);
fclose(fp);
}
writeDat()
{
FILE *fp;
int i;
fp=fopen("out.dat", "w");
fprintf(fp, "%d\n%7.1lf\n%7.2lf\n", cnt, pjz1, pjz2);
fclose(fp);
}
/* 注:
jsValue()
{
```

```

int i,q,b,s,g,cnt1=0;
for(i=0;i<300;i++)
{q=a[i]/1000;
b=a[i]/100%10;
s=a[i]/10%10;
g=a[i]%10;
if(g-q-b-s>0)
{pjz1+=a[i];
cnt++;
}
else
{pjz2+=a[i];
cnt1++;
}
}
pjz1/=cnt;
pjz2/=cnt1;
}

```

68、现有一个 10 个人 100 行的选票数据文件 IN.DAT,其数据存放的格式是每条记录的长度均为 10 位,第一位表示第一个的选中情况,第二位表示第二个人的选中情况,依此类推;内容均为字符 0 和 1,1 表示此人被选中,0 表示此人未被选中,全选或不选均为无效的选票.给定函数 READDAT()的功能是把选票数据读入到字符串数组 XX 中.请编制函数 COUNTRS()来统

计每个人的选票数并把票数依次存入 YY[0]到 YY[9]中,最后调用函数 WRITEDAT()把结果 Y 输出到 OUT.DAT 中.

```

#include"stdio.h"
char xx[100][11];
int yy[10];
int readdat(void);
void writedat(void);
void countrs(void)
{
}
void main()
{
int i;
for(i=0;i<10;i++)
yy[i]=0;
if(readdat())
return;
countrs();
writedat();
}
int readdat(void)

```



```

{
FILE *fp;
int i;
if((fp=fopen("in.dat","r"))==NULL)
return 1;
for(i=0;i<100;i++)
{if(fgets(xx[i],11,fp)==NULL)
return 1;
xx[i][10]='\0';
}
fclose (fp);
return 0;
}
void writedat(void)
{
FILE *fp;
int i;
fp=fopen("out.dat","w");
for(i=0;i<10;i++)
{fprintf(fp,"%d\n",yy[i]);
printf("%d %d\n",i+1,yy[i]);
}
fclose(fp);
}
/* 注：本题要求将那些全选取或不先的选票均视为无效，全选即为“111111111”不选
即为“000000000”，只要 xx[i] 为上述两种情况即视为无效。本题对这两种情况的判断
采用 strcmp() 函数，注意字符串的比较不能直接进行即不能用：xx[i]=="111111111"
的方式。*/
void countrs(void)
{
int i,j;
for(i=0;i<300;i++)
{if(strcmp(xx[i],"111111111")==0//strcmp(xx[i],"000000000
0")==0)
continue;
for(j=0;xx[i][j];j++)
if(xx[i][j]=='1') yy[j]++;
}
}

```

69、已知数据文件 IN.DAT 中存有 300 个四位数,并已调用读函数 READDAT()把这些数存入数

组 A 中,请编制一函数 JSVALUE(),其功能是:求出所有这些四位数是素数的个数 CNT,再求出所有满足此条件的四位数平均值 PJZ1,以及不满足条件的四位数平均值 PJZ2,最后调用写函数 WRITEDAT()把结果 CNT,PJZ1,PJZ2 输出 到 OUT.DAT 文件.

例如:5591 是素数,则该数满足条件计算平均值 PJZ1,且个数 CNT=CNT+1.

9812 是非素数,则该数不满足条件计算平均值 PJZ2。

```
#include
int a[300],cnt=0;
double pjz1=0.0,pjz2=0.0;
int isP(int m)
{
    int i;
    for(i=2;i<M;i++)
        if(m%i==0) return 0;
    return 1;
}
jsValue()
{
}
main()
{
    int i;
    readDat();
    jsValue();
    writeDat();
    printf("cnt=%d\npjz1=%7.2lf\npjz2=%7.2lf\n",cnt,pjz1,pjz2);
}
readDat()
{
    FILE *fp;
    int i;
    fp=fopen("in.dat","r");
    for(i=0;i<300;i++) fscanf(fp, "%d",&a[i]);
    fclose(fp);
}
writeDat()
{
    FILE *fp;
    int i;
    fp=fopen("out.dat","w");
    fprintf(fp,"%d\n%7.1lf\n%7.2lf\n",cnt,pjz1,pjz2);
    fclose(fp);
}
/* 注：这里利用了已有的判断素数函数 isP()来直接对某个数进行判断。*/
jsValue()
{
    int i,cnt1=0;
    for(i=0;i<300;i++)
```

```

if(isP(a[i]))
{cnt++;
pjz1+=a[i];
}
else
{cnt1++;
pjz2+=a[i];
}
pjz1/=cnt;
pjz2/=cnt1;
}

```

70、现有一个 10 个人 100 行的选票数据文件 IN.DAT,其数据存放的格式是每条记录的长度均为 10 位,第一位表示第一个的选中情况,第二位表示第二个人的选中情况,依此类推;内容均为字符 0 和 1,1 表示此人被选中,0 表示此人未被选中,若一张选票人数大于 5 个人时认为无效的选票.给定函数 READDAT()的功能是把选票并把选票数据读入到字符串数组 XX 中.请编制函数 COUNTS()来统计每个人的选票数并把票数依次存入 YY[0]到 YY[9]中,最后调用函数 WRITEDAT()把结果 YY 输出到 OUT.DAT 中.

```

#include"stdio.h"
char xx[100][11];
int yy[10];
int readdat(void);
void writedat(void);
void countrs(void)
{
}
void main()
{
int i;
for(i=0;i<10;i++)
yy[i]=0;
if(readdat())
return;
countrs();
writedat();
}
int readdat(void)
{
FILE *fp;
int i;
if((fp=fopen("in.dat","r"))==NULL)
return 1;
for(i=0;i<100;i++)
{if(fgets(xx[i],11,fp)==NULL)
return 1;
}
}

```

```

xx[i][10]='\0';
}
fclose (fp);
return 0;
}
void writedat(void)
{
FILE *fp;
int i;
fp=fopen("out.dat","w");
for(i=0;i<10;i++)
{fprintf(fp,"%d\n",yy[i]);
printf("%d %d\n",i+1,yy[i]);
}
fclose(fp);
}
/* 注：本题要求将那些选了超过 5 个人的选票视为无效票，在外层 for() 循环是用来一张
一张选票地数。在循环内的第一个 for() 循环用来数一张选票中共选了几个人，第二个 i
f() 用来将选了超过 5 人的选票去掉。*/
void countrs(void)
{
int i,j,count;
for(i=0;i<300;i++)
{count=0;
for(j=0;j<10;j++)
if(xx[i][j]=='1')
count++;
if(count>5)
continue;
for(j=0;j<10;j++)
if(xx[i][j]=='1') yy[j]++;
}
}

```

71、函数 READDAT () 实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 XX 中；
请

编制函数 SORTCHARA(), 其函数功能是: 以行为单位对字符按从小到大的顺序进行排序, 排序后的结果仍按行重新存入字符串数组 XX 中, 最后调用函数 WRITEDAT() 把结果 XX 输出到文

件 OUT.DATK .

例: 原文: dAe, BfC.

CCbbAA

结果: ., ABCdef

AACCbb

原始数据文件存放的格式是: 每行的宽度均小于 80 个字符, 含标点符号和空格.

```
#include "stdio.h"
#include "string.h"
#include "conio.h"
char xx[50][80];
int maxline=0;
int readdat(void);
void writedat(void);
void sortcharA(void)
{
}
void main()
{
clrscr();
if(readdat())
{printf("Can't open the file ENG.IN!\n");
return;
}
sortcharA();
writedat();
}
int readdat(void)
{
FILE *fp;
int i=0;
char *p;
if((fp=fopen("in.dat","r"))==NULL)
return 1;
while(fgets(xx,80,fp)!=NULL)
{p=strchr(xx[i],'\n');
if(p)
*p=0;
i++;
}
maxline=i;
fclose(fp);
return 0;
}
void writedat(void)
{FILE *fp;
int i;
fp=fopen("out1.dat","w");
for(i=0;i<MAXLINE;i++)
{printf("%s\n",xx[i]);
fprintf(fp,"%s\n",xx[i]);
```

```

}
fclose(fp);
}
/* 注：本题采用冒泡法进行排序。*/
void sortcharA(void)
{
    int i,j,k,trl;
    char ch;
    for(i=0;i<MAXLINE;I++)
    {trl=strlen(xx[i]);
    for(j=0;j<STRL-1;J++)
    for(k=j+1;k<STRL;K++)
    if(xx[i][j]>xx[i][k])
    {ch=xx[i][k];
    xx[i][k]=xx[i][j];
    xx[i][j]=ch;
    }
    }
}

```

72、下列程序的功能是:把 S 字符串中的所有字符左移一个位置,串中的第一个字符移到最后.请编写函数 CHG(CHAR *S)实现程序要求,最后调用函数 READWRITEDAT()把结果输出到

OUT.DAT 文件中.

例如:S 字符串中原有内容为:MN.123XYZ,则调用函数后,结果为:N.123XYZM.

```

#include"string.h"
#include"stdio.h"
#include"ctype.h"
#define N 81
void readwritdat();
void chg(char *s)
{
}
main()
{
    char a[N];
    printf("enter a string :");
    gets(a);
    printf("the original string is :");
    puts(a);
    chg(a);
    printf("the string after modified:");
    puts(a);
    readwritdat();
}

```

```

void readwritdat()
{
    int i;
    char a[N];
    FILE *rf,*wf;
    rf=fopen("in.dat","r");
    wf=fopen("out.dat","w");
    for(i=0;i<10;i++)
    {fscanf(rf,"%s",a);
    chg(a);
    fprintf(wf,"%s\n",a);
    }
    fclose(rf);
    fclose(wf);
}
/* 注: strcpy(s,s+1)的作用是将第二个字符及以后的字符前移一位, 注意这时最后一个字符还不是原来的第一个字符, 用*(s+i-1)=c; 来将原第一个字符存入串尾, 最后还要赋一个新的串结束符。*/
void chg(char *s)
{
    int i=strlen(s);
    char c=*s;
    strcpy(s,s+1);
    *(s+i-1)=c;
    *(s+i)='\0';
}

```

73、函数 READDAT () 实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 XX 中; 请

编制函数 CONVERTCHARA(),其函数功能是:以行为单位把字符串中的所有小写字母改写成该字母的下一个字母,如果是字母 Z,则改写成字母 A.大写字母仍为大写字母,小写字母仍为小写字母,其他字符不变.把已处理的字符串仍按行重新存入字符串数组 XX 中,最后调用函数 WRITEDAT()把结果 XX 输出到文件 OUT.DAT 中.

例:原文:AdB.Bcdza

abck.LLhj

结果:Aec.Bdeab

bcdl.LLlk

原始数据文件存放的格式是:每行的宽度均小于 80 个字符,含标点符号和空格.

```

#include"stdio.h"
#include"string.h"
#include"conio.h"
char xx[50][80];
int maxline=0;
int readdat(void);
void writdat(void);

```

```
void convertcharA(void)
{
}
void main()
{
clrscr();
if(readdat())
{printf("Can't open the file ENG.IN!\n");
return;
}
convertcharA();
writedat();
}
int readdat(void)
{
FILE *fp;
int i=0;
char *p;
if((fp=fopen("in.dat","r"))==NULL)
return 1;
while(fgets(xx[i],80,fp)!=NULL)
{p=strchr(xx[i],'\n');
if(p)
*p=0;
i++;
}
maxline=i;
fclose(fp);
return 0;
}
void writedat(void)
{FILE *fp;
int i;
fp=fopen("out3.dat","w");
for(i=0;i<MAXLINE;i++)
{printf("%s\n",xx[i]);
fprintf(fp,"%s\n",xx[i]);
}
fclose(fp);
}
/* 注：注意本题中不能将第一个 if() 的 else 去掉，若不用 else 则经过第一个 if() 后原来
字符'z'变成了'a'，当再经过第二个 if() 时又将已变成'a'的字符改成了'b'，
即实际上字母'z'变成了字母'b'，这是不行的。*/
void convertcharA(void)
```



```

{
char *p;
int i;
for(i=0;i<MAXLINE;i++)
{p=xx[i];
while(*p)
{if(*p=='z') *p='a';
else if(*p>='a'&&*p<'z') *p=*p+1;
p++;
}
}
}
}

```

74、下列程序的功能是:统计在 TT 字符串中'A'到'Z'26 个字母各自现的次数,并存入 PP 数组.请编写函数 CNT(CHAR *TT,INT PP[])实现程序的要求,最后调用函数 READWRITED

AT () 把结果输出到文件 OUT.DAT 中.(仅统计小写字母.)

例如:当输入字符串:abcdefgabcdeabc 后,输出的结果应该是:

3 3 3 2 2 2 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

#include
#include
void readwritdat();
void cnt(char *tt,int pp[])
{
}
main()
{
char tt[1000];
int pp[26], k, n;
clrscr();
printf("\nplease enter a char string:"); scanf("%s",t
t);
cnt(tt,pp);
for(k = 0; k<26; k++) printf ("%d ",pp[k]);
printf("\n");
readwritdat();
}
void readwritdat()
{
char tt[1000];
int pp[26], k, n, i;
FILE *rf, *wf;
rf = fopen("in.dat","r");
wf = fopen("out.dat","w");
for(i = 0; i<10;i++)

```

```

{
fscanf(rf,"%s",tt);
cnt(tt,pp);
for(k = 0;k<26; k++) fprintf(wf,"%d",pp[k]);
fprintf(wf,"\n");
}
fclose(rf);
fclose(wf);
}
/* 注：题中必须先对数组 pp 赋初值。pp[*tt-'a']++的作用是将对应的元素增加 1，即
当*tt 为'a'时应对 pp[0]加 1，而 pp[*tt-'a']即相当于 pp[0]，依此类推。*/
void cnt(char *tt,int pp[])
{
int i;
for(i=0;i<26;i++)
pp[i]=0;
while(*tt)
{if(*tt>='a'&&*tt<='z')
pp[*tt-'a']++;
tt++;
}
}

```

75、函数 READDAT () 实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 XX 中；
请

编制函数 STROR(),其函数功能是:以行为单位把字符串中的所有小写字母 O 左边的字符串内容移到该串的右边存放,然后并把小写字母 O 删除,余下的字符串内容移到已处理字符串的左边存放.最后把已处理的字符串仍按行重新存入字符串数组 XX 中,最后调用函数 WRITEDAT()把结果 XX 输出到文件 OUT5.DAT 中.

例如:原文:You can create an index on any field.

you have the correct record.

结果:n any field. You can create an index

rd.yu have the crrct rec

原始数据文件存放的格式是:每行的宽度均小于 80 个字符,含标点符号和空格.

```

#include"stdio.h"
#include"string.h"
#include"conio.h"
#include"ctype.h"
#include"mem.h"
unsigned char xx[50][80];
int maxline=0;
int readdat(void);
void writedat(void);
void StrOR(void)
{

```

```

}
void main()
{
clrscr();
if(readdat())
{printf("Can't open the file ENG.IN!\n");
return;
}
StrOR();
writedat();
}
int readdat(void)
{
FILE *fp;
int i=0;
char *p;
if((fp=fopen("in.dat","r"))==NULL)
return 1;
while(fgets(xx[i],80,fp)!=NULL)
{p=strchr(xx[i],'\n');
if(p)
*p=0;
i++;
}
maxline=i;
fclose(fp);
return 0;
}
void writedat(void)
{FILE *fp;
int i;
fp=fopen("out5.dat","w");
for(i=0;i<MAXLINE;i++)
{printf("%s\n",xx[i]);
fprintf(fp,"%s\n",xx[i]);
}
fclose(fp);
}

```

/* 注：题目要求的字符串中所有小写字母 o 左边的字符串内容移到该串的右边存放，即将串中“最后”一个字母 o 左右两侧的内容互换。题中第一个 while() 特环的作用是让 p1 指向最后一个字母'o'。第一个 ctrcat() 函数的作用是将 p1 以后的字符都放到新串 t 中，第二个 strcat() 函数的作用是将 p1 以前的字符连接到新串 t 的后面（注意：在些之前要让 p1 所指的单元成为 p1 前面字符串的结束位置 *p1='\0'）。这时完成左右互换。最后一个 while() 循环的作用是删除新串中的所有小写字母'o'，采用的删除方法是不是'

o'的字母一律留下，否则不留（即相当于删除。*/

```
void StrOR(void)
{
    int i;
    char *p1,*p2,t[80];
    for(i=0;i<MAXLINE;i++)
    {t[0]='\0';
    p2=xx[i];
    while(*p2)
    {if(*p2=='o') p1=p2;
    p2++;
    }
    strcat(t,p1+1);
    *p1='\0';
    strcat(t,xx[i]);
    p1=xx[i];
    p2=t;
    while(*p2)
    {if(*p2!='o') *p1++=*p2;
    p2++;
    }
    *p1='\0';
    }
}
```

76、下列程序的功能是：已知学生的记录由学号和学习成绩构成，N 名学生的数据已存入 A 数组中。找出成绩最高的学生记录（假定最高成绩的记录中唯一的），通过形参返回。

请考生编写函数 MMM(STU A[],STU *S)实现程序的要求，最后调用函数 READWRITEDAT

（）把结果输出到文件 OUT.DAT 中。

例如: KS01 87

KS09 97

KS11 67

则调用该函数后,输出 THE TOP:KS09,97

```
#include"stdio.h"
#include"string.h"
#define N 10
void readwritdat();
typedef struct ss{
    char num[10];
    int s;
}STU;
mmm(STU a[],STU *s)
{
}
main()
```

```

{
STU a[N]={{"01",81},{"02",89},{"03",66},{&quo
t;04",87},{"05",77},
{"06",90},{"07",79},{"08",61},{"09&qu
ot;,80},{"10",71}},m;
int i;
for(i=0;i<N;I++)
printf("No=%s Mark=%d\n",a[i].num,a[i].s);
mmm(a,&m);
printf("the highest: %s,%d\n",m.num,m.s);
readwritedat();
}
void readwritedat()
{
FILE *rf,*wf;
STU a[N],m;
int i;
rf=fopen("in.dat","r");
wf=fopen("out.dat","w");
for(i=0;i<10;i++)
fscanf(rf,"%s,%d",a[i].num,&a[i].s);
mmm(a,&m);
fprintf(wf,"the top: %s,%d\n",m.num,m.s);
fclose(rf);
fclose(wf);
}
/* 注：较简单。*/
mmm(STU a[],STU *s)
{
int i;
s->s=a[0].s;
for(i=1;i<N;I++)
if(a[i].s>s->s)
*s=a[i];
}

```

77、函数 READDAT () 实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 XX 中；
请

编制函数 STRCHARJL(),其函数功能是:以行为单位把字符串中的所有字符的 ASCII 值左移 4 位，如果左移后，其字符的 ASCII 值小于等于 32 或大于 100，则原字符保持不变.否则就把左移后的字符 ASCII 值再加上原字符的 ASCII 值，得到新的字符仍存入原字符串对应的位置上.最后把已处理的字符串仍按行重新存入字符串数组 XX 中,最后调用函数 WRITEDAT()把结果 XX 输出到文件 OUT7.DAT 文件中.

原始数据文件存放的格式是:每行的宽度均小于 80 个字符,含标点符号和空格.

#include

```
#include
#include
char xx[50][50];
int maxline=0;
int ReadDat(void);
void WriteDat(void);
void StrCharJR(void)
{
}
void main()
{
clrscr();
if(ReadDat())
{printf("Can't open the data file IN.DAT !\n\007");
return;
}
StrCharJR();
WriteDat();
}
int ReadDat(void)
{
FILE *fp;
int i=0;
char *p;
if((fp=fopen("IN.DAT","r"))==NULL) return 1;
while(fgets(xx[i],80,fp)!=NULL)
{p=strchr(xx[i],'\n');
if(p) *p=0;
i++;
}
maxline=i;
fclose(fp);
return 0;
}
void WriteDat(void)
{
FILE *fp;
int i;
clrscr();
fp=fopen("OUT7.DAT","w");
for(i=0;i<MAXLINE;i++)
{printf("%s\n",xx[i]);
fprintf(fp,"%s\n",xx[i]);
}
```

```

fclose(fp);
}
/* 注：亦是较简单，要看清 if() 中的条件。*/
void StrCharJR(void)
{
    int i,j;
    for(i=0;i<MAXLINE;i++)
        for(j=0;xx[i][j]!='\0';j++)
            if(xx[i][j]<<4>32&&xx[i][j]<<4<=100)
                xx[i][j]+=xx[i][j]<<4;
}

```

78、下列程序的功能是：设 A，B，C 为三个不为零的正整数，计算并输出下列不定方程组解的个数 CNT 以及满足此条件的所有 A，B，C 之和 SUM。

不定方程组为：

$$A+B+C=13$$

$$A-C=5$$

请编写函数 COUNTVALUE（）实现程序要求，最后调用函数 WRITEDAT（）把结果 CNT 和 SUM

输出到文件 OUT.DAT 中。

```

#include
int cnt,sum;
void countvalue()
{
}
void main()
{
    cnt=sum=0;
    countvalue();
    printf("cnt=%d\n",cnt);
    printf("sum=%d\n",sum);
    writedat();
}
writedat()
{
    FILE *fp;
    fp=fopen("out.dat","w");
    fprintf(fp,"%d\n%d\n",cnt,sum);
    fclose(fp);
}
/* 注：由 A-C=5 对 A+B+C=13 进行变换可得 B=18-2*A，题中要求三个数都要大于 0，所以 A
必须从 6 开始（为了保证 C 不小于等于 0），进而可得只要 B 大于 0 即可（即 18-2*A>0）。*/
void countvalue()

```

```

{
int a;
for(a=6;a<13;a++)
if(18-2*a>0)
{cnt++;
sum+=13;
}
}

```

本人只是为了上机能通过，不追求什么算法速度,答案仅供大家参考，我 9 号长春上机。

```

(tonggu)
void countvalue()
{
int i,j,k;
cnt=0;
for(i=1;i<13;i++)
for(j=1;j<13;j++)
for(k=1;k<13;k++)
if(i+j+k==13&& i-k==5)
cnt++;
sum=cnt*13;
}

```

79、函数 READDAT () 实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 XX 中; 请

编制函数 CHA(),其函数功能是:以行为单位把字符串中的第一个字符的 ASCII 值加第二个字符的 ASCII 值,得到第一个新字符,第二个字符的 ASCII 值加第三个字符的 ASCII 值,得到第二个新字符,依此类推一直处理到最后第二个字符,最后一个字符的 ASCII 值加原第一个字符的 ASCII 值,得到最后一个新的字符,得到的新字符分别存放在原字符串对应的位置上。最后把已处理的字符串逆转后按行重新存入字符串数组 XX 中,最后调用函数 WRITEDAT()把结果 XX 输出到文件 OUT9.DAT 中。

原始数据文件存放的格式是:每行的宽度均小于 80 个字符,含标点符号和空格。

```

#include
#include
#include
char xx[50][50];
int maxline=0;
int ReadDat(void);
void WriteDat(void);
void ShA(void)
{
}
void main()
{
clrscr();

```



```
if(ReadDat())
{printf("Can't open the data file IN.DAT !\n\007");
return;
}
ShA();
WriteDat();
}

int ReadDat(void)
{
FILE *fp;
int i=0;
char *p;
if((fp=fopen("IN.DAT","r"))==NULL) return 1;
while(fgets(xx[i],80,fp)!=NULL)
{p=strchr(xx[i],'\n');
if(p) *p=0;
i++;
}
maxline=i;
fclose(fp);
return 0;
}

void WriteDat(void)
{
FILE *fp;
int i;
clrscr();
fp=fopen("OUT9.DAT","w");
for(i=0;i<MAXLINE;i++)
{printf("%s\n",xx[i]);
fprintf(fp,"%s\n",xx[i]);
}
fclose(fp);
}

/* 注：本题中内嵌的 for() 循环只改变第一个字符到倒数的第二个，其中倒数最后一个
字符用最后一个语句来改变（它已不在 for() 循环中）。*/

void ShA(void)
{
int i,j;
char t1;
for(i=0;i<MAXLINE;i++)
{t1=xx[i][0];
for(j=0;xx[i][j+1]!='\0';j++)
xx[i][j]=xx[i][j]+xx[i][j+1];
```

```
xx[i][j+1]=xx[i][j+1]+t1;
}
}
```

80、下列程序的功能是：选取出 100 以上 1000 以内所有个位数字与十位数字之和被 10 除所得余数恰是百位数字的素数（如 293）。计算并输出上述这些素数的个数 CNT 以及这些素数值的和 SUM。请编写函数 COUNTVALUE（）实现程序要求，最后调用函数 WRITEDAT

（）把

结果 CNT 和 SUM 输出到文件 OUT.DAT 中。

```
#include
```

```
int cnt, sum;
```

```
void countvalue()
```

```
{
```

```
}
```

```
void main()
```

```
{
```

```
cnt=sum=0;
```

```
countvalue();
```

```
printf("cnt=%d\n",cnt);
```

```
printf("sum=%d",sum);
```

```
writedat();
```

```
}
```

```
writedat()
```

```
{
```

```
FILE *fp;
```

```
fp=fopen("out.dat", "w");
```

```
fprintf(fp, "%d\n%d\n", cnt, sum);
```

```
fclose(fp);
```

```
}
```

/* 注：内嵌的 for() 循环用来判断 i 以前的数有否能整除 i 的数若能则提前终止循环，第

二个 if() 中 j=i 的作用是配合前面的 for() 判断 i 是否是素数。*/

```
void countvalue()
```

```
{
```

```
int i, j;
```

```
for(i=100; i<1000; i++)
```

```
{for(j=2; j<I; j++)
```

```
if(i%j==0) break;
```

```
if(j==i && ((i%10+i/10%10)%10==i/100))
```

```
{cnt++;
```

```
sum+=i;
```

```
}
```

```
}
```

```
}
```

081 题 已知在文件 IN.DAT 中存有 100 个产品销售记录，每个产品销售记录由产品代码 dm(字符型 4 位)，产品名称 mc(字符型 10 位)，单价 dj(整型)，数量 sl(整型)，金额 je(长整

型)四部分组成。其中：金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat()，其功能要求：按金额从小到大进行排列，若金额相等，则按产品代码从小到大进行排列，最终排列结果仍存入结构数组 sell 中，最后调用函数 WriteDat()把结果输出到文件 OUT1.DAT 中。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include
#include
#include
#include
#include
#define MAX 100
typedef struct{
char dm[5]; /*产品代码*/
char mc[11]; /*产品名称*/
int dj; /*单价*/
int sl; /*数量*/
long je; /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();
void SortDat()
{
}
void main()
{
memset(sell,0,sizeof(sell));
ReadDat();
SortDat();
WriteDat();
}
void ReadDat()
{
FILE *fp;
char str[80],ch[11];
int i;
fp=fopen("IN.DAT","r");
for(i=0;i<100;i++){
fgets(str,80,fp);
memcpy(sell[i].dm,str,4);
memcpy(sell[i].mc,str+4,10);
memcpy(ch,str+14,4);ch[4]=0;
sell[i].dj=atoi(ch);
```

```

memcpy(ch,str+18,5);ch[5]=0;
sell[i].sl=atoi(ch);
sell[i].je=(long)sell[i].dj*sell[i].sl;
}
fclose(fp);
}
void WriteDat()
{
FILE *fp;
int i;
fp=fopen("OUT1.DAT","w");
for(i=0;i<100;i++){
fprintf(fp,"%s %s %4d %5d %101d\n",sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
}
fclose(fp);
}

```

082 题 在文件 in.dat 存放了 10 位选手参赛，6 位评委给选手打分的数据。计算各位选手的得分规则是去掉一个最高分和一个最低分，最后得分是剩下分数的算术平均值。

在下列程序中，函数 readDat()是从文件 df.dat 中读取 10 位选手的评委打分数据并存入数组 mark[10][6]中，要求编写函数 jsValue()计算出每位选手的得分并依次存入数组 score[10]中，最后调用函数 writeDat()把数组 score 中的值输出到文件 out.dat 中。

说明：第 1 位选手的得分存入 score[0]中，第 2 位选手的得分存入 score[1]中，依此类推。

部分源程序已给出。

请勿改动主函数 main()、读函数 readDat()和写函数 writeDat()的内容。 #include

float mark[10][6]; /*存放 10 位选手，6 位评委的打分数据*/

float score[10]; /*存放 10 位选手的得分*/

```

jsValue()
{
}
main()
{
int i;
readDat();
jsValue();
for(i=0;i<10;i++)printf(" 第%d 位选手得分=%7.3f\n",i+1,score[i]);
writeDat()
}
readDat()
{
FILE *fp;
int i,j;
fp=fopen("in.dat","r");
for(i=0;i<10;i++)
for(j=0;j<6;j++)fscanf(fp,"%f",&mark[i][j]);
}

```

```

fclose(fp);
}
writeDat()
{
FILE *fp;
int i;
fp=fopen("out.dat","w");
for(i=0;i<10;i++)fprintf(fp,"%7.3f\n",score[i]);
fclose(fp);
}

```

本人只是为了上机能通过，不追求什么算法速度,答案仅供大家参考，我 9 号长春上机。

```

(tonggu)
jsValue()
{
int i,j;
float max,min,sum;
for(i=0;i<10;i++)
{
max=mark[10][0];
min=mark[10][0];
sum=mark[10][0];
for(j=1;j<6;j++)
{
if(mark[i][j]>max) max=mark[i][j];
if(mark[i][j] sum+=mark[i][j];
}
score[i]=(sum-max-min)/4;
}
}

```

083 题 已知在文件 IN.DAT 中存有 100 个产品销售记录，每个产品销售记录由产品代码 dm(字符型 4 位)，产品名称 mc(字符型 10 位)，单价 dj(整型)，数量 sl(整型)，金额 je(长整型)四部分组成。其中：金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat()，其功能要求：按金额从大到小进行排列，若金额相等，则按产品代码从小到大进行排列，最终排列结果仍存入结构数组 sell 中，最后调用函数 WriteDat()把结果输出到文件 OUT3.DAT 中。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
#include
#include
#define MAX 100

```

```
typedef struct{
char dm[5]; /*产品代码*/
char mc[11]; /*产品名称*/
int dj; /*单价*/
int sl; /*数量*/
long je; /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();
void SortDat()
{
}
void main()
{
memset(sell,0,sizeof(sell));
ReadDat();
SortDat();
WriteDat();
}
void ReadDat()
{
FILE *fp;
char str[80],ch[11];
int i;
fp=fopen("IN.DAT","r");
for(i=0;i<100;i++){
fgets(str,80,fp);
memcpy(sell[i].dm,str,4);
memcpy(sell[i].mc,str+4,10);
memcpy(ch,str+14,4);ch[4]=0;
sell[i].dj=atoi(ch);
memcpy(ch,str+18,5);ch[5]=0;
sell[i].sl=atoi(ch);
sell[i].je=(long)sell[i].dj*sell[i].sl;
}
fclose(fp);
}
void WriteDat()
{
FILE *fp;
int i;
fp=fopen("OUT3.DAT","w");
for(i=0;i<100;i++){
```

```

fprintf(fp, "%s %s %4d %5d %101d\\n", sell[i].dm, sell[i].mc, sell[i].dj, sell[i].sl, sell[i].je);
}
fclose(fp);
}

```

084 题 函数 loop(s,m,n,str)对长为 n 的字符串 str, 从第 s 个字符开始的间隔为 m 的部分字符进行循环左移, 即 $str[s] \leftarrow str[s+m] \leftarrow \dots \leftarrow str[s+km] \leftarrow \dots \leftarrow str[s]$ (k 为正整数)且当字符位置 $s+k*m$ 大于等于 n 时, 要求用 $s+k*m-n$ 作新的字符位置。要求编写此函数实现其功能, 最后调用函数 WriteDat()把结果输出到文件 out.dat 中。

例: loop(1,2,8,str)

位置 0 1 2 3 4 5 6 7

调用前 str=A B C D E F G H

调用后 str=A D C F E H G B

部分源程序已给出。

请勿改动主函数 main()和写函数 WriteDat()的内容。

```

#include
#include
loop(int s,int m,int n,char str[])
{
}
main()
{
char buf[81];
strcpy(buf, "ABCDEFGH");
printf(" 调用前=%s\n", buf);
loop(1,2,8,buf);
printf(" 调用后=%s\n", buf);
writeDat();
}
writeDat()
{
FILE *in,*out;
char buf[81];
int i,s,m,n;
in=fopen("in.dat", "r");
out=fopen("out.dat", "w");
for(i=0;i<5;i++){
fscanf(in, "%d%d%d,%s", &s,&m,&n, buf);
loop(s,m,n,buf);
fprintf(out, "%s\n", buf);
}
fclose(in);
fclose(out);
}

```

```

Viod Loop(int s,int m,int n,char str[])
{ int cuur,last,i=0;
char ch;
last=s;
ch=str[s];
while(++i)
{ cuur=last;
last=(s+i*m)%n;
if(last>=s&&last+i*m>n) break;
str[cuur]=str[last];
}
str[cuur]=ch;
}

```

085 题 已知在文件 IN.DAT 中存有 100 个产品销售记录，每个产品销售记录由产品代码 dm(字符型 4 位)，产品名称 mc(字符型 10 位)，单价 dj(整型)，数量 sl(整型)，金额 je(长整型)四部分组成。其中：金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat()，其功能要求：按产品名称从小到大进行排列，若产品名称相等，则按金额从小到大进行排列，最终排列结果仍存入结构数组 sell 中，最后调用函数 WriteDat()把结果输出到文件 OUT5.DAT 中。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
#include
#include
#define MAX 100
typedef struct{
char dm[5]; /*产品代码*/
char mc[11]; /*产品名称*/
int dj; /*单价*/
int sl; /*数量*/
long je; /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();
void SortDat()
{
}
void main()
{
memset(sell,0,sizeof(sell));
ReadDat();

```



```

SortDat();
WriteDat();
}
void ReadDat()
{
FILE *fp;
char str[80],ch[11];
int i;
fp=fopen("IN.DAT","r");
for(i=0;i<100;i++){
fgets(str,80,fp);
memcpy(sell[i].dm,str,4);
memcpy(sell[i].mc,str+4,10);
memcpy(ch,str+14,4);ch[4]=0;
sell[i].dj=atoi(ch);
memcpy(ch,str+18,5);ch[5]=0;
sell[i].sl=atoi(ch);
sell[i].je=(long)sell[i].dj*sell[i].sl;
}
fclose(fp);
}
void WriteDat()
{
FILE *fp;
int i;
fp=fopen("OUT5.DAT","w");
for(i=0;i<100;i++){
fprintf(fp,"%s %s %4d %5d %101d\n",sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
}
fclose(fp);
}

```

086 题 下列程序的功能是：寻找并输出 11 至 999 之间的数 m ，它满足 m 、 m^2 和 m^3 均为回文数。所谓回文数是指其各位数字左右对称的整数，例如 121，676，94249 等。满足上述条件的数如 $m=11$ ， $m^2=121$ ， $m^3=1331$ 皆为回文数。请编制函数 `int jsValue(long m)` 实现此功能，如果是回文数，则函数返回 1，反之则返回 0。最后把结果输出到文件 `out.dat` 中。部分源程序已给出。

请勿改动主函数 `main()` 的内容。

```

#include
int jsValue(long n)
{
}
main()
{
long m;

```

```

FILE *out;
out=fopen("out.dat","w");
for(m=1;m<1000;m++){
if(jsValue(m)&&jsValue(m*m)&&jsValue(m*m*m)){
printf("m=%4ld,m*m=%6ld,m*m*m=%8ld\n",m,m*m,m*m*m);
}
}
fclose(out);
}

```

```

-----
int jsvalue(long n)
{ int i, strl, half;
char xy[20];
itoa(n, xy, 10);
strl=strlen(xy);
half=strl/2;
for(i=0; i if(xy[i]!=xy[--strl]) break;
if(i>=half) return 1;
else return 0;
}

```

本人只是为了上机能通过，不追求什么算法速度，答案仅供大家参考，我 9 号长春上机。

(tonggu)

```

int jsValue(long n)
{
int i, aa[10], j=0, b=1;
while(n)
{
aa[j++]=n%10;
n=n/10;
}
for(i=0; i
{
if(aa[i]!=aa[j-i-1]) b=0;
}
return b;
}

```

087 题 已知在文件 IN.DAT 中存有 100 个产品销售记录，每个产品销售记录由产品代码 dm(字符型 4 位)，产品名称 mc(字符型 10 位)，单价 dj(整型)，数量 sl(整型)，金额 je(长整型)四部分组成。其中：金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat()，其功能要求：按产品名称从大到小进行排列，若产品名称相等，则按金额从小到大进行排列，最终排列结果仍存入结构数组 sell 中，最后调用函数 WriteDat()把结果输出到文件 OUT7.DAT 中。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include
#include
#include
#include
#include
#define MAX 100
typedef struct{
char dm[5]; /*产品代码*/
char mc[11]; /*产品名称*/
int dj; /*单价*/
int sl; /*数量*/
long je; /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();
void SortDat()
{
}
void main()
{
memset(sell,0,sizeof(sell));
ReadDat();
SortDat();
WriteDat();
}
void ReadDat()
{
FILE *fp;
char str[80],ch[11];
int i;
fp=fopen("IN.DAT","r");
for(i=0;i<100;i++){
fgets(str,80,fp);
memcpy(sell[i].dm,str,4);
memcpy(sell[i].mc,str+4,10);
memcpy(ch,str+14,4);ch[4]=0;
sell[i].dj=atoi(ch);
memcpy(ch,str+18,5);ch[5]=0;
sell[i].sl=atoi(ch);
sell[i].je=(long)sell[i].dj*sell[i].sl;
}
fclose(fp);
```

```

}
void WriteDat()
{
FILE *fp;
int i;
fp=fopen("OUT7.DAT","w");
for(i=0;i<100;i++){
fprintf(fp,"%s %s %4d %5d %101d\n",sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
}
fclose(fp);
}

```

088 题 编写函数 jsValue，它的功能是：求 Fibonacci 数列中大于 t 的最小的一个数，结果由函数返回。其中 Fibonacci 数列 F(n)的定义为：

$F(0)=0, F(1)=1$

$F(n)=F(n-1)+F(n-2)$

最后调用函数 writeDat()读取 10 个数据 t，分别得出结果且把结果输出到文件 out.dat 中。

例如：当 $t=1000$ 时，函数值为：1597。

部分源程序已给出。

请勿改动主函数 main()和写函数 writeDat()的内容。 #include

```

int jsValue(int t)
{
}

main()
{
int n;
n=1000;
printf("n=%d,f=%d\n",n,jsValue(n));
writeDat();
}

writeDat()
{
FILE *in,*out;
int i,n,s;
in=fopen("in.dat","r");
out=fopen("out.dat","w");
for(i=0;i<10;i++){
fscanf(in,"%d",&n);
s=jsValue(n);
printf("out\\%d\n",s);
}
fclose(in);
fclose(out);
}

```

```

int jsvalue(int t)
{ int f1=0,f2=1,fn;
fn=f1+f2;
while(fn<=t)
{f1=f2;
f2=fn;
fn=f1+f2;
}
return fn;
}

```

089 题 已知在文件 IN.DAT 中存有 100 个产品销售记录，每个产品销售记录由产品代码 dm(字符型 4 位)，产品名称 mc(字符型 10 位)，单价 dj(整型)，数量 sl(整型)，金额 je(长整型)四部分组成。其中：金额=单价*数量计算得出。函数 ReadDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编制函数 SortDat()，其功能要求：按产品代码从小到大进行排列，若产品代码相等，则按金额从小到大进行排列，最终排列结果仍存入结构数组 sell 中，最后调用函数 WriteDat()把结果输出到文件 OUT9.DAT 中。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
#include
#include
#define MAX 100
typedef struct{
char dm[5]; /*产品代码*/
char mc[11]; /*产品名称*/
int dj; /*单价*/
int sl; /*数量*/
long je; /*金额*/
}PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();
void SortDat()
{
}
void main()
{
memset(sell,0,sizeof(sell));
ReadDat();
SortDat();
WriteDat();
}

```

```

void ReadDat()
{
    FILE *fp;
    char str[80],ch[11];
    int i;
    fp=fopen("IN.DAT",\r\);
    for(i=0;i<100;i++){
        fgets(str,80,fp);
        memcpy(sell[i].dm,str,4);
        memcpy(sell[i].mc,str+4,10);
        memcpy(ch,str+14,4);ch[4]=0;
        sell[i].dj=atoi(ch);
        memcpy(ch,str+18,5);ch[5]=0;
        sell[i].sl=atoi(ch);
        sell[i].je=(long)sell[i].dj*sell[i].sl;
    }
    fclose(fp);
}

void WriteDat()
{
    FILE *fp;
    int i;
    fp=fopen("OUT9.DAT",\w\);
    for(i=0;i<100;i++){
        fprintf(fp,\ %s %s %4d %5d %101d\n\ , sell[i].dm,sell[i].mc,sell[i].dj,sell[i].sl,sell[i].je);
    }
    fclose(fp);
}

```

090 题 函数 ReadDat()实现从文件 in.dat 中读取 20 行数据存放到字符串数组 xx 中(第行字符串长度均小于 80)。请编制函数 jsSort(), 其函数的功能是: 以行为单位对字符串变量的下标为奇数的字符按其 ASCII 值从大到小的顺序进行排序, 排序后的结果仍按行重新存入字符串数组 xx 中, 最后调用函数 WriteDat()把结果 xx 输出到文件 out.dat 中。

例如: 位置 0 1 2 3 4 5 6 7
源字符串 a b c d e f g h

则处理后字符串 a h c f e d g b

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
char xx[20][80];
void jsSort()
{
}

```

```
void main()
{
readDat();
jsSort();
writeDat();
}
readDat()
{
FILE *in;
int i=0;
char *p;
in=fopen(" in.dat", "r");
while(i<20&&fgets(xx[i],80,in)!=NULL){
p=strchr(xx[i], '\n');
if(p)*p=0;
i++;
}
fclose(in);
}
writeDat()
{
FILE *out;
int i;
out=fopen(" out.dat", "w");
clrscr();
for(i=0;i<20;i++){
printf(" %s\n",xx[i]);
fprintf(out, " %s\n",xx[i]);
}
fclose(out);
}
void jsSort()
{
int i,j,k,strlen;
char ch;
for(i=0;i<20;i++)
{
strlen=strlen(xx[i]);
for(j=1;j
for(k=j+2;k
if(xx[i][j]<XX[I][K])
{
ch=xx[i][j];
xx[i][j]=xx[i][k];
```

```

xx[i][k]=ch;
}
}
}

```

091 题 函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS1.DAT 中。

替代关系： $f(p)=p*11 \bmod 256$ (p 是数组中某一个字符的 ASCII 值，f(p)是计算后新字符的 ASCII 值)，如果计算后 f(p)值小于等于 32 或大于 130，则该字符不变，否则将 f(p)所对应的字符进行替代。

部分源程序已给出。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
#include
unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/
int ReadDat(void)
void WriteDat(void)
void encryptChar()
{
}
void main()
{
clrscr();
if(ReadDat()){
printf(\ 数据文件 ENG.IN 不能打开! \n\007\);
return;
}
encryptChar();
WriteDat();
}
int ReadDat(void)
{
FILE *fp;
int i=0;
unsigned char *p;
if((fp=fopen(\ eng.in\ ,\ r\ ))==NULL) return 1;
while(fgets(xx[i],80,fp)!=NULL){
p=strchr(xx[i],'\n');
if(p)*p=0;
i++;
}
}

```



```

maxline=i;
fclose(fp);
return 0;
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen("ps1.dat","w");
for(i=0;i<MAXLINE;i++){
printf("%s\n",xx[i]);
fprintf(fp,"%s\n",xx[i]);
}
fclose(fp);
}

```

本人只是为了上机能通过，不追求什么算法速度,答案仅供大家参考，我 9 号长春上机。
(tonggu)

```

void encryptChar()
{
int i,j,t;
for(i=0;i<MAXLINE;i++)
{
for(j=0;j<STRLEN(XX[i]);j++)
{
t=xx[i][j]*11%256;
if(t<=32 || t>130) continue;
xx[i][j]=t;
}
}
}

```

092 题 在文件 in.dat 中有 200 组数据，每组有 3 个数，每个数均是三位数。函数 ReadDat() 读取这 200 组数据存放到结构数组 aa 中，请编制函数 jsSort()，其函数的功能是：要求在 200 组数据中找出条件为每组中的第二个数大于第一个数加第三个数的之和，其中满足条件的个数作为函数 jsSort() 的返回值，同时把满足条件的数据存入结构数组 bb 中，再对 bb 中的数据按照每组数据的第二个数加第三个之和的大小进行降序排列(第二个数加第三个数的和均不相等)，排序后的结果仍重新存入结构数组 bb 中，最后调用函数 WriteDat()把结果 bb 输出到文件 out.dat 中。

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
typedef struct{

```

```

int x1,x2,x3;
}data;
data aa[200],bb[200];
int jsSort()
{
}
void main()
{
int count;
readDat();
count=jsSort();/*返回满足条件的个数*/
writeDat(count);
}
readDat()
{
FILE *in;
int i;
in=fopen("in.dat","r");
for(i=0; i<200; i++)
fscanf(in,"%d%d%d",&aa[i].x1,&aa[i].x2,&aa[i].x3);
fclose(in);
}
writeDat(int count)
{
FILE *out;
int i;
out=fopen("out.dat","w");
clrscr();
for(i=0; i<10; i++){
printf("%d,%d,%d 第二个数+第三个数=%d\n",bb[i].x1,bb[i].x2,bb[i].x3,bb[i].x2+bb[i].x3);
fprintf(out,"%d%d%d\n",bb[i].x1,bb[i].x2,bb[i].x3);
}
fclose(out);
}

```

093 题 函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS3.DAT 中。

替代关系： $f(p)=p*17\text{mod } 256$ (p 是数组中某一个字符的 ASCII 值， $f(p)$ 是计算后新字符的 ASCII 值)，如果计算后 $f(p)$ 值小于等于 32 或其 ASCII 值是奇数，则该字符不变，否则将 $f(p)$ 所对应的字符进行替代。

部分源程序已给出。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include

```

```
#include
#include
unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/
int ReadDat(void)
void WriteDat(void)
void encryptChar()
{
}
void main()
{
clrscr();
if(ReadDat()){
printf(\ 数据文件 ENG.IN 不能打开! \n\007\);
return;
}
encryptChar();
WriteDat();
}
int ReadDat(void)
{
FILE *fp;
int i=0;
unsigned char *p;
if((fp=fopen(\ eng.in\ ,\ r\ ))==NULL) return 1;
while(fgets(xx[i],80,fp)!=NULL){
p=strchr(xx[i],'\n');
if(p)*p=0;
i++;
}
maxline=i;
fclose(fp);
return 0;
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen(\ ps3.dat\ ,\ w\ );
for(i=0;i<MAXLINE;i++){
printf(\ %s\n\ ,xx[i]);
fprintf(fp,\ %s\n\ ,xx[i]);
}
fclose(fp);
```

```

}
void encryptChar()
{
    int i,j,t;
    for(i=0;i<MAXLINE;I++)
    {
        for(j=0;j<STRLEN(XX[I]);J++)
        {
            t=xx[i][j]*17%256;
            if(t<=32 || t%2==1) continue;
            xx[i][j]=t;
        }
    }
}

```

094 题 在文件中有 200 个正整数，且每个数均在 1000 至 9999 之间。函数 ReadDat()读取这 200 个数存放到数组 aa 中。请编制函数 jsSort()，其函数的功能是：要求按每个数的后三位的大小进行降序排列，然后取出满足此条件的前 10 个数依次存入数组 b 中，如果后三位的数值相等，则按原先的数值进行升序排列。最后调用函数 WriteDat()把结果 bb 输出到文件 out.dat 中。

例：处理前 9012 5099 6012 7025 8088

处理后 5099 8088 7025 6012 9012

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
int aa[200],bb[10];
void jsSort()
{
}
void main()
{
    readDat();
    jsSort();
    writeDat();
}
readDat()
{
    FILE *in;
    int i;
    in=fopen("in.dat","r");
    for(i=0; i<200; i++) fscanf(in,"%d",&aa[i]);
    fclose(in);
}

```

```

writeDat()
{
FILE *out;
int i;
out=fopen("out.dat","w");
clrscr();
for(i=0; i<10; i++){
printf("i=%d,%d\n",i+1,bb[i]);
fprintf(out,"%d\n",bb[i]);
}
fclose(out);
}

```

095 题 函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS5.DAT 中。

替代关系： $f(p)=p*11\text{mod } 256$ (p 是数组中某一个字符的 ASCII 值， $f(p)$ 是计算后新字符的 ASCII 值)，如果计算后 $f(p)$ 值小于等于 32 或 $f(p)$ 对应的字符是小写字母，则该字符不变，否则将 $f(p)$ 所对应的字符进行替代。

部分源程序已给出。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
#include
unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/
int ReadDat(void)
void WriteDat(void)
void encryptChar()
{
}
void main()
{
clrscr();
if(ReadDat()){
printf(" 数据文件 ENG.IN 不能打开! \n\n");
return;
}
encryptChar();
WriteDat();
}
int ReadDat(void)
{
FILE *fp;

```

```

int i=0;
unsigned char *p;
if((fp=fopen( eng.in\ ,\r\ ))==NULL) return 1;
while(fgets(xx[i],80,fp)!=NULL){
p=strchr(xx[i],'\n');
if(p)*p=0;
i++;
}
maxline=i;
fclose(fp);
return 0;
}

void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen(\\ps5.dat\\ ,\\w\\ );
for(i=0;i<MAXLINE;I++){
printf(\\ %s\\n\\ ,xx[i]);
fprintf(fp,\\ %s\\n\\ ,xx[i]);
}
fclose(fp);
}

void encryptChar()
{
{
int i,j,t;
for(i=0;i<MAXLINE;I++)
{
for(j=0;j<STRLEN(XX[I]);J++)
{
t=xx[i][j]*11%256;
if(t<=32 || t>='a' && t<='z') continue;
xx[i][j]=t;
}
}
}
}

```

096 题 函数 ReadDat()实现从文件 in.dat 中读取 20 行数据存放到字符串数组 xx 中(第行字符串长度均小于 80)。请编制函数 jsSort(), 其函数的功能是: 以行为单位对字符串按给定的条件进行排序, 排序后的结果仍按行重新存入字符串数组 xx 中, 最后调用函数 WriteDat() 把结果 xx 输出到文件 out.dat 中。

条件: 从字符串中间一分为二, 左边部分按字符的 ASCII 值升序排序, 右边部分按字符的 ASCII 值降序排序。如果原字符串长度为奇数, 则最中间的字符不参加排序, 字符仍放在原位置上。

例如: 位置 0 1 2 3 4 5 6 7 8

源字符串 h g f e a b c d
8 7 6 5 9 1 2 3 4
则处理后字符串 e f g h d c b a
5 6 7 8 9 4 3 2 1

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include
#include
#include
char xx[20][80];
void jsSort()
{
}
void main()
{
    readDat();
    jsSort();
    writeDat();
}
readDat()
{
    FILE *in;
    int i=0;
    char *p;
    in=fopen("in.dat","r");
    while(i<20&&fgets(xx[i],80,in)!=NULL){
        p=strchr(xx[i],'\n');
        if(p)*p=0;
        i++;
    }
    fclose(in);
}
writeDat()
{
    FILE *out;
    int i;
    clrscr();
    out=fopen("out.dat","w");
    for(i=0;i<20;i++){
        printf("%s\n",xx[i]);
        fprintf(out,"%s\n",xx[i]);
    }
    fclose(out);
}
```

```

void jsSort()
{
int i,j,k,strl;
char ch;
for(i=0;i<20;i++)
{
strl=strlen(xx[i]);
for(j=0;j
for(k=j+1;k
if(xx[i][j]>xx[i][k])
{
ch=xx[i][j];
xx[i][j]=xx[i][k];
xx[i][k]=ch;
}
for(j=(strl+1)/2;j<STRL;J++)
for(k=j+1;k<STRL;K++)
if(xx[i][j]<XX[I][K])
{
ch=xx[i][j];
xx[i][j]=xx[i][k];
xx[i][k]=ch;
}
}
}
}

```

097 题 函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS7.DAT 中。

替代关系： $f(p)=p*11\text{mod } 256$ (p 是数组中某一个字符的 ASCII 值， $f(p)$ 是计算后新字符的 ASCII 值)，如果原字符是大写字母或计算后 $f(p)$ 值小于等于 32，则该字符不变，否则将 $f(p)$ 所对应的字符进行替代。

部分源程序已给出。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
#include
unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/
int ReadDat(void)
void WriteDat(void)
void encryptChar()
{
}

```



```
void main()
{
clrscr();
if(ReadDat()){
printf(\ 数据文件 ENG.IN 不能打开! \n\007\);
return;
}
encryptChar();
WriteDat();
}
int ReadDat(void)
{
FILE *fp;
int i=0;
unsigned char *p;
if((fp=fopen(\ eng.in\ ,\ r\ ))==NULL) return 1;
while(fgets(xx[i],80,fp)!=NULL){
p=strchr(xx[i],'\n');
if(p)*p=0;
i++;
}
maxline=i;
fclose(fp);
return 0;
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen(\ ps7.dat\ ,\ w\ );
for(i=0;i<MAXLINE;i++){
printf(\ %s\n\ ,xx[i]);
fprintf(fp,\ %s\n\ ,xx[i]);
}
fclose(fp);
}
void encryptChar()
{
int i,j,t;
for(i=0;i<MAXLINE;i++)
{
for(j=0;j<STRLEN(XX[I]);j++)
{
t=xx[i][j]*11%256;
```

```

if(t<=32 || xx[i][j]>='A' && xx[i][j]<='Z') continue;
xx[i][j]=t;
}
}
}

```

098 题 函数 ReadDat()实现从文件 in.dat 中读取 20 行数据存放到字符串数组 xx 中(第行字符串长度均小于 80)。请编制函数 jsSort(), 其函数的功能是: 以行为单位对字符串按给定的条件进行排序, 排序后的结果仍按行重新存入字符串数组 xx 中, 最后调用函数 WriteDat() 把结果 xx 输出到文件 out.dat 中。

条件: 从字符串中间一分为二, 右边部分按字符的 ASCII 值降序排序, 排序后左边部分与右边部分进行交换。如果原字符串长度为奇数, 则最中间的字符不参加排序, 字符仍放在原位置上。

例如: 位置 0 1 2 3 4 5 6 7 8

源字符串 h g f e a b c d

8 7 6 5 9 1 2 3 4

则处理后字符串 d c b a h g f e

4 3 2 1 9 8 7 6 5

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
char xx[20][80];
void jsSort()
{
}
void main()
{
    readDat();
    jsSort();
    writeDat();
}
readDat()
{
    FILE *in;
    int i=0;
    char *p;
    in=fopen("in.dat","r");
    while(i<20&&fgets(xx[i],80,in)!=NULL){
        p=strchr(xx[i],'\n');
        if(p)*p=0;
        i++;
    }
    fclose(in);
}

```

```

}
writeDat()
{
FILE *out;
int i;
clrscr();
out=fopen("out.dat","w");
for(i=0;i<20;i++){
printf("%s\n",xx[i]);
fprintf(out,"%s\n",xx[i]);
}
fclose(out);
}
void jsSort()
{
int i,j,k,strl;
char ch;
for(i=0;i<20;i++)
{
strl=strlen(xx[i]);
for(j=(strl+1)/2;j<STRL;J++)
for(k=j+1;k<STRL;K++)
if(xx[i][j]<XX[I][K])
{
ch=xx[i][j];
xx[i][j]=xx[i][k];
xx[i][k]=ch;
}
for(j=0;j
{
ch=xx[i][j];
xx[i][j]=xx[i][(strl+1)/2+j];
xx[i][(strl+1)/2+j]=ch;
}
}
}
}

```

099 题 函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章，存入到字符串数组 xx 中；请编制函数 encryptChar()，按给定的替代关系对数组 xx 中的所有字符进行替代，仍存入数组 xx 的对应的位置上，最后调用函数 WriteDat()把结果 xx 输出到文件 PS9.DAT 中。

替代关系： $f(p)=p*11\text{mod } 256$ (p 是数组中某一个字符的 ASCII 值， $f(p)$ 是计算后新字符的 ASCII 值)，如果原字符是数字字符 0 至 9 或计算后 $f(p)$ 值小于等于 32，则该字符不变，否则将 $f(p)$ 所对应的字符进行替代。

部分源程序已给出。原始数据文件存放的格式是：每行的宽度均小于 80 个字符。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include
#include
#include
#include
unsigned char xx[50][80];
int maxline=0;/*文章的总行数*/
int ReadDat(void)
void WriteDat(void)
void encryptChar()
{
}
void main()
{
clrscr();
if(ReadDat()){
printf(\ 数据文件 ENG.IN 不能打开! \n\007\ );
return;
}
encryptChar();
WriteDat();
}
int ReadDat(void)
{
FILE *fp;
int i=0;
unsigned char *p;
if((fp=fopen(\ eng.in\ ,\ r\ ))==NULL) return 1;
while(fgets(xx[i],80,fp)!=NULL){
p=strchr(xx[i],'\n');
if(p)*p=0;
i++;
}
maxline=i;
fclose(fp);
return 0;
}
void WriteDat(void)
{
FILE *fp;
int i;
fp=fopen(\ ps9.dat\ ,\ w\ );
for(i=0;i<MAXLINE;i++){
printf(\ %s\n\ ,xx[i]);
fprintf(fp,\ %s\n\ ,xx[i]);
```

```

}
fclose(fp);
}
void encryptChar()
{
    int i,j,t;
    for(i=0;i<MAXLINE;I++)
    {
        for(j=0;j<STRLEN(XX[I]);J++)
        {
            t=xx[i][j]*11%256;
            if(t<=32 || xx[i][j]>='0' && xx[i][j]<='9') continue;
            xx[i][j]=t;
        }
    }
}

```

100 题 函数 ReadDat()实现从文件 in.dat 中读取 20 行数据存放到字符串数组 xx 中(第行字符串长度均小于 80)。请编制函数 jsSort(), 其函数的功能是: 以行为单位对字符串按给定的条件进行排序, 排序后的结果仍按行重新存入字符串数组 xx 中, 最后调用函数 WriteDat()把结果 xx 输出到文件 out.dat 中。

条件: 从字符串中间一分为二, 右边部分按字符的 ASCII 值升序排序, 排序后左边部分与右边部分进行交换。如果原字符串长度为奇数, 则最中间的字符不参加排序, 字符仍放在原位置上。

例如: 位置 0 1 2 3 4 5 6 7 8
源字符串 d c b a h g f e

8 7 6 5 9 1 2 3 4

则处理后字符串 e f g h d c b a

5 6 7 8 9 4 3 2 1

部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

#include
#include
#include
char xx[20][80];
void jsSort()
{
}
void main()
{
    readDat();
    jsSort();
    writeDat();
}
readDat()

```

```
{
FILE *in;
int i=0;
char *p;
in=fopen(" in.dat", "r");
while(i<20&&fgets(xx[i],80,in)!=NULL){
p=strchr(xx[i], '\n');
if(p)*p=0;
i++;
}
fclose(in);
}

writeDat()
{
FILE *out;
int i;
clrscr();
out=fopen(" out.dat", "w");
for(i=0;i<20;i++){
printf(" %s\n", xx[i]);
fprintf(out, " %s\n", xx[i]);
}
fclose(out);
}

void jsSort()
{
int i,j,k,strl;
char ch;
for(i=0;i<20;i++)
{
strl=strlen(xx[i]);
for(j=(strl+1)/2;j<STRL;J++)
for(k=j+1;k<STRL;K++)
if(xx[i][j]>xx[i][k])
{
ch=xx[i][j];
xx[i][j]=xx[i][k];
xx[i][k]=ch;
}
for(j=0;j
{
ch=xx[i][j];
xx[i][j]=xx[i][(strl+1)/2+j];
xx[i][(strl+1)/2+j]=ch;
```

```
}  
}  
}
```

二级 c 语言试题及答案

1)下面叙述正确的是_____。

- A)算法的执行效率与数据的存储结构无关
- B)算法的空间复杂度是指算法程序中指令(或语句)的条数
- C)算法的有穷性是指算法必须能在执行有限个步骤之后终止
- D)算法的时间复杂度是指执行算法程序所需要的时间

(1)C

知识点: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)

评 析: 算法的设计可以避开具体的计算机程序设计语言, 但算法的实现必须借助程序设计语言中提供的数据类型及其算法。数据结构和算法是计算机科学的两个重要支柱。它们是一个不可分割的整体。算法在运行过程中需辅助存储空间的大小称为算法的空间复杂度。算法的有穷性是指一个算法必须在执行有限的步骤以后结束。算法的时间复杂度是指执行算法所需要的计算工作量, 即算法执行过程中所需要的基本运算次数。

(2)以下数据结构属于非线性数据结构的是_____。

- A)队列 B)线性表 C)二叉树 D)栈

(2)C

知识点:栈和队列的定义; 栈和队列的顺序存储结构及其基本运算

评 析:线性表、栈和队列等数据结构所表达和处理的数据以线性结构为组织形式。栈是一种特殊的线性表, 这种线性表只能在固定的一端进行插入和删除操作, 允许插入和删除的一端称为栈顶, 另一端称为栈底。一个新元素只能从栈顶一端进入, 删除时, 只能删除栈顶的元素, 即刚刚被插入的元素。所以栈又称后进先出表(Last In First Out)。队列可看作是插入在一端进行, 删除在另一端进行的线性表, 允许插入的一端称为队尾, 允许删除的一端称为队头。在队列中, 只能删除队头元素, 队列的最后一个元素一定是最新入队的元素。因此队列又称先进先出表(First In First Out)。二叉树的数据结构是树型结构, 结构中数据元素之间存在着一对多的关系, 因此它是一种非线性数据结构。

(3)在一棵二叉树上第 8 层的结点数最多是_____。

- A)8 B)16 C)128 D)256

(3)C

知识点: 二叉树的定义及其存储结构

评 析:根据二叉树的性质: 二叉树第 $i(i>1)$ 层上至多有 2^{i-1} 个结点。得到第 8 层的结点数最多是 128。

(4)下面描述中, 不符合结构化程序设计风格的是_____。

- A)使用顺序、选择和重复(循环)三种基本控制结构表示程序的控制逻辑
- B)自顶向下
- C)注重提高程序的执行效率
- D)限制使用 goto 语句

(4)C

知识点: 结构化程序设计

评 析: 结构化程序设计方法的四条原则是: 1. 自顶向下; 2. 逐步求精; 3. 模块化; 4. 限制使用 goto 语句。“自顶向下”是说, 程序设计时, 应先考虑总体, 后考虑细节, 先考虑全局目标, 后考虑局部目标; “逐步求精”是说, 对复杂问题, 应设计一些子目标作过渡,

逐步细节化;“模块化”是说,一个复杂问题肯定是由若干稍简单的问题构成,解决这个复杂问题的程序,也应对应若干稍简单的问题,分解成若干稍小的部分。

(5)下面概念中,不属于面向对象方法的是_____。

A)对象、消息 B)继承、多态 C)类、封装 D)过程调用

(5)D

知识点:面向对象的程序设计方法、对象、方法、属性及继承与多态性

评 析:面向对象方法是一种运用对象、类、封装、继承、多态和消息等概念来构造、测试、重构软件的方法。面向对象方法从对象出发,发展出对象、类、消息、继承等概念。

(6)在结构化方法中,用数据流程图(DFD)作为描述工具的软件开发阶段是_____。

A)可行性分析 B)需求分析 C)详细设计 D)程序编码

(6)B

知识点:结构化设计方法

评 析:软件开发阶段包括需求分析、总体设计、详细设计、编码和测试五个阶段。其中需求分析阶段常用的工具是数据流程图和数据字典。

(7)软件生命周期中所花费用最多的阶段是_____。

A)详细设计 B)软件编码 C)软件测试 D)软件维护

(7)D

知识点: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

评 析:软件生命周期分为软件定义、软件开发及软件运行维护 3 个阶段。本题中详细设计、软件编码和软件测试都属于软件开发阶段;维护是软件生命周期的最后一个阶段,也是持续时间最长,花费代价最大的一个阶段,软件工程学的一个目的就是提高软件的可维护性,降低维护的代价。

(8)数据库系统的核心是_____。

A)数据模型 B)DBMS C)软件工具 D)数据库

(8)B

知识点:数据库的基本概念:数据库,数据库管理系统,数据库系统

评 析:数据库管理系统 DBMS 是数据库系统的核心。DBMS 是负责数据库的建立、使用和维护的软件。DBMS 建立在操作系统之上,实施对数据库的统一管理和控制。用户使用的各种数据库命令以及应用程序的执行,最终都必须通过 DBMS。另外, DBMS 还承担着数据库的安全保护工作,按照 DBA 所规定的要求,保证数据库的完整性和安全性。

(9)下列叙述中正确的是_____。

A)数据处理是将信息转化为数据的过程

B)数据库设计是指设计数据库管理系统

C)如果一个关系中的属性或属性组并非该关系的关键字,但它是另一个关系的关键字,则称其为本关系的外关键字

D)关系中的每列称为元组,一个元组就是一个字段

(9)C

知识点: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

评 析:数据处理是指将数据转换成信息的过程,故选项 A 叙述错误;设计数据库的目的实质上是设计出满足实际应用需求的实际关系模型,故选项 B 叙述错误;关系中的行称为元组,对应存储文件中的记录,关系中的列称为属性。对应存储文件中的字段,故 D 选项叙述错误。

(10)下列模式中,_____是用户模式。

A)内模式 B)外模式 C)概念模式 D)逻辑模式

(10)B

知识点：数据库的基本概念：数据库，数据库管理系统，数据库系统

评 析：数据库管理系统的三级模式结构由外模式、模式和内模式组成。外模式，或称子模式，或称用户模式，是指数据库用户所看到的数据结构，是用户看到的数据视图。模式，或称逻辑模式，是数据库中对全体数据的逻辑结构和特性的描述，是所有用户所见到的数据视图的总和。外模式是模式的一部分。内模式，或称存储模式，或称物理模式，是指数据在数据库系统内的存储介质上的表示。即对数据的物理结构和存取方式的描述。

(11)C 语言规定，程序中各函数之间_____。

- A)既允许直接递归调用也允许间接递归调用
- B)不允许直接递归调用也不允许间接递归调用
- C)允许直接递归调用不允许间接递归调用
- D)不允许直接递归调用允许间接递归调用

(11)A

知识点：函数的递归调用

评 析：c 语言规定，程序中各函数之间既允许直接递归调用也允许间接递归调用。

(12)C 语言中下列叙述正确的是_____。

- A)不能使用 do-while 语句构成的循环
- B)do-while 语句构成的循环，必须用 break 语句才能退出
- C)do-while 语句构成的循环，当 while 语句中的表达式值为非零时结束循环
- D)do-while 语句构成的循环，当 while 语句中的表达式值为零时结束循环

(12)D

知识点：do-while 语句

评 析：选项 A 是错误的，c 语言支持 do-while 语句；选项 B 是错误的，do-while 构成的循环，当 while 语句中的表达式值为零时结束循环，而不是非零；选项 C 也是错误的。

(13)以下选项中属于 C 语言的数据类型是_____。

- A)复数型 B)逻辑型 C)双精度型 D)集合型

(13)C

知识点：c 语言的数据类型

评 析：c 语言的数据类型分为基本类型、构造类型、指针类型、空类型四大类。其中，基本类型分为整型、字符型、实型三类。实型又称浮点型，包括单精度型和双精度型两种类型。

(14)在 C 语言中，不正确的 int 类型的常数是_____。

- A)32768 B)0 C)037 D)0xAF

(14)A

知识点：int 类型的范围

评 析：c 语言中 int 类型的常数的范围是：-32768~32767。c 整常数可用三种形式表示：十进制整数，八进制整数，十六进制整数。选项 A 超出范围，不正确。

(15)下列描述中不正确的是_____。

- A)字符型数组中可以存放字符串
- B)可以对字符型数组进行整体输入、输出
- C)可以对整型数组进行整体输入、输出
- D)不能在赋值语句中通过赋值运算符“=”对字符型数组进行整体赋值

(15)C

知识点：对数组的理解

评 析：c 语言规定只能逐个引用数组元素而不能一次引用整个数组。字符数组的输入、输

出可以将整个字符串一次输入或输出。所以，选项 C 的说法是不正确的。

(16)以下程序的输出结果是_____。

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

A)5 2 B)5 1 C)5 3 D)9 7

(16)C

知识点:通过指针引用数组元素

评 析: 题中 `*p=&x[1][1];` 是指将数组 x 的数组元素 `x[1][1]` 的地址赋值给指针变量 p, 使 p 指向 `x[1][1]` 这个数组元素, 那么 `p[0]` 即为指针变量 p 当前所指向的数组元素 `x[1][1]` 的值。具体执行时如下所示:

i=0 时, 输出 `p[0]`, 也就是 `x[1][1]`, 输出是 5:

i=2 时, 输出 `p[2]`, 即 `p[2+0]`, 也就是 `x[2][0]`, 输出是 3。

(17)以下程序的运行结果是_____。

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

A)运行后报错 B)6 6

C)6 12 D)5 5

(17)A

知识点: 通过指针引用数组

评 析: 题目中没有给 q 分配存储单元, 只是简单地给它赋了一个值, 所以程序的运行结果是 6 11 NULL pointer assignment, 也就是运行后报错。

(18)以下说法中正确的是_____。

A)c 语言程序总是从第一个函数开始执行

B)在 C 语言程序中, 要调用函数必须在 `main()` 函数中定义

C)c 语言程序总是从 `main()` 函数开始执行

D)c 语言程序中的 `main()` 函数必须放在程序的开始部分

(18)C

知识点: C 程序的运行顺序

评 析: c 语言的程序是由主函数 `main()` 开始运行, 由主函数来调用其他函数, 所以选项 A 错误; c 语言中定义的函数必须是并列的, 不能在一个函数中定义其他函数, 选项 B 错误; 函数必须先定义后使用, 在调用函数之前要定义函数, 而 `main()` 函数不一定要放在程序的开始部分, 故选项 D 错误。

(19)能正确表示 a 和 b 同时为正或同时为负的逻辑表达式是_____。

A)(a>=0 || b>=0) && (a<0 || b<0) B)(a>=0 && b>=0) && (a<0 && b<0)

C)(a+b>0) && (a+b<=0) D)a*b>0

(19)D

知识点：对逻辑表达式的判断

评析：逻辑表达式是指用逻辑运算符将关系表达式或逻辑量连接起来。

选项 A 中，表达式表示的是 a, b 为异号；

选项 B 中，表达式表示的是 0，因为没有满足条件的值；

选项 C 中，表达式表示的是 0，因为没有满足条件的值；

选项 D 表示的是 a 和 b 为同号。

(20)若已定义：int a[9]，*p=a；并在以后的语句中未改变 p 的值，不能表示 a[l]地址的表达式是_____。

A)p+l B)a+l C)a++ D)++p

(20)C

知识点：数组地址的表示

评析：数组名是指针常量，是个固定的指针，不能对数组名赋予一个新的地址值，也不能使这个指针常量“移动”，指向数组的其他元素。

(21)以下程序的输出结果是_____。

```
main()
{ int a=-1, b=1, k;
  if((++a<0)&&!(b--<=0))
    printf(" %d %d\n",a,b);
  else
    printf(" %d %d\n", b,a);
}
```

A)-1 1 B)0 1

C)1 0 D)0 0

(21)C

知识点：if 条件的判断

评析：逻辑运算符的优先次序如下：!(非)→&&(与)→||(或)，但本题需特别注意的是短路的概念：对于&&运算符，其两边表达式的值只要有一边为假，则整个与运算表达式的值即为假，系统在运行时，先运算&&左边的表达式，若为假，则系统不会再判断&&运算符右边的表达式了，直接运用短路原理得整个与运算表达式的值为 0。由于++a 是先运算后使用，b--是先使用后运算。所以本题在执行++a 后，a 值为 0，不满足条件，所以 a 为 1，根据短路原理，b 仍为 0，接下去执行 else 语句，输出 1 0。

(22)以下选项中，能定义 s 为合法的结构体变量的是_____。

A) typedef struct abc

```
{ double a;
char b[10];
}s;
```

B) struct

```
{ double a;
char b[10];
}s;
```

C) struct ABC

```
{ double a;
char b[10];
}
```

```

ABC s;
D ) typedef ABC
{ double a;
char b[10];
}
ABC s:

```

(22)B

知识点:结构体变量的定义

评析:定义一个结构体类型的变量,可采用三种方法:

- ①先定义结构体类型,再定义变量名;
- ②在定义类型的同时定义变量;
- ③直接定义结构类型变量,即不出现结构体名。

选项 B 符合第三种定义方法。

(23)请读程序:

```

#include<stdio.h>
#include<string. h>
main()
{
char*s1= " AbCdEf " , *s2= " aB " ;
s1++; s2++;
printf( " %d\n " ,strcmp(s1, s2));
}

```

上面程序的输出结果是_____。

A)正数 B)负数 C)零 D)不确定的值

(23)A

知识点:字符串比较函数 strcmp()的应用

评析:strcmp(X1,X2)是串比较函数,当 X1>X2 时返回值为正数,当 X1<X2 时返回值为负数,当 X1=X2 时返回值为零。

本题中“s1”、“s2”分别表示这两个串中第一个字符的地址,s1++和 s2++是将指针指向串的第二个字符,则*s1为“bCdEf”,*s2为“B”。而在字符串比较中。大小的确定是由各个字符串相应位置字符的 ASCII 码值的大小决定的。“B”的 ASCII 码值为 66,”b”的 ASCII 码值为 98,所以 s1>s2,返回值为正数。

(24)请读程序:

```

#include<stdio.h>
func(int a, int b){
int c;
c=a+b;
return C;
}
main(){
int x=6, y=7, z=8, r;
r=func((x--, y++, x+y), z--);
printf( " %d \n " , r);
}

```

上面程序的输出结果是_____。

A)11 B)20 C)21 D)31

(24)C

知识点:自增、自减运算

评析: 函数 func()的作用是返回两个形参的和, 第一个形参是 x、y 分别自减和自增后的和, 其中(x--, y++, x+y)是一个逗号表达式, 它的值应该等于 x+y, 所以整个表达式(x--,y++, x+y)的值为 13, 而第二个形参的值为 8(根据语法规则, 应当先使用, 后自增), 所以 func()的返回值为 13+8=21。

(25)请读程序:

```
#include<stdio.h>
main()
{
    int a, b;
    for(a=1, b=1; a<=100; a++){
        if(b>=20)break;
        if(b%3==1){b+=3; continue; }
        b_+=5;
    }
    printf(" %d\n ", a);
}
```

上面程序的输出结果是_____。

A)7 B)8 C)9 D)10

(25)B

知识点: break 语句和 continue 语句

评析: break 语句的作用是用于跳出循环体, 继续执行循环体下面的语句; 而 continue 语句的作用是用于跳出本次循环, 即跳过循环体中尚未执行的语句, 接着进行下一次是否执行循环的判定。“%”是求余运算符, 执行第一次循环时, 条件(b%3==1)为真, b=4, 继续下一次循环, 如此反复, 当 b=22 时, 条件(b>=20)为假, 跳出循环, 此时共循环 8 次, 即 a=8。

(26)请读程序片段(字符串内没有空格字符):

```
printf(" %d\n ",strlen(" ATS\n012\n\n "));
```

上面程序片段的输出结果是_____。

A)11 B)10 C)9 D)8

(26)C

知识点: 字符串的长度

评析: 这个语句的目的是输出“ATS\n012\n\n”这个串的长度, 在串中“\n”代表一个“\”, 为了和 printf()函数中的转义字符区分开来, 在语法上使用了两个反斜杠代替了一个反斜杠, 所以它仅仅为一个字符, 而“\1”代表数字 1, 也占一个字符, “\n”是回车换行符, 也占一个字符, 加上 A、T、s、0、1、2, 一共是 9 个字符。

(27)请选出可用作 C 语言用户标识符的一组标识符_____。

A)Void B)a3_b3 C)For D)2a

```
define _123 _abc DO
```

```
WORD IF case sizeof
```

(27)B

知识点: C 语言的标识符

评析：c 语言规定标识符只能由字母、数字和下划线三种符号组成，而且第一个字符必须是字母或下划线。

选项 A 中的 void 和 define 都和 c 语言的关键字重名，不合法；

选项 C 中的 case 和 c 语言的关键字重名，不合法；

选项 D 中的 2a 是数字打头而且 sizeof 和 c 语言的关键字重名，不合法。

(28)请选出以下程序的输出结果_____。

```
#include<stdio_h>
sub(int*s, int y)
{ static int t=3;
y=s[t]; t--;
}
main()
{ int a[]={1,2, 3, 4}, i, x=0;
for(i=0; i<4; i++){
sub(a,x); printf( " %d ", x); }
printf( " \n " );
}
```

A)1234 B)4321 C)0000 D)4444

(28)C

知识点：函数的形参和实参的关系

评析：x 作为函数 sub()的实参时，函数对 x 值的改变没有返回主函数，并不能使得 x 的值变化，所以在打印时，x 的值是始终不变的，即为 0。

(29)若有以下说明和语句，请选出哪个是对 c 数组元素的正确引用_____。

```
int c[4][5], (*cp)[5];
cp=c;
A)cp+1 B)*(cp+3) C)*(cp+1)+3 D)*(*cp+2)
```

(29)D

知识点：数组元素的引用

评析：cp=c 这个语句是将数组第 0 行的地址赋给了 cp。cp+1 使指针指向二维数组 c 的第一行；*(cp+3)是数组 c 的第三行的第 0 列的地址值；+(cp+1)+3 是数组 c 的第一行第三列的地址值。

(30)设有以下语句

```
char a=3, b=6, c;
c=a^b<<2;
```

则 c 的二进制值是_____。

A)00011011 B)00010100 C)00011100 D)00011000

(30)A

知识点:位运算

评析：“<<”是 c 语言中规定的左移运算符，例如，a=a<<2，这个语句即是将 a 的二进制数左移两位，左移一位相当于该数乘以 2，左移两位相当于该数乘以 2 的 2 次方；^是异或运算符，所以，c 的二进制值应为 00011011。

(31)设有

```
static char str[]=" Beijing " ;
则执行
```

printf(" %d \n " ,strlen(strcpy(str, " China ")));

后的输出结果为_____。

A)5 B)7 C)12 D)14

(31)A

知识点：字符串的长度

评析:在执行 printf()函数前，数组 str 的长度是 7，但是当使用 strcpy()函数将新的值赋给 str 后，strlen()函数返回的应当是现在的 str 字符串的字符个数，即是 5。

(32)以下程序的输出结果是_____。

```
#include<stdio.h>
```

```
main()
```

```
{ int i;
```

```
for(i=1; i<5; i++){
```

```
if(i%2)printf( " * " );
```

```
else continue;
```

```
printf( " # " );
```

```
}
```

```
print " $ \n " );
```

```
}
```

A)*#*#*# \$ B)#*#*#* \$ C)*#*# \$ D)#*#* \$

(32)C

知识点:if 语句的判断

评析:当 i 不可以整除 2 时打印"*,然后打印"#",不能整除 2 则执行 continue,跳过 printf("#"); 语句,结束本次循环,返回到循环的起始点。当循环结束后,打印 "\$"。

(33)有以下程序

```
#include<stdio. h>
```

```
main()
```

```
{ int c;
```

```
while((c=getchar())!= ' \n' ){
```

```
switch(C-'2'){
```

```
case 0: case 1: putchar(c+4);
```

```
case 2: putchar(c+4); break;
```

```
case 3: putchar(c+3);
```

```
default: putchar(c + 2); break; }
```

```
}
```

```
}
```

从第一列开始输入以下数据,↵代表一个回车符。

2473↵

程序的输出结果是_____。

A)668977 B)668966 C)66778777 D)6688766

(33)A

知识点: switch()语句

评析: getchar()函数是从键盘接受一个字符输入;当用户键入的字符不是回车符时,会进入一个多分支选择语句,根据表达式 c- '2' 的值进行分支选择: putchar()是在屏幕上打印一个字符,最后的结果应当为 668977。

(34)以下程序的输出结果是_____。

```
main()
{ int w=5; fun(w); printf(" \n"); }
fun(int k)
{ if(k>0) fun(k_1);
printf(" %d ", k);
}
```

A)5 4 3 2 1 B)0 1 2 3 4 5

C)1 2 3 4 5 D)5 4 3 2 1 0

(34)B

知识点：函数的递归调用

评析：函数的递归调用就是在调用一个函数的过程中又出现直接或间接地调用该函数本身。

fun 函数共被调用 6 次，即 fun(5)、fun(4)、fun(3)、fun(2)、fun(1)、fun(0)。其中 fun(5)是 main 函数调用的，其余是在 fun 函数中调用的。

(35)若 fp 是指向某文件的指针，且已读到此文件的末尾，则函数 feof(fp)的返回值是_____。

A)EOF B)0 C)非零值 D)NULL

(35)C

知识点：文件结束符的返回值

评析：函数 feof 是用来判断文件是否已读到末尾，如果已读到末尾则返回非零值，否则返回 0。

(36)算法的时间复杂度是指_____。

A)执行算法程序所需要的时间

B)算法程序的长度

C)算法执行过程中所需要的基本运算次数

D)算法程序中的指令条数

(36)C

知识点：算法复杂度的概念和意义(时间复杂度与空间复杂度)

评析：所谓算法的时间复杂度，是指执行算法所需要的计算工作量。为了能够比较客观地反映出一个算法的效率，在度量一个算法的工作量时，不仅应该与所使用的计算机、程序设计语言以及程序编制者无关，而且还应该与算法实现过程中的许多细节无关。为此，可以用算法在执行过程中所需基本运算的执行次数来度量算法的工作量。

(37)下列叙述中正确的是_____。

A)线性表是线性结构 B)栈与队列是非线性结构

C)线性链表是非线性结构 D)二叉树是线性结构

(37)A

知识点：线性结构与非线性结构的概念

评析：根据数据结构中各数据元素之间相关关系的复杂程度，一般将数据结构分为两大类：线性结构与非线性结构。如果一个非空的数据结构满足下列两个条件：(1)有且只有一个根结点；(2)每一个结点最多有一个前件，也最多有一个后件。则称该数据结构为线性结构，又称线性表。所以线性表、栈与队列、线性链表都是线性结构，而二叉树是非线性结构。

(38)下面关于完全二叉树的叙述中，错误的是_____。

A)除了最后一层外，每一层上的结点数均达到最大值

B)可能缺少若干个左右叶子结点

C)完全二叉树一般不是满二叉树

D)具有结点的完全二叉树的深度为 $\lceil \log_2 n \rceil + 1$

(38)B

知识点：二叉树的定义及其存储结构

评析：这里考察完全二叉树与满二叉树的定义及二叉树的性质。满二叉树指除最后一层外每一层上所有结点都有两个子结点的二叉树。完全二叉树指除最后一层外，每一层上的结点数均达到最大值，在最后一层上只缺少右边的若干子结点(叶子结点)的二叉树。因此选项 A 是正确的，而选项 B 是错误的。由定义可知，满二叉树肯定是完全二叉树，而完全二叉树一般不是满二叉树，因此选项 c 是正确的叙述。选项 D 即二叉树性质(5)，也是正确的。

(39)结构化程序设计主要强调的是_____。

A)程序的规模 B)程序的易读性

C)程序的执行效率 D)程序的可移植性

(39)B

知识点：结构化程序设计

评析：结构化程序设计主要强调的是结构化程序清晰易读，可理解性好，程序员能够进行逐步求精、程序证明和测试，以保证程序的正确性。

(40)在软件生命周期中，能准确地确定软件系统必须做什么和必须具备哪些功能的阶段是_____。

A)概要设计 B)详细设计 C)可行性分析 D)需求分析

(40)D

知识点：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

评析：通常，将软件产品从提出、实现、使用维护到停止使用退役的过程称为软件生命周期。也就是说，软件产品从考虑其概念开始，到该软件产品不能使用为止的整个时期都属于软件生命周期。软件生命周期的主要活动阶段为：

① 可行性研究和计划制定。确定待开发软件系统的开发目标和总的要求，给出它的功能、性能、可靠性以及接口等方面的可行方案，制定完成开发任务的实施计划。

②需求分析。对待开发软件提出的需求进行分析并给出详细定义，即准确地确定软件系统的功能。编写软件规格说明书及初步的用户手册，提交评审。

③软件设计。系统设计人员和程序设计人员应该在反复理解软件需求的基础上，给出软件的结构、模块的划分、功能的分配以及处理流程。

④软件实现。把软件设计转换成计算机可以接受的程序代码。即完成源程序的编码，编写用户手册、操作手册等面向用户的文档，编写单元测试计划。

⑤软件测试。在设计测试用例的基础上，检验软件的各个组成部分。编写测试分析报告。

⑥运行和维护。将已交付的软件投入运行，并存运行使用中不断地维护，根据新提出的需求进行必要而且可能的扩充和删改。

(41)数据流图用于抽象描述一个软件的逻辑模型，数据流图由一些特定的图符构成。下列图符名标识的图符不属于数据流图合法图符的是_____。

A)控制流 B)加工 C)数据存储 D)源和潭

(41)A

知识点：结构化分析方法，数据流图，数据字典，软件需求规格说明书

评析：数据流图从数据传递和加工的角度，来刻画数据流从输入到输出的移动变换过程。数据流图中的主要图形元素有：加工(转换)、数据流、存储文件(数据源)、源和潭。

(42)软件需求分析一般应确定的是用户对软件的_____。

A)功能需求 B)非功能需求 C)性能需求 D)功能需求和非功能需求

(42)D

知识点：结构化设计方法

评析：软件需求分析中需要构造一个完整的系统逻辑模型，理解用户提出的每一功能与性能要求，是用户明确自己的任务。因此，需求分析应确定用户对软件的功能需求和非功能需求。

(43)下述关于数据库系统的叙述中正确的是_____。

- A)数据库系统减少了数据冗余
- B)数据库系统避免了一切冗余
- C)数据库系统中数据的一致性是指数据类型的一致
- D)数据库系统比文件系统能管理更多的数据

(43)A

知识点：数据库的基本概念：数据库，数据库管理系统，数据库系统

评析：由于数据的集成性使得数据可为多个应用所共享，特别是在网络发达的今天，数据库与网络的结合扩大了数据关系的应用范围。数据的共享自身又可极大地减少数据冗余性，不仅减少了不必要的存储空间，更为重要的是可以避免数据的不一致性。所谓数据的一致性是指在系统中同一数据的不同出现应保持相同的值，而数据的不一致性指的是同一个数据在系统的不同拷贝处有不同的值。

(44)关系表中的每一横行称为一个_____。

- A)元组 B)字段 C)属性 D)码

(44)A

知识点：数据库的基本概念：数据库，数据库管理系统，数据库系统

评析：在关系数据库中，关系模型采用二维表来表示，简称“表”。二维表是由表框架及表元组组成。在表框架中，按行可以存放数据，每行数据称为元组。

(45)数据库设计包括两个方面的设计内容，它们是_____。

- A)概念设计和逻辑设计 B)模式设计和内模式设计
- C)内模式设计和物理设计 D)结构特性设计和行为特性设计

(45)A

知识点：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

评析：数据库设计可分为概念设计与逻辑设计。数据库概念设计的目的是分析数据间内存语义关联，在此基础上建立一个数据的抽象模型。数据库逻辑设计的主要工作是将 E-R 图转换为指定的 RDBMS 中的关系模型。

(46)请读程序：

```
#include<stdio. h>
main()
{
    int a; float b, c;
    scanf( " %2d%3f%4f" , &a, &b, &c);
    printf( " \na=%d, b=%C c=%f\n" ,a,b,c);
}
```

若运行时从键盘上输入 9876543210↵，则上面程序的输出结果是_____。

- A)a=98,b=765,c=4321
- B)a=10,b=432,c=8765
- C)a=98,b=765. 000000,c=4321. 000000
- D)a=98,b=765. 0,c=4321. 0

(46)C

知识点：格式输入、输出函数

评析：scanf()把刚户从键盘录入的数字的第 1、2 位存入整型变量 a；把第 3、4、5 位存入单精度实型变量 b，由于“f”是以小数形式输出单、双精度数。隐含输出 6 位小数，所以 b=4321.000000；把第 6、7、8、9 位存入单精度实型变量 c，用户录入的第 10 位被 scanf()遗弃。这时变量 fd、b、c 的值分别为：98、765. 000000、432 1. 000000。

(47)请选出以下程序的输出结果_____。

```
#include<stdio. h>
sub(x,y,z)
int x,y,*z;
{*z=y-x; }
main(){
int a,b,c;
sub(10, 5, &a); sub(7, a, &b); sub(a, b, &c);
printf( " %d, %d, %d\n " , a, b, c);
}
```

A)5,2,3 B)-5, -12, -7 C)-5, -12, -17 D)5, -2, -7

(47)B

知识点：函数的调用

评析：sub()函数的作用是形参 y 和 x 的差赋给了 z 指向的那个内存地址，所以在 sub(10, 5, &a)中，10 和 5 属于值传递，直接将数值 10 和 5 分别传递给了变量 x 和 y，而对于 a 是属于地址传递，也就是 a 与 z 指向了同一个存储单元，在执行函数后，a 的值随 *z 变化，但 b、c 值并不改变，所以此次函数被调用后，a 的值为 y-x=-5，同理可知，在 sub(7,a,&b)后，b 的值发生变化，其值为 -5-7=-12，在 sub(a,b,&c)后，c 的值发生变化，其值为 -12-(-5)=-7。

(48)若 x 是整型变量，pb 是基类型为整型的指针变量，则正确的赋值表达式是_____。

A)pb=&x; B)pb=x; C)*pb=&x; D)*pb=*x

(48)A

知识点：赋值表达式

评析：选项 A 是将变量 x 的地址赋给指针变量 pb，使 pb 指向 x，故为正确的赋值表达式。

选项 B 是将 x 的值当作地址赋给指针 pb，pb 指向一个地址等于 x 值的内存单元。

选项 C 是将 x 的地址赋给指针 pb 指向的那个地址等于 x 值的内存单元。

选项 D 是不正确的语句。

(49)若要用下面的程序片段使指针变量 p 指向一个存储整型变量的动态存储单元

```
int *p;
p=_____malloc(sizeof(int));
```

则应填入_____。

A)int B)int* C)(*int) D)(int*)

(49)D

知识点：强制类型转换

评析：不论 p 是指向什么类型的指针变量，都可以通过强制类型转换的方法使之类型一致，

强制类型转换的格式为(数据类型*)。

(50)若执行下面程序时从键盘上输入 5，

```
main()
{
int x;
```

```
scanf(" %d", &x);
if(x++>5)printf(" %d\n", x);
else printf(" %d\n", x--);
}
```

则输出是_____。

A)7 B)6 C)5 D)4

(50)B

知识点: if 条件的判断

评析: 根据 C 语言的语法, $x++$ 是在使用 x 之后, 再将 x 的值加 1, 在 if 语句中, x 的值为 5, 条件不成立, 执行 else 后面的语句, 因为 x 的值已经加 1, 为 6, 所以打印结果为 6。

(51) 设 a 、 b 和 c 都是 int 型变量, 且 $a=3$ 、 $b=4$ 、 $c=5$, 则下面的表达式中, 值为 0 的表达式是_____。

A) 'a' && 'b' B) $a \leq b$ C) $a || c \&\& b - c$ D) $!(a < b) \&\& !c || 1$

(51)D

知识点: 几种运算符的使用

评析: 选项 A: 'a' && 'b' 是字符 a 与 b 的相与, 故不为 0。

选项 B: $a \leq b$, 由题中变量赋值可知, 结果为 1。

选项 C: $a || c \&\& b - c$, 此表达式先做算术运算 $b - c$, 结果为 -1。而 $+c$ 属于单目运算符, 由于 c 初值为 5, 经过单目运算符运算后, 还是 5, 下面再进行逻辑与的运算, 即 $5 \&\& -1$ 结果为 1 (因为 C 语言中除 0 代表假外, 其他任一个数都代表真), 最后 $a || 1$, 结果为 1。

选项 D: $!(a < b) \&\& !c || 1$, 此表达式先运算最外面括号内的表达式 $(a < b) \&\& !c || 1$, 然后进行非运算, 由于 $(a < b) \&\& !c || 1$ 中先算小括号内的 $a < b$ 结果为 1, 再按逻辑运算符的运算顺序: !, &&, ||, 进行运算后得 $(a < b) \&\& !c || 1$ 的值为 1, 所以最后进行非运算知 D 选项的运算结果为 0。

(52) 设有如下程序

```
#include <stdio.h>
main()
{
    int **k, *j, i=100;
    j=&i; k=&j;
    printf(" %d\n", **k);
}
```

上述程序的输出结果是_____。

A) 运行错误 B) 100 C) i 的地址 D) j 的地址

(52)B

知识点: 指针变量的引用

评析: $j = \&i$, j 的值就是 i 的地址, $*j = 100$, 将 j 的地址赋给 k , 这时 $*k = j$, 那么, $**k = *j$, 而 $*j = 100$, 所以 $**k = 100$, 最后的打印结果应当为 100。

(53) 设有以下语句, 其中不是对 a 数组元素的正确引用的是: _____ (其中 $0 \leq i < 10$)

```
int a[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, *p = a;
```

A) $a[p - a]$ B) $\&a[i]$ C) $p[i]$ D) $*(a + i)$

(53)D

知识点: 通过指针引用数组元素

评析: 观察程序可知, a 实际上就是数组 a 的首地址, 所以 “ $*(a + i)$ ” 表示的就是数组 a 中的

第 i 个元素的值, 进而, 我们可以知道 $*(a+i)$ 必然不是对 a 数组元素的正确引用。

(54)有以下程序:

```
#include<stdio. h>
main(){
char c[6];
int i=0;
for(; i<6; c[i]=getchar(), i++);
for(i=0; i<6; i++)putchar(c[i]);
printf( " \n " );
}
```

如果从键盘上输入:

ab<回车>

c<回车>

def<回车>

则输出结果为_____。

A)a B)a C)ab D)abcdef

b b c

c c d

d d

e

f

(54)C

知识点: 字符数据的输入、输出

评析: 1. getchar()

此函数的作用是从终端(或系统隐含指定的输入设备)输入一个字符。请注意: **getchar()** 只能接受一个字符(回车符也算是一个字符)。**getchar** 函数得到的字符可以赋给一个字符变量或整型变量, 也可不赋给任何变量, 作为表达式的一部分。

2. putchar()

此函数的作用是向终端输出一个字符, 也可以输出控制字符, 如回车符。使输出的当前位置移到下一行的开头。

本题在输入字符时, ab 和 c 后面的回车符分别赋给了 c[2]和 c[4], 所以, 正确答案为 C。

(55)下面程序

```
#include<stdio. h>
#include<string. h>
main()
{ char*p1: " abc ", *p2= " ABC ",str[50]= " xyz ";
strcpy(str+2, strcat(p1, p2));
printf( " %s\n " , str);
}
```

的输出是_____。

A)xyzabcABC B)zabcABC

C)yzabcABC D)xyabcABC

(55)D

知识点: 字符串的连接和拷贝

评析：strcat(字符串 1，字符串 2)的作用是连接两个字符串数组中的字符串，把字符串 2 接到字符串 1 的后面，结果放在字符串数组 1 中，本题中 strcat(p1, p2)函数将 *p1 和 *p2 指向的两个串连接了起来，将字符串 abcABC 放到了 *p1 所指向的存储单元中；而 strcpy(字符串 1，字符串 2)函数将字符串 2 复制到字符串 1 中去，本题将字符串 abcABC 复制到了 str+2 所指向的存储单元中，即将原 str 数组中的字符 z 及 z 向后的所有字符覆盖，所以在打印输出时，str 的值已经等于“xyabcABC”。

(56)下面程序

```
int aa[3][3]={ {2}, {4}, {6} };
main()
{ int i, *p=&aa[0][0];
  for(i=0; i<2; i++){
    if(i==0)aa[i][i+1]=*p+1;
    else ++p;
    printf(" %d ", *p);
  }
}
```

的输出是_____。

A)23 B)26 C)33 D)36

(56)A

知识点：for 循环语句

评析：观察题目，可以发现，*p=&aa[0][0]语句实际是将数组 aa 的首地址赋给了指针变量 p，将 i 的值带入 for 循环中，i=0 时，aa[0][1]=3，*p=2；*p 输出的是指针所指向的数组值，此时 p 所指向的是 aa[0][0]，输出 2，而 i=1 时执行了 ++p 操作，使指针向后移动指向了 aa[0][1]，所以在输出 *p 时应为 3。

(57)下面程序

```
main()
{ int x=100, a=10, b=20, ok1=5, ok2=0;
  if(a<b)
  if(b!=15)
  if(!ok1)x=1;
  else if(ok2)x=10;
  X=-1;
  Ptintf(" %d\n", x);
}
```

的输出是_____。

A)-1 B)0 C)1 D)不确定的值

(57)A

知识点：判断语句的循环嵌套

评析：第一个判断值为真，过渡到下一个判断，第二个判断为真，过渡到第三个判断……如此循环，在打印输出语句的前一行，程序给变量 x 赋了值，为一 1，所以，无论前期如何变化，最后的 x 值依然为一 1。

(58)下面程序

```
main()
{
```

```
int x=32;
```

```
printf( " %d\n " , x=x<<1);
```

```
}
```

的输出是_____。

A)100 B)160 C)120 D)64

(58)D

知识点：位运算

评析：<<是c语言中规定的左移运算符，例如， $a=a<<2$ ，这个语句即是将a的二进制数左移两位，左移一位相当于该数乘以2，左移两位相当于该数乘以2的2次方。所以， $x<<1=32*2=64$ 。

(59)以下程序的输出结果是_____。

```
#include<stdio. h>
```

```
#define FUDGE(y) 2. 84+y
```

```
#define PR(a) printf( " %d " , (int))(a))
```

```
#define PRINT l(a) PR(a); putchar( ' \n' )
```

```
main()
```

```
{ int x=2;
```

```
PRINT l(FUDGE(5)*X);
```

```
}
```

A)11 B)12 C)13 D)15

(59)B

知识点：带参数的宏定义

评析：在程序中如果有带实参的宏，则按#define 命令行中指定的字符串从左到右进行置换，如果串中包含宏中的形参，则将程序语句中相应的实参代替形参，如果宏定义中的字符串中的字符不是参数字符，则保留，这样就形成了置换的字符串。

根据以上原则，我们将实参带入已经定义的宏中，可以得出答案，最后打印出来的是“12”。

(60)以下程序段给数组所有的元素输入数据，请选择正确答案填入：

```
#include<stdio.h>
```

```
main()
```

```
{
```

```
int a[10], i=0;
```

```
while(i<10)scanf( " %d " , _____);
```

```
:
```

```
:
```

```
}
```

A)a+(i++) B)&a[i+1] C]a+i D]&a[++i]

(60)A

知识点：格式输入函数 scanf()

评析：因为a实际上就是数组a的首地址，而a+x则是数组中第x个元素的地址，所以在四个选项中，选项B和C只能输入一个数据，选项D不能给a[0]输入数据，只有选项A可以完成给数组所有的元素输入数据的任务。

(61)字符(char)型数据在微机内存中的存储形式是_____。

A)反码 B)补码

C)EBCDIC 码 D)ASCII 码

(61)D

知识点: 字符数据在内存中的存储形式

评析: 将一个字符常量放到一个字符变量中, 实际上并不是把该字符本身放到内存单元中去, 而是将该字符的 ASCII 码值放到存储单元中。

(62)下面程序的输出是_____。

```
typedef union
{ long x[2];
  int y[4];
  char z[8];
}MYTYPE;
MYTYPE them;
main()
{ printf( " %d \n " , sizeof(them)); }
```

A)32 B)16 C)8 D)24

(62)C

知识点: 共用体的长度

评析: sizeof(x)是一个标准 c 函数, 它的返回值是 x 型的数据结构占用的内存字节数。题目中定义了一个共用体, 共用体变量在内存中所占的长度等于最长的成员的长度, 所以共用体 MYTYPE 需要 8 个字节的存储空间, 由它定义的变量的长度当然也是 8。

(63)不能把字符串: “Hello!” 赋给数组 b 的语句是_____。

A)char b[10]={ ‘H’, ‘e’, ‘l’, ‘l’, ‘o’, ‘!’ };

B)char b[10]={ ‘h’, ‘e’, ‘l’, ‘l’, ‘o’, ‘!’ };

C)char b[10]; strcpy(b, " Hello! ");

D)char b[10]= " Hello! " ;

(63)B

知识点: 给数组赋值

评析: 在 c 语言中, 大写字母和小写字母被认为是两个不同的字符, 因此, “hello!” 和 “Hello!” 是两个不同的字符串。

(64)下面程序的输出是_____。

```
main()
{ int x=3, y=6, a=0;
  while(x++!=(y-=1))
  { a+=1;
    if(y<x)break;
  }
  printf( " x: %d, y: %d, a=%d\n " , x, y, a);
}
```

A)x=4, y=4,a=1 B)x=5, y=5,a=1

C)x=5, y=4, a=3 D)x=5, y=4, a=1

(64)D

知识点: 运算符

评析: 注意在 x++ 中 x 值的引用时, 这里应当是先引用, 后自加, 具体执行过程如下:

第一次 while 循环: x=3 与 y=5 比较, 条件为真, 执行 a=a+1=1; 此时, x 的值已为 4, 判断 y<x 不成立, 继续执行循环;

第二次 while 循环：x=4 与 y=4 比较，条件为假，此时 x 的值已为 5，退出 while 循环，执行 printf。

(65)下面程序的输出是_____。

```
main()
{char*s: " 12134211 ";
int v1=0,v2=0,V3=0,v4=0, k;
for(k=0; s[k]; k++)
switch(s[k])
{ default: V4++;
case '1': v1++;
case '3': v3++;
case '2': v2++;
}
printf( " v1=%d, v2=%d, v3=%d, v4=%d\n " , v1, v2,v3, V4);
}
```

A)v1=4, v2=2, v3=1, v4=1 B)v1=4, V2=9, V3=3, V4=1

C)v1=5, v2=8, v3=6, v4=1 D)v1=8, V2=8, V3=8, v4=8

(65)C

知识点：switch()语句

评析：当 switch 后面括弧内的表达式的值与某一个 case 后面的常量的表达式的值相等时，就执行此 case 后面的语句，若所有的 case 中的常量表达式的值都没有与表达式的值匹配的，就执行 default 后面的语句。本题中，for 循环的条件是 s 数组的数组元素值，只要将数组的值一一代入程序逐步求解即可。

(66)下面程序的输出是_____。

```
main()
{int k=11;
printf( " k=%d, k=%o, k=%x\n " , k, k, k);
}
```

A)k=11. k=12, k=11 B)k=11, k=13, k=13

C)k=11, k=013, k=0xb D)k=11, k=13, k=B

(66)D

知识点：格式字符

评析：在 c 语言格式字符的输出中，“%d”是以带符号的十进制形式输出整数，即 k=11：“%o”是以 8 进制无符号形式输出整数(不输出前导符 0)，即 k=13：“%x”是以 16 进制无符号形式输出整数(不输出前导符 0x)，即 k=B。

(67)以下叙述中正确的是_____。

A)c 语言比其他语言高级

B)C 语言可以不用编译就能被计算机识别执行

c)c 语言以接近英语国家的自然语言和数学语言作为语言的表达形式

D)C 语言出现的最晚，具有其他语言的一切优点

(67)C

知识点：c 语言风格

评析：计算机语言分为低级语言、汇编语言和高级语言，c 语言属于高级语言，但并不是说 c 语言比其他语言高级，所以选项 A 错误；除了低级语言外，其他各种语言都必须编译成能

被计算机识别的二进制数才能执行，选项 B 错误；c 语言出现于 1972 年到 1973 年间，并不是出现最晚的语言，所以选项 D 也是错误的。

(68)下列可用于 c 语言用户标识符的一组是_____。

- A)void define WORD B)a3_b3 _123 Car
C)For -abc IFCase D)2a D0 sizeof

(68)B

知识点：c 语言的标识符

评析：c 语言规定标识符只能由字母、数字和下划线 3 种字符组成，且第一个字符必须为字母或下划线，所以排除 c 和 D。c 语言还规定标识符不能为 c 语言的关键字，从而选项 A 是错误的，因为 void 为关键字。

(69)请选出正确的程序段_____。

A)int*p B)int*s, k;
Scanf(" %d " ,p); *s=100;
.....

C)int*s, k; D)int*s, k;
Char*p, c; char*p, e;
s=&k; s=&k;
p=&c; p=&c;
*p= 'a'; s=p;
..... *s=l;
.....

(69)C

知识点：指针变量

评析：本题的 A 和 B 犯了一个同样的错误，即指针变量 p 定义后并没有指向具体的变量。也就是说，p 中没有确定的地址值，它的值是不可预见的，所指向的单元也是不可预见的，因此不能进行赋值操作。另外，在选项 D 中，s 是 int 指针变量，p 是 char 型指针变量，所指向的内存单元所占用的字节数是不同的，因而不能将字符指针变量 p 的值赋给整型指针变量 s。

(70)若有下面的说明和定义，则 sizeof(struct aa)的值是_____。

```
struct aa
{
    int r1; double r2; float r3;
    union uu{char u1[5]; long u2[2]}ua;
}mya;
```

A)30 B)29 C)24 D)22

(70)D

知识点：结构体和共用体变量所占存储空间大小的计算方法

评析：结构体变量所占内存长度是各成员占的内存长度之和，每个成员分别占有自己的内存单元；共用体变量所占的内存长度等于最长的成员的长度。结构体变量 aa 中，成员 r1 占 2 个字节，r2 占 8 个字节，r3 占 4 个字节，共用体 ua 占 8 个字节，所以共占用 2+8+4+8=22 个字节。

(71)算法的空间复杂度是指_____。

- A)算法程序的长度 B)算法程序中的指令条数
C)算法程序所占的存储空间 D)算法执行过程中所需要的存储空间

(71)D

知识点：算法的复杂度

评析：一个算法的空间复杂度，一般是指执行这个算法所需的内存空间。

一个算法所占用的存储空间包括算法程序所占的空间、输入的初始数据所占的存储空间以及算法执行过程中所需要的额外空间。

(72)下列关于栈的叙述中正确的是_____。

- A)在栈中只能插入数据 B)在栈中只能删除数据
C)栈是先进先出的线性表 D)栈是先进后出的线性表

(72)D

知识点：栈的输入输出操作

评析：栈是限定在一端进行插入与删除的线性表。

栈是按照“先进后出”的或“后进先出”的原则组织数据的，因此，栈也被称为“先进后出”表或“后进先出”表。

(73)在深度为 5 的满二叉树中，叶子结点的个数为_____。

- A)32 B)31 C)16 D)15

(73)C

知识点：二叉树的概念

评析：所谓满二叉树是指除最后一层外，每层上的所有结点都有两个子结点。也就是说，在满二叉树中，每一层上的结点数都达到最大值，即在满二叉树的第 K 层上有 2^{k-1} 个结点，且深度为 m 的满二叉树有 2^m 个结点。

在满二叉树中，最后一层的结点数就是叶子结点的个数，本题中深度为 5，故叶子结点数为 $2^5 - 1 = 24 = 16$ 。

(74)对建立良好的程序设计风格，下面描述正确的是_____。

- A)程序应简单、清晰、可读性好 B)符号名的命名要符合语法
C)充分考虑程序的执行效率 D)程序的注释可有可无

(74)A

知识点：程序设计风格

评析：要形成良好的程序设计风格，主要应注重和考虑下述一些因素：符号名的命名应具有一定的实际含义，以便于对程序功能的理解；正确的注释能够帮助读者理解程序；程序编写应优先考虑清晰性，除非对效率有特殊要求，程序编写要做到清晰第一，效率第二。

(75)下面对对象概念描述错误的是_____。

- A)任何对象都必须有继承性 B)对象是属性和方法的封装体
C)对象间的通讯靠消息传递 D)操作是对象的动态性属性

(75)A

知识点：对象的概念

评析：对象是由数据和容许的操作组成的封装体，与客观实体有直接的对应关系。对象之间通过传递消息互相联系，以模拟现实世界中不同事物彼此之间的联系。

(76)下面不属于软件工程的 3 个要素的是_____。

- A)工具 B)过程 C)方法 D)环境

(76)D

知识点：软件：[程]的要素

评析：软件工程包括 3 个要素，即方法、工具和过程。

(77)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

(77)B

知识点：软件设计工具

评析：程序流程图(PFD)是一种传统的、应用广泛的软件过程设计表示工具，通常也称为程序框图，其箭头代表的是控制流。

(78)在数据管理技术的发展过程中，经历了人工管理阶段、文件系统阶段和数据库系统阶段。其中数据独立性最高的阶段是_____。

A)数据库系统 B)文件系统 C)人工管理 D)数据项管理

(78)A

知识点：数据管理技术的发展

评析：在数据管理技术的发展过程中，经历了人工管理阶段、文件系统阶段和数据库系统阶段。其中数据独立性最高的阶段是数据库系统。

(79)用树形结构来表示实体之间联系的模型称为_____。

A)关系模型 B)层次模型 C)网状模型 D)数据模型

(79)B

知识点：数据库模型

评析：层次模型是最早发展出来的数据库模型。它的基本结构是树形结构，这种结构方式在现实世界中很普遍，如家族结构、行政组织机构，它们自顶向下、层次分明。

(80)关系数据库管理系统能实现的专门关系运算包括_____。

A)排序、索引、统计 B)选择、投影、连接
C)关联、更新、排序 D)显示、打印、制表

(80)B

知识点：关系的运算操作

评析：关系数据库管理系统能实现的专门关系运算，包括选择运算、投影运算、连接运算。

(81)下列语句中符合 C 语言语法的赋值语句是_____。

A)a=7+b+c=a+7 B)a=a+7
C)a=7+b,b++, a+7; D)a=7+b, c。 a+7;

(81)D

知识点：C 表达式类型

评析：本题的考查点是赋值语句。

c 语言中，赋值语句具有其他高级语言的赋值语句的一切特点和功能。但也有不同：

①c 语言中的赋值号“=”是一个运算符，在其他大多数语言中赋值号不是运算符。

②关于“赋值表达式”这一概念，其他大多数高级语言并没有，但在 c 语言中必须区分：例如：i=i+1 是一个表达式，而不是语句；i=i+1; 是一个语句。可以看到，一个表达式的最后加一个分号就成了一个语句。

由于赋值语句是由赋值表达式加分号构成，而赋值表达式是赋值运算符“=”将一个变量和一个表达式连接起来的式子，所以选项 A、B 均不是合法的赋值语句；选项 c 中，存在两种运算符：逗号运算符和赋值运算符，其中赋值运算符的优先级高，逗号表达式“a=7+b, b++, a+7”的值就是表达式“a+7”的值，即选项 C 也就可以表示为：a+7;。由此可见，选项 C 也不是一个合法的赋值语句。选项 D 是用逗号运算符连接的两个赋值语句。因此符合题目要求的应该是选项 D。

(82)下面程序

```
main()
{int y=9;
for(; y>0; y--){
```

```

if(y%3==0)
{ printf( " %d ",--y);
continue;
}
}
}

```

的输出是_____。

A)741 B)852 C)963 D)875421

(82)B

知识点：用 if 语句实现选择结构

评析：本题的考查点是 if 判断语句。

“%”是求余运算符，所以 if 判断语句中实际是指出只有当 y 可以被 3 整除时方可以继续；--y 是先进行 y 的自减运算，再使用 y 值。所以，最后的打印结果应当为“852”。

(83)下面程序

```

main()
{
int a=-1,b=4,k;
k=(a++<=0)&&!(b--<=0));
printf( " %d%d%d\n " ,k,a,b);
}

```

的输出是_____。

A)0 0 3 B)0 1 2 C)1 0 3 D)1 1 2

(83)C

知识点：c 表达式类型

评析：本题的考查点是逻辑与及自增、自减运算。

a++是先使用 a 值再将 a 的值加 1，b--是先使用 b 的值再将 b 的值减 1，所以题中的逻辑运算表达式的值应当为真，即为 1，而经过运算后，a 和 b 的值已经分别变成 0 和 3 了。

(84)下面的程序片段

```

y=-1;
if(X!--0)
if(x>0)y=1; else y=0;

```

所表示的数学函数关系是_____。

-1(x<0) 1(x<0)

A) y= 0(x=0) B) y= -1(x=0)

1(x=0) 0(x>0)

0(x=0) -1(x<0)

C) y= -1(x=0) D) y= 1(x=0)

1(x>0) 0(x>0)

(84)C

知识点：用 if 语句实现选择结构

评析：本题的考查点是根据程序写出数学函数关系。

题中的两个判断语句实际上指出了 y 取值的条件，即当 x 大于零时 y 取 1；当 x 小于零时 y 取 0；当 x 等于零时 y 值不变，为一 1。

(85)下面的程序

```

main()
{ int x=3,y=0,z=0;
if(x=y+z)printf( " * * * * " );
else printf( " # # # # " );
}
_____O

```

A)有语法错误不能通过编译

B)输出* * * *

C)可以通过编译，但是不能通过连接，因而不能运行

D)输出# # # #

(85)D

知识点:c 运算符的种类、运算优先级和结合性

评析:本题的考查点是运算符在表达式中的处理原则。

本题中把 y+z 的值赋给 x，由于 y 和 z 都等于 0，所以 x 的值也为 0，即表达式值为 false，输出####。特别提醒：if 语句中 if 后面的表达式一般为逻辑或关系表达式，但也可以是任意的数值类型(包括整型、实型、字符型、指针型数据)。

(86)下面函数

```

int fun1(char*x)
{ char*y=x;
while(*y++);
return(y-x-1);
}

```

的功能是_____。

A)求字符串的长度 B)比较两个字符串的大小

c)将字符串 x 复制到字符串 y D)将字符串 x 连接到字符串 y 后面

(86)A

知识点: while 和 do-while 循环结构

评析: 本题的考查点是 while()循环语句。

本题首先要注意的是对 y++ 的运算，由于++和*为同一优先级别，且结合方向为自右向左，因此它相当于*(y++)。由于++在 y 的右侧，是“后加”，因此先对 y 的原值进行运算，然后使 y 的值改变。在程序中，首先将指针 x 和 y 指向同一个字符串的首地址。举个例子，假如。x=" abc":

第一次判断 while 循环条件: *y= 'a'，然后使得 y 加 1，指向了字符 b;

第二次判断 while 循环条件: *y= 'b'，然后使得 y 加 1，指向了字符 c;

第三次判断 while 循环条件: *y= 'c'，然后使得 y 加 1，指向了字符 \0(字符串结束标志);

第四次判断 while 循环条件: *y= '\0'，然后使得 y 加 1，指向了字符 '\0' 后面的一个存储空间;

执行 return(y-x-1);，返回值为 3，所以此函数的功能是求字符串的长度，需要注意的是，在返回长度时一定要减 1，否则计算出的串长度将会多 1。

(87)已知字母 A 的 ASCII 码为十进制的 65，下面程序

```

main()
{ char ch1,ch2;
ch1= 'A' + '5' - '3';
ch2= 'A' + '6' - '3';
}

```

```
print 削%d,%Cn",ch1,ch2);
}
```

的输出是_____。

A)67,D B)B,C C)C,D D)不确定的值

(87)A

知识点：数据的输入与输出，输入输出函数的调用。

评析：本题的考查点是格式输出函数 printf 的格式字符。

由于字符“5”和“3”的 ASCII 码相差为 2。所以 ch1 经过运算后的值应为 65+2=67；同理，ch2 经过运算后的值应为 65+3=68，即是字符“D”。

(88)以下程序的输出结果是_____。

```
#include<stdio. h>
```

```
main()
```

```
{
```

```
int a,b,d=241;
```

```
a=d / 100%9;
```

```
b=(-1)&&(-1);
```

```
printf( " %d, %d\n " , a, b);
```

```
}
```

A)6, 1 B)2,1 C)6,0 D)2,0

(88)B

知识点：c 运算符的种类、运算优先级和结合性

评析：本题的考查点是运算符的应用。

第一个表达式中，运算的方向是从左至右，所以 a 的值为 2；

第二个表达式中，等式右边是一个逻辑表达式，由于结果为真，所以表达式的值为 1，即 b 为 1。

(89)以下程序的输出结果是_____。

```
#include<stdio. h>
```

```
main()
```

```
{
```

```
printf( " %d\n " ,NULL);
```

```
}
```

A)不确定的(因变量无定义) B)O

C)-1 D)l

(89)B

知识点：“文件包含”处理

评析：本题的考查点是对“NULL’”的理解。

NULL 在 c 编译器中是一个值为 0 的常量，它的定义在头文件“stdio. H”中，本题的程序在一开始就包含了这个头文件，所以，在打印的时候，将打印出 0。

(90)以下程序的输出结果是_____。

```
main()
```

```
{ int k=4,m=1, p;
```

```
p=func(k, m); printf( " %d, " , p);
```

```
p=func(k, m); printf( " %d\n " , p);
```

```
}
```



```
func(int a,int b)
{ static int m=0, i=2;
i+=m+1; m=i+a+b;
return(m);
}
```

A)8,17 B)8,16 C)8,20 D)8,8

(90)A

知识点：函数的正确调用

评析：本题的考查点是函数的调用。

将 k 和 m 的值带入函数 func 中，第一次调用完 func()后，其中 m 和 i 的值要保留，并在第二次调用时使用，这是因为两者都是局部静态变量。模块的代码开始运行后，使用 static 语句声明的变量会一直保持其值，直至该模块复位或重新启动。所以最后的结果是 8 和 17。

(91)以下程序调用 findmax 函数求数组中值最大的元素在数组中的下标，请选择填空：

```
#include<stdio. h>
findmax(int*s,int t,int*k)
{ int p;
for(p=0, *k=p; p<t; p++)
if(s[p]>s[*k])_____;
}
main()
{ int a[10], i,k;
for(i=0; i<10; i++)scanf( " %d " ,&a[i]);
fmdmax(a,10,&k);
printf( " %d, %d\n " , k, a[k]);
}
```

A)k=p B)*k=p-s C)k=p-s D)*k=p

(91)D

知识点：一维数组和多维数组的定义、初始化和引用

评析：本题的考查点是求数组中最大值的下标。

k 是一个指针，它的值是一个地址，要通过它为主函数的变量改变数据，必须使用 *p，这样就可以把一个下标数赋给 p 指针指向的那个内存单元，所以横线处应填入 *k=p。

(92)以下函数调用语句中含 _____ 个实参。

```
func((exp1,exp2), (exp3,exp4,exp5));
```

A)1 B)2 C)4 D)5

(92)B

知识点：形参与实参，参数值的传递

评析：本题的考查点是对实参的理解。

实参可以是常量、变量或表达式，(exp1,exp2)是逗号表达式，它的值是 exp2 的值；

(exp3,exp4,exp5)是逗号表达式，它的值是 exp5 的值。所以该函数调用语句含有的实参个数是 2，即：(exp1,exp2)和(exp3,exp4,exp5)。

(93)在 c 语言中，要求运算数必须是整型的运算符是_____。

A)% B)/ C)< D)!

(93)A

知识点：位运算符的含义及使用

评析：本题的考查点是运算符。

题目的四个选项中,(B)(C)(D)都不要求运算数必须为整数,参与模运算(%)的运算数必须是整型数据。

(94)C 语言提供的合法的数据类型关键字是_____。

A)Double B)float C)integer D)Char

(94)B

知识点:c 的数据类型

评析：本题的考查点是 c 语言中的关键字。

选项 A 和 D 中第一个字母大写, 不正确;

选项 c 是 PASCAL 中的整型关键字, 不是 c 语言中的关键字。

(95)若有以下说明:

int a[10]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, *p=a;

则值为 6 的表达式是_____。

A)串 p+6 B)木(p+6)C)半 p+=5 D)p+5

(95)C

知识点: 指针数组

评析：本题的考查点是通过指针引用数组元素。

选项 A, *p+6, 其值等于 1+6=7;

选项 B, *(p+6), 其值等于 a[6]=7;

选项 c, *p+=5, 其值等于*p=*p+5=1+5=6;

选项 D, p+5 是个地址值。

(96)下面程序的输出是_____。

```
main()
{
    int m=0xa,n=2;
    m+=a;
    printf( " %X\n " , m);
}
```

A)C B)c C)99 D)2

(96)A

知识点: 格式字符

评析：格式控制符 x 表示数据按十六进制形式输出(不输出前导符 0x)。本题在执行语句 m+=n; 后, m 的值变为字母 c, 又因为在输出 a-f 时, 如果格式控制符用 X, 则以大写字母输出;如果用 x, 则以小写字母输出, 故本题答案为 A。

(97)以下程序的输出结果是_____。

```
#include<string. h>
main()
{char*a=" abcdefghi " ; int k;
fun(a); puts(a);
}
fun(char *s)
{ int x, y; char c;
for(x=0, y=strlen(s)-1; x<y; x++, y--)
{ c=s[y]; s[y]=s[x]; s[x]=c; }
```

```

}
A)ihgfedcba B)abcdefghi C)abcdedcba D)ihgfeefghi
(97)A

```

知识点：字符串与字符数组

评析：本题的考查点是字符串的应用。

观察子函数 fun()可知，它是实现一个数组首末元素位置互相依次交换。联系主函数，可以看出，是对字符串中的元素依次进行首末位置的对调。

(98)以下程序的输出结果是_____。

```

main()
{ union{char i[2];
int k;
}r;
r. i[0]=2; r. i[1]=0;
printf( " %d\n " , r. k);
}

```

A)2 B)1 C)0 D)不确定

(98)A

知识点：结构体和共用体类型数据的定义方法和引用方法

评析：本题的考查点是共用体类型数据的特点。

根据共用体的定义可知：共用体 r 的成员 k 和成员 i[2]是共用同一段内存空间，所以，当程序给 r.i[0]赋值后，实际上，共用体成员 k 的值也确定了，为 2。所以打印输出的结果应当为 2。

(99)以下程序的输出结果是_____。

```

#define f(x) x*x
main()
{ int a=6, b=2, c;
c=f(a) / f(b);
printf( " %d\n " , c);
}

```

A)9 B)6 C)36 D)18

(99)C

知识点：宏定义：不带参数的宏定义；带参数的宏定义

评析：本题的考查点是带参数的宏定义。

观察程序段可知，对带参数的宏的展开只是将语句中的宏名后面括号内的实参字符串代替#define 命令行中的形参。本题中 c=f(a) / f(b)用#define f(x)X*X 代替得 c=f(6) / f(2)，即 c=6*6 / 2*2，结果为 36。

(100)以下程序的输出结果是_____。

```

main()
{ int i;
for(i=1; i<6; i++)
{ if(i%2){printf( " # " ); continue; }
printf( " * " );
}
printf( " \n " );
}

```

```

}
A)#####B)#####C)*****D)#####
(100)A

```

知识点：用 if 语句实现选择结构

评析：本题的考查点是 if 条件的判断。

i 从 1 到 5 循环 5 次，i 为 1 时，满足条件，打印输出“#”；i 为 2 时，条件为假，打印输出“*”。也就是说，当 i 为奇数时，条件为真，程序打印输出“撑”，否则，输出“*”。

(101)有如下程序

```

main()
{
int n=9;
while(n>6)
{
n--;
printf( " %d " , n);
}
}

```

该程序的输出结果是_____。

A)987 B)876 C)8765 D)9876

(101)B

知识点：while 和 do-while 循环结构

评析：本题考查点是 while 语句循环次数的问题。

符合 n>6 的 n 只有 7,8,9；所以循环次数为 3。选项 c、D 可以排除。又因 n 在输出之前要执行 n--，所以输出结果只能是 876。

(102)执行下面的程序段

```

int x=35;
char z= 'A';
int B;
B=((x&15)&&(z< 'a' ));

```

后，B 的值为_____。

A)0 B)1 C)2 D)3

(102)B

知识点：简单的位运算

评析：本题的考查点是逻辑与及位与运算。

从整个表达式来看是个逻辑表达式，而与运算符“&&”的右边的子表达式中由于 z 的值为“A”，而 A 的 ASCII 码的值小于 a 的 ASCII 码值，所以这个子表达式的值为真，即 1；在与运算符“&&”的左边的子表达式是个基于位运算的子表达式，将 x 的值与 15 做位与运算，表达式值不为零，所以两个子表达式的与值应当为 1。

(103)若 fl 是指向某文件的指针，且已读到此文件的末尾，则函数 feof(fl) 的返回值是_____。

A)EOF B)0 C)非零值 D)NULL

(103)C

知识点：文件的打开与关闭

评析：本题的考查点是文件结束符的返回值问题。

函数 feof 是用来判断文件是否已读到末尾，如果已读到末尾则返回非零值，否则返回 0。

(104)若有以下定义：int t[3][2]；能正确表示 t 数组元素地址的表达式是_____。

A)&t[3][2] B)t[3] C)t[1] D)*t[2]

(104)C

知识点：指针数组，指向指针的指针

评析：本题的考查点是数组元素地址的表示问题。

选项 A 和 B 两个表达式都越界了；

选项 D 中，*t[2]是目标变量，即一个整型值，而不是地址值。

(105)以下对结构体类型变量的定义中，不正确的是_____。

A)typedef struct aa

{

int n:

float m:

}aa;

aa tdl;

B)#define aa struct aa

aa{

int n:

float m:

}tdl;

C)struct

{

int n:

float m:

}aa;

struct aa tdl:

D)struct

{

int n:

float m:

}tdl;

(105)C

知识点：结构体和共用体类型数据的定义方法和引用方法

评析：本题的考查点是结构体类型变量的定义。

在选项 C 中，aa 是一个结构体变量，而不是结构体名，所以 struct aa tdl： 是非法的。

(106)算法一般都可以用_____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

(106)D

知识点：算法的基本概念

评析：算法的控制结构给出了算法的基本框架，它不仅决定了算法中各操作的执行顺序，而且也直接反映了算法的设计是否符合结构化原则。一个算法一般都可以用顺序、选择、循环三种基本控制结构组合而成。

(107)数据的存储结构是指_____。

- A)数据所占的存储空间量
- B)数据的逻辑结构在计算机中的表示
- C)数据在计算机中的顺序存储方式
- D)存储在外存中的数据

(107)B

知识点：数据的存储结构

评析：数据的逻辑结构在计算机存储空间中的存放形式称为数据的存储结构。

(108)设有下列二叉树：

对此二叉树中序遍历的结果为_____。

- A)ABCDEF B)DBEAFC C)ABDECF D)DEBFCA

(108)B

知识点：二叉树的中序遍历

评析：所谓中序遍历是指在访问根结点、遍历左子树与遍历右子树这三者中，首先遍历左子树，然后访问根结点，最后遍历右子树；并且在遍历左、右子树时，仍然先遍历左子树，然后访问根结点，最后遍历右子树。

(109)在面向对象方法中，一个对象请求另一对象为其服务的方式是通过发送_____。

- A)调用语句 B)命令 C)口令 D)消息

(109)D

知识点：面向对象的程序设计方法

评析：面向对象的世界是通过对象与对象间彼此的相互合作来推动的，对象间的这种相互合作需要一个机制协助进行，这样的机制称为消息。消息是一个实例与另一个实例之间传递的信息，它请求对象执行某一处理或回答某一要求的信息，它统一了数据流和控制流。

(110)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

(110)A

知识点：软件测试

评析：确认测试的任务是验证软件的功能和性能及其他特性是否满足了需求规格说明中确定的各种需求，以及软件配置是否完全、正确。

(111)下列工具中为需求分析常用工具的是_____。

- A)PAD B)PFD C)N-S D)DFD

(111)D

知识点：软件的需求分析

评析：常见的需求分析方法有：结构化分析方法和面向对象的分析方法。结构化分析的常用工具有：数据流图(DFD)、数据字典(DD)、判定树和判定表等。

(112)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

(112)C

知识点：软件工程的基本概念

评析：在软件设计过程中，必须遵循软件工程的基本原则：这些原则包括抽象、信息隐蔽、模块化、局部化、确定性、一致性、完备性和可靠性。

(113)索引属于_____。

- A)模式 B)内模式 C)外模式 D)概念模式

(113)B

知识点: 数据库系统的内部结构体系

评析: 内模式 (Internal Schema) 又称物理模式 (Physical Schema), 它给出了数据库物理存储结构与物理存取方法, 如数据存储的文件结构、索引、集簇及 hash 等存取方式与存取路径。

(114) 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

(114) D

知识点: 关系数据模型

评析: 在关系数据库中, 用二维表来表示实体之间联系。

(115) 将 E-R 图转换到关系模式时, 实体与联系都可以表示成_____。

A) 属性 B) 关系 C) 键 D) 域

(115) B

知识点: 实体联系模型及 E-R 图

评析: 关系是由若干个不同的元组所组成, 因此关系可视为元组的集合, 将 E-R 图转换到关系模式时, 实体与联系都可以表示成关系。

(116) 有如下程序

```
main()
{
    char ch[2][5]={ " 6937 " , " 8254 " }, *p[2];
    int i,j, s=0;
    for(i=0; i<2; i++)p[i]: ch[i];
    for(i=0; i<2; i++)
        for(j=0; p[i][j]> '\0'; j+=2)
            s=10*s+(p[i][j]- '0' );
    printf( " %d\n " , s);
}
```

该程序的输出结果是_____。

A) 69825 B) 63825 C) 6385 D) 693825

(116) C

知识点: 指针数组

评析: 本题 “ch[2][5]={ “6937”, “8254” }” 用于 2 行 5 列的二维数组初始化, ch[0][0]~ch[0][4] 的值分别为: 6, 9, 3, 7, \0; ch[1][0]~ch[1][4] 的值分别为: 8, 2, 5, 4, \0; 第一个 for 循环用于将数组指针分别指向二维数组的第一行和第二行; 第二个 for 嵌套循环用于取出 p[0][0], p[0][2], p[1][0], p[1][2] 四个存储单元的数组组成一个四位数进行输出。具体执行过程如下:

i=0:

j=0: s=0+(p[0][0]-'0'); 即 s=6 (“p[0][0]-0” 通过 ASCII 码相减的方法将字符转换成数字)

j=2: s=60+(p[0][2]-'0'): 即 s=63

i=1:

j=0: s=630+(p[1][0]-'0'): 即 s=638

j=2: s=6380+(p[1][2]-'0'): 即 s=6385

(117) 若变量已正确定义并赋值, 下面符合 c 语言的表达式是_____。

A) a: b+1 B) a=b=c+2 C) int 18. 5%3 D) a=a+7=c+b

(117) B

知识点: 表达式类型和求值规则

评析：选项 A 和 c 一望而知其错；对于表达式 $a=a+7=c+b$ (答案 D)，是先执行赋值表达式 $a+7=c+b$ ，再把这个表达式的值赋给 a，显然，表达式 $a+7=c+b$ 非法，因为赋值表达式的左侧不能为常量或表达式。

(118)C 语言运算对象必须是整型的运算符是_____。

A)% B)/ C)= D)<。

(118)A

知识点：运算符及其运算对象

评析：在 c 语言中，模运算(即取余运算%)的运算对象必须是整型数据(常量、变量或表达式)。

(119)若已定义 x 和 y 为 double 类型，则表达式 $x=1, y=-x+3/2$ 的值是_____。

A)1 B)2 C)2. 0 D)2. 5

(119)C

知识点：表达式的运算

评析：这是一个逗号表达式，它的值应为表达式 $y=-x+3/2$ 的值，而前一个表达式已给 x 赋值 1，在没有进行类型转换的情况下， $3/2$ 的值为 1，所以 $x+3/2$ 的值应为 2.0。

(120)若变量 a、i 已正确定义，且 i 已正确赋值，合法的语句是_____。

A)a=1 B)++i; C)a=a++=5; D)a=int(i);

(120)B

知识点：基本语句中的表达式语句

评析：选项 D 显然是错的，而选项 A 没有分号结束，不是 c 语句； $a++=5$ (答案 c)违反了赋值表达式左侧必须为一个变量(不能是常量或表达式)的规定。

(121)有如下程序

```
main()
{ int y=3,x=3,z=1;
printf( " %d%d\n " , (++x, y++), z+2);
}
```

运行该程序的输出结果是_____。

A)3 4 B)4 2 C)4 3 D)3 3

(121)D

知识点：基本语句中的输入输出语句

评析：逗号表达式(++x, y++)的值应该是 y++的值，由于 y++是先引用后自增，所以 y++的值是 3。

(122)能正确表示逻辑关系：“ $a \geq 10$ 或 $a \leq 0$ ”的 c 语言表达式是_____。

A) $a \geq 10$ or $a \leq 0$ B) $a \geq 0 \mid a \leq 10$

C) $a \geq 10 \&\& a \leq 0$ D) $a \geq 10 \parallel a \leq 10$

(122)D

知识点：c 语言表达式

评析：c 语言中的“或”关系用“||”表示，其他相关逻辑运算符“与”用“&&”表示；运算符“非”用“!”来表示。

(123)有如下程序

```
main()
{ int x=1, a=0, b=0;
switch(x){
case 0: b++;
case 1: a++;
```



```

case 2: a++; b++;
}
printf( " a=%d, b=%d\n " , a, b);
}

```

该程序的输出结果是_____。

A)a=2,b=1 B)a=1,b=1 C)a=1,b=0 D)a=2,b=2

(123)A

知识点:用 switch 语句实现多分支选择结构

评析:在这个 switch 语句中,因为 x 的值为 1,所以执行 case 1: 后面的 a++, 这样 a=1。但又由于其下没有 break 语句,所以其后面的语句“a++; b++”也将被执行,这样一来, a=2, b=1。

(124)有如下程序

```

main()
{ float x=2. 0,y;
if(x<0. 0)y=0. 0;
else if(x<10. 0)y=1. 0 / x;
else y=1. 0;
printf( " %f\n " , y);
}

```

该程序输出结果是_____。

A)0. 000000 B)0. 250000 C)0. 500000 D)1. 000000

(124)C

知识点:用 if 语句实现选择结构

评析:本题考查的是 if...else 语句的使用。x=2.0, 符合第二个 IF 语句的条件 x<10. 0, 所以执行 y=1.0 / x 语句, 即 y=1. 0 / 2. 0=0. 500000。

(125)有如下程序

```

main()
{ int i,sum;
for(i=1; i<=3; sum++)sum+=i;
printf( " %d\n " ,sum);
}

```

该程序的执行结果是_____。

A)6 B)3 C)死循环 D)0

(125)C

知识点: for 循环结构

评析: 在循环过程中, i 的值一直没有被改变, 所以这是一个死循环。

(126)有如下程序

```

main()
{ int x=3;
do
{ printf( " %d " , x--); }
while(!x);
}

```

该程序的执行结果是_____。

A)321 B)3 C)不输出任何内容 D)陷入死循环

(126)B

知识点: do-while 循环结构

评析: 本题 x 赋初值为 3, 当执行 `print(“%d”,x--);` 时, 由于 `x--` 的作用是先运算 x, 再将 x 的值减 1, 所以, `printf` 输出的 x 值为 3, 等输出后, x 的值减 1 变为 2, 继续执行, 判断条件!x 为假, 循环只被执行一次。

(127)设有说明语句: `char a= '\ 72';` 则变量 a _____。

A)包含 1 个字符 B)包含 2 个字符

C)包含 3 个字符 D)说明不合法

(127)A

知识点: c 语言运算符

评析: 转义字符常量 `'\xx'` 可以把 `'\'` 后面的数字转换为对应的 ASCII 字符。

(128)有如下函数调用语句

`func(rec1,rec2+rec3,(rec4,rec5));`

该函数调用语句中, 含有的实参个数是_____。

A)3 B)4 C)5 D)有语法错

(128)A

知识点: 函数的正确调用

评析: `(rec4,rec5)` 是逗号表达式, 它的值是 `rec5` 的值。所以该函数调用语句含有的实参个数是 3。

(129)有如下程序

```
int func(int a, intb)
```

```
{ return(a+b); }
```

```
main()
```

```
{int x=2, y=5, z=8, r;
```

```
r=func(func(x,y), z);
```

```
printf( " %d \n " , r);
```

```
}
```

该程序的输出结果是_____。

A)12 B)13 C)14 D)15

(129)D

知识点: 函数参数的传递

评析: 调用函数 `func(x,y)`, 返回 `x+y`, 即 7; 再调用函数 `func(7, z)`, 返回 `7+z`, 即 15。

(130)有如下程序段

```
int *p,a=10,b--1;
```

```
p=&a; a=*p+b;
```

执行该程序段后, a 的值为_____。

A)12 B)11 C)10 D)编译出错

(130)B

知识点: 指针变量的概念

评析: 执行 `p=&a;` 后, p 指向整数 a, *p 即 p 的目标变量, 也就是 a; 所以 `a=*p+b` 等价于 `a=a+b`, 可知 a 的值最终为 11。

(131)有如下程序

```
long fib(int n)
```

```
{ if(n>2)return(fib(n-1)+fib(n-2));
else return(2);
}
```

```
main()
```

```
{ printf( " %ld\n ",fib(3)); }
```

该程序的输出结果是_____。

A)2 B)4 C)6 D)8

(131)B

知识点：函数的递归调用

评析：函数 fib 内部有两个递归调用，当 n=3 时，fib(n-1)返回 2，fib(n-2)也返回 2，所以 fib(n-1)+fib(n-2)的值为 4。

(132)在 c 语言中，函数中变量的隐含存储类别是_____。

A)auto B)static C)extern D)无存储类别

(132)A

知识点：函数中变量的存储类别

评析：函数中变量的存储类别包括 auto，static，extern，其中 auto 为隐含存储类别，static 为静态存储类别，register 是寄存器存储类别。

(133)有如下程序

```
#define n 2
```

```
#define m N+1
```

```
#define NUM 2*m+1
```

```
main()
```

```
{ int i;
```

```
for(i=1; i<=NUM; i++)printf( " %d\n ",i);
```

```
}
```

该程序中的 for 循环执行的次数是_____。

A)5 B)6 C)7 D)8

(133)B

知识点：宏定义的应用

评析：在 c 语言中，宏定义在编译时将被直接替换。所以 NUM 最后会被替换成 2*N+1+1，即 2*2+1+1，值为 6。因此，for 循环执行的次数为 6。

(134)有以下函数

```
char *fun(char*p)
```

```
{return p; }
```

该函数的返回值是_____。

A)无确切的值 B)形参 p 中存放的地址值

C)一个临时存储单元的地址 D)形参 p 自身的地址值

(134)B

知识点：返回指针值的指针函数

评析：p 本身就是一个字符型指针变量，返回 p 也就是返回变量 p 中存放的地址值。

(135)有如下程序段

```
int a=14,b=15,x;
```

```
char c= 'A';
```

```
x=(a&& b) &&(c< 'B' );
```

执行该程序段后，x 的值为_____。

A)true B>false C)0 D)1

(135)D

知识点:逻辑表达式的运算

评析: $a \& \& b$ 的值为 true, $c < 'B'$ 的值也为 true, 所以 $(a \& \& b) \& \& (c < 'B')$ 的值为 true。但 x 是 int 型变量, 所以 $(a \& \& b) \& \& (c < 'B')$ 的值最后要从 bool 型转换为 int 型赋给 x。这样 x 的值应为 1。

(136)若 fp 是指向某文件的指针, 且已读到此文件末尾, 则库函数 feof(fp) 的返回值是_____。

A)EOF B)0 C)非零值 D)NULL

(136)C

知识点: 库函数的返回值

评析: 函数 feof 是用来判断文件是否已读到末尾, 如果已读到末尾则返回非零值, 否则返回 0。

(137)有如下说明

`int a[10]={1,2,3,4,5,6,7,8,9,10}, *p=a;`

则数值为 9 的表达式是_____。

A)*p+9 B)*(p+8) C)*p+=9 D)p+8

(137)B

知识点: 指针变量的概念

评析: p 是指针, p+8 是地址值, D 是错的; *p 是 p 的目标变量, 即 a[0], 值为 1, 所以 *p+9 的值是 10, 而 *p+=9 的值是重新给 a[0]赋了值(1+9), 所以 A 和 C 也是错的。p+8 是地址值, 指向 a[8], 那么。(p+8)的值就是 a[8]的值, 即 9。

(138)若有以下定义: `int t[3][2]`, 能正确表示 t 数组元素地址的表达式是_____。

A)&t[3][2] B)t[3] C)t[i] D)*t[2]

(138)C

知识点: 数组的概念

评析: A 和 B 两个表达式都已越界了; *t[2]是目标变量, 即一个整数, 而不是地址值。

(139)有如下程序

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

该程序的输出结果是_____。

A)18 B)19 C)20 D)21

(139)A

知识点: 循环的嵌套使用

评析: 题中的外循环只执行了 2 次:

第 1 次: $a[1][0]=3$, $a[1][1]=4$, 所以 $s=7$;

第 2 次: $a[2][0]=5$, $a[2][1]=6$, $a[2][2]=0$, 所以 $s=7+5+6+0=18$ 。

(140)若已建立如下图所示的单向链表结构:

在该链表结构中，指针 p、s 分别指向图中所示结点，则不能将 s 所指的结点插入到链表末尾仍构成单向链表的语句组是_____。

- A) $p=p->next$; $s->next=p$; $p->next=s$;
 B) $p=p->next$; $s->next=p->next$; $p->next=s$;
 C) $s->next=NULL$; $p=p->next$; $p->next=s$;
 D) $p=(*p).next$; $(*s).next=(*p).next$; $(*p).next=s$;

(140)A

知识点：单向链表的建立

评析：在答案 A 中： $p=p->next$; $s->next=p$; $p->next=s$; s 的确已插到了链表的末尾，但它的 next 却并没有为 NULL，而是指向了它的直接前趋 p，这样它就小是一个单向链表(单向链表最后一个结点的 next 指针一定是一个 NULL)。

(141)在下列选项中，_____不是一个算法一般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

(141)C

知识点：算法的基本特征

评析：作为一个算法，一般应具有以下几个基本特征。

- ①可行性
 ②确定性
 ③有穷性
 ④拥有足够的情报

(142)希尔排序法属于_____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)建堆排序法

(142)B

知识点:希尔排序算法

评析:希尔排序法的基本思想是:将整个无序序列分割成若干小的子序列分别进行插入排序。所以希尔排序法属于插入类排序，但它对简单插入排序做了很大的改进。

(143)下列关于队列的叙述中正确的是_____。

- A)在队列中只能插入数据 B)在队列中只能删除数据
 C)队列是先进先出的线性表 D)队列是先进后出的线性表

(143)C

知识点:队列的基本概念

评析:队列是指允许在一端进行插入、而在另一端进行删除的线性表。它又称为“先进先出”或“后进后出”的线性表，体现了“先来先服务”的原则。

(144)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) $N+1$ B) N C) $(N+1)/2$ D) $N/2$

(144)B

知识点：顺序查找排序算法

评析:在进行顺序查找过程中，如果被查的元素是线性表中的最后一个，或者被查元素根本不在线性表中，则为了查找这个元素需要与线性表中所有元素进行比较，这是顺序查找最坏的情况。

(145)信息隐蔽的概念与下述_____概念直接相关。

- A)软件结构定义 B)模块独立性

C)模块类型划分 D)模拟耦合度

(145)B

知识点: 软件工程中的基本概念

评析: 信息隐蔽是指在一个模块内包含的信息(过程或数据), 对于不需要这些信息的其他模块来说是不能访问的。

模块独立性是指每个模块只完成系统要求的独立的子功能, 并且与其他模块的联系最少且接口简单。

衡量软件的模块独立性的度量标准是耦合性和内聚性。一个模块的耦合性越强, 则该模块的模块独立性越强。而内聚性是信息隐蔽和局部化概念的自然扩展。

(146)面向对象的设计方法与传统的面向过程的方法有本质不同, 它的基本原理是_____。

- A)模拟现实世界中不同事物之间的联系
- B)强调模拟现实世界中的算法而不强调概念
- C)使用现实世界的概念抽象地思考问题从而自然地解决问题
- D)鼓励开发者在软件开发的绝大部分中都用实际领域的概念去思考

(146)C

知识点: 面向对象程序设计的概念

评析: 面向对象的设计方法与传统的面向过程的方法有本质不同。它的基本原理是, 使用现实世界的概念抽象地思考问题从而自然地解决问题。它强调模拟现实世界中的概念而不强调算法, 它鼓励开发者在软件开发的绝大部分中都用应用领域的概念去思考。

(147)在结构化方法中, 软件功能分解属于软件开发中的_____阶段。

- A)详细设计 B)需求分析 C)总体设计 D)编程调试

(147)C

知识点: 软件设计的过程

评析: 总体设计过程通常由两个主要阶段组成: 系统设计, 确定系统的具体实现方案; 结构设计-确定软件结构。为确定软件结构, 首先需要从实现角度把复杂的功能进一步分解。分析员结合算法描述仔细分析数据流图中的每个处理, 如果一个处理的功能过分复杂, 必须把它的功能适当地分解成一系列比较简单的功能。

(148)软件调试的目的是_____。

- A)发现错误 B)改正错误
- C)改善软件的性能 D)挖掘软件的潜能

(148)B

知识点: 软件调试的概念及其要点

评析: 由程序调试的概念可知: 程序调试活动由两部分组成。其一是根据错误的迹象确定程序中错误的确切性质、原凶和位置。其二, 对程序进行修改, 排除这个错误。所以程序调试的目的就是诊断和改正程序中的错误。

(149)按条件 f 对关系 R 进行选择, 其关系代数表达式为_____。

$\sigma_{f(R)}$

- A) $\sigma_{f(R)}$ B) $f \cap (R)$ C) $\sigma_{f(R)}$ D) $\sigma_{f(R)}$

(149)C

知识点: 关系运算的基本概念

评析: 选择运算是一个一元运算, 关系 R 通过选择运算(并由该运算给出所选择的逻辑条件)后仍为一个关系。这个关系是由 R 中那些满足逻辑条件的元组所组成。如果关系的逻辑条件为 C 则 R 满足 f 的选择运算可以写成: $\sigma_{f(R)}$ 。

(150)数据库的物理设计是为一个给定的逻辑结构选取一个适合应用环境的_____的过程, 包

括确定数据库在物理设备上的存储结构和存取方法。

A)逻辑结构 B)物理结构 C)概念结构 D)层次结构

(150)B

知识点：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

评析：根据数据库物理设计的概念可知：为一个给定的逻辑数据模型选取一个最合适应用要求的物理结构的过程，就是数据库的物理设计。数据库的物理结构就是指数据库在物理设备上的存储结构与存取方法。

(151)下列叙述中正确的是_____。

A)C 语言编译时不检查语法

B)C 语言的子程序有过程和函数两种

C)C 语言的函数可以嵌套定义

D)C 语言中，根据函数能否被其他源文件调用，被区分为内部函数和外部函数

(151)D

知识点：c 语言基本概念

评析：选项 A 的错误在于编译过程中是检查语法的，若发现源程序有语法错误，则系统会提示出错信息；选项 B 的错误在于 C 语言中，子程序的作用是调用函数来完成的。无过程的概念；选项 C 的错误在于函数不可以嵌套定义，但可以嵌套调用。

(152)以下变量 x、y、z 均为 double 类型且已正确赋值，不能正确表示数学式子 $x / (y * z)$ 的 C 语言表达式是_____。

A) $x / y * z$ B) $x * (1 / (y * z))$ C) $x / y * 1 / z$ D) $x / y / z$

(152)A

知识点：运算符的优先级

评析：按照自左向右的运算逻辑，选项 A 是先做 x / y ，然后再乘以 x，显然与题意不符。

(153)若 a 为 int 类型，且其值为 3，则执行完表达式 $a += a -- = a * a$ 后，a 的值是_____。

A)-3 B)9 C)-12 D)6

(153)C

知识点：运算符的优先级

评析：c 语言中对赋值运算的运算顺序是自右向左。本题中，计算 $a * a$ 之后，a 的值依然为 3，表达式 $a * a$ 的值为 9； $a = 9$ 。 $a = 3 - 9 = -6$ ； $a += -6$ 。 $a = -6 + -6 = -12$ 。

(154)设 x、y、t 均为 int 型变量，则执行语句： $x = y = 3$ ； $t = ++x \parallel ++y$ ；后，y 的值为_____。

A)不定值 B)4 C)3 D)1

(154)C

知识点：逻辑表达式的运算

评析：C 语言中在做逻辑或运算时，自左向右判断各个参加或运算的表达式，一旦运算到某式的值为非零时，表明整个表达式一定为真，余下的语句则不再进行运算。本题中， $++x$ 的值为真，则 $++y$ 并没有进行运算，所以 y 值不变。

(155)若执行以下程序时从键盘上输入 9，则输出结果是_____。

```
main()
{
    int n;
    scanf(" %d: ", &n);
    if(n++<10)printf(" %d\n", n);
    else printf(" %d\n", n--);
}
```

A)11 B)10 C)9 D)8

(155)B

知识点：复合语句和 if 语句

评析：n++是在语句结束后再加 1，因此，if 语句中相当于判断的是 9<10，当然是执行 if 语句后面的 printf 语句，而这时，因为 if 语句被执行。n 的值也被加 1，所以在打印时，n 值为 10，而 n--是存 printf 语句执行完毕后再减 1 的。

(156)有以下程序段

```
int k=0;
```

```
while(k=1)k++;
```

while 循环执行的次数是_____。

A)无限次 B)有语法错，不能执行

C)一次也不执行 D)执行 1 次

(156)A

知识点：while 循环语句

评析：乍一看，还以为程序要表达的意思是当 k 等于 1 的时候做 k++，但是 c 语言里关系表达式的等于应该是“==”。一个等于号表示赋值，即这里面重复地把 1 赋给 k，自然表达式 k=1 的值总为 1，while 后面的表达式恒为真，当然会进入死循环。

(157)以下程序执行后 sum 的值是_____。

```
main()
```

```
{
```

```
int i, sum;
```

```
for(i=1; i<6; i++)sum+=i;
```

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

```
}
```

A)15 B)14 C)不确定 D)0

(157)C

知识点：for 循环语句

评析：变量 sum 在使用时未赋初值，所以无法确定 sum 最终的值。

(158)有以下程序段

```
int x=3;
```

```
do
```

```
{
```

```
printf( " %d ",x-=2);
```

```
}
```

```
while(!--x);
```

其输出结果是_____。

A)1 B)3 C)0 D)死循环

(158)C

知识点：do-while 循环语句

评析：本题中循环执行了两次，第一次，执行完 printf 语句后，x=1，而--x 是先自减后执行语句的，所

以(!--x)的值为非零，所以又执行了一次。第二次 x 的值为-2，--x 的值为-3，求非后，值为 0，跳出循环。

(159)若变量 c 为 char 类型，能正确判断出 c 为小写字母的表达式是_____。

A) 'a' <=c<= 'z' B)(c>= 'a') || (c<= 'z')
 C)('a'<=c)and('z'>=c) D)(c>='a')&&(C<='z')

(159)D

知识点：变量的灵活应用

评析：(c>= 'a')和(c<= 'z')的关系应该是“与”的关系。

(160)以下所列的各函数首部中，正确的是_____。

A)void play(vat a: Integer,var b: Integer)

B)void play(int a,b)

C)void play(int a,int b)

D)Sub play(a as integer,b as integer)

(160)C

知识点：函数的定义

评析：除选项 c 项外，其他各项都没有按照 c 语言中的函数定义规则定义函数。

(161)以下程序的输出结果是_____。

```
fun(int x,int y,int z)
```

```
(Z=x*x+y*y; )
```

```
main()
```

```
{
```

```
int a=31;
```

```
fun(5, 2,a);
```

```
printf( " %d " , a);
```

```
}
```

A)0 B)29 C)31 D)无定值

(161)C

知识点：函数的调用和参数的传递

评析：本题中，a 的值并没有改变，因为传递到函数中的是变量值。

(162)下列程序段的输出结果是_____。

```
void fun(int *x,int *y)
```

```
{printf( " %d%d " , *x, *y); *x=3; *y=4; }
```

```
main()
```

```
{
```

```
int x=1, y=2;
```

```
fun(&y, &x);
```

```
printf( " %d%d " , x, y);
```

```
}
```

A)2 1 4 3 B)1 2 1 2 C)1 2 3 4 D)2 1 1 2

(162)A

知识点：函数的参数传递

评析：本题中函数传递的是变量的地址，如果形参在函数中发生改变，相对应的实参也就有了改变。

(163)下列程序的输出结果是_____。

```
main()
```

```
{
```

```
char a[10]={9,8,7,6,5,4,3,2,1, 0}, *p=a+5;
```

```
printf( " %d " , *--p);
}
```

A)非法 B)a[4]的地址 C)5 D)3

(163)C

知识点：指针和数组

评析：本题中 `char*p=a+5`；即相当于 `p=&a[5]`，而 `--p`，是先将 `p` 减 1，则 `printf` 语句要输出的值就是 `a[4]`。

(164)下列程序的运行结果是_____。

```
void fun(int *a,int *b)
{
    int*k;
    k=a; a=b; b=k;
}
main()
{
    int a=3, b=6, *x=&a, *y=&b;
    fun(x, y);
    printf( " %d%d " , a, b);
}
```

A)6 3 B)3 6 C)编译出错 D)0 0

(164)B

知识点：函数的调用和变量的作用域

评析：本题中主函数里的 `x`、`y`，`fun` 函数里的 `a`、`b`、`k`，这些都是指针，`fun` 函数中只是将 `a`、`b` 这两个指针交换了位置，而并没有改变主函数中变量 `a`、`b` 的值。

(165)若有定义：`int*p[3]`；，则以下叙述中正确的是_____。

A)定义了一个基类型为 `int` 的指针变量 `p`，该变量有三个指针

B)定义了一个指针数组 `p`，该数组含有三个元素，每个元素都是基类型为 `int` 的指针

C)定义了一个名为 `tp` 的整型数组，该数组含有三个整型元素

D)定义了一个可指向一维数组的指针变量 `p`，所指一维数组应具有三个 `int` 类型元素

(165)B

知识点：数组的定义

评析：由于运算符 `[]` 优先级比 `*` 高，`int+p[3]`；相当于 `int*(p[3])`；表示数组 `p` 的三个元素都是指针变量，且每个元素都是基类型为 `int` 的指针。

(166)当调用函数时，实参是一个数组名，则向函数传送的是_____。

A)数组的长度 B)数组的首地址

C)数组每一个元素的地址 D)数组每个元素中的值

(166)B

知识点：指针和数组的关系

评析：当调用函数时，实参是一个数组名，则向函数传送的是数组的首地址，函数中的形参可定义成以下 3 种形式：

①形参定义成数组：

②形参定义成可变长数组：

③形参定义为指针变量。

(167)设有以下说明语句

struct ex

```
{ int x; float y; char z; }example;
```

则下面的叙述中不正确的是_____。

A)struct 是结构体类型的关键字 B)example 是结构体类型名

C)x, v, z 都是结构体成员名 D)struct ex 是结构体类型

(167)B

知识点：结构体的基本概念

评析：example 是结构体变量名。

(168)以下只有在使用时才为该类型变量分配内存的存储类说明是_____。

A)auto 和 static B)auto 和 register

C)register 和 static D)extern 和 register

(168)B

知识点：变量的存储类型

评析：extern、register、static、auto 分别是定义外部变量、静态变量、自动变量，其中，自动变量和寄存器变量属于动态存储，调用时分配单元；而静态变量和外部变量属于静态存储，在整个程序运行时都存在。

(169)以下程序的输出结果是_____。

```
main()
```

```
{
```

```
int i, k, a[10], p[3];
```

```
k=5;
```

```
for(i=0; i<10; i++)a[i]=i;
```

```
for(i=0; i<3; i++)p[i]=a[i*(i+1)];
```

```
for(i=0; i<3; i++)k+=p[i]*2;
```

```
printf( " %d\n " , k);
```

```
}
```

A)20 B)21 C)22 D)23

(169)B

知识点：循环语句和数组

评析：按照程序的流程走一遍，可以得到，p[0]=0，p[1]=2，p[3]=6；执行三次 k+=p[i]*2，相当于 k=5+0*2+2*2+6*2=21。

(170)当执行下面的程序时，如果输入 ABC，则输出结果是_____。

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

```
#include "string. h "
```

```
main()
```

```
{ char ss[10]= " 12345 " ;
```

```
gets(ss); strcat(ss, " 6789 " );
```

```
printf( " %s\n " , ss);
```

```
}
```

A)ABC6789 B)ABC67 C)12345ABC6 D)ABC456789

(170)C

知识点：c 语言字符串函数。

评析：strcat(str1,str2)合并字符串函数的作用是把 str2 所指字符串的内容连接到 str1 字符串的后面，自动删去 str1 原来串中的'\0'。为了进行这项操作，要求 str1 所指的字符串后向

有足够的空间来容纳 str2 所指字符串中的内容。函数值为 str1 所指第一个字符的地址。

```
(171)main(int argc, char*argv[])
{
while(--argc>0)printf( " %s " , argv[argc]);
printf( " \n " );
}
```

假定以上程序经编译和连接后生成可执行文件 PROG. EXE, 如果在此可执行文件所在目录的 DOS 提示符下键入: PROG ABCDEFGHIJKL / , 则输出结果为_____。

A)ABCDEFG B)IJHL

C)ABCDEFGHIJKL D)IJKLABCDEFGH

(171) C

知识点: 指针数组作函数参数

评析: 本题主函数带有两个参数, 一个 int 型变量 argc, 另一个是 char 型的一维一级指针数组 argv。当主函数被执行时, 系统自动将根据命令行的情况, 分别给主函数的两个参数赋值。argc 中存放命令行中命令字和参数的总和的个数, argv 用来存放命令行中命令字和所有参数的字符串, 并规定 argv[0]存放命令字符串, argv[1]用来存放第一个参数的字符串, argv[2]用来存放第 2 个参数的字符串, 依次类推。题中 argc 的值是 2, 执行完 --argc 后为 1, 输出 argv[1], 此数组存在第一个参数的字符串 ABCDEFGHIJKL。

(172)以下程序的输出结果是_____。

```
long fun(int n)
{ long s;
if(n==1 || n==2)s=2;
else s=n-fun(n-1);
return S;
}
main()
{printf( " %ld \n " ,fun(3)); }
```

A)1 B)2 C)3 D)4

(172) A

知识点: 递归函数

评析: 这是一个递归函数。递归的结束条件是, n=1 或者 n=2。按照程序顺序, 即可得出本题结果为 1。

(173)以下程序的输出结果是_____。

```
#define SQR(X) X*X
main()
{ int a=16, k=2, m=1;
a /=SQR(k+m) / SQR(k+m);
printf( " %d\n " ,a);
}
```

A)16 B)2 C)9 D)1

(173) B

知识点: 宏定义

评析: C 语言在预编译时遇到带实参的宏名, 则按命令行中指定的字符串从左到右进行置换。在做这道题时, 也不妨运用置换法。得到: a /=k+m*k+m / k+m*k+m=16 / 7=2。注: a 为

整型，所以在做除法时，自动取整。

(174)假定建立了以下链表结构，指针 p、q 分别指向如图所示的结点，则以下可以将 q 所指结点从链表中删除并释放该结点的语句组是_____。

- A) free(q); p->next=-q->next;
- B) (*p). next=(*q). next; free(q);
- C) q=(*q). next; (*p). next=q; free(q);
- D) q=q->next; p->next=q; p=p->next; free(p);

(174) B

知识点：链表的应用

评析：选项 A 的错误在于，先已经把 q 结点给释放，无法再进行余下语句了；选项 C 和 D 的错误在于因为有了 q=(*q).next 做了赋值，则 free(q)释放了原来 q 后面的一个结点。

(175)以下程序的输出结果是_____。

```
main()
{ int x=05; char z= 'a';
  printf( " %d\n " , (x&1)&&(z<'Z' )); }
```

A)0 B)1 C)2 D)3

(175) B

知识点：综合应用

评析：“&”是位运算符表示按位与运算。“&&”是逻辑运算符，当其两边的表达式都为真时返真值。关系表达式(x&1)&&(z<'z')，(x&1)为真，(z<'z')也为真，两者相与，自然也为真，以整型输出，则输出1。

(176)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

(176) C

知识点：算法的基本概念

评析：计算机算法是指解题方案的准确而完整的描述，它有以下几个基本特征：可行性、确定性、有穷性和拥有足够的情报。

(177)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

(177) C

知识点：线性表的概念

评析：栈和队列都是一种特殊的操作受限的线性表，只允许在端点处进行插入和删除。二者的区别是：栈只允许在表的一端进行插入或删除操作，是一种“后进先出”的线性表；而队列只允许在表的一端进行插入操作，在另一端进行删除操作，是一种“先进先出”的线性表。

(178)已知二叉树 BT 的后序遍历序列是 dabec，中序遍历序列是 debac，它的前序遍历序列是_____。

- A)cedba B)acbed C)decab D)deabc

(178) A

知识点：树的遍历

评析：二叉树 BT 的后序遍历序列为 dabec，故 BT 的根结点为 c(后序遍历序列的最后一个结点为树的根结点)；而 BT 的中序遍历序列是 debac，即遍历序列中最后一个结点为根结点，

说明 BT 的右子树为空。由 BT 的后序遍历序列和中序遍历序列可知 BT 的左子树(LST)的后序遍历序列和中序遍历序列分别为 dabe 和 deba(树是递归定义的): 故 LST 的根结点是 e。再由 LST 的中序遍历序列可知其左子树为 d。因此 BT 的前序遍历序列为 cedba。

(179)在下列几种排序方法中, 要求内存量最大的是_____。

A)插入排序 B)选择排序 C)快速排序 D)归并排序

(179)D

知识点: 各种排序方法的实现过程及实现机制

评析: 快速排序的基本思想是, 通过一趟排序将待排序记录分割成独立的两部分, 其中一部分记录的关键字均比另一部分记录的关键字小, 再分别对这两部分记录继续进行排序, 以达到整个序列有序; 插入排序的基本操作是指将无序序列中的各元素依次插入到已经有序的线性表中, 从而得到一个新的序列; 选择排序的基本思想是: 扫描整个线性表, 从中选出最小的元素, 将它交换到表的最前面(这是它应有的位置), 然后对剩下的子表采用同样的方法, 直到表空为止; 归并排序是将两个或两个以上的有序表组合成一个新的有序表。

(180)在设计程序时, 应采纳的原则之一是_____。

A)程序结构应有助于读者理解 B)不限制 goto 语句的使用
C)减少或取消注解行 D)程序越短越好

(180)A

知识点: 结构化程序设计的原则

评析: 滥用 goto 语句将使程序流程无规律, 可读性差; 添加的注解行有利于对程序的理解。不应减少或取消; 程序的长短要依照实际需要而定, 并不是越短越好。

(181)下列不属于软件调试技术的是_____。

A)强行排错法 B)集成测试法 C)回溯法 D)原因排除法

(181)B

知识点: 软件调试的各种方法及各种方法的区别

评析: 调试的关键在于推断程序内部的错误位置及原因。主要的调试方法有强行排错法、回溯法和原因排除法。

(182)下列叙述中, 不属于软件需求规格说明书的作用的是_____。

A)便于用户、开发人员进行理解和交流
B)反映出用户问题的结构, 可以作为软件开发工作的基础和依据
C)作为确认测试和验收的依据
D)便于开发人员进行需求分析

(182)D

知识点: 软件需求规格说明书的作用

评析: 软件需求规格说明书(SRS, Software Requirement Specification)是需求分析阶段的最后成果, 是软件开发中的重要文档之一。它有以下几个方面的作用: ①便于用户、开发人员进行理解和交流; ②反映出用户问题的结构, 可以作为软件开发工作的基础和依据; ③作为确认测试和验收的依据。

(183)在数据流图(DFD)中, 带有名字的箭头表示_____。

A)控制程序的执行顺序 B)模块之间的调用关系
C)数据的流向 D)程序的组成成分

(183)C

知识点: 软件设计工具

评析: 数据流相当于一条管道, 并有一级数据(信息)流经它。在数据流图中, 用标有名字的箭头表示数据流。数据流可以从加工流向加工, 也可以从加工流向文件或从文件流向加工,

并且可以从外部实体流向系统或从系统流向外部实体。

(184) SQL 语言又称为_____。

- A)结构化定义语言 B)结构化控制语言
C)结构化查询语言 D)结构化操纵语言

(184)C

知识点：结构化查询语言的定义

评析：结构化查询语言(SQL, Structured Query Language)是集数据定义、数据操纵和数据控制功能于一体的数据库语言。

(185)视图设计一般有 3 种设计次序，下列不属于视图设计的是_____。

- A)自顶向下 B)由外向内 C)由内向外 D)自底向上

(185)B

知识点：数据库概念设计中视图设计的基本概念

评析：视图设计一般有 3 种设计次序，它们分别是自顶向下、自底向上和由内向外，它们又为视图设计提供了具体的操作方法，设计者可根据实际情况灵活掌握，可以单独使用也可混合使用。

(186)以下有 4 组用户标识符，其中合法的一组是_____。

- A)FOR B)4d C)f2_G3 D)WORD

-sub DO IF void

Case Size abc define

(186)C

知识点：c 语言标识符

评析：c 语言的标识符的定义为：以字母或下划线开头的由字母、数字字符、下划线组成的字符串，而且标识符不能与关键字相同。

(187)下列叙述中正确的是_____。

- A)C 程序中注释部分可以出现在程序中任何合适的地方
B)花括号“{”和“}”只能作为函数体的定界符
C)构成 C 程序的基本单位是函数，所有函数名都可以由用户命名
D)分号是 C 语句之间的分隔符，不是语句的一部分

(187)A

知识点：c 语言程序的构成和格式

评析：c 程序中注释部分用“/*”和“*/”括起来，可以出现在程序中任何合适的地方；花括号“{”和“}”不仅可作为函数体的定界符，也可作为复合语句的定界符；构成 c 程序的基本单位是函数，分为两类：系统提供的标准函数和用户自定义函数，只有用户自定义函数名可以由用户命名：一个语句必须在最后出现分号，分号是 c 语句中不可缺少的部分。

(188)以下合法的赋值语句是_____。

- A)X=Y=100 B)D--; C)X+Y; D)C=int(A+B)

(188)B

知识点：c 语言的赋值语句

评析：本题中的答案 A 与 D 都缺少“;”，而答案 c 中，表达式是不能独立成为语句的，答案 B 使用了 C 语言的自减运算符就相当于 D=D-1，所以答案 B 为赋值语句。

(189)设 x,y 均为 . int 型变量，且 x=10, y=3, 则 printf(" %d,%d\n " ,x--,y); 语句的输出结果是_____。

- A)10,3 B)9,3 C)9,2 D)10,2

(189)D

知识点: c 语言的自减运算符

评析: c 语言的自减运算符有前置与后置两种形式。当自减运算符前置时, 变量先自减 1 后再参与表达式运算; 而后置时, 变量先参与表达式运算后再自减 1。

(190)x、Y、z 被定义为 int 型变量, 若从键盘给 x、Y、z 输入数据, 正确的输入语句是_____。

A)INPUT X,Y,Z; B)scanf(" %d%d%d ", &x, &Y, &z);

C)scanf(" %d%d%d ", X, Y, Z); D)read(" %d%d%d ", &X, &Y, &Z);

(190)B

知识点: C 语言的格式化输入语句

评析: c 语言中没有 INPUT 与 read 语句, scanf()语句的格式为:

scan(“格式字符串”, 变量地址 1, 变量地址 2, ...);

本题答案 c 中, 应为变量地址。

(191)以下程序的输出结果是_____。

```
main()
```

```
{ int a=3;
```

```
printf(" %d\n ", (a+=a-a*a));
```

```
}
```

A)-6 B)1 C)2 D)-12

(191)D

知识点: c 语言的赋值运算符

评析: 首先 $a+=b$ 相当于 $a=a+b$; $a-=b$ 相当于 $a=a-b$, 且赋值运算符的结合方向是自右至左, 所以表达式 $a+=a-a*a$ 。a 先运算最右边的 $a*a$ 得 9, 再运算 $a=a-9$, 即 $a=3-9$, 所以此时 a 的值由 3 变成了 -6, 最后运算 $a=a+(6)$, 即 $a=(-6)+(-6)-12$ 。

(192)设 a、b、c、d、m、n 均为 int 型变量, 且 $a=5$ 、 $b=6$ 、 $c=7$ 、 $d=8$ 、 $m=2$ 、 $n=2$, 则逻辑表达式 $(m=a>b) \ \&\& \ (n=c>d)$ 运算后, n 的值为_____。

A)0 B)1 C)2 D)3

(192)A

知识点: c 语言的比较运算符与逻辑运算符

评析: c 语言中比较表达式的运算结果为 0 或 1。0 代表不成立, 1 表示成立。

(193)以下程序的输出结果是_____。

```
main()
```

```
{ int num=0;
```

```
while(num<=2)
```

```
{ num++; printf(" %d\n,num"); }
```

```
}
```

A) B) C) D)

1 1 1 1

2 2 2

3 3

4

(193)B

知识点: C 语言的 while 循环语句

评析: c 语言中 while 语句的一般形式如下:

```
while(表达式)
```

```
{
```


语句部分:

```
}
```

该语句的执行过程是: 先计算表达式的值, 若成市(即值为非零)则重新执行语句部分(循环体), 然后再次计算表达式的值, 并重复上述过程, 直到表达式的值为“假”(值为零)时, 退出循环, 并转下一语句去执行。本题在输出 num 的值为 3 之后, 再判断 while 的循环条件 $3 \leq 2$, 不成立, 所以不再输出 Num 的值, 故本题的答案选 B。

(194)以下各选项企图说明一种新的类型名, 其中正确的是_____。

A)typedef vl int; B)typedef v2=int;

C)typedef int v3; D)typedef v4: int

(194)C

知识点: c 语言的类型定义

评析: c 语言中可以使用 typedef 来重新定义已有的数据类型, 相当于为数据类型取个别名。

(195)在调用函数时, 如果实参是简单变量, 它与对应形参之间的数据传递方式是_____。

A)地址传递 B)单向值传递

C)由实参传给形, 再由形参传回实参 D)传递方式由用户指定

(195)B

知识点: c 语言函数的参数的传值方式

评析: C 语言函数中的参数传递方式有传值与传址两种方式, 传值方式是指将实参的值复制一份传递给形参, 而传址方式是指将实参的变量地址传递给形参, 也就是实参的引用。

(196)以下函数值的类型是_____。

```
fun(float x)
```

```
{ float y;
```

```
y=3*x-4;
```

```
return y;
```

```
}
```

A)int B)不确定 C)void D)float

(196)A

知识点: C 语言函数的缺省的函数类型

评析: c 语言中如果函数前不加任何数据类型时, 缺省函数的类型为整型, 函数的类型就是函数返回值的类型。

(197)以下选项中, 非法的字符常量是_____。

A) '\t' B) '\17' C) "\n" D) '\xaa'

(197)C

知识点: c 语言的字符常量与转义字符

评析: c 语言中字符常量足以单引号括起来的单个字符, 或为以“\”与三位八进制数值或两位十六进制数值代替单个字符。

(198)若有说明: int i,j=2, *p=&i; , 则能完成 i=j 赋值功能的语句是_____。

A)i=*p; B)*p=&j; C)i=&j; D)i=**p;

(198)B

知识点: c 语言中指针赋值运算

评析: 指针是一种特殊的变量, 它是用来存放变量地址的变量。本题中指针变量 p 用于存放整型变量 i 的地址, 即指针变量 p 所指向的存储单元为变量 i 所在的存储单元, 改变 *p 的值即为改变 i 的值。又因为“&”和“*”两个运算符的优先级别相同, 按自右而左的方向结合, 所以 *&j 先进行 &j 运算, 得 j 的地址, 再进行 * 运算, 取所得地址里而的值, 故 *&j 与 j

等价。

(199)假定 `int` 类型变量占用两个字节，若有定义：`int x[10]={0, 2, 4};`，则数组 `x` 在内存中所占字节数是_____。

A)3 B)6 C)10 D)20

(199)D

知识点：c 语言的数组的定义

评析：c 语言中可以为数组赋初值，或部分赋初值。当数组定义后，系统就为其分配内存空间，而不论其中有没有内容。因此，本题中数组 `x[10]` 不论是否为其元素初始化，它所分配的存储空间仍为 $2 \times 10 = 20$ 个字节。

(200)以下程序的输出结果是_____。

```
main()
{int i, a[10];
for(i=9>: 0; i--)a[i]=10-i;
printf( " %d%d%d ", a[2], a[5], a[8]);
}
```

A)258 B)741 C)852 D)369

(200)C

知识点：循环语句给数组元素赋值

评析：c 语言中 `for` 循环语句的一般形式为：

`for(表达式 1; 表达式 2; 表达式 3)`

```
{
语句部分;
}
```

在本题运行时主要注意的是当 `i=9` 时，`a[i]=10-9=1`；`i=8` 时，`a[i]=10-8=2`；`i=7` 时，`a[i]=10-7=3`；……依此类推，直到 `i=0` 时，`a[i]=10-0=10`；此时，`i` 的值已变为 -1，判断 `for` 的循环条件，不成立，然后输出 `a[2]`，`a[5]`，`a[8]` 分别为 8，5，2。

(201)以下数组定义中不正确的是_____。

A)`n a[2][3];` B)`int b[][3]。{0,1, 2};`;

C)`int c[100][100]={0};` D)`int d[3][]={{1, 2}, {1, 2, 3}, {1, 2, 3, 4}};`;

(201)D

知识点：c 语言的数组定义与数组的初始化

评析：一维数组的定义方式为：

类型说明符 数组名[常量表达式];

选项 A 符合此定义形式，正确；c 语言中多维数组赋初值时可以部分赋值，也可以不指定除第一维以外的其他维的大小，故选项 c 正确；另外，如果对全部数组元素都赋初值，则定义数组时对第一维的长度可以不指定，但第二维的长度不能省，所以选项 B 正确，而选项 D 是错误的。

(202)以下程序的输出结果是_____。

```
main()
{ int a[4][4]={{1, 3, 5}, {2, 4, 6}, {3, 5, 7}};
printf( " %d%d%d%d\n ", a[0][3], a[1][2], a[2][1], a[3][0]);
}
```

A)0650 B)1470 C)5430 D)输出值不定

(202)A

知识点：c 语言中的二维数组的初始化

评析：对未给出初始值的整数数组元素，被默认初始化为零。

(203)以下程序的输出结果是_____。

```
main()
{ char st[20]= " hello \0\t\t ";
printf( " %d%d\n ",strlen(st), sizeof(st));
}
```

A)99 B)520 C)1320 D)2020

(203)B

知识点：c 语言的字符串与库函数

评析：c 语言中字符串是以 ‘\0’ 字符结束的，且 strlen()函数计算的是 ‘\0’ 字符前的所有字符的个数。本题中 strlen(st)应为 5。数组定义以后系统就为其分配相应大小的内存空间，而不论其中有没有内容。sizeof()函数是计算变量或数组的所分配到的内存空间的大小。所以本题的 sizeof(st)为 20。

(204)以下选项中，不能正确赋值的是_____。

A)char s[10]; s1=" chest " ; B)char s2[]={ ‘C’, ‘t’, ‘e’, ‘s’, ‘t’ } ;
C)char s3[20]= " Chest " ; D)char*s4=" Ctest\n "

(204)A

知识点：c 语言的字符数组和字符串的概念

评析：c 语言中不能将字符串常量直接赋给数组。但在赋初值时可以。

(205)以下程序输出正确的是_____。

```
amovep(n *p,int(*a)[3],int n)
{int i,j;
for(i=0; i<n; i++)
for(j=0; j<n; j++){ *p=a[i][j]; p++; }
}
main()
{int *p, a[3][3]={1, 3, 5}, {2, 4, 6}};
p=(int*)malloc(100);
amovep(p, a, 3);
printf( " %d%d\n ", p[2], p[5]); free(p);
}
```

A)56 B)25 C)34 D)程序错误

(205)A

知识点：指针与数组的操作

评析：本题 main 函数中定义了指引 p 和二维数组 a，通过函数 amovep 将数组的值存入指针 p 所指向的存储单元中，a 的各元素分别为：a[0][0]=1，a[0][1]=3，a[0][2]=5，a[1][0]=2，a[1][1]=4，a[1][2]=6，a[2][0]=0，a[2][1]=0，a[2][2]=0。通过 malloc()函数给指针分配内存空间，free()函数用于释放指针变量所用内存空间。在主函数中通过 amovep(p, a,3)调用函数 amovep，使得实参 p 与形参 p，实参数组 a 与形参中指向数组的指针变量共用同一存储空间。具体运行过程如下：

i=0:

j=0: p[0]=a[0][0]，即 p[0]=1;

j=1: p[1]=a[0][1]，即 p[1]=3;

```

j=2: p[2]=a[0][2], 即 p[2]=5;
i=1:
j=0: p[3]=a[1][0], 即 p[3]=2;
j=1: p[4]=a[1][1], 即 p[4]=4;
j=2: p[5]=a[1][2], 即 p[5]=6;
i=2:

```

```

j=0: p[6]=a[2][0], 即 p[6]=0;
j=1: p[7]=a[2][1], 即 p[7]=0;
j=2: p[8]=a[2][2], 即 p[8]=0;
因此, 输出 p[2], p[5]为 56。

```

(206)以下程序的输出结果是_____。

```

Struct HAR
{int x,y; struct HAR*p; }h[2];
main()
{ h[0]. x=1; h[0]. y=2;
h[1]. x=3; h[1]. y=4;
h[0]. p=&h[1]; h[1]. p=h;
printf( " %d%d\n " , (h[0]. p)->x,(h[1]. p)->y);
}

```

A)12 B)23 C)14 D)32

(206)D

知识点: c 语言中的结构体与链表

评析: 本题中是一个含有两个结点的循环链表。

C 语言中结构体的定义为:

struct 结构体类型名

```

{
成员项表:
};

```

(207)以下程序的输出结果是_____。

```

int a, b;
Void fun()
{ a=100; b=200; }
main()
{ int a=5, b=7;
fun();
printf( " %d%d\n " , a, b);
}

```

A)100200 B)57 C)200100 D)75

(207)B

知识点: c 语言中的变量作用域

评析: 全程变量是可以在子函数中对其值作改变, 且它也可作为函数问的值传递。但当函数或子函数中定义了与全程变量名称相同的局部变量, 则全程变量将被屏蔽。所以本题主函数的局部变量 a, b 把同名的全程变量屏蔽了。

(208)以下程序的输出结果是_____。

```
#define M(x,y,z) x*y+z
main()
{ int a=1, b=2, c=3;
printf( " %d\n ",M(a+b, b+c, c+a));
}
```

A)19 B)17 C)15 D)12

(208)D

知识点: c 语言的宏定义

评析: c 语言的宏定义包括不带参数的宏定义与带参数的宏定义。本题为带参数的宏定义,其形式为: #define 宏名(形参表) 字符串

本题的 M(a+b,b+c,c+a)被字符串 a+b*b+c+c+a 代替。

(209)整型变量 x 和 Y 的值相等, 且为非 0 值, 则以下选项中结果为 0 的表达式是_____。

A)X||Y B)X | Y C)X&Y D)X^Y

(209)D

知识点: c 语言的逻辑运算符与位运算符

评析: “||”为或运算符, 当其左右表达式中只要一个为非零则整个表达式的值 1。|是按位或, &是按位与, ^是按位异或, 这三个位运算符是按值的二进制位来比较的。

(210)下面的程序执行后, 文件 test. t 中内容是_____。

```
#include<stdio. h>
void fun(char*fname,char*st)
{ FILE*myf; int i;
myf=fopen(fname, "w");
for(i=0; i<strlen(st); i++)fputc(st[i], myf);
fclose(myf);
}
main()
{fun( " test. t ", " new world " ); fun( " test. t ", " hello, " ); }
```

A)hello. B)new worldhello C)new world D)hello,rld

(210)A

知识点: c 语言中的文件操作方面的知识

评析: c 语言中文件有文本文件与二进制文件, 对文件的使用前必须先打开, 打开方式有只读、写入、读写等方式。

(211)数据结构中, 与所使用的计算机无关的是数据的_____。

A)存储结构 B)物理结构 c)逻辑结构 D)物理和存储结构

(211)C

知识点: 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

评析: 数据结构概念一般包括 3 个方面的内容, 数据的逻辑结构、存储结构及数据上的运算集合。数据的逻辑结构只抽象地反映数据元素之间的逻辑关系, 而不管它在计算机中的存储表示形式。

(212)栈底至栈顶依次存放元素 A、B、C、D, 在第五个元素 E 入栈前, 栈中元素可以出栈, 则出栈序列可能是_____。

A)ABCED B)DBCEA C)CDABE D)DCBEA

(212)D

知识点：栈和队列的定义：栈和队列的顺序存储结构及其基本运算

评析：栈操作原则上“后进先出”，栈底至栈顶依次存放元素 A、B、C、D，则表明这 4 个元素中 D 是最后进栈，B、C 处于中间，A 最早进栈。所以出栈时一定是先出 D，再出 C，最后出 A。

(213)线性表的顺序存储结构和线性表的链式存储结构分别是_____。

- A)顺序存取的存储结构、随机存取的存储结构
- B)随机存取的存储结构、顺序存取的存储结构
- C)随机存取的存储结构、随机存取的存储结构
- D)任意存取的存储结构、任意存取的存储结构

(213)B

知识点：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

评析：顺序存储结构中，数据元素存放在一组地址连续的存储单元中，每个数据元素地址可通过公式 $LOC(a_i) = LOC(a_1) + (i-1)L$ 计算得到，从而实现了随机存取。对于链式存储结构，要对某结点进行存取，都得从链的头指针指向的结点开始，这是一种顺序存取的存储结构。

(214)在单链表中，增加头结点的目的是_____。

- A)方便运算的实现 B)使单链表至少有一个结点
- C)标识表结点中首结点的位置 D)说明单链表是线性表的链式存储实现

(214)A

知识点：线性单链表、双向链表与循环链表的结构及其基本运算

评析：头结点不仅标识了表中首结点的位置，而且根据单链表(包含头结点)的结构，只要掌握了表头，就能够访问整个链表，因此增加头结点的目的是为了便于运算的实现。

(215)软件设计包括软件的结构、数据接口和过程设计，其中软件的过程设计是指_____。

- A)模块间的关系 B)系统结构部件转换成软件的过程性描述
- C)软件层次结构 D)软件开发过程

(215)B

知识点：结构化分析方法，数据流图，数据字典，软件需求规格说明书

评析：软件设计包括软件结构设计、数据设计、接口设计和过程设计。其中结构设计是定义软件系统各主要部件之间的关系；数据设计是将分析时创建的模型转化为数据结构的定义；接口设计是描述软件内部、软件和操作系统之间及软件与人之间如何通信；过程设计则是把系统结构部件转换成软件的过程性描述。

(216)为了避免流程图在描述程序逻辑时的灵活性，提出了用方框图来代替传统的程序流程图，通常也把这种图称为_____。

- A)PAD 图 B)N-S 图 C)结构图 D)数据流图

(216)B

知识点：结构化分析方法，数据流图，数据字典，软件需求规格说明书

评析：常见的过程设计工具有：程序流程图、N-S 图、PAD 图和 HIPO 图。其中，为了避免流程图在描述程序逻辑时的灵活性，提出了用方框图来代替传统的程序流程图，通常也把这种图称为 N-S 图。

(217)数据处理的最小单位是_____。

- A)数据 B)数据元素 C)数据项 D)数据结构

(217)C

知识点：数据库的基本概念：数据库，数据库管理系统，数据库系统

评析：数据处理的最小单位是数据项；由若干数据项组成数据元素；而数据是指能够被计算

机识别、存储和加工处理的信息载体；数据结构是指数据之间的相互关系和数据运算。

(218)下列有关数据库的描述，正确的是_____。

- A)数据库是一个 DBF 文件 B)数据库是一个关系
C)数据库是一个结构化的数据集合 D)数据库是一组文件

(218)C

知识点：数据库的基本概念：数据库，数据库管理系统，数据库系统

评析：数据库(Database，简称 DB)是数据的集合，它具有统一的结构形式并存放于统一的存储介质内，是多种应用数据的集成，并可被各个应用程序所共享。数据库中的数据具有“集成”、“共享”之特点。

(219)单个用户使用的数据视图的描述称为_____。

- A)外模式 B)概念模式 C)内模式 D)逻辑模式

(219)A

知识点:数据库的基本概念：数据库，数据库管理系统,数据库系统

评析：外模式是用户的数据视图，也就是用户所见到的数据模式；全局数据视图的描述称为概念模式，即数据库中全部数据的整体逻辑结构的描述；物理存储数据视图的描述称为内模式，即数据库在物理存储方面的描述，存储模式即为内模式。

(220)需求分析阶段的任务是确定_____。

- A)软件开发方法 B)软件开发工具
C)软件开发费用 D)软件系统功能

(220)D

知识点：结构化分析方法，数据流图，数据字典，软件需求规格说明书

评析：需求分析是软件定义时期的最后一个阶段，它的基本任务就是详细调查现实世界要处理的对象(组织、部门、企业等)，充分了解原系统的工作概况，明确用户的各种需求，然后在此基础上确定新系统的功能。选项 A)软件开发方法是在总体设计阶段完成的任

(221)若变量 a 是 int 类型，并执行了语句：a= 'A' +1.6;，则正确叙述是_____。

- A)a 的值是字符 C B)a 的值是浮点型
c)不允许字符型和浮点型相加 D)a 的值是字符 " A " 的 ASCII 值加上 1

(221)D

知识点：C 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

评析：在赋值表达式中，赋值符右边的值的类型会自动转换成赋值符左边的变量的类型。在本题中，先用 " A " 的 ASCII 码值加上 1.46 得到 66.46，然后将它强制转换成 int 型，转换后，实数的小数部分全部舍去。

(222)以下程序段的输出结果是_____。

```
int a=1234;
printf( " %2dha " ,a);
```

- A)12 B)34 C)1234 D)提示出错、无结果

(222)C

知识点：数据的输入与输出，输入输出函数的调用

评析：本题主要考查对 printf 函数域宽描述符的理解，在这里我们要特别注意：输出数据的实际精度并不主要取决于格式项中的域宽与精度，也不取决于输入的数据精度，而主要取决于数据在机器内的存储精度。

(223)以下选项中不属于 C 语言的类型的是_____。

- A)signed short int B)unsigned char

C)signed long D)long short

(223)D

知识点: c 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

评析: c 语言中归纳起来, 基本类型数据有以下几种:

```
[signed]char;
unsigned char;
[signed]short[int];
unsigned short[int];
[signed]long[int];
unsigned long[int];
float;
double.
```

(224)以下程序的输出结果是_____。

```
main()
{int a=5,b=4,c=6,d;
printf( " %d \n " , d=a>c?(a>c?a: c): (b));
}
```

A)5 B)4 C)6 D)不确定

(224)B

知识点: C 运算符的种类、运算优先级和结合性

评析: 本题最重要的分析 $d=a>c?(a>c?a:c):(b)$, 首先运算括号内的式子 $a>c?a:c$, 它的值是 c 的值 6, 即式子化成 $d=a>c?6:4$ 。显然 $a=c$, 所以将 4 赋给 d。

(225)以下程序中, while 循环的循环次数是_____。

```
main()
{ int i=0;
while(i<10)
{if(i<1)continue;
if(i==5)break;
i++;
}
.....
}
```

A)1 B)10 C)6 D)死循环, 不能确定次数

(225)D

知识点: while 和 do-while 循环结构

评析: 题目中, 进入循环后, 先执行一个条件语句, 如果 i 的值小于 1, 那么直接进入下一轮循环, 因为 i 的初始值是 0, 小于 1, 故直接进入下一轮循环, 又因为 i 的值始终没有改变, 所以这个循环成了死循环。

(226)以下程序的输出结果是_____。

```
main()
{int a=0,i;
for(i=1; i<5; i++)
{ switch(i)
{ case 0:
```



```

case 3: a+=2:
case 1:
case 2: a+=3:
default: a+=5;
}
}
printf( " %d\n " , a)
}

```

A)31 B)13 C)10 D)20

(226)A

知识点: 用 switch 语句实现多分支选择结构

评析: switch 结构的执行过程为: 进入 switch 结构后, 对条件表达式进行计算, 然后从上至下去找与条件表达式的值相匹配的 case; 以此作为入口, 执行 switch 结构中后面的各语句。第一次 for 循环中, switch 结构的条件表达式 i 的值是 1, 则从 case 1 开始执行后面的语句, 先执行 a+=3, a 的值变成 3。接着执行 a+=5, a 的值变成了 8。然后进入第二次循环, switch 结构条件表达式 i 的值 2, 则从 case 2 开始执行后面的语句, 结束第二轮循环时 a 的值是 16。在第三轮循环中, switch 结构条件表达式的值是 3, 则从 case 3 后面的语句开始执行, a+=2 语句使 a 的值变成了 18, 然后执行 a+=3, 和 a+=5, 第三轮循环结束时, a 的值是 26。第四轮循环中, switch 结构条件表达式的值是 4, 从 default 处开始执行, 执行一个 a+=5 的操作, a 的值变成了 31, i 的值经修改变成了 5, 不再满足循环条件, 退出循环。

(227)以下程序的输出结果是_____。

```

main()
{int a=4, b=5, c=0, d;
d=!a&&!b || !c;
printf( " %d\n " , d);
}

```

A)1 B)0 C)非 0 的数 D)-1

(227)A

知识点: 表达式语句, 空语句, 复合语句

评析: !a 值为 0, 故!a&&!b 的值为 0, !c 的值为 1, 而 0||1 的值应该是 1。所以答案是 A。

(228)以下程序的输出结果是_____。

```

#include<stdio. h>
main()
{int i=0, a=0;
while(i<20)
{for(;; )
{if((i%10)==0)break;
else i--;
}
i+=11; a+=i;
}
printf( " %d\n " , a);
}

```

A)21 B)32 C)33 D)11

(228)B

知识点: for 循环结构

评析: 本题中, 有两个循环语句。首先, $i=0$, 进入 while 循环, for 语句中对循环没有任何条件, 故直接进入 for 循环, 因为 $0\%10$ 结果还是 0, 所以跳出 for 循环, 执行 $i+=11$, i 的值变为 11, 执行 $a+=i$, a 的值变为 11; 接着进入下一轮 while 循环, 在 for 循环中, 因为 i 的值是 11, 对 10 取余结果为 1, 所以执行 i 自减, i 的值变成 10, 进入新一轮 for 循环, 因为 $10\%10=0$, 所以跳出 for 循环, 执行 $i+=11$, i 的值变成 21, 执行 $a+=i$, 也就是将 21+11 的和赋给 a , $a=32$, 因为此时 i 的值不再小于 20, 所以结束循环, 故输出结果是 32。

(229)以下程序的输出结果是_____。

```
int f()
{static int i=0;
int s=1;
s+=i; i++;
return S;
}
main()
{int i, a=0;
for(i=0; i<5; i++)a+=f();
printf(" %d\n", a);
}
```

A)20 B)24 C)25 D)15

(229)D

知识点: 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

评析: 本题主要考的是对变量存储属性的理解, 一个变量被指定为静态变量, 在编译时就为其分配了存储空间, 程序一开始执行使被建点, 直到该程序执行结束都存在, 而不像动态变量只存在于函数或分程序被调用期间。在函数多次被调用的过程中静态局部变量的值具有可继承性。在第一次调用函数结束时, i 的值是 1, 返回值 1, 第二次调用函数时, i 的值保持为 1, 执行自加操作后, j 的值变成了 2, 到了第三次调用时, i 的值保持了上次调用结束时的值, 再执行自加, 值变成了 3。到第五次调用结束叫, i 的值是 5。而 s 的值不具备 i 的这种可继承性, 每次调用时, 它都被重新赋值为 1, 再执行下面的操作。

(230)若有以下程序

```
#include<stdio.h>
void f(int n);
main()
{void f(int n);
f(5);
}
voidf(int n)
{printf(" %d\n", n); }
```

则以下叙述中不正确的是_____。

- A)若只在主函数中对函数 f 进行说明, 则只能在主函数中正确调用函数 f
- B)若在主函数前对函数 f 进行说明, 则主函数和其后的其他函数中都可以正确调用函数 f
- C)对于以上函数程序, 编译时系统会提示出错信息; 提示对 f 函数重复说明

D)函数 f 无返回值，所以可用 void 将其类型定义为无值型

(230)C

知识点：函数的定义方法

评析：一个函数在一个文件中的定义只能有一次，但对它的声明却可以有很多个。一个函数可以正确调用在当前函数之前声明的函数。对于本题来说，如果有一个新的函数 fNew()是在 f()函数之后说明的，那么即使主函数 tp 没有对函数 f 进行说明，在 lNew()中同样可以正确调用函数 f()。

(231)在 C 语言中，形参的缺省存储类别是_____。

A)auto B)register C)static D)extern

(231)A

知识点：形式参数与实在参数，参数值的传递

评析：程序进行编译时，并不为形式参数分配存储空间。只有在被调用时，形式参数才临时地占有存储空间。形式参数用关键字 auto 作存储类别的声明时，关键字“auto”可以省略，auto 不写则隐含确定为“自动存储类别”，它属于动态存储方式。

(232)若指针 p 已正确定义，要使 p 指向两个连续的整型动态存储单元，不正确的语句是_____。

A)p=2*(int*)malloc(sizeof(int)); B)p=(int*)malloc(2*sizeof(int))

C)p=(int*)malloc(2*2) D)p=(int*)calloc(2,sizeof(int))

(232)A

知识点：用指针作函数参数

评析：malloc 函数的作用是在内存开辟指定大小的存储空间，并将此存储空间的地址作为函数值带回，

它的原型为 void*malloc(unsigned int size)，函数值为指针，这个指针指向 void 类型，也就是不规定指向任何具体的类型。如果要将此返回地址赋给一个指向特定类型的指针变量，则应进行强制类型转换。calloc 函数的模型是：void*calloc(unsigned int num,unsigned int size)，作用是分配 num 个大小为 size 字节的空间。

(233)以下程序的输出结果是_____。

```
main()
```

```
{ int x=040;
```

```
printf( " %0 \ 11 " , x<<1);
```

```
}
```

A)100 B)80 C)64 D)32

(233)A

知识点：简单的位运算

评析：040 用二进制表示就是 00100000，左移一位，得到的值应该是 01000000，化成八进制应该是 0100。

(234)若要打开 A 盘上的 user 子目录下名为 abc. txt 的文本文件进行读、写操作，下面符合此要求的函数调用是_____。

A)fopen(" A: \user\abc. txt " , " r ")B)fopen(" A: \\user\abc. txt " , " r+ ")

C)fopen(" A: \user\abc. txt " , " rb ")D)fopen(" A: \\user\abc. txt " , " w ")

(234)B

知识点：文件的打开与关闭(fopen, fclose)

评析:本题考查点是 fopen 函数中文件使用方式的指定。

文件使用方式“r”，表示以“只读”方式打开一个字符文件。

文件使用方式“r+”，表示以“读写”方式打开一个字符文件。

文件使用方式“rb”，表示以“只读”方式打开一个二进制文件。

文件使用方式“w”，表示以“只写”方式打开一个字符文件。

(235)以下不能正确进行字符串赋初值的语句为_____。

A)char str[5]="good! " B)char str[]="good! "

C)char *str="good! " D)char str[5]: { 'g', 'o', 'o', 'd' }

(235)A

知识点：字符串与字符数组

评析：因为用字符串作初值为字符数组赋值时，系统总会自动在字符串的末尾补上一个“\0”，作为字符串的结束标志。如果如 A 中定义的话，本来应该把第 6 个字符也就是“\0”赋给 str 数组，但此数组只含 5 个元素，因此最后一个字符“\0”未能放入 str 数组，而是放到了 str 数组之后的存储单元中，这就可能会破坏其他数据区或程序本身。

(236)若有下面的说明和定义：

```
struct test
```

```
{int m1; char m2; float m3;
```

```
union uu{char u1[5]; int u2[2]; }ua;
```

```
}myaa;
```

则 sizeof(struct test)的值是_____。

A)12 B)16 C)14 D)9

(236)A

知识点：结构体和共用体类型数据的定义方法和引用方法

评析：在定义了结构体变量后，系统会为之分配内存单元，在内存中一个整型占 2 个字节，一个字符占 1 个字节，一个数组元素占 1 个字节，一个浮点型占 4 个字节。结构体变量所占内存长度是各成员所占内存长度之和；而共用体变量所占的内存长度等于最长的成员的长度。所以 test 在内存中占 12 个字节(2+1+4+5=12)。

(237)若有定义：int aa[8];，则以下表达式中不能代表数组元素 aa[1]的地址的是_____。

A)&aa[0]+1 B)&aa[1] C)aa[0]++ D)aa+1

(237)C

知识点：指针与指针变量的概念，指针与地址运算符

评析：答案 A 中&aa[0]表示 aa[0]的地址，加上 1 就是 aa[1]的地址。同样 aa[i]的地址也可以直接用&aa[i]或 aa+i 表示。

(238)以下程序的输出结果是_____。

```
f(int b[], int m,int n)
```

```
{int i, s=0;
```

```
for(i=m; i<n; i=i+2)s=s+b[i];
```

```
return S;
```

```
}
```

```
main()
```

```
{inc x,a[]={1,2,3,4,5,6,7,8,9};
```

```
x=f(a,3,7);
```

```
printf( " %d\n ",x);
```

```
}
```

A)10 B)18 C)8 D)15

(238)A

知识点：函数的正确调用，嵌套调用，递归调用

评析：本题中，主函数调用 f 函数，在 f 函数中执行了两次循环，第一次 i=3，s=0+b[i]=4，然后 i 的值被修改成 5，仍然满足循环条件；进入第二轮循环，s=4+b[5]=4+6=10，随后，i 的值被修改成 7，不再满足循环条件，退出循环，返回 s 的值。所以最后输出结果是 10。

(239)若有以下定义和语句：

```
int s[4][5], (*ps)[5];
```

```
ps=s;
```

则对 s 数组元素的正确引用形式是_____。

A)ps+1 B)*(ps+3) C)ps[0][2] D)*(ps+1)+3

(239)C

知识点：一维数组和多维数组的定义、初始化和引用

评析：选项 A 得到的是 s 数组第一行的地址，选项 B 得到的是 s 数组第三行第零列元素的地址，而选项 D 得到的是 s 数组第一行第三列元素的地址。只有选项 C 正确引用了 ps 数组第零行第二列的元素。

(240)以下程序的输出结果是_____。

```
main()
```

```
{int b[3][3]={0, 1, 2, 0, 1, 2, 0, 1, 2}, i, j, t=1;
```

```
for(i=0; i<3; i++)
```

```
for(j=i; j<=i; j++)t=t+b[i][b[j][j]];
```

```
printf(" %d\n", t);
```

```
}
```

A)3 B)4 C)1 D)9

(240)B

知识点：循环的嵌套

评析：本程序中，有两个循环语句嵌套使用，在第一次外循环中，i=0，执行内循环时，将 j 赋值为 i 的值 0，执行一次 t=t+b[i][b[j][j]]，得到 t=1，退出内循环，接着进入第二次外循环，在第二次外循环结束时 t=2，然后执行第三次外循环，得到 t=4，退出循环，输出 t 的值。选项 B)软件开发工具是在实现阶段需完成的任务：选项 c)软件开发费用是在可行性研究阶段需完成的任务。

241)以下程序的输出结果是_____。

```
#include <stdio. h>
```

```
#include <string. h>
```

```
main()
```

```
{char b1[8]= " goddwer ", b2[8], *pb=b1+3;
```

```
while(--pb>=b1)strcpy(b2,pb);
```

```
printf(" %d\n", strlen(b2));
```

```
}
```

A)8 B)3 C)1 D)7

(241)D

知识点：库函数的正确调用

评析：本题中，每次执行循环，都是将 pb 指针指向的内容复制到 b2 中，第一次循环中，将 b1[2]~b1[7]六个字符复制到 b2 中，第二次循环，将 b[1]~b[7]七个字符复制到 b2 中，第三次循环，将 b1 数组的全部内容复制到 b2 中。而后一次复制会覆盖前面的复制，最后 b2 的实际长度应该是八，而 strlen()函数测出的是一个字符串中“\0”之前的全部字符的个数，

所以最后输出的值是 7。

(242)在说明语句: `int*f()`; 中, 标识符 `f` 代表的是_____。

- A)一个用于指向整型数据的指针变量
- B)一个用于指向一维数组的行指针
- C)一个用于指向函数的指针变量
- D)一个返回值为指针型的函数名

(242)D

知识点: 返回指针值的指针函数

评析: 一个函数在编译时被分配给一个入口地址, 这个入口地址就称为函数的指针。可以用一个指针变量指向函数, 然后通过该指针变量调用此函数。`int*f()`表示 `f` 是一个函数, 它带回一个指针值, 这个指针是指向一个整型数据的。

(243)不合法的 `main` 函数命令行参数表示形式是_____。

- A)`main(int a,char*c[])` B)`main(int arc,char**arv)`
- C)`main(int argc,char*argv)` D)`main(int argv,char*arge[])`

(243)C

知识点: 指针数组, 指向指针的指针, `main` 函数的命令行参数

评析: 本题主要考查了对 `main` 函数参数的了解, `main` 函数可以有两个形参, 一般形式是: `main(int argc,char*argv[])`, 也就是说, 它的第一个形参是一个整型变量。第二个形参是一个指针数组, 其元素指向字符型数据。

(244)以下程序的输出结果是_____。

```
int x=3;
main()
{int i;
for(i=1; i<x; i++)incre();
}
incre()
{static intx=1;
x*=x+1;
printf(" %d ", x);
}
```

- A)3 3 B)2 2 C)2 6 D)2 5

(244)C

知识点: 内部函数与外部函数

评析: `incre()`中定义一个静态局部变量 `x`, 它的作用范围仅限于本函数中, 而不会影响 `main()` 函数中的 `x` 值, 所以 `main()`中执行两次 `for` 循环, 第一次调用 `incre()`函数时, 函数中 `x` 的初值是 1, `x*=x+1` 语句可以写成 `x=x*(x+1)`, 结果 `x` 的值变成 2, 随后输出值 2; 在第二次调用 `incre()`函数时, `x` 的值继承了上次调用的运行结果, 变成了 2, 执行了 `x*=x+1` 后, `x` 的值是 6, 故又输出了数字 6。

(245)若有以下定义:

```
struct link
{int data;
struct link *next;
}a, b, c, *p, *q;
```

且变量 `a` 和 `b` 之间已有如图所示的链表结构:

指针 p 指向变量 a, q 指向变量 c。则能够把 c 插入到 a 和 b 之间并形成新的链表的语句组是_____。

- A)a. next=c; c. next=b; B)p. next=q; q. next=p. next;
C)p->next=&c; q->next=p->next; D)(*p). next=q; (*q). next=&b;
(245)D

知识点: 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

评析: 本题考的是指针的运用。即先将元素 a 的指针指向 q 指针所指向的元素即 c, 然后将 c 的指针指向 b 元素, 采取的是直接取元素地址的方式。

(246)算法分析的目的是_____。

- A)找出数据结构的合理性 B)找出算法中输入和输出之间的关系
C)分析算法的易懂性和可靠性 D)分析算法的效率以求改进
(246)D

知识点: 算法分析

评析: 算法分析是指对一个算法的运行时间和占用空间做定量的分析, 一般计算出相应的数量级, 常用时间复杂度和空间复杂度表示。分析算法的目的就是要降低算法的时间复杂度和空间复杂度, 提高算法的执行效率。

(247)n 个顶点的强连通图的边数至少有_____。

- A)n-1 B)n(n-1) C)n D)n+1
(247)C

知识点: 强连通图

评析: 在有向图中, 若任意两个顶点都连通, 则称该图是强连通图, 这样的有向图的形状是环状, 因而至少应有 n 条边。

(248)已知数据表 A 中每个元素距其最终位置不远, 为节省时间, 应采用的算法是_____。

- A)堆排序 B)直接插入排序 C)快速排序 D)直接选择排序
(248)B

知识点: 插入排序算法

评析: 当数据表 A 中每个元素距其最终位置不远, 说明数据表 A 按关键字值基本有序, 在待排序序列基本有序的情况下, 采用插入排序所用时间最少, 故答案为选项 B。

(249)用链表表示线性表的优点是_____。

- A)便于插入和删除操作 B)数据元素的物理顺序与逻辑顺序相同
C)花费的存储空间较顺序存储少 D)便于随机存取
(249)A

知识点: 链表的结构

评析: 链式存储结构克服了顺序存储结构的缺点: 它的结点空间可以动态申请和释放; 它的数据元素的逻辑次序靠结点的指针来指示, 不需要移动数据元素。故链式存储结构下的线性表便于插入和删除操作。

(250)下列不属于结构化分析的常用工具的是_____。

- A)数据流图 B)数据字典 C)判定树 D)PAD 图
(250)D

知识点: 结构化分析的常用工具

评析: 结构化分析的常用工具有数据流图、数据字典、判定树和判定表。而 PAD 图是常见的过程设计工具中的图形设计工具。

(251)软件开发的结构化生命周期方法将软件生命周期划分成_____。

- A)定义、开发、运行维护
- B)设计阶段、编程阶段、测试阶段
- C)总体设计、详细设计、编程调试
- D)需求分析、功能定义、系统设计

(251)A

知识点: 软件生命周期

评析: 通常, 将软件产品从提出、实现、使用维护到停止使用退役的过程称为软件生命周期。它可以分为软件定义、软件开发及软件运行维护三个阶段。

(252)在软件工程中, 白盒测试法可用于测试程序的内部结构。此方法将程序看做是_____。

- A) 循环的集合 B)地址的集合 C)路径的集合 D)目标的集合

(252)C

知识点:软件测试方法

评析: 软件的自盒测试方法是把测试对象看做一个打开的盒子, 它允许测试人员利用程序内部的逻辑结构及有关信息, 设计或选择测试用例, 对程序所有逻辑路径进行测试。

(253)在数据管理技术发展过程中, 文件系统与数据库系统的主要区别是数据库系统具有_____。

- A)数据无冗余 B)数据可共享
- C)专门的数据管理软件 D)特定的数据模型

(253)D

知识点:数据库系统的基本特点

评析: 在文件系统中, 相互独立的记录其内部结构的最简单形式是等长同格式记录的集合, 易造成存储空间大量浪费, 不方便使用。而在数据库系统中, 数据是结构化的, 这种结构化要求在描述数据时不仅描述数据本身, 还要描述数据间的关系, 这正是通过采用特定的数据模型来实现的。

(254)分布式数据库系统不具有的特点是_____。

- A)分布式 B)数据冗余
- C)数据分布性和逻辑整体性 D)位置透明性和复制透明性

(254)B

知识点:数据库系统的基本概念

评析: 分布式数据库系统具有数据分布性、逻辑整体性、位置透明性和复制透明性的特点, 其数据也是分布的; 但分布式数据库系统中数据经常重复存储, 数据也并非必须重复存储, 主要视数据的分配模式而定。若分配模式是一对多, 即一个片段分配到多个场地存放, 则是冗余的数据库, 否则是非冗余的数据库。

(255)下列说法中, 不属于数据模型所描述的内容的是_____。

- A)数据结构 B)数据操作 C)数据查询 D)数据约束

(255)C

知识点:数据模型的基本概念

评析: 数据模型所描述的内容有 3 个部分, 它们是数据结构、数据操作和数据约束。其中, 数据模型中的数据结构主要描述数据的类型、内容、性质, 以及数据库的联系等; 数据操作主要是描述在相应数据结构上的操作类型与操作方式; 数据模型中的数据约束主要描述数据结构内数据间的语法、语义联系, 它们之间的制约与依存关系, 以及数据动态变化的规则, 以保证数据的正确、有效与相容。

(256)以下选项中合法的实型常数是_____。

A)5E2. 0 B)E-3 C). 2E0 D)1. 3E

(256)C

知识点：实型数据的指数表示方式

评析：以指数形式表示实数时，字母 e(或 E)之前必须有数字，且 e 后面的指数必须为整数。

(257)已知大写字母 A 的 ASCII 码是 65，小写字母 a 的 ASCII 码是 97，则用八进制表示的字符常量 ‘\101’ 是_____。

A)字符 A B)字符 a C)字符 e D)非法的常量

(257)A

知识点：字符与 ASCII 码的转换，

评析：八进制 101 转换为 10 进制即为 65，而字母 A 的 ASCII 码是 65，所以用八进制表示的字符常量 ‘\101’ 是 A。

(258)设 a 和 b 均为 double 型常量，且 a=5.5、b=2.5，则表达式(int)a+b / b 的值是_____。

A)6. 500000 B)6 C)5. 500000 D)6. 000000

(258)D

知识点：各类数值型数据的混合运算

评析：各类数值型数据在进行混合运算时，不同类型的数据要先转换成同一类型，然后进行运算。转换的规则为：总是转换为精度更大的数据类型，字符数据转换为整数，short 型转为 int 型，float 型数据在运算时转换成双精度型，以提高运算精度。

(259)已知 i、j、k 为 int 型变量，若从键盘输入：1,2,3 /，使 i 的值为 1、j 的值为 2、k 的值为 3，以下选项中正确的输入语句是_____。

A)scanf(" %2d%2d%2d " ,&i,&j, &k);

B)scanf(" %d %d %d " , &i, &j, &k);

C)scanf(" %d, %d, %d " , &i, &j, &k);

D)scanf(" i=%d,j=%d, k=%d " , &i, &j, &k);

(259)C

知识点：格式输入函数

评析：如果在“格式控制”字符串中除了格式说明以外还有其他字符，则在输入数据时应输入与这些字符相同的字符。

(260)若有以下程序：

```
main()
{int k=2, i=2, m;
m=(k+=i*=k);
printf( " %d, %d\n " , m, i);
}
```

执行后的输出结果是_____。

A)8,6 B)8,3 C)6,4 D)7,4

(260)C

知识点：复合的赋值运算符

评析：在赋值符“=”之前加上其他运算符，可以构成复合的运算符。此运算为右结合性。

即从右往左计算。i*=k 即 i*i*k=2*2=4，k=k+i=2+4=6，然后将 k 的值赋给 m。

(261)已有定义：int x=3, y=4, z=5;，则表达式!(x+y)+z-1&& y+z / 2 的值是_____。

A)6 B)0 C)2 D)1

(261)D

知识点：运算符的优先级

评 析：在本题中，运算符的优先级分别为： $!>/>+,->\&\&$ ，即： $!(3+4)+5-1\&\&4+5/2=0+5-1\&\&4+2=4\&\&6=1$ 。

(262)以下选项中，与 $k=n++$ 完全等价的表达式是_____。

A) $k=n,n=n+1$ B) $n=n+1,k=n$ C) $k=++n$ D) $k+=n+1$

(262)A

知识点：自增及赋值运算

评 析： $n++$ 是自增运算，是先使用，然后再使 n 加 1，分开来写即为： $k=n,n=n+1$ 。

(263)以下程序的功能是：按顺序读入 10 名学生 4 门课程的成绩，计算出每位学生的平均分并输出，程序如下：

```
main()
{ int n, k;
float score,sum,ave;
sum=0. 0;
for(n=1; n<=10; n++)
{ for(k=1; k<=4; k++)
{scanf( " %f " ,&score);
sum+=score;
}
ave=sum / 4. 0;
printf( " NO%d: %f\n " ,n,ave);
}
}
```

上述程序运行后结果不正确，调试中发现有一条语句在程序中的位置不正确。这条语句是_____。

A) $sum=0. 0;$ B) $sum+=score;$
C) $ave=sum / 4. 0;$ D) $printf(" NO%d: %f\n " ,n,ave);$

(263)A

知识点：for 循环语句的使用

评 析：题目中要求每位学生的平均分数，首先要求每位学生的总分，也就是程序中的 sum ，所以 sum 应在第一个 for 循环体内赋初值，如果在循环体外赋初值就会得到所有学生的分数总和，不合题意。

(264)有以下程序

```
main()
{ int a=15, b=21, m=0;
switch(a%3)
{ case 0: m++; break;
case 1: m++;
switch(b%2)
{ default: m++;
case 0: m++; break;
}
}
printf( " %d\n " , m);
}
```

程序运行后的输出结果是_____。

A)1 B)2 C)3 D)4

(264)A

知识点: switch 语句

评 析: 因为 $a \% 3$ 是个常数, 所以只执行一次就跳出 switch 结构。即只执行 $m++$, 此时 m 为 1。

(265)若有说明: $\text{int } n=2, *p=\&n, *q=p;$, 则以下非法的赋值语句是_____。

A) $p=q;$ B) $*p=*q;$ C) $n=*q;$ D) $p=n;$

(265)D

知识点: 指针变量的引用

评 析: 指针变量不同于整型变量和其他类型的变量, 它是用来存放地址(指针)的, 不能将一个整型变量(或任何其他非地址类型的数据)赋给一个指针变量, 这样的赋值是不合法的。

本题中的答案 D 就属于这种情况, 所以 $p=n;$ 是非法的赋值语句。

(266)有以下程序

```
Void fun(char*c, int d)
{ *c=*c+1; d=d+1;
printf( " %c, %c, " , *c, d);
}
main()
{ char a= 'A', b= 'a';
fun(&b, a); printf( " %c, %c\n " , a, b);
}
```

程序运行后的输出结果是_____。

A)B,a,B,a B)a, B, a, B C)A,b, A, b D)b, B, A, a

(266)D

知识点: 函数的调用

评 析: 将 a, b 代入函数得 $*\&b=*\&b+1=a+1=b$, 所以打印出第一个字母 b :

$a=a+1=A+1=B$, 打印出第二个字符 B :

$a= 'A'$, 输出字母 A ;

$b= 'a'$, 输出字母 a 。

(267)以下程序中函数 sort 的功能是对 a 所指数组中的数据进行由大到小的排序。

```
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;
sort(&aa[3], 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)10,9,8,7,6,5,4,3,2,1,
C)1,2,3,8,7,6,5,4,9,10, D)1,2,10,9,8,7,6,5,4,3,

(267)C

知识点：对数组的操作

评 析：本题并不是对数组的所有元素进行排序，而是对从 a[3] 开始往后的 5 个元素进行排序。

(268)有以下程序

```
main()
{char a[]={ 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', '\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)9,9 B)8,9 C)1,8 D)9,8

(268)D

知识点：计算字符数组与字符串的长度

评 析：sizeof() 函数是计算字符数组的长度，因为 ‘\0’ 也是一个字符，要计算在内。在字符数组末尾加上 ‘\0’ 作为字符串结束标志，strlen() 是计算字符串长度的函数，遇到 ‘\0’ 即认为是字符串结束，不把空字符计入字符串的长度。

(269)以下程序中的函数 reverse 的功能是将 a 所指数组中的内容进行逆置。

```
void reverse(int a[], int n)
{int i, t;
for(i=0; i<n / 2; i++)
{t=a[i]; a[i]=a[n-1-i]; a[n-1-i]=t; }
}
main()
{int b[10]={1,2,3,4,5,6,7,8,9,10}; int i, s=0;
reverse(b, 8);
for(i=6; i<10; i++)s+=b[i];
printf( " %d\n ", s);
}
```

程序运行后的输出结果是_____。

A)22 B)10 C)34 D)30

(269)A

知识点：对数组的操作

评 析：本题并不是将整个数组中的内容逆置，而是逆置前 8 个元素。逆置后的数组为：8, 7, 6, 5, 4, 3, 2, 1, 9, 10 通过 for 循环计算 b[6]+b[7]+b[8]+b[9] 即 2+1+9+10=22。

(270)有以下程序

```
#include<string. h>
main()
{ char*p=" abcde \0fghjik \0 " ;
printf( " %d\n ",strlen(p));
}
```

程序运行后的输出结果是_____。

A)12 B)15 C)6 D)5

(270)D

知识点: 计算字符串的长度

评 析: “\0” 是字符串标志,当遇到此标志时字符串已结束.所以字符串的长度为 5.

(271)程序中头文件 `typel.h` 的内容是:

```
#define N 5
#define M1 N*3
```

程序如下:

```
#include " typel.h "
#define M2 N*2
main()
{ int i;
i=M1+M2;
printf( " %d\n " ,i);
}
```

程序编译后运行的输出结果是_____。

A)10 B)20 C)25 D)30

(271)C

知识点:宏定义

评 析:宏定义就是用一个指定的标识符(即名字)来代表一个字符串,它的一般形式为:

```
#define 标识符 字符串
```

这种方法使用户能以一个简单的名字代替一个长的字符串,因此把这个标识符(名字)称为

” 宏名 ” 宏名一般习惯用大写字母表示, 以与变量名相区别。使用宏名代替一个字符串,可以减少程序中重复书写某些字符串的工作量。如本题用宏名 `M1` 代替字符串 `N*3`; `M2` 代替字符串 `N*2`。

$i=M1+M2=N*3+N*2=5*3+5*2=25$ 。

(272)以下叙述中错误的是_____。

A)二进制文件打开后可以先读文件的末尾, 而顺序文件不可以

B)在程序结束时, 应当用 `fclose` 函数关闭已打开的文件

c)利用 `fread` 函数从二进制文件中读数据, 可以用数组名给数组中所有元素读入数 据

D)不可以用 `FILE` 定义指向二进制文件的文件指针

(272)D

知识点: 对文件的操作

评 析: 每个被使用的文件都在内存中开辟一个区, 用来存放文件的有关信息, 这些信息是保存在一个名为 `FILE` 的结构体类型的结构体变量中的。而文件又分为 `ASCII` 文件和二进制文件, 所以, 可以用 `FILE` 定义指向二进制文件的文件指针。

(273)有以下程序

```
#include<string. h>
main(int argc,char*argv[])
{int i,len=0;
for(i=1; i<argc; i++)len+=strlen(argv[i]);
printf( " %d\n " , len);
}
```

程序编译连接后生成的可执行文件是 `ex1. exe`, 若运行时输入带参数的命令行是:

exl abcdefg 10 /

则运行的结果是_____。

A)22 B)17 C)12 D)9

(273)D

知识点：指针数组作 main 函数的形参

评 析：本题主函数带有两个参数，一个 int 型变量 argc，另一个是 char 型的一维一级指针数组 argv。当主函数被执行时，系统自动将根据命令行的情况，分别给主函数的两个参数赋值。argc 用于存放命令行中命令字和参数的总和的个数 4，argv 用来存放命令行中命令字和所有参数的字符串，并规定 argv[0]存放命令字符串 exl，argv[1]用来存放第一个参数的字符串 abcd，argv[2]用来存放第二个参数的字符串

efg，argv[3]用来存放第三个参数的字符串 10。那么 for 循环将循环 3 次分别将 argv 数组中元素的长度累加到 len 中，所以答案为 9。

(274)有以下程序

```
int fa(int x)
{ return x*x; }
int fb(int x)
{ return x*x*x; }
int f(int(*f1)(), int(*f2)(), int x)
{ return f2(X)-f1(x); }
main()
{ int i;
  i=f(fa,fb,2); printf( " %d\n ",i);
}
```

程序运行后的输出结果是_____。

A)-4 B)1 C)4 D)8

(274)C

知识点：函数的嵌套调用

评 析：f(fa,fb,2)即为 fb(2)-fa(2)=2*2*2-2*2=8-4=4。

(275)有以下程序

```
void ss(char*s,char t)
{while(*s)
{if(*S==t)*s=t- 'a' + 'A';
S++;
}
}
main()
{char str1[100]= " abcddefefdbd ", c= 'd';
ss(str1, c); printf( " %s\n ",str1);
}
```

程序运行后的输出结果是_____。

A)ABCDDEFEDBD B)abcDDfefDbD

C)abcAAfefAbA D)Abcddefefdbd

(275)B

知识点：将字符串中指定的小写字母转换为大写字母

评 析：在主函数中因为 c 被赋值为字符 'd'，所以函数的功能是将字符串中的小写字母 d 转换为大写。

(276)设有如下定义：

```
struct sk
{ int a;
float b;
} data;
int *p;
```

若要使 p 指向 data 中的 a 域，正确的赋值语句是_____。

A)p=&a; B)p=data. a; C)p=&data. a; D)*p. data. a

(276)C

知识点：指向结构体变量的指针

评 析：将 data.a 的起始地址赋给指针变量 p，也就是使 p 指向 data.a。

(277)有以下程序

```
#include <stdlib.h>
struct NODE
{ int num; struct NODE *next; }
main()
{ struct NODE *p, *q, *r;
p=(struct NODE*)malloc(sizeof(struct NODE));
q=(struct NODE*)malloc(sizeof(struct NODE));
r=(struct NODE*)malloc(sizeof(struct NODE));
p->num=10; q->num=20; r->num=30;
p->next=q; q->next=r;
printf( " %d\n " , p->num+q->next->num);
}
```

程序运行后的输出结果是_____。

A)10 B)20 C)30 D)40

(277)D

知识点：用指针处理链表

评 析：p->num 即为 10，q->next->num 即为 r->num，为 30，所以 p->num+q->next->num=10+30=40。

(278)若有以下说明和定义

```
typedef int *INTEGER
```

```
INTEGER p, *q;
```

以下叙述正确的是_____。

A)p 是 int 型变量 B)p 是基类型为 int 的指针变量

C)q 是基类型为 int 的指针变量 D)程序中可用 INTEGER 代替 int 类型名

(278)B

知识点：用 typedef 定义类型

评 析：typedef int*INTEGER; (声明 INTEGER 为整型指针类型);

INTEGER p,*q; (定义 p,*q 为整型指针类型)，所以 p 是基类型为 int 的指针变量；*q 是基类型为 int 的指针变量；程序中可用*INTEGER 代替 int 类型名。

(279)有以下程序

```
main()
{unsigned char a, b, c;
a=0x3; b=a10x8; c=b<<1;
printf( " %d%d\n ", b, c);
}
```

程序运行后的输出结果是_____。

A)-11 12 B)-6 -13 C)12 24 D)11 22

(279)D

知识点：位运算

评 析：0x3 表示 16 进制表，”|” 为按位或，”<<” 为按位左移，将 a，b 转换为二进制数进行位运算，再将结果以十进制打印出来，结果为 11 22。

(280)以下程序中函数 f 的功能是将 n 个字符串，按由大到小的顺序进行排序。

```
#include<string.h>
void f(char p[][10], int n)
{char t[20]; int i, j;
for(i=0; i<n-1; i++)
for(j=h1. j<n. j++)
if(strcmp(p[i], p[j])<0)
{strcpy(t, p[i]); strcpy(p[i], p[j]); strcpy(p[j], t); }
}
main()
{char p[][10]={ " abc ", " aabdfg ", " abbd ", " dcdbe ", " cd " }; int i;
f(p, 5); printf( " %d\n ",strlen(p[0]));
}
```

程序运行后的输出结果是_____。

A)6 B)4 C)5 D)3

(280)C

知识点：比较字符串的大小

评 析：比较字符串的大小是从字符串的第一个字母开始比较，如果第一个字母相同则比较第二个字母，以此类推，直至字符串结束。strcmp(字符串 1，字符串 2)的作用是比较字符串 1 和字符串 2：字符串 1=字符串 2，函数值为 0；字符串 1>字符串 2，函数值为 1；字符串 1<字符串 2，函数值为 -1。strcpy(字符数组 1，字符串 2)的作用是将字符串 2 复制到字符数组 1 中去。结合本题可知，比较后的字符串数组为：

p[0]="dcdbe"

p[1]="cd"

p[2]="abc"

p[3]="abbd"

p[4]="aabdfg"

所以，strlen(p[0])=5。

填空题

(1)算法的复杂度主要包括时间复杂度和___【1】___复杂度。

(1)【1】空间

知识点：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)

评 析：一个程序在计算机上运行时所耗费的时间由下列因素所决定：程序运行时所需输入的

数据总量,对源程序进行编译所需时间,计算机执行每条指令所需时间,程序中的指令重复执行的次数。前三条取决于实现算法的计算机软、硬件系统,习惯上常常把语句重复执行的次数作为算法运行时间的相对量度,称作算法的时间复杂度。算法在运行过程中需辅助存储空间的大小称为算法的空间复杂度。

(2)数据的物理结构在计算机存储空间中的存放形式称为数据的____【2】____。

(2)【2】内模式或物理模式或存储模式

知识点:数据库的基本概念:数据库,数据库管理系统,数据库系统

评 析:模式也称逻辑模式或概念模式,是数据库中全体数据的逻辑结构和特征的描述,共数是所有用户的公共数据视图。例如数据记录由哪些数据项构成,数据项的名字、类型、取值范围等。

外模式是模式的子集,所以也称子模式或用户模式,是数据库用户能够看见的和使用的、局部的逻辑结构和特征的描述,是与某一应用有关的数据的逻辑表示。

内模式也称物理模式或存储模式。一个数据库只有一个内模式,它是数据物理结构和存储方式的描述,是数据库内部的表示方法。例如,记录的存储方式是顺序存储、索引按照什么方式组织;数据是否压缩存储,是否加密等。

(3)若按功能划分,软件测试的方法通常分为____【3】____测试方法和黑盒测试方法。

(3)【3】白盒

知识点:软件测试的寺法,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

评 析:软件测试的方法有三种:动态测试、静态测试和正确性证明。设计测试实例的方法一般有两类:黑盒测试方法和白盒测试方法。在使用黑盒法设计测试实例时,测试人员将程序看成一个“黑盒”,也就是说,他不关心程序内部是如何实现的,而只是检查程序是否符合它的“功能说明”,所以使用黑盒法设计的测试用例完全是根据程序的功能说明来设计的;如用白盒法,则需要了解程序内部的结构,此时的测试用例是根据程序的内部逻辑来设计的。如果想用白盒法发现程序中所有的错误,则至少必须使程序中每种可能的路径都执行一次。实际上这是不可能的,即使测遍所有的路径,仍不一定能保证符合相应的功能要求。

(4)数据库三级模式体系结构的划分,有利于保持数据库的____【4】____。

(4)【4】数据独立性

知识点:数据库的基本概念:数据库,数据库管理系统,数据库系统

评 析:数据库的三级模式体系结构把数据的组织管理工作进行了划分,把数据的具体组织留给 DBMS 管理,使用户能逻辑地、抽象地处理数据,而不必关心数据在计算机中的具体表示方式与存储方式,保证了数据的独立性。

(5)在关系运算中,查找满足一定条件的元组的运算称之为____【5】____。

(5)【5】选择

知识点: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

评 析:在关系数据库中,选择运算也是一个元运算,关系 R 通过选择运算(并由该运算给出所选择的逻辑条件)后仍为一个关系。这个关系是由 R 中的那些满足逻辑条件的元组所组成。

(6)在 C 语言中(以 16 位 PC 机为例),一个 float 型数据在内存中所占的字节数为 4: 一个 double 型数据在内存中所占的字节数为____【6】____。

(6)【6】8

知识点:数据类型

评 析:在一般 16 位 PC 机中,一个 float 型数据在内存中占 4 个字节(32 位),一个 double 型数据占 8 个字节。

(7)若运行时给变量 x 输入 12,则以下程序的运行结果是____【7】____;

```
main(){
int x, y;
scan( "%d", &x);
y=x>127x+10: x-12;
printf( "%d\n", y);
}
```

(7) 【7】 0

知识点：三目运算符的用法

评 析：三目运算符的形式为(表达式 1)?(表达式 2): (表达式 3)。当(表达式 1)值为真时, (表达式 2)的值为整个表达式的值;当(表达式 1)的值为假时, (表达式 2)的值为整个表达式的值。

题中因为 $x > 12$ 为假, 所以结果为 $x - 12$, 即为 0。

(8)设有以下定义和语句, 则 $*(p+2)+1$ 的值为___【8】___。

```
int[3][2]={10,20,30,40,50,60}, (*p)[2];
```

```
p=a;
```

(8) 【8】 60

知识点：指针数组

评 析：本题定义了一个 3 行 2 列的二维数组 a, 并定义了一个指向两个元素的一维数组指针 p, 让 p 指向二维数组 a 的首行。则代码 $*(p+2)+1$ 中的 p+2 指向二维数组 a 的第三行 a[2], $*(p+2)$ 指向 a[2][0], $*(p+2)+1$ 指向 a[2][1], $*(p+2)+1$ 是引用 a[2][1], 其值是 60。

(9)若要使指针 p 指向一个 double 类型的动态存储单元, 请填空。

```
P=___【9】___malloc(sizeof(double));
```

(9) 【9】 (double*)

知识点：强制类型转换

评 析：函数 malloc 返回的是 void*, 所以若要使指针指向一个 double 类型, 则必须进行类型转换, 类型转换格式为: (数据类型*)。

(10)函数 mycmp(char*s,char*t)的功能是比较字符串 s 和 t 的大小, 当 s 等于 t 时返回 0, 当 $s > t$ 时返回正值, 当 $s < t$ 时返回负值。请填空。

```
mycmp(char*s,char*t)
{ while(*S==*t)
{ if(*s== '\0')return 0;
++s; ++t;
}
return(___【10】___);
}
```

(10) 【10】 *s-*t 或-*t+*S

知识点：比较字符串的大小

评 析：函数的两个形参是两个字符指针变量, 字符串的首地址是指针变量的值, *s 和 *t 表示字符串数组 s 和 t 的第一个字符, 在 while 循环中, 是通过 s 和 t 值的不断变化来改变指针所指向的字符, 要改变 s 和 t 所指不同字符的 ASCII 码的差值, 必须使用 “*s-*t” 得出。

(11)下列程序的运行结果是___【11】___。

```
main()
{ union EXAMPLE
{struct
{ int x; int y; }in;
```

```

int a;
int b;
}e;
e. a=1; e. b=2;
e. in. x=e. a*e. b;
e. in. y=e. a+e. b;
printf(" %d, %d\n", e. in. x,e. in. y);
}

```

(11) 【11】 4,8

知识点：共用体

评 析：c 语言规定，共用体中各个成员变量在内存中占用同一段内存单元，即 e. in. x 和 e. a、e. b 占用同一段内存单元；

执行语句 e. b=2 后，e. in. x=e.a=e. b=2；

执行语句 e. in. x=e. a+e. b 时，e. a 的值为 2，e. b 的值为 2，所以运算得出 e. in. x 的值为 4；

执行语句 e. in. y=e. a+e. b 时，由于 e. in. x 和 e. a、e. b 占用的是同一段内存单元，所以此时的 e. a、e. b 的值为 4，因此运算得出 e. in. y 的值为 4+4=8。程序最后的打印结果为“4，8”。

(12)在对文件进行操作的过程中，若要求文件的位置回到文件的开头，应当调用的函数是 ____ 【12】 ____ 函数。

(12) 【12】 rewind 或 fseek

知识点：文件的定位

评 析：在对文件进行操作的过程中，若要求文件的位置回到文件的开头，可以调用以下两个函数来完成：

①rewind 函数；

②fseek 函数。

其中 rewind 函数的作用是将位置指针重新返回文件的开头，这个函数没有返回值；而 fseek 函数可以实现文件指针的定位，它的调用方式如下：

fseek(文件类型，位移量，起始点)。起始点用 0、1、2 表示，0 表示“文件开始”，1 表示“文件当前位置”，2 表示“文件末尾”。

(13)以下函数用于求出一个 2×4 矩阵中的最大元素值。

```

max_value(arr)
int arr[][4];
{
    int i,j, max;
    max=arr[0][0];
    for(i=0; i<2; i++)
    for(j=0; ____ 【13】 ____; j++)
    if(____ 【14】 ____>max)max=____ 【15】 ____;
    return(max);
}

```

(13) 【13】 j<4

【14】 arr[i][j]

【15】 arr[i][j]

知识点：二维数组的应用

评 析：2*4 矩阵实际上就是一个二维数组，结合本题应为 `arr[2][4]`，所以在第一个循环中应当填写“4”作为循环的终止条件；第二个横线处是要判断当前的数组元素值是否大于 `max`，所以应当填写 `arr[i][j]`；第三个横线处是要将当前大于 `max` 的数组元素的值赋给变量 `max`，所以应当填写“`arr[i][j]`”。

(14)在先左后右的原则下，根据访问根结点的次序，二叉树的遍历可以分为三种：前序遍历、____【1】____遍历和后序遍历。

(14)【1】中序

知识点：二叉树的前序、中序和后序遍历

评 析：在先左后右的原则下，根据访问根结点的次序，二叉树的遍历可以分为三种：前序遍历、中序遍历和后序遍历。

前序遍历是指在访问根结点、遍历左子树与遍历右子树这三者中，首先访问根结点，然后遍历左子树，最后遍历右子树；并且遍历左、右子树时，仍然先访问根结点，然后遍历左子树，最后遍历右子树。中序遍历指在访问根结点、遍历左子树与遍历右子树这三者中，首先遍历左子树，然后访问根结点，最后遍历右子树；并且遍历左、右子树时，仍然先遍历左子树，然后访问根结点，最后遍历右子树。后序遍历指在访问根结点、遍历左子树与遍历右子树这三者中，首先遍历右子树，然后访问根结点，最后遍历左子树；并且遍历左、右子树时，仍然先遍历右子树，然后访问根结点，最后遍历左子树。

(15)结构化程序设计方法的主要原则可以概括为自顶向下、逐步求精、____【2】____和限制使用 `goto` 语句。

(15)【2】模块化

知识点：结构化程序设计

评 析：结构化程序设计方法的主要原则可以概括为自顶向下、逐步求精、模块化和限制使用 `goto` 语句。自顶向下：程序设计时，应先考虑总体，后考虑细节；先考虑全局目标，后考虑局部目标。不要一开始就过多追求众多的细节，先从最上层总目标开始设计，逐步使问题具体化。逐步求精：对复杂问题，应设计一些子目标作过渡，逐步细化。模块化：一个复杂问题，肯定是由若干稍简单的问题构成。模块化是把程序要解决的总目标分解为分目标，再进一步分解为具体的小目标，把每个小目标称为一个模块。限制使用 `goto` 语句。

(16)软件测试是保证软件质量的重要手段，而软件测试的主要和重要的测试方法是通过测试数据和____【3】____的设计来实现。

(16)【3】测试实例

知识点：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

评 析：进行软件测试时，应精心设计测试实例和选择测试数据，以对系统进行全面测试。

(17)数据库系统的三级模式分别为____【4】____模式、内部级模式与外部级模式。

(17)【4】概念或概念级

知识点：数据库的基本概念：数据库，数据库管理系统，数据库系统

评 析：数据库系统在其内部具有三级模式及二级映射，三级模式分别是概念级模式、内部级模式和外部级模式。

概念模式是数据库系统中全局数据逻辑结构的描述，是全体用户(应用)公共数据视图。

内模式又称物理模式，它给出了数据库物理存储结构与物理存取方法，如数据存储的文件结构、索引、集簇及 `hash` 等存取方式与存取路径，内模式的物理性主要体现在操作系统及文件级上，它还未深入到设备级上(如磁盘及磁盘操作)。

外模式也称子模式或用户模式，它是用户的数据视图，也就是用户所见到的数据模式，它由

概念模式推导而出。

(18)数据字典是各类数据描述的集合，它通常包括 5 个部分，即数据项、数据结构、数据流、**【5】**和处理过程。

(18) **【5】** 数据存储

知识点：数据库设计方法和步骤

评 析：数据字典是各类数据描述的集合，它通常包括 5 个部分，即数据项，是数据的最小单位；数据结构，是若干数据项有意义的集合；数据流，可以是数据项，也可以是数据结构，表示某一处理过程的输入或输出；数据存储，处理过程中存取的数据，常常是手工凭证、手工文档或计算机文件；处理过程。

(19)下面程序的输出是 **【6】** 。

```
main()
{int arr[10], i, k=0;
for(i=0; i<10; i++)
arr[i]=i;
for(i=1; i<4; i++)
k+=arr[i]+i;
printf( " %d\n ",k); }
```

(19) **【6】** 12

知识点：数组

评 析：本题通过第一个 for 循环将数组 arr[0]~arr[9]分别赋值为 0~9，通过第二个 for 循环的三次循环累加，求出结果为 12，具体分析如下：

i=1: k=0+arr[1]+1 即 k=2;

i=2: k=2+arr[2]+2 即 k=6;

i=3: k=6+arr[3]+3 即 k=12。

(20)若 a=10, b=20, 则表达式!(a<b)的值是 **【7】** 。

(20) **【7】** 0

知识点：逻辑非运算

评 析：已知 a=10, b=20, 所以逻辑表达式 a<b 的值为 true, 即为 1, 在这个表达式前面有一个逻辑运算符!, 表示反操作, 所以整个语句的值应当为 false, 即为 0。

(21)下面程序的输出是 **【8】** 。

```
main()
{enum em{em1=3,em2=1,em3};
char*aa[]={ "AA", "BB", "CC", "DD" };
printf( " %s%s%s\n ", aa[em1], aa[em2], aa[em3]);
}
```

(21) **【8】** DDBBCC

知识点：对枚举概念的理解

评 析：c 语言对枚举的定义规定：在枚举中声明的各个枚举元素，如果没有明确指出某个枚举元素的值，而它的上一个元素存在并有明确值的情况下，则这个枚举元素的值为其上一个元素的值+1。

在本题中，没有明确说明枚举元素 em3 的值，则 em3=em2+1=1+1=2，进而可知，在 printf() 打印函数中，要打印的数组元素是 aa[3]、aa[i]、aa[2]，因此最后的打印结果应当为“DDBBCC”。

(22)若想通过以下输入语句使 a=5.0, b=4, c=3, 则输入数据的形式应该是 **【9】** 。

```
int b, c; float a;
scanf( " %e%d, c=%d ", &a, &b, &c)
```

(22) 【9】 5.0,4,c=3

知识点：格式输入函数 scanf()

评 析：scanf(格式控制，地址表列)，如果在“格式控制”字符串中除了格式说明以外还有其他字符，则在输入数据时应输入与这些字符相同的字符。所以此题中输入数据的形式是 5.0,4,c=3。

(23)下列程序的输出结果是____ 【10】 ____。

```
int t(int x,int y,int cp,int dp)
{ cp=x*x+y*y;
  dp=x*x-y*y;
}
main()
{ int a=4, b=3, c=5, d=6;
  t(a,b,c,d);
  printf( " %d%d\n ",c,d);
}
```

(23) 【10】 5 6

知识点：函数的调用

评 析：本题中 a, b, c, d 是实参，x, y, cp, dp 是形参。c 语言规定，实参变量对形参变量的数据传递是“值传递”，即单向传递，只由实参传给形参，而不能由形参传回来给实参。在内存中，实参单元与形参单元是不同的单元。在调用函数时，给形参分配存储单元，并将实参对应的值传递给形参，调川结束后，形参单元被释放，实参单元仍保留并维持原值。因此，程序的输出结果是 5 6。

(24)音以下定义和语句，则 sizeof(a)的值是____ 【11】____，而 sizeof(a. share)的值是____ 【12】____。

```
struct date
{ int day;
  int month;
  int year;
  union { int share1;
         float share2;
        }share;
}a;
```

(24) 【11】 10

【12】 4

知识点：结构体变量所占内存长度

评 析：结构体变量所占内存长度是各成员占的内存长度之和。每个成员分别占有其自己的内存单元。int

占 2 个字节，float 占 4 个字节，共用体变量所占的内存长度等于最长的成员的长度。所以，sizeof(a. share)的值是 4，sizeof(a)的值是 2+2+2+4=10。

(25)下述函数用于统计一行字符中的单词个数，单词之间用空格分隔。

```
word__num(str)
char str[];
```

```

{int i,num=0,word=0;
for(i=0; str[i]!='\0'; i++)
if(str[i]!=' ')word++;
else if(word==0)
{
word=1;
num++;
}
return(num);
}

```

(25) 【13】 ‘\0’ 或 0 或 NULL

【14】 str[i]

【15】 num++ 或 num=num+1 或 num+=1

知识点：字符数组的应用

评 析:观察题目要求，可以知道以下几点：

①for 循环的结束条件应当是：str[i] 已是字符串的最后一个字符；

②str[i] 代表字符串 str 中的第 i+1 个字符；

③整型变量 num 的值是要记录的单词的个数。

c 语言中规定字符串的最后一个字符是一个隐含的字符串结束符 “\0”，所以在题中第一个空中应填写 “\0”；题中第二个空应填写 “str[i]”，以判断当前位置的字符是否为空格；题中第三个空中应当填写 “num++”，通过变量 num 的加 1 累加得到字符串中的单词个数。

(26) 设一棵完全二叉树共有 500 个结点，则在该二叉树中有 ____ 【1】 ____ 个叶子结点。

(26) 【1】 250

知识点：二叉树的概念

评 析:所谓完全二叉树是指除最后一层外，每一层上的结点数均达到最大值；在最后一层上只缺少右边的若干结点。

具有 n 个结点的完全二叉树，其父结点数为 $\text{int}(n/2)$ ，而叶子结点数等于总结点数减去父结点数。本题 $n=500$ ，故父结点数等于 $\text{int}(500/2)=250$ ，叶子结点数等于 $500-250=250$ 。

(27) 在最坏情况下，冒泡排序的时间复杂度为 ____ 【2】 ____。

(27) 【2】 $n(n-1)/2$ 或 $O(n(n-1)/2)$

知识点：算法的概念

评 析:冒泡排序法是一种最简单的交换类排序方法，它是通过相邻数据元素的交换逐步将线性表变成有序。

假设线性表的长度为 n，则在最坏的情况下，冒泡排序需要经过 $n/2$ 遍的从前往后的扫描和 $n/2$ 遍的从后往前的扫描，需要的比较次数为 $n(n-1)/2$ 。

(28) 面向对象的程序设计方法中涉及的对象是系统中用来描述客观事物的一个 ____ 【3】 ____。

(28) 【3】 实体

知识点：面向对象程序设计的概念

评 析:面向对象的程序设计方法中涉及的对象是系统中用来描述客观事物的一个实体，是构成系统的一个基本单位，它由一组表示其静态特征的属性和它可执行的一组操作组成。

(29) 软件的需求分析阶段的工作，可以概括为四个方面： ____ 【4】 ____、需求分析、编写需求规格说明书和需求评审。

(29) 【4】 需求获取

知识点：软件设计的软件需求分析阶段

评 析：软件的需求分析阶段的工作，可以概括为四个方面：需求获取、需求分析、编写需求规格说明书和需求评审。

需求获取的目的是确定对目标系统的各方面需求。涉及到的主要任务是建立获取用户需求的方法框架，并支持和监控需求获取的过程。

需求分析是对获取的需求进行分析和综合，最终给出系统的解决方案和目标系统的逻辑模型。

编写需求规格说明书作为需求分析的阶段成果，可以为用户、分析人员和设计人员之间的交流提供方便，可以直接支持目标软件系统的确认，又可以作为控制软件开发进度的依据。

需求评审是对需求分析阶段的工作进行的复审，验证需求文档的一致性、可行性、完整性和有效性。

(30) ____ 【5】 ____是数据库应用的核心。

(30) 【5】 数据库设计

知识点：数据库设计概念

评 析：数据库设计是数据库应用的核心。在数据库应用系统中的一个核心问题就是设计一个能满足用户要求，性能良好的数据库，这就是数据库设计。

(31)若有以下定义，则不移动指针 p，且通过指针 p 引用值为 98 的数组元素的表达式是 ____ 【6】 ____。

int w[10]={23,54,10,33,47,98,72,80,61}, *p=w;

(31)若有以下定义，则不移动指针 p，且通过指针 p 引用值为 98 的数组元素的表达式是 ____ 【6】 ____。

int w[10]={23,54,10,33,47,98,72,80,61}, *p=w;

(31) 【6】 +(p+5)或 p[5]

知识点：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

评 析：本题的考查点是通过指针引用数组元素。

98 是数组 w 的第 5 个元素(最开始的为第 0 个)，而通过 *p=w 已经将 p 指向了数组 w，要想不移动指针 p 而引用 98。可以有以下两种方法：p[5]、*(p+5)。

(32)以下程序的输出结果是 ____ 【7】 ____。

```
#define MAX(x,y)(x)>(y)?(x): (y)
```

```
main()
```

```
{
```

```
int a=5, b=2,c=3, d=3, t;
```

```
t=MAX(a+b, c+d)*10;
```

```
printf( " %d\n " , t);
```

```
}
```

(32) 【7】 7

知识点：宏定义：不带参数的宏定义；带参数的宏定义

评 析：本题的考查点是宏定义。宏替换后，表达式 t=MAX(a+b, c+d)*10 即变为：

t=(a+b)>(c+d)?(a+b): (c+d)*10; 由于(a+b)>(c+d)为真， 所以 t=(a+b)，即为 7。

(33)若已经定义 int a=25,b=14,c=19;，以下三目运算符(?:)所构成的语句的执行结果是 ____ 【8】 ____。

```
a++<=2&&b--<=2&&c++?printf( " * * *
```

```
a=%d, b=%d, c=%d\n " , a, b, c): printf( " a=%d, b=%d, c=%d\n " , a, b, c);
```

(33) 【8】 a=26,b=14, c=19

知识点：位运算符的含义及使用

评 析：本题的考查点是三目运算符。

这种条件表达式的一般形式为：表达式 1 ? 表达式 2 : 表达式 3，三目运算符的执行顺序：先求解表达式 1，若非 0 则求解表达式 2，此时表达式 2 的值就作为整个表达式的值，若表达式 1 的值为 0，则求解表达式 3，此时表达式 3 的值就作为整个表达式的值。c 语言在做一系列表达式相与，从左到右分别计算各表达式的值，一旦遇到一个表达式为 false，则剩余的表达式都不再进行运算。该题中，`a++<=2` 的值为 false，

则 `b--<=2` 和 `c++` 就都不再运算。

(34)以下程序中用户由键盘输入一个文件名，然后输入一串字符(用#结束输入)存放到 此文件中，形成文本文件，并将字符的个数写到文件的尾部。请填空。

```
#include<stdio. h>

main()
{ FILE *fp;
char ch,filename[32]; int count=0;
printf( " Input the filename :  " );
scanf( " %s " ,filename);
if((fp=fopen(____【9】____, " w+ " ))==NULL)
{ printf( " Can't open file: %s\n" ,filename); exit(0); }
printf( " Enter data: \n " );
while((ch=getchar())!='#')
{ fputc(ch,fp); count++; }
fprintf(____【10】____, " \n%d\n " ,count);
fclose(fp);
}

(34)【9】 filename
【10】 fd
```

知识点：文件的打开与关闭

评 析：本题的考查点是对文件的操作。

`fopen()`函数实现打开文件的功能，通常的调用方式为：`FILE *fp;fp=fopen(文件名,使用文件方式)`，因此，第一个横线处要求填写要打开文件的名字 `filename`。`fprintf()`函数的一般调用方式为：`fprintf(文件指针,格式字符串,输出表列)`，所以第二个横线处应填写 `fp`。

(35)下面的函数 `strcat(str1,str2)`实现将字符串 `str2` 拼接到字符串 `str1` 后面的功能。请填空使之完整。

```
char*strcat(str1,str2)
char*str1,*str2;
{ char*t=str1;
while(____【11】____)str1++;
while(____【12】____);
return(t); }

(35)【11】 *str1!='\0' 或 *str1==0 或 *str1==NULL
【12】 *str1++=*str2++或 *str1++=*str2,*str2++
```

知识点：字符串与字符数组

评 析：本题的考查点是字符串连接函数 `strcat()`。

函数 `strcat(str1, str2)`实现将字符串 `str2` 连接到字符串 `str1` 后面，所以首先要找到字符串 `str1`

的串尾，根据 c 语言的语法规则，一个串的串尾一定是一个隐含字符 “\0”，而在程序中，对字符串中字符的访问是通过两个指针变量来完成的，因此要找到字符串 str1 的串尾，要判断 *str1 是否为 “\0”，要找到字符串 str2 的串尾，要判断 *str2 是否为

“\0”，程序中必须可以使字符串中字符逐一顺序体现，所以在题中我们填写 “*str1” 和 “*str1++=*str++”。

(36) 以下函数 fun 用于求两个整数 a 和 b 的最大公约数。

```
fun(a,b)
int a,b;
{int i,j, m, n;
if(a>b)
{m=a; a=b; ____【13】____; }
i=a; j=b;
while((n=____【14】____)!=0)
{j=i; i=____【15】____; }
return(i);
}
```

(36) 【13】 b=m

【14】 j%i

【15】 n

知识点：while 和 do-while 循环结构

评 析：本题的考查点是求最大公约数问题。

函数中的变量 i 和 j 分别存放两个形参的最小数和最大数，在语句 i=a 和 j=b 执行之前，要将 a 和 b 分别变为最小数和最大数，当 a>b 时，程序使用中间变量将两者交换，所以题中第一个空中应当填写 “b=m”；为求两个整数的最大公约数，程序的 while 循环条件必须为 j 整除 i 成立，所以题中第二个空中应当填写 “j%i”；程序要通过 i 的值返回最大公约数，因此在循环中要将每次整除的公约数赋给 i，以保证最后返回的是两个整数的最大公约数，所以第三个空中应当填写 “n”。

(37) 在运算过程中，能够使空表与非空表的运算统一的结构是 ____【1】____。

(37) 【1】 循环链表

知识点：线性单链表、双向链表与循环链表的结构及其基本运算

评 析：在链表的运算过程中，采用链接方式即循环链表的结构把空表与非空表的运算统一起来。循环链表具有两个特点：

①在循环链表中增加了一个表头结点，其数据域为任意或根据需要来设置，指针域指向线性表的第一个元素的结点。循环链表的头指针指向表头结点。

②循环链表中最后一个结点的指针不是空，而是指向表头结点。

(38) 软件工程研究的内容主要包括： ____【2】____ 技术和软件工程管理。

(38) 【2】 软件开发

知识点：软件工程的基本概念

评 析：基于软件工程的目标，软件工程的理论和技术性研究的内容主要包括：软件开发技术和软件工程管理。

软件开发技术包括：软件开发方法学、开发过程、开发工具和软件工程环境，其主体内容是软件开发方法学。

软件工程管理包括：软件管理学、软件工程经济学、软件心理学等内容。

(39) 与结构化需求分析方法相对应的是 ____【3】____ 方法。

(39) 【3】 结构化设计

知识点：程序设计方法

评 析：与结构化需求分析方法相对应的是结构化设计方法。结构化设计就是采用最佳的可行方法设计系统的各个组成部分以及各个成分之间的内部联系的技术。也就是说，结构化设计是这样—个过程，它决定用哪些方法把哪些部分联系起来，才能解决好某个具体且有清楚定义的问题。

(40)关系模型的完整性规则是对关系的某种约束条件，包括实体完整性、__【4】__和自定义完整性。

(40) 【4】 参照完整性

知识点：关系数据库的数据约束定义

评 析：关系模型允许定义三类数据约束，它们是实体完整性、参照完整性以及用户定义的完整性约束，其中前两种完整性约束由关系数据库系统自动支持。

实体完整性约束要求关系的主键中属性值不能为空，这是数据库完整性的最基本要求，因为主键是惟一决定元组的，如为空则其惟一性就成为不可能的了。

参照完整性约束是关系之间相关联的基本约束，它不允许关系引用不存在的元组：即在关系中的外键要么是所关联关系中实际存在的元组，要么是空值。

自定义完整性是针对具体数据环境与应用环境由用户具体设置的约束，它反映了具体应用中数据的语义要求。

(41)数据模型按不同的应用层次分为三种类型，它们__【5】__数据模型、逻辑数据模型和物理数据模型。

(41) 【5】 概念

知识点：关系数据库的数据模型

评 析：数据模型按不同的应用层次分为3种类型，它们是概念数据模型、逻辑数据模型和物理数据模型。概念数据模型简称概念模型，它是一种面向客观世界、面向用户的模型；它与具体的数据库管理系统无关。

逻辑数据模型又称数据模型，它是一种面向数据库系统的模型，该模型着重于在数据库系统一级的实现。

物理数据模型又称物理模型，它是一种面向计算机物理表示的模型，此模型给出了数据模型在计算机上物理结构的表示。

(42)表示“整数x的绝对值大于5”时值为“真”的C语言表达式是__【6】__。

(42) 【6】 $x > 5 \parallel x < -5$ 或 $x < -5 \parallel x > 5$

知识点：逻辑表达式的应用

评 析：要使整数x的绝对值大于5，则x大于5或x小于-5，用C语言表示即为： $x > 5 \parallel x < -5$ 或简写 $x < -5 \parallel x > 5$ 。

故本题答案为： $x > 5 \parallel x < -5$ 或简写 $x < -5 \parallel x > 5$ 。

(43)以下程序的输出结果是__【7】__。

```
main()
{ unsigned short a=65536; int b;
printf(" %d\n", b: a);
}
```

(43) 【7】 0

知识点：数据类型及C定义方法

评 析：对于一个 unsigned short 来说，它能取的最大值是 65535。这里给 a 赋值 65536，已经超出了它的取值范围，这样它的高位将被截掉，只把低位赋给它(全零)。所以 a 的值实际

为 0。

(44)若有定义: int a=10, b=9, c=8;, 接着顺序执行下列语句, 变量 b 中的值是__【8】__。

```
c=(a-=(b-5));
```

```
c=(a%11)+(b=3);
```

(44)【8】3

知识点:表达式及其变量

评 析: 这里: b-5=4, a-=4 之后 a 的值为 6, 并把它赋给 c。此时 b 的值并未被改变。在第 2 个表达式中, 也只有赋值表达式 b=3 改变了 b 的值。所以两行语句执行完毕, b 的值应该为 3。

(45)函数 pi 的功能是根据以下近似公式求 π 值:

$$(\pi^2/6) = 1 + 1/(2*2) + 1/(3*3) + \dots + 1/(n*n)$$

现在请你在下面的函数中填空, 完成求 π 的功能。

```
#include "math. h "
```

```
{ double s=0. 0; long i;
```

```
for(i=1; i<=n; i++)s=s+__【9】__;
```

```
return(sqrt(6*s));
```

```
}
```

(45)【9】1. 0 / (i*i)或(double)1 / (i*i)

知识点:for 循环语句

评 析:表达式 $1 + 1/(2*2) + 1/(3*3) + \dots + 1/(n*n)$ 可写为: $1/(1*1) + 1/(2*2) + 1/(3*3) + \dots + 1/(n*n)$

即: $\sum 1/(i \times i)$

i=1

对应的 c 语言的表达式: s=s+1. 0 / (i*i)。注: 必须要写成 1. 0 的形式, 否则 1 / (i*i) 返回值为其值截尾取整, 即恒为 0。

(46)若输入字符串: abcde↵, 则以下 while 循环体将执__【10】__次。

```
while((ch=getchar())=='e')printf(" * ");
```

(46)【10】0

知识点: while 循环语句

评析: 函数 getchar()是从键盘得到用户输入的一个字符。用户输入的第 1 个字符是 a, 不管后面输入的是什么 ch 的值都是 'a', 因此条件(ch=getchar())=='e'为假, 这个循环不会被执行。

(47)以下函数用来求出两整数之和, 并通过形参将结果传回, 请填空。

```
Void func(int x,int y, __【11】__z)
```

```
{ *z=x+y; }
```

(47)【11】int*

知识点: 函数参数的正确运用

评析:从题中代码可知: z 应该是一个整型指针, 因此应填: int*

(48)若有以下定义, 则不移动指针 p, 且通过指针 p 引用值为 98 的数组元素的表达式是__【12】__。

```
int w[10]={23,54,10,33,47,98,72,80,61}, *p=w;
```

(48)【12】p[5]或*(p+5)

知识点:指针变量的概念

评析:98 是数组 w 的第 5 个元素(最开始的为第 0 个),而通过 *p=w 已经将 p 指向了数组 w,要想不移动指针 p 而引用 98,可以有以下两种写法: p[5]、*(p+5)。

(49)设在主函数中有以下定义和函数调用语句,且 fun 函数为 void 类型;请写出 fun 函数的首部 **【13】** ____。要求形参名为 b。

```
main()
{ double s[10][22];
  int n;
  .....
  fun(s);
  .....
}
```

(49) **【13】** void fun(double b[10][22])

知识点:函数的定义

评析:本题答案: void fun(double h[10][22])

(50)以下程序的输出结果是 **【14】** ____。

```
void fun()
{ static int a=0;
  a+=2; printf( " %d " , a);
}
main()
{ int cc;
  for(cc=1; cc<4; cc++)fun();
  printf( " \n " );
}
```

(50) **【14】** 246

知识点:变量类型的作用域

评析:循环 for(cc=1; cc<4; cc++)被执行了 3 次。在函数 fun 中,由于 a 是 static 型变量,所以函数第 1 次调用后, a=2; 第 2 次调用后, a=4;第 3 次调用后, a=6。

(51)以下程序的输出结果是 **【15】** ____。

```
#define MAX(x,y)(x)>(y)?(x): (y)
main()
{ int a=5, b=2, c=3, d=3, t;
  t=MAX(a+b,c+d)*10;
  printf( " %d\n " , t);
}
```

(51) **【15】** 7

知识点:宏定义

评析:在 c 语言中,宏定义是直接替换的,所以在对表达式 MAX(a+b,c+d)*10 进行或替换后,表达式变为: (a+b)>(c+d)?(a+b): (c+d)*10

在这个条件表达式中, (a + b)>(c+d)为真,所以用 (a+b)的值作为整个表达式的值(否则用 (c+d)*10 的值作为整个表达式的值),而(a+b)的值为 7。所以整个表达式的值为 7。

(52)栈的基本运算有三种:入栈、退栈和 **【1】** ____。

(52) **【1】** 读栈顶元素

知识点:对栈的操作

评析：栈的基本运算有三种：入栈、退栈和读栈顶元素。入栈运算是指在栈顶位置插入一个新元素。这个运算有两个基本操作：首先将栈顶指针进一(即 top 加 1)，然后将新元素插入到栈顶指针指向的位置。退栈运算是指取出栈顶元素并赋给一个指定的变量。这个运算有两个基本操作：首先将栈顶元素(栈顶指针指向的元素)赋给一个指定的变量，然后将栈顶指针退一(即 top 减 1)。读栈顶元素是指将栈顶元素赋给一个指定的变量。这个运算不删除栈顶元素，只是将它的值赋给一个变量。

(53)在面向对象方法中，信息隐蔽是通过对象的___【2】___性来实现的。

(53)【2】封装

知识点：软件工程的基本概念

评析：软件工程的基本原则包括抽象、信息隐蔽、模块化、局部化、确定性、一致性、完备性和可验证性。

信息隐蔽是指采用封装技术，将程序模块的实现细节隐藏起来，使模块接口尽量简单。

(54)数据流的类型有___【3】___和事务型。

(54)【3】变换型

知识点：数据流的概念

评析：典型的数据流类型有两种：变换型和事务型。

变换型：是指信息沿输入通路进入系统，同时由外部形式变换为内部形式，进入系统的信息通过变换中心，经加工处理以后再沿输出通路变换成外部形式离开软件系统。

事务型：在很多软件应用中，存在某种作业数据流，它可以引发一个或多个处理，这些处理能够完成该作业要求的功能，这种数据流就叫做事务。

(55)数据库系统中实现各种数据管理功能的核心软件称为___【4】___。

(55)【4】数据库管理系统或 DBMS

知识点：数据库管理系统的概念

评析：数据库管理系统是数据库的机构，它是一种系统软件，负责数据库中的数据组织、数据操纵、数据维护、控制及保护和数据服务等。数据库管理系统是数据库系统的核心。

(56)关系模型的数据操纵即是建立在关系上的数据操纵，一般有___【5】___、增加、删除和修改四种操作。

(56)【5】查询

知识点：对关系模型数据的操作

评析：关系模型的数据操纵即是建立在关系上的数据操纵，一般有查询、增加、删除和修改四种操作。

数据查询：用户可以查询关系数据库中的数据，它包括一个关系内的查询以及多个关系间的查询。

数据删除的基本单位是一个关系内的元组，它的功能是将指定关系内的指定元组删除。

数据增加仅对一个关系而言，在指定关系中插入一个或多个元组。

数据修改是在一个关系中修改指定的元组和属性。

(57)下列程序的输出结果是___【6】___。

```
main()
{
    int a=1, b=2;
    a=a+b; b=a-b; a=a-b;
    printf(" %d, %d\n", a, b);
}
```

(57)【6】2,1

知识点：赋值表达式

评析：在本题中，执行 $a=a+b$ 时， $a=1$ ， $b=2$ ， $a=1+2=3$ ；执行 $b=a-b$ 时， $a=3$ ， $b=2$ ， $b=3-2=1$ ；执行 $a=a-b$ 时， $a=3$ ， $b=1$ ， $a=3-1=2$ 。因此，本题的输出是 2,1。

(58)下列程序的输出结果是 16.00，请填空。

```
main()
{
    int a=9, b=2;
    float x=___【7】___, y=1.1, z;
    z=a / 2+b*x / y+1 / 2;
    printf( " %5. 2f\n ",z);
}
```

(58)【7】6.6

知识点：算术表达式

评析：本题已知结果 $z=16.00$ ，即 $16.00=a / 2+b*x / y+1 / 2$ ，分别把 a 、 b 代入，得到 $16.00=9 / 2+2*x / 1.1+1 / 2$ 。

因为 $a=9$ ， a 是整型，所以 $9 / 2$ 的值在没有进行类型转换时，等于 4， $1 / 2$ 同理，等于 0。

代入求一元一次方程，很容易就可以得出 $x=6.6$ 。

(59)用以下语句调用库函数 malloc，使字符指针 st 指向具有 11 个字节的动态存储空间，请填空。

```
St=(char*)___【8】___;
```

(59)【8】malloc(11)

知识点：动态分配内存

评析：malloc 函数的格式是：void*(或 char*)malloc(size)，size 表示应分配的存储区。此题要分配 11 字节的单元，把 11 代入即可。

(60)下列程序段的输出结果是___【9】___。

```
main()
{char b[]=" Hello,you ";
  b[5]=0;
  printf( " %s\n ", b);
}
```

(60)【9】Hello

知识点：字符数组

评析：字符串中，数值 0 或符号 '\0' 表示字符串的结束。本题中， $b[5]$ 被赋了 0 值，表明字符串 b 的第六个字符就是结束标志。因此，只输出前五个字符，即 Hello。

61)下列程序段的输出结果是___【10】___。

```
int n= 'c':
switch(n++)
{default: print " error " , }; break;
case 'a':
case 'A':
case 'b':
case 'B': printf( "good" ); break;
case 'c': case 'C': printf( "pass" );
case 'd': case 'D': printf( "wam" );
```

}

(61) 【10】 passwarn

知识点: Switch 语句的使用

评析: `n++`是在执行完其所在的语句后再加1, 因此, 在执行 `case` 的时候, `n` 的值依然为 'c', 执行 `case 'c'` 后面的语句, 先打印出 "pass"; 在执行完 `case 'c'` 后, 未遇到 `break` 跳出 `switch`, 便接着执行下面的语句, 又打印出 `warn`。所以此题输出结果是: `passwarn`。

(62) 以下程序通过函数指针 `p` 调用函数 `fun`, 请在填空栏内, 写出定义变量 `p` 的语句。

```
void fun(int*x,int*y)
```

```
{ ..... }
```

```
main()
```

```
{ int a=10,b=20;
```

```
___ 【11】 ___:
```

```
p=fun;  p(&a, &b);
```

```
.....
```

}

(62) 【11】 `void(*p)();`

知识点: 函数指针的定义

评析: 要定义 `p` 为指向函数的指针变量, 需写成: `void(*p)();`, 注意 `*p` 两侧的括弧不可以省略, 表示 `p` 先与 `*` 结合, 是指针变量, 再与后面的 `()` 结合, 表示此指针变量指向函数, 这个函数无返回值(同 `fun` 函数)。

(63) 下列程序的输出结果是___ 【12】 ___。

```
void fun(int*n)
```

```
{
```

```
while((*n)--);
```

```
printf( " %d " , ++(*n));
```

```
}
```

```
main()
```

```
{ int a=100;
```

```
fun(&a);
```

```
}
```

(63) 【12】 0

知识点: 综合应用

评析: 在函数 `fun()` 中, `while((*n)--)` 是先引用 `*n` 的值, 再做 `(*n)--` 运算, 所以循环结束时 `*n` 的值为 0,

再做 `(*n)--` 运算后, `*n` 的值为 -1; 执行 `++(*n)` 后, `*n` 的值是 0。

(64) 以下程序的输出结果是___ 【13】 ___。

```
main()
```

```
{
```

```
int arr[]={30,25,20,15, 10,5}, *p=arr;
```

```
p++;
```

```
printf( " %d \n " , *(p+3));
```

```
}
```

(64) 【13】 10

知识点: 综合应用

评析：整型指针最初被赋的值是数组 arr 的头指针，即指向数组的第一个元素 30，p++后，指针指向数组的下一个元素，即 *p=arr[1]=25，在执行 *(p+3) 时，则相当于 a[1+3]，即 10。因此，输出应为 10。

(65) 下面程序把从终端读入的文本(用 @ 作为文本结束标志)输出到一个名为 bi. dat 的新文件中，请填空。

```
#include "stdio. h "
FILE *fp;
main()
{ char ch;
if((fp=fopen(__【14】__))==NULL)exit(0);
while((ch=getchar())!= '@')fputc(ch,fp);
fclose(fp);
}
```

(65) 【14】 “bi. Dat”，“w”，或 “bi. dat”，“w+” 或 “bi. dat”，“r+”

知识点：文件的操作

评析：本题的考点是 fopen() 函数，此函数的格式是 fopen(文件名，使用文件方式)，题目的要求是将从键盘读入的文本写到文件中，因此选择的打开方式应该是 “w” (只写)。或者是读写 “r+” 或 “w+”。

(66) 若变量 n 中的值为 24，则 prnt 函数共输出__【15】__行。

```
void prnt(int n, int aa[])
{ int i;
for(i=1; i<=n; i++)
{ printf( " %6d ",aa[i]);
if(!(i%5))printf( " \n " );
}
printf( " \n " );
}
```

(66) 【15】 5

知识点：综合应用

评析：本题的关键在于 if(!(i%5))printf(“\n”); 这句，即如果 i 可被 5 整除时，则换行。i 值为 24，可以被 5 整除 4 次，并余 4，因此，prnt 函数共输出 5 行，最后一行有 4 个数。

(67) 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__【1】__。

(67) 【1】 空间复杂度和时间复杂度

知识点：算法的各种属性及其相互作用

评析：算法的复杂性是指对一个在有限步骤内终止算法和所需存储空间大小的估计。算法所需存储空间大小是算法的空间复杂度，算法的计算量是算法的时间复杂性。

(68) 数据结构包括数据的逻辑结构、数据的__【2】__以及对数据的操作运算。

(68) 【2】 存储结构

知识点：数据结构包含的 3 个方面及各方面之间的关系。

评析：数据结构包括 3 个方面，即数据的逻辑结构、数据的存储结构及对数据的操作运算。

(69) 一个类可以从直接或间接的祖先中继承所有属性和方法。采用这个方法提高了软件的__【3】__。

(69) 【3】 可重用性

知识点：继承的优点

评析：继承的优点：相似的对象可以共享程序代码和数据结构，从而大大减少了程序中的冗余，提高软件的可重用性。

(70)面向对象的模型中，最基本的概念是对象和____【4】____。

(70)【4】类

知识点：面向对象方法的基本概念

评析：面向对象模型中，最基本的概念是对象和类。对象是现实世界中实体的模型化；将属性集和方法集相同的所有对象组合在一起，可以构成一个类。

(71)软件维护活动包括以下几类：改正性维护、适应性维护、__【5】__维护和预防性维护。

(71)【5】完善性

知识点：软件维护的分类

评析：软件维护活动包括以下几类：改正性维护、适应性维护、完善性维护和预防性维护。

改正性维护

是指在软件交付使用后，为了识别和纠正软件错误、改正软件性能上的缺陷、排除实施中的误使用，应当进行的诊断和改正错误的过程；适应性维护是指为了使软件适应变化，而去修改软件的过程；完善性维护是指为了满足用户对软件提出的新功能与性能要求，需要修改或再开发软件，以扩充软件功能、增强软件性能、改进加工效率、提高软件的可维护性；预防性维护是为了提高软件的可维护性、可靠性等，为以后进一步改进软件打下良好基础。

(72)语句：x++、++x、X=X+1、x=1+x；，执行后都使变量 x 中的值增 1，请写出一条同一功能的赋值语句(不得与列举的相同)____【6】____。

(72)【6】x+=1；

知识点：c 语言的语法

评析：在 c 语言中使变量增 1 的方法有：自增运算符与赋值语句。其中自增运算符有前置与后置方式，赋值语句也有两种类型。

本题中的主要增 1 方式有：X++、++X、X=X+1、X=1+X、X+=1；。

(73)设 Y 是 int 型变量，请写出判断 Y 为奇数的关系表达式____【7】____。

(73)【7】Y%2==1 或 Y%2!=0

知识点：c 语言的语法

评析：判断变量是否为奇数可以用变量与 2 取模，判断结果是 1 或不为 0。本题具体做法如下：Y%2==1 或 Y%2!=0。

(74)以上程序运行后的输出结果是____【8】____。

```
main()
{int i=10, j=0;
do
{j=j+i; i--; }
while(i>2);
printf( " %d\n " , j);
}
```

(74)【8】52

知识点：c 语言中的 do-while 语句

评析：do-while 语句的形式为：

```
do
{
语句;
}while(条件表达式)
```

当条件表达式为非零时，继续执行循环体，直到条件表达式为零时退出循环。

(75)设有如下程序：

```
main()
{int n1, n2;
scanf( " %d " , &n2);
while(n2!=0)
{ n1=n2%10;
n2=n2 / 10;
printf( " %d " , n1);
}
}
```

程序运行后，如果从键盘上输入 1298，则输出结果为__【9】__。

(75) 【9】 8921

知识点：c 语言中的循环语句与取模和整除运算符。

评析：本题的程序的功能是将输入的整数反向输出。

(76)以下程序输出的最后一个值是__【10】__。

```
int ff(int n)
{ static int f=1;
f=f*n;
return f;
}
main()
{ int i;
for(i=1; i<=5; i++) printf( " %(1 \n " , ff(i));
}
```

(76) 【10】 120

知识点：c 语言的函数

评析：函数的定义形式为：

函数类型 函数名(形参表)

{

类型说明语句;

执行语句;

}

静态变量的类型说明符是 **static**，静态局部变量属于静态存储方式，它具有以下特点：

①静态局部变量属于静态存储类别，在静态存储区内分配存储单元。在程序整个运行期间都不释放。

②对静态局部变量是在编译时赋初值的，即只赋初值一次，在程序运行时它已有初值。以后每次调用函数时不再重新赋初值而只是保留上次函数调用结束时的值。

③如在定义局部变量时不赋初值的话，则对静态局部变量来说，编译时自动赋初值 0(对数值型变量)或空字符(对字符变量)。(注意：C 语言中的非静态变量在定义时，系统并不会自动给它赋初值)

④虽然静态局部变量在函数调用结束后仍然存在，但其他函数是不能引用它的。

本题中函数的功能是：与 for 语句一起求一个整数的阶乘。

(77)以下函数的功能是：求 x 的 y 次方，请填空。

```
double fun(double x,int y)
{int i;
double z;
for(i=1, z=x; i<y; i++)z=z*___【11】___;
return z;
}
```

(77) 【11】 x

知识点: c 语言中的函数

评析: 函数的定义形式为:

函数类型 函数名(形参表)

```
{
```

类型说明语句:

执行语句:

```
}
```

本题中函数的功能是: 累积变量以求得变量的 y 次方。

(78)以下程序运行后的输出结果是___【12】___。

```
main()
{char s[]=" 9876 ",*p;
for(p=s; p<s+2; p++)printf(" %s\n", p);
}
```

(78) 【12】 9876

876

知识点: c 语言中的指针

评析: 指针是一种数据类型, 这种数据类型的变量用来存放内存中分配的存储单元的首地址。

指针的定义:

类型说明符 t 指针变量名:

(79)若有定义语句: char s[100], d[100]; int j=0, i=0;, 且 s 中已赋字符串, 请填空以实现复制。(注: 不使用逗号表达式)

```
while(s[i]){d[j]=___【13】___; j++; }
d[j]=0;
```

(79) 【13】 s[i++]

知识点: c 语言中的 while 语句与数组

评析: 本题中为了能实现字符串的复制, 需要使字符数组 s 从头到尾依次遍历其所有元素。

本题应使用 i 的自增后置来实现。

(80)若有如下结构体说明:

```
struct STRU
{int a,b; char c; double d;
struct STRU*pl, *p2;
};
```

请填空, 以完成对 t 数组的定义, t 数组的每个元素为该结构体类型。

___【14】___t[20]

(80) 【14】 struct STRU

知识点: c 语言中的结构体

评析: 结构体类型是构造数据类型, 是用户自己定义的一种类型。

结构体类型的定义:

struct 结构体类型名

```
{
成员项表;
};
```

定义结构体变量的形式为:

struct 结构体类型名 变量 1, 变量 2, ...

其中变量包括: 一般变量、指针变量、数组变量等。

(81)以下程序段打开文件后, 先利用 fseek 函数将文件位置指针定位在文件末尾, 然后 调用 ftell 函数返回当前文件位置指针的具体位置, 从而确定文件长度, 请填空。

```
FILE *myf; long fl;
```

```
myf=__【15】__( " test. t ", " rb " );
```

```
fseek(myf(), SEEK_END); fl+ftell(myf);
```

```
fclose(myf);
```

```
printf( " %ld\n " ,fl);
```

(81) 【15】 fopen

知识点: c 语言中的文件操作

评析: c 语言中的文件分为: ASCII 文件与二进制文件。文件在使用前打开, 使用后要关闭。

打开文件的函数为: fopen(), 调用形式为: fp=fopen(" 文件名", " 使用文件方式");

关闭文件的函数为: fclose(), 调用形式为: fclose(fp);, 其中 fp 为文件指针。

(82)算法的基本特征是可行性、确定性、__【1】__和拥有足够的情报。

(82) 【1】 有穷性

知识点: 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)

评析: 算法是指解题方案的准确而完整的描述。它有 4 个基本特征, 分别是可行性、确定性、有穷性和拥有足够的情报。

(83)顺序存储方法是把逻辑上相邻的结点存储在物理位置__【2】__的存储单元中。

(83) 【2】 相邻

知识点: 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。

评析: 常用的存储表示方法有 4 种, 顺序存储、链式存储、索引存储、散列存储。其中, 顺序存储方法是把逻辑上相邻的结点存储在物理位置也相邻的存储单元中。

(84)Jackson 结构化程序设计方法是英国的 M. Jackson 提出的, 它是一种面向__【3】__的设计方法。

(84) 【3】 数据结构

知识点: 结构化程序设计

评析: 结构化分析方法主要包括: 面向数据流的结构化分析方法(SA-Structured analysis), 面向数据结构的 Jackson 方法(JSD-Jackson system development method)和面向数据结构的结构化数据系统开发方法(DSSD-Data structured system development method)。

(85)数据库设计分为以下 6 个设计阶段: 需求分析阶段、__【4】__、逻辑设计阶段、物理设计阶段、实施阶段、运行和维护阶段。

(85) 【4】 概念设计阶段或数据库概念设计阶段

知识点: 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

评析: 数据库设计分为以下 6 个设计阶段: 需求分析阶段、概念设计阶段、逻辑设计阶段、物理设计阶段、实施阶段及数据库运行和维护阶段。

(86)数据库保护分为：安全性控制、___【5】___、并发性控制和数据的恢复。

(86)【5】完整性控制

知识点:数据库的基本概念：数据库，数据库管理系统，数据库系统

评析:考查考生对数据库基本知识的了解。

安全性控制：防止未经授权的用户有意或无意存取数据库中的数据，以免数据被泄露、更改或破坏；完整性控制：保证数据库中数据及语义的正确性和有效性，防止任何对数据造成错误的操作；并发性控制：正确处理多用户、多任务环境下的并发操作，防止错误发生；数据的恢复：当数据库被破坏或数据不正确时，使数据库能恢复到正确的状态。

(87)若从键盘输入 58，则以下程序的输出结果是___【6】___。

```
main()
{ int a;
scanf( " %d " , &a);
if(a>50)printf( " %d " , a);
if(a>40)printf( " %d " , a);
if(a>30)printf( " %d " , a);
}
```

(87)【6】585858

知识点：用 if 语句实现选择结构

评析:本题首先为 a 输入一个值 58，然后执行三个判断语句，在每一个判断中，如果满足判断条件的话，就输出 a，因为三次判断，条件都满足，所以三次输出 a。

(88)以下程序的输出结果是___【7】___。

```
main()
{ int a=177;
printf( " %o\n " , a);
}
```

(88)【7】261

知识点：数据的输入与输出，输入输出函数的调用；不同数制间的转换

评析：本题考点是 printf 函数的格式字符。“%o”表示以八进制整数形式输出。

(89)以下程序的输出结果是___【8】___。

```
main()
{int a=0;
a+=(a=8);
printf( " %d\n " , a);
}
```

(89)【8】16

知识点:C 运算符的种类、运算优先级和结合性

评析:本题主要考的是运算符的优先级。

a+=(a=8)可以写成 a=a+(a=8)的形式，括号的优先级高于“+”，而“+”的优先级又高于“=”，所以先执行括号内的运算，将 a 赋值为 8，然后再执行+运算。

(90)以下定义的结构体类型拟包含两个成员，其中成员变量 info 用来存放整型数据；成员变量 link 是指向自身结构的指针。请将定义补充完整。

```
struct node
{int info;
___【9】___link;
```

```
};
```

(90) 【9】 struct node*

知识点: 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

评析: 本题考的是如何定义一个指向结构体变量的指针.

一个结构体变量指针就是该变量所占据的内存段的起始地址。可以设一个指针变量, 用来指向一个结构体变量, 此时该指针变量的值是结构体变量的起始地址。指针变量也可以用来指向结构体数组中的元素。

(91) 以下程序的输出结果是___ 【10】 ___。

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

(91) 【10】 6

知识点: for 循环结构

评析: 在逗号表达式内按自左至右顺序求解, 因此本题共执行三次循环, 具体情形如下:

第一次当 i=0 时, 执行 i++, s+=i, 则 i=1, s=1;

第二次当 i=1 时, 执行 i++, s+=i, 则 i=2, s=3;

第三次当 i=2 时, 执行 i++, s+=i, 则 i=3, s=6。

当 i=3 后退出循环。

(92) 以下程序的输出结果是___ 【11】 ___。

```
main()
{char*p= " abcdefgh " ,。 r;
long*q;
q=(long*)p;
q++;
r==(char*)q;
printf( " %s\n " , r);
}
```

(92) 【11】 efgh

知识点: 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

评析: 本题考点是指针的指向数据类型。

在本题中指针变量 p, r 指向字符型数据, 而指针变量 q 指向长整型数据。首先通过类型转换将指针 p 的值赋给 q, 然后执行 q 自加, 因为 q 是定义为指向长整型数据的变量, 而长整型数据一般占 4 个字节的空间, 所以实际上, q 的值增加了 4, 当再将 q 的值经过强行类型转换赋给 r 的时候, 实际上这时候 r 是指向了字符 e, 所以最后打印字符串时, 得到的结果是 “efgh”。

(93) 以下程序的输出结果是___ 【12】 ___。

```
main()
{int x=0;
sub(&x,8, 1);
printf( " %d\n " , x);
}
```

```
sub(int *a,int n,int k)
{ if(k<=n)sub(a,n / 2,2*k);
*a+=k;
}
```

(93) 【12】 7

知识点：函数的正确调用，嵌套调用，递归调用

评析：本题是一个递归调用。主函数中调用 sub 函数流程如下：

```
sub(*x, 8, 1)-sub(*x, 4, 2)-sub(*x, 2, 4)-x=x+k=4
||
x=x+k=7 x=x+k=6
```

(94)设有如下宏定义

```
#define MYSWAP(z,x,y) {z=x; x=y; y=z; }
```

以下程序段通过宏调用实现变量 a，b 内容的交换，请填空。

```
float a=5,b=16, c;
MYSWAP(__ 【13】 __, a, b);
```

(94) 【13】 c

知识点：宏定义：不带参数的宏定义；带参数的宏定义

评析：本题最重要的是理解宏 MYSWAP(z,x,y)的作用：通过 z 实现 x，y 的内容交换。所以 MYSWAP(c,a,b)可以通过 c 实现 a，b 内容的交换。

(95)以下程序用来统计文件字符的个数，请填空。

```
#include "stdio. h "
main()
{FILE*fp; long num=0;
if((fp=fopen( " fname. dat " , " r " ))==NULL)
{printf( " Open error\n " ); exit(0); }
while(__ 【14】 __)
{num++; }
printf( " num=%ld\n " ,num);
fclose(fp);
}
```

(95) 【14】 起 fgetc(fp)!=EOF 或!feof(fp)

知识点：文件的读写(fputc, fgetc, fvuts, fgets, fread, fwrite, fprintf,fscanf 函数)，文件的定位(rewind, fseek 函数)

评析：本程序考的是 fgetc 函数。

此函数的功能是从指针变量 fp 指向的文件中读入一个字符，如果执行 fgetc 函数时遇到文件结束符，则函数返回文件结束符 EOF。

(96)以下程序中，select 函数的功能是：在 N 行 M 列的二维数组中，选出一个最大值作为函数值返回，并通过形参传回此最大值所在的行下标，请填空。

```
#define N 3
#define M 3
select(int a[N][M], int*n)
{int i,j, row=0,colum=0;
for(i=0; i<N; i++)
for(j=0; j<M; j++)
```



```

if(a[i][j]>a[row][column]){row=i; column=j; }
*n=row;
return(____【15】__);
}
main()
{int a[N][M]={9, 11,23,6,1, 15, 9, 17,20}, max,n;
max=select(a, &n);
printf( " max=%d,line=%d\n " ),max,n);
}

```

(96) 【15】 a[row][column]

知识点：形式参数与实在参数，参数值的传递

评析：本题通过判断语句，将较大值的行下标赋给 row，列下标赋给 column，循环结束后，a[row][column]

中是数组元素的最大值，所以应该把它的行下标赋给，n，将 a[row][column]的值返回。

(97)测试的目的是暴露错误，评价程序的可靠性；而____【1】____的目的是发现错误的位置并改正错误。

(97) 【1】 调试

知识点:软件测试的目的

评析：软件测试的目标是在精心控制的环境下执行程序，以发现程序中的错误，给出程序可靠性的鉴定。调试也称排错，它是一个与测试有联系又有区别的概念。具体来说，测试的目的是暴露错误，评价程序的可靠性，而调试的目的是发现错误的位置，并改正错误。

(98)在最坏情况下，堆排序需要比较的次数为____【2】____。

(98) 【2】 $O(n\log 2n)$

知识点：堆排序

评析：在最坏情况下，冒泡排序所需要的比较次数为 $n(n-1)/2$ ；简单插入排序所需要的比较次数为 $n(n-1)/2$ ；希尔排序所需要的比较次数为 $O(n^{1.5})$ ；堆排序所需要的比较次数为 $O(n\log 2n)$ 。

(99)若串 s="Program"，则其子串的数目是____【3】____。

(99) 【3】 29

知识点：子串的概念

评析：串 s 中共有 7 个字符，由于串中字符各不相同，则其子串中字符个数为 0 的串有 1 个，即空串；字符个数为 1 的字符串有 7 个；字符个数为 2 的字符串有 6 个；字符个数为 3 的字符串有 5 个；字符个数为 4 的字符串有 4 个；字符个数为 5 的字符串有 3 个；字符个数为 6 的字符串有 2 个；字符个数为 7 的字符串有 1 个；共有 $1+2+3+4+5+6+7+1=29$ 。

(100)一个项目具有一个项目主管，一个项目主管可管理多个项目，则实体 " 项目主管 " 与实体 " 项目 " 的联系属于____【4】____的联系。

(100) 【4】 1 对多或 1: N

知识点：实体间的联系

评析：两个实体集间的联系实际上是实体集间的函数关系，这种函数关系可以有 3 种，即一对一(1: 1)

的联系、一对多(1: N)或多对一(N: 1)的联系和多对多(N: N)的联系。

(101)数据库管理系统常见的数据模型有层次模型、网状模型和____【5】____三种。

(101) 【5】 关系模型

知识点：数据模型

评析：数据库管理系统是位于用户与操作系统之间的一层系统管理软件，是一种系统软件，是用户与数据库之间的一个标准接口，其总是基于某种数据模型，可以分为层次模型、网状模型和关系模型。

(102)以下程序运行后的输出结果是____【6】_____。

```
main()
{ int x=10,y=20,t=0;
  if(x==y)t=x; x=y; y=t;
  printf( " %d, %d\n " , x, y);
}
```

(102)【6】20,0

知识点：赋值语句

评析：在本题给出的程序中，if 条件为假，不执行 t=x; 语句，而去执行 x=y; 和 y=t; 语句，所以打印出 x 的值为 20，y 的值为 0；但是如果程序这样写 {t=x; x=y; y=t; }，那么就是实现 x 与 y 的交换。

(103)以下程序运行后的输出结果是____【7】_____。

```
main()
{ int x=15;
  while(x>10&& x<50)
  {x++;
   if(x / 3){x++; break; }
   else continue;
  }
  printf( " %d\n " , x);
}
```

(103)【7】17

知识点：while 循环语句

评析：当 x=15 时，while 条件为真，执行 x++；，这时 x 的值为 16，if 条件为真，执行 x++；后跳出循环体，此时 x 的值为 17。

(104)有以下程序：

```
#include<stdio. h>
main()
{char c;
  while((c=getchar())!= ' ?' )putchar(--c);
}
```

程序运行时，如果从键盘输入：Y?N? /，则输出结果为____【8】_____。

(104)【8】X

知识点：字符输入函数 getchar()

评析：getchar() 只能接收一个字符，当从键盘输入 Y?N? / 后，系统判断出第一个字符 Y!= ' ?'。条件为真，输出 X，继续循环，这时 while 条件不成立，循环结束。

(105)以下函数的功能是计算 $s=1/1 + 1/2! + 1/3! + \dots + 1/n!$ ，请填空。

```
double fun(int n)
{double s=0. 0,fac=1. 0; int i;
  for(i=1; i<=n; i++)
  {fac=fac____【9】_____;
```

```

s=s+fac;
}
return S;
}

```

(105) 【9】 $1/i$ 或 $1/i$ 或 $1/i$ 或 $(1.0/i)$ 或 $(double)i$

知识点: for 循环语句

评析: 该程序实现的功能是求各个因式的和, 观察表达式可以看出, 只要将 i 的值的倒数乘以 fac 就可得到每个因式的值。

(106) 下面程序的运行结果是: ____ 【10】 ____。

```

#define N 10
#define s(x)x*x
#define f(x)(x*x)
main()
{
    int i, i2;
    i=1000 / s(N); i2=1000 / f(N);
    printf(" %d%d\n", i, i2);
}

```

(106) 【10】 1000 10

知识点: 宏定义

评析: 根据宏定义, 变量 $i1$ 的值为 $1000 / 10*10=1000$, 变量 $i2$ 的值为 $1000 / (10*10)=10$ 。

(107) 下面程序的运行结果是: ____ 【11】 ____。

```

typedef union student
{
    char name[10];
    long sno;
    char sex;
    float score[4];
}STU;
main()
{
    STU a[5];
    printf(" %d\n", sizeof(a));
}

```

(107) 【11】 80

知识点: 共用体变量所占内存长度的计算

评析: 根据共用体变量的特点可知, 共用体变量所占的内存长度等于最长的成员的长度, 所以变量 STU 所占的字节数等于其成员 $score[4]$ 所占的字节数, 即为 16 个字节, 最终打印出共用体数组 $a[5]$ 的长度为 $16*5=80$ 。

(108) 若 fp 已正确定义为一个文件指针, $d1.dat$ 为二进制文件, 请填空, 以便为“读”而打开此文件: $fp=fopen$ (____ 【12】 ____);。

(108) 【12】 “ $d1.dat$ ”, “ rb ”

知识点: 文件的打开函数 $fopen()$

评析: ANSI C 规定了标准输入输出函数库, 用 $fopen()$ 函数来实现打开文件, 其调用方式为 (fp 已定义为一个文件指针): $fp=fopen$ (文件名, 使用文件方式);, “ $d1.dat$ ” 是文件名, “ rb ” 是为输入打开一个二进制文件, 使用方式为 “只读”。

(109) 设有定义: $int\ n, *k=&n$; , 以下语句将利用指针变量 k 读写变量 n 中的内容, 请将语

句补充完整。

```
scanf( " %d " ,____ 【13】 ____);
```

```
printf("%d\n" , ____ 【14】 ____);
```

(109) 【13】 k

【14】 *k

知识点：指针变量的引用

评析：本题中的第一空，要求输入变量 n 的地址，而指针变量 k 指向整型变量 n，故填写 k；第二空要求打印输出变量 n 的值，应该填入 *k。

(110) fun 函数的功能是：首先对 a 所指的 N 行 N 列的矩阵，找出各行中的最大的数，再将这 N 个最大值中的最小的那个数作为函数值返回。请填空。

```
#include<stdio. h>
```

```
#define N 100
```

```
int fun(int(*a)[N])
```

```
{int row,col,max,min;
```

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

```
{for(max=a[row][0], col=1; col<N; col++)
```

```
if(____ 【15】 ____ )max=a[row][col];
```

```
if(row==0)min=max;
```

```
else if(max<min)min=max;
```

```
}
```

```
return min;
```

```
}
```

(110) 【15】 max<a[row][col] 或 max<=a[row][col] 或 a[row][col]>max 或 a[row][col]>=max

知识点：求矩阵最大值的算法

评析：本题空格所在的 if 条件要求判断出每一行中的最大数，应该填写条件 max<a[row][col]。