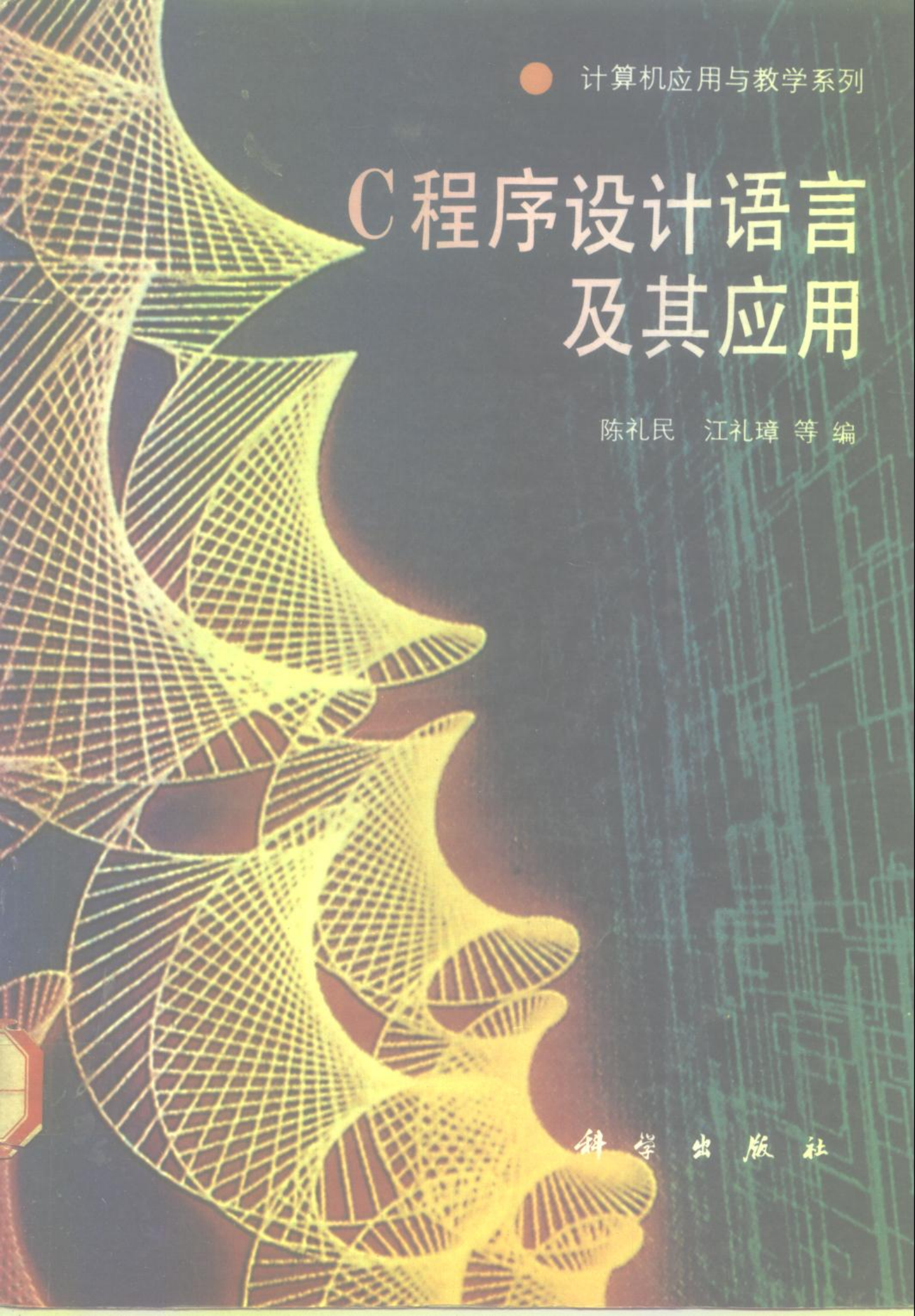

计算机应用与教学系列

# C程序设计语言及其应用

陈礼民　江礼璋　等 编

科学出版社

计算机应用与教学系列

# C程序设计语言及其应用

陈礼民　江礼璋　等　编


科学出版社

1993

(京) 新登字 092 号

## 内 容 简 介

本书是以最新的 ANSI C 标准编写的，能很好地满足目前流行的各种 C 语言版本的要求. 本书详细地介绍了 C 语言的语法及其程序设计方法，主要内容包括数据类型、各种语句、程序结构、程序设计方法、函数、指针、结构、联合、位域、枚举类型、文件及其 I/O，并另辟一章介绍动态分配、链表和树、C 和 dBASE 的接口、图形函数等高级应用.

本书注重实用，书中除了有丰富的上机实例外，还包含了许多 C 语言的工具和实用程序，便于直接引用或在这个基础上做进一步开发. 书后还附有 C 编程中的常见错误和函数库等多种实用的附录，便于参考、查找.

本书适合作为各理工科大专院校的计算机专业教材，也可供其它专业选用；对于广大科技人员和计算机工作者也是一本很好的参考书.



计算机应用与教学系列

**C 程序设计语言及其应用**

陈礼民 江礼建 等 编

责任编辑 冯长芳 刘晓融

科学普及出版社出版

北京东黄城根北街 16 号

邮政编码：100707

国防科工委印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1993年1月第 一 版 开本：787×1092 1/16

1993年1月第一次印刷 印张：16

印数：1—8 000 字数：365 000

ISBN 7-03-003332-9/TP · 246

**定价：12.00 元**

# 前　言

C 语言从 1972 年创建以来已近 20 年了．由于 UNIX 操作系统的蓬勃发展，也由于 C 语言自身的优点，发展到今天，C 语言已广泛为人们所接受，成为当今最受欢迎的主要计算机语言之一．

C 语言从传统 C 语言发展到目前的标准 C 语言，已日趋成熟．ANSI 标准化草案的问世，就是 C 语言趋向规范化的标志．许多以 ANSI C 为范本的 C 编译器已相继问世，尤其是在微机上，Turbo C 和 MS-C 已广为流传，用 ANSI C 替代传统 C 势在必行．目前，形形色色新的 C 语言的材料不少，但作为正式出版的 C 语言教材，绝大部分还是介绍传统 C 的．因此，根据 ANSI C 写一本适合当前要求的 C 语言书籍，是很有必要的．

作者依据以下三个原则编写本书：

(1) 深入浅出．

希望本书有广泛的适应性，即兼顾科学性和通俗性，使它既可以适应广大科技工作者的需要，也可以作为大专院校的教材．书中主要介绍当前流行的 ANSI C，对一些日益被采用的新概念和新内容，如函数原型、流、二进制文件的读写与应用等都做了介绍；也兼顾了对传统 C 有需要的用户，并且进行了特点比较，概念更为明确．在叙述上力求透彻易懂，引入变量的存区、地址等概念，通过图解和程序实例，运用比较以及示范易错写法等种种手段，力求深入浅出以适应读者的不同要求．

(2) 全面和简练．

C 语言发展到现在已广泛地应用于各个领域．作为一本介绍 C 语言及其应用的书籍，既不能等同于一般资料，不分主次地罗列素材，也不能只讲“骨头”，把一大堆实际使用中的问题留给读者自己去摸索和体会．本书在一些主要内容上作了比较全面和深入的讨论，能满足一般应用要求．

(3) 注重实用．

C 语言有丰富的函数库，学会应用函数库和常用例程是学习 C 语言的一个重要内容．限于篇幅和时间，许多书籍一般很少专门讲解这一类应用问题．本书寓语法和应用于一体，在讲解语法的同时，介绍了程序的开发、结构化程序设计以及程序的良好风格，并注意引用一些实用的库函数和实例．根据多年教学经验，本书还列出了 C 编程中的常见错误，以帮助读者解决实际问题．

本书的最后一章是高级应用，包含文件、链表、C 和 dBASE 的接口等实际应用以及图形的常识和应用，并给出了一个完整的示范性的图形程序．本书最后给出了 4 个附录（包括函数库）和 1 个索引，这些内容增加了本书的实用性．

本书第一、四、五、六章由陈礼民编写；第二章由陈礼民、张骥华编写；第三、七章由李淑琴编写；第八章由江礼璋、陈礼民编写；第九章由江礼璋、李淑琴、陈礼民编写；附录由陈礼民、张骥华、李淑琴编写．

本书由复旦大学于玉副教授主审，刘开瑛教授在本书的撰写中给予了热情指导，特在此表示我们的深切谢意．对参与编写、录入、调试程序的山西大学计算机系的部分学生也在此表示感谢．最后，在本书出版之际，我们特别表示我们对药英副教授的感激之情，她生前对本书出版给予了大力支持．

希望本书能成为读者喜欢的教材和参考书，但限于作者水平，不当之处在所难免，还望读者和专家们不吝赐教．

编者

# 目 录

# 第一章　C语言简介

## 1.1　概　述

### 1.1.1　C语言的由来和发展

C语言起源于BCPL语言．1970年，K.L.Thompson和D.M.Ritchie为了改善他们的软件开发环境，完成了第一个UNIX系统，并且接着用汇编改写了UNIX系统．但是，由于汇编语言的移植性，程序开发效率和易读性都比不上高级语言．为了摆脱汇编语言的困扰，K.L.Thompson打算用一种高级语言来描述系统，结果产生了B语言．由于种种原因，B语言未能流行起来．于是，D.M.Ritchie在B语言中加入了多种数据类型和控制结构，从而形成了一种新的语言．为了表示它来源于B而又区别于B，便从BCPL中取了第二个字母，取名为C．1972年，C语言正式在PDP-11的UNIX系统上运行．1973年，人们用C语言重写了UNIX操作系统的内核，开创了成功地把高级语言用于系统软件开发的先例．

为了改变用C语言写出的软件的可移植性，Steve Johnson在1975年实现了第一个“可移植的C编译”．UNIX系统是建立在C基础上的，因此，C的可移植性也为UNIX的可移植性奠定了基础，从而使UNIX操作系统开始为人们所接受，并广泛应用到各种机器上；反过来UNIX的发展又极大地促进了C语言的发展和应用．

70年代是微机发展的时代．鉴于C语言小巧精悍、功能强，人们开始把它移植到非UNIX系统的微机上，并在微机CP／M-80上实现了C编译．这一开拓性的工作表明了C语言已发展成为一个独立的通用程序设计语言．1972年C语言的诞生和1978年Ritchie等人的《C程序设计语言》一书的出版，标志着C语言从一个系统程序设计语言发展成一个通用程序设计语言的阶段已经完成．

80年代，微机由于其性能价格比的不断提高而得到了广泛应用．微机对软件的需要促进了C的蓬勃发展，并出现多种C编译系统．为了提高C语言的适应性，以利于统一和发展，美国国家标准协会ANSI成立了一个X3J11委员会，并于1988年5月公布了ANSI C语言标准X3J11／88-002．该标准于1989年获得通过，使C语言逐步规范化、统一化，从而提高了C语言的品质．

C语言的标准化是它成熟的标志，也是C语言发展的第二个阶段．它表明C已发展成一个面向多种应用、脱离任何一种机器和操作系统的通用程序设计语言．通常，人们把1988年以后出现的符合ANSI标准的C语言称做ANSI C或标准C；而把1988年以前出现的C语言称做传统C．标准C的出现促进了UNIX操作系统的推广应用，并进一步推动了C的发展．目前，广泛流行的Turbo C，Microsoft C和Lattice C都是以ANSI C为标准扩展而成的．

### 1.1.2 C 语言的主要特点

60 年代，先后出现了许多高级语言，使程序员不必了解系统的内部结构，只要按语言所提供的语句格式、语法和语义说明，即可编写出所需程序．因此，高级语言很受用户欢迎．

C 语言是 70 年代初期设计的，它在很多方面继承和发扬了高级语言的成功经验与特色，如数据类型、多种存储类别、自由格式、模块化结构、参数的传值方式、结构化的控制以及分别编制等等．但是，它是作为一种系统程序设计的工具而设计的，因此决定了它具有许多特殊性：

(1) 由于起初的 C 是面向系统程序设计的，它十分注意效率．C 的语言特征和机器提供的功能相匹配，可读性远胜于汇编，这样可以大幅度提高软件开发效率；为了追求高质的目标码，要求简捷明快，所以 C 语言的表达非常简练；为了摆脱数据表示的严格区分的约束，C 基本上可以自由翻译，允许"隐式转换"，给编程带来很大的方便．总的来讲，C 语言目标码的执行效率仅比汇编低 10–20%．另外，由于 C 语言的语句表达简练，编译时又不做太多的安全检查，所以编译效率是其它语言所不能比拟的．

(2) 由于灵活性高就不能规模大，所以 C 语言只有一个很小的内核，而其强大的功能是通过函数库来实现的．

(3) C 语言是一种中级语言．它是介于汇编和高级语言之间的一种语言，这是因为系统程序设计要求程序能触及到系统内部，能反映计算机硬件结构，如一块内存的申请、寄存器的位操作及逻辑运算等等．因此，C 除了具有高级语言的能力外，还具有汇编语言的能力．这些能力，在目前高级语言是很少有的．

C 语言主要特性如下：

(1) 程序设计模块化．C 语言是一种模块化的程序设计语言，而函数是 C 语言程序的基本单位．从结构上看，C 语言程序是由一系列函数和外部变量说明构成的．一个源程序由多个代表程序段的文件组成，这种程序的构造方法能适应程序模块化的要求．

(2) 函数方式操作．C 语言虽然只有 10 个基本语句，但可以通过函数方式，构造许多使用者所需要的功能模块，例如字符串运算、基本 I／O 操作等．使用这些函数，如同使用基本语句那样方便．

(3) 数据类型多样化．C 语言具有字符串、不同长度的整数和浮点数的基本数据类型以及由此导出的指针、数组、结构和联合等导出数据类型．这种多样化的数据类型，增强了程序的处理能力．

(4) 运算功能齐备．在 C 语言中除了一般的高级语言使用的算术运算和逻辑运算外，还有位操作功能（按位与，按位或，按位非）、半加运算以及移位运算等．此外，C 还有自加、自减运算（如 x++，y—，++z），组合运算（如+=，−=，＊=）．

(5) 存储类型丰富．C 语言对已定义的变量，可追加存储类型说明和定义．存储类型有局部于函数的和外部于函数的、静态的和非静态的、寄存器变量和非寄存器变量等．存储类型说明还可以是复合说明，例如静态外部变量、静态局部变量等．存储类型的说明，表示不同变量作用域和存在时间的长短，即变量的可见性和寿命．值得指出的是，变量的构造类型和存储类型也可以用于函数定义，指明函数返回值的构造类型和存

储类型.

(6) 具有预处理功能. C语言有三种预处理: #include用于在程序某处插入事先设计的源程序段文件, 以达到共享某些具有通用性的源文件; #define用比较简短的字符串替代较长的字符串, 宏替换还可以有函数宏, 如C语言中的getchar和putchar; 另外还有条件编译#if, #ifdef等. 这些功能给程序的组织和编译提供了便利.

(7) 程序简洁. 丰富的数据类型使数据的表示简练、紧凑; 34种运算符能使表达式简短, 如用i++表示i=i+1; 用&x表示取x变量的地址; 在程序书写上, 用"{"和"}"代替begin, end, 用int, char代替integer, character; 用宏定义可以缩写常数、变量、表达式、函数等等.

## 1.2 C语言程序的基本结构

### 1.2.1 程序及其格式

C语言是由函数组成的, 如例1.1.

**例1.1**

```
/ * file 1 * /
#include <stdio.h>
main( ) {
    int i, j, sum;
    i=1;
    j=2;
    sum=i+j;
    printf("%d", sum);    }
```

这是一个函数, 其中, main是函数名, 叫主函数, 其后必须跟着一对圆括号. 一对花括号包含的是函数要完成的工作部分, 即函数的内容, 也叫函数体. 它的第一行是说明语句, 说明变量的类型; 第二、三、四句是赋值语句. #include是预处理, printf是一个函数的调用.

C语言程序的书写要注意如下几点:

(1) 一个程序可以有注解行和预处理行. 注解用左右"/"和"*"括起来. 如/*…*/; 而预处理行则由"#"引导.

(2) 程序语句的书写格式一般为每一行书写一句, 但也可以书写多句. 分号";"是语句的结束标志. C程序的书写格式非常自由, 即使把例1.1写成

```
main( ) {int i, j, sum;  i=1;  j=2;  sum=i+j; …}
```

的形式也是正确的.

(3) C程序一般采用小写字母, 只有定义某些常数时, 才使用大写字母.

(4) 空格或空行能增加程序的可读性. 在名字和操作符前后可以任意设置空格和空行, C编译能跳过它们, 正确地读取有用信息.

(5) 编写程序须先从编写函数开始, 函数由函数名和函数体组成. 函数名必须跟一对圆括号. 函数体由花括号作定界符, 标明一个程序块的起迄.

### 1.2.2 程序的结构特点

C是一种模块化的语言, 用它编写的程序具有如下特点:

1. 模块结构性

C 语言程序由若干个函数组成，函数是程序的构件.这充分体现了程序的模块结构性，如例 1.2 .

**例 1.2**

```
fun( ) {
    int i, j, sum;
    i=1; j=2; sum=i+j;
    return(sum);    }
```

```
main( ) {
    int k;
    k=fun(1);
    printf("k=%d\n", k);    }
```

上述程序由三个函数组成：一个是主函数 main，一个是函数 fun，另一个是库函数 printf . 每一个函数都是具有不同功能的程序模块 . 这里，函数 fun( ) 为求和，函数 printf 为打印.

2. 模块调用

在 C 程序中，主函数是必须出现的，有且仅有一个，而且程序总是从主函数开始执行 . 如例 1.2 中，函数 fun 虽然排在主函数的前面，但是执行时还是从主函数开始的. 在一个大程序中，主函数往往只起调用函数的作用 . 例 1.2 中，main 主函数调用函数 fun( )，由 fun( ) 把计算的结果通过 return 语句送回给主函数，并赋值给变量 k，最后调用函数 printf 把结果打印出来 .

函数调用也即模块调用 . 一个程序是通过主函数调用各个模块来完成程序的总体功能的，所以在 C 程序中的模块结构性非常突出 .

### 1.2.3 函数

除了主函数和库函数外，C 语言允许用户定义自己的函数 . 函数由函数名和函数体构成，参数及其类型说明写在函数名后的圆括号内，如例 1.3 .

**例 1.3**

```
abs(int i) {
    if (i<0)
    i=-i;
    return(i);    }
```

上述函数中，i 叫形式参数，简称形参 . int 说明 i 是整型的形参 . 一个函数相当于一个程序模块，为完成一定功能，它需要和外界通讯，输入需要加工的数据，输出处理完的数据 . 在函数中的形参将数据传入函数，而 return 语句把数据传送回给调用者 .

多个参数构成一个参数表，如：

```
fun(int i, float f)
{...}
```

参数间用逗号隔开. 如果没有数据需要传送，参数表可以是空的(或用 viod 表示)，如：

```
fun(viod)
{...}
```

或

```
fun( )
{...}
```

函数的返回值，代表函数对外的性质，所以函数返回值的数据类型也就是函数的类型 . 例 1.2 中，如果返回值是 float 型的，则应改为

```
float fun(float f)
{if (f<0);
f=-f;  return(f);  }
```

在函数名 fun 前面的 float，说明函数的返回值是浮点型的数据．当函数没有返回值时，可用 void 表示，以免产生错误．

在 C 语言中，函数都是外部的，即函数只能在函数之外定义；函数的定义是不能嵌套的，所以在函数中只能调用函数，而不能定义函数．

### 1.2.4 函数 printf

在一般情况下，主函数中最后一行是由函数 printf 组成的打印语句．在函数 printf 中，双引号包含部分是它的第一个参数，其中的字符串可以按原样印出（除了控制部分以外）．符号"%"及其跟随着的字符控制第二个、第三个、…参数的打印格式．如例 1.2 中的字符"%d"控制参数 k 以十进制整数形式打印．

下面列出常用的一些打印格式，关于函数 printf 的详细使用方法，请看以后章节．

| 格式串 | 相应的参数输出形式 |
|---|---|
| %d | 十进制整数 |
| %c | 单个字符 |
| %s | 字符串 |
| %f | 十进制浮点数 |

## 1.3 程序的开发过程

程序的开发包括源文件的建立、编译和执行三步，依赖于操作系统的文本编辑、编译系统．但是，它们的开发过程相同．

### 1.3.1 编辑源文件

C 语言程序是以文件方式储存的，用户写一个程序，也就是要建立一个用 C 语言书写的文件．这个文件的名字必须有扩展名 .c，用以指明这是 C 语言编写的程序，否则，编辑将不予理睬．

编辑程序种类很多，在 UNIX 操作系统中，可使用 vi，ed；在 MS-DOS(或 PC-DOS) 操作系统中，可使用 edlin，也可以使用一些功能更强的编辑软件，如 wordstar 等．录入文件可使用下面的命令：

```
ed example.c          (UNIX)
edlin example.c       (MS-DOS)
```

在需要修改编辑过的文件时，也可以使用上述编辑命令把文件调出来，再进行修改．

### 1.3.2 编译——生成目标程序

经过编辑的文件是字符型的源文件，需要经过编译才能成为二进制的目标代码文件．编译过程同时也是语法检查过程．C 的编译一般使用 cc，如：

```
cc example                    (这时，一般文件不带扩展名.c)
```

cc 命令将调用编译程序对程序 example.c 进行编译．当然，不同的编译可能用不同的标识符来代表编译命令，如在 Macrosoft C 中，编译命令为 Qcl，Cl；在 Turbo C 中，编

译命令为 Tcc .

### 1.3.3 链接装入——生成可执行文件

编译后产生的目标文件还不能直接执行 . 这是因为: 一，编译后的目标程序并没有真正装入内存，它的地址是相对的，即浮动的；二，程序运行所需的库函数以及目标程序模块，尚未链接在一起，不能生成真正的可执行文件，所以程序编译过后，还要使用 link 命令，如:

link　文件名

进行链接装入 . 在不少系统中， cc 命令并不只是编译命令，而是包含了编译命令和链接命令两种功能, 如 Turbo C 的 Tcc, MS–C 的 Qcl(和 UNIX 系统的 cc 命令很相似).因此，在这样的系统中就不必再使用 link 命令了. 总之, 经过链接即生成可执行文件.

——UNIX 系统下产生 a.out 可执行文件

——MS–DOS 下产生.exe 型可执行文件

链接过后，文件就可执行，执行命令的形式为

a.out　　回车　　(UNIX)

文件名　回车　　(MS–DOS)

以上提供的只是 C 语言编程的一般开发过程，在具体使用时，请参考各系统提供的关于编辑、编译程序和链接程序的有关手册，按照规定格式使用命令 .