

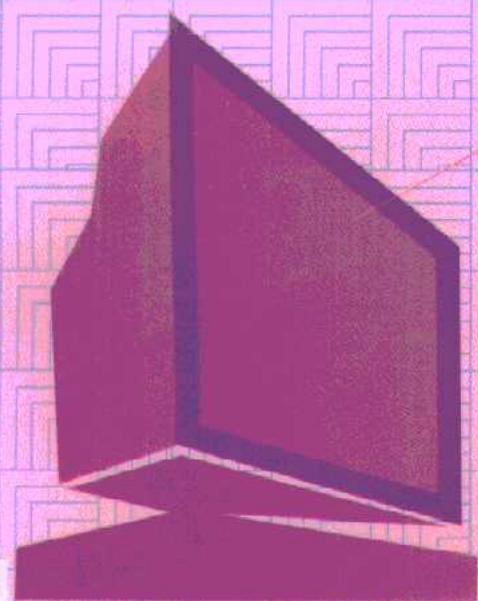
全国计算机等级考试

二级教程

C 程序设计

刘瑞挺 主编

高福成 李宗耀 李兰友 编著



C 程序设计

南开大学出版社

TP3
L7364

全国计算机等级考试

二 级 教 程

C 程序设计

刘瑞挺 主编
高福成 李宗耀 李兰友 编著



A0910031

南 开 大 学 出 版 社

• 天 津 •

内 容 提 要

本书是根据教育部考试中心制定的《全国计算机等级考试二级考试大纲》中有关C语言程序设计的要求编写的。本书以应用为目的,以程序设计为主线,系统介绍了C语言及其程序设计技术。全书共十一章,包括C语言的基本数据类型、数据运算、程序控制结构、数组、函数、指针、复合数据类型、文件和编译预处理等。书中突出了重点,分解了难点,配以大量的应用实例和多种类型的习题,方便自学,是应试人员考前必读教材,也可供软件开发人员参考。

图书在版编目(CIP)数据

C 程序设计/刘瑞挺主编. —天津:南开大学出版社,2000.1 重印
(全国计算机等级考试系列丛书)
全国计算机等级考试二级教程
ISBN 7-310-01306-9

I. C… II. 刘… III. C语言-程序设计-水平考试-教材
IV. TP312

中国版本图书馆 CIP 数据核字(1999)第 35233 号

出版发行 南开大学出版社
地址:天津市南开区卫津路 94 号
邮编:300071 电话:(022)23508542
出版人 张世甲
承 印 天津市宝坻县第二印刷厂印刷
经 销 全国各地新华书店
版 次 1999 年 10 月第 1 版
印 次 2000 年 1 月第 2 次印刷
开 本 787mm×1092mm 1/16
印 张 17.5
字 数 440 千字
印 数 5001—10000
定 价 25.00 元

前 言

C 语言是计算机程序设计语言中的小字辈,但由于其卓越的性能,使它风靡全世界,成为最受欢迎的语言之一。当今流行的面向对象语言 C++ 及 Internet 网上的 JAVA 语言就来源于 C,参加全国计算机等级考试三级(偏软)的应试者也必须掌握 C 语言。

目前参加计算机等级考试的应试者中,选择 C 语言的人数不断增加。学习计算机语言的唯一目的是应用,而应用要通过程序设计来体现。进行程序设计,需要很强的逻辑思维能力,是一种极富创造性的智力劳动。可以这样认为,语言是一种技能,程序设计是一门科学。因此,任何计算机语言及其程序设计的基本特点就是理论性和实践性并重,教学上应强调科学训练与技能培养并存。基于这一认识,本书的宗旨是以应用为目的、以程序设计为主线,系统介绍 C 语言及其程序设计技术,让读者在大量的程序设计实践中自然而然地熟悉和掌握语言。

本书是根据教育部考试中心制定的《全国计算机等级考试二级考试大纲》中有关 C 语言程序设计的要求编写的,从选材上注意了以下几点:

1. 立足基点,掌握制高点。基点是会用语言来进行程序设计,学会一大批基本的程序设计单元;制高点是掌握程序设计的基本思维方法和一定的技巧。因此,本书根据实用和适用的原则,精选大量的例题对 C 语言的各种语法现象加以阐述。这些例题不是零星的程序段,而是完整的实际题目,以便于读者建立整体的观点。

2. 突出重点,分解难点。本书的重点章节是第 4、6、7、10 章,书中对此进行了较大篇幅的阐述。对那些难点问题,如指针、数据在函数间的传递、存储类型等,则分散在多个章节内讨论。

3. 为了使读者能及时考查自己的学习效果,每一章都安排了大量的习题,包括选择题、填空题、编程题和程序调试题共四种题型,可满足读者参加等级考试、发挥程序设计的创造性以及上机实践的需要。

根据 C 语言自身的特点,本书从内容上作了如下安排:

第 1 章介绍简单的 C 程序设计;第 2、5、9 章从易到难介绍 C 语言的各种数据结构;第 3 章专门介绍 C 语言的运算功能;第 4 章集中介绍 C 语言的控制结构,体现结构化程序设计的特点;第 7、10 章介绍函数和文件,这是模块化

程序设计的需要;第6章介绍指针,这是C语言的精华和特色,也是学习的难中之难;第8章介绍了C语言环境中的特殊问题,即编译预处理和分割编译。附录部分给出了C语言的运算符、ASCII代码表、Turbo C常用库函数及其标题文件、Turbo C编译错误信息,便于读者查阅。最后给出了本书中的习题(选择题和填空题)参考答案。

参加本书编写的有高福成(第1、2、3、4、5、6章,7、8、9章部分)、李宗耀(7、8章部分)、李兰友(第10章)和高茵(第9章部分及全部程序调试、绘制流程图和习题解答),最后由高福成统编和定稿。南开大学出版社的领导和编辑为本书的出版付出了大量的劳动,在此表示由衷的感谢。

编著者 谨识

1999.5

第 1 章 C 程序设计初步知识

C 语言是一种通用的程序设计语言,它以表达简明、使用灵活、结构化的流程控制、丰富的数据结构和操作符集合、良好的程序可移植性和高效率的目标代码为特征。C 语言不仅具有高级语言的特点,还具有低级语言的功能,因此它既可用于编写系统程序,也可用于编写不同领域的应用程序。因此,C 语言成为国际上流行的一种高级语言。

1.1 一个简单的 C 程序

我们先通过一个简单的 C 语言程序,了解一个 C 程序是什么样子。

例 1.1 输入圆的半径,计算圆的周长和面积。

```
#include <stdio.h>
#define PI 3.14159
main()                /* 函数名 */
{                    /* 函数体开始 */
    float r,a,c        /* r:半径;a:面积;c:周长 */
    r=2.5;             /* 给定 r 的值 */
    a=PI*r*r;          /* 计算 a */
    c=2*PI*r           /* 计算 c */
    printf("r=%f,a=%f,c=%f\n",r,a,c); /* 输出 r,a,c */
}                    /* 函数体结束 */
```

以上程序运行结果如下:

r=2.500000, a=19.634956, c=15.707965

上面的程序中,main 是主函数名,C 语言规定必须用 main 作主函数名,其后的一对圆括号不能省略;函数体是程序的定义和执行部分,左花括号表示函数体的开始,右花括号表示函数体的结束;函数体中有五条 C 语句,第一条语句定义了三个浮点型变量(分别表示半径、面积和周长),其后三条语句分别对这三个变量进行赋值,最后一条语句在屏幕上显示计算结果。

在程序的开始,还有两个用“#”开头的命令行,它们的含义将在下面介绍。

1.2 C 程序的基本结构

我们再通过下一个例子,了解 C 程序的基本结构。

例 1.2 计算二整数相乘的程序

```
#include <stdio.h>                /* 文件蕴含 */
main()                            /* 主函数名 */
{                                /* 函数体开始 */
    int x,y,z;                    /* 局部变量类型定义 */
```

```

printf("enter the value of x,y:");          /* 屏幕提示信息 */
scanf("%d%d",&x,&y);                        /* 输入变量值 */
z=product(x,y);                            /* 调用计算乘积的函数 */
printf("x=%d,y=%d\n",x,y);                /* 打印 x 和 y 的值 */
printf("x*y=%d\n",z);                      /* 打印乘积值 */
}                                           /* 函数体结束 */

product(int a,int b)                       /* 子函数及其形式参数 */
{ int c;                                   /* 定义局部变量 */
  c=a*b;                                   /* 计算乘积 */
  return(c);                              /* 返回值 */
}

```

这是一个简单的含多个函数的 C 程序,它含有一个主函数和一个子函数。主函数负责输入两个变量的值,然后调用子函数,最后打印结果。子函数负责接收从主函数传送过来的两个数,计算其乘积,并将该乘积返回主函数。

程序运行时,屏幕首先提示信息:

enter the value of x,y:_

提示信息末尾的“_”是闪烁的光标,用户若从键盘输入5和8(5和8用空格隔开)并按回车键后,则程序输出为

x=5,y=8

x*y=40

C 程序的基本结构是:

1. 一个 C 程序由一个或多个 C 函数组成,其中必须有一个用“main”命名的主函数。函数是完成某个整体功能或某个模块的最小单位。C 函数相当于 FORTRAN 语言中的“程序段”和 Pascal 语言中的“过程”。

2. 一个 C 函数由若干条 C 语句组成,C 语句是完成某种程序功能(如赋值、输入、输出等)的最小单位。

3. 一条 C 语句由若干个基本单词组成,基本单词是构成语句的最小单位。C 程序的一般格式如图 1.1 所示,其中 f1() 到 fn() 表示用户定义的函数。

```

全局变量说明
main()
{ 局部变量说明
  语句序列
}
f1(形式参数表)
{ 局部变量说明
  语句序列
}
f2(形式参数表)
{ 局部变量说明
  语句序列
}
:
fn(形式参数表)
{ 局部变量说明
  语句序列
}

```

图 1.1 C 程序的一般格式

1.2.1 基本单词

C 语言共有五种基本单词,即关键字(亦称保留字)、标识符、常数、操作符和分隔符。例如语句

float r,a,c;

中,float 是关键字,代表数据类型是浮点型;r,a 和 c 是标识符(这里表示变量名);又如语句

a=PI*r*r, c=2*PI*r;

中,“=”、“*”和“,”是操作符,其中前两个是运算符,第三个是有特定用途的标点符号;“2”是常数,PI 是符号常数;在“a=PI*r*r,”和“c=2*PI*r”之间可以用一个或多个空格分开,这些空格

是分隔符。

1. 关键字

关键字是 C 语言中有特定意义和用途、不得作为它用的字符序列,其中 ANSI C 标准规定的关键字有 32 个,Turbo C 扩充的关键字有 11 个,见表 1.1。

表 1.1 Turbo C 关键字一览表

ANSI C 标准关键字					
auto	default	float	register	signed	volatile
break	do else	for	return	switch	while
case	double enum	goto	short	typedef	
char	else extern	if	sizeof	union	
const	enum float	int	static	unsigned	
continue	extern for	long	struct	void	
Turbo C 扩充关键字					
asm	interrupt	near	far	_ds	pascal
_ss	cdecl	_cs	huge	_es	

所有 C 关键字都是小写字母。在 C 语言中,大写字母和小写字母是有区别的,如 else 是关键字,而 ELSE 或 Else 就不是关键字。

2. 标识符

标识符是 C 语言中用来表示变量名、数组名、函数名、指针名、结构名、联合名、枚举常数名、用户定义的数据类型名及语句标号等用途的字符序列,可由 1~32 个字符组成,第一个字符必须是字母或下划线,后面的字符可以是字母、数字或下划线。例如

AB, Ab, aB, ab, A_b, _ab, ab_s2d, W_length 等都是正确的标识符,而 A+B,

A'B, A.B, 2abc, α , β , d% 等是错误的标识符。

标识符除了大小写不等价外,也不能与 C 关键字相同。例如 AB 和 Ab 是两个不同的标识符,ELSE 可以作为标识符,它不会与 C 关键字 else 混淆。

顺便指出,在 C 编译系统的库函数中,经常使用以下划线“_”打头的函数名或变量名,所以在程序中也应尽量避免使用以“_”打头的标识符,免得与库函数冲突。

3. 常数

常数包括数值常数(如 123、-23.5、1.2E4 等)、字符常数(如 'a'、'B'、'c' 等)、字符串常数(如 "xyz"、"good morning" 等)、符号常数以及枚举常数。

4. 操作符

操作符包括各种运算符(如 +、-、*、/),有特定意义的标点符号(如花括号、方括号、圆括号、逗号)等。

5. 分隔符

分隔符用来分隔相邻的标识符、关键字和常数,最常用的分隔符是空格,此外还有制表符、换行符、换页符等。

1.2.2 C 语句

C 语句是组成 C 程序的基本单位,具有独立的程序功能。所有的 C 语句都以分号结尾。

1. 简单语句

任何 C 表达式加上分号后,就构成一条 C 语句。如

```
i=0;
x=x+1;
```

2. 复合语句

一组 C 语句用花括号括住,就构成复合语句。如

```
while(i<10)
{   sum=sum+i;
    i++;
}
```

复合语句被视为一个整体,通常用在条件分支或循环语句中。有时为了数据隐藏的目的,用复合语句形成一个代码块,块中定义的局部变量不会对程序的其他部分发生副作用。

3. 空语句

只有一个分号的语句称为空语句。例如

```
for(i=0; i<1000; i++)
;
```

由一条 for 语句(循环语句)和一条空语句组成。空语句用作循环语句的循环体,表示什么也不做。事实上,这个循环的功能是延迟一小段时间。有时,空语句被用作转向点。

4. 注释

严格地讲,注释不是 C 语句。一个字符串(中文或英文)以“/*”开始,而以“*/”结束称为注释,如

```
/* This is a main function */
```

注释既不被编译也不被执行,使用注释主要是为了增加程序的可读性。

1.2.3 函数

函数(有关函数的详细介绍见第7章)是一个相对独立的程序段或模块。一个 C 程序由一个主函数和任意多个其他函数组成,所有函数都具有相同的结构。

1. 函数名

主函数有固定的名称 main,其他函数则可以根据标识符的命名方法任意取名。主函数通常包括了整个程序的轮廓,由它再调用其他函数。

2. 形式参数

在函数名的后面有一对圆括号,其中放置一个或多个形式参数,简称形参、虚参或哑元。一个函数也可以没有形式参数,但圆括号不能省略。

3. 函数体

用花括号包围起来的部分是函数体,即函数的主体。它主要有两大部分,第一部分是本函数内部用到的局部变量类型定义(C 语言中,所有的变量都要先定义后使用),第二部分是语句序列,完成本函数的功能。

1.2.4 C 程序的书写风格

C 程序的书写格式完全自由。

1. 每个函数在整个程序文件中的位置任意。主函数不一定出现在程序的开始处,但不管主函数位于程序中的何处,程序运行时总是从主函数开始。

2. 每个程序行中的语句数量任意。既允许一行内写几条语句,也允许一条语句分几行书写,但每条语句都必须以分号(;)结束。有时还可以在程序的适当地方(如两个函数之间)加进一个或多个空行,使程序结构更加清晰。

3. 注释的位置任意。注释可以出现在程序的任何地方,既可以独占一行或几行,也可以出现在某语句的开头或结尾处。如果注释占有几行,则每一行都要以“/*”开始,以“*/”结束,“*”和“/”之间不能有空格。注释对程序的编译和运行没有影响,但使用注释可以明显增加程序的可读性。

4. 尽管C程序的书写几乎没有限制,但为使程序清晰易读,通常每行写一条语句;不同结构层次的语句从不同的位置开始,即按缩进格式书写成阶梯形状,可以用Tab键或空格键调整各行的起始位置。

1.2.5 库函数和标题文件

例1.1和例1.2中的第一行都是“#include <stdio.h>”,这是调用C编译程序提供的标题文件,因为程序中用到的库函数scanf()和printf()都是在标题文件stdio.h中定义的。库函数不是C语言本身的组成部分,而是由C编译系统提供的一些非常有用的功能函数。例如,C语言没有输入输出语句,也没有直接处理字符串的语句,但是C编译系统以库函数的方式提供了这些功能。另外,还有大量的数学函数及其他函数可供用户直接调用。这些库函数的说明、类型和宏定义都保存在相应的标题文件(也称头文件)中,而对应的子程序则存放在运行库(.lib)中,用户只要在程序的函数外部用

```
#include <标题文件>
```

或

```
#include "标题文件"
```

包含指定的标题文件,就可以调用其中定义的库函数。例1.1和例1.2就是通过标题文件stdio.h调用了库函数scanf()和printf()来进行变量的输入和输出的。

Turbo C 提供的标题文件如下(标题文件名不区分大小写):

ALLOC.H	动态分配函数(ANSI C)
ASSERT.H	诊断函数(ANSI C)
BIOS.H	ROM-BIOS 函数
CONIO.H	屏幕处理函数
CTYPE.H	字符处理函数(ANSI C)
DIR.H	目录处理函数
DOS.H	DOS 接口函数
ERRNO.H	错误码定义函数(ANSI C)
FCNTL.H	OPEN 函数使用的常数定义
FLOAT.H	与机器有关的浮点数定义及限制(ANSI C)
GRAPHICS.H	图形函数
IO.H	UNIX 用 I/O 子程序

LIMITS. H	与机器有关的各种限制(ANSI C)
LOCATE. H	国家和语言标识函数(ANSI C)
MATH. H	数学函数(ANSI C)
MEM. H	内存操作函数
PROCESS. H	spawn()和 exec()函数
SETJMP. H	非局部转移
SHARE. H	文件共享
SIGNAL. H	中断信号处理函数(ANSI C)
STDARG. H	可变长度参数表定义(ANSI C)
STDDEF. H	公用常数定义(ANSI C)
STDIO. H	标准输入输出函数(ANSI C)
STDLIB. H	标准实用函数(ANSI C)
STRING. H	字符串处理函数(ANSI C)
TIME. H	系统时间函数(ANSI C)
VALUES. H	与机器有关的常数
SYS. H	为打开文件定义的常数
SYS. H	为 ftime()函数所作的说明
SYS. H	与时间函数联用的类型定义

其中,未注明(ANSI C)的都是由 Turbo C 定义的标题文件。

从技术角度讲,任何函数都可以自行编制,完全可以不使用 C 编译系统提供的函数库。然而很少这样做,因为 C 并未提供任何执行 I/O 操作的方法。绝大多数 C 程序都包含对标准函数库的调用。函数库是一种重要的软件资源,读者应在学习 C 语言本身的同时,逐步了解和掌握各种库函数的功能和用法,充分利用它们,避免重复自行编制,以免事倍功半。

需要说明的是,不同的 C 编译系统提供的库函数在数量、种类、名称及使用上都有一些差异,甚至连标题文件的名称也不一定相同。正如我们上面所看到的,Turbo C 就在 ANSI C 标准建议的100多个库函数的基础上扩充到374个。但就 ANSI C 建议的15类标准函数而言,各类 C 编译系统大致都是相同的。

1.3 简单的 C 程序设计

我们知道,一个程序一般需要具备三个功能,即输入数据、数据处理和输出结果。C 语言中的数据处理功能主要是由赋值语句完成的,而数据的输入、输出则要调用系统提供的输入/输出函数。本节先介绍最常用的输入、输出函数及赋值语句,以能进行最简单的程序设计。

1.3.1 格式输出函数 printf()简介

格式输出函数 printf()的一般形式为

printf("格式控制字符串",输出项目清单)

其中,printf 是函数名,其后的括号中是该函数的参数;格式控制字符串用双引号括住,用来规定输出格式;输出项目清单中包含零个或多个输出项,它们可以是常数、变量或表达式,相互之

间用逗号隔开。例如

```
printf("%d",125);
printf("x=%f,y=%f",x,y)
printf("Hi,good morning!");
```

1. 输出格式

输出格式由格式控制字符串加以规定,它关系到数据输出格局的安排。格式控制字符串可包括以下三种符号:

(1) 格式转换说明符

最简单的格式转换说明符是在“%”后跟一个特定的字母,用来与输出项的数据类型相匹配:

%d	输出十进制整数(decimal)
%x	输出十六进制整数(hexadecimal)
%o	输出八进制整数(octal)
%f	输出浮点数,保留6位小数(float)
%c	输出单个字符(character)
%s	输出一个字符串(string)
%%	输出一个%

(2) 转义序列

转义序列由“\”后跟一个特定的字母组成,用以输出某些特殊字符和不可见字符。例如:

\n	回车换行(Enter)
\t	制表符(Tab,光标右移8位)
\a	报警铃响(alert)
\\	反斜杠(\)
\"	双引号(")

(3) 其他字符

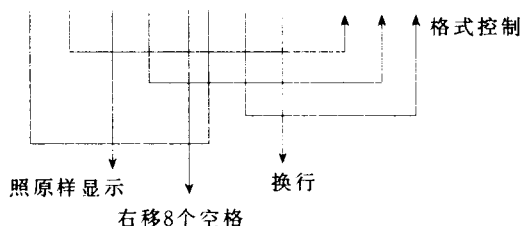
除以上两项字符以外的其他字符都视为普通字符,输出时照原样显示。

2. 函数功能

printf()函数的功能是将输出项目的值按对应的格式转换说明符的规定,在标准输出设备——显示器屏幕上显示出来,格式转换说明符中的转义序列起换行、换页等控制功能,格式转换说明符中出现的普通字符则照原样显示。

假设 a 和 b 是整型变量(a=123,b=1455),c 是浮点型变量(c=3.14159265),则 printf("a=%d,b=%x\tc=%f",a,b,c)的控制关系如下所示:

```
printf("a=%d,b=%x\tc=%f\n",a,b,c);
```



该语句的输出结果为

```
a=123,b=5af      c=3.141593
```

又如, `printf("Ten years is\na long time\nfor human's life. \n")` 中不含输出项, 格式控制字符串中除“\n”外都是普通字符, 照原样显示, 由于“\n”的作用, 结果输出三行:

```
Ten years is
a long time
for human's life.
```

3. 说明

(1) 整个格式控制字符串必须用双引号括住。如果有输出项目, 则格式控制字符串与第一个输出项目之间用逗号隔开。

(2) 如果有多个输出项目, 各输出项目之间用逗号隔开。

(3) 格式转换说明符的个数应与输出项的个数相等, 且顺序和类型应一一对应, 例如

```
printf("%d %d %d %d", x, y, z)
```

中, 四个格式转换说明符对应三个输出项, 这是错误的; 如果 `x`、`y`、`z` 不是整型, 则使用“%d”格式也是错误的;

(4) 利用输出格式控制中的三项内容恰当组合, 可形成不同的显示效果, 如表格、图案等。

1.3.2 格式输入函数 scanf() 简介

格式输入函数 `scanf()` 的一般形式为

```
scanf("格式控制字符串", 输入项目清单)
```

例如, `scanf("%d", &x);`

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

其中, 格式控制字符串通常只包含格式转换说明符, 其含义与 `printf()` 类似, 同样用“%”后跟一个字母表示, 输入项目清单要求至少有一个输入项, 且必须是变量的地址(`&x`、`&a`、`&b` 分别表示变量 `x`、`a`、`b` 的地址)。

`scanf` 函数的功能是从标准输入设备—键盘接收数据, 按格式转换说明符将数据转换成相应的格式存放到对应的变量中。

下面就 `scanf` 函数的使用说明几点:

1. 当程序执行到 `scanf()` 时, 将停止继续执行, 等待用户从键盘输入数据。如果只有一个输入项, 则键入数据后再键回车键; 如果有多个输入项, 则键入的数据之间应用一个或多个空格、Tab(键盘上的 Tab 键)或回车键隔开。例如

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

```
scanf("%f%f", &x, &y);
```

你可以这样输入(↓表示回车键):

```
5 ↓
```

```
8 ↓
```

```
1.2 ↓
```

```
3.5 ↓
```

也可以这样输入:

5 8 ↓
1.2 3.5 ↓

还可以这样输入:

5 ↓
8 1.2 ↓
3.5 ↓

2. 在格式控制字符串中通常只出现格式转换说明符。如果出现其他字符,这些字符应该照原样输入,否则 scanf 函数从输入数据中找不到这样的字符串时,就自行终止。例如

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

输入数据时,两个数据之间必须有逗号,即

5,6 ↓

如果输入

5 6 ↓

则变量 a 获得数值 5,scanf 找不到与格式控制字符串中对应的“,”,函数自行终止,变量 b 未获得数据。

因此,不能用 scanf 函数去显示一个提示信息。比如初学者希望在输入数据前屏幕上能显示:

```
enter a value;
```

字样,于是写出下列形式

```
scanf("enter a value: %d",&x);
```

以实现人机对话功能。实际上,程序运行时,屏幕上并不显示提示信息。格式控制字符串中出现的 enter a value: 不会出现在屏幕上,而是要求用户照原样从键盘输入。如果 x 的值是 10,那么应该从键盘输入

enter a value:10 ↓

显然,在格式控制字符串中加入不必要的字符是不可取的。为了实现上述形式的人机对话,可用 printf 函数显示提示信息,用 scanf 函数输入数据,即

```
printf("enter a value:");
```

```
scanf("%d",&x);
```

这样,屏幕上会先显示

```
enter a value: _
```

用户只需键入

10 ↓

就可以了。

3. 和 printf 函数一样,scanf 也要求格式转换说明符与输入项在个数、顺序、类型上一一对应。例如

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

若键盘输入

56.3145 35 ↓

则 b 的值就不能正确读入。

顺便指出,由于各种 C 编译程序都提供 printf 和 scanf 函数,在编译连接时,会自动将用户程序与这两个标准库文件相连,因此,可以不用 #include <stdio.h> 就能直接调用这两个函数。如果要调用其他库函数,则不能省略 #include。

1.3.3 简单的赋值语句

输入函数只能把用户输入的常数赋给指定的变量,赋值语句则可以对表达式进行计算,然后把计算结果赋给指定的变量,其基本格式为

`v=e;`

其中,v 是变量名,e 是表达式。赋值语句的功能是先计算表达式 e 的值,并把 e 的值转换成和 v 相同的数据类型后赋给变量 v。

1.3.4 简单程序设计举例

例1.3 为使例1.1的程序更具通用性,在每次程序运行中,可由用户从键盘输入半径 r 的值,计算并打印该半径所对应的圆的周长 c、面积 a、球的表面积 s 和球的体积 v,这样,输入不同的半径值,就会得到不同的运行结果。程序如下:

```
#include <stdio.h>
main()
{ float r,c,a,s,v;
  printf("Input value of r:");
  scanf("%f",&r);
  c=2*3.141593*r;
  a=3.141593*r*r;
  s=4*a;
  v=s*r/3;
  printf("r=%f\t",r);
  printf("c=%f\n",c);
  printf("a=%f\t",a);
  printf("s=%f\t",s);
  printf("v=%f\n",v);
}
```

某次程序运行结果如下:

```
Input value of r:2.5 ↓
r=2.500000      c=15.707965
a=19.634956     s=78.539825      v=65.449852
```

例1.4 从键盘输入一个小写字母,打印该字母及其对应的十进制 ASCII 代码值,然后打印该字母对应的大写字母及其对应的十进制 ASCII 代码值。

我们知道,同一个英文字母,其大小写字母的十进制 ASCII 代码值相差32,于是有了小写字母就容易得到大写字母;反之也一样。程序可编制如下:

```
main()
```

```
{ char ch1,ch2;  
  scanf("%c",&ch1);  
  printf("%c,%d\n",ch1,ch1);  
  ch2=ch1-32;  
  printf("%c,%d\n",ch2,ch2);  
}
```

程序某次运行结果为：

```
d ↓  
d,100  
D,68
```

1.4 C 程序的编译与运行

C 语言是一种编译型的程序设计语言，一个 C 程序要经过编辑、编译、连接和运行四个步骤，才能得到运行结果。

1. 编辑

编辑是指 C 语言源程序的输入和修改，最后以文本文件的形式存放在磁盘上，文件名由用户自己选定，扩展名一般为“.c”，例如 a.c、sample.c、gfc.c 等。编辑所使用的软件是编辑程序。它是提供给用户的书写环境，用来输入和修正源程序，如 UNIX 系统提供的行编辑程序 ed、全屏幕编辑程序 vi、DOS 系统提供的全屏幕编辑程序 edit、UCDOS 系统提供的字处理软件 WPS、Windows 系统提供的字处理程序 word 等，都可以用来编辑 C 源程序。如果使用字处理软件，则必须使用非文书文件方式或纯文本文件方式存盘。

2. 编译

编译是把 C 源程序翻译成可重定位的二进制目标程序。编译过程由编译程序完成。编译程序自动对源程序进行句法和语法检查，当发现这类错误时，就将错误的类型和在程序中的位置显示出来，以帮助用户修正源程序中的错误。如果未发现句法和语法错误，就自动形成目标代码并对目标代码进行优化后生成目标文件。目标文件的名称由编译系统自动规定，也可以由用户指定。不同的编译系统或不同版本的编译程序，它们的启动命令可能不同，生成的目标文件的扩展名也不一定相同。

3. 连接

连接也称链接或装配，是用连接程序将编译过的目标程序和程序中用到的库函数连接装配在一起，形成可执行的目标程序。可执行目标文件的扩展名由系统自动确定。

4. 运行

运行是将可执行的目标文件投入运行，以获取程序的运行结果。通常，在 DOS 操作系统提示符下直接键入可执行文件名，或在 Windows 操作系统下用鼠标双击可执行文件名即可。

上述四个步骤示于图 1.2。图中，带箭头的实线表示操作流程，带箭头的虚线表示操作所需要的条件和产生的结果。例如，在 Turbo 2.0 集成环境中的操作步骤为：在编辑菜单下进行输入和编辑，产生 C 源程序文件（扩展名为.c）；对该文件进行编译，生成可重定位的目标文件（扩展名为.obj）；再将目标文件进行连接装配，生成可执行的目标文件（扩展名为.exe），也称可执

行文件;运行可执行文件即可获得运行结果。

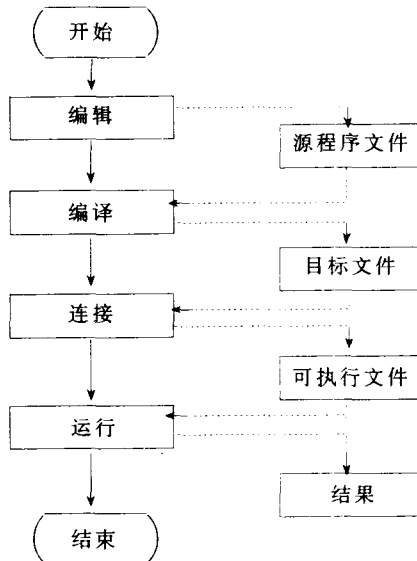


图1.2 C 程序的开发过程

1.5 Turbo C 2.0集成环境

Turbo C 2.0是美国 Borland 公司开发的一个 C 语言集成环境,它集编辑、编译、连接及运行功能于一身,使得 C 程序的编辑、调试和测试非常简捷,编译和连接速度极快,使用也很方便。

1. Turbo C 2.0的启动

Turbo C 2.0可以在 DOS 或 Windows 下运行。

(1) DOS 环境下启动

假设 Turbo C 2.0安装在 C 盘根目录下的 TC 子目录中,要启动 Turbo C,只要先启动 DOS。如果需要使用汉字,则再启动汉字 DOS(但在大多数汉字 DOS 下,Turbo C 不直接支持使用汉字,在 UC DOS 或天汇 DOS 下,Turbo C 可直接处理汉字,这给用户带来方便),然后可按下面两种方式进入 Turbo C 集成环境:

① 直接进入 TC 子目录,启动 Turbo C,即

C>CD C:\TC ↓

C:\>tc ↓

② 先进入用户子目录(假设用户子目录为 C:\USER),然后启动 Turbo C,即

C>CD C:\USER ↓

c:\USER>c:\TC\tc ↓

使用前一种方式进入 Turbo C,用户的程序文件将存放在 TC 子目录下;使用后者,用户的程序文件将存放在用户子目录下。

(2) Windows 95下启动