



C 语 言 程 序 设 计

```

-v timeout destination-list
-v timeout destination-list
-v timeout destination-list

```

```

Ping the specified host until stopped.
To see statistics and continue - type Control-Break;
To stop - type Control-C.
Resolve addresses to hostnames.
Number of echo requests to send.
Send buffer size.
Set Don't Fragment flag in packet.
Time To Live.
Type Of Service.
Record route for count hops.
Timestamp for count hops.
Loose source route along host-list.
Strict source route along host-list.
Timeout in milliseconds to wait for each reply.

```

```

Patch1      ATMREG.SIM      Bmc.La
SABE.DLL    CDLLA1g.EXE    CDLLA1
S16.EXE     CDLLA1g.DLL    CDLLA4
S64.EXE     CDLLA1g.DLL    Coffee
log         CDLLA1g.DLL    (Config
action Wizard) CDLLA1g.DLL    (Core1)
lpf.lrs     (Current)       (Debug)
tsul.exe    (Desktop)       Directx
svcr.exe    (Device Cache)    DMRTA
huc11       EPSI700.LK     explore
svcr-cf     FeatherFeature.bmp Gome71
action.hop  (Help)         hhv
Skr5.000    HomeSite.log          ICCO2
QDES.DLL    iccsig.dat             idr.Ld
sv.log      IsUn0804.exe          IsUn08
M1          MPCRS.INI      MCRP12
M2          MPCRS.INI      MCRP12
M3          MPCRS.INI      MCRP12
M4          MPCRS.INI      MCRP12
M5          MPCRS.INI      MCRP12
M6          MPCRS.INI      MCRP12
M7          MPCRS.INI      MCRP12
M8          MPCRS.INI      MCRP12
M9          MPCRS.INI      MCRP12
M10         MPCRS.INI      MCRP12
M11         MPCRS.INI      MCRP12
M12         MPCRS.INI      MCRP12
M13         MPCRS.INI      MCRP12
M14         MPCRS.INI      MCRP12
M15         MPCRS.INI      MCRP12
M16         MPCRS.INI      MCRP12
M17         MPCRS.INI      MCRP12
M18         MPCRS.INI      MCRP12
M19         MPCRS.INI      MCRP12
M20         MPCRS.INI      MCRP12
M21         MPCRS.INI      MCRP12
M22         MPCRS.INI      MCRP12
M23         MPCRS.INI      MCRP12
M24         MPCRS.INI      MCRP12
M25         MPCRS.INI      MCRP12
M26         MPCRS.INI      MCRP12
M27         MPCRS.INI      MCRP12
M28         MPCRS.INI      MCRP12
M29         MPCRS.INI      MCRP12
M30         MPCRS.INI      MCRP12
M31         MPCRS.INI      MCRP12
M32         MPCRS.INI      MCRP12
M33         MPCRS.INI      MCRP12
M34         MPCRS.INI      MCRP12
M35         MPCRS.INI      MCRP12
M36         MPCRS.INI      MCRP12
M37         MPCRS.INI      MCRP12
M38         MPCRS.INI      MCRP12
M39         MPCRS.INI      MCRP12
M40         MPCRS.INI      MCRP12
M41         MPCRS.INI      MCRP12
M42         MPCRS.INI      MCRP12
M43         MPCRS.INI      MCRP12
M44         MPCRS.INI      MCRP12
M45         MPCRS.INI      MCRP12
M46         MPCRS.INI      MCRP12
M47         MPCRS.INI      MCRP12
M48         MPCRS.INI      MCRP12
M49         MPCRS.INI      MCRP12
M50         MPCRS.INI      MCRP12
M51         MPCRS.INI      MCRP12
M52         MPCRS.INI      MCRP12
M53         MPCRS.INI      MCRP12
M54         MPCRS.INI      MCRP12
M55         MPCRS.INI      MCRP12
M56         MPCRS.INI      MCRP12
M57         MPCRS.INI      MCRP12
M58         MPCRS.INI      MCRP12
M59         MPCRS.INI      MCRP12
M60         MPCRS.INI      MCRP12
M61         MPCRS.INI      MCRP12
M62         MPCRS.INI      MCRP12
M63         MPCRS.INI      MCRP12
M64         MPCRS.INI      MCRP12
M65         MPCRS.INI      MCRP12
M66         MPCRS.INI      MCRP12
M67         MPCRS.INI      MCRP12
M68         MPCRS.INI      MCRP12
M69         MPCRS.INI      MCRP12
M70         MPCRS.INI      MCRP12
M71         MPCRS.INI      MCRP12
M72         MPCRS.INI      MCRP12
M73         MPCRS.INI      MCRP12
M74         MPCRS.INI      MCRP12
M75         MPCRS.INI      MCRP12
M76         MPCRS.INI      MCRP12
M77         MPCRS.INI      MCRP12
M78         MPCRS.INI      MCRP12
M79         MPCRS.INI      MCRP12
M80         MPCRS.INI      MCRP12
M81         MPCRS.INI      MCRP12
M82         MPCRS.INI      MCRP12
M83         MPCRS.INI      MCRP12
M84         MPCRS.INI      MCRP12
M85         MPCRS.INI      MCRP12
M86         MPCRS.INI      MCRP12
M87         MPCRS.INI      MCRP12
M88         MPCRS.INI      MCRP12
M89         MPCRS.INI      MCRP12
M90         MPCRS.INI      MCRP12
M91         MPCRS.INI      MCRP12
M92         MPCRS.INI      MCRP12
M93         MPCRS.INI      MCRP12
M94         MPCRS.INI      MCRP12
M95         MPCRS.INI      MCRP12
M96         MPCRS.INI      MCRP12
M97         MPCRS.INI      MCRP12
M98         MPCRS.INI      MCRP12
M99         MPCRS.INI      MCRP12
M100        MPCRS.INI      MCRP12

```

CHINESE FISCAL AND TAXATION LONG-DISTANCE LEARNING

姚益平 主编



C 语言 程 序 设 计

姚益平 主编

TP312C/38.

湖南电子音像出版社



湖南大学现代远程教育
全国税务系统远程教育

系列教材

C 语言程序设计

姚益平 主编

策 划：谭慧渊 刘镜波 蒋菊香

责任编辑：杨许国 肖家红

装帧设计：黄 弋 赵 慧

湖南出版集团

湖南电子音像出版社出版发行

长沙市展览馆路 66 号 邮编：410005

湖南印刷一厂印刷

2002 年 2 月第 1 版 2002 年 2 月第 1 次印刷

开本：850 × 1168 1/32 印张：12

字数：272832 印数：8,000

ISBN 900352-03-1/G4 · 95

定价：90.00 元 / 套(光盘配书)

编写说明

现代远程教育是 20 世纪 80 年代以来国际教育发展的共同趋势。1998 年 9 月,教育部批准湖南大学等四所大学首批试办现代远程教育,标志着我国现代远程教育已正式启动。湖南大学的现代远程教育,在探索中不断前进,特别是与国家税务总局合作开办的主要面向行业的财税远程教育,在办学模式、教学手段等方面正在实现跨越式发展。

在全国税务系统远程学历教育领导小组的领导下和在全国税务系统远程学历教育教学指导委员会的指导下,我们根据湖南大学本科学历教育教学大纲和新形势下社会对财经类人才素质的要求,组织全国相关专业的著名教授、学者、专家编写了这套系列教材及学习指导书,并配有电子光盘、VCD 光盘、网络课件等教学资源。

本书由姚益平主编

由于时间原因,错漏之处在所难免,敬请同行专家批评指正。

目 录

第一章 绪论	1
第一节 C语言的发展和特点	1
第二节 C语言程序的格式和结构特点	4
第三节 C语言程序的开发过程	8
第四节 简单的输入与输出	10
第二章 数据与运算	15
第一节 常量	15
第二节 变量及其数据类型	21
第三节 变量的存储类型和使用范围	26
第四节 变量的初始化	31
第五节 数组	33
第六节 字符型数组与字符串	36
第七节 多维数组	38
第八节 赋值和算术运算	41
第九节 关系运算和逻辑运算	45
第十节 位操作	47
第十一节 其它运算与运算顺序	52
第三章 流程控制	57
第一节 结构化程序设计和流程控制	57
第二节 条件分支	63
第三节 开关分支	70
第四节 WLibe 循环	75

第五节	for 循环	80
第六节	do ~ while 循环	85
第七节	循环的中途退出	88
第八节	goto 语句与标号	95
第四章	指针	98
第一节	指针变量	98
第二节	指针的说明与初始化	102
第三节	指针运算	106
第四节	指针与数组	111
第五节	字符指针与字符串	116
第六节	指针数组	118
第七节	多级指针	124
第五章	函数	130
第一节	模块化软件与 C 语言程序结构	130
第二节	函数的定义和调用	134
第三节	函数间的参数传递	140
第四节	使用返回值和全局变量传递数据	148
第五节	数组与函数	151
第六节	字符串与函数	156
第七节	指针型函数	160
第八节	递归函数	165
第九节	指向函数的指针	168
第十节	编译预处理	175
第十一节	分割编译	183
第十二节	命令行变元	188
第六章	结构体和联合体	190
第一节	结构的定义和说明	190
第二节	结构的使用形式和初始化	196

第三节	结构数组	201
第四节	结构指针	204
第五节	结构在函数间的传递	208
第六节	结构型和结构指针型函数	212
第七节	结构嵌套	215
第八节	位结构	217
第九节	联合体	219
第十节	枚举(enum)	223
第十一节	类型定义	225
第七章	C 标准函数库	228
第一节	程序库与首标文件	228
第二节	I/O 函数	230
第三节	字符串函数与字符函数	275
第四节	数学函数	300
第五节	时间、日期及其它有关系统的函数	314
第六节	动态分配函数	320
第七节	其它函数	326

第一章 绪论

随着电子计算机的迅速发展和广泛应用,C 程序设计语言(简称 C 语言)在计算机软件开发中的作用日益重要,几乎各种型号的计算机上都配置有 C 语言编译系统。本章简要介绍 C 语言的历史和特点:C 语言程序的格式和结构特点,以及它的开发过程;此外,对 C 语言程序中经常使用的输入输出给予简单介绍。

第一节 C 语言的发展和特点

C 语言是一种编译型程序语言,它的前身是马丁·理查德(Martin Ricchards)在 60 年代开发的 BCPL 语言。

1970 年,肯·苏姆普逊(Ken Thompson)在软件开发工作中,继承和发展了 BCPL 语言的上述特点,从而提出了“B 语言”。此后,在美国贝尔研究所进行的更新型的小型机 PDP-11 的 UNIX 操作系统的开发工作中,戴尼斯 M.利奇(Dennis M.Ritchie)和布朗 W.卡尼汉(Brian W.Kernighan)对 B 语言做了进一步的充实和完善,于 1972 年推出了一种新型的程序语言——C 语言。

为了使 UNIX 操作系统推广,1977 年 Dennis M.Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。1978 年 Brian W.Kernighan 和 Dennis M.Ritchie 出版了名著《The C Programming Language》,从而使 C 语言成为目前世界上流行最广泛的高级程序设计语言。

随着微型计算机的日益普及,出现了许多 C 语言版本。由于没有统一的标准,使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况,美国国家标准研究所 (ANSI) 为 C 语言制定了一套 ANSI 标准,成为现行的 C 语言标准。

C 语言发展迅速,而且成为最受欢迎的语言之一,主要因为它具有强大的功能。许多著名的系统软件都是由 C 语言编写的。用 C 语言加上一些汇编语言子程序,就更能显示 C 语言的优势了。

归纳起来 C 语言具有下列特点:

1.C 语言简洁、紧凑、方便、灵活,C 语言一共只有 32 个关键字,9 种控制语句,程序书写自由,主要用小写字母表示,压缩了一些不必要的成分。

2.C 语言是处于汇编语言和高级语言之间的一种记述性程序语言。C 语言是比较靠近硬件与系统的,即与汇编语言比较接近。它把高级语言的基本结构和语句与低级语言的实用性结合起来,可以像汇编语言一样对位、字节和地址进行操作,而这三者是计算机最基本的工作单元。C 语言即有面向硬件和系统,像汇编语言那样可以直接访问硬件的功能;又有高级语言面向用户、容易记忆、便于阅读和书写的优点。

3.C 语言是一种结构化程序设计语言,即程序的逻辑结构可以用顺序、选择和循环三种基本结构组成。C 语言具有诸如 if-else、for、while、do~while、switch~case 等结构化语句,十分便于采用由顶向下、逐步细化的结构化程序设计技术。因此,用 C 语言编制的程序,具有容易理解、便于维护的优点。

4.C 语言是便于模块化软件设计的程序语言。C 语言程序的函数结构,十分利于把整体程序分割成若干相对独立的功能模块,并且为程序模块间的相互调用以及数据传递提供了便利。这一特点也为把大型软件模块化,由多人同时进行集体性开发的软件工程技术方法提供了强有力的支持。

5.C 语言具有种类丰富的运算符。C 的运算符包含的范围很广泛,共有种 34 个运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理。从而使 C 的运算类型极其丰富,表达式类型多样化,灵活使用各种运算符可以实现在其它高级语言中难以实现的运算。除一般高级语言使用的+、-、*、/四则运算及与 (AND)、或 (OR)、非(NOT)等逻辑运算功能外,还可实现以二进制位(bit)为单位的位与(&)、位或(|)、位非(~)、位异或(^)以及移动操作(>>、<<=等位运算。并且具有如 a++、b--等单项运算和+=、-=、*=、/=等复合运算功能。

6.C 语言程序中可以使用如 #define、#include 等编译处理语句,能够进行字符串或特定参数的宏定义,以及实现对外部文本文件的读取和合并。同时还具有 #if、#else 等条件编译预处理语句。这些功能的使用提高了软件开发的工作效率,并为程序的组织和编译提供了便利。

7.C 语言的数据类型丰富,具有现代化语言的各种数据结构。C 的数据类型有:整型、实型、字符型、数组类型、指针类型、结构体类型、联合体类型等,能用来实现各种复杂的数据类型的运算,且计算功能、逻辑判断功能强大。并引入了指针概念,使程序效率更高。另外 C 语言具有强大的图形功能,支持多种显示器和驱动器。

8.C 语法限制不太严格、程序设计自由度大。一般的高级语言语法检查比较严,能够检查出几乎所有的语法错误。而 C 语言允许程序编写者有较大的自由度,放宽了语法检查。例如,对数组下标越界不作检查,由程序编写者自己保证程序的正确性。对变量的类型使用比较灵活。例如,整型量与字符型数据以及逻辑型数据可以通用。

9.C 语言程序生成代码质量高,程序执行效率高,一般只比汇编程序生成的目标代码效率低 10%~20%。

10.C 语言程序具有较高的移植性。C 语言的语句中,没有依存于

硬件的输入/输出语句,程序的输入/输出功能都是通过调用输入/输出函数实现的。而这些函数是由系统提供的独立于C语言程序本身、并且不依存于机器硬件系统,从而便于在硬件结构不同的机种实现程序的移植。

由于C语言具有上述众多特点,近年来迅速地得到广泛普及和应用,特别是在微处理机和微型计算机的软件开发、以及各种软件工具的开发中,使用C语言的趋势日益增强,呈现出C语言有可能取代汇编语言的发展倾向。所以C语言被称为“高级汇编语言”。

第二节 C 语言程序的格式和结构特点

任何一种计算机程序语言,都具有特定的语法规则和一定的表现形式。程序的书写格式和程序的构成规则是程序语言表现形式的一个重要方面。按照规定的格式和构成规则书写程序,不仅可以使程序设计人员和使用程序的人容易理解,更重要的是把程序输入给计算机时,计算机能够充分认识从而能够正确执行它。

一、C 语言程序的格式

1.C 语言的保留字都是小写的。C 语言中区分大写与小写。在 C 程序中保留字不能用于其它目的,即不允许将保留字作为变量名或函数名使用。C 程序习惯上使用小写英文字母。

2.C 语言程序是由一个个语句组成。每个语句都具有规定的语法格式和特定的功能。

3.C 语言程序不使用行序号。但在本书中,为了解说方便,有时在给出的程序实例中加有序号。需要注意的是,向计算机输入这些程序时,不要输入这样的行序号。

4.C 语言程序使用分号“;”作为语句的终止符或分隔符。

5.C 语言程序不存在程序行的概念。一个程序可以自由地使用任意的书写行,即一行中可以有多个语句,一个语句也可以占用任

意多行,但需要注意语句之间必须用“;”分隔。6.C 语言程序中用大括弧{ }表示程序结构层次范围。一个完善的程序模块要用一对大括弧表示该程序模块的范围。此外,程序体中若干结构化语句,如 if、for、while、else、do 等,常常由若干语句组成语句块,这样的语句也要求用大括弧包围,以表示该结构的范围。

7.C 语言程序中,为了增强可读性,可以使用适量的空格和空行。但是,变量名、函数名以及 C 语言本身使用的单词(在 C 中称为保留字,如 if、for、while、char、int...等),不能在中间插入空格。除此之外的空格和空行是可以任意设置的,C 编译系统无视这样的空格和空行综上所述,C 语言程序的书写格式自由度较高、灵活性很强,有较大的任意性。但是,为了避免程序书写的层次混乱不清,便于人们阅读和理解,一般都采用有一定格式的习惯书写方法。这样的书写格式并不是计算机要求的,而是为了给人们提供便利。

C 语言程序书写格式的要点是:

- 1.一般情况下,每个语句占用一行。
- 2.同结构层次的语句,从不同的起始位置开始,即在同一结构层次中的语句,缩进同样的字数。
- 3.表示结构层次的大括弧,写在该结构化语句第一个字母的下方,与结构化语句对齐,并占用一行。
- 4.语句中不同单词间可以加有空格,程序不同部分的行间可以加有空行。

采用这样格式书写的程序,结构层次清晰,充分体现了结构化程序的特点,十分便于阅读和理解。除了这里介绍的书写格式外,还有一些其它的常用书写格式,但都大同小异。

二、C 语言程序的结构特点

一个完整的 C 语言程序是一个或多个具有相对独立功能的程序模块结合而成的。这样的程序模块称为函数。从格式上看,每个函数是由函数名和大括号对{}包围的若干语句组成的。为了更直观地

了解 C 语言程序的结构特点,下面再给出一个程序例子。

例 1.1 C 语言程序结构特点

```
1: /* print string as upperccase */
2: #include <stdio.h>
3: #define SIZE 80
4: main()
5: {
6: char str[SIZE];
7: int i;
8:
9: gets(str);
10: for(i=0; str[i] != '\0'; i++)
11: putupper(str[i]);
12: }
13: putupper(ch)
14: char ch;
15: {
16: char cc;
17:
18: cc=(ch>='a' && ch <='z') ? ch+'A'-'a':ch;
19: putchar(cc);
20: }
```

可以看出,上述程序是由名字称为 main 和 putupper 的两个函数组成。在组成 C 语言程序的函数中必须有一个且只能有一个名字为 main 的函数,它叫做主函数。除主函数之外的函数由用户命名,如上例中的 putupper 函数。

C 语言程序的执行是从主函数开始的,主函数中的所有语句执行完毕,则程序执行结束。如例 1.1 的程序是从第 5 行的“{”开始

的,执行到第 12 行的“}”结束。在执行第 11 行的语句时,程序控制转移到函数 putupper 中。执行完函数 putupper 的语句后,再返回主函数中继续运行。这个转移过程叫做调用函数 putupper。因此看出,C 语言程序中,main 函数之外的其它函数都是在执行 main 函数时,通过嵌套调用而得以执行的。在程序中除了可以调用用户自己编制的函数外,还可以调用由系统提供的标准函数,如例 1.1 中的 gets 和 putchar 等都是标准函数。

程序前部带有 # 号的语句是编译预处理语句,如例 1.1 中第 2 行和第 3 行的语句。关于它们的作用在第五章中详细讨论。和其它程序语言一样,C 语言程序中也可以使用注释。注释部分的格式是:

```
/* 注释内容 */
```

如例 1.1 中的第 1 行就是注释部分。注释可以加在程序的任意部位,它可以占用一行以上的位置,也可写在语句的后面。注释部分作为源程序的一个部分驻留在源程序清单中,但对源程序进行编译时,系统将无视注释部分。

C 语言函数的一般格式如下所:

```
函数名(参数)
参数说明;
{
    数据说明部分;
    执行语句部分;
}
```

有的函数不带参数,则参数说明部分也就不存在了。但函数名后的圆括号不能省略。如例 1.1 中 main() 函数不带参数,而 putupper

0函数有一个参数 ch,第 14 行是该参数的说明。第 6、7 行和第 16 行分别是两个函数内的数据说明部分,而第 9、10、11 行和第 18、19 行是执行语句部分。

C 语言程序的函数模块结构特点,使得程序整体结构分明、层次清楚,它为模块化软件设计方法提供了有力的支持。关于程序各部分的意义及功能,后面各章中将陆续讨论。

第三节 C 语言程序的开发过程

使用 C 语言开发程序时的基本过程如图 1-1 所示。

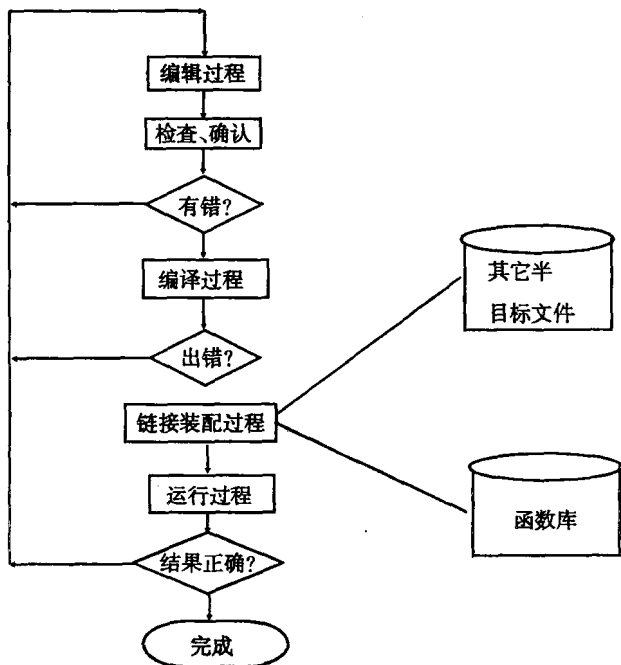


图 1-1 C 语言程序的开发过程

一、源文件的编辑

程序开发人员把自己编制的 C 语言程序输入给计算机时,首先要使用编辑程序(又称为编辑器)建立 C 语言程序的源文件。建立后的源文件以文本文件形式存储在计算机的文件系统(软盘或硬盘)中。源文件的名字由用户选定,但要求有一定的后缀“c”。例如:

file.c

program.c

test.c

二、编译过程

在对源文件进行编译前,要再次进行检查和确认。发现问题后,则重新启动编辑程序对源文件进行修改。确认正确后,则开始对源文件进行编译。

编译过程是通过编译程序(又称为编译器)进行的。系统对源文件进行编译的同时,还对源文件的语法和程序的逻辑结构等进行检查。当发现错误时,可指出错误的位置和种类。此时要再次使用编辑程序进行排错修正。

三、链接过程

编译后产生的半目标数据集是可再定位的程序模块,不能直接用于运行。半目标文件要经过链接装配过程,把它和其它半目标程序模块,以及系统提供的标准函数库等进行链接后,才生成可以运行的可执行目标文件。链接过程也是使用系统提供的链接程序(又称链接装配器)进行。生成的可执行数据集也暂时存储于文件系统中。

可执行数据集生成后,就可以在计算机操作系统的支持下运行。若执行结果达到预期的目的,则开发工作到此完成。否则,要进一步检查修改源程序,再经过编辑-编译-链接的过程,直到取得正确的运行结果为止。

以上介绍了 C 语言程序的一般开发过程。读者在从事实际的

程序开发工作中,应该参考系统提供的编辑器,编译系统和链接器的有关使用手册,按照规定的命令格式,选择合适的选择项完成开发过程。

第四节 简单的输入与输出

制代码和一个完整的计算机程序,常常要求具备输入输出功能。C语言本身没有配备完成输入输出的语句。C语言程序的输入输出功能是通过调用系统提供的标准函数实现的。在全面学习C语言程序设计之前,为了便于以后各章内容的叙述和讨论,本节首先简单介绍两种经常使用的输入输出标准函数。

一、格式化输入输出函数

格式化输入输出函数是按指定的格式完成输入输出过程。其中 `printf()` 函数的功能是向标准输出设备上输出信息。

1、输出函数 `printf()`

输出函数 `printf()` 的一般使用形式如下:

`Printf("输出格式",输出项系列);`

如使用该函数

```
printf("v=%f\n",v);
```

其中“`v=%f \n`”是给定的输出格式,而 `v` 是输出项,它们之间用逗号分隔。

`printf()` 函数的功能是按照给定的输出格式、向标准输出设备输出信息。输出格式中用%打头后面跟有一个字母的部分称为转换说明符,它规定了输出项的输出形式。常用的转换说明符及其意义如下所示: