精通C语言速成

● 李华凯　谢兴明　编著 ●

中国计量出版社

精通C语言速成

李华凯　谢兴明　　编著

中国计量出版社

(京)新登字 024 号

图书在版编目(CIP)数据

精通 C 语言速成/李华凯　谢兴明编著. —北京中国计量出版社,1995.7
ISBN 7-5026-0809-5

Ⅰ.精…　Ⅱ.李…　Ⅲ.表处理-编译程序　Ⅳ.TP314

中国版本图书馆 CIP 数据核字(95)和 10044 号

精通 C 语言速成

李华凯　谢兴明　　编著

*

中国计量出版社出版
北京和平里西街甲 2 号
邮政编码　100013
河北永清第一胶印厂印刷
新华书店北京发行所发行

*

开本 787×1092/16　印张:18　　字数:415 千字
1995 年 9 月　第 1 版　1995 年 9 月　第 1 次印刷

*

印数 1—6000　　**定价:23.00 元**

目　录

第一章 C语言简介

1.1 C语言的历史

C语言是Dennis Ritchie于1972年在AT&T的贝尔实验室为了完成UNIX操作系统而开发出来的计算机语言。

C语言可以用在编写编译程序、操作系统、文字处理、数值计算以及数据库管理等方面。

1.2 C语言的特性

C语言具有以下特性：

具有高级语言的方便和低级语言的功能

高级语言就是比较接近人类日常用语的计算机语言。例如：BASIC、FORTRAN、COBOL与PASCAL等，这种语言比较容易学习和使用。

低级语言就是比较接近计算机内部用语的语言。例如：汇编语言，通常我们都使用低级语言来控制计算机的硬件与外围设备。例如：键盘与屏幕。

移植性高

所谓移植性高就是在某一台计算机上所编写的程序，只要稍加修改，就可以在另外一台计算机上执行。例如在IBM PC上编写的C程序，只要稍加修改，就可以在其他系统的计算机上执行。

执行速度快

处理同样一个问题，用其他计算机语言所写成的程序其执行时间可能是C程序的几倍。

1.3 解释程序与编译程序

高级语言需要通过一个解释程序，将人们下达的命令翻译成计算机看得懂的机器语言，翻译程序有下列两种：

- 解释程序
- 编译程序

解释程序

解释程序每次只翻译一行语句，并马上送给计算机执行，接着再继续翻译下一行语句。每

次执行之后的结果会立刻显示在屏幕上。传统BASIC就是使用解释程序来翻译的语言。

解释程序的优点是操作简单，并且在程序执行的过程中会随时指出程序的错误，特别适合初学者使用。其缺点是每次使用程序时都要再重新翻译，比较浪费时间。

编译程序

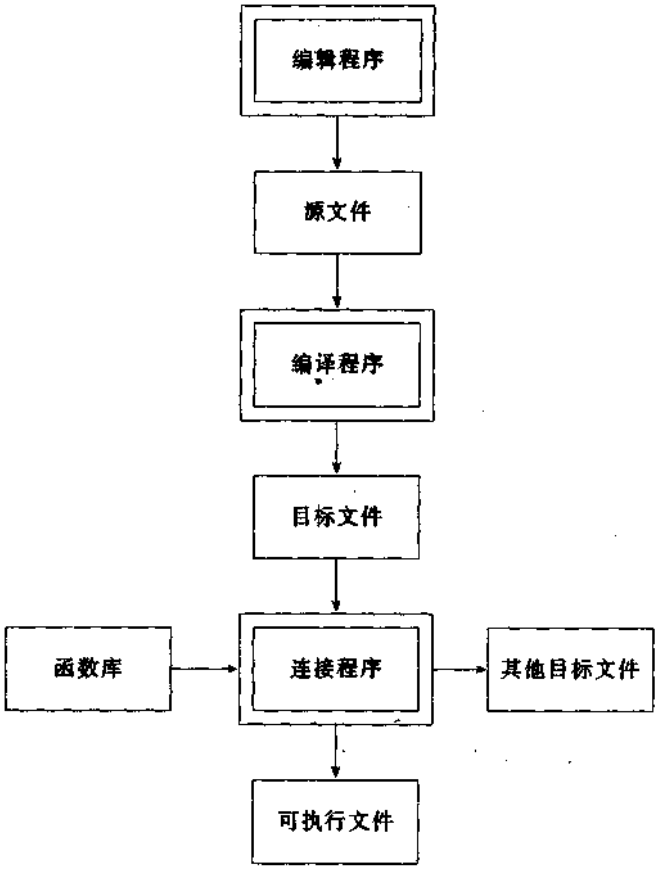
在编辑程序中所写成的程序称为源程序或源文件。编译程序会将整个源文件一次编译完毕，这种经编译程序翻译后的文件称为目标文件。C语言就是使用编译程序来翻译的语言。

编译程序的优点是一旦产生目标文件，以后要再执行程序时，不需要再重新翻译，比较节省时间。其缺点是操作过程比较复杂。

1.4 函数库与连接程序

函数库中存放着一般函数、数学函数与控制外围设备的输入/输出函数。

连接程序是用来将一个或多个目标文件与函数库连接起来，以产生一个可执行文件。只有可执行文件才可以直接在计算机上执行。

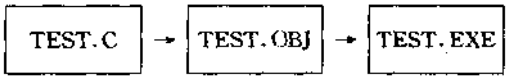


1.5 文件名称

文件名称的一般格式如下：

主文件名.扩展名

主文件名不能超过8个字符，扩展名不能超过3个字符。源文件的扩展名必须为C，目标文件的扩展名计算机会自动设定为OBJ，可执行文件的扩展名计算机会自动设定为EXE。例如：



源文件　　目标文件　　可执行文件

1.6 C程序的结构

一个C程序的结构如下：

```
/* a simple program */
main()
{
  printf ("Good bye my love.");
}
```

执行结果

Good bye my love.

说明

- /* a simple program */

  注解是以/*开始，以*/结束。我们常用注解来说明程序内容的意义，以增加程序的可读性。

- main()

  C程序是由一个或多个函数所组成，并由main()函数开始执行。()符号表示函数的意思。

- {

  左大括号表示main()函数的开始，C程序的内容都写在左大括号之下。

- printf()

  printf()是一个输出函数，其功能是将数据显示在屏幕上或从打印机输出。

- "Good bye my love."

  C的字符串要用双引号(")括起来，表示它是printf()所要输出的内容。

- ;

  分号是语句的结束符号，程序中的每一个语句后面都必须有分号。

- }

  右大括号表示main()函数的结束。

第二章 常数、变量和数据类型

本章将分别叙述常数、变量和数据类型，下面首先从数据开始。

2.1 数据

数据包括常数与变量两种。常数具有固定不变的值，而变量的值会随着程序的执行而改变。例如：

num＝3；

其中3是常数而num是变量。

2.2 常数

常数包括下列几种：

整数常数

整数常数是不带小数点的数，整数常数有下列三种：

- 八进制整数常数
- 十进制整数常数
- 十六进制整数常数

十进制整数常数可以使用0～9这10个数字，但首位数字不可以是0。例如：25、－320。八进制整数常数可使用0～7这8个数字，但首位数字必须是0。例如：015、－023。十六进制整数常数可以使用0～9与A～E(或a～e)这16个数字，但首位数字的左边必须加上0X或0x。例如：0X2A、0xE3。

整数常数的范围介于－32768和32768之间。在IBM PC中，Turbo C是以2个字节(16个位)来存储整数常数。如果使用的整数常数超出上面的范围时，可在此整数常数的最右边加上L或l，使其转换成长整数常数。例如：78943L或－78943l。长整数常数的范围介于－2147483648和2147483647之间，并以4个字节(32个位)来存储。

浮点数常数

浮点数常数就是带有小数点的数。浮点数常数有下列两种表示法：

- 十进制表示法

例如：32.45、－0.783

- 科学计数表示法

例如：1.035E＋07、6.52e－05

浮点数常数的范围介于3.4E－38和3.4E＋38之间，并以4个字节来存储。

双精度浮点数常数

双精度浮点数常数的范围介于 1.7E－308 和 1.7E＋308 之间，并以 8 个字节来存储。

字符常数

字符常数是由一个字符与单引号(')所组成，例如：'A'、'b'、'5'、'＊'。有些字符是无法显示在屏幕上或从打印机上输出的。例如按〈Enter〉键或〈Tab〉键所产生的字符，要使用这种字符，必须在其前面加上反斜杠(\)。

字符	用　途
\n	换行(使光标跳到下一行的开头)
\t	水平跳格(相当于按 Tab 键)
\v	垂直跳格
\b	退一格
\r	归位(相当于 Enter 键)
\f	跳页
\\	输出\字符
\"	输出"字符

字符常数是以 1 字节来存储的。

每一个字符都有一个整数与之对应以代表这个字符，这个整数称为这个字符的 ASCII 码。例如：A 的 ASCII 码为 65，而 a 的 ASCII 码为 97。

字符串常数

字符串常数是由零个或多个字符，并加上双引号(")所组成。例如："You are my darling."、"BMW－318i"。

Turbo C 是以一个字节来存储字符串常数中的每一个字符，且会在字符串常数的后面自动为其加上一个空字符(\0)以表示字符串的结束。所以一个有 N 个字符的字符串常数需要 N＋1 字节的存储位置。例如字符串常数"ABCD"，需要 5 个字节的存储位置。

A	B	C	D	\0

2.3 变量

变量名称

变量名称的命名规则如下：

- 必须由英文字母、数字或下划线(_)所组成。
- 第一个字符必须是英文字母或下划线。
- Turbo C 编译程序认为大写与小写字母是不同的，所以 cont 与 CONT 是两个不同的变量。习惯上是以小写来表示变量名称，而用大写来表示常数名称。

• Turbo C 可以辨认变量名称的前 32 个字符。

例如:正确的变量名称:

1. count25
2. test_15

例如:错误的变量名称:

1. 25count
2. test－15

变量的定义

所有的变量在使用之前都必须定义,以指定变量的类型与名称。例如:

```
int a,b;
char ch;
```

变量之间是以逗号隔开,并以分号作为定义语句的结束。上面的定义方式也可以写成:

```
int a;
int b;
char ch;
```

使用这种定义方式的目的是:容易在语句的右边加上变量的注解。

变量初始值的设定

我们可以在定义变量的同时给变量设定初始值,其方法是在变量名称的后面加上一个等号与常数即可。例如:

```
int a=5;
char ch='A',CW='$';
```

上述的定义方式也可以写成:

```
int a;
char ch,cw;
a=5;
ch='A';
CW='$';
```

2.4 数据类型

C 语言的基本数据类型有下列四种:

char	字符类型	用来存放字符数据
int	整数类型	用来存放整数数据
float	浮点数类型	用来存放浮点数数据
double	双精度浮点数类型	用来存放双精度浮点数数据

2.5 数据类型修饰符

基本数据类型的前面可以加修饰符，以改变数据类型的长度或意义。C 语言的修饰符有下列四种：

- signed　　具有正负号
- unsigned　　不具有正负号
- long　　长形
- short　　短形
- signed、unsigned、long 与 short 修饰符可以用在整数与字符数据类型中，例如：unsigned int、long char。
- long 也可以用在 double 数据类型中，例如：long double。
- 使用修饰符时，后面的 int 可以省略不写，例如：short int 也可以写成 short。
- 对整数数据类型使用 signed 是多余的，因为 Turbo C 缺省的整数即具有正负号。
- 在 IBM PC 中，Turbo C 是以 4 个字节来存储 long int。
- 在 IBM PC 中，Turbo C 是以 2 个字节来存储 short int(有的大型机器是用 4 个字节)。
- 常用的数据类型在 IBM PC 中其字符长度与范围如下：

数据类型	字符长度	范围
char	8	0～255
int	16	－32768～32768
unsigned int	16	0～65535
short int	16	－32768～32767
long int	32	－2147483648～2147483647
float	32	3.4E－38～3.4E＋38
double	64	1.7E－308～1.7E＋308

第三章 输入与输出

C语言没有输入与输出语句，而是使用函数库中的函数来执行输入与输出数据。常用的输入与输出函数如下：

- printf()函数：将数值、字符或字符串显示在屏幕上。
- scanf()函数：读取从键盘输入的数值、字符或字符串。
- getchar()函数：读取从键盘输入的一个字符。
- putchar()函数：将一个字符显示到屏幕上。

严格的说，上述函数都不属于C语言本身，而只属于C语言的标准输入与输出函数，但由于C语言本身就没有输入与输出指令，所以久而久之人们就自然的称以上函数是C语言的输入与输出指令。

3.1 printf()函数

printf()函数的一般格式如下：

printf("控制字",参数1,参数2,……);

printf()函数会按照控制字符串所规定的格式，将数据转换规格后显示到屏幕上。控制字符串包括下列两种字符：

- 一般字符
- 转换规格

一般字符

一般字符包括A～Z、a～z、0～9、#、*、! 等。printf()函数会将一般字符原封不动地显示到屏幕上。

转换规格

printf()函数会将数据(变量)按照其所对应的转换规格所规定的格式显示到屏幕上。转换规格是以%开头，并以转换字符结尾。printf()函数常用的转换规格如下：

转换规格	输出格式
%d	十进制整数
%u	没有正负号的十进制整数
%o	没有正负号的八进制整数
%x	没有正负号的十六进制整数
%f	以十进制表示的浮点数

（续表）

转换规格	输出格式
%e	以科学计数法表示的浮点数
%c	字符
%s	字符串

转换规格的修饰符

转换规格加上修饰符后的一般格式如下：

%[－][w][.P]转换字符

－（负号）

C 语言输出数据时是向右靠齐，加上负号可以使输出数据改为向左靠齐。

w（栏宽）

w 是用来设定数据输出的栏宽。例如：%5d、%10f、%8e、%25s。若输出数据的长度超过所设定的栏宽时，仍然会以输出数据的实际长度输出。若输出数据的长度小于所设定的栏宽时，则输出数据以向右靠齐的方式输出。

.P（精确值）

- 设定以十进制表示的浮点数所输出的小数点以后的位数。
- 设定以科学计数表示的浮点数所输出的有效数字。
- 设定字符串输出的字符个数。

例如：

%10.3f 栏宽 10 位，输出小数点以后 3 位。

%10.3e 栏宽 10 位，输出 3 位有效数字。

%25.6s 栏宽 25 位，只输出字符串的前 6 个字符。

下面通过具体程序实例说明 printf()函数的用法：

例 1：将字符串 Introduction to C language 打印两次，且将它打印在同一行中。

```
/* Print string 2 times in a line. */

void main()
{
   printf("Introduction to C language.");
   printf("Introduction to C language.");
}
```

执行结果

Introduction to C language. Introduction to C language.

在例 1 中，我们可以看到字符串 Introduction to C language 在同一行中打印了两次。C 语言提供了一种控制字符，可让我们将上述字符串分别打印在不同的两行中，这个控制字符是\n，这个字符主要的目的是指示输出部件跳纸打印输出字符。

例 2：重复打印字符串 Introduction to C language，但是将它分两行打印出来。

```
/* Print string 2 times in a line. */

void main()
{
   printf("Introduction to C language. \n");
   printf("Introduction to C language. \n");
}
```

执行结果

Introduction to C language.
Introduction to C language.

例3：打印字符串 Introduction to C language 两次，但是按不同的格式将它打印出来。

```
/* Another output instruction. */

void main()
{
   printf("Introduction\n to C language. \n");
   printf("Introduction to\n C language. \n");
}
```

执行结果

Introduction
  to C language.
Introduction to
  C language.

例4：基本整数输出的实例应用，本程序将会打印 exercise ch34，但是3和4分别用整数变量将它打印出来。

```
/* Print the string exercise ch34. */
void main()
{
   int i,j;
   i=3;
   j=4;
   printf("exercise ch%d  %d.c\n",i,j);
}
```

执行结果

exercise ch34

例5：格式化输出某一整数变量值的应用。

```
/* Formatted the integer output. */

void main()
```