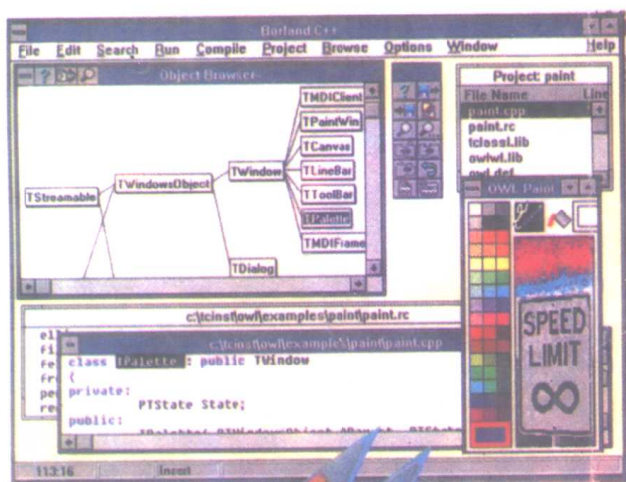


实用 C 语言入门



游志男 赖森禄 编著

科学出版社

希望

实用 C 语言入门

游志男 赖森禄 编著

雪 燕 改编

亦鸥 希望 审订

科 学 出 版 社

1993

(京)新登字 092 号

内 容 简 介

本书是学习和使用 C 程序设计语言的入门读物,书中以大量的实例来介绍 C 程序设计语言,并使用了大量的计算机专业英语,是一本内容新颖的入门读物。

本书可供从事计算机软件设计、开发和应用的技术人员学习参考。

需要本书的用户,可与北京 8721 信箱联系。电话 2567329, 邮码 100080。

版 权 声 明

本书繁体字中文版原名为《实用 C 程式语言入门》由松岗电脑图书资料股份有限公司出版。版权归松岗公司所有。本书简体字中文版版权由松岗公司授予北京希望电脑公司,由北京希望电脑公司和科学出版社独家出版、发行。未经出版者书面许可,本书的任何部分不得以任何形式或任何手段复制或传播。

实用 C 语言入门

游志男 赖森禄 编著

雪 燕 改编

亦鸥 希望 审订

责任编辑 张建荣

科学出版社 出版

北京东黄城根北街 16 号

邮政编码,100717

双青印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

★

1993 年 12 月第 一 版 开本,787×1092 1/16

1993 年 12 月第一次印刷 印张,17 5/8

印数,1-4 000 字数,401 000

ISBN 7-03-004019-8/TP·322

定价:19.00 元

目 录

第零章 C语言概述	1
0.1 C语言的来源	1
0.2 C语言的特性	1
0.3 C语言是时势的潮流	2
0.4 结论	3
第一章 准备工作	4
1.1 准备工具	4
1.2 第一个程序	4
1.3 编程技巧	5
1.4 编译、执行第一个程序	6
1.5 C语言编译工作流程	7
1.6 介绍基本元件——字符集	8
1.7 结论	12
1.8 综合练习	13
1.9 第一章综合练习解答	13
第二章 程序结构(一)——声明和表达式	14
2.1 Declarations(声明语句)	15
2.2 Expressions(表达式)	18
2.3 综合练习	30
2.4 第二章综合练习解答	31
第三章 基本函数调用	32
3.1 Library Function Reference Manual(库函数参考手册)	32
3.2 printf() function	33
3.3 scanf() function	39
3.4 getch() function 与 getche() function	42
3.5 putch() function	43
3.6 getchar() 与 putchar()	44
3.7 gets() function	45
3.8 结论	46
3.9 综合练习	46
3.10 第三章综合练习解答	47
第四章 程序结构(二)	49
4.1 if statement	49
4.2 while statement	54

4.3	do...while statement	58
4.4	for statement	61
4.5	switch...case statement	65
4.6	break 与 continue statement	69
4.7	结论	72
4.8	综合练习	72
4.9	第四章综合练习解答	74
第五章	函数	78
5.1	Function 用途	78
5.2	Function Definitions(函数定义)	79
5.3	Function Declarations(函数声明)	81
5.4	Function Prototype	82
5.5	传值方式——call by value	85
5.6	Recursion(递归)	89
5.7	Storage class、Variable Lifetime 与 Visiable	94
5.8	Data Type Conversions(数据类型转换)	100
5.9	结论	102
5.10	综合练习	102
5.11	第五章综合练习解答	103
第六章	预处理器	106
6.1	Preprocessor Directives(预处理命令)	106
6.2	#define 与 #undef	106
6.3	include file	113
6.4	Conditional Compilation(条件编译)	115
6.5	Other Preprocessor Directives	118
6.6	结论	119
6.7	综合练习	119
6.8	第六章综合练习解答	120
第七章	数组和字符串	122
7.1	为什么需要使用 Array	122
7.2	Array Decalations(数组声明)	123
7.3	Array Initializations (数组初值设定)	128
7.4	Multi-dimensional Arrays(多维数组)	130
7.5	Array 的 memory address 表示	133
7.6	Array as function argument(作函数参数的数组)	136
7.7	String(字符串)	139
7.8	常用的 String Library Function	145
7.9	结论	154
7.10	综合练习	155

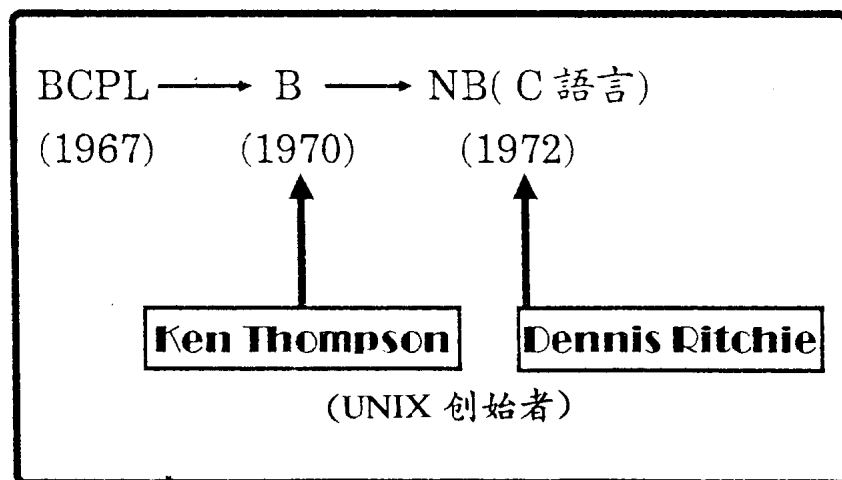
7.11 第七章综合练习解答	156
第八章 Pointer(指针)	159
8.1 Pointer 概论	159
8.2 Pointer variable Declarations(指针变量声明)	159
8.3 Pointer Assignment(指针赋值)	160
8.4 Pointer as Arguments-call by address	162
8.5 Pointer point to Pointer(双重指针)	167
8.6 Pointer And Array(指针与数组)	170
8.7 Pointer And String(指针与字符串)	178
8.8 Pointer And Function(指针与函数)	184
8.9 Pointer And Absolute address(指针与绝对地址)	189
8.10 结论	194
8.11 综合练习	194
8.12 第八章综合练习解答	196
第九章 Structure、Union 与 Enum	200
9.1 Structure declarations	200
9.2 Structure Initialization(结构初值设定)	203
9.3 Nested Structure(嵌套结构)	207
9.4 structure Array(结构数组)	208
9.5 Pointer 与 Structure	214
9.6 Structure As Argument	215
9.7 Dynamic Data Structure(动态数据结构)	221
9.8 unions(联合)	251
9.9 自定义 data type——num(枚举)	252
9.10 结论	256
9.11 综合练习	256
9.12 第九章综合练习解答	258
附录 A Microsoft C V6.0 Setup	262
附录 B Turbo C 2.0 Installation	266
附录 C 内存模式	270
附录 D ANSI C	272

第零章 C 语言概述

以往国内的信息工业大致分为两大类:工程类与商业类。工程类的程序语言着重于 FORTRAN 语言,而商业类则着重于 COBOL 语言。今天,C 语言将逐渐取代它们的地位。

0.1 C 语言的来源

C 语言的鼻祖是一种称为 BCPL (Basic Compound Programming Language)的组合语言。1970 年 Ken Thompson 为了在 UNIX 操作系统上,设计 FORTRAN 语言的 Compiler,从 BCPL 定义出另一套语言,称为 B 语言。B 语言由于太过于简单,只适用于较低的层次,对于解决特定的问题,受到相当的限制,于是又由 Dennis Ritchie 于 1972 年在贝尔实验室,将 B 语言修改为 NB 语言,也就是后来的 C 语言。

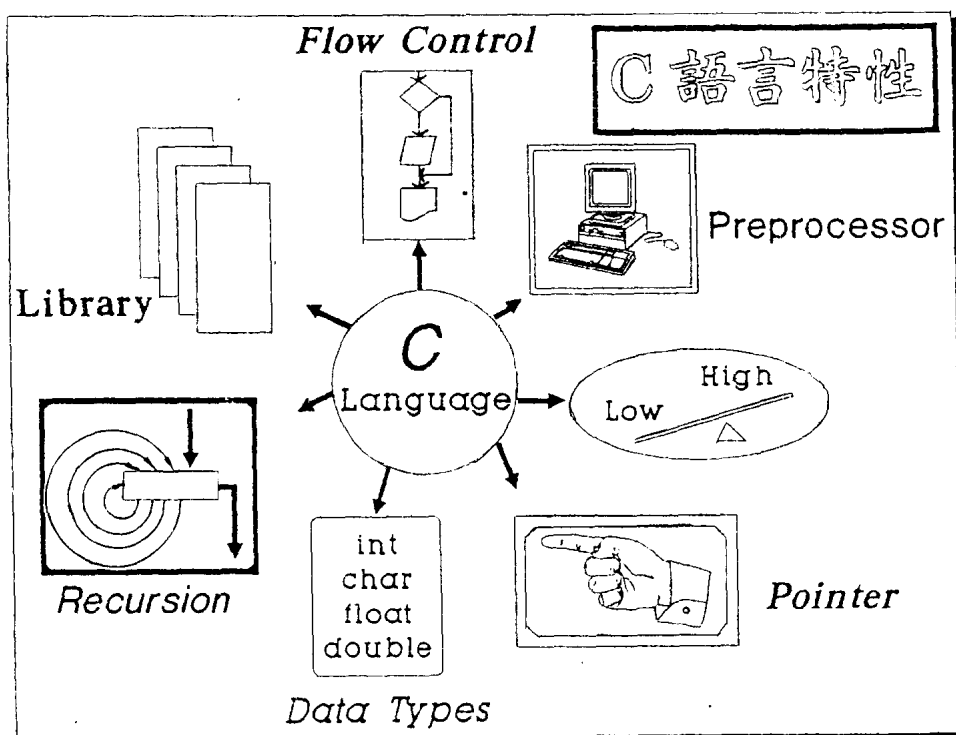


C 语言简史

0.2 C 语言的特性

- C 语言提供了丰富的 Library (库函数),供程序设计者使用,并且可以让程序设计者自己来设计库函数,以加强他们对应用程序的开发。
- C 语言提供多种而且具有弹性的数据类型,让程序设计者可以不受束缚。
- C 语言具有对位处理的低层功能,同时具有高级语言的程序语句。
- C 语言具有强有力的结构化流程控制。

- C 语言提供指针变量,对内存直接管理。指针变量可以用于算术运算。
- C 语言提供 recursive (递归)处理能力。
- C 语言提供 Preprocessor (预处理器)功能,使程序具有 macro (宏)指令,也可以提高程序的可移植性。
- C 语言是一种通用的程序语言,也相当适合于系统程序的开发。
- 对于只能用结构化语言处理的工作,C 语言几乎都可以处理。
- 具有结构化特性的 C 语言,使用简易的语法将程序模块化(modular),使程序易懂易维护。



0.3 C 语言是时势的潮流

C 语言的兴起,源于 UNIX 操作系统。因为 UNIX 系统的百分之九十都是由 C 语言所写的。由于 C 语言具有系统程序语言及一般通用程序语言的双重特性,许多专业软件公司,纷纷在不同的操作系统上开发 C Compiler,以争夺广大的用户市场。可见 C 语言流传相当广泛。用各式各样 C 语言开发的套装软件,如雨后春笋般地纷纷出笼,像数据库管理系统、电子表、绘图软件、CAD/CAM 应用软件、文书处理、OA(办公室自动化)、网络系统及科学、工程应用软件等等。C 语言将伴随着 UNIX 系统,逐渐走入明日的信息世界。

0.4 结论

本章介绍 C 语言的简史及其丰富的特性。本书将针对这些特性逐一说明,相信读者短时间内就能掌握 C 程序的基本语法并运用自如,进而学会如何处理丰富的程序库,建立实用的工具,提高效率成为优秀的程序设计师。

第一章 准备工作

这一章,首先让我们来做一些准备动作。前半段,准备应有的工具;后半段,了解 C 的基本字符。

1.1 准备工具

软件

一套 C COMPILER,如:Microsoft C、Turbo C。一套文本处理器,如:PE2、WORDSTAR 等等。

注:MS-C 6.0 PWB 内含文本处理器;
Turbo C IDE 内含文本处理器。

操作系统

如 MS-DOS 3.3、MS-DOS 4.01 等等。

硬件

一部 80X86 系列主机。如:IBM PC/XT 或 AT 或兼容主机。
Turbo C 最好配有硬盘;
Microsoft C V6.0 则需要至少 6M bytes 以上硬盘空间。

参考手册

一本 C COMPILER 软件公司的程序库参考手册。
一本 DOS 使用手册。
一本 PC BIOS 参考手册。

1.2 第一个程序

程序名称: first. c

```
void main (void)
{
    printf ("HELLO. WORLD !");
}
```

执行结果:

屏幕显示 HELLO, WORLD!

程序说明:

1. void main(void) Function—main 表示主程序。
 写在 main() 之前的 void: 表示 Function 无返回值;
 写在 main() 之内的 void: 表示 Function 不传 Arguments(参数);
 传统 C 语言 Function 书写格式: 不写 void;
 ANSI C 语言 Function 书写格式: void 要写;
 * 请参阅第五章。
2. 注意: 大小写英文字母, 在 C 语言中是不相同的。
 如 main、MAIN、Main、mAin 在 C 语言中, 是截然不同的。
3. printf() Function 调用, “HELLO, WORLD!” 称为 Function 的 argument。printf() 的目的是将 argument 数据输出。
4. 注意 Statement (语句) 的结束符“;”。
5. 左右大括号“{”, “}”, 表示 Block (程序模块) 的开始与结束。Block 内还可以包含另一个 Block。

main() function

C 程序事实上都是由 Function (函数) 所组成。而 Function 的形式很简单, 如下所示:

```
void main(void)
...
{
    Function 内容;
...
}
```

main() 这一个 Function, 是 C 语言固定要有的 Function。它表示整个程序的起点, 也就是程序执行的开头。当然, 你可以在程序里编写自己定义的其它 Function (往后还有很多范例)。但是, 可千万别忘记 main()。

另外, 当你购买 C Compiler 时, 磁盘内都会含有很多 Function 供你调用, 我们称这些 Function 为 Library Function (库函数)。第一个程序内的 printf() 就是 Standard Library Function 内的一个 Function。本书的后面章节还会提到。

1.3 编程技巧

C 语言的书写是自由格式。

例一:

```
void main(void)
```

```
{
    printf("Hello!")
}
```

例二:

```
void main(void) {printf("Hello!");}
```

自由格式就是不限行号位置。所以对 C Compiler 来说, (例一) 与 (例二) 是完全一样的。但是, 以程序设计者观点来看, (例一) 会使看程序的人较容易看懂。写程序的人应有一份共识, 为了自己以后程序的维护方便及其他的人能阅读, 应该在程序适当位置加上注解 (例三)。另外值得注意一点, 就是设定名称时应该尽量选取有意义名称, 如此可以增加程序的阅读性, 亦可提高编程技巧。

例三:

```
/* 程式名称:pgm01.C */
/* 用 途:讲解范例 */
/* 备 注:          */
void main(void) /* 主程序 */
{
    printf("Hello!"); /* Function Call */
} /* 主程序结束 */
/* ... */:表示注解
```

1.4 编译、执行第一个程序

编译程序, 我们将介绍 Turbo C 2.0 及 MS-C 6.0 的编译方法。有关这两种编译器的安装请参阅附录 (A) (B)。

Turbo C V2.0 Command Line 编译

Turbo C 的编译有两个方法: 1. 采用集成式开发环境。2. Command-line 编译法。

本书在此仅介绍 command-line 的编译方式。有关集成式开发环境, 请参阅 Turbo C 的使用手册。

Microsoft C V6.0 Command Line 编辑

使用 Microsoft C V6.0 版本的用户, 请参阅如下用法。

```
C:\C600>cl first.c
```

```
Microsoft(R) C Optimizing Compiler Version 6.00
```

```
Copyright(c) Microsoft Corp 1984-1990. All rights reserved.
```

```
first.c
```

```
Microsoft(R) Segmented-executable Linker Version 5.10
```

```
Copyright(c) Microsoft Corp 1984-1990. All rights reserved.
```

```
Object Modules[.OBJ];first.obj/farcall
```

Run File [first.exe]; "first.exe"/noi
List File [NUL.MAP];NUL
Libraries [.LIB];
Definitions File [NUL.DEF];;
C:\C600>first
HELLO, WORLD!

1.5 C 语言编译工作流程

图 1.1 介绍 C 语言在 IBM PC 的环境下的工作流程图。

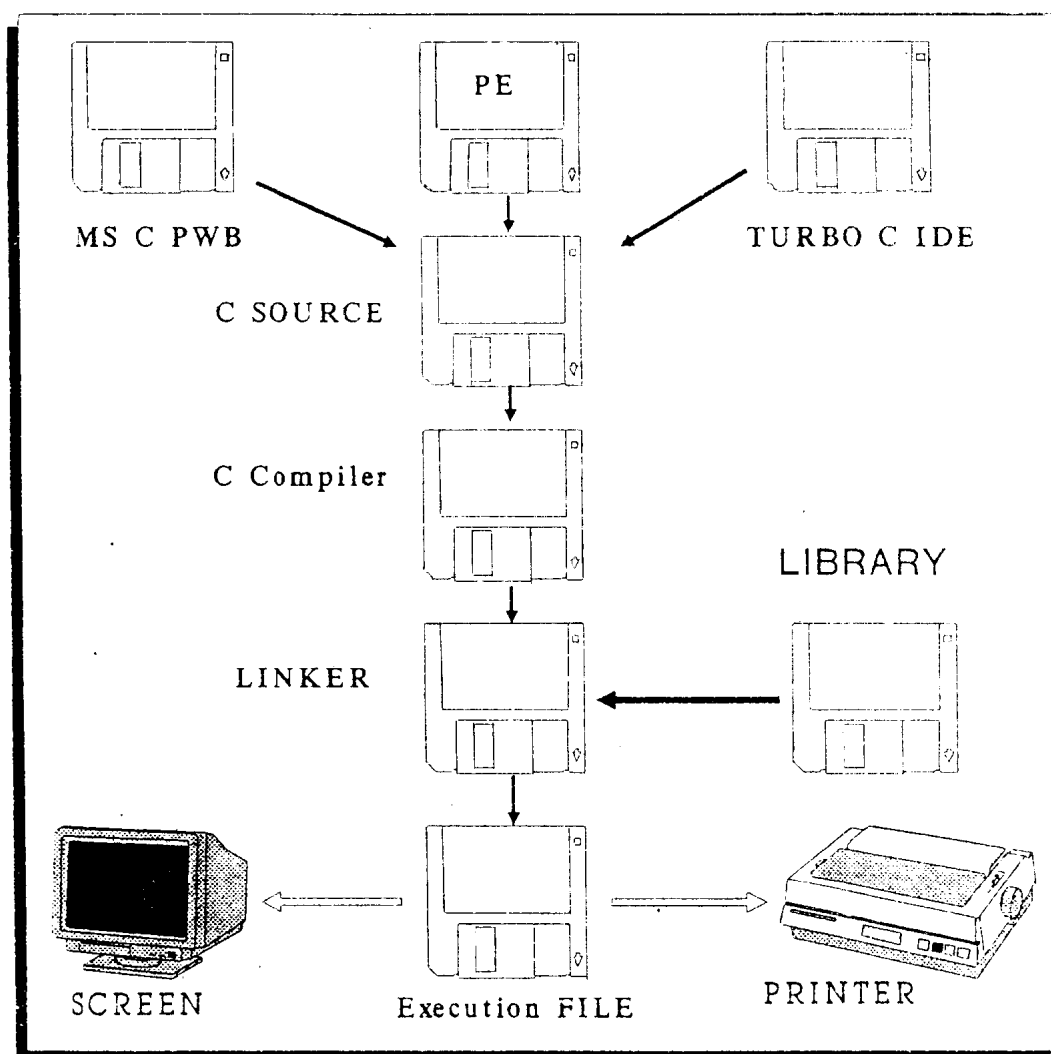


图 1.1

1.6 介绍基本元件——字符集

C 语言的程序语句是由文字、符号与数字所组成,这些都是最基本的字符。下面就介绍这些字符集合。

1. Character(字符)

在 C 语言中,用到的基本字符包括有:字母与数字、空白字符、特殊符号字符、脱序字符、运算符等。

字母与数字:字母指的是英文大小写字母,数字是指阿拉伯数字 0~9。

注意:在 C 语言中,英文字母的大小写是有区别的。

White-Space Characters(空白字符)

空白字符是指无法显示出来的字符。在 ASCII 字码表内 它的值都小于 32。如 SPACE、TAB、Line-feed、Form-feed、Carrige Return 等等。在程序内主要用途在隔离每一个单项,以提高程序的可读性。如下所示:

```
Void main (void) 0D 0A
{
    0D 0A
    printf("Hello ! \n"); 0D 0A
} 0D 0A 1A
```

其中,0D 为回车符,0A 为换行符,1A 为文件结束符。

Punctuation and Special Characters (特殊符号)

特殊字符在 C 语言中,各拥有不同的用途,我们留到程序范例再做说明。下面列出所有特殊符号字符:

表 1.1

字符	符号名称	字符	符号名称
,	逗号		垂直线
.	句号		空白符号
:	分号	/	斜线
?	问号	~	波浪符
'	单引号	_	下划线
"	双引号	#	井字号
(	左括号	%	百分号
)	右括号	&	AND 符号
[	左中括号	^	脱字号
]	右中括号	*	星号
{	左大括号	-	减号
}	右大括号	=	等号
<	左尖括号	+	加号
>	右尖括号	\	反斜线
!	感叹号		

Escape Sequence (转义符)

用来表示空白及不可打印的字符,一般是用在终端或打印机的字符串输出。转义符,是一个反斜线字符加上一个字母或数字所组成。下表列出 C 语言中的转义符。

表 1.2

转义符	用途
\n	换行
\t	水平跳格
\v	垂直跳格
\b	倒退
\r	Return
\f	跳页
\'	单引号
\"	双引号
\\	反斜线
\ddd	8 进位值 ASCII 码表示方式
\xdd	16 进位值 ASCII 码表示方式

Operators(运算符)

运算符主要用在运算语句内,用于数值运算。下表列出 C 语言的运算符。

表 1.3 算术、关系逻辑算符

运算符	名称	运算符	名称
!	逻辑否定	<	小于
~	位补数	>	大于
++	递加	==	等于
--	递减	<=	小于等于
+	加	>=	大于等于
-	减	!=	不等于
*	乘		位 OR
/	除	&	位 AND
%	余数	^	位 EXOR
<<	位左移	&&	逻辑 AND
>>	位右移		逻辑 OR

表 1.4 赋值运算符

运算符	名称	运算符	名称
=	赋值	>>=	位右移并赋值
+=	加并赋值	<<=	位左移并赋值
-=	减并赋值	&=	位 AND 并赋值
*=	乘并赋值	^=	位 EXOR 并赋值
/=	除并赋值	=	位 OR 并赋值
%=	取余数并赋值		

2. Constant(常数)

常数种类分为整数常数、浮点常数、字符常数、字符串常数。常数的表示值永远是固定的,不可以更改它。

整数常数

整数常数有三种表示方式:十进制、八进制、十六进制。分别如下所示:

十进制: 123 使用 0~9 数字。

八进制: 0123 与十进制差别:

1. 使用 0~7 数字。
2. 数值最前面为数字零——“0”

十六进制: 0X123 或 与十进制差别:

1. 使用 0~9 与 A、B、C、D、E、F(或 a、b、c、d、e、f);
2. 数值最前面为字符“0X”或“0x”。

整数常数的负值表示方式是在前面加一个负号。整数的范围,视其数据类型而定,下一章将讨论。

整数常数,可强迫设定其数据类型为长整数型(long int)。如下:

十进制	八进制	十六进制
或 12L	014L	0XEL

121 0141 0xel

浮点常数

浮点常数,仅用十进制来表示。如:

3.75
 .025
 -1.63
 4.3E-2 (4.3×10^{-2})
 -0.0089e2 (-0.0089×10^{-2})
 e 或 E 后面数值表示指数值。

字符常数

字符常数是使用一对单引号,将一个文字或数字括起来。如:

'C' 大写字母 C
 '\b' 倒退
 '\x16' ASCII ESC
 '\\ ' 倒斜线
 '!' 感叹号

字符串常数

字符串常数是使用一对双引号,将两个以上的字符括起来。如:

"This is a sample."
 "3211900"
 "Hello,world\n"

3. Identifier (标识符)

标识符是一个用来表示某种数据结构体的名称。如变量名称、函数名称。程序设计者必须去声明它或定义它,才能在程序内使用它。

标识符是一个或多个字母、数字及下划线所组成,但是第一个字符必须为字母或下划线。如:

a
 v
 templ
 Value
 _cursor_position

请注意:templ 与 TEMP1 完全不同。

4. Keyword (保留字)

保留字是 C Compiler 对这些字串有特殊的用途而设定。程序设计者不可再去声明或定