

附录 A C 语言运算符表

运算符按照优先级大小由上向下排列，在同一行的运算符具有相同优先级。第二行是所有的一元运算符。

运算符	解释	结合方式
() [] -> .	括号（函数等），数组，两种结构成员访问	由左向右
! ~ ++ -- + - * & (类型) sizeof	否定，按位否定，增量，减量，正负号， 间接，取地址，类型转换，求大小	由右向左
* / %	乘，除，取模	由左向右
+ -	加，减	由左向右
<< >>	左移，右移	由左向右
< <= >= >	小于，小于等于，大于等于，大于	由左向右
== !=	等于，不等于	由左向右
&	按位与	由左向右
^	按位异或	由左向右
	按位或	由左向右
&&	逻辑与	由左向右
	逻辑或	由左向右
?:	条件	由右向左
= += -= *= /= &= ^= = <<= >>=	各种赋值	由右向左
,	逗号（顺序）	由左向右

关于条件运算符的结合方式，参看第 3.4.2 节的解释。

附录 B C 语言速查

本附录按照分类描述 C 语言各方面的特征，对于在本书正文里有详细介绍的东西，这里只给出简单介绍和索引。在下面介绍中，诸如 (2.3.4) 之类的写法表示讨论有关问题的章节编号。

1, 关键字

C 语言总共有 32 个关键字：

auto	break (4.3.2)	case (4.5)
char (2.3.3, 5.1)	const (5.3.5)	continue (4.3.2)
default (4.5)	do (4.3.1)	double (2.3.2)
else (3.6.1)	enum (8.4)	extern (5.3.1)
float (2.3.2, 5.1)	for (3.6.3)	goto (4.3.2)
if (3.6.1)	int (2.3.1, 5.1)	long (2.3.1, 5.1)
register (5.3.5)	return (3.3.1, 3.2.3, 5.2.4)	short (5.1)
signed (5.1)	sizeof (6.2)	static (5.3.4, 5.3.5)
struct (8.1.1)	switch (4.5)	typedef (7.6)
union (8.3)	unsigned (5.1)	void (3.3.1, 5.2.4)
volatile	while (3.6.2, 4.3.1)	

注：关键字 `auto` 用于说明自动变量，通常不用；`volatile`（易变的）表示该变量不经过赋值，其值也可能被改变（例如表示时钟的变量、表示通信端口的变量等）。

2, 基本类型 (2.3, 5.1, 7)

C 语言的基本类型包括下面列出的类型，字符类型和各种整型对应的无符号类型，还有各种指针类型，关于指针类型的介绍见第七章。

字符类型	短整数类型	整数类型	长整数类型
char	short int	int	long int
浮点数类型	双精度浮点数类型	长双精度浮点数类型	
float	double	long double	

注：(1) 字符类型和所有整数类型都有对应的无符号类型，类型名加 `unsigned`；
(2) `short int` 和 `long int` 可以分别简写为 `short` 和 `long`；
(3) `unsigned int` 类型可以简写为 `unsigned`。

3, 字面量的书写形式 (2.3, 5.1)

字面量是程序里直接写出的数据，包括各种整数、浮点数、字符和字符串。

整数（各种整型的字面值）由一串数字构成。第一个数字不是 0 的是十进制数；以 0 开头的是八进制数；以 0x 或 0X 开头的数字序列表示十六进制数。加字母 `u` 或 `U` 后缀表示无符号整数；加字母 `l` 或 `L` 后缀表示长整数。(2.3.1, 5.1)

浮点数（`float`、`double` 和 `long double` 类型的字面值）的基本部分包括一个整数部分，一个小数点和一个小数部分。此后可以有一个指数后缀，其形式是字母 `e` 或 `E` 后跟一

个可以带符号的整数，表示十进制的指数值；最后还可以有一个字母 f, F, l 或 L 作为类型后缀。f 或 F 表示是 **float** 类型的字面量，l 或 L 表示是 **long double** 类型，无后缀是 **double** 类型。整数或小数部分两者可以缺一个，小数点和指数部分两者可以缺一个。(2.3.2, 5.1)

字符是单引号括起来的一个字符，特殊字符的写法见下面的特殊字符表。(2.3.3)

字符串字面量是由一对双引号界定的一个字符序列，其中可以写各种能直接写出来的字符，以及按下表列出的换意序列方式描述的特殊字符。

以换意序列描述的特殊字符包括：

换行符：	\n	水平制表符：	\t	垂直制表符：	\v
退格符：	\b	回车符：	\r	换页符：	\f
鸣铃符：	\a	反斜线符：	\\	问号：	\?
单引号：	\'	双引号：	\"		

任意字符均可以用换意符号（反斜线符）后跟八进制或十六进制表示，形式为：

八进制表示：	\ooo	十六进制表示：	\xhh
--------	------	---------	------

其中 ooo 表示三位八进制数字，最左一位只能是 0 到 3；hh 表示两位十六进制数字，0 到 9 以及 A 到 F 或 a 到 f。

4，运算符与表达式

所有运算符的优先级及结合方式见附录 A。下面给出各运算符的简单意义，并列出有关它们的讨论所在的章节。

运算符	意义	章节	运算符	意义	章节
()	括号，函数		[]	数组成分	6.1.2
->	由指针取结构成分	8.1.3	.	结构成分	8.1.2
!	逻辑非	3.4.3	~	字位取反	5.5
++	增量	3.7.1	--	减量	3.7.1
+	（一元）正号	2.4.1	-	（一元）负号	2.4.1
&	（一元）取地址	7.1.1	(类型)	类型强制	2.4.4
sizeof	数据或类型的大小	6.2	*	算术乘	2.4.1
/	算术除	2.4.1	%	求余数（取模）	2.4.1
+	（二元）算术加	2.4.1	-	（二元）算术减	2.4.1
<<	字位左移	5.5	>>	字位右移	5.5
<	小于	3.4.1	<=	小于等于	3.4.1
>	大于	3.4.1	>=	大于等于	3.4.1
==	等于	3.4.1	!=	不等于	3.4.1
&	（二元）字位与	5.5	^	字位异或	5.5
	字位或	5.5	&&	逻辑与	3.4.3
	逻辑或	3.4.3	?:	条件运算符	3.4.2
,	逗号	3.7.2	=	赋值	3.2.2
+= -= *= /= %= &= ^= = <<= >>=			各种更新赋值		3.7.3

常量表达式：

常量表达式用在 **case** 后面，描述数组和字段的长度，给枚举常量指定值，给外部变量和局部静态变量提供初始值，给数组、结构变量的成分提供初始值，以及用在条件预处理命令中。常量表达式在程序加工时被求值，这种表达式里不能包含：赋值运算符、增减量运算符、函数调用、逗号运算符。

对要求整数值的常量表达式，其中的运算对象只能是整数、枚举值、字符、**sizeof** 表达式，浮点数常量必须强制转换到某个整型。浮点数常量表达式的运算对象只能是整数、浮点数、字符、枚举值、**sizeof** 表达式。地址常量表达式应表示一个指针值，可以是外部变量、静态变量、函数的地址。这个指针值可以通过运算符**&**作用于外部或静态对象而得到的，或是由数组名、函数名直接创建的。表达式计算的结果或者就是一个已经说明了的外部或静态变量的地址，或者是一个这种地址加或减一个常数。创建过程中不应间接访问变量的值。

在预处理命令**#if** 之后的常量表达式中不允许出现 **sizeof**、枚举常量或类型强制。

5, 基本控制结构

这里列出 C 语言的各种控制结构，并给出简短的解释。

空语句只包含一个分号。它的执行什么也不做，通常用于在程序中填补位置（3.7.4）。

break;

中止语句。退出最内层的循环语句或者 **switch** 语句（4.3.2）。

continue;

继续语句。停止最内层循环体的一次执行，准备开始循环的下次执行（4.3.2）。

goto 标号;

跳转语句。使程序执行立即转到指定标号处。标号的作用范围是它所在的函数体（4.3.2）。

return;

return 表达式;

返回语句。立即结束当前函数的执行，控制返回函数调用处。第一种形式用于无返回值的函数，第二种形式用于有返回值的函数，在执行时表达式被求值，得到的值转换成函数所要求的返回值类型，作为函数的返回值（3.3.1, 3.2.3, 5.2.4）。

复合语句在形式上是一对花括号，其中可以有一系列的定义或说明，后跟一系列的语句。这两部分都可以没有（3.1）。

if (表达式) 语句₁

if (表达式) 语句₁ else 语句₂

条件语句。表达式应为算术类型或指针类型。首先对表达式求值，若得到的值非 0 就执行语句₁ 部分；若得到值 0，有 **else** 部分时执行语句₂，否则整个条件语句立即结束（3.6.1）。

while (表达式) 语句

当循环语句。表达式应为算术或指针类型。首先对表达式求值，得到值 0 时整个循环语句立即结束；得到非 0 值时执行语句部分，然后重复这整个过程（3.6.2）。

do 语句 while (表达式);

首先执行语句，执行完毕后对表达式求值。当求出的值非 0 时重复前面的过程；否则整个语句结束。这里的表达式也必须是算术类型或指针类型的（4.3.1）。

for (初值表达式; 条件表达式; 步进表达式) 语句

首先求值初值表达式。然后顺序地反复做：求值条件表达式，执行语句，求值步进表达式。一旦某次条件表达式求出值 0，整个结构的执行就立即结束。这里的条件表达式必须是算术类型或指针类型的（3.6.3）。

switch (表达式) {case 常量表达式: 语句序列 case 常量表达式: 语句序列 ...}

这里的复合语句称为 **switch** 的体，“**case 常量表达式:**”称为 **case** 标号。表达式和常量表达式（称为 **case** 常量）可以是各种字符类型或整型，一个 **switch** 语句的各个 **case** 后的常量表达式的值必须互不相同。语句的执行过程是：首先求值表达式，用这个值顺序与各 **case** 常量表达式的值比较。发现相等时，就从该标号 **case** 后的语句序列开始执行。**switch** 体中可以有一个 **default** 标号，当所有 **case** 标号都不匹配时执行转到 **default** 标号处。如果没有 **default** 标号，没有标号匹配时整个语句立即结束。人们通常在每个语句序列的最后放一个 **break** 语句，使语句序列执行完毕时执行能退出 **switch** 语句。如果被执行序列的最后没有 **break** 语句，执行将进展到随后的语句（**case** 标号的作用就像是普通标号，只表示一个位置）。语句序列可以是空（4.5）。

6，变量的定义与说明

简单变量定义及初始化问题见（3.2.1，3.2.3，5.3.5），数组定义及初始化见（6.1.1，6.1.2），指针变量的有关问题见（7.1），结构变量、联合变量见（8.1.1，8.3 等）。

各种变量有关情况请参看：自动变量（3.2.1，5.3.2，5.3.3），寄存器变量（5.3.5），外部变量（5.3.1，5.3.2，5.3.5），局部静态变量（5.3.4），外部静态变量（5.3.5），常变量（5.3.5）。

变量外部说明的形式及作用参看（5.3.1）。

7，数组、结构、联合、枚举的描述和类型说明

数据描述机制：数组（6.1.1，6.5）、结构（8.1.1）、联合（8.3）、枚举（8.4）。

类型定义的有关问题见（7.6，7.7，8.1.1），复杂类型的描述和解读见（7.8）。

8，函数定义、调用和原型说明

函数定义的基本形式是：

返回值类型 函数名 参数表 函数体

其中函数名是一个标识符；函数体是一个复合语句；参数表由一对圆括号界定，内有一系列的参数描述，每个描述说明了函数的一个形式参数的名字和它的类型。描述间用逗号分隔。无参函数的参数表可以写为 **()** 或 **(void)**（3.3，3.5，5.2）。

函数调用是一种基本表达式，它的形式是：

函数名 实际参数表

其中实际参数表由一对圆括号界定，里面是一系列的实际参数表达式，表达式间用逗号分隔。无参函数的调用形式是函数名之后跟一对空的圆括号（3.3，3.5，5.3）。

函数原型的书写形式、意义和作用请参看（5.2.6，5.2.7，5.2.8，8.6）。

9，预处理命令

预处理命令中最常用的是文件包含`#include` (5.4.1)，宏定义`#define` (5.4.2) 和一组与条件保留有关的命令`#if`、`#elif`、`#else`、`#endif` (5.4.3)。

另两个命令`#ifdef`、`#ifndef` 也用在条件保留方面：

`#ifdef` 标识符 等价于 `#if defined` 标识符

`#ifndef` 标识符 等价于 `#if !defined` 标识符

与`#define` 有关的还有两个特殊运算符`#`和`##`。

C 语言的预处理命令还有`#line`、`#error` 和`#pragma`。有的命令主要是用于编写生成 C 程序的程序。读者要了解这方面的情况，请查看参考文献[1][2]。

C 预处理部分还提供了几个特殊的预定义常量，这些常量比较特殊，它们的形式类似，前后都有两个下划线符号。有关常量列在表里：

<code>__LINE__</code>	值为一个整数，表示出现位置的行号
<code>__FILE__</code>	值为字符串，是被编译的文件名（本文件的名字）
<code>__TIME__</code>	值为字符串，以“hh:mm:ss”形式表示的当前时间
<code>__DATE__</code>	值为字符串，以“Mmm dd yyyy”形式表示的当前日期
<code>__STDC__</code>	对于标准的 C 语言实现，值为 1

10，标准函数库

有关库函数及其相关问题请参看有关章节的介绍：

标准流输入输出	(9.3)
文件输入输出	(9.2)
输入输出的格式描述	(9.3.2, 9.3.3)
直接输入输出	(9.2.5)
文件定位	(9.2.3)
输入输出出错处理	(9.2.4)
各种数学函数	(2.5.2, 9.5.1)
字符类型判断	(6.4.4)
字母大小写转换	(6.4.4)
字符串操作	(6.4.4, 9.5.1)
存储区操作	(9.5.1)
串数据转换	(9.5.2)

伪随机数生成	(5.5)
动态存储管理	(7.5.2)
执行控制	(9.5.2)
与环境交换信息	(9.5.2)
查找与排序	(9.5.2)
整数带余除操作	(9.5.2)
时间处理	(4.2.4, 9.5.3)
与实现有关的规定	(9.5.4)
条件诊断	(9.5.5)
变长参数表函数的定义	(9.5.6)
非局部控制转移	(9.5.7)
异常条件与处理	(9.5.8)

本书没有介绍的标准库函数包括：与本地化有关的类型与函数、多字节处理函数、有变长度表类型 `va_list` 参数的输入和输出函数等。需要进一步了解这些函数的读者请参看参考文献[1]或[2]。