第3章

数据类型、运算符与表达式

目　　的

(1) 掌握 C 语言的数据类型。
(2) 掌握整型、实型、字符型数据的常量及变量。
(3) 掌握 C 语言中的各种运算符。
(4) 掌握运算符的优先级与结合性。

重　　点

(1) 各种基本数据类型的常量和变量。
(2) C 语言中各种运算符的使用。

难　　点

(1) 不同类型的数据在内存中的物理存储形式。
(2) 混合表达式中运算符的运算顺序。

知识要点概述

一、数据类型

C 语言中的数据类型分为：字符类型、整类型、实类型、指针类型、构造类型和空类型，C 语言中的数据有常量和变量之分，它们分别属于以上这些类型。

常量是指在程序设计过程中已知的、在程序中直接写出的数值。变量在内存中占据一定的存储单元。C 语言的变量具有 3 个要素：变量名、数据类型和变量的值，三者不要混淆。变量名是变量的名字，用标识符来标识，变量是用来存储数据的，存储的数据就是变量

的值，数据类型决定了变量所占字节的多少。

C 语言提供了丰富的运算符，共有 44 种。

在表达式求值时，遵循的规则是：同一优先级的运算符，运算次序由结合方向决定。不同优先级的运算符出现在同一表达式中时，按运算符的优先级的高低次序执行。

在进行混合运算时，如果一个运算符两侧的运算对象的数据类型不同，系统则按“先转换、后运算”规则进行处理。

二、常量

(1) 常量：在程序运行过程中，其值不能被改变的量。

(2) 常量定义：在程序运行过程中，其值不能被改变的量称为常量。常量常区分不同的类型，如 12、0、－3 为整型常量，'a'、'D'为字符常量。实型常量如 4.5、－1.234，字符串常量如“a”、“abc”、“1”，用双引号表示。

(3) 符号常量：用一个标识符代表一个常量的，称为符号常量，即标识符形式的常量。常量不同于变量，它的值在作用域内不能改变，也不能再被赋值。

符号常量即是用一个标识符来代替一个常量；符号常借助于预处理命令 # define 来实现。

定义形式：# define　标识符　字符串

如：# define　PI　3.1415926535

说明：

① 习惯上，符号常量用大写字母表示。

② 定义符号常量时，不能以“；”结束。

③ 一个 # define 占一行，且要从第一列开始书写。

④ 一个源程序文件中可含有若干个 define 命令，不同的 define 命令中指定的“标识符”不能相同。

三、变量

(1) 变量定义：其值可以改变的量称为变量。

(2) 标识符的命名规范。和其他高级语言一样，用来标识变量名、符号常量名、函数名、数组名、类型名、文件名的有效字符序列称为标识符，C 语言中的标识符命名规范如下。

① 变量名只能由字母、数字和下划线 3 种字符组成，且第一个字符必须为字母或下划线。

② C 语言中标识符的长度(字符个数)无统一规定，随系统而不同。许多系统(如 IBM PC 的 MS C)取 8 个字符，假如程序中出现的变量名长度大于 8 个字符，则只有前面 8 个字符有效，后面的不被识别。

③ C 语言有 32 个关键字(例如 if、else、while)，它们已有专门含义，不应采用与它们同名的变量名。

④ C 语言将大写字母和小写字母认为是两个不同字母。

(3) 变量：在程序运行过程中，其值会发生变化。

① 每个变量必须有一个名字，变量名是标识符。

② 标识符是用来标识数据对象的，是一个数据对象的名字。

③ 命名规则：以字母或下划线开始，后跟字符、数字或下划线。

例：x1，_average，lotus_1_2_3，#abc，1fs，M. D. Jhon

④ 变量名不能是关键字(即保留字，是 C 编译程序中保留使用的标识符。如：auto、break、char、do、else、if、int 等)。

⑤ 变量必须先定义再使用。

四、整型数据

(1) 整型常量

整型常量即整常数。C 语言整常数可用以下 3 种表示形式。

① 十进制表示。如 231、－56.478。

② 八进制表示。以 0 开头的数是八进制数。如 0123 即$(123)_8=1\times8^2+2\times8^1+3\times8^0=64+16+3=83$。

③ 十六进制表示。以 0x 开头的数是十六进制。如 0x123 即$(123)_{16}=1\times16^2+2\times16^1+3\times16^0=256+32+3=291$。

(2) 整型变量

整型变量分为：基本型、短整型、长整型和无符号型 4 种。

① 基本型，以 int 表示。

② 短整型，以 short int 表示或以 short 表示。

③ 长整型，以 long int 表示或以 long 表示。

④ 无符号型，存储单元中全部二进制位(bit)用作存放数本身，而不包括符号。无符号型中又分为无符号整型、无符号短整型和无符号长整型，分别以 unsigned int、unsigned short 和 unsigned long 表示。

(3) 整型数据的取值范围

C 标准没有具体规定各类型所占内存字节数，各种机器处理上有所不同，以 IBM PC 为例，数的范围如表 3.1 所示。

表　3.1

类　　型	所占位数	数的范围
int	16	$-2^{15}\sim(2^{15}-1)$
short[int]	16	$-2^{15}\sim(2^{15}-1)$
long[int]	32	$-2^{31}\sim(2^{31}-1)$
unsigned[int]	16	$0\sim(2^{16}-1)$
unsigned short	16	$0\sim(2^{16}-1)$
unsigned long	32	$0\sim(2^{32}-1)$

(4) 整型常量的分类

① 一个整常量,如果其值在-32 768～32 767 范围内,认为它是 int 型,它可以赋值给 int 型和 long int 型变量。

② 一个整常量,如果其值超过了上述范围,而在-2 147 483 648～2 147 483 647 范围内,则认为它是 long int 型,可以将它赋值给一个 long int 型变量。

③ 如果某一计算机系统的 C 版本确定的 short int 与 int 型在内存中占据的长度相同,则一个 int 型的常量同时是一个 short int 型常量。

④ 常量中无 unsigned 型。但一个非负值的整常量可以赋值给 unsigned 型整变量,只要它的范围不超过变量的取值范围即可。例如:将 50 000 赋给一个 unsigned int 型变量是可以的,而将 70 000 赋给它是不行的(溢出)。

⑤ 在一个整常量后面加一个字母 l 或 L,则认为是 long int 型常量。

五、实型数据

(1) 实型常量

实数在 C 语言中又称为浮点数。实数有以下两种表示形式。

① 十进制形式。它由数字和小数点组成(注意必须有小数点)。例如:0.123、.123、123.0、0.0 都是十进制数形式。

② 指数形式。如 123.56e4 或 123.56E4 都代表 123.56×10^4。但字母 e(或 E)之前必须有数字,e 后面指数必须为整数。例如:e3、2.1e3.5、.e3、e 等都不是合法的指数形式。

(2) 实型变量

① 实型数据在内存中的存放形式。一个实型数据一般在内存中占 4 字节(32 位)。实型数据是按照指数形式存储的。

② 实型变量的分类:单精度 float、双精度 double、长双精度 long double。

六、字符型数据

(1) 字符常量

① 普通形式的字符常量:用引号(即撇号)括起来的一个字符,如'a'、'D'、'$'、'?'等都是字符常量。

② 转义符:以"\"开头的字符序列。

常用的以"\"开头的特殊字符见表 3.2。

表 3.2

字符形式	功　能	字符形式	功　能
\n	换行	\f	走纸换页
\t	横向跳格	\\	反斜杠字符
\v	竖向跳格	\'	单引号字符
\b	退格	\ddd	1～3 位八进制所代表的字符
\r	回车	\xhh	1～2 位十六进制数所代表的字符

(2) 字符变量：字符变量用来存放字符常量，只能放一个字符。例：char c1＝'a'，c2＝'A'；一个字符变量在内存中占一字节。

七、字符串常量

(1) 括在一对双引号中的 0 个或多个字符组成的序列。双引号仅作界限符。如："C language programming"、"a\\n"、"＃123"、" "等为字符串常量。

(2) 字符串常量的实际存储：在存储完字符串中的有效字符后还应存储字符串结束标志'\0'。

八、变量赋初值

在定义变量时对变量进行赋值称为变量的初始化。

格式：类型说明符　变量 1＝值 1，变量 2＝值 2……

如：int　a＝3，b＝4，c＝5；

　　float　x＝3.4，y＝0.75；

　　char　ch1＝'K'，ch2＝'P'；

用运算符和括号将运算对象(数据)连接起来的、符合 C 语法规则的句子称为表达式。

优先级是指表达式中包含多个运算符时，先进行优先级高的运算符操作，然后再进行优先级低的运算符操作。

当表达式中包含的几个运算符的优先级全相同时，由运算符的结合性来决定它们的运算顺序。

(1) 从左至右；

(2) 从右至左。

九、算术运算符与算术表达式

(1) 基本的算术运算符：＋、－、×、/和％。优先级：×、/、％　高于 ＋和－。

结合性：左结合性。

(2) 算术表达式：用算术运算符和括号将运算对象(操作数)连接起来的符合 C 语法规则的式子称为算术表达式。

(3) 强制类型转换运算符：(类型名)(表达式)。

(4) 自增、自减运算符：++ 和-- 。

作用是使变量的值增一或减一。

十、赋值运算符与赋值表达式

简单的赋值运算符：＝，除逗号表达式外，优先级最低。

复合赋值运算符：(＋＝)、×＝、％＝等。

赋值表达式：＜变量＞＜赋值运算符＞＜ 表达式/值＞。

十一、逗号运算符与逗号表达式

逗号运算符：,,所有运算符中优先级最低。

逗号表达式：表达式 1,表达式 2,…,表达式 n。

求解过程：先求表达式 1,再求表达式 2,依次求下去,直到求出表达式 n,整个逗号表达式的值就是表达式 n 的值。

典型例题分析

【例 1】 在下列符号中,可以选用哪些作变量名？哪些不可以？

a3B　3aB　Ⅱ　+a　-b　*x　$　_b5_　if　next_　day　e_2　OK?　Integer　MAXNUMBER　i*j

【解】 _b5_　a3B　next_　day　e_2　MAXNUMBER 可作变量名,其他的作变量名不可以。

(1) MAXNUMBER 可作变量名。习惯上符号常量名用大写,变量名用小写以示区别,但大写字母作变量名并无错误。

(2) if、integer 属于保留字,保留字不可作变量名。

(3) Ⅱ　+a　-b　*x　$　OK?　i*j 不可作变量名,因为变量名只能由字母、数字和下划线 3 种字符组成。

(4) 3aB 不可作变量名,因为变量名的第一个字母必须为字母或下划线。

【例 2】 下面 4 个选项中,均是不合法的浮点数的选项是_____。

(A) 160.　0.12　e3　　(B) 123　2e4.2　.e5

(C) -.18 123e4　0.0　　(D) -.e3　.234　1e3

【解】 答案为 B。

(1) 160.　0.12　-.18　123e4　0.0　.234　1e3 是实数的十进制形式或指数形式。

(2) e3　2e4.2　.e5　-.e3 不是正确的指数形式。因为正确的字母 e(或 E)之前必须有数字,e 后面指数必须为整数。对于数据表示形式.e5 以及-.e3,e 前的.与-.不是有效的数字表示形式。

(3) 123 是整数形式。

【例 3】

```
main( )
{ float a;
   a = 111111.666666;
   printf(" %f",a);
  }
```

【解】 输出结果：111111.640621。

【详解】

(1) 一个实型常量不分 float 型和 double 型。一个实型常量可以赋给一个 float 型或 double 型变量。根据变量的类型截取实型常量中相应的有效位数字。

(2) 由于 float 型变量只能接收 7 位有效数字,因此在把 111111.666666 赋给 a 时,a 只接收了 111111.6,由于输出函数 printf 中的%f 格式表示输出小数点后的 6 位小数,所以 111111.6 后的 40621 属于无意义数字。

(3) 如果 a 改为 double 型,则能全部接收上述 12 位数字。

【例 4】 若有说明语句: char c='\729';则变量 c _____。

(A) 包含 1 个字符　　(B) 包含 2 个字符

(C) 包含 3 个字符　　(D) 说明不合法

【解】 答案为 D。

【详解】 "\"后可以有 1～3 位八进制所代表的字符,本题中"\"后的"72"属于八进制所代表的字符,而"9"则不属于八进制位所代表的字符,则'\729'中包含了两个字符常量'\72'和'9'。而字符常量是用引号(即撇号)括起来的一个字符,所以答案为 D。

【例 5】 将小写字母转换成大写字母。

```
main( )
{ char c1;
 c1 = 'a';
 c1 = c1 - 32;
 printf(" % c",c1);
}
```

【解】 输出结果为 A。

【详解】

(1) 'a'的 ASCII 码为 97,所以 c1='a';语句的功能是把 97 赋值给了 c1。

(2) c1=c1-32;语句的功能是把 97-32 的值 65 赋值给 c1。

(3) printf 函数中的%c 格式表示以字符方式输出。ASCII 码值为 65 的字符为 A,所以运行结果为 A

【例 6】 下面不正确的字符串常量是_____。

(A) 'abc'　　(B) "12'12"　　(C) "0"　　(D)"　"

【解】 答案为 A

【详解】 'abc'是用单引号引来的,所以'abc'不是正确的字符串常量。

【例 7】 在 C 语言中,要求运算数必须是整型的运算符是_____。

(A) /　　(B) ++　　(C) ! =　　(D) %

【解】 答案为 D。

【详解】 对于%运算符来说,要求两侧均为整型数据,所以表达式 3.5%2 与 3%2.0 是错误的。

【例 8】 写出下列程序的输出结果。

```
main( )
{ printf(" % d, % d\n",5/3,5 % 3);
 printf(" % d, % d\n", - 5/ - 3, - 5 % - 3);
 printf(" % d, % d\n", - 5/3, - 5 % 3);
 printf(" % d, % d\n",5/ - 3,5 % - 3);
}
```

【解】 输出结果为：

```
1,2
1,-2
-1,-2
-1,2
```

【详解】 两个同号整数相除时结果为正整数，如 5/3、－5/－3 的结果值为 1。两个异号整数相除时结果为负整数，多数机器采取“向零取整”法，即－5/3＝－1，5/－3＝－1，但如果参加运算的两个数中有一个数为实数时结果为实数。对于求余(%)运算，运算结果与第一个数的符号相同。

优先级别：先×、/、%后＋、－

运算量：双元运算量，%前后必须为整数。

左右结合性：自左至右参与运算。

【例 9】 若 x 和 n 均是 int 型变量，且 x 和 n 的初值均为 5，则计算表达式后 x 的值为_____，n 的值为_____。

```
x+=n++
```

【解】 答案为 10　6。

【详解】 根据优先级别运算表达式 n++ ，因为 n++ 是后缀表示形式，所以 n 先参与运算，再运算表达式 x＋＝n，则 x 为 10，最后 n 自加为 6。

【例 10】 写出以下程序的输出结果。

```
main(  )
{ int x,y,m,n;
 x=2;y=2
m=x++ *5;
n=++y*5;
printf("%d,%d,%d,%d",x,y,m,n);
}
```

【解】 输出结果为 3,3,10,15。

【程序详解】 对于后缀来说是先使用后运算，所以 m 的值为 x 在自加以前的 2×5 得 10 赋值给 m 后，x 自加变为 3。对于前缀来说是先运算后使用，所以 m 的值为 x 在自加以后的 3×5 得 15 赋值给 n。

【例 11】 已知 x＝43，ch＝'A'，y＝0;，则表达(x＞＝y&&ch＜'B'&&！y)的值是_____。

(A) 0　　(B) 语法错　　(C) 1　　(D)“真”

【解】 答案为 C。

【详解】 C 语言不提供逻辑性数据“真”和“假”，在进行逻辑运算时则把“非零”作为“真”，把 0 作为“假”，所以表达式！y 的运算结果是 1。

【例 12】 写出下面程序的输出结果。

```
main( )
{ int x,y,z;
```

```
 x = y = z = 0;
  ++ x&&++ y||++ z;
 printf(" % d, % d, % d",x,y,z);
 x = y = z = 0;
  ++ x||++ y&&++ z;
 printf(" % d, % d, % d",x,y,z);
}
```

【解】 输出结果为

```
1,1,0
1,0,0
```

【程序说明】

(1) 因为 && 的优先级别高于||,所以表达式++x&&++y||++z 是一个或表达式,根据||的一端为 0,则不必再计算另一端的原则,先计算表达式++x&&++y 的值为 1,因为 1 或任何值都为 1,所以表达式++z 没有运算,输出结果为: 1,1,0。

(2) 表达式++x||++y&&++z 也是一个或表达式,同样根据||的一端为 0,则不必再计算另一端的原则,先计算表达式++z 的值为 1,因为 1 或任何值都为 1,所以表达式++y&&++z 没有运算,输出结果为: 1,0,0。

【例 13】 若有以下定义,则能使值为 3 的表达式是_____。

```
int k = 7,x = 12;
```

(A) x%=(k%5)　　(B) x%=(k-k%5)

(C) x%=k-k%5　　(D) (x%=k)-(k%=5)

【解】 答案为 D。

【详解】 表达式(x%=k)-(k%=5)完全等价于(x=x%k)-(k=k%5)等价于 5-2,此表达式的结果为 3。

【例 14】 若 a 是 int 型变量,且 a 的初值为 6,则计算表达式后 a 的值为_____。

```
a += a - = a * a
```

【解】 答案为-60。

【详解】 表达式从左向右运算,先计算表达式 a=a-36 后 a 为-30,再计算表达式 a=a+a 后 a 的值变为-60。

【例 15】 若有条件表达式(exp)? a++ :b--,则以下表达式中能完全等价于表达式(exp)的是_____。

【解】 答案为 exp!=0。

【说明】 对于表达式 e1? e2:e3 ,e1 一般为算术表达式、逻辑表达式、关系表达式,结果为 1(真)或 0(假)。也可以为数值 exp,结果为非 0(真)或 0(假),在本例中与 exp 完全等价的表达式是 exp!=0。

【例 16】 以下程序的运行结果是_____。

```
main()
  { int k = 4,a = 3,b = 2,c = 1;
   printf(" % d",k<a? k:c<b? c:a);
}
```

【解】 答案为1。

【程序解析】 条件表达式是从右向左运算，所以在本例中先计算表达式 c<b? c:a 的值，把各数值代入此表达式的值为1。再计算表达式 k<a? k:1 的值，因为 k<a 为假，所以整个表达式的值为1。

【例17】 以下符合C语言语法的赋值表达式是_____。

(A) d=9+e+f=d+9　　(B) d=(9+e,f=d+9)

(C) d=9+e,e++ ,d+9　　(D) d=9+e++ =d+7

【解】 答案为B。

【解析】 表达式 d=9+e+f=d+9 中 9+e+f=d+9 是不正确的表示形式，因为赋值号(=)左边不能是表达式。表达式 d=9+e,e++ ,d+9 是逗号表达式，因为赋值运算符(=)的优先级别高于逗号运算符(,)。表达式 d=9+e++ =d+7 中 9+e++ =d+7 是不正确的表达式，因为赋值号(=)左边不能是表达式。

【例18】 假设所有变量均为整型，则表达式 a=2,b=5,b++ ,a+b 的值是_____。

【解】 答案为8。

【解析】 根据逗号运算符从左向右运算的原则，首先把2和5分别赋值给了a,b。再计算表达式 b++ ,b 变为6，再计算表达式 a+b 的值，最后整个表达式的值是8。

【例19】 写出以下程序的运算结果。

```
main( )
{float x;
 int i;
 x = 3.6;
 i = (int)x
 printf("x = %f,i = %d",x,i);
}
```

【解】 运算结果为 x=3.600 000,i=3。

【程序解析】 变量 x 进行强制类型运算后，其类型仍为 float 型，值仍为3.6。

【例20】 写出以下程序的运行结果。

```
main()
{ int a = - 1;
 printf(" %d, %o",a,a);
}
```

【解】 运行结果为－1 177 777。

【程序解析】 －1在内存单元中(以补码形式存放)为$(1111111111111111)_2$，转换为八进制数为$(177\ 777)_8$。

(1) x 格式：以无符号十六进制形式输出整数。对长整型可以用“%lx”格式输出。同样也可以指定字段宽度用“%mx”格式输出。

(2) u 格式：以无符号十进制形式输出整数。对长整型可以用“%lu”格式输出。同样也可以指定字段宽度用“%mu”格式输出。

(3) c 格式：输出一个字符。

(4) s 格式：用来输出一个串。有以下几种用法。

① %s：例如：printf("%s","CHINA")输出“CHINA”字符串(不包括双引号)。

② %ms：输出的字符串占 m 列，如字符串本身长度大于 m，则突破 m 的限制，将字符串全部输出。若串长小于 m，则左补空格。

③ %－ms：如果串长小于 m，则在 m 列范围内，字符串向左靠，右补空格。

④ %m.ns：输出占 m 列，但只取字符串中左端 n 个字符。这 n 个字符输出在 m 列的右侧，左补空格。

⑤ %－m.ns：其中 m、n 含义同上，n 个字符输出在 m 列范围的左侧，右补空格。如果 n＞m，则自动取 n 值，即保证 n 个字符正常输出。

(5) f 格式：用来输出实数(包括单、双精度)，以小数形式输出。有以下几种用法。

① %f：不指定宽度，整数部分全部输出并输出 6 位小数。

② %m.nf：输出共占 m 列，其中有 n 位小数，如数值宽度小于 m 左端补空格。

③ %－m.nf：输出共占 n 列，其中有 n 位小数，如数值宽度小于 m 右端补空格。

(6) e 格式：以指数形式输出实数。可用以下形式。

① %e：数字部分(又称尾数)输出 6 位小数，指数部分占 5 位或 4 位。

② %m.ne 和%－m.ne：m、n 和“－”字符含义与前相同。此处 n 指数据的数字部分的小数位数，m 表示整个输出数据所占的宽度。

(7) g 格式：自动选 f 格式或 e 格式中较短的一种输出，且不输出无意义的零。

【例 21】 请将下面各数用八进制数和十六进制数表示。

(1) 10　(2) 32　(3) 75　(4) －617　(5) －111　(6) 2483　(7) －28 654　(8) 21 003

【解】

(1) $(10)_{10}=(12)_8=(a)_{16}$

(2) $(32)_{10}=(40)_8=(20)_{16}$

(3) $(75)_{10}=(113)_8=(4b)_{16}$

(4) $(-617)_{10}=(176\ 627)_8=(fd97)_{16}$

(5) $(-111)_{10}=(177\ 621)_8=(ff91)_{16}$

(6) $(2\ 483)_{10}=(4\ 663)_8=(963)_{16}$

(7) $(-28\ 654)_{10}=(110\ 022)_8=(9\ 012)_{16}$

(8) $(21\ 003)_{10}=(51\ 013)_8=(520b)_{16}$

【例 22】 字符常量和字符串常量有什么区别?

【解】 字符常量是一个字符，在程序中字符是用单引号括起来的。字符串变量由 0 个或若干个字符组合而成，在程序中字符串是用双引号括起来的，在存储时系统自动在字符串最后面加一个结束符号'\0'。

【例 23】 将“China”译成密码。密码规律：用原来的字母后面第 4 个字母代表原来的字母。例如，字母“A”后面第 4 个字母是“E”，用“E”代替“A”。因此，“China”应译为“Glmre” 请编一程序，用赋初值的方法使 c1、c2、c3、c4、c5 这 5 个变量的值分别为"C"、"h"、"i"、"n"、"a"。经过运算，使 c1、c2、c3、c4、c5 分别变为"G"、"l" 、"m"、"r"、"e"并输出。

【解】

```
 #include "stdio.h"
main()
```