

本章重点：C 语言的各种数据类型；C 语言的各种运算符。

本章难点：各种数据类型的存储方式；各种运算符的优先级。

计算机程序处理的数据不仅仅是简单的数字,还有需要处理的信息,包括数字、字符、声音、图像和视频。这些数据以一定的形式进行存储。数据在内存中存放的形式和可以进行的操作由数据类型决定。对数据的不同操作可以构成各种各样的表达式。

3.1 数据类型分类

数据是程序的操作对象,不同类型的数据有不同的存储方式和操作。在 C 语言中,数据类型非常丰富,如图 3.1 所示。

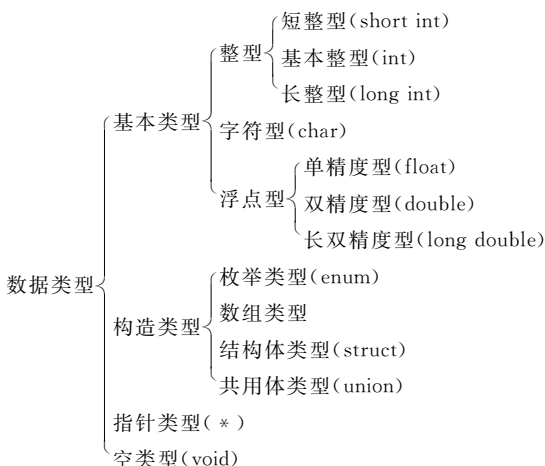


图 3.1 C 语言的数据类型

3.2 整型数据

1. 整型数据的存储方式

整型数据是没有小数部分的数值,在 C 语言中可以用三种形式表示。

(1) 十进制整数,如 123、-456、89。

(2) 八进制整数,如 0123 表示八进制整数 123,开头的 0 表示八进制数。 $(123)_8 =$

$$1\times8^2+2\times8^1+3\times8^0=(83)_{10}。$$

(3) 十六进制整数,如 0x2F 表示十六进制整数 2F,开头的 0x 表示十六进制数。
 $(2F)_{16}=2\times16^1+15\times16^0=(47)_{10}。$

数据在内存中是以二进制形式存储的,C语言对不同的数据类型分配不同长度规格的存储空间,不同长度规格的存储空间对应的数据取值范围又是不同的。即使同样长度规格的存储空间表示的数据范围还与是否有符号,是定点表示还是浮点表示有关。最后不同数据类型分配的存储长度还与编译系统有关。Visual C++ 6.0 对整型数据分配 4 个字节的存储空间。

实际上,整型数据是以补码的形式进行存储的。一个正整数的补码跟该整数的原码(即该数的二进制形式)相同。而一个负整数的补码则将该数的绝对值的二进制形式进行按位取反,末尾加 1。如果是有符号整数,则最左边的一位表示符号位,符号位用 0 表示正,用 1 表示负。

2. 整型数据的分类

整型数据可分为:基本型、短整型、长整型和无符号型。

(1) 基本型:以 int 表示。

(2) 短整型:以 short int 表示。

(3) 长整型:以 long int 表示。

(4) 无符号型: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、unsigned long 表示。无符号型分配的存储空间都用来存储数据本身,不含符号位。

各类整型数据的值的范围包括负数到正数如表 3.1 所示。

表 3.1 各类整型数据的存储空间和取值范围

类 型	字节数	取 值 范 围
基本整型(int)	2	$-32\,768\sim32\,767$,即 $-2^{15}\sim(2^{15}-1)$
	4	$-2\,147\,483\,648\sim2\,147\,483\,647$,即 $-2^{31}\sim(2^{31}-1)$
无符号基本整型(unsigned int)	2	$0\sim65\,535$,即 $0\sim(2^{16}-1)$
	4	$0\sim4\,294\,967\,295$,即 $0\sim(2^{32}-1)$
短整型(short int)	2	$-32\,768\sim32\,767$,即 $-2^{15}\sim(2^{15}-1)$
无符号短整型(unsigned short int)	2	$0\sim65\,535$,即 $0\sim(2^{16}-1)$
长整型(long int)	4	$-2\,147\,483\,648\sim2\,147\,483\,647$,即 $-2^{31}\sim(2^{31}-1)$
无符号长整型(unsigned long int)	4	$0\sim4\,294\,967\,295$,即 $0\sim(2^{32}-1)$
双长整型(long long int)	8	$-9\,223\,372\,036\,854\,775\,808\sim9\,223\,372\,036\,854\,775\,807$, 即 $-2^{63}\sim(2^{63}-1)$
无符号双长整型(unsigned long long int)	8	$0\sim18\,446\,744\,073\,709\,551\,615$,即 $0\sim(2^{64}-1)$

3.3 浮点型数据

1. 浮点型数据的表示形式

C语言中的浮点型数据实际上就是实数类型的数据,有时也称之为实型数据,其有两种表示形式。

(1) 十进制小数形式。它由数字和小数点组成,如 5.462、0.234、23.0 等都是十进制的浮点型数据。

(2) 指数形式。例如 345e3、345E3 或 345E+3 都表示 546×10^3 ,字母 e(或 E)之前必须有数字,而字母 e(或 E)之后的指数必须为一个整数。一个实型数据可以有多种指数表示形式。例如 345.67 可以表示为 345.67e0、34.567e1、3.4567e2、0.34567e3 等。其中 3.4567e2 称为“规范化的指数形式”,字母 e(或 E)之前的小数部分中,小数点左边有且只有一位非零的数字。

2. 浮点型数据的分类

浮点型数据分为单精度型(float)、双精度型(double)和长双精度型(long double),见表 3.2。

表 3.2 浮点型数据的存储空间和取值范围

类 型	字 节 数	有 效 数 字	数 值 范 围
单精度型(float)	4	6~7	约 $-3.4 \times 10^{-38} \sim 3.4 \times 10^{38}$
双精度型(double)	8	15~16	约 $-1.7 \times 10^{-308} \sim 1.7 \times 10^{308}$
长双精度型(long double)	16	18~19	约 $-1.2 \times 10^{-4932} \sim 1.2 \times 10^{4932}$

3.4 字符型数据

字符类型的数据即字符型数据,它可以分为字符和字符串两种。C 语言中的字符表示是用单引号括起来的一个字符。如'a'、'A'、'+'等。

字符型数据的存储空间和值的范围如表 3.3 所示。

表 3.3 字符型数据的存储空间和取值范围

类 型	字 节 数	取 值 范 围
有符号字符型(signed char)	1	$-128 \sim 127$,即 $-2^7 \sim (2^7 - 1)$
无符号字符型(unsigned char)	1	$0 \sim 255$,即 $0 \sim (2^8 - 1)$

用反斜杠(\)开头引导的字符称为转义字符,其意思是将反斜杠(\)后面的字符转变成另外的意义。如\n中的 n 并不是代表字母 n,而是表示换行符。常见的转义字符如表 3.4 所示。

表 3.4 常用转义字符

字 符 形 式	功 能
\n	换行
\t	横向跳格(跳到下一个输出区)
\v	竖向跳格
\b	退格
\r	回车
\f	走纸换页
\\	反斜杠字符
\'	单引号字符
\ddd	1~3 位八进制所代表的字符
\xhh	1~2 位十六进制所代表的字符

3.5 常量与变量的定义

3.5.1 常量

在C语言中,数据可以用常量和变量进行存放。常量是指在程序运行过程其值不会发生改变的量。

1. 整型常量

C语言中合法的整型常量,例如:

255,0,-8,76(十进制整型常量);

0233,0111,0107(以数字0开头表示八进制整型常量);

0xAF,0x82,0xF4(以0x开头表示十六进制整型常量);

486L,8350l(用L或l表示长整型常量)。

C语言中非法的整型常量,例如:

086(8不是八进制的数码);

4F(F是十六进制的数码,4F前面缺少0x);

0xK5(K不是十六进制的数码)。

2. 实型常量

C语言中的实型常量只能用十进制形式表示。例如58.75,589.03,1.56E+2,5.6E-2,0.12e2都是合法的实型常量,而3.5E+4.8,E5,e-8都是不合法的实型常量。

3. 字符常量

'a','x','\n'都是合法的字符常量。

4. 字符串常量

"Jiaying","ab","++"都是合法的字符串常量。

5. 符号常量

在C语言程序可以用标识符定义一个常量,称为符号常量。其定义格式为:

```
#define 标识符 常量
```

例如:

```
#define PI 3.14159
#define MAX 1000
#define MIN 10
```

在程序中,某个标识符一旦被定义为符号常量,那么在程序运行中都会将该标识符替换成对应的常量。

3.5.2 变量

1. 变量的定义

变量是以某个标识符为名字,其值可以被修改的量。标识符的定义必须符合如下的规则。

(1) 标识符是由大小写字母、数字和下划线所组成的序列,不能以数字开头。例如:a,A,a1,_a,abc都是合法的标识符,而4a,a-3则是非法的标识符。

(2) 标识符区分大小写,abc 和 ABC 是两个不同的标识符。

(3) 标识符的长度有一定的限制。C89 要求 C 语言编译器所能识别的标识符长度为不多于 31 个字符,而 C99 要求 C 语言编译器所能识别的标识符长度为不多于 63 个字符。如果超出这个长度,超出的字符可能无法识别。

(4) 普通标识符的定义不能使用 C 语言中具有特殊含义的关键字。

变量的使用必须遵循“先定义,后使用”的原则,变量的定义格式为:

类型标识符 变量名 1,变量名 2, ... ;

其中的类型标识符表示定义的变量保存的数据的数据类型,可以在一条语句定义同一数据类型多个变量,变量之间用逗号分隔。例如:

```
int i,j,k;
float x,y;
char c1,c2;
```

2. 变量的赋值

变量定义完成后可以使用赋值运算符“=”进行赋值。变量赋值可以在定义的时候赋初值或者在定义完成后赋值。例如:

```
int i = 1, j = 2, k = 3;
```

或者

```
int i,j,k;
i = 1;
j = 2;
k = 3;
```

赋值过程中一定要保证“=”右边的常量(或表达式)与“=”左边的变量类型相一致或能够进行数据转换成一致。变量赋值类型不一致将导致程序出现错误。另外,对多个变量进行赋值时,不能在定义这多个变量的同时进行连续赋值,例如:

```
int i = j = k = 1; /* 非法赋值 */
```

但是,先定义多个变量,再连续赋值是允许的,例如:

```
int i,j,k;
i = j = k = 1; /* 正确 */
```

【例 3.1】 整型变量和字符变量的定义和赋值。

编写程序:

```
#include <stdio.h>
int main()
{ int x = 10, y;
  char c1 = 'a', c2;
  y = x + 10;
  c2 = c1 - 32;
  printf(" %d, %c", y, c2);
  return 0;
}
```

运行结果：

20, A

3.6 运算符和表达式

3.6.1 C语言运算符简介

运算符是一种特定的数学或逻辑运算的符号。C语言的运算符非常丰富,主要有如下几类。

- (1) 算术运算符：+、-、*、/、%。
- (2) 关系运算符：>、<、==、>=、<=、!=。
- (3) 逻辑运算符：!、&&、||。
- (4) 位运算符：<<、>>、~、|、^、&。
- (5) 赋值运算符：=。
- (6) 条件运算符：?:。
- (7) 逗号运算符：,。
- (8) 指针运算符：*、&。
- (9) 求字节数运算符：sizeof。
- (10) 强制类型转换运算符：类型。
- (11) 分量运算符：.、->。
- (12) 下标运算符：[]。
- (13) 其他运算符。

本章只介绍赋值运算符、算术运算符、条件运算符和逗号运算符,其他运算符将在以后的章节中陆续介绍。

3.6.2 算术运算符和算术表达式

1. C语言的基本算术运算符

C语言中的算术运算通过算术运算符来实现。表 3.5 为 C 语言中的基本算术运算符及其说明。

表 3.5 C语言的基本算术运算符

运算符	名称	运算对象	功能	示例	示例值
+	加	两个实数或整数	和	4+2	6
-	减	两个实数或整数	差	4-2	2
*	乘	两个实数或整数	积	4*2	8
/	除	两个实数或整数(右操作数不能为0)	商	4/2	2
%	模	两个整数(右操作数不能为0)	相除后的余数	4%2	0

几点说明：

- (1) 算术运算符的操作数有两个,又称为双目算术运算符;
- (2) 在 C 语言中,两个整数相除的结果为整数;
- (3) +、-、*、/运算中,如果两个操作数中一个操作数为 float 类型,另一个操作数为 double 类型,则都按 double 类型进行运算。

2. 算术运算表达式及算术运算符的优先级

通过算术运算符把各个运算对象(操作数)连接起来的式子称为算术表达式。这里的运算对象可以是常量、变量和函数等。例如:

```
5 + (b/a - 2.8) + 'A'
```

表达式往往涉及多种运算,有多个运算符,所以就存在优先级的问题,即先做哪种运算,后做哪种运算。C 语言规定了算术运算符的优先级别为: *、/、% 要高于 +、-,并且它们都比赋值运算符的优先级要高。当一个表达式既有赋值运算符又有算术运算符的时候,按照 C 语言规定的优先级,应该先做算术运算,然后再做赋值运算。

3. 自增、自减运算符

自增、自减运算符的作用是使变量的值加 1 或减 1。例如: ++i, i++。--i, i--。

++i, i++ 的作用相当于 i=i+1; --i, i-- 的作用相当于 i=i-1。现在的问题是 ++i 和 i++ 有何区别? --i 和 i-- 又有何区别?

假设 i=3,则 j=++i 与 j=i++ 有何区别?

执行 j=++i 后, i=4, j=4; 而执行 j=i++ 后, i=4, j=3。由此可见, j=++i 是在使用 i 进行赋值前使 i 加 1,而 j=i++ 是在使用 i 进行赋值后再使 i 加 1。虽然最后 i 的值都加了 1,但 j 的结果却不一样。相类似 --i 和 i-- 也是同样的情况。

另外,自增运算符(++)和自减运算符(--)只能用于变量,不能用于常量或表达式,并且只能用于整型常量。例如, 6++ 或 x-- 都是错误的。

3.6.3 条件运算符和条件运算表达式

条件运算符有三个操作对象,称为三目运算符。它由两个符号“?”和“:”组成,由条件运算符连接起来的式子成为条件运算表达式。其一般形式为:

```
<表达式 1>?<表达式 2>:<表达式 3>
```

条件运算表达式的求解过程是:先求解表达式 1 的值,如果它的值为真(非 0 值),则求解表达式 2 的值并把它作为整个表达式的值;如果表达式 1 的值为假(0 值),则求解表达式 3 的值并把它作为整个表达式的值。例如:

```
b = a>100?200:300
```

其中, a>100? 200:300 为条件运算表达式,其求解过程为:

- (1) 如果 a>100 成立,则条件运算表达式的值为 200,即 b=200;
- (2) 如果 a>100 不成立,则条件运算表达式的值为 300,即 b=300。

C 语言中条件运算符的优先级要高于赋值运算符。

3.6.4 逗号运算符和逗号表达式

逗号运算符是 C 语言提供了一种特殊运算符,用逗号运算符把若干个表达式连接起来的式称为逗号表达式。逗号表达式的一般形式为:

表达式 1,表达式 2

例如:

`a = (x = 5, x + 1)`

逗号表达式的求解过程是:先求解表达式 1,再求解表达式 2。整个表达式的值是表达式 2 的值。`a=(x=5,x+1)`的求解过程为:先执行 `x=5`,然后执行 `x+1=6`,最后再执行赋值 `a=6`。

逗号表达式的一般形式还可以扩展为:

表达式 1,表达式 2,表达式 3, ..., 表达式 n

表达式 n 的值为这个逗号表达式最后的值。

3.7 本章小结

本章介绍了数据类型、运算符和表达式的有关知识,具体包括如下几方面。

1. 在 C 语言中数据都属于某种类型,不同数据类型的数据的存储和操作是不一样的。整型数据用二进制的补码形式存储,字符型数据用其 ASCII 码的值存储,实数则用指数形式存储。

2. 在程序中要存储数据时,首先要先定义相应的内存变量,内存变量的命名要按照相应的规则,并且在其名字确定后要指定其存储的数据类型。

3. 数据有常量和变量之分,符号常量是一种特殊的常量,它不占用存储空间,也不能指定相应的数据类型,它只是一个字符串,用来代替一个已知的常量。

4. ANSI C 标准并没有具体规定各种数据类型的存储空间,而是由各个 C 编译系统自主决定。Visual C++ 编译系统中,short: 2 字节、int: 4 字节、long: 4 字节、字符型: 1 字节、float: 4 字节、double: 8 字节。一般也可以用运算符 `sizeof(类型名)` 或 `sizeof(变量名)` 测试所分配的存储空间。

5. 在 C 语言中,字符和字符串是两个不同的概念,字符用加单引号,字符串要加双引号。

6. 自增(++)和自减(--)是 C 语言的一个特色,它使用起来非常方便,也可以使程序清晰、简练,但如果用得不好也会产生副作用。

习 题 3

一、填空题

1. 以下程序的输出结果是_____。

```
int main()
```



```

{
    int a = 0;
    a += (a = 8);
    printf(" %d\n", a);
    return 0;
}

```

2. 以下程序的输出结果是_____。

```

int main()
{
    unsigned short a = 65536;
    int b;
    printf(" %d\n", b = a);
    return 0;
}

```

3. 数字符号 0 的 ASCII 码十进制表示为 48, 数字符号 9 的 ASCII 码十进制表示为_____。

4. 设有以下变量定义, 并已赋确定的值:

```
char w; int x; float y; double z;
```

则表达式: $w * x + z - y$ 所求得的数据类型为_____。

5. 设 a、b、c 为整型数, 且 $a=2, b=3, c=4$, 则执行完以下语句: $a * = 16 + (b++) - (++ c)$; 后, a 的值是_____。

6. 若有定义: $\text{int } a=10, b=9, c=8$; 执行下列语句后, 变量 b 中的值是_____。

```
c = (a -= (b - 5)); c = (a % 11) + (b = 3);
```

二、选择题

1. 若变量 a 是 int 类型, 并执行了语句: $a = 'A' + 1.6$; , 则正确的叙述是_____。

- A. a 的值是字符 C
- B. a 的值是浮点型
- C. 不允许字符型和浮点型相加
- D. a 的值是字符 'A' 的 ASCII 值加上 1

2. 以下选项中不属于 C 语言的类型的是_____。

- A. signed short int
- B. unsigned long int
- C. unsigned int
- D. long short

3. 在 16 位 C 编译系统上, 若定义 $\text{long } a$; , 则能给 a 赋 40000 的正确语句是_____。

- A. $a = 20000 + 20000$;
- B. $a = 4000 * 10$;
- C. $a = 30000 + 10000$;
- D. $a = 4000L * 10L$;

4. 以下选项中合法的实型常数是_____。

- A. 5E2.0
- B. E-3
- C. .2E0
- D. 1.3E

5. 以下选项中合法的用户标识符是_____。

- A. long
- B. _2Test
- C. 3Dmax
- D. A.dat

6. 已知大写字母 A 的 ASCII 码值是 65, 小写字母 a 的 ASCII 码是 97, 则用八进制表示的字符常量 '\101' 是_____。

- A. 字符 A B. 字符 a C. 字符 e D. 非法的常量

7. 设 a 和 b 均为 double 型变量, 且 $a = 5.5$ 、 $b = 2.5$, 则表达式 $(\text{int})a + b/b$ 的值是_____。

- A. 6.500 000 B. 6 C. 5.500 000 D. 6.000 000

8. 若有以下程序:

```
int main()
{
    int k = 2, i = 2, m;
    m = (k += i * = k);
    printf(" %d, %d\n", m, i);
    return 0;
}
```

执行后的输出结果是_____。

- A. 8,6 B. 8,3 C. 6,4 D. 7,4

9. 与数学式子 $\frac{3x^n}{2x-1}$ 对应的 C 语言表达式是_____。

- A. $3 * x^n(2 * x - 1)$ B. $3 * x ** n(2 * x - 1)$
C. $3 * \text{pow}(x, n) * (1/(2 * x - 1))$ D. $3 * \text{pow}(n, x)/(2 * x - 1)$

10. 以下选项中, 与 $k = n++$ 完全等价的表达式是_____。

- A. $k = n, n = n + 1$ B. $n = n + 1, k = n$ C. $k = ++n$ D. $k++ = n + 1$