第 3 章　基本数据类型与表达式

本章重点：C 语言的数据类型；C 语言的运算符号；变量。

本章难点：数据类型的存储方式；运算符的优先级；变量的定义。

计算机程序处理的数据不仅仅是简单的数字，而是计算机处理的信息，包括数字、字符、声音、图像和视频。这些信息以一定的数据形式进行存储。数据在内存中存储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。

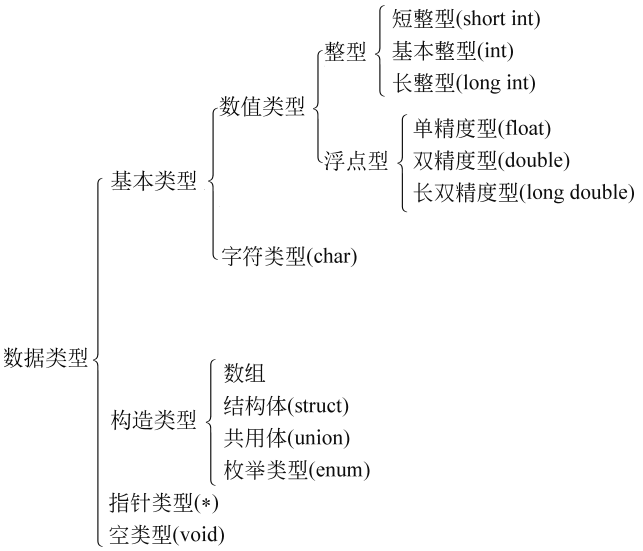

图 3.1　C 语言的数据类型

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 语言中可以用如下三种形式表示：

(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,其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。一个正整数的补码与该整数的原码(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

int是C语言的基本整数类型,此外,C语言还提供了四个可以修饰int的关键字:short、long、signed、unsigned,其中signed可以省略,所有没有标明unsigned的整数类型默认都是有符号整数。利用这四个关键字,C语言标准定义了以下整数类型。

(1) 短整型:以short int表示。

(2) 基本型:以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。

不同位数的CPU,数据的存储空间和取值范围不同。表3.1是32位CPU的各类整型数据的存储空间和取值范围。

表3.1 32位CPU的各类整型数据的存储空间和取值范围

名 称	全称类型说明符	缩写类型说明符	字节数	取值范围
短整型	short int	short	2	$-32768\sim32767$,即$-2^{15}\sim(2^{15}-1)$
无符号短整型	unsigned short int	unsigned short	2	$0\sim65535$,即$0\sim(2^{16}-1)$
基本整型	int	int	4	$-2147483648\sim2147483647$,即$-2^{31}\sim(2^{31}-1)$
无符号基本整型	unsigned int	unsigned	4	$0\sim4294967295$,即$0\sim(2^{32}-1)$
长整型	long int	long	4	$-2147483648\sim2147483647$,即$-2^{31}\sim(2^{31}-1)$
无符号长整型	unsigned long int	unsigned long	4	$0\sim4294967295$,即$0\sim(2^{32}-1)$

续表

名　　称	全称类型说明符	缩写类型说明符	字节数	取 值 范 围
双长整型	long long int	long long	8	−9223372036854775808～9223372036854775807，即-2^{63}～$(2^{63}-1)$
无符号双长整型	unsigned long long int	unsigned long long	8	0～18446744073709551615，即 0～$(2^{64}-1)$

说明：如果编译器不支持 C99 标准，则不能使用 long long 和 unsigned long long。

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，有时候也被称为实型数据，其有如下两种表示形式：

(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。如 345e3、345E3 或 345E+3 都表示 345×10^3，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。一个浮点型数据可以有多种指数表示形式。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”，即字母 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)。其存储空间和取值范围如表 3.2 所示。

表 3.2　浮点型数据的存储空间和取值范围

名　　称	类型说明符	字节数	有效数字	取 值 范 围
单精度型	float	4	6～7	$-3.4*10^{-38}$～$3.4*10^{38}$
双精度型	double	8	15～16	$-1.7*10^{-308}$～$1.7*10^{308}$
长双精度型	long double	16	18～19	$-1.2*10^{-4932}$～$1.2*10^{4932}$

说明：signed、unsigned 不能用于修饰浮点类型。浮点类型可以处理正数，也能处理负数。没有无符号浮点型。

3.4　字符型数据

字符类型的数据即字符型数据，它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取值范围如表 3.3 所示。

表 3.3　字符型数据的存储空间和取值范围

名　　称	类型说明符	字节数	取 值 范 围
有符号字符型	signed char 或 char	1	−128～127,即-2^7～(2^7-1)
无符号字符型	unsigned char	1	0～255,即 0～(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　常见的转义字符

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

在 C 语言中,数据可以用常量和变量进行存储。

3.5.1　常 量

常量是指在程序运行过程中其值不会发生改变的量。

在基本数据类型中,常量可分为整型常量、浮点型常量、字符型常量(包括字符常量和字符串常量)和符号常量,现分别介绍如下。

1. 整型常量

C 语言中合法的整型常量示例如下:

- 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示例如下:

- 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. 浮点型常量

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

3. 字符常量

'a','x','\n'都是合法的字符常量,为由单引号标识的单个字符。

4. 字符串常量

"Jiaying","ab","++"都是合法的字符串常量,为由双引号标识符的多个字符。

5. 符号常量

在 C 语言程序中可以用标识符定义一个常量,称为符号常量。其定义格式如下:

```
#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。符号常量习惯用大写表示。

3.5.2 变量

1. 变量的定义

在程序的运行过程中,需要保存很多临时数据和中间结果,这就需要用到变量。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、数字和下画线所组成的序列,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不多于 31 个字符,而 C89 要求 C 语言编译器所能识别的标识符长度不多于 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,变量的定义格式如下:

```
类型标识符 变量名 1,变量名 2,…;
```

其中,类型标识符表示定义的变量保存的数据的数据类型,可以在一条语句中定义同一数据类型的多个变量。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时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;
```

赋值过程中一般要保证“=”右边的常量跟“=”左边的变量类型相一致。变量赋值类型不一致时程序将会出现错误。另外,对变量进行赋值时不能连续赋值。例如：

```
int i=j=k=1;  /*非法赋值*/
```

另外,在C语言的赋值中有一种特殊的赋值运算符,就是复合赋值运算符,由赋值运算符“=”之前加上其他二目运算符构成。例如“n+=8;”这个语句相当于“n=n+8;”。

C语言中有如下的复合赋值运算符：

- n+=a;相当于n=n+a;
- n-=a;相当于n=n-a;
- n*=a;相当于n=n*a;
- n/=a;相当于n=n/a;
- n%=a;相当于n=n%a;
- n<<=a;相当于n=n<<a;
- n>>=a;相当于n=n>>a;
- n&=a;相当于n=n&a;
- n^=a;相当于n=n^a;
- n|=a;相当于n=n|a;

【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

运算符是一种实现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。

本章主要介绍算术运算符、关系运算符、逻辑运算符、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

算术运算是我们最熟悉的运算，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 语言中的基本算术运算符

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时，按照C语言规定的优先级，应该先进行算术运算，然后再进行赋值运算。

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，是单目运算符。例如：

```
++i,i++(前置,后置)
--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？

例如，假设i=3，则：

```
j=++i;
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。--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 关系运算符和关系表达式

1. 关系运算符

关系运算是C语言中一种比较简单的运算，也可以认为是一种比较运算。将两个数进行比较，判断这两个数是否满足给定的关系。例如：a>b，“>”是一种关系，表示“大于”。如果a赋值为4，b赋值为3，那么a>b成立，结果为“真”；如果a赋值为3，b赋值为4，那么a>b不成立，结果为“假”。

在C语言中有6种关系运算符，分别为小于“<”、小于或等于“<=”、大于“>”、大于或等于“>=”、等于“==”、不等于“!=”。

C 语言中的关系运算符都是双目运算符号(需要两个操作数),操作数可以是数值型数据和字符型数据。如果关系成立,则关系运算的值为 1(表示逻辑真);如果关系不成立,则关系运算的值为 0(表示逻辑假)。例如:

4＞3　关系运算的值为 1

4＜3　关系运算的值为 0

6 种关系运算符中,＜、＜＝、＞、＞＝的优先级相同,＝＝、!＝的优先级也相同,但＜、＜＝、＞、＞＝的优先级要高于＝＝、!＝。另外,关系运算符的优先级要低于算术运算符,高于赋值运算符。具体关系如图 3.2 所示。

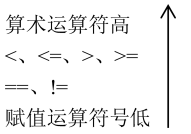

图 3.2　关系运算符的优先级

在用关系运算符“＝＝”时,如果操作数是两个浮点数,由于浮点数是用近似值表示,存在存储误差,所以有时可能会得出错误的结果。

例如,表达式 1/3.0＝＝0。

一般人都会认为这个关系运算的值是 1(为真),但实际上 1/3.0 的值约为 0.333333。因为 1/3.0 是一个用浮点数表示的近似值。另外,在 C 语言中“＝＝”是关系运算符,而“＝”是赋值运算符。因此,整个表达式的结果是 0,即为假。

2. 关系表达式

用关系运算符把两个表达式连接起来的式子称为关系表达式。

例如:

```
a<b,a+2>b+2,a+b>c+d,'A'>'B',c=5>d=3,(a>b)>(c<d)
```

关系表达式的值也是一个逻辑值,即“真”和“假”。由于在 C 语言中没有逻辑型数据,所以用 1 代表真,用 0 代表假。如果 a＝1,b＝2,c＝3,那么 a＜b 关系表达式的值为 1。

对于 a＋2＞b＋2,由于表达式中既有算术运算符,又有关系运算符,根据运算符的优先级,算术运算符的优先级要高于算术运算符,所以应该先做算术运算,再做关系运算。因此关系表示的值为 0。

3.6.4　逻辑运算符和逻辑表达式

1. 逻辑运算符

在现实生活中进行条件判断时往往需要判断不止一个条件,如果有多个条件,那么如何表示条件之间的关系呢? 例如,判断一个年份是否闰年的条件有两个:这个年份能被 4 整除,但不能被 100 整除或者这个年份能被 4 整除,又能被 400 整除。这个时候就需要用到逻辑运算符。

在 C 语言中有 3 个逻辑运算符:“!”“&&”“||”。分别表示逻辑非运算、逻辑与运算、逻辑或运算。

1) 逻辑非

逻辑非是单目运算,其运算符为“!”。运算规则:若逻辑非运算符后面的操作数的值为 0(逻辑假),则逻辑非的运算结果为 1(逻辑真);若逻辑非运算符后面的操作数的值为 1(逻辑真),则逻辑非的运算结果为 0(逻辑假)。在 C 语言中进行逻辑判断时,数据的值为非 0,则认为是逻辑真;数据的值为 0,则认为是逻辑假。例如:

```
int a=1,b=5;
!b                  //运算结果为0,因为b是非0值表示逻辑真,再进行逻辑非运算后结果为0
!(a>b)              //运算结果为1,因为a>b的运算结果是逻辑假,再进行逻辑非运算后结果为1
```

2) 逻辑与

逻辑与是双目运算,其运算符为“&&”。运算规则:若参与逻辑与运算的两个操作数的值均为1(逻辑真),则逻辑与的运算结果为1(逻辑真);若参与逻辑与运算的两个操作数的值中有一个为0(逻辑假),则逻辑与的运算结果为0(逻辑假)。例如:

```
int a=1,b=5;
a&&b                //运算结果为1,因为a和b的值均为逻辑真
(a>b)&&(a>0)        //运算结果为0,因为a>b的运算结果是逻辑假
```

3) 逻辑或

逻辑或是双目运算,其运算符为“||”。运算规则:若参与逻辑与运算的两个操作数的值均为0(逻辑假),则逻辑与的运算结果为0(逻辑假);若参与逻辑与运算的两个操作数的值中有一个为1(逻辑真),则逻辑与的运算结果为1(逻辑真)。例如:

```
int a=1,b=5;
a||b                //运算结果为1,因为a的值为逻辑真
(a>b)||(a>0)        //运算结果为1,因为a>0的运算结果是逻辑真
```

逻辑运算的运算规则也可以用表3.6表示。

表3.6 逻辑运算的真值表

a	b	!a	!b	a&&b	a||b
真	真	假	假	真	真
真	假	假	真	假	真
假	假	真	真	假	假
假	真	真	假	假	真

根据逻辑运算符的运算规律,假设用内存变量year存储年份的值,那么要判断一个年份是否是闰年的条件可以表示为

```
(year%4==0&&year%100!=0)||(year%4==0&&year%400==0)
```

2. 逻辑表达式

用逻辑运算符把两个表达式连接起来的式子称为逻辑表达式。

例如:

```
!(a<b)&&c,(a<b)||(a>c),(a<b)&&(a>c)||(b>c),!(a>b)&&b
```

逻辑表达式的值也是一个逻辑值,即“真”和“假”。C语言中逻辑运算符的优先级为逻辑非(!)→逻辑与(&&)→逻辑或(||)。如果a=1,b=2,c=3,那么!(a<b)&&c逻辑表达式的值为0,因为!(a<b)的运算结果为0;(a<b)||(a>c)逻辑表达式的值为1,因为a<b的运算结果为1。

在表达式运算中，逻辑非“!”运算的优先级要高于算术运算，而逻辑与“&&”运算和逻辑或“||”运算的优先级要低于关系运算，如图 3.3 所示。

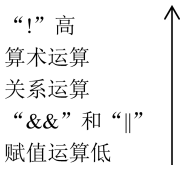

图 3.3　运算符的优先级

在逻辑表达式的运算过程中，并不是所有的逻辑运算都要被执行。

(1) a&&b&&c，只有 a 为真时，才需要判断 b 的值，只有 a 和 b 都为真时，才需要判断 c 的值。

(2) a||b||c，只要 a 为真，就不必判断 b 和 c 的值，只有 a 为假，才判断 b，只有 a 和 b 都为假，才需要判断 c。

例如：当 a=1，b=2，c=3，d=4，m 和 n 的原值为 1 时，执行表达式(m=a>b)&&(n=c>d)后，由于“a>b”的值为 0，因此 m=0，而“n=c>d”不被执行，因此 n 的值不是 0 而是仍然保持原值 1。

3.6.5　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称为条件运算表达式。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，其求解过程如下：

(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。

3.6.6　逗号运算符和逗号表达式

逗号运算符是 C 语言提供的一种特殊运算符，用逗号运算符把若干表达式连接起来的表达式称为逗号表达式。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。例如，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,字符要加单引号,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。特别是前置和后置的区别。

(7) C语言具有丰富的运算符,如算术运算符、关系运算符、逻辑运算符、条件运算符、逗号运算符等,运算符有结合方向及优先级特性。由运算符结合操作数构成了各类的表达式。

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 码十进制数表示为 48，数字符号 9 的 ASCII 码十进制数表示为_____。

4. 设有变量定义“char w;int x;float y;double z;”，并已赋确定的值，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，则执行完语句“a * =16+(b++)-(++c);”后，a 的值是_____。

6. 若有定义“int a=10，b=9，c=8;”执行语句“c=(a--(b-5)); c=(a%11)+(b=3);”后，变量 b 中的值是_____。

二、单项选择题

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“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. 以下选项中合法的浮点型常数是(　　)。

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(int)a+b/b 的值是(　　)。

A. 6.500000　　B. 6　　C. 5.500000　　D. 6.000000

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 * pow(x,n) * (1/(2 * x－1))　　D. 3 * 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