

数据类型、运算符和表达式

本章要点：

- 基本概念
- 基本数据类型
- 常量、变量
- 输入、输出函数
- 运算符与表达式

3.1 基本概念

3.1.1 C 语言的字符集

C 语言的字符集是指编译器可识别的字符，是构成 C 语言的最基本元素，由字母、数字、空格、标点和特殊字符组成。

(1) 大小写字母：a~z, A~Z。

(2) 0~9 的数字。

(3) 标点和特殊符号：

空格 ~ ! # % ^ * () _ - + =
{ } [] ; : ' " , . / ? \ |

读者一定要记住：@、\$ 和 ` 不属于 C 语言字符集中的字符，但双引号中的字符串可以包含这 3 个字符。

C 语言的常用字符与 ASCII 码对照表见附录 A。

(4) 空白符。空格符、制表符、换行符等统称为空白符。空白符只在字符常量和字符串常量中起作用。在其他地方出现时，只起间隔作用，编译程序对它们忽略不计。因此在程序中使用空白符与否，对程序的编译不发生影响，但在程序中适当的地方使用空白符可以使程序更加清晰可读。

3.1.2 标识符

标识符包括模块名、变量名、常量名、标号名、子程序名以及数据区名、缓冲区名等。这些名字应含义鲜明、能正确地提示代表的实体且符合语言的命名规范。

标识符的命名规则如下。

- (1) 由字母、数字和下划线组成。
- (2) 首字符必须是字母或者下划线,不能由数字开头。
- (3) 标识符不能与关键字重名。
- (4) 标识符区分大小写字母,例如: name_1 与 Name_1 是不同的标识符。

3.1.3 关键字

关键字是由 C 语言规定的具有特定意义的字符串,通常也称为保留字,所有的关键字一律采用小写字母。C 语言的关键字分为以下几类。

- (1) 类型说明符: 用于定义、说明变量、函数或其他数据结构的类型,例如, int、char、float 等。
- (2) 语句定义符: 用于表示一个语句的功能,如: if else 就是条件语句的语句定义符。
- (3) 预处理命令: 用于表示一个预处理命令,如前面各例中用到的 include。

C 语言共有 32 个关键字,具体内容见附录 B。

【例 3.1】 判断以下字符串哪些是合法的标识符? 哪些是 C 语言的关键字?

- ① if ② IF ③ 5m ④ _4text ⑤ ww@126 ⑥ switch ⑦ book
⑧ _cake ⑨ int ⑩ 5char ⑪ static ⑫ x * y ⑬ void ⑭ For
⑮ unsigned

依据标识符的特点,必须以字母或者下划线开头,由字母、数字和下划线组成,且不能为 C 语言的关键字,查看附录 B 可知 C 语言的关键字,最终确定以下结果。

合法的标识符有: ②④⑦⑧⑭。

C 语言的关键字有: ①⑥⑨⑪⑬⑮。

C 语言区分大小写字母,因此 IF、For 不属于 C 语言的关键字,而 if 为 C 语言关键字。

3.2 基本数据类型

计算机的出现为人类节省了大量的计算时间,慢慢地人们的需求越来越大,不仅需要计算机进行计算,更需要它能够处理各类事务。因此,需要计算机能够处理更为丰富的数据,C 语言的数据类型就是为了处理各种各样的数据而诞生的。

在 C 语言编程世界里,有哪些数据类型呢? 这些数据类型的诞生与现实世界是分不开的,分析一下现实世界中的数据主要有数值类型数据和字符类型数据两种,前者顾名思义,“数值类数据”与“数”分不开,例如日常生活中涉及的各种商品的价格、数量、需求量、销售量等。同样是商品方面的问题,除了上面提到的各种数值类型数据之外,还有商品名称、商品种类、生产厂家等数据,人们称为字符类型数据。

在 C 语言中字符类型数据也可以归到数值类型中,参与数值运算,将在后面章节中遇到。C 语言规定在程序中使用的每个数据都必须属于某种类型,它还提供了丰富的数

据类型,主要有基本类型、构造类型、指针类型和空类型四大类,具体分类如图 3.1 所示。

C 语言中常用的基本数据类型主要有整型、单精度型、双精度型、字符型。

3.2.1 整型

取值为整数的数据类型称为整型,数据类型名为 int。在 16 位系统里占 2 字节,32 位系统里占 4 字节,例如: 890、67、10000 等。

整型数据分为 int(整型)、short int(短整型)、long int(长整型)、unsigned int(无符号整型)、signed int(有符号整型),其中 short int 可简写为 short, long int 可简写为 long, unsigned int 等同于 int。长整型一般占 4 字节,整型数据的类型、数据范围及所占字节数见表 3.1。

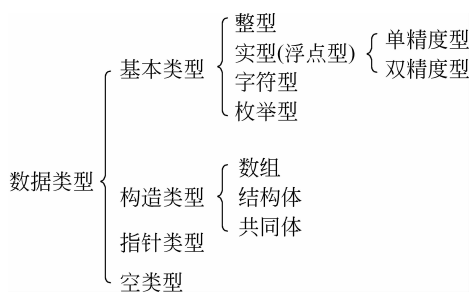


图 3.1 数据类型

表 3.1 VC++ 运行环境下整型数据

类 型	类 型 名	数 据 范 围	字 节
整型	int(基本类型符)	-2147483648~2147483647 即 $-2^{31} \sim (2^{31}-1)$	4
无符号整型	unsigned[int]	0~4294967295 即 $0 \sim (2^{32}-1)$	4
短整型	short[int]	-32768~32767 即 $-2^{15} \sim (2^{15}-1)$	2
无符号短整型	unsigned short[int]	0~65535 即 $0 \sim (2^{16}-1)$	2
长整型	long[int]	-2147483648~2147483647 即 $-2^{31} \sim (2^{31}-1)$	4
无符号长整型	unsigned long[int]	0~4294967295 即 $0 \sim (2^{32}-1)$	4

TC 运行环境下 int 类型占 2 字节,同 short int 类型。

3.2.2 实型(浮点型)

浮点型数据取值一般为实数,根据数据的范围分为单精度浮点型(float)和双精度浮点型(double)两种类型。

1. 单精度(float)

单精度实型,类型名为 float,占 4 字节,取值范围为 $-10^{38} \sim +10^{38}$,数据的有效位数为 7 位数字,例如 5.398、-143.467 等。

2. 双精度(double)

双精度实型,类型名为 double,占 8 字节,取值范围为 $-10^{308} \sim +10^{308}$,数据的有效位数为 15 位数字。双精度浮点型的范围要比单精度浮点型大,例如 1.77554545、-567922.6088 等。

3.2.3 字符型

字符型取值为一个 ASCII 字符,数据类型名为 char,占 1 字节,如 'A'、'9' 等。

构造类型是在基本数据类型基础上构造出来的数据类型。指针类型指的是内存单元地址。对于空类型(void),没有空类型数据,只有空类型指针,指针和空类型将在后面章节介绍。

【例 3.2】 现调查长春职业技术学院信息分院 2012 级 10 班学生的基本情况如下。

班级编号: 120110

学号: 12011001~12011050

姓名: 不超过 3 个汉字

团员: 80%为团员,可用“Y”代表是团员,“N”代表不是团员

入学成绩: 356.5~485

根据 C 语言基本数据类型进行数据分析,分析班级编号、学号、姓名、团员、入学成绩各项内容是否属于基本数据类型,如果属于基本数据类型,确定其数据类型。

本书所采用的运行环境为 VC++ 6.0,根据基本数据类型的定义、范围确定各项内容的数据类型如下。

“班级编号”、“学号”均可确定为整型(int)。

“姓名”的内容为汉字,不属于数值型数据,又因为一个汉字占两个字节,也不能使用字符型(char)数据,因此,“姓名”不能用基本数据类型。

“团员”可以使用“Y”和“N”,为单个字符,因此,可以定义为字符型数据。

“入学成绩”中有小数,数据变化范围较小,因此,可确定为单精度数据类型。

3.3 常量、变量

3.3.1 常量

对于基本数据类型,按其取值是否可改变又分为常量和变量两种。在程序执行过程中,其值不发生改变的称为常量,其值可变的称为变量。它们可与数据类型结合起来分类,按数据类型可分为整型常量、整型变量、浮点常量、浮点变量、字符常量、字符变量、枚举常量、枚举变量。按照常量值的表现形式可分为直接常量和符号常量。

1. 直接常量(字面常量)

整型常量: 67、0、-78。

浮点常量: 89.7、-114.34。

字符常量: 'a'、'b'。

2. 符号常量

用标识符代表一个常量。在 C 语言中,可以用一个标识符来表示一个常量,称之为符号常量。

符号常量在使用之前必须先定义,其一般形式如下:

```
#define 标识符 常量
```

其中,#define 也是一条预处理命令(预处理命令都以“#”开头),称为宏定义命令,其功能是把该标识符定义为其后的常量值。一经定义,以后在程序中所有出现该标识符的地方均代之以该常量值。

习惯上符号常量的标识符用大写字母,变量标识符用小写字母,以示区别。

【例 3.3】 符号常量的简单应用。

```
#define PI 3.14
void main()
{
    float s,r;
    r=5;
    s=PI*r*r;
    printf("s=%f",s);
}
```

说明如下。

- (1) 用标识符代表一个常量,称为符号常量。
- (2) 符号常量与变量不同,它的值在其作用域内不能改变,也不能再被赋值。
- (3) 使用符号常量的好处是:含义清楚;能做到“一改全改”。

3. 整型常量

整型常量就是整数,其表现形式有 3 种。

(1) 十进制整型常量。十进制常量在常量中只能出现 0~9 的数字,且可带正、负号,如-93、10、74 等。

(2) 八进制整型常量。八进制常量是以数字 0 开头的八进制数,其中数字为 0~7。例如:0213 表示八进制的 211,十进制值为 $2 \times 8^2 + 1 \times 8^1 + 3 \times 8^0 = 139$,0110 表示八进制的 110,十进制为 72,-012 表示八进制的-12,十进制为-10。

(3) 十六进制整型常量。十六进制常量是以 0x 或 0X 开头的十六进制数。其中每个数字可以是 0~9、a~f 或 A~F 中的数字或英文字母。例如:0X112,代表十六进制的 112,十进制的值为: $1 \times 16^2 + 1 \times 16^1 + 2 \times 16^0 = 274$,0X13 表示十六进制的 13,十进制 19,0x5a 表示十六进制的 5a,十进制 90。

一般情况下,采用十进制表示整型常量,而表示地址通常用八进制或者十六进制。

4. 浮点常量

浮点型常量也称为实型常量,即实数,其表现形式可以是小数,也可以是指数。

(1) 十进制小数形式。十进制小数形式由数字与小数点组成,如 5.83、0.23、-5.5、0.0、30. 等。十进制小数形式中的小数不可省,小数点前或者小数点后可以没有数字,但不可以前、后同时没有数字。

(2) 指数形式。指数形式的格式由两部分组成:十进制小数形式或十进制整数形式部分和指数部分。其中指数部分是在 e 或 E(相当于数学中幂底数 10)后跟整数阶码(可带符号的整数指数)。例如:3.654E5、12.69E3、0.12e8 等,其中小数点前有一位非零数字(且只有一位非零数字)称为规范化的指数形式。

5. 字符常量

字符型常量是括在半角单引号内的一个字符。这些字符通常是 ASCII 码字符,它们的值即该字符的 ASCII 码值,如 'a'、'@'、'7' 等。

6. 字符串常量

字符串常量是由半角双引号括起来的字符序列。字符串常量与字符常量有严格的不同,编译程序在每个字符串的后面自动加上一个空字符 \0 以示区别。例如:"I am a student."、"China" 等是字符串常量。

【例 3.4】 已知 12011001 班某同学的基本信息见表 3.2,根据所给数据,确定各字段的值属于什么类型常量。

表 3.2 张旭同学基本信息

学号	姓名	性别	团员	入学成绩
12011001	张旭	男	Y	458.8

根据各字段常量的特点,不难看出所有常量均为直接常量,具体类型见表 3.3。

表 3.3 常量类型确定

学号	姓名	性别	团员	入学成绩
12011001 (整型常量)	"张旭" (字符串常量)	"男" (字符串常量)	'Y' (字符常量)	458.8 (浮点常量)

将表 3.3 中字符串常量输出,其形式如下:

```
printf("张旭");
printf("男");
```

字符常量与字符串常量形式不同,'Y'为字符常量,而"Y"为字符串常量。如果变量被定义为字符变量,那么可以使用单引号形式的字符常量为其赋值,不可以使用双引号形式的字符串常量为其赋值。C 语言规定,在每一个字符串常量的结尾加一个“字符串结束标志”,以便系统根据这个标志判断字符串是否结束。C 语言规定以 \0 作为字符串结束标志,\0 是一个 ASCII 码为 0 的字符,也就是一个“空操作符”,即它不会引起任何控制动作,也不是一个可显示的字符。例如:'Y'是字符常量,占一个字节;"Y"是字符串常量,占两个字节,其中一个放 \0,但在输出时不输出 \0,\0 是系统自动加的,因此,在写字符串

时不需要写\0。

3.3.2 变量

其值可以改变的量,称为变量。一个变量应该有一个名字,在内存中占据一定的存储单元。变量必须先定义后使用。要区分变量名和变量值是两个不同的概念,如图 3.2 所示。

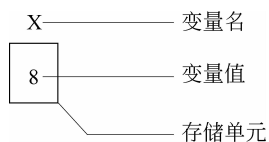


图 3.2 变量与变量名

1. 变量定义

在程序中,常量是可以不经说明而直接引用的,而变量则必须先定义后使用。

变量定义的一般形式:

类型说明符 变量名标识符,变量名标识符,...

例如:

```
int x1,x2;           /* x1,x2 为整型变量 */
float x,y;          /* x,y 为单精度浮点型变量 */
unsigned y1,y2;     /* y1,y2 为无符号整型变量 */
```

2. 变量定义说明

(1) 允许在一个类型说明符后,定义多个相同类型的变量,各变量名之间用逗号间隔。

(2) 类型说明符与变量名之间至少用一个空格间隔。

(3) 最后一个变量名之后必须以“;”号结尾。

(4) 变量定义必须放在变量使用之前,一般放在函数体的开头部分。

3. 变量赋初值

在程序中常常需要对变量赋初值,以便使用变量。语言程序中可有多种方法为变量提供初值。本小节先介绍在作变量定义的同时给变量赋以初值的方法,这种方法称为初始化。

(1) 一般形式

类型说明符 变量 1=值 1,变量 2=值 2,...

(2) 说明

① 可以给被定义变量部分赋初值。例如:

```
int x,y=3,z=8;
```

② 可以为多个变量同时赋同一初值。例如:

```
float x=1.0,y=1.0,z=1.0;
```

其中, x, y, z 变量的初值均为 1.0, 不可以写成如下形式。

```
float x=y=z=1.0;
```

【例 3.5】 定义变量并赋初值。

```
#include<stdio.h>
void main()
{
    int a=25;
    int b,c=36;
    char c1='a',c2='B';
}
```

应注意, 在定义中不允许连续赋值, 如 `int a=b=28` 是不合法的。

【例 3.6】 定义例 3.2 基本数据类型数据。

```
#include<stdio.h>
void main()
{
    int classNum, studenNum;    /* classNum 为班级编号变量, studenNum 为学号变量 */
    char league;               /* league 为团员变量 */
    float score;               /* score 为入学成绩变量 */
}
```

3.3.3 字符数据在内存中的存储形式

将字符常量存放在字符变量中, 其实并不是将该字符本身存放在内存单元中, 而是将该字符所对应的 ASCII 码存放在内存单元中。例如字符 A 的 ASCII 码为 65, 字符 B 的 ASCII 码为 66, 字符 C 的 ASCII 码为 67, 字符 a 的 ASCII 码为 97, 字符 b 的 ASCII 码为 98, 字符 c 的 ASCII 码为 99, 大小写字母之间的 ASCII 码值相差 32, 其他字符所对应的 ASCII 码参照附录 A。

因为字符在内存单元的存储形式为 ASCII 码, 因此, 它的存储形式与整数的存储形式类似, 这样字符型数据与整型数据可以相互通用。一个字符既可以以字符形式输出, 也可以以整数形式输出, 如果是以字符形式输出, 则输出字符本身, 如果以整数形式输出, 输出该字符所对应的 ASCII 码, 也可以将字符数据参与算术运算, 此时, 用该字符所对应的 ASCII 码进行算术运算。

【例 3.7】 分别以字符形式及整数形式输出两个字符及对应的 ASCII 码。

```
#include<stdio.h>
void main()
{
    char c1,c2;
    c1='a';
    c2='b';
    printf("%c,%d,%c,%d",c1,c1,c2,c2);
}
```

运行结果如下：

```
a,97,b,98
```

【例 3.8】 将小写字母转换为大写字母。

```
#include<stdio.h>
void main()
{
    char ch1,ch2;
    ch1='a';
    ch2='b';
    ch1=ch1-32;
    ch2=ch2-32;
    printf("%c,%c",ch1,ch2);
}
```

运行结果如下：

```
A,B
```

【例 3.9】 字符数据参与算术运算。

```
#include<stdio.h>
void main()
{
    char c1;
    int n;
    c1='A';
    n=c1*10;
    printf("%d",n);
}
```

运行结果如下：

```
650
```

因为字符 A 的 ASCII 码为 65,在算术运算中,取其 ASCII 码进行运算,因此运行结果为 650。

【例 3.10】 将整型数据赋给字符变量。

```
#include<stdio.h>
void main()
{
    char str1,str2;
    str1=65;
    str2=66;
    printf("%d,%c,%d,%c",str1,str1,str2,str2);
}
```

运行结果如下：

```
65, A, 66, B
```

3.4 输入/输出函数

3.4.1 格式输入/输出函数

计算机向外部设备(如显示器、打印机、磁盘等)输出数据称为“输出”，从外部设备(如键盘、鼠标、扫描仪、光盘、磁盘)向计算机输入数据称为“输入”。输入与输出是以计算机主机为主体而言的。

C语言本身不提供输入/输出语句,输入/输出操作由函数实现。在C标准函数库中提供了一些输入/输出函数,如printf()函数(格式输出)、scanf()函数(格式输入)、putchar()函数(字符输出)、getchar()函数(字符输入)等。

1. 格式输出函数 printf()

printf()函数包含在stdio.h文件中,因此在程序的开头需要将其包含进来。

1) 功能

按照用户指定的格式,向系统的输出设备(例如:屏幕)输出若干个任意类型数据。

2) 格式

```
printf("格式控制字符串",输出表列);
```

例如:

```
printf("请输入账号:\n");  
  
printf("%d,%c",i,str);
```

3) 说明

(1)“格式控制字符串”是用双引号括起来的字符串,也称“转换控制字符串”,它指定输出数据项的类型和格式。它包括格式说明、普通字符、转义字符。

① 格式说明字符串。格式说明符由“%”和格式字符串组成,它的作用是将输出的数据项转换为指定的格式输出。例如:

%d——按十进制整型输出;

%ld——按十进制长整型输出;

%c——按字符型输出;

%f——按单精度型输出;

%7.2f——按单精度型输出,输出宽度为7,保留两位小数;

%5d——按十进制整型输出,宽度为5,数据不足宽度时,左补空格,即右对齐;

%-5d——按十进制整型输出,宽度为5,数据不足宽度时,右补空格,即左对齐。