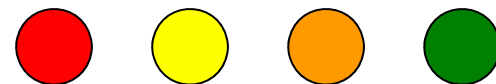




程序设计基础 (C语言)

郎大鹏



第2章 数据表示及数据运算

- 2.1 数据类型的一般概念
- 2.2 常量与变量
- 2.3 整型数据
- 2.4 浮点型数据
- 2.5 字符型数据
- 2.6 变量赋初值
- 2.7 不同数据类型数据间的混合运算
- 2.8 运算符和表达式
- 2.9 运算符的优先级和结合性



第2章 数据表示及数据运算

- 2.1 数据类型的一般概念
- 2.2 常量与变量
- 2.3 整型数据
- 2.4 浮点型数据
- 2.5 字符型数据
- 2.6 变量赋初值
- 2.7 不同数据类型数据间的混合运算
- 2.8 运算符和表达式
- 2.9 运算符的优先级和结合性



第2章 数据表示及数据运算

- 2.1 数据类型的一般概念



第2章 数据表示及数据运算

- 2.1 数据类型的一般概念
- 2.2 常量与变量
- 2.3 整型数据
- 2.4 浮点型数据
- 2.5 字符型数据
- 2.6 变量赋初值
- 2.7 不同数据类型数据间的混合运算
- 2.8 运算符和表达式
- 2.9 运算符的优先级和结合性



第2章 数据表示及数据运算

- 2.2 常量与变量

在程序执行过程中，其值不能被改变的量称为**常量**。

例：12、0、-7 为整型常量

3.14、-2.8 为浮点型常量

'a'、'b'、'c' 为字符常量

符号常量在使用之前必须先定义，其一般形式为：

#define 标识符 常量

其中**#define**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。



第2章 数据表示及数据运算

- 2.2 常量与变量

在程序执行过程中，其值不能被改变的量称为常量。

符号常量在使用之前必须先定义，其一般形式为：

#define 标识符 常量

其中**#define**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。

使用符号常量的好处如下：

(1) 含义清楚。(2) 在需要改变一个常量时能做到“一改全改”。





例2.1 符号常量的使用。程序代码如下：

```
#define PI 3.14
```

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
    float area, r;
```

```
    r = 10;
```

```
    area = PI* r * r;
```

```
    printf("area=%f\n", area);
```

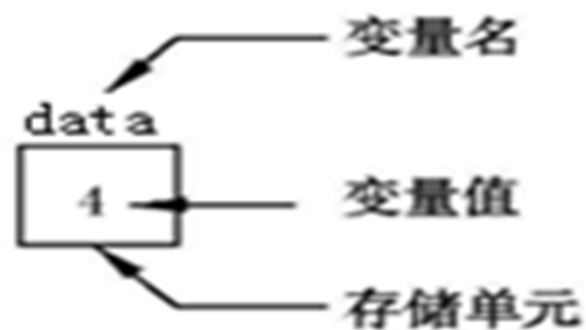
```
}
```



第2章 数据表示及数据运算

- 2.2 常量与变量

在程序执行过程中，其值可变的量称为变量。一个变量必须有一个名字，在内存中占据一定的存储单元，在该存储单元中存放变量的值。



第2章 数据表示及数据运算

- 2.2 常量与变量

变量名是一种标识符，所谓标识符就是程序中用来为符号常量、变量、函数、数组、文件等命名的有效字符序列。标识符的命名规则

如下：

- ✓ 只能由字母、数字和下划线组成。
- ✓ 第一个字符必须为字母或下划线。
- ✓ 不能使用C语言中的关键字。
- ✓ 区分大小写字母，`sum`和`Sum`是不同的标识符。



第2章 数据表示及数据运算

- 2.2 常量与变量

变量则必须作强制定义，即“先定义，后使用”。这样做的目的有以下几点：

- ✓凡未被事先定义的，不作为变量名，这就能保证程序中的变量名使用得正确。
- ✓一个变量被指定为某一确定的数据类型，在编译时就能为其分配相应的存储单元。
- ✓一个变量被指定为某一确定的数据类型，便于在编译时据此检查所进行的运算是否合法。



第2章 数据表示及数据运算

- 2.1 数据类型的一般概念
- 2.2 常量与变量
- 2.3 整型数据
- 2.4 浮点型数据
- 2.5 字符型数据
- 2.6 变量赋初值
- 2.7 不同数据类型数据间的混合运算
- 2.8 运算符和表达式
- 2.9 运算符的优先级和结合性



第2章 数据表示及数据运算

- 2.3 整型数据

1. 整型变量的分类

整型变量可分为基本型、短整型、长整型和无符号型4种。

- (1) 基本型

类型声明符为`int`，在内存中占2个字节。

- (2) 短整型

类型声明符为`short int`或`short`，所占字节和取值范围均与基本型相同。

- (3) 长整型

类型声明符为`long int`或`long`，在内存中占4个字节。

- (4) 无符号型

类型声明符为`unsigned`，存储单元中全部二进位(bit)都用作存放数本身，而不包括符号。



类型声明符	字节数	取值范围	
<u>int</u>	2	-32768~32767	即 $-2^{15} \sim (2^{15}-1)$
<u>short [int]</u>	2	-32768~32767	即 $-2^{15} \sim (2^{15}-1)$
<u>long [int]</u>	4	-2147483648~2147483647	即 $-2^{31} \sim (2^{31}-1)$
<u>unsigned [int]</u>	2	0~65535	即 $0 \sim (2^{16}-1)$
<u>unsigned short</u>	2	0~65535	即 $0 \sim (2^{16}-1)$
<u>unsigned long</u>	4	0~4294967295	即 $0 \sim (2^{32}-1)$



第2章 数据表示及数据运算

- 2.3 整型数据

整型变量的定义

C规定在程序中所有用到的变量都必须在程序中定义，即“强制类型定义”。

变量定义的一般形式为：

类型声明符 变量名标识符1, 变量名标识符2, ...;

例如：

```
int a, b, c;          /* a、b、c为整型变量*/  
long m, n;            /* m、n为长整型变量*/  
unsigned int p, q;    /* p、q为无符号整型变量*/
```



第2章 数据表示及数据运算

- 2.3 整型数据

变量定义时应注意以下几点：

- (1) 允许在一个类型声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。类型声明符与变量名之间至少用一个空格间隔。
- (2) 最后一个变量名之后必须以分号(;)结束。
- (3) 变量定义必须放在变量使用之前。
- (4) 可在定义变量的同时给出变量的初值。其格式为：

类型声明符 变量名标识符1 = 初值1, 变量名标识符2 = 初值2, ...;

例： **int a=1,b=2;**



- 例2.2 整型变量的定义与初始化。程序代码如下
:
- `#include <stdio.h>`
- `void main( )`
- `{`
- `int a = 3, b = 5;`
- `printf("a+b=%d\n", a + b);`
- `}`





变量在内存中的存储形式

数据在内存中是以二进制补码形式存放的

整型数据的溢出

一个int型变量的取值范围是-32768~32767，超过这个范围则溢出



第2章 数据表示及数据运算

- 2.3.2 整型常量

整型常量就是整常数。在C语言中，使用的整常数有八进制、十六进制和十进制3种，使用不同的前缀来相互区分。除了前缀外，C语言中还使用后缀来区分不同的整数。

1. 八进制整常数

八进制整常数必须以0开头，即以0作为八进制数的前缀。

2. 十六进制整常数

十六进制整常数的前缀为0X或0x。

3. 十进制整常数

十进制整常数没有前缀，数码取值为0~9。

4. 整型常数的后缀

长整型数是用后缀“L”或“l”来表示的。

5. 整型常数的类型



第2章 数据表示及数据运算

- 2.1 数据类型的一般概念
- 2.2 常量与变量
- 2.3 整型数据
- 2.4 浮点型数据
- 2.5 字符型数据
- 2.6 变量赋初值
- 2.7 不同数据类型数据间的混合运算
- 2.8 运算符和表达式
- 2.9 运算符的优先级和结合性



第2章 数据表示及数据运算

- 2.4 浮点型数据

1. 浮点型变量的分类

按照数值的取值范围不同分为以下三种：

- 1) 单精度型：类型声明符为`float`，在内存中占4个字节。
- 2) 双精度型：类型声明符为`double`，在内存中占8个字节。
- 3) 长双精度型：类型声明符为`long double`，在内存中占16个字节。

2. 浮点型变量的定义

浮点型变量声明的格式和书写规则与整型相同。例如：

```
float x, y;          /* x、y为单精度变量*/
```

也可在声明变量为浮点型的同时，给变量赋初值。例如：

```
double a = 0.2, b = 1.3, c = 5.1
```



第2章 数据表示及数据运算

- 2.4 浮点型数据

1. 浮点型常量的表示形式

有两种形式：十进制数形式和指数形式。

- 1) 十进制数形式

由数码0~9和小数点组成。例如0.0、.25、5.789、0.13、5.0、300.、-267.8230 等均为合法的实数。

- 2) 指数形式

一般格式是“实数+e（或E）+整数”，记为 $a E n$ ，其值为 $a \times 10^n$ 。

2. 浮点型常量的类型

许多C编译系统将实型常量作为双精度来处理



- 例2.4 演示float和double的区别。程序代码如下：
- #include <stdio.h>
- void main()
- {
- float a;
- double b;
- a = 33333.333333;
- b = 33333.3333333333;
- printf("a=%f\nb=%f\n", a, b);
- }



第2章 数据表示及数据运算

- 2.1 数据类型的一般概念
- 2.2 常量与变量
- 2.3 整型数据
- 2.4 浮点型数据
- 2.5 字符型数据
- 2.6 变量赋初值
- 2.7 不同数据类型数据间的混合运算
- 2.8 运算符和表达式
- 2.9 运算符的优先级和结合性



第2章 数据表示及数据运算

- 2.5 字符型数据

字符常量是用一对单引号括起来的单个字符，如 ‘A’、‘a’、‘X’、‘?’、‘\$’ 等都是字符常量。注意单引号是定界符，不是字符常量的一部分。

C语言还允许用一种特殊形式的字符常量，即**转义字符**。转义字符以反斜线“\”开头，后跟一个或几个字符。



第2章 数据表示及数据运算

- 2.5 字符型数据

转义字符	表示含义
\\	反斜杠字符 “\”
\'	单引号字符
\”	双引号字符
\n	换行，将当前位置移到下一行开头
\t	水平制表，横向跳到下一个输出区
\r	回车，将当前位置移到本行开头
\f	换页，将当前位置移到下页开头
\b	退格，将当前位置移到前一列
\ddd	1到3位八进制数所代表的字符
\xhh	1到2位十六进制数所代表的字符





区分一下：

65

065

0x65

n

'n'

"n"

'\n'

'\65'

'\x65'



第2章 数据表示及数据运算

- 2.5 字符型数据

字符变量用来存放字符常量，即单个字符。每个字符变量被分配一个字节的内存空间，因此只能存放一个字符。

字符变量的类型声明符为**char**，字符变量类型声明的格式如下：

```
char a, b;          /* 定义字符变量 a和b */  
a = 'x', b = 'y';   /* 给字符变量a和b分别赋值'x'和'y' */
```

字符数据在内存中以**ASCII**码存储，它的存储形式与整数的存储形式相类似，所以也**可以把它们看成是整型量**。

注意：

整型数据为2个字节

字符数据为1个字节

当整型数据按字符型量处理时，只有低8位参与处理。



- 例2.6 字符变量的使用。程序代码如下：
- `#include <stdio.h>`
- `void main()`
- `{`
- `char a, b;`
- `a = 120;`
- `b = 121;`
- `printf("%c,%c\n%d,%d\n", a, b, a, b);`
- `}`



- 例2.7 将小写字母转换成大写字母。程序代码如下：
- `#include <stdio.h>`
- `void main()`
- `{`
- `char a, b;`
- `a = 'x';`
- `b = 'y';`
- `a = a-32; /* 把小写字母转换成大写字母 */`
- `b = b-32;`
- `printf("%c,%c\n%d,%d\n", a, b, a, b); /* 以字符型和整型输出 */`
- `}`



第2章 数据表示及数据运算

- 2.5.3 字符串常量

字符串常量是由一对双引号括起来的字符序列。如：“CHINA”、“C program”、“\$12.5”等都是合法的字符串常量。

可以输出一个字符串，例如：

```
printf("Hello world!");
```

思考：

不能把一个字符串赋值给一个字符变量。那么，

‘a’ 和 “a” 究竟有什么区别呢？





C语言规定，在每一个字符串的结尾加一个字符串结束标记，以便系统据此判断字符串是否结束。C语言规定以字符'\0'作为字符串结束标记。'\0'是一个ASCII码为0的字符，也就是空操作字符，即它不引起任何控制动作，也不是一个可显示的字符。

如字符串“WORLD”在内存中的实际存放形式为：



第2章 数据表示及数据运算

- 2.1 数据类型的一般概念
- 2.2 常量与变量
- 2.3 整型数据
- 2.4 浮点型数据
- 2.5 字符型数据
- 2.6 变量赋初值
- 2.7 不同数据类型数据间的混合运算
- 2.8 运算符和表达式
- 2.9 运算符的优先级和结合性



第2章 数据表示及数据运算

- 2.6 变量赋初值

在C语言中允许在定义变量的同时初始化，如：

```
int a=3;
```

```
float f=3.21;
```

如果对几个变量都赋予相同的初值（如1），应写成：

```
int x=1,y=1,z=1;
```

注意：不能写成`int x=y=z=1`；但可以写成：

```
int x, y, z;
```

```
x=y=z=1;
```





变量初始化是在程序运行时赋予初值的，相当于一个赋值语句
如：

`int a=1;`

相当于： `int a;`
 `a=1;`

`int a, b, c=5;`

相当于： `int a, b, c;`
 `c=5;`



第2章 数据表示及数据运算

- 2.1 数据类型的一般概念
- 2.2 常量与变量
- 2.3 整型数据
- 2.4 浮点型数据
- 2.5 字符型数据
- 2.6 变量赋初值
- 2.7 不同数据类型数据间的混合运算
- 2.8 运算符和表达式
- 2.9 运算符的优先级和结合性



第2章 数据表示及数据运算

- 2.7 不同数据类型数据间的混合运算

整型、实型(包括单精度和双精度)、字符型数据间可以混合运算。

`10+'a'+1.5-12.34*'b'`

转换的方法有两种，一是自动转换，二是强制转换。

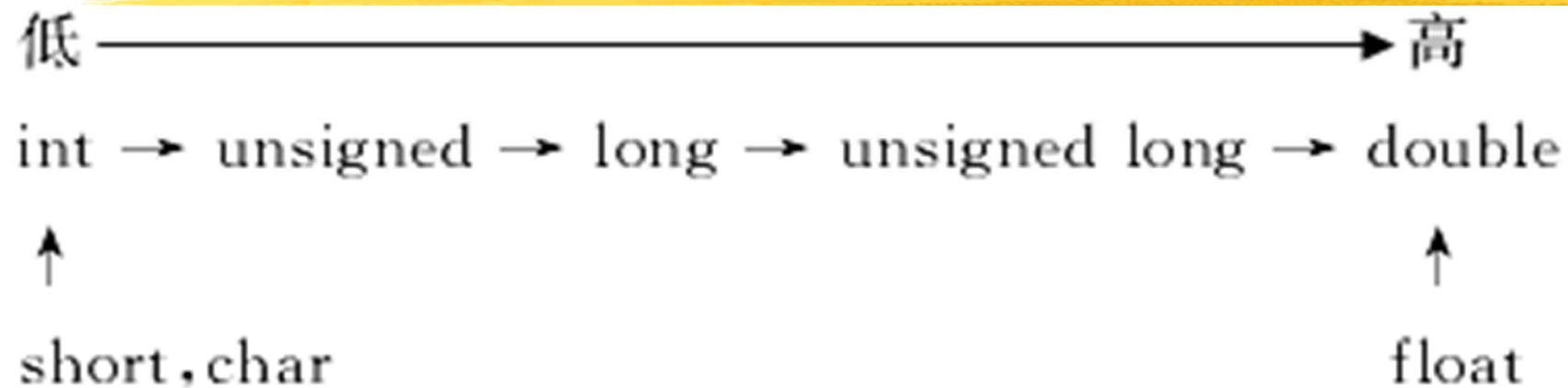


第2章 数据表示及数据运算

自动类型转换

- 若参与运算的类型不同，则先转换成同一类型，然后进行运算。
- 转换按数据长度增加的方向进行，以保证精度不降低。如`int`型和`long`型混合运算时，先把`int`型转成`long`型后再进行运算。
- 所有的浮点运算都是以双精度进行的，即使仅含`float`单精度量运算的表达式也要先转换成`double`型，再作运算。
- `char`型和`short`型参与运算时，必须先转换成`int`型。
- 在赋值运算中，赋值号两边量的数据类型不同时，赋值号右边量的类型将转换为左边量的类型。如果右边量的数据类型长度比左边长时，将丢失一部分数据，这样会降低精度，丢失的部分按四舍五入向前舍入。





图中**纵向**向上的箭头表示必定发生的转换，如字符型数据必先转成整型，单精度数据先转成双精度数据等。

图中**横向**的箭头表示当运算对象为不同的类型时转换的方向。





假设i为整型变量，f为单精度实型变量，d为双精度实型变量，e为长整型变量。则表达式 $10+'a'+i*f-d/e$ 的运算次序如下：

- (1) 进行 $10+'a'$ 的运算，先将 'a'转换成整数97，运算结果为107。
- (2) 进行 $i*f$ 的运算，先将 'i'和 'f'都转换成双精度型，运算结果为双精度型。
- (3) 整数107和 $i*f$ 的积相加，先将整数107转换成双精度型(107.000000)，运算结果为双精度型。
- (4) 进行 d/e 的运算，先将e转换成双精度型，运算结果为双精度型。
- (5) 将 $10+'a'+i*f$ 的结果与 d/e 的商相减，结果为双精度型。

上述的类型转换是由系统自动进行的。



强制类型转换

(类型声明符) (表达式)

其功能是把表达式的运算结果强制转换成类型声明符所表示的类型。

例如：(float)a把a转换为实型，(int)(x+y)把x+y的结果转换为整型。

? (int)(2.5+3.6)

? (int)2.5+3.6



- 试一试:

$$8/4*(\text{int})2.5/(\text{int})(1.25*(3.7+2.3))$$

$x=2.5$, $a=7$, $y=4.7$, 则

$$x+a\%3*(\text{int})(x+y)\%2/4 = ?$$



第2章 数据表示及数据运算

- 2.1 数据类型的一般概念
- 2.2 常量与变量
- 2.3 整型数据
- 2.4 浮点型数据
- 2.5 字符型数据
- 2.6 变量赋初值
- 2.7 不同数据类型数据间的混合运算
- 2.8 运算符和表达式
- 2.9 运算符的优先级和结合性



第2章 数据表示及数据运算

- 2.8 运算符和表达式

- 1. 算术运算符

运算符	名称	举例	功能
-	取负值	$-x$	取 x 的负值
+	加	$x+y$	求 x 与 y 的和
-	减	$x-y$	求 x 与 y 的差
*	乘	$x*y$	求 x 与 y 的积
/	除	x/y	求 x 与 y 的商
%	求余(或模)	$x\%y$	求 x 除以 y 的余数



第2章 数据表示及数据运算

• 例:

运算	结果
----	----

$2+3$	5
-------	---

$2-3$	-1
-------	----

$2*3$	6
-------	---

$2/3$	0
-------	---

$2\%3$	2
--------	---

$2.0/3$	0.666666
---------	----------

$2.0\%3$	非法
----------	----



第2章 数据表示及数据运算

- (1) 减法运算符“-”可作取负值运算符，这时为单目运算符。例如 $-(x+y)$ 、 -10 等。
- (2) 使用除法运算符“/”时，若参与运算的变量均为整数时，其结果也为整数(舍去小数)；若除数或被除数中有一个为负数，则舍入的方向是不固定的。例如： $-7/4$ ，在有的机器上得到结果为 -1 ，而有的机器上得到结果 -2 。多数机器上采取“向零取整”原则，例如： $7/4=1$ ， $-7/4=-1$ ，取整后向零靠拢。
- (3) 使用求余运算符(模运算符)“%”时，要求参与运算的变量必须均为整型，其结果值为两数相除所得的余数。一般情况下，所得的余数与被除数符号相同。例如：
 $7\%4=3$ ， $10\%5=0$ ， $-8\%5=-3$ ， $8\%-5=3$ 。



第2章 数据表示及数据运算

2. 算术表达式

用运算符、圆括号将运算对象连接起来的符合C语法规则的式子，称为**表达式**。

用算术运算符、圆括号将运算对象(或称操作数)连接起来的符合C语法规则的式子，称为**C算术表达式**。其中运算对象可以是常量、变量、函数等。

例如：

`a*b/c-1.5+'a'`



第2章 数据表示及数据运算

C算术表达式的书写形式与数学中表达式的书写形式是有区别的，在使用时要注意以下几点：

(1) C表达式中的乘号不能省略。例如：

数学式 b^2-4ac ，相应的C表达式应写成 $b*b-4*a*c$ 。

(2) C表达式中只能使用系统允许的标识符。例如：

数学式 πr^2 相应的C表达式应写成 $3.14*r*r$ 。

(3) C表达式中的内容必须书写在同一行，不允许有分子分母形式，必要时利用圆括号保证运算的顺序。例如：

$$\frac{a+b}{c+d}$$

数学式相应的C表达式应写 $(a+b)/(c+d)$ 。

(4) C表达式不允许使用方括号和花括号，只能使用圆括号帮助限定运算顺序。可以使用多层圆括号，但左右括号必须配对，运算时从内层圆括号开始，由内向外依次计算表达式的值。



第2章 数据表示及数据运算

3. 算术运算符的优先级和结合性

C语言规定了在表达式求值过程中各运算符的优先级和结合性。

(1) **优先级**：是指当一个表达式中如果有多个运算符时，则计算是有先后次序的，这种计算的先后次序称为运算符的优先级。

(2) **结合性**：是指当一个运算对象两侧的运算符的优先级别相同时，进行运算的结合方向。按“从右向左”的顺序运算，称为右结合性；按“从左向右”的顺序运算，称为左结合性。表2.6中给出了算术运算符的优先级和结合性。

运算种类	结 合 性	优 先 级
*、/、%	从左向右	高 ↓ 低
+, -	从左向右	

第2章 数据表示及数据运算

例2.9 算术运算符/和%举例。程序代码如下：

```
#include <stdio.h>
void main()
{
    int x,y;
    x=10;
    y=3;
    printf("%d\n",x/y);
    printf("%d\n",x%y);
    x=1;
    y=2;
    printf("%d\n",x/y);
    printf("%d\n",x%y);
}
```



自增、自减运算符

运算符	名 称	举 例	等价运算
++	自加 1	i++或++i	i=i+1
--	自减 1	i--或--i	i=i-1

自增、自减运算符可以用在运算量之前(如++i、--i)，称为**前置**运算；自增、自减运算符可以用在运算量之后(如i++、i--)，称为**后置**运算。

简单规则：

前置——先自加，后引用

后置——先引用，后自加

int i=3,j; int i=3,j;

j=++i; j=i++;



例2.10 以下程序的输出结果是()。

```
#include<stdio.h>
void main ()
{
int i=010,j=10;
printf("%d ",++i);
printf("%d\n",j- -);
}
```

A) 11 10 B) 9 10 C) 010 9 D) 10 9



第2章 数据表示及数据运算

- 2.8 运算符和表达式

位操作运算符

- **&**
 - 按位与运算
- **|**
 - 按位或运算
- **^**
 - 按位异或运算
- **<<**
 - 按位左移运算
- **>>**
 - 按位右移运算
- **~**
 - 按位求反



第2章 数据表示及数据运算

- 2.8 运算符和表达式

赋值运算符

- 下面两个语句是等价的

- `i = i + 2;`

- `i += 2;`

- `+`、`-`、`*`、`/`、`%`、`<<`、`>>`、`&`、`^`、`|` 运算符都可以按此种方式处理

- 这种形式看起来更直观，而且执行效率一般也能更高一些



第2章 数据表示及数据运算

- 2.8 运算符和表达式

逗号运算符

- 表达式1, 表达式2, ..., 表达式n
- 多数情况下, 并不使用整个逗号表达式的值, 更常见的情况是要分别得到各表达式的值
- 主要用在循环语句中, 同时对多个变量赋初值等

```
for (i = 0 , j = 0; i < j; i++, j++)
```



第2章 数据表示及数据运算

- 2.8 运算符和表达式

条件表达式

- 把a和b中的最大值放入z中
 - `if (a > b)`
 `z = a;`
 `else`
 `z = b;`
 - `z = (a > b) ? a : b;`
- 此种表达式切忌用得过于繁杂



第2章 数据表示及数据运算

- 2.1 数据类型的一般概念
- 2.2 常量与变量
- 2.3 整型数据
- 2.4 浮点型数据
- 2.5 字符型数据
- 2.6 变量赋初值
- 2.7 不同数据类型数据间的混合运算
- 2.8 运算符和表达式
- 2.9 运算符的优先级和结合性



第2章 数据表示及数据运算

- 2.9 运算符的优先级和结合性

优先级

- () [] -> .
- ! ~ ++ -- + - * & (类型) sizeof
- * / %
- + -
- << >>
- < <= > >=
- == !=
- &
- ^
- |
- &&
- ||
- ? :
- = += -= *= /= %= &= ^= |= <<= >>=
- ,

注意：能背下优先级？

用括号来控制运算顺序更直观、
方便，并减少出错的概率



第2章 数据表示及数据运算

- 2.9 运算符的优先级和结合性

结合性

- ✓ 多数运算符为左结合；
- ✓ 少数为右结合：
 - 一元运算符
 - 赋值运算符
 - 条件运算符



第2章 数据表示及数据运算

- 2.6 变量赋初值

- 1. 变量赋值运算

格式:

V=expression; /* 将expression的值赋给变量V*/

int a=2; /*定义同时赋值*/

int b;

b=3+2; /*先定义，后赋值*/

说明:

(1) =是赋值号（动词）；

(2) 赋值运算的左值只能是变量。

(3) 赋值号两边类型应该一致，如不一致，需进行类型转换。

int a=2.5 ; /* 2→a */

a=a+2;

iCount=iCount+10;

a+2=12; /*Error!*/



作业

一、填空题

- (1) 在C语言中，用“\”开头的字符序列称为转义字符。转义字符“\n”的功能是_____；转义字符“\r”的功能是_____。
- (2) 运算符“%”两侧运算对象的数据类型必须都是_____；运算符“++”和“--”运算对象的数据类型必须是_____。
- (3) 表达式 $8/4*(int)2.5/(int)(1.25*(3.7+2.3))$ 值的数据类型为_____。
- (4) 表达式 $(3+10)/2$ 的值为_____。
- (5) 设 $x=2.5$ ， $a=7$ ， $y=4.7$ ，则算术表达式 $x+a\%3*(int)(x+y)\%2/4$ 的值是_____。



二、选择题

(1) 下列4组选项中，均不是C语言关键字的选项是_____。

- | | | |
|------------|------|--------|
| A. define | IF | type |
| B. getc | char | printf |
| C. include | case | scanf |
| D. while | go | pow |

(2) 下列4组选项中，均是合法转义字符的选项是_____。

- | | | | |
|----------|-------|-------|---------|
| A. \" | \\ | \\n | B. \ |
| | \\017 | \\ | |
| C. \\018 | \\f | 'xab' | D. \\0 |
| | 'xlf' | | '\\101' |

(3) 已知字母 'b' 的ASCII码值为98，如ch为字符型变量，则表达式ch='b'+5-2'的值为_____。

- | | | | |
|------|------|--------|--------|
| A. e | B. d | C. 102 | D. 100 |
|------|------|--------|--------|



(4) 以下表达式值为3的是_____。

- A. $16-13\%10$ B. $2+3/2$ C. $14/3-2$
D. $(2+6)/(12-9)$

(5) 以下叙述不正确的是_____。

- A. 在C程序中，逗号运算符的优先级最低
B. 在C程序中，MAX和max是两个不同的变量
C. 若a和b类型相同，在计算了赋值表达式a=b后，b中的值将放入a中，而b中的值不变
D. 当从键盘输入数据时，对于整型变量只能输入整型数值，对于实型变量只能输入实型数值



(6) 以下非法的赋值语句是

A. $n=(i=2,++i);$ B. $j++;$ C. $++(i+1);$ D. $x=j>0;$

(7) 以下选项中合法的实型常数是

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

(8) 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

(9) 与数学式子 $3*x^n/(2x-1)$ 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) 已有定义: `int x=3,y=4,z=5;` , 则表达式 `!(x+y)+z-1&& y+z/2` 的值是

A. 6 B. 0 C. 2 D. 1

(11) 若有定义: `int a=8, b=5, c;` , 执行语句 `c=a/b+0.4;` 后, c 的值为

A. 1.4 B. 1 C. 2.0 D. 2

(12) 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。



(13) 以下选项中合法的字符常量是

A. "B" B. '\010' C. 68 D. D

(14) 设正x、y均为整型变量，且x=10, y=3，则以下语句的输出结果是

```
printf("%d,%d\n",x--,--y);
```

A. 10,3 B. 9,3 C. 9,2 D. 10,2



三、写出以下程序运行的结果。

(1)

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
char c1 = '6', c2 = '0';
```

```
printf("%c,%c,%d\n", c1, c2, c1-c2);
```

```
}
```





(2)

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
int x = 010, y = 10, z = 0x10;
```

```
printf("%d,%d,%d\n", x, y, z);
```

```
}
```





(3)

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
int a = 2, b = 3;
```

```
float x = 3.9, y = 2.3;
```

```
float r;
```

```
r = (float)(a + b) / 2 + (int)x % (int)y;
```

```
printf("%f\n", r);
```

```
}
```





(4)

```
#include <stdio.h>
```

```
void main()
```

```
{ int a=5,b=4,c=6,d;
```

```
printf("%d\n",d=a>b?(a>c?a:c):(b));
```

```
}
```

(5)

```
#include <stdio.h>
```

```
void main()
```

```
{ int a=4,b=5,c=0,d;
```

```
d=!a&&!b||!c;
```

```
printf("%d\n",d);
```

```
}
```





(6)

```
#include <stdio.h>
void main()
{ int a=177;
printf("%o\n",a);
}
```

(7)

```
#include <stdio.h>
void main()
{ int a=0
a+=(a=8);
printf("%d\n",a);
}
```





(8)

```
#include <stdio.h>
```

```
void main()
```

```
{ int a=5,b=4,c=3,d;
```

```
d=(a>b>c);
```

```
printf("%d\n",d);
```

```
}
```



四、编程题

- (1) 输入长方形的长和宽，输出长方形的周长和面积。
- (2) 输入一个字符，输出其ASCII代码。
- (3) 输入3个整数，计算并输出它们的平均值。
- (4) 已知整型变量a、b、c的值，根据以下算式编写程序求y的值。



五、设a和n已定义为整型变量，a=12，求下面表达式运算后a的值。

(1) $a+=a$

(2) $a-=2$

(3) $a*=2+3$

(4) $a/=a+a$

(5) $a\%=(n\%=2)$ ，n的值等于5

(6) $a+=a-=a*=a$

