

第2章

用C语言编写简单程序





Contents

- **数据类型、运算符和表达式**
- **格式输入与输出**
- **C语言的语句**
- **顺序结构程序设计**



2.1 引例

例2-1 求摄氏温度100°C对应的华氏温度，计算公式如下：

$f = (9/5)c + 32$ ，式中：c表示摄氏温度，f表示华氏温度。

```
#include <stdio.h>
void main( )
{
    double celsius,fahr;           /*定义两个实型变量*/
    celsius=100;                   /*对变量celsius赋值*/
    fahr=9.0/5.0*celsius+32;       /*温度转换计算*/
    printf("celsius=%f,fahr=%f\n",celsius,fahr);
                                    /*显示计算结果*/
}
```



2.2 数据类型

2.2.1 C语言的数据类型

- 基本数据类型
 - 整型**int**
 - 实型（浮点型）：单精度型**float**、双精度型**double**
 - 字符型**char**
- 构造数据类型
 - 数组、结构体、共用体、枚举
- 指针类型
- 空类型



2.2.2 常量和变量

- 在程序执行过程中，其值不发生改变的量称为**常量**，其取值可变的量称为**变量**。
- **常量**可以不经说明而直接引用。
- **变量**则必须先定义后使用。变量定义的一般形式为：
数据类型 变量名表；
- 对于某些有特定含义的、经常使用的常量可以用**符号常量**来代替。可增加程序的可读性和可维护性。符号常量定义的一般格式为：

#define 符号常量 常量

大写字母



2.2.3 整型数据

■ 整型常量 (整常数)

- 八进制整常数

八进制整常数必须以0开头

Eg : 016(十进制为14) 0101(十进制为65)

- 十六进制整常数

十六进制整常数的前缀为0X或0x

Eg : 0X1A(十进制为26) 0XA0 (十进制为160)

- 十进制整常数

Eg : -258 678 1828

■ 整型变量

可以根据数据的取值范围和所占内存的字节数，
将变量定义为六种整型类型：

整型类型 变量名表；

Eg : **int a,b,c;**

地址	存储单元	变量名
110200000	100	a
110200004	200	b

a=100;
b=200;



VC++6.0中各整型类型的表示、分配的字节数和取值范围

类型定义关键字	名称	分配的字节数	取值范围
[signed] int	基本整型	4	-2147483648 ~2147483647
[signed] short [int]	短整型	2	-32768~32767
[signed] long [int]	长整型	4	-2147483648 ~2147483647
unsigned [int]	无符号基本整型	4	0~4294967295
unsigned short [int]	无符号短整型	2	0~65535
unsigned long [int]	无符号长整型	4	0~4294967295

2.2.4 实型数据（浮点型数据）

■ 实型常量（实数/浮点数）

- 十进制小数形式

由0~9和小数点组成

Eg : 0.0、-2.87、3.8、4.、.77等

- 十进制指数形式

格式：**a E n** （其值为 $a \times 10^n$ ）

↓
十进制数

↓
十进制整常数

Eg : 3.4E6 (即 3.4×10^6), 9.6E-4 (即 9.6×10^{-4})



■ 实型变量

可以根据数据的取值范围和所占内存的字节数，
将变量定义为六种整型类型：

类型定义 关键字	名称	分配的 字节数	取值范围	有效数字位数
float	单精度型	4	$-3.4\text{E}+38 \sim 3.4\text{E}+38$	7
double (默认)	双精度型	8	$-1.7\text{E}+308 \sim 1.7\text{E}+308$	15~16



2.2.5 字符型数据

■ 字符常量

- 用单引号括起来的一个字符

Eg : 'x'、'y'、'\$'、'？'

- 转义字符

格式： 以反斜线\开头的字符序列

此字符序列具有特定的含义，故称“转义”字符。

主要用来表示那些用一般字符不便于表示的控制代码



常用转义字符及功能

\n	换行
\t	横向跳到下一制表位置
\"	双引号
\'	单引号
\\	反斜线
\ddd	1~3位八进制数所代表的字符 (ASCII)
\xhh	1~2位十六进制数所代表的字符 (ASCII)

■ 字符变量

- 字符变量说明的格式：

char 变量表；

Eg：char a,b;

- 占用一个字节的内存空间

Eg：a = 'x' ;

a	0	1	1	1	1	0	0	0
---	---	---	---	---	---	---	---	---

并不是把字符常量'x' 存放a内存单元中，存放的是'x'的ASCII码

∴也可以把字符变量a看成是整型变量，C语言允许字符变量赋以整型值并按整型数据形式输出，也允许整型变量赋以字符值并按字符型数据输出。

But，整型数据至少占用2B空间，而字符型数据仅占用1B空间。



2.2.6 变量赋初值

C语言程序中可在定义变量的同时赋以初值，这种方法称为**初始化**。一般形式为：

类型说明符 变量1= 值1，变量2= 值2，……；

Eg : int a=5, b=6;

float x=4.7, y=38.6, z=8.72;



2.3 运算符与表达式

2.3.1 C语言运算符简介

- 运算符具有不同的优先级，共分为15级。1级最高，15级最低，优先级较高的先于优先级较低的进行运算；
- 运算符具有结合性。在一个运算量两侧的运算符优先级相同时，则按运算符的结合性所规定的结合方向处理。C语言中各运算符的结合性分为两种，即**左结合性**(自左至右运算)和**右结合性**(自右至左运算)。



2.3.2 算术运算符与算术表达式

1. 基本算术运算符

- 加法 $+$ ，双目运算，左结合性
- 减法 $-$ ，双目运算，左结合性
- 乘法 $*$ ，双目运算，左结合性
- 除法 $/$ ，双目运算，左结合性。**注意：当双目均为整型数据时，结果也为整型，舍去小数。如：18/4，结果为4，而非4.5**
- 求余 $%$ ，双目运算，左结合性。**要求：双目均为整型。如：7%3，结果为1**

优先级：“先乘除后加减”，即 $*$ 、 $/$ 、 $%$ 优于 $+$ 、 $-$

2. 负值运算符

-，单目运算符，如-5，-y，具有右结合性。

3. 自增、自减运算符

- 自增运算符**++**，使变量的值增1
- 自减运算符**--**，使变量的值减1
- 均为单目运算符，只能用于变量

++i	i增1后再参与其他运算
--i	i减1后再参与其他运算
i++	i参与运算后，i的值再增1
i--	i参与运算后，i的值再减1

例如： `int x=5, y;`

`y=x++;`

结果：y的值为5，x的值为6

`y=++x;`

结果：y的值为6，x的值为6

4. 算术表达式：由算术运算符和括号将运算对象连接起来的式子

2.3.3 赋值运算符与赋值表达式

- 赋值表达式一般形式为：**变量=表达式**；

例如： $x=a+b$;
 $y=b+x++$;

- 赋值运算符具有右结合性，如： $x=y=z=8$ ，可理解为 $x=(y=(z=8))$
- 如果赋值运算符=两边的数据类型不相同，系统将自动进行类型转换，即把=右边的类型转换成左边的类型
- 复合赋值运算符， $+=$ 、 $-=$ 、 $*=$ 、 $/=$ 、 $\%=$ 、 $<<=$ 、 $>>=$ 、 $\&=$ 、 $\wedge=$ 、 $|=$
如： $x+=8$ ； 相当于 $x=x+8$;
 $a*=b+6$ ； 相当于 $a=a*(b+6)$;



2.3.4 逗号运算符与逗号表达式

- 逗号运算符 “**,**” , 其功能是把两个表达式连接起来 , 组成一个表达式 , 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 , ... , 表达式n
其求值过程 : 先计算表达式1的值 , 然后计算表达式2值 , , 最后计算表达式n的值 , **并以表达式n的值作为整个逗号表达式的值。**
例如 : $a=3, b=5, c=a+b$ 的值为8
- 并不是在所有出现逗号的地方都组成逗号表达式 , 如在变量说明中、函数参数表中逗号只是用作各变量之间的**分隔符**。

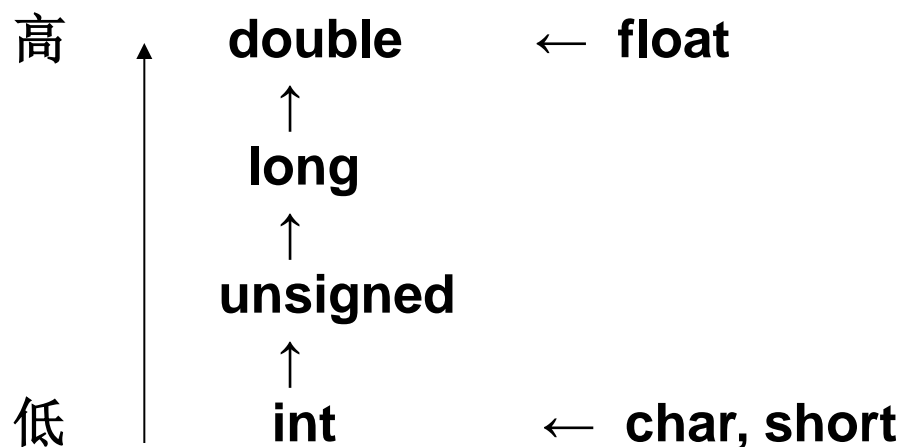


2.4 各类数值型数据间的混合运算

- 在C语言中，允许整型、实型、字符型数据间进行混合运算。
- 但这些数据首先要转换成同一类型，然后再作运算。
- 数据类型的转换包括**自动转换**和**强制转换**。自动转换由C语言编译系统自动完成，强制转换则通过特定的运算完成。

2.4.1 自动类型转换

1. 非赋值运算的类型转换



水平方向：自动从右向左转换

垂直方向：经水平方向转换后，若类型仍不相同，再自动转换成**其中**级别最高的类型



2. 赋值运算的类型转换

当赋值运算符两侧的类型不一致时，其转换规则是将赋值运算符**右侧**的类型**转换为左侧**变量的类型，**然后再进行赋值操作**。这种情况下，可能会引起**数值溢出或产生舍入误差**。

例如：
`int i;`
`i=5.67;`
结果i的值为5



2.4.2 强制类型转换

- 使用强制转换运算符，可以将一个表达式转换成给定的类型。
- 其一般形式为：
 (数据类型名) 表达式
 例如，(long)a 强制将a临时转换为长整型
 (float)(x+y) 强制将x+y的结果临时转换为单精度实型
- 无论是强制转换还是自动转换，都只是一种作用于本次运算的临时性转换，而不会改变数据原来的类型。



2.5 数据的输入输出

C语言中，所有的输入输出都是由**库函数**完成地方，在使用这些函数时，需要使用**预处理命令**

“**#include**”将头文件stdio.h包含到用户源程序中，
即**#include <stdio.h>**



2.5.1 标准字符输入输出函数

1. 标准字符输出函数 `putchar()`

功能：向标准输出设备输出一个字符

调用形式：`putchar(c);`

其中，`c`为一个字符常量、字符变量、整型常量、整型变量或整型表达式。

2. 标准字符输入函数 `getchar()`

功能：从标准输入设备（一般为键盘）读入一个字符，并立即在显示器上显示该字符（称作回显）

调用形式：`getchar();`

2.5.2 格式输出函数 printf()

- 功能：向显示器输出若干个任意类型的数据
- 调用形式：**printf(“格式控制字符串” , 输出表列) ;**

用于指定输出格式

格式字符串 + 非格式字符串



% [附加格式说明符]格式字符

Page23 表2-3、2-4



原样输出，在显示中起提示作用

列出了各个输出项，要求格式字符串和各输出项在**数量**和**类型**上应该**一一对应**



例2-2 格式输出整数型和实数型。

```
#include <stdio.h>
void main( )
{
    int a=18;
    float b=3.1415926;
    printf( "a=%d , b=%f\n",a,b);
}
```

运行结果：

a=18 , b=3.1415926

2.5.3 格式输入函数 **scanf()**

- 功能：按用户指定的格式从键盘上把数据输入到指定的变量之中。
- 调用形式：**scanf(“格式控制字符串” , 地址表列) ;**

↓
格式字符串：

% [附加格式说明符] 格式字符

- ↓
- ✓ 需要接受输入数据的**所有变量的地址**，而不是变量本身（变量名前加地址运算符 **&**）
 - ✓ 各地址之间要用逗号 **,** 分隔



■ scanf()的常用附加格式说明符及其作用：

字母l	用于输入long型整数（%ld）或double型实数（%lf）
字母m	用于指定输入数据的宽度


■ 注意事项：

1) scanf()函数中没有精度控制

scanf("%8.4f" ,&a); **非法 x**

2) 输入多个数值数据时，若格式控制串中有规定的间隔符时，则在输入数值时必须使用该间隔符；
否则，可用空格、Tab键、回车键作间隔。

例如： scanf("%d,%d,%d" ,&a,&b,&c);
运行时输入 6,7,8<回车键>



```
#include <stdio.h>
void main( )
{
    double celsius,fahr;
    printf( "请输入摄氏温度：" );
    scanf( "%lf" ,&celsius);
    fahr=9.0/5.0*celsius+32;
    printf( "celsius=%lf, fahr=%lf\n",celsius,fahr);
}
```

运行结果：

请输入摄氏温度：100 ✓

celsius=100.000000, fahr=212.000000



2.6 顺序结构程序设计

2.6.1 C 语言的语句

1. 表达式语句

组成：**表达式、分号 “ ; ”**

执行：就是计算表达式的值

2. 函数调用语句

组成：**函数名、实际参数、分号 “ ; ”**

执行：调用函数体并把实际参数赋予函数定义中的形式参数，然后执行被调函数体中的语句，求取函数值

例如：`printf(“This is the first C program”);`
调用库函数，输出字符串。



3. 控制语句

功能：用于完成一定的控制功能，以实现结构化程序设计

分类：C语言有九种控制语句。可分成以下三类：

- 1) **条件判断语句：**if语句、switch语句
- 2) **循环语句：**while语句、do while语句、for语句
- 3) **转向语句：**goto语句、break语句、continue语句、return语句

4. 空语句

组成：只有分号“；”组成的语句

功能：什么也不执行

5. 复合语句

组成：把若干条语句用大括号{}括起来组成的语句

执行：执行时复合语句是一条语句，而不是多条语句

2.6.2 顺序结构程序设计举例

- 顺序结构是由一组按先后书写顺序执行的程序块所组成
- 顺序结构是结构化程序设计中最简单的一种。

例2-4 输入三角形的三条边长，求该三角形的面积

```
#include <stdio.h>
#include <math.h>
void main( )
{   double a,b,c,s,area;
    printf("请输入三角形的三条边 a,b,c:");
    scanf("%lf,%lf,%lf",&a,&b,&c);
    s=1.0/2*(a+b+c);
    area=sqrt(s*(s-a)*(s-b)*(s-c));
    printf("a=%7.2f b=%7.2f c=%7.2f\n",a,b,c);
    printf("s=%7.2f area=%7.4f\n",s,area);
}
```

运行结果：

请输入三角形的三条边 a,b,c:6,8,10 ✓

a= 6.00 b= 8.00 c= 10.00

s= 12.00 area=24.0000



例2-5 从键盘上输入一个小写字母，要求改用大写字母输出。

```
#include <stdio.h>
void main( )
{
    char c1,c2;
    printf("Please input a lower letter: ");
    scanf("%c",&c1);
    c2=c1-32;
    /*小写字母与大写字母的ASCII差值为32*/
    printf("Upper letter is %c\n",c2);
}
```

运行结果：

Please input a lower letter:d ✓
Upper letter is D



例2-6 读入圆的半径 r ，计算该圆的周长及面积。

```
#define PI 3.141593
#include <stdio.h>
void main()
{
    double r,circum,area;
    printf("请输入半径:");
    scanf("%lf",&r);    /*双精度变量格式输入必须用%lf*/
    circum=2*r*PI;
    area=PI*r*r;
    printf("r=%10.2lf,circum=%10.2lf,area=%10.2lf\n",r,circum,area);
}
```

运行结果：

请输入半径:3 ✓

r= 3.00,circum= 18.85,area= 28.27



End of Chapter 2