2

数据的定义和运算

每种程序设计语言都有自己独特的语法体系，都有自己能处理的数据范围。掌握语法体系和数据范围对学习程序设计语言来说是非常重要的。本单元主要介绍 C 语言的常量与变量、数据类型、运算符和表达式、数据类型转换和标识符，这些都是 C 语言程序设计的基础。

内容摘要：

- 基本数据类型
- 常量和变量
- 标识符命名
- 常用运算符和表达式
- 运算符的优先级与结合性

学习目标：

- 理解常量和变量的含义
- 熟悉基本数据类型：int、char、float 和 double
- 使用算术运算符
- 理解类型转换

任务 1　常量和变量

知识与技能：

- C 语言的数据类型分类

- 掌握 C 语言程序设计中的常量和变量
- 熟悉常量、变量的表示方法

任务引导

在程序设计中，所有的程序都会涉及到待处理的数据。不同类型的数据既可以以常量的形式出现，也可以以变量的形式出现。C 语言既提供了丰富的数据类型对不同的数据加以描述，又提供了丰富的运算符和表达式对数据进行加工。

根据数学知识，学生的平均成绩和总成绩都可以利用公式来求，用 sum 代表学生的总成绩，用 avg 代表学生的平均成绩，用 N 代表学生课程数量，这里 N 是固定不变的，而总成绩和平均成绩是可变的，这些元素如何在 C 语言里进行定义？这就是本任务要重点讲述的内容。

知识点介绍

1. 数据类型

在本任务中，只介绍数据类型说明，其他说明在以后各单元中陆续介绍。所谓数据类型是按被说明量的性质、表示形式、占据存储空间的多少、构造特点来划分的。在 C 语言中，数据类型可分为基本数据类型、构造数据类型、指针类型、空类型四大类，如图 2-1-1 所示。

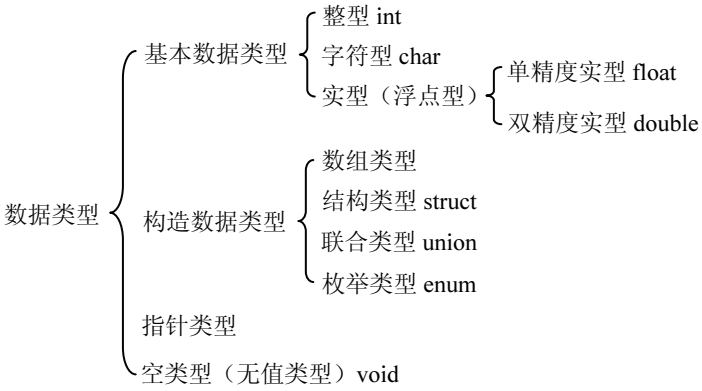

图 2-1-1　C 语言的数据类型

2. 常量（直接常量、符号常量）

在程序执行过程中，其值不发生改变的量称为常量。C 程序设计中的常量分为直接常量和符号常量两类。

（1）直接常量。

直接常量分为算术型运算常量和字符型常量两种。

算术型运算常量也叫整型常量或实型常量，如 2、0、-24、3.1415926 等。

字符型常量是用双引号或单引号括起来的一串字符，如'F'、"123a"、"HK"、"2a+b="等。

（2）符号常量。

用一个标识符代表一个常量，这样的标识符称为符号常量。程序中使用符号常量可提高程序的易读性、可修改性，便于调试程序，减少出错机会。如用 PI 代表 3.1415926。

符号常量的值在其作用域内不能改变，也不能再被赋值。如在程序中，对 PI 重新赋值：PI=2;，这样是不允许的。

习惯上，符号常量名用大写，变量名用小写。

3. 变量（变量的定义、变量的赋值）

变量是指其值可以改变的量。一个变量应该有一个名字（标识符）标明存储单元，在该存储单元中存放变量的值。变量名就是这个量的代号，就像每个人都有名字一样，而变量值是这个量的取值。所有的 C 语言变量必须先定义后使用。

变量定义的一般形式如下：

```
数据类型符    变量名列表;
```

例如：

```
int a;
```

数据类型符用来说明变量的数据类型，数据类型可以是 C 语言中任意一种基本数据类型或构造数据类型。

变量取名应符合标识符的命名规则，最后一个变量名之后必须以“;”结尾。例如：

```
char c1;
```

变量名列表中多个变量之间用逗号隔开，数据类型符与变量名之间必须用空格隔开。例如：

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

4. 变量赋值

定义变量后，在使用之前需要给定一个初始值。在 C 语言中，可以通过赋值运算符“=”给变量赋值。变量赋值语句的一般格式为：

```
变量名=表达式;
```

变量赋值一般有以下两种情况：

（1）先定义变量，后赋值。例如：

```
int   r;
r=1;
```

（2）变量的初始化。

在定义变量的同时为其赋值，称为变量的初始化。定义的变量可以全部初始化，也可以部分初始化。对于上面的语句也可以这样写：

```
int   r=1;
```

即定义整型变量 r 的同时，对其赋初值为 1。

注意：

① 变量在某一时刻只有一个确定的值，变量获得新值后，其原值将不再存在。例如：

```
int   r;r=1; r=2;
```

该程序执行后，变量 r 的值是 2，而不是 1。

② 定义多个同类型的变量时，如果给所有变量赋同一个值，只能逐个处理。例如有三个整型变量 x、y、z，且初值均为 10，可以写成下面的形式：

```
int   x=10,y=10,z=10;
```

③ 如果变量的类型与所赋数据的类型不一致，所赋数据将被转换成与变量相同的类型。例如下面的定义是合法的：

```
int   x=10.5;
long   y=99;
```

该程序执行后，变量 x 的值是整数 10（只将整数部分赋给变量 x），变量 y 的值是长整数 99。

任务的实现

【例 2-1-1】程序代码如下：

```
#define   PI   3.1415926                    /*PI 为符号常量*/
void main()
{
   float   r,circle,area;                   //r、circle、area 为单精度型的变量
   r=2.3;                                   /*对 r 赋值*/
   circle=2*PI*r;                           /*求圆的周长*/
   area=PI*r*r;                             /*求圆的面积*/
   printf("circle=%f\narea=%f\n",circle,area);   /*输出圆的周长和面积*/
}
```

程序运行结果如图 2-1-2 所示。

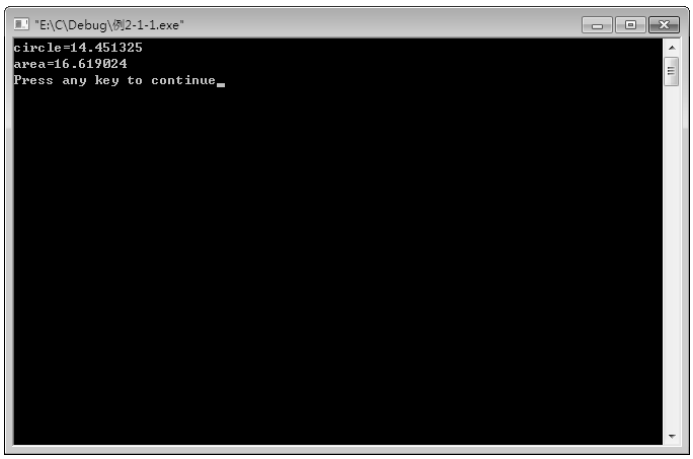

图 2-1-2　求圆的周长和面积

说明：

（1）编译预处理命令#define 可以用来定义符号常量，该程序用#define 命令行定义 PI 代表常量 3.1415926，后续程序语句中出现的符号 PI 都代表 3.1415926，若需要修改符号常量 PI 的值，只需在#define 命令行中进行修改。

（2）程序中定义的符号常量命名时最好遵循“见名知意”的原则，这样可以增加程序的可读性。为了与变量名相区别，习惯上符号常量用大写表示。符号常量不占用内存，在预编译结束后就不存在了，因此不能对符号常量赋以新值。

知识扩展

1. 标识符

（1）标识符是对变量名、函数名、标号和其他各种用户定义的对象命名。

（2）命名规则：标识符由字母、数字、下划线组成，且第一个字符必须是字母或下划线。

（3）标识符区分大小写。

（4）标识符的有效长度取决于具体的 C 编译系统。

（5）标识符的书写一般采用具有一定实际含义的单词，这样可提高程序的可读性。

（6）标识符不能与 C 语言的关键字同名，也不能与自定义函数或 C 语言库函数同名。

2. 关键字

关键字是具有固定名字和特定含义的特殊标识符，也称保留字，不允许程序设计者将它们另作别用。

C 语言的关键字有 32 个。

3. return 语句

return 用于从函数返回，使执行的函数返回到函数的调用点。

return 的一般形式为：

```
return 表达式;
```

其中表达式就是函数返回的值。

函数使用 return 的次数不受限制，但是当函数遇到第一个 return 时，函数停止执行，返回到函数的调用点。

任务 2　数据类型

知识与技能：

- 掌握 C 语言的基本数据类型
- 掌握基本数据类型的输入和输出方法

任务引导

计算机中有各种各样的程序，每个程序需要处理的信息类型也各不相同，包括文字、数字、图形、声音、动画等，这些信息在程序中可以通过不同的数据类型进行定义，因此使用各种数据类型实现常量、变量数据的定义是程序设计的基本能力。

知识点介绍

1. 整型数据

（1）整型常量的表示方法。

整型常量即整常数。在 C 语言中，整常数可用以下三种形式表示：

- 十进制整数。如 123、-456.4。
- 八进制整数。以 0 开头的数是八进制数。如 0123 表示八进制数 123，等于十进制数 83；-013 表示八进制数-13，即十进制数-11。
- 十六进制整数。以 0x 开头的数是十六进制数。如 0x153 代表十六进制数 153，等于十进制数 339；-0x12 等于十进制数-18。

（2）整型变量。

整型数据在内存中是以二进制形式存放的，例如：

```
int i;          /*定义为整型变量*/
i=10;           /*给 i 赋以整数 10*/
```

注意：十进制数 10 的二进制形式为 1010，Turbo C 2.0 和 Turbo C++ 3.0 为一个整型变量在内存中分配 2 个字节的存储单元（不同的编译系统为整型数据分配的字节数是不相同的，Visual C++ 6.0 则分配 4 个字节）。

数值是以补码（Complement）表示的。

整数类型分为如表 2-2-1 所示的几类。

表 2-2-1 整数类型分类

类型	类型说明符	长度	数的范围
基本型	int	2 字节	-32768～32767
短整型	short	2 字节	-2^{15}～$2^{15}-1$
长整型	long	4 字节	-2^{31}～$2^{31}-1$
无符号整型	unsigned	2 字节	0～65535
无符号短整型	unsigned short	2 字节	0～65535
无符号长整型	unsigned long	4 字节	0～($2^{32}-1$)

（3）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”。例如：

```
int a,b                 /*指定变量 a、b 为整型*/
unsigned short c,d;     /*指定变量 c、d 为无符号短整型*/
long e,f;               /*指定变量 e、f 为长整型*/
```

（4）整型数据的溢出。

在给整型变量赋值时，要注意其取值范围，如果将一个大于 32767 或小于-32767 的数赋值给一个 int 型变量则会产生溢出。

（5）整型常量的类型。

① 一个整数，如果其值在-32768～+32767 范围内，认为它是 int 型，它可以赋值给 int 型和 long int 型变量。

② 一个整数，如果其值超过了上述范围，而在-2147483637～+2147483647 范围内，则认为它是长整型，可以将它赋值给一个 long int 型变量。

③ 如果所用的 C 版本（如 Turbo C）分配给 short int 与 int 型数据在内存中占据的长度相同，则它的表数范围与 int 型相同。因此一个 int 型的常量同时也是一个 short int 型常量，可以赋给 int 型或 short int 型变量。

④ 一个整常量后面加一个字母 u 或 U，认为是 unsigned int 型，如 12345u，在内存中按 unsigned int 规定的方式存放（存储单元中最高位不作为符号位，而用来存储数据）。如果写成 -12345u，则先将-12345 转换成其补码 53191，然后按无符号数存储。

⑤ 在一个整常量后面加一个字母 l 或 L，则认为是 long int 型常量。例如 123l、432L、0L 等。这往往用于函数调用中。如果函数的形参为 long int 型，则要求实参也为 long int 型。

2. 浮点型数据

带有小数点的数称为浮点型数据，也叫实型数据。

（1）浮点型常量。

在 C 语言中，浮点型常量有两种表示方式：

- 十进制小数形式：由数字 0～9 和小数点组成，必须有小数点，但小数点前后的 0 可以省略。例如 5.34、0.54、.54 都是十进制小数形式的合法表示。
- 指数形式：指数形式又称为科学记数法，由十进制小数、阶码标志“e”或“E”以及阶码（只能为整数，可以带符号）组成。如 1.8e-3、-1.25e-6、-.1e-4 都是合法的表示形式；E2、e2.1、e3.2 都不是合法的表示形式。

注意：字母 e（或 E）之前必须有数字，且 e 后面的指数必须为整数。

（2）浮点型变量。

1）浮点型数据在内存中的存放形式。一个浮点型数据一般在内存中占 4 个字节（32 位）。与整型数据的存储方式不同，浮点型数据是按照指数形式存储的。系统把一个浮点型数据分成小数部分和指数部分，分别存放。指数部分采用规范化的指数形式。

2）浮点型变量的分类。浮点型变量分为单精度型（float）、双精度型（double）和长双精度型（long double）三类，如表 2-2-2 所示。

表 2-2-2　浮点型变量分类

类型	位数	数的范围	有效数字
float	32	10^{-37}～10^{38}	6～7 位
double	64	10^{-307}～10^{308}	15～16 位
long double	128	10^{-4931}～10^{4932}	18～19 位

3）浮点型数据的误差。由于浮点型变量的存储单元有限，单精度实型的有效位数是 7 位，双精度实型的有效位是 16 位，Turbo C 中规定小数点后最多保留 6 位，因此在进行赋值和计算时会产生误差。

3. 字符型数据

（1）字符常量。

用单引号包含的一个字符是字符型常量，只能包含一个字符，有些以“\”开头的特殊字符称为转义字符：

\n　换行

\t　横向跳格

\r　回车

\\　反斜杠

\ddd　ddd 表示 1～3 位八进制数字

\xhh　hh 表示 1～2 位十六进制数字

（2）字符变量。

字符型变量用来存放字符常量，注意只能放一个字符。

字符变量的定义形式为：

```
char c1,c2;
```

在本函数中可以用下面的语句对 c1 和 c2 赋值：

```
c1='a'; c2='b';
```

一个字符变量在内存中占一个字节。

（3）字符数据在内存中的存储形式及其使用方法。

将一个字符常量放到一个字符变量中，实际上并不是把该字符本身放到内存单元中去，而是将该字符相应的 ASCII 代码放到存储单元中。

这样就使字符型数据和整型数据之间可以通用。一个字符数据既可以以字符形式输出，也可以以整数形式输出。

（4）字符串常量。

字符串常量是一对双引号括起来的字符序列。C 规定以字符‘\0’作为字符串结束标志。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”，以便系统据此判断字符串是否结束。

合法的字符串常量："How do you do."、"CHINA"、"a"、"$123.45"。

可以输出一个字符串，如：

```
printf("How do you do.");
```

‘a’是字符常量，“a”是字符串常量，二者不同。

注意：不能把一个字符串常量赋值给一个字符变量。

4. 变量赋初值

（1）C 语言允许在定义变量的同时使变量初始化，例如：

```
int a=3;                //指定 a 为整型变量，初值为 3
float f=3.56;           //指定 f 为浮点型变量，初值为 3.56
char c= 'a';            //指定 c 为字符变量，初值为'a'
```

（2）可以给被定义变量的一部分赋初值，例如：

```
int a,b,c=5;            //表示指定 a、b、c 为整型变量，但只对 c 初始化，c 的初值为 5
```

（3）如果对几个变量赋以同一个初值，应写成：

```
int a=3,b=3,c=3;        //表示 a、b、c 的初值都是 3
```

不能写成：

```
int   a=b=c=3;
```

注意：初始化不是在编译阶段完成的，而是在程序运行时执行本函数时赋初值的，相当于有一个赋值语句。

任务的实现

【例 2-2-1】求两数之和。

```
#include <stdio.h>
void main()
{
    int a,b,c,d;            //指定 a、b、c、d 为整型变量
    unsigned u;             /*指定 u 为无符号整型变量*/
    a=13;b=-25;u=12;
    c=a+u;d=b+u;
    printf("a+u=%d,b+u=%d\n",c,d);
}
```

程序运行结果如图 2-2-1 所示。

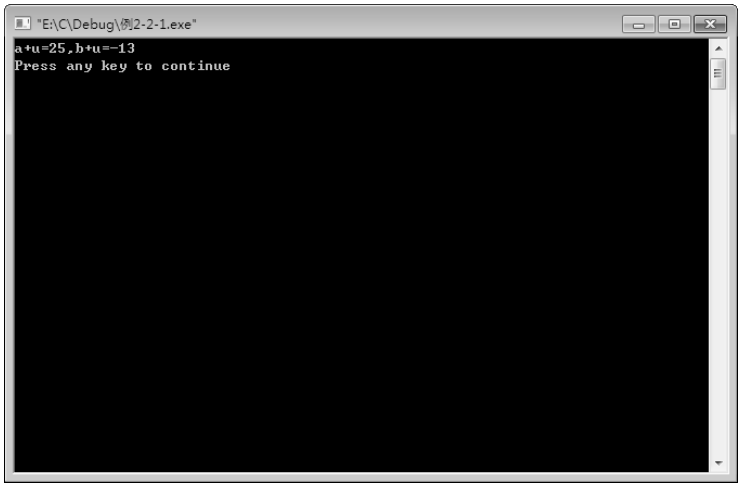

图 2-2-1　例 2-2-1 程序运行结果

【例 2-2-2】浮点型数据的舍入误差。

```
#include <stdio.h>
void main()
{
    float a,b;
    a = 123456.789e5;
    b = a + 20 ;
    printf("%f\n",b);
}
```

程序运行结果如图 2-2-2 所示。

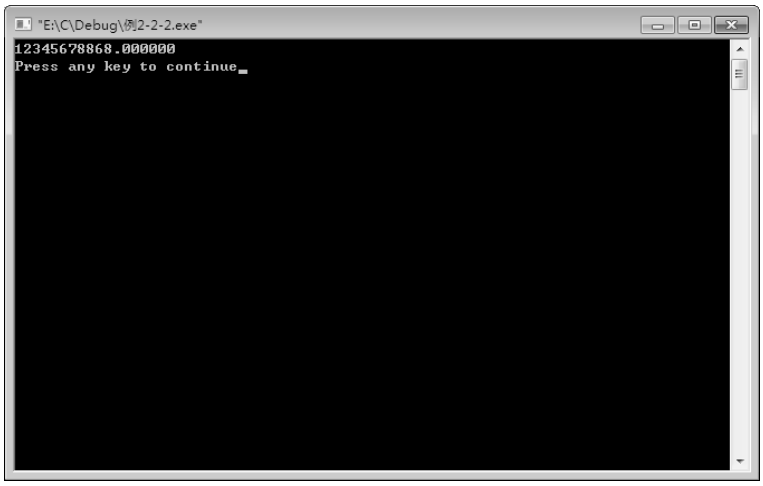

图 2-2-2　例 2-2-2 程序运行结果

【例 2-2-3】转义字符的使用。

```
#include <stdio.h>
void main()
{
    printf(" ab c\t de\rf\tg\n");
    printf("h\ti\b\bj k\n");
}
```

程序运行结果如图 2-2-3 所示。

【例 2-2-4】向字符变量赋整数。

```
#include <stdio.h>
void main()
{
    char c1,c2;
    c1=97;
    c2=98;
    printf("%c %c\n",c1,c2);
    printf("%d %d\n",c1,c2);
}
```

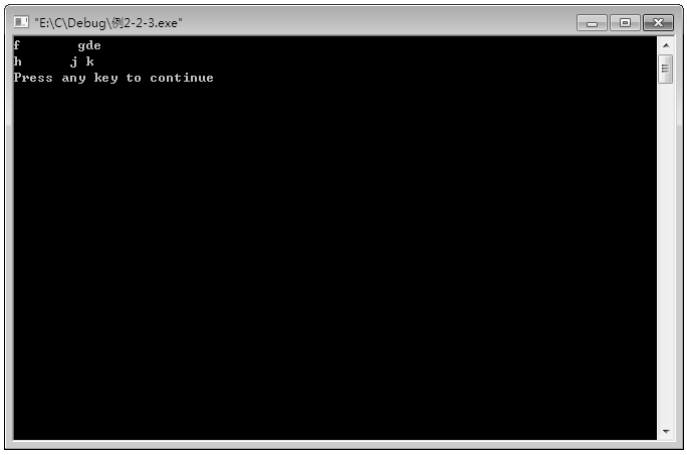

图 2-2-3　例 2-2-3 程序运行结果

程序运行结果如图 2-2-4 所示。

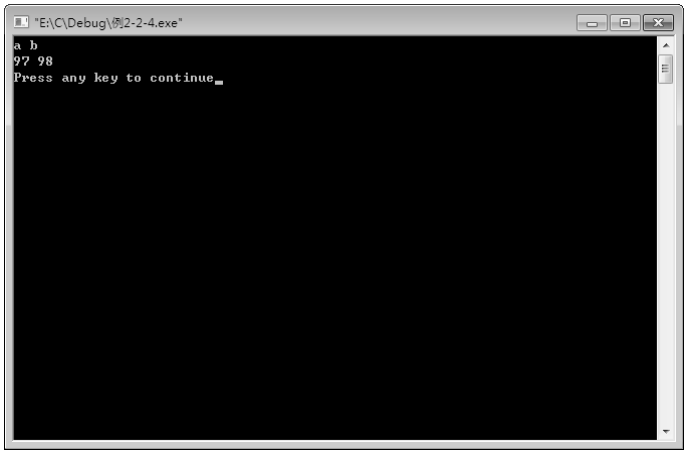

图 2-2-4　例 2-2-4 程序运行结果

【例 2-2-5】大小写字母的转换。

```
#include <stdio.h>
void main()
{
    char c1,c2;
    c1='a';
    c2='b';
    c1=c1-32;
    c2=c2-32;
    printf("c1=%c\nc2=%c\n",c1,c2);
}
```

程序运行结果如图 2-2-5 所示。

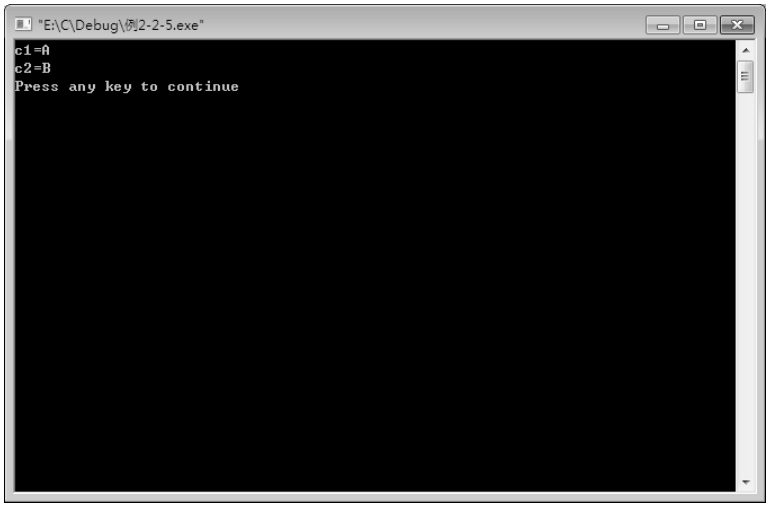

图 2-2-5　例 2-2-5 程序运行结果

说明：有些系统（如 Turbo C）将字符变量定义为 signed char 型，其存储单元中的最高位作为符号位，它的取值范围是-128 ~ 127。如果在字符变量中存放一个 ASCII 码为 0 ~ 127 间的字符，由于字节中最高位为 0，因此用%d 输出字符变量时输出的是一个正整数。如果在字符变量中存放一个 ASCII 码为 128 ~ 255 间的字符，由于在字节中最高位为 1，用%d 格式符输出时就会得到一个负整数。

知识的扩展

1. 编译预处理

编译预处理是在编译源程序之前根据预处理命令对源程序进行的预加工，由编译系统中的预处理程序完成。使用预处理命令可以改进程序设计环境，提高编程效率，使程序易读、易改。

格式：以符号“#”开头，例如：

```
#include<math.h>
```

位置：宏定义与文件包含命令一般放在程序的开头（原则上可以放在程序中的任意位置）。

作用域：从定义起直到其所在源程序的末尾。

2. 文件包含

文件包含是指一个源文件可以将另外一个源文件的全部内容包含进来，其一般形式为：

```
#include  "文件名"
```

或者

```
#include<文件名>
```

功能是把指定文件的内容插入到该#include 命令所在之处。其中，"文件名"是首先在当前目录中寻找文件，如果找不到，再到一系列系统预先设定的目录中去找；而<文件名>则不在当前目录中寻找，而是直接到系统预先设定的目录中去找该文件。

说明：

（1）由于#include 命令常出现在源文件的头部，所以也称被包含进来的文件为“头文件”。

（2）C 编译系统本身也提供有许多这样的头文件，如 stdio.h、string.h、math.h。

（3）一个#include 命令只能指定一个被包含的文件。

任务 3　运算符与表达式

知识与技能：

- 了解 C 语言常用的运算符
- 掌握数据的基本计算方法
- 理解常用数学函数的用法

任务引导

几乎每一个程序都需要进行运算，对数据进行加工处理。完成对程序中不同类型的数据定义之后，可以通过运算符将数据连接组成表达式，实现对数据的计算。C 语言运算符非常丰富，把除了控制语句和输入输出以外的几乎所有的基本操作都作为运算符处理。

知识点介绍

1. C 语言运算符简介

用来表示各种运算的符号称为“运算符”。有些运算符只需要一个运算对象，这种运算符称为“单目运算符”，有的需要两个运算对象，称为“双目运算符”，最多的则需要 3 个运算对象，称为“三目运算符”。

用运算符把运算对象连接在一起所组成的式子称为“表达式”。根据表达式中运算符的不同，在 C 语言里表达式分为算术表达式、赋值表达式、关系表达式、逻辑表达式、条件表达式和逗号表达式等。每种表达式按照运算符所规定的运算规则进行运算，最终都会得到一个结果，它称为表达式的值。

C 语言的运算符有以下几类：

- 算术运算符：+、-、*、/、%。
- 关系运算符：>、<、==、>=、<=、!=。
- 逻辑运算符：!、&&、||。
- 位运算符：<<、>>、~、|、^、&。
- 赋值运算符：=及其扩展赋值运算符。
- 条件运算符：?:。
- 逗号运算符：,。

- 指针运算符：*和&。
- 求字节数运算符：sizeof。
- 强制类型转换运算符：(类型)。
- 分量运算符：.、->。
- 下标运算符：[ ]。
- 其他：如函数调用运算符()。

2. 算术运算符和算术表达式

算术表达式是由算术运算符把数值型运算对象连接在一起构成的表达式，如表 2-3-1 列出了 C 语言中的算术运算符及其含义。

表 2-3-1 基本算术运算符及其含义

运算符	含义	运算对象个数	示例
-	取负	单目	-a
+	取正	单目	+a
*	乘法	双目	a*b
/	除法	双目	a/b
%	整除取余	双目	a%b
+	加法	双目	a+b
-	减法	双目	a-b
++	自加 1	单目	a++ ++a
--	自减 1	单目	a-- --a

除法运算符“/”的运算规则与运算对象的数据类型有关。如果两个运算对象都是整型的，则结果是取商的整数部分，舍去小数，也就是做整除；如果两个运算对象中至少有一个是实型，那么结果就是实型的，也就是一般的除法。

模运算符“%”的两个运算对象必须是整型的，结果是整除后的余数，符号与被除数相同。

自加 1、自减 1 运算符都是单目运算符，其运算对象只能是变量，且变量的数据类型限于整型、字符型、指针型、整型数组元素等。

自加 1、自减 1 运算符实施的具体操作是自动将运算对象实行加 1 或减 1，然后把运算结果回存到运算对象中。

注意：自加 1、自减 1 运算符的与众不同之处在于它们既能出现在运算对象之前，例如++i、--j，称为“前缀运算符”，也就是平常所说的“加加在前、减减在前”；也能出现在运算对象之后，例如 i++、j--，称为“后缀运算符”，也就是平常所说的“加加在后、减减在后”。

前缀式自加 1、自减 1 运算符和后缀式自加 1、自减 1 运算符之间的差别在于实施其增、减操作的时间。前缀式自加 1、自减 1 运算符，是先对运算对象完成加、减 1 和回存操作，然

后才能使用该运算对象的值，也就是先加减后参加运算；后缀式自加 1、自减 1 运算符，是先使用该运算对象的值，然后才去完成加、减 1 和回存操作，也就是先参加运算后加减。

【思考】如何理解-++a、-a++和 a---b。

很明显，对于-++a 应理解成-(++a)。而-a++表达式，因为+、-以及++、--的优先级是相同的，应当遵从自右向左的结合性，它相当于-(a++)；错误的理解是按照自左向右的结合性，它相当于(-a)++，这显然是不正确的，因为“-a”是表达式，而++只能作用于变量，而不能是常量或表达式。当表达式中出现连续多个+或-的情形时，应该从左向右尽可能多地将若干+或-组成一个运算符，因此 a---b 应理解为(a--)-b，而不是 a-(--b)。

3. 赋值运算符和赋值表达式

（1）基本赋值运算符。基本赋值运算符简称“赋值运算符”，它是双目运算符，使用时左边必须是变量，右边是表达式，如表 2-3-2 列出了 C 语言中的赋值运算符及其含义。

表 2-3-2　赋值运算符及其含义

运算符	运算对象个数	含义	示例
=	双目	将表达式的值赋给变量	a=b
+=	双目	加表达式值后赋给变量	a+=b 等价于 a=a+b
-=	双目	减表达式值后赋给变量	a-=b 等价于 a=a-b
*=	双目	乘表达式值后赋给变量	a*=b 等价于 a=a*b
/=	双目	除表达式值后赋给变量	a/=b 等价于 a=a/b
%=	双目	模表达式值后赋给变量	a%=b 等价于 a=a%b

格式为：

变量=表达式;

（2）算术自反赋值运算符。

算术自反赋值运算符的作用是把“运算”和“赋值”两个动作结合起来，成为一个运算符。算术自反赋值运算符都是双目运算符，运算符左边必须是变量，右边是表达式。以“+=”为例（其他见表 2-3-2），其格式为：

变量+=表达式;

含义是先把运算符左边“变量”的当前值与右边“表达式”的值进行“+”运算，然后把结果赋给左边的变量。例如“a+=b”等价于“a=a+b”。

注意：

① 应该把算术自反运算符右边的表达式作为一个整体来对待。如“a*=b+c”等价于“a=a*(b+c)”，而不能把它理解为“a=a*b+c”。

② 运算符“%”要求必须是整型数据，算术自反运算符“%=”也只能用于整型数据，变量的当前值和右边表达式的值都应该是整型的。

4. 关系运算符与关系表达式

所有关系运算符都是双目的。用关系运算符将两个运算对象连接起来所形成的表达式称为“关系表达式”。关系运算符和关系表达式将在第 4 单元详细讲述。

5. 逻辑运算符与逻辑表达式

逻辑运算符中，逻辑非是前缀式的单目运算符，逻辑与、逻辑或是双目运算符。逻辑运算符与逻辑表达式将在第 4 单元详细讲述。

6. 条件运算符与条件表达式

由“?”和“:”两个符号组合成条件运算符，它是 C 语言里唯一的三目运算符，其一般格式为：

表达式 1?表达式 2:表达式 3;

条件运算符与条件表达式将在第 4 单元详细讲述。

7. 逗号运算符与逗号表达式

逗号运算符就是把逗号（,）作为运算符，利用它来把若干个表达式“连接”在一起，这样构成的表达式整体称为“逗号表达式”。逗号表达式的一般格式为：

表达式 1,表达式 2,表达式 3,…,表达式 n

逗号表达式的执行过程是：从左到右顺序计算各个表达式的值，并把最右边表达式的值作为该逗号表达式的最终取值，也就是说“表达式 n”的值是整个逗号表达式的值。

关于逗号表达式，要注意以下几点：

- 逗号表达式是可以嵌套的。
- 程序中使用逗号表达式，通常是分别求逗号表达式内各表达式的值，并不一定要求整个逗号表达式的值。
- 并不是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间隔符。

8. 位运算符

C 语言中的位逻辑运算符，除了“位非”外，都是双目的。由于它们都是按照二进制的相应位一位一位地进行运算，所以称它们是位逻辑运算符。由位逻辑运算符和运算对象构成的表达式称为“位逻辑表达式”。如表 2-3-3 列出了 C 语言中的位逻辑运算符及其含义。

表 2-3-3 C 语言中的位逻辑运算符及其含义

名称	运算符	运算对象个数	含义
位非	~	单目	相应位是 1 为 0，相应位是 0 为 1
位与	&	双目	相应位都是 1 才是 1，否则为 0
位或	|	双目	相应位只要有一个为 1，就是 1
位异或	^	双目	相应位不同为 1，否则为 0

位非运算符“~”为单目运算符，具有右结合性，功能是对参与运算的数的各二进位按位

求反。

位异或运算符“^”是双目运算符，功能是参与运算的两个数各对应的二进位相异或，当两个对应的二进位相异时结果为 1，否则为 0。参与运算的数以补码出现。

位或运算符“|”是双目运算符，功能是参与运算的两个数各对应的二进位相或，只要对应的两个二进位有一个为 1 时结果就为 1，否则为 0。参与运算的两个数均以补码出现。

位与运算符“&”是双目运算符，功能是参与运算的两个数各对应的二进位相与。只有对应的两个二进位均为 1 时结果才为 1，否则为 0。参与运算的数以补码方式出现。

任务的实现

【例 2-3-1】程序代码如下：

```
#include "stdio.h"
main()
{
    float   c=94,english=88,math=75,computer=90,sum,average;
    sum= c+english+math+computer;
    average=sum/4;
    printf("该同学的总成绩为：%f\n",sum);
    printf("该同学的平均成绩为：%f\n",average);
}
```

程序运行结果如图 2-3-1 所示。

图 2-3-1　例 2-3-1 程序运行结果

说明：

（1）printf()的功能是按用户指定的格式把指定的数据显示到终端（一般为显示器屏幕）。

（2）%f 输出的总成绩按小数形式输出，默认 6 位小数。printf 函数的格式符及意义在第 3 单元中有详细阐述。

知识扩展

1. 表达式中数据类型的自动转换

自动转换发生在不同数据类型的量进行混合运算时，该过程由编译系统自动完成。自动转换遵循以下规则：

- 若参与运算量的类型不同，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。例如，int 型和 long 型运算时，先把 int 型转成 long 型后再进行运算。
- 所有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 型后再进行运算。
- char 型和 short 型参与运算时，必须先转换成 int 型。
- 在赋值运算中，赋值运算符两边量的数据类型不同时，赋值运算符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，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一位舍入。

2. 表达式中数据的强制类型转换

强制类型转换是通过类型转换运算来实现的，其格式为：

```
(类型说明符) (表达式)
```

功能是把表达式的运算结果强制转换成类型说明符所表示的类型。例如：

```
(float) x     //把 x 转换为浮点型（也可写作 float x）
(int)(x+y)   //把 x+y 的结果转换为整型
```

在使用强制转换时应注意以下问题：

- 类型说明符和表达式都必须加括号（单个变量可不加）。例如，若(int)(x+y)写成(int)x+y，则变成将 x 转换成 int 型之后再与 y 相加。
- 强制转换或自动转换都只是为了本次运算的需要而对变量的数据长度进行的临时性转换，而不改变数据说明时对该变量定义的类型及其在内存中的存储形式。

本单元小结

本单元主要讲述了 C 语言的常量与变量、数据类型、运算符和表达式等。在 C 语言中，数据类型可分为：基本数据类型、构造数据类型、指针类型、空类型四大类。其中基本数据类型包括整型数据、实型数据和字符型数据。在程序执行过程中，其值不发生改变的量称为常量。变量是指其值可以改变的量。

在学完本单元后，要求学生能够熟练地在程序中应用以上知识点来解决实际问题。

习题 2

选择题

1．以下关于 C 语言的叙述中正确的是（　　）。

A）C 语言中的注释不可以夹在变量名或关键字的中间

B）C 语言中的变量可以在使用之前的任何位置进行定义

C）在 C 语言算术表达式的书写中，运算符两侧的运算数类型必须一致

D）C 语言的数值常量中夹带空格不影响常量值的正确表示

2．以下 C 语言用户标识符中不合法的是（　　）。

A）_1　　B）AaBc　　C）a_b　　D）a--b

3．若有定义：double a=22;int i=0,k=18;，则不符合 C 语言规定的赋值语句是（　　）。

A）a=a++,i++;　　B）i=(a+k)<=(i+k);

C）i=a%11;　　D）i=!a;

4．以下关于 C 语言数据类型使用的叙述中错误的是（　　）。

A）若要准确无误差地表示自然数，应使用整数类型

B）若要保存带有多位小数的数据，应使用双精度类型

C）若要处理如“人员信息”等含有不同类型的相关数据，应自定义结构体类型

D）若只处理“真”和“假”两种逻辑值，应使用逻辑类型

5．以下选项中，能用作用户标识符的是（　　）。

A）void　　B）8_8　　C）_0_　　D）unsigned

填空题

1．C 语言的基本数据类型为_____、_____和_____。

2．假定 x 和 y 为 double 型，则表达式 x=2，y=x+1/2 的值是_____。

3．设 int a=2，则执行完语句 a+=a-=a*a;后，a 的值是_____。

4．执行下面程序中的输出语句后，a 的值是_____。

```
#include "stdio.h"
void main()
{
    int a;
    printf("%d\n",(a=3*2,a*3,a+4));
}
```