# 数据和C

本章中您将学习下列内容:

* 关键字：

int,short,long,unsigned,char,float,double,\_Bool,\_Complex,

\_Imaginary

* 运算符：

sizeof

* 函数：
* scanf()
* C使用的基本数据类型
* 整数类型和浮点数类型的区别。
* 对上述类型，如何书写常量和声明变量。
* 使用printf()和scanf()函数读写各种数据类型的值。

程序离不开数据。将数字、文字和单词输入计算机，目的是希望计算机能够处理这些数据。例如，要计算机计算利息支付或者显示经过排序的葡萄酒商列表。除了数据读取，本章的内容还包括更有趣的对数据的操作练习。

本章研究数据类型中的两大系列：整数类型和浮点数类型。C语言提供属于这两个系列的多种数据类型。本章介绍这些数据类型的名称、如何声明它们、如何以及何时使用它们。您还将发现常量和变量的区别，并且作为奖励，您将很快看到第一个交互式的程序。

## 3.1 示例程序

这里仍以一个示例程序作为开端。正如同前面章节中那样，我们将解释您感到不熟悉的地方。该程序的大致意图应该是很清晰的，因此请试着编译并运行程序清单3.1中的源代码。为节省时间，输入源代码时，可略去注释。

程序清单 3.1 rhodium.c程序

/\* rhodium.c --用金属铑衡量您的体重 \*/

#include <stdio.h>

int main(void)

{

float weight; /\* 用户体重 \*/

float value; /\* 相等重量的铑的价值 \*/

printf("Are you worth your weight in rhodium?\n");

printf("Let’s check it out.\n");

printf("Please enter your weight in pounds: ");

/\* 从用户处获取输入 \*/

scanf("%f",&weight);

/\* 假设铑为每盎司770美元 \*/

/\* 14.5833把常衡制的英磅转为金衡制的盎司 \*/

value = 770 \* weight \* 14.5833;

printf("Your weight in rhodium is worth $%.2f.\n",value);

printf("You are easily worth that! If rhodium prices

drop.\n");

printf("eat more to maintain your value.\n");

return 0;

}

**错误和警告**

如果输入这个程序的过程中出现错误（error），比如少了一个分号，编辑器会给出语法错误消息。即使输入正确，编译器还可能发出像这样的警告(warning): “警告——从double类型转换为float类型时有可能丢失数据。”错误信息表明程序中存在错误，不能对其编译。警告则表明尽管代码正确但有可能不是程序员所要的。警告不终止编译。这里的警告和C语言怎样处理770这样的值有关。本例不必理会此问题，本章稍后将对这个警告消息进行说明。

输入此程序时您可以把770改为贵金属铑的现价，但是不要改动14.5833，这是相当于1英镑的盎司数。盎司金衡制用于衡量贵金属，而英镑常衡制用于衡量人（无论贵贱）的体重。

注意，“enter your weight”的意思是用来通知计算机输入您的体重值，然后按Enter或Return键（不要只是键入体重后就一直等着）。按回车键是用来通知计算机已经完成了键入回答的工作。程序希望您输入一个数字（比如150），而不是单词（比如 too much）。如果键入字母而非数字，程序运行将产生问题，这个问题需要使用if语句解决（请参见第7章“C控制语句：分支和跳转”）。所以这里先请输入数字。下面是程序的示例输出结果：

Are you worth your weight in rhodium?

Let’s check it out.

Please enter your weight in pounds: **150**

Your weight in rhodium is worth $1684371.12.

You are easily worth that! If rhodium prices drop.

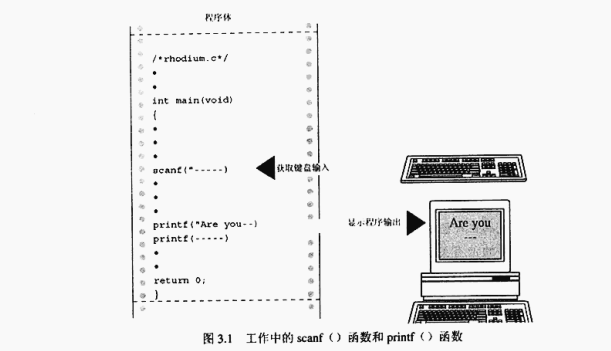
eat more to maintain your value.

**此程序中的新元素**

此程序中包含C语言如下的一些新元素：

* 请注意代码中使用了一种新的变量说明。前面例子中只有整型变量（int）,而本例中还包含了一个浮点变量（float）类型，以便处理更大范围内的数据。float类型可以处理带有小数点的数字。
* 程序还示范了常量的几种新写法，您现在就可以使用带有小数点的数了。
* 要打印这种新的变量类型，请在printf()代码中使用%f说明符来处理浮点值。对%f说明符使用.2修饰词可以精准控制输出格式，使浮点数显示到小数点后两位。
* 使用scanf()函数为程序提供键盘输入。%f指示scanf()从键盘读取一个浮点数，&weight指定将输入值赋于名为weight的变量中。scanf()函数使用&符号指示weight变量的位置。下一章将进一步讨论&符号，现在请相信此处您需要它。
* 也许本程序最突出的新特点是它的交互性。计算机向您询问信息，并使用您输入的数字。与非交互性程序相比较，交互性程序使用起来更为有趣。更重要的是，交互性方法使程序更加灵活。例如，本示例程序可以用于任何合理的体重，而不是150磅。不必每次重写，程序即可针对不同体重进行计算。scanf()和printf()函数使这种交互性成为可能。scanf()函数从键盘读取数据并将其传递给程序，而printf()函数则从程序读取数据并将其传递给屏幕。两个函数一起使用，就可以建立人机之间的双向通信（请参见图3.1），这使计算机的使用更加饶有趣味。

本章解释上述新特性列表中的前两项：各种数据类型的变量和常量。第4章“字符串和格式化输入/输出”将介绍后3项，本章仍将继续使用scanf()和printf()的有限功能。



## 3.2 变量与常量数据

在程序的指示下，计算机可以做很多事，比如数值计算、名字排序、执行语音或视频命令、计算彗星轨道、准备邮寄列表、拨电话号码、画图、做决策以及任何其他可以想象到的事。要完成这些任务，程序需要使用数据，即承载信息的数字与字符。有些数据可以在程序使用之前预先设定并在整个运行过程中没有变化，这称为常量。另外的数据在程序运行过程中可能变化或被赋值，这称为变量。在示例程序中weight是变量，而14.5833则是常量。770.0呢？虽然铑的价格不是一成不变的，但此程序中把它作为常量对待。变量与常量的区别在于，变量的值可以在程序执行过程中变化与指定，而常量则不可以。

## 3.3 数据：数据类型关键字

除了变量和常量的区别，各种数据类型间也有不同。一些数据类型是数字，而另一些则是字母（更广泛的地说是字符）。计算机需要一种方法来区分和使用这些不同的类型。C通过识别一些基本的数据类型做到这一点。如果是常量数据，编译器一般通过其书写来辨认其类型，比如：42是整数，而42.100是浮点数。变量则需要在声明语句中指定其类型。稍后会介绍有关声明变量的详细内容。我们首先了解C语言的基本数据类型。K&R C给出了7个数据类型相关的关键字。C90标准向其中添加了2个关键字，C99标准又添加了另外3个，如表3.1所示。

表3.1 C的数据关键字

|  |  |  |
| --- | --- | --- |
| 原来的K&R关键字 | C90关键字 | C99关键字 |
| int | signed | \_Bool |
| long | void | \_Complex |
| short |  | \_Imaginary |
| unsigned |  |  |

int 关键字提供C使用的基本的整数类型。下面3个关键字（long、short和unsigned）以及ANSI附加的signed