



21 世 纪 计 算 机 科 学 与 技 术 系 列 教 材

C 语言

程序设计

主编 李 健



电子科技大学出版社

21 世纪计算机科学与技术系列教材

C 语言 程 序 设 计

主 编 李 健

副主编 张 杰 周立友 牛朝晖

编 委 (按姓氏笔画为序)

王建良 王 丰 李全军

李秀龙 李秋潭 孙中升

张广辉 张月玲 张玉孔

张 伟 张 华 宋作玲

肖恩忠 陈 衡 苗乃祥

徐加金 高晓峰

电子科技大学出版社

图书在版编目(CIP)数据

C 语言程序设计 / 李健主编. —成都: 电子科技大学出版社, 2006. 6

(21 世纪计算机科学与技术系列教材)

ISBN 7-81114-179-5

I. C… II. 李… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 068605 号

内 容 简 介

本书是作者根据多年的教学经验编写的,本着精选理论、强化实践、突出技能的原则,在内容编排上,结构清晰、新颖,内容前导后续衔接、过渡自然,充分体现了本书易学易用的特点;在文字叙述上,条理清晰、简洁,便于读者阅读。全书包括两篇。第一篇分 12 章,系统阐述了 C 语言的基本概念和语法规则,通过典型例题的分析,着重强调了利用 C 语言进行程序设计的方法,同时,每章后附有大量的习题供读者练习。第二篇结合第一篇中每章的内容,编写了相应的实验指导书,并提供了所有参考程序(均经过本书作者上机调试通过),以利于读者全面地、系统地理解和掌握 C 语言程序设计。

本书可作为高等院校计算机及其他理工类专业的本、专科教材,特别适合用作高职高专教材,也可以供有关的工程技术人员参考。

21 世纪计算机科学与技术系列教材

C 语言程序设计

主编 李 健

出 版	电子科技大学出版社(成都市建设北路二段四号 邮编 610054)
责任编辑	张致强
发 行	新华书店
印 刷	安徽蚌埠广达印务有限公司
开 本	787×1092mm 1/16 印张 18.75 字数 500 千字
版 次	2006 年 6 月第一版
印 次	2006 年 6 月第一次印刷
书 号	ISBN 7-81114-179-5 / TP·44
印 数	1—3000 册
定 价	24.90 元

前 言

C语言是一种短小精悍的计算机高级程序设计语言,它是根据结构化程序设计原则设计并实现的。C语言应用广泛,结构简单、数据类型丰富、表达能力强、使用灵活方便。C语言既有高级语言的优点,又具有低级语言的许多特点。用C语言编写的程序,具有速度快、效率高、代码紧凑、可移植性好的优点。利用C语言,可编制各种系统软件和应用软件。

本书分两篇。第一篇由12章组成。系统阐述了C语言的基本概念和语法规则,通过典型例题的分析,着重强调了利用C语言进行程序设计的方法。第二篇结合第一篇中各章的内容,相应编写了12个实验的指导书,并提供了所有参考程序。

本书是作者根据多年C语言教学经验积累,面向计算机及其他理工类专业的本、专科学生编写的,在编写过程中努力做到结构紧凑、概念准确、叙述流畅、重点突出、例题丰富、实用性强、通俗易懂。本书各章节都引入了大量的实例来说明相关的内容,力求让读者尽快上手。本书在编写上有以下的特点:

1. 精选理论,强化实践,突出技能。

2. 结构清晰、新颖。

3. 在内容的组织上考虑了C语言的特点和读者群的特点,力求用条理、简洁、通俗的语言,将概念表达得很准确、很清楚。

4. 本着循序渐进的原则,对C语言的基本概念和语法规则作了系统讲解。全书的内容前导后续衔接、过渡自然,知识点的组织思路清晰,安排恰当,尽量避免后续知识提前出现给读者造成不必要的障碍。这是作者多年教学经验的总结,也是本书的一大特色。

5. 本书提供了大量典型的例题,通过对典型例题的分析和学习,加深读者对知识的理解和掌握。

6. 将教材和实验指导书合而为一,降低了成本,节约了读者的开支。

7. 本书例题程序和实验参考程序均在Turbo C环境下调试通过。由于篇幅有限,书中的程序只给出了一种参考程序,读者在学习过程中可以举一反三。

8. 本书的例题和实验参考程序书写规范,读者通过学习和效仿,可逐步养成良好的编程习惯。

作者认为,要学好C语言,除了勤于思考外,加强实践才能收到好的效果。在多年的教学实践过程中,我们发现有不少人在学习C语言时,感到入门难,掌握起来更难,对很多问题的理解支离破碎。所以,建议初学者一定要多读、多写程序、多上机调试程序,只有这样才能尽快地掌握和运用C语言去解决实际问题。基于此,我们根据

多年的教学经验和体会,编写了这本教材。

本书由李健主编,张杰、周立友、牛朝晖任副主编。李健编写了第一编第1~6章的主要内容,张杰编写了第一编第7~9章的主要内容,牛朝晖编写了第一编第10~12章的主要内容,周立友编写了第二编的主要内容。李秀龙、张华、张伟参加了第一编第1~6章的编写,王丰、张玉孔参加了第7~9章的编写,王建良、孙中升参加了第10~12章的编写。张广辉、李秋潭参加了第二编内容的编写。张伟、王建良参加了本书大纲的讨论制定,并编写了部分内容。徐加金、肖恩忠、陈衡参加了本书的程序调试。张月玲、苗乃祥、宋作玲、高晓峰、李全军对本书进行了归档整理。全书由李健统稿定稿。

本书的编写和出版得到了许多友人的支持和帮助,在此表示衷心感谢。

由于作者水平有限,书中难免有错误之处,请读者批评指正。作者信箱:wfxylj@163.com。

编 者

2006年6月

目 录

第一编 理论 基础

第 1 章 程序设计基础.....	(3)
1.1 程序设计概述.....	(3)
1.2 C 语言程序设计.....	(7)
习题 1.....	(14)
第 2 章 数据类型、运算符及表达式.....	(16)
2.1 数据类型.....	(16)
2.2 常 量.....	(17)
2.3 变 量.....	(19)
2.4 运算符和表达式.....	(21)
2.5 表达式中数据类型的自动转换.....	(27)
习题 2.....	(29)
第 3 章 简单的 C 程序设计.....	(32)
3.1 程序的逻辑控制结构.....	(32)
3.2 基本的输入输出函数.....	(34)
3.3 顺序结构程序设计举例.....	(40)
习题 3.....	(43)
第 4 章 选择结构程序设计.....	(46)
4.1 if 语句.....	(46)
4.2 switch 语句.....	(55)
4.3 goto 语句.....	(57)
4.4 选择结构应用举例.....	(58)
习题 4.....	(60)
第 5 章 循环结构程序设计.....	(65)
5.1 循环控制语句.....	(65)
5.2 循环体中的控制命令.....	(73)
5.3 循环嵌套.....	(76)
5.4 循环结构程序实例.....	(77)
习题 5.....	(80)
第 6 章 数 组.....	(87)
6.1 一维数组.....	(87)

6.2	字符数组及字符串	(91)
6.3	二维数组	(99)
6.4	数组应用实例	(103)
	习题 6	(107)
第 7 章	函 数	(112)
7.1	函数的定义及使用	(112)
7.2	函数中变量的属性	(119)
7.3	函数的嵌套和递归	(123)
7.4	数组作为函数的参数	(127)
	习题 7	(131)
第 8 章	指 针	(137)
8.1	概述	(137)
8.2	指针变量的定义和使用	(138)
8.3	数组与指针	(141)
8.4	指针作为函数的参数	(148)
8.5	指针函数和指向函数的指针变量	(156)
8.6	动态内存管理函数	(159)
	习题 8	(163)
第 9 章	结构体	(168)
9.1	结构体类型	(168)
9.2	结构体变量	(170)
9.3	结构体数组	(174)
9.4	结构体指针	(178)
9.5	链 表	(180)
	习题九	(193)
第 10 章	共用体、枚举和位运算	(197)
10.1	共用体	(197)
10.2	枚 举	(201)
10.3	位运算	(204)
	习题 10	(209)
第 11 章	编译预处理和数据类型再命名	(212)
11.1	编译预处理	(212)
11.2	数据类型再命名	(221)
	习题 11	(223)
第 12 章	文 件	(226)
12.1	文件概述	(226)
12.2	文件的基本操作	(229)
12.3	文件的数据块读写操作	(233)
12.4	文件的其他操作	(236)
	习题 12	(241)

第二编 实验部分

实验一 C 语言程序设计初步.....	(247)
实验二 数据类型、运算符及表达式.....	(249)
实验三 顺序结构程序设计.....	(252)
实验四 分支结构程序设计.....	(254)
实验五 循环结构程序设计.....	(258)
实验六 数 组.....	(261)
实验七 函 数.....	(264)
实验八 指 针.....	(267)
实验九 结构体.....	(271)
实验十 位运算.....	(275)
实验十一 编译预处理.....	(278)
实验十二 文 件.....	(281)
附 录	
附录 A 常用 ASCII 码字符对照表	(285)
附录 B C 语言的运算符.....	(286)
附录 C Turbo C 2.0 编译错误信息一览表.....	(288)
参考文献.....	(291)

第一编

理 论 基 础

第1章 程序设计基础

【内容提要】

本章内容是程序设计的基础知识，简要介绍了程序设计的基本概念和 C 语言程序设计的入门知识，主要包括计算机程序的概念、算法的概念及其描述方法、程序设计的基本问题、程序错误和测试、C 语言程序的基本结构及特点、C 语言程序的上机实现等内容。

【学习目标】

- ※掌握程序设计的基本概念，包括程序、算法、程序设计以及程序的调试和测试等。
- ※掌握算法设计和描述的基本方法，能对简单的问题设计算法，并用流程图表达出来。
- ※了解 C 语言程序结构的基本特点，能够用 TC2.0 集成环境运行简单的 C 语言程序。

1.1 程序设计概述

1.1.1 计算机语言和程序

计算机语言是计算机能够理解和识别的软件系统，它通过一定的方式向计算机传送操作指令，从而使计算机能够按照人们的意愿进行各种操作处理。计算机能够识别并执行这些指令的前提是，在设计和组织这些指令时必须符合计算机语言的规则。任何一种计算机语言都有一定的使用规则，我们通常称之为语法规则。只有按照特定的语法规则设计的指令，才是有效的指令，计算机才能够执行它。

计算机语言的种类有很多，总体上经过了由低级语言到高级语言的发展过程，目前广泛使用的是计算机高级语言，如 Basic、FoxPro、C++、Java、Delphi 以及本书介绍的 C 语言等。

要学习计算机语言，必须注意学习它的语法规则，就像学汉语要学汉语语法、学英语要学英语语法一样。学习文字语言的目的是为了实现人们之间更好地交流，而学习计算机语言的目的是为了设计计算机程序，使计算机按照人们的意愿去自动处理问题。

所谓计算机程序就是按照计算机语言规则组织起来的一组命令，或者说计算机程序是计算机能够自动执行的一组指令的集合。

1.1.2 算法

算法就是求解问题的方法，是在有限步骤内求解某一问题所使用的一组定义明确的规则，是计算机处理问题所需要的过程。算法的建立通常需要一个逐步求精的过程，先找出解决问题的基本思路，把解决问题的基本过程表达出来，确立粗略的算法框架，然后对框架中的内容进行逐步细化，添加必要的细节，使之成为较为详细的算法描述。

一个算法通常由一系列求解步骤来完成，各操作步骤之间有严格的顺序关系，每一个步骤所规定的操作必须有明确的含义，不能存在二义性。计算机能够在执行有限的步骤后给出正确的结果。

在进行算法设计时，还应该注意：对于同一个问题，可以有不同的解决方法，即一个问题有多种算法。因此，即便使用同一种计算机语言，一个问题也可能有多个不同的计算机程序，认识到这一点，对学习程序设计是非常重要的。

现在，我们从算法的角度对“计算 10 000 以内的所有奇数和”的问题作进一步讨论，并给出它的算法描述。

最直观的理解，计算 10 000 以内的所有奇数和，就是求以下代数式的值：

$$1+3+5+7+\cdots+9\,999$$

假若用 i 表示当前要加的数， i 开始取值为 1，每加一次， i 的值增加 2；用 s 表示已经累加取得的结果，开始取值为 0。那么，对问题求解的过程就是不断地将 i 加到 s 中，直到 i 的值超过 9 999 时，便将累加的结果显示在计算机屏幕上。

下面是包含了执行步骤的算法描述，是用自然语言对算法进行描述的常见形式。

“计算 10 000 以内的所有奇数和”问题的算法：

- 步骤① 为 i 和 s 赋初值，使 $i=1$ 、 $s=0$ ；继续下一步骤。
- 步骤② 判断 i 的值，若 $i < 10\,000$ ，则继续执行下一步骤；否则，转步骤⑥。
- 步骤③ s 加上 i ，继续执行下一步骤。
- 步骤④ i 加上 2，继续执行下一步骤。
- 步骤⑤ 转步骤②。
- 步骤⑥ 显示 s 的值，继续执行下一步骤。
- 步骤⑦ 结束。

按照上述算法给定的 7 个步骤，就能求解“奇数和”问题。若选用一种计算机语言正确描述这个算法，就会得到求解“奇数和”问题的计算机程序，运行该程序，将得到“奇数和”问题的计算结果。

1.1.3 程序流程图

为了使算法描述表达得更清晰，更容易实现程序编写，在进行程序设计时通常使用专门的算法表达工具对算法加以描述，如流程图、N-S 图、PAD 图、伪码等。

流程图是最早使用的一种算法描述工具，它采用不同的几何图形来表示算法的各个步骤，每个几何图形表示不同性质的操作。表 1-1 列出了常用的流程图符号及其功能。

表 1-1 常用的流程图符号及其功能

流程图符号	符号功能
	开始、结束
	处理

续表

流程图符号	符号功能
	判断
	输入、输出
	流程方向

图 1-1 “计算 10 000 以内的所有奇数和”的程序流程图。

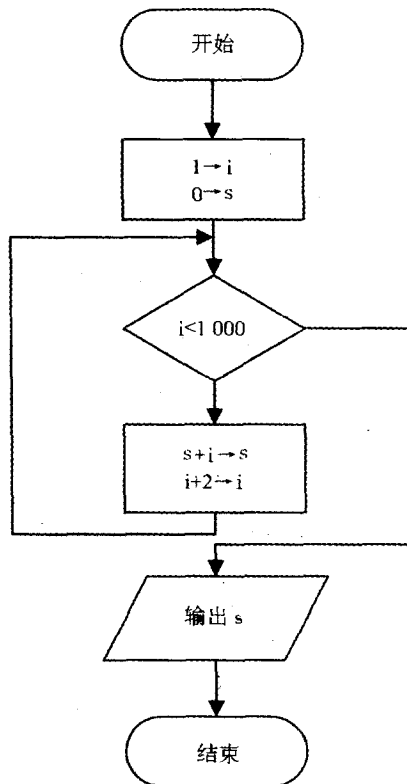


图 1-1 “计算 10 000 以内的所有奇数和”的程序流程图

1.1.4 程序设计

程序设计是用计算机语言实现算法的过程。计算机程序是一段文本代码，编写计算机程序需要在文本编辑环境下进行，高级语言系统一般都提供相应的编程环境，当然也可以使用

像 Windows 记事本程序那样的文本编辑软件。编写程序人员只有对所用程序设计语言深刻理解、熟练掌握、正确运用,才能编写出高质量的程序代码。

编写程序的基本要求是保证语法上的正确性,只有语法正确的程序才能通过编译系统的语法检查。然后是保证逻辑的正确性,这样程序执行后才能得到正确结果。逻辑的正确性对一个复杂的程序并不容易做到,通常需要一个反复测试和编辑修改的过程。

高质量的程序还应体现在以下四个方面:可靠性高、运行速度快、占用存储空间小和易懂性。通常这四个方面不能同时满足,要根据具体情况权衡利弊,兼顾某些方面。在计算机速度越来越快、内存越来越大的今天,程序的易懂性显得更为重要。这是因为一个程序除在计算机上运行外,还要求人能够看懂。只有看懂程序,才能对程序中出现的问题快速地修改,才能根据需要容易地扩充其功能和改善其性能。

要编写容易读懂的程序代码,从一开始就应养成良好的编程习惯,主要体现在以下几个方面:

(1) 合理使用注释。

注释语句是每个程序设计语言都要提供的语句。注释语句对程序的执行结果没有影响,是帮助阅读程序的人理解程序的,它为程序员与程序读者之间建立了重要的信息沟通渠道。在一些正规的程序文本中,注释信息会占用大量的篇幅。

在编写程序时,一般要在程序的开头编写对程序整体说明的注释,在程序模块(如子程序、函数、过程等)前编写解释该模块作用的注释,在较难理解的语句后编写对该语句说明的注释。

(2) 要使用含义鲜明的符号名。

符号名包括函数名、变量名、常量名等。这些名字应能反映它所代表的实际东西,有实际意义,使其能见名知义。例如使用 `total` 表示总量,使用 `average` 表示平均值,使用 `sum` 表示累加和等。

(3) 程序格式化。

尽量使程序布局合理、层次清晰。一个程序如果写得密密麻麻,没有空白,往往是很难看懂的。恰当地利用空格、空行和缩进,可使程序清晰明瞭。

1.1.5 程序的错误和测试

程序的错误通常有两种:语法错误和逻辑错误。程序的语法错误,是指程序编写时因不符合程序语言的语法规则而造成的错误。程序存在语法错误时,程序不能正常运行。程序的逻辑错误,是指程序能够正常运行,但得不到要求的正确结果。程序的语法错误在编译运行阶段语言系统就会指出来,查错纠错比较容易。对于复杂一些的程序,存在逻辑错误时查找起来则比较困难。

要保证程序的正确性或验证程序的正确性是一个关键的、极为困难的问题。目前,比较实用的验证程序的方法是采用测试的方法。但是,测试只能证明程序有错,而无法确保程序完全正确,也许通过一系列的测试,程序中还包含未发现的错误。

测试是假设程序中存在错误,通过运行程序来尽可能地发现错误。在测试时,输入一组预先设计好的数据,检查运行后是否得到正确的结果。这组输入的数据称为测试用例。如何设计测试用例是测试的关键。目前常用的测试方法有黑盒法和白盒法。

黑盒法把程序看成一个黑盒子，只测试程序是否满足它的功能，不考虑程序的内部逻辑和特性。因此，要发现程序中的错误，需要运用穷举法输入每一种数据进行测试。但事实上要测试每一种数据往往是不可能做到的，因此在程序测试领域出现了众多测试技术，如等价分类法、边值分析法、错误推测法和因果分析法等。

白盒法又称为逻辑覆盖法。使用白盒法需要了解程序内部的详细情况，并在此基础上设计测试用例。测试时，程序中的每一条语句至少要执行一次，最彻底的是覆盖程序中的每一条路径。常用的覆盖标准有：语句覆盖、判定覆盖、条件覆盖、判定 / 条件覆盖和条件组合覆盖等。

通过测试发现错误后，要对源程序进行检查，找出错误的位置，并予以改正。这种去除程序错误的过程称为调试。

目前，大多数开发工具都提供了跟踪调试工具，一般都提供了设置断点、单步执行、查看变量或表达式值等功能，熟练使用这些调试工具可以提高调试效率。

1.2 C 语言程序设计

1.2.1 C 语言的诞生和发展

C 语言 1972 年由美国的 Dennis Ritchie 设计发明的，并首次在 Unix 操作系统的 DEC PDP-11 计算机上使用。它由早期的编程语言 BCPL (Basic Combined Programming Language) 发展演变而来。在 1970 年，AT&T 贝尔实验室的 Ken Thompson 根据 BCPL 语言设计出较先进的并取名为 B 的语言，最后导致了 C 语言的问世。1983 年，美国国家标准化协会 (ANSI) 根据 C 语言问世以来各种版本对 C 的发展和扩充制定了 C 的标准，称为 ANSI C。1987 年 ANSI 又公布了新的标准——87 ANSI C。目前流行的 C 编译系统都是以它为基础的。

在 C 的基础上，1983 年又由贝尔实验室的 Bjarne Stroustrup 推出了 C++。C++ 进一步扩充和完善了 C 语言，成为一种面向对象的程序设计语言。C++ 提出了一些更为深入的概念，它所支持的这些面向对象的概念容易将问题空间直接地映射到程序空间，为程序员提供了一种与传统的结构化程序设计不同的思维方式和编程方法，因而也增加了整个语言的复杂性，掌握起来有一定的难度。

1.2.2 C 语言的特点

C 语言具有以下特点：

(1) C 语言是一种结构化语言，它层次清晰，便于按模块化方式组织程序，易于调试和维护。

(2) C 语言的表现能力和处理能力极强，它不仅具有丰富的运算符和数据类型，便于实现各类复杂的数据结构，还可以直接访问内存的物理地址。

(3) 由于 C 语言实现了对硬件的编程操作，因此 C 语言集高级语言和低级语言的功能于一体，既可用于系统软件的开发，也适合于应用软件的开发。

(4) C语言还具有效率高、可移植性强等特点,因此广泛地移植到了各种类型的计算机上,从而形成了多种版本的C语言。

(5) C语言可以直接操纵硬件,并且生成的目标代码质量高。

1.2.3 C程序的概念

本节通过几个小程序引出了C语言程序设计的基本概念,使读者对C语言程序和程序设计有一个初步认识。

例 1-1 一个加法程序。

```
/* 加法程序 */
main()
{
    int a,b;
    a=8;
    b=2000;
    printf("%d\n",a+b);
}
```

这个C程序含以下概念:

程序行、主函数、数据类型、变量、赋值、表达式、系统函数、输出、输出格式、函数体、注释。

这个程序共由8行组成,每一行称为一个程序行;

第1行是程序的注释,是一些关于程序的说明性信息;

第2行的main()称为主函数;

第4、5、6、7行的a、b是程序中的变量;

第4行的int限定变量a、b只能代表整数,即a、b的数据类型为整型;

第5、6行实现了赋值功能,分别使a、b具有8和2000的值;

第7行的功能是将表达式a+b的值输出在屏幕上,其中printf()是实现输出的一个系统函数,“%d\n”是输出数据的格式控制信息;

花括号对({})之间的所有信息构成了main()函数的函数体。

从上面的叙述可以知道,这个加法程序的功能是非常单一的,它并不能对任意的两个数进行加法运算,即它不是一个通用的加法程序。

例 1-2 最简单的C语言程序。

```
main()
{
    printf("Hello,world!\n");
}
```

这是一个最简单的C语言程序,函数体只有一个输出函数语句,用于输出一个特定的文本信息。该程序执行结果是在屏幕显示以下字符串:

Hello,world!

总结上述程序,不难发现它们具有一个共同的特点,即每个程序都是由main()函数构成

的，由于 main() 函数的函数体内容不同，程序也就各自具有了不同的功能。事实上，任何一个 C 语言程序，main() 函数都是不可缺少的。main() 函数的一般结构如下：

```
main()
{
    函数体语句
}
```

函数体通常分为两部分，前半部分一般是一些说明语句，用于对变量等进行必要的定义说明，也称为定义数据结构；后半部分是一些执行语句，完成具体的操作。从操作的角度来认识，程序的功能是由函数体的可执行语句完成的，也就是说，函数的可执行部分实现了问题的“算法”。由此，对“程序”的概念可以理解为：

程序 = 数据结构 + 算法

下面是一个稍微复杂一些的例子，它告诉读者，一个 C 语言程序，在结构上不仅包括 main() 函数，还可以包括其他独立的函数，一个函数可以在另一个函数中被使用。

例 1-3 求最大数程序。

以下程序实现了求两个数中最大数的功能，当程序执行时，用户从键盘输入任意两个整数，计算机就会将其中较大的数显示在屏幕上。

```
main()                                /* 主函数 */
{
    int x,y,large;                    /* 定义变量 */
    scanf("%d,%d",&x,&y);             /* 从键盘输入两个数 */
    large=max(x,y);                  /* 利用 max() 函数找出 x、y 的最大数并赋给 large */
    printf("The Max number is %d\n",large); /* 输出 */
}
/* 以下是一个自定义函数 */
int max(int x,int y)                 /* 求 x、y 中最大值的函数，它是独立于 main() 函数的 */
{
    int z;
    if (x>y)
        z=x;                        /* 如果 x 大于 y，就将 x 的值赋给 z */
    else
        z=y;                        /* 否则就将 y 的值赋给 z */
    return(z);
}
```

这个程序包括了两个函数：一个是主函数 main()；另一个是 max() 函数，其功能是求出两个整数 x、y 中最大的一个数，这个函数是独立定义的，它在 main() 函数中被使用。在执行时，程序从 main() 函数开始，先由 scanf() 函数从键盘读取两个数字，这时需要由用户从键盘上输入两个数（如 3、5）。此时 x 被赋值 3，y 被赋值 5。然后执行第 6 行，将 x 和 y 的值传入 max() 函数中。在 max() 函数中经过判断后，z 中的值就是两个数的最大值。用 return 语句将 z 的值成为函数的返回值，此时程序又回到第 6 行，将 max() 函数的返回值赋值给变量 large。第 7 行将变量 large 的值输出到屏幕上。