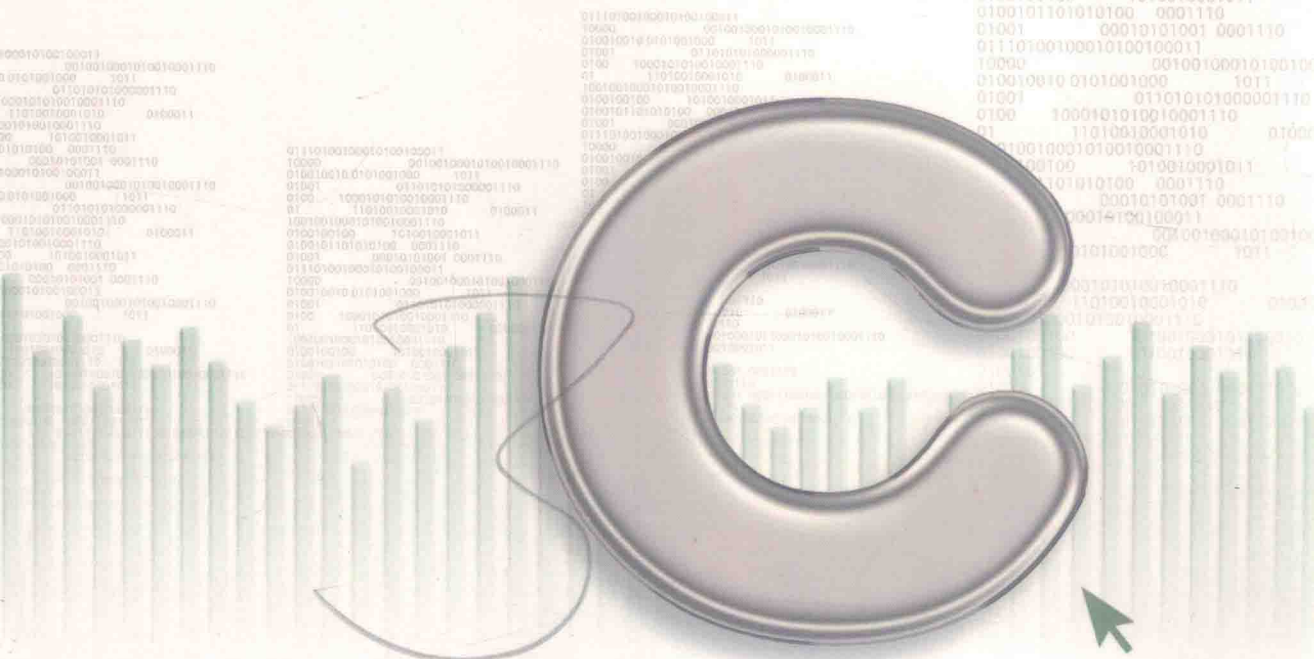




高等教育应用型人才十三五规划教材
高等教育应用型人才 计算机类专业 规划教材



C语言程序设计 案例教程

◎ 胡则辉 寿周翔 李 静 主编

本书提供免费教学资源，包括电子教案、试题库和案例源程序文件



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

高等教育应用型人才十三五规划教材

高等教育应用型人才计算机类专业规划教材

C 语言程序设计案例教程

胡则辉 寿周翔 李 静 主 编

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书理论与实践紧密结合,注重知识的系统性。全书共9章,内容包括:概述、数据类型、选择结构、循环结构、数组、函数、指针、结构体和共用体,以及文件等。本书通过大量精选的典型示例详细地阐述编程思想、语法应用、程序调试,激发读者的学习兴趣,加快读者对编程知识的理解,加深读者对相关知识间相互关系的理解,打造读者编程的基石。本书力求化繁为简、学以致用,重在培养读者的编程思想和编程能力,对教学中的重点、难点通过案例以分步迭代的方式依次递进、分层剖析,以使读者能快速抓住重点、难点,掌握编程中“必会”的知识点,而不至于迷失在语法的海洋中。

本书不仅适合作为应用型本科院校C语言课程的教材,而且适合作为程序设计爱好者的参考用书。

本书提供免费教学资源,包括电子教案、试题库和案例源程序文件。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

C语言程序设计案例教程 / 胡则辉, 寿周翔, 李静主编. —北京: 电子工业出版社, 2018.7

ISBN 978-7-121-34409-1

I. ①C… II. ①胡…②寿…③李… III. ①C语言-程序设计-高等学校-教材 IV. ①TP312.8

中国版本图书馆CIP数据核字(2018)第120635号

策划编辑: 李 静

责任编辑: 朱怀永 文字编辑: 李 静

印 刷: 北京虎彩文化传播有限公司

装 订: 北京虎彩文化传播有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路173信箱 邮编 100036

开 本: 787×1092 1/16 印张: 13 字数: 283 千字

版 次: 2018年7月第1版

印 次: 2018年7月第1次印刷

定 价: 38.00元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: (010) 88254604, lijing@phei.com.cn

前言



人人都应该学习一门计算机语言，因为它将教会你如何思考。

——苹果公司联合创始人 乔布斯

未来社会，随着信息技术和人工智能的发展，程序设计已经渗入各行各业，因此，学习编程并不是程序员的“专利”。

C 语言是面向过程的结构化、模块化的主程序程序设计语言，它具有功能丰富、表达能力强、使用灵活方便、应用面广等特点，“C 语言程序设计”是本、专科院校普遍开设的一门专业基础课。

编者在近 20 年的程序设计教学中发现，初学者在程序设计学习过程中遇到的最大困难是容易迷失在语法的海洋中，以及较难建立编程思维。语法是编程的基础，掌握语法是能够灵活组织程序实现逻辑功能的前提，而编程思维是逻辑思维、抽象思维和计算思维的综合，需要通过复杂的编程训练逐步建立起来。作者在程序设计课堂教学中采用基于案例的分步迭代教学法进行教学，很好地解决了以上两个问题，本书的编写也采用基于案例的分步迭代教学思想。

基于案例的分步迭代教学法是指教师通过案例引导学生掌握基础知识，并架起学习新知识的支点，然后运用知识迁移、协作讨论等方式培养学生的思维模式，从而达到培养学生解决实际问题的能力。分步迭代教学方法按照人们认识实践的规律：学习阶段→模仿阶段→独立应用阶段，把该规律应用到程序设计教学过程中，可以训练学生逐步掌握程序语言编码调试能力，以及运用程序语言进行项目开发的能力。

本书在内容上由浅入深，由简单到复杂，循序渐进，合理分散重点和难点，兼顾够用和全面，将部分平常编程中较少用到或者比较冷僻的语法放入知识拓展中，以便学生在开始学习时重点掌握那些必会的语法知识；在每章的重点或难点语法讲解中，加入了“练一练”和“想一想”两个小环节，使学生进一步巩固理解相应的知识点；通过案例将 C 语言的理论与实践应用紧密结合，让学生能够快速学以致用，在案例讲解中通过问题分析逐步引导学生如何

分析、解决问题，引导学生建立编程思维。

另外，编者也在教学实践中总结了初学者学好编程的 7 个原则。

1. 从实践中学习。编程应该注重实践，而不仅仅靠看书和学习理论。

2. 从案例中学习。编程应关注问题和解决问题的方法及程序，而不是将重点放在程序语言的语法上。

3. 从错误中学习。学习编程应该拥抱错误而不是害怕它们，因为你每发现和改正一个错误，你的编程水平可能就因此而获得提升。

4. 从互联网上学习。IT 技术是日新月异的，书本上的知识容易过时，而优秀的程序员都喜欢在网上提问和分享，在互联网上总能学到当下流行的技术，也经常能够很容易地获得问题的答案。

5. 向同学学习。在同一班级学习的同学往往面临同样的问题，一起讨论交流，是一个高效的相互学习的方法。

6. 持续写代码。编程不像游泳或骑单车一样是习得性技能，习得性技能一旦学会就不会忘掉，但编程的技能需要持续练习，否则技能是会退化的。

7. 经常提问。会提问的人是会思考的。学习编程的过程中不可避免地会碰到各种一时解决不了的问题，经常提问并记录，哪怕你暂时找不到答案。

本书由胡则辉、寿周翔、李静主编，其中，第 1、2、8、9 章由胡则辉编写，第 3、4 章及附录由寿周翔编写，第 5、6、7 章由李静编写。

本书提供免费教学资源，包括电子教案、试题库和案例源程序文件。

本书的出版得到电子工业出版社的支持和帮助，在此表示衷心感谢。由于编者水平有限，书中疏漏与不妥之处在所难免，恳请同行和读者批评指正。

编者

目录

第 1 章 概 述 1

- 1.1 计算机程序 1
- 1.2 计算机语言 1
- 1.3 C 语言的发展及其特点 4
- 1.4 C 程序格式与结构特点 5
- 小练习 8

第 2 章 数据类型 10

- 2.1 C 语言数据类型 10
- 2.2 常量 11
- 2.3 变量 14
 - 2.3.1 整型变量 15
 - 2.3.2 实型变量 21
 - 2.3.3 字符变量 23
- 2.4 输入与输出函数 printf() 和 scanf() 26
 - 2.4.1 用 printf() 函数实现输出 26
 - 2.4.2 用 scanf() 函数实现输入 28
- 2.5 字符输入与输出函数 31
- 小练习 37

第 3 章 选择结构 39

- 3.1 应用场景 39

3.2 关系运算符和关系表达式	39
3.3 逻辑运算符和逻辑表达式	41
3.4 if 语句实现选择结构	43
3.4.1 if 语句的常用形式	44
3.4.2 if 语句的嵌套	49
3.5 switch 语句实现多分支选择结构	51
编程练习	53

第 4 章 循环结构 57

4.1 应用场景	57
4.2 用 for 语句实现循环	58
4.3 用 while 语句实现循环	62
4.4 用 do-while 语句实现循环	64
4.5 循环嵌套	65
4.6 流程转移控制	67
4.7 循环应用综合举例	69
编程练习	74

第 5 章 数组 79

5.1 应用场景	79
5.2 一维数组	80
5.3 二维数组	90
5.4 字符数组	93
编程练习	99

第 6 章 函数 102

6.1 应用场景	102
6.2 函数的定义	103
6.3 函数的调用和声明	105
6.4 变量的作用域和生存期	112
编程练习	115

第 7 章 指针 117

7.1 应用场景	117
7.2 指针和指针变量	118
7.3 指针变量的定义和引用	118
7.3.1 指针变量的定义、初始化及赋值	119
7.3.2 指针变量的引用	128
7.4 指针变量的指针运算符及运算	135
编程练习	137

第 8 章 结构体和共用体 139

8.1 结构体	139
8.1.1 结构体的定义、引用和初始化	141
8.1.2 结构体数组与结构体指针	147
8.1.3 使用指针处理链表	154
8.2 共用体（选学）	156
8.3 枚举类型（选学）	158
8.4 用 typedef 自定义类型（选学）	159
编程练习	160

第 9 章 文件 162

9.1 C 语言文件的相关知识	162
9.2 文件的打开与关闭	164
9.3 顺序读写文件	166
9.3.1 读写字符	166
9.3.2 读写字符串	170
9.3.3 格式化读写	171
9.3.4 二进制方式读写	173
9.3.5 随机读写文件	175
编程练习	177

附录 A ASCII 码表 178

附录 B	C 语言中的关键字	180
附录 C	C 语言运算符和优先级	181
附录 D	C 语言库函数	183
附录 E	C 语言错误的中英文对照信息	187
附录 F	在 Visual C++ 6.0 环境下运行 C 程序的方法	193
参考文献		199

第 1 章 概 述

学习目标

1. 理解指令、程序、软件的概念及其相互之间的关系。
2. 掌握 C 语言程序的基本结构。

1.1 计算机程序

1945 年，冯·诺依曼首先提出“存储程序”的概念和二进制原理，计算机运行过程中，把要执行的程序（一组计算机能识别和执行的，用来完成一定功能的指令序列）和处理的数据首先存入主存储器（内存），计算机执行程序时，将自动地按顺序从主存储器中取出指令（可以被计算机理解并执行的基本操作命令）逐条执行。这一基本原理一直使用至今。

计算机的操作都是由程序控制的，为了让计算机完成指定操作，必须编写各种计算机程序。计算机程序开发是周而复始的，需要经历：问题分析、设计算法、编写程序、测试、运行结果分析、修改调整等一系列过程，专业从事这种工作的人员叫作程序员。

1.2 计算机语言

人与人之间使用语言进行交流，人和计算机交流信息并控制计算机完成指定的操作就需要使用计算机语言。计算机每执行的一个动作、一个步骤，都是按照计算机语言编写的程序执行的，而程序都是使用编程语言编写的。所以人们要控制计算机，一定要学会使用计算机语言。计算机程序设计语言的发展，经历了机器语言、汇编语言到高级语言的三个阶段。

1. 机器语言

目前计算机都使用二进制原理，所以计算机能直接识别和执行的是由 0 和 1 组成的二进

制机器指令（计算机能直接识别和执行的二进制代码）。与计算机交流的最原始语言是机器语言（机器指令的集合）。由于机器语言直接使用二进制，所以机器语言具有灵活、直接执行和速度快等优点，但难学、难记、难检查、难修改、难以推广使用、可移植性差（软件可移植性是指一种计算机上的软件转移到其他计算机上并正确运行的能力）。机器语言依赖于具体的机器，不同型号的计算机的机器语言是不相通的，按照一种计算机的机器指令编写的程序，往往无法在另一种计算机上直接执行，难以移植。用机器语言编写程序，编程人员要首先熟记所用计算机的全部指令代码和代码的含义。编写程序时，程序员得自己处理每条指令和每个数据的存储分配和输入输出，还得记住编程过程中每步所使用的工作单元处在何种状态，这是一件十分烦琐且极易出错的工作。目前，除了计算机生产厂家的专业人员外，绝大多数的程序员已经不再学习机器语言。例如，

0000, 0000, 000000010000 代表 LOAD A, 16

其中，0000 代表加载（LOAD），0000 代表暂存器 A，000000010000 代表地址为 16 的存储器。

2. 汇编语言

为了降低编程难度，克服机器语言的缺点，人们设计汇编语言（用助记符代替机器指令的操作码，用地址符号代替指令或操作数的地址）。汇编语言比机器语言简单好记，但在不同的设备中，汇编语言对应不同的机器语言指令集，也依赖于具体机器，编写的程序还是难以移植，导致汇编语言仍然难以普及。汇编语言编写的程序需通过汇编程序（把汇编语言编写的程序转换为机器指令）才能被计算机执行。

机器语言和汇编语言都是完全依赖于具体机器的，也称为低级语言。机器语言用二进制指令代码编写程序，而汇编语言（也称为符号语言）用指令助记符来编写程序。

3. 高级语言

计算机语言具有高级语言和低级语言之分，高级语言主要相对于机器语言和汇编语言，它是较接近自然语言和数学公式的编程，基本脱离了机器的硬件系统，具有良好的可移植性。它是以人类的日常语言为基础的一种编程语言，使用一般人易于接受的文字及表达习惯，从而编写程序更容易，具有较高的可读性。为了提高编程效率，克服低级语言的不足，20 世纪 50 年代人们创造了第一个计算机高级语言（FORTRAN 语言）。

高级语言并不是某一种具体的语言，而是包括很多编程语言，如流行的 Java，C++，C#，Pascal，Python，Lisp，Prolog，FoxPro，C 语言等，这些语言的语法、命令格式都不相同。

计算机无法直接识别高级语言编写的程序，必须用编译程序将源程序（用高级语言编写的程序称为源程序）转换为目标程序（机器指令的程序），计算机执行目标程序，得到最终结果。使用高级语言编程效率较高，经程序编译后高级语言的一个语句对应多条机器指令。由

C语言构成的指令序列称为C源程序，源程序文件的后缀名为“.c”。源程序经过C编译程序编译生成后缀为“.obj”的目标文件（二进制文件），然后由称为“连接程序”（link）的软件，把目标文件与C语言提供的各种库函数连接起来，生成后缀为“.exe”的可执行文件。在编写大型程序时，由于汇编语言难用难懂，从而，需要一个更加符合人类思维逻辑框架的编程语言，就像C，C++等一系列的语言，他们的可操作性非常强，适合编写一些大型程序，同样，它也需要“编译器”把他们编译成机器语言才能够被机器识别。



想一想

用机器语言、汇编语言、高级语言编写的程序哪种可移植性最好？



知识扩展

请通过网络或图书馆查阅资料，初步了解各种高级语言的特点并进行比较。



小练习

- 在下列关于计算机语言的说法中，不正确的是（ ）。
 - Pascal 和 C 都是编译执行的高级语言
 - 高级语言程序比汇编语言程序更容易从一种计算机移植到另一种计算机上
 - C++是历史上的第一种支持面向对象的计算机语言
 - 与汇编语言相比，高级语言程序更容易阅读
- 汇编语言（ ）。（多选）
 - 是一种与具体硬件无关的程序设计语言
 - 在编写复杂程序时，相对于高级语言而言代码量较大，且不易调试
 - 可以直接访问寄存器、内存单元、I/O 端口
 - 随着高级语言的诞生，如今已完全被淘汰，不再使用
- 下列关于高级语言的说法正确的有（ ）。（多选）
 - Ada 是历史上第一种高级语言
 - Pascal 和 C 都是编译执行的高级语言
 - C++是历史上的第一种支持面向对象的语言
 - 编译器将高级语言程序转变为目标代码
 - 高级语言程序比汇编语言程序更容易从一种计算机移植到另一种计算机上

参考答案：1. C 2. BC 3. BDE

1.3 C 语言的发展及其特点

为了描述和实现 UNIX 操作系统, 1972~1973 年, 贝尔实验室的 D.M.Ritchie 博士在 B 语言基础上设计了 C 语言。C 语言是当前国际上流行的计算机高级语言。C 语言具有以下特点。

1. 简捷紧凑、灵活方便

C 语言一共只有 32 个关键字, 9 种控制语句, 程序书写自由, 编程效率高(实现相同功能时比其他高级语言编写的源程序更短)。它将高级语言的基本结构和语句与低级语言的实用性相结合。

2. 运算符丰富

C 语言的运算符包含范围很广泛, 共有 34 种运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理, 从而使 C 语言的运算类型极其丰富, 表达式类型多样化, 灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

3. 数据结构丰富

C 语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。C 语言拥有用来实现各种复杂的数据类型的运算, 并引入指针概念, 程序效率更高。另外, C 语言具有强大的图形功能, 支持多种显示器和驱动器, 且计算功能、逻辑判断功能强大。

4. C 语言是结构化语言

结构化编程语言的显著特点是代码及数据的分隔化, 即程序的各个部分除了必要的信息交流外彼此独立。结构化程序分为顺序结构、选择结构和循环结构三种形式。这种结构化方式可使程序层次清晰, 便于使用、维护及调试。C 语言是以函数形式提供给用户的, 这些函数可方便地调用, 并具有多种循环、条件语句控制程序流程, 从而使程序完全结构化。

5. 语法限制不太严格、程序设计自由度大

一般的高级语言语法检查比较严格, 能够查出绝大多数语法错误。而 C 语言允许程序编写者有较大的自由度, 但必须严格检查程序, 否则容易出错。

6. 允许直接访问物理地址, 可以直接对硬件进行操作

C 语言既具有高级语言的功能, 又具有低级语言的许多功能, 能够像汇编语言一样对位、字节和地址进行操作, 而这三者是计算机最基本的工作单元, 可以用来写系统软件。

7. C 语言程序生成代码质量高，程序执行效率高

一般只比汇编程序生成的目标代码效率低 10%左右。

8. C 语言适用范围大，可移植性好

C 语言的一个突出优点就是适用多种操作系统，如 DOS、UNIX，也适用于多种机型。适用于不同平台，对于一台计算机编写的 C 程序只需做较少修改或不做修改，就能在另一台计算机上轻松地运行。

对操作系统，以及需要对硬件进行操作的场合，C 语言明显优于其他高级语言，许多大型应用软件都是用 C 语言编写的。C 语言具有绘图能力强，可移植性，并具备很强的数据处理能力，因此适于编写系统软件，三维、二维图形和动画，它是数值计算的高级语言。

与其他高级语言比较。

BASIC：初学者的入门语言。

FORTRAN：用于科学计算。

COBOL：用于银行、商业管理（历史原因：由美国海军开发，用于美国银行系统）。

PASCAL：世界上第一种结构化语言，严谨，是较理想的教学语言；但过于刻板。

C：称为 PASCAL 的姊妹语言，这是从结构化方面来看的。但 C 语言比较灵活，自由度高，适于编写各种软件。



想一想

现实生活中哪些问题编程实现时需要使用顺序结构、选择结构、循环结构？



知识扩展

通过查阅资料详细了解各用人单位对 C 语言的要求，思考嵌入式开发为什么偏爱选择 C 语言作为编程语言？

1.4 C 程序格式与结构特点

【例 1-1】



问题描述

在屏幕上输出以下一行信息：

Hangzhou welcome you!



问题分析

C 语言为我们提供了 `printf` 函数：按照一定格式输出信息。



程序实现

```
/*以下程序用于输出一行以下信息：
Hangzhou welcome you!*/
#include<stdio.h>//也可以写成#include"stdio.h"
int main ()
{
    printf ("Hangzhou welcome you!\n ");
    return 0; //返回 0，也可以写成 return (0);
}
```

例 1-1 程序运行结果如图 1-1 所示。

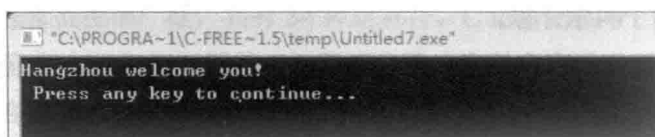


图 1-1 例 1-1 程序运行结果

说明：

(1) `/*.....*/`和`//`都是用来注释的，`/*.....*/`用来注释一段，而`//`用来注释一行，注释部分在程序运行时不会被执行。合理的注释可以提高程序的可读性。

(2) `#include<stdio.h>`（也可以写成`#include"stdio.h"`），预处理，由于下面用到系统函数 `printf`（用于输出结果），而这一系统函数在头文件 `stdio.h` 中定义，所以需要包含。

(3) `int main()`：前面的 `int` 表示主函数最后返回的结果是整型数值。`main()` 是主函数。关于主函数注意以下内容。

① 每个 C 语言程序都必须有且只有一个主函数（`main` 函数）。

② `main` 函数后的“`()`”，中间的参数部分可以是空的，但 `()` 不能省略。

③ 程序总是从 `main` 函数开始执行（`main` 函数可以放在程序的任意位置），到程序执行完后，从主函数结束执行，不论 `main` 函数在程序中的位置如何。

(4) 函数体用大括号 `{}` 括住。

(5) 其中，`printf` 称为输出函数，双引号内的字符串原样输出，“`\n`”是换行符（一些非显示字符，难以用一般形式表示，故使用这种特殊形式表示，称为“转义字符”）。

(6) 本程序只有一个语句，语句必须用分号结束（每个语句必定有分号）。

(7) 编写 C 程序时除汉字以外其他所有字符都在英文状态下输入。

【例 1-2】



问题描述

输入两个整数，输出其中最大的一个整数。



问题分析

可以定义一个变量，然后将输入的两个整数进行比较，将其中一个大的值赋给这个变量。程序为：if a>b c=a else c=b 或者 if a>b return (a) else return (b)。



程序实现

```
#include <stdio.h>
int max (int x, int y);
int main ()
{
    int a, A, m;
    printf ("请输入两个整数 (中间用, ""隔开, 输入完成后按回车键): \n");
    scanf ("%d, %d", &a, &A); m=max (a, A);
    printf ("%d 和%d 相比最大的是: %d\n", \
        a, A, m);
    return 0;
}
int max (int x, int y)
{
    int z;
    z=x>y?x: y;
    return (z); }
```

例 1-2 程序运行结果如图 1-2 所示。

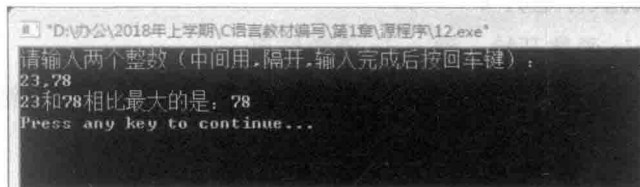


图 1-2 例 1-2 程序运行结果

说明:

(1) int max (int x, int y); 自定义函数声明，注意后面的；不要漏掉。若自定义函数的现代码在被调用函数前则不需要声明，若在其后则必须声明。C 语言的基本组成部分是函数，分为系统函数（一般需要预处理，要注意用到的系统函数在哪个头文件里）和自定义函数。

(2) `int a, A, m;` 变量定义, 注意 C 语言区分大小写。大写的 `A` 与小写的 `a` 表示两个不同的变量。

(3) `printf` 称为格式输出函数, 它的参数有两部分, 用双括号括住部分称为“格式控制字符串”, 另一部分称为“输出项”。而“格式控制字符串”又分为两部分: 格式说明与普通字符。由 `%` 开始, 以格式字符结束。用于表示对应的输出项的数据类型和格式。

(4) `scanf` 称为格式输入函数, 用于从键盘输入变量的值, 它的参数由两部分组成: 格式控制字符串和地址列表。(由 `%` 开始以格式字符结束, 表示输入数据的类型和格式; 表示用户输入数据时同时输入的符号, 用于分隔输入数据, 通常使用逗号或空格。)注意: 输入参数时的格式要与 `scanf` 中的分隔符一致。地址列表由若干地址组成, 如本例中的 `&a`, `&b`; `&a`, `&b` 分别表示变量 `a`, `b` 在内存中的地址。

(5) C 程序书写格式自由, 一行可以写几个语句 (例: `scanf ("%d, %d", &a, &A); m=max(a, A);` (若语句结束用; 隔开), 一个语句也可以写成多行 (最后用 \ 分开), 例如,

```
printf("%d 和 %d 相比最大的是: %d\n",
    \a, A, m);
```



想一想

如何在不对以上案例程序大幅修改的情况下实现: 输入三个整数, 输出最大的一个整数。

小练习

一、选择题

- 一个 C 程序的执行是从 ()。
 - 本程序的 `main` 函数开始, 到 `main` 函数结束
 - 本程序文件的第一个函数开始, 到本程序文件的最后一个函数结束
 - 本程序的 `main` 函数开始, 到本程序文件的最后一个函数结束
 - 本程序文件的第一个函数开始, 到本程序 `main` 函数结束
- 在 C 语言中, 每个语句必须以 () 结束。
 - 回车符
 - 冒号
 - 逗号
 - 分号
- C 语言规定: 在一个源程序中, `main` 函数的位置 ()。
 - 必须在最开始
 - 必须在系统调用的库函数的后面
 - 可以任意
 - 必须在最后
- 一个 C 语言程序由 ()。