HZ BOOKS

高等院校计算机教材系列

C/C++程序设计教程

秦维佳 伞宏力 侯春光 孟艳红 编著

为教师提供教学课件

机械工业出版社
China Machine Press

C/C++
程序设计教程

高等院校计算机教材系列

C/C++ 程序设计教程

秦维佳 伞宏力 侯春光 孟艳红 编著

机械工业出版社
China Machine Press

C/C++语言是程序设计的入门语言，也是理工类大学生必修的一门程序设计课程。本书在实例的选择上从易到难，循序渐进，使读者能够逐步了解C/C++语言的精髓，掌握结构化程序设计的方法，并初步了解面向对象的程序设计方法。

全书共分12章，5个附录。内容包括程序设计的基本概念，C语言的数据，结构化程序设计方法，数组、函数、指针的概念和实例，结构体与共用体，位运算，文件的概念和从文件中输入/输出数据，C++面向对象程序设计的基本概念，类的继承与多态和C++ I/O流。附录介绍了C语言的预编译命令，Visual C++集成环境的使用方法，C/C++常用的库函数，运算符的优先级与结合性等。

本书可作为大学本科、专科学生学习C/C++语言程序设计课程的教材，也可作为C/C++语言自学者的教材或参考书。

图书在版编目(CIP)数据

C/C++程序设计教程/秦维佳等编著. —北京：机械工业出版社，2007.4
(高等院校计算机教材系列)

ISBN 978-7-111-20609-5

Ⅰ.C…　Ⅱ.秦…　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159357号

机械工业出版社(北京市西城区百万庄大街22号　邮政编码　100037)
策划编辑：武恩玉
责任编辑：杨庆燕
三河市明辉印装有限公司印刷·新华书店北京发行所发行
2007年2月第1版第1次印刷
184mm×260mm·18.75印张
定价：29.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：(010)68326294

前　言

计算机应用能力是新世纪人才不可缺少的基本素质。程序设计是高等院校工科各专业计算机应用能力培养的重要技术基础。C/C++是目前国内外广泛使用的一种程序设计语言，也是国内外大学讲授程序设计方法的首选语言。

本书主要讲授C语言的语法及程序设计方法。由于TC 2.0的集成调试环境是DOS系统界面，因此不支持鼠标操作。计算机软、硬件技术发展到今天的水平，输入设备、输入方法很多，已经有许多读者不太适应这种环境了。C++是在C语言的基础上发展起来的，是C语言的超集，因此在C++集成环境下，调试C语言程序是完全可以实现的，为此我们编写了C/C++程序设计教程。第1.~9章介绍C语言的知识，所有程序分别在TC 3.0和Visual C++6.0集成环境下调试两遍；第10～12章介绍C++程序设计的知识，所有程序在Visual C++6.0集成环境下经过调试。通过本书的学习，不仅可掌握C语言的知识，也可了解C++程序设计的思想和熟悉C++程序调试的环境。在学习C程序设计时，同时了解C++语言是一种非常好的学习程序设计的方法，可为今后进一步深入学习和自学Visual C++程序设计打下良好的基础。

本书内容丰富、注重实践，由浅入深、便于理解，案例广泛、图文并茂，通俗易懂、便于自学。全书共分12章，5个附录。第1章介绍程序设计的基本概念，算法的描述和C/C++程序的结构以及上机的基本操作；第2章介绍C语言的数据；第3章介绍结构化程序设计方法；第4章介绍数组的概念和实例；第5章介绍函数的概念和实例；第6章介绍指针的概念和实例；第7章介绍结构体与共用体；第8章主要介绍位运算；第9章介绍文件的概念和从文件中输入/输出数据；第10～12章介绍C++面向对象程序设计的基本概念，类的继承与多态和C++ I/O流。附录介绍了C语言的预编译命令，Visual C++集成环境的使用方法，C/C++常用的库函数，运算符的优先级与结合性等。

C/C++是一门实践性很强的课程，实践是学好本课程的重要环节。为此我们同时编写了《C/C++程序设计实验教程》一书，给出了每章的知识结构、学习要点、上机实训、习题及参考答案。供读者学习时借鉴和参考。本书还提供实例源代码和电子课件，需要者可登录华章网站下载。

本书由秦维佳组织编写。其中第1～3章由秦维佳编写；第4～5章由孟艳红编写；第6～9章由侯春光编写；第10～12章由伞宏力编写。秦维佳对全书进行了统编、程序调试与定稿。在教材编写过程中寿业勇、于彤彤、赵翠红、赵爽等阅读了初稿，并提出了许多宝贵意见，在此表示衷心感谢。

本书可作为大学本科、专科学生学习C/C++语言程序设计课程的教材，也可作为C/C++语言自学者的教材或参考书。

由于编者水平有限，错误在所难免，请广大读者批评指正。

编　者

2006年10月

目　录

第1章　C语言及程序设计初步

C语言是目前世界上最流行、使用最广泛的高级程序设计语言。C语言的主要特色是兼顾了高级语言和汇编语言的特点，简洁、丰富、可移植。目前有许多应用程序都是由C语言来编写的，例如，数据结构、数据库系统、人机接口控制、数值分析、图像处理等。C语言受到人们的极大推崇。

本章将带你走进C语言的世界，初步了解C语言的功能，教你编写第一个C语言程序。在这一过程中，读者要了解C/C++发展简史，理解算法的基本概念，了解结构化程序设计方法，了解C语言程序的基本结构，掌握C语言的程序调试步骤。

1.1　C/C++历史及特点

1.1.1　C语言的历史

C语言是在20世纪70年代初问世的。

1978年美国电话电报公司(AT&T)的贝尔实验室正式发布了C语言。同时由B. W. Kernighan和D. M. Ritchit合著了著名的《The C Programming Language》一书。通常简称为K&R C，也有人称之为K&R C标准。但是，在K&R C中并没有定义一个完整的标准C语言。K&R C通常被作为C编译器所支持的最基本的C语言部分。

1983年，由美国国家标准学会(ANSI)组建了一个委员会，着手制定ANSI的标准C。

1988年，ANSI公布了标准ANSI C。这个标准的大部分特性已经由现代的编译系统所支持。

1989年，国际标准化组织(ISO)采纳了ANSI标准，称ANSI/ISO standard C(ANSI X3.159－1989)。标准化的一个目的是扩展K&R C。在ANSI标准化的过程中，一些新的特征被加了进去。在ANSI标准化后，C语言的标准在一段相当长的时间内都保持不变。C语言的标准在20世纪90年代才经历了改进，这就是ISO9899：1999(1999年出版)。这个版本就是通常提及的C99。它被ANSI于2000年3月采用。

早期的C语言主要用于UNIX系统。由于C语言的强大功能和各方面的优点逐渐为人们认识，到了20世纪80年代，C开始进入其他操作系统，并很快在各类大、中、小和微型计算机上得到了广泛的使用，成为当代最优秀的程序设计语言之一。

1.1.2　C语言的特点

1. C语言是一种功能强大的高级语言

C语言可以直接访问内存的物理地址，进行位(bit)一级的操作。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。它把高级语言的基本结构和语句与低级语言的实用性结合起来。C语言可以像汇编语言一样对位、字节和地址进行操作，而这三者正是计算机最基本的工作单元。

2. 结构化语言

C语言是一种结构化语言。它用函数作为程序的基本单位，使程序层次清晰，便于按模块化方式组织程序。结构化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。这种结构化方式可使程序层次清晰，便于使用、维护以及调

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向，从而使程序完全结构化。

3. 功能齐全

C 语言的表现能力和处理能力极强。它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。C 语言具有各种各样的数据类型，并引入了指针概念，可使程序效率更高。另外 C 语言也具有强大的图形功能，支持多种显示器和驱动器。其计算功能、逻辑判断功能也比较强大，可以实现决策目的。因此既可用于系统软件的开发，也适合于应用软件的开发。

4. 适用范围大

C 语言还具有效率高、可移植性强等特点。另外，C 语言还有一个突出的优点就是适合多种操作系统，如 DOS、UNIX 等，也适用于多种机型。因此 C 语言被广泛地移植到了各类型计算机上，从而形成了多种版本的 C 语言。

目前最流行的 C 语言有以下几种：

- Microsoft C 或称 MS C
- Borland Turbo C 或称 Turbo C
- AT&T C

这些 C 语言版本不仅实现了 ANSI C 标准，而且在此基础上各自作了一些扩充，使之更加方便、完善。

1.2 程序与程序设计

1.2.1 程序

其实在日常生活中，我们总是在不断地编写程序，并执行程序。例如，我们要用全自动洗衣机洗衣服，应该如何去做呢？这就需要编写一个洗衣服的程序，洗衣服的程序如下：

1)把脏衣服放到洗衣机中。

2)插好电源，打开水龙头。

3)设置洗衣服的命令，如经济、普通、轻揉。

4)放入洗涤剂。

5)按下开始按钮。

6)等待洗衣机清洗完毕。

当洗衣机清洗完毕后，蜂鸣器发出响声，即可将衣服取出晾晒。这就是人们洗衣服的程序。上述 6 个步骤，第 1)、2)步可以互换，第 3)、4)步也可以互换，同样能将衣服洗干净。所以干一件事情的“程序”并不唯一。因此“程序”一词对我们并不陌生。

在计算机中，程序是计算机的一组指令。如果没有程序，计算机就什么动作都不会做。与现实生活中的“程序”一样，计算机解决问题的程序也不唯一。要使计算机能完成人们预定的工作，就必须把要完成工作的具体步骤编写成计算机能执行的一条条指令。计算机执行这些指令后，就能完成指定的功能，这样的指令序列就是程序。

定义：程序是计算机执行后，能完成特定功能的指令序列。

计算机程序主要包括两个方面的内容：数据结构和算法。数据结构描述数据对象及数据对象之间的关系；算法描述数据对象的处理过程。

计算机程序有以下性质：

1)目的性：程序有明确的目的，程序执行后能完成指定的功能。

2)分步性：程序由计算机可执行的一系列基本步骤组成。

3)有序性：程序的执行步骤是有序的，不可随意改变程序步骤的执行顺序。

4)有限性：程序包含的指令序列是有限的。

5)操作性：有意义的程序总是对某些对象进行操作，完成预定的功能。

1.2.2 程序设计语言

目前，通用的计算机不能识别自然语言，只能识别特定的计算机语言。计算机语言即程序设计语言是程序设计的工具，一般分为低级语言和高级语言。

低级语言直接依赖于计算机硬件，不同的机型使用的低级语言是完全不一样的。高级语言则不依赖于计算机硬件，用高级语言编写的程序可以方便地、几乎不加修改地用在不同类型的计算机上。

1. 低级语言

低级语言包括机器语言和汇编语言。

机器语言是直接使用二进制代码表示的指令来编程的语言，它依赖于不同机型的指令系统。必须准确地牢记每一条指令的二进制代码，才能编写程序，因此不是一件容易的事情。机器语言的优点是执行速度快，并且可以直接对硬件进行操作。缺点是可读性差、可移植性差。

汇编语言是机器语言的符号表示。为了能够方便地编写程序，人们用一些符号和简单的语法来表示机器指令。例如，“101110001110100000000011”用汇编语言表示就是“MOV AX，1000”，该指令的功能是将 1000 送入寄存器 AX 中，这要比机器语言清楚多了，也方便多了。但是计算机的 CPU 不能识别汇编语言，需要一个“翻译”程序将汇编语言翻译成机器语言，这个程序称为“汇编器”。汇编语言与机器语言的指令是一一对应的，除了提高了可读性，汇编语言从根本上没有改变机器语言的特点。

2. 高级语言

高级语言是一种叙述性语言，是一种比较接近自然语言和数学语言的程序设计语言。高级语言与人类所惯用的语法比较接近，所以容易编写、排错。因此，高级语言的出现大大提高了程序员的工作效率，降低了程序设计难度，并且改善了程序质量。但是，相对于低级语言，高级语言对硬件的控制能力比较差，执行效率也比较低。常见的高级语言有 BASIC、FORTRAN、COBOL、PASCAL 等。

本书要讲的 C 语言不仅具有高级语言的优点，还兼顾了低级语言的特点，可以像汇编语言一样对位、字节和地址进行操作，实现了对硬件的编程，利用低级语言的特点来提高程序代码的执行效率。

1.2.3 程序设计

程序设计是一门技术，需要相应的理论、技术、方法和工具来支持。程序设计就是根据问题的需求，设计数据结构和算法，编制程序和调试程序，使计算机程序能完成所需要的任务。程序首先应能正确完成任务，并且是可靠的。简单地说，程序设计是设计和编制程序的过程。由于程序在使用过程中，环境或约束条件会有所改变，这就需要修改程序的功能。因此，好的程序应该具有可靠性、易读性、可维护性等良好的特性。为了满足这些特性要求，

应采用正确的程序设计方法，以便从程序设计方法上保证能设计出具有良好特性的程序。

除了好的程序设计方法和技术之外，程序设计风格也是很重要的。程序设计风格会极大地影响软件的质量和可维护性，良好的程序设计风格可以使程序结构清晰合理，使程序代码便于维护，因此程序设计风格对保证程序的质量非常重要。

一般来讲，程序设计风格是指编写程序时所表现出的特点、习惯和逻辑思路。程序是由人来编写的，为了测试和维护程序，往往需要阅读和跟踪程序，因此程序设计的风格强调简单和清晰，程序必须是可以理解的。著名的“清晰第一，效率第二”的论点已成为当今主导的程序设计风格。

要形成良好的程序设计风格，主要注重考虑以下因素。

1. 源程序文档化

源程序文档化应考虑如下几点：

1)符号名的命名：符号名包括文件名和变量名，符号名的命名应具有一定的实际含义，以利于对程序功能进行理解；符号名的命名还应该有规律，以利于对程序文件进行整理。

2)程序注释：正确的注释能够帮助读者理解程序。注释一般分为序言性注释和功能性注释。序言性注释通常位于每个程序的开头部分，它给出程序的整体说明。主要描述内容包括：程序标题、程序功能说明、主要算法、程序位置、接口说明、程序设计者、开发简历、审查者、修改日期等。功能性注释的位置一般嵌入在源程序体中，主要描述语句或程序做什么。

3)视觉组织：为了使程序的结构一目了然，可以在程序中利用空格、空行、缩进等技巧使程序层次清晰。

2. 数据说明的方法

在编写程序时，需要注意数据说明的风格，以便使程序中的数据说明更易于理解和维护。一般应注意如下几点：

1)数据说明的次序规范化。鉴于程序理解、阅读和维护的需要，使数据说明次序固定，可以使数据的属性容易查找，这样有利于测试、排错和维护。

2)说明语句中变量安排有序化。当一个说明语句说明多个变量时，变量按照字母顺序排列为好。

3)使用注释语句来说明复杂数据的结构。

3. 语句的结构

程序应该简单易懂，语句构造应该简单直接，不应该为提高效率而把语句复杂化。一般应该注意如下几点：

1)在一行只写一条语句。

2)编写程序首先考虑清晰性。

3)除非对效率有特殊要求，否则编写程序时，要做到清晰第一，效率第二。

4)首先保证程序正确，然后再提高速度。

5)避免使用大量的临时变量以防使程序的可读性下降。

6)避免使用无条件转移语句。

7)尽可能使用库函数。

8)避免使用复杂的条件嵌套语句。

9)模块功能尽可能单一，即一个模块完成一个功能。

10)不要对不良的程序修修补补，不良的程序要重新编写，避免因修改带来新的问题。

4. 输入和输出

输入/输出是用户最关心的问题，输入/输出的方式和格式应尽可能方便用户的使用，因为系统能否被用户接受，往往取决于输入/输出的风格。无论是批处理的输入和输出方式，还是交互式的输入和输出方式，在设计和编程时都应考虑如下原则：

1)对所有的输入数据都要检验其合法性。

2)检查输入项之间的合理性。

3)输入数据尽可能少，操作尽可能的简单。

4)在以交互输入/输出方式进行输入时，要采用人—机会话的方式给出明确的提示信息和运行的状态信息。

5)设计输出格式。

1.3 算法与算法的表示方法

程序设计离不开算法，程序 = 算法 + 数据结构。算法是指解决问题的方法和步骤。

1.3.1 算法的概念

为解决一个实际问题而采取的方法和步骤称为“算法”。对于同一个问题，可能有不同的方法和步骤，即可能有不同的算法。

【例 1.1】 给出求 1 +2 +3 +… +100 = ? 的算法描述。

第 1 种算法：

```
步骤 1：1 +2 =3
步骤 2：3 +3 =6
……
步骤 99:4950 +100 =5050
```

第 2 种算法：

```
步骤 1：0 +100 =100
步骤 2：1 +99 =100
步骤 3：2 +98 =100
……
步骤 50：49 +51 =100
步骤 51：100 * 50 =5000
步骤 52：5000 +50 =5050
```

第 3 种算法：

```
步骤 1：i =1,s =0
步骤 2：如果 i >100,则算法结束,s 即为所求结果,输出 s;否则转向步骤 3 执行.
步骤 3：s =s +i,i =i +1
步骤 4：转向步骤 2 执行.
```

第 4 种算法：

```
步骤 1：i =100,s =0
步骤 2：如果 i <1,则算法结束,s 即为所求结果,输出 s;否则转向步骤 3 执行.
步骤 3：s =s +i,i =i -1
步骤 4：转向步骤 2 执行.
```

可以看出，解决同一个问题有多种算法。当然，算法有优劣之分，有的算法简练，有的

算法烦琐。一般来说，总是希望采用计算简单、运算步骤少的算法。上面4个算法中，算法1的步骤最多，算法3和算法4的步骤最少，质量最优。因此，为了有效地解决问题，在保证算法正确的前提下，要考虑算法的质量，在多种算法中选择合适的算法。

1.3.2 算法的特性

算法是一个有穷的指令集，是解决某一问题的运算序列。一个算法一般应具有以下几个基本特征：

1)可行性：算法中描述的操作都可以通过已经实现的基本运算执行有限次来实现。

2)确定性：算法的确定性，是指算法中的每一个步骤都必须是有明确定义的，不允许有模棱两可的解释，也不允许有多义性。对于每一种情况，需要执行的动作都应严格地、清晰地规定。

3)有穷性：算法的有穷性，是指算法必须能在有限的时间内做完，即算法必须能在执行有限个步骤之后终止。算法的有穷性还应包括合理的执行时间的含义。假如，一个算法需要执行千万年，显然失去了实用价值。

4)输入：算法总是要施加到运算对象上，提供运算对象的初始情况。因此，一个算法必须有0个或多个输入。

5)输出：一个算法应该有一个或多个输出，输出的量是算法计算的结果。如果没有输出，则无法知道结果。

1.3.3 算法的常用表示方法

1. 流程图

流程图是用图形的方式来表示算法，是软件制作过程的设计表示工具，流程图表达直观、清晰，易于学习和掌握。通常用一些几何图形来代表各种不同性质的操作。美国国家标准化协会(ANSI)规定了一些常用流程图符号，已被大多数国家接受，如图1-1所示。

【例1.2】 画出计算并输出两个数之和的程序流程图，如图1-2所示。

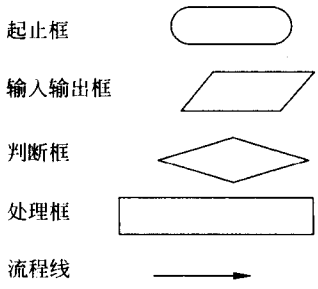

图1-1 流程图符号

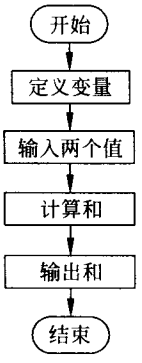

图1-2 程序流程图举例

2. N-S结构图

N-S图是用方框图来代替传统的程序流程图的技术。在这种流程图中，把流程线全部去掉，全部算法写在一个矩形框内，在框内还可以包含其他框。N-S结构图的基本图形有5种，如图1-3所示。

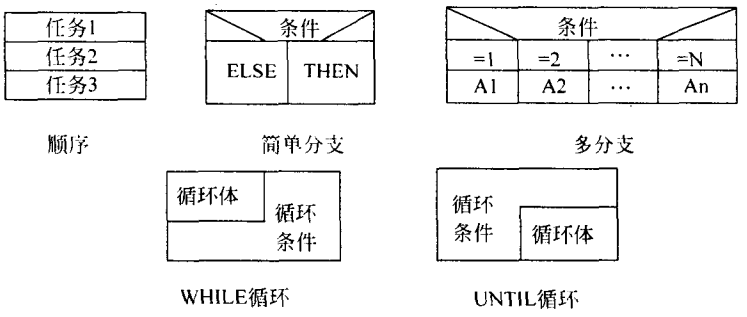

图 1-3 N-S 图的基本图形

3. PAD 图

PAD 图是问题分析图(Problem Analysis Diagram)。它是继程序流程图和 N-S 结构图之后，提出的又一种描述算法的图形工具。基本图形有 5 种，如图 1-4 所示。

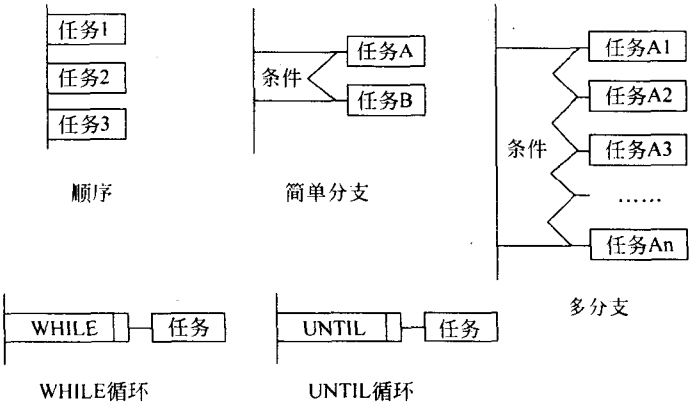

图 1-4 PAD 图的基本图形

4. 伪代码表示法

伪代码是介于自然语言和计算机语言之间的文字和符号，用来表示算法，即用与程序设计语言很相近的伪代码描述处理过程的细节，如结构化语言。伪代码中，关键字用英文表示，其他的可用汉字，也可用英文，只要便于书写和阅读即可。用伪代码表示算法无固定、严格的语法规则，只要求把意图表达清楚，书写格式清晰易懂即可。

【例 1.3】 某工厂对工人的超产奖励政策为：该厂生产两种产品 A 和 B，凡工人每月的实际生产量超过计划指标者均有奖励。奖励政策为：

对于产品 A 的生产者，超产数 N 小于或等于 100 件时，每超产 1 件奖励 2 元；N 大于 100 件小于等于 150 件时，大于 100 件的部分每件奖励 2.5 元，其余的每件奖励金额不变；N 大于 150 件时，超过 150 件的部分每件奖励 3 元，其余按超产 150 件以内的方案处理。

对于产品 B 的生产者，超产数 N 小于或等于 50 件时，每超产 1 件奖励 3 元；N 大于 50 件小于等于 100 件时，大于 50 件的部分每件奖励 4 元，其余的每件奖励金额不变；N 大于 100 件时，超过 100 件的部分每件奖励 5 元，其余按超产 100 件以内的方案处理。将该超产奖励政策用伪代码描述。

伪代码描述如下：

```
If 产品 = A
   Case 数量 <= 100
      奖金 = 2 × 数量
   Case 数量 > 100 and 数量 <= 150
      奖金 = 200 + (数量 - 100) × 2.5
   Case 数量 > 150
      奖金 = 200 + 125 + (数量 - 150) × 3
Else
   Case 数量 <= 50
      奖金 = 3 × 数量
   Case 数量 > 50 and 数量 <= 100
      奖金 = 150 + (数量 - 50) × 4
   Case 数量 > 100
      奖金 = 150 + 200 + (数量 - 100) × 5
EndIf
```

1.4 C语言程序的基本结构

1.4.1 程序的组成

计算机程序是指计算机为解决某个问题或完成某项任务而编制的指令序列。例如，输入两个数，然后计算并输出这两个数的和。下面是一段比较简单的C语言程序，以此为例，介绍C语言程序的组成。

【例1.4】 编写程序计算并输出两个数的和。

程序如下：

```
 1  /* ------------ sum of x add y ------------ */
 2  #include <stdio.h>
 3  void main()                    /* 定义主函数,整数加法器程序 */
 4  {
 5    int x,y;                     /* 变量定义 */
 6    int z;                       /* 变量定义 */
 7    scanf("%d,%d",&x,&y);        /* 输入两个整数,数据间以逗号分隔 */
 8    z = x + y;                   /* 求 x + y 的和,并存入 z */
 9    printf("Sum = %d",z);        /* 输出结果,即 z 的值 */
10  }
```

1，2，3，……，10是为了方便阅读和解说而编写的行号，在编辑程序时不用输入。

通常，一个程序包含数据输入、处理、数据输出这三部分内容。此程序的功能是，在运行时接收用户输入的两个数，然后计算这两个数之和，并输出和。

分析说明：

该程序包括三部分：注释、预处理命令及函数定义。

语句1：用“/*”和“*/”括起来的是注释行。注释行用于说明程序的主要功能，编译系统会跳过注释行，不对其进行翻译。

语句2：以#号开头的是预处理命令。这些命令是在编译系统翻译代码之前需要由预处理程序处理的语句。本例中的#include <stdio.h>语句是请求预处理程序将文件stdio.h包含到程序中，作为程序的一部分。文件stdio.h中包含一些重要的定义，如果没有文件stdio.h，则scanf("%d,%d",&x,&y);与printf("Sum = %d",z);这两条语句就不能通过编译系统的“翻译”。

语句3：定义主函数。每个C程序必须包含一个主函数main()，也只能包含一个主函数。用{}括起来的部分是一个程序模块，在C语言中称为分程序，每个函数至少有一个分