新世纪应用型高等教育
计算机类课程规划教材

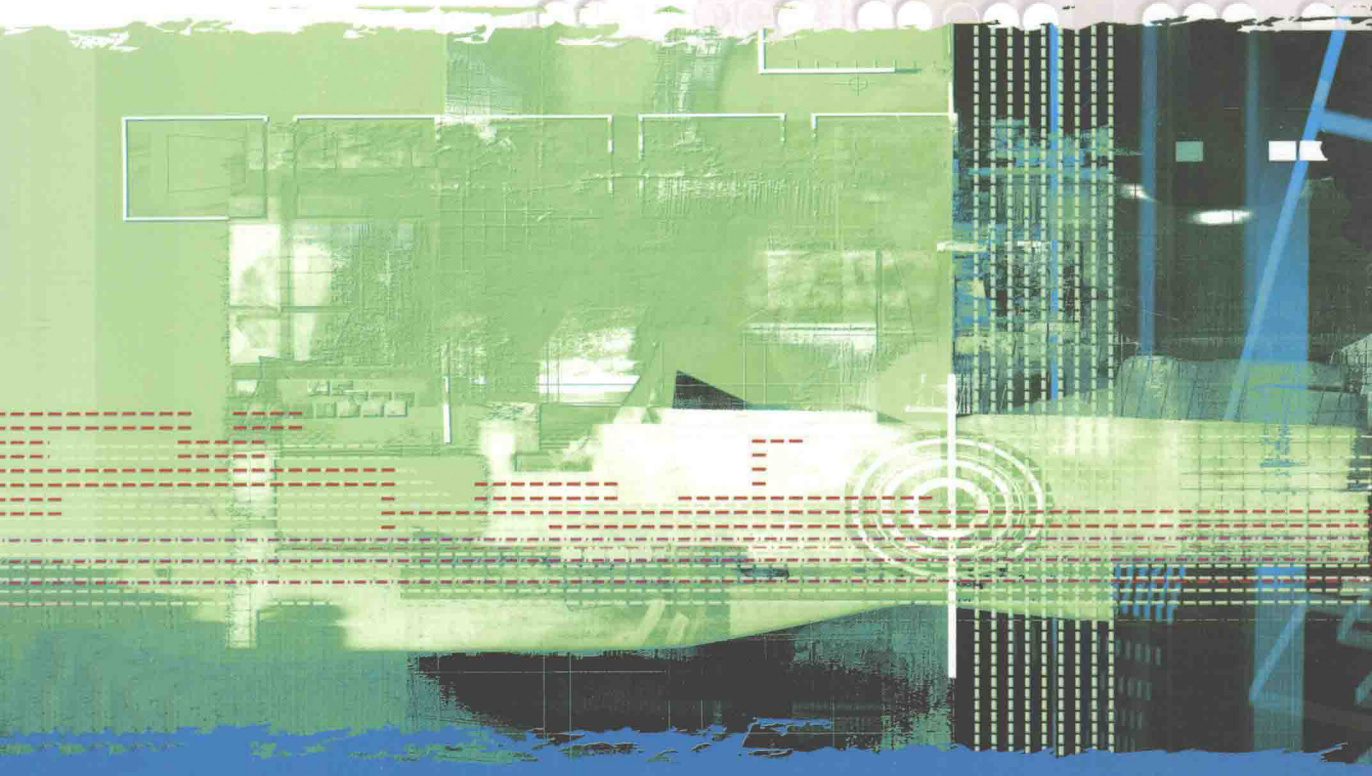
新世纪

C语言程序设计

新世纪应用型高等教育教材编审委员会 组编

主编 万 琼 盛国栋

主审 李 娟



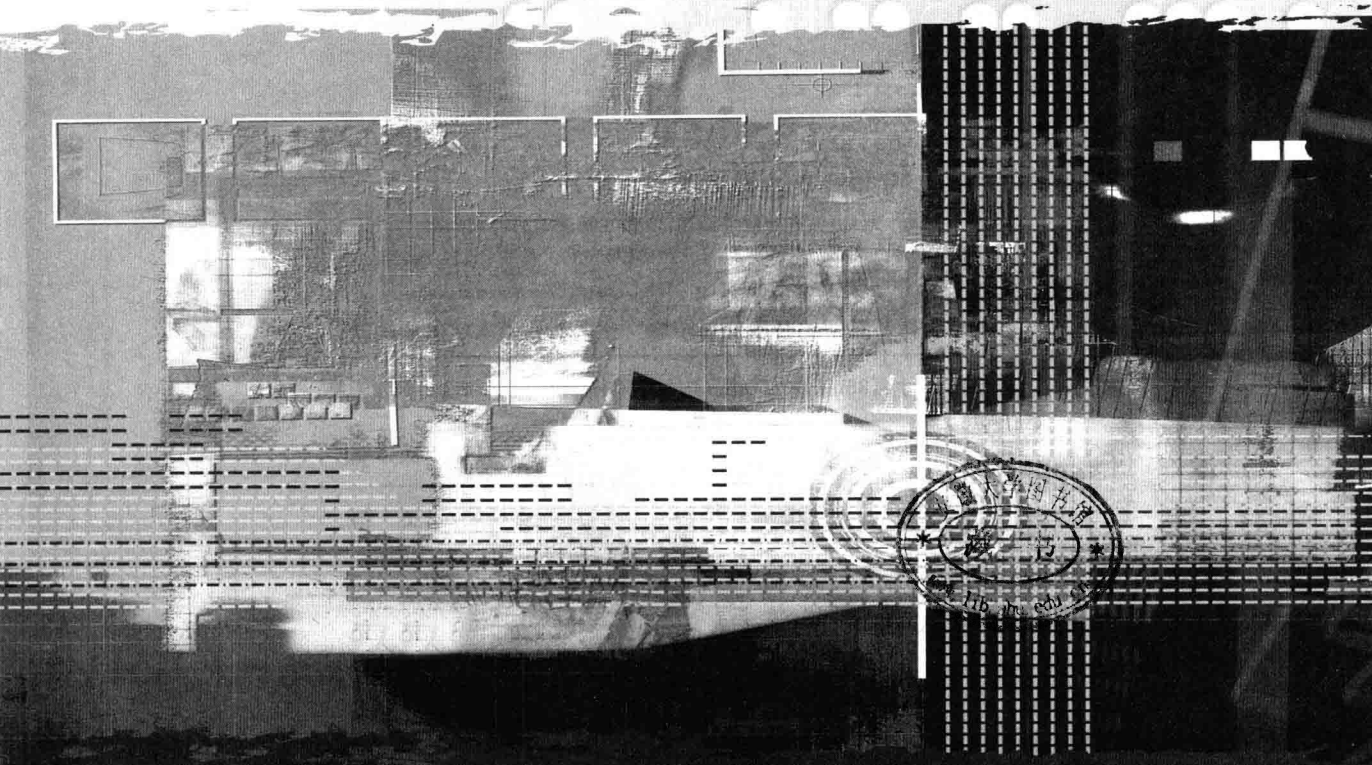
大连理工大学出版社

新世纪应用型高等教育
计算机类课程规划教材

C语言程序设计

C YUYAN CHENGXU SHEJI

新世纪应用型高等教育教材编审委员会 组编
主编 万 琼 盛国栋
副主编 亚森·艾则孜
安尼瓦尔·加马力 于 丽
主审 李 娟



大连理工大学图书馆 藏书 lib.dlut.edu.cn

大连理工大学出版社

图书在版编目(CIP)数据

C语言程序设计 / 万琼，盛国栋主编. — 大连 ：大连理工大学出版社，2012.8

新世纪应用型高等教育计算机类课程规划教材

ISBN 978-7-5611-7236-0

Ⅰ. ①C… Ⅱ. ①万… ②盛… Ⅲ. ①C语言－程序设计－高等学校－教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2012)第199491号

大连理工大学出版社出版

地址：大连市软件园路80号　邮政编码：116023

发行：0411-84708842　邮购：0411-84703636　传真：0411-84701466

E-mail：dutp@dutp.cn　URL：http://www.dutp.cn

大连美跃彩色印刷有限公司印刷　　大连理工大学出版社发行

幅面尺寸：185mm×260mm　　印张：20　　字数：462千字

印数：1～2000

2012年8月第1版　　2012年8月第1次印刷

责任编辑：马　双　　责任校对：王　倩

封面设计：张　莹

ISBN 978-7-5611-7236-0　　定　价：39.50元

前　言

C语言程序设计是我国大部分高校都开设的专业基础课程，是计算机相关专业程序设计入门非常重要的必修课程。C语言具有很多优点，如语言简洁、功能丰富、数据结构多样、可移植性好、目标程序效率高等，是一种适合用来编写系统软件和应用软件的语言。

在编写本书过程中，作者结合自己多年在高等院校从事C语言程序设计教学的经验，理论结合实际，力求通俗易懂。本书在体系结构安排上，根据教学目的和要求，各章以示例入手，尽可能将概念、知识点与例题结合起来，每章结尾均对该章内容进行小结，章末均附有不同类型的习题，除第一章外，每章均设置有数量不等的实验内容。

本书可作为应用型本科、高职高专院校的C语言程序设计课程的教材，也可作为各类计算机培训班学员学习C语言程序设计的教材以及软件开发人员和自学人员的技术参考书。

本书由万琼、盛国栋担任主编，亚森·艾则孜、安尼瓦尔·加马力、于丽担任副主编，赵旭东、贺一峰、张扬、吴霞、王亚娟参编。其中，第7、8章由万琼编写，第5、10章由盛国栋编写，第2、4章由亚森·艾则孜编写，第1、3章由于丽编写，第6、9章由安尼瓦尔·加马力编写。赵旭东参编第7、8章，贺一峰参编第5、10章，张扬参编第2、4章，王亚娟参编第1、3章，吴霞参编第6、9章。

新世纪

由于作者水平有限，加上时间仓促，书中难免有错误和不足之处，恳请专家和读者批评指正。

所有意见和建议请发往：dutpgz@163.com

欢迎访问我们的网站：http://www.dutpbook.com

联系电话：0411-84707492　84706104

编　者

2012年8月

试读结束，需要全本PDF请购买 www.ertongbook.com

目　录

第1章 程序设计基础

本章教学目标

通过本章的学习，要求了解程序、程序设计语言、语言处理程序及算法的基本概念；熟悉设计程序的基本原则、算法的表示方法、结构化程序设计方法；掌握自然语言、程序流程图、伪代码表示算法的方法、结构化程序设计的基本结构。

本章教学内容

本章教学共需2学时，其中理论教学2学时，实践教学0学时。主要教学内容如下：

1. 程序和程序设计语言
2. 算法
(1)算法的概念
(2)算法的表示方法
①自然语言
②程序流程图
③N-S流程图
④伪代码
⑤计算机语言
3. 结构化程序设计
(1)基本控制结构
(2)设计方法

本章教学重点

本章的教学重点内容如下：(1)设计程序的基本原则
(2)算法的表示方法
(3)结构化程序设计的基本结构

其中要注意的难点问题是：算法的表示方法

1.1 程序和程序设计语言

电子计算机是20世纪人类最伟大的发明之一，它对人类影响极其深远。自计算机问世以来，随着计算机硬件和软件的不断升级换代以及计算机应用领域的迅速扩大，计算机程序设计语言也有了很大的发展。

试读结束，需要全本PDF请购买 www.ertongbook.com

1.1.1 程序设计和程序设计语言介绍

程序设计也可称为一门工程设计，它是根据要解决的问题，使用某种程序设计语言，设计出能够完成这一任务的计算机指令序列。

程序设计语言是人与计算机进行交流的一种形式语言，是人利用计算机分析问题、解决问题的一个基本工具。就如同人类社会中，自然语言是人与人之间用来表达意思、交流思想的工具一样。自然语言是山字、词、句和语法等构成的一个系统；而计算机程序设计语言是山字、词和语法等构成的指令系统。

最初程序员使用的程序设计语言是原始的计算机指令，即能够被计算机直接识别的一系列二进制数，称为机器语言。

在机器语言的基础上，人们设计出了汇编语言，它可以将机器语言用便于人们记忆和阅读的助记符来表示，如 ADD，SUI3，MOV 等。计算机运行汇编程序时，首先将用助记符写成的源程序转换成机器能够识别的指令，然后再运行机器指令程序，得到所要的结果。

随着计算机应用的发展，人们开发出了高级程序设计语言，1972 年至 1973 年间，贝尔实验室的 D. M. Ritchie 在 B 语言的基础上设计出了 C 语言，后来 C 语言又做了多次改进。

C 语言是国际卜广泛流行的、很有发展前途的计算机高级语言。它适合作为系统描述语言，既可以用来写系统软件，同时也可用来写应用软件。

在使用计算机程序设计语言设计程序时，要达到的目标是：在保证程序正确的前提下，力求程序可读性强、容易维护、移植性好。程序的可读性是指程序要有良好的书写风格，用简单易懂的语句编写程序，书写风格包括语句的对齐、规范的注释等。容易维护是指当业务规则发生变化时，要求以最小的开销对程序功能进行更改或增加。移植性好是指编好的程序可以在不同的计算机和操作系统卜运行，并且运行的结果一样。

程序语言的发展，总是从低级到高级，从具体到抽象，直到可以用自然语言来描述。

1.1.2 语言处理程序

计算机不能直接执行用高级语言编制的程序（源程序），那么，计算机怎样鉴别源程序，确定它的正确性，如何运行并获得预期的计算结果，这些是语言处理程序要解决的问题。

用高级语言（或汇编语言）编写的程序，即源程序是语言处理程序要处理的对象，语言处理程序把源程序翻译成语义等价的计算机能够识别的低级语言程序（目标程序）。由此可见，对语言处理程序来说，源程序是其处理对象（输入程序），目标程序是其处理结果（输出程序），它是在高级语言和计算机之间起到翻译作用的程序。

语言处理程序可用汇编语言或高级语言写成。它是为用户设计的编程服务软件，其作用是将高级语言源程序翻译成计算机能识别的目标程序。语言处理程序环境一般由汇编程序、编译程序或解释程序与相应的操作程序等组成，一般称之为编译（程序）系统

或解释(程序)系统等。

在软件领域中,包含程序设计语言和相关语言处理程序及其他相关工具的程序设计环境,通常被称为某种程序设计语言的集成开发环境。不同语言的语言处理程序是不同的,同一种高级语言也可能存在不同级别、不同版本的语言处理程序。程序员可以在不了解语言处理程序的情况下,借助集成开发环境完成程序的编制和运行。常用集成开发环境有多种,如微软的 Visual Studio. Net 可以称为 C++、VB、C # 等语言的集成开发环境,Borland 的 C++ Builder、Delphi、JBuilder 分别为 C++、Pascal、Java 语言的集成开发环境。

1.1.3　程序设计的基本原则

程序设计也是一种工程设计(通常称为软件工程),程序设计追求的目标是程序的可靠性、可理解性、可维护性和执行效率。但执行效率与可维护性、可理解性一般是相互矛盾的,因此程序设计的目标是设计出可靠、易读而且代价合理的程序。

程序设计时应该遵循的几个基本原则:

(1)正确性:正确可靠无疑是程序设计最基本的要求,程序设计错误将导致设计的程序工作毫无意义。

(2)有效性:一个好的程序运行效率要高,既要考虑减少计算机运行时间,又要考虑节省计算机存储空间。

(3)鲁棒性:程序设计时必须考虑到可能发生的异常事件并做出相应处理。一个好的程序应当是鲁棒的、健壮可靠的,即使用户非法操作(如不正确的输入)程序也能继续正常工作。

(4)可理解性:程序设计应选择恰当的组织方式及布置格式,正确地编写、组织、管理与程序有关的程序清单、说明书、使用手册等文字资料,来提高它的可理解性。

(5)可维护性:是指纠正程序出现的错误和缺陷以及为满足新的要求进行修改、扩充或压缩的容易程度。

(6)可移植性:简单地说就是指源代码在不同的平台上工作的兼容性。当程序移植到不同平台上,修改的代码越少,程序可移植性越好。

1.2　算　法

1.2.1　算法的概念

算法(algorithm)是为解决某一个具体问题而采取的确定的、有限步骤的方法。计算机算法即计算机能够执行的算法,是指以一步接一步的方式来详细描述计算机如何将输入转化为所要求的输出的过程。通常计算机算法分为两类:

- 数值计算算法:是利用数学的计算方法来得到运算结果。
- 非数值计算算法:则常用逻辑运算或数据运算来分析解决一些事务管理的问题。

在学习数学过程中我们知道，要解一个方程需要用一定的方法和步骤(算法)。下面通过求解一个数学题目来理解算法的概念。

例 1.1 求一个一元二次方程的解。

使用自然语言表示算法如下：

步骤 1：将方程写为标准形式，即 $ax^2+bx+c=0$。利用 b^2-4ac 的值来判断方程解的情况：无解、有一个解或有两个解。

步骤 2：求 b^2-4ac 的值进行判断，如果该值小于 0，则此方程无解，执行步骤 5；如果等于 0，则此方程有一个解，执行步骤 3；否则方程有两个解，执行步骤 4。

步骤 3：求 $x=\frac{-b}{2a}$的值，为方程唯一的解，执行步骤 5。

步骤 4：求 $x_1=\frac{-b+\sqrt{b^2-4ac}}{2a}$，$x_2=\frac{-b-\sqrt{b^2-4ac}}{2a}$的值，为方程的两个有效解，执行步骤 5。

步骤 5：解题结束。

以上是在求解一元二次方程时所采用的方法和步骤。程序设计也是要利用类似方法和步骤控制计算机来实现解题或完成某项工作。这种方法和步骤就是计算机程序设计的算法。人们根据经验将一些典型和成熟的算法汇编成册，供大家学习和使用，但是算法设计的步骤和方法并不是固定不变的。如例 1.1 的求解算法中，也可以把步骤 2、3、4 顺序和内容做一些调整，照样可正确求解，这说明完成某一任务，并不一定只有一种算法，但算法应具备一些特征。

一个算法应该具备以下五个特征：

- 有穷性(有限性)。任何一种算法都应在有限的操作步骤内完成，哪怕提出的解题方法是失败的。
- 确定性(唯一性)。解题算法中的任何一个操作步骤都应是清晰无误的，不会产生歧义或者误解。
- 可行性(能行性)。解题算法中的任何一个操作步骤在现有计算机软、硬件条件下和逻辑思维中都能够实现。
- 有 0 到多个输入。解题算法中可以没有数据输入，也可以同时输入多个需要算法处理的数据。
- 一个算法执行结束之后必须有数据处理结果输出，哪怕是输出错误的数据结果，没有输出的算法是毫无意义的。

1.2.2 算法的表示方法

虽然算法与计算机程序密切相关，但二者也存在区别：计算机程序是算法的一个实例，是将算法通过某种计算机语言表达出来的具体形式；同一个算法可以用任何一种计算机语言来表达。

描述算法可以用自然语言、程序流程图、伪代码、计算机语言、PDL 图以及 N-S 图等。下面介绍描述算法的几种常用方法。

1. 用自然语言描述算法

用人们日常使用的语言来描述算法，称为算法的自然语言描述。如例 1.1 中就是采用自然语言描述算法的方式。为加深对自然语言描述算法的理解，下面再看一个例子。

例 1.2　用自然语言描述求解 S=1×2×3……×(n−1)×n 的算法。

①确定 n 的一个值；

②假设 i 的初始值为 1；

③假设 S 的初始值为 1；

④如果 i≤n 时，转去执行⑤，否则转去执行⑦；

⑤计算 S 乘以 i 的值后，重新赋值给 S；

⑥计算 i 加 1 的值，然后将该值重新赋给 i，转去执行④；

⑦输出 S 的值，算法结束。

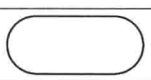
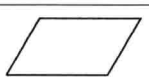
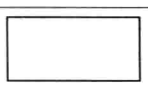
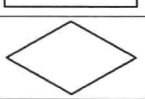
从上面这个描述求解的过程中不难发现，自然语言描述算法的方法比较容易掌握。但存在很大缺陷：如果算法中含有多分支或循环操作，则很难表述清楚。此外，使用自然语言描述算法还很容易造成歧义。

2. 用程序流程图描述算法

程序流程图(Program Flow Chart)是软件开发者最熟悉的一种算法表达工具，它独立于任何程序设计语言。它的优点是直观、清晰、易于掌握，便于转化成任何计算机程序设计语言。因此，它是软件开发者常用的算法表示方式。

在学习使用流程图描述算法之前，先对流程图中的一些常用符号作一解释。

表 1-1　　**程序流程图的常用符号**

符　号	名　称	作　用
	开始框或结束框	表示算法的开始或结束
	输入框或输出框	表示算法过程中，从外部获取的信息(输入)，然后将处理过的信息输出
	处理框	表示算法过程中，需要处理的内容，只有一个入口和一个出口
	判断框	表示算法过程中分支结构的条件，通常用菱形框上面的顶点表示入口，其余顶点表示出口
	流程线	算法过程中流程指向的方向

例 1.3　用程序流程图描述例 1.2 求解过程的算法。

从上面算法流程图中，可以比较清晰地看出求解问题的执行过程。但是程序流程图的符号在使用过程中不容易规范，特别是在标准中没有严格规定流程线的用法，流程线能够指示流程控制方向的随意转移，很容易造成算法中操作步骤的执行次序混乱，而且不便于开发人员交流。

3. 用 N-S 图描述算法

N-S 图是 1973 年由美国学者 I. Nassi 和 B. Shneiderman 提出的一种新的流程图形式，N 和 S 是两学位者姓氏的首字母。在这种流程图中，摒弃了带箭头的流程线，算法的

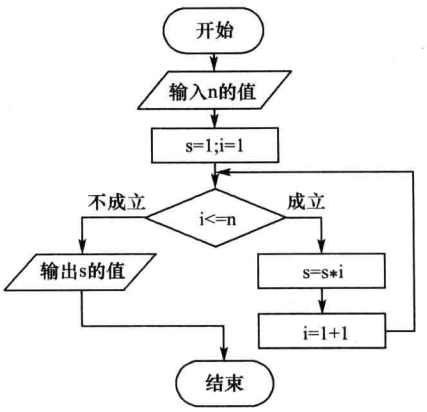


图 1-1 用程序流程图描述例 1.2 的算法

具体内容都写在一个矩形框内，框内又可以包含其他的从属框。

N-S 图表示不同程序结构的符号如下：

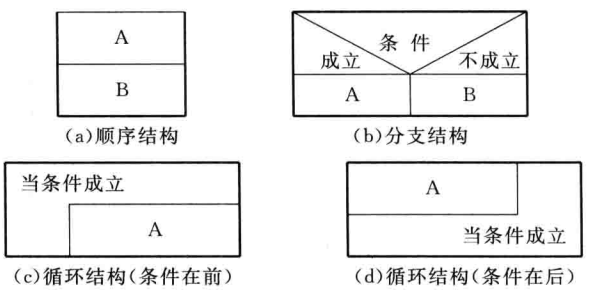


图 1-2 N-S 图表示不同程序结构的符号

(1)顺序结构：图 1-2(a)表示顺序结构，程序流程从 A 框到 B 框。

(2)分支结构：图 1-2(b)表示分支结构，判断条件是否成立。条件成立执行 A 框，条件不成立执行 B 框。

(3)循环结构：图 1-2(c)和图 1-2(d)分别表示循环结构。图 1-2(c)表示当循环条件成立时，执行 A；否则，终止执行 A。图 1-2(d)表示执行 A，直到循环条件不成立时终止执行 A。

以上三种结构框的矩形框中又可以包含这三种结构，从而组成复杂的 N-S 图。下面来看一个实例。

例 1.4 用 N-S 图描述例 1.2 求解过程的算法。

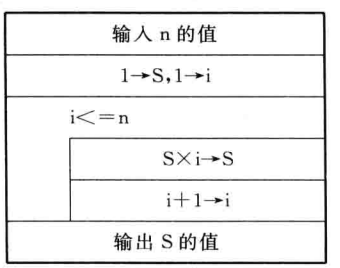


图 1-3 用 N-S 图描述例 1.2 中的算法

4. 用伪代码描述算法

伪代码(Pseudocode)作为一种算法描述语言,同使用自然语言还是使用流程图描述算法一样,都只是表述了编程者解决问题的一种思路,它们均无法被计算机直接接受并进行操作。使用伪代码的目的是为了使被描述的算法可以容易地以任何一种编程语言(Pascal,C,Java,C#等)实现。

例 1.5　用伪代码描述例 1.2 求解过程的算法。

(1)Start;
(2)输入 n 的值;
(3)S ← 1;
(4)i ← 1;
(5)do while i<=n
(6){ S ← S × i;
(7) i ← i + 1;}
(8)输出 S 的值;
(9)end。

伪代码是一种用来书写程序或描述算法时使用的非正式、透明的表述方法。它并非是一种编程语言,这种方法针对的是一台虚拟的计算机。伪代码是用介于自然语言和计算机语言之间的文字符号来描述算法的。

5. 用计算机语言描述算法

这种方法是直接用计算机语言书写算法,一步到位。选用的程序设计语言不一样,算法描述的形式也不一样,计算机语言描述算法必须严格遵循所用语言的语法规则。

例 1.6　用 C 语言描述例 1.2 求解过程的算法。

```
/* 源文件名:AL1_6.c */
#include <stdio.h>
void main()
{   int i,n;
    long S;
    printf("请输入整数 n 的值:");
    scanf("%d",&n);
    i=1;S=1;
    while(i<=n)
    {   S=S*i;
        i=i+1;
    }
        printf("%d! =%ld",n,S);
}
```

使用 Visual C++ 6.0 或 Turbo C 2.0 对上面 C 语言描述的算法程序进行编译、连接、运行,输入 5 后按回车,屏幕显示:

请输入整数 n 的值:5 ↙

试读结束，需要全本PDF请购买 www.ertongbook.com