C语言程序设计

孙晓华 姜孝军 吴亚明 主编

吉林大学出版社
JILIN UNIVERSITY PRESS

C 语言程序设计

主　编　孙晓华　姜孝军　吴亚明

副主编　陈　思　刘宇洋

吉林大学出版社

图书在版编目(CIP)数据

C语言程序设计 / 孙晓华，姜孝军，吴亚明主编. -- 长春 : 吉林大学出版社，2012.7
ISBN 978-7-5601-8729-7

Ⅰ. ①C… Ⅱ. ①孙… ②姜… ③吴… Ⅲ. ①C语言-程序设计 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2012)第164552号

书　名：C语言程序设计
作　者：孙晓华　姜孝军　吴亚明　主编
责任编辑：刘子贵　责任校对：卢　婵
吉林大学出版社出版、发行
开本：787×1092毫米　1/16
印张：15　字数：362千字
ISBN 978-7-5601-8729-7

封面设计：刘　川
长春日升印业有限公司　印刷
2012年7月第1版
2012年7月第1次印刷
定价：36.00元

版权所有　翻印必究

社址：长春市明德路501号　邮编：130021
发行部电话：0431-88580026/28/29
网址：http://www.jlup.com.cn
E-mail：jlup@mail.jlu.edu.cn

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。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。

本书系作者根据多年从事C语言课程教学实践经验和应用C语言的体会，在多次使用以及逐步完善的讲义基础上，并广泛参考有关资料编写而成的。在结构上符合学习逻辑，内容充实，选材上注重系统性和实用性。全书精选了大量典型例题，帮助读者加深理解和掌握所学内容。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. 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，将概念表达得准确、清楚。

3. 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，尽量避免了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。

4. 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。

本书由孙晓华、姜孝军、吴亚明主编，陈思、刘宇洋任主编。具体分工如下：孙晓华（哈尔滨理工大学）编写了第一、二、三、六章；姜孝军（吉林工程技术师范学院）编写了第八、十章；吴亚明（绥化学院）编写了第七、九、十二章；陈思（长春师范学院）编写了第四、十一章；刘宇洋（黑龙江职业学院）编写了第五章。

由于作者水平有限，书中难免有错误之处，请读者批评指正。

编　者

2012年4月

目 录

此为试读，需要完整PDF请访问：www.ertongbook.com

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、程序的错误和测试、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。

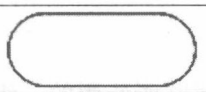
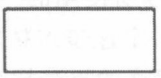
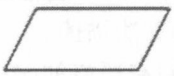
按照上述算法给定的 7 个步骤，就能求解“奇数和”问题。若选用一种计算机语言正确描述这个算法，就会得到求解“奇数和”问题的计算机程序，运行该程序，将得到“奇数和”问题的计算结果。

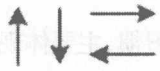
1.1.3 程序流程图

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
	输入、输出
	判断

流程图符号	符号功能
	流程方向

如图 1 - 1 所示为“计算 10 000 以内的所有奇数和”的程序流程图。

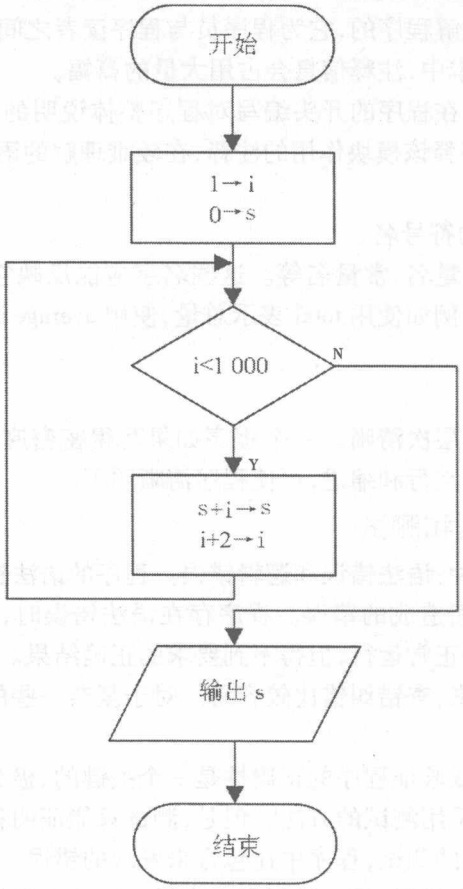


图 1 - 1　“计算 10 000 以内的所有奇数和”的程序流程图

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，可使程序清晰明了。

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 语言是 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,最后促成了 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 ANSIC。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 Strou - strup 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)
```


此为试读,需要完整PDF请访问: www.ertongbook.com

}

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2 000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,word! \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,word!

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 的值作为函数的返回值,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。

1.2.4　标识符与保留字

在程序中使用的变量名、函数名等统称为标识符。除库函数的函数名由系统定义外,其余都由用户自定义。C 语言规定,任何一个标识符只能是由字母(A ~ Z,a ~ z)、数字(0 ~ 9)和下划线(_)构成的字符串,其他符号不能出现在标识符中,并且标识符的第一个字符必须是字母或下划线。

以下标识符是合法的:

max　　class1　　max_1　　_add　　num007

以下标识符是非法的:

12d　　不允许以数字开头

sre * T　　使用了非法字符“ * ”

book - 1　　出现非法字符“ - ”(减号)

在使用标识符时还必须注意以下几点:

(1)标准 C 不限制标识符的长度,但它受各种版本的 C 语言编译系统的限制,同时也受

到具体机器的限制。例如,在某版本 C 语言中规定标识符前八位有效,当两个标识符前八位相同时,则被认为是同一个标识符。

(2)在标识符中,大小写是有区别的,例如 max 和 MAX 是两个不同的标识符。

(3)标识符虽然可由程序员随意定义,但标识符是用于标识某个量的符号,因此,命名应尽量有相应的意义,以便阅读理解。

(4)保留字不能用做用户定义的标识符。所谓保留字(也称关键字)是由 C 语言规定的具有特定意义的字符串,例如标准输入输出函数名 scanf 和 printf,控制命令字 if、while 等都是系统的保留字。

1.2.5 C 语言程序的基本特点

C 语言程序主要有以下基本特点:

(1)C 语言程序完全是由函数构成的,而且每个程序可由一个或多个函数组成。C 语言程序的函数化结构使得 C 语言程序非常容易实现模块化,便于阅读和维护。

(2)一个源程序不论由多少个函数组成,都有且只能有一个 main( ) 函数,即主函数。

(3)C 语言程序总是从 main( ) 函数开始执行,而不论 main( ) 函数在程序的什么位置,也就是说,可以将 main( ) 函数放在程序的任何位置。

(4)每一个语句都必须以分号结尾,但函数头和花括号"}"之后不能加分号。

(5)C 语言中没有专门的输入、输出语句,输入输出是通过 scanf( ) 和 printf( ) 两个库函数实现的。

(6)标识符、关键字之间必须至少用一个空格进行分隔,以便互相之间区别开来。

(7)C 语言程序对字符的大、小写有严格的区别,标识符的大写形式和小写形式是不一样的。

(8)C 语言程序中可以用"/* … */"对任何部分进行注释。好的程序都要有必要的注释以提高程序的可读性。

1.2.6 C 语言程序的上机实现

在计算机上实现一个 C 语言程序通常包括 4 个阶段:编辑、编译、连接和运行。

(1)编辑(Edit):利用文本编辑软件录入编写的 C 语言程序,并以磁盘文件的形式保存起来,这类文件称为 C 语言的源程序文件,一般使用".c"作为文件扩展名。

(2)编译(Compile):将编辑好的源程序转化成二进制目标代码。计算机只能执行二进制代码,C 语言源程序只有转化成机器指令序列后才能被计算机识别执行,这个过程由专门的编译程序完成。在编译阶段,编译程序首先对源程序进行自动分析,检查程序中存在的语法错误,给出出错报告,编程人员根据报告可以立即对程序进行编辑修改。这个过程往往会多次反复,直到正常编译结束为止。

(3)连接(Link):编译所生成的目标文件(*.obj)是相对独立的模块,还不能直接执行,需要通过连接程序把它和其他目标文件以及系统所提供的库函数进行连接装配,生成可执行文件才能执行。这期间可能出现缺少库函数等连接错误,同样连接程序会报告错误信息。用户根据错误报告再修改源程序,再编译,再连接,直到程序正确无误后形成可执行文件。可执行文件的名字可自由指定,默认的执行文件的名字与源文件的名字一致,可执行文件的扩展名为".exe"。

(4)运行(Run):执行在连接阶段生成的可执行文件,得到运行结果。当然,也可能由于解决问题的算法存在问题而使源程序编写具有逻辑错误,导致错误的运行结果。也可能由于语义上的错误,例如用 0 做除数,使得运行时出现错误。出现这些情况时,就需要对算法进行检查,重新编制源程序,直至得到正确的运行结果。

要上机实现一个 C 语言程序,首先需要安装 C 语言的支持系统。使用较多的 C 语言系统是 Turbo C 2.0,它是一个集成的开发环境,源文件编辑、编译、连接及运行等,都可在这个环境下进行。当然,目前广为流行的面向对象的语言系统 Visual C ++ 6.0 也支持 C 语言程序。以下是使用 TC 2.0 实现 C 语言程序的基本方法:

(1)启动 TC 2.0,进入如图 1 -2 所示的 TC 2.0 集成环境窗口。窗口上方的命令可以通过按下 Alt 键和命令首字母的组合形式去调用它。

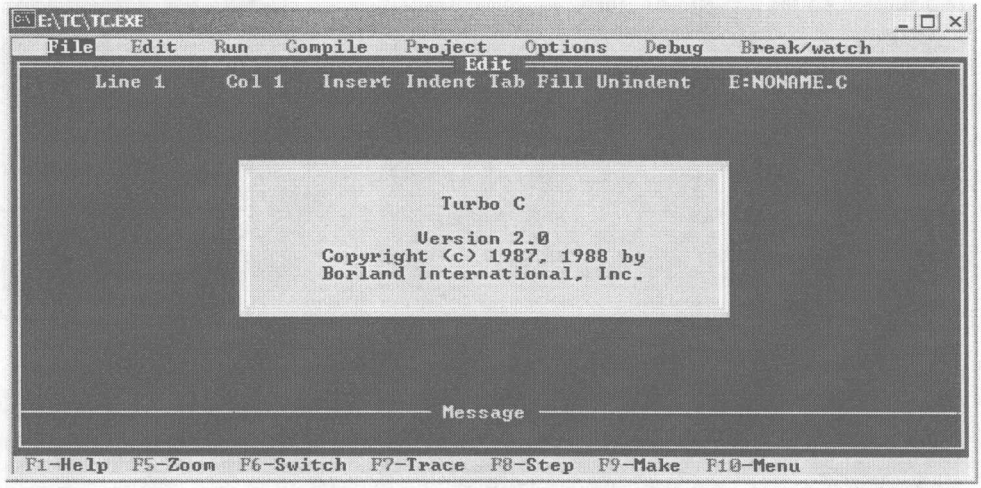


图 1 -2　TC 2.0 集成环境窗口

(2)使用“Edit”命令,使 TC 集成环境进入编辑状态,录入源程序,如图 1 -3 所示。

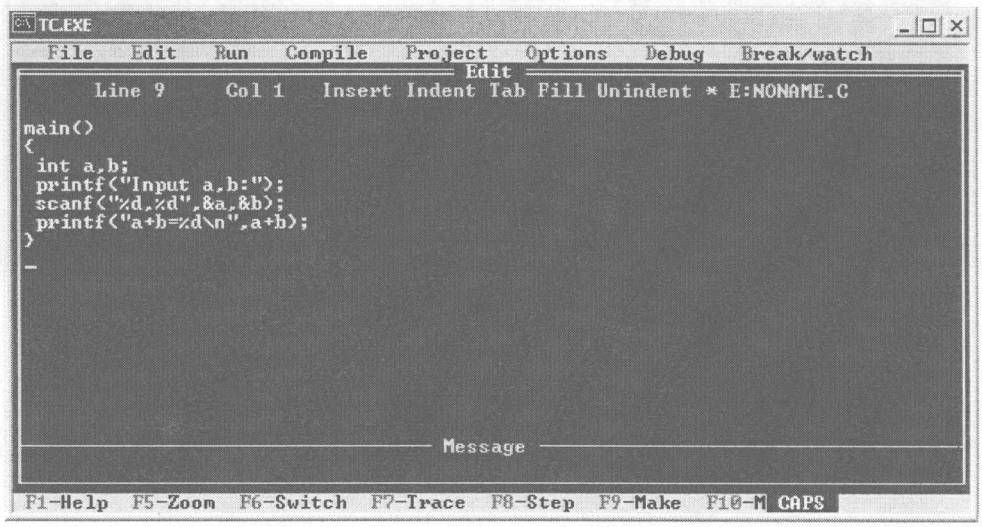


图 1 -3　“Edit”命令的编辑窗口

(3)使用“Compile”命令,对源程序进行编译、连接,生成可执行文件,图 1 -4 为“Compile”命令的窗口。选择“Compile”的“Compile to OBJ”,则进行编译,得到一个后缀为 . obj 的目标程序;然后再选择“Compile”的“Link EXE file”, 进行连接操作,可得到一个后缀为 . exe 的可执行文件。也可以使用“Compile”的“Make EXE file” 或按“F9”键,将编译和连接合为一个步骤进行,一次完成编译和连接。在屏幕上会显示编译或连接时有无错误和有几个错误,图 1 -5 为编译完成后的窗口,图 1 -6 为连接生成可执行文件后的窗口。

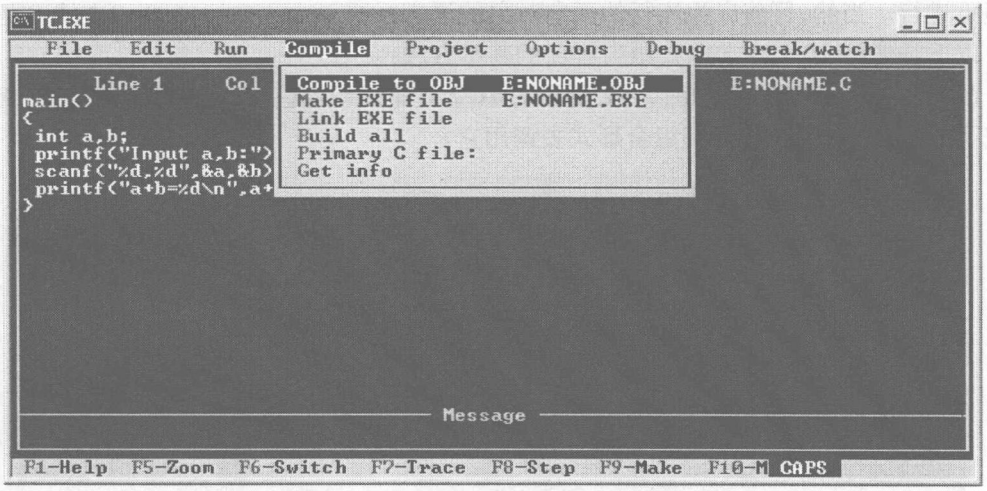


图 1 -4 为“Compile”命令的窗口

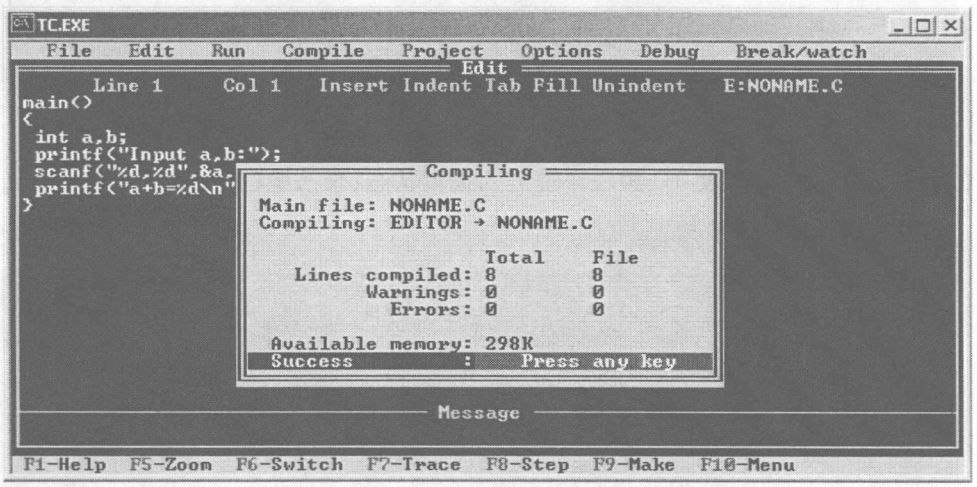


图 1 -5 编译完成后的窗口

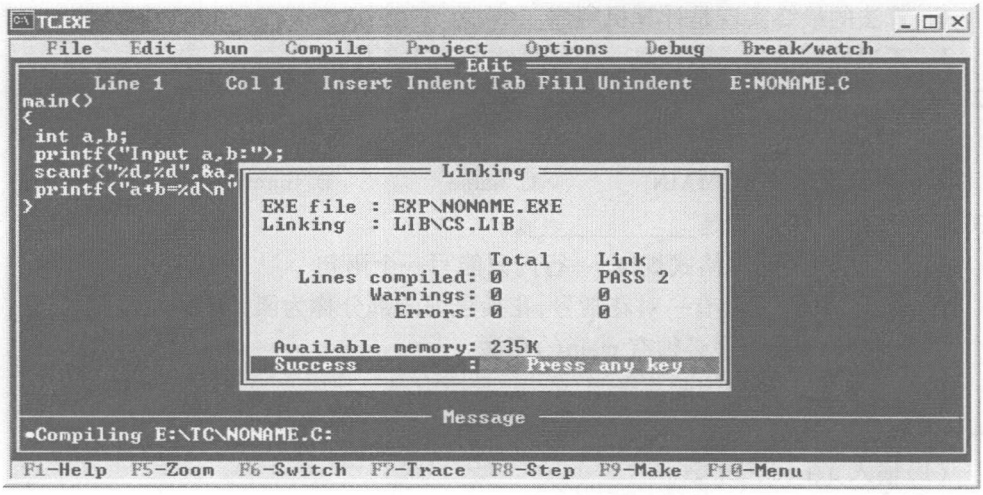


图 1-6　连接生成可执行文件后的窗口

(4)使用“Run”命令运行程序。Alt + F5 组合键可以实现程序窗口和运行结果窗口的切换,是查看程序运行结果时常用的一个控制键。

本章小结

算法是解决具体问题的方法,一个问题可以有不同的算法,算法中的每一个步骤都必须有确切的含义,一个算法在经过有限步骤之后能够结束。对算法的描述方法有多种,程序流程图是常用的算法描述方法。

程序的错误有两种,即语法错误和逻辑错误。语法错误在编译阶段能够被系统发现并指出来,而逻辑错误系统无法检查,需要用多种方法进行测试。常用的测试法有黑盒测试法和白盒测试法。

C 语言是一种结构化的高级语言,C 语言程序的突出特点是函数化结构。任何一个 C 语言程序都是由若干个函数构成的,而且有且仅有一个称为主函数的 main( )函数,即当程序只有一个函数时,该函数必然是 main( )函数。任何函数都具有如下的一般结构:

```
函数名( )
{
函数体
}
```

在 C 语言程序中,标识符的大小写形式具有不等价性,如变量名 NAME 和 name 是两个不同的变量。保留字不能用做用户定义的标识符。

C 语言程序的上机实现分为编辑、编译、连接和运行四个过程,其中的编辑可以通过任何文本编辑软件实现。

习题 1

一、选择题

1. 以下关于算法的描述不正确的是_____。

A. 任何一个问题,它的实现算法是唯一的

B. 描述算法常用的表达工具有流程图、N - S 图、PAD 图、伪码等

此为试读,需要完整PDF请访问: www.ertongbook.com