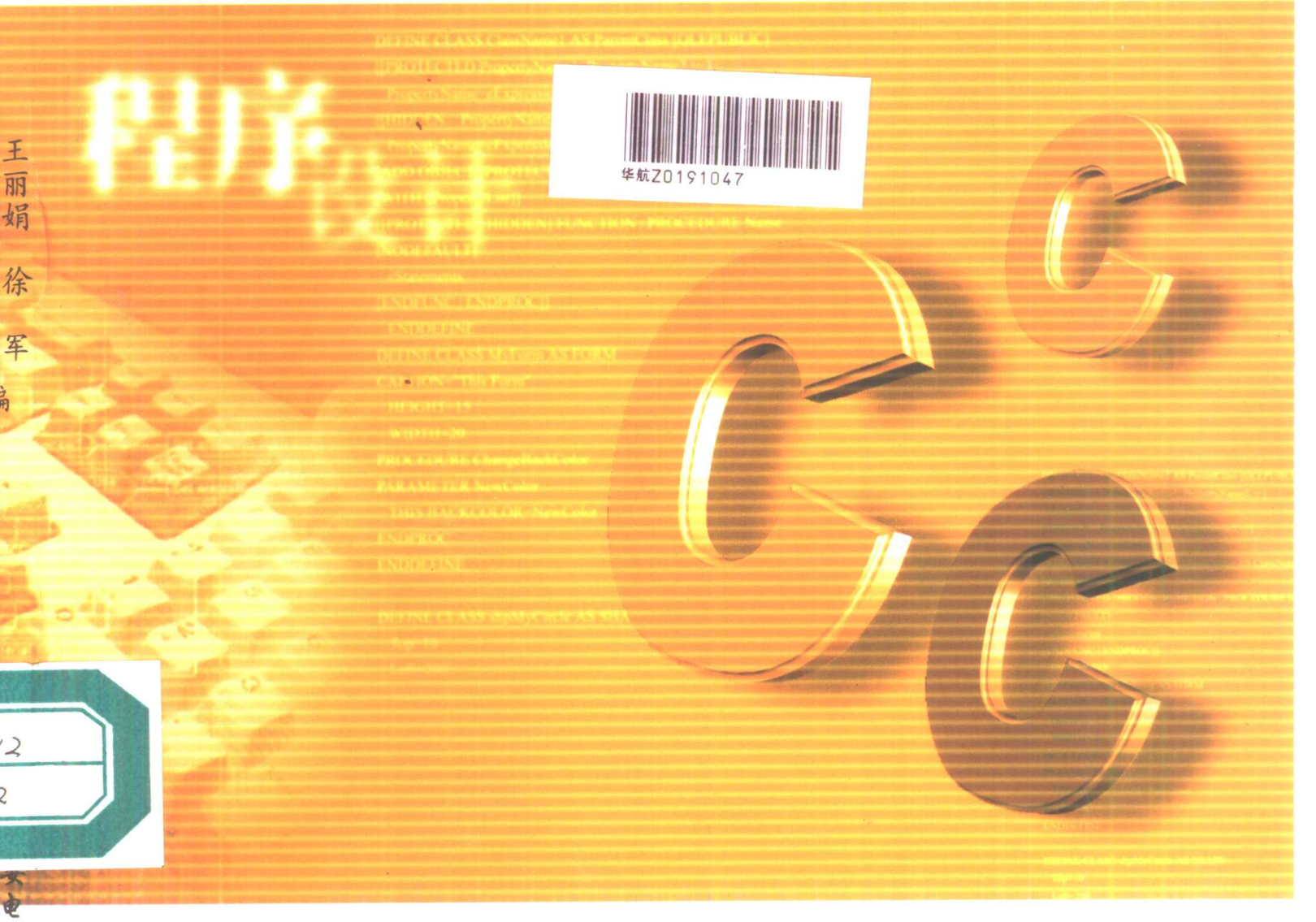
21世纪

高等学校电子信息类系列教材

C程序设计

王丽娟 徐 军
戴宝华 荣 政 编



华航Z0191047

西安电子科技大学出版社
http://www.xduph.com

★ 21世纪高等学校电子信息类系列教材

C程序设计

王丽娟　徐　军
戴宝华　荣　政　编

西安电子科技大学出版社
2000

内 容 简 介

本书系统地介绍了C语言的基本知识及C程序设计。全书共12章，前10章介绍了C语言的基本概念、语法规则以及利用C语言进行程序设计的结构化程序设计方法。第十一章介绍了上机知识，并配有具体查错示例，以加强实践环节。第十二章对C++语言中部分与C相关的概念用对比的方式进行了介绍，并将面向过程的程序设计思想与面向对象的程序设计思想进行了对比阐述，以利于读者今后的学习。本书有配套书《〈C程序设计〉学习指导》，帮助读者深入学习。

凡具有计算机基础知识的读者，都可通过本套书将C语言作为第一门计算机语言进行学习。本套书既可作为大专院校和计算机培训班的教材，又可作为计算机等级考试(二级)的参考用书。

图书在版编目(CIP)数据

C程序设计/王丽娟等编. —西安：西安电子科技大学出版社，2000.8
21世纪高等学校电子信息类系列教材
ISBN 7-5606-0906-6
Ⅰ.C… Ⅱ.王… 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(2000)第37615号

责任编辑 马乐惠 杨宗周
出版发行 西安电子科技大学出版社(西安市太白南路2号)
电 话 (029)8227828 邮 编 710071
http://www.xduph.com E-mail:xdupfxb@pub.xaonline.com
经 销 新华书店
印 刷 陕西画报社印刷厂
版 次 2000年8月第1版 2000年8月第1次印刷
开 本 787毫米×1092毫米 1/16 印张 15
字 数 351千字
印 数 1～8 000册
定 价 18.00元
ISBN 7-5606-0906-6/TP·0482

如有印装问题可调换
本书封面贴有西安电子科技大学出版社的激光防伪标志，无标志者不得销售。

前　言

“C 程序设计”作为计算机文化基础、技术基础和应用基础三个层次中第二层次的一门主要课程，是所有理工科高校的必修课，又是全国计算机等级考试二级的主要语言。并且以其为核心的C＋＋是目前广为流行的面向对象程序设计的主要语言之一，所以C语言已成为广大计算机应用人员和计算机爱好者、初学者的必学语言。

在多年非计算机专业的计算机系列课程教学工作中，我们深切地感到多数学习者总停留在“学会C语言的基本语法、理论，编写简单的C语言程序，通过书面考试”的水平上。实际上C语言课程的学习过程是一个很好的机会，如果能利用好这个过程，它的意义远大于仅学会一种计算机语言，而是培养了良好的编程习惯和工作作风，提高了独立思考问题、解决问题的能力，为后续课程的学习和今后的工作打下良好的基础。

为此，我们编写了这套书，试图从三个方面实现上述目标：

一、面向初学者，使略有计算机基础知识的人都能较容易地“学会”用C语言编程。

本书以“浅显易懂、简练清晰”为原则，突出重点内容，回避了许多容易使初学者困惑，易产生副作用的繁琐细节和一些繁难的内容。本书更注重易用性、实践性。所有程序书写形式严谨、细节考虑周到，并全部在 Turbo C 系统下调试通过。另外，以专门一章详述上机操作，重点介绍了调试查错的思想方法和具体步骤，使学习者能通过计算机语言学习的捷径——在上机编程实践中，真正学会C语言程序设计。

二、在学会的基础上，使学习者能发挥C语言的特长，编写出优化的、符合现代程序模式的“好”程序。

本书将结构化程序设计的思想贯穿于始终，用了相当大的篇幅帮助学习者建立起结构化程序设计的基本思想，指导C的编程实践。使学习者有一个高的起点，少走弯路，不仅能写对程序，而且能辨别程序的优劣，写出好程序，并能上机顺利实现。

我们同时编写了较有特色的配套书《〈C 程序设计〉学习指导》，帮助学习者更好地学习。

三、向C＋＋的过渡。本书第十二章介绍了如何从C转入C＋＋，并对“类”和相关概念以及面向过程与面向对象的程序设计思想做了对比介绍。希望学习者在学完之后，对C＋＋ 有一个感性的认识，对今后的C＋＋的学习有一定的帮助。

全书由王丽娟主编并编写第九、十、十二章，第一、二、三、四章由戴宝华编写，第五、六及第十一章由荣政编写，第七、八章由徐军编写，王长山担任主审。

希望这套书能给众多C语言的学习者以切实的帮助。由于我们水平有限，时间仓促，其中必有不足之处，殷切盼望读者能提出宝贵的意见。

编者

2000 年 6 月

目　录

此为试读，需要完整PDF请访问：www.ertongbook.com

第一章

C 语 言 概 述

1.1 C 语言的发展简史与特色

1.1.1 C 语言发展简史

C 语言是目前世界上最广泛使用的通用计算机语言。用它既可编写计算机系统软件，也可编写各种应用软件，所以在数百种计算机语言中，C 语言仍然是目前最流行、最受欢迎的计算机语言。

最初，C 语言是为写 UNIX 多用户、多任务计算机操作系统而设计的。1971 年，美国电话与电报公司(AT&T)贝尔实验室的丹尼斯·里奇(Dennis Ritchie)先生，在 B 语言的基础上写成了 C 语言，该语言于 1972 年投入了使用。1973 年他又同肯·汤普森(Ken Thompson)用 C 语言重写了 UNIX 操作系统(原系统由汇编语言编写)，使之成了以后各种版本 UNIX 的发展基础，之后，C 语言发展成为通用的程序设计语言。1983 年美国国家标准化协会 ANSI 为了使 C 语言得到更快更好的发展，成立了一个委员会，制定了 C 语言标准，称为 ANSI C。1987 年又公布了新标准——87ANSI C。目前流行的各种 C 版本都是以这个标准为基础的。

现在，Windows 已成为计算机的主要操作系统，相应的基于 Windows 的程序开发多采用 C++，它虽是一种面向对象的语言，但其核心内容仍是标准 C。

1.1.2 C 语言的特色

C 语言能得到快速的发展，受到用户的广泛欢迎，是因为它独具特色。

1. C 与其它语言的比较

作为高级语言的 C 语言常被称为中级计算机语言。这并不意味着 C 语言功能差难以使用，或不像某些高级语言那样完善，也不是说使用 C 语言编程会涉及到使用汇编语言时所出现的繁琐的问题。C 语言被称为中级语言，只是意味着它把其它高级语言的基本结构与低级语言的实用性结合了起来。

1) C 与汇编语言比较

C 语言允许对位、字节和地址进行操作(指针)，这三者是计算机最基本的工作单元，

在编制系统程序时要经常用到，所以它适用于写系统程序。由于汇编语言是非结构化语言，含有大量的跳转、子程序调用以及变址，这种结构的缺陷使得汇编语言程序难以读懂，难以维护，也不能移植。而C语言的结构化、模块化克服了汇编程序难读、难维护的缺点。C语言又具有汇编语言的功能，目标代码长度也差不多，效率几乎与汇编相近，且具有很好的可移植性。

2) C与其它高级语言比较

C有丰富的运算符，达34种(见2.5节)，其中有很多运算符对应于常用的机器指令，比如++等可直接编译成机器代码，使用起来简单精练。

C有多样化的表达式类型，因此可将C语言说成是表达式语言。C语言中的表达式几乎无处不在，而在其它高级语言中，表达式则要受到很大的限制。

C的数据类型丰富，具有现代语言的各种数据结构。C的数据类型有：整型，实型，字符型，数组，指针，结构体，共用体等。它们能用来实现各种复杂的数据结构：链表，树，队列，栈等。其中指针类型使参数传递简单、迅速，节省内存，其它高级语言中除Pascal语言外，均无指针数据类型，而Pascal语言的编译器也是用C语言写的。

C的输入输出使用的是数据流，而其它高级语言使用的是记录。

C程序生成的机器代码质量高，内存占用少，运行速度快，程序执行效率高。这是衡量一种语言优劣的重要指标之一。如用FORTRAN等语言编程生成的代码长，占用内存多，不能用于实时操作。

2. C是结构化语言

C语言是以函数为模块来编写源程序的，所以C程序是模块化的。

C语言具有结构化的控制语句，如if～else语句，switch语句，while语句，do～while语句，for语句等。因此是结构化的理想语言，符合现代编程风格的要求。

结构化语言的一个显著特点是代码和数据的分隔化，即代码和数据分开存贮，互相隔离；程序的各个部分除了必要的信息交流外，彼此互不影响，相互隔离。

实现分隔化的一个方法是使用若干子程序(函数)，在子程序中分别定义各自的局部(暂时)变量。由于使用局部变量，编程者能保证子程序(函数)运行时不会对程序的其它部分产生副作用。

由于C语言具有结构化语言的这些特点，程序之间很容易实现程序段的共享。如你编制了一个分隔化的函数，仅仅需要知道如何去调用它，以及它实现的是什么功能，而无须知道其功能是如何实现的。请注意：过多地使用全局变量(在整个程序中都可使用的变量)可能会由于副作用，使错误蔓延到整个程序。

3. C是编程者的语言

C语言的其它主要优点如下：

C语言简洁、紧凑，使用方便灵活；一共只有32个关键字(27个来自kerninghan和Ritchie的标准，5个由ANSI标准委员会增补)，9种控制语句，它们构成了C语言的全部指令；程序书写形式自由，压缩了一切不必要的成分。

C语言很少限制、很少缺陷、模块结构、彼此独立的函数和一些十分紧凑的关键字，使得C语言能达到接近汇编语言的高效率和广泛的应用范围，所以在许多情况下它是编程者首选的计算机语言。

4. C 的“缺点”

1）语法限制不严格

C 程序在运行时不做诸如数组边界检查和变量类型兼容性检查等错误检查，而是由编程者自己保证程序的正确性。故初学者在编程过程中应给予相当的重视，否则很容易发生错误。

2）程序设计自由度大

C 语言虽有五种固有的数据类型，但不像 Pascal、Ada 语言那样有很强的类型区分。实际上 C 语言允许几乎所有数据类型的转换。例如在大部分表达式中，字符型和整型数据都可自由地混合使用；所有类型均可作逻辑型(非零为真，零为假)；可自己定义新的类型(typedef)等，还可以把某类型强制转换为指定的类型，编程者可自由发挥。

以上这些“缺点”实际上也是 C 语言的特点，它可使程序员有更大的自主性，能编出灵活、优质的程序。但对于初学者来说这些特点却不易掌握，似乎是“缺点”，只有在熟练掌握 C 语言程序设计之后，才能体会出其灵活的特性。正是 C 语言的这种灵活性使它的适应性更强，应用范围更广，成为目前最流行的语言。

1.2 简单的 C 程序介绍

例 1.1 打印一个语句。

程序：

```
main( )
  { printf("A simple c program. \n");
  }
```

经编译后运行结果如下：

A simple c program.

程序中 main 表示“主函数”，每一个 C 程序都必须有一个 main 函数。函数体由花括号{ }括起来。本例 main 函数内的 printf 是 C 语言中的标准输出函数调用语句，而语句最后必须有一个分号。双引号内的字符串原样输出，“\n”是换行符，即在输出“A simple c program. ”后回车换行。

例 1.2 求两数中的小者。

程序：

```
main( )                                  /* 主函数 */
  { int a , b, c;                        /* 定义变量 */
    scanf("%d, %d", &a, &b);             /* 输入 */
    c=min(a, b);                         /* 调用函数 */
    print("min=%d\n", c);                /* 输出 */
  }
  int min(int x, int y)                  /* 定义函数 */
      { int z;                           /* 定义局部变量 */
        if (x>y)          z=y;
```

```
    else              z=x;
    return(z);                          /* 返回z值*/
}
```

经编译后运行如下：

14，4↙

min=4

再次运行：

6，29↙

min=6

本程序包括两个函数：主函数 main 和子函数 min。

min 函数的作用是将 x 和 y 中较小者的值赋给变量 z，并通过 return 语句将 z 的值返回。变量 x，y 是形式参数，在调用 min 函数时，由实参 a，b 分别对应传送给 x，y。变量 z 是局部变量，仅在 min 函数内有效，z 的值是通过函数名 min 返回给主调函数，并赋给变量 c。

在主函数 main 中，int a，b，c；是变量的定义部分，说明变量 a，b，c 是整型变量，a，b，c 也是局部变量，仅在主函数内有效。scanf 是标准输入函数，在此为输入函数调用语句，因此最后以分号结束。该 scanf 函数的作用是输入 a 和 b 的值，&a 和 &b 表示存放变量 a 和变量 b 的地址，键入的值将按该地址送到相应内存单元中存贮。而“%d，%d”仅指定了输入变量 a 和变量 b 的格式，%d 表示按十进制整数类型格式输入，两个%d 之间的逗号表示实际上机输入两个整型数时，两数之间也要用逗号分隔。printf 是标准输出函数，双引号内为输出控制字符串，其中“min=”表示原样输出，“%d”表示把 c 变量的值按十进制整型数输出，“\n”表示最后要回车换行。

在程序中各行后面的“/* … */”是注释部分，它可以出现在任何位置，但必须配对使用，在/和 * 之间是不能出现空格的，计算机系统将注释部分忽略不予编译。注释是给编程者或用户看的，因此可以是英文，也可以是中文，为了便于理解，我们采用中文来注释。在调试过程中，有时也用注释符暂时“删掉”某段程序，以缩小查错范围。

本例中出现了许多概念：形参、实参、函数返回值、取地址等等。我们只作了简单的解释，如不理解，可先不予深究，学到后续章节时，问题就能迎刃而解。

通过上述两个例子，我们可以看到：

(1) C 程序是由函数构成的。一个 C 程序至少包含一个 main 函数，也可以包含一个 main 函数和若干个其它函数。因此函数是 C 程序的基本单位，C 又被称为函数式语言。被调用的函数可以是用户自编的(如 min 函数)，也可以是由系统提供的(如 scanf 和 printf)。所有的函数又是独立的，这种特点使得程序的模块化非常容易实现。

(2) 一个函数由两部分组成：函数的首部和函数体。函数的首部就是每个函数的第一行，包括函数的类型、函数名和形参(及形参的类型说明)，函数名后的形参必须用一对圆括号括起，如没有形参时(如 main 函数)圆括号也不能省略，它已成为函数的标志。

函数体由一对花括号{ }括起。函数体一般包括说明部分(包括变量定义和函数声明)和函数执行部分。在某些情况下可以没有说明部分(如例 1.1)，有时甚至可以既无说明部分也无执行部分。如：

此为试读，需要完整PDF请访问：www.ertongbook.com

dummy( ) { }

它是一个空函数，什么也不干，但这是合法的，并且在开始开发一个程序时，也是非常有用的。我们常用空函数来留出一个接口，以便以后可换成其它函数(见例 4.7)。

(3) main 函数通常位于程序之首，实际上它位于程序的开头、最后及函数与函数之间均是合法的，但不管在什么位置，一个 C 程序总是从 main 函数开始执行的。

(4) C 程序书写格式自由，一行内可以写几个语句，一个语句可以分写在多行上。每个语句和数据定义的最后必须有一个分号，分号是 C 语句的必要组成部分。

(5) C 语言本身没有输入输出语句。输入和输出操作是由调用系统提供的标准输入输出函数(如 scanf 和 printf)来完成的。

(6) 可以用/* … */对 C 程序中的任何部分作注释，它可增加程序的可读性。

1.3 用 C 语言解决实际问题的步骤

1.3.1 一个实例的求解过程

本节我们以计算面积的实际问题为例，说明用 C 语言编程解决实际问题的步骤。

1. 问题提出

计算 y=f(x)曲线，x 轴(y=0)，x=a，x=b 四条曲线所围成的面积。参看图 1.1。

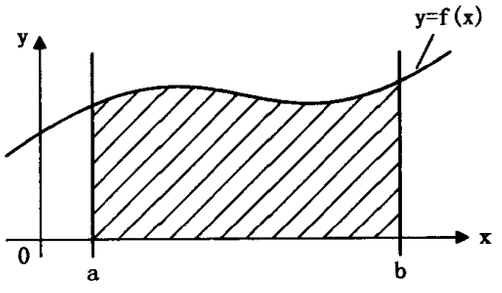


图 1.1 计算面积

2. 分析问题

用户需求：计算四条曲线所围的面积。

已知条件：四条曲线(即 x=a，x=b，y=0，y=f(x))。

需要进行的处理：计算面积，注意误差和精度。

需要用到的软、硬件环境：用 C 语言编程，在个人微机上运行，使用 Turbo C 编译系统。

进行可行性分析：用数值计算方法完全能够实现。

经过分析，我们已做到心中有数。

3. 确定处理方案

根据实际问题选用适当的数学模型(本例属科学计算)。根据高等数学中学到的知识，我们提出的数学模型是 $S=\int_a^b f(x)dx$，也就是在[a，b]区间内求 f(x)函数的定积分。

4. 根据处理方案确定操作步骤

有了数学模型，当f(x)为一个复杂函数时，用解析法将难以求解。我们可选用适当的数值计算方法：例如求解常微分方程组可选用龙格—库塔方法，求解偏微分方程可以选用差分法或有限元法。关于数值计算则属于“数值分析”或“计算方法”学科研究的范围。在本例中我们采用求定积分的基本方法：将该面积分成若干小块，计算每小块的面积，我们采用矩形法，然后将小面积累加起来，参看图 1.2。

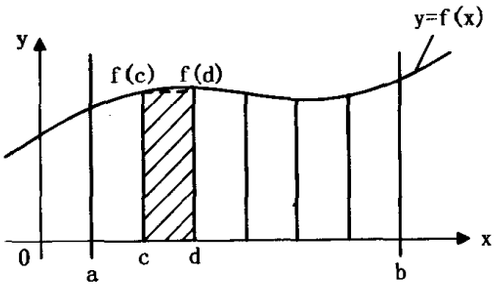


图 1.2 矩形法求积分

在图 1.2 中画阴影线的小面积有 4 条边，第一条长度 f(c)，第二条长度 f(d)，第三条长度 d−c，而第四条线我们采用近似的办法：以第一条线高度作 x 轴的平行线得 y=f(c)，则小矩形面积为(d−c)·f(c)；或以第二条线高度作 x 轴平行线得 y=f(d)，则小矩形面积为(d−c)·f(d)。然后将各小块面积累加起来。以上过程归纳起来就得出了近似公式，最后得到一个操作步骤，也就是算法，如下所示：

(1) 读 a，b 和 n(分成 n 块)。

(2) 求高 h=(b−a)/n。

(3) s=0，面积的初始值为 0。

(4) i=1，先计算第一个小面积。

(5) s=s+h*f(a+i*h)，用后点计算；或 s=s+h*f(a+(i−1)*h)，用前点计算，小面积累加起来。

(6) i=i+1，取下一个小面积。

(7) 如 i<=n 转(5)，否则打印 s。

5. 根据操作步骤编写源程序

程序：

```
main( )
  { float a, b, h, s;
    float f(float);
    int i, n;
    scanf("%f, &f, %d", &a, &b, &n);
    h=(b−a)/n;
    s=0;
    i=1;
```

```
    while(i<=n)
      { s=s+h*f(a+(i-1)*h);                    /* 用前点计算 */
        i++;
      }
    printf("a=%f, b=%f, n=%d, s=%f\n", a, b, n, s);
}
```

关于 f( )函数可以自己写一个函数，也可以使用标准数学函数，如 sin(x)，cos(x)，exp(x)等。如果是标准数学库函数，则在第一行前应加上

#include <math. h>

将第三行 float f(float)；去掉，程序中函数名 f 改成标准函数名即可。

6. 输入程序并上机调试

上机输入源程序并调试，直到改正了所有的编译错误和运行错误。在调试过程中应该精心选择典型数据进行测试。本例当计算有误差或精度不足时可加大分块数量，或采用其它计算方法：如改为梯形法或抛物线法(即辛普生法)等。要避免因调试数据不能反映实际数据的特征而引起的计算偏差和运行错误。

7. 整理分析计算结果

写出有关文档资料，并将该结果应用于解决实际问题。

图 1.3 表示计算机处理一个实际问题的主要过程：

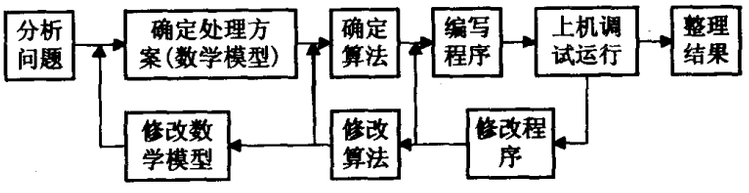


图 1.3 用计算机处理一个实际问题

以上介绍的是一个简单问题的计算操作步骤。如果处理的是一个较大的问题，我们可以采用“自顶向下，逐步细化”的方法，将其分解成若干个小问题，然后采用结构化的方法将其逐一解决。这一点将在第七章中详述。如果处理的是一个庞大复杂的任务，即利用 C 语言开发一个大型应用系统，例如管理信息系统，数据库应用系统，计算机辅助教学系统或实时控制系统等，就需要采用软件工程的方法。这方面的内容已超出本课程的范围，有兴趣的读者可以参看有关软件工程方面的书籍和资料。

1.3.2 算法

1. 什么是算法

为解决一个问题而采取的方法和步骤，就称为“算法”。对同一个问题，可以有不同的解题方法和步骤，也就有不同的算法。例如求 $\int_a^b f(x)dx$ 可采用矩形法，梯形法，辛普生法(即抛物线法)等。为了有效地解题，不仅需要保证算法正确，还要考虑算法的质量，选择较好的算法，或进行算法的优化。

2. 算法的特点

一个算法，必须具有以下特点：

(1) 仅有有限的操作步骤，即“有穷性”(无死循环)。

(2) 算法的每一个步骤应当是确定的，即无“二义性”。

(3) 有适当的输入，即有确定的条件。

(4) 有输出结果。没有输出的算法是无意义的。

(5) 算法中的每一个步骤都应当有效执行(无死语句)。

3. 算法的重要性

掌握最基本的、常用的算法是很重要的，算法设计是整个程序设计的核心。著名计算机科学家沃思(Wirth)曾提出一个公式：程序＝算法＋数据结构。对初学者而言，数据结构即是语言提供的各种数据类型，无大难点。这样编程的任务主要是考虑算法问题，当然还要学会基本的语句和语法。对于许多学习者，往往满足于学会基本的语法和语句，而疏于算法优化的考虑。实际上，语言是千变万化的，即使同一门语言，也是不断改进的，算法才是根本，养成了讲究算法的好习惯，不管用什么语言编程，都会很快编出好的程序。所以建议读者每编一个程序之前要尽量选择好的算法，之后要反复琢磨自己编的程序，尽量优化改进之。本书后续程序举例及配套的《〈C 程序设计〉学习指导》中多处着重介绍了算法的选择和优化，初学者可模仿之予以重点学习。

4. 算法的表示方法

算法的表示方法很多，我们经常采用以下方式：

1) 用自然语言表示

自然语言可以是中文，英文，数学表达式等等。用自然语言表示通俗易懂，缺点是可能文字冗长，不太严格，表达分支和循环的结构不很方便。本章 1.3.1 节一个实例的求解过程 4 即是用自然语言书写的算法。

2) 用传统流程图表示

传统流程图可用一些图框和流程线来表示各种类型的操作。优点是直观形象，易于理解，缺点是占用篇幅较多，画传统流程图既费时又不方便，也不易修改。

标准规定的传统流程图常用符号如图 1.4 所示。

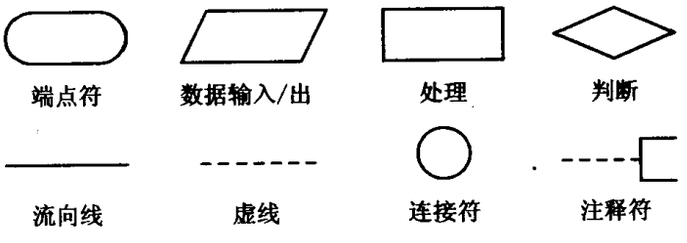


图 1.4　传统流程图常用符号

例 1.3　求 $\int_a^b f(x)dx$，用传统流程图表示，见图 1.5。

在画流向线时，向下和向右不必加箭头，否则须加箭头表示流程方向。

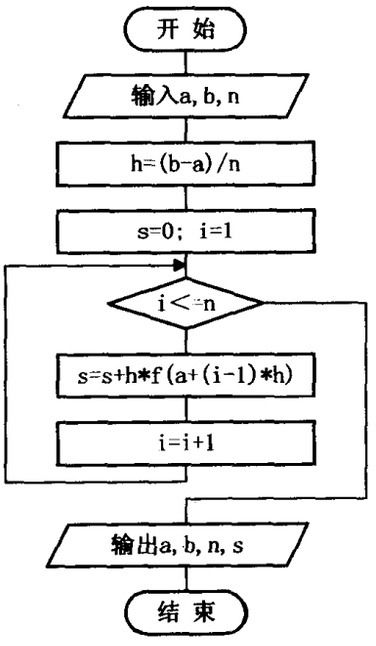


图 1.5　矩形法求积分

3）用 N－S 流程图表示

1973 年美国学者 I. Nassi 和 B. Shneiderman 提出了一种新的流程图形式。在这种流程图中，完全去掉了带箭头的流程线，全部算法写在一个矩形框内，在该框内还可以包含其它从属于它的框，这种流程图以二人姓命名为 N－S 结构化流程图，它适用于结构化程序设计。具体图形符号将在第三章中介绍。

例 1.4　求 $\int_a^b f(x)dx$，用 N－S 结构化流程图表示，见图 1.6。

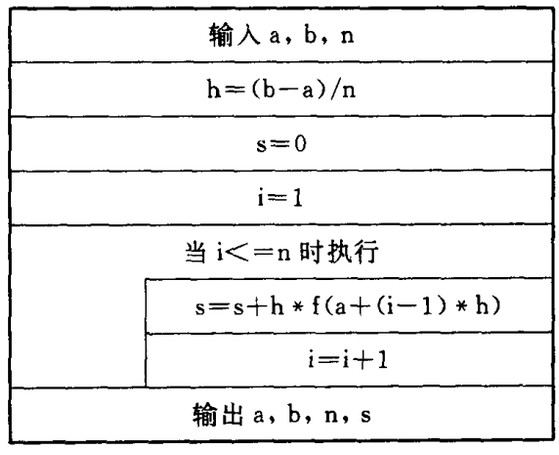


图 1.6　矩形法求积分

4）用伪代码表示

用传统流程图和 N－S 流程图表示算法直观易懂，但画起来费时，而修改流程图也很麻烦。为了设计算法时方便，常用一种称为伪代码的工具。

伪代码是用介于自然语言和计算机语言之间的文字和符号来描述算法的，它不用图形符号，因此书写方便，格式紧凑，也比较好懂，同时也便于向计算机语言算法（即程序）

过渡。

例 1.5 求 $\int_a^b f(x)ds$，用伪代码来表示。

开始(BEGIN)

输入(Input) a，b，n

(b－a)/n⇒h

0⇒s

1⇒i

当(while) i＜＝n 执行(do)

s＋h＊f(a＋(i－1)＊h)⇒s

i＋1⇒i

循环到此结束(End do)

输出(Print) a，b，n，s

算法结束(END)

上面介绍了几种常用的算法表示方法，在程序设计中可以根据需要和习惯选用，在此我们建议选用N－S结构化流程图。当然，简单的题目可以不写算法，直接写出程序。但是刚开始学习编程或编写大程序时，或分析别人写的程序时，最好写出算法，以便于理顺思路。

此为试读，需要完整PDF请访问：www.ertongbook.com