

计 算 机 基 础 教 育 丛 书



谭浩强 主编

C 语言程序设计

高等院校非计算机专业教学用书

周海燕 赵重敏 齐华山 编

科 学 出 版 社
龙 门 书 局



计 算 机 基 础 教 育 丛 书



谭浩强 主编

C 语言程序设计

高等院校非计算机专业教学用书

周海燕 赵重敏 齐华山 编



科 学 出 版 社
龙 门 书 局

内 容 简 介

本书是大学计算机基础教育丛书之一。

本书由十一个单元组成，内容主要包括程序设计的基本常识，C语言的基本概念，设计顺序结构的C程序、带有分支结构的C程序、带有循环结构的C程序，函数，变量的地址及指针操作，数组的概念及应用，字符串，结构体类型和共用体类型，文件以及五个附录。

C语言是目前国内外广泛使用的程序设计语言。本书按照提出问题、分析问题、解决问题的写作原则，以C语言为描述工具，重点介绍了设计程序中具体问题的解决方法，并融入了典型的例题和习题。每道习题均给出了解题指导，以引导读者尽快掌握程序设计方法。在附录中给出了习题的全部参考答案。

本书主要适用于计算机程序设计的初学者，可作为高等院校非计算机专业师生教学、自学用书，同时也可作为社会相关领域培训教材。

需要本书或需要得到技术支持的读者，请与北京海淀083信箱书刊部（邮编100080）联系。电话：010-62562329，010-62531267。传真：010-62579874。网址：www.bhp.com.cn，E-mail：qrh@hope.com.cn。

计算机基础教育丛书

C语言程序设计

谭浩强 主编

周海燕 赵重敏 齐华山 编著

责任编辑：郭淑珍

科学出版社

龙 门 书 局 出版

北京东黄城根北街16号

邮政编码：100717

北京媛明印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

2001年1月第一版

开本：787×1092 1/16

2001年1月第一次印刷

印张：15.25

印数：1—5000

字数：386 000

ISBN 7-03-005814-3/TP·745

定价：21.00元

《计算机基础教育丛书》序

自 80 年代初以来,高等学校中非计算机专业的计算机教育迅速发展起来,最初只是理工类部分专业开设计算机课程,继而很快扩展到农、林、医、经济、管理、师范、文科以至体育、艺术各专业。几乎所有的高校都已开设了计算机课程,并且被列入必修的基础课。计算机知识已成为当代大学生必备的知识了。显然,到 21 世纪,不会使用计算机的人,就不配当大学生。

20 年来,计算机普及和计算机基础教育的内容发生了很大的变化。初期带有一定的扫盲性质,主要是知识性的普及。而现在,则侧重于应用,强调根据需要组织学习内容,学以致用。80 年代,多数学校只开设一门高级语言课(多数是 BASIC 语言),现在大多数学校都根据不同特点设计了计算机基础教育的层次结构,由浅到深、逐步深入地进行学习。计算机基础教育无论在广度上和深度上都比十多年前大大地扩展和提高了。

全国高等院校计算机基础教育研究会自 1984 年成立以来,团结了广大高校教师,深入探索教学目标、课程设置、教学内容和教学办法。在总结许多学校经验的基础上,于 1985 年提出了“按四个层次组织教学”的方案,受到全国高校的欢迎,许多高校按此指导思想规划了课程。经过了十多年的实践,又取得了许多有益的经验。许多学校按照计算机公共基础—计算机技术基础—计算机应用课程这样三个层次开展教学。

一个好的计算机基础教育方案应当具有以下一些特点:

1. 跟踪信息科学技术及应用的发展,面向 21 世纪,体现出先进性;
2. 从国情和校情出发,充分考虑实际需要的可能条件,做到切实可行。切忌照搬外国、照搬计算机专业、照搬外校,体现出可行性;
3. 要有一定的弹性,有必修课,也有选修课,以适应不同专业、不同基础、不同需要的学生,体现出灵活性;
4. 方案应当是动态的,随着发展而不断丰富完善,体现出可拓展性。

在 1998 年举行的全国高等院校计算机基础教育会'98 学术年会上,研究会课程建设委员会向大会提出了两个参考方案,一个适用于理工科专业,一个偏重于非理工科专业,引起各校的兴趣。许多学校希望提供按照这两个方案编写的教材,以便更好地推动教学。根据大家的要求,我们特委托这两个方案的起草人张基温教授、王启智教授分别组织编写了两套教材,供各校选用和参考。第一批拟出版以下几册:

1. 大学计算机公共基础实用教程
2. 大学计算机公共基础实验教程
3. 大学计算机软件技术基础教程
4. 大学计算机硬件技术基础教程
5. 大学计算机应用基础教程
6. 大学计算机应用实验教程

JS8P2/01

7. 大学计算机技术基础教程
8. 大学计算机技术实验教程
9. C 语言程序设计

应当说明,在教学改革过程中,应当提倡百花齐放,推陈出新。中国如此之大,各校差别如此明显,根本不可能用一本统一的教材包打天下。不同内容、不同风格、不同层次的教材不断涌现,是教育改革深入和学术繁荣的标志。各校可以从不同内容、不同风格、不同层次的教材中选择适合自己需要的教材,在实践中优胜劣汰,学生才能真正用到优秀的教材。

我们组织本丛书只是为了抛砖引玉,希望能有助于教学改革的深入,有助于教材的百花齐放,有助于计算机的普及教育。我们将会根据广大读者的意见把本丛书组织得符合教学的需要,符合广大读者的需要。

全国高等院校计算机基础教育研究会
理事长 谭浩强
2000年12月

前 言

随着我国计算机普及浪潮一浪高过一浪，整个社会的计算机应用水平和技术在不断提高。计算机不仅作为信息时代的工具广泛应用于各行各业，它还代表着一种文化日渐影响着人们的思维方式、工作方式乃至生活方式。

在现代知识结构中，《程序设计》是集中体现上述两方面优势的必修课程。作为应用，它是人与计算机沟通的桥梁。通过编写程序，人们可以将自己的意愿“告诉”计算机，以求得问题的解决；作为文化，它培养人们的抽象思维、逻辑思维以及统筹规划能力，这是信息社会所应具备的素质。

以往，我们在介绍程序设计时，都是以一门具体语言为主线，围绕着语法知识展开的。实质上，设计程序是为了解决问题，是一种缜密思维方式的培养。程序设计的实现要有具体语言为依托，但并不受具体语言的限制。为使初学者能够将注意力集中在掌握程序设计的方法上，本书试图在写作中做一点改革尝试，尽量做到以任务驱动为主导，按照提出问题、分析问题、解决问题的写作原则，导出必要的概念和方法，从而达到掌握知识要点的目的。同时，本书在内容安排上，只将语言本身作为描述算法的工具，以介绍典型算法为侧重点，尽量做到在对算法描述的过程中讨论语法现象。

全书划分成十一个单元，分别介绍了程序设计的基本常识，C语言程序的构造及组成，设计带有顺序、分支、循环和独立模块程序的方法，处理有关数列、二维表、字符串应用问题的方法，以及结构体、文件的应用。其中第一至五单元由周海燕编写，第六、十、十一单元由赵重敏编写，第七至九单元由齐华山编写。谭浩强教授审阅了全书并提出了宝贵的意见。由于作者水平所限，书中难免不妥之处，恳请读者批评指正。

作者

2001.1

目 录

| | |
|-------------------------------------|--------------------------------------|
| 第一单元 程序设计的基本知识..... 1 | 任务三 解决多种循环问题..... 61 |
| 任务一 建立程序和程序设计的初步概念... 1 | 任务四 学会简单的程序调试方法..... 64 |
| 任务二 了解算法及其在程序设计中的 重要地位..... 2 | 练习与指导..... 65 |
| 任务三 了解结构化程序设计的思想..... 4 | 第六单元 函数..... 70 |
| 练习与指导..... 8 | 任务一 了解模块化程序设计..... 70 |
| 第二单元 C程序设计的基本知识..... 10 | 任务二 掌握函数的定义方法..... 72 |
| 任务一 了解C语言程序的结构和组成..... 10 | 任务三 掌握带参函数的调用方法..... 73 |
| 任务二 掌握C语言中基本类型常量和 变量的使用..... 13 | 任务四 了解函数的递归调用..... 77 |
| 任务三 掌握算术表达式的运用..... 19 | 任务五 了解变量的作用域..... 83 |
| 任务四 掌握赋值表达式的使用方法..... 21 | 任务六 了解变量的存储类别..... 86 |
| 任务五 掌握自增、自减和逗号运算符 的使用方法..... 22 | 任务七 比较内部变量和外部变量..... 90 |
| 练习与指导..... 23 | 练习与指导..... 90 |
| 第三单元 设计顺序结构的C程序..... 25 | 第七单元 变量的地址及指针操作..... 94 |
| 任务一 了解顺序结构的程序..... 25 | 任务一 建立指针的概念..... 94 |
| 任务二 掌握构成顺序结构程序的基本 语句的使用..... 26 | 任务二 掌握函数调用时的地址传递..... 97 |
| 任务三 了解复合语句和空语句的作用..... 31 | 任务三 掌握有关指针的运算..... 101 |
| 任务四 进行简单的程序设计..... 31 | 练习与指导..... 102 |
| 任务五 掌握Turbo C环境下的简单 上机步骤..... 33 | 第八单元 数组的概念及应用..... 105 |
| 练习与指导..... 36 | 任务一 利用一维数组实现对同类型 批量数据的处理..... 105 |
| 第四单元 设计带有分支结构的C程序..... 38 | 任务二 掌握通过指针对一维数组进行 操作的方法..... 110 |
| 任务一 了解分支结构的程序..... 38 | 任务三 掌握与一维数组有关的参数 传递..... 116 |
| 任务二 掌握判断条件的描述方法..... 38 | 任务四 用多维数组表示按行和列格式 存放的信息表..... 119 |
| 任务三 解决简单分支问题..... 40 | 练习与指导..... 134 |
| 任务四 解决多分支问题..... 43 | 第九单元 字符串..... 138 |
| 任务五 了解程序测试用例的选取方法..... 50 | 任务一 理解字符串的概念..... 138 |
| 练习与指导..... 51 | 任务二 通过一维字符数组来存取 字符串..... 138 |
| 第五单元 设计带有循环结构的C程序..... 53 | 任务三 通过字符指针来操作字符串..... 141 |
| 任务一 了解循环结构的程序..... 53 | 练习与指导..... 147 |
| 任务二 解决单重循环问题..... 53 | |

| | |
|-------------------------------|-----|
| 第十单元 结构体类型和共用体类型 | 150 |
| 任务一 了解结构体类型..... | 150 |
| 任务二 掌握结构体变量的操作方法..... | 151 |
| 任务三 掌握结构体类型数组的 操作方法..... | 155 |
| 任务四 掌握与结构体有关的参数传递.. | 158 |
| 任务五 了解动态链表..... | 160 |
| 任务六 掌握动态开辟存储单元的方法.. | 162 |
| 任务七 掌握单向动态链表的操作方法.. | 163 |
| 任务八 了解共用体类型..... | 171 |
| 练习与指导..... | 174 |
| 第十一单元 文件 | 176 |
| 任务一 了解文件的概念..... | 176 |
| 任务二 掌握文件操作的方法..... | 176 |

| | |
|----------------------------------|-----|
| 任务三 掌握对文本文件进行单个字符 的读写方法..... | 179 |
| 任务四 掌握对文本文件进行字符串的 读写方法..... | 181 |
| 任务五 掌握文件的格式化读写方式..... | 182 |
| 任务六 掌握文件的数据块读写方式..... | 184 |
| 任务七 了解文件复制方法..... | 185 |
| 任务八 了解文件定位函数..... | 187 |
| 练习与指导..... | 188 |
| 附录一 C语言的关键字 | 190 |
| 附录二 运算符的优先级和结合性 | 191 |
| 附录三 常用字符与ASCII代码对照表 | 192 |
| 附录四 C库函数 | 193 |
| 附录五 练习参考答案 | 197 |

任务一 建立程序和程序设计的初步概念

问题一 什么是程序

“程序”原本并不是计算机学科的专用名词。事实上，我们在做任何事情时都遵循着一定的程序。就拿日常生活中的“炒鸡蛋”来说，便可以用以下的程序来描述：

- (1) 准备好烹炒所需的原料和炊具:鸡蛋、食油、盐、炒锅、容器、筷子。
- (2) 把鸡蛋磕入容器并加适量食盐搅匀。
- (3) 将炒锅放在炉灶上并点燃炉火。
- (4) 倒入适量的食油,待油热后倒入鸡蛋烹炒。
- (5) 当鸡蛋从液体状变为固体状则烹炒结束。

这个程序告诉我们：炒鸡蛋需要哪些连贯的步骤，每一步应该怎样去做。一般地说这些步骤是不能随意颠倒的。

随着计算机的问世和普及，“程序”二字渐渐被专业化了。通常特指能够指挥计算机自动完成某项任务指令的集合。

程序要做的就是对所要实现的事件的描述，这些描述应包含两方面的内容，一个是对要实现的“动作”的描述，称之为“算法”；一个是对这些动作所操作的“对象”的描述，称之为“数据结构”。构成程序所使用的描述语言称作“程序设计语言”。程序设计语言分为机器语言、汇编语言和高级语言。

问题二 程序是如何运行的

我们知道计算机只能识别和处理由0和1组成的二进制的代码和数据，即只能读懂“机器语言”。而目前广为流行的、诸如：QBASIC、PASCAL、C等程序设计语言均属于“高级语言”。这些语言都是使用西方语言文字和各类符号来表达的，从而给人们学习和记忆带来了极大的方便，但计算机却不能直接接受它们的指挥。

用高级语言编写的程序称作“源程序”。把源程序变为机器可以执行的二进制指令有两种途径：一种是解释方式，一种是编译方式。

1. 解释方式

QBASIC语言采用的就是解释执行方式，它的整个处理过程可由图1.1示意。

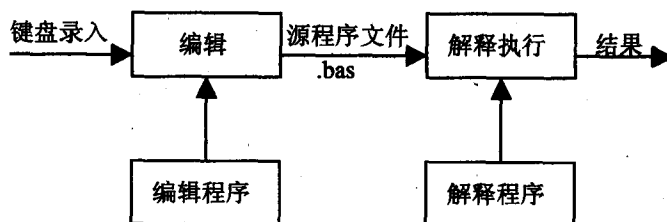


图1.1 解释执行方式

首先通过“编辑程序”将源程序录入到计算机中来建立一个源程序文件(.bas)；然后通过“解释程序”对源程序进行翻译；每翻译一句，就执行一句，如果发现错误则立即

终止解释执行过程，需要修改程序后重新运行。如果程序中没有错误，则一直解释执行完全部程序。最终并不产生翻译后的二进制代码程序。这中间用到的编辑程序和解释程序都属于系统软件。

有的书中将这种方式形象地比作“口译”：说一句，译一句，并不形成译文。

2. 编译方式

C语言采用的就是编译方式。它的整个处理过程见图1.2。

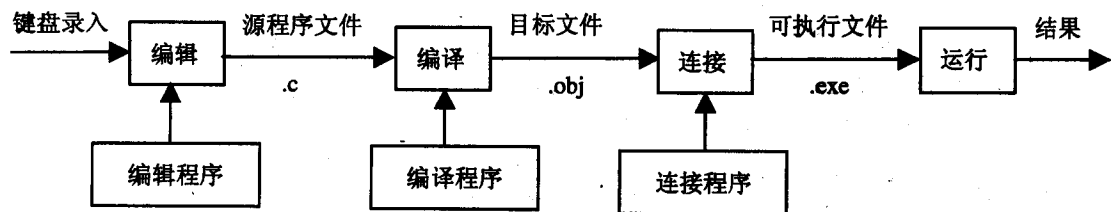


图1.2 编译执行方式

编译方式与解释方式不同。在编译方式下，经过编辑产生了源程序文件(.c)后，并不能立即执行，而先要经过称作“编译程序”的软件，对源程序的语法和结构进行检查，在没有错误的前提下，生成目标程序文件(.obj)。目标程序文件虽然也由二进制代码组成，但仍不可以直接运行（它只是可重定位的目标程序模块），还要由称作“连接程序”的软件，将目标文件与系统提供的函数库以及其他目标程序连接装配成一个完整的可执行程序(.exe)。可执行程序文件完全可以脱离语言环境独立运行。这种方式恰似于“笔译”，只有当反复修改定稿后才形成最终的译文。

在编译方式下用到的编辑程序、编译程序和连接程序都属于系统软件。目前，很多软件都将其集成在一起以方便使用。Turbo C就是这样一种集成开发环境。

问题三 简单程序设计要经历哪些步骤

开发一个简单的应用程序大致要经过以下几个步骤：

1. 确定解题方案。针对具体问题决定采用什么样的数据组织形式来描述问题，数据如何存储等有关数据结构；对于纯数学计算则应选取适当的数学模型；划分程序模块结构。
2. 确定算法。根据选取的数据结构，设计出具体的操作步骤。
3. 编码和调试。选用较为适宜的计算机语言进行编码，编写出源程序文件，并进行调试，改正所有的编译错误和运行错误。
4. 程序测试。要精心设计测试所用的数据（简称测试用例），使得程序对各种合法数据都能有正确的结果，对非法数据给出提示，保证程序的健壮性。
5. 编写文档。为保障应用程序的正确使用，便于维护和修改，应由设计人员提供用户使用说明书、程序技术说明书等文档。

任务二 了解算法及其在程序设计中的重要地位

常听一些初学者反映：学了程序设计语言，自己仍旧不会编写程序。这中间一个重

要的原因，就是没有掌握基本的算法。著名的计算机科学家沃思（N.Wirth）提出过一个经典公式：

程序 = 算法 + 数据结构

也就是说，编写一个程序的关键是合理组织数据和设计算法。数据是程序的核心而算法是灵魂。程序设计语言则仅仅是实现算法的工具。因此，读者在学习程序设计课程的全过程中，始终要重视基本算法的掌握和积累。

问题一 什么是算法

广义地讲，算法是指为解决某个具体问题而采取的方法和步骤。事实上，在日常生活中，我们做每一件事都遵循着一定的算法。在本书中，我们关心的仅限于用计算机来解决问题的算法。

计算机解题算法分为两大类：数值运算算法和非数值运算算法。

数值运算算法解决的是求数值解的问题，例如：用辗转相除法求两个数的最大公约数、牛顿迭代法求方程的根、梯形法求定积分等等。这类算法相对比较成熟，学习的重点就是理解和掌握它。

非数值运算的算法涉及的内容十分广泛，难以规范化。其中某些典型的应用有比较成熟的算法。比如排序算法中有：冒泡法、选择法等等；查找算法中有二分法等等。但很大一部分非数值运算问题，都需要参考已有的类似算法，针对具体问题重新设计。本书将结合实例介绍一些典型的算法，帮助读者理解其中的设计思想和方法，并通过有针对性的练习引导读者举一反三。

问题二 一个算法应具有哪些特点

在设计一个算法时用什么衡量它的正确与否呢？通常，可以用是否满足以下五个特性来界定。

1. 有穷性

一个算法应包含有限个操作步骤，其中每一步都应在合理的时间范围内完成。这个合理的限度没有严格的标准，但要受所处理问题的约束。比如：一个为导弹发射后计算并纠正轨道偏离的应用程序，需要运行半小时后才出结果，这显然就超出了合理的限度。

2. 确定性

算法中的每个步骤都必须是确定的，不允许有“歧义性”。比如：“如果 x 大于等于0，则输出YES；如果 x 小于等于0，则输出NO。在这一描述中，当 x 等于0时，同时满足两种情况，既要输出YES，又要输出NO，这就是不确定的。

3. 有效性

算法中的每个步骤都应该是能够有效地执行的，并能得到确定的结果。比如：对一个负数执行开平方运算，就是无效的算法。

4. 有零个或多个输入

在某些算法中，处理的数据可以通过输入从外界获得，如：计算 $n!$ ， n 的值应该从键盘输入。有些情况下也可以没有输入，如：计算 $5!$ ，常数5是已知的，就不需要输入信息。

5. 有一个或多个输出

算法的实现是以得到运算结果为目的的，因此，没有输出的算法是毫无意义的。比

如：设计“寻找50以内，能同时被3、5、7、11整除的数”这一算法时，若找到了，应该依次输出找到的数值；同时应考虑到：若没找到满足条件的数值，应该输出“未找到”信息。否则，程序没有任何输出，如同不存在一样。

问题三 怎样表示一个算法

在进行算法设计时，可以使用不同的算法描述工具。常见的有：自然语言、传统流程图、N-S图、伪代码、PAD图等。下面我们针对同一个问题，分别介绍使用自然语言、传统流程图、N-S图描述的算法。

待处理的问题：计算 $n!(n \geq 0)$ 。

1. 用自然语言描述算法

(1) 读入 n 的值

(2) 如果 $n < 0$ 则打印“输入错”信息，执行步骤(4)

(3) 如果 $n \geq 0$ 则

① 给存放连乘积的变量 fac 置初值1。

② 给代表乘数的变量 i 置初值1。

③ 进行连乘运算 $fac = fac * i$ 。

④ 乘数增1： $i = i + 1$ 。

⑤ 如果 i 不大于 n ，返回重复执行步骤③，否则按执行步骤⑥。

⑥ 输出 $fac(n!)$ 的值。

(4) 结束算法

从以上介绍可以看出，“自然语言”就是人们日常生活中使用的语言，描述中可以使用汉语、英语和数学符号等。

使用自然语言描述算法通俗易懂，但文字不免冗长，在表达上往往容易出现漏洞，引起理解上的歧义性。所以，一般用于算法较简单的情况。

2. 用传统流程图描述算法

用传统流程图描述的 $n!$ 的算法见图1.3。

可以看出：用传统流程图表示算法直观形象，便于理解。图中所用符号采用美国国家标准协会ANSI规定的通用符号（见图1.4）。

使用传统流程图的缺点是流向线不受约束，使用者可以随心所欲，从而造成阅读和修改上的困难。但它仍是人们乐于使用的工具之一。

3. 用N-S图描述算法

用N-S图描述的 $n!$ 的算法见图1.5。

N-S图是用美国两位学者（J.Nassi和B.Schneidermam）名字的首字母命名的。由他二人提出的这一算法描述工具中，完全取消了流向线，所有算法都以三种基本控制结构（顺序、分支、循环）作为基础，从而可以保证算法的质量。有关内容将在任务三中介绍。

任务三 了解结构化程序设计的思想

问题一 为什么要采用结构化的程序设计方法

在早期的计算机应用中，由于计算机的运算速度慢、内存容量小、硬件价格昂贵，

程序设计人员为尽量节省内存空间、提高运算速度，往往在编程技巧上下功夫，有时会为省去几条语句而使算法变得晦涩难懂。由于当时程序的规模一般都比较小，程序员采用的是手工作坊式的开发方法，这项工作往往被看得很神秘。

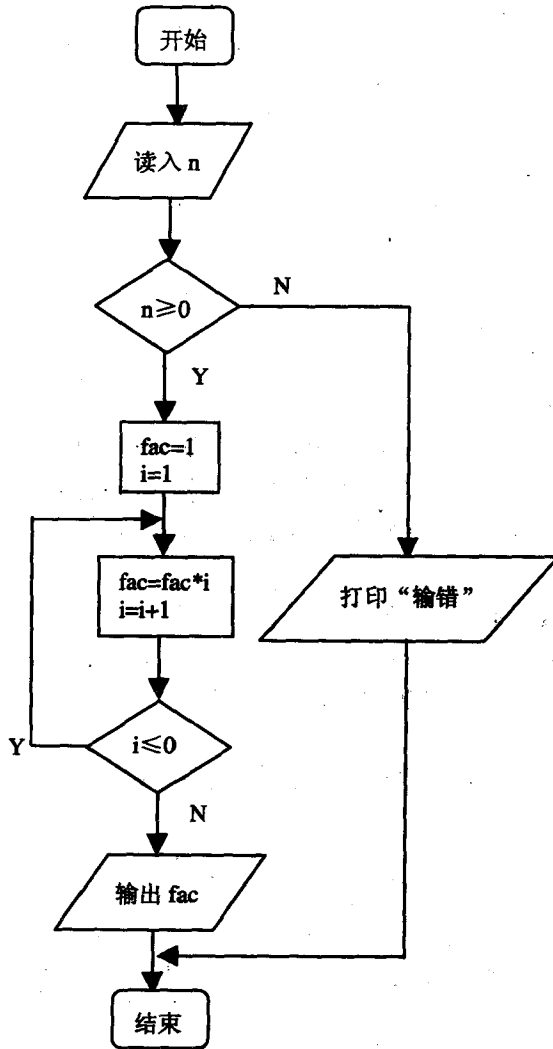


图1.3 计算n!的流程图

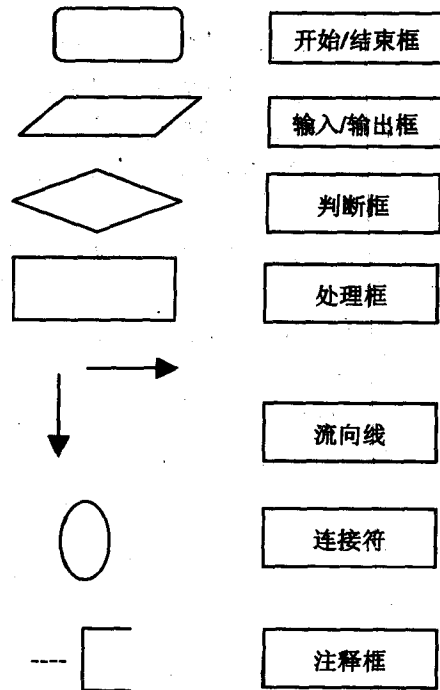


图1.4 流程图使用的基本符号

随着计算机的发展，硬件成本急剧下降，软件要处理的问题越来越复杂，程序的规模就越来越大。一些大型的软件需要数十人、上百人的合作方可完成。由于传统程序设计方法的个人随意性，未考虑到协同工作的需要，使得软件设计周期加长、成本攀升、出错率猛增。以至于许多软件尚未问世，就因故障率太高或已过时而被丢弃。软件的开发制约了计算机应用的发展，从而形成了“软件危机”。

为了解决软件危机，由世界许多著名计算机科学家参加的大讨论，对传统的程序设计方法进行了革命性的改造，并提出了结构化程序设计的思想。其核心是：一个好的程序应该具有层次化的模块结构，并且只使用顺序、分支和循环三种基本结构组合而成。

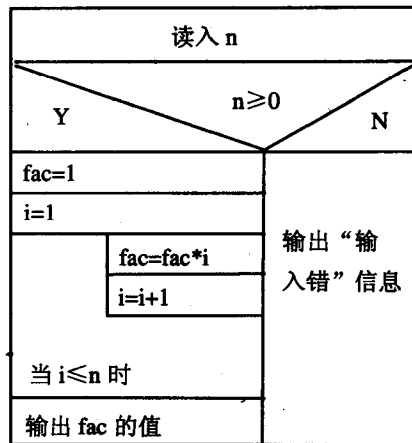


图1.5 计算n!的N-S图

问题二 什么是结构化程序的基本组成结构

结构化程序采用三种基本结构组成，用这三种基本结构的组合可以表示任何复杂的算法结构。为了配合结构化程序设计，美国学者提出了去掉流向线的结构化流程图（N-S图）。下面结合结构化程序设计的三种基本结构来介绍这种流程图的基本构成。

1. 顺序结构

顺序结构是最简单的一种基本结构。这种结构的程序是按照语句出现的先后顺序执行的。

相应的N-S图见1.6。其中A、B两框是按先后顺序执行的。

2. 分支结构

又称作选择结构。分支结构中必定包含一个条件判断，根据给定的条件是否成立来决定是选择执行A框还是B框。相应的N-S图见图1.7。

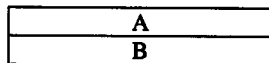


图1.6 顺序结构的N-S图

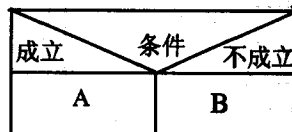


图1.7 分支结构的N-S图

3. 循环结构

又称作重复结构。一般分为当型循环和直到型循环。相应的N-S图见图1.8。它们将根据各自的循环条件来决定重复执行A框的次数。当型循环是在条件成立的情况下执行A框；直到型循环是在条件成立时终止执行A框。这两种循环的判断条件，既可以放在执行A框之前，也可以放在执行A框之后（注：C语言中没有直到型循环）。

由上述三种基本结构组成的结构化程序满足以下基本条件：

- (1) 整个程序只有一个入口和一个出口。
- (2) 没有执行不到的语句。

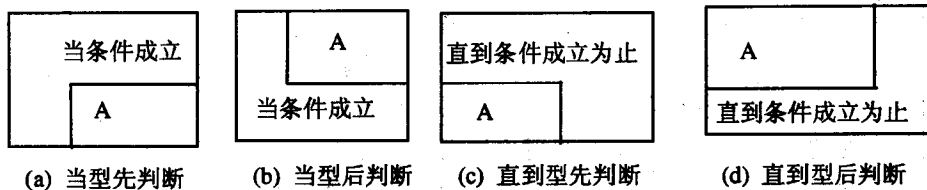


图1.8 循环结构的N-S图

问题三 如何构造出结构化的程序

用三种基本结构组成的程序必定是结构化的。但面对一个较复杂的问题，不可能一步到位，直接设计出用这三种结构描述的算法。为保证获得结构化的程序，就应该采用“自顶向下、逐步细化、模块化”的程序设计方法。

也就是说，在对一个具体问题进行分析的基础上，将其分解成若干个子问题，再分别对具体的子问题进行分解，逐步细化到每个子问题都在易于理解和处理的范围内。这时每个子问题就可以用一个单独的模块实现（在C语言中，一个模块就是一个函数）。从而可以构造出一个层次分明、功能划分明确的结构体系。此时，对于完成单一功能的模块而言，采用三种基本结构描述也就不困难了。

作为示例，请看这样一个简单问题：输入一个班学生的某科成绩，计算出平均得分，输出高于平均分学生的学号及得分。

首先将“成绩统计”问题分解为三个子问题（如图1.9所示）。由于这三个子问题的任务已经明确，功能相对单一，所以可以将一个子问题划分为一个模块。然后再对每个模块进一步细化，画出具体的算法流程图（此处略）。

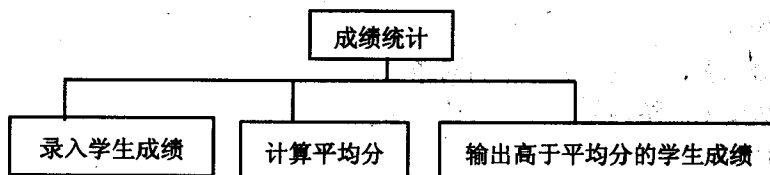


图1.9 模块划分示意图

问题四 如何评价一个程序的优劣

保证程序设计的质量是贯穿整个程序开发过程中不容忽视的问题。那么衡量一个程序优劣的标准是什么呢？国际标准化组织（ISO）软件委员会提出了判别软件质量的标准。依照国际标准，在我国国家标准中，也将有关衡量软件质量的准则归纳为以下6个方面：

- (1) 功能正确性
- (2) 运行可靠性
- (3) 使用方便性
- (4) 效率
- (5) 可维护性
- (6) 可移植性

显然，“功能正确性”是对程序的起码要求。离开了这一基本点，其它条件无从谈起。

“运行可靠性”是与程序内部所潜在的故障密切相关的。当某一种特定条件的组合恰恰执行到了具有隐藏问题的流程时，就会导致运行失败。提高可靠性的有效措施是加强程序测试。这方面的内容在有关软件工程的书籍中有详尽的介绍。

“使用方便性”主要体现在良好的用户界面上。因为大部分的应用程序开发出来是为非计算机专业的广大用户服务的，使用方便的软件才会受欢迎，容易推广。

“效率”通常是指程序的运算速度和内存开销，即“时间经济性”与“资源经济性”。

“可维护性”和“可移植性”与程序中所采用的算法、数据结构、控制结构、语句结构乃至程序风格都有一定的关系。

在这些指标中，程序的效率与可维护性之间有时会产生矛盾。比如：要提高程序的运行速度或节省内存单元，可能要采用比较复杂的算法或引入一些设计技巧，从而必然降低程序的可读性，增加维护的复杂性。在计算机发展的早期，硬件价格高，运行速度慢，这样做是值得的。随着计算机的发展，硬件价格不断下降，计算机的内存容量，运算速度不断提高，再利用这种手段就得不偿失了。因此，目前衡量程序质量的标准已从“效率第一”转变为“清晰第一”。当然，对于某些在时间上有特殊要求的实时操作可以例外。

练习与指导

1.1. 什么是算法？它应具有那些特性？

解题指导：

基本概念题，参见任务二中的问题二。

1.2. 为什么要采用结构化的程序设计方法？

解题指导：

基本概念题，参见任务三中的问题一。

1.3. 结构化程序设计的三种基本结构是什么？基本设计方法是什么？

解题指导：

基本概念题，参见任务三中的问题三。

1.4. 评价一个程序应该从哪几方面入手？

解题指导：

基本概念题，参见任务三中的问题四。

1.5. 是否在任意一台计算机上都可以录入并运行C程序？C语言的源程序要经历哪些操作步骤才能运行？这中间都生成什么文件？

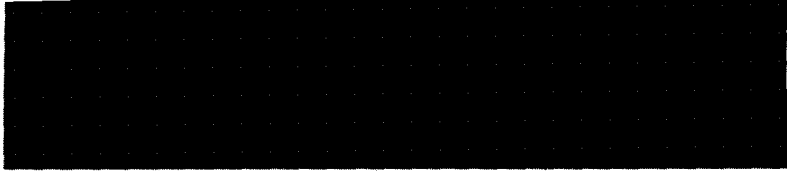
解题指导：

基本概念题，参见任务一中的问题二。

1.6. 请用传统流程图和N-S图，描述“求50以内奇数之和”的算法。

解题指导：

(1) 存放累加和的变量置初值0。

- 
- (2) 加数从1开始每次增2（保证为奇数），循环放入累加和中。
 - (3) 若加数没有超过50，重复进行累加，否则输出累加结果。
 - (4) 传统流程图和N-S图的表示方法参见任务二和任务三。