

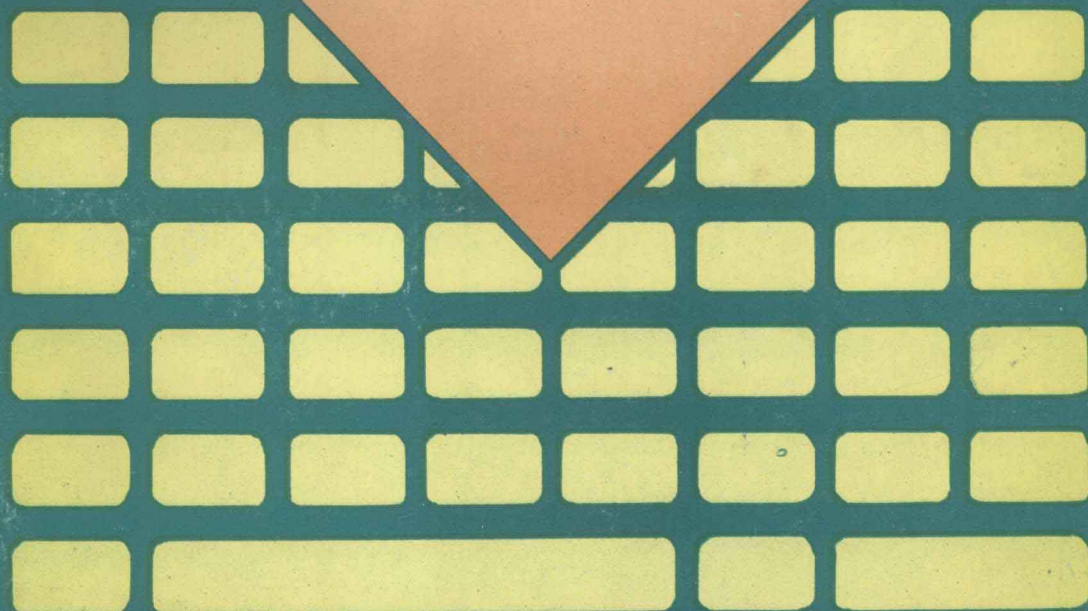
江苏省普通高校计算机等级考试系列教材

TRUE BASIC

程序设计教程

江苏省教育委员会组织编写

孙建国 牛又奇 编著



南京大学出版社

江苏省教育委员会组织编写
江苏省普通高校计算机等级考试系列教材

TRUE BASIC 程序设计教程

孙建国 牛又奇 编著

南京大学出版社

内 容 提 要

True BASIC 语言是一种完全结构化的程序设计语言,它具有简单易学,模块结构清晰,作图功能强,运行效率高等特点。

本书系统地介绍了 True BASIC (V2.03 版)的各种语言成分,通过实例介绍了程序设计的基本方法和技巧。深入浅出,通俗易懂,内容丰富。书中含有大量例题,每章末附有习题,供教学选用。全书内容覆盖《江苏省普通高校非计算机专业学生计算机基础知识和应用能力等级考试大纲》规定的二级 True BASIC 语言考试内容。可作为高等院校非计算机专业学习计算机的教材,也可作为广大科技工作者自学计算机软件的参考书。

江苏省教育委员会组织编写
江苏省普通高校计算机等级考试系列教材
TRUE BASIC 程序设计教程

孙建国 牛又奇 编著

孙志挥 审阅

*

南京大学出版社出版

(南京大学校内 邮政编码 210093)

江苏省新华书店发行 江苏省宜兴印刷厂印刷

*

开本 787×1092 1/16 印张 13 字数 324 千

1996 年 6 月第 1 版 1996 年 6 月第 1 次印刷

印数 1—4000

ISBN7-305-02968-8/TP·151

定价:13.00 元

江苏省普通高校计算机等级考试 系列教材编委会

顾 问 葛锁网 张福炎 邢汉承

主任委员 邱坤荣

副主任委员 陈华生

委 员 (以姓氏笔划为序)

牛又奇 江正战 江邦人 朱 敏

陈凤兰 陈良宽 高岳兴 奚抗生

梅镇武 殷新春 蔡绍稷

工作人员 郭新宇 鞠 勤 钱国兴

出版说明

21 世纪人类将进入信息化社会,对人才的素质和知识能力结构提出了全新的要求,高等学校面临着十分艰巨的任务。当前,计算机知识和能力不仅是本专科学生必备的知识能力,而且已成为高等教育各学科的重要组成部分,并成为衡量人才素质的重要尺度之一。高校计算机基础教育如何主动适应经济建设和社会发展,如何提高教育质量,是一个值得重视和研究的新课题。

江苏省教委为了主动适应经济建设和社会发展以及科技进步的需要,强化跨世纪人才素质的培养,从 1993 年起组织全省普通高校非计算机专业学生进行计算机基础知识和应用能力等级考试,推动了计算机教学内容体系的改革,大面积提高了课程教学质量,1994 年 12 月在江苏省普通高校计算机等级考试指导委员会下面成立了江苏省普通高校计算机等级考试系列教材编委会,根据 1995 年重新修改的《江苏省普通高校非计算机专业学生计算机基础知识和应用能力等级考试大纲》,组织编写、审定一套服务于我省普通高校计算机等级考试的计算机基础教育教材(第一批为六册:《计算机应用基础教程》、《FORTRAN 77 程序设计教程》、《PASCAL 程序设计教程》、《C 程序设计教程》、《TRUE BASIC 程序设计教程》、《数据库(FOXBASE+)教程》),拟在一到两年内分步出版。

这套教材,主要是编者总结了非计算机专业学生计算机基础知识和应用能力等级考试的经验在较长期教学实践基础上成稿,部分书稿是从师生反映较好的讲义中选择修改,其特点为内容丰富,结构严谨,条理清楚,通俗易懂,理论和实践紧密相结合,既可作为普通高校非计算机专业学生学习计算机的一套很好的教材,也可作为广大工程技术人员、科技工作者、管理干部自学计算机知识的宝贵丛书。

这套教材分别由南京大学出版社、东南大学出版社、苏州大学出版社承担出版工作。

我们谨向参与这套教材的编审和出版社编辑表示感谢。限于水平和经验,这套教材还会有许多缺点和疏漏之处,恳请使用这套教材的单位、广大教师和学生提出宝贵的批评或建议,以使这套教材的质量得到进一步提高。

江苏省普通高校计算机
等级考试系列教材编委会

1995 年 6 月

前 言

True BASIC 是 BASIC 语言创始者 J. G. Kemeny 和 T. E. Kurtz 对 BASIC 语言作了重大改进和发展而推出的 BASIC 语言的最新版本。True BASIC 严格遵循美国国家标准 BASIC 的规定。它不仅保留了原 BASIC 语言通俗易懂,易学易用的特点,还是一种完全结构化的程序设计语言,它不仅适用于科学计算、数据处理,还具有丰富的作图和音乐功能。True BASIC 同时提供了解释执行和编译执行两种工作方式。用户可在解释方式下对程序进行编辑和调试,而采用编译方式运行调试好的程序、提高程序运行效率。True BASIC 是一种理想的学习程序设计的语言。

本书以 V2.03 版的 True BASIC 作为蓝本。本教程内容覆盖了《江苏省普通高校非计算机专业学生计算机基础知识和应用能力等级考试大纲》二级中的 True BASIC 语言考试要求的全部内容。

本课程是一门实践性和应用性很强的课程。本书在各章节中编入了大量的经过上机测试的程序设计实例,通过例题深入浅出地介绍如何使用 True BASIC 语言编写各类程序、相关的算法知识及程序设计的方法和技巧。各章备有习题,按照难易程度分为两类,一类为加深对课程内容的理解,一类为提高编程能力,以满足不同专业、不同层次的学生需要。全书分为九章:第一章介绍常用算法和结构化程序设计的基本知识,第二章至第九章介绍了 True BASIC 语言及其程序设计基础。附录中有出错信息及解释, True BASIC 系统命令一览表, True BASIC 系统语法一览表,供同学学习参考使用。

本书由牛又奇编写第一至五章,孙建国编写第六至九章。东南大学孙志挥教授审阅了全部书稿,提出了许多的宝贵意见和建议。在此表示由衷的感谢。

由于编者水平有限,时间仓促,难免存在错误,敬请广大读者批评指正。

编者

1996 年 6 月

目 录

第一章 算法与程序设计方法概论	(1)
§ 1.1 程序与算法	(1)
§ 1.2 结构化程序设计方法	(3)
§ 1.3 算法设计的常用策略	(5)
习题一	(6)
第二章 True BASIC 基础	(7)
§ 2.1 True BASIC 简介	(7)
§ 2.2 True BASIC 的安装、启动与基本操作	(7)
§ 2.3 True BASIC 程序的组成	(17)
§ 2.4 数据	(18)
§ 2.5 标准函数、算术表达式与字符串操作	(19)
习题二	(22)
第三章 顺序结构程序设计	(24)
§ 3.1 程序的一般流程	(24)
§ 3.2 赋值语句与键盘输入语句	(25)
§ 3.3 输出语句	(26)
§ 3.4 读数与置数语句	(32)
§ 3.5 几个简单语句	(34)
习题三	(35)
第四章 分支结构程序设计	(37)
§ 4.1 判别条件的描述方法	(37)
§ 4.2 IF-THEN 结构	(40)
§ 4.3 多分支与 ELSE IF 语句	(43)
§ 4.4 SELECT-CASE 结构	(47)
习题四	(48)
第五章 循环结构程序设计	(50)
§ 5.1 基本循环结构	(50)
§ 5.2 DO 循环	(51)
§ 5.3 FOR 循环	(55)
§ 5.4 程序设计示例	(59)
习题五	(64)
第六章 数组	(67)
§ 6.1 数组与数组元素	(67)
§ 6.2 数组简单应用举例	(72)
§ 6.3 矩阵处理	(83)

习题六.....	(102)
第七章 文件	(105)
§ 7.1 文件的基本概念	(105)
§ 7.2 文件操作的基本语句	(106)
§ 7.3 正文文件	(110)
§ 7.4 记录文件	(121)
§ 7.5 字节文件	(129)
习题七.....	(131)
第八章 自定义函数与子程序	(132)
§ 8.1 自定义函数	(132)
§ 8.2 子程序定义与调用	(144)
§ 8.3 递归程序设计	(150)
§ 8.4 程序的链接	(155)
§ 8.5 应用举例	(156)
习题八.....	(159)
第九章 图形	(163)
§ 9.1 作图环境	(163)
§ 9.2 基本作图语句	(167)
§ 9.3 动画	(173)
§ 9.4 图画子程序与图画变换	(174)
§ 9.5 图形输入和多窗口操作	(178)
习题九.....	(179)
附录 A 出错信息及解释	(180)
附录 B TRUE BASIC 系统命令一览表	(189)
附录 C TRUE BASIC 系统语法一览表	(191)

第一章 算法与程序设计方法概论

§ 1.1 程序与算法

现代电子计算机是在程序的控制下工作的。可以说,没有程序,计算机将一无所能。在具备必要的硬件条件下,计算机可以做某件事或不能做某件事,将取决于有无相应的程序。

程序是什么? 程序就是为解决某一问题而设计的一个指令序列。通常,一个程序由一系列的操作规则和对操作对象的描述(说明)组成,也就是说,人们通过程序告诉计算机做什么(处理对象)和怎么做(操作步骤)。计算机执行程序,就是按规定的步骤处理操作对象,解决特定问题。

要使程序能解决特定的问题,达到预定的目的,就必须保证程序的“正确性”和“可行性”。因此,在设计程序时,首先必须从实际问题出发,考虑计算机的工作特点并确定解决某个问题所需要的方法和步骤,这就是所谓的算法设计。

1.1.1 算法的概念

广义而言,算法就是解决某个问题或处理某件事的方法和步骤。人们在日常生活和工作中做任何事,都必须遵从一定的章法,才能顺利完成。

狭义而言,算法是专指用计算机解决某一问题的方法和步骤。著名计算机科学家 D. E. Knuth 在其《计算机程序设计技巧》一书中为算法所下的定义是:“一个算法,就是一个有穷规则的集合,其中之规则规定了一个解决某一特定类型问题的运算序列。”

计算机算法可以分为两大类:一类是数值计算算法,主要是解决用一般数学解析方法难以处理的一些数学问题,如求解超越方程的根、求定积分、解微分方程等;另一类是非数值计算算法,如对非数值信息的排序、查找等。

研究解决各种特定类型问题的算法已成为一个称作“算法设计”的专门学科。虽然现代电子计算机功能强大,但其基本部件仅能执行诸如数据的传递(输入、输出和取数、存数等)、算术运算(加、减、乘、除)、逻辑运算(与、或、非等)及比较、判断与转移等操作。因此,研究如何通过这些简单操作的组合去解决复杂的问题,也是“算法设计”的重要研究课题。

对于同一问题的求解,往往可以设计出多种不同的算法。不同的算法在运行效率、占用内存的多少等方面有所差异。评价一个算法的好坏优劣,也有不同的角度和标准。一般而言,主要看算法是否正确,运行的效率和占用系统资源的多少等。

1.1.2 算法示例

【例 1.1】 求两个自然数的最大公约数的算法。

- S1. 输入两个自然数 M, N ;
- S2. 求 M 除以 N 的余数 R ;
- S3. 使 $M=N$, 即用 N 代换 M ;
- S4. 使 $N=R$, 即用 R 代换 N ;
- S5. 若 $R \neq 0$, 则重复执行 S2, S3, S4 (循环), 否则转 S6;
- S6. 输出 M , M 即为最大公约数。

本算法是由古希腊数学家欧几里德提出的, 所以又称为“欧几里德算法”。算法中的 S1、S2、S3... 叫算法步骤, 每个算法步骤明确规定所要进行的操作。除过如算法步骤 S5, 根据条件可改变执行顺序外, 各算法步骤都是依排列的次序顺序执行。

欧几里德算法是求两个自然数的最大公约数的经典算法。读者可自行设定两个数对该算法进行测试。

【例 1.2】 在 N 个字符数据中, 查找有无特定的字符串。

- S1. 输入 N 和要查找的数据 S ;
- S2. 使 $I=1$, I 用于计数;
- S3. 读字符数据 X ;
- S4. 若 $X=S$, 输出“找到 S ”信息, 算法结束; 否则转 S5;
- S5. 使 $I=I+1$, 计数器计数;
- S6. 若 $I \leq N$, 则重复执行 S3, S4, S5 (循环); 否则转 S7;
- S7. 输出“找不到”信息。

本算法给出了顺序查找字符信息的方法。

上面的两个算法示例, 第 1 个是数值计算算法, 第 2 个是非数值计算算法。

1.1.3 算法的特征

作为算法, 应具备以下特征:

1) 确定性

算法的每个步骤都应确切无误, 没有歧义性。

2) 可行性

算法的每个步骤都必须是计算机能够有效执行的, 并可得到确定的结果。

3) 有穷性

一个算法包含的步骤必须是有限的, 并在一个合理的时间限度内可以执行完毕。“有穷”是个相对的概念, 随着计算机性能的提高, 过去使用低速计算机需要执行若干年的算法 (相当于“无穷”), 现在使用新的高速计算机只需较短的时间就可执行完毕, 则又相当于“有穷”。

4) 输入性

执行算法时, 计算机可从外部取得数据。一个算法可以有多个输入, 但也可以没有输入 (0 个输入), 因为计算机也可自动产生一些必须的数据。

5) 输出性

一个算法必须有 1 个或多个输出。计算机是人们用于“解题”的工具, 因此算法必须具备向计算机外部输出结果的步骤, 否则, 该算法将毫无意义。

1.1.4 算法的表示和实现

算法可以多种方式来表示。如用自然语言即人们使用的英语、汉语等来表示；用图形（即后面将要述及的流程图）来表示；使用某种代码符号（如伪代码）来表示等等。由于图形的描述方法，具有形象、直观、清晰的特点，所以流程图得到广泛的应用。

算法仅仅提供了解决某类问题可采用的方法和步骤，还必须使用某一种计算机程序设计语言把算法描述出来，也就是说，使用某种程序设计语言所提供的语言成分，遵照规定的语法规则，去实现算法，这就是所谓的编程或编码（设计程序）。

因此，学习计算机程序设计语言，除了要学习该语言的各种语言成分的意义、功能、执行方式及各种语法规则之外，更重要的是还要了解该语言的特点，学会使用该语言实现各种算法的方法。

§ 1.2 结构化程序设计方法

随着计算机性能的提高，应用的普及，人们对程序的需求大增，程序也越来越复杂。由于程序设计本身的特点，使得程序设计的周期加长，程序调试（查错和排错）、维护（处理隐藏错误和完善）的工作量也大幅度增加。在此背景下，人们开始对程序设计的方法进行研究。

1965年，E. W. Dijkstra 最早提出了结构化程序设计的思想，其后又经其他多位学者的理论研究和努力，从而确立了结构化的程序设计方法。

前已述及，同一个问题可以设计出多种不同的算法，自然也可设计出多个不同的程序。结构化程序设计方法的核心是规定了算法的三种基本结构——顺序结构、分支结构和循环结构。并从理论上证明：无论多么复杂的问题，其算法都可表示为这三种基本结构的组合。依照结构化的算法编写的程序，结构清晰、易于理解、易于验证其正确性，也易于查错和排错。人们现已公认，结构化程序设计方法是一种“良好风格的”程序设计技术。

1.2.1 顺序、分支和循环结构

图 1.1 是三种基本结构的图形表示。

图 1.1(a)是顺序结构。其中的每个步骤(A 与 B)顺序执行。

图 1.1(b)是分支结构。其中 e 为判决条件，进入分支结构，首先判断 e 成立与否，再根据判决结果，选择执行 A 部分或 B 部分（只能选择执行其中一个）后退出。

循环结构又可分为两类，一类如图 1.1(c-1)所示，称为“当型循环”，另一类如图 1.1(c-2)所示，称为“直到型循环”。循环结构中的 A 是要重复执行的操作步骤，叫作“循环体”；e 为控制循环执行的条件。“当型循环”是“当”条件 e 成立（即为“真”），就继续执行 A，否则（即为“假”），就结束循环；而“直到型循环”是反复执行 A，“直到”条件 e 成立（即为“真”），就结束循环。

由图 1.1 可以看出，三种基本结构共同具有以下特点：

- (1) 只有单一的入口和单一的出口；
- (2) 结构中的每一个部分都有执行到的可能；
- (3) 结构内不存在永不终止的死循环。

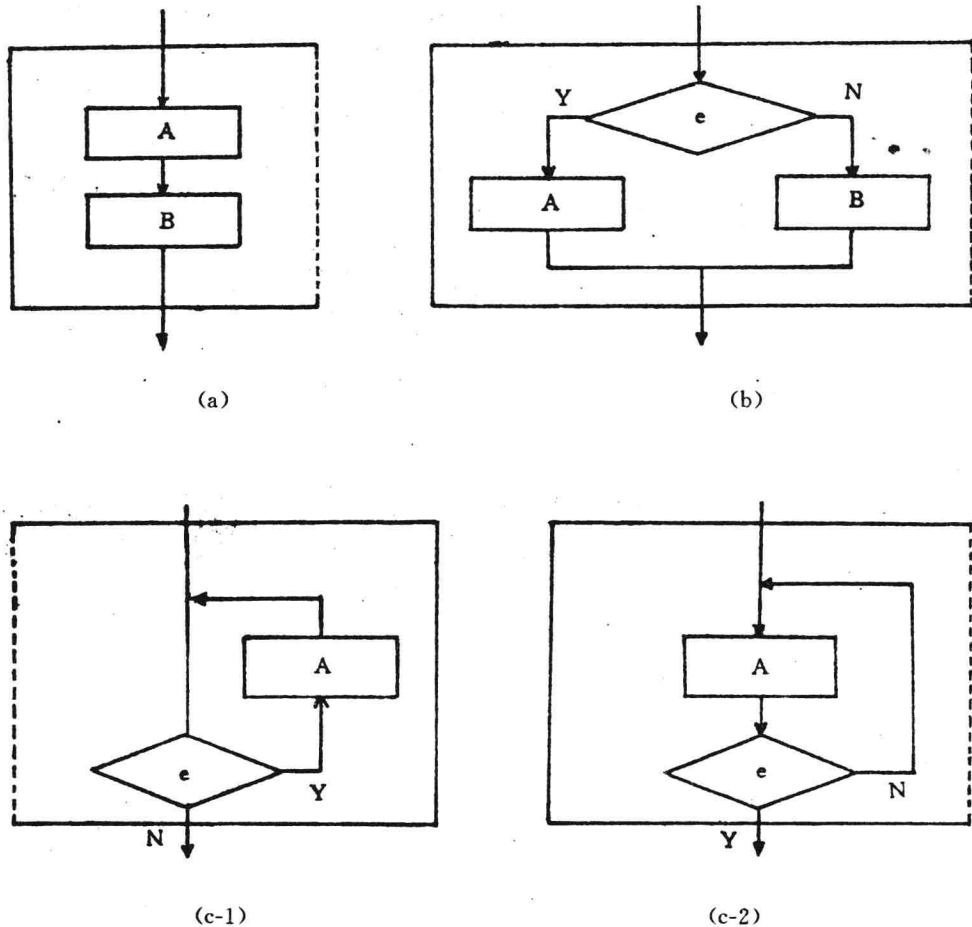


图 1.1

1.2.2 程序设计的一般过程

一般设计一个计算机程序的过程包括以下几个步骤：

(1) 问题分析。通过问题分析，要明确程序的任务和功能要求；需要处理的数据的数量、类型和来源，输入的方式；需要得到哪些处理结果，以何种格式输出等等。

(2) 建模及构造算法。在问题分析的基础上，进一步可确定问题的处理方法。首先是建立数学模型。对于数值计算方面的问题，通常只要选定计算公式即可；对于非数值计算方面的问题，则要根据具体问题经分析后，确定处理的数学模型。

有了数学模型，就可设计具体的算法，即处理的具体方法步骤。遵照结构化程序设计的原则，设计的算法应以“结构清晰”为第一。由于算法的好坏，往往会影响到处理的速度、精度以及程序的可维护性，所以在结构清晰的前提下，要仔细选择算法。通常对于简单的问题，如求若干个数之和或积，根据某种条件对数据进行分类处理或分类计数，从若干个数中选大、选小，对一批数进行排序或从一组有序排列的数据中查找某个数据等，都有公认的经典算法。在学习程序设

计语言时,应结合实例,熟练地掌握这些常用的算法。而对于复杂的问题,则采用“软件工程”的方法,通过把它分解为若干个相对独立的功能模块,使每个模块功能相对比较单一,算法比较简单,模块程序规模不大。这样复杂的问题也就转化为一个个简单问题的求解,算法设计的难度也就降低了。在算法确定后,为看起来清晰、直观,便于编程,一般还以图形方式(即后面将专门介绍的流程图)表示出来。

(3)程序编码。将算法用某种计算机语言描述出来,就称为“编程”或“程序编码”。只要算法正确,并较熟练地掌握了编程用的语言的各种语言成分及相关的语法规则,编程并不困难。

(4)测试与维护。刚编好的程序由于在问题分析、算法设计及编码过程中的考虑不周或失误,在将程序录入计算机时的差错等等原因,有可能存在这样或那样的缺陷和错误,必须通过测试检查并排除。测试又称为“调试”,是保证程序的正常工作必不可少的一步。使程序结构化、模块化,不仅可减少程序可能出现的错误,而且,即使有错,也便于检查和排除。

一些规模较大、比较复杂的程序,测试不可能一次性完成。在程序投入正常工作后,又可能发现一些隐藏很深的错误或程序存在的不尽人意的方面。维护就是在程序投入正常运行后,不断改正那些隐藏很深的错误或对程序的功能等进行改进。

§ 1.3 算法设计的常用策略

初学程序设计的人,最感困难的就是不知如何设计算法,换句话讲,就是不会根据问题确定解题的方法和步骤。下面介绍一些解题的基本思路,可能会对初学者有所帮助。

1. 迭代

迭代是科学计算领域常用的一种近似求解方法。所谓“迭代”,即是从一个已知的或事先估计的粗略的近似值 x_0 出发,通过一个迭代公式:

$$x_n = q(x_{n-1}) \quad n=1, 2, 3, \dots$$

可依次求得 $x_1, x_2, x_3, \dots$ 。只要迭代过程是收敛的,理论上讲,经无穷多次迭代,就可得到问题的精确解。实际运算时,则可给出一个允许误差 ϵ ,当前后两次近似值的绝对误差小于等于 ϵ 时,就结束迭代,而把最后一次迭代得到的近似值作为问题的解。

迭代算法可通过循环实现。输入迭代的初始值 x 和 ϵ ;在循环体内,将 x 赋给 x_0 ,再按迭代公式计算新的 x 值;然后就 $|x - x_0| \leq \epsilon$ 进行判决,若满足条件,则结束循环,输出 X ;否则,继续执行循环体。

迭代算法的关键是找出迭代公式。有了迭代公式,构造一个迭代算法并不困难。

2. 穷举

穷举又称为“枚举”。所谓“穷举”,即是通过将问题所有可能的解逐一测试的方法,找出问题的真解。

穷举算法一般通过多重嵌套的循环实现。首先确定由多个因子构成的解向量每个因子的可能取值范围;然后取其中 $n-1$ 个因子,构成 $n-1$ 重嵌套的循环,每重循环对应一个因子,循环的参数则由该因子的可能取值范围及变化规律决定;在最内层的循环体内,先确定第 n 个因子,再判别每个可能的因子组合是否满足给定的条件,并将满足条件的因子组合,即真解输出。

穷举算法的关键是确保遍历所有的可能解答,作到不重复也不遗漏。因此,仔细地设置各个循环的控制参数,至关重要。利用穷举法,有时也可用于证明给出的问题无解。

3. 递推

递推与“迭代”有些相似。所谓“递推”，也是已知一个初始值，同时已知未知值与已知值的相互关系(计算公式)，因而可以通过已知求出未知值，逐次前推，最终求出所要的结果。但与“迭代”不同的是，它不是一个收敛的过程。递推的终止以求出所要的结果或规定范围的结果为止。

递推算法也使用循环实现。

4. 递归

简单地讲，所谓“递归”就是一个过程直接或间接调用它自己。有许多函数或过程可递归定义，例如求阶乘的函数可定义为：

$$n! = \begin{cases} 1 & (n=0 \text{ 或 } n=1) \\ n \times (n-1)! & (n>1) \end{cases}$$

按阶乘的递归定义，要求 $n!$ 就需要先求出 $(n-1)!$ ，求 $(n-1)!$ 又要先求出 $(n-2)!$ ，逐次上溯，直到给出的初始边界条件(已知)，然后再逐次递推回来，最终求出所要的解。

采用标准的分支结构，就可轻而易举地实现递归算法。一般来说，一个递归算法也可通过递推实现，而且，采用递推算法运行的效率还要比采用递归算法高。但由于递归算法易于理解，算法描述更为简单，也更容易实现，所以在程序设计中，人们仍广泛地使用递归。

5. 分解

把一个复杂的问题，尽可能划分为若干较小的部分，分而治之，是解决规模较大且复杂问题的有效途径。

分解的方法有多种，最常见的是按功能分解。每一个小部分实现某种比较单纯的功能，解决某个方面的问题。由于每个小部分功能单一，规模较小，从而降低了算法设计的难度。各个小部分有机地联系在一起，就可实现复杂问题的求解。

习 题 一

1. 什么是算法？算法与程序设计有何关系？
2. 算法有哪些特性？
3. 结构化程序设计方法规定了哪几种基本算法结构？在执行方式上，各种结构有何特点？采用结构化程序设计方法有哪些优越性？
4. 简述用计算机解题的一般过程。

第二章 True BASIC 基础

§ 2.1 True BASIC 简介

True BASIC 是由 BASIC 语言的创始者 J. G. Kemeny 和 T. E. Kurtz 等于 1985 年 11 月根据美国国家标准 ANSI BASIC(1984)设计的 BASIC 语言新版本。它既保持了原 BASIC 语言易学、清晰、易懂的特色,又摒弃了原有 BASIC 版本中的缺陷,如版本繁杂、通用性和可移植性差、仅在解释方式下工作,运行速度较慢及非结构化等。按照程序设计方法学的原则,经过对原 BASIC 语言的扩充与改造,使其成为一种结构化、模块化、标准化的语言系统。

True BASIC 提供了功能强大的全屏幕编辑系统,更便于用户进行程序编辑、调试和运行。

True BASIC 同时提供了解释执行和编译执行两种工作方式。用户可在解释方式下对程序进行编辑和调试,而采用编译方式运行调试好的程序,提高程序运行的效率。

True BASIC 增加了许多功能极强的语句,是一种完全结构化的程序设计语言。

True BASIC 还具有较强的图形、动画和音乐功能,可用于开发绘图、游戏等程序。

更重要的是 True BASIC 是一种标准化的语言,具有很好的可移植性。它还兼容以前的各种 BASIC 版本,即用老版本开发的 BASIC 程序,一般都可直接在 True BASIC 环境中运行。True BASIC 还可给用户提供更比原 BASIC 更大的用户空间。

True BASIC 的特色及其强大的功能,不仅对于初学者还是有经验的程序设计人员,都有很大的吸引力,实践也表明, True BASIC 确实如它的创始者所宣称的“True BASIC 的出现开辟了 BASIC 语言的新纪元”。

§ 2.2 True BASIC 的安装、启动与基本操作

2.2.1 True BASIC 的运行环境与系统安装

1. True BASIC 的运行环境

(1) 硬件环境 IBM 系列微机及其兼容机,要求至少具有 192KB 内存容量,有一个双面软盘驱动器或硬盘。为了能充分发挥系统的功能,特别是彩色图形与动画功能,最好配置具有较高分辨率的彩色显示器。另外,最好配备一台打印机,以打印输出运行结果。

(2) 软件环境 PC-DOS(或 MS-DOS)2.0 以上版本。

2. 系统安装

True BASIC 系统由一个主文件 HELLO. EXE 及若干个辅助文件组成,存放在一张软盘上。在配置有硬盘的机器上使用时,可在硬盘上先建立一个名为 TB 的子目录,再使用 DOS 的

COPY 命令把软盘上的 True BASIC 系统复制到硬盘上的 TB 子目录中。具体操作过程如下:

加电开机,从硬盘上启动操作系统;

C>MD TB<回车> (在 C 盘根目录下建立 TB 子目录);

C>CD TB<回车> (改变当前目录为 TB 子目录);

将 True BASIC 系统软盘插入 A 软驱,关上软驱门;

C>copy A:*. * <回车> (拷贝系统文件)。

2.2.2 True BASIC 的启动与退出

1. 系统启动

以配有硬盘的机器系统为例,改变当前目录为 TB 子目录,在系统提示符下键入 HELLO 并回车,即:

C>HELLO<回车>

系统进入 True BASIC,屏幕将显示:

(编辑窗口)	
Untitled	
True BASIC here. Version 2.03	
Copyright (c) 1988. by True BASIC, Inc. All rights reserved.	
Type SPLIT to see a menu of the function keys.	
Ok. —	
(操作窗口)	

这表明 True BASIC 系统已启动成功,处于正常工作状态。

在 True BASIC 下,屏幕被分为两个部分。上半部分称为编辑窗口(editing window),用来输入程序及进行程序编辑;下半部分称为操作窗口或历程窗口(history window),用于输入控制程序工作的各种操作命令及显示程序的执行结果。操作窗口保留有最近执行的命令序列及相应的信息,用户可以回溯查询。

编辑窗口通常占 17 行,操作窗口占 8 行。必要时可以使用操作命令改变两个窗口的大小及屏幕的色彩。

启动 True BASIC 后,光标最初位于操作窗口的命令提示符“Ok.”后,此时按 F1 键可将光标移至编辑窗口;在编辑窗口下,按 F2 键,则又可将光标移到操作窗口。

光标位于那个窗口,就表明那个窗口为当前工作窗口,可接收用户的输入。

2. 系统退出

在操作窗口为当前工作窗口,且光标位于最后一个“OK.”后时,键入系统退出命令 BYE 并回车,即

Ok. BYE<回车>

就可退出 True BASIC 系统,返回 DOS。

注意:在退出系统前,必须先使用存盘命令(后述)将当前编辑窗口的文件内容存盘。如果忘记存盘,系统会自动给出下面的提示信息:

Do you want to save this file? (你要保存这个文件吗?)

操作者可键入 Y 或 N,表示要保存或不要保存该文件。若键入 N,就直接退出;键入 Y,系统又给出提示:

Save your program as:

要求操作者输入程序名。当操作者键入程序名并回车,系统就按给定的程序名将编辑窗口的内容存盘而后退出。

2.2.3 True BASIC 的基本操作

启动 True BASIC,在系统处于正常工作状态之后,用户就可以从键盘输入程序或者使用操作命令将保存在磁盘上的程序读入内存,并显示在屏幕的编辑窗口内。利用 True BASIC 提供的编辑功能,可非常方便地对程序进行增、删、改;使用各种操作命令,可显示磁盘文件目录、读磁盘程序文件、将程序文件存入磁盘、控制程序运行、对程序进行调试等等。了解 True BASIC 的各种基本操作命令并熟练地掌握其基本操作,是学习和使用 True BASIC 的第一步。

1. 程序输入及简单编辑

【例 2.1】 这是一个求面积为 10 平方厘米的圆的直径的简单 True BASIC 程序,下面以此为例来介绍程序输入及简单的编辑方法。

1) 输入程序

```
■ ! The example of True BASIC program.
■ let s=10
■ let d=sqr(s*4/3.14159)
■ print "the area of circle s=";s
■ print "the diameter of circle d=";d
■ end
```

Untitled

True BASIC here. Version 2.03

Copyright (c) 1988. by True BASIC, Inc. All rights reserved.

Type SPLIT to see a menu of the function keys.

Ok. —

输入在编辑窗口进行,按 F1 键将光标移至编辑窗口,就可逐行输入程序,输完一行,按回车键表示行结束,光标自动移到下一行。屏幕上每行最前面的■是系统自动添加的,称为行标志。

2) 简单程序编辑

True BASIC 支持全屏幕编辑,输入并显示在编辑窗口的程序若有错漏之处,可随意移动