

True BASIC 程序设计

(第二版)

谭浩强 张基温 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

True BASIC 是由 BASIC 语言的两位创始人推出的新的 BASIC 版本,对 BASIC 语言作了重大的改进和发展。本书系统地介绍了 True BASIC 的结构化与模块化程序设计方法,同时通过大量例题介绍算法及如何编程,每章后均附有习题。本书采用了新的体系编写,从算法入手,使读者在学习之后能具有较强的算法设计和程序设计的能力,养成良好的程序设计风格。

在这次修订中,改正了原书中个别错处,并补充了些内容,尤其是增加了第九章文件,以适应读者需要。

本书可作为高等院校、中专和各类计算机培训班的教材,也可供自学参考。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标志,无标志者不得销售。

图书在版编目(CIP)数据

True BASIC 程序设计/谭浩强,张基温编著. - 北京:清华大学出版社,1994
(计算机基础教育丛书/谭浩强等主编)

ISBN 7-302-01484-1

.T... . 谭... 张... .BASIC 语言-程序设计 .TP312 BA

中国版本图书馆 CIP 数据核字(94)第 01910 号

出版者:清华大学出版社(北京清华大学校内,邮编 100084)

印刷者:北京振华印刷厂

发行者:新华书店总店北京科技发行所

开 本: 787× 1092 1/16 印张: 21.25 字数: 529 千字

版 次: 1994 年 8 月第 2 版 1995 年 12 月第 4 次印刷

书 号: ISBN 7-302-01484-1/TP·590

印 数: 62001—77000

定 价: 17.80 元

这是薄一波同志为本书作者潭浩强教授的题词,祝贺他在我国计算机教育事业中取得的成绩,勉励他作出更大的贡献。

这是张劲夫同志赠本书作者谭浩强教授的题词,祝贺他在我国计算机教育事业中取得的成绩,勉励他作出更大的贡献。

《计算机基础教育丛书》出版说明

近年来,我国的计算机应用事业迅速发展,大批科技人员、大中学生、管理人员、以及各行各业的在职人员都迫切要求学习计算机知识,他们已经认识到,计算机知识是当代知识分子的知识结构中不可缺少的重要部分。

计算机应用人才队伍由两部分人组成:一部分是从计算机专业毕业的计算机专门人才,他们是计算机应用人才队伍中的骨干力量;另一部分是各行各业中从事计算机应用的人才。他们既熟悉本专业的业务,又掌握计算机应用的技术,人数众多,是计算机应用人才队伍的基本力量。他们掌握计算机知识的情况和应用计算机的能力在相当大程度上决定了我国计算机应用的水平。因此,在搞好计算机专业教育的同时,在广大非计算机专业中开展计算机基础教育是十分必要的。

非计算机专业中的计算机教学,无论就目的、内容、教学体系、教材、教学方法等各方面都与计算机专业有很大的不同,它以应用为目的,以应用为出发点。如果不注意这个特点,将会事倍功半。广大非计算机专业的师生、在职干部迫切希望有一套适合他们的教材,以便循序渐进地迈入计算机应用领域,并且不断地提高自己的水平。我们在前几年陆续编写了一些适合初学者使用的教材,受到广大群众的欢迎。许多读者勉励我们在此基础上进一步摸索和总结规律,为我国的广大非计算机专业人员编写一整套合适的教材。

近年来,全国许多专家、学者在这个领域作了有益的探索,写出了一批受到群众欢迎的计算机基础教育的教材。特别是全国高等学校计算机基础教育研究会作了大量的工作,在集思广益的基础上,提出了在高等学校的非计算机专业中进行计算机教育的四个层次的设想,受到广泛的注意和支持。我们认为:计算机的应用是分层次的,同样,计算机人才的培养也是分层次的;非计算机专业中各个领域的情况不同,也不能一律要求,在进行计算机教育时也应当有不同的层次。对于每一个学习计算机知识的人,还有一个由浅入深,逐步提高的过程。

我们认为,编辑出版一套全面而有层次的计算机基础教育的教材,目前不仅是十分必要的,而且是完全有条件的。在全国高等学校计算机基础教育研究会和许多同志的积极推动和清华大学出版社的大力支持下,我们决定编辑《计算机基础教育丛书》。它的对象是:高等学校非计算机专业的学生、计算机继续教育或培训班的学员、广大在职自学人员。

本丛书包括计算机科学技术的一些最基本的内容,例如计算机各种常用的

高级语言、计算机软件技术基础、计算机硬件技术基础、微型计算机的原理与应用、算法与数据结构、数据库基础、计算机辅助设计基础、微机网络与应用、系统分析与设计等,形成多层次的结构,读者可以根据需要与可能选学。

本丛书的宗旨是针对广大非计算机专业的需要和特点来组织教材,从实际出发,用读者容易理解的体系和叙述方法,深入浅出、循序渐进地帮助读者更好地掌握课程的基本内容。希望我们的丛书能在这方面具有自己的风格。在实践中接受检验。

本丛书的作者大多数是高等学校中有较丰富教学经验的教师。但是,由于计算机科学技术的飞速发展以及我们的水平有限,丛书肯定存在许多不足,丛书的书目和内容也应当不断发展和更新。我们热情地希望得到社会各界和广大读者的批评指正。

主编 谭浩强 林定基 刘瑞挺

1988.10

前 言

近年来,计算机的应用已在我国迅速推广,计算机的知识已成为当代科技人员和管理人员的知识结构中不可缺少的一部分。为了使用计算机,人们需要学习计算机语言。BASIC 语言是一种既容易学又有广泛应用价值的计算机高级语言。前几年在我国已有近千万人学习了 BASIC 语言。不少人从学习 BASIC 语言入门,迈进了计算机应用的领域。BASIC 的突出优点是充分考虑初学者的特点,它是为初学者设计的一种计算机语言,从 1964 年 BASIC 问世以来,它以极其迅速的速度在全世界推广开来。它受到千百万初学者的热烈欢迎,在我国同样也是如此。BASIC 语言在计算机的推广普及中已经并将继续发挥巨大的作用。

在计算机科学技术迅速发展的当今,所有的语言都在不断地发展和完善,特别是要适应结构化程序设计的要求。1984 年,BASIC 语言的创始人 John G. Kemeny 和 Thomas E. Kurtz 对 BASIC 语言作了重大的改进和发展,提出了名为 True BASIC(意为“真正的 BASIC”)的新的 BASIC 版本,它严格遵循美国国家标准 BASIC 的规定。True BASIC 是一种比较理想的计算机语言,它不仅完全适应结构化与模块化程序设计的要求,而且保留了 BASIC 语言的优点——易学易懂,程序易编写易调试。它不仅适用于数值计算、数据处理,还有丰富的作图功能。这两位 BASIC 的创始人宣称,True BASIC 的出现将开辟 BASIC 的新纪元。

True BASIC 的出现的确给 BASIC 带来了新的生命力。学习 True BASIC 将会使读者养成编制程序的良好风格。应当大力推广 True BASIC。

为了适应发展,并且满足广大计算机初学者和爱好者的要求,我们在 1988 年编写了本书的初版,系统地介绍 True BASIC 程序设计。我们不是简单地介绍 True BASIC 语法规则,而是着眼于算法,因为算法是程序设计的灵魂,语言只是实现算法的一种工具。学习语言的目的是利用语言工具进行程序设计。本书把重点放在提高编写程序的能力上。

在写法上,本书采取科学性与通俗性相结合的方式,通过大量例题深入浅出地介绍如何使用 True BASIC 语言编写各类程序。即使从未接触过计算机的读者,也能循序渐进地学懂本书的内容。

本书初版出版后,受到广大读者的广泛欢迎,已重印 7 次。我们在这次修订中,改正了原书中一些错处,并补充了一些内容,尤其是增加了第九章文件,以适应读者需要。

马素霞同志帮助编著者进行了修订工作,对原书进行整理、修改、补充,使本书的质量得以提高。

由于我们水平有限,书中难免存在错误之处,恳请专家和读者不吝指正。

谭浩强 张基温

1993.9

目 录

第一章 计算机算法	(1)
1.1 算法与计算机	(1)
1.1.1 算法的特征	(1)
1.1.2 计算机——实现算法的有力工具	(6)
1.1.3 计算机科学是研究算法的科学	(8)
1.2 算法的表示	(9)
1.2.1 概述	(9)
1.2.2 流程图	(10)
1.2.3 三种基本结构	(11)
1.2.4 N-S 结构流程图	(14)
1.3 用“逐步细化”方法进行算法设计	(18)
1.4 程序设计语言	(25)
1.4.1 概述	(25)
1.4.2 程序设计语言的发展	(25)
1.4.3 程序设计语言的功能	(29)
1.4.4 程序设计语言的使用	(30)
习题	(31)
第二章 True BASIC 程序设计初步	(34)
2.1 概述	(34)
2.1.1 True BASIC 简介	(34)
2.1.2 True BASIC 程序的组成	(36)
2.1.3 True BASIC 字符集	(40)
2.1.4 使用 True BASIC	(40)
2.2 数据描述	(43)
2.2.1 数据类型	(43)
2.2.2 常量与变量	(44)
2.3 True BASIC 表达式	(46)
2.3.1 算术操作符	(46)
2.3.2 标准函数	(46)
2.3.3 数学表达式	(46)
2.3.4 字符串表达式	(49)
2.4 数据传送	(50)
2.4.1 PRINT 语句	(50)

2.4.2	LET 语句	(53)
2.4.3	INPUT 语句	(54)
2.4.4	DATA/READ 语句与 RESTORE 语句	(57)
2.4.5	数据传送语句小结	(61)
2.5	选取型程序结构.....	(62)
2.5.1	逻辑表达式	(62)
2.5.2	IF 型结构控制	(64)
2.5.3	CASE 型结构控制	(71)
2.6	循环型程序结构.....	(75)
2.6.1	DO 循环	(75)
2.6.2	FOR 循环	(84)
2.7	程序设计举例.....	(92)
2.7.1	穷举	(92)
2.7.2	迭代	(97)
2.7.3	确定性模拟.....	(107)
2.7.4	概率性模拟.....	(109)
	习题.....	(119)
第三章	数组.....	(129)
3.1	用数组组织数据	(129)
3.1.1	几个基本概念.....	(129)
3.1.2	定义数组.....	(133)
3.1.3	数组测试函数.....	(135)
3.1.4	简单应用举例.....	(135)
3.2	数组的输入与输出	(147)
3.2.1	MAT PRINT 语句	(147)
3.2.2	MAT INPUT 语句	(149)
3.2.3	MAT READ 语句	(152)
3.3	数组赋值与运算	(153)
3.3.1	MAT 赋值语句	(153)
3.3.2	数组加.....	(155)
3.3.3	矩阵乘.....	(156)
3.3.4	数值量乘数组.....	(157)
3.3.5	内部数组常数.....	(157)
3.3.6	矩阵函数.....	(160)
3.4	排序	(160)
3.4.1	选择排序.....	(161)
3.4.2	插入排序.....	(163)
3.4.3	交换排序.....	(169)
	习题.....	(172)

第四章 函数与子程序.....	(178)
4.1 函数	(178)
4.1.1 自定义函数.....	(178)
4.1.2 外部函数——使用局部变量.....	(186)
4.1.3 递归函数.....	(190)
4.1.4 按功能定义函数.....	(193)
4.2 子程序	(193)
4.2.1 子程序的定义和调用.....	(194)
4.2.2 内部子程序与外部子程序.....	(194)
4.2.3 带参子程序——虚实结合.....	(196)
4.2.4 递归子程序与递归程序设计.....	(202)
4.3 库文件	(218)
4.3.1 库文件的概念.....	(218)
4.3.2 库文件的形成与使用.....	(219)
4.4 模块化程序设计	(219)
4.4.1 设计大程序的策略——模块化.....	(219)
4.4.2 模块间的层次结构.....	(222)
4.4.3 采用自顶向下、逐步细化的设计方法	(223)
4.5 程序的连接	(227)
4.5.1 连接语句(Chain 语句)	(227)
4.5.2 响应语句(Program 语句)	(228)
习题.....	(228)
第五章 字符串.....	(232)
5.1 基本概念	(232)
5.1.1 字符串常数与字符串变量.....	(232)
5.1.2 字符串的比较.....	(233)
5.1.3 字符串连接与子字符串.....	(233)
5.2 字符串传送	(234)
5.2.1 用 READ/DATA 或 INPUT 语句传送字符串常数	(234)
5.2.2 LINE INPUT 语句	(235)
5.2.3 用 LET 语句传送字符串数据	(236)
5.3 字符串函数	(237)
5.3.1 测字符串长度函数.....	(237)
5.3.2 字符串转换函数.....	(238)
5.3.3 字符串重复函数.....	(242)
5.3.4 子字符串查找函数.....	(242)
5.3.5 删除首尾空格函数.....	(243)
习题.....	(244)
第六章 程序设计方法和风格.....	(245)

6.1	程序质量标准	(245)
6.1.1	关于程序质量的几个概念.....	(245)
6.1.2	从效率第一到清晰第一.....	(247)
6.2	结构化程序设计	(248)
6.3	程序设计的风格	(249)
6.3.1	基本风格:简短朴实	(250)
6.3.2	程序文档化.....	(251)
6.3.3	使用过程的具体原则.....	(255)
6.3.4	使用控制结构的具体原则.....	(256)
6.3.5	提高表达式的可读性.....	(259)
	习题.....	(260)
第七章	数据的输入与输出.....	(262)
7.1	数据的输入	(262)
7.1.1	输入风格.....	(262)
7.1.2	单键输入(GET KEY) 语句	(264)
7.1.3	测试按任一键(KEY INPUT 函数)	(265)
7.2	数据输出格式的控制	(266)
7.2.1	显示的区宽和边界.....	(266)
7.2.2	TAB 定位	(268)
7.2.3	自定义输出数据项格式(PRINT USING 语句).....	(270)
7.2.4	MAT PRINT USING 语句	(274)
7.2.5	USING \$ 函数	(275)
	习题.....	(275)
第八章	图形.....	(278)
8.1	显示模式与图形窗口	(278)
8.1.1	显示器的工作模式.....	(278)
8.1.2	图形窗口坐标.....	(278)
8.2	画图	(280)
8.2.1	画点.....	(280)
8.2.2	画线.....	(281)
8.2.3	画矩形.....	(282)
8.2.4	画圆与椭圆.....	(282)
8.3	着色	(283)
8.3.1	前景颜色与背景颜色.....	(283)
8.3.2	用 BOX AREA 语句画实体矩形	(285)
8.3.3	用 PLOT AREA 语句画实体图形	(285)
8.3.4	用 FLOOD 语句着色.....	(285)
8.3.5	图形中的正文设置.....	(285)
8.4	动画	(286)

8.5	图画	(286)
8.5.1	图画的定义与调用	(287)
8.5.2	图画的变换	(287)
	习题	(288)
第九章	数据文件	(289)
9.1	有关概念	(289)
9.1.1	文件名	(289)
9.1.2	文件的存取方式与文件指针	(289)
9.1.3	文件组织	(290)
9.1.4	通道, 文件的打开与关闭	(291)
9.1.5	数据文件操作的一般过程	(293)
9.2	正文文件操作	(293)
9.2.1	写正文文件	(293)
9.2.2	写屏幕与打印机输出	(295)
9.2.3	读正文文件	(295)
9.2.4	按数据项处理正文文件	(296)
9.3	记录文件操作	(297)
9.3.1	True BASIC 记录文件的特点及其属性设置	(297)
9.3.2	记录文件的存取	(298)
9.3.3	多数据项记录的拼装与分解	(301)
9.4	字节文件操作	(302)
9.4.1	字节文件的存取特点	(302)
9.4.2	字节文件应用举例	(304)
9.5	文件管理	(305)
	习题	(306)
附录 A	编辑键一览表	(308)
附录 B	True BASIC 系统命令一览表	(309)
B.1	文件操作	(309)
B.2	编辑	(309)
B.3	查错	(309)
B.4	更改行号	(309)
B.5	其它	(310)
附录 C	True BASIC 语法一览表	(311)
C.1	简单语句	(312)
C.2	循环结构	(313)
C.3	选择结构	(314)
C.4	矩阵语句	(314)
C.5	程序单位	(315)
C.6	图形处理	(315)
C.7	文件处理	(317)
C.8	出错处理	(319)
附录 D	IBM-PC 字符与 ASCII 代码对照表	(321)

D.1	显示符号集	(321)
D.2	获得键的符号集	(324)
D.3	打印机符号集	(326)

第一章 计算机算法

电子计算机自 1946 年问世以来,以迅猛的速度发展,在短短的四十多年中,已经历了四个发展阶段,即所谓的“ 四代 ”——电子管计算机、晶体管计算机、集成电路计算机,大规模集成电路计算机,现在正在研制着第五代计算机。

计算机的出现是科学技术发展史上的一场伟大的革命,对人类社会产生了深远的影响。现在几乎每个领域都在大力开展计算机的应用。

计算机并不神秘,学会使用计算机也非难事。计算机是按照人们的意旨进行工作的。为了进行运算,人们必须事先准备好数据和程序。所谓程序,是指用计算机语言表示的操作步骤。例如,一个 BASIC 程序就是用 BASIC 语言写出的让计算机如何进行操作的一系列指令。

为了编好程序,就必须事先整理出解题的思路,正确地拟定出每一个操作步骤。“ 算法 ”就是研究这个问题的。本章将介绍算法、计算机和程序的有关知识。

1.1

1.1.1 算法的特征

初学者往往认为计算机解题是一个不可思议的过程。其实,计算机解题的过程与操作机器、执行任务、演奏音乐、打算盘、打太极拳、炒菜、做饭等极为类似。

广义地说,做任何事情都是一个过程,这个过程可以看作为在规定条件下能够进行的基本操作所组成的序列。例如,开汽车的过程可以看作是如下一个可进行的基本操作序列:

1. 掏出钥匙; 2. 打开车门; 3. 坐在座位上; 4. 打开电门; 5. 打火; 6. 加油; 7. 踩离合器;。或者说,我们在做任何事之前,都要在规定的解题环境之内,用所允许的基本操作去构造解题的步骤,只不过在做那些我们已经熟悉(习惯)的工作时,无须再有意识地考虑它罢了。例如,歌手唱歌、乐队演奏,都要为他们事先设计乐谱,乐谱就是由音符组成的序列,而音符就是音乐中所规定的基本操作的符号。同样,打太极拳的图谱、做菜的菜谱、珠算的口诀、工作计划、生产流程、治病处方、.....,都代表了在不同的解题环境中,为实现某一目的所应完成的基本操作序列。只不过在不同的解题环境下,对基本操作有不同的定义。这种解题操作序列称为“ 算法 ”。这里“ 解题 ”二字是泛指解决某一问题,而不只是指“ 计算 ”。计算机科学家 D. E. Knuth 说:“ 一个算法是一个有穷规则的集合,其中的规则规定了一个解决某一特定类型问题的解答。”算法的概念源于数学,当然包括了在数学中的应用。

例 1.1.1 有三个数 a, b, c , 找出其中最大的数。

我们先考虑处理这类问题的思路,先将 a 和 b 两数相比,将大者放在变量 \max 中,然后再让 c 与 \max 相比,如果 $c > \max$, 将 c 的值送到 \max 中,最后 \max 就是三个数中最大的数。将它写成以下形式:

S1: 将 a 与 b 比, 如是 $a > b$, 则使 a 的值作为 max 的值, 否则, 使 b 的值作为 max 的值。
可以写成: 如 $a > b$, 则 $a = \max$, 否则, $b = \max$ 。

S2: 将 c 与 max 比, 如 $c > \max$, 则 $c = \max$ 。

最后 max 的值就是三个数中最大的数。

上面用 S1, S2 表示步骤的次序, 在写算法时常用这种形式的标记。S 是 step(步)的首字母。S1 代表“第一步”, S2 代表“第二步”, ……。

例 1.1.2 有两个变量 a 和 b, 要求将它们的值互换。例如 $a = 3, b = 4$, 互换后, $a = 4, b = 3$ 。

为了进行两个变量的值互换, 须引入一个临时变量。正如两个瓶子内的液体互换需要用第三个瓶子作为过渡一样, 见图 1.1。其算法可表示如下:

S1: $a = c$ (将 a 的值送给变量 c);

S2: $b = a$ (将 b 的值送给变量 a);

S3: $c = b$ (将 c 的值送给变量 b)。

通过以上三个步骤实现了两个变量值的交换。这个方法在以后程序设计中常会用到。

例 1.1.3 求 $1 + 2 + 3 + 4 + 5$, 即求 $\sum_{n=1}^5 n$ 。

图 1.1

先用最原始的方法进行:

S1: 先进行 $1 + 2$ 的运算, 相加的和(结果)放在变量 N 中(N 的值为 3)。

S2: 将 N 再加 3, 和仍放在 N 中(此时 N 的值为 6)。

S3: 将 N 的值再加 4, 和仍放在 N 中(此时 N 的值为 10)。

S4: 将 N 的值再加 5, 和仍放在 N 中(此时 N 的值为 15)。

可将算法写成如下形式:

S1: 令 $N = 1$

S2: 令 $I = 2$

S3: N 与 I 相加, 结果放在 N 中

S4: 使 I 的值加 1, 即 $I = I + 1$

S5: 如果 $I > 5$, 返回 S3, 否则算法终止

最后 N 的值就是 $\sum_{n=1}^5 n$ 的结果。

在执行这个算法时, S3 和 S4 两个步骤要重复执行多次。直到 $I > 5$ 为止(请读者考虑 S3 和 S4 重复执行多少次?)。可以看到, 这个算法是采用循环的方法来处理。请考虑: 在 S5 步骤中的循环继续的条件为什么是“ $I > 5$ ”, 如果改写为“ $I < 5$ ”会出现什么情况?

显然, 这个算法是比较好的。如果要求的是 $\sum_{n=1}^{100} n$, 即 $1 + 2 + \dots + 100$, 上述算法基本上可不必改, 只需将 S5 步骤中的“若 $I > 5$ ”改为“若 $I > 100$ ”即可。读者可自己写出此完整的算法, 并分析它的执行过程。

例 1.1.4 有 10 个数, 找出其中最大的数。

本题的思路: 如同打擂台一样, 台上先站着一个人, 第二个人上台与之比较, 胜者留在台

上,然后第三个人上台再与台上的人(胜者)比较,胜者留在台上,比完九轮之后,在台上的人便是最后胜者。

今先将输入的第一个数放在 MAX 中,将输入的第二个数与之相比,如第二个数大于 MAX,则它取代 MAX 的原值。然后第三个数再与 MAX 比,大者放在 MAX 中……,一直到比完 9 次为止。可以写成以下的算法:

S1: 输入一个数,放在 MAX 中。

S2: $I \leftarrow I + 1$ (I 用来累计比较次数)。

S3: 输入一个数,放在 X 中。

S4: 若 $X > MAX$, 则 $MAX \leftarrow X$ 。

S5: $I \leftarrow I + 1$ (I 的值加 1)。

S6: 若 $I \leq 9$, 返回 S3 继续执行。否则, 停止。此时 MAX 的值就是 10 个数中最大的数。

例 1.1.5 求两个正整数 m 和 n ($m > 0, n > 0, m > n$) 的最大公约数。

求两个正整数的最大公约数的算法是介绍算法时常常引用的一个典型例子。为了使读者容易理解,我们首先看看手工求 $M = 60$ 和 $n = 33$ 的最大公约数的过程:

S1: 以 n(33) 除 m(60), 得余数 r(27)。

S2: 判断 r 是否等于零: 若 $r = 0$, 则 n 为解, 若 $r \neq 0$, 则进行 S3 的操作(现 $r = 27$, 故进入操作 S3)。

S3: 以 n 作为新的 m(33), 以 r 作为新的 n(27) 求新的 m/n 的余数 r(6)。

S4: 判断 r 是否为零: 若 $r = 0$, 则前一个 n 即为解; 否则要继续 S5 的操作。

S5: 以 n 作为新的 m(即使 $27 = m$), 以 r 作为新的 n(即使 $6 = n$), 求新的余数 r(3)。

S6: 判断上一个 r 是否等于零: 若 $r = 0$, 则前一个 n 即为解; 否则执行下面的操作。

S7: 再以 n 作为新的 m($6 = m$), r 作为新的 n($3 = n$), 求新的 r($r = 0$)。

S8: 判断上一个 r 是否等于零。这里已有 $r = 0$, 所以算法结束, $n = 3$ 即为 60 与 33 的最大公约数。

这种算法称为辗转相除。算法中使用的基本操作: 整除求余和判断。它们都是进行算术运算时所允许的操作, 只不过判断这一操作是在人脑中完成的, 并不需要显式地写出来罢了。

从上面的算法中可以看出, S8, S6, S4 与 S2 所完成的操作是相同的, S7, S5 与 S3 所完成的操作也是相同的。或者说, S8, S6, S4 是 S2 的重复, S7, S5 是 S3 的重复。经过加工整理, 可以得到下面的一个简单明了的算法 2:

S1: 置数。将两个数中的大数放到 m 中, 小数放到 n 中。

S2: 求余。求 m/n 的余数 r。

S3: 判断。若 $r = 0$, 则 n 就是所求最大公约数; 若 $r \neq 0$, 执行下一步。

S4: 置换。使 n 值作为新的 m 值($n \leftarrow m$), 使 r 值作为新的 n 值($r \leftarrow n$)。

这个算法的 S4, 使用了循环, 控制 S2, S3, S4 重复执行几次, 使算法变得简单明了。稍作变换, 我们还可以将它变换为下面的算法 3:

S1: 置数。将两个数中大数放到 m 中, 小数放到 n 中。

S2: 重复执行下面的序列, 直到求得 $r = 0$ 为止。

S2.1: 求余。求 m/n 的余数 r。

S2.2: 置换。 $n = m, r = n$ 。

S3: 输出 m 。

上面的算法对任何 m 与 n 都是适用的。要理解这个算法或要写出这个算法需要经过一定的抽象, 即从具体问题出发, 经过提炼、归纳, 找出解决本类问题的普遍规律。譬如在智力竞赛时, 对于“ $1+2+\dots+100$ ”这样一个题目, 有人按 作 $1+2=3$, 作 $3+3=6$, 作 $6+4=10$,的方法去作; 而有人找到了另一种高效率的算法: $100+(1+99)+(2+98)+\dots+(49+51)+50$, 共得 50 个 100, 1 个 50, 即和为 5050。一旦得到这样一种方法, 当遇到题目为 $1+2+\dots+1000$ 时, 他决不会重新冥思苦想地另寻算法, 而是运用前面的方法, 很快算出 $1000+(1+999)+(2+998)+\dots+(499+501)+500=500500$ 。

例 1.1.6 求正整数 A 和 B 的积, A 和 B 为任意位数。

先看一下人工计算是怎样进行的。假如 $A=3412, B=513$, 乘法竖式为

$$\begin{array}{r} 3412 \text{---} A \\ \times 513 \text{---} B \\ \hline 10236 \\ 3412 \quad \text{部分积} \\ 17060 \\ \hline 1750356 \text{---} \text{积} \end{array}$$

它们是这样进行的: 以 A 为被乘数, 分别以 B 的个位数、十位数、百位数作为乘数, 求出它们的部分积。个位数与 A 相乘, 得到第一行部分积, 十位数与 A 相乘得到部分积(第二行)应左移一位, 或者认为将该位数字乘以 10 再和 A 相乘; 同样由乘百位数得到的第三行部分积应左移二位, 或者认为将该数乘以 100 再和 A 相乘。把各部分积相加即得最后的积。

现在 A 和 B 的位数都是任意的, 那么怎样用一般抽象的方法表示此算法呢?

将乘数 B 表示为 $b_n b_{n-1} \dots b_2 b_1 b_0$, b_0 表示个位数字, b_1 表示十位的数字, b_2 表示百位的数字, 其余类推。用 C 表示某一时刻的部分之和。用 i 表示乘数的位数, $i=0$ 表示个位(因为 $10^0=1$), $i=1$ 表示十位(因为 $10^1=10$), $i=2$ 表示百位(因为 $10^2=100$),。

算法可以表示如下:

S1: 使 $i=0, C=0$ (未进行乘数运算前, 部分积之和为零), 并给 n 赋初值。

S2: 当 $i \leq n$ 时, 重复执行下面的操作:

S2.1: $C + A \times b \times 10^i \leftarrow C$ (乘 10^i 即左移 i 位, 第一个部分积为 $0 + A \times b \times 10^0 = A \times b_0$);

S2.2: $i + 1 \leftarrow i$ (i 增 1)。

S3: 输出 C 。

请读者按上述算法进行 853×28654 的运算。

例 1.1.7 求两个整数 $X(X \geq 0)$ 和 $Y(Y > 0)$ 的整数商和余数(规定只能用加法和减法运算)。

除法运算实际上是用加法和减法来实现的。其办法是, 从 X 中一次又一次地减去 Y , 直到 $X < Y$ 为止, 所减的次数即为商。例如 $5 \div 2$ 的商为 2, 余数为 1。可以将 5 先减 2, 余 3, 再减一次 2, 余 1。每减一次, 商加 1。5 能减两次 2, 所以商就得 2。此时余数为 1, 不能再减 2, 结束。用 Q 代表商, R 代表余数, 算法可以表示为