

新世纪

计算机基础教育丛书

丛书主编

谭浩强

QBASIC程序设计

题解与上机指导

谭浩强 张基温 编著
马素霞 薛淑斌



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



2-44

8/7

TJ312.4
THG/7

新世纪
计算机基础教育丛书

丛书主编

谭浩强

QBASIC程序设计 题解与上机指导

谭浩强

张基温

马素霞

薛淑斌

编 著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>

(京)新登字 158 号

JS362/24
内 容 简 介

本书是和《QBASIC 程序设计》(谭浩强等编著,清华大学出版社出版)一书配套使用的参考书。书中包括《QBASIC 程序设计》各章的全部习题和参考解答,按照结构化程序设计的方法编写了可供运行的 QBASIC 程序,并提供了学习本课程的实验指导和实验安排。读者通过阅读本书提供的例题和程序,可以进一步熟悉结构化的程序设计方法,更深入地掌握 QBASIC 语言。

本书除作为《QBASIC 程序设计》的配套用书外,还是所有 QBASIC 程序设计学习者的一本很好的参考书,可供大学、高职、高专、中专以及计算机培训班学生使用,也可供读者自学参考。

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签,无标签者不得销售。

图书在版编目(CIP)数据

QBASIC 程序设计题解与上机指导/谭浩强等编著. —北京:清华大学出版社,2000

(新世纪计算机基础教育丛书/谭浩强主编)

ISBN 7-302-03910-0

I. Q… I. 谭… II. BASIC 语言-程序设计-自学参考资料 N. TP312

中国版本图书馆 CIP 数据核字(2000)第 30041 号

出版者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 北京市人民文学印刷厂

发行者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 11.25 **字数:** 267 千字

版 次: 2000 年 8 月第 1 版 2000 年 8 月第 1 次印刷

书 号: ISBN 7-302-03910-0/TP·2285

印 数: 0001~8000

定 价: 14.00 元

21世纪终于来临了,在新的世纪,人们自然对未来有许多美好的愿望和设想。现代科学技术的飞速发展,改变了世界,也改变了人类的生活。作为新世纪的大学生,应当站在时代发展的前列,掌握现代科学技术知识,调整自己的知识结构和能力结构,以适应社会发展的要求。新世纪需要具有丰富现代科学知识、能够独立解决面临任务、充满活力、有创新意识的新型人才。

掌握计算机知识和应用无疑是培养新型人才的一个重要环节。计算机既是现代科学技术的结晶,又是大众化的工具。学习计算机知识不仅是为了掌握一种技能,更重要的是:它能启发人们对先进科技的向往,激发创新意识,推动对新知识的学习,培养自学能力,锻炼动手实践的本领。因而它是高等学校全面素质教育中极为重要的一部分。

自20世纪80年代初以来,高等学校中计算机教育(尤其是非计算机专业中的计算机教育)发展迅速,计算机教育的内容不断扩展,程度不断提高,它所起的作用也愈来愈显著。

在实践中,大家已认识到,计算机应用人才队伍是由两部分人组成的:一部分是计算机专业出身的计算机专业人才,这部分人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应用计算机的人员。这后一部分人一般并非从计算机专业毕业,他们人数众多,既熟悉自己所从事的专业,又掌握计算机的应用知识,善于用计算机作为工具去解决本领域中的任务。他们是计算机应用人才队伍中的基本力量。事实上,大部分应用软件都是由非计算机专业出身的计算机应用人员研制的。他们具有的这个优势是其他人难以代替的。从这个事实可以看到在非计算机专业中深入进行计算机教育的必要性。

非计算机专业中的计算机教育,无论目的、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,决不应该照搬计算机专业的模式和做法。全国高等院校计算机基础教育研究会自1984年成立以来,始终不渝地探索高校计算机基础教育的特点和规律,在80年代中期,最早提出了按层次进行教育的方案。计算机应用是分层次的,不同的人在不同的层次上使用着计算机,同样,计算机教育也是分层次的,以适应不同应用层次的要求。全国有一千多所高等学校,好几百个专业,学校的类

型、条件和基础差别很大,不可能按同一模式、同一要求、同一内容进行教学。按层次组织教学,可以使不同专业、不同学校能够根据自己的情况选择教学内容,做到“各取所需”。

经过十多年的实践,几经调整,许多高校形成了按以下三个层次组织教学的方案:第一层次为计算机公共基础,学习计算机基本知识和基本操作;第二层次为计算机技术基础,内容包括程序设计、数据库、网络和多媒体等;第三层次为计算机应用课程,结合专业应用的需要学习有关计算机应用课程。每一层次中设立若干门课程,包括必修课和选修课。

1988年起,我们根据层次教学方案,组织编写了“计算机基础教育丛书”,邀请有丰富教学经验的专家学者先后编写了20多种教材,由清华大学出版社出版。丛书出版后,迅速受到广大高校师生的欢迎,对高等学校的计算机基础教育起了积极的推动作用。广大读者反映这套教材定位准确、内容丰富、通俗易懂,符合广大非计算机专业学生的特点。许多高校都采用了我们编写的教材。丛书总发行量达到700多万册,这在全国是罕见的。

在新世纪来临之际,我们在该丛书成功的基础上组织了这套“新世纪计算机基础教育丛书”,以适应新形势的要求。本丛书有以下特点:

(1) 内容新颖。根据新世纪的需要,重新确定丛书的内容,以符合计算机科学技术的发展和教学改革的要求。本丛书除保留了原丛书中经过实践考验、且深受群众欢迎的优秀教材外,还新编写了许多新的教材,在这些教材中反映了近年来迅速得到推广应用的一些计算机新技术,以后还将根据发展不断补充新的内容。

(2) 适合按层次组织教学的需要。在新世纪大多数学校是采用层次教学模式的,但不同的学校和专业所达到的层次不同,本丛书采用模块形式,提供了各种课程的教材,内容覆盖高校计算机基础教育的三个层次。既有供理工类专业用的,也有供文科和经济类专业用的;既有必修课的教材,也包括一些选修课的教材供选用。各类学校都可以从中选择到合适的教材。

(3) 符合大学非计算机专业学生的特点。本丛书针对非计算机专业学生的特点,以应用为目的,以应用为出发点,强调实用性。本丛书的作者都是长期在第一线从事高校计算机基础教育的教授和副教授,对学生的基础、特点和认识规律有深入的研究,在教学实践中积累了丰富的经验,可以说,每一本教材都是他们长期教学经验的总结。在教材的写法上,既注意概念的严谨和清晰,又特别注意采用读者容易理解的方法阐明看来深奥难懂的问题,做到例题丰富,通俗易懂,便于自学。这一点是本丛书一个十分重要的特点。书是写给读者看的,读者如果看不懂,只能算失败。

(4) 采用多样化的形式。除了文字教材这一基本形式外,有些教材还配有习题解答和上机指导,我们还准备采用现代教学方式,陆续制作电子出版物,以利于学生自学。

总之,本丛书的指导思想是:内容新颖、概念清晰、实用性强、通俗易懂、层次配套。简单概括为:“新颖、清晰、实用、通俗、配套”。我们经过多年实践形成的这一套行之有效的创作风格相信会受到广大读者欢迎。判别一本书的优劣,读者最有发言权。

本丛书多年来得到各方面人士的指导、支持和帮助,尤其是得到全国高等院校计算机基础教育研究会的各位专家和各高校的老师们的支持和帮助,我们在此表示由衷的感谢。

本丛书肯定有不足之处,竭诚希望得到广大读者的批评指正。

丛 书 主 编
全国高等院校计算机基础教育研究会理事长
谭 浩 强
2000 年 1 月 1 日



现在,我国已经欣起了第三次计算机普及的高潮,愈来愈多的人参加到学习计算机的行列中来。计算机的学习和应用是分层次的。在学习了计算机的初步知识和基本操作后,不少人希望学习程序设计的知识。应该说,程序设计是计算机应用人员必备的知识 and 能力。由于 QBASIC 语言易学易用,因此它成为许多人学习程序设计时的首选语言。

为了帮助广大读者学习程序设计,不久前我们编写了《QBASIC 程序设计》一书,由清华大学出版社出版。该书针对初学者的特点,进行了深入浅出的介绍,同时注重算法的概念,突出结构化程序设计思想。该书出版后,很受欢迎,被许多大学、高职、高专和计算机培训班选为教材。为了帮助大家更好地学习 QBASIC 程序设计课程,根据广大读者的要求,我们编写了这本书。

本书包括了《QBASIC 程序设计》一书的全部习题和完整的参考解答。对于编程题,都按结构化编程方法编写出 QBASIC 源程序,用 Turbo C 2.0 调试通过,而且附有运行结果,供读者对照程序进行分析。对部分较复杂的问题,还给出了流程图,并作了算法分析,以帮助读者理解程序的思路。本书的 1~9 章与教材的各章相对应,题号也与教材的题号一致,以方便读者查阅。

应当说明,本书介绍的算法和程序不是惟一的,甚至不一定是最好的,只是一种参考解答。对同一个问题,可以有不同的解决方法。读者最好能写出不同于本书的程序,并与本书的程序作比较,这样收获会更大。读者完全有可能写出比本书更好的程序。

由于讲课的时间有限,教师不可能在课堂上介绍太多的程序,应当要求学生课外阅读更多的程序,并且自己编写程序和上机调试。在本书中提供了各种不同类型的程序,学生通过阅读这些程序,可以进一步丰富自己的知识,熟悉结构化的编程方法,更深入地掌握 QBASIC 语言。教师也可从中选择部分程序在课堂上讲授,或者指定学生自学。

除了提供教材各章参考答案外,在本书第 10 章中提供了学习本课程的上机实验的要求。上机实验的目的不仅是为了验证程序正确,还应当通

过上机培养调试程序的能力。在实验过程中要善于发现问题、思考问题、解决问题。我们在第 11 章中安排了 10 个实验,实验的顺序与教材的顺序一致,以配合教学的进行。在每个实验中,我们都提供了一些启发性的问题供大家在实验过程中思考和实践,希望读者重视实验,在实验中开动脑筋,使实验取得更好的效果。

本书是由谭浩强教授、张基温教授、马素霞副教授、薛淑斌高级工程师在《Ture BASIC 程序设计题解》(谭浩强、张基温、马素霞编著)的基础上重新编写而成的。马素霞副教授在改编过程中承担了主要部分的工作。薛淑斌高工编写了第 8 章、第 9 章。祖向荣副教授编写了实验 2~5 的初稿。最后由谭浩强教授修改定稿。

本书不但可以作为《QBASIC 程序设计》的参考书,而且所有 QBASIC 学习者都可以从本书中得到裨益。它是学习 QBASIC 的一本很好的参考书。

本书有不妥之处,请专家和读者不吝指正。

作者

2000 年 3 月

目 录

Catalog Catalog Catalog Catalog

1	程序设计的初步知识	1
2	QBASIC 程序设计初步	10
3	选择结构程序设计	18
4	循环结构程序设计	29
5	数组与记录	51
6	函数与子程序	88
7	字符串	113
8	屏幕控制与图形	122
9	数据文件	135
10	上机实验指导	145
10.1	上机实验的指导思想和要求	145
10.2	程序的调试和测试	146



11.1	实验 1	QBASIC 的基本操作	149
11.2	实验 2	顺序结构程序设计	152
11.3	实验 3	选择结构程序设计	154
11.4	实验 4	循环结构程序设计	157
11.5	实验 5	一维数组	159
11.6	实验 6	二维数组和记录	160
11.7	实验 7	函数与子程序	163
11.8	实验 8	字符串	166
11.9	实验 9	图形	167
11.10	实验 10	文件	169

第1章 程序设计的初步知识

- 1.1 有人说：“人工解题的最好方法不一定是计算机解题的最好方法。”对吗？为什么？

解：人工解题的最好方法不一定就是计算机解题的最好方法。

首先，人脑与电脑的物理组成不同；其次，人脑与电脑的工作方式不同。因此，人工解题与计算机解题的机制是不同的。人工解题时不仅可以利用算术方法，还可以利用公理系统进行复杂的推理及演算。而计算机并不具有思维能力，不能像人一样用公理系统按演绎方式解题。计算机只能执行数据传送、逻辑运算、算术运算、比较、判断及转移等简单的基本操作。用计算机解题的方法必须能够分解成计算机能够执行的基本操作。由于计算机所能执行的操作与人所能执行的操作不同，因此，人能执行的某些操作计算机不能直接执行。而计算机运算速度快，适合于执行多次重复性的操作。当求解过程能被描述为多次重复的简单操作时，更能发挥计算机的解题优势。这一点是人所不能比拟的。

- 1.2 请叙述“设计算法”和“实现算法”这两个概念。怎样才算实现了一个算法？写出源程序算不算实现了一个算法？

解：“算法”就是解决问题的步骤。因此，“设计算法”就是指使用算法描述工具写出解题步骤的过程，这个过程常表现为从概要到详细、从抽象到具体。常用的描述工具有：流程图、N-S图、PAD图、IPO图、判定表、伪代码等。

“实现算法”是指用计算机来实现算法所描述的功能。按照算法编写程序，在计算机上调试、编译、运行，并验证结果的正确性，才算实现了一个算法。写出源程序并不算实现了一个算法。实际上，源程序只不过是算法的另一种描述形式，即是用程序设计语言描述的算法。

- 1.3 什么是结构化算法和结构化程序设计？它们有些什么特征？

解：由顺序结构、分支结构、循环结构三种基本结构顺序组成的算法属于结构化算法。

在解决问题时采用模块化的原则将大的、复杂的问题进行分解，得到若干个功能相对独立的子问题；还可以对子问题继续分解，直到每一个子问题都是人们的智力所能及的问题，再分别设计解决每一个子问题的算法。在进行算法设计时，采用自顶向下、逐步求精的方法设计出结构化算法，再用结构化的程序设计语言来描述。在所有的结构化程序设计语言中，都有直接实现三种基本结构的语句，所以很自然地就得到了结构化的程序。使用非结构化的程序设计语言也可以编写结构化的程序，这需要借助 GOTO 语句来实现三种基本结构。从对问题的分析到设计出结构化的算法，再到编写出结构化的程序，这样的过程就称为结构化程序设计。

结构化算法的特征是：算法必须由三种基本结构（以及由它们派生出来的其他基本结构）组成。三种基本结构具有以下共同的特点：

- (1) 只有一个入口。
- (2) 只有一个出口。
- (3) 每一个基本结构中的每一部分都有机会被执行到。
- (4) 结构内不存在死循环。

由这三种基本结构组成的算法在总体上呈现顺序结构,结构之间通过出口和入口相接,不存在无规律的转移,只在基本结构内才允许存在分支或转移。

结构化程序设计的特征是在程序设计过程中采用自顶向下、逐步细化以及模块化的原则,设计出的程序结构清晰,容易阅读和修改。

1.4 下面是打印出最初 n 个素数的算法,请用 N-S 图表示。

- (1) p 置初值 2, k 置初值 1, $s=0$, 输入 n 的值。
- (2) 打印 p 。
- (3) 使 $p+1 \Rightarrow p$ 。
- (4) 当 $k < n$ 时,重复做下面的工作,否则转(5)。
 - ① $2 \Rightarrow i$ 。
 - ② 当 $i \leq \sqrt{p}$ 且 $s=0$ 时,重复做以下工作:
 - ③ p/i 的余数 $\Rightarrow r$;
 - ④ 如 $r \neq 0$, 则 $i+1 \Rightarrow i$, 否则 $s=-1$ 。
 - ⑤ 如 $s=0$, 则打印 p , $k+1 \Rightarrow k$ 。
 - ⑥ $s=0$ 。
 - ⑦ $p+2 \Rightarrow p$ 。
- (5) 结束。

请弄清楚这个算法。 p 是待判断的数。 k 表示已找到的素数的个数。 s 作为“标志”,当 $s=-1$ 时表示 p 能被某一数整除, p 不是素数。

解: N-S 图见图 1.1。

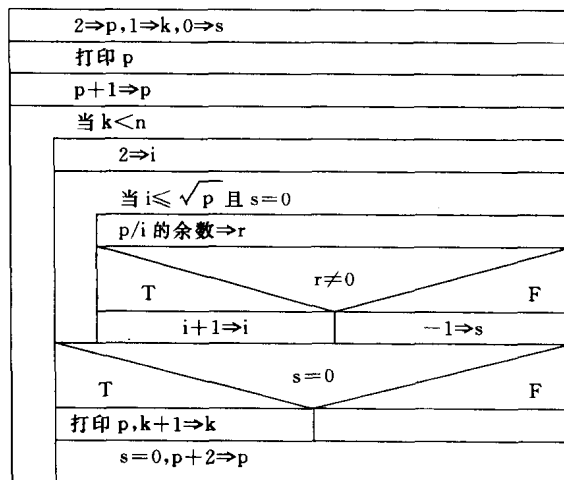


图 1.1

1.5 求 y 的值,请用 N-S 图表示算法。

$$y = \begin{cases} -1 & (x < 0) \\ 0 & (x = 0) \\ 1 & (x > 0) \end{cases}$$

解: N-S 图见图 1.2。

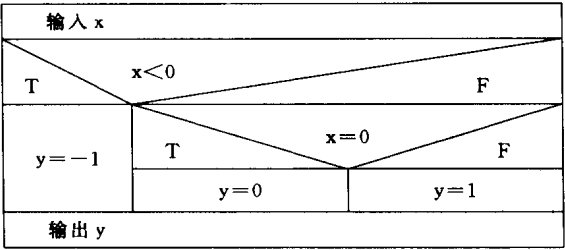


图 1.2

请用逐步细化的方法设计下列各题的算法(最后用 N-S 图描述)。

1.6 计算 $s = 1 + 2 + 2^2 + \cdots + 2^{15}$ 。

解: 初步算法:

S1: $0 \Rightarrow s$ 。

S2: 计算 s 的值。

S3: 输出 s 的值。

对 S2 细化: 设变量 p 存每项的值, 变量 i 控制循环次数。

S2.1: $1 \Rightarrow p, 0 \Rightarrow i$ 。

S2.2: 当 $i \leq 15$, 重复以下操作:

S2.2.1: $p + s \Rightarrow s$;

S2.2.2: $p * 2 \Rightarrow p$;

S2.2.3: $i + 1 \Rightarrow i$ 。

算法的 N-S 图如图 1.3 所示。

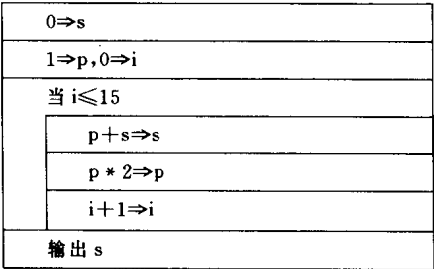


图 1.3

1.7 计算 $s = a_0 + a_1x + a_2x^2 + \cdots + a_{10}x^{10}$ 。

解: 初步算法:

S1: $0 \Rightarrow s$ 。

S2: 计算 s 的值。

S3: 输出 s 。

对 S2 细化: 设变量 p 存 x^i 的值, 变量 i 控制循环的次数。当 i 从 0 到 10 变化时, 将 $a_i x_i$ 累加到变量 s 上。今只设一个变量 a 存系数, 故 11 个系数不能同时输入, 需要边输入边计算。具体算法如下:

S2.1: 输入 x 的值。

S2.2: $0 \Rightarrow i, 1 \Rightarrow p$ 。

S2.3: 当 $i \leq 10$ 重复执行以下操作:

S2.3.1: 输入 x^i 的系数 $a_i \Rightarrow a$ 。

S2.3.2: $a * p + s \Rightarrow s$ 。

S2.3.3: $p * x \Rightarrow p$ 。

算法的 N-S 图如图 1.4 所示。

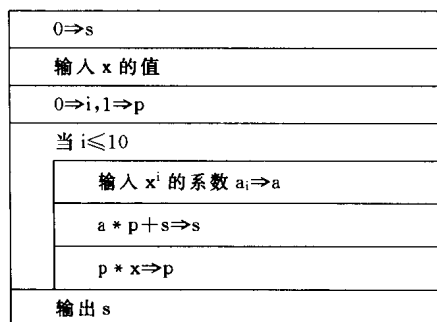


图 1.4

1.8 计算 $s = 1 + (1 + 2!) + \cdots + (1 + 2! + \cdots + 10!)$ 。

解: 初步算法:

S1: $0 \Rightarrow s$ 。

S2: 计算 s 的值。

S3: 输出 s 的值。

对 S2 细化: 设变量 t 存累加项的值, 变量 p 存 $i!$ 的值, 变量 i 控制循环的次数。当 i 从 1 到 10 变化时, 计算 p 的值; 将 p 累加到变量 t 上, 再将 t 累加到变量 s 上。具体算法如下:

S2.1: $0 \Rightarrow t, 1 \Rightarrow p$ 。

S2.2: $1 \Rightarrow i$ 。

S2.3: 当 $i \leq 10$ 重复执行以下操作:

S2.3.1: $p * i \Rightarrow p$ 。

S2.3.2: $t + p \Rightarrow t$ 。

S2.3.3: $s + t \Rightarrow s$ 。

S2.3.4: $i + 1 \Rightarrow i$ 。

算法的 N-S 图如图 1.5 所示。

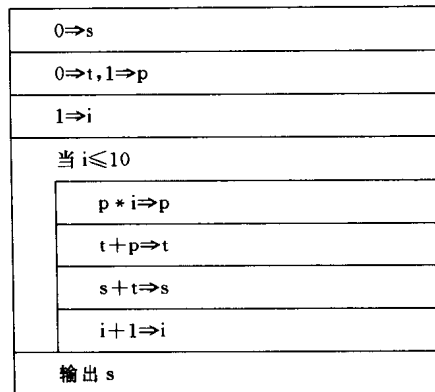


图 1.5

1.9 求方程 $ax^2+bx+c=0$ 的根。

解：算法：

S1：输入方程的系数 a、b、c，并限制 a、b 不同时为 0。

S2：如果 $a=0$ ，则计算一次方程 $bx+c=0$ 的根并输出。

否则，求二次方程 $ax^2+bx+c=0$ 的根并输出。

求二次方程 $ax^2+bx+c=0$ 的根的算法如下：

S2.1：求判别式 b^2-4ac 的值 disc。

S2.2：如果 $\text{disc} > 0$ ，则计算两个不相等的实根： $x_{1,2} = (-b \pm \sqrt{\text{disc}})/(2a)$ ；

如果 $\text{disc} = 0$ ，则计算两个相等的实根： $x_1 = x_2 = -b/(2a)$ ；

如果 $\text{disc} < 0$ ，计算两个复根： $x_{1,2} = (-b \pm \sqrt{\text{disc}i})/(2a)$ 。

算法的 N-S 图如图 1.6 所示。其中，计算二次方程的根的算法如图 1.7 所示。

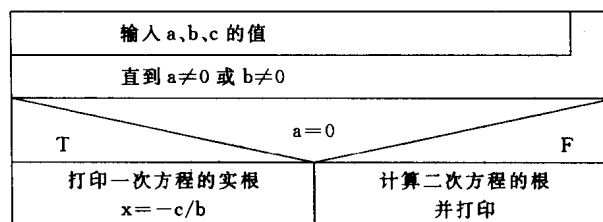


图 1.6

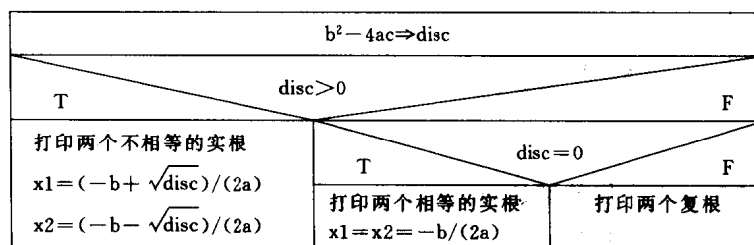


图 1.7

1.10 将 a, b, c 三个变量的值互换。即将 a 的原值送给 b, b 的原值送给 c, c 的原值送给 a 。

解：初步算法：

S1：输入 a, b, c 的值。

S2：将 a, b, c 的值互换。

S3：输出 a, b, c 的值。

设变量 m 为临时存储单元。算法的 N-S 图如图 1.8 所示。

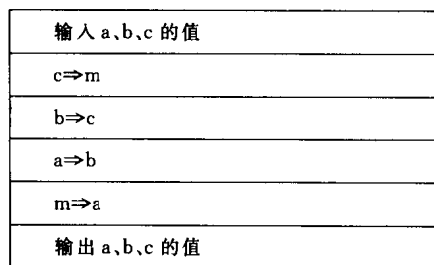


图 1.8

1.11 给定一个年份，判断是不是闰年。

解：闰年的条件是：能被 4 整除且不能被 100 整除，或能被 400 整除的年份。

初步算法：

S1：输入年份 $year$ 。

S2：判断 $year$ 是否是闰年。

S3：输出结果。

设变量 $leap$ 来记 $year$ 是否是闰年。 $leap=1$ 表示 $year$ 是闰年； $leap=0$ 表示 $year$ 不是闰年。算法的 N-S 图如图 1.9 所示。

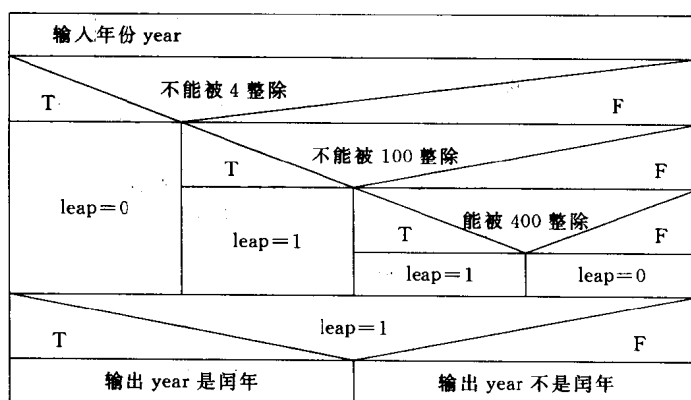


图 1.9

1.12 验证：任何一个大于 2 的偶数都是两个素数之和。请将 6~100 之间的偶数表示为两个素数之和(如 $6=3+3, 8=3+5$ 等)。这就是著名的哥德巴赫猜想。

解：初步算法见图 1.10。其中，求素数 A, B 使 $N=A+B$ 的算法如图 1.11 所示。

判断 B 是素数的算法(见图 1.12)

(1) 根据素数的定义,只能被 1 及自身整除的数为素数。对任一整数 B ,用 $2 \sim B-1$ 中的每一个数去除 B ,如果都不能除尽,则 B 是素数,否则 B 不是素数。但实际上,并不需要用 $2 \sim B-1$ 范围内的所有数去除 B ,而只需在 $2 \sim \sqrt{B}$ 范围内判断即可。因为,如果存在 $B_1、B_2(B_1 \leq B_2)$ 使 $B=B_1 * B_2$,则必有 $B_1 \leq \sqrt{B}$ 。因此,如果 $2 \sim \sqrt{B}$ 中不存在整除 B 的数,则 $\sqrt{B}+1 \sim B-1$ 范围内也一定不存在整除 B 的数。

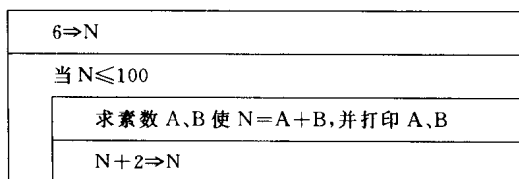


图 1.10

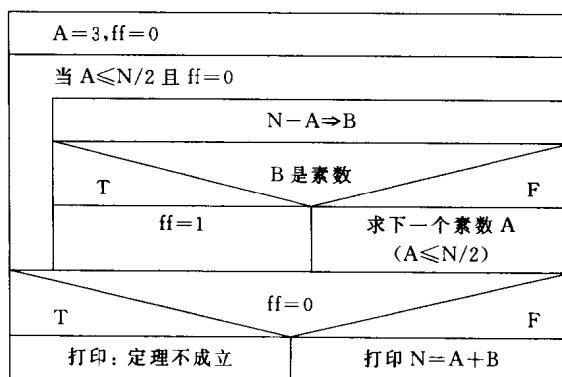


图 1.11

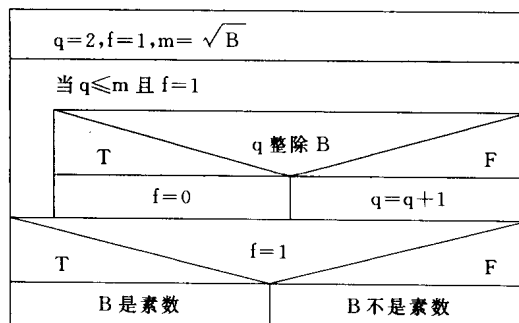


图 1.12

设变量 f 来标识是否判断出 B 是素数。如已判断出 B 不是素数,则 $f=0$,否则 $f=1$ 。

(2) 求下一素数 $A(A \leq N/2)$ 的算法

当 $A \leq N/2$ 且 A 不是素数时,重复执行以下操作:

S1: $A+2 \Rightarrow A$ 。

S2: 判断 A 是否是素数。

设变量 fa 标识是否找到下一个素数 A 。算法的 N-S 图如图 1.13 所示。