

中小学生的理想计算机课外辅导读物 信息学(计算机)奥林匹克竞赛的最好教材

信息学[计算机]奥林匹克

基础篇上册

俞仁励 陈琦 等编著 北京大学出版社

信息学
[计算机]
奥林匹克丛书



G634.67

信息学(计算机)奥林匹克丛书

信息学(计算机)奥林匹克新版

(基础篇上册)

俞仁励 陈 琦
朱鹤翔 李燕平 编著
田亚农

北京  大学出版社

内 容 简 介

本书是一本 BASIC 语言程序设计的入门教材, 适合于学习过 BASIC 语言的基本语句和程序设计的基本方法(相当于《信息学(计算机)奥林匹克新版(预备篇)》)的读者使用。

本书用大量精心挑选的例子介绍了程序设计的一些基本算法和解决有关整数问题、逻辑推理、字符图形等问题的方法。在引导读者程序设计入门的同时, 本书不失时机地讲解了结构化程序的设计方法和程序调试方法, 帮助读者从一开始就养成良好的程序设计风格。本书每章后附有充实的习题, 供读者练习。

本书可作为中小学计算机课外活动的教材, 也可供计算机教师和程序设计爱好者参考。

图书在版编目(CIP)数据

信息学(计算机)奥林匹克: 基础篇 上册/俞仁励等
编·一北京: 北京大学出版社, 1996. 3
ISBN 7-301-03077-0

I. 信… II. 俞… III. 电子计算机-程序设计-基本知识 IV. TP311

书 名: 信息学(计算机)奥林匹克新版(基础篇上册)

著作责任者: 俞仁励 陈 琦 等

责任编辑: 郭佑民

标准书号: ISBN 7-301-03077-0/TP·0285

出 版 者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内 100871

电 话: 出版部 2752015 发行部 2559712 编辑部 2752032

排 印 者: 中国科学院印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787×1092 毫米 16 开本 8.5 印张 22 千字

1996 年 4 月第一版 1996 年 4 月第一次印刷

定 价: 13.00 元

序 言

科学技术普及工作是传播科学知识、科学思想和科学方法,提高全民科学文化素质的关键措施,是实施“科教兴国”战略,加速科学技术进步,促进社会主义物质文明和精神文明建设的重要内容。在科技竞争、人才竞争已成为世界各国竞争焦点的今天,科学技术的普及活动也已成为衡量综合国力的一个重要尺度。从培养 21 世纪人才着眼,抓好青少年科技教育,是一件十分有意义的事情,这也是我们编写这套丛书的初衷。

众所周知,现代信息技术正在对人类社会产生着难以估量的深远影响,并将成为新世纪的一个标志。将有关信息科学知识纳入到未来人才的知识结构中来,是时代的需要。人们越来越认识到:计算,跟语言一样是人类社会每时每刻都不可缺少的事情;计算已经成为与理论研究和实验研究并列的第三种研究方法;计算机成了“人类通用智力工具”;计算机和现代通信网络结合成了教育面向世界、面向未来、面向现代化的支撑环境;信息技术与基础教育相结合已经成为当今世界的大趋势。国家教委副主任柳斌同志在论述“为什么要重视计算机教育”时说:“经验证明,计算机技术越是高度发展,计算机人才就越重要,计算机教育就越重要。只有培养一批又一批掌握现代化已经成熟的电子计算机技术人才,并不断发展和提高我国的计算机技术水平,我们才能加速我国走向现代化,走向世界,走向未来的历史进程。”

信息学奥林匹克竞赛最根本的一条宗旨是推动计算机在青少年中的普及。从 1989 年到 1995 年已经成功地举行过七届世界大赛。在这七届大赛中,我国的青少年选手取得了名列前茅的好成绩。这表明,中国孩子有志向、有能力、有条件掌握先进的信息科学知识,具备成为合格的跨世纪人才的素质条件。学科奥林匹克对青少年树立自立、自信、自尊、自强精神,起到了激励和鼓舞作用。

信息学奥林匹克竞赛属于课外因材施教活动,重在培养能力,特别是创造能力。按照教育心理学的观点,创造能力的培养必须以实践为基础。在计算机上编程解题是一项极好的实践活动,它可以训练观察能力、逻辑思维能力、抽象化形式化描述问题能力、规划能力、动手动脑分析问题和解决问题的能力。总之,这是一项用电脑帮助开发人脑的有益活动。正因为如此,热心于普及教育的老师们集体编著了这套丛书。这套丛书按不同年龄段设计了不同的内容,分层次由浅入深,介绍计算机的典型算法和基本数据结构知识,重点讲解编程解题的思路与技巧,有丰富的例题和习题,许多都是能够引发兴趣的题目,有的有相当难度。这里需要强调:计算机是实践性很强的学科,不上机是学不会的。阅读丛书应和上机实践紧密配合才能真正学懂,学到手。另外,书上给出的方法也只是为了抛砖引玉。我们鼓励创造,我们特别希望读者编出的程序更有效更高明。“精心育桃李,切望青胜蓝”是我们的座右铭。

中国计算机学会普及委员会主任
国际信息学奥林匹克中国队总教练

吴文虎

1995 年 12 月于清华园

前 言

本书是《信息学（计算机）奥林匹克丛书》的新版基础篇上册，是作者们汲取了广大读者的宝贵意见并根据近几年来计算机程序设计活动的新内容所写成。

本篇供学习了基本 BASIC 语言（相当于《信息学（计算机）奥林匹克新版（预备篇）》）后有兴趣继续深入学习计算机程序设计的读者选用。本篇共三册即正文上下两册及习题集一册。上册结合程序设计经常遇到的问题类型详细地逐步地介绍一些基本算法，下册则是对上册提出的问题与算法从综合角度出发，深入一步去研究和讨论。每章都附有难易适当的习题，两册的习题及其解法与相应程序都编入在习题集中。这样做的目的是为了使读者能较顺利地越过程序设计入门的屏障，实现质的飞跃。

上、下两册的有些章节的章目类似，读者可以根据自己的情况来决定学习方法。如可以按章节内容把上下两册的某些章节合并学习；也可以先学习上册对一些基本算法有了初步的理解和掌握后再学习下册。

考虑到大多数读者学习计算机程序设计的入门语言是 BASIC 语言，书中所列程序皆用 GW BASIC 语言写成并调试通过。使用其他版本 BASIC 语言的读者，有些程序可以直接录入运行，有些则要作适当修改。本册的第八章第一节提供了 QUICK BASIC 语言与 GW BASIC 语言关于程序的三种基本结构的语句构成规范，可供读者修改时参考。

本篇除了上述的一些特点外，还有两个特点提请读者注意：

(1) 青少年程序设计所涉及的问题类型十分广泛，如果读者对于问题所涉及的学科没有基本常识，则程序设计无从做起；再则由于程序设计活动的广泛开展，程序设计的题目类型越来越广泛，难度也越来越大；程序设计者不能“完全”把求解的任务推给计算机，设计者应当对题目作较充分的分析，用较优的算法去求解，因此作者希望读者学点奥林匹克数学和看点科普读物。

(2) 程序设计不是语句堆积，要十分注重程序的结构性和可读性，要从一开始就养成良好的程序设计习惯。为此，读者可以先阅读上册的第八章“程序设计方法”。

上面两点渗透在本篇的各章节中。

本篇可作为中小学计算机课外小组的教材，也可供计算机教师和计算机程序设计爱好者参考使用。

本篇的作者都是长期在第一线做计算机普及教育工作的教师。他们辅导中小學生参加过各类计算机竞赛，并取得较好的成绩。他们有着丰富的教学和辅导经验。本篇是他们多年来呕心沥血一点一滴积累的经验总结。现在把它们毫无保留地奉献给广大读者。

参加上册编写工作的有：

北京第一六一中学陈琦：第一章至第四章；北京第二中学朱鹤翔：第七章，第九章第一节至第五节；北京实验大学李燕平：第五章至第六章；北京房山第二中学田亚农：第八章；北

京第一六一中学俞仁励：第九章第六节；全书由俞仁励统稿。

本书的编写得到北京大学出版社郭佑民老师的大力支持和帮助，谨此表示感谢。

由于编写时间仓促且作者水平所限，书中难免会有错误和不妥之处，敬请读者提出宝贵意见和建议。

俞仁励

1995年10月9日于北京

目 录

第一章 编程中常用的整数性质	(1)
第一节 整数及有关的一些判断.....	(1)
第二节 整数的数字分离和局部截取.....	(2)
第三节 素数、素因数分解、最大公约数、最小公倍数.....	(5)
第四节 平方数、立方数问题.....	(7)
第五节 综合举例.....	(9)
习题一	(11)
第二章 编程的基础方法	(13)
第一节 顺序法和逆算法	(13)
第二节 穷举法	(14)
第三节 模拟法和公式法	(17)
习题二	(23)
第三章 字符串处理	(25)
第一节 字符串函数	(25)
第二节 字符串的比较和排序	(27)
第三节 字符串的基本操作	(29)
习题三	(30)
第四章 文本图形	(32)
第一节 基本几何图形	(32)
第二节 对称图形	(33)
第三节 具有“空缺”(空心)的图形.....	(35)
第四节 连续图形	(37)
习题四	(38)
第五章 逻辑表达式与逻辑判断	(42)
第一节 逻辑值与逻辑运算	(42)
第二节 简单的逻辑推理	(44)
第三节 综合练习	(45)
习题五	(47)
第六章 综合训练题型	(50)
第一节 由程序写出运行结果	(50)
第二节 按要求完成或修改程序	(53)
第三节 编写程序	(55)
第四节 编程并上机调试	(58)
习题六	(60)

第七章 综合举例	(65)
习题七	(93)
第八章 程序设计方法	(94)
第一节 三种基本程序结构	(94)
第二节 结构化程序设计方法	(100)
第三节 程序编写风格	(104)
第四节 程序设计的准则	(107)
习题八	(108)
第九章 调试程序	(110)
第一节 什么叫调试程序	(110)
第二节 语法错误处理	(110)
第三节 输入错误处理	(113)
第四节 逻辑错误处理	(113)
第五节 优化程序	(113)
第六节 程序调试综合举例	(118)
习题九	(126)

第一章 编程中常用的整数性质

在 BASIC 程序设计中,很大部分是用来解决数学计算方面的问题,在文本图形或高精度作图的程序中,也常需要数学计算方面的知识。因此为了深入学习 BASIC 编程的知识及技巧,我们先学习一些用 BASIC 语言、表达式或程序来解决基本数学方面的知识。

第一节 整数及有关的一些判断

一、整数的判断

若 $X = \text{INT}(X)$ 则 X 为整数。若同时有 $X > 0$ 则 X 为正整数,否则 X 为负整数或零。

若 $X \neq \text{INT}(X)$ 则 X 不是整数。

二、整除的判断

在 $A \div B$ 的除法运算中

若 $A/B = \text{INT}(A/B)$, 则 A 能被 B 整除。

若 $A/B \neq \text{INT}(A/B)$, 则 A 不能被 B 整除。

有时也用 $P = A/B$ 判断,若 $P = \text{INT}(P)$, 则 A 能被 B 整除(其中 P 为 A 被 B 除所得的商)。

也可以用 $Q = A - \text{INT}(A/B) * B$ 判断,若 $Q = 0$, 则 A 能被 B 整除(其中 Q 为 A 除以 B 的余数)。

在 GW BASIC 中,另有求模运算符 MOD,如 $Q = A \text{ MOD } B$, 即 Q 为 A 被 B 除后的余数;但应用这个运算符时应该注意:被除数 A 与除数 B 必须是整数,否则会先自动将 A 与 B 取整而得出误差很大的结果。

三、奇偶数的判断

由上述原理可以推导出:

若 $X/2 = \text{INT}(X/2)$, 则 X 是偶数。

若 X 为整数也可以用 $X/2 \neq \text{INT}(X/2)$ 来判断 X 为奇数。若 X 不一定是整数时,则不可用这种方法来判断 X 是否是奇数,可以用 $(X+1)/2 = \text{INT}((X+1)/2)$ 来判断 X 是奇数。

四、余数的判断

当 A 被 B 除后的余数为 C 时,判断方法是: $(A-C)/B = \text{INT}((A-C)/B)$ 成立,或 $C = A - \text{INT}(A/B) * B$ 成立。在 GW BASIC 中则可以直接用 MOD 运算符 $C = A \text{ MOD } B$ 来判断。

【例 1】 某四位数被 2, 3, 4, ..., 9, 10 去除时,它的余数分别是 1, 2, 3, ..., 8, 9, 求出所有具有这种性质的四位数。

分析:由此数被 2 除余 1,可知此数一定是奇数,又此数被 10 除余 9,此数的个位数字一定是 9,又因 9999 能被 9 整除,所以只需从 1009 到 9989 对每 10 个数去一一列举寻找即可。

程序如下:

```
10 FOR X=1009 TO 9989 STEP 10
20 FOR I=3 TO 9:IF (X-I+1)/I<>INT((X-I+1)/I) THEN 40
30 NEXT I:PRINT X,
40 NEXT X:END

第 20 句也可以写为 20 FOR I=3 TO 9: IF X MOD I<>I-1 THEN
40
```

想一想:①如果求具有这样性质的最小的四位数,怎样修改这个程序?②能否把 I 循环的初值改为 6?

【例 2】 已知某数是 2, 3, 5 的倍数,被 11 除余 10,被 13 除余 7,被 17 除余 13。求满足这些条件的最小自然数。

分析:由于此数是 2, 3, 5 的倍数,而 2, 3, 5 的最小公倍数是 30,某数一定也是 30 的倍数。所以可用变量 X 的值每次增加 30 去一一列举判断。又由于各个除数与余数之间没有明显的数学关系,所以可用 READ/DATA 语句来解决这个问题。想一想 RESTORE 语句起什么作用?

程序如下:

```
5 X=0
10 X=X+30:RESTORE
20 FOR I=1 TO 3:READ A,B
30 IF(X-B)/A<>INT((X-B)/A)THEN 10
40 NEXT I:PRINT X:END
50 DATA 11,10,13,7,17,3
```

第 30 句也可以直接用求模运算符写作 30 IF B<>X MOD A THEN 10。

第二节 整数的数字分离和局部截取

在一些数值计算中,需要将一个数分离成各位数字或者截取出其中的某几位。这就需要应用数字分离或局部截取的方法。

我们掌握了以下三种方法以后,就可以根据题意及题目类型熟练灵活地使用其中的某一种方法。

一、取整函数法

这种方法适用于位数不多或者只取其中某一位或某一段数字的情况;而且分离截取后可直接用于计算。

若 X 为一个四位整数,则它的各位数字如下:

千位数字 $A = \text{INT}(X/1000)$

百位数字 $B = \text{INT}(X/100) - 10 * A$

十位数字 $C = \text{INT}(X/10) - 100 * A - 10 * B$

个位数字 $D = X - 1000 * A - 100 * B - 10 * C$

这种方法适合于各位数字连续分离, 如果只分离或截取其中某一位或某一段, 可用下面的方法:

如只取百位数字时, 可取 $\text{INT}(X/100) - \text{INT}(X/1000) * 10$ 。

若把一个五位数 X 分别截取成前三位和后两位时, 则可用:

前三位数 $A = \text{INT}(X/100)$

后两位数 $B = X - \text{INT}(X/100) * 100$ 或 $B = X - A * 100$

其他类似情况可以此类推。

【例 3】 求满足下列条件的三位数:

(1) 它的各位数字不同且不为零;

(2) 这个数等于所有由它的各个数字所组成的两位数的和。 [132, 264, 396]

分析: 由三个不为零的数字 A, B, C 分别组成的两位数包括 AB, BA, AC, CA, BC, CB , 可知 A, B, C 这三个数字在十位及个位中各出现过两次, 所以这些两位数的和可概括成 $22 * (A + B + C)$ 。

程序如下:

```
10 FOR X=123 TO 987
20 A=INT(X/100):B=INT(X/10)-10*A:C=X-100*A-10*B
30 IF A=B OR B=C OR A=C OR B=0 OR C=0 THEN 50
40 IF 22*(A+B+C)=X THEN PRINT X
50 NEXT X:END
```

【例 4】 99^{99} 的值最后的三位数字分别是几? [899]

分析: 如果先用 99 累乘 99 次, 再将所得的积截取出最后三位数, 显然是不可能的, 因为其积早已“溢出”而使程序中止运行, 所以要每自乘一次就截取出最后三位。

程序如下:

```
10 X=1
20 FOR I=1 TO 99: X=X*99
30 X=X-INT(X/1000)*1000
40 NEXT I: PRINT X:END
```

二、求商取余数组法

这种方法适用于不大于 8 位的自然数, 而且从低位逐次向高位分离出各位数字。将分离出来的各位数字的值赋值于数组中。可以直接用来运算、排序、比较、交换等。

先由算术式的实例来分析这种编程方法的原理及步骤。如分离 128 这个数的各位数字:

10	128	
10	12	……余 8 (个位数字), 存于数组 A (1) 中
10	1	……余 2 (十位数字), 存于数组 A (2) 中
	0	……余 1 (百位数字), 存于数组 A (3) 中

这就是说, 每一步被 10 除所得的商即成为下一步运算的被除数, 而余数则是被分离出来

的一位数字。直到所得的商是 0 时为止。

【例 5】 将一个不大于 7 位的自然数 X 的各位数字分离，并按从高位往低位的顺序分行打印出来。

分析：原理如前面所述。在程序中各变量所代表的定义如下：P——商，X——被除数，Q——余数，N——数组下标。

程序如下：

```
10 INPUT X;IF X>INT(X) OR X<=0 OR X>=1E+07 THEN 10
15 N=0;DIM A(7)
20 P=INT(X/10);Q=X-P*10;N=N+1;A(N)=Q
30 IF P>0 THEN X=P;GOTO 20
35 PRINT
40 FOR I=N TO 1 STEP -1;PRINT A(I);NEXT I;END
```

注：当 X 不超过 15 位数时，可以用双精度变量，记作 X#。

三、子字符串函数法

这种方法适用于：(1) 多位数字 (255 位之内)；(2) 可以由低位往高位，也可以由高位往低位逐位分离；(3) 也可以任意截取或分离出数字中的任何一位或几位；(4) 由各位数字组合成一个多位数也很方便。

由于把数字转变成了数字字符构成的字符串，所以分离、截取或合成后不能直接用于数学运算。必须先用 VAL(X) 函数把字符串转化成数值后才可用来进行数学运算。

另外在 GW BASIC 语言中，把一个正数或 0 用 STR\$(X) 函数转化成数字字符的字符串后，它的长度比原数的实际长度增加一位 (符号位)，在使用时应该注意这一点。

【例 6】 由键盘输入一个多位自然数，再将它重新排列成一个位数不变，与原数相同数字组成的最大的数。

分析：为了避免所输入的数位数大于八位而形成科学记数法的形式或“溢出”，应直接以字符串的形式输入，这样还可以避免第一位符号位的出现。程序中第 20 句、30 句是为保证所输入的是自然数的字符串。第 40 句则是将各位数字分离后存放于字符串数组中。第 50—80 句将分离后的各位数字按降序排序，并组合成一个新的字符串 Y\$。排序的原理则是利用了 0—9 这十个数码大小与它们的 ASCII 码的大小相一致的原理。

程序如下：

```
10 INPUT X$;L=LEN(X$);IF LEFT$(X$,1)="" THEN 10
20 FOR I=1 TO L;IF MID$(X$,I,1)<"0" OR MID$(X$,I,1)>"9" \
THEN 10
30 NEXT I;DIM A$(L);Y$=""
40 FOR I=1 TO L;A$(I)=MID$(X$,I,1);NEXT I
50 FOR I=1 TO L-1;FOR J=I+1 TO L
60 IF A$(I)<A$(J) THEN T$=A$(I);A$(I)=A$(J);A$(J)=T$
70 NEXT J;Y$=Y$+A$(I)
80 NEXT I;Y$=Y$+A$(I)
```

```
90 PRINT Y$:END
```

注：也可省略第 40 句分离数字这一步，用替换语句将字符串中的字符直接交换。这时将第 60 句改为：

```
60 IF MID$(X$,I,1) < MID$(X$,J,1) THEN T$ = MID$(X$,I,1):  
MID$(X$,I,1) = MID$(X$,J,1):MID$(X$,J,1) = T$ 同时也用不着连接成字符串  
Y$ 这一步了。
```

第三节 素数、素因数分解、最大公约数、最小公倍数

一、素数

大于 1，并且除 1 和它自身外，无其他因数的自然数叫素数（在小学算术课本中叫质数）。

2 是最小的素数，除 2 以外，所有的偶数显然不是素数。

为了减少循环次数，对于大于 3 的自然数，判断它是否是素数，可用以下的程序。

【例 7】 打印出 100—200 之间的所有素数。

分析：因除 2 以外所有的偶数都不是素数，所以只需要从 101 至 199 之间的奇数中来找素数。由于先把偶数排除在外，所以也用不着用偶数做除数（因数）来判断了。只需要用 $3-\sqrt{X}$ 之间所有的奇数当除数来进行判断就可以了。

SQR(X) 函数叫平方根函数，也叫开平方函数，它相当于数学上的 $\sqrt{X}$ ，即求一个这样的数，它自己乘自己等于 X，例如： $\sqrt{9}=3$ ，因为 $3 \times 3=9$ 。

程序如下：

```
10 FOR X=101 TO 199 STEP 2  
20 FOR I=3 TO SQR(X) STEP 2  
30 IF X/I=INT(X/I) THEN 50  
40 NEXT I:PRINT X,  
50 NEXT X:END
```

二、素因数分解

任何大于 1 的非素数整数叫合数，合数都可以分解成为素数的乘积，叫做素因数分解，在小学数学中叫做分解质因数。

我们先分析小学数学课本中所学过的把一个数分解成质因数的方法，从中找到 BASIC 编程的原理和步骤。

例如，把 2200 分解质因数：

算 法	说 明
2 2200	……除数从 2 开始来试除
2 1100	……商 1100，作为下一步的被除数
2 550	……再继续用 2 来试除，商为 550
5 275	……2, 3, 4 不能去整除 275，除数增加到 5
5 55	
11 11	……除数从 5 增加到 11

1最后当商为1时,分解过程结束

所以 $2200 = 2 \times 2 \times 2 \times 5 \times 5 \times 11$

【例8】 键盘输入一个大于3的合数,把它分解成素因数连乘积的形式。

分析:下面程序中的第20句、30句是为了保证输入的数是“大于3的非素数”。在编程中这一步是必要的步骤,我们称做输入屏蔽,在竞赛中上机答卷和“黑匣子”阅卷中尤其必不可少(详见第九章)。

为了简化程序,第25句 THEN 后面的语句是为了判定合数之用,也是为了打印第一个素因数而用。

另外为了简化程序,在第20句中的除数和第50句中除数(因数)也没有按素数或奇数去找,而是每次的记数都增加1。

程序如下:

```
10 INPUT X:IF X<4 OR X<>INT(X) THEN 10
20 FOR I=2 TO SQR(X)
25 IF X/I=INT(X/I) THEN PRINT X;"=";I;:X=X/I;GOTO 40
30 NEXT I:PRINT"NO!";GOTO 10
40 IF X/I=INT(X/I) THEN PRINT" * ";I;:X=X/I;GOTO 40
50 IF X<>1 THEN I=I+1;GOTO 40
60 END
```

三、求最大公约数

在数学中求两个自然数的最大公约数(最大公因数)的方法如下。以求100,140两数的最大公约数为例:

算 法	说 明
$\begin{array}{r rr} 2 & 100 & 140 \\ \hline & 50 & 70 \end{array}$	从2开始试除,整除后的商作为新被除数
$\begin{array}{r rr} 2 & 25 & 35 \\ \hline & 5 & 7 \end{array}$	不能被2整除后,再依次用3,4,5去除
	当以6试除时,6大于其中的一个被除数,运算结束

所以100与140的最大公约数为 $2 \times 2 \times 5 = 20$

【例9】 键盘输入两个自然数,求出它们的最大公约数。

分析:最大公约数是两数的公因数的累乘之积,所以要用累乘器,并且先将其赋值为1。具体每个步骤和过程的原理与例8和前面的算术式分析是一样的。

程序如下:

```
10 INPUT A,B:P=1:I=2
20 IF A/I=INT(A/I) AND B/I=INT(B/I) THEN P=P*I:A=A/I:B=B/I:
   GOTO 20
30 I=I+1:IF I<=A AND I<=B THEN 20
40 PRINT P:END
```

如果求两个以上自然数的最大公约数,可以先求出两个数的最大公约数,再求这个数与第三个数的最大公约数,……,依此类推下去。这时应当把求二个自然数最大公约数的程序

段写成子程序。

四、求最小公倍数

【例 10】 求两个自然数的最小公倍数。

方法一：以其中一个自然数为循环变量的初值和步长值，在循环过程中循环变量的值逐倍增加，当它又是另一个自然数的整倍数时，此时循环变量的值就是它们的最小公倍数。

程序如下：

```
10 INPUT X,Y:IF X<>INT(X) OR Y<>INT(Y) THEN 10
20 FOR I=X TO X*Y STEP X
30 IF I/Y=INT(I/Y) THEN PRINT I:END
40 NEXT I
```

方法二：我们先用数学方法看，求 100 与 140 两数的最小公倍数。列竖式求：

$$\begin{array}{r|rr} 2 & 100 & 140 \\ \hline 2 & 50 & 70 \\ \hline 5 & 25 & 35 \\ \hline & 5 & 7 \end{array}$$

所以它们的最小公倍数为： $2 \times 2 \times 5 \times 7 = 700$ 。

因它们的最大公约数为 $2 \times 2 \times 5 = 20$ ，所以最小公倍数也可以写成 $20 \times 5 \times 7 = 700$ 。

如果用代数式表示，求 A 与 B 的最小公倍数如下：

$$\begin{array}{r|rr} P_1 & A & B \\ \hline P_2 & A_1 & B_1 \\ & A_2 & B_2 \\ & \cdot & \cdot \\ & \cdot & \cdot \\ & \cdot & \cdot \\ \hline P_n & \cdot & \cdot \\ & a & b \end{array}$$

则 A 与 B 的最大公约数 $P = P_1 \times P_2 \times \cdots \times P_n$

而 $P \times a = A$ ， $P \times b = B$

所以它们的最小公倍数为：

$$\begin{aligned} Q &= P \times a \times b = P \times a \times b \times \frac{P}{P} = \frac{P \times a \times P \times b}{P} \\ &= \frac{A \times B}{P} \end{aligned}$$

由此得出结论：两数的最小公倍数是这两个数的积被它们的最大公约数去除。

它们的 BASIC 程序请同学们参考例 9 求最大公约的程序补充完成。

第四节 平方数、立方数问题

为了说明有关平方数和立方数问题在编程时应该注意什么事项，我们先看看下面程序的运行结果。

```
10 INPUT M,N:IF M<>INT(M) OR N<>INT(N)
   OR M>=N THEN 10
```

```

20 FOR I=M TO N:X=I*I
30 IF I=SQR(X)THEN PRINT I;"YES!";GOTO 50
40 PRINT I;"NO!";
50 NEXT I:END

```

从数学运算规则来看,应该全部都打印成“YES!”,可是这个程序运行的结果却是绝大多数都打印成了“NO!”。

这是因为 GW BASIC 语言在进行乘方与开方运算时有一定的误差,并且运行的速度也比乘除、加减法慢。

为了解决这类问题,可利用修正误差的方法,如把 30 句的关系表达式改成:

```
ABS(I-SQR(X))<1E-6
```

另一种方法就是采取从平方根来计算平方数,也就是用乘法来代替开方运算。这种方法一方面避免了在开方时系统本身的误差问题,又可以大大减少运算次数。

同样道理,在乘方运算中也有这种问题,如将程序中的 30 句改为:

```
30 IF X=I*I THEN PRINT I;" YES!";GOTO 50
```

所以在编程时,要把乘方的运算尽量写成乘法运算的形式。

为了由平方根来求平方数,我们应记住以下规律:

平方数的位数	循环变量初值	循环变量终值
1	1	3
2	4	9
3	10	31
4	32	99
5	100	316
6	317	999
7	1000	3162
8	3163	9999

【例 11】 编程找出六位平方数,且它的前三位与后三位都是平方数。

[225625, 114400, 324900]

分析:六位平方数的平方根为 317—999,三位平方数的平方根为 10—31,由平方根求出平方数既克服了系统中本身的误差,又大大的减少了运算次数。下面程序的外循环是求出六位的平方数,并把它局部截取成前三位和后三位两个数。内循环是判断这两个三位数是否都是平方数。

程序如下:

```

10 FOR I=317 TO 999: X=I*I
20 A=INT(X/1000);B=X-A*1000;K=0
30 FOR J=10 TO 31: Y=J*J
40 IF A=Y OR B=Y THEN K=K+1

```



```

50 NEXT J: IF K=2 THEN PRINT X
60 NEXT I: END

```

【例 12】 用 0—9 这 10 个数不许重复拼凑出两个自然数，让它们分别是同一个数的平方和立方数。
〔 $69^2=4761$, $69^3=328509$ 〕

分析：下面程序中的 30 句到 50 句是为了检索这 10 个数字是否是“不许重复”的。

程序如下：

```

10 FOR I=48 TO 99: A=I*I: B=A*I
20 X$=STR$(A)+STR$(B): IF LEN(X$)<>12 THEN 60
30 FOR J=2 TO 11: FOR K=J+1 TO 12
40 IF MID$(X$,J,1)=MID$(X$,K,1) THEN 60
50 NEXT K,J: PRINT I;"^2=";A,I;"^3=";B: END
60 NEXT I

```

【例 13】 如果有一对自然数，它们的和与差是一对完全平方数，我们称这两个数为“亲和数”，例如 8 和 17 这对数： $17+8=25$ ， $17-8=9$ ，25 和 9 都是平方数，因此 8 和 17 是一对“亲和数”，编程求出 100 以内的所有亲和数。

分析：因为求 100 以内的两个“亲和数”，所以这两数之和一定小于 200，又因其和是一个平方数，所以两数之和最大值不会大于 196（即 14^2 ）根据数学四则应用题所学过的，大数 = (和 + 差) ÷ 2，小数 = (和 - 差) ÷ 2。可以由两数的和与差求出这两个数。下面程序中各变量所表示的意义如下：X 为两数之和，Y 为两数之差，A 为大数；B 为小数。根据下式： $A+B=X$ ， $A-B=Y$ ，如果 X，Y 都是整数，则 A 与 B 也一定同时是整数，所以在 40 句中只需判断 A 与 B 中有一个是不是整数就可以了。

程序如下：

```

10 FOR I=2 TO 16: X=I*I
20 FOR J=1 TO I-1: Y=J*J
30 A=(X+Y)/2: B=(X-Y)/2
40 IF A<100 AND A=INT(A) THEN PRINT A;" ";B,
50 NEXT J,I: END

```

第五节 综合举例

通过前面几节例题的编程实例，我们可以看出，在编写解决有关数值计算和性质的程序中，一般的原则是：

外层循环——控制数值的范围。

内层循环——控制数值的特点和性质。

如果数值的范围已知——一般用循环语句。

如果数值的范围未知——一般用计数器、累加器和条件语句来控制。

【例 14】 求连续若干个自然数，使其之和为 1000，共有多少组这样的数，并分别打印出它们的算式来。

分析：因连续的自然数累加之和为 1000，所以只需从 1—500 中去穷举寻找就可以了。下