



中小学信息学 奥林匹克难题详解

吴清俊 韩天刚 贾海 冯平 编著

化学工业出版社

中 小 学

信息学奥林匹克难题详解

吴清俊 韩天刚 贾 海 冯 平 编著

化学工业出版社
· 北 京 ·

(京)新登字 039 字

内 容 提 要

本书从历届各地中小学计算机竞赛试题中精选出具有代表性的难题十八个,其中包含了循环、判断、数组的典型应用,涉及到计算机制作音乐、图画、动画、文件处理、链表应用、逻辑推理、求反等方面的知识。围绕这些题目以题解、分析、讨论、深化、推广等环节展开了详细地讨论。有些题目还特别列出了多种解答,给读者以启发。最后对每个类型的题目还做了一般性归纳,使读者能做到举一反三,从而提高解决相似题目的能力。

本书适合于中、小学计算机爱好者学习使用。可做为中、小学生计算机辅导教师的教学参考书,也适用于初步入门学习程序设计的成人使用。

图书在版编目(CIP)数据

中小学信息学奥林匹克难题详解/吴清俊等编著,

—北京:化学工业出版社,1996

ISBN 7-5025-1590-9

I. 中… II. 吴… III. ①计算机课-中学-试题②计算机课-小学-试题 IV. G634.676

中国版本图书馆 CIP 数据核字(96)第 01298 号

出版发行:化学工业出版社(北京市朝阳区惠新里 3 号)

社长:俸培宗 总编辑:蔡剑秋

经 销:新华书店北京发行所
印 刷:北京市通县京华印刷厂
装 订:北京市通县京华印刷厂
版 次:1996 年 7 月第 1 版
印 次:1996 年 7 月第 1 次印刷
开 本:787×1092 1/16
印 张:10 $\frac{1}{4}$
字 数:250 千字
印 数:1—3000
定 价:17.40 元

前 言

自从小平同志号召计算机要从娃娃抓起以来,全国及许多省市每年都要举行信息学奥林匹克大赛。用以选拔人材,激励青少年向信息科学奋进。并且在此基础上从全国青少年中选拔出了参加国际信息学奥林匹克竞赛的代表队,参加了世界大赛。这些优秀的青少年代表在国际大赛中取得了辉煌的成果,为祖国争得了荣誉,为中国人民争了气。信息学奥林匹克竞赛已受到社会各界的重视,面对这一赛事,不少单位把全国各省市的历年的竞赛试题收集成册,并附上解答程序出书发行,受到了各地中小学生的欢迎。但这些试题集多是试题的搜集,所列程序也只是简单的答案,没有讲明为什么要这样解答,也没有给出解题的方法,所以它们只是一种资料性的参考书籍。针对这一情况,并根据七、八年来组织计算机竞赛的实践经验,我们从历年各种试题中精选出具有典型意义的,比较难解的题目十八个。以题为纲,通过分析题意,给出多种解法。在分析对比中,引导学生掌握一些比较好的解题方法,并进行深入分析讨论,使学生获得更多的有关编程的知识和技巧。每个题目,大体上由试题、解题、分析、讨论、深化、推而广之几个环节组成,并给出解题程序。分析讨论中,既有理论分析,也有具体示范,特别是列出了多种解题方法,在对比中给人以启发,有些解法还是很有深意和新意的。书中所有程序都在计算机上运行实现,所以一些学生即使不能读懂书中全部内容,但照着程序演练,也会大有收益,并能取得明显的进步。对于学习程序设计入门的成年读者,也可把本书做为一本科普性、趣味性的编程入门练习书来阅读。

阅读本书的方法最好是先看题目,根据题目,自己先试着编写程序,再把自己编程结果与书中给出的解答相对照,这样收获可能会大一些。有一些题目自己不会解,看完书中解说仍搞不懂时,可以按照书中给出的程序,装入计算机中,进行运行操作,慢慢自可明白其中的意义。

本书十八个题目难易程度不一,我们在每个题目后头标上*号以表明其难易程度。我们认为对于中小学生而言,打*者经过自己思考与阅读,本题是可以读懂的。打**者,经过与同学们之间的商讨、钻研与交流也可以读懂。打***也许要再读一点其他书籍或经高级同学及辅导教师的指点,才能逐步把它搞懂。阅读本书时,先从“*”号入手,逐步去读那些难一点的题目,最后把那些最难的题也攻读下来。当你攻下了一个难题之后,你将会感到其乐无穷。我们盼望同学们要有一股钻劲,用不攻下难题决不罢休的拼劲去解它们。

本书中对部分程序附有用 Quick BASIC 和 C 语言编的程序,供同学们对照参考。用 BASIC 编的程序用 B 字来标识,用 Quick BASIC 语言编的程序用 Q 来标识,用 C 语言编的程序用 C 来标识。

本书由吴清俊、韩天刚执笔编写,贾海校阅了本书有关章节,并用 Quick BASIC 和 C 语言编写了部分解题程序,冯平同志校正了全书的文稿,并将它们录入计算机中,使本书得以成稿,付出了艰苦的劳动。

此书的编写得到河南省计算机学会的支持与鼓励,在此表示感谢。

编者

1995. 11. 20

目 录

第一题	老太太卖鸡蛋 [*]	1
第二题	链环 [* *]	5
第三题	水手分桃 [*]	11
第四题	水仙花数 [*]	15
第五题	打印数字三角形及其他图形 [*]	21
第六题	奇妙的四位数 [*]	30
第七题	处理素数 [*]	38
第八题	数制转换及其应用 (上) [*]	45
第九题	数制转换及其应用 (下) [*]	56
第十题	称重与分币 [*]	64
第十一题	计算机唱歌 [* *]	74
第十二题	计算机作图 [* *]	82
第十三题	醉鼠问题 [* *]	92
第十四题	迷宫问题 [* * *]	103
第十五题	学生成绩报表 [* * *]	111
第十六题	海盗藏宝 [* *]	128
第十七题	约瑟夫问题 [* * *]	135
第十八题	九个数的有趣组合 [*] [* * *]	143
主要参考文献		156

第 一 题

老太太卖鸡蛋

题一：某老太太卖鸡蛋时，总是查不准数，只知道从篮子里每次取 2、3、4、5、6 个时，最后分别都剩一个鸡蛋，当每次取 7 个时就不剩了。请编一程序，帮老太太算篮子里总共有多少鸡蛋。（题分：20 分）[选自 1995 年新乡市小学生计算机竞赛上机试题]。

解：计算机最大优点就是运算速度快，因此在求解问题时，可以用“逐个测试求解法”，也可称为“多次猜测法”或“尝试法”。这种方法在编程时主要要掌握好两个环节。（一）要把逐个猜测对象的数据范围及取值方法确定好，以便逐个选出来进行猜测。（二）要把合乎要求或者不合乎要求的判据确定好，以便对选出来的对象进行取舍。在一般情况下，第一个环节用循环语句来实现，第二个环节用条件判断语句来实现。本题就是一道典型的判断学生是否具有应用“逐个猜测法”来编程解题能力的考题。

解题程序 1:

[B1-1]

```
10 FOR I=7 TO 1000 STEP 7
```

```
20 IF I-INT(I/2)*2 <> 1 THEN 80
```

```
30 IF I-INT(I/3)*3 <> 1 THEN 80
```

```
40 IF I-INT(I/4)*4 <> 1 THEN 80
```

```
50 IF I-INT(I/5)*5 <> 1 THEN 80
```

```
60 IF I-INT(I/6)*6 <> 1 THEN 80
```

```
70 PRINT " I="; I: END
```

```
80 NEXT I
```

[Q1-1. BAS]

```
FOR i = 7 TO 1000 STEP 7
```

```
IF i - INT(i/2) * 2 <> 1 THEN 80
```

```
IF i - INT(i/3) * 3 <> 1 THEN 80
```

```
IF i - INT(i/4) * 4 <> 1 THEN 80
```

```
IF i - INT(i/5) * 5 <> 1 THEN 80
```

```
IF i - INT(i/6) * 6 <> 1 THEN 80
```

```
PRINT "i="; i:END
```

```
80 NEXT i
```

[C1-1. C]

```
#include " math. h"
```

```
main ()
```

```
{
```

```
int i;
```

```
for (i=7; i<=1000; i=i+7)
```

```

{
if (i%2! =1) continue;
if (i%3! =1) continue;
if (i%4! =1) continue;
if (i%5! =1) continue;
if (i%6! =1) continue;
printf (" i=%d\n", i);
}
}

```

解题程序 2:

[B1-2]

```

10 FOR I=7 TO 1000 STEP 7
20 IF I MOD 2<>1 THEN 80
30 IF I MOD 3<>1 THEN 80
40 IF I MOD 4<>1 THEN 80

```

```

50 IF I MOD 5<>1 THEN 80
60 IF I MOD 6<>1 THEN 80
70 PRINT " I="; I; END
80 NEXT I

```

评：过去有些人认为编程语句越少越好，当今编程更提倡编程思路的清晰明了，可读性要好，尤其对小学和初中学生，无需附加注释，就能把程序的思路读明白。由于计算机运算速度的加快，内存容量的加大，一个程序多加十条、二十条语句，并不似以往那样事态严重。所以宁愿多增加几条语句，来使程序的可读性完善。

基于以上标准来看，以上两个解，编写的都很不错。循环语句决定了要进行猜测的对象，用步进为 7，使所选的对象都可被 7 整除，这样的设置是对的。20 到 60 语句是判据，如果通过了这 5 条判据，就是符合要求的对象，而予以打印，否则另选一数，继续往下猜。

用 $I - \text{INT}(I/2) * 2 <> 1$ 的判据与用 $I \text{ MOD } 2 <> 1$ 的判据都是说明该数可被 2 (或 3、4、5、6) 整除而余 1。除了以上两种写法，采用 $(I-1)/2 <> \text{INT}((I-1)/2)$ 这种判法也是可以的，各种判法大同小异，各有千秋，用求整来解，概念比较清晰，用求余来解，编程手法简要。

讨论：(1) 题中 70 语句有 `PRINT " I="; I; END` 句，标准答案也认为得 301 者即为满分，其实根据题意这样的解答是不对的。因为老太太卖鸡蛋，并未给出鸡蛋的上限数，难道不可能是 301、721、1141、1561、1981、2041、2821……等多个数吗？所以不要加 `" : END"` 句，而要让程序不断地循环下去以求出多解，这正是计算机求多解的妙处。本题用 `END` 将循环结束，求出一个解，令人误认为这是唯一解，这就错了。如给出老太太卖蛋数在 1000 个以内，则就有二解，如给出老太太卖蛋数在 500 个以内，则循环语句用 `TO 1000` 是定值错误。

(2) 老太太卖蛋判据共有五个，应当用谁为首，这是很有技巧的问题。题解 1、2 均以 2、3、4、5、6 为序，这是不好的，应当以 6、5、4、3、2 的次序为好。因为 6 通不过，3、2 也必然通不过，判 6 就是判 2、3，而判 2，并不等于判 6。且 6、5、4 数值为大，大者先判，跨距应当由大逐步缩小。同理 6、5、4、3、2 五个判据也没有必要都用，用 6、5、4 三个判据，或者用 5、4、3 三个判据就可以了。因为它们都包含了可被 6、5、4、3、2 整除的内涵。因此程序可以精简，采用三个判据的形式。这是编程的深化，程序简化了，但此时应当进行注释。

解题程序 3:

```

[B1-3]
10 N=7
20 A=N-1
30 IF A/5 <> INT (A/5) THEN 70
40 IF A/4 <> INT (A/4) THEN 70
50 IF A/3 <> INT (A/3) THEN 70
60 PPINT " "; N;
70 N=N+7
80 IF N>3000 THEN PRINT " END":
  END
90 GOTO 20

```

注: 凡能被 5、4、3 整除者, 也能被 6、2 整除, 因为 $5 \times 4 \times 3$ 中含有 2、6 之因素, 用 6、5、4 来整除, 其理一样, 故用三个判据。

解题程序 4:

```

[B1-4]
10 N=N+7
20 FOR J=6 TO 4 STEP-1
30 IF N MOD J <> 1 THEN 70
40 NEXT J
50 PRINT " "; N;
60 IF N >=3000 THEN PRINT
  " END": END
70 GOTO 10

```

解题程序 5a:

```

[B1-5A]
10 I=7
20 I=I+7
30 IF I MOD 2=1 AND I MOD 3=1 AND I MOD 4=1 AND I
  MOD 5=1 AND I MOD 6=1 THEN PRINT I
40 IF I>3000 THEN END
50 GOTO 20

```

解题程序 5b:

```

[B1-5B]
10 FOR I=7 TO 1000 STEP 7
20 K=I-1
30 IF K/2=INT (K/2) AND K/3=INT (K/3) AND K/4=INT (K/4)
  AND K/5=INT (K/5) AND K/6=INT (K/6) THEN PRINT I
40 NEXT I

```

解题程序 5a 与 5b 初看好像很简练, 只用了 4、5 条语句, 但它实质上与解题程序 1、2 并无区别。30 号语句用 AND 运算, 把所有判断都集中在一条语句之中, 它并未简化计算机的运算, 而使判断语句复杂化, 给程序调试时, 带来麻烦和不便, 所以在编程中应尽量不要采用这种多重逻辑判断的结构形式。相比之下还是宜于采用解题程序 1、2 的判断形式, 而不要采用本程序的 A AND B AND C AND D 的这种结构形式。

深化: 换一个思路来想问题, 就是把判据做为选择对象, 把选择对象做为判据来编程, 也就是凡能被 7 整除的对象即为选中了, 那么拿什么来做选择对象呢? 根据以前分析已知要能被 6、5、4、3、2 整除之数必含有 60 这个因素, 30 还不行, 因为 30 这个数不可被 4 整除, 只有 60 是可被 4 整除的因素, 因此它必须是 $(I * 60 + 1)$ 这样一个数, 由此可得解题程序 6。

解题程序 6:

```

[B1-6]
10 FOR I=1 TO 3000 STEP 60
20 IF I/7=INT (I/7) THEN PRINT " "; I;
30 NEXT I
40 END

```

解题程序 6 编写得可谓简而又简了，但也有不足，就是解题思路不太明了。

解题程序 7:

```

[B1-7]
10 X=X+60
20 Y=(X+1)MOD 7
30 IF Y=0 THEN PRINT X+1
40 IF X>3000 THEN END
50 GOTO 10

```

解题程序 7 的方法与解题程序 6 大同小异，可以参考。当然用解题程序 8 的方法来求解也是可以的，但 FOR I=7 TO 3000 STEP 7 步距不如 FOR I=1 TO 3000 STEP 60 步距大，所以其搜索速度不如后者快，因此解题程序 6、7 比解题程序 8 好。

解题程序 8:

```

[B1-8]
10 FOR I=7 TO 3000 STEP 7
20 IF I MOD 60=1 THEN PRINT " "; I;
30 NEXT I

```

推而广之：请编程序求解以下两题。

题二：现在仓库有一堆货，用小车运出，如果每次运出二包、三包、四包、五包或六包，七包，八包，最后都剩下一包，如果每次运出九包，最后则剩下四包，问这堆货总共有几包？

题三：六一儿童节，组织大型团体操，按三人，四人，五人，六人，七人，八人成一阵列组，最后都剩下 2 人，如果按 9 人组成阵列，最后一组还少 1 人，问这群小朋友共有多少人？

第二题

链 环

题一：设有六位数 1abcde 乘 3 后变为 abcde1。试求这个六位数。（程序最优给满分，否则扣 5 分）（题分：20 分）[选自 1995 年新乡市小学生计算机竞赛上机试题]

解：根据本题题意，可采用逐个猜测的编程方法来求解，关键是要把搜索方法与判别依据确定好。

解题程序 1：

```
[B2-1]
10 FOR A=1 TO 9
20 FOR B=1 TO 9
30 FOR C=1 TO 9
40 FOR D=1 TO 9
50 FOR E=1 TO 9
60 S=100000+A*10000+B*1000+C
   *100+D*10+E
70 H=A*100000+B*10000+C*
   1000+D*100+E*10+1
80 IF S*3=H THEN PRINT S: END
90 NEXT E
100 NEXT D
110 NEXT C
120 NEXT B
130 NEXT A
```

从解题程序 1 来看，编程者的意图就是分别对 A、B、C、D、E 进行猜测，当 $S*3=H$ 时，即为所求之解，打印出结果。这种编程思路明确，只是搜索时间太长。

只有 $7*3=21$ ，E 才能为 7，才合题意，因此 50 号语句可改为 $E=7$ 而不必对 E 进行猜测，90 号语句可以删去成为一个四重循环，加快了搜索的速度。

解题程序 2：

```
[B2-2]
10 FOR X=10000 TO 99999
20 A=100000+X: B=10*X+1
30 IF A*3=B THEN PRINT A: END
40 NEXT X
```

显然，解题程序 2 优于解题程序 1。它的思路是令 $X=abcde$ ，所以 $A=1abcde$ ，令 X 前进一位加 1，则 $B=X*10+1=abcde1$ ，当 $A*3=B$ 时，即搜索到了所求之解，这个思路是很巧妙的。当然这个程序还可改进，因为 X 的最低位数必然为 7，所以可将语句 10 改为如下形式。

```
10 FOR X=10007 TO 99997 STEP 10
```

经过以上改进，搜索跨距加大了 10 倍，搜索速度也有明显的改进。

评论：评论就是把评述与讨论结合在一起进行分析。

首先讨论搜索速度，解题程序 2 要搜索约 10 万个数据，经过改进，也要搜索约 1 万个数据。解题程序 1 也要搜索 6 千 5 百多个数据，所以这两个程序的运行求解速度都是很慢的。在运行过程中，解题程序 1 比解题程序 2 还明显的慢，除了搜索数据要花时间外，解题 1 中语

句 60 与语句 50 也是要花时间的。经过以下改进, 搜索速度可以明显提高, 所以如何选用搜索变量, 如何列写判据式子, 都是很重要的。特别是在一些运行速度较低的机器上, 或者较大程序, 有较多多重使用的程序环节时, 在编程时更要注意这个问题。

对解题程序 1 的改进如下:

[B2-3]

```

10 FOR A=10000 TO 90000 STEP 10000
20 FOR B=1000 TO 9000 STEP 1000
30 FOR C=100 TO 900 STEP 100
40 FOR D=10 TO 90 STEP 10
50 E=7
60 T=A+B+C+D+E
70 S=100000+T
80 H=10*T+1
90 IF S*3=H THEN PRINT S; END
100 NEXT D
110 NEXT C
120 NEXT B
130 NEXT A

```

本题可否进一步简化搜索的程序呢? 当然是有的, 比如本题本来是一个六位数即 $abcde$, 现在确定首位为 1, 此数乘 3 后应为 $abcde1$ 。尾数为 1, 使尾数乘 3 后为 1 者, 只能为 7, 所以只是对 $1abcd7$ 进行猜测, 这里只要猜测 4 个数就可以了。进而推之, $d7 \times 3 = 71$, 3 乘 d 能得 7 的数者只能是 5, 因为 $5 \times 3 = 15$, 5 加上进位数 2 得 7。 $1a \times 3 = ax$, a 应当是几呢? 首先可知 $1 \times 3 = 3$, 3 加上进位数后应为 a 。而在 3 的乘法中, 进位数只有 0、1、2、三个数。 1×3 后再加上进位数, 只有三种可能情况, 即 $3+0=3$ 、 $3+1=4$ 、 $3+2=5$ 。形成进位数为 2 的被 3 乘的数是 7、8、9 均大于 5, 所以 a 是 7、8、9 是不可能的。形成进位数为 0 的被 3 乘的数是 1、2、3。1、2 小于 3, 所以 a 是 1、2 也是不可能的。形成进位数为 1 的被 3 乘的数是 4、5、6。5、6 大于 4, 所以 a 是 5、6 也是不可能的。由此可知 a 只能是在 3、4 之间猜测。这样 $1abcd7$ 就变成了在 $13bc57$ 与 $14bc57$ 之间进行猜测了, 猜测时要搜索的数据范围当然明显缩小。运算速度必然会明显提高, 这样可得解题程序 3a、3b。

解题程序 3a:

[B2-3A]

```

10 FOR A=30000 TO 40000 STEP 10000
20 FOR B=1000 TO 9000 STEP 1000
30 FOR C=100 TO 900 STEP 100
40 D=57
50 T=A+B+C+D
60 S=100000+T
70 H=T*10+1
80 IF S*3=H THEN PRINT S; END
90 NEXT C, B, A

```

解题程序 3b:

[B2-3B]

```

10 FOR X=10057 TO 99957 STEP 100
20 A=10000+X; B=10**+1
30 IF 3*A=B THEN PRINT A
40 NEXT X

```

采用以上方法, 运算速度会明显提高。题意中有这样一句话, 既“程序最优给满分, 否则扣 5 分”, 何谓最优程序, 这涉及到了学术观点与美学观点。出题者可能更欢喜解题程序 2 的做法, 其理由是: 简单, 只用了一个循环。但是他们可能没有考虑到循环虽只有一层, 但要搜索的对象要有十万个, 经过改进, 也要搜索万个数据。解题程序 1, 看似复杂, 但搜索的数据对象只有 6 千个, 所以程序 1 并不一定比程序 2 差。这只有技巧的不同, 并不能称某个程序为最优者。评论一个程序编的是否完善, 应当考虑以下几条: ①编程思路是简单明了的。

②程序运行是可靠的和有效率的。③程序的编写是简单紧凑的。总之对一个程序要做综合分析,要进行深入分析,不要太简单化和表面化。

前面已经谈到为了加快解题的速度,正确选择搜索对象和搜索步距至关重要。因此由 1abcde 要猜测 5 个数或 10 万个数值,改为猜测 1abcd7 要猜 4 个数或 1 万个数值,使五重循环变为四重循环;或者步距由 1 变为 10。进一步推算,把猜测数变为 13bc57 与 14bc57,要猜的数变为 3 个,或者步距变为 100,要猜的数值只有 2 百余个,运算速度明显提高。如果再分析下去,可把要猜的数定界为 13b857 与 14b857,这样只有二十几个数要猜了,同理还可定界为 132857 与 142857,只要猜一个数就可以了,进而定界为 142857,到了这时,还要编什么程序去猜这个数。用计算机解题的奥妙就显示不出来,寻找一个折中的部位,解题既快,又不似演算一道数学题目,这是需要思索和权衡的。要发表的评论是:这个命题出的不完善。一个唯一解,经过逆推很容易得出解来,用这样问题来命题编程,应当说命题工作是不完善的,试问解题程序 4 的程序能算是最优的答案吗?

解题程序 4:

[B2-4]

```
10 FOR X=30857 TO 49857 STEP 1000
```

```
20 A=100000+X; B=X*10+1
```

```
30 IF 3*A=B THEN PRINT A
```

```
40 NEXT X
```

[C2-4. C]

```
main ()
```

```
{
```

```
long int x, a, b;
```

```
for (x=30857; x<=49857; x=x+1000)
```

```
{
```

```
a=100000+x
```

```
b=10*x+1;
```

```
if (a*3!=b) continue;
```

```
printf (" a=%d", a);
```

```
}
```

```
}
```

题二: 设有六位数 Mabcde 乘 3 后变为 abcdeM, 试求这个六位数为几?

解: 本命题只把 1 变为 M, 这样题解就不是 1 而是一个变量 M, 它可以是 1、2、3、4、5、6、7、8、9, 所以题解面宽了, 搜索对象的定界难了, 也不可以简单地用 7、57、857 之类的数定界求解了, 显然编程难度加大了, 要研究的范围广泛得多了。在解这个题时, 首先分析, 要知道 M 只能是 1、2 或 3, 因为大于 3, 在乘 3 之后, 就要产生进位数, 这个 Mabcde 六位数经乘 3 后就会成为 7 位数了, 与题意不符, 所以对 M 只要在 1 至 3 之间猜测。

把解题程序 2 稍加改进就可得到本题的解。它存在的缺点是要搜索的数据太多了, 为了求解, 运算时间长了, 比解题程序 2 运算的时间更长了, 以至让人难以忍受。其所以不易缩短运行时间, 就在于令 M 为一变量后, 确定尾数的定界比 M=1 时更难了。

解题程序 5:

[B2-5]

5 FOR I=1 TO 3

10 FOR X=10000 TO 99999

20 A=I*100000+X: B=10*X+I

30 IF A*3=B THEN PRINT A

40 NEXT X

50 NEXT I

由于程序运行时间太长,要限制对数据的搜索范围,加大搜索的步距,可以改进成解题程序 5a、5b。

解题程序 5a:

[B2-5A]

1 W(1)=7

2 W(2)=4

3 W(3)=1

5 FOR I=1 TO 3

6 Z=W(I)

10 FOR X=10000+Z TO 99999 STEP

10

20 A=I*100000+X: B=10*X+I

30 IF A*3=B THEN PRINT A

40 NEXT X, I

解题程序 5b:

[B2-5B]

1 W(1)=857

2 W(2)=714

3 W(3)=571

5 FOR I=1 TO 3

6 Z=W(I)

10 FOR X=10000+Z TO 99999 STEP

1000

20 A=I*100000+X: B=10*X+I

30 IF A*3=B THEN PRINT A

40 NEXT X, I

请读者分析,在以上程序中,采用了什么技术,从而提高了程序的运算速度。

深化:用猜测求解问题,其优点是编程简捷,思路简要。只要掌握好搜索对象与步距和搜索结果的判据两个环节,程序是比较容易编写的。它的缺点是如果搜索对象及其步距选择得不好,其求解的运行时间就会很长。介绍采用查表技术来解本题。所谓查表技术,就是事先编好一个表,让计算机来查这个表,对号入座查出结果,方法十分简要。

分析题意,可以知道某数乘 3,只有九种情况:

即:

 $1 \times 3 = 3$ 进位为 0,

 $2 \times 3 = 6$ 进位为 0,

 $3 \times 3 = 9$ 进位为 0,

 $4 \times 3 = 12$ 进位为 1,

 $5 \times 3 = 15$ 进位为 1,

 $6 \times 3 = 18$ 进位为 1,

 $7 \times 3 = 21$ 进位为 2,

 $8 \times 3 = 24$ 进位为 2,

 $9 \times 3 = 27$ 进位为 2.

在本题中:

$$\begin{array}{r} \text{M a b c d e} \\ \times \quad \quad \quad 3 \\ \hline \text{a b c d e M} \end{array}$$

由式中可知:

 $e = (d \times 3) \text{ 的个位数} + (e \times 3) \text{ 的进位数};$
 $d = (c \times 3) \text{ 的个位数} + (d \times 3) \text{ 的进位数};$
 $c = (b \times 3) \text{ 的个位数} + (c \times 3) \text{ 的进位数};$
 $(d \times 3) \text{ 的个位数} = e - (e \times 3) \text{ 的进位数};$
 $(c \times 3) \text{ 的个位数} = d - (d \times 3) \text{ 的进位数};$
 $(b \times 3) \text{ 的个位数} = c - (c \times 3) \text{ 的进位数};$

要求左侧的数是个位数，而右侧的数是已知的数，所以可以以乘积的个位数为变量做出一组函数，如表 2-1。

表 2-1 以乘积的个位数为变量做出一组函数

乘积的个位数	被 3 乘的被乘数	乘上被乘数产生的进位数	乘积的个位数	被 3 乘的被乘数	乘上被乘数产生的进位数
1	7 (7×3=21)	2	6	2 (2×3=6)	0
2	4 (4×3=12)	1	7	9 (9×3=27)	2
3	1 (1×3=3)	0	8	6 (6×3=18)	1
4	8 (8×3=24)	2	9	3 (3×3=9)	0
5	5 (5×3=15)	1			

这样可以用数组来表现它们的关系如下：

A(1)=7, B(1)=2;

A(6)=2, B(6)=0;

A(2)=4, B(2)=1;

A(7)=9, B(7)=2;

A(3)=1, B(3)=0;

A(8)=6, B(8)=1;

A(4)=8, B(4)=2;

A(9)=3, B(9)=0;

A(5)=5, B(5)=1;

例如 abcd7×3, a、b、c、d 应当为几？由于

$$\begin{array}{r} a \ b \ c \ d \ 7 \\ \times \quad \quad \quad 3 \\ \hline \quad \quad \quad 71 \end{array}$$

7×3 是 21, 查表 2-1 知 A(1)=7, B(1)=2, A(1)-B(1)=5; A(5) 知 d 应为 5, A(5)-B(5)=4, A(4) 知 c 应为 8, A(4)-B(4)=6, A(6) 知 b 应为 2; A(6)-B(6)=2; A(2) 知 a 应为 4, 由此可以编出一程序求出题解。因为它是直接查表求解，所以其运行速度是极快的。读者可以按所示程序输入计算机内，对它运行，将会看到求解是在瞬间获得。这样的编程求解显然比以前的解法优秀得多，编程技巧也奥妙多了。

解题程序 6:

[B2-6]

10 FOR I=1 TO 9

100 P=A(P)-B(P)

110 NEXT I

20 READ A(I), B(I)

120 IF W(6)<>J THEN 180

30 NEXT I

130 PRINT J;";";

40 DATA 7, 2, 4, 1, 1, 0, 8, 2, 5

140 FOR I=6 TO 1 STEP-1

50 DATA 1, 2, 0, 9, 2, 6, 1, 3, 0

150 PRINT W(I);

60 FOR J=1 TO 3

160 NEXT I

70 P=J

170 PRINT: PRINT

80 FOR I=1 TO 6

180 NEXT J

90 W(I)=A(P)

200 END

[Q2-6. BAS]

FOR i=1 TO 9

READ a(i),b(i)

NEXT i

DATA 7,2,4,1,1,0,8,2,5

DATA 1,2,0,9,2,6,1,3,0

FOR j=1 TO 3

P=J

FOR i=1 TO 6

W(i)=a(P)

P=a(P)-b(P)

NEXT i

IF w(6)<>j THEN 180

PRINT j;"":

FOR i=6 TO 1 STEP-1

PRINT w(i);

NEXT i

PRINT;PRINT

180 NEXT j

END

推而广之:请读者编一程序求解。

题三:设有一 12 位的数 Mabcde fghijk 乘 3 后变为 abcde fghijkM,试求出这个可能的 12 位数,并把结果打印出来。

第三题

水手分桃

题一：遇难船上的五名水手，带着一个猴子漂流到一个荒岛上。他们采集一大批野桃作为后几天的口粮。为防意外，夜间轮流看守。第一个水手值班时，独自把桃分成五份，还余一个扔给了猴子，自己藏了一份。轮到第二个水手值班时，他又把剩下的桃子分成了五份，正好也剩一个，扔给了猴子，自己也拿了一份。当轮到第三、四、五个水手值班时，都如此办理。第二天早上五个水手又一齐把剩下的桃子分成五份，仍剩一个桃子扔给猴子。请编程计算一下这堆桃子共有多少个？（题分：30分）[选自1995年新乡市小学计算机竞赛上机试题]

解：求解本题，最好采用猜测法进行猜测，关键是要确定一个判据，以确定所猜之数已经被找到。这一堆桃子，减去一个后能分成五份，也就是说它能被5整除或原数除5余1。所以其判据 $[(X-1)/5]$ 应为整数或者 $X \bmod 5 = 1$ 。

每次将该数乘4应为下一次分桃的出发点，如此循环6次，如果次次均符合上述要求，则所求的数就找到了。

解题程序 1：

```
[B3-1]
10 Y=1                                GOTO 20
20 X=Y                                60 NEXT I
30 FOR I=1 TO 6                       70 PRINT Y
40 X=(X-1)/5*4                        80 END
50 IF X<>INT(X) THEN Y=Y+5;
```

评论：解题程序1是基本形式，其他的题解不过是对它的完善而已。把10号语句改为 $Y=11$, $Y=101$, 或 $Y=10001$, 都是可以的，但也都是没有太多的道理的。问题在于要确定出由何数起始去猜，起始猜数定的大一些，当然可以加快运算速度，但在当今计算机的高速运算下这一点差异是不算一回事的。如果说道理，就是编程者在编程和调试时是逐步把 $Y=1$ 修改到 $Y=10001$ 去的。但是必须注意的是其尾数必须是一个1字。如果尾数不为1，每经过增加5步距之后，其尾数就不是一或六，这样就永远得不到 $(X-1)/5$ 为整数的结果，计算机进行推算也就永远不会找到结果，它将变成了一个不断循环而又找不到求解的运行程序了。40号语句和50号语句做如下修改也是可以的：

```
40 IF X MOD 5 <> 1 THEN Y=Y+5 : GOTO 20
50 X=(X-1)/5*4
```

这只是判据的形式不同，其道理则和前述是一样的。值得进一步讨论的是，80号语句把问题结束了，得出了唯一解，这是误解。其实它后续还有许多解，最好把80号语句改为：

```
80 IF Y > 99999 THEN END
```

```
90 Y=Y+5 : GOTO 20
```

```
100 END
```

按照编程技巧要求，以多采用模块化语句，少采用 GOTO 语句为原则，本题改为如下形式更为恰当。

解题程序 2:

```
[B3-2]
```

```
10 FOR J=1 TO 99999 STEP 5
```

```
60 NEXT I
```

```
20 X=J
```

```
70 PRINT J
```

```
30 FOR I=1 TO 6
```

```
80 NEXT J
```

```
40 IF X MOD 5 < > 1 THEN 80
```

```
100 END
```

```
50 X= (X-1) /5 * 4
```

```
[Q3-2. BAS]
```

```
FOR j=1 TO 99999 STEP 5
```

```
x=j
```

```
FOR i=1 TO 6
```

```
x=(x-1) /5 * 4
```

```
NEXT i
```

```
PRINT j
```

```
80 NEXT j
```

```
END
```

```
[C3-2. C]
```

```
main ()
```

```
{
```

```
long int j, x;
```

```
int i;
```

```
for (j=1; j<=9999; j=j+5)
```

```
{
```

```
x=j
```

```
for (i=1; i<=6; i++)
```

```
{
```

```
if (x%5 !=1) break;
```

```
x= (x-1)/5 * 4;
```

```
}
```

```
}
```

```
printf("j=%d",j);
```

```
}
```

解题程序 3: