

IBM PC 译丛

数据和文件管理

朱科志 译

辽宁省电子计算机学会
《小型微型计算机系统》编辑部

前 言

IBM—PC, XT & AT及其兼容机(长城0520等),是我国微型计算机普及率最高的机种,为促其开发和应用,辽宁省电子计算机学会曾于八四年出版过一套《IBM—PC译丛》(共十九本,350万字,俗称黄皮书),深受用户欢迎。但由于印数有限,未能满足用户需要。为此,辽宁省电子计算机学会和《小型微型计算机系统》编辑部决定重新排印这套丛书。

再版过程中,我们对原书目录进行重新编选,淘汰一些对用户关系不甚密切的部分编译程序、专用软件,以及内容雷同的硬件技术手册,增加了广大用户迫切需要的dBASE III、dBASE IV、IBM—PC 通讯与网络等重要内容,对保留的内容,译者结合自己的实践经验,进行了重新的校订,这样便使这套丛书更为实用。

这次重新排印,时间很紧,尽管做了不少努力,更正一些错误,还可能会出现新的疏漏,敬请读者批评指正。

辽宁省电子计算机学会

《小型微型计算机系统》编辑部

目 录

- | | |
|---------------------------|--------------------------|
| 1. IBM PC/AT操作指南 | 2. IBM PC/XT硬件技术手册 |
| 3. IBM PC磁盘操作系统 | 4. IBM PC CP/M—86 |
| 5. IBM PC数据和文件管理 | 5. IBM PC宏汇编 |
| 7. IBM PC汇编语言程序 | 6. IBM PC BASIC |
| 9. IBM PC FORTRAN
编译程序 | 10. IBM PC PASCAL |
| 11. IBM PC COBOL编译程序 | 12. IBM PC FORTH语言 |
| 13. IBM PC C语言 | 14. IBM PC dBASE III用户手册 |
| 15. IBM PC dBASE IV用户手册 | 16. IBM PC通讯与网络 |

目 录

绪 言

第一章 指 针

下标	3
尼姆游戏, G1A和启发式	
程序设计	3
蒙特卡洛方法	7
随机的正文生成	11
随机的消息选择	14
值的正态分布	18

第二章 排 序

排序算法的分类	21
强力排序	21
交换排序	23
二分排序	26
树型排序	29
多键排序	31
小结	33
参考文献	36

第三章 字 符 串

字处理	37
随机词的选择	37
模式匹配	40
正文的加密	42
正文的重排	45
正文的分析	47

第四章 线性列表和链接列表

堆栈	51
队列	52
双端队列	54
链接列表	55
单向链接列表	55

双向链接列表	56
环形链接列表	56
环形双向链接列表的应用	57

第五章 顺序存取文件

顺序查找技术	62
顺序文件的存取	63
分组求和	66
顺序文件的合并	69
索引的产生	72

第六章 直接存取文件

文件的查找	81
二分查找	81
内插查找	84
散列编址技术	85
大文件的排序	87
磁盘排序	87
分离键排序	89
分段分离键排序	91
ISAM文件处理技术	93
ISAM 的三个存贮区	93
DOS的物理特性	93
ISAM 的结构	94
ISAM 的存取	95
溢出区	95
ISAM 的插入	95

第七章 树形结构

二叉树	97
二叉顺序查找树	102
内存中的单键BSST	102
内存中的双键BSST	104
内存中的多键BSST	107

磁盘上的BSST	109	主程序——菜单驱动程序.....	120
树与环形链接列表.....	111	删除和均衡.....	121
第八章 倒排文件		记录的插入.....	121
次要键.....	118	随机存取.....	124
记录的结构.....	118	按排序次序进行显示.....	127
记录的内容.....	119	物理记录的显示.....	134
使用指针表存取文件.....	120	结束语.....	137

绪 言

我们经常听到这样的问题：

“什么是数据结构？”

“理解了查找技术，为什么有助于编写较好的程序呢？”

“在现实世界上什么是堆栈、队列、双端队列和树形结构？”

“一个文件经过倒排是否会损失其内容？”

程序员之所以提出这样的问题是有原因的。这些问题起源于对某些用来处理数据的基本程序设计概念缺乏理解。大多数程序员的知识不外乎一批厂家提供的手册和一、两门正规的BASIC课程。他们充满信心地认为他们有能力应付列表、数组、下标和顺序查找技术。遗憾的是，这种程序设计水平，这些技术，尚不足以帮助他们编写出漂亮有效的软件。

在学校里和咨询业务中，我们经常明确地指明编写“可用的”程序（即有利于用户的程序）的极端重要性。我们还强调程序员本人是一个好程序的唯一用户的情况是极为罕见的，因为根据定义，好程序是可供多人使用的程序。好的程序员为了写出容易使用的程序也必须煞费苦心。一个好程序具备以下性质：

- 它应该是结构良好的，使得它的作者和它的所有读者都感到自己有能力改变它的一部分（增加、删除和修改它的一些模块），而不致于给未涉及的部分带来不利的影响。

- 它应当配有完整清晰的解释，使得它的逻辑容易理解。

- 只要交互式更有利于用户，它就应该采取交互式操作。

- 如果有合适的介质存贮信息，它就应该使用文件。

- 为提高计算机的使用效率，它应该采用适当的算法缩短排序和查找的时间，减少磁盘存取次数，避免使用特大的内存数组和不恰当的数据存取技术。

其中的前三个性质在《IBM PC的BASIC技术》之类的书中已作过详尽地讨论。后两个性质是本书的重点。

许多大学都有一门叫作“数据结构”的课程，它是给学过两个学期的程序设计的学

生开设的。它的目的是进一步提高学生的程序设计技术，并介绍一些多年来为使程序运行得更有效而开发的常用过程。这门课程的内容同本书几乎相同，而且叙述的次序也相差无几。同一个学生在掌握了这些技术之后处理内存和文件中大量数据的能力就有了显著的提高，处理起来更轻松，更成功。数据结构课程比起正规训练中的其他课程说来，在培养学生具备适应将来工作需要的、编写面向用户的程序的能力方面，具有更重要的意义。

IBM个人计算机的普及方兴未艾，因此工业界权威人士预言它将成为八十年代中期最受欢迎的三、四种个人计算机之一。大多数人可能只购买小型的IBM PC系统，尽管他们最终的配置可能包括很多外部设备。本书给出的程序所用的系统就是这样一个小型系统，具有64K用户可用内存、两台320K的双密度磁盘驱动器和一台黑白显示器。此

外我们还使用了一台 Brother 公司的 HR—1 型字符打印机来产生全部的程序清单。我们使用了版本为1.10的IBM磁盘操作系统 (DOS) 及其配备的高级BASIC语言。我们感到利用这套配置开发我们的应用不会受到任何拘束。

我们在这本书里为了简化和阐明数据结构这个主题着实下了一番苦功。我们把本书叫作“数据和文件管理”，是因为我们力图指明：数据结构这个题目只有同数据管理的实践方面联系起来才是有用的。本书作为例子给出的程序将告诉读者这些技术在许多常见的应用中该是多么有用。我们衷心地希望其中的某些例子能够引起你足够的兴趣，使你能够象改编小说那样修改这些程序。我们把编写这些程序当作一大乐事，我们希望你在使用它们时也会享受到其中的快乐。

第一章 指 针

BASIC 之所以成为微型计算机上最常用的问题求解语言，理由举不胜举，最重要的是它的内在简单性。它容易学习，使用方便。许多中学和大学都把它当作计算机的入门课程。

BASIC 得以普及还有一个重要的原因，往往为人所忽视，这就是它的高度灵活性。换句话说，BASIC 给出的程序设计结构是用其它语言难以实现的，甚至是不可能的。例如，BASIC 对数组下标唯一的要求是只能取数值表达式，但在 FORTRAN 的许多版本中，下标形式极为有限，只有几种简单的变形。

下标

“下标” (subscript) 是程序员使用的指向数组的指针，因此它必须具有尽可能多的变形形式。

最基本的下标形式只能存取一个给定的数组元素。请看下面的程序片断：

```
100 DIM D ( 50 )
110 P = 17
120 V = 4
130 D ( P ) = V
```

在130行上变量P被用作下标，这时它扮演了指向数组D的指针的角色。这个片断的总效果是把值4存放到数组D的第17个位置上。

假如，这个程序片断又增加了一行：

```
140 D ( D ( P ) ) = 237.8
```

这次数组D的指针是变量D (P)，后者也使用了一个指针，它是 P = 17，这是上面的110行定义的。据120行和130行可知，D (P) = D (17) = 4，从而 D (D (P)) = D (4) 因此D (4) 被定义作237.8，即值237.8被存放到D (4) 中。这种寻址方式叫作“间接寻址” (indirect addressing)，这是因为计算机使用了一个中间位置(D(17))来确定值传送的最终目的地。

在介绍指针的同时，需要指出FORTRAN 的大多数版本不准许下标变量再用下标变量作为下标。切记！

本章为阐明指针的用法给出了一些程序例子，它们提出了一个共同性的问题——数组指针来自何处？你将会发现数组指针可以选自一个存贮池 (pool)，可以随机生成，也可以计算得到。它们有别于更常见的数组指针来源，诸如FOR—NEXT循环索引、计数器等。

尼姆游戏，G1A和启发式程序设计

第一个程序G1A，是尼姆游戏(Nim)的有趣变形。它的玩法是：有一堆棋子，最初是13枚；两个游戏者依次从堆中取走1到3枚，谁若拿到最后一枚谁就算输。如果第二个游戏

者能牢记这样一条原则:给对手留下的棋子总数总要等于9、5或1,那末他就会百战百胜。

在计算机的发展初期,人们经常讨论如何给计算机编制能给它们增加类似智能的性质的程序。于是诞生了一门叫作“人工智能”(Artificial Intelligence, AI)的新学科。一种用于智能模拟的技术叫作“启发式程序设计”(heuristic programming),这是一种旨在发现或揭示某些基本原理的程序设计技术。

1965年布拉克(H. D. Block)在《科学美国人(American Scientist)》杂志上发表了一篇文章,描述了一种能够“学习”如何玩赢尼姆游戏的机器。这种机器最初叫作G1,它能巧妙地仿效人类的学习方法。最初它好象心不在焉地玩。玩过一阵子之后,一盘没完它就知道认输了,似乎它已经意识到继续玩下去也无法挽回败局。经过多次磨炼,终于它成了常胜将军,这时它已经掌握了致胜的诀窍。

为了缩短学习过程,布拉克在G1的基础上创造了G1A。现在我们把这台机器编成程序奉送给你。这个程序亲自在四个杯子中放上了全部可能采取的动作。请记住,每个杯子中放着分别写有1、2和3的三个纸条。随着游戏的进展,G1A从相应的杯子中随机地取出一个动作。当有迹象表明G1A已经输定了时,它就在输前所取的最后一个动作上标上-1,从此以后再不采取这个动作了。最后,除那些能致胜的动作外,所有杯子中的动作都标上了-1。以后G1A只采取那些没标-1的动作。

作为习题,请读者按照这个程序的执行路线,看看G1A是怎样聪明起来的。这一定是十分有趣的。

这个程序有些地方还是值得特殊说明的:

- 用户的全部输入都是用INKEY\$函数处理的。请注意程序中向用户索取姓名缩写的部分,这里利用INKEY\$建立了一个包含3个字符的、不用ENTER键结束的输入字符串。

- LOCATE指令广泛地用于显示战局发生变化时的游戏状态。

- 用图形符号(扑克中的红心)显示棋子。注意,为了从堆中随机地挪走棋子,还是要花点功夫的。

- G1A显示的消息颇有性格。G1A输了,它用小写回答,显得那末卑谦。它赢了,就用大写,好一副自命不凡的神情。

- G1A分别用图形符号——扑克中的黑桃和方块表示轮到自己 and 对手取子了。

```
10 ' 文件名: "g1a"
20 ' 目的: 玩尼姆游戏
30 ' 作者: jdr & jpg 9/82
40  DEFINT A-Z
50  DIM CUP(4, 3): CY(13), CX(13)
60  DIM PERM(6, 3) '用来装1、2、3的6种排列
70  ' 用来存放自己和对手的胜利消息
80  DIM H1$(6), H2$(6), O1$(13), O2$(13), O3$(13)
90  RANDOMIZE: CLS
100 ' 标题屏幕
110 LOCATE 5, 1: PRINT STRING$(80, 1) 'ASCII字符001是个微笑的面孔
120 LOCATE 7, 25: PRINT "N I M W I T H G 1 A"
130 LOCATE 9, 1: PRINT STRING$(80, 1)
140 ' 将纸条1、2、3装入4个杯子
150 FOR I=1 TO 4
160     FOR J=1 TO 3: CUP(I, J)=J: NEXT J
170 NEXT I
```



```

180 ' 用1、2、3的全排列填写数组PERM
190 FOR I=1 TO 6
200     FOR J=1 TO 3: READ PERM(I, J): NEXT J
210 NEXT I
220 ' 用漂亮话填写自己的胜利消息
230 FOR I=1 TO 6: READ H1$(I), H2$(I): NEXT I
240 ' 用挖苦话填写对手的胜利消息
250 FOR I=1 TO 13: READ O1$(I), O2$(I), O3$(I): NEXT I
260 LOCATE 12, 1: PRINT STRING$(54, 32), WHO$ = " / "
270 ' ** 要对手通名报姓
280 LOCATE 12, 3: PRINT "enter your initials": T$=INKEY$
290 IF T$="" THEN 270
300 WHO$=WHO$+T$: LOCATE 12, 25: PRINT WHO$
310 IF LEN(WHO$) <> 6 THEN 270
320 GOSUB 5000: WHO$=RIGHT$(WHO$, 3): CLS
330 ' ** 设定屏幕显示
340 LOCATE 7, 10: PRINT "G1A": ' 计算机的名字是 "G1A"
350 LOCATE 9, 10: PRINT WHO$: ' 印对手的名字
360 LOCATE 8, 10: PRINT WIN: ' 印自己赢的次数
370 LOCATE 10, 10: PRINT LOSE: ' 印对手赢的次数
380 GOSUB 1000 ' 收拾残局
390 ' ** 重摆棋子
400 LOCATE 7, 6: PRINT " ": ' 清除自己的取子标记
410 LOCATE 9, 6: PRINT STRING$(3, 4): ' 设定对手的取子标记(方块)
420 ' ** 读取对手的取子枚数
430 LOCATE 15, 10: PRINT "Take 1 to 3 chips": T$=INKEY$
440 IF T$="" THEN 420
    ELSE TAKE=VAL(T$): LOCATE 15, 30: PRINT " / ": T$:
    GOSUB 5000
450 IF TAKE<1 OR TAKE>3 OR TAKE>CHIPS
    THEN GOSUB 6000: LOCATE 15, 10: PRINT "improper move": GOSUB 5000:
    GOTO 420
460 GOSUB 2000
470 IF CHIPS<=0 THEN GOSUB 4000: GOTO 330
480 ' 该G1A取子了
490 GOSUB 6000: LOCATE 9, 6: PRINT " ":
500 LOCATE 7, 6: PRINT STRING$(3, 6):
510 ' A是个随机排列选择器
520 GOSUB 5000: A=INT(RND*6+1)
530 FOR I=1 TO 3
540 '     检查第A个杯子
550 ' B是CHIPS除以4所得的余数
560 K=PERM(A, I): B=CHIPS-4*INT(CHIPS/4)
570 IF B=0 THEN B=4
580 ' 如此杯非空, 取一着试之
590 ' 如此杯为零, 认输, 标记纸条, 重开棋
600 IF CUP(B, K) <=0 THEN NEXT I: CUP(BCUP, KCARD)=-1: HOW=1:
610 TAKE=CUP(B, K)
620 ' 如取到最后一下子, 认输, 标记纸条, 重开棋.
630 IF TAKE>=CHIPS THEN CUP(B, K)=-1: HOW=2: GOSUB 3000: GOTO 330
640 ' 注意名词单、复数之别, 'chip'和'chips'
650 LOCATE 15, 10: PRINT "G1A takes": TAKE: LEFT$("chips", 4+INT(TAKE/2)).
660 GOSUB 5000: GOSUB 5800
670 BCUP=B: KCARD=K
680 GOSUB 2000: GOTO 390
1000' **** 子例程: 摆好13个棋子
1010 N=0
1020 FOR I=-1 TO 1

```

```

1030 L=ABS(I): M=62+L
1040 FOR J=1 TO 5-L
1050 N=N+1: CX(N)=M+J+J: CY(N)=I+I+8
1060 LOCATE CY(N), CX(N): PRINT CHR$(3);
1070 NEXT J
1080 NEXT I
1090 CHIPS=13: LOCATE 7, 51: PRINT 'chips': LOCATE 8, 51: PRINT 'left';
1100 LOCATE 9, 51: PRINT CHIPS: R=INT(RND*13+1): P=1+INT(RND*11+1)
1110 RETURN
2000' **** 子例程: 挪走1至3个棋子
2010 FOR I=1 TO TAKE
2020 R=R+P
2030 IF R>13 THEN R=R-13
2040 LOCATE CY(R), CX(R): PRINT ' ';
2050 NEXT I
2060 CHIPS=CHIPS-TAKE: LOCATE 9, 51: PRINT CHIPS:
2070 RETURN
3000' **** 子例程: 印自己失败的消息
3010 GOSUB 5000: GOSUB 5000: CLS
3020 A=INT(RND*6+1): LOCATE 7, 1: PRINT STRING$(80, 254)
3030 IF HOW=2 THEN LOCATE 9, 25: PRINT 'G1A': H2$(A): 'acknowledges
    defeat'
3040 IF HOW=1 THEN LOCATE 9, 25: PRINT 'G1A': H1$(A): "concedes the game"
3050 LOCATE 12, 1: PRINT STRING$(80, 254): LOSE+1
3060 GOSUB 5000: GOSUB 5000: GOSUB 5000: CLS
3070 RETURN
4000' **** 子例程: 印自己胜利的消息
4010 CLS: A=INT(RND*13+1): B=INT(RND*13+1): C=INT(RND*13+1)
4020 LOCATE 7, 1: PRINT STRING$(80, 254)
4030 LOCATE 9, 15: PRINT 'THE': O1$(A): 'G1A HAS':
4040 PRINT O2$(B): 'THE': O3$(C): ' ': WHO$
4050 LOCATE 11, 1: PRINT STRING$(80, 254)
4060 WIN=WIN+1
4070 GOSUB 5000: GOSUB 5000: GOSUB 5000: CLS
4080 RETURN
5000' ***** 子例程: 延长一段时间:
5010 FOR I=1 TO 1500: NEXT I
5020 RETURN
6000' ****子例程: 清消息
6010 LOCATE 15, 1: PRINT STRING$(80, 32);
6020 RETURN
7000' ****数据语句
7010'
7020'
7030 DATA 1, 2, 3, 1, 3, 2, 2, 1, 3, 2, 3, 1, 3, 1, 2, 3, 2, 1
7040' G1A的失败消息
7050 DATA cordially, respectfully, graciously politely
7060 DATA affably, humbly, congenially, modestly,
7070 DATA meekly, amicably, courteously, agreeably
7080' G1A的胜利消息
7090 DATA AWESOME, ANNIHILATED, PROSAIC, DREADED
7100 DATA EXTERMINATED, VAPID, PUISSANT, OBLITERATED, SLUGGISH
7110 DATA EMINENT, DEMOLISHED, DOLTISH, EXALTED, CQNOQUERED, OBTUSE
7120 DATA INTREPID, VANQUISHED, INFERIOR, SPLENDID DEVASTATED
7130 DATA INSIPID, SAPIENT, EXTIRPATED, MAWKSH, IERUDITE, SUBJUGATED
7140 DATA BUNGLING, FORMIDABLE, CRUSHED, FLACCID, REDOUBTABLE
7150 DATA FLATTENED, INEPT, BRILLIANT, STOMPED, IGNORANT
7160 DATA MAGNIFICENT, DESTROYED, STUPID

```

GOSUB 3000; GOTO 330

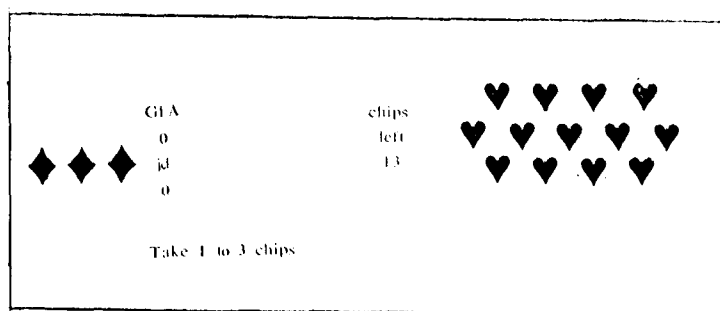


图1

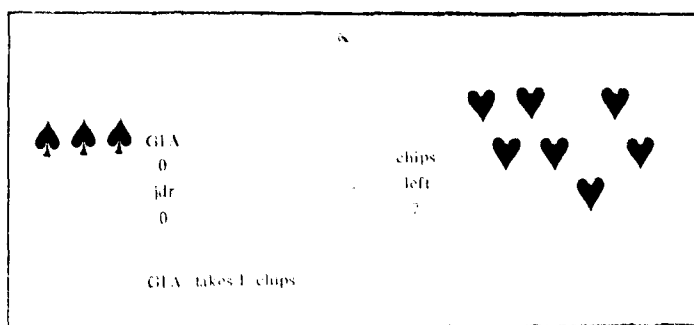


图2

蒙特卡洛方法

下一个程序叫作 **JOBSTEPS**，用来说明如何使用随机数作为指针，以便给数组分配内容。该程序的功能是要确定两个机械加工序列，这两个序列将分别让两个工人来完成，要求作到公平合理，就是说他们各自工作的总时间要尽可能地一样多。他们能够承担的加工工种有25种之多，每个序列包含的加工工种数目不受限制，只要两人工作的总时间相同即可。他们的老板是此程序的使用者，他先输入总的工作时数，比如说是4个小时，而后该程序印出两张工作计划表，各包含一串加工操作。注意，不允许在两张表上出现同一种操作。

一次接一次地搅乱一个数组的内容，而后再来考查看这种次序产生的解法是否优越于前一种，这种技术是“蒙特卡洛技术”（Monte Carlo Technigue），蒙特卡洛本来是摩纳哥的一个著名赌场的名字。

这个问题并非微不足道，用铅笔和纸来设计这样的工作计划表差不多得用一个小时。程序 **JOBSTEPS** 所做的事情是从25种操作中随机地选取一组操作，再求出它们所耗时间的总和。只要算出的和数一达到老板指定的总时数（4个小时），就显示这个总和。这时老板可以让计算机程序另选一组操作，使其时间总和更接近4个小时，他也可

以表示接受这组操作。

在第一个工人的工作计划表就这样地确定下来之后，程序就把这组操作的时间总和当作追逐的目标，再从没有分配的操作中随机地为第二个工人选出一张工作计划表。计算机显示它的第一次尝试，这时老板可以表示接受或拒绝。拒绝会强迫计算机再提出一份更好的计划表。每一次更接近目标（第一个工人的工作计划表的总时间）的尝试都要显示出来，让老板表态。最终老板总会接受的，这时显示两个工人的工作计划表和所耗费的总时间。

```
10 ' 文件名: "jobsteps"
20 ' 目的: 用蒙特卡洛方法制定工作计划
30 ' 作者: jpg & jdr 9/82
40 '
50 DIM T$(25), T(25), K(25)
60 ' T$=操作 T=各种操作的时间(分)
70 ' K=选择指针
80 RANDOMIZE: CLS
90 ' 读操作及时间
100 INPUT "What is target time (in minutes) ": TT
110 FOR I=1 TO 25
120 READ T$(I), T(I): K(I)=0
130 NEXT I
140 S=0
150 T1=10 ' 设定目标时间S和建议时间TT之差的上界
160 GOSUB 1000
170 PRINT "Suggested time is": S
180 LINE INPUT "Is this acceptable? (y=yes) ": A$
190 IF A$ <> "y" THEN T1=ABS(S-TT): GOTO 160
200 FOR I=1 TO 25
210 IF K(I) > 0 THEN K(I)=-K(I) ' 标明为第一个工人选择的操作
220 NEXT I
230 GOSUB 1000
240 PRINT "Suggested time is": S
250 LINE INPUT "Is this acceptable? (y=yes) ": A$
260 IF A$ <> "y" THEN T1=ABS(S-TT): GOTO 230
270 PRINT "Schedule for both workers"
280 S1=0: S2=0
285 FOR I=1 TO 25
290 IF K(I) < 0 THEN P=ABS(K(I)): PRINT T$(P), T(P): S1=S1+T(P)
300 NEXT I
310 PRINT "Sum, Worker 1: ": S1
330 FOR I=1 TO 25
340 IF K(I) > 0 THEN PRINT T$(K(I)), T(K(I)): S2=S2+T(K(I))
350 NEXT I
360 PRINT "Sum, Worker 2: ": S2
370 STOP
500 DATA "stamping", 31.7, "spooling", 42.0, "flanging", 25.4
510 DATA "milling", 40.1, "cutting", 32.5, "degreasing", 24.7
520 DATA "pithing", 34.8, "polarizing", 30.3, "rolling", 31.7
530 DATA "cascading", 22.2, "wafting", 44.8, "leveling", 15.0
540 DATA "plating", 29.1, "chafing", 38.2, "fluting", 38.5
550 DATA "sanding", 53.9, "bending", 26.5, "stressing", 27.7
560 DATA "testing", 51.4, "polishing", 20.1, "packing", 44.2
570 DATA "blunting", 32.2, "merging", 37.8, "gnashing", 23.4
580 DATA "flushing", 25.0
1000' **** 子例程: 返回与TT之差不超过T1的和数S
1010 S=0
```

```

1020 FOR I=1 TO 25
1030   IF K(I) > 0 THEN K(I) = 0
1040 NEXT I
1050 R=INT(RND*25+1) '产生一个随机数
1060' 检查该操作是否已分配
1070   IF K(R) < > 0 THEN 1050 ' 如是, 另取一操作
1080   S=S+T(R): K(R)=R ' 累加此随机时间
1090   IF S>TT THEN IF S-TT<T1 THEN RETURN ELSE 1010
        ELSE IF TT-S<T1 THEN RETURN ELSE 1050
9999 END

```

What is target time (in minutes)? 120 目标时间是多少(分)? 120

Suggested time is 119.8 建议时间是119.8

Is this acceptable? (y=yes) y 可以接受吗? 可

Suggested time is 128.2 建议时间是 128.2

Is this acceptable? (y=yes) n 可以接受吗? 不可

Suggested time is 119.9

Is this acceptable? (y=yes) y

Schedule for both workers 二个工人的工作计划表

chafing 38.2

sanding 53.9

stressing 27.7

Sum, Worker 1: 119.8

第4个工人的总工时 119.8

plating 29.1

bending 26.5

polishing 20.1

packing 44.2

Sum, Worker 2: 119.9

第2个工人的总工时: 119.9

下面给出一个完全类似的程序 INHERIT, 它可将一份遗产公平地分配给两个继承人。在一本介绍 BASIC 的书中有个程序, 能够将一批歌曲按时间平分到一张唱片的两面上。这里的 INHERIT 就是仿照那个程序编写的。

在这个程序的 DATA 语句中指出了死者遗产包括25件不同的财物, 并列出了它们的估计价值。该程序将搅乱指向各件财物的指针, 遗产监督人认为需要搅乱多少次就搅乱多少次。而后程序开列出两份价值相等的遗产份额。

注意, 410行在打印机上印出了字符±, 而没有印IBM 标示字符编辑字段的首、尾位置所用的反斜线(\)字符。

```

10 ' 文件名: "inherit"
20 ' 目的: 用蒙特卡洛方法平分遗产
30 ' 作者: jpg & jdr 9/82
40 '
50 DIM T$(25), V(25), K(25), L(25)
60 ' T$=财物描述 V=价值
70 ' K=随机指针 L=第二随机指针
80 RANDOMIZE: CLS
90 ' 读财物描述和价值
100 ' N=分配给两个继承人的财物总件数
110 INPUT "How many of the 25 items to be distributed"; N
120 LPRINT "This distribution has"; N; "of the 25 items."
130 FOR I=1 TO 25
140   READ T$(I), V(I): K(I)=I
150 NEXT I
160 F$ = "Smallest difference is $$$###, ### in try ###"
170 C=600001: ' 设定两个继承人分得财物的价值差额上限
180 INPUT "How many scrambles"; N5
190 LPRINT "Selected number of scrambles="; N5

```

```

200 FOR Q=1 TO N5 '将所有财物的次序打乱N5次
210   FOR I=1 TO N
220     J=INT(RND*N+1): Z=K(I): K(I)=K(J): K(J)=Z
230   NEXT I
240 ' 求出每个继承人分得财物的总价值
250   Z1=0: Z2=0: N2=INT(N/2)
260   FOR I=1 TO N2
270     J=K(I): Z1=Z1+V(J): J=K(I+N2): Z2=Z2+V(J)
280   NEXT I
290   IF N/2 < > INT(N/2) THEN Z2=Z2+V(N)
300   B=ABS(Z1-Z2) 'B=两人分得财物总价值之差
310   IF B >= C THEN 370 'C=先前的最小差额
320   C=B: C1=Z1: C2=Z2
330   FOR I=1 TO N: L(I)=K(I): NEXT I
340   PRINT USING F$: C, Q
350   LPRINT USING F$: C, Q
360   IF C=0 THEN 380 '妙极了! 差额是0!
370 NEXT Q '再打乱一次
380 PRINT: PRINT
390 LPRINT: LPRINT
400 F1$ = " Estate #: "
410 F2$ = " ± ± $ $ ###, ###.##"
420 F3$ = " ====="
430 F4$ = " Total: $ $ ###, ###.##"
440 RPINT USING F1$: 1: LPRINT USING F1$: 1
450 FOR I=1 TO N2: J=L(I)
460   PRINT USING F2$: T$(J), V(J): LPRINT USING F2$: T$(J), V(J)
470 NEXT I
480 PRINT F3$: LPRINT F3$
490 PRINT USING F4$: C1: LPRINT USING F4$: C1
500 PRINT: PRINT: LPRINT: LPRINT
510 PRINT USING F1$: 2: LPRINT USING F1$: 2
520 FOR I=N2+1 TO N: J=L(I)
530   PRINT USING F2$: T$(J), V(J): LPRINT USING F2$: T$(J), V(J)
540 NEXT I
550 PRINT F3$: LPRINT F3$
560 PRINT USING F4$: C2: LPRINT USING F4$: C2
570 STOP
580 DATA "Stamp collection", 5200, "China hutch", 4300
590 DATA "Sterling silverware", 2450, "Crystal chandelier", 2400
600 DATA "Coin collection", 6000, "China", 3200
610 DATA "Packard coupe", 6200, "Diamond solitaire", 3550
620 DATA "Sail boat", 2200, "Gold statuette", 8200
630 DATA "Overstuffed chairs", 2100, "Paarl necklace", 3300
640 DATA "Bedroom suite", 5300, "Diamond pendant", 3300
650 DATA "Grand piano", 3400, "Oboe", 1200, "Spinnet", 2500
660 DATA "Movie camera", 2400, "Station wagon", 4250
670 DATA "Convertible", 3800, "Dining room suite", 3850
680 DATA "Living room suite", 3400, "Motorcycle", 2200
690 DATA "Jade figurine", 2350, "Painting", 3450
9999 END

```

This distribution has 18 of the 25 items.

这次分配25件中的18件

Selected number of scrambles=25

指定的打乱次数=25

Smallest difference is \$ 700 in try 1 第1次试分差额为\$ 700

Smallest difference is \$ 500 in try 6

Smallest difference is \$ 400 in try 17

Smallest difference is \$ 300 in try 25

遗产 Estate 1:

Sterling silverware	\$2, 450.00
Spinnet	\$2, 500.00
China hutch	\$4, 300.00
Bedroom suite	\$5, 300.00
Oboe	\$1, 200.00
Packard coupe	\$6, 200.00
Pearl necklace	\$3, 300.00
Sail boat	\$2, 200.00
Coin collection	\$8, 000.00

=====
总和 Total: \$33, 450.00

遗产 Estate 2:	
Crystal chandelier	\$2, 400.00
Diamond pendant	\$3, 300.00
Stamp collection	\$5, 200.00
Movie camera	\$2, 400.00
Gold statuette	\$8, 200.00
Diamond solitaire	\$3, 550.00
Overstuffed chairs	\$2, 100.00
China	\$3, 200.00
Grand piano	\$3, 400.00

=====
总和Total \$33, 750.00

随机的正文生成

随机指针的另一个用途是用来产生正文。下面的两个程序以前都曾出现在文献中。第一个程序BLIP是本书作者之一(JDR)编写的,曾发表在1978年8月出版的《创造性计算(Creative Computing)》杂志上。第二个程序SIMP曾以几种不同的形式出现在多种杂志上,但是最为流行的形式是我们这里给出的。

程序BLIP能够产生形如“—— is the ——of——”的警句、格言、陈词滥调,甚至胡说八道。例如,“肥胖是暴食的结果”(Obesity is the retult of gluttony),若稍加修改,也能得到同样顺口的发人深省的连珠妙语,诸如“肥胖是成功的结果”(Obesity is the result of success)和“肥胖是成功的兄弟”(Obesify is the brother of success)。这样演变下去,将会无休无止,但是其中蕴含的“真理”却有待读者去解释。

```

10 ' 文件名: "blip"
20 ' 目的: 警句制造程序
30 ' 作者: jdr & jpg 9/82
40 ' 参考文献 Creative Computing, Aug. 1979, p. 116
50 '
60  RANDOMIZE: CLS
70 ' 本程序的要点是产生 "Blip is the blap of bleep." 形式的警句
80 ' X$=Blip Y$=Blap Z$=Bleep.
90  DIM X$(25), Y$(25), Z$(25)
100 FOR I=1 TO 25: READ X$(I), Y$(I), Z$(I): NEXT I
110  INPUT "How many aphorisms would you like": N
120  FOR I=1 TO N
130      J=INT(RND*25+1): K=INT(RND*25+1): L=INT(RND*25+1)
140      PRINT X$(J); "is the"; Y$(K); "of" Z$(L)
150  NEXT I
160 ' **** 数据语句
170  DATA Sanity, anathema, love., Agony, quagmire, ardor.
180  DATA Distrust, blight, beauty., Politics, harlot, hate.

```

190 DATA Greed, friend, filth., Gluttony, mother, evil.
 200 DATA Health, father, excess., Hysteria, sister, fantasy.
 210 DATA Brevity, brother, racism., Charity, enemy, laughter.
 220 DATA Security, bandit, naivete., Science, house, apoplexy.
 230 DATA Celibacy, kidney, sadism., Decency, harpy, lust.
 240 DATA Abuse, laxative, sex., Weakness, follicle, leprosy.
 250 DATA Death, heart, suicide., Bravery, reward, deceit.
 260 DATA Genius, genesis, loyalty., Alcohol, liver, success.
 270 DATA Violence, curator, idiocy., Apathy, eunuch, progress.
 280 DATA Money, result, failure., Garbage, apex, release.
 290 DATA Obesity, bladder, avarice.
 9999 END

Abuse is the blight of lust.
 Death is the brother of deceit.
 Charity is the harlot of release.
 Garbage is the curator of success.
 Abuse is the curator of rldase.
 Health is the genesis of apoplexy.
 Money is the father of sex.
 anity is the sister of progress.
 Hysterig is the enemy of failure.
 Agony the bladder of success.
 Security is the reward of loyalty.
 Bravery is the heart of laughter.
 Bravery is the quagmire of love.
 Agony is the genesis of failure.
 Obesity is the father of evil.
 Braveny is the harlot of excess.
 Money is the heart of excess.
 Apathy is the brother of idiocy.
 Bravery is the blight of excess.
 Charity is the anathema of excess.
 Celibacy is the laxative of release.
 Security is the eunuch of lust.
 Brevity is the genesis of apoplexy.
 Science is the eunuch of lust.
 Greed is the friend of laughter.
 Health is the kidney of success.
 Money is the curator of lust.
 Abuse is the curator of evil.
 Health is the brother of excess.
 Greed is the result of deceit.
 Decency is the follicle of avarice.
 Violence is the bendit of sadism.
 Agony is the kidney of progress.
 Violence is the friend of evil.
 Alcohol is the friend of excess.
 Bravery is the sister of avarice.
 Sanity is the kidney of love.
 Death is the liver of beauty.
 Apathy is the enemy of failure.
 Obesity is the reward of idiocy.
 Gluttony is the friend of deceit.
 Gluttony is the result of failure.
 Weakness is theheart of fantasy.
 Decency is the kidney of progress.
 Charity is the curator of apoplexy.

程序SIMP 对于计算领域的贡献尚难断言。它的目的并不很重要，只是要让计算机产生一些同在系统分析报告和进展报告中经常出现的技术套话相类似的文字材料。这个程序的迷人之处在于它的输出极易阅读。它输出的文字材料看起来和听起来都是如此美妙，以至让人不能不想把其中的几段插到他的报告里，看看是否有人能够着出这是计算机产生的正文。请高声地朗读这个程序输出的材料吧！你会感到很难把它同人写的报告区别开来。

```

10 ' 文件名: "Simp"
20 ' 目的: 产生玄妙哲理
30 ' 作者: jpg & jdr (after other like efforts) 9/82
40 '
50 RANDOMIZE; CLS
60 DIM X$(40)
70 FOR I=1 TO 40: READ X$(I): NEXT I
80 INPUT "How many paragraphs"; P
90 INPUT "How many sentences per paragraph"; S
100 CLS
110 FOR I=1 TO P
120 PRINT " ";
130 FOR J=1 TO S
140 A=INT(RND*10+1); B=INT(RND*10+11); C=INT(RND*10+21);
    D=INT(RND*10+31)
150 PRINT X$(A); X$(B); X$(C); X$(D); " "
160 NEXT J: PRINT
170 NEXT I
500 ' *****
510 ' 数据语句
520 DATA "In particular, ", "On the other hand, "
530 DATA "However, ", "Similarly, ", "In this regard, "
540 DATA "As a resultant implication, ", "For example, "
550 DATA "Based on integral subsystem considerations, "
560 DATA "Thus, ", "With respect to specific goals, "
570 ' ****
580 DATA "a large portion of the interface coordination communication"
590 DATA "a constant flow of effective information"
600 DATA "the characterization of specific criteria"
610 DATA "initialization of critical subsystem development"
620 DATA "the product configuration baseline"
630 DATA "the fully integrated test program"
640 DATA "any associated supporting element"
650 DATA "the incorporation of additional mission constraints"
660 DATA "the independent functional principle"
670 DATA "a primary interrelationship between system and/or
    subsystem technologies"
680 ' ****
690 DATA "must utilize and be functionally interwoven with"
700 DATA "maximizes the probability of project success and minimizes
    the cost and time required for"
710 DATA "adds explicit performance limits to"
720 DATA "necessitates that urgent consideration be applied to"
730 DATA "requires considerable systems analysis and trade-off studies to arrive at"
740 DATA "is further compounded, when taking into account"
750 DATA "presents extremely interesting challenges to"
760 DATA "recognizes the importance of other systems and the necessity for"
770 DATA "effects a significant implementation to"
780 DATA "adds overriding performance constraints to"

```