

Turbo Pascal 程序设计技巧示例

卢中辉 姚庭宝 编著

电子工

F
5244
(2)

73.874
C459
73.87429
C459

Turbo Pascal 程序设计技巧示例

圖書

卢中辉 姚庭宝 编著

電子工業出版社

(京)新登字 055 号

图书在版编目(CIP)数据

Turbo Pascal 程序设计技巧示例/卢中辉,姚庭宝编著. —北京:电子工业出版社,1994.9

ISBN7-5053-2596-5

I. T...

Ⅱ. ①卢...②姚...

Ⅲ. Pascal 语言-程序库-程序设计

Ⅳ. TP312PA

电子工业出版社出版

北京市海淀区万寿路 173 信箱 (100036)

电子工业出版社发行 各地新华书店经销

电子工业出版社计算机排版室排版

人民卫生出版社印刷厂印刷

*

开本: 850×1168 毫米 1/32 印张: 8 字数: 214 千字

1994 年 12 月第一版 1994 年 12 月北京第一次印刷

印数: 6000 册 定价: 9.80 元

ISBN 7-5053-2596-5/TP·784

前 言

本书专为已了解 Turbo Pascal 语言特性及程序设计基本方法与技巧的读者而写。书中精选了一百个较典型、具有一定技巧性的程序设计示例，近三分之一系近几年来国内外各类程序竞赛试题，并对其十类四十例就题意分析、算法选择、数据组织、程序设计思路以及编程实现等方面进行详述，提供多种解法，以期取得促进思维、举一反三之功效。全书着力于培养与加强广大读者的程序设计兴趣、提高灵活运用 Turbo Pascal 解决应用问题的本领并深入体现 Turbo Pascal 的潜在能力。

鉴于目前国内的计算机程序设计竞赛主要使用 Turbo Pascal 5.5 或 6.0 编程，为能与其接轨，本书中所有示例均按此要求编程，并在 Turbo Pascal 5.5 及 6.0 编译系统系统下都可编译通过且运行正确。为方便读者，促进交流，我们已将书中全部源程序代码存入软盘，并愿提供软盘拷贝。需要者可与作者联系。

对于那些既对 Turbo Pascal 深感兴趣而又苦于所知颇少、尚未入门的读者，请参阅电子工业出版社于一九九三年二月出版的《Turbo Pascal 大全》(姚庭宝、张帜编著)。该书提供了大量有关 Turbo Pascal 概念特性以及程序设计方法与技巧方面的信息，足以成为顺利阅读并理解本书的基础。

在本书编写过程中，杜秀荣和马兰春两位女士给予我们全力的支持，并参与了整理、定稿工作，在此表示衷心感谢。

虽然我们竭力想为读者奉献一本能体现并促进 Turbo Pascal 程序设计技巧的好书，但限于我们的经验与水平，本书定有不当之处，敬请广大专家、读者指正。

姚庭宝教授

长沙国防科技大学七系

卢中辉副教授

湖南省中小学计算机教育研究中心

一九九三年十月

目 录

第一章 显示图案	(1)
【题 1】 显示图案 1(1984 年全国青少年程序设计竞赛 高中组上机试题)	(1)
【题 2】 显示图案 2	(6)
【题 3】 显示图案 3	(9)
【题 4】 显示图案 4	(13)
第二章 逻辑判断	(18)
【题 5】 四大湖问题(1986 年湖南省青少年程序设计 竞赛试题).....	(18)
【题 6】 配备侦察员问题(1986 年广东省青少年程序 设计竞赛试题).....	(21)
【题 7】 白帽子和红帽子问题	(24)
【题 8】 四对夫妻问题	(27)
第三章 数据模拟	(34)
【题 9】 豆子下落问题(1986 年全国青少年程序设计 竞赛试题).....	(34)
【题 10】 扑克牌发牌	(36)
【题 11】 打印螺旋方阵	(41)
【题 12】 猴子分桃子	(46)
第四章 整数问题	(51)
【题 13】 填数字	(51)
【题 14】 巧分数字	(55)
【题 15】 回文数	(59)
【题 16】 01 数字串.....	(62)

第五章 数据处理	(68)
【题 17】 产生数字列	(68)
【题 18】 双重排序	(72)
【题 19】 查找足球八强(1986 年全国青少年程序设计 竞赛上机试题)	(77)
【题 20】 数制转换	(83)
第六章 数值计算	(88)
【题 21】 计算 π 的值	(88)
【题 22】 计算多项式的值	(93)
【题 23】 高精度乘幂运算	(94)
【题 24】 高精度除法运算(1993 年全国青少年信息学 奥林匹克竞赛试题)	(99)
第七章 递归与分治	(106)
【题 25】 字母排列	(106)
【题 26】 取数问题	(111)
【题 27】 路径问题	(116)
【题 28】 快速排序	(119)
第八章 回溯法	(125)
【题 29】 八皇后问题	(125)
【题 30】 排球运动员站位问题(1986 年全国青少年程 序设计竞赛试题)	(137)
【题 31】 填数问题 1(1990 年长沙市青少年计算机选拔 考试试题)	(141)
【题 32】 填数问题 2(第三届国际信息学奥林匹克竞赛 第一轮试题)	(146)
第九章 数据结构	(155)
【题 33】 火车调度问题	(155)
【题 34】 路径问题(1989 年上海市程序设计竞赛 试题)	(163)

【题 35】	液晶数字	(167)
【题 36】	地图四色问题	(171)
第十章	图的搜索	(179)
【题 37】	棋子移动问题(1987 年北京市程序设计竞赛 试题)	(179)
【题 38】	杯子分溶剂问题	(189)
【题 39】	中国盒子问题(1989 年第一届国际信息学奥 林匹克竞赛试题)	(196)
【题 40】	重排九宫问题	(216)
练习题		(230)

第一章 显示图案

计算机在正文方式下显示图案,题目变化多,具有趣味性。编程序时要灵活运用标准函数,使程序简洁、结构清楚,尽量使程序具有通用性。

【题1】显示图案1

(1984年全国青少年程序设计竞赛高中组上机试题)

```

          9
        9 9 9
      8 8 8 8 8
    8 8 8 8 8 8 8
  7 7 7 7 7 7 7 7 7
7 7 7 7 7 7 7 7 7 7 7
6 6 6 6 6 6 6 6 6 6 6 6
 6 6 6 6 6 6 6 6 6 6 6 6 6
5 5 5 5 5 5 5 5 5 5 5 5 5 5
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
   4 4 4 4 4 4 4 4 4 4 4 4 4
    3 3 3 3 3 3 3 3 3 3 3 3
     3 3 3 3 3 3 3 3 3 3
      2 2 2 2 2 2 2 2 2
       2 2 2 2 2 2 2
        1 1 1 1 1
         1 1 1
          0

```

正文方式下显示图案用循环程序设计实现。通常要确定每行第一个字符的位置、每行显示字符的数目以及要显示的字符等。它们与图案的行号之间有一个函数关系。可列个表帮忙寻找函数关系。如果把图案显示在正中间,则函数关系如下:

行 号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
第一个字符的位置	40	39	38	37	36	35	34	33	32	31	32	33	34	35	36	37	38	39	40
显示字符的数目	1	3	5	7	9	11	13	15	17	19	17	15	13	11	9	7	5	3	1
显示的字符	9	9	8	8	7	7	6	6	5	5	4	4	3	3	2	2	1	1	0

设行号用变量 i 表示,接下来的工作就是如何确定三个函数关系。

(1) 确定第一个字符的位置与 i 的函数关系

变量 i 增大时,函数值先是减小,到过了 $i=9$ 时,函数值增大,函数在 $i=9$ 时有最小值 31,且关于 $i=9$ 对称。绝对值函数也具有类似的性质,故用绝对值函数能方便地构造此函数关系,即 $31 + \text{abs}(i-9)$,它用于表示第一个字符的位置。从而显示此字符前应留的空格数为 $30 + \text{abs}(i-9)$ 。

(2) 确定显示字符的数目与 i 的函数关系

变量 i 与显示字符的数目也是关于 $i=9$ 的对称函数, $i=9$ 时有最大值 19,每当 i 增大或减小 1 时,函数相应地增大或减小 2,所以函数关系可表示为 $19 - 2 * \text{abs}(i-9)$ 。它可转化为循环变量的初值和终值分别取作 0 和 $18 - 2 * \text{abs}(i-9)$ 。

(3) 确定显示的字符与 i 的函数关系

每当变量 i 增大 2 时,函数值相应地减小 1,故函数关系可表示为 $9 - i \text{ div } 2$ 。

程序 1

```

program ex1_a;
var
  i, j : integer;
begin
  for i := 0 to 18 do
    begin
      write(' ', 30 + abs(i - 9));
      for j := 0 to 18 - 2 * abs(i - 9) do
        write(9 - i div 2);
    end
  end

```

```

        writeln
    end
end.

```

运行程序 1 能显示题目中的图案。但要编写好程序,就不能满足于已编出了的程序,要不断地改进完善。由于图案上下对称,若将行号改为-9 至 9,则可简化程序。

程序 2

```

program ex1_b;
var
    i, j : integer;
begin
    for i := -9 to 9 do
        begin
            write(' ', 30 + abs(i));
            for j := 0 to 18 - 2 * abs(i) do
                write(9 - (i + 9) div 2);
            writeln
        end
    end.

```

一个好的程序应该有一定的通用性。引进常数标识符 n 用于表示菱形顶上的数字,可使程序更清楚,且具有通用性。把 n 改为小于 9 的正整数时,可显示类似的图案。比如,在 $n=4$ 时显示图案为:

```

          4
        4 4 4
      3 3 3 3 3
    3 3 3 3 3 3 3
  2 2 2 2 2 2 2 2 2
  2 2 2 2 2 2 2
    1 1 1 1 1
      1 1 1
        0

```

程序 3 中引进变量 k , 用于表示第 i 行第一个字符离纵向中间对称轴的距离。

程序 3

```

program ex1_c;
const
    n = 9;
var
    i, j, k : integer;
begin
    for i := -n to n do
        begin
            k := n - abs(i);
            write(' ':39 - k);
            for j := 0 to 2 * k do
                write(n - (i + n) div 2);
            writeln
        end
    end.

```

进一步增大题目的难度,改为显示空心菱形,空心菱形边的厚度为 m 个字符。可用“填空法”编程序,即在空的地方显示空格。比如 $n=6$, $m=3$ 时显示图案为:

```

      0
    6 6 6
  5 5 5 5 5
 3 5 5 5 5 5
 4 4 4      4 4 4
 4 4 4      4 4 4
3 3 3      3 3 3
3 3 3      3 3 3
 2 2 2      2 2 2
  2 2 2  2 2 2
    1 1 1 1 1
      1 1 1
        0

```

程序 4

```

program ex1_d;

```

```

const
    n = 9; m = 3;
var
    i, j, k : integer;
begin
    for i := -n to n do
        begin
            k := n - abs(i);
            write(' ', 39 - k);
            for j := -k to k do
                if abs(j) > k - m
                    then write(n - (i + n) div 2)
                    else write(' ');
            writeln
        end
    end.

```

空心图形也可用“跳跃法”编程序,即遇到空心部分,跳到后面有字符的地方。这时要用标准库单元 crt 中的预定义过程语句 gotoxy 来给光标定位。

程序 5

```

program ex1_e;
uses
    crt;
const
    n = 9; m = 3;
var
    i, j, k : integer;
begin
    clrscr;
    for i := -n to n do
        begin
            k := n - abs(i);

```

```

for j := k to k do
begin
    gotoxy(40 + j, 12 + i);
    write(n := (i + n) div 2);
    if (k >= m) and (j = m - k + 1) then
        j := k - m;
end
end
end.

```

【题2】显示图案2

```

1  0  2  0  3  0  4  0  5
    1  0  2  0  3  0  4
        1  0  2  0  3
            1  0  2
                1
                    -1  0  -2
                        -1  0  -2  0  -3
                            -1  0  -2  0  -3  0  -4
                                -1  0  -2  0  -3  0  -4  0  -5

```

本题的难点是怎样确定要显示的数据,在同一行中数据是不同的,且有些数据是负的。可采用逐步深入的方法解决。先把有显示的数据简化为全是1,编程序显示出形状相同的图案:

```

1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1
    1 1 1 1 1
      1 1 1
        1
          1 1 1
            1 1 1 1 1
              1 1 1 1 1 1 1
                1 1 1 1 1 1 1 1 1

```

这个图案也是上下对称的,行号变化可从-4到4。可用上题类似的方法列出行号与第一个字符的位置以及行号与该行显示数据的数目之间的函数关系:

行 号	-4	-3	-2	-1	0	1	2	3	4
第一个字符的位置	27	30	33	36	39	36	33	30	27
显示数据的数目	9	7	5	3	1	3	5	7	9

行号用变量 i 表示, 第一个字符的位置为 $39 - 3 * \text{abs}(i)$, 显示数据的数目为 $2 * \text{abs}(i) + 1$. 引进常数标识符 $n=4$, 改变 n 的大小可显示大小不同的类似图案。

程序 1

```

program ex2_a;
const
    n = 4;
var
    i, j : integer;
begin
    for i := -n to n do
        begin
            write(' ', 39 - 3 * abs(i));
            for j := 0 to 2 * abs(i) do
                write(1;3);
            writeln
        end
    end.

```

运行程序 1 后显示如上所述的全 1 图案。

接下来研究怎样在同一行中显示不同的数据, 目前暂且不考虑数据的负号, 即欲显示下述图案:

```

      1 0 2 0 3 0 4 0 5
        1 0 2 0 3 0 4
          1 0 2 0 3
            1 0 2
              1
                1 0 2
                  1 0 2 0 3
                    1 0 2 0 3 0 4
                      1 0 2 0 3 0 4 0 5

```

由于每一行第一个数据都是 1,且后面的数据变化情况相同,因此可列出相对序列号与数据的函数关系:

序列号	0	1	2	3	4	5	6	7	8
数 据	1	0	2	0	3	0	4	0	5

序列号用变量 j 表示, j 为偶数时,数据为 $j/2+1$, j 为奇数时,数据为 0. 只要找到一个表达式,它在 j 为偶数时为 1,奇数时为 0,就能把上面的两种情况统一用一个表达式表示。 $(j+1) \bmod 2$ 就是在 j 为偶数时为 1,奇数时为 0,从而,表达式 $((j+1) \bmod 2) * (j/2+1)$ 就能正确表示序列号与数据的函数关系。

程序 2

```

program ex2_b;
const
    n = 4;
var
    i, j : integer;
begin
    for i := -n to n do
        begin
            write(' ', 38 - 3 * abs(i));
            for j := 0 to 2 * abs(i) do
                write(((j + 1) mod 2) * (j / 2 + 1); 3; 0);
            writeln
        end
    end.

```

运行程序 2 后。将显示上述不带负号且数据不同的图案。

最后要解决数据的符号问题,即欲显示本题一开始给出的图案。从图案可见,当 $i \leq 0$ 时数据非负, $i > 0$ 时数据非正。故可构造一个函数,令其在 $i \leq 0$ 时为 1,在 $i > 0$ 时为 -1,将它与前面程序中显示的数据相乘,就能得到题目中要显示的数据。此函数可用 $\text{abs}(0.5 - i) / (0.5 - i)$ 来实现,而程序 3 即成为完成本题的最终形式。

程序 3

```
program ex2_c;  
const  
    n = 4;  
var  
    i, j : integer;  
begin  
    for i := -n to n do  
        begin  
            write(' ', 38 - 3 * abs(i));  
            for j := 0 to 2 * abs(i) do  
                write(abs(0.5 - i) / (0.5 - i) * ((j + 1) mod 2) * (j / 2 + 1); 3; 0);  
            writeln  
        end  
    end.  
end.
```

【题 3】显示图案 3

```
      0  
    1  2  
  3  4  5  
6  7  8  9  
  A  B  C  D  E  
  F  G  H  I  J  K  
 L  M  N  O  P  Q  R  
S  T  U  V  W  X  Y  Z
```

本题要显示的图案中有数字和字母,可统一作为字符处理。使用与前两题类似的方法,可以用二重循环来实现。主要问题是确定要显示的字符。

字符“0”到“9”的 ASCII 码是 48 到 57,字符“A”到“Z”的 ASCII 码是 65 到 90,关键是确定要显示字符的 ASCII 码。有两种方法,一种方法是用变量来构造要显示字符的 ASCII 码,另一种方法是用数学的方法计算要显示字符的 ASCII 码。首先用一种普通的方法。变量 s 的值首

先是“0”的 ASCII 码 48, 然后每次加 1, 当 s 的值为“9”的 ASCII 码 57 后, 用 if 语句来个“跳跃”, 变为“A”的 ASCII 码 65。

程序 1

```
program ex3_a;  
const  
    n := 7;  
var  
    i, j, s : byte;  
begin  
    s := 47;  
    for i := 0 to n do  
        begin  
            write(' ', 39 - i);  
            for j := 0 to i do  
                begin  
                    s := s + 1;  
                    if s = 58 then  
                        s := 65;  
                    write(chr(s); 2)  
                end;  
            writeln  
        end  
    end.
```

程序可作一些小的改进。只要确定了每行第一个字符的 ASCII 码后, 就能很方便地确定本行中各个字符的 ASCII 码。

程序 2

```
program ex3_b;  
const  
    n := 7;  
var  
    i, j, s : byte;  
    • 10 •
```