JAPLA 研究会資料 2018/8/4

Jプログラムにより「誰でもできる奇数次の魔方陣」をつくってみよう

西川　利男

Jは配列処理のプログラム言語である。さりとて、統計計算や線形代数の計算以外はやってはいけないなどと言うことはない。今年の猛暑を忘れさせるJの楽しみとして、以下のようなJによる魔方陣のプログラミングをやってみた。

伊達宗行、「理科で歴史を読み直す」ちくま新書、(2010)

この本は理科の考え方で、縄文時代から南方熊楠、宮沢賢治までの歴史をアルスとして考えてみようという、伊達先生のユニークな語り口で、読者を楽しませてくれる。

この中のp. 227に以下のような魔方陣の話がある。

1. Jでやってみよう　最初のステップをJでやってみよう。ここでは、他の言語にはない

伊達宗行(だて・むねゆき)
1929年仙台生まれ。52年東北大学理学部卒業。55年東北大学理学研究科物理専攻中退。理学博士。専門は物性物理学。大阪大学理学部長、日本原子力研究所先端基礎研究センター長、日本物理学会会長、日本学術会議会員、同第四部長などを歴任。大阪大学名誉教授。現在、研究助成財団新世代研究所理事長。著書に『新しい物性物理』『極限の科学』(以上、講談社ブルーバックス)、『「数」の日本史』(日経ビジネス人文庫)がある。

「理科」で歴史を読みなおす　伊達宗行

†誰でもできる奇方陣

建部の魔方陣ほど高度のものでなければ、奇数の方陣はわけなくできる。いくら大きいのでも同じルールで作れるのである。その一つをお見せしよう。

図5－12に太線でかこまれた枠に五方陣を作る。まず1を中央真下に置く。そして右斜め下へ2、3、……と置いて行く。すると3がはみ出すが、3の位置に対応する元の枠に移す。4も同じようにする。そして5まで行くと1にぶつかるが、その時は1の左斜め下に行く。すると10まで行けるが、ここでまた図のように11に行く。このようにして16までを書き込んでである。以下12と8の中間に25が来て完結するのを確認していただきたい。

このやり方はやさしく、覚えやすいので即興で書いて見せることもできる。魔方陣を知らない友人を驚嘆させることは請け合

11		7		3	
4	12		8	16	4
	5	13		9	
10		1	14		10
	6		2	15	
11		7		3	
				16	

図5－12　五方陣（奇数方陣）の作り方

Jの次の機能が効果的に使われている。（以下のプログラムを参照）

・配列要素の入れ替え　amend (})
　（新しい配列）=:（入れ替え値）　（入れ替え位置）　}　（元の配列）
・法 Modulus　つまり割った余り　|

```
M =: 5 5$0
```

```
   M
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
   M =: 1 (<3, 2) } M
   M
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 1 0 0
0 0 0 0 0
   M =: 2 (<4, 3) } M
   M
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 1 0 0
0 0 0 2 0
   M =: 3 (< (5|5), (5|4)) } M
   M
0 0 0 0 3
0 0 0 0 0
0 0 0 0 0
0 0 1 0 0
0 0 0 2 0
   M =: 4 (< (5|6), (5|5)) } M
   M
0 0 0 0 3
4 0 0 0 0
0 0 0 0 0
0 0 1 0 0
0 0 0 2 0
```

このように、1からステップごとに値を増やし、斜めにいれる。行や列からはみ出したら、次の行や列に進める。

２．ステップを進めるには－いろいろの試行錯誤の末、やっとのプログラム
さらにスマホ版を完成させた

最初のステップでは、あざやかに進んでいったが、やがてステップが進むに従い、以前に入れた値との衝突が起こる。このときの解決操作を伊達先生の本に書いてあるが、プログラムとしてはけっこう頭を悩ました。（楽しませてくれた。）

およそスマートなコーディングとは言えないが、衝突かどうかをテストして、次のステップをきめる。やっとのことで、魔方陣の各値が決められるようになった。

よくＪのプログラムはループなしでといわれているが、無理をしないでループ構造でステップを進めるようにした。

さらに、スマホ版として、Huewei タブレット上のＪ Android の上で、動くようにした。そのＪプログラムはつぎのとおりである。

```
NB. Magic Square / Sumaho Version ==================================
NB.   2018/7/29 by T.Nishikawa
wr =: 1!:2&2
N =: 5
M =: (N, N)$0
P =: (N + 1) % 2
Q =: (N - 1) % 2
I =: 1
go =: 3 : 0
wr 'n = ', (":I)
wr M =: I (<P, Q) } M
if. 25 <: I do.  goto_fin.  end.
NB. test of next
  P =: N|P+1
  Q =: N|Q+1
  t =. (<P, Q) { M
wr 'test (', (":P),',', (":Q), ') = ', (": t)
if. 0 = t
  do.
    M =: I (<P, Q) } M
  else.
    P =: N|P + 1
    Q =: N|Q - 1
    M =: I (<P, Q) } M
end.
I =: I + 1
label_fin.
''
)
```

動詞 go ''を繰り返し実行することで、魔方陣はワンステップずつ、作られていく。

実行にあたり、注意することとして、途中の配列Ｍおよびインデックスなど P,Q,I の値をグローバルで使用しているので、再度実行するときは、clear ''とした後、プログラムを新しくロードして行う。

```
   go ,,
n = 1
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 1 0 0
0 0 0 0 0
test (4,3) = 0

   go ,,
n = 2
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 1 0 0
0 0 0 2 0
test (0,4) = 0

   go ,,
n = 3
0 0 0 0 3
0 0 0 0 0
0 0 0 0 0
0 0 1 0 0
0 0 0 2 0
test (1,0) = 0

   go ,,
n = 4
0 0 0 0 3
4 0 0 0 0
0 0 0 0 0
0 0 1 0 0
0 0 0 2 0
test (2,1) = 0

 (途中省略)

   go ,,
n = 20
11  0  7 20  3
 4 12  0  8 16
17  5 13  0  9
10 18  1 14  0
 0  6 19  2 15
test (1,4) = 16
   go ,,
n = 21
11  0  7 20  3
 4 12  0  8 16
17  5 13 21  9
10 18  1 14  0
 0  6 19  2 15
test (3,4) = 0

   go ,,
n = 22
11  0  7 20  3
 4 12  0  8 16
```

```
17  5 13 21  9
10 18  1 14 22
 0  6 19  2 15
test (4,0) = 0

   go ,,
n = 23
11  0  7 20  3
 4 12  0  8 16
17  5 13 21  9
10 18  1 14 22
23  6 19  2 15
test (0,1) = 0

   go ,,
n = 24
11 24  7 20  3
 4 12  0  8 16
17  5 13 21  9
10 18  1 14 22
23  6 19  2 15
test (1,2) = 0

   go ,,
n = 25
11 24  7 20  3
 4 12 25  8 16
17  5 13 21  9
10 18  1 14 22
23  6 19  2 15
test (2,3) = 21
```

３．魔方陣としてのチェックを行う。

```
   MAG =: M
   MAG
11 24  7 20  3
 4 12 25  8 16
17  5 13 21  9
10 18  1 14 22
23  6 19  2 15
```

・タテ方向の和

```
   +/MAG
65 65 65 65 65
```

・ヨコ方向の和

```
   +/"(1) MAG
65 65 65 65 65
```

・斜め方向（＼）の和

```
   (i.5),.(i.5)
0 0
1 1
2 2
3 3
4 4
   <"(1) (i.5),.(i.5)
+---+---+---+---+---+
|0 0|1 1|2 2|3 3|4 4|
+---+---+---+---+---+
   (<"(1) (i.5),.(i.5)) { MAG
11 12 13 14 15
   +/ (<"(1) (i.5),.(i.5)) { MAG
65
```

・斜め方向（／）の和

```
   (|.i.5),.(i.5)
4 0
3 1
2 2
1 3
0 4
   <"(1) (|.i.5),.(i.5)
+---+---+---+---+---+
|4 0|3 1|2 2|1 3|0 4|
+---+---+---+---+---+
   (<"(1) (|.i.5),.(i.5)) { MAG
23 18 13 8 3
   +/ (<"(1) (|.i.5),.(i.5)) { MAG
65
```

すでに作成したが、以下はパソコン上で、プログラムの進行のチェックのため、トレース機能をプログラムの実行状況である。

全体のプログラムは最後にあげてある。

```
load'f:¥j402¥user¥magic_square.ijs'
```

10 ステップ以降、トレースさせて、実行する。

```
MAG =: 10 run 5

No. 0========
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

No. 1========
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 1 0 0
0 0 0 0 0

No. 2========
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 1 0 0
0 0 0 2 0

No. 3========
0 0 0 0 3
0 0 0 0 0
0 0 0 0 0
0 0 1 0 0
0 0 0 2 0

No. 4========
0 0 0 0 3
4 0 0 0 0
0 0 0 0 0
0 0 1 0 0
0 0 0 2 0

No. 5========
0 0 0 0 3
4 0 0 0 0
0 5 0 0 0
0 0 1 0 0
0 0 0 2 0

No. 6========
0 0 0 0 3
4 0 0 0 0
0 5 0 0 0
0 0 1 0 0
```

0 6 0 2 0

No. 7========
0 0 7 0 3
4 0 0 0 0
0 5 0 0 0
0 0 1 0 0
0 6 0 2 0

No. 8========
0 0 7 0 3
4 0 0 8 0
0 5 0 0 0
0 0 1 0 0
0 6 0 2 0

No. 9========
0 0 7 0 3
4 0 0 8 0
0 5 0 0 9
0 0 1 0 0
0 6 0 2 0

```
No. 10========
 0 0 7 0 3
 4 0 0 8 0
 0 5 0 0 9
10 0 1 0 0
 0 6 0 2 0

No. 11========
p q t:1 1 0
OK
1 1
11  0 7 0 3
 4 12 0 8 0
 0  5 0 0 9
10  0 1 0 0
 0  6 0 2 0

No. 12========
p q t:2 2 0
OK
2 2
11  0  7 0 3
 4 12  0 8 0
 0  5 13 0 9
10  0  1 0 0
 0  6  0 2 0

No. 13========
p q t:3 3 0
OK
3 3
11  0  7  0 3
 4 12  0  8 0
 0  5 13  0 9
10  0  1 14 0
 0  6  0  2 0

No. 14========
p q t:4 4 0
OK
4 4
11  0  7  0  3
 4 12  0  8  0
 0  5 13  0  9
10  0  1 14  0
 0  6  0  2 15
No. 15========
p q t:0 0 11
NO
11  0  7  0  3
 4 12  0  8 16
 0  5 13  0  9
10  0  1 14  0
 0  6  0  2 15

No. 16========
p q t:2 0 0
OK
2 0
```

```
11  0  7  0  3
 4 12  0  8 16
17  5 13  0  9
10  0  1 14  0
 0  6  0  2 15

No. 17========
p q t:3 1 0
OK
3 1
11  0  7  0  3
 4 12  0  8 16
17  5 13  0  9
10 18  1 14  0
 0  6  0  2 15

No. 18========
p q t:4 2 0
OK
4 2
11  0  7  0  3
 4 12  0  8 16
17  5 13  0  9
10 18  1 14  0
 0  6 19  2 15

No. 19========
p q t:0 3 0
OK
0 3
11  0  7 20  3
 4 12  0  8 16
17  5 13  0  9
10 18  1 14  0
 0  6 19  2 15

No. 20========
p q t:1 4 16
NO
11  0  7 20  3
 4 12  0  8 16
17  5 13 21  9
10 18  1 14  0
 0  6 19  2 15

No. 21========
p q t:3 4 0
OK
3 4
11  0  7 20  3
 4 12  0  8 16
17  5 13 21  9
10 18  1 14 22
 0  6 19  2 15

No. 22========
p q t:4 0 0
OK
4 0
```

```
11  0  7 20  3
 4 12  0  8 16
17  5 13 21  9
10 18  1 14 22
23  6 19  2 15

No. 23========
p q t:0 1 0
OK
0 1
11 24  7 20  3
 4 12  0  8 16
17  5 13 21  9
10 18  1 14 22
23  6 19  2 15
```

```
No. 24========
p q t:1 2 0
OK
11 24  7 20  3
 4 12 25  8 16
17  5 13 21  9
10 18  1 14 22
23  6 19  2 15

No. 25========
p q t:2 3 21
NO
11 24  7 20  3
 4 12 25  8 16
17  5 13 21  9
10 18 26 14 22
23  6 19  2 15
```

```
NB. 魔方陣の検証
   MAG
11 24  7 20  3
 4 12 25  8 16
17  5 13 21  9
10 18  1 14 22
23  6 19  2 15

   +/ MAG
65 65 65 65 65
   +/"(1) MAG
65 65 65 65 65

NB. 斜め左上から右下へ
     (i.5),.(i.5)
0 0
1 1
2 2
3 3
4 4
   <"(1) (i.5),.(i.5)
+---+---+---+---+---+
|0 0|1 1|2 2|3 3|4 4|
+---+---+---+---+---+
   (<"(1) (i.5),.(i.5)) { MAG
11 12 13 14 15
   +/ (<"(1) (i.5),.(i.5)) { MAG
65

NB. 斜め左下から右上へ
   |: i. 5
0 1 2 3 4
   |. i. 5
4 3 2 1 0
   <"(1) (|. i.5),.(i.5)
+---+---+---+---+---+
|4 0|3 1|2 2|1 3|0 4|
+---+---+---+---+---+
   (<"(1) (|. i.5),.(i.5)){ MAG
23 18 13 8 3
   +/ (<"(1) (|. i.5),.(i.5)){ MAG
65
```

```
  NB. Magic Square for Odd Order
NB.   2018/7/19 T.Nishikawa

rd =: 1!:1
wr =: 1!:2&2

NB. Magic Square for Odd order ================================
NB.       run 5 => do 5x5 magic square
NB.    10 run 5 => trace after 10 repetitions 5x5 magic square
run =: 3 : 0
(*: y.) run y.
:
trace =. x.
N =. y.
M =. (N, N)$0
NB. M =. 5 5$0
NB. N =. 5
i =. 0
p =. (N + 1) % 2
q =. (N - 1) % 2
while. i < (>: *: N)
NB. while. i < 17
  do.

     YN =. rd 1
NB.      if. 'n' = YN do. return. end.
     wr 'No. ', (": i), '========'

     wr M
     MA =. M
NB. test current value with previous one whether or not collide ========
     t =. (<p, q) { M
if. i > trace do. wr 'p q t:', ": p, q, t end.
     if. 0 = t
          do.
if. i > trace do.
               wr 'OK'
               wr p, q
              end.
             M =. (>:i) (<p, q) } M
if. i > trace do.  wr M end.
             p =. N|p + 1
             q =. N|q + 1
             i =. i + 1
           else.
if. i > trace do.
                wr 'NO'
                wr p, q
               end.
             pp =. N|p + 1
             qq =. N|q - 1
if. i > trace do.  wr p, qq end.
             M =. (>:i) (<pp, qq) } M
if. i > trace do.  wr M  end.
             p =. N|pp + 1
             q =. N|qq + 1
             i =. i + 1
     end.
   end.
```

MA
)