Chapter 1: Fundamental Study of ‘Prototype Squares’ and ‘Do-it-After-the-Model Transformation’: **Kanji Setsuda**

Section 3: Prototype Squares of Order 5 and Three Types of Magic Squares of Order 5

#1. What are the 'Prototype Squares' of Order 5?

Let's study about the ‘Prototype Squares’ of order 5 and their structural relations with three types of magic squares of order 5: (1) Self-complementary, (2) Pan-diagonal, and (3) Simultaneous: both self-complementary and pan-diagonal.

**** Two Sets of Squares and ‘Do-it-After-the-Model Transformation’ ****

```
                    .--------------.
                    |Representative|
                .-->|    Model     |--->.
                |   |  Nominated   <--. |
                |   '--------------'  | |
                | (Application 'DAMT')| |
 .--------------|-------.      .------|-v---------.
 |   Solution           |      |    Solution      |
 |    Set of            |      |     Set of       |
 |  'Prototype          |      |   'Object        |
 |    Squares'          |      |     Squares'     |
 |                      | <---> |                 |
 |   (Original)         | 'One- |    (Target)     |
 '----------------------'  to-  '-----------------'
                 One' Correspondence
```

The next list shows some samples of ‘Prototype Squares’ of order 5. The top square 1/ below is assumed as a representative of all and made after the square array 5 by 5.
Any of the Prototype Squares is designed for a mother solution whose child should be an object solution made by the single ‘**D**o-it-**A**fter-the-**M**odel **T**ransformation’.

**** Sample List of ‘Prototype Squares’ of Order 5 ****

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

We want to have the object solutions of self-complementary type re-composed by 'DAM Transformation' finally such as:

```
**  Sample List of Self-Complementary Magic Squares 5x5  **
               1/                2/                3/                4/
  1 23 17 22  2     1 23 17 22  2     1 23 17 22  2     1 23 17 22  2
 21  8 11  6 19    20  7 15  5 18    19  6 11  8 21    18  5 15  7 20
 12 10 13 16 14    12 10 13 16 14    16 14 13 12 10    16 14 13 12 10
  7 20 15 18  5     8 21 11 19  6     5 18 15 20  7     6 19 11 21  8
 24  4  9  3 25    24  4  9  3 25    24  4  9  3 25    24  4  9  3 25
               5/                6/                7/                8/
  1 23 17 22  2     1 23 17 22  2     1 23 17 22  2     1 23 17 22  2
  8 21 11 19  6     7 20 15 18  5     6 19 11 21  8     5 18 15 20  7
 12 10 13 16 14    12 10 13 16 14    16 14 13 12 10    16 14 13 12 10
 20  7 15  5 18    21  8 11  6 19    18  5 15  7 20    19  6 11  8 21
 24  4  9  3 25    24  4  9  3 25    24  4  9  3 25    24  4  9  3 25
               9/               10/               11/               12/
  1 23 18 21  2     1 23 18 21  2     1 23 18 21  2     1 23 18 21  2
 22  6 10 12 15    22 12  6 10 15    20 14  4 10 17    19 15 12 10  9
  7 17 13  9 19     7  9 13 17 19    11  7 13 19 15     4  6 13 20 22
 11 14 16 20  4    11 16 20 14  4     9 16 22 12  6    17 16 14 11  7
 24  5  8  3 25    24  5  8  3 25    24  5  8  3 25    24  5  8  3 25
  . . . . . .
```

Look at the next figures below. They are the primary forms of Prototype Squares and object solutions. The one on the left hand side is made after the regular array 5 by 5, and each value of 25 positions is taken as equal to its position name itself.

```
[Primary Forms of Prototype Square and Model Solution]

24 25 21 22 23 24 25 21 22     3 25 17 14  6  3 25 17 14
      .--------------.               .--------------.
 4  5|  1|  2|  3|  4|  5| 1  2   12  9|  1|23|20|12|  9| 1 23
     |--+--+--+--+--|               |--+--+--+--+--|
 9 10|  6|  7|  8|  9|10| 6  7    21 18|15|  7|  4|21|18|15  7
     |--+--+--+--+--|               |--+--+--+--+--|
14 15|11|12|13|14|15|11 12    10  2|24|16|13|10|  2|24 16
     |--+--+--+--+--|               |--+--+--+--+--|
19 20|16|17|18|19|20|16 17    19 11|  8|  5|22|19|11|  8  5
     |--+--+--+--+--|               |--+--+--+--+--|
24 25|21|22|23|24|25|21 22     3 25|17|14|  6|  3|25|17 14
      '--------------'               '--------------'
 4  5  1  2  3  4  5  1  2    12  9  1 23 20 12  9  1 23
```

The one on the right hand side is the primary representative form of object squares. It is selected not only for the 'Model Solution', but also for the purpose of getting the rules of 'DAM Transformation' after it.

This time I would choose this sample for the only common Model for all three types of object squares. It originally belongs to the 'simultaneous type', since it has the properties of both self-complementary and pan-diagonal at the same time.

#2. 'DAM Transformation' and the Common Rules of Transformation

"**D**o-it-**A**fter-the-**M**odel **T**ransformation" is the way of making our object solutions from the Prototype Squares. Rules of transformation are taken after the Model itself and dictated as follows:

```
(1) Put the value of original  n1 into the object n1.
(2) Put the value of original n23 into the object n2.
(3) Put the value of original n20 into the object n3.
(4) Put the value of original n12 into the object n4.
(5) Put the value of original  n9 into the object n5.
```

```
 (6) Put the value of original n15 into the object n6.
 (7) Put the value of original  n7 into the object n7.
 (8) Put the value of original  n4 into the object n8.
 (9) Put the value of original n21 into the object n9.
(10) Put the value of original n18 into the object n10.
 . . . . . . . . .
(24) Put the value of original  n3 into the object n24.
(25) Put the value of original n25 into the object n25.
```

```
/**/
/* Do-it-After-the-Model Transformation */
void damt(void){
  d[1]=c[1];    d[2]=c[23];   d[3]=c[20];   d[4]=c[12];   d[5]=c[9];
  d[6]=c[15];   d[7]=c[7];    d[8]=c[4];    d[9]=c[21];   d[10]=c[18];
  d[11]=c[24];  d[12]=c[16];  d[13]=c[13];  d[14]=c[10];  d[15]=c[2];
  d[16]=c[8];   d[17]=c[5];   d[18]=c[22];  d[19]=c[19];  d[20]=c[11];
  d[21]=c[17];  d[22]=c[14];  d[23]=c[6];   d[24]=c[3];   d[25]=c[25];
}
/**/
```

Let's assume them as the only common rules for all three types of object squares of order 5, and let's apply this set of rules to each Prototype Square and get each corresponding object solution of all types.

First of all we have to make the three sets of Prototype Squares, which contain:
(1) 48544 solutions for Self-complementary type
(2) 3600 solutions for Pan-diagonal type
(3) 16 solutions for Simultaneous type: both self-complementary and pan-diagonal.

#3. How to make the Prototype Squares for Self-Complementary MS55

How do we have to define to make each set of Prototype Squares of order 5? What are the basic conditions for getting all solutions we want to have?

We must study the Model Solution at first and get everything from it.

When we assume the next diagram as another basic form of Prototype Squares and define such the conditions of it as listed below for the self-complementary magic objects, we can always get the appropriate Prototype solutions we have just wanted.

```
 [Basic Diagram and Conditions for 'Prototype Squares 5x5']
    'Prototype Squares'              The Model Solution
 24 25 21 22 23 24 25 21 22       3 25 17 14  6  3 25 17 14
       .--------------.                 .--------------.
  4  5| 1| 2| 3| 4| 5| 1  2      12  9| 1|23|20|12| 9| 1 23
       |--+--+--+--+--|                 |--+--+--+--+--|
  9 10| 6| 7| 8| 9|10| 6  7      21 18|15| 7| 4|21|18|15  7
       |--+--+--+--+--|                 |--+--+--+--+--|
 14 15|11|12|13|14|15|11 12      10  2|24|16|13|10| 2|24 16
       |--+--+--+--+--|                 |--+--+--+--+--|
 19 20|16|17|18|19|20|16 17      19 11| 8| 5|22|19|11| 8  5
       |--+--+--+--+--|                 |--+--+--+--+--|
 24 25|21|22|23|24|25|21 22       3 25|17|14| 6| 3|25|17 14
       '--------------'                 '--------------'
  4  5  1  2  3  4  5  1  2      12  9  1 23 20 12  9  1 23
* Basic Conditions *
  n1+n23+n20+n12+n9=C    ... (1) | n1+n15+n24+n8+n17=C    ... (6)
  n15+n7+n4+n21+n18=C    ... (2) | n23+n7+n16+n5+n14=C    ... (7)
```

```
  n24+n16+n13+n10+n2=C   ... (3) | n20+n4+n13+n22+n6=C   ... (8)
  n8+n5+n22+n19+n11=C    ... (4) | n12+n21+n10+n19+n3=C  ... (9)
  n17+n14+n6+n3+n25=C    ... (5) | n9+n18+n2+n11+n25=C   ...(10)
* Self-Complementary Conditions: *
  n1+n25=n23+n3=n20+n6=n12+n14=n9+n17=n15+n11=n7+n19
  =n4+n22=n21+n5=n18+n8=n24+n2=n16+n10=n13+n13=PSM       ...(cp)
* List-forming Inequality Conditions: *
  n1<n25; n1<n9; n1<n17; and n23>n15;
```

Position names of variables n1, n2, n3, . . . , n24, n25 are just the same with the ones of primary Prototype Square. Pay your special attention to that.

#4. Apply ‘DAMT’ to those Prototypes and Re-compose our Object Squares.

Let’s apply ‘Do-it-After-the-Model Transformation’ to each Prototype Squares 5x5 and make it into each self-complementary magic square 5x5, shall we?

We use those rules of transformations described before in the chapter #2.

The next list shows the result of transformations. Each couple has the Prototype on the left and our recomposed object on the right, accompanied with the corresponding numbers found in the old list of S-C MS55 solutions.

```
['Prototype Squares' and Self-Complementary MS55 Reconstructed]
               1/P    S/1                          12/P    S/12
  1  2  3  4  5    1 23 20 12  9         1  2  3  5  4    1 23 19 12 10
  6  7  8  9 10   15  7  4 21 18         7  6  9 10  8   15  6  5 22 17
 11 12 13 14 15   24 16 13 10  2        11 12 13 14 15   24 18 13  8  2
 16 17 18 19 20    8  5 22 19 11        18 16 17 20 19    9  4 21 20 11
 21 22 23 24 25   17 14  6  3 25        22 21 23 24 25   16 14  7  3 25
              19/P    S/21                         38/P    S/40
  1  2  3  6  5    1 23 18 12 11         1  2  3  7  8    1 23  5 22 14
  8 19 16 11 22    9 19  6 21 10        21 20 17 14 16   11 20  7 18  9
 17 12 13 14  9   24  4 13 22  2        15 22 13  4 11   24 10 13 16  2
  4 15 10  7 18   16  5 20  7 17        10 12  9  6  5   17  8 19  6 15
 21 20 23 24 25   15 14  8  3 25        18 19 23 24 25   12  4 21  3 25
              41/P    S/45                         48/P    S/51
  1  2  3  8  4    1 23 12  9 20         1  2  3  9  5    1 23 11 22  8
 14  5 15 20 10   19  5  8 22 11        15 19 16  8 12    6 19  9 21 10
  7  9 13 17 19   24 16 13 10  2        20 22 13  4  6   24 14 13 12  2
 16  6 11 21 12   15  4 18 21  7        14 18 10  7 11   16  5 17  7 20
 22 18 23 24 25    6 17 14  3 25        21 17 23 24 25   18  4 15  3 25
              56/P    S/57                         63/P    S/67
  1  2  3 10  7    1 23 14 21  6         1  2  3 11  4    1 23  9 18 14
 12  8  9  6  4   11  8 10 19 17        17 20 21 14 16    7 20 11 22  5
 15 21 13  5 11   24 22 13  4  2        19 18 13  8  7   24 10 13 16  2
 22 20 17 18 14    9  7 16 18 15        10 12  5  6  9   21  4 15  6 19
 19 16 23 24 25   20  5 12  3 25        22 15 23 24 25   12  8 17  3 25
              70/P    S/77                         83/P    S/91
  1  2  3 12  4    1 23 16  5 20         1  2  3 14  5    1 23  8 11 22
 10  9 19 20 18   15  9 12 22  7        18  6 19 22 10   17  6 14 21  7
 11  5 13 21 15   24  8 13 18  2         9 11 13 15 17   24 16 13 10  2
  8  6  7 17 16   19  4 14 17 11        16  4  7 20  8   19  5 12 20  9
 22 14 23 24 25    6 21 10  3 25        21 12 23 24 25    4 15 18  3 25
```

96/P

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

S/100

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

103/P

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

S/103

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

110/P

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

S/113

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

118/P

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

S/121

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

125/P

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

S/126

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

128/P

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

S/130

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

147/P

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

S/147

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

154/P

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

S/160

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

165/P

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

S/176

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

412/P

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

S/412

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

524/P

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

S/526

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

660/P

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

S/660

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

770/P

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

S/772

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

868/P

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

S/868

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

933/P

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

S/933

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

988/P

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

S/989

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

1010/P

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

S/1013

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

1072/P

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

S/1073

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

1089/P					S/1089					1104/P					S/1104				
1	2	15	5	7	1	11	17	22	14	1	2	16	11	5	1	10	23	17	14
9	23	18	14	6	10	23	5	19	8	3	19	20	14	4	8	19	11	21	6
16	22	13	4	10	24	20	13	6	2	18	17	13	9	8	24	22	13	4	2
20	12	8	3	17	18	7	21	3	16	22	12	6	7	23	20	5	15	7	18
19	21	11	24	25	12	4	9	15	25	21	15	10	24	25	12	9	3	16	25

1115/P					S/1115					1118/P					S/1118				
1	2	17	16	11	1	9	18	23	14	1	2	18	17	12	1	8	21	20	15
8	22	21	14	6	7	22	16	15	5	5	23	22	15	10	7	23	17	14	4
19	23	13	3	7	24	20	13	6	2	19	20	13	6	7	24	16	13	10	2
20	12	5	4	18	21	11	10	4	19	16	11	4	3	21	22	12	9	3	19
15	10	9	24	25	12	3	8	17	25	14	9	8	24	25	11	6	5	18	25

1119/P					S/1119					1187/P					S/1187				
1	3	2	4	5	1	24	20	8	12	1	3	4	2	5	1	22	20	10	12
6	7	10	12	15	17	7	4	21	16	6	7	9	12	11	18	7	2	21	17
9	8	13	18	17	23	11	13	15	3	8	10	13	16	18	23	15	13	11	3
11	14	16	19	20	10	5	22	19	9	15	14	17	19	20	9	5	24	19	8
21	22	24	23	25	14	18	6	2	25	21	24	22	23	25	14	16	6	4	25

1274/P					S/1276					1371/P					S/1374				
1	3	5	2	4	1	21	20	15	8	1	3	6	2	4	1	20	8	21	15
6	10	9	8	7	14	10	2	22	17	18	17	16	15	7	14	17	2	22	10
12	15	13	11	14	23	19	13	7	3	12	21	13	5	14	23	19	13	7	3
19	18	17	16	20	9	4	24	16	12	19	11	10	9	8	16	4	24	9	12
22	24	21	23	25	18	11	6	5	25	22	24	20	23	25	11	5	18	6	25

1412/P					S/1412					1464/P					S/1465				
1	3	7	2	10	1	19	20	17	8	1	3	8	2	4	1	18	14	11	21
6	22	12	8	21	11	22	2	16	14	12	19	20	21	17	16	19	2	22	6
15	17	13	9	11	23	5	13	21	3	10	11	13	15	16	23	9	13	17	3
5	18	14	4	20	12	10	24	4	15	9	5	6	7	14	20	4	24	7	10
16	24	19	23	25	18	9	6	7	25	22	24	18	23	25	5	15	12	8	25

1498/P					S/1498					1538/P					S/1539				
1	3	9	2	4	1	17	20	19	8	1	3	10	2	4	1	16	18	11	19
6	16	12	8	5	11	16	2	22	14	8	21	20	19	17	14	21	2	22	6
15	19	13	7	11	23	21	13	5	3	12	11	13	15	14	23	9	13	17	3
21	18	14	10	20	12	4	24	10	15	9	7	6	5	18	20	4	24	5	12
22	24	17	23	25	18	7	6	9	25	22	24	16	23	25	7	15	8	10	25

1579/P					S/1579					1622/P					S/1623				
1	3	11	2	6	1	15	22	10	17	1	3	12	4	2	1	14	18	17	15
4	21	18	17	19	14	21	2	20	8	8	21	20	15	7	10	21	4	24	6
12	10	13	16	14	23	7	13	19	3	16	17	13	9	10	23	19	13	7	3
7	9	8	5	22	18	6	24	5	12	19	11	6	5	18	20	2	22	5	16
20	24	15	23	25	9	16	4	11	25	24	22	14	23	25	11	9	8	12	25

1631/P					S/1631					1637/P					S/1637				
1	3	14	4	5	1	12	18	19	15	1	3	15	4	8	1	11	24	20	9
8	24	20	15	9	10	24	4	21	6	2	21	14	9	7	10	21	4	18	12
16	19	13	7	10	23	17	13	9	3	16	20	13	6	10	23	19	13	7	3
17	11	6	2	18	20	5	22	2	16	19	17	12	5	24	14	8	22	5	16
21	22	12	23	25	11	7	8	14	25	18	22	11	23	25	17	6	2	15	25

1647/P					S/1648					1657/P					S/1657				
1	3	16	14	5	1	10	22	15	17	1	3	17	12	6	1	9	24	16	15
4	20	24	17	7	8	20	14	21	2	2	21	22	15	7	8	21	12	20	4
18	15	13	11	8	23	19	13	7	3	18	16	13	10	8	23	19	13	7	3
19	9	2	6	22	24	5	12	6	18	19	11	4	5	24	22	6	14	5	18
21	12	10	23	25	9	11	4	16	25	20	14	9	23	25	11	10	2	17	25

				1662/P	S/1662				
1	4	2	3	5	1	24	19	10	11
7	6	9	11	12	18	6	3	21	17
8	10	13	16	18	22	14	13	12	4
14	15	17	20	19	9	5	23	20	8
21	23	24	22	25	15	16	7	2	25

				2466/P	S/2466				
1	5	2	4	3	1	24	20	12	8
6	7	10	8	9	15	7	4	23	16
11	12	13	14	15	21	17	13	9	5
17	18	16	19	20	10	3	22	19	11
23	22	24	21	25	18	14	6	2	25

				2857/P	S/2860				
1	6	2	3	4	1	24	17	12	11
9	5	10	11	8	19	5	3	22	16
7	12	13	14	19	20	18	13	8	6
18	15	16	21	17	10	4	23	21	7
22	23	24	20	25	15	14	9	2	25

				3479/P	S/3479				
1	7	2	3	4	1	24	11	12	17
15	18	20	17	21	16	18	3	22	6
10	12	13	14	16	19	5	13	21	7
5	9	6	8	11	20	4	23	8	10
22	23	24	19	25	9	14	15	2	25

				3948/P	S/3948				
1	8	2	3	4	1	24	21	9	10
5	6	11	10	12	19	6	3	22	15
7	9	13	17	19	18	14	13	12	8
14	16	15	20	21	11	4	23	20	7
22	23	24	18	25	16	17	5	2	25

				4336/P	S/4337				
1	9	2	4	3	1	24	12	20	8
14	21	19	8	15	10	21	4	23	7
16	20	13	6	10	17	11	13	15	9
11	18	7	5	12	19	3	22	5	16
23	22	24	17	25	18	6	14	2	25

				4603/P	S/4603				
1	10	2	3	4	1	24	19	12	9
7	5	11	9	8	20	5	3	22	15
6	12	13	14	20	16	18	13	8	10
18	17	15	21	19	11	4	23	21	6
22	23	24	16	25	17	14	7	2	25

				5067/P	S/5067				
1	11	2	3	14	1	24	19	16	5
7	8	6	5	17	22	8	3	12	20
4	16	13	10	22	15	9	13	17	11
9	21	20	18	19	6	14	23	18	4
12	23	24	15	25	21	10	7	2	25

				5378/P	S/5381				
1	12	2	3	5	1	24	17	7	16
9	11	18	16	20	22	11	3	21	8
4	7	13	19	22	14	6	13	20	12
6	10	8	15	17	18	5	23	15	4
21	23	24	14	25	10	19	9	2	25

				5807/P	S/5808				
1	14	2	3	8	1	24	9	16	15
17	19	21	15	22	20	19	3	18	5
6	16	13	10	20	12	4	13	22	14
4	11	5	7	9	21	8	23	7	6
18	23	24	12	25	11	10	17	2	25

				6155/P	S/6155				
1	15	2	3	4	1	24	19	12	9
7	18	20	9	21	16	18	3	22	6
10	12	13	14	16	11	5	13	21	15
5	17	6	8	19	20	4	23	8	10
22	23	24	11	25	17	14	7	2	25

				6394/P	S/6396				
1	16	2	3	4	1	24	11	21	8
15	12	17	8	6	19	12	3	22	9
7	21	13	5	19	10	20	13	6	16
20	18	9	14	11	17	4	23	14	7
22	23	24	10	25	18	5	15	2	25

				6725/P	S/6725				
1	17	2	3	4	1	24	15	20	5
11	12	16	5	7	18	12	3	22	10
8	20	13	6	18	9	19	13	7	17
19	21	10	14	15	16	4	23	14	8
22	23	24	9	25	21	6	11	2	25

				6937/P	S/6938				
1	18	2	3	4	1	24	15	20	5
11	17	19	5	12	16	17	3	22	7
10	20	13	6	16	8	14	13	12	18
14	21	7	9	15	19	4	23	9	10
22	23	24	8	25	21	6	11	2	25

				7139/P	S/7139				
1	19	2	3	8	1	24	15	20	5
11	10	14	5	9	22	10	3	18	12
4	20	13	6	22	7	17	13	9	19
17	21	12	16	15	14	8	23	16	4
18	23	24	7	25	21	6	11	2	25

				7282/P	S/7283				
1	20	2	3	4	1	24	15	17	8
11	12	19	8	10	21	12	3	22	7
5	17	13	9	21	6	16	13	10	20
16	18	7	14	15	19	4	23	14	5
22	23	24	6	25	18	9	11	2	25

				7426/P	S/7426				
1	21	2	4	9	1	24	18	19	3
8	14	16	3	15	20	14	4	17	10
6	19	13	7	20	5	11	13	15	21
11	23	10	12	18	16	9	22	12	6
17	22	24	5	25	23	7	8	2	25

				7519/P	S/7521				
1	22	2	5	6	1	24	17	15	8
9	10	19	8	12	23	10	5	20	7
3	15	13	11	23	4	14	13	12	22
14	18	7	16	17	19	6	21	16	3
20	21	24	4	25	18	11	9	2	25

```
         7672/P   S/7672                  7756/P   S/7756
 1 23  2  6 11    1 24 16 19  5      1 24  3  8  9    1 23 20 16  5
10 14 18  5 17   22 14  6 15  8      6 11 19  5 14   22 11  8 17  7
 4 19 13  7 22    3  9 13 17 23      4 16 13 10 22    2 12 13 14 24
 9 21  8 12 16   18 11 20 12  4     12 21  7 15 20   19  9 18 15  4
15 20 24  3 25   21  7 10  2 25     17 18 23  2 25   21 10  6  3 25
         7793/P   S/7794                 14601/P   S/14601
 2  1  3  4  5    2 23 20 12  8      3  1  2  4  5    3 24 20 10  8
 6 16 11  8 19    9 16  4 21 15      6  9  7  8 15   12  9  4 21 19
17 12 13 14  9   25  7 13 19  1     14 10 13 16 12   25 11 13 15  1
 7 18 15 10 20   11  5 22 10 17     11 18 19 17 20    7  5 22 17 14
21 22 23 25 24   18 14  6  3 24     21 22 24 25 23   18 16  6  2 23
        21805/P   S/21805                27597/P   S/27597
 4  1  2  3  5    4 24 19 10  8      5  1  2  3  4    5 24 11 16  9
 7  9  6  8 15   12  9  3 21 20     15  8  6  9  7   12  8  3 22 20
14 10 13 16 12   25 11 13 15  1     14 16 13 10 12   25 19 13  7  1
11 18 20 17 19    6  5 23 17 14     19 17 20 18 11    6  4 23 18 14
21 23 24 25 22   18 16  7  2 22     22 23 24 25 21   17 10 15  2 21
        33193/P   S/33193                37313/P   S/37313
 6  1  2  4  3    6 24 18  5 12      7  1  2  3  4    7 24 17  6 11
 8 10  9 12 19   11 10  4 23 17      9 12  8 11 21   10 12  3 22 18
15  5 13 21 11   25  7 13 19  1     16  6 13 20 10   25  5 13 21  1
 7 14 17 16 18    9  3 22 16 15      5 15 18 14 17    8  4 23 14 16
23 22 24 25 20   14 21  8  2 20     22 23 24 25 19   15 20  9  2 19
        41081/P   S/41081                44129/P   S/44129
 8  1  2  3  4    8 24  5 12 16      9  1  2  6 15    9 24  4 12 16
21  6  7 16  9   15  6  3 22 19     22  7  3 16 21   18  7  6 11 23
11 12 13 14 15   25 17 13  9  1      8 12 13 14 18   25  5 13 21  1
17 10 19 20  5    7  4 23 20 11      5 10 23 19  4    3 15 20 19  8
22 23 24 25 18   10 14 21  2 18     11 20 24 25 17   10 14 22  2 17
        46533/P   S/46533                48065/P   S/48065
10  1  2  8  3   10 24  4 15 12     11  1  4  6  7   11 22 18  2 12
22  6  7 12  5    9  6  8 23 19      8  9  5 12 23   10  9  6 19 21
17 15 13 11  9   25 21 13  5  1     16  2 13 24 10   25  3 13 23  1
21 14 19 20  4    7  3 18 20 17      3 14 21 17 18    5  7 20 17 16
23 18 24 25 16   14 11 22  2 16     19 20 22 25 15   14 24  8  4 15
[Count = 48544/48544]  OK!

** Monitor List of Solution Correspondences between Old and New **
  ??: 0
   1: 1111111111111111111111111111111111111111111111111111111111111111
  65: 1111111111111111111111111111111111111111111111111111111111111111
 129: 1111111111111111111111111111111111111111111111111111111111111111
 193: 1111111111111111111111111111111111111111111111111111111111111111
 257: 1111111111111111111111111111111111111111111111111111111111111111
 321: 1111111111111111111111111111111111111111111111111111111111111111
 385: 1111111111111111111111111111111111111111111111111111111111111111
 449: 1111111111111111111111111111111111111111111111111111111111111111
 513: 1111111111111111111111111111111111111111111111111111111111111111
 577: 1111111111111111111111111111111111111111111111111111111111111111
 641: 1111111111111111111111111111111111111111111111111111111111111111
 705: 1111111111111111111111111111111111111111111111111111111111111111
 769: 1111111111111111111111111111111111111111111111111111111111111111
 833: 1111111111111111111111111111111111111111111111111111111111111111
 897: 1111111111111111111111111111111111111111111111111111111111111111
 961: 1111111111111111111111111111111111111111111111111111111111111111
```

```
 1025: 1111111111111111111111111111111111111111111111111111111111111111
 1089: 1111111111111111111111111111111111111111111111111111111111111111
 1153: 1111111111111111111111111111111111111111111111111111111111111111
 1217: 1111111111111111111111111111111111111111111111111111111111111111
 1281: 1111111111111111111111111111111111111111111111111111111111111111
 1345: 1111111111111111111111111111111111111111111111111111111111111111
 1409: 1111111111111111111111111111111111111111111111111111111111111111
 1473: 1111111111111111111111111111111111111111111111111111111111111111
 1537: 1111111111111111111111111111111111111111111111111111111111111111
 1601: 1111111111111111111111111111111111111111111111111111111111111111
 1665: 1111111111111111111111111111111111111111111111111111111111111111
 1729: 1111111111111111111111111111111111111111111111111111111111111111
 1793: 1111111111111111111111111111111111111111111111111111111111111111
 1857: 1111111111111111111111111111111111111111111111111111111111111111
 1921: 1111111111111111111111111111111111111111111111111111111111111111
 1985: 1111111111111111111111111111111111111111111111111111111111111111
 2049: 1111111111111111111111111111111111111111111111111111111111111111

 3969: 1111111111111111111111111111111111111111111111111111111111111111
 4033: 1111111111111111111111111111111111111111111111111111111111111111
 4097: 1111111111111111111111111111111111111111111111111111111111111111
 4161: 1111111111111111111111111111111111111111111111111111111111111111
 4225: 1111111111111111111111111111111111111111111111111111111111111111
 4289: 1111111111111111111111111111111111111111111111111111111111111111
 4353: 1111111111111111111111111111111111111111111111111111111111111111
 4417: 1111111111111111111111111111111111111111111111111111111111111111
 4481: 1111111111111111111111111111111111111111111111111111111111111111
 4545: 1111111111111111111111111111111111111111111111111111111111111111
 4609: 1111111111111111111111111111111111111111111111111111111111111111
 4673: 1111111111111111111111111111111111111111111111111111111111111111
 4737: 1111111111111111111111111111111111111111111111111111111111111111
 4801: 1111111111111111111111111111111111111111111111111111111111111111
 4865: 1111111111111111111111111111111111111111111111111111111111111111

24001: 1111111111111111111111111111111111111111111111111111111111111111
24065: 1111111111111111111111111111111111111111111111111111111111111111
24129: 1111111111111111111111111111111111111111111111111111111111111111
24193: 1111111111111111111111111111111111111111111111111111111111111111
24257: 1111111111111111111111111111111111111111111111111111111111111111
24321: 1111111111111111111111111111111111111111111111111111111111111111
24385: 1111111111111111111111111111111111111111111111111111111111111111
24449: 1111111111111111111111111111111111111111111111111111111111111111
24513: 1111111111111111111111111111111111111111111111111111111111111111
24577: 1111111111111111111111111111111111111111111111111111111111111111
24641: 1111111111111111111111111111111111111111111111111111111111111111

47873: 1111111111111111111111111111111111111111111111111111111111111111
47937: 1111111111111111111111111111111111111111111111111111111111111111
48001: 1111111111111111111111111111111111111111111111111111111111111111
48065: 1111111111111111111111111111111111111111111111111111111111111111
48129: 1111111111111111111111111111111111111111111111111111111111111111
48193: 1111111111111111111111111111111111111111111111111111111111111111
48257: 1111111111111111111111111111111111111111111111111111111111111111
48321: 1111111111111111111111111111111111111111111111111111111111111111
48385: 1111111111111111111111111111111111111111111111111111111111111111
48449: 1111111111111111111111111111111111111111111111111111111111111111
48513: 11111111111111111111111111111111

 OK!
```

I must report we could successfully make all the self-complementary MS55 without any fails and there exists the complete 'one-to-one correspondence' between the two solution sets of Prototype Squares and object solutions.

#5. Case of Pan-diagonal Magic Squares of Order 5

You may probably know there are 3600 pan-diagonal magic squares of order 5.

```
 ** Sample Solutions of Pan-Diagonal Magic Squares 5x5: **
              1/                2/                3/                4/
   1 25 13 19  7     1 25 13 19  7     1 25 14 18  7     1 25 14 18  7
  18  9  2 21 15    14 17  6  5 23    19  8  2 21 15    13 17  6  5 24
  22 11 20  8  4    10  3 24 12 16    22 11 20  9  3    10  4 23 12 16
  10  3 24 12 16    22 11 20  8  4    10  4 23 12 16    22 11 20  9  3
  14 17  6  5 23    18  9  2 21 15    13 17  6  5 24    19  8  2 21 15
              5/                6/                7/                8/
   1 25 18 14  7     1 25 18 14  7     1 25 19 13  7     1 25 19 13  7
  19 12  6  5 23    13  9  2 21 20    18 12  6  5 24    14  8  2 21 20
  10  3 24 17 11    22 16 15  8  4    10  4 23 17 11    22 16 15  9  3
  22 16 15  8  4    10  3 24 17 11    22 16 15  9  3    10  4 23 17 11
  13  9  2 21 20    19 12  6  5 23    14  8  2 21 20    18 12  6  5 24
  . . . . . . .
```

Let's try to recompose all those 3600 pan-diagonal magic squares only by 'DAM Transformation' and the solution set of Prototype Squares.

Since we want to use the same 'DAMT' and the same rules of transformations for this case, we must make the original set of Prototype Squares first of all.
What and how do we have to do, then?

We have to study the next two diagrams below and learn what conditions we can take for the Prototype Squares consulting with the Model Solution itself.

```
 [Basic Diagrams and Conditions for Prototype Squares 5x5]
    'Prototype Squares'               The Model Solution
 24 25 21 22 23 24 25 21 22       3 25 17 14  6  3 25 17 14
       .--------------.                 .--------------.
  4  5| 1| 2| 3| 4| 5| 1  2      12  9| 1|23|20|12| 9| 1 23
      |--+--+--+--+--|                |--+--+--+--+--|
  9 10| 6| 7| 8| 9|10| 6  7      21 18|15| 7| 4|21|18|15  7
      |--+--+--+--+--|                |--+--+--+--+--|
 14 15|11|12|13|14|15|11 12      10  2|24|16|13|10| 2|24 16
      |--+--+--+--+--|                |--+--+--+--+--|
 19 20|16|17|18|19|20|16 17      19 11| 8| 5|22|19|11| 8  5
      |--+--+--+--+--|                |--+--+--+--+--|
 24 25|21|22|23|24|25|21 22       3 25|17|14| 6| 3|25|17 14
       '--------------'                 '--------------'
  4  5  1  2  3  4  5  1  2      12  9  1 23 20 12  9  1 23
* Basic Conditions *
  n1+n23+n20+n12+n9=C    ... (1) | n1+n15+n24+n8+n17=C    ... (6)
  n15+n7+n4+n21+n18=C    ... (2) | n23+n7+n16+n5+n14=C    ... (7)
  n24+n16+n13+n10+n2=C   ... (3) | n20+n4+n13+n22+n6=C    ... (8)
  n8+n5+n22+n19+n11=C    ... (4) | n12+n21+n10+n19+n3=C   ... (9)
  n17+n14+n6+n3+n25=C    ... (5) | n9+n18+n2+n11+n25=C    ...(10)
* Pandiagonal Conditions: *
  n1+n7+n13+n19+n25=C    ...(11) | n1+n18+n10+n22+n14=C   ...(12)
  n23+n4+n10+n11+n17=C   ...(13) | n23+n15+n2+n19+n6=C    ...(14)
  n20+n21+n2+n8+n14=C    ...(15) | n20+n7+n24+n11+n3=C    ...(16)
  n12+n18+n24+n5+n6=C    ...(17) | n12+n4+n16+n8+n25=C    ...(18)
  n9+n15+n16+n22+n3=C    ...(19) | n9+n21+n13+n5+n17=C    ...(20)
* List-forming Inequality Conditions: *
  n1<n25; n1<n9; n1<n17; and n23>n15     ...(lf)
```

They are all the conditions we now need for making the Prototype Squares.
Self-complementary conditions are not necessary this time, since all pairs of 26 are not always located symmetrically in the pan-diagonal object, although the Model itself is of self-complementary type because it is selected from simultaneous type.

The conditions for Equal Cross-Sums below are not necessary now, but we know they will surely help us. You may well add these conditions, if necessary. They will never spoil our job, but surely make our program run faster.

```
n1+n7=n2+n6, n2+n8=n3+n7, n3+n9=n4+n8, n4+n10=n5+n9, n5+n6=n1+n10,
n6+n12=n7+n11, n7+n13=n8+n12, . . . , n19+n25=n20+n24, n20+n21=n16+n25,
and n1+n8=n3+n6, n2+n9=n4+n7, n3+n10=n5+n8, n4+n6=n1+n9, n5+n7=n2+n10,
n6+n13=n8+n11, n7+n14=n9+n12, n8+n15=n10+n13, n9+n11=n6+n14, . . . .
```

Let’s make all the Prototype Squares 5x5 now and apply our ‘DAMT’ to them one after another, and get every solution of pan-diagonal magic squares 5x5, shall we?
Take your kind look at the next result list of our recent experiment.

['**Prototype Squares**'(/P) and **Pan-diagonal MS55** Reconstructed(R/)]

```
                1/P   R/                          2/P   R/
  1  2  3  4  5   1 23 20 12  9      1  2  3  4  5   1 18 25 12  9
  6  7  8  9 10  15  7  4 21 18      6  7  8  9 10  15  7  4 16 23
 11 12 13 14 15  24 16 13 10  2     11 12 13 14 15  19 21 13 10  2
 16 17 18 19 20   8  5 22 19 11     21 22 23 24 25   8  5 17 24 11
 21 22 23 24 25  17 14  6  3 25     16 17 18 19 20  22 14  6  3 20
                3/P   R/                          4/P   R/
  1  2  3  4  5   1 23 15 17  9      1  2  3  4  5   1 23 20  7 14
  6  7  8  9 10  20  7  4 21 13     11 12 13 14 15  10 12  4 21 18
 16 17 18 19 20  24 11 18 10  2      6  7  8  9 10  24 16  8 15  2
 11 12 13 14 15   8  5 22 14 16     16 17 18 19 20  13  5 22 19  6
 21 22 23 24 25  12 19  6  3 25     21 22 23 24 25  17  9 11  3 25
                5/P   R/                          6/P   R/
  1  2  3  4  5   1 18 25  7 14      1  2  3  4  5   1 23 10 17 14
 11 12 13 14 15  10 12  4 16 23     11 12 13 14 15  20 12  4 21  8
  6  7  8  9 10  19 21  8 15  2     16 17 18 19 20  24  6 18 15  2
 21 22 23 24 25  13  5 17 24  6      6  7  8  9 10  13  5 22  9 16
 16 17 18 19 20  22  9 11  3 20     21 22 23 24 25   7 19 11  3 25
                7/P   R/                          8/P   R/
  1  2  3  4  5   1 23 15  7 19      1  2  3  4  5   1 13 25  7 19
 16 17 18 19 20  10 17  4 21 13     16 17 18 19 20  10 17  4 11 23
  6  7  8  9 10  24 11  8 20  2      6  7  8  9 10  14 21  8 20  2
 11 12 13 14 15  18  5 22 14  6     21 22 23 24 25  18  5 12 24  6
 21 22 23 24 25  12  9 16  3 25     11 12 13 14 15  22  9 16  3 15
                9/P   R/                         10/P   R/
  1  2  3  4  5   1 23 10 12 19      1  2  3  4  5   1 18 15  7 24
 16 17 18 19 20  15 17  4 21  8     21 22 23 24 25  10 22  4 16 13
 11 12 13 14 15  24  6 13 20  2      6  7  8  9 10  19 11  8 25  2
  6  7  8  9 10  18  5 22  9 11     11 12 13 14 15  23  5 17 14  6
 21 22 23 24 25   7 14 16  3 25     16 17 18 19 20  12  9 21  3 20
               11/P   R/                         12/P   R/
  1  2  3  4  5   1 13 20  7 24      1  2  3  4  5   1 18 10 12 24
 21 22 23 24 25  10 22  4 11 18     21 22 23 24 25  15 22  4 16  8
  6  7  8  9 10  14 16  8 25  2     11 12 13 14 15  19  6 13 25  2
 16 17 18 19 20  23  5 12 19  6      6  7  8  9 10  23  5 17  9 11
 11 12 13 14 15  17  9 21  3 15     16 17 18 19 20   7 14 21  3 20
```

				13/P	R/				
1	2	3	5	4	1	23	19	12	10
6	7	8	10	9	14	7	5	21	18
11	12	13	15	14	25	16	13	9	2
16	17	18	20	19	8	4	22	20	11
21	22	23	25	24	17	15	6	3	24

				16/P	R/				
1	2	3	5	4	1	23	19	7	15
11	12	13	15	14	9	12	5	21	18
6	7	8	10	9	25	16	8	14	2
16	17	18	20	19	13	4	22	20	6
21	22	23	25	24	17	10	11	3	24

				19/P	R/				
1	2	3	5	4	1	23	14	7	20
16	17	18	20	19	9	17	5	21	13
6	7	8	10	9	25	11	8	19	2
11	12	13	15	14	18	4	22	15	6
21	22	23	25	24	12	10	16	3	24

				22/P	R/				
1	2	3	5	4	1	18	14	7	25
21	22	23	25	24	9	22	5	16	13
6	7	8	10	9	20	11	8	24	2
11	12	13	15	14	23	4	17	15	6
16	17	18	20	19	12	10	21	3	19

				25/P	R/				
1	2	4	3	5	1	24	20	12	8
6	7	9	8	10	15	7	3	21	19
11	12	14	13	15	23	16	14	10	2
16	17	19	18	20	9	5	22	18	11
21	22	24	23	25	17	13	6	4	25

				28/P	R/				
1	2	4	3	5	1	24	20	7	13
11	12	14	13	15	10	12	3	21	19
6	7	9	8	10	23	16	9	15	2
16	17	19	18	20	14	5	22	18	6
21	22	24	23	25	17	8	11	4	25

				31/P	R/				
1	2	4	3	5	1	24	15	7	18
16	17	19	18	20	10	17	3	21	14
6	7	9	8	10	23	11	9	20	2
11	12	14	13	15	19	5	22	13	6
21	22	24	23	25	12	8	16	4	25

				34/P	R/				
1	2	4	3	5	1	19	15	7	23
21	22	24	23	25	10	22	3	16	14
6	7	9	8	10	18	11	9	25	2
11	12	14	13	15	24	5	17	13	6
16	17	19	18	20	12	8	21	4	20

				37/P	R/				
1	2	4	5	3	1	24	18	12	10
6	7	9	10	8	13	7	5	21	19
11	12	14	15	13	25	16	14	8	2
16	17	19	20	18	9	3	22	20	11
21	22	24	25	23	17	15	6	4	23

				40/P	R/				
1	2	4	5	3	1	24	18	7	15
11	12	14	15	13	8	12	5	21	19
6	7	9	10	8	25	16	9	13	2
16	17	19	20	18	14	3	22	20	6
21	22	24	25	23	17	10	11	4	23

				43/P	R/				
1	2	4	5	3	1	24	13	7	20
16	17	19	20	18	8	17	5	21	14
6	7	9	10	8	25	11	9	18	2
11	12	14	15	13	19	3	22	15	6
21	22	24	25	23	12	10	16	4	23

				46/P	R/				
1	2	4	5	3	1	19	13	7	25
21	22	24	25	23	8	22	5	16	14
6	7	9	10	8	20	11	9	23	2
11	12	14	15	13	24	3	17	15	6
16	17	19	20	18	12	10	21	4	18

				49/P	R/				
1	2	5	3	4	1	25	19	12	8
6	7	10	8	9	14	7	3	21	20
11	12	15	13	14	23	16	15	9	2
16	17	20	18	19	10	4	22	18	11
21	22	25	23	24	17	13	6	5	24

				73/P	R/				
1	3	2	4	5	1	22	20	13	9
6	8	7	9	10	15	8	4	21	17
11	13	12	14	15	24	16	12	10	3
16	18	17	19	20	7	5	23	19	11
21	23	22	24	25	18	14	6	2	25

				97/P	R/				
1	3	4	2	5	1	24	20	13	7
6	8	9	7	10	15	8	2	21	19
11	13	14	12	15	22	16	14	10	3
16	18	19	17	20	9	5	23	17	11
21	23	24	22	25	18	12	6	4	25

				121/P	R/				
1	3	5	2	4	1	25	19	13	7
6	8	10	7	9	14	8	2	21	20
11	13	15	12	14	22	16	15	9	3
16	18	20	17	19	10	4	23	17	11
21	23	25	22	24	18	12	6	5	24

				145/P	R/				
1	4	2	3	5	1	22	20	14	8
6	9	7	8	10	15	9	3	21	17
11	14	12	13	15	23	16	12	10	4
16	19	17	18	20	7	5	24	18	11
21	24	22	23	25	19	13	6	2	25

				217/P	R/				
1	5	2	3	4	1	22	19	15	8
6	10	7	8	9	14	10	3	21	17
11	15	12	13	14	23	16	12	9	5
16	20	17	18	19	7	4	25	18	11
21	25	22	23	24	20	13	6	2	24

```
           289/P    R/                      361/P    R/
 1  6 16 21 11    1 20 14  8 22     1 11 16 21  6    1 20  9 13 22
 2  7 17 22 12   13  7 21  5 19     2 12 17 22  7    8 12 21  5 19
 3  8 18 23 13   25  4 18 12  6     3 13 18 23  8   25  4 18  7 11
 4  9 19 24 14   17 11 10 24  3     4 14 19 24  9   17  6 15 24  3
 5 10 20 25 15    9 23  2 16 15     5 15 20 25 10   14 23  2 16 10
           433/P    R/                      505/P    R/
 1 16 11 21  6    1 15  9 18 22     1 21 11 16  6    1 15  9 23 17
 2 17 12 22  7    8 17 21  5 14     2 22 12 17  7    8 22 16  5 14
 3 18 13 23  8   25  4 13  7 16     3 23 13 18  8   20  4 13  7 21
 4 19 14 24  9   12  6 20 24  3     4 24 14 19  9   12  6 25 19  3
 5 20 15 25 10   19 23  2 11 10     5 25 15 20 10   24 18  2 11 10
           577/P    R/                     1153/P    R/
 2  1  3  4  5    2 23 20 11  9     3  1  2  4  5    3 22 20 11  9
 7  6  8  9 10   15  6  4 22 18     8  6  7  9 10   15  6  4 23 17
12 11 13 14 15   24 17 13 10  1    13 11 12 14 15   24 18 12 10  1
17 16 18 19 20    8  5 21 19 12    18 16 17 19 20    7  5 21 19 13
22 21 23 24 25   16 14  7  3 25    23 21 22 24 25   16 14  8  2 25
          1729/P    R/                     2305/P    R/
 4  1  2  3  5    4 22 20 11  8     5  1  2  3  4    5 22 19 11  8
 9  6  7  8 10   15  6  3 24 17    10  6  7  8  9   14  6  3 25 17
14 11 12 13 15   23 19 12 10  1    15 11 12 13 14   23 20 12  9  1
19 16 17 18 20    7  5 21 18 14    20 16 17 18 19    7  4 21 18 15
24 21 22 23 25   16 13  9  2 25    25 21 22 23 24   16 13 10  2 24
          2881/P    R/                     3025/P    R/
 6  7  8  9 10    6 23 20  2 14     7  6  8  9 10    7 23 20  1 14
11 12 13 14 15    5 12  9 21 18    12 11 13 14 15    5 11  9 22 18
 1  2  3  4  5   24 16  3 15  7     2  1  3  4  5   24 17  3 15  6
16 17 18 19 20   13 10 22 19  1    17 16 18 19 20   13 10 21 19  2
21 22 23 24 25   17  4 11  8 25    22 21 23 24 25   16  4 12  8 25
          3169/P    R/                     3313/P    R/
 8  6  7  9 10    8 22 20  1 14     9  6  7  8 10    9 22 20  1 13
13 11 12 14 15    5 11  9 23 17    14 11 12 13 15    5 11  8 24 17
 3  1  2  4  5   24 18  2 15  6     4  1  2  3  5   23 19  2 15  6
18 16 17 19 20   12 10 21 19  3    19 16 17 18 20   12 10 21 18  4
23 21 22 24 25   16  4 13  7 25    24 21 22 23 25   16  3 14  7 25
          3457/P    R/
10  6  7  8  9   10 22 19  1 13
15 11 12 13 14    4 11  8 25 17
 5  1  2  3  4   23 20  2 14  6
20 16 17 18 19   12  9 21 18  5
25 21 22 23 24   16  3 15  7 24
```

[Monitor List of Solution Correspondences between Old and New]

```
  ??: 0
   1: 111111111111111111111111111111111111111111111111111111111111
  61: 111111111111111111111111111111111111111111111111111111111111
 121: 111111111111111111111111111111111111111111111111111111111111
 181: 111111111111111111111111111111111111111111111111111111111111
 241: 111111111111111111111111111111111111111111111111111111111111
 301: 111111111111111111111111111111111111111111111111111111111111
 361: 111111111111111111111111111111111111111111111111111111111111
 421: 111111111111111111111111111111111111111111111111111111111111
 481: 111111111111111111111111111111111111111111111111111111111111
 541: 111111111111111111111111111111111111111111111111111111111111
```

```
 601: 111111111111111111111111111111111111111111111111111111111111
 661: 111111111111111111111111111111111111111111111111111111111111
 721: 111111111111111111111111111111111111111111111111111111111111
 781: 111111111111111111111111111111111111111111111111111111111111
 841: 111111111111111111111111111111111111111111111111111111111111
 901: 111111111111111111111111111111111111111111111111111111111111
 961: 111111111111111111111111111111111111111111111111111111111111
1021: 111111111111111111111111111111111111111111111111111111111111
1081: 111111111111111111111111111111111111111111111111111111111111
1141: 111111111111111111111111111111111111111111111111111111111111
1201: 111111111111111111111111111111111111111111111111111111111111
1261: 111111111111111111111111111111111111111111111111111111111111
1321: 111111111111111111111111111111111111111111111111111111111111
1381: 111111111111111111111111111111111111111111111111111111111111
1441: 111111111111111111111111111111111111111111111111111111111111
1501: 111111111111111111111111111111111111111111111111111111111111
1561: 111111111111111111111111111111111111111111111111111111111111
1621: 111111111111111111111111111111111111111111111111111111111111
1681: 111111111111111111111111111111111111111111111111111111111111
1741: 111111111111111111111111111111111111111111111111111111111111
1801: 111111111111111111111111111111111111111111111111111111111111
1861: 111111111111111111111111111111111111111111111111111111111111
1921: 111111111111111111111111111111111111111111111111111111111111
1981: 111111111111111111111111111111111111111111111111111111111111
2041: 111111111111111111111111111111111111111111111111111111111111
2101: 111111111111111111111111111111111111111111111111111111111111
2161: 111111111111111111111111111111111111111111111111111111111111
2221: 111111111111111111111111111111111111111111111111111111111111
2281: 111111111111111111111111111111111111111111111111111111111111
2341: 111111111111111111111111111111111111111111111111111111111111
2401: 111111111111111111111111111111111111111111111111111111111111
2461: 111111111111111111111111111111111111111111111111111111111111
2521: 111111111111111111111111111111111111111111111111111111111111
2581: 111111111111111111111111111111111111111111111111111111111111
2641: 111111111111111111111111111111111111111111111111111111111111
2701: 111111111111111111111111111111111111111111111111111111111111
2761: 111111111111111111111111111111111111111111111111111111111111
2821: 111111111111111111111111111111111111111111111111111111111111
2881: 111111111111111111111111111111111111111111111111111111111111
2941: 111111111111111111111111111111111111111111111111111111111111
3001: 111111111111111111111111111111111111111111111111111111111111
3061: 111111111111111111111111111111111111111111111111111111111111
3121: 111111111111111111111111111111111111111111111111111111111111
3181: 111111111111111111111111111111111111111111111111111111111111
3241: 111111111111111111111111111111111111111111111111111111111111
3301: 111111111111111111111111111111111111111111111111111111111111
3361: 111111111111111111111111111111111111111111111111111111111111
3421: 111111111111111111111111111111111111111111111111111111111111
3481: 111111111111111111111111111111111111111111111111111111111111
3541: 111111111111111111111111111111111111111111111111111111111111
[Count = 3600/3600]  OK!
```

How beautiful this monitor list is! It does not only indicate we could recompose the complete set of pan-diagonal solutions, but also demonstrate there certainly exists the one-to-one correspondence between the two sets of squares by ‘DAMT’.

#6. Case of Simultaneous MS55: Both Self-complementary and Pan-diagonal

The simultaneous magic squares of both self-complementary and pan-diagonal type are so rare that they are thought to be the most precious jewels we have ever had. All those of order 5 were once discovered and announced with the complete solution list by Prof. Mutsumi Suzuki in his famous homepage of magic squares.

His list contains 16 solutions.

** Standard Solutions of Simultaneous Magic Squares 5x5: **
 *** Both Self-complementary and Pan-diagonal ***

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

[Count = 16]

Let's try to recompose those 16 jewels by our DAMT and Prototype Squares now.
We want to use the same 'DAMT' and the same rules of transformations for this case, too. We must make another original set of Prototype Squares for them at first.
What and how do we have to do now?

[**Basic Diagrams and Conditions for Prototype Squares 5x5**]

```
    'Prototype Squares'           The Model Solution
 24 25 21 22 23 24 25 21 22     3 25 17 14  6  3 25 17 14
       .--------------.                .--------------.
  4  5|  1|  2|  3|  4|  5|  1  2    12  9|  1|23|20|12|  9|  1 23
       |--+--+--+--+--|                |--+--+--+--+--|
  9 10|  6|  7|  8|  9|10|  6  7    21 18|15|  7|  4|21|18|15  7
       |--+--+--+--+--|                |--+--+--+--+--|
 14 15|11|12|13|14|15|11 12    10  2|24|16|13|10|  2|24 16
       |--+--+--+--+--|                |--+--+--+--+--|
 19 20|16|17|18|19|20|16 17    19 11|  8|  5|22|19|11|  8  5
       |--+--+--+--+--|                |--+--+--+--+--|
 24 25|21|22|23|24|25|21 22     3 25|17|14|  6|  3|25|17 14
       '--------------'                '--------------'
  4  5  1  2  3  4  5  1  2    12  9  1 23 20 12  9  1 23
```

* Basic Conditions *

```
 n1+n9+n12+n20+n23=C    ... (1) | n1+n8+n15+n17+n24=C    ... (6)
 n4+n7+n15+n18+n21=C    ... (2) | n5+n7+n14+n16+n23=C    ... (7)
 n2+n10+n13+n16+n24=C   ... (3) | n4+n6+n13+n20+n22=C    ... (8)
 n5+n8+n11+n19+n22=C    ... (4) | n3+n10+n12+n19+n21=C   ... (9)
 n3+n6+n14+n17+n25=C    ... (5) | n2+n9+n11+n18+n25=C    ...(10)
```

* Pandiagonal Conditions: *

```
 n1+n7+n13+n19+n25=C    ...(11) | n1+n18+n10+n22+n14=C   ...(12)
```

```
  n23+n4+n10+n11+n17=C   ...(13) | n23+n15+n2+n19+n6=C    ...(14)
  n20+n21+n2+n8+n14=C    ...(15) | n20+n7+n24+n11+n3=C    ...(16)
  n12+n18+n24+n5+n6=C    ...(17) | n12+n4+n16+n8+n25=C    ...(18)
  n9+n15+n16+n22+n3=C    ...(19) | n9+n21+n13+n5+n17=C    ...(20)
* Self-Complementary Conditions: *
  n1+n25=n23+n3=n20+n6=n12+n14=n9+n17=n15+n11=n7+n19
  =n4+n22=n21+n5=n18+n8=n24+n2=n16+n10=n13+n13=CC        ...(21)
* List-forming Inequality Conditions: *
  n1<n25; n1<n9; n1<n17; and n23>n15;
```

We have to study the two diagrams above and learn from the Model Solution about what conditions we have to take for the new Prototype Squares.

1. Complementary pairs of 26 are all located symmetrically in the object Model. We have to keep the Self-complementary Conditions active in the Prototype, too.
   `n1+n25=n2+n24=n3+n23=....=n11+n15=n12+n14(=n13+n13)=26`

2. Every 5 entries on each pan-diagonal in the Prototype are re-used to make the object. Pan-diagonal Conditions have to be held strictly in the Prototype.

3. Every row and every column in the object add up to the same constant 65. What in the original make them magic? New Basic Conditions must be defined as above.

4. How do we have to deal with the equality conditions of Cross-Sums like these?
   `n1+n7=n2+n6, n2+n8=n3+n7, n3+n9=n4+n8, n4+n10=n5+n9, n5+n6=n1+n10,`
   `n6+n12=n7+n11, n7+n13=n8+n12, .... , n19+n25=n20+n24, n20+n21=n16+n25`
   `and n1+n8=n3+n6, n2+n9=n4+n7, n3+n10=n5+n8, n4+n6=n1+n9, n5+n7=n2+n10,`
   `n6+n13=n8+n11, n7+n14=n9+n12, n8+n15=n10+n13, n9+n11=n6+n14, ....`
   These conditions are not always necessary at first, but we know they will surely help us. They will never spoil us to fail, but make our program run faster.

5. List-forming Inequality Conditions are not always necessary. But without using any of these conditions, you have to have the new Prototype Squares as many as 128 (=16x8). You may well use them. But, you must not put {n1<n5; n1<n21; n1<n25 and n2>n6} in place of those conditions above. They should make some serious conflicts.
   The locations of {n5, n21} in the Prototype Squares are no longer meaningful in the object ones. You must have the new conditions {n1<n25; n1<n9; n1<n17 and n23>n15} for you to get all 16 standard solutions of Prototype Squares for our simultaneous magic squares 5x5: both self-complementary and pan-diagonal.

The next list shows all sixteen couples of the Prototype Squares and the recomposed object solutions accompanied with the decomposition diagrams by 5th increment number system.

```
** 'Prototype Squares' and Simultaneous MS55 Reconstructed **
  ** Both Self-complementary and Pan-diagonal: with /D5i **

           1/P   /d5i                          1/R   /d5i
  1  2  3  4  5   0 0 0 0 0   0 1 2 3 4    1 23 20 12  9   0 4 3 2 1   0 2 4 1 3
  6  7  8  9 10   1 1 1 1 1   0 1 2 3 4   15  7  4 21 18   2 1 0 4 3   4 1 3 0 2
 11 12 13 14 15   2 2 2 2 2   0 1 2 3 4   24 16 13 10  2   4 3 2 1 0   3 0 2 4 1
 16 17 18 19 20   3 3 3 3 3   0 1 2 3 4    8  5 22 19 11   1 0 4 3 2   2 4 1 3 0
 21 22 23 24 25   4 4 4 4 4   0 1 2 3 4   17 14  6  3 25   3 2 1 0 4   1 3 0 2 4
```

2/P	/d5i	
1 2 3 4 5	0 0 0 0 0	0 1 2 3 4
16 17 18 19 20	3 3 3 3 3	0 1 2 3 4
11 12 13 14 15	2 2 2 2 2	0 1 2 3 4
6 7 8 9 10	1 1 1 1 1	0 1 2 3 4
21 22 23 24 25	4 4 4 4 4	0 1 2 3 4

3/P	/d5i	
1 4 3 2 5	0 0 0 0 0	0 3 2 1 4
6 9 8 7 10	1 1 1 1 1	0 3 2 1 4
11 14 13 12 15	2 2 2 2 2	0 3 2 1 4
16 19 18 17 20	3 3 3 3 3	0 3 2 1 4
21 24 23 22 25	4 4 4 4 4	0 3 2 1 4

4/P	/d5i	
1 4 3 2 5	0 0 0 0 0	0 3 2 1 4
16 19 18 17 20	3 3 3 3 3	0 3 2 1 4
11 14 13 12 15	2 2 2 2 2	0 3 2 1 4
6 9 8 7 10	1 1 1 1 1	0 3 2 1 4
21 24 23 22 25	4 4 4 4 4	0 3 2 1 4

5/P	/d5i	
2 1 3 5 4	0 0 0 0 0	1 0 2 4 3
7 6 8 10 9	1 1 1 1 1	1 0 2 4 3
12 11 13 15 14	2 2 2 2 2	1 0 2 4 3
17 16 18 20 19	3 3 3 3 3	1 0 2 4 3
22 21 23 25 24	4 4 4 4 4	1 0 2 4 3

6/P	/d5i	
2 1 3 5 4	0 0 0 0 0	1 0 2 4 3
17 16 18 20 19	3 3 3 3 3	1 0 2 4 3
12 11 13 15 14	2 2 2 2 2	1 0 2 4 3
7 6 8 10 9	1 1 1 1 1	1 0 2 4 3
22 21 23 25 24	4 4 4 4 4	1 0 2 4 3

7/P	/d5i	
2 5 3 1 4	0 0 0 0 0	1 4 2 0 3
7 10 8 6 9	1 1 1 1 1	1 4 2 0 3
12 15 13 11 14	2 2 2 2 2	1 4 2 0 3
17 20 18 16 19	3 3 3 3 3	1 4 2 0 3
22 25 23 21 24	4 4 4 4 4	1 4 2 0 3

8/P	/d5i	
2 5 3 1 4	0 0 0 0 0	1 4 2 0 3
17 20 18 16 19	3 3 3 3 3	1 4 2 0 3
12 15 13 11 14	2 2 2 2 2	1 4 2 0 3
7 10 8 6 9	1 1 1 1 1	1 4 2 0 3
22 25 23 21 24	4 4 4 4 4	1 4 2 0 3

9/P	/d5i	
4 1 3 5 2	0 0 0 0 0	3 0 2 4 1
9 6 8 10 7	1 1 1 1 1	3 0 2 4 1
14 11 13 15 12	2 2 2 2 2	3 0 2 4 1
19 16 18 20 17	3 3 3 3 3	3 0 2 4 1
24 21 23 25 22	4 4 4 4 4	3 0 2 4 1

10/P	/d5i	
4 1 3 5 2	0 0 0 0 0	3 0 2 4 1
19 16 18 20 17	3 3 3 3 3	3 0 2 4 1
14 11 13 15 12	2 2 2 2 2	3 0 2 4 1
9 6 8 10 7	1 1 1 1 1	3 0 2 4 1
24 21 23 25 22	4 4 4 4 4	3 0 2 4 1

3/R	/d5i	
1 23 10 12 19	0 4 1 2 3	0 2 4 1 3
15 17 4 21 8	2 3 0 4 1	4 1 3 0 2
24 6 13 20 2	4 1 2 3 0	3 0 2 4 1
18 5 22 9 11	3 0 4 1 2	2 4 1 3 0
7 14 16 3 25	1 2 3 0 4	1 3 0 2 4

2/R	/d5i	
1 23 20 14 7	0 4 3 2 1	0 2 4 3 1
15 9 2 21 18	2 1 0 4 3	4 3 1 0 2
22 16 13 10 4	4 3 2 1 0	1 0 2 4 3
8 5 24 17 11	1 0 4 3 2	2 4 3 1 0
19 12 6 3 25	3 2 1 0 4	3 1 0 2 4

4/R	/d5i	
1 23 10 14 17	0 4 1 2 3	0 2 4 3 1
15 19 2 21 8	2 3 0 4 1	4 3 1 0 2
22 6 13 20 4	4 1 2 3 0	1 0 2 4 3
18 5 24 7 11	3 0 4 1 2	2 4 3 1 0
9 12 16 3 25	1 2 3 0 4	3 1 0 2 4

5/R	/d5i	
2 23 19 11 10	0 4 3 2 1	1 2 3 0 4
14 6 5 22 18	2 1 0 4 3	3 0 4 1 2
25 17 13 9 1	4 3 2 1 0	4 1 2 3 0
8 4 21 20 12	1 0 4 3 2	2 3 0 4 1
16 15 7 3 24	3 2 1 0 4	0 4 1 2 3

7/R	/d5i	
2 23 9 11 20	0 4 1 2 3	1 2 3 0 4
14 16 5 22 8	2 3 0 4 1	3 0 4 1 2
25 7 13 19 1	4 1 2 3 0	4 1 2 3 0
18 4 21 10 12	3 0 4 1 2	2 3 0 4 1
6 15 17 3 24	1 2 3 0 4	0 4 1 2 3

6/R	/d5i	
2 23 19 15 6	0 4 3 2 1	1 2 3 4 0
14 10 1 22 18	2 1 0 4 3	3 4 0 1 2
21 17 13 9 5	4 3 2 1 0	0 1 2 3 4
8 4 25 16 12	1 0 4 3 2	2 3 4 0 1
20 11 7 3 24	3 2 1 0 4	4 0 1 2 3

8/R	/d5i	
2 23 9 15 16	0 4 1 2 3	1 2 3 4 0
14 20 1 22 8	2 3 0 4 1	3 4 0 1 2
21 7 13 19 5	4 1 2 3 0	0 1 2 3 4
18 4 25 6 12	3 0 4 1 2	2 3 4 0 1
10 11 17 3 24	1 2 3 0 4	4 0 1 2 3

9/R	/d5i	
4 23 17 11 10	0 4 3 2 1	3 2 1 0 4
12 6 5 24 18	2 1 0 4 3	1 0 4 3 2
25 19 13 7 1	4 3 2 1 0	4 3 2 1 0
8 2 21 20 14	1 0 4 3 2	2 1 0 4 3
16 15 9 3 22	3 2 1 0 4	0 4 3 2 1

11/R	/d5i	
4 23 7 11 20	0 4 1 2 3	3 2 1 0 4
12 16 5 24 8	2 3 0 4 1	1 0 4 3 2
25 9 13 17 1	4 1 2 3 0	4 3 2 1 0
18 2 21 10 14	3 0 4 1 2	2 1 0 4 3
6 15 19 3 22	1 2 3 0 4	0 4 3 2 1

```
            11/P   /d5i                                  10/R   /d5i
  4  5  3  1  2   0 0 0 0 0  3 4 2 0 1    4 23 17 15  6   0 4 3 2 1  3 2 1 4 0
  9 10  8  6  7   1 1 1 1 1  3 4 2 0 1   12 10  1 24 18   2 1 0 4 3  1 4 0 3 2
 14 15 13 11 12   2 2 2 2 2  3 4 2 0 1   21 19 13  7  5   4 3 2 1 0  0 3 2 1 4
 19 20 18 16 17   3 3 3 3 3  3 4 2 0 1    8  2 25 16 14   1 0 4 3 2  2 1 4 0 3
 24 25 23 21 22   4 4 4 4 4  3 4 2 0 1   20 11  9  3 22   3 2 1 0 4  4 0 3 2 1
            12/P   /d5i                                  12/R   /d5i
  4  5  3  1  2   0 0 0 0 0  3 4 2 0 1    4 23  7 15 16   0 4 1 2 3  3 2 1 4 0
 19 20 18 16 17   3 3 3 3 3  3 4 2 0 1   12 20  1 24  8   2 3 0 4 1  1 4 0 3 2
 14 15 13 11 12   2 2 2 2 2  3 4 2 0 1   21  9 13 17  5   4 1 2 3 0  0 3 2 1 4
  9 10  8  6  7   1 1 1 1 1  3 4 2 0 1   18  2 25  6 14   3 0 4 1 2  2 1 4 0 3
 24 25 23 21 22   4 4 4 4 4  3 4 2 0 1   10 11 19  3 22   1 2 3 0 4  4 0 3 2 1
            13/P   /d5i                                  13/R   /d5i
  5  2  3  4  1   0 0 0 0 0  4 1 2 3 0    5 23 16 12  9   0 4 3 2 1  4 2 0 1 3
 10  7  8  9  6   1 1 1 1 1  4 1 2 3 0   11  7  4 25 18   2 1 0 4 3  0 1 3 4 2
 15 12 13 14 11   2 2 2 2 2  4 1 2 3 0   24 20 13  6  2   4 3 2 1 0  3 4 2 0 1
 20 17 18 19 16   3 3 3 3 3  4 1 2 3 0    8  1 22 19 15   1 0 4 3 2  2 0 1 3 4
 25 22 23 24 21   4 4 4 4 4  4 1 2 3 0   17 14 10  3 21   3 2 1 0 4  1 3 4 2 0
            14/P   /d5i                                  15/R   /d5i
  5  2  3  4  1   0 0 0 0 0  4 1 2 3 0    5 23  6 12 19   0 4 1 2 3  4 2 0 1 3
 20 17 18 19 16   3 3 3 3 3  4 1 2 3 0   11 17  4 25  8   2 3 0 4 1  0 1 3 4 2
 15 12 13 14 11   2 2 2 2 2  4 1 2 3 0   24 10 13 16  2   4 1 2 3 0  3 4 2 0 1
 10  7  8  9  6   1 1 1 1 1  4 1 2 3 0   18  1 22  9 15   3 0 4 1 2  2 0 1 3 4
 25 22 23 24 21   4 4 4 4 4  4 1 2 3 0    7 14 20  3 21   1 2 3 0 4  1 3 4 2 0
            15/P   /d5i                                  14/R   /d5i
  5  4  3  2  1   0 0 0 0 0  4 3 2 1 0    5 23 16 14  7   0 4 3 2 1  4 2 0 3 1
 10  9  8  7  6   1 1 1 1 1  4 3 2 1 0   11  9  2 25 18   2 1 0 4 3  0 3 1 4 2
 15 14 13 12 11   2 2 2 2 2  4 3 2 1 0   22 20 13  6  4   4 3 2 1 0  1 4 2 0 3
 20 19 18 17 16   3 3 3 3 3  4 3 2 1 0    8  1 24 17 15   1 0 4 3 2  2 0 3 1 4
 25 24 23 22 21   4 4 4 4 4  4 3 2 1 0   19 12 10  3 21   3 2 1 0 4  3 1 4 2 0
            16/P   /d5i                                  16/R   /d5i
  5  4  3  2  1   0 0 0 0 0  4 3 2 1 0    5 23  6 14 17   0 4 1 2 3  4 2 0 3 1
 20 19 18 17 16   3 3 3 3 3  4 3 2 1 0   11 19  2 25  8   2 3 0 4 1  0 3 1 4 2
 15 14 13 12 11   2 2 2 2 2  4 3 2 1 0   22 10 13 16  4   4 1 2 3 0  1 4 2 0 3
 10  9  8  7  6   1 1 1 1 1  4 3 2 1 0   18  1 24  7 15   3 0 4 1 2  2 0 3 1 4
 25 24 23 22 21   4 4 4 4 4  4 3 2 1 0    9 12 20  3 21   1 2 3 0 4  3 1 4 2 0
 [Count = 16/16]

** Monitor List of Solution Correspondences **
 ??:  0
  1:  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 OK!
```

#7. Comment

We could successfully make three types of solution sets of Prototype Squares of order 5. Each of them proves to be certainly one-to-one corresponding to each type of magic squares of order 5.

If you only have a single set of simple transformation rules and an appropriate set of Prototype Squares, you can always reconstruct the corresponding set of magic squares automatically, whenever you apply 'Do-it-After-the-Model Transformation'.

It is really amazing. But why could we do that? What makes it possible?

What do the Prototype Squares really mean?

I guess they are the basis of all object squares, and especially the regular array 5x5 has the master key to open all doors of every world, since we always have to make any type of magic squares on the basic array itself.

Isn't this basic array the Mother of all magic squares?

I feel we have to study more about this life-giving Mother herself.

Isn't it true that the solution set of Prototype Squares is just same with the set of Basic Forms of object squares, when I extend its concept to that which should contain all the conceptual diagrams of various basic transformations?

I am now thinking so, and getting more convinced of it.

You cannot deny that the top of the Prototype Squares is just the Basic Form of all, can you? Each of the next ones should indicate just how the Basic Form is transformed. Therefore, you can really make it into each object solution by our DAM Transformation.

One of the most wonderful things in our magic square world is the very Model and DAM Transformation, don't you think so?

A single Model Solution and the simple rules of transformation defined after that are really capable of making all the object squares provided with the solution set of Prototype Squares. In the case of order 5, the only common Model could make all the three types of magic squares 5x5: (1) Self-complementary, (2) Pan-diagonal and (3) Simultaneous type of both self-complementary and pan-diagonal.

It means that the single common Model selected is just the only one fundamental solution, the representative of all, doesn't it? How amazing it is!

(The original article was written on July 7, 2001; Revised on March 5, 2003, June 15, 2005; and October 17, 2006 by Kanji Setsuda working on MacOS Xcode 2.1)

#8. Additional Report

I introduced a new different way of composing Prototype Squares at that time of this revision. I did not use those old familiar definitions using Equality of Cross-sums, what I would have taken often before. This method was invented with my new idea that making Prototype Squares is just the inverse transformation of 'DAMT' and we could really make them by solving the 'inverse function'.

That is the reason why I want to take a new list of definitions like these below.

But to tell the truth, this new method is very slow for us to get all we want to have, even if we use our newest personal computer to count them up through. Though it is not impractical, it will really take a long time to do that job completely.

You might prefer to take those familiar conditions of Equal Cross-sums.

But the old method of composing Prototype Squares with those equal Cross-sums is not truly universal, since you cannot always make all with it. Though you may use it for the case of pan-diagonal magic squares of order 4 and 5, you may not use it for order 7, because you can only get the smaller set of special solutions of objects.

We really want to have the constant, universal method that can make the complete solution set of every object. 'Inverse function' of DAMT is the only answer I could get.

But why could we really use those conditions of Equal Cross-sums in the very case of order 5? We must make some logical proof for that possibility before all.

I tried to discover those conditions under the list of our first definitions below, and I have finally found them making such elaborate algebraic calculations as follows:

[**Basic Diagrams and Conditions for Prototype Squares 5x5**]

```
       /Prototype                    /Pan-diagonal Model
  24 25 21 22 23 24 25 21 22     3 25 17 14  6  3 25 17 14
        .--------------.               .--------------.
   4  5| 1| 2| 3| 4| 5| 1  2     12  9| 1|23|20|12| 9| 1 23
        |--+--+--+--+--|               |--+--+--+--+--|
   9 10| 6| 7| 8| 9|10| 6  7     21 18|15| 7| 4|21|18|15  7
        |--+--+--+--+--|               |--+--+--+--+--|
  14 15|11|12|13|14|15|11 12     10  2|24|16|13|10| 2|24 16
        |--+--+--+--+--|               |--+--+--+--+--|
  19 20|16|17|18|19|20|16 17     19 11| 8| 5|22|19|11| 8  5
        |--+--+--+--+--|               |--+--+--+--+--|
  24 25|21|22|23|24|25|21 22      3 25|17|14| 6| 3|25|17 14
        '--------------'               '--------------'
   4  5  1  2  3  4  5  1  2     12  9  1 23 20 12  9  1 23
```

* Basic Conditions *

```
n1+n9+n12+n20+n23=C    ... (1) | n1+n8+n15+n17+n24=C    ... (6)
n4+n7+n15+n18+n21=C    ... (2) | n5+n7+n14+n16+n23=C    ... (7)
n2+n10+n13+n16+n24=C   ... (3) | n4+n6+n13+n20+n22=C    ... (8)
n5+n8+n11+n19+n22=C    ... (4) | n3+n10+n12+n19+n21=C   ... (9)
n3+n6+n14+n17+n25=C    ... (5) | n2+n9+n11+n18+n25=C    ...(10)
```

* Pandiagonal Conditions: *

```
n1+n7+n13+n19+n25=C    ...(11) | n1+n18+n10+n22+n14=C   ...(12)
n4+n10+n11+n17+n23=C   ...(13) | n2+n6+n15+n19+n23=C    ...(14)
n2+n8+n14+n20+n21=C    ...(15) | n3+n7+n11+n20+n24=C    ...(16)
n5+n6+n12+n18+n24=C    ...(17) | n4+n8+n12+n16+n25=C    ...(18)
n3+n9+n15+n16+n22=C    ...(19) | n5+n9+n13+n17+n21=C    ...(20)
```

* List-forming Inequality Conditions: *

```
n1<n25; n1<n9; n1<n17; and n23>n15;
```

** **Algebraic Calculations** #1: **

```
n5+n8+n11+n19+n22=C    ... (4) ->  n5+n8+n11+n22=C-n19    ... (4)'
n3+n10+n12+n19+n21=C   ... (9) ->  n3+n10+n12+n21=C-n19   ... (9)'
n1+n7+n13+n19+n25=C    ...(11) ->  n1+n7+n13+n25=C-n19    ...(11)'
n2+n6 +n15+n19+n23=C   ...(14) ->  n2+n6+n15+n23=C-n19    ...(14)'
```

Therefore

```
n1+n7+n13+n25=n2+n6+n15+n23=n3+n10+n12+n21=n5+n8+n11+n22
  A   +   B   =  C  +   D   =  E   +   F   =  G  +   H        ...(21)

n1+n9+n12+n20+n23=C    ... (1) | n1+n8+n15+n17+n24=C    ... (6)
n4+n7+n15+n18+n21=C    ... (2) | n5+n7+n14+n16+n23=C    ... (7)
n1+n7+n13+n19+n25=C    ...(11) | n1+n10+n14+n18+n22=C   ...(12)
n3+n7+n11+n20+n24=C    ...(16)
---------------------------------------------------------------
(n1+n3+n4+n5+n7+n8+n9+n10+n11+n12+n13+n14+n15+n16+n17+n18+n19+n20+n21+n22+n23+n24
+n25)+n1+n1+n1+n7+n7+n7+n14+n15+n18+n20+n23+n24=7*C
```

Add (n2+n6) to both sides.

```
5*C+n1+n1+n1+n7+n7+n7+n14+n15+n18+n20+n23=7*C+n2+n6
n1+n1+n1+n7+n7+n7+n14+n15+n18+n20+n23+n24=2*C+n2+n6         ...(22)
```

In the same way

```
n2+n10+n13+n16+n24=C   ... (3) | n4+n6+n13+n20+n22=C    ... (8)
n3+n6+n14+n17+n25=C    ... (5) | n2+n9+n11+n18+n25=C    ...(10)
n2+n8+n14+n20+n21=C    ...(15) | n2+n6+n15+n19+n23=C    ...(14)
n5+n6+n12+n18+n24=C    ...(17)
---------------------------------------------------------------
(n2+n3+n4+n5+n6+n8+n9+n10+n11+n12+n13+n14+n15+n16+n17+n18+n19+n20+n21+n22+n23+n24
+n25)+n2+n2+n2+n6+n6+n6+n13+n14+n18+n20+n24+n25=7*C
```

Add (n1+n7) to both sides.

```
5*C+n2+n2+n2+n6+n6+n6+n13+n14+n18+n20+n24+n25=7*C+n1+n7
n2+n2+n2+n6+n6+n6+n13+n14+n18+n20+n24+n25=2*C+n1+n7         ...(23)
```

(22)+(23)

```
  n1+n1+n1+n7+n7+n7+n14+n15+n18+n20+n23+n24+2*C+n1+n7=2*C+n2+n6+n2+n2+n2+n6+n6+n6+n
13+n14+n18+n20+n24+n25
  n1+n1+n1+n1+n7+n7+n7+n7+n15+n23=n2+n2+n2+n2+n6+n6+n6+n6+n13+n25
```

Therefore

```
  4*(n1+n7)+n15+n23=4*(n2+n6)+n13+n25
      4A   +  D   =    4C   +   B                          ...(24)
```

(24)+(21)

```
   4A + D = 4C + B...(24)
    A + B =  C + D...(21)
  -------------------------
   5A     = 5C
```

Therefore A=C

In other words: n1+n7 = n2+n6

Return to (21): B=D

Therefore n13+n25=n15+n23

**** Algebraic Calculations #2: ****

```
  n1+n9+n12+n20+n23=C   ... (1) ->  n1+n9+n12+n23=C-n20   ... (1)'
  n4+n6+n13+n20+n22=C   ... (8) ->  n4+n6+n13+n22=C-n20   ... (8)'
  n2+n8+n14+n20+n21=C   ...(15) ->  n2+n8+n14+n21=C-n20   ...(15)'
  n3+n7+n11+n20+n24=C   ...(16) ->  n3+n7+n11+n24=C-n20   ...(16)'
```

Therefore

```
  n1+n9+n12+n23=n4+n6+n13+n22=n2+n8+n14+n21=n3+n7+n11+n24
    A  +   B    =  C  +   D    =  E   +  F    =  G  +   H       ...(21)'
```

```
  n1+n8+n15+n17+n24=C   ... (6) | n2+n10+n13+n16+n24=C  ... (3)
  n5+n8+n11+n19+n22=C   ... (4) | n2+n9+n11+n18+n25=C   ...(10)
  n2+n6+n15+n19+n23=C   ...(14) | n2+n8+n14+n20+n21=C   ...(15)
  n4+n8+n12+n16+n25=C   ...(18)
---------------------------------------------------------------
  (n1+n2+n4+n5+n6+n8+n9+n10+n11+n12+n13+n14+n15+n16+n17+n18+n19+n20+n21+n22+n23+n24
+n25)+n2+n2+n2+n8+n8+n8+n11+n15+n16+n19+n24+n25=7*C
```

Add (n3+n7) to both sides.

```
  5*C+n2+n2+n2+n8+n8+n8+n11+n15+n16+n19+n24+n25=7*C+n3+n7
  n2+n2+n2+n8+n8+n8+n11+n15+n16+n19+n24+n25=2*C+n3+n7        ...(22)'
```

In the same way

```
  n4+n7+n15+n18+n21=C   ... (2) | n5+n7+n14+n16+n23=C   ... (7)
  n3+n6+n14+n17+n25=C   ... (5) | n3+n10+n12+n19+n21=C  ... (9)
  n1+n7+n13+n19+n25=C   ...(11) | n3+n7+n11+n20+n24=C   ...(16)
  n3+n9+n15+n16+n22=C   ...(19)
---------------------------------------------------------------
  (n1+n3+n4+n5+n6+n7+n9+n10+n11+n12+n13+n14+n15+n16+n17+n18+n19+n20+n21+n22+n23+n24
+n25)+n3+n3+n3+n7+n7+n7+n14+n15+n16+n19+n21+n25=7*C
```

Add (n2+n8) to both sides.

```
  5*C+n3+n3+n3+n7+n7+n7+n14+n15+n16+n19+n21+n25=7*C+n2+n8
  n3+n3+n3+n7+n7+n7+n14+n15+n16+n19+n21+n25=2*C+n2+n8        ...(23)'
```

(22)'+(23)'

```
  n2+n2+n2+n8+n8+n8+n11+n15+n16+n19+n24+n25+2*C+n2+n8=2*C+n3+n7+n3+n3+n3+n7+n7+n7+n
14+n15+n16+n19+n21+n25
  n2+n2+n2+n2+n8+n8+n8+n8+n11+n24=n3+n3+n3+n3+n7+n7+n7+n7+n14+n21
```

Therefore

```
  4*(n2+n8)+n11+n24=4*(n3+n7)+n14+n21
      4E   +  H   =    4G   +   F                          ...(24)'
```

Compare it with (21)'

```
  n1+n9+n12+n23=n4+n6+n13+n22=n2+n8+n14+n21=n3+n7+n11+n24
    A  +   B    =  C  +   D    =  E   +  F    =  G  +   H       ...(21)'
```

(24)'+(21)'

```
   4E + H = 4G + F...(24)
```

```
    E + F =  G + H...(21)
  -------------------------
   5E     = 5G
Therefore E=G
In other words: n2+n8 = n3+n7
Return to (21): F=H
Therefore       n14+n21=n11+n24
```

Repeating the same kind of calculations, you can get such the equations as follows:

```
n1+n7=n2+n6; n2+n8=n3+n7; n3+n9=n4+n8; n4+n10+n5+n9;  n5+n6=n1+n10;
n6+n12=n7+n11; n7+n13=n8+n12; n8+n14=n9+n13; n9+n15=n10+n14;
n10+n11=n6+n15;  n11+n17=n12+n16; n12+n18=n13+n17; n13+n19=n14+n18;
n14+n20=n15+n19; n15+n16=n11+n20; n16+n22=n17+n21; n17+n23=n18+n22;
n18+n24=n19+n23; n19+n25=n20+n24; n20+n21=n16+n25;  n21+n2=n22+n1;
n22+n3=n23+n2; n23+n4=n24+n3; n24+n5=n25+n4; n25+n1=n21+n5;
```

Do you notice you may call them as “The Equality of Cross-Sums”?

You can always expect this property true and define it at the first step, whenever you should make the Prototype Squares for any Pan-diagonal magic squares 5x5.

The next program list demonstrates how to use our newest, simplest definitions.

```
/** 'Prototype Squares' and Simultaneous MS55: **/
/**  Both Self-complementary and Pan-diagonal  **/
/** 'ProtoT5Sml.c' built by Kanji Setsuda **/
/**  on Jun.14, 2005; Sep.27, 2006 **/
/**  Working on MacOSX & Xcode2.2  **/
/**/
/* Using Library */
#include <stdio.h>
/**/
/* Global Variables */
short int cnt, cnt1, cnt2;
short cnt3;
short LSM, PSM, N3C;
short nm[26], uflg[26];
short dnm[26];
short ln[17];
short an[5][26];
short sn[17][26], pn[17][26];
/**/
/* Sub-Routines 1: */
void deep01(void), deep02(void), deep03(void), deep04(void);
void deep05(void), deep06(void), deep07(void), deep08(void);
void deep09(void), deep10(void), deep11(void), deep12(void);
void deep13(void), deep14(void);
void ansrecord(void);
void prans(short x);
/**/
/* Sub-Routines 2: */
void stp01(void), stp02(void), stp03(void), stp04(void);
void stp05(void), stp06(void), stp07(void), stp08(void);
void stp09(void), stp10(void), stp11(void), stp12(void);
void stp13(void), stp14(void);
void protorecord(void);
void sortp(void);
void exc(short x);
void printall(short x, short y), pr2ans(short x);
void pransdcmp(short x);
void damt(void);
short findoldn(void);
void prmon(void);
```

```
/**/
/* Main Program */
int main(){
 short m;
 short n;
  printf("\n** Standard Solutions of Simultaneous Magic Squares 5x5: **\n");
  printf(" ***  Both Self-complementary and Pan-diagonal  ***\n");
  for(n=0;n<26;n++){nm[n]=0; uflg[n]=0;}
  LSM=65; PSM=26; N3C=39; cnt=0; cnt3=0;
  nm[13]=13; uflg[13]=1;
  deep01();
  if(cnt3>0){prans(cnt3);}
  printf(" [Count = %d]\n",cnt);
  printf("\n** 'Prototype Squares' and Simultaneous MS55 Reconstructed **\n");
  printf("** Both Self-complementary and Pan-diagonal: List of Pairs **\n");
  for(n=0;n<26;n++){nm[n]=0; uflg[n]=0;}
  for(m=0;m<cnt;m++){ln[m]=0;}
  LSM=65; PSM=26; cnt1=0;
  nm[13]=13; uflg[13]=1;
  stp01();
  sortp();
  printall(1,cnt1);
  if(cnt3>0){pr2ans(cnt3);}
  printf(" [Count = %d/%d]\n",cnt1,cnt);
  printf("\n** Monitor List of Solution Correspondences **\n");
  prmon();
  printf(" OK!\n");
 return 0;
}
/**/
/**  Basic Form for PMS55  **
            .--.--.--.--.--.
  2  3  4  5| 1| 2| 3| 4| 5| 1  2  3  4
            |--+--+--+--+--|
  7  8  9 10| 6| 7| 8| 9|10| 6  7  8  9
            |--+--+--+--+--|
 12 13 14 15|11|12|13|14|15|11 12 13 14
            |--+--+--+--+--|
 17 18 19 20|16|17|18|19|20|16 17 18 19
            |--+--+--+--+--|
 22 23 24 25|21|22|23|24|25|21 22 23 24
            '--'--'--'--'--'
*/
/* Define Level 1: */
/* Set N1 & n25 & n1<n25 */
void deep01(){
 short a,b;
  for(a=1;a<13;a++){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[1]=a; nm[25]=b;
      uflg[a]=1; uflg[b]=1;
      deep02();
      uflg[b]=0; uflg[a]=0;}
  }
}
/* Set N2 & n24 */
void deep02(){
 short a,b;
  for(a=25;a>0;a--){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[2]=a; nm[24]=b;
      uflg[a]=1; uflg[b]=1;
      deep03();
      uflg[b]=0; uflg[a]=0;}
  }
```

```
}
/* Set n6=N3C-n1-n2 & n6<n2 & n20 */
void deep03(){
 short a,b;
  a=N3C-nm[1]-nm[2];
  if((0<a)&&(a<nm[2])){
    b=N3C-nm[25]-nm[24];
    if(a+b==PSM){
     if((uflg[a]==0)&&(uflg[b]==0)){
      nm[6]=a; nm[20]=b;
      uflg[a]=1; uflg[b]=1;
      deep04();
      uflg[b]=0; uflg[a]=0;}}}
}
/* Set N3 & n23 */
void deep04(){
 short a,b;
  for(a=25;a>0;a--){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[3]=a; nm[23]=b;
      uflg[a]=1; uflg[b]=1;
      deep05();
      uflg[b]=0; uflg[a]=0;}
  }
}
/* Set N4 & n22 */
void deep05(){
 short a,b;
  for(a=1;a<26;a++){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[4]=a; nm[22]=b;
      uflg[a]=1; uflg[b]=1;
      deep06();
      uflg[b]=0; uflg[a]=0;}
  }
}
/* Set n5=LSM-n1-n2-n3-n4 & n1<n5 & n21(>n1) */
void deep06(){
 short a,b;
  a=LSM-nm[1]-nm[2]-nm[3]-nm[4];
  if((nm[1]<a)&&(a<26)){b=PSM-a;
    b=LSM-nm[22]-nm[23]-nm[24]-nm[25];
    if((a+b==PSM)&&(b>nm[1])){
     if((uflg[a]==0)&&(uflg[b]==0)){
      nm[5]=a; nm[21]=b;
      uflg[a]=1; uflg[b]=1;
      deep07();
      uflg[b]=0; uflg[a]=0;}}}
}
/* Set n10=N3C-n4-n5 & n16 */
void deep07(){
 short a,b;
  a=N3C-nm[4]-nm[5];
  if((0<a)&&(a<26)){
    b=N3C-nm[22]-nm[21];
    if(a+b==PSM){
     if((uflg[a]==0)&&(uflg[b]==0)){
      nm[10]=a; nm[16]=b;
      uflg[a]=1; uflg[b]=1;
      deep08();
      uflg[b]=0; uflg[a]=0;}}}
}
/* Set n11=LSM-n1-n6-n16-n21 & n15 */
void deep08(){
```

```
 short a,b;
  a=LSM-nm[1]-nm[6]-nm[16]-nm[21];
  if((0<a)&&(a<26)){b=PSM-a;
    b=LSM-nm[5]-nm[10]-nm[20]-nm[25];
    if(a+b==PSM){
     if((uflg[a]==0)&&(uflg[b]==0)){
      nm[11]=a; nm[15]=b;
      uflg[a]=1; uflg[b]=1;
      deep09();
      uflg[b]=0; uflg[a]=0;}}}
}
/* Set n7=LSM-n3-n11-n20-n24 & n19 */
void deep09(){
 short a,b;
  a=LSM-nm[3]-nm[11]-nm[20]-nm[24];
  if((0<a)&&(a<26)){b=PSM-a;
    b=LSM-nm[2]-nm[6]-nm[15]-nm[23];
    if(a+b==PSM){
     if((uflg[a]==0)&&(uflg[b]==0)){
      nm[7]=a; nm[19]=b;
      uflg[a]=1; uflg[b]=1;
      deep10();
      uflg[b]=0; uflg[a]=0;}}}
}
/* Set n9=LSM-n3-n15-n16-n22 & n17 */
void deep10(){
 short a,b;
  a=LSM-nm[3]-nm[15]-nm[16]-nm[22];
  if((0<a)&&(a<26)){b=PSM-a;
    b=LSM-nm[4]-nm[10]-nm[11]-nm[23];
    if(a+b==PSM){
     if((uflg[a]==0)&&(uflg[b]==0)){
      nm[9]=a; nm[17]=b;
      uflg[a]=1; uflg[b]=1;
      deep11();
      uflg[b]=0; uflg[a]=0;}}}
}
/* Set n8=LSM-n6-n7-n9-n10 & n18 */
void deep11(){
 short a,b;
  a=LSM-nm[6]-nm[7]-nm[9]-nm[10];
  if((0<a)&&(a<26)){b=PSM-a;
    b=LSM-nm[20]-nm[19]-nm[17]-nm[16];
    if(a+b==PSM){
     if((uflg[a]==0)&&(uflg[b]==0)){
      nm[8]=a; nm[18]=b;
      uflg[a]=1; uflg[b]=1;
      deep12();
      uflg[b]=0; uflg[a]=0;}}}
}
/* Set n12=LSM-n5-n6-n18-n24 & n14 */
void deep12(){
 short a,b,c,d;
  a=LSM-nm[5]-nm[6]-nm[18]-nm[24];
  if((0<a)&&(a<26)){b=PSM-a;
    b=LSM-nm[2]-nm[8]-nm[20]-nm[21];
    if(a+b==PSM){
      c=LSM-nm[4]-nm[8]-nm[16]-nm[25];
      d=LSM-nm[2]-nm[7]-nm[17]-nm[22];
      if((a==c)&&(a==d)){
       if((uflg[a]==0)&&(uflg[b]==0)){
        nm[12]=a; nm[14]=b;
        uflg[a]=1; uflg[b]=1;
        deep13();
```

```
        uflg[b]=0; uflg[a]=0;}}}}
}
/* Checks Sums of Some 4 Entries */
void deep13(void){
 short sm1, sm2;
  sm1=nm[4]+nm[9]+nm[14]+nm[19]+nm[24];
  sm2=nm[1]+nm[10]+nm[14]+nm[18]+nm[22];
  if((sm1==LSM)&&(sm2==LSM)){deep14();}
}
/* Checks Sums of Some 4 Entries */
void deep14(void){
 short sm1, sm2, sm3, sm4;
  sm1=nm[1]+nm[7]+nm[13]+nm[19]+nm[25];
  sm2=nm[3]+nm[8]+nm[13]+nm[18]+nm[23];
  sm3=nm[5]+nm[9]+nm[13]+nm[17]+nm[21];
  sm4=nm[11]+nm[12]+nm[13]+nm[14]+nm[15];
  if((sm1==LSM)&&(sm2==LSM)&&(sm3==LSM)&&(sm4==LSM)){ansrecord();}
}
/**/
/* Record the Answers */
void ansrecord(){
 short n;
  for(n=1;n<26;n++){sn[cnt][n]=nm[n];}
  cnt++; cnt2++;
  an[cnt3][0]=cnt;
  for(n=1;n<26;n++){an[cnt3][n]=nm[n];}
  cnt3++; if(cnt3==4){prans(cnt3); cnt3=0;}
}
/**/
/* Print the Answers */
void prans(short x){
 short l,l5,m,n;
  for(n=0;n<x;n++){printf("%15d/ ",an[n][0]);}
  printf("\n");
  for(l=0;l<5;l++){l5=l*5;
    for(m=0;m<x;m++){
      printf(" %3d%3d%3d%3d%3d ",
        an[m][l5+1],an[m][l5+2],an[m][l5+3],an[m][l5+4],an[m][l5+5]);
    }
    printf("\n");
  }
}
/**/
/* Make the Prototype Squares and Apply 'DAMT' */
/** Basic Form for Prototype Squares 5x5 **
  .--.--.--.--.--.               .--.--.--.--.--.
  | 1| 2| 3| 4| 5|    23 20 12  9| 1|23|20|12| 9| 1 23 20 12
  |--+--+--+--+--|               |--+--+--+--+--|
  | 6| 7| 8| 9|10|     7  4 21 18|15| 7| 4|21|18|15  7  4 21
  |--+--+--+--+--|               |--+--+--+--+--|
  |11|12|13|14|15|    16 13 10  2|24|16|13|10| 2|24 16 13 10
  |--+--+--+--+--|               |--+--+--+--+--|
  |16|17|18|19|20|     5 22 19 11| 8| 5|22|19|11| 8  5 22 19
  |--+--+--+--+--|               |--+--+--+--+--|
  |21|22|23|24|25|    14  6  3 25|17|14| 6| 3|25|17 14  6  3
  '--'--'--'--'--'               '--'--'--'--'--'
** Basic Conditions for Prototype Squares 5x5 **
  n1+n23+n20+n12+n9=C    ...rw1;  | n1+n15+n24+n8+n17=C    ...cl1;
  n15+n7+n4+n21+n18=C    ...rw2;  | n23+n7+n16+n5+n14=C    ...cl2;
  n24+n16+n13+n10+n2=C   ...rw3;  | n20+n4+n13+n22+n6=C    ...cl3;
  n8+n5+n22+n19+n11=C    ...rw4;  | n12+n21+n10+n19+n3=C   ...cl4;
  n17+n14+n6+n3+n25=C    ...rw5;  | n9+n18+n2+n11+n25=C    ...cl5;
** Pan-diagonal Conditions for Prototype Squares 5x5 **
  n1+n7+n13+n19+n25=C    ...pd1;  | n1+n18+n10+n22+n14=C   ...pb1;
  n23+n4+n10+n11+n17=C   ...pd2;  | n23+n15+n2+n19+n6=C    ...pb2;
```

```
  n20+n21+n2+n8+n14=C    ...pd3;  | n20+n7+n24+n11+n3=C    ...pb3;
  n12+n18+n24+n5+n6=C    ...pd4;  | n12+n4+n16+n8+n25=C    ...pb4;
  n9+n15+n16+n22+n3=C    ...pd5;  | n9+n21+n13+n5+n17=C    ...pb5;
** Self-complementary Conditions for Prototype Squares 5x5 **
  n1+n25=n23+n3=n20+n6=n12+n14=n9+n17=n15+n11=
  n7+n19=n4+n22=n21+n5=n18+n8=n24+n2=n16+n10=n13+n13=
  n10+n16=n2+n24=n8+n18=n5+n21=n22+n4=n19+n7=n11+n15=
  n17+n9=n14+n12=n6+n20=n3+n23=n25+n1=SC ...scc
** List-forming Inequality Conditions **
  n1<n9, n1<n17, n1<n25 and n15<n23
*/
/* Define Level 1: */
/* Set N1 & n25 & n1<n25 */
void stp01(){
 short a,b;
  for(a=1;a<13;a++){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[1]=a; nm[25]=b;
      uflg[a]=1; uflg[b]=1;
      stp02();
      uflg[b]=0; uflg[a]=0;}
  }
}
/* Set N23 & n3 */
void stp02(){
 short a,b;
  for(a=25;a>0;a--){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[23]=a; nm[3]=b;
      uflg[a]=1; uflg[b]=1;
      stp03();
      uflg[b]=0; uflg[a]=0;}
  }
}
/* Set n15=N3C-n1-n23 & n15<n23 & n11 */
void stp03(){
 short a,b;
  a=N3C-nm[1]-nm[23];
  if((0<a)&&(a<nm[23])){
    b=N3C-nm[25]-nm[3];
    if(a+b==PSM){
     if((uflg[a]==0)&&(uflg[b]==0)){
      nm[15]=a; nm[11]=b;
      uflg[a]=1; uflg[b]=1;
      stp04();
      uflg[b]=0; uflg[a]=0;}}}
}
/* Set N20 & n6 */
void stp04(){
 short a,b;
  for(a=25;a>0;a--){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[20]=a; nm[6]=b;
      uflg[a]=1; uflg[b]=1;
      stp05();
      uflg[b]=0; uflg[a]=0;}
  }
}
/* Set N12 & n14 */
void stp05(){
 short a,b;
  for(a=1;a<26;a++){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[12]=a; nm[14]=b;
```

```
        uflg[a]=1; uflg[b]=1;
        stp06();
        uflg[b]=0; uflg[a]=0;}
   }
}
/* Set n9=LSM-n1-n23-n20-n12 & n1<n9 & n17(>n1) */
void stp06(){
 short a,b;
   a=LSM-nm[1]-nm[23]-nm[20]-nm[12];
   if((nm[1]<a)&&(a<26)){b=PSM-a;
     b=LSM-nm[25]-nm[3]-nm[6]-nm[14];
     if((a+b==PSM)&&(b>nm[1])){
      if((uflg[a]==0)&&(uflg[b]==0)){
        nm[9]=a; nm[17]=b;
        uflg[a]=1; uflg[b]=1;
        stp07();
        uflg[b]=0; uflg[a]=0;}}}
}
/* Set n18=N3C-n12-n9 & n8 */
void stp07(){
 short a,b;
   a=N3C-nm[12]-nm[9];
   if((0<a)&&(a<26)){
     b=N3C-nm[14]-nm[17];
     if(a+b==PSM){
      if((uflg[a]==0)&&(uflg[b]==0)){
        nm[18]=a; nm[8]=b;
        uflg[a]=1; uflg[b]=1;
        stp08();
        uflg[b]=0; uflg[a]=0;}}}
}
/* Set n24=LSM-n1-n15-n8-n17 & n2 */
void stp08(){
 short a,b;
   a=LSM-nm[1]-nm[15]-nm[8]-nm[17];
   if((0<a)&&(a<26)){b=PSM-a;
     b=LSM-nm[25]-nm[11]-nm[18]-nm[9];
     if(a+b==PSM){
      if((uflg[a]==0)&&(uflg[b]==0)){
        nm[24]=a; nm[2]=b;
        uflg[a]=1; uflg[b]=1;
        stp09();
        uflg[b]=0; uflg[a]=0;}}}
}
/* Set n7=LSM-n20-n24-n11-n3 & n19 */
void stp09(){
 short a,b;
   a=LSM-nm[20]-nm[24]-nm[11]-nm[3];
   if((0<a)&&(a<26)){b=PSM-a;
     b=LSM-nm[6]-nm[2]-nm[15]-nm[23];
     if(a+b==PSM){
      if((uflg[a]==0)&&(uflg[b]==0)){
        nm[7]=a; nm[19]=b;
        uflg[a]=1; uflg[b]=1;
        stp10();
        uflg[b]=0; uflg[a]=0;}}}
}
/* Set n21=LSM-n20-n2-n8-n14 & n5 */
void stp10(){
 short a,b;
   a=LSM-nm[20]-nm[2]-nm[8]-nm[14];
   if((0<a)&&(a<26)){b=PSM-a;
     b=LSM-nm[6]-nm[24]-nm[18]-nm[12];
     if(a+b==PSM){
```

```
     if((uflg[a]==0)&&(uflg[b]==0)){
      nm[21]=a; nm[5]=b;
      uflg[a]=1; uflg[b]=1;
      stp11();
      uflg[b]=0; uflg[a]=0;}}}
}
/* Set n4=LSM-n15-n7-n21-n18 & n22 */
void stp11(){
 short a,b;
  a=LSM-nm[15]-nm[7]-nm[21]-nm[18];
  if((0<a)&&(a<26)){b=PSM-a;
    b=LSM-nm[11]-nm[19]-nm[5]-nm[8];
    if(a+b==PSM){
     if((uflg[a]==0)&&(uflg[b]==0)){
      nm[4]=a; nm[22]=b;
      uflg[a]=1; uflg[b]=1;
      stp12();
      uflg[b]=0; uflg[a]=0;}}}
}
/* Set n16=LSM-n9-n15-n22-n3 & n10 */
void stp12(){
 short a,b,c,d;
  a=LSM-nm[9]-nm[15]-nm[22]-nm[3];
  if((0<a)&&(a<26)){b=PSM-a;
    b=LSM-nm[23]-nm[4]-nm[11]-nm[17];
    if(a+b==PSM){
      c=LSM-nm[12]-nm[4]-nm[8]-nm[25];
      d=LSM-nm[23]-nm[7]-nm[5]-nm[14];
      if((a==c)&&(a==d)){
       if((uflg[a]==0)&&(uflg[b]==0)){
        nm[16]=a; nm[10]=b;
        uflg[a]=1; uflg[b]=1;
        stp13();
        uflg[b]=0; uflg[a]=0;}}}}
}
/* Checks Sums of Some 4 Entries */
void stp13(void){
 short sm1, sm2;
  sm1=nm[12]+nm[21]+nm[10]+nm[19]+nm[3];
  sm2=nm[1]+nm[18]+nm[10]+nm[22]+nm[14];
  if((sm1==LSM)&&(sm2==LSM)){stp14();}
}
/* Checks Sums of Some 4 Entries */
void stp14(void){
 short sm1, sm2, sm3, sm4;
  sm1=nm[1]+nm[7]+nm[13]+nm[19]+nm[25];
  sm2=nm[20]+nm[4]+nm[13]+nm[22]+nm[6];
  sm3=nm[9]+nm[21]+nm[13]+nm[5]+nm[17];
  sm4=nm[24]+nm[16]+nm[13]+nm[10]+nm[2];
  if((sm1==LSM)&&(sm2==LSM)&&(sm3==LSM)&&(sm4==LSM)){protorecord();}
}
/**/
/* Record the Answers */
void protorecord(){
 short n;
  for(n=1;n<26;n++){pn[cnt1][n]=nm[n];}
  cnt1++;
}
/**/
/* Print the Prototypes and the Objects */
void printall(short x, short y){
 short m,n;
  for(m=0;m<4;m++){for(n=0;n<26;n++){an[m][n]=0;}}
  cnt3=0;
```

```
  for(m=x-1;m<y;m++){
    an[cnt3*2][0]=m+1;
    for(n=1;n<26;n++){nm[n]=pn[m][n]; an[cnt3*2][n]=nm[n];}
    damt();
    an[cnt3*2+1][0]=findoldn();
    for(n=1;n<26;n++){an[cnt3*2+1][n]=dnm[n];}
  //cnt3++; if(cnt3==2){pr2ans(cnt3); cnt3=0;}
    cnt3++; if(cnt3==1){pransdcmp(cnt3); cnt3=0;}
  }
}
/**/
/* Print the Answers */
void pr2ans(short x){
 short l,l5,m,m2,n;
  for(n=0;n<x;n++){
    printf("%14d/P   S/%d",an[n*2][0],an[n*2+1][0]);
    if(n+1<x){
      if(an[n*2][0]<10){printf("             ");}
      else{printf("            ");}}
  }
  printf("\n");
  for(l=0;l<5;l++){l5=l*5;
    for(m=0;m<x;m++){m2=m*2;
      printf(" ");
      for(n=1;n<6;n++){
        printf("%3d",an[m2][l5+n]);}
      printf("  ");
      for(n=1;n<6;n++){
        printf("%3d",an[m2+1][l5+n]);}
      if(m+1<x){printf("  ");}
    }
    printf("\n");
  }
}
/**/
/* Print All Answers with /D5i */
void pransdcmp(short x){
 short c,c2,l,l5,n;
  for(c=0;c<x;c++){c2=c*2;
    printf("%14d/P   /d5i%32d/R   /d5i\n",an[c2][0],an[c2+1][0]);
    for(l=0;l<5;l++){l5=l*5;
      printf(" ");
      for(n=1;n<6;n++){printf("%3d",an[c2][l5+n]);}
      printf("  ");
      for(n=1;n<6;n++){printf("%2d",(an[c2][l5+n]-1)/5);}
      printf(" ");
      for(n=1;n<6;n++){printf("%2d",(an[c2][l5+n]-1)%5);}
      printf("   ");
      for(n=1;n<6;n++){printf("%3d",an[c2+1][l5+n]);}
      printf("  ");
      for(n=1;n<6;n++){printf("%2d",(an[c2+1][l5+n]-1)/5);}
      printf(" ");
      for(n=1;n<6;n++){printf("%2d",(an[c2+1][l5+n]-1)%5);}
      printf("\n");
    }
  }
}
/**/
/* Do-it-After-the-Model Transformation */
void damt(void){
  dnm[1]=nm[1];   dnm[2]=nm[23];  dnm[3]=nm[20];  dnm[4]=nm[12];  dnm[5]=nm[9];
  dnm[6]=nm[15];  dnm[7]=nm[7];   dnm[8]=nm[4];   dnm[9]=nm[21];  dnm[10]=nm[18];
  dnm[11]=nm[24]; dnm[12]=nm[16]; dnm[13]=nm[13]; dnm[14]=nm[10]; dnm[15]=nm[2];
  dnm[16]=nm[8];  dnm[17]=nm[5];  dnm[18]=nm[22]; dnm[19]=nm[19]; dnm[20]=nm[11];
```

```
  dnm[21]=nm[17]; dnm[22]=nm[14]; dnm[23]=nm[6];  dnm[24]=nm[3];  dnm[25]=nm[25];
}
/**/
/* Sort the Prototypes */
void sortp(){
 int m,mx,d1,d2;
 short f;
  mx=cnt1-1;
  do{f=0;
   for(m=0;m<mx;m++){
    d1=(((pn[m][1]*25+pn[m][2])*25+pn[m][3])*25+pn[m][4])*25+pn[m][5];
    d2=(((pn[m+1][1]*25+pn[m+1][2])*25+pn[m+1][3])*25+pn[m+1][4])*25+pn[m+1][5];
    if(d1>d2){exc(m); f=1;}
    else if((d1==d2)&&(pn[m][6]>pn[m+1][6])){exc(m); f=1;}
    else if((d1==d2)&&(pn[m][6]==pn[m+1][6])&&(pn[m][7]>pn[m+1][7])){exc(m); f=1;}
    else if((d1==d2)&&(pn[m][6]==pn[m+1][6])&&(pn[m][7]==pn[m+1][7])){
     if(pn[m][11]>pn[m+1][11]){exc(m); f=1;}
     else if((pn[m][11]==pn[m+1][11])&&(pn[m][16]>pn[m+1][16])){exc(m); f=1;}
    }
   }
   mx--;
  }while(f>0);
}
/**/
void exc(short x){
 short n;
 short d;
  for(n=0;n<26;n++){
    d=pn[x][n]; pn[x][n]=pn[x+1][n]; pn[x+1][n]=d;
  }
}
/**/
/* Find its Old Number */
short findoldn(){
 short fn,m;
 short cn,n;
  fn=0;
  for(m=0;m<cnt;m++){cn=0;
    for(n=1;n<26;n++){if(dnm[n]==sn[m][n]){cn++;}else{break;}}
    if(cn==25){fn=m+1; break;}
  }
  if(cn<25){fn=0;}
  ln[fn]++;
 return fn;
}
/**/
/* Print the Monitor List of Correspondence */
void prmon(){
 short m;
  printf(" ??:%3d",ln[0]);
  for(m=1;m<=cnt1;m++){
    if(m%16==1){printf("\n%3d:",m);}
    printf("%3d",ln[m]);
  }
  printf("\n");
}
/**/
```

Aren't there any other good Models for our 'DAM Transformation'?

I once selected such the best Model as mentioned above for the only common Model for all the three types of magic squares of order 5, after undergoing the strict examination of checking if it was possible to transform itself back to the original form by its own 'DAMT'.

But is it really true it is only the best Model for our purpose? Can't we really reconstruct all the object solutions with any other models and their 'DAMTs'?

We must make some actual experiments examining if it is true here and now.

Suppose you select any one of simultaneous MS55 among the list of 16 standard solutions, and assume it for the Model to dictate the transformation rules after it.

Next you make all the 16 Prototype Squares, to each of which you apply your new 'DAMT and reconstruct each of your object solutions. Can you really come back to the original 16 solutions of your simultaneous MS55?

The next two lists show the results of my recent experiments.

```
** 'Prototype Squares' and Simultaneous MS55 Reconstructed: **
 *** Both Self-complementary and Pan-diagonal: Model #1: ***
               1/P    R/1                      2/P    R/3
   1  2  3  4  5       1 23 20 14  7       1  2  3  4  5       1 23 10 14 17
   6  7  8  9 10      15  9  2 21 18      16 17 18 19 20      15 19  2 21  8
  11 12 13 14 15      22 16 13 10  4      11 12 13 14 15      22  6 13 20  4
  16 17 18 19 20       8  5 24 17 11       6  7  8  9 10      18  5 24  7 11
  21 22 23 24 25      19 12  6  3 25      21 22 23 24 25       9 12 16  3 25
               3/P    R/2                      4/P    R/4
   1  4  3  2  5       1 23 20 12  9       1  4  3  2  5       1 23 10 12 19
   6  9  8  7 10      15  7  4 21 18      16 19 18 17 20      15 17  4 21  8
  11 14 13 12 15      24 16 13 10  2      11 14 13 12 15      24  6 13 20  2
  16 19 18 17 20       8  5 22 19 11       6  9  8  7 10      18  5 22  9 11
  21 24 23 22 25      17 14  6  3 25      21 24 23 22 25       7 14 16  3 25
               5/P    R/5                      6/P    R/7
   2  1  3  5  4       2 23 19 15  6       2  1  3  5  4       2 23  9 15 16
   7  6  8 10  9      14 10  1 22 18      17 16 18 20 19      14 20  1 22  8
  12 11 13 15 14      21 17 13  9  5      12 11 13 15 14      21  7 13 19  5
  17 16 18 20 19       8  4 25 16 12       7  6  8 10  9      18  4 25  6 12
  22 21 23 25 24      20 11  7  3 24      22 21 23 25 24      10 11 17  3 24
               7/P    R/6                      8/P    R/8
   2  5  3  1  4       2 23 19 11 10       2  5  3  1  4       2 23  9 11 20
   7 10  8  6  9      14  6  5 22 18      17 20 18 16 19      14 16  5 22  8
  12 15 13 11 14      25 17 13  9  1      12 15 13 11 14      25  7 13 19  1
  17 20 18 16 19       8  4 21 20 12       7 10  8  6  9      18  4 21 10 12
  22 25 23 21 24      16 15  7  3 24      22 25 23 21 24       6 15 17  3 24
               9/P    R/9                     10/P    R/11
   4  1  3  5  2       4 23 17 15  6       4  1  3  5  2       4 23  7 15 16
   9  6  8 10  7      12 10  1 24 18      19 16 18 20 17      12 20  1 24  8
  14 11 13 15 12      21 19 13  7  5      14 11 13 15 12      21  9 13 17  5
  19 16 18 20 17       8  2 25 16 14       9  6  8 10  7      18  2 25  6 14
  24 21 23 25 22      20 11  9  3 22      24 21 23 25 22      10 11 19  3 22
              11/P    R/10                    12/P    R/12
   4  5  3  1  2       4 23 17 11 10       4  5  3  1  2       4 23  7 11 20
   9 10  8  6  7      12  6  5 24 18      19 20 18 16 17      12 16  5 24  8
  14 15 13 11 12      25 19 13  7  1      14 15 13 11 12      25  9 13 17  1
  19 20 18 16 17       8  2 21 20 14       9 10  8  6  7      18  2 21 10 14
  24 25 23 21 22      16 15  9  3 22      24 25 23 21 22       6 15 19  3 22
              13/P    R/13                    14/P    R/15
   5  2  3  4  1       5 23 16 14  7       5  2  3  4  1       5 23  6 14 17
  10  7  8  9  6      11  9  2 25 18      20 17 18 19 16      11 19  2 25  8
  15 12 13 14 11      22 20 13  6  4      15 12 13 14 11      22 10 13 16  4
  20 17 18 19 16       8  1 24 17 15      10  7  8  9  6      18  1 24  7 15
  25 22 23 24 21      19 12 10  3 21      25 22 23 24 21       9 12 20  3 21
```

```
             15/P    R/14                  16/P    R/16
  5  4  3  2  1      5 23 16 12  9      5  4  3  2  1      5 23  6 12 19
 10  9  8  7  6     11  7  4 25 18     20 19 18 17 16     11 17  4 25  8
 15 14 13 12 11     24 20 13  6  2     15 14 13 12 11     24 10 13 16  2
 20 19 18 17 16      8  1 22 19 15     10  9  8  7  6     18  1 22  9 15
 25 24 23 22 21     17 14 10  3 21     25 24 23 22 21      7 14 20  3 21
```

```
** Monitor List of Solution Correspondences **
 ??:  0
  1:  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

```
** 'Prototype Squares' and Simultaneous MS55 Reconstructed: **
***  Both Self-complementary and Pan-diagonal: Model #3:  ***
              1/P    R/3                    2/P    R/1
  1  2  3  4  5      1 23 10 14 17      1  2  3  4  5      1 23 20 14  7
  6  7  8  9 10     15 19  2 21  8     16 17 18 19 20     15  9  2 21 18
 11 12 13 14 15     22  6 13 20  4     11 12 13 14 15     22 16 13 10  4
 16 17 18 19 20     18  5 24  7 11      6  7  8  9 10      8  5 24 17 11
 21 22 23 24 25      9 12 16  3 25     21 22 23 24 25     19 12  6  3 25
              3/P    R/4                    4/P    R/2
  1  4  3  2  5      1 23 10 12 19      1  4  3  2  5      1 23 20 12  9
  6  9  8  7 10     15 17  4 21  8     16 19 18 17 20     15  7  4 21 18
 11 14 13 12 15     24  6 13 20  2     11 14 13 12 15     24 16 13 10  2
 16 19 18 17 20     18  5 22  9 11      6  9  8  7 10      8  5 22 19 11
 21 24 23 22 25      7 14 16  3 25     21 24 23 22 25     17 14  6  3 25
              5/P    R/7                    6/P    R/5
  2  1  3  5  4      2 23  9 15 16      2  1  3  5  4      2 23 19 15  6
  7  6  8 10  9     14 20  1 22  8     17 16 18 20 19     14 10  1 22 18
 12 11 13 15 14     21  7 13 19  5     12 11 13 15 14     21 17 13  9  5
 17 16 18 20 19     18  4 25  6 12      7  6  8 10  9      8  4 25 16 12
 22 21 23 25 24     10 11 17  3 24     22 21 23 25 24     20 11  7  3 24
              7/P    R/8                    8/P    R/6
  2  5  3  1  4      2 23  9 11 20      2  5  3  1  4      2 23 19 11 10
  7 10  8  6  9     14 16  5 22  8     17 20 18 16 19     14  6  5 22 18
 12 15 13 11 14     25  7 13 19  1     12 15 13 11 14     25 17 13  9  1
 17 20 18 16 19     18  4 21 10 12      7 10  8  6  9      8  4 21 20 12
 22 25 23 21 24      6 15 17  3 24     22 25 23 21 24     16 15  7  3 24
              9/P    R/11                  10/P    R/9
  4  1  3  5  2      4 23  7 15 16      4  1  3  5  2      4 23 17 15  6
  9  6  8 10  7     12 20  1 24  8     19 16 18 20 17     12 10  1 24 18
 14 11 13 15 12     21  9 13 17  5     14 11 13 15 12     21 19 13  7  5
 19 16 18 20 17     18  2 25  6 14      9  6  8 10  7      8  2 25 16 14
 24 21 23 25 22     10 11 19  3 22     24 21 23 25 22     20 11  9  3 22
             11/P    R/12                  12/P    R/10
  4  5  3  1  2      4 23  7 11 20      4  5  3  1  2      4 23 17 11 10
  9 10  8  6  7     12 16  5 24  8     19 20 18 16 17     12  6  5 24 18
 14 15 13 11 12     25  9 13 17  1     14 15 13 11 12     25 19 13  7  1
 19 20 18 16 17     18  2 21 10 14      9 10  8  6  7      8  2 21 20 14
 24 25 23 21 22      6 15 19  3 22     24 25 23 21 22     16 15  9  3 22
             13/P    R/15                  14/P    R/13
  5  2  3  4  1      5 23  6 14 17      5  2  3  4  1      5 23 16 14  7
 10  7  8  9  6     11 19  2 25  8     20 17 18 19 16     11  9  2 25 18
 15 12 13 14 11     22 10 13 16  4     15 12 13 14 11     22 20 13  6  4
 20 17 18 19 16     18  1 24  7 15     10  7  8  9  6      8  1 24 17 15
 25 22 23 24 21      9 12 20  3 21     25 22 23 24 21     19 12 10  3 21
```

```
            15/P    R/16                            16/P    R/14
  5  4  3  2  1     5 23  6 12 19     5  4  3  2  1     5 23 16 12  9
 10  9  8  7  6    11 17  4 25  8    20 19 18 17 16    11  7  4 25 18
 15 14 13 12 11    24 10 13 16  2    15 14 13 12 11    24 20 13  6  2
 20 19 18 17 16    18  1 22  9 15    10  9  8  7  6     8  1 22 19 15
 25 24 23 22 21     7 14 20  3 21    25 24 23 22 21    17 14 10  3 21

** Monitor List of Solution Correspondences **
 ??:  0
  1:  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

Why! The results show any models could really perform well, say, enough to be called as 'good Models'. I was extremely surprised to know that any of 16 solutions could be the good Model and reconstruct all members of its family. It indicates they were naturally born equal and complete.

Any one knows all, and any one can make all. How impressive it is!

My philosophical mind is getting more active and would hardly sleep.

But why is it made so 'democratic'? What does it really mean, I say, mathematically?

I have not yet known the exact answer for it, but I feel there must be some great 'symmetry' among the object solutions, and we should be able to discover it just among the solution set of Prototype Squares, for it is the collective set of Basic Forms. It might have any 'cyclic' structure to transform one form into another and make it back into the most basic one after all, whenever and wherever it should start.

(Written on October 17, 2006 by Kanji Setsuda working on MacOS X & Xcode 2.2)

E-Mail Address:<jag12001@nifty.ne.jp>