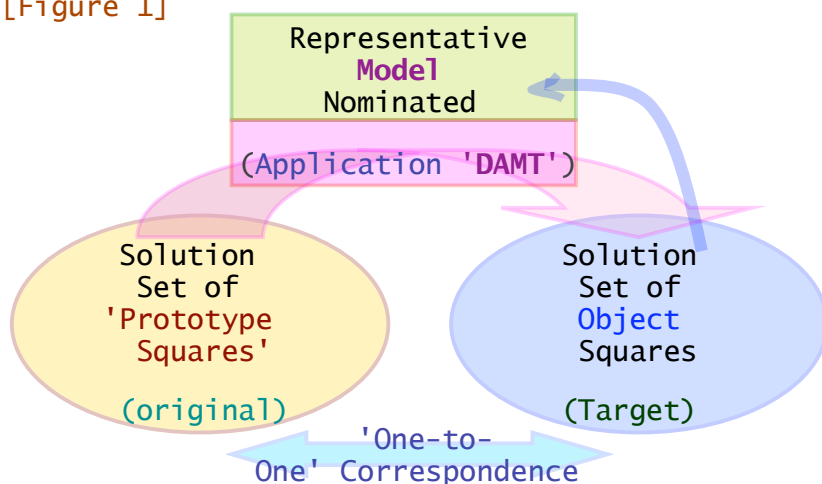


Chapter 1: Fundamental Study of 'Prototype' Squares and 'Do-it-After-the-Model' Transformation: Kanji Setsuda

Section 2: Prototype Squares of Order 4 and 'Do-it-After-the-Model Transformation'

Let's go on studying about the two 'parallel worlds' of magic squares and the beautiful 'bridge' between them.

** 'Prototype Squares' and 'Do-it-After-the-Model Transformation' **
 [Figure 1]



Now we are going to study those of order 4. One is the world of 'Prototype Squares', and the other is the world of magic squares: (1) Self-complementary type and (2) Pan-diagonal type or Complete Pan-magic one.

Look at each example of these three squares shown below.

[Figure 2: The Primary Representative Models]

'Prototype Squares' of order 4	Self-Complementary Magic Squares of order 4	Complete Pan-Magic Squares of order 4																																																
<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>5</td><td>6</td><td>7</td><td>8</td></tr> <tr><td>9</td><td>10</td><td>11</td><td>12</td></tr> <tr><td>13</td><td>14</td><td>15</td><td>16</td></tr> </table>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	<table border="1"> <tr><td>1</td><td>15</td><td>14</td><td>4</td></tr> <tr><td>12</td><td>6</td><td>7</td><td>9</td></tr> <tr><td>8</td><td>10</td><td>11</td><td>5</td></tr> <tr><td>13</td><td>3</td><td>2</td><td>16</td></tr> </table>	1	15	14	4	12	6	7	9	8	10	11	5	13	3	2	16	<table border="1"> <tr><td>1</td><td>15</td><td>4</td><td>14</td></tr> <tr><td>12</td><td>6</td><td>9</td><td>7</td></tr> <tr><td>13</td><td>3</td><td>16</td><td>2</td></tr> <tr><td>8</td><td>10</td><td>5</td><td>11</td></tr> </table>	1	15	4	14	12	6	9	7	13	3	16	2	8	10	5	11
1	2	3	4																																															
5	6	7	8																																															
9	10	11	12																																															
13	14	15	16																																															
1	15	14	4																																															
12	6	7	9																																															
8	10	11	5																																															
13	3	2	16																																															
1	15	4	14																																															
12	6	9	7																																															
13	3	16	2																																															
8	10	5	11																																															

These three are all the primary representative models of (1) 'Prototype Squares', (2) Self-complementary and (3) Complete Pan-magic squares of Order 4.

We already know that each type of the latter two has the set of as many as 48 'standard' solutions (equivalent to 384 'primitive' solutions).

The primary prototype square at the left end is made after the regular array 4x4. The value of each entry is always equal to the value of each position name itself. $n_1=1, n_2=2, n_3=3, \dots, n_{10}=10, \dots, n_{15}=15$ and $n_{16}=16$

#1. 'Prototype Squares' for Self-complementary Magic Squares of Order 4

15 16 13 14 15 16 13 14 [Figure 3: Extended Form]

3	4	1	2	3	4	1	2	Complementary pairs of 17: $1+16=2+15=3+14=4+13$ $=5+12=6+11=7+10=8+9=17$ Eight pairs are all located symmetrically.
7	8	5	6	7	8	5	6	
11	12	9	10	11	12	9	10	
15	16	13	14	15	16	13	14	
3	4	1	2	3	4	1	2	

$1+2+3+4=10, 5+6+7+8=26,$
 $9+10+11+12=42, \dots,$
 $1+5+9+13=28, 2+6+10+14=32,$
 $\dots, 4+8+12+16=40$

But $1+6+11+16=34, 1+8+11+14=34, 2+7+12+13=34, 2+5+12+15=34,$
 $3+8+9+14=34, 3+6+9+16=34, 4+5+10+15=34, 4+7+10+13=34$
Four entries on every Pan-diagonal add up to the same sum 34.

Each Prototype Square is already the Simultaneous type of both Self-complementary and Pan-diagonal, as you see, although no row and no column add up to the same magic constant 34.

On top of that, it has some other interesting properties such as:

- $1+6=2+5, 2+7=3+6, 3+8=4+7, 5+10=6+9, 6+11=7+10, 7+12=8+11,$
 $9+14=10+13, 10+15=11+14, 11+16=12+15$ (Equality of Cross-sums)
- $1+7=3+5, 2+8=4+6, 1+8=4+5, 5+11=7+9, 5+12=8+9,$
 $6+12=8+10, 9+15=11+13, 10+16=12+14, 9+16=12+13,$
 $1+10=2+9, 5+14=6+13, 1+14=2+13, 2+11=3+10, 6+15=7+14,$
 $2+15=3+14, 3+12=4+11, 7+16=8+15, 3+16=4+15,$
 $1+11=3+9, 2+12=4+10, 5+15=7+13, 6+16=8+14, 1+16=4+13$
 (Equality of Cross-sums in broad sense)
- $1+4=2+3, 5+8=6+7, 9+12=10+11, 13+16=14+15,$
 $1+13=5+9, 2+14=6+10, 3+15=7+11, 4+16=8+12$
 (The sum of outer entries is equal to the sum of inner entries in a line.)

1-2. How many different patterns can you make for this set of prototype squares?
 Try to rearrange all entries and make new patterns under those strict conditions.
 Put it no inequality conditions like $n1 < n4, n1 < n13, n1 < n16$ or $n2 > n5$ now before all.
 Of course, you can use your personal computer to make and count them through.
 And you will get 384 different patterns in all like the list shown below.

*** List 1: 'Prototype Squares' of Order 4 (Part) ***

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

25/ 2 1 4 3 6 5 8 7 10 9 12 11 14 13 16 15	26/ 2 1 4 3 10 9 12 11 6 5 8 7 14 13 16 15	27/ 2 1 6 5 4 3 8 7 10 9 14 13 12 11 16 15	28/ 2 1 6 5 10 9 14 13 4 3 8 7 12 11 16 15	29/ 2 1 10 9 4 3 12 11 6 5 14 13 8 7 16 15	30/ 2 1 10 9 6 5 14 13 4 3 12 11 8 7 16 15
31/ 2 4 1 3 6 8 5 7 10 12 9 11 14 16 13 15	32/ 2 4 1 3 10 12 9 11 6 8 5 7 14 16 13 15	33/ 2 4 6 8 1 3 5 7 10 12 14 16 9 11 13 15	34/ 2 4 6 8 10 12 14 16 1 3 5 7 9 11 13 15	35/ 2 4 10 12 1 3 9 11 6 8 14 16 5 7 13 15	36/ 2 4 10 12 6 8 14 16 1 3 9 11 5 7 13 15
37/ 2 6 1 5 4 8 3 7 10 14 9 13 12 16 11 15	38/ 2 6 1 5 10 14 9 13 4 8 3 7 12 16 11 15	39/ 2 6 4 8 1 5 3 7 10 14 12 16 9 13 11 15	40/ 2 6 4 8 10 14 12 16 1 5 3 7 9 13 11 15	41/ 2 6 10 14 1 5 9 13 4 8 12 16 3 7 11 15	42/ 2 6 10 14 4 8 12 16 1 5 9 13 3 7 11 15
43/ 2 10 1 9 4 12 3 11 6 14 5 13 8 16 7 15	44/ 2 10 1 9 6 14 5 13 4 12 3 11 8 16 7 15	45/ 2 10 4 12 1 9 3 11 6 14 8 16 5 13 7 15	46/ 2 10 4 12 6 14 8 16 1 9 3 11 5 13 7 15	47/ 2 10 6 14 1 9 5 13 4 12 8 16 3 11 7 15	48/ 2 10 6 14 4 12 8 16 1 9 5 13 3 11 7 15
49/ 3 1 4 2 7 5 8 6 11 9 12 10 15 13 16 14	50/ 3 1 4 2 11 9 12 10 7 5 8 6 15 13 16 14	51/ 3 1 7 5 4 2 8 6 11 9 15 13 12 10 16 14	52/ 3 1 7 5 11 9 15 13 4 2 8 6 12 10 16 14	53/ 3 1 11 9 4 2 12 10 7 5 15 13 8 6 16 14	54/ 3 1 11 9 7 5 15 13 4 2 12 10 8 6 16 14
55/ 3 4 1 2 7 8 5 6 11 12 9 10 15 16 13 14	56/ 3 4 1 2 11 12 9 10 7 8 5 6 15 16 13 14	57/ 3 4 7 8 1 2 5 6 11 12 15 16 9 10 13 14	58/ 3 4 7 8 11 12 15 16 1 2 5 6 9 10 13 14	59/ 3 4 11 12 1 2 9 10 7 8 15 16 5 6 13 14	60/ 3 4 11 12 7 8 15 16 1 2 9 10 5 6 13 14
61/ 3 7 1 5 4 8 2 6 11 15 9 13 12 16 10 14	62/ 3 7 1 5 11 15 9 13 4 8 2 6 12 16 10 14	63/ 3 7 4 8 1 5 2 6 11 15 12 16 9 13 10 14	64/ 3 7 4 8 11 15 12 16 1 5 2 6 9 13 10 14	65/ 3 7 11 15 1 5 9 13 4 8 12 16 2 6 10 14	66/ 3 7 11 15 4 8 12 16 1 5 9 13 2 6 10 14
67/ 3 11 1 9 4 12 2 10 7 15 5 13 8 16 6 14	68/ 3 11 1 9 7 15 5 13 4 12 2 10 8 16 6 14	69/ 3 11 4 12 1 9 2 10 7 15 8 16 5 13 6 14	70/ 3 11 4 12 7 15 8 16 1 9 2 10 5 13 6 14	71/ 3 11 7 15 1 9 5 13 4 12 8 16 2 10 6 14	72/ 3 11 7 15 4 12 8 16 1 9 5 13 2 10 6 14
73/ 4 2 3 1 8 6 7 5 12 10 11 9 16 14 15 13	74/ 4 2 3 1 12 10 11 9 8 6 7 5 16 14 15 13	75/ 4 2 8 6 3 1 7 5 12 10 16 14 11 9 15 13	76/ 4 2 8 6 12 10 16 14 3 1 7 5 11 9 15 13	77/ 4 2 12 10 3 1 11 9 8 6 16 14 7 5 15 13	78/ 4 2 12 10 8 6 16 14 3 1 11 9 7 5 15 13

. . .(Skip). . .

355/ 15 16 7 8 11 12 3 4 13 14 5 6 9 10 1 2	356/ 15 16 7 8 13 14 5 6 11 12 3 4 9 10 1 2	357/ 15 16 11 12 7 8 3 4 13 14 9 10 5 6 1 2	358/ 15 16 11 12 13 14 9 10 7 8 3 4 5 6 1 2	359/ 15 16 13 14 7 8 5 6 11 12 9 10 3 4 1 2	360/ 15 16 13 14 11 12 9 10 7 8 5 6 3 4 1 2
361/ 16 8 12 4 14 6 10 2 15 7 11 3 13 5 9 1	362/ 16 8 12 4 15 7 11 3 14 6 10 2 13 5 9 1	363/ 16 8 14 6 12 4 10 2 15 7 13 5 11 3 9 1	364/ 16 8 14 6 15 7 13 5 12 4 10 2 11 3 9 1	365/ 16 8 15 7 12 4 11 3 14 6 13 5 10 2 9 1	366/ 16 8 15 7 14 6 13 5 12 4 11 3 10 2 9 1
367/ 16 12 8 4 14 10 6 2 15 11 7 3 13 9 5 1	368/ 16 12 8 4 15 11 7 3 14 10 6 2 13 9 5 1	369/ 16 12 14 10 8 4 6 2 15 11 13 9 7 3 5 1	370/ 16 12 14 10 15 11 13 9 8 4 6 2 7 3 5 1	371/ 16 12 15 11 8 4 7 3 14 10 13 9 6 2 5 1	372/ 16 12 15 11 14 10 13 9 8 4 7 3 6 2 5 1

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

[Count = 384]

1-3. How can you combine these 384 patterns of Prototype Squares above to those 384 primitive solutions of Self-complementary Magic Squares below one by one?

**** List 2: Self-complementary Magic Squares 4*4: 384 Solutions ****

1/	2/	3/	4/	5/	6/
1 15 14 4	1 15 14 4	1 15 12 6	1 15 12 6	1 15 8 10	1 15 8 10
12 6 7 9	8 10 11 5	14 4 7 9	8 10 13 3	14 4 11 5	12 6 13 3
8 10 11 5	12 6 7 9	8 10 13 3	14 4 7 9	12 6 13 3	14 4 11 5
13 3 2 16	13 3 2 16	11 5 2 16	11 5 2 16	7 9 2 16	7 9 2 16
7/	8/	9/	10/	11/	12/
1 14 15 4	1 14 15 4	1 14 12 7	1 14 12 7	1 14 8 11	1 14 8 11
12 7 6 9	8 11 10 5	15 4 6 9	8 11 13 2	15 4 10 5	12 7 13 2
8 11 10 5	12 7 6 9	8 11 13 2	15 4 6 9	12 7 13 2	15 4 10 5
13 2 3 16	13 2 3 16	10 5 3 16	10 5 3 16	6 9 3 16	6 9 3 16
13/	14/	15/	16/	17/	18/
1 12 15 6	1 12 15 6	1 12 14 7	1 12 14 7	1 12 8 13	1 12 8 13
14 7 4 9	8 13 10 3	15 6 4 9	8 13 11 2	15 6 10 3	14 7 11 2
8 13 10 3	14 7 4 9	8 13 11 2	15 6 4 9	14 7 11 2	15 6 10 3
11 2 5 16	11 2 5 16	10 3 5 16	10 3 5 16	4 9 5 16	4 9 5 16
19/	20/	21/	22/	23/	24/
1 8 15 10	1 8 15 10	1 8 14 11	1 8 14 11	1 8 12 13	1 8 12 13
14 11 4 5	12 13 6 3	15 10 4 5	12 13 7 2	15 10 6 3	14 11 7 2
12 13 6 3	14 11 4 5	12 13 7 2	15 10 4 5	14 11 7 2	15 10 6 3
7 2 9 16	7 2 9 16	6 3 9 16	6 3 9 16	4 5 9 16	4 5 9 16
25/	26/	27/	28/	29/	30/
2 16 13 3	2 16 13 3	2 16 11 5	2 16 11 5	2 16 7 9	2 16 7 9
11 5 8 10	7 9 12 6	13 3 8 10	7 9 14 4	13 3 12 6	11 5 14 4
7 9 12 6	11 5 8 10	7 9 14 4	13 3 8 10	11 5 14 4	13 3 12 6
14 4 1 15	14 4 1 15	12 6 1 15	12 6 1 15	8 10 1 15	8 10 1 15
31/	32/	33/	34/	35/	36/
2 13 16 3	2 13 16 3	2 13 11 8	2 13 11 8	2 13 7 12	2 13 7 12
11 8 5 10	7 12 9 6	16 3 5 10	7 12 14 1	16 3 9 6	11 8 14 1
7 12 9 6	11 8 5 10	7 12 14 1	16 3 5 10	11 8 14 1	16 3 9 6
14 1 4 15	14 1 4 15	9 6 4 15	9 6 4 15	5 10 4 15	5 10 4 15
37/	38/	39/	40/	41/	42/
2 11 16 5	2 11 16 5	2 11 13 8	2 11 13 8	2 11 7 14	2 11 7 14
13 8 3 10	7 14 9 4	16 5 3 10	7 14 12 1	16 5 9 4	13 8 12 1
7 14 9 4	13 8 3 10	7 14 12 1	16 5 3 10	13 8 12 1	16 5 9 4
12 1 6 15	12 1 6 15	9 4 6 15	9 4 6 15	3 10 6 15	3 10 6 15
43/	44/	45/	46/	47/	48/
2 7 16 9	2 7 16 9	2 7 13 12	2 7 13 12	2 7 11 14	2 7 11 14
13 12 3 6	11 14 5 4	16 9 3 6	11 14 8 1	16 9 5 4	13 12 8 1
11 14 5 4	13 12 3 6	11 14 8 1	16 9 3 6	13 12 8 1	16 9 5 4
8 1 10 15	8 1 10 15	5 4 10 15	5 4 10 15	3 6 10 15	3 6 10 15
49/	50/	51/	52/	53/	54/
3 16 13 2	3 16 13 2	3 16 10 5	3 16 10 5	3 16 6 9	3 16 6 9
10 5 8 11	6 9 12 7	13 2 8 11	6 9 15 4	13 2 12 7	10 5 15 4
6 9 12 7	10 5 8 11	6 9 15 4	13 2 8 11	10 5 15 4	13 2 12 7
15 4 1 14	15 4 1 14	12 7 1 14	12 7 1 14	8 11 1 14	8 11 1 14

55/ 3 13 16 2 10 8 5 11 6 12 9 7 15 1 4 14	56/ 3 13 16 2 6 12 9 7 10 8 5 11 15 1 4 14	57/ 3 13 10 8 16 2 5 11 6 12 15 1 9 7 4 14	58/ 3 13 10 8 6 12 15 1 16 2 5 11 9 7 4 14	59/ 3 13 6 12 16 2 9 7 10 8 15 1 5 11 4 14	60/ 3 13 6 12 10 8 15 1 16 2 9 7 5 11 4 14
61/ 3 10 16 5 13 8 2 11 6 15 9 4 12 1 7 14	62/ 3 10 16 5 6 15 9 4 13 8 2 11 12 1 7 14	63/ 3 10 13 8 16 5 2 11 6 15 12 1 9 4 7 14	64/ 3 10 13 8 6 15 12 1 16 5 2 11 9 4 7 14	65/ 3 10 6 15 16 5 9 4 13 8 12 1 2 11 7 14	66/ 3 10 6 15 13 8 12 1 16 5 9 4 2 11 7 14
67/ 3 6 16 9 13 12 2 7 10 15 5 4 8 1 11 14	68/ 3 6 16 9 10 15 5 4 13 12 2 7 8 1 11 14	69/ 3 6 13 12 16 9 2 7 10 15 8 1 5 4 11 14	70/ 3 6 13 12 10 15 8 1 16 9 2 7 5 4 11 14	71/ 3 6 10 15 16 9 5 4 13 12 8 1 2 7 11 14	72/ 3 6 10 15 13 12 8 1 16 9 5 4 2 7 11 14
73/ 4 15 14 1 9 6 7 12 5 10 11 8 16 3 2 13	74/ 4 15 14 1 5 10 11 8 9 6 7 12 16 3 2 13	75/ 4 15 9 6 14 1 7 12 5 10 16 3 11 8 2 13	76/ 4 15 9 6 5 10 16 3 14 1 7 12 11 8 2 13	77/ 4 15 5 10 14 1 11 8 9 6 16 3 7 12 2 13	78/ 4 15 5 10 9 6 16 3 14 1 11 8 7 12 2 13

... (Skip) ...

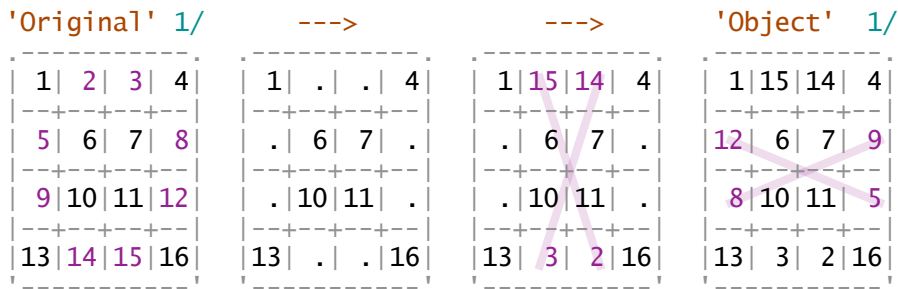
355/ 15 1 10 8 6 12 3 13 4 14 5 11 9 7 16 2	356/ 15 1 10 8 4 14 5 11 6 12 3 13 9 7 16 2	357/ 15 1 6 12 10 8 3 13 4 14 9 7 5 11 16 2	358/ 15 1 6 12 4 14 9 7 10 8 3 13 5 11 16 2	359/ 15 1 4 14 10 8 5 11 6 12 9 7 3 13 16 2	360/ 15 1 4 14 6 12 9 7 10 8 5 11 3 13 16 2
361/ 16 9 5 4 3 6 10 15 2 7 11 14 13 12 8 1	362/ 16 9 5 4 2 7 11 14 3 6 10 15 13 12 8 1	363/ 16 9 3 6 5 4 10 15 2 7 13 12 11 14 8 1	364/ 16 9 3 6 2 7 13 12 5 4 10 15 11 14 8 1	365/ 16 9 2 7 5 4 11 14 3 6 13 12 10 15 8 1	366/ 16 9 2 7 3 6 13 12 5 4 11 14 10 15 8 1
367/ 16 5 9 4 3 10 6 15 2 11 7 14 13 8 12 1	368/ 16 5 9 4 2 11 7 14 3 10 6 15 13 8 12 1	369/ 16 5 3 10 9 4 6 15 2 11 13 8 7 14 12 1	370/ 16 5 3 10 2 11 13 8 9 4 6 15 7 14 12 1	371/ 16 5 2 11 9 4 7 14 3 10 13 8 6 15 12 1	372/ 16 5 2 11 3 10 13 8 9 4 7 14 6 15 12 1
373/ 16 3 9 6 5 10 4 15 2 13 7 12 11 8 14 1	374/ 16 3 9 6 2 13 7 12 5 10 4 15 11 8 14 1	375/ 16 3 5 10 9 6 4 15 2 13 11 8 7 12 14 1	376/ 16 3 5 10 2 13 11 8 9 6 4 15 7 12 14 1	377/ 16 3 2 13 9 6 7 12 5 10 11 8 4 15 14 1	378/ 16 3 2 13 5 10 11 8 9 6 7 12 4 15 14 1
379/ 16 2 9 7 5 11 4 14 3 13 6 12 10 8 15 1	380/ 16 2 9 7 3 13 6 12 5 11 4 14 10 8 15 1	381/ 16 2 5 11 9 7 4 14 3 13 10 8 6 12 15 1	382/ 16 2 5 11 3 13 10 8 9 7 4 14 6 12 15 1	383/ 16 2 3 13 9 7 6 12 5 11 10 8 4 14 15 1	384/ 16 2 3 13 5 11 10 8 9 7 6 12 4 14 15 1

[Count = 384]

At first we have to find the way how to transform the primary representative form of Prototype Squares (Let's call it the "original") into the one of Self-complementary Magic Squares (Let's call it the "object"). And then we have to examine if we can apply the very transformation method to the other 'original' squares one after another to make them all into our 'object' solutions without any fails.

Let me introduce you the concept of transformation I always assume recently. It looks like as shown in the next diagrams.

[Figure 4: The Transformation]



- (1) Hold 8 entries on the two primary diagonals still in the new 'object' form.
- (2) What do we have to do for the rest 8 ones?
Find four complementary pairs of 17 such as: (2, 15), (3, 14), (5, 12) and (8, 9); and exchange their locations with each partner as shown above.

1-4. Apply this way of transformation to each solution of the original, and you will get each other solution of the object. Examine if this transformation works all right.

The "Rules of Transformation" below should express this transformation method in another way. They are directly dictated after the object Model literally.

- (1) Put the value of n1(original) into n1(object).
- (2) Put the value of n15(original) into n2(object).
- (3) Put the value of n14(original) into n3(object).
- (4) Put the value of n4(original) into n4(object).
-
- (9) Put the value of n8(original) into n9(object).
-
- (15) Put the value of n2(original) into n15(object).
- (16) Put the value of n16(original) into n16(object).

In your computer program these rules might be easy for you to write.

Once you select the representative model, make the set of these rules above **after the model** and apply it to the other 'prototype' solutions one after another, you can automatically **transform** them and have got all the rest object squares at last.

Let's call this way of making objects as "**Do-it-After-the-Model Transformation**".

The next list shows the result of this transformation method called 'D.A.M.T.'

Each number on the shoulder of object ('S/n') refers to our old independent List of Self-complementary Solutions.

**** List 3: 'Prototype Squares' 4*4 and S-C MS44 Recomposed ****

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

10/P S/10	11/P S/11	12/P S/12
1 3 5 7 1 14 12 7	1 3 9 11 1 14 8 11	1 3 9 11 1 14 8 11
9 11 13 15 8 11 13 2	2 4 10 12 15 4 10 5	5 7 13 15 12 7 13 2
2 4 6 8 15 4 6 9	5 7 13 15 12 7 13 2	2 4 10 12 15 4 10 5
10 12 14 16 10 5 3 16	6 8 14 16 6 9 3 16	6 8 14 16 6 9 3 16
13/P S/13	14/P S/14	15/P S/15
1 5 2 6 1 12 15 6	1 5 2 6 1 12 15 6	1 5 3 7 1 12 14 7
3 7 4 8 14 7 4 9	9 13 10 14 8 13 10 3	2 6 4 8 15 6 4 9
9 13 10 14 8 13 10 3	3 7 4 8 14 7 4 9	9 13 11 15 8 13 11 2
11 15 12 16 11 2 5 16	11 15 12 16 11 2 5 16	10 14 12 16 10 3 5 16
16/P S/16	17/P S/17	18/P S/18
1 5 3 7 1 12 14 7	1 5 9 13 1 12 8 13	1 5 9 13 1 12 8 13
9 13 11 15 8 13 11 2	2 6 10 14 15 6 10 3	3 7 11 15 14 7 11 2
2 6 4 8 15 6 4 9	3 7 11 15 14 7 11 2	2 6 10 14 15 6 10 3
10 14 12 16 10 3 5 16	4 8 12 16 4 9 5 16	4 8 12 16 4 9 5 16
19/P S/19	20/P S/20	21/P S/21
1 9 2 10 1 8 15 10	1 9 2 10 1 8 15 10	1 9 3 11 1 8 14 11
3 11 4 12 14 11 4 5	5 13 6 14 12 13 6 3	2 10 4 12 15 10 4 5
5 13 6 14 12 13 6 3	3 11 4 12 14 11 4 5	5 13 7 15 12 13 7 2
7 15 8 16 7 2 9 16	7 15 8 16 7 2 9 16	6 14 8 16 6 3 9 16
22/P S/22	23/P S/23	24/P S/24
1 9 3 11 1 8 14 11	1 9 5 13 1 8 12 13	1 9 5 13 1 8 12 13
5 13 7 15 12 13 7 2	2 10 6 14 15 10 6 3	3 11 7 15 14 11 7 2
2 10 4 12 15 10 4 5	3 11 7 15 14 11 7 2	2 10 6 14 15 10 6 3
6 14 8 16 6 3 9 16	4 12 8 16 4 5 9 16	4 12 8 16 4 5 9 16
25/P S/25	26/P S/26	27/P S/27
2 1 4 3 2 16 13 3	2 1 4 3 2 16 13 3	2 1 6 5 2 16 11 5
6 5 8 7 11 5 8 10	10 9 12 11 7 9 12 6	4 3 8 7 13 3 8 10
10 9 12 11 7 9 12 6	6 5 8 7 11 5 8 10	10 9 14 13 7 9 14 4
14 13 16 15 14 4 1 15	14 13 16 15 14 4 1 15	12 11 16 15 12 6 1 15
28/P S/28	29/P S/29	30/P S/30
2 1 6 5 2 16 11 5	2 1 10 9 2 16 7 9	2 1 10 9 2 16 7 9
10 9 14 13 7 9 14 4	4 3 12 11 13 3 12 6	6 5 14 13 11 5 14 4
4 3 8 7 13 3 8 10	6 5 14 13 11 5 14 4	4 3 12 11 13 3 12 6
12 11 16 15 12 6 1 15	8 7 16 15 8 10 1 15	8 7 16 15 8 10 1 15
31/P S/31	32/P S/32	33/P S/33
2 4 1 3 2 13 16 3	2 4 1 3 2 13 16 3	2 4 6 8 2 13 11 8
6 8 5 7 11 8 5 10	10 12 9 11 7 12 9 6	1 3 5 7 16 3 5 10
10 12 9 11 7 12 9 6	6 8 5 7 11 8 5 10	10 12 14 16 7 12 14 1
14 16 13 15 14 1 4 15	14 16 13 15 14 1 4 15	9 11 13 15 9 6 4 15
34/P S/34	35/P S/35	36/P S/36
2 4 6 8 2 13 11 8	2 4 10 12 2 13 7 12	2 4 10 12 2 13 7 12
10 12 14 16 7 12 14 1	1 3 9 11 16 3 9 6	6 8 14 16 11 8 14 1
1 3 5 7 16 3 5 10	6 8 14 16 11 8 14 1	1 3 9 11 16 3 9 6
9 11 13 15 9 6 4 15	5 7 13 15 5 10 4 15	5 7 13 15 5 10 4 15
37/P S/37	38/P S/38	39/P S/39
2 6 1 5 2 11 16 5	2 6 1 5 2 11 16 5	2 6 4 8 2 11 13 8
4 8 3 7 13 8 3 10	10 14 9 13 7 14 9 4	1 5 3 7 16 5 3 10
10 14 9 13 7 14 9 4	4 8 3 7 13 8 3 10	10 14 12 16 7 14 12 1
12 16 11 15 12 1 6 15	12 16 11 15 12 1 6 15	9 13 11 15 9 4 6 15
40/P S/40	41/P S/41	42/P S/42
2 6 4 8 2 11 13 8	2 6 10 14 2 11 7 14	2 6 10 14 2 11 7 14
10 14 12 16 7 14 12 1	1 5 9 13 16 5 9 4	4 8 12 16 13 8 12 1
1 5 3 7 16 5 3 10	4 8 12 16 13 8 12 1	1 5 9 13 16 5 9 4
9 13 11 15 9 4 6 15	3 7 11 15 3 10 6 15	3 7 11 15 3 10 6 15

43/P S/43	44/P S/44	45/P S/45
2 10 1 9 2 7 16 9	2 10 1 9 2 7 16 9	2 10 4 12 2 7 13 12
4 12 3 11 13 12 3 6	6 14 5 13 11 14 5 4	1 9 3 11 16 9 3 6
6 14 5 13 11 14 5 4	4 12 3 11 13 12 3 6	6 14 8 16 11 14 8 1
8 16 7 15 8 1 10 15	8 16 7 15 8 1 10 15	5 13 7 15 5 4 10 15
46/P S/46	47/P S/47	48/P S/48
2 10 4 12 2 7 13 12	2 10 6 14 2 7 11 14	2 10 6 14 2 7 11 14
6 14 8 16 11 14 8 1	1 9 5 13 16 9 5 4	4 12 8 16 13 12 8 1
1 9 3 11 16 9 3 6	4 12 8 16 13 12 8 1	1 9 5 13 16 9 5 4
5 13 7 15 5 4 10 15	3 11 7 15 3 6 10 15	3 11 7 15 3 6 10 15
49/P S/49	50/P S/50	51/P S/51
3 1 4 2 3 16 13 2	3 1 4 2 3 16 13 2	3 1 7 5 3 16 10 5
7 5 8 6 10 5 8 11	11 9 12 10 6 9 12 7	4 2 8 6 13 2 8 11
11 9 12 10 6 9 12 7	7 5 8 6 10 5 8 11	11 9 15 13 6 9 15 4
15 13 16 14 15 4 1 14	15 13 16 14 15 4 1 14	12 10 16 14 12 7 1 14
52/P S/52	53/P S/53	54/P S/54
3 1 7 5 3 16 10 5	3 1 11 9 3 16 6 9	3 1 11 9 3 16 6 9
11 9 15 13 6 9 15 4	4 2 12 10 13 2 12 7	7 5 15 13 10 5 15 4
4 2 8 6 13 2 8 11	7 5 15 13 10 5 15 4	4 2 12 10 13 2 12 7
12 10 16 14 12 7 1 14	8 6 16 14 8 11 1 14	8 6 16 14 8 11 1 14
55/P S/55	56/P S/56	57/P S/57
3 4 1 2 3 13 16 2	3 4 1 2 3 13 16 2	3 4 7 8 3 13 10 8
7 8 5 6 10 8 5 11	11 12 9 10 6 12 9 7	1 2 5 6 16 2 5 11
11 12 9 10 6 12 9 7	7 8 5 6 10 8 5 11	11 12 15 16 6 12 15 1
15 16 13 14 15 1 4 14	15 16 13 14 15 1 4 14	9 10 13 14 9 7 4 14
58/P S/58	59/P S/59	60/P S/60
3 4 7 8 3 13 10 8	3 4 11 12 3 13 6 12	3 4 11 12 3 13 6 12
11 12 15 16 6 12 15 1	1 2 9 10 16 2 9 7	7 8 15 16 10 8 15 1
1 2 5 6 16 2 5 11	7 8 15 16 10 8 15 1	1 2 9 10 16 2 9 7
9 10 13 14 9 7 4 14	5 6 13 14 5 11 4 14	5 6 13 14 5 11 4 14
61/P S/61	62/P S/62	63/P S/63
3 7 1 5 3 10 16 5	3 7 1 5 3 10 16 5	3 7 4 8 3 10 13 8
4 8 2 6 13 8 2 11	11 15 9 13 6 15 9 4	1 5 2 6 16 5 2 11
11 15 9 13 6 15 9 4	4 8 2 6 13 8 2 11	11 15 12 16 6 15 12 1
12 16 10 14 12 1 7 14	12 16 10 14 12 1 7 14	9 13 10 14 9 4 7 14
64/P S/64	65/P S/65	66/P S/66
3 7 4 8 3 10 13 8	3 7 11 15 3 10 6 15	3 7 11 15 3 10 6 15
11 15 12 16 6 15 12 1	1 5 9 13 16 5 9 4	4 8 12 16 13 8 12 1
1 5 2 6 16 5 2 11	4 8 12 16 13 8 12 1	1 5 9 13 16 5 9 4
9 13 10 14 9 4 7 14	2 6 10 14 2 11 7 14	2 6 10 14 2 11 7 14
67/P S/67	68/P S/68	69/P S/69
3 11 1 9 3 6 16 9	3 11 1 9 3 6 16 9	3 11 4 12 3 6 13 12
4 12 2 10 13 12 2 7	7 15 5 13 10 15 5 4	1 9 2 10 16 9 2 7
7 15 5 13 10 15 5 4	4 12 2 10 13 12 2 7	7 15 8 16 10 15 8 1
8 16 6 14 8 1 11 14	8 16 6 14 8 1 11 14	5 13 6 14 5 4 11 14
70/P S/70	71/P S/71	72/P S/72
3 11 4 12 3 6 13 12	3 11 7 15 3 6 10 15	3 11 7 15 3 6 10 15
7 15 8 16 10 15 8 1	1 9 5 13 16 9 5 4	4 12 8 16 13 12 8 1
1 9 2 10 16 9 2 7	4 12 8 16 13 12 8 1	1 9 5 13 16 9 5 4
5 13 6 14 5 4 11 14	2 10 6 14 2 7 11 14	2 10 6 14 2 7 11 14
73/P S/73	74/P S/74	75/P S/75
4 2 3 1 4 15 14 1	4 2 3 1 4 15 14 1	4 2 8 6 4 15 9 6
8 6 7 5 9 6 7 12	12 10 11 9 5 10 11 8	3 1 7 5 14 1 7 12
12 10 11 9 5 10 11 8	8 6 7 5 9 6 7 12	12 10 16 14 5 10 16 3
16 14 15 13 16 3 2 13	16 14 15 13 16 3 2 13	11 9 15 13 11 8 2 13

. . .(Skip). . .

376/P S/376				377/P S/377				378/P S/378															
16	14	12	10	16	3	5	10	16	14	15	13	16	3	2	13	16	14	15	13	16	3	2	13
15	13	11	9	2	13	11	8	8	6	7	5	9	6	7	12	12	10	11	9	5	10	11	8
8	6	4	2	9	6	4	15	12	10	11	9	5	10	11	8	8	6	7	5	9	6	7	12
7	5	3	1	7	12	14	1	4	2	3	1	4	15	14	1	4	2	3	1	4	15	14	1
379/P S/379				380/P S/380				381/P S/381															
16	15	8	7	16	2	9	7	16	15	8	7	16	2	9	7	16	15	12	11	16	2	5	11
12	11	4	3	5	11	4	14	14	13	6	5	3	13	6	12	8	7	4	3	9	7	4	14
14	13	6	5	3	13	6	12	12	11	4	3	5	11	4	14	14	13	10	9	3	13	10	8
10	9	2	1	10	8	15	1	10	9	2	1	10	8	15	1	6	5	2	1	6	12	15	1
382/P S/382				383/P S/383				384/P S/384															
16	15	12	11	16	2	5	11	16	15	14	13	16	2	3	13	16	15	14	13	16	2	3	13
14	13	10	9	3	13	10	8	8	7	6	5	9	7	6	12	12	11	10	9	5	11	10	8
8	7	4	3	9	7	4	14	12	11	10	9	5	11	10	8	8	7	6	5	9	7	6	12
6	5	2	1	6	12	15	1	4	3	2	1	4	14	15	1	4	3	2	1	4	14	15	1

*** Monitor List of Correspondence between Old List and New ***

```

??: 0
1: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
25: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
49: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
73: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
97: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
121: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
145: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
169: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
193: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
217: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
241: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
265: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
289: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
313: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
337: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
361: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

```

[Count = 384/384]

I tried to find the solution number of each transformed answer by consulting with our old independent solution list of Self-complementary Magic Squares 4x4.

The results are shown in the monitor list above. I could find neither duplications nor drop-outs of any solution, but I could finally know there is the beautiful 'One-to-one Correspondence' between the two sets of solutions.

It tells us whenever you have the solution set of 384 'Prototype Squares' and 'DAM Transformation', you can always reconstruct the complete set of 384 solutions of Self-complementary Magic Squares of Order 4. It is amazing!

What does this set of Prototype Squares really mean, then? What purpose does it exist for? I can only say now it seems to act like a 'variety generator' for the object. It also seems to be the set of 'Basic Forms' for our Self-complementary Magic objects.

#2. Prototype Squares for Pan-diagonal Magic Squares of Order 4

How about for Pan-diagonal MS44? Do we have to get another set of 384 'Prototype Squares' of Order 4 corresponding to them? Do we also have to have another 'Do-it-After-the-Model Transformation'? And with both of them can we really recompose those 384 solutions of Pan-diagonal Magic Squares of Order 4?

2-1. First of all, we have to find and pick up the 'Model' out of the solution set of Pan-diagonal Magic Squares of Order 4. It is used for dictating the 'DAM Transformation' and also for making the solution set of 'Prototype Squares'.

1/P				1/S				1/C				Do you know the next three types have the same common solution set? (1) Pan-diagonal Magic Squares 4x4, (2) Complete Pan-magic Squares 4x4, (3) Composite & Complete PMS44;
1	2	3	4	1	15	14	4	1	15	4	14	
5	6	7	8	12	6	7	9	12	6	9	7	
9	10	11	12	8	10	11	5	13	3	16	2	
13	14	15	16	13	3	2	16	8	10	5	11	

Let's take the example of Complete Pan-magic Square listed right for our temporary Model. For it has as much structural similarity as possible with the original Prototype left and with the Self-complementary MS44 listed in the center.

2-2. We have to compose the new 384 Prototype Squares for Complete Pan-magic Squares 4x4 next. But can you guess what you have to do to get those solutions?

We have to learn everything from the Model solution itself. We can only compose them in the same, usual way just as we have made any ordinary magic squares, but under the new Basic Conditions and Complete Complementary Conditions below.

We just want to solve the 'inverse function' of our 'DAMT', when we suppose our DAMT to be a function for making the Prototypes into Complete Pan-magic objects.

```

/**/
/* Make 'Prototype Squares' of Order 4 */
/** Basic Forms for Prototype Squares for Self-complementary Type **
  3  2 16 13  3  2 16 13  3  2

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

** Basic Conditions for Prototype Squares 4x4: **
n1+n15+n14+n4=C ...rw1; | n1+n12+n8+n13=C ...cl1;
n12+n6+n7+n9=C ...rw2; | n15+n6+n10+n3=C ...cl2;
n8+n10+n11+n5=C ...rw3; | n14+n7+n11+n2=C ...cl3;
n13+n3+n2+n16=C ...rw4; | n4+n9+n5+n16=C ...cl4;
** Self-complementary Conditions for Prototype Squares 4x4: **
n1+n16=n15+n2=n14+n3=n4+n13=n12+n5=n6+n11=n7+n10=n9+n8=CC...scc
*/
/* Set N1 & n16=PSM-N1 */
void sstp01(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[1]=a; nm[16]=b;
uflg[a]=1; uflg[b]=1;
sstp02();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set N15 & n2=PSM-N15 */
void sstp02(){

```

```

short a,b;
for(a=16;a>0;a--){b=PSM-a;
  if((uflg[a]==0)&&(uflg[b]==0)){
    nm[15]=a; nm[2]=b;
    uflg[a]=1; uflg[b]=1;
    sstp03();
    uflg[b]=0; uflg[a]=0;}
}
}
/* Set N14 & n3=PSM-N14 */
void sstp03(){
short a,b;
for(a=16;a>0;a--){b=PSM-a;
  if((uflg[a]==0)&&(uflg[b]==0)){
    nm[14]=a; nm[3]=b;
    uflg[a]=1; uflg[b]=1;
    sstp04();
    uflg[b]=0; uflg[a]=0;}
}
}
/* Set n4=LSM-n1-n15-n14 & n13 */
void sstp04(){
short a,b;
a=LSM-nm[1]-nm[15]-nm[14];
if((0<a)&&(a<17)){
  b=LSM-nm[3]-nm[2]-nm[16];
  if(a+b==PSM){
    if((uflg[a]==0)&&(uflg[b]==0)){
      uflg[a]=1; uflg[b]=1;
      nm[4]=a; nm[13]=b;
      sstp05();
      uflg[b]=0; uflg[a]=0;}}}
}
}
/* Set N12 & n5=PSM-N12 */
void sstp05(){
short a,b;
for(a=16;a>0;a--){b=PSM-a;
  if((uflg[a]==0)&&(uflg[b]==0)){
    uflg[a]=1; uflg[b]=1;
    nm[12]=a; nm[5]=b;
    sstp06();
    uflg[b]=0; uflg[a]=0;}
}
}
/* Set N6 & n11=PSM-n6 */
void sstp06(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
  if((uflg[a]==0)&&(uflg[b]==0)){
    uflg[a]=1; uflg[b]=1;
    nm[6]=a; nm[11]=b;
    if(nm[1]+nm[6]+nm[11]+nm[16]==LSM){sstp07();}
    uflg[b]=0; uflg[a]=0;}
}
}
/* Set n8=LSM-n1-n12-n13 & n9 */
void sstp07(){
short a,b;
a=LSM-nm[1]-nm[12]-nm[13];

```

```

if((0<a)&&(a<17)){
  b=LSM-nm[16]-nm[5]-nm[4];
  if(a+b==PSM){
    if((uflg[a]==0)&&(uflg[b]==0)){
      uflg[a]=1; uflg[b]=1;
      nm[8]=a; nm[9]=b;
      sstp08();
      uflg[b]=0; uflg[a]=0;}}}}
}
/* Set n7=LSM-n12-n6-n9 & n10 */
void sstp08(){
  short a,b,c,d;
  a=LSM-nm[12]-nm[6]-nm[9];
  if((0<a)&&(a<17)){
    b=LSM-nm[8]-nm[11]-nm[5];
    if(a+b==PSM){
      c=LSM-nm[14]-nm[11]-nm[2];
      d=LSM-nm[15]-nm[6]-nm[3];
      if((a==c)&&(b==d)){
        if((uflg[a]==0)&&(uflg[b]==0)){
          nm[7]=a; nm[10]=b;
          uflg[a]=1; uflg[b]=1;
          if(nm[4]+nm[7]+nm[10]+nm[13]==LSM){recpspr();}
          uflg[b]=0; uflg[a]=0;}}}}}}
}
/**/
/* Record the Solutions and Print */
void recpspr(){
  short n;
  for(n=1;n<17;n++){d[cnt3][n]=nm[n]; pts[cnt1][n]=nm[n];}
  pts[cnt1][0]=cnt1+1;
  cnt1++; d[cnt3][0]=cnt1;
  //cnt3++; if(cnt3==6){pr6ans(); cnt3=0;}
}
/**/
/* Make another 'Prototype Squares' of Order 4 */
/** Basic Forms for Prototype Squares for Complete Pan-magic Type **
10 5 11 8 10 5 11 8 10 5

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

** Basic Conditions for Prototype Squares 4x4: **
n1+n15+n4+n14=C ...rw1; | n1+n12+n13+n8=C ...cl1;
n12+n6+n9+n7=C ...rw2; | n15+n6+n3+n10=C ...cl2;
n13+n3+n16+n2=C ...rw3; | n4+n9+n16+n5=C ...cl3;
n8+n10+n5+n11=C ...rw4; | n14+n7+n2+n11=C ...cl4;
** Complete Complementary Conditions for Prototype Squares 4x4: **
n1+n16=n15+n2=n4+n13=n14+n3=n12+n5=n6+n1=n9+n81=n7+n10=CC...ccc
*/
/* Set N1 & n16=PSM-N1 */
void cstp01(){
  short a,b;

```

```

for(a=1;a<17;a++){b=PSM-a;
  if((uflg[a]==0)&&(uflg[b]==0)){
    nm[1]=a; nm[16]=b;
    uflg[a]=1; uflg[b]=1;
    cstp02();
    uflg[b]=0; uflg[a]=0;}
}
}
/* Set N15 & n2=PSM-N15 */
void cstp02(){
  short a,b;
  for(a=16;a>0;a--){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[15]=a; nm[2]=b;
      uflg[a]=1; uflg[b]=1;
      cstp03();
      uflg[b]=0; uflg[a]=0;}
}
}
/* Set N4 & n13=PSM-N4 */
void cstp03(){
  short a,b;
  for(a=1;a<17;a++){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      uflg[a]=1; uflg[b]=1;
      nm[4]=a; nm[13]=b;
      cstp04();
      uflg[b]=0; uflg[a]=0;}
}
}
/* Set n14=LSM-n1-n15-n4 & n3 */
void cstp04(){
  short a,b;
  a=LSM-nm[1]-nm[15]-nm[4];
  if((0<a)&&(a<17)){
    b=LSM-nm[16]-nm[2]-nm[13];
    if(a+b==PSM){
      if((uflg[a]==0)&&(uflg[b]==0)){
        uflg[a]=1; uflg[b]=1;
        nm[14]=a; nm[3]=b;
        cstp05();
        uflg[b]=0; uflg[a]=0;}}}
}
/* Set N12 & n5=PSM-N12 */
void cstp05(){
  short a,b;
  for(a=16;a>0;a--){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[12]=a; nm[5]=b;
      uflg[a]=1; uflg[b]=1;
      cstp06();
      uflg[b]=0; uflg[a]=0;}
}
}
/* Set N6 & n11=PSM-N6 */
void cstp06(){
  short a,b;
  for(a=1;a<17;a++){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){

```

```

    uflg[a]=1; uflg[b]=1;
    nm[6]=a; nm[11]=b;
    if(nm[1]+nm[6]+nm[16]+nm[11]==LSM){cstp07();}
    uflg[b]=0; uflg[a]=0;}
}
}
/* Set n8=LSM-n1-n12-n13 & n9 */
void cstp07(){
short a,b;
a=LSM-nm[1]-nm[12]-nm[13];
if((0<a)&&(a<17)){
b=LSM-nm[16]-nm[5]-nm[4];
if(a+b==PSM){
if((uflg[a]==0)&&(uflg[b]==0)){
uflg[a]=1; uflg[b]=1;
nm[8]=a; nm[9]=b;
if(nm[14]+nm[9]+nm[3]+nm[8]==LSM){cstp08();}
uflg[b]=0; uflg[a]=0;}}}
}
}
/* Set n7=LSM-n12-n6-n9 & n10 */
void cstp08(){
short a,b,c,d;
a=LSM-nm[12]-nm[6]-nm[9];
if((0<a)&&(a<17)){
b=LSM-nm[5]-nm[11]-nm[8];
if(a+b==PSM){
c=LSM-nm[14]-nm[2]-nm[11];
d=LSM-nm[3]-nm[15]-nm[6];
if((a==c)&&(b==d)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[7]=a; nm[10]=b;
uflg[a]=1; uflg[b]=1;
reccpspr();
uflg[b]=0; uflg[a]=0;}}}}
}
}
/**/
/* Record the Solutions */
void reccpspr(){
short n;
for(n=1;n<17;n++){d[cnt3][n]=nm[n]; ptc[cnt2][n]=nm[n];}
ptc[cnt2][0]=cnt2+1;
cnt2++; d[cnt3][0]=cnt2;
}
}
/**/

```

Here I put the two programs of making Prototypes for both Self-complementary MS44 and Complete Pan-magic type listed as above. Compare them carefully to know how similar their styles of making Prototypes are.

The results appear very similar to each other, too. Watch the following list, and you will surely be surprised with the fact itself.

```

** Comparison of 'Prototype Squares' of Order 4 between **
** Self-Complementary Type and Complete Pan-magic Type **

```

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

4/S	4/C	5/S	5/C	6/S	6/C
1 2 5 6	1 2 5 6	1 2 9 10	1 2 9 10	1 2 9 10	1 2 9 10
9 10 13 14	9 10 13 14	3 4 11 12	3 4 11 12	5 6 13 14	5 6 13 14
3 4 7 8	3 4 7 8	5 6 13 14	5 6 13 14	3 4 11 12	3 4 11 12
11 12 15 16	11 12 15 16	7 8 15 16	7 8 15 16	7 8 15 16	7 8 15 16
7/S	7/C	8/S	8/C	9/S	9/C
1 3 2 4	1 3 2 4	1 3 2 4	1 3 2 4	1 3 5 7	1 3 5 7
5 7 6 8	5 7 6 8	9 11 10 12	9 11 10 12	2 4 6 8	2 4 6 8
9 11 10 12	9 11 10 12	5 7 6 8	5 7 6 8	9 11 13 15	9 11 13 15
13 15 14 16	13 15 14 16	13 15 14 16	13 15 14 16	10 12 14 16	10 12 14 16
10/S	10/C	11/S	11/C	12/S	12/C
1 3 5 7	1 3 5 7	1 3 9 11	1 3 9 11	1 3 9 11	1 3 9 11
9 11 13 15	9 11 13 15	2 4 10 12	2 4 10 12	5 7 13 15	5 7 13 15
2 4 6 8	2 4 6 8	5 7 13 15	5 7 13 15	2 4 10 12	2 4 10 12
10 12 14 16	10 12 14 16	6 8 14 16	6 8 14 16	6 8 14 16	6 8 14 16
13/S	13/C	14/S	14/C	15/S	15/C
1 5 2 6	1 5 2 6	1 5 2 6	1 5 2 6	1 5 3 7	1 5 3 7
3 7 4 8	3 7 4 8	9 13 10 14	9 13 10 14	2 6 4 8	2 6 4 8
9 13 10 14	9 13 10 14	3 7 4 8	3 7 4 8	9 13 11 15	9 13 11 15
11 15 12 16	11 15 12 16	11 15 12 16	11 15 12 16	10 14 12 16	10 14 12 16
16/S	16/C	17/S	17/C	18/S	18/C
1 5 3 7	1 5 3 7	1 5 9 13	1 5 9 13	1 5 9 13	1 5 9 13
9 13 11 15	9 13 11 15	2 6 10 14	2 6 10 14	3 7 11 15	3 7 11 15
2 6 4 8	2 6 4 8	3 7 11 15	3 7 11 15	2 6 10 14	2 6 10 14
10 14 12 16	10 14 12 16	4 8 12 16	4 8 12 16	4 8 12 16	4 8 12 16
19/S	19/C	20/S	20/C	21/S	21/C
1 9 2 10	1 9 2 10	1 9 2 10	1 9 2 10	1 9 3 11	1 9 3 11
3 11 4 12	3 11 4 12	5 13 6 14	5 13 6 14	2 10 4 12	2 10 4 12
5 13 6 14	5 13 6 14	3 11 4 12	3 11 4 12	5 13 7 15	5 13 7 15
7 15 8 16	7 15 8 16	7 15 8 16	7 15 8 16	6 14 8 16	6 14 8 16
22/S	22/C	23/S	23/C	24/S	24/C
1 9 3 11	1 9 3 11	1 9 5 13	1 9 5 13	1 9 5 13	1 9 5 13
5 13 7 15	5 13 7 15	2 10 6 14	2 10 6 14	3 11 7 15	3 11 7 15
2 10 4 12	2 10 4 12	3 11 7 15	3 11 7 15	2 10 6 14	2 10 6 14
6 14 8 16	6 14 8 16	4 12 8 16	4 12 8 16	4 12 8 16	4 12 8 16
25/S	25/C	26/S	26/C	27/S	27/C
2 1 4 3	2 1 4 3	2 1 4 3	2 1 4 3	2 1 6 5	2 1 6 5
6 5 8 7	6 5 8 7	10 9 12 11	10 9 12 11	4 3 8 7	4 3 8 7
10 9 12 11	10 9 12 11	6 5 8 7	6 5 8 7	10 9 14 13	10 9 14 13
14 13 16 15	14 13 16 15	14 13 16 15	14 13 16 15	12 11 16 15	12 11 16 15
28/S	28/C	29/S	29/C	30/S	30/C
2 1 6 5	2 1 6 5	2 1 10 9	2 1 10 9	2 1 10 9	2 1 10 9
10 9 14 13	10 9 14 13	4 3 12 11	4 3 12 11	6 5 14 13	6 5 14 13
4 3 8 7	4 3 8 7	6 5 14 13	6 5 14 13	4 3 12 11	4 3 12 11
12 11 16 15	12 11 16 15	8 7 16 15	8 7 16 15	8 7 16 15	8 7 16 15
31/S	31/C	32/S	32/C	33/S	33/C
2 4 1 3	2 4 1 3	2 4 1 3	2 4 1 3	2 4 6 8	2 4 6 8
6 8 5 7	6 8 5 7	10 12 9 11	10 12 9 11	1 3 5 7	1 3 5 7
10 12 9 11	10 12 9 11	6 8 5 7	6 8 5 7	10 12 14 16	10 12 14 16
14 16 13 15	14 16 13 15	14 16 13 15	14 16 13 15	9 11 13 15	9 11 13 15
34/S	34/C	35/S	35/C	36/S	36/C
2 4 6 8	2 4 6 8	2 4 10 12	2 4 10 12	2 4 10 12	2 4 10 12
10 12 14 16	10 12 14 16	1 3 9 11	1 3 9 11	6 8 14 16	6 8 14 16
1 3 5 7	1 3 5 7	6 8 14 16	6 8 14 16	1 3 9 11	1 3 9 11
9 11 13 15	9 11 13 15	5 7 13 15	5 7 13 15	5 7 13 15	5 7 13 15

37/S	37/C	38/S	38/C	39/S	39/C
2 6 1 5	2 6 1 5	2 6 1 5	2 6 1 5	2 6 4 8	2 6 4 8
4 8 3 7	4 8 3 7	10 14 9 13	10 14 9 13	1 5 3 7	1 5 3 7
10 14 9 13	10 14 9 13	4 8 3 7	4 8 3 7	10 14 12 16	10 14 12 16
12 16 11 15	12 16 11 15	12 16 11 15	12 16 11 15	9 13 11 15	9 13 11 15
40/S	40/C	41/S	41/C	42/S	42/C
2 6 4 8	2 6 4 8	2 6 10 14	2 6 10 14	2 6 10 14	2 6 10 14
10 14 12 16	10 14 12 16	1 5 9 13	1 5 9 13	4 8 12 16	4 8 12 16
1 5 3 7	1 5 3 7	4 8 12 16	4 8 12 16	1 5 9 13	1 5 9 13
9 13 11 15	9 13 11 15	3 7 11 15	3 7 11 15	3 7 11 15	3 7 11 15
43/S	43/C	44/S	44/C	45/S	45/C
2 10 1 9	2 10 1 9	2 10 1 9	2 10 1 9	2 10 4 12	2 10 4 12
4 12 3 11	4 12 3 11	6 14 5 13	6 14 5 13	1 9 3 11	1 9 3 11
6 14 5 13	6 14 5 13	4 12 3 11	4 12 3 11	6 14 8 16	6 14 8 16
8 16 7 15	8 16 7 15	8 16 7 15	8 16 7 15	5 13 7 15	5 13 7 15
46/S	46/C	47/S	47/C	48/S	48/C
2 10 4 12	2 10 4 12	2 10 6 14	2 10 6 14	2 10 6 14	2 10 6 14
6 14 8 16	6 14 8 16	1 9 5 13	1 9 5 13	4 12 8 16	4 12 8 16
1 9 3 11	1 9 3 11	4 12 8 16	4 12 8 16	1 9 5 13	1 9 5 13
5 13 7 15	5 13 7 15	3 11 7 15	3 11 7 15	3 11 7 15	3 11 7 15
49/S	49/C	50/S	50/C	51/S	51/C
3 1 4 2	3 1 4 2	3 1 4 2	3 1 4 2	3 1 7 5	3 1 7 5
7 5 8 6	7 5 8 6	11 9 12 10	11 9 12 10	4 2 8 6	4 2 8 6
11 9 12 10	11 9 12 10	7 5 8 6	7 5 8 6	11 9 15 13	11 9 15 13
15 13 16 14	15 13 16 14	15 13 16 14	15 13 16 14	12 10 16 14	12 10 16 14

[Counts = 384/S : 384/C]

Yes. They have come as just the same common sets of solutions with each other. Only the Models and 'DAM Transformations' are different from each other.

Now you can reconstruct the two solution sets of your object at the same time, if you use this common set of Prototypes and the two types of 'DAMT', as you guess.

```

/* Do-it-After-the-Model Transformation System */
/* ProtoT/ ModelS/ ModelC/
  |-----| |-----| |-----| | | | | | | | | |
  | 1| 2| 3| 4| | 1|15|14| 4| | 1|15| 4|14|
  |-----| |-----| |-----|
  | 5| 6| 7| 8| |12| 6| 7| 9| |12| 6| 9| 7|
  |-----| |-----| |-----|
  | 9|10|11|12| | 8|10|11| 5| |13| 3|16| 2|
  |-----| |-----| |-----|
  |13|14|15|16| |13| 3| 2|16| | 8|10| 5|11|
  |-----| |-----| |-----|
*/
/* Transform Prototype into Self-Complementary Object */
void damts(void){
  dm[1]=nm[1];   dm[2]=nm[15];   dm[3]=nm[14];   dm[4]=nm[4];
  dm[5]=nm[12];  dm[6]=nm[6];   dm[7]=nm[7];   dm[8]=nm[9];
  dm[9]=nm[8];   dm[10]=nm[10];  dm[11]=nm[11]; dm[12]=nm[5];
  dm[13]=nm[13]; dm[14]=nm[3];   dm[15]=nm[2];  dm[16]=nm[16];
}
/**/
/* Transform Prototype into Complete Pan-magic Object */
void damtc(void){
  dm[1]=nm[1];   dm[2]=nm[15];   dm[3]=nm[4];   dm[4]=nm[14];
  dm[5]=nm[12];  dm[6]=nm[6];   dm[7]=nm[9];   dm[8]=nm[7];
  dm[9]=nm[13];  dm[10]=nm[3];  dm[11]=nm[16]; dm[12]=nm[2];
  dm[13]=nm[8];  dm[14]=nm[10]; dm[15]=nm[5];   dm[16]=nm[11];
}

```

[Figure 5]


```

/**/
/* Print 2 Trios of Recomposed Solutions by DAMT [Primitive Set] */
void prp2trios(short x){
  short l,l4,m,n,p;
  for(m=0;m<x;m=m+2){
    for(n=1;n<17;n++){nm[n]=pts[m][n];}
    damts(); for(n=1;n<17;n++){d[0][n]=dm[n];}
    for(n=1;n<17;n++){nm[n]=ptc[m][n];}
    damtc(); for(n=1;n<17;n++){d[1][n]=dm[n];}
    for(n=1;n<17;n++){nm[n]=pts[m+1][n];}
    damts(); for(n=1;n<17;n++){d[2][n]=dm[n];}
    for(n=1;n<17;n++){nm[n]=ptc[m+1][n];}
    damtc(); for(n=1;n<17;n++){d[3][n]=dm[n];}
    printf("%10d/P      /S      /C%12d/P      /S      /C\n",
      pts[m][0],pts[m+1][0]);
    for(l=0;l<4;l++){l4=l*4;
      for(p=1;p<5;p++){printf("%3d",pts[m][l4+p]);}
      printf(" ");
      for(p=1;p<5;p++){printf("%3d",d[0][l4+p]);}
      printf(" ");
      for(p=1;p<5;p++){printf("%3d",d[1][l4+p]);}
      printf(" ");
      for(p=1;p<5;p++){printf("%3d",pts[m+1][l4+p]);}
      printf(" ");
      for(p=1;p<5;p++){printf("%3d",d[2][l4+p]);}
      printf(" ");
      for(p=1;p<5;p++){printf("%3d",d[3][l4+p]);}
      printf("\n");
    }
  }
}
/**/

```

**** Two Types of Magic Squares 4x4 Recomposed by 'DAMT' from Prototypes: ****
*** Primitive Sets: Prototype/P, Self-complementary/S, Complete Pan-magic/C ***

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

11/P			/S			/C			12/P			/S			/C								
1	3	9	11	1	14	8	11	1	14	11	8	1	3	9	11	1	14	8	11	1	14	11	8
2	4	10	12	15	4	10	5	15	4	5	10	5	7	13	15	12	7	13	2	12	7	2	13
5	7	13	15	12	7	13	2	6	9	16	3	2	4	10	12	15	4	10	5	6	9	16	3
6	8	14	16	6	9	3	16	12	7	2	13	6	8	14	16	6	9	3	16	15	4	5	10
13/P			/S			/C			14/P			/S			/C								
1	5	2	6	1	12	15	6	1	12	6	15	1	5	2	6	1	12	15	6	1	12	6	15
3	7	4	8	14	7	4	9	14	7	9	4	9	13	10	14	8	13	10	3	8	13	3	10
9	13	10	14	8	13	10	3	11	2	16	5	3	7	4	8	14	7	4	9	11	2	16	5
11	15	12	16	11	2	5	16	8	13	3	10	11	15	12	16	11	2	5	16	14	7	9	4
15/P			/S			/C			16/P			/S			/C								
1	5	3	7	1	12	14	7	1	12	7	14	1	5	3	7	1	12	14	7	1	12	7	14
2	6	4	8	15	6	4	9	15	6	9	4	9	13	11	15	8	13	11	2	8	13	2	11
9	13	11	15	8	13	11	2	10	3	16	5	2	6	4	8	15	6	4	9	10	3	16	5
10	14	12	16	10	3	5	16	8	13	2	11	10	14	12	16	10	3	5	16	15	6	9	4
17/P			/S			/C			18/P			/S			/C								
1	5	9	13	1	12	8	13	1	12	13	8	1	5	9	13	1	12	8	13	1	12	13	8
2	6	10	14	15	6	10	3	15	6	3	10	3	7	11	15	14	7	11	2	14	7	2	11
3	7	11	15	14	7	11	2	4	9	16	5	2	6	10	14	15	6	10	3	4	9	16	5
4	8	12	16	4	9	5	16	14	7	2	11	4	8	12	16	4	9	5	16	15	6	3	10
19/P			/S			/C			20/P			/S			/C								
1	9	2	10	1	8	15	10	1	8	10	15	1	9	2	10	1	8	15	10	1	8	10	15
3	11	4	12	14	11	4	5	14	11	5	4	5	13	6	14	12	13	6	3	12	13	3	6
5	13	6	14	12	13	6	3	7	2	16	9	3	11	4	12	14	11	4	5	7	2	16	9
7	15	8	16	7	2	9	16	12	13	3	6	7	15	8	16	7	2	9	16	14	11	5	4
21/P			/S			/C			22/P			/S			/C								
1	9	3	11	1	8	14	11	1	8	11	14	1	9	3	11	1	8	14	11	1	8	11	14
2	10	4	12	15	10	4	5	15	10	5	4	5	13	7	15	12	13	7	2	12	13	2	7
5	13	7	15	12	13	7	2	6	3	16	9	2	10	4	12	15	10	4	5	6	3	16	9
6	14	8	16	6	3	9	16	12	13	2	7	6	14	8	16	6	3	9	16	15	10	5	4
23/P			/S			/C			24/P			/S			/C								
1	9	5	13	1	8	12	13	1	8	13	12	1	9	5	13	1	8	12	13	1	8	13	12
2	10	6	14	15	10	6	3	15	10	3	6	3	11	7	15	14	11	7	2	14	11	2	7
3	11	7	15	14	11	7	2	4	5	16	9	2	10	6	14	15	10	6	3	4	5	16	9
4	12	8	16	4	5	9	16	14	11	2	7	4	12	8	16	4	5	9	16	15	10	3	6
25/P			/S			/C			26/P			/S			/C								
2	1	4	3	2	16	13	3	2	16	3	13	2	1	4	3	2	16	13	3	2	16	3	13
6	5	8	7	11	5	8	10	11	5	10	8	10	9	12	11	7	9	12	6	7	9	6	12
10	9	12	11	7	9	12	6	14	4	15	1	6	5	8	7	11	5	8	10	14	4	15	1
14	13	16	15	14	4	1	15	7	9	6	12	14	13	16	15	14	4	1	15	11	5	10	8
27/P			/S			/C			28/P			/S			/C								
2	1	6	5	2	16	11	5	2	16	5	11	2	1	6	5	2	16	11	5	2	16	5	11
4	3	8	7	13	3	8	10	13	3	10	8	10	9	14	13	7	9	14	4	7	9	4	14
10	9	14	13	7	9	14	4	12	6	15	1	4	3	8	7	13	3	8	10	12	6	15	1
12	11	16	15	12	6	1	15	7	9	4	14	12	11	16	15	12	6	1	15	13	3	10	8
29/P			/S			/C			30/P			/S			/C								
2	1	10	9	2	16	7	9	2	16	9	7	2	1	10	9	2	16	7	9	2	16	9	7
4	3	12	11	13	3	12	6	13	3	6	12	6	5	14	13	11	5	14	4	11	5	4	14
6	5	14	13	11	5	14	4	8	10	15	1	4	3	12	11	13	3	12	6	8	10	15	1
8	7	16	15	8	10	1	15	11	5	4	14	8	7	16	15	8	10	1	15	13	3	6	12
31/P			/S			/C			32/P			/S			/C								
2	4	1	3	2	13	16	3	2	13	3	16	2	4	1	3	2	13	16	3	2	13	3	16
6	8	5	7	11	8	5	10	11	8	10	5	10	12	9	11	7	12	9	6	7	12	6	9
10	12	9	11	7	12	9	6	14	1	15	4	6	8	5	7	11	8	5	10	14	1	15	4
14	16	13	15	14	1	4	15	7	12	6	9	14	16	13	15	14	1	4	15	11	8	10	5

2 4 6 8	2 13 11 8	2 13 8 11	2 4 6 8	2 13 11 8	2 13 8 11
1 3 5 7	16 3 5 10	16 3 10 5	10 12 14 16	7 12 14 1	7 12 1 14
10 12 14 16	7 12 14 1	9 6 15 4	1 3 5 7	16 3 5 10	9 6 15 4
9 11 13 15	9 6 4 15	7 12 1 14	9 11 13 15	9 6 4 15	16 3 10 5
33/P	/S	/C	34/P	/S	/C
2 4 10 12	2 13 7 12	2 13 12 7	2 4 10 12	2 13 7 12	2 13 12 7
1 3 9 11	16 3 9 6	16 3 6 9	6 8 14 16	11 8 14 1	11 8 1 14
6 8 14 16	11 8 14 1	5 10 15 4	1 3 9 11	16 3 9 6	5 10 15 4
5 7 13 15	5 10 4 15	11 8 1 14	5 7 13 15	5 10 4 15	16 3 6 9
35/P	/S	/C	36/P	/S	/C
2 6 1 5	2 11 16 5	2 11 5 16	2 6 1 5	2 11 16 5	2 11 5 16
4 8 3 7	13 8 3 10	13 8 10 3	10 14 9 13	7 14 9 4	7 14 4 9
10 14 9 13	7 14 9 4	12 1 15 6	4 8 3 7	13 8 3 10	12 1 15 6
12 16 11 15	12 1 6 15	7 14 4 9	12 16 11 15	12 1 6 15	13 8 10 3
37/P	/S	/C	38/P	/S	/C
2 6 4 8	2 11 13 8	2 11 8 13	2 6 4 8	2 11 13 8	2 11 8 13
1 5 3 7	16 5 3 10	16 5 10 3	10 14 12 16	7 14 12 1	7 14 1 12
10 14 12 16	7 14 12 1	9 4 15 6	1 5 3 7	16 5 3 10	9 4 15 6
9 13 11 15	9 4 6 15	7 14 1 12	9 13 11 15	9 4 6 15	16 5 10 3
39/P	/S	/C	40/P	/S	/C
2 6 10 14	2 11 7 14	2 11 14 7	2 6 10 14	2 11 7 14	2 11 14 7
1 5 9 13	16 5 9 4	16 5 4 9	4 8 12 16	13 8 12 1	13 8 1 12
4 8 12 16	13 8 12 1	3 10 15 6	1 5 9 13	16 5 9 4	3 10 15 6
3 7 11 15	3 10 6 15	13 8 1 12	3 7 11 15	3 10 6 15	16 5 4 9
41/P	/S	/C	42/P	/S	/C
2 10 1 9	2 7 16 9	2 7 9 16	2 10 1 9	2 7 16 9	2 7 9 16
4 12 3 11	13 12 3 6	13 12 6 3	6 14 5 13	11 14 5 4	11 14 4 5
6 14 5 13	11 14 5 4	8 1 15 10	4 12 3 11	13 12 3 6	8 1 15 10
8 16 7 15	8 1 10 15	11 14 4 5	8 16 7 15	8 1 10 15	13 12 6 3
43/P	/S	/C	44/P	/S	/C
2 10 4 12	2 7 13 12	2 7 12 13	2 10 4 12	2 7 13 12	2 7 12 13
1 9 3 11	16 9 3 6	16 9 6 3	6 14 8 16	11 14 8 1	11 14 1 8
6 14 8 16	11 14 8 1	5 4 15 10	1 9 3 11	16 9 3 6	5 4 15 10
5 13 7 15	5 4 10 15	11 14 1 8	5 13 7 15	5 4 10 15	16 9 6 3
45/P	/S	/C	46/P	/S	/C
2 10 6 14	2 7 11 14	2 7 14 11	2 10 6 14	2 7 11 14	2 7 14 11
1 9 5 13	16 9 5 4	16 9 4 5	4 12 8 16	13 12 8 1	13 12 1 8
4 12 8 16	13 12 8 1	3 6 15 10	1 9 5 13	16 9 5 4	3 6 15 10
3 11 7 15	3 6 10 15	13 12 1 8	3 11 7 15	3 6 10 15	16 9 4 5
47/P	/S	/C	48/P	/S	/C
3 1 4 2	3 16 13 2	3 16 2 13	3 1 4 2	3 16 13 2	3 16 2 13
7 5 8 6	10 5 8 11	10 5 11 8	11 9 12 10	6 9 12 7	6 9 7 12
11 9 12 10	6 9 12 7	15 4 14 1	7 5 8 6	10 5 8 11	15 4 14 1
15 13 16 14	15 4 1 14	6 9 7 12	15 13 16 14	15 4 1 14	10 5 11 8

[Counts = 384/P, 384/S, 384/C]

The new list above shows the result of my newest experiment with the new set of common Prototype Squares and the two DAM Transformations dictated as above.

2-3. Let's have another experiment here to try to make the new Prototype Squares under the new Model and finally reproduce the 48 Standard Solutions of Complete Pan-magic Squares of Order 4 with the new Do-it-After-the-Model Transformation.

Do you think we can really reproduce that familiar set of standard solutions?

```

/* Make 'Prototype Squares' 4x4 and Apply 'DAMT' */
/*
    [Basic Form]                [New Model]
15 16 13 14 15 16 13 14   5 11 14 4 5 11 14 4
3  4 | 1 | 2 | 3 | 4 | 1  2  10  8 | 1 |15|10| 8 | 1 15
   |---+---+---+---|
7  8 | 5 | 6 | 7 | 8 | 5  6   3 13 |12| 6 | 3 |13|12  6
   |---+---+---+---|
11 12| 9 |10|11|12| 9 10  16  2 | 7 | 9 |16| 2 | 7  9
   |---+---+---+---|
15 16|13|14|15|16|13 14   5 11|14| 4 | 5 |11|14  4
   |---+---+---+---|
3  4  1  2  3  4  1  2  10  8  1 15 10  8  1 15
[Basic Definitions]
n1+n15+n10+n8=34 ...rw1; | n1+n12+n7+n14 ...cl1;
n12+n6+n3+n13=34 ...rw2; | n15+n6+n9+n4 ...cl2;
n7+n9+n16+n2=34 ...rw3; | n10+n3+n16+n5 ...cl3;
n14+n4+n5+n11=34 ...rw4; | n8+n13+n2+n11 ...cl4;
n1+n16=n2+n15=n3+n14=n4+n13=n5+n12=n6=n11=n7+n10=n8+n9=17 ...cmpltcc
n1<n14; n1<n8; n1<n11; and n15<n12;
*/
/* Set N1 & n16=PSM-N1 */
void stp01(){
    short a,b;
    for(a=1;a<17;a++){b=PSM-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            uflg[a]=1; uflg[b]=1;
            nm[1]=a; nm[16]=b;
            stp02();
            uflg[b]=0; uflg[a]=0;}
    }
}
/* Set N15 & n2=PSM-N15 */
void stp02(){
    short a,b;
    for(a=16;a>0;a--){b=PSM-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            uflg[a]=1; uflg[b]=1;
            nm[15]=a; nm[2]=b;
            stp03();
            uflg[b]=0; uflg[a]=0;}
    }
}
/* Set N10 & n7=PSM-N10 */
void stp03(){
    short a,b;
    for(a=16;a>0;a--){b=PSM-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            uflg[a]=1; uflg[b]=1;
            nm[10]=a; nm[7]=b;
            stp04();
            uflg[b]=0; uflg[a]=0;}
    }
}
/* Set n8=LSM-n1-n15-n10 & n8>N1 */
/* Set n9=PSM-n8 */
void stp04(){
    short a,b;
    a=LSM-nm[1]-nm[15]-nm[10];

```

```

if((nm[1]<a)&&(a<17)){
  b=LSM-nm[16]-nm[2]-nm[7];
  if(a+b==PSM){
    if((uflg[a]==0)&&(uflg[b]==0)){
      uflg[a]=1; uflg[b]=1;
      nm[8]=a; nm[9]=b;
      stp05();
      uflg[b]=0; uflg[a]=0;}}
}
/* Set N12(<N15) & n5=PSM-N12 */
void stp05(){
  short a,b;
  for(a=nm[15]-1;a>0;a--){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      uflg[a]=1; uflg[b]=1;
      nm[12]=a; nm[5]=b;
      stp06();
      uflg[b]=0; uflg[a]=0;}
  }
}
/* Set n14=LSM-n1-n12-n7 & n14>N1 */
/* Set n9=PSM-n14 */
void stp06(){
  short a,b;
  a=LSM-nm[1]-nm[12]-nm[7];
  if((nm[1]<a)&&(a<17)){
    b=LSM-nm[16]-nm[5]-nm[10];
    if(a+b==PSM){
      if((uflg[a]==0)&&(uflg[b]==0)){
        uflg[a]=1; uflg[b]=1;
        nm[14]=a; nm[3]=b;
        if(nm[8]+nm[3]+nm[9]+nm[14]==LSM){stp07();}
        uflg[b]=0; uflg[a]=0;}}
  }
}
/* Set N6 & n11=PSM-N6 & n11>N1 */
void stp07(){
  short a,b;
  for(a=1;a<17;a++){b=PSM-a;
    if((b>nm[1])&&(uflg[a]==0)&&(uflg[b]==0)){
      uflg[a]=1; uflg[b]=1;
      nm[6]=a; nm[11]=b;
      if(nm[1]+nm[6]+nm[16]+nm[11]==LSM){stp08();}
      uflg[b]=0; uflg[a]=0;}
  }
}
/* Set n4=LSM-n12-n6-n9 & n13=PSM-n4 */
void stp08(){
  short a,b,c,d;
  a=LSM-nm[15]-nm[6]-nm[9];
  if((0<a)&&(a<17)){
    b=LSM-nm[2]-nm[11]-nm[8];
    if(a+b==PSM){
      c=LSM-nm[14]-nm[5]-nm[11];
      d=LSM-nm[3]-nm[12]-nm[6];
      if((a==c)&&(b==d)){
        if((uflg[a]==0)&&(uflg[b]==0)){
          uflg[a]=1; uflg[b]=1;
          nm[4]=a; nm[13]=b;
          prtrecord();
        }
      }
    }
  }
}

```

```

        uflg[b]=0; uflg[a]=0;}}}}
}
/**/
/* Record the Prototype Solutions */
void prtrecord(){
    short n;
    pt[cnt1][0]=cnt1+1;
    for(n=1;n<17;n++){pt[cnt1][n]=nm[n];}
    cnt1++;
}
/**/
/* Print the List of Solutions */
void prlist(short x){
    short m,n,c2;
    for(m=0;m<x;m++){
        c2=cnt3*2;
        //dt[c2][0]=pt[m][0];
        dt[c2][0]=m+1;
        for(n=1;n<17;n++){nm[n]=pt[m][n]; dt[c2][n]=nm[n];}
        damt();
        for(n=1;n<17;n++){dt[c2+1][n]=dnm[n];}
        n=findnmb(); dt[c2+1][0]=n; lnm[n]++;
        cnt3++; if(cnt3==3){pr6ans(); cnt3=0;}
    }
}
/**/
/* Print 3 Pairs of Solutions */
void pr6ans(){
    short l,l4,m,n;
    printf("%10d/P%4d/C%19d/P%4d/C%19d/P%4d/C\n",
        dt[0][0],dt[1][0],dt[2][0],dt[3][0],dt[4][0],dt[5][0]);
    for(l=0;l<4;l++){l4=l*4;
        for(m=0;m<6;m++){
            for(n=1;n<5;n++){printf("%3d",dt[m][l4+n]);}
            if(m<5){
                if(m%2==1){printf(" ");}else{printf(" ");}}
        }
        printf("\n");
    }
}
/**/
/*
        Proto/
        -----
    3  4| 1| 2| 3| 4| 1  2  10  8| 1|15|10| 8| 1 15
        |-----|
    7  8| 5| 6| 7| 8| 5  6   3 13|12| 6| 3|13|12  6
        |-----|
    11 12| 9|10|11|12| 9 10  16  2| 7| 9|16| 2| 7  9
        |-----|
    15 16|13|14|15|16|13 14   5 11|14| 4| 5|11|14  4
        |-----|
*/
/* Transform Prototype into 'Complete' Object */
void damt(void){
    dnm[1]=nm[1];   dnm[2]=nm[15];   dnm[3]=nm[10];   dnm[4]=nm[8];
    dnm[5]=nm[12];  dnm[6]=nm[6];    dnm[7]=nm[3];    dnm[8]=nm[13];
    dnm[9]=nm[7];   dnm[10]=nm[9];   dnm[11]=nm[16];  dnm[12]=nm[2];
    dnm[13]=nm[14]; dnm[14]=nm[4];   dnm[15]=nm[5];   dnm[16]=nm[11];
}
/**/

```

** Prototype Squares of Order 4 and 'Complete' Pan-magic Squares: **

** Recomposed by DAMT: Standard Solutions Listed by Kanji Setsuda **

1/P				6/C				2/P				4/C				3/P				5/C			
1	2	3	4	1	15	10	8	1	2	3	4	1	15	6	12	1	2	5	6	1	15	10	8
5	6	7	8	12	6	3	13	9	10	11	12	8	10	3	13	3	4	7	8	14	4	5	11
9	10	11	12	7	9	16	2	5	6	7	8	11	5	16	2	9	10	13	14	7	9	16	2
13	14	15	16	14	4	5	11	13	14	15	16	14	4	9	7	11	12	15	16	12	6	3	13
4/P				2/C				5/P				3/C				6/P				1/C			
1	2	5	6	1	15	4	14	1	2	9	10	1	15	6	12	1	2	9	10	1	15	4	14
9	10	13	14	8	10	5	11	3	4	11	12	14	4	9	7	5	6	13	14	12	6	9	7
3	4	7	8	13	3	16	2	5	6	13	14	11	5	16	2	3	4	11	12	13	3	16	2
11	12	15	16	12	6	9	7	7	8	15	16	8	10	3	13	7	8	15	16	8	10	5	11
7/P				10/C				8/P				9/C				9/P				8/C			
1	3	2	4	1	14	11	8	1	3	2	4	1	14	7	12	1	3	5	7	1	14	4	15
5	7	6	8	12	7	2	13	9	11	10	12	8	11	2	13	9	11	13	15	8	11	5	10
9	11	10	12	6	9	16	3	5	7	6	8	10	5	16	3	2	4	6	8	13	2	16	3
13	15	14	16	15	4	5	10	13	15	14	16	15	4	9	6	10	12	14	16	12	7	9	6
10/P				7/C				11/P				12/C				12/P				11/C			
1	3	9	11	1	14	4	15	1	5	2	6	1	12	7	14	1	5	3	7	1	12	6	15
5	7	13	15	12	7	9	6	9	13	10	14	8	13	2	11	9	13	11	15	8	13	3	10
2	4	10	12	13	2	16	3	3	7	4	8	10	3	16	5	2	6	4	8	11	2	16	5
6	8	14	16	8	11	5	10	11	15	12	16	15	6	9	4	10	14	12	16	14	7	9	4
13/P				18/C				14/P				16/C				15/P				17/C			
2	1	4	3	2	16	9	7	2	1	4	3	2	16	5	11	2	1	6	5	2	16	9	7
6	5	8	7	11	5	4	14	10	9	12	11	7	9	4	14	4	3	8	7	13	3	6	12
10	9	12	11	8	10	15	1	6	5	8	7	12	6	15	1	10	9	14	13	8	10	15	1
14	13	16	15	13	3	6	12	14	13	16	15	13	3	10	8	12	11	16	15	11	5	4	14
16/P				14/C				17/P				15/C				18/P				13/C			
2	1	6	5	2	16	3	13	2	1	10	9	2	16	5	11	2	1	10	9	2	16	3	13
10	9	14	13	7	9	6	12	4	3	12	11	13	3	10	8	6	5	14	13	11	5	10	8
4	3	8	7	14	4	15	1	6	5	14	13	12	6	15	1	4	3	12	11	14	4	15	1
12	11	16	15	11	5	10	8	8	7	16	15	7	9	4	14	8	7	16	15	7	9	6	12
19/P				22/C				20/P				21/C				21/P				20/C			
2	4	1	3	2	13	12	7	2	4	1	3	2	13	8	11	2	4	6	8	2	13	3	16
6	8	5	7	11	8	1	14	10	12	9	11	7	12	1	14	10	12	14	16	7	12	6	9
10	12	9	11	5	10	15	4	6	8	5	7	9	6	15	4	1	3	5	7	14	1	15	4
14	16	13	15	16	3	6	9	14	16	13	15	16	3	10	5	9	11	13	15	11	8	10	5
22/P				19/C				23/P				24/C				24/P				23/C			
2	4	10	12	2	13	3	16	2	6	1	5	2	11	8	13	2	6	4	8	2	11	5	16
6	8	14	16	11	8	10	5	10	14	9	13	7	14	1	12	10	14	12	16	7	14	4	9
1	3	9	11	14	1	15	4	4	8	3	7	9	4	15	6	1	5	3	7	12	1	15	6
5	7	13	15	7	12	6	9	12	16	11	15	16	5	10	3	9	13	11	15	13	8	10	3
25/P				30/C				26/P				28/C				27/P				29/C			
3	1	4	2	3	16	9	6	3	1	4	2	3	16	5	10	3	1	7	5	3	16	9	6
7	5	8	6	10	5	4	15	11	9	12	10	6	9	4	15	4	2	8	6	13	2	7	12
11	9	12	10	8	11	14	1	7	5	8	6	12	7	14	1	11	9	15	13	8	11	14	1
15	13	16	14	13	2	7	12	15	13	16	14	13	2	11	8	12	10	16	14	10	5	4	15
28/P				26/C				29/P				27/C				30/P				25/C			
3	1	7	5	3	16	2	13	3	1	11	9	3	16	5	10	3	1	11	9	3	16	2	13
11	9	15	13	6	9	7	12	4	2	12	10	13	2	11	8	7	5	15	13	10	5	11	8
4	2	8	6	15	4	14	1	7	5	15	13	12	7	14	1	4	2	12	10	15	4	14	1
12	10	16	14	10	5	11	8	8	6	16	14	6	9	4	15	8	6	16	14	6	9	7	12
31/P				34/C				32/P				33/C				33/P				32/C			
3	4	1	2	3	13	12	6	3	4	1	2	3	13	8	10	3	4	7	8	3	13	2	16
7	8	5	6	10	8	1	15	11	12	9	10	6	12	1	15	11	12	15	16	6	12	7	9

11 12 9 10	5 11 14 4	7 8 5 6	9 7 14 4	1 2 5 6	15 1 14 4
15 16 13 14	16 2 7 9	15 16 13 14	16 2 11 5	9 10 13 14	10 8 11 5
34/P	31/C	35/P	40/C	36/P	38/C
3 4 11 12	3 13 2 16	4 2 3 1	4 15 10 5	4 2 3 1	4 15 6 9
7 8 15 16	10 8 11 5	8 6 7 5	9 6 3 16	12 10 11 9	5 10 3 16
1 2 9 10	15 1 14 4	12 10 11 9	7 12 13 2	8 6 7 5	11 8 13 2
5 6 13 14	6 12 7 9	16 14 15 13	14 1 8 11	16 14 15 13	14 1 12 7
37/P	39/C	38/P	36/C	39/P	37/C
4 2 8 6	4 15 10 5	4 2 8 6	4 15 1 14	4 2 12 10	4 15 6 9
3 1 7 5	14 1 8 11	12 10 16 14	5 10 8 11	3 1 11 9	14 1 12 7
12 10 16 14	7 12 13 2	3 1 7 5	16 3 13 2	8 6 16 14	11 8 13 2
11 9 15 13	9 6 3 16	11 9 15 13	9 6 12 7	7 5 15 13	5 10 3 16
40/P	35/C	41/P	44/C	42/P	43/C
4 2 12 10	4 15 1 14	4 3 2 1	4 14 11 5	4 3 2 1	4 14 7 9
8 6 16 14	9 6 12 7	8 7 6 5	9 7 2 16	12 11 10 9	5 11 2 16
3 1 11 9	16 3 13 2	12 11 10 9	6 12 13 3	8 7 6 5	10 8 13 3
7 5 15 13	5 10 8 11	16 15 14 13	15 1 8 10	16 15 14 13	15 1 12 6
43/P	42/C	44/P	41/C	45/P	46/C
4 3 8 7	4 14 1 15	4 3 12 11	4 14 1 15	5 1 6 2	5 16 3 10
12 11 16 15	5 11 8 10	8 7 16 15	9 7 12 6	13 9 14 10	4 9 6 15
2 1 6 5	16 2 13 3	2 1 10 9	16 2 13 3	7 3 8 4	14 7 12 1
10 9 14 13	9 7 12 6	6 5 14 13	5 11 8 10	15 11 16 12	11 2 13 8
46/P	45/C	47/P	48/C	48/P	47/C
5 1 7 3	5 16 2 11	6 2 5 1	6 15 4 9	6 2 8 4	6 15 1 12
13 9 15 11	4 9 7 14	14 10 13 9	3 10 5 16	14 10 16 12	3 10 8 13
6 2 8 4	15 6 12 1	8 4 7 3	13 8 11 2	5 1 7 3	16 5 11 2
14 10 16 12	10 3 13 8	16 12 15 11	12 1 14 7	13 9 15 11	9 4 14 7

[Count = 48/48]

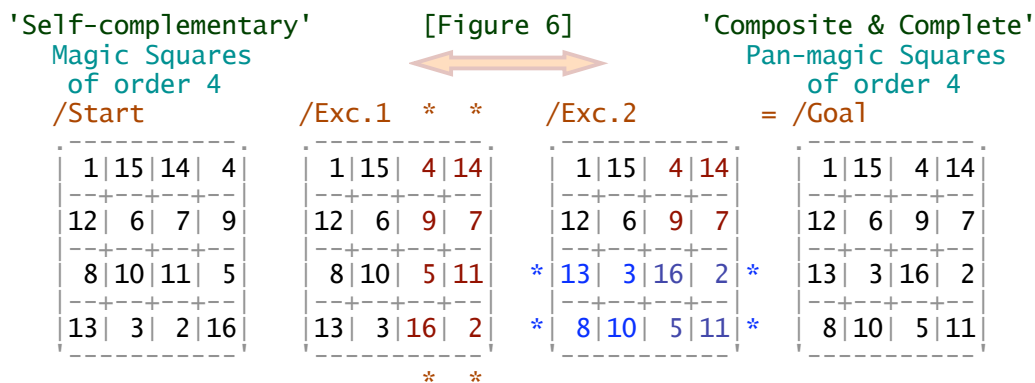
** Monitor List of Correspondence between Old and New **

?: 0
 1: 1
 25: 1

We could successfully reproduce the complete set of 48 standard solutions again and find the one-to-one correspondence between the Prototypes and the Complete Pan-magic squares. It is amazing. But why? What do they really mean?
 The set of Prototype Squares looks like the complete set of 'Basic Forms'.

#3. Another Way of Making our Objects by Prototype Squares and 'DAMT'

We already know how to transform Self-complementary MS44 into Complete Pan-magic ones or how to remake the latter back into the former.



The only things you have to do are:

- (1) Exchange the whole 4 entries between the 3rd and 4th columns.
- (2) Exchange the whole 4 entries between the 3rd and 4th rows.

As the result we can get a Complete Pan-magic square from a Self-complementary one. We can actually get a 'Composite & Complete' Pan-magic square 4x4, I say. All complementary pairs of 17 are located on the pan-diagonals, nothing elsewhere. Every 4 entries of any 2 x 2 block within the object also add up to same total 34.

We can take this 'type converter' and add it to our job list now.

- (1) At first let's make 384 solutions of Self-complementary Squares by the 384 Prototype Squares and 'DAM Transformation' in the same way as we just made.
- (2) And let's make the 384 'Composite & Complete' Pan-magic squares from the 384 Self-complementary ones by this type converter.

Why don't you do this experiment? Yes, let's try it now right here. The next long list shows the result of my recent experiment.

[Prototype Squares 4x4 and Re-compositions of Self-complementary(/S) and Complete(/C) Pan-magic Squares 4x4 Recomposed by serial Transformations]

[1] 1/P 1/S 1/C	[2] 2/P 2/S 2/C
1 2 3 4 1 15 14 4 1 15 4 14	1 2 3 4 1 15 14 4 1 15 4 14
5 6 7 8 12 6 7 9 12 6 9 7	9 10 11 12 8 10 11 5 8 10 5 11
9 10 11 12 8 10 11 5 13 3 16 2	5 6 7 8 12 6 7 9 13 3 16 2
13 14 15 16 13 3 2 16 8 10 5 11	13 14 15 16 13 3 2 16 12 6 9 7
[3] 3/P 3/S 3/C	[4] 4/P 4/S 4/C
1 2 5 6 1 15 12 6 1 15 6 12	1 2 5 6 1 15 12 6 1 15 6 12
3 4 7 8 14 4 7 9 14 4 9 7	9 10 13 14 8 10 13 3 8 10 3 13
9 10 13 14 8 10 13 3 11 5 16 2	3 4 7 8 14 4 7 9 11 5 16 2
11 12 15 16 11 5 2 16 8 10 3 13	11 12 15 16 11 5 2 16 14 4 9 7
[5] 5/P 5/S 5/C	[6] 6/P 6/S 6/C
1 2 9 10 1 15 8 10 1 15 10 8	1 2 9 10 1 15 8 10 1 15 10 8
3 4 11 12 14 4 11 5 14 4 5 11	5 6 13 14 12 6 13 3 12 6 3 13
5 6 13 14 12 6 13 3 7 9 16 2	3 4 11 12 14 4 11 5 7 9 16 2
7 8 15 16 7 9 2 16 12 6 3 13	7 8 15 16 7 9 2 16 14 4 5 11
[7] 7/P 7/S 7/C	[8] 8/P 8/S 8/C
1 3 2 4 1 14 15 4 1 14 4 15	1 3 2 4 1 14 15 4 1 14 4 15
5 7 6 8 12 7 6 9 12 7 9 6	9 11 10 12 8 11 10 5 8 11 5 10
9 11 10 12 8 11 10 5 13 2 16 3	5 7 6 8 12 7 6 9 13 2 16 3
13 15 14 16 13 2 3 16 8 11 5 10	13 15 14 16 13 2 3 16 12 7 9 6
[9] 9/P 9/S 9/C	[10] 10/P 10/S 10/C
1 3 5 7 1 14 12 7 1 14 7 12	1 3 5 7 1 14 12 7 1 14 7 12
2 4 6 8 15 4 6 9 15 4 9 6	9 11 13 15 8 11 13 2 8 11 2 13
9 11 13 15 8 11 13 2 10 5 16 3	2 4 6 8 15 4 6 9 10 5 16 3
10 12 14 16 10 5 3 16 8 11 2 13	10 12 14 16 10 5 3 16 15 4 9 6
[11] 11/P 11/S 11/C	[12] 12/P 12/S 12/C
1 3 9 11 1 14 8 11 1 14 11 8	1 3 9 11 1 14 8 11 1 14 11 8
2 4 10 12 15 4 10 5 15 4 5 10	5 7 13 15 12 7 13 2 12 7 2 13
5 7 13 15 12 7 13 2 6 9 16 3	2 4 10 12 15 4 10 5 6 9 16 3
6 8 14 16 6 9 3 16 12 7 2 13	6 8 14 16 6 9 3 16 15 4 5 10
[13] 13/P 13/S 13/C	[14] 14/P 14/S 14/C
1 5 2 6 1 12 15 6 1 12 6 15	1 5 2 6 1 12 15 6 1 12 6 15
3 7 4 8 14 7 4 9 14 7 9 4	9 13 10 14 8 13 10 3 8 13 3 10
9 13 10 14 8 13 10 3 11 2 16 5	3 7 4 8 14 7 4 9 11 2 16 5
11 15 12 16 11 2 5 16 8 13 3 10	11 15 12 16 11 2 5 16 14 7 9 4

[15]	15/P	15/S	15/C	[16]	16/P	16/S	16/C
1	5 3 7	1 12 14 7	1 12 7 14	1	5 3 7	1 12 14 7	1 12 7 14
2	6 4 8	15 6 4 9	15 6 9 4	9	13 11 15	8 13 11 2	8 13 2 11
9	13 11 15	8 13 11 2	10 3 16 5	2	6 4 8	15 6 4 9	10 3 16 5
10	14 12 16	10 3 5 16	8 13 2 11	10	14 12 16	10 3 5 16	15 6 9 4
[17]	17/P	17/S	17/C	[18]	18/P	18/S	18/C
1	5 9 13	1 12 8 13	1 12 13 8	1	5 9 13	1 12 8 13	1 12 13 8
2	6 10 14	15 6 10 3	15 6 3 10	3	7 11 15	14 7 11 2	14 7 2 11
3	7 11 15	14 7 11 2	4 9 16 5	2	6 10 14	15 6 10 3	4 9 16 5
4	8 12 16	4 9 5 16	14 7 2 11	4	8 12 16	4 9 5 16	15 6 3 10
[19]	19/P	19/S	19/C	[20]	20/P	20/S	20/C
1	9 2 10	1 8 15 10	1 8 10 15	1	9 2 10	1 8 15 10	1 8 10 15
3	11 4 12	14 11 4 5	14 11 5 4	5	13 6 14	12 13 6 3	12 13 3 6
5	13 6 14	12 13 6 3	7 2 16 9	3	11 4 12	14 11 4 5	7 2 16 9
7	15 8 16	7 2 9 16	12 13 3 6	7	15 8 16	7 2 9 16	14 11 5 4
[21]	21/P	21/S	21/C	[22]	22/P	22/S	22/C
1	9 3 11	1 8 14 11	1 8 11 14	1	9 3 11	1 8 14 11	1 8 11 14
2	10 4 12	15 10 4 5	15 10 5 4	5	13 7 15	12 13 7 2	12 13 2 7
5	13 7 15	12 13 7 2	6 3 16 9	2	10 4 12	15 10 4 5	6 3 16 9
6	14 8 16	6 3 9 16	12 13 2 7	6	14 8 16	6 3 9 16	15 10 5 4
[23]	23/P	23/S	23/C	[24]	24/P	24/S	24/C
1	9 5 13	1 8 12 13	1 8 13 12	1	9 5 13	1 8 12 13	1 8 13 12
2	10 6 14	15 10 6 3	15 10 3 6	3	11 7 15	14 11 7 2	14 11 2 7
3	11 7 15	14 11 7 2	4 5 16 9	2	10 6 14	15 10 6 3	4 5 16 9
4	12 8 16	4 5 9 16	14 11 2 7	4	12 8 16	4 5 9 16	15 10 3 6
[25]	25/P	25/S	25/C	[26]	26/P	26/S	26/C
2	1 4 3	2 16 13 3	2 16 3 13	2	1 4 3	2 16 13 3	2 16 3 13
6	5 8 7	11 5 8 10	11 5 10 8	10	9 12 11	7 9 12 6	7 9 6 12
10	9 12 11	7 9 12 6	14 4 15 1	6	5 8 7	11 5 8 10	14 4 15 1
14	13 16 15	14 4 1 15	7 9 6 12	14	13 16 15	14 4 1 15	11 5 10 8
.							
[47]	47/P	47/S	47/C	[48]	48/P	48/S	48/C
2	10 6 14	2 7 11 14	2 7 14 11	2	10 6 14	2 7 11 14	2 7 14 11
1	9 5 13	16 9 5 4	16 9 4 5	4	12 8 16	13 12 8 1	13 12 1 8
4	12 8 16	13 12 8 1	3 6 15 10	1	9 5 13	16 9 5 4	3 6 15 10
3	11 7 15	3 6 10 15	13 12 1 8	3	11 7 15	3 6 10 15	16 9 4 5
[49]	49/P	49/S	49/C	[50]	50/P	50/S	50/C
3	1 4 2	3 16 13 2	3 16 2 13	3	1 4 2	3 16 13 2	3 16 2 13
7	5 8 6	10 5 8 11	10 5 11 8	11	9 12 10	6 9 12 7	6 9 7 12
11	9 12 10	6 9 12 7	15 4 14 1	7	5 8 6	10 5 8 11	15 4 14 1
15	13 16 14	15 4 1 14	6 9 7 12	15	13 16 14	15 4 1 14	10 5 11 8
[51]	51/P	51/S	51/C	[52]	52/P	52/S	52/C
3	1 7 5	3 16 10 5	3 16 5 10	3	1 7 5	3 16 10 5	3 16 5 10
4	2 8 6	13 2 8 11	13 2 11 8	11	9 15 13	6 9 15 4	6 9 4 15
11	9 15 13	6 9 15 4	12 7 14 1	4	2 8 6	13 2 8 11	12 7 14 1
12	10 16 14	12 7 1 14	6 9 4 15	12	10 16 14	12 7 1 14	13 2 11 8
.							
[95]	95/P	95/S	95/C	[96]	96/P	96/S	96/C
4	12 8 16	4 5 9 16	4 5 16 9	4	12 8 16	4 5 9 16	4 5 16 9
2	10 6 14	15 10 6 3	15 10 3 6	3	11 7 15	14 11 7 2	14 11 2 7
3	11 7 15	14 11 7 2	1 8 13 12	2	10 6 14	15 10 6 3	1 8 13 12
1	9 5 13	1 8 12 13	14 11 2 7	1	9 5 13	1 8 12 13	15 10 3 6
[97]	97/P	97/S	97/C	[98]	98/P	98/S	98/C
5	1 6 2	5 16 11 2	5 16 2 11	5	1 6 2	5 16 11 2	5 16 2 11
7	3 8 4	10 3 8 13	10 3 13 8	13	9 14 10	4 9 14 7	4 9 7 14
13	9 14 10	4 9 14 7	15 6 12 1	7	3 8 4	10 3 8 13	15 6 12 1
15	11 16 12	15 6 1 12	4 9 7 14	15	11 16 12	15 6 1 12	10 3 13 8

Two types of magic squares of order 4 could successfully be recomposed by this method. There certainly exists the One-to-one Correspondence among the Prototype Squares, Self-complementary MS44 and Complete Pan-magic ones. How beautiful!

#4. Of course, you can choose and list out the 48 Standard Solutions of our objects under the list-forming inequality conditions such as $n1 < n4$; $n1 < n13$; $n1 < n16$; and $n2 > n5$; just after you have got each of the primitive sets of 384 solutions.

```

/**/
/* Print 2 Trios of Recomposed Solutions by DAMT [Standard Set] */
void prs2trios(){
  short l,l4,m,n,p;
  for(m=0;m<tcnt;m=m+2){
    for(n=0;n<4;n++){for(p=0;p<17;p++){d[n][p]=0;}}
    for(n=1;n<17;n++){nm[n]=pts[m][n];}
    damts();
    if((dm[1]<dm[4])&&(dm[1]<dm[13])&&(dm[1]<dm[16])&&(dm[5]<dm[2])){
      cnt1++; d[0][0]=cnt1; for(n=1;n<17;n++){d[0][n]=dm[n];}}
    for(n=1;n<17;n++){nm[n]=ptc[m][n];}
    damtc();
    if((dm[1]<dm[4])&&(dm[1]<dm[13])&&(dm[1]<dm[16])&&(dm[5]<dm[2])){
      cnt2++; d[1][0]=cnt2; for(n=1;n<17;n++){d[1][n]=dm[n];}}
    for(n=1;n<17;n++){nm[n]=pts[m+1][n];}
    damts();
    if((dm[1]<dm[4])&&(dm[1]<dm[13])&&(dm[1]<dm[16])&&(dm[5]<dm[2])){
      cnt1++; d[2][0]=cnt1; for(n=1;n<17;n++){d[2][n]=dm[n];}}
    for(n=1;n<17;n++){nm[n]=ptc[m+1][n];}
    damtc();
    if((dm[1]<dm[4])&&(dm[1]<dm[13])&&(dm[1]<dm[16])&&(dm[5]<dm[2])){
      cnt2++; d[3][0]=cnt2; for(n=1;n<17;n++){d[3][n]=dm[n];}}
    printf("%10d/P%11d/S%11d/C%12d/P%11d/S%11d/C\n",
      pts[m][0],d[0][0],d[1][0],pts[m+1][0],d[2][0],d[3][0]);
    if((d[0][0]>0)||(d[1][0]>0)||(d[2][0]>0)||(d[3][0]>0)){
      for(l=0;l<4;l++){l4=l*4;
        for(p=1;p<5;p++){printf("%3d",pts[m][l4+p]);}
        printf(" ");
        for(p=1;p<5;p++){printf("%3d",d[0][l4+p]);}
        printf(" ");
        for(p=1;p<5;p++){printf("%3d",d[1][l4+p]);}
        printf(" ");
        for(p=1;p<5;p++){printf("%3d",pts[m+1][l4+p]);}
        printf(" ");
        for(p=1;p<5;p++){printf("%3d",d[2][l4+p]);}
        printf(" ");
        for(p=1;p<5;p++){printf("%3d",d[3][l4+p]);}
        printf("\n");
      }
    }
  }
}
}
}
}
/**/

```

The next list shows those 48 standard solutions picked up.

```

*** Two Types of Magic Squares 4x4 Recomposed by 'DAMT' from Prototypes: ***
* Standard Sets: Prototype/P, Self-complementary/S, Complete Pan-magic/C *

```

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

3/P			3/S			3/C			4/P			4/S			4/C								
1	2	5	6	1	15	12	6	1	15	6	12	1	2	5	6	1	15	12	6	1	15	6	12
3	4	7	8	14	4	7	9	14	4	9	7	9	10	13	14	8	10	13	3	8	10	3	13
9	10	13	14	8	10	13	3	11	5	16	2	3	4	7	8	14	4	7	9	11	5	16	2
11	12	15	16	11	5	2	16	8	10	3	13	11	12	15	16	11	5	2	16	14	4	9	7
5/P			5/S			5/C			6/P			6/S			6/C								
1	2	9	10	1	15	8	10	1	15	10	8	1	2	9	10	1	15	8	10	1	15	10	8
3	4	11	12	14	4	11	5	14	4	5	11	5	6	13	14	12	6	13	3	12	6	3	13
5	6	13	14	12	6	13	3	7	9	16	2	3	4	11	12	14	4	11	5	7	9	16	2
7	8	15	16	7	9	2	16	12	6	3	13	7	8	15	16	7	9	2	16	14	4	5	11
7/P			7/S			7/C			8/P			8/S			8/C								
1	3	2	4	1	14	15	4	1	14	4	15	1	3	2	4	1	14	15	4	1	14	4	15
5	7	6	8	12	7	6	9	12	7	9	6	9	11	10	12	8	11	10	5	8	11	5	10
9	11	10	12	8	11	10	5	13	2	16	3	5	7	6	8	12	7	6	9	13	2	16	3
13	15	14	16	13	2	3	16	8	11	5	10	13	15	14	16	13	2	3	16	12	7	9	6
9/P			0/S			0/C			10/P			9/S			9/C								
1	3	5	7	0	0	0	0	0	0	0	0	1	3	5	7	1	14	12	7	1	14	7	12
2	4	6	8	0	0	0	0	0	0	0	0	9	11	13	15	8	11	13	2	8	11	2	13
9	11	13	15	0	0	0	0	0	0	0	0	2	4	6	8	15	4	6	9	10	5	16	3
10	12	14	16	0	0	0	0	0	0	0	0	10	12	14	16	10	5	3	16	15	4	9	6
11/P			0/S			0/C			12/P			10/S			10/C								
1	3	9	11	0	0	0	0	0	0	0	0	1	3	9	11	1	14	8	11	1	14	11	8
2	4	10	12	0	0	0	0	0	0	0	0	5	7	13	15	12	7	13	2	12	7	2	13
5	7	13	15	0	0	0	0	0	0	0	0	2	4	10	12	15	4	10	5	6	9	16	3
6	8	14	16	0	0	0	0	0	0	0	0	6	8	14	16	6	9	3	16	15	4	5	10
13/P			0/S			0/C			14/P			11/S			11/C								
1	5	2	6	0	0	0	0	0	0	0	0	1	5	2	6	1	12	15	6	1	12	6	15
3	7	4	8	0	0	0	0	0	0	0	0	9	13	10	14	8	13	10	3	8	13	3	10
9	13	10	14	0	0	0	0	0	0	0	0	3	7	4	8	14	7	4	9	11	2	16	5
11	15	12	16	0	0	0	0	0	0	0	0	11	15	12	16	11	2	5	16	14	7	9	4
15/P			0/S			0/C			16/P			12/S			12/C								
1	5	3	7	0	0	0	0	0	0	0	0	1	5	3	7	1	12	14	7	1	12	7	14
2	6	4	8	0	0	0	0	0	0	0	0	9	13	11	15	8	13	11	2	8	13	2	11
9	13	11	15	0	0	0	0	0	0	0	0	2	6	4	8	15	6	4	9	10	3	16	5
10	14	12	16	0	0	0	0	0	0	0	0	10	14	12	16	10	3	5	16	15	6	9	4
17/P			0/S			0/C			18/P			0/S			0/C								
19/P			0/S			0/C			20/P			0/S			0/C								
21/P			0/S			0/C			22/P			0/S			0/C								
23/P			0/S			0/C			24/P			0/S			0/C								
25/P			13/S			13/C			26/P			14/S			14/C								
2	1	4	3	2	16	13	3	2	16	3	13	2	1	4	3	2	16	13	3	2	16	3	13
6	5	8	7	11	5	8	10	11	5	10	8	10	9	12	11	7	9	12	6	7	9	6	12
10	9	12	11	7	9	12	6	14	4	15	1	6	5	8	7	11	5	8	10	14	4	15	1
14	13	16	15	14	4	1	15	7	9	6	12	14	13	16	15	14	4	1	15	11	5	10	8
27/P			15/S			15/C			28/P			16/S			16/C								
2	1	6	5	2	16	11	5	2	16	5	11	2	1	6	5	2	16	11	5	2	16	5	11
4	3	8	7	13	3	8	10	13	3	10	8	10	9	14	13	7	9	14	4	7	9	4	14
10	9	14	13	7	9	14	4	12	6	15	1	4	3	8	7	13	3	8	10	12	6	15	1
12	11	16	15	12	6	1	15	7	9	4	14	12	11	16	15	12	6	1	15	13	3	10	8
29/P			17/S			17/C			30/P			18/S			18/C								
2	1	10	9	2	16	7	9	2	16	9	7	2	1	10	9	2	16	7	9	2	16	9	7
4	3	12	11	13	3	12	6	13	3	6	12	6	5	14	13	11	5	14	4	11	5	4	14
6	5	14	13	11	5	14	4	8	10	15	1	4	3	12	11	13	3	12	6	8	10	15	1
8	7	16	15	8	10	1	15	11	5	4	14	8	7	16	15	8	10	1	15	13	3	6	12

	31/P	19/S	19/C	32/P	20/S	20/C
2	4 1 3	2 13 16 3	2 13 3 16	2 4 1 3	2 13 16 3	2 13 3 16
6	8 5 7	11 8 5 10	11 8 10 5	10 12 9 11	7 12 9 6	7 12 6 9
10	12 9 11	7 12 9 6	14 1 15 4	6 8 5 7	11 8 5 10	14 1 15 4
14	16 13 15	14 1 4 15	7 12 6 9	14 16 13 15	14 1 4 15	11 8 10 5
	33/P	0/S	0/C	34/P	21/S	21/C
2	4 6 8	0 0 0 0	0 0 0 0	2 4 6 8	2 13 11 8	2 13 8 11
1	3 5 7	0 0 0 0	0 0 0 0	10 12 14 16	7 12 14 1	7 12 1 14
10	12 14 16	0 0 0 0	0 0 0 0	1 3 5 7	16 3 5 10	9 6 15 4
9	11 13 15	0 0 0 0	0 0 0 0	9 11 13 15	9 6 4 15	16 3 10 5
	35/P	0/S	0/C	36/P	22/S	22/C
2	4 10 12	0 0 0 0	0 0 0 0	2 4 10 12	2 13 7 12	2 13 12 7
1	3 9 11	0 0 0 0	0 0 0 0	6 8 14 16	11 8 14 1	11 8 1 14
6	8 14 16	0 0 0 0	0 0 0 0	1 3 9 11	16 3 9 6	5 10 15 4
5	7 13 15	0 0 0 0	0 0 0 0	5 7 13 15	5 10 4 15	16 3 6 9
	37/P	0/S	0/C	38/P	23/S	23/C
2	6 1 5	0 0 0 0	0 0 0 0	2 6 1 5	2 11 16 5	2 11 5 16
4	8 3 7	0 0 0 0	0 0 0 0	10 14 9 13	7 14 9 4	7 14 4 9
10	14 9 13	0 0 0 0	0 0 0 0	4 8 3 7	13 8 3 10	12 1 15 6
12	16 11 15	0 0 0 0	0 0 0 0	12 16 11 15	12 1 6 15	13 8 10 3
	39/P	0/S	0/C	40/P	24/S	24/C
2	6 4 8	0 0 0 0	0 0 0 0	2 6 4 8	2 11 13 8	2 11 8 13
1	5 3 7	0 0 0 0	0 0 0 0	10 14 12 16	7 14 12 1	7 14 1 12
10	14 12 16	0 0 0 0	0 0 0 0	1 5 3 7	16 5 3 10	9 4 15 6
9	13 11 15	0 0 0 0	0 0 0 0	9 13 11 15	9 4 6 15	16 5 10 3
	41/P	0/S	0/C	42/P	0/S	0/C
	43/P	0/S	0/C	44/P	0/S	0/C
	45/P	0/S	0/C	46/P	0/S	0/C
	47/P	0/S	0/C	48/P	0/S	0/C
	49/P	0/S	25/C	50/P	0/S	26/C
3	1 4 2	0 0 0 0	3 16 2 13	3 1 4 2	0 0 0 0	3 16 2 13
7	5 8 6	0 0 0 0	10 5 11 8	11 9 12 10	0 0 0 0	6 9 7 12
11	9 12 10	0 0 0 0	15 4 14 1	7 5 8 6	0 0 0 0	15 4 14 1
15	13 16 14	0 0 0 0	6 9 7 12	15 13 16 14	0 0 0 0	10 5 11 8
	51/P	25/S	27/C	52/P	26/S	28/C
3	1 7 5	3 16 10 5	3 16 5 10	3 1 7 5	3 16 10 5	3 16 5 10
4	2 8 6	13 2 8 11	13 2 11 8	11 9 15 13	6 9 15 4	6 9 4 15
11	9 15 13	6 9 15 4	12 7 14 1	4 2 8 6	13 2 8 11	12 7 14 1
12	10 16 14	12 7 1 14	6 9 4 15	12 10 16 14	12 7 1 14	13 2 11 8
	53/P	27/S	29/C	54/P	28/S	30/C
3	1 11 9	3 16 6 9	3 16 9 6	3 1 11 9	3 16 6 9	3 16 9 6
4	2 12 10	13 2 12 7	13 2 7 12	7 5 15 13	10 5 15 4	10 5 4 15
7	5 15 13	10 5 15 4	8 11 14 1	4 2 12 10	13 2 12 7	8 11 14 1
8	6 16 14	8 11 1 14	10 5 4 15	8 6 16 14	8 11 1 14	13 2 7 12
	55/P	0/S	31/C	56/P	0/S	32/C
3	4 1 2	0 0 0 0	3 13 2 16	3 4 1 2	0 0 0 0	3 13 2 16
7	8 5 6	0 0 0 0	10 8 11 5	11 12 9 10	0 0 0 0	6 12 7 9
11	12 9 10	0 0 0 0	15 1 14 4	7 8 5 6	0 0 0 0	15 1 14 4
15	16 13 14	0 0 0 0	6 12 7 9	15 16 13 14	0 0 0 0	10 8 11 5
	57/P	0/S	0/C	58/P	29/S	33/C
3	4 7 8	0 0 0 0	0 0 0 0	3 4 7 8	3 13 10 8	3 13 8 10
1	2 5 6	0 0 0 0	0 0 0 0	11 12 15 16	6 12 15 1	6 12 1 15
11	12 15 16	0 0 0 0	0 0 0 0	1 2 5 6	16 2 5 11	9 7 14 4
9	10 13 14	0 0 0 0	0 0 0 0	9 10 13 14	9 7 4 14	16 2 11 5

	59/P		0/S		0/C		60/P		30/S		34/C
3	4 11 12	0	0 0 0	0	0 0 0	3	4 11 12	3	13 6 12	3	13 12 6
1	2 9 10	0	0 0 0	0	0 0 0	7	8 15 16	10	8 15 1	10	8 1 15
7	8 15 16	0	0 0 0	0	0 0 0	1	2 9 10	16	2 9 7	5	11 14 4
5	6 13 14	0	0 0 0	0	0 0 0	5	6 13 14	5	11 4 14	16	2 7 9
	61/P		0/S		0/C		62/P		31/S		0/C
3	7 1 5	0	0 0 0	0	0 0 0	3	7 1 5	3	10 16 5	0	0 0 0
4	8 2 6	0	0 0 0	0	0 0 0	11	15 9 13	6	15 9 4	0	0 0 0
11	15 9 13	0	0 0 0	0	0 0 0	4	8 2 6	13	8 2 11	0	0 0 0
12	16 10 14	0	0 0 0	0	0 0 0	12	16 10 14	12	1 7 14	0	0 0 0
	65/P		0/S		0/C		66/P		0/S		0/C
	67/P		0/S		0/C		68/P		0/S		0/C
	69/P		0/S		0/C		70/P		0/S		0/C
	71/P		0/S		0/C		72/P		0/S		0/C
	63/P		0/S		0/C		64/P		32/S		0/C
3	7 4 8	0	0 0 0	0	0 0 0	3	7 4 8	3	10 13 8	0	0 0 0
1	5 2 6	0	0 0 0	0	0 0 0	11	15 12 16	6	15 12 1	0	0 0 0
11	15 12 16	0	0 0 0	0	0 0 0	1	5 2 6	16	5 2 11	0	0 0 0
9	13 10 14	0	0 0 0	0	0 0 0	9	13 10 14	9	4 7 14	0	0 0 0
	73/P		0/S		35/C		74/P		0/S		36/C
4	2 3 1	0	0 0 0	4	15 1 14	4	2 3 1	0	0 0 0	4	15 1 14
8	6 7 5	0	0 0 0	9	6 12 7	12	10 11 9	0	0 0 0	5	10 8 11
12	10 11 9	0	0 0 0	16	3 13 2	8	6 7 5	0	0 0 0	16	3 13 2
16	14 15 13	0	0 0 0	5	10 8 11	16	14 15 13	0	0 0 0	9	6 12 7
	75/P		33/S		37/C		76/P		34/S		38/C
4	2 8 6	4	15 9 6	4	15 6 9	4	2 8 6	4	15 9 6	4	15 6 9
3	1 7 5	14	1 7 12	14	1 12 7	12	10 16 14	5	10 16 3	5	10 3 16
12	10 16 14	5	10 16 3	11	8 13 2	3	1 7 5	14	1 7 12	11	8 13 2
11	9 15 13	11	8 2 13	5	10 3 16	11	9 15 13	11	8 2 13	14	1 12 7
	77/P		35/S		39/C		78/P		36/S		40/C
4	2 12 10	4	15 5 10	4	15 10 5	4	2 12 10	4	15 5 10	4	15 10 5
3	1 11 9	14	1 11 8	14	1 8 11	8	6 16 14	9	6 16 3	9	6 3 16
8	6 16 14	9	6 16 3	7	12 13 2	3	1 11 9	14	1 11 8	7	12 13 2
7	5 15 13	7	12 2 13	9	6 3 16	7	5 15 13	7	12 2 13	14	1 8 11
	79/P		0/S		41/C		80/P		0/S		42/C
4	3 2 1	0	0 0 0	4	14 1 15	4	3 2 1	0	0 0 0	4	14 1 15
8	7 6 5	0	0 0 0	9	7 12 6	12	11 10 9	0	0 0 0	5	11 8 10
12	11 10 9	0	0 0 0	16	2 13 3	8	7 6 5	0	0 0 0	16	2 13 3
16	15 14 13	0	0 0 0	5	11 8 10	16	15 14 13	0	0 0 0	9	7 12 6
	81/P		0/S		0/C		82/P		37/S		43/C
4	3 8 7	0	0 0 0	0	0 0 0	4	3 8 7	4	14 9 7	4	14 7 9
2	1 6 5	0	0 0 0	0	0 0 0	12	11 16 15	5	11 16 2	5	11 2 16
12	11 16 15	0	0 0 0	0	0 0 0	2	1 6 5	15	1 6 12	10	8 13 3
10	9 14 13	0	0 0 0	0	0 0 0	10	9 14 13	10	8 3 13	15	1 12 6
	83/P		0/S		0/C		84/P		38/S		44/C
4	3 12 11	0	0 0 0	0	0 0 0	4	3 12 11	4	14 5 11	4	14 11 5
2	1 10 9	0	0 0 0	0	0 0 0	8	7 16 15	9	7 16 2	9	7 2 16
8	7 16 15	0	0 0 0	0	0 0 0	2	1 10 9	15	1 10 8	6	12 13 3
6	5 14 13	0	0 0 0	0	0 0 0	6	5 14 13	6	12 3 13	15	1 8 10
	85/P		0/S		0/C		86/P		39/S		0/C
4	8 2 6	0	0 0 0	0	0 0 0	4	8 2 6	4	9 15 6	0	0 0 0
3	7 1 5	0	0 0 0	0	0 0 0	12	16 10 14	5	16 10 3	0	0 0 0
12	16 10 14	0	0 0 0	0	0 0 0	3	7 1 5	14	7 1 12	0	0 0 0
11	15 9 13	0	0 0 0	0	0 0 0	11	15 9 13	11	2 8 13	0	0 0 0

		87/P				0/S				0/C				88/P			40/S			0/C					
4	8	3	7	0	0	0	0	0	0	0	0	0	0	4	8	3	7	4	9	14	7	0	0	0	0
2	6	1	5	0	0	0	0	0	0	0	0	0	0	12	16	11	15	5	16	11	2	0	0	0	0
12	16	11	15	0	0	0	0	0	0	0	0	0	0	2	6	1	5	15	6	1	12	0	0	0	0
10	14	9	13	0	0	0	0	0	0	0	0	0	0	10	14	9	13	10	3	8	13	0	0	0	0
		89/P				0/S				0/C				90/P			0/S			0/C					
		91/P				0/S				0/C				92/P			0/S			0/C					
		93/P				0/S				0/C				94/P			0/S			0/C					
		95/P				0/S				0/C				96/P			0/S			0/C					
		97/P				0/S				0/C				98/P			0/S			45/C					
5	1	6	2	0	0	0	0	0	0	0	0	0	0	5	1	6	2	0	0	0	0	5	16	2	11
7	3	8	4	0	0	0	0	0	0	0	0	0	0	13	9	14	10	0	0	0	0	4	9	7	14
13	9	14	10	0	0	0	0	0	0	0	0	0	0	7	3	8	4	0	0	0	0	15	6	12	1
15	11	16	12	0	0	0	0	0	0	0	0	0	0	15	11	16	12	0	0	0	0	10	3	13	8
		99/P				0/S				0/C				100/P			0/S			46/C					
5	1	7	3	0	0	0	0	0	0	0	0	0	0	5	1	7	3	0	0	0	0	5	16	3	10
6	2	8	4	0	0	0	0	0	0	0	0	0	0	13	9	15	11	0	0	0	0	4	9	6	15
13	9	15	11	0	0	0	0	0	0	0	0	0	0	6	2	8	4	0	0	0	0	14	7	12	1
14	10	16	12	0	0	0	0	0	0	0	0	0	0	14	10	16	12	0	0	0	0	11	2	13	8
		101/P				41/S				0/C				102/P			42/S			0/C					
5	1	13	9	5	16	4	9	0	0	0	0	0	0	5	1	13	9	5	16	4	9	0	0	0	0
6	2	14	10	11	2	14	7	0	0	0	0	0	0	7	3	15	11	10	3	15	6	0	0	0	0
7	3	15	11	10	3	15	6	0	0	0	0	0	0	6	2	14	10	11	2	14	7	0	0	0	0
8	4	16	12	8	13	1	12	0	0	0	0	0	0	8	4	16	12	8	13	1	12	0	0	0	0
		103/P				0/S				0/C				104/P			0/S			0/C					
		105/P				0/S				0/C				106/P			43/S			0/C					
5	6	7	8	0	0	0	0	0	0	0	0	0	0	5	6	7	8	5	11	10	8	0	0	0	0
1	2	3	4	0	0	0	0	0	0	0	0	0	0	13	14	15	16	4	14	15	1	0	0	0	0
13	14	15	16	0	0	0	0	0	0	0	0	0	0	1	2	3	4	16	2	3	13	0	0	0	0
9	10	11	12	0	0	0	0	0	0	0	0	0	0	9	10	11	12	9	7	6	12	0	0	0	0
		107/P				0/S				0/C				108/P			0/S			0/C					
		109/P				0/S				0/C				110/P			0/S			0/C					
		111/P				0/S				0/C				112/P			44/S			0/C					
5	7	6	8	0	0	0	0	0	0	0	0	0	0	5	7	6	8	5	10	11	8	0	0	0	0
1	3	2	4	0	0	0	0	0	0	0	0	0	0	13	15	14	16	4	15	14	1	0	0	0	0
13	15	14	16	0	0	0	0	0	0	0	0	0	0	1	3	2	4	16	3	2	13	0	0	0	0
9	11	10	12	0	0	0	0	0	0	0	0	0	0	9	11	10	12	9	6	7	12	0	0	0	0
		113/P				0/S				0/C				114/P			0/S			0/C					
		115/P				0/S				0/C				116/P			0/S			0/C					
		117/P				0/S				0/C				118/P			0/S			0/C					
		119/P				0/S				0/C				120/P			0/S			0/C					
		121/P				0/S				0/C				122/P			0/S			47/C					
6	2	5	1	0	0	0	0	0	0	0	0	0	0	6	2	5	1	0	0	0	0	6	15	1	12
8	4	7	3	0	0	0	0	0	0	0	0	0	0	14	10	13	9	0	0	0	0	3	10	8	13
14	10	13	9	0	0	0	0	0	0	0	0	0	0	8	4	7	3	0	0	0	0	16	5	11	2
16	12	15	11	0	0	0	0	0	0	0	0	0	0	16	12	15	11	0	0	0	0	9	4	14	7
		123/P				0/S				0/C				124/P			0/S			48/C					
6	2	8	4	0	0	0	0	0	0	0	0	0	0	6	2	8	4	0	0	0	0	6	15	4	9
5	1	7	3	0	0	0	0	0	0	0	0	0	0	14	10	16	12	0	0	0	0	3	10	5	16
14	10	16	12	0	0	0	0	0	0	0	0	0	0	5	1	7	3	0	0	0	0	13	8	11	2
13	9	15	11	0	0	0	0	0	0	0	0	0	0	13	9	15	11	0	0	0	0	12	1	14	7
		125/P				45/S				0/C				126/P			46/S			0/C					
6	2	14	10	6	15	3	10	0	0	0	0	0	0	6	2	14	10	6	15	3	10	0	0	0	0
5	1	13	9	12	1	13	8	0	0	0	0	0	0	8	4	16	12	9	4	16	5	0	0	0	0
8	4	16	12	9	4	16	5	0	0	0	0	0	0	5	1	13	9	12	1	13	8	0	0	0	0
7	3	15	11	7	14	2	11	0	0	0	0	0	0	7	3	15	11	7	14	2	11	0	0	0	0

127/P	0/S	0/C	128/P	0/S	0/C
129/P	0/S	0/C	130/P	47/S	0/C
6 5 8 7	0 0 0 0	0 0 0 0	6 5 8 7	6 12 9 7	0 0 0 0
2 1 4 3	0 0 0 0	0 0 0 0	14 13 16 15	3 13 16 2	0 0 0 0
14 13 16 15	0 0 0 0	0 0 0 0	2 1 4 3	15 1 4 14	0 0 0 0
10 9 12 11	0 0 0 0	0 0 0 0	10 9 12 11	10 8 5 11	0 0 0 0
131/P	0/S	0/C	132/P	0/S	0/C
133/P	0/S	0/C	134/P	0/S	0/C
135/P	0/S	0/C	136/P	48/S	0/C
6 8 5 7	0 0 0 0	0 0 0 0	6 8 5 7	6 9 12 7	0 0 0 0
2 4 1 3	0 0 0 0	0 0 0 0	14 16 13 15	3 16 13 2	0 0 0 0
14 16 13 15	0 0 0 0	0 0 0 0	2 4 1 3	15 4 1 14	0 0 0 0
10 12 9 11	0 0 0 0	0 0 0 0	10 12 9 11	10 5 8 11	0 0 0 0
137/P	0/S	0/C	138/P	0/S	0/C
139/P	0/S	0/C	140/P	0/S	0/C
141/P	0/S	0/C	142/P	0/S	0/C
143/P	0/S	0/C	144/P	0/S	0/C
145/P	0/S	0/C	146/P	0/S	0/C
147/P	0/S	0/C	148/P	0/S	0/C
149/P	0/S	0/C	150/P	0/S	0/C

.

[Counts = 384/P, 48/S, 48/C] OK!

If you want to have this list, you must not make the solution set of 48 'Standard' Prototype Squares at first. You should hold on making the 384 'Primitive' ones.

Otherwise you would miss a part of One-to-one Correspondences among the 3 sets of solutions. As you see in the last half of the list above, you could not always find all the complete correspondences among the three solutions. It is quite embarrassing.

#5. We could certainly reconstruct all the 384 Self-complementary Squares and all the 384 Complete pan-magic ones, whenever we have got the common set of 'Prototype Squares' and the appropriate Models and DAMTs. But why? What makes the same set of 384 Prototype Squares work all right? Why is another set of them unnecessary?

Let's think a little about them now, shall we?

[Figure 7: Various types of Magic Squares and their Properties]

Types: /Properties/	Prototype squares	Self- Complementary	Pan- magic	'C & C' pan-magic
Complementary pairs located symmetrically	0	0	X	X
4 entries on pan-diagonals add up to 34	0	X	0	0
Every row & Column add up to the same 34	X	0	0	0
Any 2 x 2 block within add up to 34	X	X	0	0
Equality of Cross-sums	0	X	X	X

How many Standard Solutions?	48	48	48	48
------------------------------------	----	----	----	----

Why can we build such a beautiful 'bridge' as the One-to-one Correspondence between the parallel worlds of Prototype squares and of Object solutions?

We can guess there exists anything deep, like the 'inner structural relation' among them. We can also guess that the common set of Prototype Squares should mean the only one common Origin of everything.

#6. I must confess here that the basic idea of 'Prototype squares' and 'Do-it-After-the-Model Transformation' is not my original invention. To tell the truth, I borrowed it from Drs. Ollerenshaw and Brae, British researchers. But I examined it in every case of every order intensively and tried to extend their idea.

I say, I could almost have got the new concept and the tools reborn.

Drs. Ollerenshaw and Brae presented it in their page only about 'Most-Perfect Pan-diagonal' magic squares (that are equivalent to our 'C & C' pan-magic squares) of Orders 4, 8, 12, 16,

But this idea itself could be applied to broader objects, I thought, to any other typical magic squares in every order. I had to extend the concept of 'Prototype Squares' and the application 'DAMT' explicitly. I also had to improve my knowledge about the way how to make the set of Prototype Squares and how to select the representative model out of object squares.

In the case of 'Composite & Complete' Magic Squares of Order 4, Drs. Ollerenshaw & Brae discovered the three 'Principal Squares' among the 'Reversible' ones.

They are equivalent to No.1, 3, and 5 in our lists above.

** 'Principal Squares' **											
1/PT				/SC				/CC			
1	2	3	4	1	15	14	4	1	15	4	14
5	6	7	8	12	6	7	9	12	6	9	7
9	10	11	12	8	10	11	5	13	3	16	2
13	14	15	16	13	3	2	16	8	10	5	11
3/PT				/SC				/CC			
1	2	5	6	1	15	12	6	1	15	6	12
3	4	7	8	14	4	7	9	14	4	9	7
9	10	13	14	8	10	13	3	11	5	16	2
11	12	15	16	11	5	2	16	8	10	3	13
5/PT				/SC				/CC			
1	2	9	10	1	15	8	10	1	15	10	8
3	4	11	12	14	4	11	5	14	4	5	11
5	6	13	14	12	6	13	3	7	9	16	2
7	8	15	16	7	9	2	16	12	6	3	13

These three solutions are the same with the ones classified before as 'Fundamental Three' in Self-complementary and 'C & C' Pan-magic squares of order 4.

Recent study revealed us that the 4-dimensional extra-cubic magic forms of order 2 is certainly the origin of these squares.

There are three different patterns of them. They are equivalent to those three below. The 'One-to-one Correspondence' could be found among them.

Everything tells us about any deep relationships among them: the beautiful bridges built over the three different worlds.

** 'Fundamental Three': EC02^4 and Developed Forms **

1/n1	/n2	1/SC	/CC
1---12	8---13	1 15 8 10	1 15 10 8
15----6	10----3	14 4 11 5	14 4 5 11
14- -7	11- -2	12 6 13 3	7 9 16 2
4----9	5---16	7 9 2 16	12 6 3 13
2/n1	/n2	2/SC	/CC
1----8	12---13	1 15 12 6	1 15 6 12
15---10	6----3	14 4 7 9	14 4 9 7
14- -11	7- -2	8 10 13 3	11 5 16 2
4----5	9---16	11 5 2 16	8 10 3 13
3/n1	/n2	3/SC	/CC
1----8	14---11	1 15 14 4	1 15 4 14
15---10	4----5	12 6 7 9	12 6 9 7
12- -13	7- -2	8 10 11 5	13 3 16 2
6----3	9---16	13 3 2 16	8 10 5 11

#7. Why are there always only three Fundamentals?

Why are there only three 'fundamental' solutions in all and no more?

I think the 'C&C' Pan-magic Squares could tell us about the secret.

It depends upon the miraculous unity that three types of magic squares of order 4:

(1) Pan-diagonal, (2) Complete, and (3) C&C pan-magic have the same solution set.

Therefore you can find the same property as the most strict conditions of C&C pan-magic type equally among all types of pan-diagonal magic squares of order 4.

It is always very important how the 'adjacent numbers next to n1' could appear and take certain values, I have found. It is the problem of permutations.

In the 'Composite and Complete' Magic Squares 4x4, many kinds of complementary pairs are desperately needed.

[Figure 8: 'Composite and Complete' Squares]

15 16 13 14 15 16 13 14	
3 4 1 2 3 4 1 2	n1+ n2+ n5+ n6=S....(1)
7 8 5 6 7 8 5 6	n2+ n3+ n6+ n7=S....(2)
11 12 9 10 11 12 9 10	n3+ n4+ n7+ n8=S....(3)
15 16 13 14 15 16 13 14	n5+ n6+ n9+n10=S....(4)
3 4 1 2 3 4 1 2	n6+ n7+n10+n11=S....(5)
	n7+ n8+n11+n12=S....(6)
	n9+n10+n13+n14=S....(7)
	n10+n11+n14+n15=S....(8)
	n11+n12+n15+n16=S....(9)
	n1+n2+n3+n4=K...(10);
	n1+n5+n9+n13=K...(11);
	n1+n6+n11+n16=K...(12) (S=K=34)

- (1) n1+n5=n3+n7=n10+n14=n12+n16=C1;
n2+n6=n4+n8=n9+n13=n11+n15=D1; C1+D1=K(=34)
- (2) n5+n9=n7+n11=n2+n14=n4+n16=C2;
n1+n13=n6+n10=n3+n15=n8+n12=D2; C2+D2=K
- (3) n1+n2=n9+n10=n7+n8=n15+n16=C3;
n3+n4=n5+n6=n11+n12=n13+n14=D3; C3+D3=K
- (4) n2+n3=n5+n8=n10+n11=n13+n16=C4;
n1+n4=n6+n7=n9+n12=n14+n15=D4; C4+D4=K
- (5) n1+n11=n6+n16=n2+n12=n7+n13=n3+n9=n8+n14=n4+n10=n5+n15=17

The last 8 pairs of 17 are all located on its pan-diagonals.
 Each group of the others consists of 2 sets of 4 pairs of C1~4 and D1~4.
 [Notice that each group has (n1+nN) without any exceptions.]

Are there so many complementary pairs available as eight?
 Yes, there are. But not very many varieties.

**** Complementary Pairs For Magic Square 4x4 ****

- CC=6: C1(1,5), C2(2,4),
 DD=28: D1(16,12), D2(15,13),
- CC=7: C1(1,6), C2(2,5), C3(3,4),
 DD=27: D1(16,11), D2(15,12), D3(14,13),
- CC=8: C1(1,7), C2(2,6), C3(3,5),
 DD=26: D1(16,10), D2(15,11), D3(14,12),
- CC=9: C1(1,8), C2(2,7), C3(3,6), C4(4,5),
 DD=25: D1(16,9), D2(15,10), D3(14,11), D4(13,12), OK!
- CC=10: C1(1,9), C2(3,7), C3(4,6),
 DD=24: D1(16,8), D2(14,10), D3(13,11),
- CC=11: C1(1,10), C2(2,9), C3(5,6),
 DD=23: D1(16,7), D2(15,8), D3(12,11),
- CC=12: C1(1,11), C2(2,10), C3(3,9),
 DD=22: D1(16,6), D2(15,7), D3(14,8),
- CC=13: C1(1,12), C2(2,11), C3(3,10), C4(4,9),
 DD=21: D1(16,5), D2(15,6), D3(14,7), D4(13,8), OK!
- CC=14: C1(1,13), C2(2,12), C3(3,11),
 DD=20: D1(16,4), D2(15,5), D3(14,6),
- CC=15: C1(1,14), C2(2,13), C3(5,10), C4(6,9),
 DD=19: D1(16,3), D2(15,4), D3(12,7), D4(11,8), OK!
- CC=16: C1(1,15), C2(3,13), C3(5,11), C4(7,9),
 DD=18: D1(16,2), D2(14,4), D3(12,6), D4(10,8), OK!
- CC=DD=17: 1(1,16), 2(2,15), 3(3,14), 4(4,13),
 5(5,12), 6(6,11), 7(7,10), 8(8,9); OK!

There are only 5 groups available. It is the fate we should accept when we want to make pan-magic squares of order 4 in classical style using the integers 1~16.

The last group of CC=DD=17 is used only for making pan-diagonals.

How and where do you place the other 4 groups?

Notice that each group has (1, N) at the top position. Take it and suppose the 'representative pair' of its group. If you assume the 'standard form' with n1=1, the adjacent number next to n1 must take any value of N: {15, 14, 12, 8}.

That means each of {n2, n4, n5, n13} must take any value of {15, 14, 12, 8} but nothing else.

Once they are determined, all the other entries should be determined accordingly, say, in one way. It is amazing.

After all 'N sitting next to 1' is always acting a very important role.

The variety of appearances of {15, 14, 12, 8} on {n2, n4, n5, n13} is calculated

as follows: ${}^4C_2 / {}^2P_2 = 6 / 2 = 3$

[Figure 9: The Numbers sitting next to N1]

```
n16 n13      (n1+n2+n5+n6)+(n1+n4+n13+n16)=34+34=68;
      |
      |      But n1=1; n6+n16=17;
n4--n1--n2--n3--n4  2+(n2+n5+n4+n13)+17=68
      |
      |      Therefore n2+n4+n5+n13=49
      |      {15, 14, 12, 8} are reasonable.
n5  n6  n7  n8
```

(The original written on Jul. 4, 2001; Feb.26, 2003; Oct.17, 2006;
Newest revision on Aug. 2, 2010 by Kanji Setsuda on MacOSX and Xcode 3.1)

E-Mail Address: jag12001@nifty.com

[Appendix: Program List]

```
/** Comparison of 'Prototype' Squares of Order 4 between **/
/** Self-Complementary Type and Complete Pan-magic Type **/
/** 'ProtoT4S&C384.c' Dictated by Kanji Setsuda **/
/** on Mar.8, 2003; Sep.26, 2006; Aug.1, 2010 **/
/** Working with MacOSX & Xcode 3.1.2 **/
/**/
#include <stdio.h>
/**/
short cnt1, cnt2, cnt3, tcnt;
short LSM, PSM;
short nm[17], uflg[17], dm[17];
short d[7][17];
short pts[385][17];
short ptc[385][17];
/**/
/* Sub-Routines */
void sstp01(void), sstp02(void), sstp03(void);
void sstp04(void), sstp05(void), sstp06(void);
void sstp07(void), sstp08(void), sstp09(void);
void recpspr(void), pr6ans(void);
void cstp01(void), cstp02(void), cstp03(void);
void cstp04(void), cstp05(void), cstp06(void);
void cstp07(void), cstp08(void), cstp09(void);
void recpspr(void);
void pr3pairs(short x);
void damts(void), damtc(void);
void prp2trios(short x);
void prs2trios();
/**/
/* Main Program */
int main(){
short m,n;
printf("\n** Comparison of 'Prototype' Squares of Order 4 between **\n");
printf("** Self-Complementary Type and Complete Pan-magic Type **\n");
printf("\n** 'Prototype' Squares of Order 4 for Self-complementary Type **\n");
PSM=17; LSM=34;
for(n=0;n<17;n++){nm[n]=0; uflg[n]=0;}
cnt1=0; cnt3=0;
sstp01(); //Make 'Prototype Squares' for Self-complementary Type */
printf(" [Count = %d]\n",cnt1);
printf("\n** 'Prototype' Squares of Order 4 for Complete Pan-magic Type **\n");
for(n=0;n<17;n++){nm[n]=0; uflg[n]=0;}
cnt2=0; cnt3=0;
cstp01(); //Make 'Prototype Squares' for Complete Pan-magic Type */
printf(" [Count = %d]\n",cnt2);
```

```

printf("\n** Compare 'Prototype' Squares 4x4 between S-C and Complete PM Types **\n");
pr3pairs(50);
if(m<cnt1){printf(" . . . . \n");}
printf(" [Counts = %d/S : %d/C]\n",cnt1,cnt2);
tcnt=cnt1; cnt1=0; cnt2=0;
printf("\n *** Two Types of Magic Squares 4x4 Recomposed by 'DAMT' from Prototypes: ***\n");
printf(" * Primitive Sets: Prototype/P, Self-complementary/S, Complete Pan-magic/C *\n");
prp2trios(50);
printf(" . . . . \n");
printf(" [Counts = %d/P, %d/S, %d/C]\n",tcnt,tcnt,tcnt);
cnt1=0; cnt2=0;
printf("\n *** Two Types of Magic Squares 4x4 Recomposed by 'DAMT' from Prototypes: ***\n");
printf(" * Standard Sets: Prototype/P, Self-complementary/S, Complete Pan-magic/C *\n");
prs2trios();
printf(" . . . . \n");
printf(" [Counts = %d/P, %d/S, %d/C] OK!\n",tcnt,cnt1,cnt2);
return 0;
}
/**/
/* Make 'Prototype Squares' for Self-complementary Type */
/* Set N1 & n16=PSM-N1 */
void sstp01(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[1]=a; nm[16]=b;
uflg[a]=1; uflg[b]=1;
sstp02();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set N15 & n2=PSM-N15 */
void sstp02(){
short a,b;
for(a=16;a>0;a--){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[15]=a; nm[2]=b;
uflg[a]=1; uflg[b]=1;
sstp03();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set N14 & n3=PSM-N14 */
void sstp03(){
short a,b;
for(a=16;a>0;a--){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[14]=a; nm[3]=b;
uflg[a]=1; uflg[b]=1;
sstp04();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set n4=LSM-n1-n15-n14 & n13 */
void sstp04(){
short a,b;
a=LSM-nm[1]-nm[15]-nm[14];
if((0<a)&&(a<17)){
b=LSM-nm[3]-nm[2]-nm[16];
if(a+b==PSM){
if((uflg[a]==0)&&(uflg[b]==0)){
uflg[a]=1; uflg[b]=1;
nm[4]=a; nm[13]=b;
sstp05();
}
}
}
}

```

```

        uflg[b]=0; uflg[a]=0;}}
}
/* Set N12 & n5=PSM-N12 */
void sstp05(){
    short a,b;
    for(a=16;a>0;a--){b=PSM-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            uflg[a]=1; uflg[b]=1;
            nm[12]=a; nm[5]=b;
            sstp06();
            uflg[b]=0; uflg[a]=0;}
        }
}
/* Set N6 & n11=PSM-n6 */
void sstp06(){
    short a,b;
    for(a=1;a<17;a++){b=PSM-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            uflg[a]=1; uflg[b]=1;
            nm[6]=a; nm[11]=b;
            if(nm[1]+nm[6]+nm[11]+nm[16]==LSM){sstp07();}
            uflg[b]=0; uflg[a]=0;}
        }
}
/* Set n8=LSM-n1-n12-n13 & n9 */
void sstp07(){
    short a,b;
    a=LSM-nm[1]-nm[12]-nm[13];
    if((0<a)&&(a<17)){
        b=LSM-nm[16]-nm[5]-nm[4];
        if(a+b==PSM){
            if((uflg[a]==0)&&(uflg[b]==0)){
                uflg[a]=1; uflg[b]=1;
                nm[8]=a; nm[9]=b;
                sstp08();
                uflg[b]=0; uflg[a]=0;}}}
}
/* Set n7=LSM-n12-n6-n9 & n10 */
void sstp08(){
    short a,b,c,d;
    a=LSM-nm[12]-nm[6]-nm[9];
    if((0<a)&&(a<17)){
        b=LSM-nm[8]-nm[11]-nm[5];
        if(a+b==PSM){
            c=LSM-nm[14]-nm[11]-nm[2];
            d=LSM-nm[15]-nm[6]-nm[3];
            if((a==c)&&(b==d)){
                if((uflg[a]==0)&&(uflg[b]==0)){
                    nm[7]=a; nm[10]=b;
                    uflg[a]=1; uflg[b]=1;
                    if(nm[4]+nm[7]+nm[10]+nm[13]==LSM){recpspr();}
                    uflg[b]=0; uflg[a]=0;}}}}
}
/**/
/* Record the Solutions and Print */
void recpspr(){
    short n;
    for(n=1;n<17;n++){d[cnt3][n]=nm[n]; pts[cnt1][n]=nm[n];}
    pts[cnt1][0]=cnt1+1;
    cnt1++; d[cnt3][0]=cnt1;
    //cnt3++; if(cnt3==6){pr6ans(); cnt3=0;}
}
/**/
/* Print 3 Pairs of Solutions */

```

```

void pr6ans(){
short l,l4,m,n;
printf("%11d/%13d/%13d/%13d/%13d/%13d\n",
d[0][0],d[1][0],d[2][0],d[3][0],d[4][0],d[5][0]);
for(l=0;l<4;l++){l4=l*4;
for(m=0;m<6;m++){
for(n=1;n<5;n++){printf("%3d",d[m][l4+n]);}
if(m<5){printf(" ");}
}
printf("\n");
}
}
/**/
/* Make 'Prototype Squares' for Complete Pan-magic Type */
/* Set N1 & n16=PSM-N1 */
void cstp01(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[1]=a; nm[16]=b;
uflg[a]=1; uflg[b]=1;
cstp02();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set N15 & n2=PSM-N15 */
void cstp02(){
short a,b;
for(a=16;a>0;a--){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[15]=a; nm[2]=b;
uflg[a]=1; uflg[b]=1;
cstp03();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set N4 & n13=PSM-N4 */
void cstp03(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
uflg[a]=1; uflg[b]=1;
nm[4]=a; nm[13]=b;
cstp04();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set n14=LSM-n1-n15-n4 & n3 */
void cstp04(){
short a,b;
a=LSM-nm[1]-nm[15]-nm[4];
if((0<a)&&(a<17)){
b=LSM-nm[16]-nm[2]-nm[13];
if(a+b==PSM){
if((uflg[a]==0)&&(uflg[b]==0)){
uflg[a]=1; uflg[b]=1;
nm[14]=a; nm[3]=b;
cstp05();
uflg[b]=0; uflg[a]=0;}}}
}
/* Set N12 & n5=PSM-N12 */
void cstp05(){
short a,b;
for(a=16;a>0;a--){b=PSM-a;

```



```

        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[12]=a; nm[5]=b;
            uflg[a]=1; uflg[b]=1;
            cstp06();
            uflg[b]=0; uflg[a]=0;}
    }
}
/* Set N6 & n11=PSM-N6 */
void cstp06(){
    short a,b;
    for(a=1;a<17;a++){b=PSM-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            uflg[a]=1; uflg[b]=1;
            nm[6]=a; nm[11]=b;
            if(nm[1]+nm[6]+nm[16]+nm[11]==LSM){cstp07();}
            uflg[b]=0; uflg[a]=0;}
    }
}
/* Set n8=LSM-n1-n12-n13 & n9 */
void cstp07(){
    short a,b;
    a=LSM-nm[1]-nm[12]-nm[13];
    if((0<a)&&(a<17)){
        b=LSM-nm[16]-nm[5]-nm[4];
        if(a+b==PSM){
            if((uflg[a]==0)&&(uflg[b]==0)){
                uflg[a]=1; uflg[b]=1;
                nm[8]=a; nm[9]=b;
                if(nm[14]+nm[9]+nm[3]+nm[8]==LSM){cstp08();}
                uflg[b]=0; uflg[a]=0;}}
    }
}
/* Set n7=LSM-n12-n6-n9 & n10 */
void cstp08(){
    short a,b,c,d;
    a=LSM-nm[12]-nm[6]-nm[9];
    if((0<a)&&(a<17)){
        b=LSM-nm[5]-nm[11]-nm[8];
        if(a+b==PSM){
            c=LSM-nm[14]-nm[2]-nm[11];
            d=LSM-nm[3]-nm[15]-nm[6];
            if((a==c)&&(b==d)){
                if((uflg[a]==0)&&(uflg[b]==0)){
                    nm[7]=a; nm[10]=b;
                    uflg[a]=1; uflg[b]=1;
                    reccspr();
                    uflg[b]=0; uflg[a]=0;}}}}
    }
}
/**/
/* Print the Solutions */
void reccspr(){
    short n;
    for(n=1;n<17;n++){d[cnt3][n]=nm[n]; ptc[cnt2][n]=nm[n];}
    ptc[cnt2][0]=cnt2+1;
    cnt2++; d[cnt3][0]=cnt2;
//cnt3++; if(cnt3==6){pr6ans(); cnt3=0;}
}
/**/
/* Print 3 Pairs of Solutions */
void pr3pairs(short x){
    short l,l4,m,n,p;
    for(m=0;m<x;m=m+3){
        printf("%10d/S%4d/C%19d/S%4d/C%19d/S%4d/C\n",
            pts[m][0],ptc[m][0],pts[m+1][0],ptc[m+1][0],pts[m+2][0],ptc[m+2][0]);
        for(l=0;l<4;l++){l4=l*4;

```

```

        for(n=m;n<(m+3);n++){
            for(p=1;p<5;p++){printf("%3d",pts[n][l4+p]);}
            printf(" ");
            for(p=1;p<5;p++){printf("%3d",ptc[n][l4+p]);}
            if(n<(m+2)){printf(" ");}
        }
        printf("\n");
    }
}
}
/**/
/* Do-it-After-the-Model Transformation System */
/* Transform Prototype into Self-Complementary Object */
void damts(void){
    dm[1]=nm[1];    dm[2]=nm[15];    dm[3]=nm[14];    dm[4]=nm[4];
    dm[5]=nm[12];    dm[6]=nm[6];    dm[7]=nm[7];    dm[8]=nm[9];
    dm[9]=nm[8];    dm[10]=nm[10];    dm[11]=nm[11];    dm[12]=nm[5];
    dm[13]=nm[13];    dm[14]=nm[3];    dm[15]=nm[2];    dm[16]=nm[16];
}
/**/
/* Transform Prototype into Complete Pan-magic Object */
void damtc(void){
    dm[1]=nm[1];    dm[2]=nm[15];    dm[3]=nm[4];    dm[4]=nm[14];
    dm[5]=nm[12];    dm[6]=nm[6];    dm[7]=nm[9];    dm[8]=nm[7];
    dm[9]=nm[13];    dm[10]=nm[3];    dm[11]=nm[16];    dm[12]=nm[2];
    dm[13]=nm[8];    dm[14]=nm[10];    dm[15]=nm[5];    dm[16]=nm[11];
}
/**/
/* Print 2 Trios of Recomposed Solutions by DAMT [Primitive Set] */
void prp2trios(short x){
    short l,l4,m,n,p;
    for(m=0;m<x;m=m+2){
        for(n=1;n<17;n++){nm[n]=pts[m][n];}
        damts(); for(n=1;n<17;n++){d[0][n]=dm[n];}
        for(n=1;n<17;n++){nm[n]=ptc[m][n];}
        damtc(); for(n=1;n<17;n++){d[1][n]=dm[n];}
        for(n=1;n<17;n++){nm[n]=pts[m+1][n];}
        damts(); for(n=1;n<17;n++){d[2][n]=dm[n];}
        for(n=1;n<17;n++){nm[n]=ptc[m+1][n];}
        damtc(); for(n=1;n<17;n++){d[3][n]=dm[n];}
        printf("%10d/P          /S          /C%12d/P          /S          /C\n",
            pts[m][0],pts[m+1][0]);
        for(l=0;l<4;l++){l4=l*4;
            for(p=1;p<5;p++){printf("%3d",pts[m][l4+p]);}
            printf(" ");
            for(p=1;p<5;p++){printf("%3d",d[0][l4+p]);}
            printf(" ");
            for(p=1;p<5;p++){printf("%3d",d[1][l4+p]);}
            printf(" ");
            for(p=1;p<5;p++){printf("%3d",pts[m+1][l4+p]);}
            printf(" ");
            for(p=1;p<5;p++){printf("%3d",d[2][l4+p]);}
            printf(" ");
            for(p=1;p<5;p++){printf("%3d",d[3][l4+p]);}
            printf("\n");
        }
    }
}
}
}
/**/
/* Print 2 Trios of Recomposed Solutions by DAMT [Standard Set] */
void prs2trios(){
    short l,l4,m,n,p;
    for(m=0;m<tcnt;m=m+2){
        for(n=0;n<4;n++){for(p=0;p<17;p++){d[n][p]=0;}}
    }
}
}
}

```

```

for(n=1;n<17;n++){nm[n]=pts[m][n];}
damts();
if((dm[1]<dm[4])&&(dm[1]<dm[13])&&(dm[1]<dm[16])&&(dm[5]<dm[2])){
    cnt1++; d[0][0]=cnt1; for(n=1;n<17;n++){d[0][n]=dm[n];}}
for(n=1;n<17;n++){nm[n]=ptc[m][n];}
damtc();
if((dm[1]<dm[4])&&(dm[1]<dm[13])&&(dm[1]<dm[16])&&(dm[5]<dm[2])){
    cnt2++; d[1][0]=cnt2; for(n=1;n<17;n++){d[1][n]=dm[n];}}
for(n=1;n<17;n++){nm[n]=pts[m+1][n];}
damts();
if((dm[1]<dm[4])&&(dm[1]<dm[13])&&(dm[1]<dm[16])&&(dm[5]<dm[2])){
    cnt1++; d[2][0]=cnt1; for(n=1;n<17;n++){d[2][n]=dm[n];}}
for(n=1;n<17;n++){nm[n]=ptc[m+1][n];}
damtc();
if((dm[1]<dm[4])&&(dm[1]<dm[13])&&(dm[1]<dm[16])&&(dm[5]<dm[2])){
    cnt2++; d[3][0]=cnt2; for(n=1;n<17;n++){d[3][n]=dm[n];}}
printf("%10d/P%11d/S%11d/C%12d/P%11d/S%11d/C\n",
    pts[m][0],d[0][0],d[1][0],pts[m+1][0],d[2][0],d[3][0]);
if((d[0][0]>0)||(d[1][0]>0)||(d[2][0]>0)||(d[3][0]>0)){
    for(l=0;l<4;l++){l4=l*4;
        for(p=1;p<5;p++){printf("%3d",pts[m][l4+p]);}
        printf(" ");
        for(p=1;p<5;p++){printf("%3d",d[0][l4+p]);}
        printf(" ");
        for(p=1;p<5;p++){printf("%3d",d[1][l4+p]);}
        printf(" ");
        for(p=1;p<5;p++){printf("%3d",pts[m+1][l4+p]);}
        printf(" ");
        for(p=1;p<5;p++){printf("%3d",d[2][l4+p]);}
        printf(" ");
        for(p=1;p<5;p++){printf("%3d",d[3][l4+p]);}
        printf("\n");
    }
}
}
}
}
/**/

```