Chapter 6: Fundamental Studies of Multi-Dimensional Extra-Cubic Magic Forms and their Decompositions: Kanji Setsuda

Section 3-5: Multiple Type of Composite Pan-magic Squares of Order 8 and Its Transformation System

#1. We have studied about 6-dimensional extra-cubic magic forms of order 2 and their decompositions. We have obtained 15 fundamental solutions of 'Composite and Complete' Magic Cubes of order 4 and 10 fundamental solutions of 'C&C' Magic Squares of order 8 by dimension-converting down from the origin.
In the former section I showed we could reconstruct all 90 solutions of C&C Magic Squares of order 8 from the Fundamental Ten by our original transformation system.

What we have to do next is analysing 360 solutions of Composite and Pan-Diagonal magic squares of order 8 and finding out which family they belong to.
Can we say that they do really belong to the same family as our 90 solutions of 'C&C' magic squares do? Can we demonstrate any evidence for that? Can we find any transformation system for making the Fundamental Ten or 90 of 'C&C' into all 360 of 'C&P'?

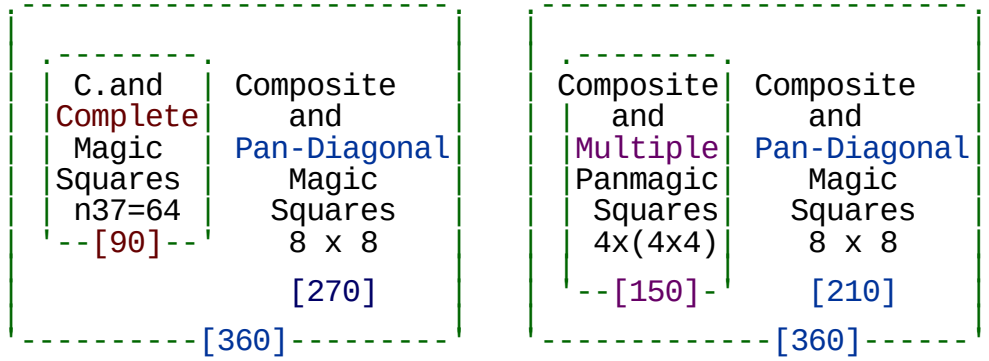
#2. We know the set of 90 solutions of 'C&C' is the subset of 360 solutions of 'C&P' magic squares of order 8. But we don't yet know if the other 270 solutions also belong to the same noble family whose origin is that 6-dimensional extra-cubic object of order 2.

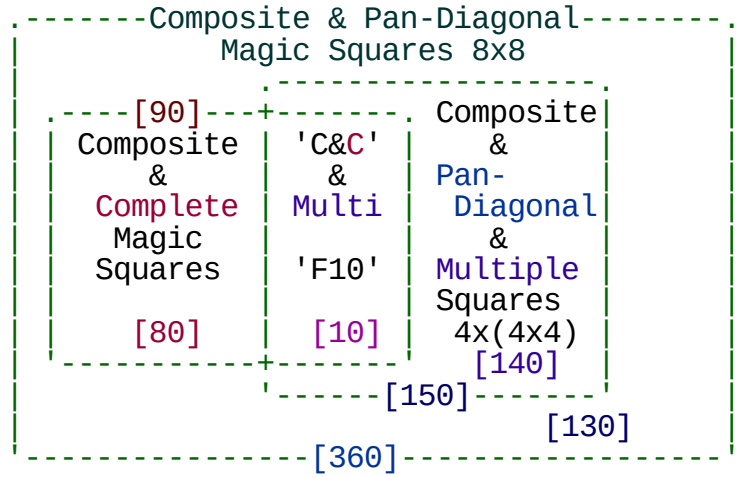
I often tried to study 360 and to find any family tree of them in vain. It was too heavy for me to work on it, because the scale of everything was too big and comparison of each to the others was too hard for me to go on. It takes too much time to do any job.
I tried it many times, but gave it up all times.

I decided to change my goal object to the smaller one in scale.

This time I am going to take the 150 solutions of “multiple” type of 'C&P' magic sqaures of order 8. Each of them always includes 4 squares of order 4 within itself. They make the subset of C&P magic squares of order 8. And it is more important, I think, that the set of F10 is a part of the set of those 150. We can now expect all 150 solutions could be made from the F10, can't we?

***** Solution Set and Subsets of *****
'Composite and Pan-Diagonal' Magic Squares 8×8





#3. Let me define all objects we are discussing here with some simultaneous equations as usual.
(1) I mean any 'Composite' type of magic squares of order 8 is ruled by such conditions as follows:

****'Composite' Type of Magic Squares 8x8****
***** Basic Form and Basic Conditions *****

62	63	64	57	58	59	60	61	62	63	64	57	58	59	60	61
6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5
14	15	16	9	10	11	12	13	14	15	16	9	10	11	12	13
22	23	24	17	18	19	20	21	22	23	24	17	18	19	20	21
30	31	32	25	26	27	28	29	30	31	32	25	26	27	28	29
38	39	40	33	34	35	36	37	38	39	40	33	34	35	36	37
46	47	48	41	42	43	44	45	46	47	48	41	42	43	44	45
54	55	56	49	50	51	52	53	54	55	56	49	50	51	52	53
62	63	64	57	58	59	60	61	62	63	64	57	58	59	60	61
6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5

'Composite' Conditions:

n1+n2+n9+n10=130; n2+n3+n10+n11=130; n3+n4+n11+n12=130; n4+n5+n12+n13=130;
n5+n6+n13+n14=130; n6+n7+nn14+n15=130; n7+n8+n15+n16=130; n8+n1+n16+n9=130
n9+n10+n17+n18=130; n10+n11+n18+n19=130; n11+n12+n19+n20=130; n12+n13+n20+n21=130;
n13+n14+n21+n22=130; n14+n15+n22+n23=130; n15+n16+n23+n24=130; n16+n9+n24+n17=130
n17+n18+n25+n26=130; n18+n19+n26+n27=130; n19+n20+n27+n28=130; n20+n21+n28+n29=130;
n21+n22+n29+n30=130; n22+n23+n30+n31=130; n23+n24+n31+n32=130; n24+n17+n32+n25=130
n25+n26+n33+n34=130; n26+n27+n34+n35=130; n27+n28+n35+n36=130; n28+n29+n36+n37=130;
n29+n30+n37+n38=130; n30+n31+n38+n39=130; n31+n32+n39+n40=130; n32+n25+n40+n33=130
n33+n34+n41+n42=130; n34+n35+n42+n43=130; n35+n36+n43+n44=130; n36+n37+n45+n46=130;
n37+n38+n45+n46=130; n38+n39+n46+n47=130; n39+n40+n47+n48=130; n40+n33+n48+n41=130
n41+n42+n49+n50=130; n42+n43+n50+n51=130; n43+n44+n51+n52=130; n44+n45+n52+n53=130;
n45+n46+n53+n54=130; n46+n47+n54+n55=130; n47+n48+n55+n56=130; n48+n41+n56+n49=130
n49+n50+n57+n58=130; n50+n51+n58+n59=130; n51+n52+n59+n60=130; n52+n53+n60+n61=130;
n53+n54+n61+n62=130; n54+n55+n62+n63=130; n55+n56+n63+n64=130; n56+n49+n64+n57=130
n57+n58+n1+n2=130; n58+n59+n2+n3=130; n59+n60+n3+n4=130; n60+n61+n4+n5=130;
n61+n62+n5+n6=130; n62+n63+n6+n7=130; n63+n64+n7+n8=130; n64+n57+n8+n1=65

If you have all of these 'Composite' conditions colored black above, you don't have to have the rest equations colored green at first, because you can calculate them afterward and find they are always true.
(2) If you want to have the ‘ Composite and Pan-Diagonal’ magic type, you must add the next three conditions at least.

Additional Conditions:

```
n1+n2+n3+n4+n5+n6+n7+n8=260;
n1+n9+n17+n25+n33+n41+n49+n57=260;
n1+n10+n19+n28+n37+n46+n55+n64=260;
```

You don't have to assume all those pandiagonal equations below at first, because they are always true if you have all these 'Composite' conditions and three additional equations above.

But in your program sheets, the equations below may sometimes be so useful for you to find your answers quickly. You had better use them whenever you might have a chance to write them, even for your check procedures in your program.

Pan-Diagonals:

```
n1+n10+n19+n28+n37+n46+n55+n64=260  ...(pd1)
n1+n16+n23+n30+n37+n44+n51+n58=260  ...(pd2)
n2+n11+n20+n29+n38+n47+n56+n57=260  ...(pd3)
n2+n9+n24+n31+n38+n45+n52+n59=260   ...(pd4)
n3+n12+n21+n30+n39+n48+n49+n58=260  ...(pd5)
n3+n10+n17+n32+n39+n46+n53+n60=260  ...(pd6)
n4+n13+n22+n31+n40+n41+n50+n59=260  ...(pd7)
n4+n11+n18+n25+n40+n47+n54+n61=260  ...(pd8)
n5+n14+n23+n32+n33+n42+n51+n60=260  ...(pd9)
n5+n12+n19+n26+n33+n48+n55+n62=260 ...(pd10)
n6+n15+n24+n25+n34+n43+n52+n61=260 ...(pd11)
n6+n13+n20+n27+n34+n41+n56+n63=260 ...(pd12)
n7+n16+n17+n26+n35+n44+n53+n62=260 ...(pd13)
n7+n14+n21+n28+n35+n42+n49+n64=260 ...(pd14)
n8+n9+n18+n27+n36+n45+n54+n63=260  ...(pd15)
n8+n15+n22+n29+n36+n43+n50+n57=260 ...(pd16)
```

(3) I mean the 'Complete' type of magic squares as a little different from any 'Pan-diagonal' magic ones.
In such a lower order as 4, both 'Complete' and 'Pandiagonal' mean the same thing, since they always take the same forms however different you might define them. But in such a higher order as 8, they do always appear in different forms.
If you want to have any 'Composite & Complete' magic square, you must always add the next equations as follows:

‘Complete’Conditions:

```
   n1+n37 = n2+n38 = n3+n39 = n4+n40 = n5+n33 = n6+n34
 = n7+n35 = n8+n36 = n9+n45 = n10+n46 = n11+n47 = n12+n48
 = n13+n41 = n14+n42 = n15+n43 = n16+n36 = n17+n53 = n18+n54
 = n19+n55 = n20+n56 = n21+n49 = n22+n50 = n23+n51 = n24+n52
 = n25+n61 = n26+n62 = n27+n63 = n28+n64 = n29+n57 = n30+n58
 = n31+n59 = n32+n60 = 65
```

They are quite similar to the definition of 'Self-Complementary' type. Yes, they stand for the constant sum of all Complementary pairs, but each Complementary pair is located in quite a different way. Since there is a definite correspondence between them, you can transform any Complete type into Self-complementary one or make any Self-complementary type into Complete in reverse, whenever you want to.

Any 'Complete' magic square must have n37=64; because n1+n37=65; and n1=1 in our 'standard' form. It is always very important for any pan-magic square what value n37 really takes.

The solution set of 'Complete' magic squares of order 8 makes the subset of 'Pan-diagonal' magic ones. It means that a 'Complete' magic square is a special type of Pandiagonal ones.

(4) If you want to have any 'Multiple' type of panmagic square, you must add the next two conditions at least.

'Multiple'Conditions:

n1+n2+n3+n4=130; n1+n9+n17+n25=130;
n9+n10+n11+n12=130; n2+n10+n18+n26=130; n17+n18+n19+n20=130;
n3+n11+n19+n27=130;
n25+n26+n27+n28=130; n4+n12+n20+n28=130;
n5+n6+n7+n8=130; n33+n41+n49+n57=130;
n13+n14+n15+n16=130; n5+n13+n21+n29=130;
. . . . .

If you add these two equations to the Composite conditions above, you can surely have the one with four squares of order 4 included within itself, although any pan-diagonal of it does not always add up to 130.

You may add such equations above as colored green to the two, but you don't have to do it. You may well to use them so to improve your program for working quickly.

#4. The new object goal of ours is the solution set of multiple type of C&P magic squares of order 8. It is shown in the next list with as many as 150 solutions.

The last 10 solutions of them is certainly the Fundamental Ten with n37=64. They are naturally the same with the F10 taken out of 6-dimensional extra-cubic object of order 2.

How can we discover and invent our original transformation system of making these F10 into all those 150 solutions of multiple C&P?

[Multiple Type of 'Composite' Pan-Magic Squares 8x8
Solution List: All of 150 Multiple 'C&P' MS88]

(If you want to have got the full size list, please click here.)

```
                  1/52                          2/52                          3/54
 1 64  2 63  4 61  3 62    1 64  2 63  4 61  3 62    1 64  2 63  6 59  5 60
60  5 59  6 57  8 58  7   56  9 55 10 53 12 54 11   62  3 61  4 57  8 58  7
29 36 30 35 32 33 31 34   29 36 30 35 32 33 31 34   27 38 28 37 32 33 31 34
40 25 39 26 37 28 38 27   44 21 43 22 41 24 42 23   40 25 39 26 35 30 36 29
49 16 50 15 52 13 51 14   49 16 50 15 52 13 51 14   49 16 50 15 54 11 53 12
12 53 11 54  9 56 10 55    8 57  7 58  5 60  6 59   14 51 13 52  9 56 10 55
45 20 46 19 48 17 47 18   45 20 46 19 48 17 47 18   43 22 44 21 48 17 47 18
24 41 23 42 21 44 22 43   28 37 27 38 25 40 26 39   24 41 23 42 19 46 20 45
```

```
                    4/54                    5/55                    6/55
 1 64  2 63  6 59  5 60    1 64  3 62  7 58  5 60    1 64 25 40 49 16 41 24
56  9 55 10 51 14 52 13   63  2 61  4 57  8 59  6   63  2 39 26 15 50 23 42
27 38 28 37 32 33 31 34   26 39 28 37 32 33 30 35    4 61 28 37 52 13 44 21
46 19 45 20 41 24 42 23   40 25 38 27 34 31 36 29   62  3 38 27 14 51 22 43
49 16 50 15 54 11 53 12   49 16 51 14 55 10 53 12    7 58 31 34 55 10 47 18
 8 57  7 58  3 62  4 61   15 50 13 52  9 56 11 54   57  8 33 32  9 56 17 48
43 22 44 21 48 17 47 18   42 23 44 21 48 17 46 19    6 59 30 35 54 11 46 19
30 35 29 36 25 40 26 39   24 41 22 43 18 47 20 45   60  5 36 29 12 53 20 45
                    7/55                    8/55                    9/55
 1 64  3 62  7 58  5 60    1 64  4 61  7 58  6 59    1 63  4 62  7 57  6 60
56  9 54 11 50 15 52 13   56  9 53 12 50 15 51 14   56 10 53 11 50 16 51 13
26 39 28 37 32 33 30 35   25 40 28 37 31 34 30 35   25 39 28 38 31 33 30 36
47 18 45 20 41 24 43 22   48 17 45 20 42 23 43 22   48 18 45 19 42 24 43 21
49 16 51 14 55 10 53 12   49 16 52 13 55 10 54 11   49 15 52 14 55  9 54 12
 8 57  6 59  2 63  4 61    8 57  5 60  2 63  3 62    8 58  5 59  2 64  3 61
42 23 44 21 48 17 46 19   41 24 44 21 47 18 46 19   41 23 44 22 47 17 46 20
31 34 29 36 25 40 27 38   32 33 29 36 26 39 27 38   32 34 29 35 26 40 27 37
                   10/56                   11/56                   12/56
 1 64  3 62  7 58  5 60    1 64 25 40 49 16 41 24    1 64  2 63  6 59  5 60
63  2 61  4 57  8 59  6   63  2 39 26 15 50 23 42   62  3 61  4 57  8 58  7
26 39 28 37 32 33 30 35    4 61 28 37 52 13 44 21   27 38 28 37 32 33 31 34
40 25 38 27 34 31 36 29   62  3 38 27 14 51 22 43   40 25 39 26 35 30 36 29
50 15 52 13 56  9 54 11    8 57 32 33 56  9 48 17   51 14 52 13 56  9 55 10
16 49 14 51 10 55 12 53   58  7 34 31 10 55 18 47   16 49 15 50 11 54 12 53
41 24 43 22 47 18 45 20    5 60 29 36 53 12 45 20   41 24 42 23 46 19 45 20
23 42 21 44 17 48 19 46   59  6 35 30 11 54 19 46   22 43 21 44 17 48 18 47
. . . . . .
                  139/63                  140/63                  141/64
 1 64  3 62 11 54  9 56    1 64  3 62  7 58  5 60    1 63  6 60 22 44 17 47
59  6 57  8 49 16 51 14   55 10 53 12 49 16 51 14   62  4 57  7 41 23 46 20
22 43 24 41 32 33 30 35   26 39 28 37 32 33 30 35   11 53 16 50 32 34 27 37
48 17 46 19 38 27 40 25   48 17 46 19 42 23 44 21   56 10 51 13 35 29 40 26
53 12 55 10 63  2 61  4   57  8 59  6 63  2 61  4   43 21 48 18 64  2 59  5
15 50 13 52  5 60  7 58   15 50 13 52  9 56 11 54   24 42 19 45  3 61  8 58
34 31 36 29 44 21 42 23   34 31 36 29 40 25 38 27   33 31 38 28 54 12 49 15
28 37 26 39 18 47 20 45   24 41 22 43 18 47 20 45   30 36 25 39  9 55 14 52
                  142/64                  143/64                  144/64
 1 63  6 60 38 28 33 31    1 63  6 60 14 52  9 55    1 63 10 56 26 40 17 47
62  4 57  7 25 39 30 36   62  4 57  7 49 15 54 12   62  4 53 11 37 27 46 20
11 53 16 50 48 18 43 21   19 45 24 42 32 34 27 37    7 57 16 50 32 34 23 41
56 10 51 13 19 45 24 42   48 18 43 21 35 29 40 26   60  6 51 13 35 29 44 22
27 37 32 34 64  2 59  5   51 13 56 10 64  2 59  5   39 25 48 18 64  2 55  9
40 26 35 29  3 61  8 58   16 50 11 53  3 61  8 58   28 38 19 45  3 61 12 54
17 47 22 44 54 12 49 15   33 31 38 28 46 20 41 23   33 31 42 24 58  8 49 15
46 20 41 23  9 55 14 52   30 36 25 39 17 47 22 44   30 36 21 43  5 59 14 52
                  145/64                  146/64                  147/64
 1 63 10 56 42 24 33 31    1 63 18 48 50 16 33 31    1 63  4 62 20 46 17 47
62  4 53 11 21 43 30 36   62  4 45 19 13 51 30 36   60  6 57  7 41 23 44 22
 7 57 16 50 48 18 39 25    7 57 24 42 56 10 39 25   13 51 16 50 32 34 29 35
60  6 51 13 19 45 28 38   60  6 43 21 11 53 28 38   56 10 53 11 37 27 40 26
23 41 32 34 64  2 55  9   15 49 32 34 64  2 47 17   45 19 48 18 64  2 61  3
44 22 35 29  3 61 12 54   52 14 35 29  3 61 20 46   24 42 21 43  5 59  8 58
17 47 26 40 58  8 49 15    9 55 26 40 58  8 41 23   33 31 36 30 52 14 49 15
46 20 37 27  5 59 14 52   54 12 37 27  5 59 22 44   28 38 25 39  9 55 12 54
                  148/64                  149/64                  150/64
 1 63  4 62 36 30 33 31    1 63  4 62 12 54  9 55    1 63  4 62  8 58  5 59
60  6 57  7 25 39 28 38   60  6 57  7 49 15 52 14   56 10 53 11 49 15 52 14
13 51 16 50 48 18 45 19   21 43 24 42 32 34 29 35   25 39 28 38 32 34 29 35
56 10 53 11 21 43 24 42   48 18 45 19 37 27 40 26   48 18 45 19 41 23 44 22
29 35 32 34 64  2 61  3   53 11 56 10 64  2 61  3   57  7 60  6 64  2 61  3
40 26 37 27  5 59  8 58   16 50 13 51  5 59  8 58   16 50 13 51  9 55 12 54
17 47 20 46 52 14 49 15   33 31 36 30 44 22 41 23   33 31 36 30 40 26 37 27
44 22 41 23  9 55 12 54   28 38 25 39 17 47 20 46   24 42 21 43 17 47 20 46
```

[2/52 2/54 5/55 7/56 6/58 15/59 17/60 30/61 26/62 30/63 **10/64**]

#5. How can we make the F10 into all 150 solutions?

We need such a certain system of transformations as found when we once made F10 into 90 solutions of 'C&C' magic squares of order 8.

And for our tools we must have the set of "exchange-a-half" transformation methods in which we can exchange only 32 elements of each of the two objects compared and we must keep the rest half 32 unchanged at all. We need such a powerful method as to transform many objects in the same common way.

(1) Find any similar solution to each of F10 out of 140 solutions list. Compare each one to the other and check if a half of 64 elements are the same. Collect those similar pairs.

(2) Watch and examine them, and collect the only pairs whose 'Unmatched' patterns are the same and common. Make them classified into each group of the same pattern.

(3) Imagine any common transformation method for each group, and check if it works well. And you can discover and invent your transformation methods at last.

(4) If you could find so few pairs as to transform each one into the other, make yourself think of the next 'generation'. You may well make some granddaughters from each of the first daughters.

Let me show you my recent result of first comparisons in the next list.

[Multiple Type of 'Composite' Pan-Magic Squares 8x8
Find Any Similar Object to the Fundamental Ten]

```
[ 1]                141/64      111/63
    1 63  6 60 22 44 17 47       1 64  5 60 21 44 17 48
   62  4 57  7 41 23 46 20      62  3 58  7 42 23 46 19
   11 53 16 50 32 34 27 37      12 53 16 49 32 33 28 37
   56 10 51 13 35 29 40 26      55 10 51 14 35 30 39 26
   43 21 48 18 64  2 59  5      43 22 47 18 63  2 59  6
   24 42 19 45  3 61  8 58      24 41 20 45  4 61  8 57
   33 31 38 28 54 12 49 15      34 31 38 27 54 11 50 15
   30 36 25 39  9 55 14 52      29 36 25 40  9 56 13 52
                   <-- Unmatched -->
     * 63  6  * 22  *  * 47     * 64  5  * 21  *  * 48
     *  4 57  * 41  *  * 20     *  3 58  * 42  *  * 19
    11  *  * 50  * 34 27  *    12  *  * 49  * 33 28  *
    56  *  * 13  * 29 40  *    55  *  * 14  * 30 39  *
     * 21 48  * 64  *  *  5     * 22 47  * 63  *  *  6
     * 42 19  *  3  *  * 58     * 41 20  *  4  *  * 57
    33  *  * 28  * 12 49  *    34  *  * 27  * 11 50  *
    30  *  * 39  * 55 14  *    29  *  * 40  * 56 13  *

[ 2]                141/64      121/63
    1 63  6 60 22 44 17 47       1 64  6 59 21 44 18 47
   62  4 57  7 41 23 46 20      62  3 57  8 42 23 45 20
   11 53 16 50 32 34 27 37      11 54 16 49 31 34 28 37
   56 10 51 13 35 29 40 26      56  9 51 14 36 29 39 26
   43 21 48 18 64  2 59  5      43 22 48 17 63  2 60  5
   24 42 19 45  3 61  8 58      24 41 19 46  4 61  7 58
   33 31 38 28 54 12 49 15      33 32 38 27 53 12 50 15
   30 36 25 39  9 55 14 52      30 35 25 40 10 55 13 52
                   <-- Unmatched -->
     * 63  * 60 22  * 17  *     * 64  * 59 21  * 18  *
     *  4  *  7 41  * 46  *     *  3  *  8 42  * 45  *
     * 53  * 50 32  * 27  *     * 54  * 49 31  * 28  *
     * 10  * 13 35  * 40  *     *  9  * 14 36  * 39  *
     * 21  * 18 64  * 59  *     * 22  * 17 63  * 60  *
     * 42  * 45  3  *  8  *     * 41  * 46  4  *  7  *
     * 31  * 28 54  * 49  *     * 32  * 27 53  * 50  *
     * 36  * 39  9  * 14  *     * 35  * 40 10  * 13  *
```

```
[ 3]                141/64     131/63
    1 63  6 60 22 44 17 47      1 64  5 60 21 44 17 48
   62  4 57  7 41 23 46 20     61  4 57  8 41 24 45 20
   11 53 16 50 32 34 27 37     12 53 16 49 32 33 28 37
   56 10 51 13 35 29 40 26     56  9 52 13 36 29 40 25
   43 21 48 18 64  2 59  5     43 22 47 18 63  2 59  6
   24 42 19 45  3 61  8 58     23 42 19 46  3 62  7 58
   33 31 38 28 54 12 49 15     34 31 38 27 54 11 50 15
   30 36 25 39  9 55 14 52     30 35 26 39 10 55 14 51
                  <-- Unmatched -->
     * 63  6  * 22  *  * 47     * 64  5  * 21  *  * 48
    62  *  *  7  * 23 46  *    61  *  *  8  * 24 45  *
    11  *  * 50  * 34 27  *    12  *  * 49  * 33 28  *
     * 10 51  * 35  *  * 26     *  9 52  * 36  *  * 25
     * 21 48  * 64  *  *  5     * 22 47  * 63  *  *  6
    24  *  * 45  * 61  8  *    23  *  * 46  * 62  7  *
    33  *  * 28  * 12 49  *    34  *  * 27  * 11 50  *
     * 36 25  *  9  *  * 52     * 35 26  * 10  *  * 51

[ 4]                142/64     112/63
    1 63  6 60 38 28 33 31      1 64  5 60 37 28 33 32
   62  4 57  7 25 39 30 36     62  3 58  7 26 39 30 35
   11 53 16 50 48 18 43 21     12 53 16 49 48 17 44 21
   56 10 51 13 19 45 24 42     55 10 51 14 19 46 23 42
   27 37 32 34 64  2 59  5     27 38 31 34 63  2 59  6
   40 26 35 29  3 61  8 58     40 25 36 29  4 61  8 57
   17 47 22 44 54 12 49 15     18 47 22 43 54 11 50 15
   46 20 41 23  9 55 14 52     45 20 41 24  9 56 13 52
                  <-- Unmatched -->
     * 63  6  * 38  *  * 31     * 64  5  * 37  *  * 32
     *  4 57  * 25  *  * 36     *  3 58  * 26  *  * 35
    11  *  * 50  * 18 43  *    12  *  * 49  * 17 44  *
    56  *  * 13  * 45 24  *    55  *  * 14  * 46 23  *
     * 37 32  * 64  *  *  5     * 38 31  * 63  *  *  6
     * 26 35  *  3  *  * 58     * 25 36  *  4  *  * 57
    17  *  * 44  * 12 49  *    18  *  * 43  * 11 50  *
    46  *  * 23  * 55 14  *    45  *  * 24  * 56 13  *

[ 5]                142/64     122/63
    1 63  6 60 38 28 33 31      1 64  6 59 37 28 34 31
   62  4 57  7 25 39 30 36     62  3 57  8 26 39 29 36
   11 53 16 50 48 18 43 21     11 54 16 49 47 18 44 21
   56 10 51 13 19 45 24 42     56  9 51 14 20 45 23 42
   27 37 32 34 64  2 59  5     27 38 32 33 63  2 60  5
   40 26 35 29  3 61  8 58     40 25 35 30  4 61  7 58
   17 47 22 44 54 12 49 15     17 48 22 43 53 12 50 15
   46 20 41 23  9 55 14 52     46 19 41 24 10 55 13 52
                  <-- Unmatched -->
     * 63  * 60 38  * 33  *     * 64  * 59 37  * 34  *
     *  4  *  7 25  * 30  *     *  3  *  8 26  * 29  *
     * 53  * 50 48  * 43  *     * 54  * 49 47  * 44  *
     * 10  * 13 19  * 24  *     *  9  * 14 20  * 23  *
     * 37  * 34 64  * 59  *     * 38  * 33 63  * 60  *
     * 26  * 29  3  *  8  *     * 25  * 30  4  *  7  *
     * 47  * 44 54  * 49  *     * 48  * 43 53  * 50  *
     * 20  * 23  9  * 14  *     * 19  * 24 10  * 13  *

[ 5}                142/64       <-- Unmatched -->      122/63
    1 63  6 60 38 28 33 31     * 63  * 60 38  * 33  *     1 64  6 59 37 28 34 31
   62  4 57  7 25 39 30 36     *  4  *  7 25  * 30  *    62  3 57  8 26 39 29 36
   11 53 16 50 48 18 43 21     * 53  * 50 48  * 43  *    11 54 16 49 47 18 44 21
   56 10 51 13 19 45 24 42     * 10  * 13 19  * 24  *    56  9 51 14 20 45 23 42
   27 37 32 34 64  2 59  5     * 37  * 34 64  * 59  *    27 38 32 33 63  2 60  5
   40 26 35 29  3 61  8 58     * 26  * 29  3  *  8  *    40 25 35 30  4 61  7 58
   17 47 22 44 54 12 49 15     * 47  * 44 54  * 49  *    17 48 22 43 53 12 50 15
   46 20 41 23  9 55 14 52     * 20  * 23  9  * 14  *    46 19 41 24 10 55 13 52
```

```
[ 6}                142/64      <-- Unmatched -->     132/63
    1 63  6 60 38 28 33 31    * 63  6  * 38  *  * 31    1 64  5 60 37 28 33 32
   62  4 57  7 25 39 30 36   62  *  *  7  * 39 30  *   61  4 57  8 25 40 29 36
   11 53 16 50 48 18 43 21   11  *  * 50  * 18 43  *   12 53 16 49 48 17 44 21
   56 10 51 13 19 45 24 42    * 10 51  * 19  *  * 42   56  9 52 13 20 45 24 41
   27 37 32 34 64  2 59  5    * 37 32  * 64  *  *  5   27 38 31 34 63  2 59  6
   40 26 35 29  3 61  8 58   40  *  * 29  * 61  8  *   39 26 35 30  3 62  7 58
   17 47 22 44 54 12 49 15   17  *  * 44  * 12 49  *   18 47 22 43 54 11 50 15
   46 20 41 23  9 55 14 52    * 20 41  *  9  *  * 52   46 19 42 23 10 55 14 51
[ 7}                143/64      <-- Unmatched -->     113/63
    1 63  6 60 14 52  9 55    * 63  6  * 14  *  * 55    1 64  5 60 13 52  9 56
   62  4 57  7 49 15 54 12    *  4 57  * 49  *  * 12   62  3 58  7 50 15 54 11
   19 45 24 42 32 34 27 37   19  *  * 42  * 34 27  *   20 45 24 41 32 33 28 37
   48 18 43 21 35 29 40 26   48  *  * 21  * 29 40  *   47 18 43 22 35 30 39 26
   51 13 56 10 64  2 59  5    * 13 56  * 64  *  *  5   51 14 55 10 63  2 59  6
   16 50 11 53  3 61  8 58    * 50 11  *  3  *  * 58   16 49 12 53  4 61  8 57
   33 31 38 28 46 20 41 23   33  *  * 28  * 20 41  *   34 31 38 27 46 19 42 23
   30 36 25 39 17 47 22 44   30  *  * 39  * 47 22  *   29 36 25 40 17 48 21 44
[ 8}                143/64      <-- Unmatched -->     123/63
    1 63  6 60 14 52  9 55    * 63  * 60 14  *  9  *    1 64  6 59 13 52 10 55
   62  4 57  7 49 15 54 12    *  4  *  7 49  * 54  *   62  3 57  8 50 15 53 12
   19 45 24 42 32 34 27 37    * 45  * 42 32  * 27  *   19 46 24 41 31 34 28 37
   48 18 43 21 35 29 40 26    * 18  * 21 35  * 40  *   48 17 43 22 36 29 39 26
   51 13 56 10 64  2 59  5    * 13  * 10 64  * 59  *   51 14 56  9 63  2 60  5
   16 50 11 53  3 61  8 58    * 50  * 53  3  *  8  *   16 49 11 54  4 61  7 58
   33 31 38 28 46 20 41 23    * 31  * 28 46  * 41  *   33 32 38 27 45 20 42 23
   30 36 25 39 17 47 22 44    * 36  * 39 17  * 22  *   30 35 25 40 18 47 21 44
[ 9}                143/64      <-- Unmatched -->     133/63
    1 63  6 60 14 52  9 55    * 63  6  * 14  *  * 55    1 64  5 60 13 52  9 56
   62  4 57  7 49 15 54 12   62  *  *  7  * 15 54  *   61  4 57  8 49 16 53 12
   19 45 24 42 32 34 27 37   19  *  * 42  * 34 27  *   20 45 24 41 32 33 28 37
   48 18 43 21 35 29 40 26    * 18 43  * 35  *  * 26   48 17 44 21 36 29 40 25
   51 13 56 10 64  2 59  5    * 13 56  * 64  *  *  5   51 14 55 10 63  2 59  6
   16 50 11 53  3 61  8 58   16  *  * 53  * 61  8  *   15 50 11 54  3 62  7 58
   33 31 38 28 46 20 41 23   33  *  * 28  * 20 41  *   34 31 38 27 46 19 42 23
   30 36 25 39 17 47 22 44    * 36 25  * 17  *  * 44   30 35 26 39 18 47 22 43
[10}                144/64      <-- Unmatched -->     114/63
    1 63 10 56 26 40 17 47    * 63 10  * 26  *  * 47    1 64  9 56 25 40 17 48
   62  4 53 11 37 27 46 20    *  4 53  * 37  *  * 20   62  3 54 11 38 27 46 19
    7 57 16 50 32 34 23 41    7  *  * 50  * 34 23  *    8 57 16 49 32 33 24 41
   60  6 51 13 35 29 44 22   60  *  * 13  * 29 44  *   59  6 51 14 35 30 43 22
   39 25 48 18 64  2 55  9    * 25 48  * 64  *  *  9   39 26 47 18 63  2 55 10
   28 38 19 45  3 61 12 54    * 38 19  *  3  *  * 54   28 37 20 45  4 61 12 53
   33 31 42 24 58  8 49 15   33  *  * 24  *  8 49  *   34 31 42 23 58  7 50 15
   30 36 21 43  5 59 14 52   30  *  * 43  * 59 14  *   29 36 21 44  5 60 13 52
    . . . . . . .
```

#6. The next thing we have to do is to collect any similar pairs whose 'Unmatched' patterns are the same and common and to put them into some groups.
The next list shows a part of the result:

```
** Collect Similar Pairs whose 'Unmatched' Patterns **
  **  are the same and Put them into each Group  **
```

Type 1:

```
[ 1]                141/T1      <-- Unmatched -->     111/63
    1 63  6 60 22 44 17 47    * 63  6  * 22  *  * 47    1 64  5 60 21 44 17 48
   62  4 57  7 41 23 46 20    *  4 57  * 41  *  * 20   62  3 58  7 42 23 46 19
   11 53 16 50 32 34 27 37   11  *  * 50  * 34 27  *   12 53 16 49 32 33 28 37
   56 10 51 13 35 29 40 26   56  *  * 13  * 29 40  *   55 10 51 14 35 30 39 26
   43 21 48 18 64  2 59  5    * 21 48  * 64  *  *  5   43 22 47 18 63  2 59  6
   24 42 19 45  3 61  8 58    * 42 19  *  3  *  * 58   24 41 20 45  4 61  8 57
   33 31 38 28 54 12 49 15   33  *  * 28  * 12 49  *   34 31 38 27 54 11 50 15
   30 36 25 39  9 55 14 52   30  *  * 39  * 55 14  *   29 36 25 40  9 56 13 52
```

```
[ 2]                142/T1      <-- Unmatched -->      112/63
    1 63  6 60 38 28 33 31    * 63  6  * 38  *  * 31    1 64  5 60 37 28 33 32
   62  4 57  7 25 39 30 36    *  4 57  * 25  *  * 36   62  3 58  7 26 39 30 35
   11 53 16 50 48 18 43 21   11  *  * 50  * 18 43  *   12 53 16 49 48 17 44 21
   56 10 51 13 19 45 24 42   56  *  * 13  * 45 24  *   55 10 51 14 19 46 23 42
   27 37 32 34 64  2 59  5    * 37 32  * 64  *  *  5   27 38 31 34 63  2 59  6
   40 26 35 29  3 61  8 58    * 26 35  *  3  *  * 58   40 25 36 29  4 61  8 57
   17 47 22 44 54 12 49 15   17  *  * 44  * 12 49  *   18 47 22 43 54 11 50 15
   46 20 41 23  9 55 14 52   46  *  * 23  * 55 14  *   45 20 41 24  9 56 13 52
[ 3]                143/T1      <-- Unmatched -->      113/63
    1 63  6 60 14 52  9 55    * 63  6  * 14  *  * 55    1 64  5 60 13 52  9 56
   62  4 57  7 49 15 54 12    *  4 57  * 49  *  * 12   62  3 58  7 50 15 54 11
   19 45 24 42 32 34 27 37   19  *  * 42  * 34 27  *   20 45 24 41 32 33 28 37
   48 18 43 21 35 29 40 26   48  *  * 21  * 29 40  *   47 18 43 22 35 30 39 26
   51 13 56 10 64  2 59  5    * 13 56  * 64  *  *  5   51 14 55 10 63  2 59  6
   16 50 11 53  3 61  8 58    * 50 11  *  3  *  * 58   16 49 12 53  4 61  8 57
   33 31 38 28 46 20 41 23   33  *  * 28  * 20 41  *   34 31 38 27 46 19 42 23
   30 36 25 39 17 47 22 44   30  *  * 39  * 47 22  *   29 36 25 40 17 48 21 44
[ 4]                144/T1      <-- Unmatched -->      114/63
    1 63 10 56 26 40 17 47    * 63 10  * 26  *  * 47    1 64  9 56 25 40 17 48
   62  4 53 11 37 27 46 20    *  4 53  * 37  *  * 20   62  3 54 11 38 27 46 19
    7 57 16 50 32 34 23 41    7  *  * 50  * 34 23  *    8 57 16 49 32 33 24 41
   60  6 51 13 35 29 44 22   60  *  * 13  * 29 44  *   59  6 51 14 35 30 43 22
   39 25 48 18 64  2 55  9    * 25 48  * 64  *  *  9   39 26 47 18 63  2 55 10
   28 38 19 45  3 61 12 54    * 38 19  *  3  *  * 54   28 37 20 45  4 61 12 53
   33 31 42 24 58  8 49 15   33  *  * 24  *  8 49  *   34 31 42 23 58  7 50 15
   30 36 21 43  5 59 14 52   30  *  * 43  * 59 14  *   29 36 21 44  5 60 13 52
[ 5]                145/T1      <-- Unmatched -->      115/63
    1 63 10 56 42 24 33 31    * 63 10  * 42  *  * 31    1 64  9 56 41 24 33 32
   62  4 53 11 21 43 30 36    *  4 53  * 21  *  * 36   62  3 54 11 22 43 30 35
    7 57 16 50 48 18 39 25    7  *  * 50  * 18 39  *    8 57 16 49 48 17 40 25
   60  6 51 13 19 45 28 38   60  *  * 13  * 45 28  *   59  6 51 14 19 46 27 38
   23 41 32 34 64  2 55  9    * 41 32  * 64  *  *  9   23 42 31 34 63  2 55 10
   44 22 35 29  3 61 12 54    * 22 35  *  3  *  * 54   44 21 36 29  4 61 12 53
   17 47 26 40 58  8 49 15   17  *  * 40  *  8 49  *   18 47 26 39 58  7 50 15
   46 20 37 27  5 59 14 52   46  *  * 27  * 59 14  *   45 20 37 28  5 60 13 52

  . . . . .
```

Type 2:

```
[11]                141/T2      <-- Unmatched -->      121/63
    1 63  6 60 22 44 17 47    * 63  * 60 22  * 17  *    1 64  6 59 21 44 18 47
   62  4 57  7 41 23 46 20    *  4  *  7 41  * 46  *   62  3 57  8 42 23 45 20
   11 53 16 50 32 34 27 37    * 53  * 50 32  * 27  *   11 54 16 49 31 34 28 37
   56 10 51 13 35 29 40 26    * 10  * 13 35  * 40  *   56  9 51 14 36 29 39 26
   43 21 48 18 64  2 59  5    * 21  * 18 64  * 59  *   43 22 48 17 63  2 60  5
   24 42 19 45  3 61  8 58    * 42  * 45  3  *  8  *   24 41 19 46  4 61  7 58
   33 31 38 28 54 12 49 15    * 31  * 28 54  * 49  *   33 32 38 27 53 12 50 15
   30 36 25 39  9 55 14 52    * 36  * 39  9  * 14  *   30 35 25 40 10 55 13 52
[12]                142/T2      <-- Unmatched -->      122/63
    1 63  6 60 38 28 33 31    * 63  * 60 38  * 33  *    1 64  6 59 37 28 34 31
   62  4 57  7 25 39 30 36    *  4  *  7 25  * 30  *   62  3 57  8 26 39 29 36
   11 53 16 50 48 18 43 21    * 53  * 50 48  * 43  *   11 54 16 49 47 18 44 21
   56 10 51 13 19 45 24 42    * 10  * 13 19  * 24  *   56  9 51 14 20 45 23 42
   27 37 32 34 64  2 59  5    * 37  * 34 64  * 59  *   27 38 32 33 63  2 60  5
   40 26 35 29  3 61  8 58    * 26  * 29  3  *  8  *   40 25 35 30  4 61  7 58
   17 47 22 44 54 12 49 15    * 47  * 44 54  * 49  *   17 48 22 43 53 12 50 15
   46 20 41 23  9 55 14 52    * 20  * 23  9  * 14  *   46 19 41 24 10 55 13 52
[13]                143/T2      <-- Unmatched -->      123/63
    1 63  6 60 14 52  9 55    * 63  * 60 14  *  9  *    1 64  6 59 13 52 10 55
   62  4 57  7 49 15 54 12    *  4  *  7 49  * 54  *   62  3 57  8 50 15 53 12
   19 45 24 42 32 34 27 37    * 45  * 42 32  * 27  *   19 46 24 41 31 34 28 37
   48 18 43 21 35 29 40 26    * 18  * 21 35  * 40  *   48 17 43 22 36 29 39 26
   51 13 56 10 64  2 59  5    * 13  * 10 64  * 59  *   51 14 56  9 63  2 60  5
   16 50 11 53  3 61  8 58    * 50  * 53  3  *  8  *   16 49 11 54  4 61  7 58
   33 31 38 28 46 20 41 23    * 31  * 28 46  * 41  *   33 32 38 27 45 20 42 23
   30 36 25 39 17 47 22 44    * 36  * 39 17  * 22  *   30 35 25 40 18 47 21 44
```

```
[14]                144/T2      <-- Unmatched -->     124/63
    1 63 10 56 26 40 17 47    * 63  * 56 26  * 17  *    1 64 10 55 25 40 18 47
   62  4 53 11 37 27 46 20    *  4  * 11 37  * 46  *   62  3 53 12 38 27 45 20
    7 57 16 50 32 34 23 41    * 57  * 50 32  * 23  *    7 58 16 49 31 34 24 41
   60  6 51 13 35 29 44 22    *  6  * 13 35  * 44  *   60  5 51 14 36 29 43 22
   39 25 48 18 64  2 55  9    * 25  * 18 64  * 55  *   39 26 48 17 63  2 56  9
   28 38 19 45  3 61 12 54    * 38  * 45  3  * 12  *   28 37 19 46  4 61 11 54
   33 31 42 24 58  8 49 15    * 31  * 24 58  * 49  *   33 32 42 23 57  8 50 15
   30 36 21 43  5 59 14 52    * 36  * 43  5  * 14  *   30 35 21 44  6 59 13 52

   . . . . .
```

Type 3:

```
[21]                141/T3      <-- Unmatched -->     131/63
    1 63  6 60 22 44 17 47    * 63  6  * 22  *  * 47    1 64  5 60 21 44 17 48
   62  4 57  7 41 23 46 20   62  *  *  7  * 23 46  *   61  4 57  8 41 24 45 20
   11 53 16 50 32 34 27 37   11  *  * 50  * 34 27  *   12 53 16 49 32 33 28 37
   56 10 51 13 35 29 40 26    * 10 51  * 35  *  * 26   56  9 52 13 36 29 40 25
   43 21 48 18 64  2 59  5    * 21 48  * 64  *  *  5   43 22 47 18 63  2 59  6
   24 42 19 45  3 61  8 58   24  *  * 45  * 61  8  *   23 42 19 46  3 62  7 58
   33 31 38 28 54 12 49 15   33  *  * 28  * 12 49  *   34 31 38 27 54 11 50 15
   30 36 25 39  9 55 14 52    * 36 25  *  9  *  * 52   30 35 26 39 10 55 14 51
[22]                142/T3      <-- Unmatched -->     132/63
    1 63  6 60 38 28 33 31    * 63  6  * 38  *  * 31    1 64  5 60 37 28 33 32
   62  4 57  7 25 39 30 36   62  *  *  7  * 39 30  *   61  4 57  8 25 40 29 36
   11 53 16 50 48 18 43 21   11  *  * 50  * 18 43  *   12 53 16 49 48 17 44 21
   56 10 51 13 19 45 24 42    * 10 51  * 19  *  * 42   56  9 52 13 20 45 24 41
   27 37 32 34 64  2 59  5    * 37 32  * 64  *  *  5   27 38 31 34 63  2 59  6
   40 26 35 29  3 61  8 58   40  *  * 29  * 61  8  *   39 26 35 30  3 62  7 58
   17 47 22 44 54 12 49 15   17  *  * 44  * 12 49  *   18 47 22 43 54 11 50 15
   46 20 41 23  9 55 14 52    * 20 41  *  9  *  * 52   46 19 42 23 10 55 14 51
[23]                143/T3      <-- Unmatched -->     133/63
    1 63  6 60 14 52  9 55    * 63  6  * 14  *  * 55    1 64  5 60 13 52  9 56
   62  4 57  7 49 15 54 12   62  *  *  7  * 15 54  *   61  4 57  8 49 16 53 12
   19 45 24 42 32 34 27 37   19  *  * 42  * 34 27  *   20 45 24 41 32 33 28 37
   48 18 43 21 35 29 40 26    * 18 43  * 35  *  * 26   48 17 44 21 36 29 40 25
   51 13 56 10 64  2 59  5    * 13 56  * 64  *  *  5   51 14 55 10 63  2 59  6
   16 50 11 53  3 61  8 58   16  *  * 53  * 61  8  *   15 50 11 54  3 62  7 58
   33 31 38 28 46 20 41 23   33  *  * 28  * 20 41  *   34 31 38 27 46 19 42 23
   30 36 25 39 17 47 22 44    * 36 25  * 17  *  * 44   30 35 26 39 18 47 22 43
   . . . . .
```

#7. Let's imagine and invent any transformation method for each group. If you find anything new, write some rules of transformation in your program codes, and examine them carefully and know if they work all right.

Let me show you some examples of my inventions and the result of my experiments:

```
/**/
/* Rules of Trnsformation */
/**/
void trnsf1(){
 short n;
    for(n=0;n<65;n++){d[n]=c[n];}
    d[2]=c[37]; d[3]=c[40]; d[5]=c[34]; d[8]=c[35];
         d[10]=c[45]; d[11]=c[48]; d[13]=c[42];   d[16]=c[43];
    d[17]=c[54]; d[20]=c[55]; d[22]=c[49];   d[23]=c[52];
         d[25]=c[62]; d[28]=c[63]; d[30]=c[57];   d[31]=c[60];
    d[34]=c[5]; d[35]=c[8]; d[37]=c[2]; d[40]=c[3];
         d[42]=c[13]; d[43]=c[16]; d[45]=c[10];   d[48]=c[11];
    d[49]=c[22]; d[52]=c[23]; d[54]=c[17];   d[55]=c[20];
         d[57]=c[30]; d[60]=c[31]; d[62]=c[25];   d[63]=c[28];
}
```

```
/**/
void trnsf2(){
 short n;
    for(n=0;n<65;n++){d[n]=c[n];}
    d[2]=c[37]; d[4]=c[39]; d[5]=c[34]; d[7]=c[36];
         d[10]=c[45]; d[12]=c[47]; d[13]=c[42];   d[15]=c[44];
    d[18]=c[53]; d[20]=c[55]; d[21]=c[50];   d[23]=c[52];
         d[26]=c[61]; d[28]=c[63]; d[29]=c[58];   d[31]=c[60];
    d[34]=c[5]; d[36]=c[7]; d[37]=c[2]; d[39]=c[4];
         d[42]=c[13]; d[44]=c[15]; d[45]=c[10];   d[47]=c[12];
    d[50]=c[21]; d[52]=c[23]; d[53]=c[18];   d[55]=c[20];
         d[58]=c[29]; d[60]=c[31]; d[61]=c[26];   d[63]=c[28];
}
/**/
void trnsf3(){
 short n;
    for(n=0;n<65;n++){d[n]=c[n];}
    d[2]=c[37]; d[3]=c[40]; d[5]=c[34]; d[8]=c[35];
         d[9]=c[46]; d[12]=c[47]; d[14]=c[41];    d[15]=c[44];
    d[17]=c[54]; d[20]=c[55]; d[22]=c[49];   d[23]=c[52];
         d[26]=c[61]; d[27]=c[64]; d[29]=c[58];   d[32]=c[59];
    d[34]=c[5]; d[35]=c[8]; d[37]=c[2]; d[40]=c[3];
         d[41]=c[14]; d[44]=c[15]; d[46]=c[9];    d[47]=c[12];
    d[49]=c[22]; d[52]=c[23]; d[54]=c[17];   d[55]=c[20];
         d[58]=c[29]; d[59]=c[32]; d[61]=c[26];   d[64]=c[27];
}
/* Diagrams: Concept of Transformations */
/*
                     0/B      --- Transform -->      1/T
   1  2  3  4  5  6  7  8    *  2  3  *  5  *  *  8    1 37 40  4 34  6  7 35
   9 10 11 12 13 14 15 16    * 10 11  * 13  *  * 16    9 45 48 12 42 14 15 43
  17 18 19 20 21 22 23 24   17  *  * 20  * 22 23  *   54 18 19 55 21 49 52 24
  25 26 27 28 29 30 31 32   25  *  * 28  * 30 31  *   62 26 27 63 29 57 60 32
  33 34 35 36 37 38 39 40    * 34 35  * 37  *  * 40   33  5  8 36  2 38 39  3
  41 42 43 44 45 46 47 48    * 42 43  * 45  *  * 48   41 13 16 44 10 46 47 11
  49 50 51 52 53 54 55 56   49  *  * 52  * 54 55  *   22 50 51 23 53 17 20 56
  57 58 59 60 61 62 63 64   57  *  * 60  * 62 63  *   30 58 59 31 61 25 28 64
                     0/B      --- Transform -->      2/T
   1  2  3  4  5  6  7  8    *  2  *  4  5  *  7  *    1 37  3 39 34  6 36  8
   9 10 11 12 13 14 15 16    * 10  * 12 13  * 15  *    9 45 11 47 42 14 44 16
  17 18 19 20 21 22 23 24    * 18  * 20 21  * 23  *   17 53 19 55 50 22 52 24
  25 26 27 28 29 30 31 32    * 26  * 28 29  * 31  *   25 61 27 63 58 30 60 32
  33 34 35 36 37 38 39 40    * 34  * 36 37  * 39  *   33  5 35  7  2 38  4 40
  41 42 43 44 45 46 47 48    * 42  * 44 45  * 47  *   41 13 43 15 10 46 12 48
  49 50 51 52 53 54 55 56    * 50  * 52 53  * 55  *   49 21 51 23 18 54 20 56
  57 58 59 60 61 62 63 64    * 58  * 60 61  * 63  *   57 29 59 31 26 62 28 64
                     0/B      --- Transform -->      3/T
   1  2  3  4  5  6  7  8    *  2  3  *  5  *  *  8    1 37 40  4 34  6  7 35
   9 10 11 12 13 14 15 16    9  *  * 12  * 14 15  *   46 10 11 47 13 41 44 16
  17 18 19 20 21 22 23 24   17  *  * 20  * 22 23  *   54 18 19 55 21 49 52 24
  25 26 27 28 29 30 31 32    * 26 27  * 29  *  * 32   25 61 64 28 58 30 31 59
  33 34 35 36 37 38 39 40    * 34 35  * 37  *  * 40   33  5  8 36  2 38 39  3
  41 42 43 44 45 46 47 48   41  *  * 44  * 46 47  *   14 42 43 15 45  9 12 48
  49 50 51 52 53 54 55 56   49  *  * 52  * 54 55  *   22 50 51 23 53 17 20 56
  57 58 59 60 61 62 63 64    * 58 59  * 61  *  * 64   57 29 32 60 26 62 63 27
*/
/**/
```

**** Check if each Transformation Works All Right ****
****** By Transformation Type 1, 2 and 3 ******

```
[ 1]                141/T1   <-- Matched -->  111/63
    1 64  5 60 21 44 17 48   * * * * * * * *    1 64  5 60 21 44 17 48
   62  3 58  7 42 23 46 19   * * * * * * * *   62  3 58  7 42 23 46 19
   12 53 16 49 32 33 28 37   * * * * * * * *   12 53 16 49 32 33 28 37
   55 10 51 14 35 30 39 26   * * * * * * * *   55 10 51 14 35 30 39 26
   43 22 47 18 63  2 59  6   * * * * * * * *   43 22 47 18 63  2 59  6
   24 41 20 45  4 61  8 57   * * * * * * * *   24 41 20 45  4 61  8 57
   34 31 38 27 54 11 50 15   * * * * * * * *   34 31 38 27 54 11 50 15
   29 36 25 40  9 56 13 52   * * * * * * * *   29 36 25 40  9 56 13 52
[ 2]                142/T1   <-- Matched -->  112/63
    1 64  5 60 37 28 33 32   * * * * * * * *    1 64  5 60 37 28 33 32
   62  3 58  7 26 39 30 35   * * * * * * * *   62  3 58  7 26 39 30 35
   12 53 16 49 48 17 44 21   * * * * * * * *   12 53 16 49 48 17 44 21
   55 10 51 14 19 46 23 42   * * * * * * * *   55 10 51 14 19 46 23 42
   27 38 31 34 63  2 59  6   * * * * * * * *   27 38 31 34 63  2 59  6
   40 25 36 29  4 61  8 57   * * * * * * * *   40 25 36 29  4 61  8 57
   18 47 22 43 54 11 50 15   * * * * * * * *   18 47 22 43 54 11 50 15
   45 20 41 24  9 56 13 52   * * * * * * * *   45 20 41 24  9 56 13 52
[ 3]                143/T1   <-- Matched -->  113/63
    1 64  5 60 13 52  9 56   * * * * * * * *    1 64  5 60 13 52  9 56
   62  3 58  7 50 15 54 11   * * * * * * * *   62  3 58  7 50 15 54 11
   20 45 24 41 32 33 28 37   * * * * * * * *   20 45 24 41 32 33 28 37
   47 18 43 22 35 30 39 26   * * * * * * * *   47 18 43 22 35 30 39 26
   51 14 55 10 63  2 59  6   * * * * * * * *   51 14 55 10 63  2 59  6
   16 49 12 53  4 61  8 57   * * * * * * * *   16 49 12 53  4 61  8 57
   34 31 38 27 46 19 42 23   * * * * * * * *   34 31 38 27 46 19 42 23
   29 36 25 40 17 48 21 44   * * * * * * * *   29 36 25 40 17 48 21 44
[ 4]                144/T1   <-- Matched -->  114/63
    1 64  9 56 25 40 17 48   * * * * * * * *    1 64  9 56 25 40 17 48
   62  3 54 11 38 27 46 19   * * * * * * * *   62  3 54 11 38 27 46 19
    8 57 16 49 32 33 24 41   * * * * * * * *    8 57 16 49 32 33 24 41
   59  6 51 14 35 30 43 22   * * * * * * * *   59  6 51 14 35 30 43 22
   39 26 47 18 63  2 55 10   * * * * * * * *   39 26 47 18 63  2 55 10
   28 37 20 45  4 61 12 53   * * * * * * * *   28 37 20 45  4 61 12 53
   34 31 42 23 58  7 50 15   * * * * * * * *   34 31 42 23 58  7 50 15
   29 36 21 44  5 60 13 52   * * * * * * * *   29 36 21 44  5 60 13 52
  . . . . .
[11]                141/T2   <-- Matched -->  121/63
    1 64  6 59 21 44 18 47   * * * * * * * *    1 64  6 59 21 44 18 47
   62  3 57  8 42 23 45 20   * * * * * * * *   62  3 57  8 42 23 45 20
   11 54 16 49 31 34 28 37   * * * * * * * *   11 54 16 49 31 34 28 37
   56  9 51 14 36 29 39 26   * * * * * * * *   56  9 51 14 36 29 39 26
   43 22 48 17 63  2 60  5   * * * * * * * *   43 22 48 17 63  2 60  5
   24 41 19 46  4 61  7 58   * * * * * * * *   24 41 19 46  4 61  7 58
   33 32 38 27 53 12 50 15   * * * * * * * *   33 32 38 27 53 12 50 15
   30 35 25 40 10 55 13 52   * * * * * * * *   30 35 25 40 10 55 13 52
[12]                142/T2   <-- Matched -->  122/63
    1 64  6 59 37 28 34 31   * * * * * * * *    1 64  6 59 37 28 34 31
   62  3 57  8 26 39 29 36   * * * * * * * *   62  3 57  8 26 39 29 36
   11 54 16 49 47 18 44 21   * * * * * * * *   11 54 16 49 47 18 44 21
   56  9 51 14 20 45 23 42   * * * * * * * *   56  9 51 14 20 45 23 42
   27 38 32 33 63  2 60  5   * * * * * * * *   27 38 32 33 63  2 60  5
   40 25 35 30  4 61  7 58   * * * * * * * *   40 25 35 30  4 61  7 58
   17 48 22 43 53 12 50 15   * * * * * * * *   17 48 22 43 53 12 50 15
   46 19 41 24 10 55 13 52   * * * * * * * *   46 19 41 24 10 55 13 52
[13]                143/T2   <-- Matched -->  123/63
    1 64  6 59 13 52 10 55   * * * * * * * *    1 64  6 59 13 52 10 55
   62  3 57  8 50 15 53 12   * * * * * * * *   62  3 57  8 50 15 53 12
   19 46 24 41 31 34 28 37   * * * * * * * *   19 46 24 41 31 34 28 37
   48 17 43 22 36 29 39 26   * * * * * * * *   48 17 43 22 36 29 39 26
   51 14 56  9 63  2 60  5   * * * * * * * *   51 14 56  9 63  2 60  5
   16 49 11 54  4 61  7 58   * * * * * * * *   16 49 11 54  4 61  7 58
   33 32 38 27 45 20 42 23   * * * * * * * *   33 32 38 27 45 20 42 23
   30 35 25 40 18 47 21 44   * * * * * * * *   30 35 25 40 18 47 21 44
```

```
[14]                144/T2   <-- Matched -->  124/63
    1 64 10 55 25 40 18 47   * * * * * * * *    1 64 10 55 25 40 18 47
   62  3 53 12 38 27 45 20   * * * * * * * *   62  3 53 12 38 27 45 20
    7 58 16 49 31 34 24 41   * * * * * * * *    7 58 16 49 31 34 24 41
   60  5 51 14 36 29 43 22   * * * * * * * *   60  5 51 14 36 29 43 22
   39 26 48 17 63  2 56  9   * * * * * * * *   39 26 48 17 63  2 56  9
   28 37 19 46  4 61 11 54   * * * * * * * *   28 37 19 46  4 61 11 54
   33 32 42 23 57  8 50 15   * * * * * * * *   33 32 42 23 57  8 50 15
   30 35 21 44  6 59 13 52   * * * * * * * *   30 35 21 44  6 59 13 52
  . . . . . .
[21]                141/T3   <-- Matched -->  131/63
    1 64  5 60 21 44 17 48   * * * * * * * *    1 64  5 60 21 44 17 48
   61  4 57  8 41 24 45 20   * * * * * * * *   61  4 57  8 41 24 45 20
   12 53 16 49 32 33 28 37   * * * * * * * *   12 53 16 49 32 33 28 37
   56  9 52 13 36 29 40 25   * * * * * * * *   56  9 52 13 36 29 40 25
   43 22 47 18 63  2 59  6   * * * * * * * *   43 22 47 18 63  2 59  6
   23 42 19 46  3 62  7 58   * * * * * * * *   23 42 19 46  3 62  7 58
   34 31 38 27 54 11 50 15   * * * * * * * *   34 31 38 27 54 11 50 15
   30 35 26 39 10 55 14 51   * * * * * * * *   30 35 26 39 10 55 14 51
[22]                142/T3   <-- Matched -->  132/63
    1 64  5 60 37 28 33 32   * * * * * * * *    1 64  5 60 37 28 33 32
   61  4 57  8 25 40 29 36   * * * * * * * *   61  4 57  8 25 40 29 36
   12 53 16 49 48 17 44 21   * * * * * * * *   12 53 16 49 48 17 44 21
   56  9 52 13 20 45 24 41   * * * * * * * *   56  9 52 13 20 45 24 41
   27 38 31 34 63  2 59  6   * * * * * * * *   27 38 31 34 63  2 59  6
   39 26 35 30  3 62  7 58   * * * * * * * *   39 26 35 30  3 62  7 58
   18 47 22 43 54 11 50 15   * * * * * * * *   18 47 22 43 54 11 50 15
   46 19 42 23 10 55 14 51   * * * * * * * *   46 19 42 23 10 55 14 51
[23]                143/T3   <-- Matched -->  133/63
    1 64  5 60 13 52  9 56   * * * * * * * *    1 64  5 60 13 52  9 56
   61  4 57  8 49 16 53 12   * * * * * * * *   61  4 57  8 49 16 53 12
   20 45 24 41 32 33 28 37   * * * * * * * *   20 45 24 41 32 33 28 37
   48 17 44 21 36 29 40 25   * * * * * * * *   48 17 44 21 36 29 40 25
   51 14 55 10 63  2 59  6   * * * * * * * *   51 14 55 10 63  2 59  6
   15 50 11 54  3 62  7 58   * * * * * * * *   15 50 11 54  3 62  7 58
   34 31 38 27 46 19 42 23   * * * * * * * *   34 31 38 27 46 19 42 23
   30 35 26 39 18 47 22 43   * * * * * * * *   30 35 26 39 18 47 22 43
[24]                144/T3   <-- Matched -->  134/63
    1 64  9 56 25 40 17 48   * * * * * * * *    1 64  9 56 25 40 17 48
   61  4 53 12 37 28 45 20   * * * * * * * *   61  4 53 12 37 28 45 20
    8 57 16 49 32 33 24 41   * * * * * * * *    8 57 16 49 32 33 24 41
   60  5 52 13 36 29 44 21   * * * * * * * *   60  5 52 13 36 29 44 21
   39 26 47 18 63  2 55 10   * * * * * * * *   39 26 47 18 63  2 55 10
   27 38 19 46  3 62 11 54   * * * * * * * *   27 38 19 46  3 62 11 54
   34 31 42 23 58  7 50 15   * * * * * * * *   34 31 42 23 58  7 50 15
   30 35 22 43  6 59 14 51   * * * * * * * *   30 35 22 43  6 59 14 51
  . . . . . .
```

They seem to work well. By these three types of transformation we could really make 30 solutions of the second generation from F10 of the first.

#8. What do we have to do for the next step?
Why don't you make any granddaughters from the 30 daughters of the second generation? Yes, let's.

.....
I tried it several times, but I gave it up at last.
I had to make too many generations from these 30 of the second generation.
I felt we need more daughters of the second generation than 30.
But how could we find them beside 30?

Why don't you put each F10 upside down or leftside right in reverse, and compare it to No1.~140 solutions of M&C&P?
I found we need the so-called 'Mirror Reflection' operation for our purpose.
I show you its concept as follows:

```
/**/
/* Mirror Reflection */
void reflect(){
     d[1]=c[1]; d[2]=c[9]; d[3]=c[17]; d[4]=c[25];
          d[5]=c[33]; d[6]=c[41]; d[7]=c[49]; d[8]=c[57];
     d[9]=c[2]; d[10]=c[10]; d[11]=c[18]; d[12]=c[26];
          d[13]=c[34]; d[14]=c[42]; d[15]=c[50]; d[16]=c[58];
     d[17]=c[3]; d[18]=c[11]; d[19]=c[19]; d[20]=c[27];
          d[21]=c[35]; d[22]=c[43]; d[23]=c[51]; d[24]=c[59];
     d[25]=c[4]; d[26]=c[12]; d[27]=c[20]; d[28]=c[28];
          d[29]=c[36]; d[30]=c[44]; d[31]=c[52]; d[32]=c[60];
     d[33]=c[5]; d[34]=c[13]; d[35]=c[21]; d[36]=c[29];
          d[37]=c[37]; d[38]=c[45]; d[39]=c[53]; d[40]=c[61];
     d[41]=c[6]; d[42]=c[14]; d[43]=c[22]; d[44]=c[30];
          d[45]=c[38]; d[46]=c[46]; d[47]=c[54]; d[48]=c[62];
     d[49]=c[7]; d[50]=c[15]; d[51]=c[23]; d[52]=c[31];
          d[53]=c[39]; d[54]=c[47]; d[55]=c[55]; d[56]=c[63];
     d[57]=c[8]; d[58]=c[16]; d[59]=c[24]; d[60]=c[32];
          d[61]=c[40]; d[62]=c[48]; d[63]=c[56]; d[64]=c[64];
}
/*
                      0/B      --- Transform -->     Reflect/
    1  2  3  4  5  6  7  8    *  2  3  4  5  6  7  8    1  9 17 25 33 41 49 57
    9 10 11 12 13 14 15 16    9  * 11 12 13 14 15 16    2 10 18 26 34 42 50 58
   17 18 19 20 21 22 23 24   17 18  * 20 21 22 23 24    3 11 19 27 35 43 51 59
   25 26 27 28 29 30 31 32   25 26 27  * 29 30 31 32    4 12 20 28 36 44 52 60
   33 34 35 36 37 38 39 40   33 34 35 36  * 38 39 40    5 13 21 29 37 45 53 61
   41 42 43 44 45 46 47 48   41 42 43 44 45  * 47 48    6 14 22 30 38 46 54 62
   49 50 51 52 53 54 55 56   49 50 51 52 53 54  * 56    7 15 23 31 39 47 55 63
   57 58 59 60 61 62 63 64   57 58 59 60 61 62 63  *    8 16 24 32 40 48 56 64
*/
/**/
```

I found 22 new similar pairs to the reflected F10.
I take the next ten solutions for our purpose, because they make a group which can react only to the transformation type 2.

****** Find Anything Similar to the Reflected F10: ******

```
[ 1]              141R/64      <-- Unmatched -->      92/62
    1 62 11 56 43 24 33 30    * 62 11  * 43  *  * 30    1 64  9 56 41 24 33 32
   63  4 53 10 21 42 31 36    *  4 53  * 21  *  * 36   63  2 55 10 23 42 31 34
    6 57 16 51 48 19 38 25    6  *  * 51  * 19 38  *    8 57 16 49 48 17 40 25
   60  7 50 13 18 45 28 39   60  *  * 13  * 45 28  *   58  7 50 15 18 47 26 39
   22 41 32 35 64  3 54  9    * 41 32  * 64  *  *  9   22 43 30 35 62  3 54 11
   44 23 34 29  2 61 12 55    * 23 34  *  2  *  * 55   44 21 36 29  4 61 12 53
   17 46 27 40 59  8 49 14   17  *  * 40  *  8 49  *   19 46 27 38 59  6 51 14
   47 20 37 26  5 58 15 52   47  *  * 26  * 58 15  *   45 20 37 28  5 60 13 52
```

```
[ 2]                141R/64      <-- Unmatched -->      94/62
    1 62 11 56 43 24 33 30    * 62  * 56 43  * 33  *    1 64 11 54 41 24 35 30
   63  4 53 10 21 42 31 36    *  4  * 10 21  * 31  *   63  2 53 12 23 42 29 36
    6 57 16 51 48 19 38 25    * 57  * 51 48  * 38  *    6 59 16 49 46 19 40 25
   60  7 50 13 18 45 28 39    *  7  * 13 18  * 28  *   60  5 50 15 20 45 26 39
   22 41 32 35 64  3 54  9    * 41  * 35 64  * 54  *   22 43 32 33 62  3 56  9
   44 23 34 29  2 61 12 55    * 23  * 29  2  * 12  *   44 21 34 31  4 61 10 55
   17 46 27 40 59  8 49 14    * 46  * 40 59  * 49  *   17 48 27 38 57  8 51 14
   47 20 37 26  5 58 15 52    * 20  * 26  5  * 15  *   47 18 37 28  7 58 13 52
[ 3]                141R/64      <-- Unmatched -->     101/62
    1 62 11 56 43 24 33 30    * 62 11  * 43  *  * 30    1 64  9 56 41 24 33 32
   63  4 53 10 21 42 31 36   63  *  * 10  * 42 31  *   61  4 53 12 21 44 29 36
    6 57 16 51 48 19 38 25    6  *  * 51  * 19 38  *    8 57 16 49 48 17 40 25
   60  7 50 13 18 45 28 39    *  7 50  * 18  *  * 39   60  5 52 13 20 45 28 37
   22 41 32 35 64  3 54  9    * 41 32  * 64  *  *  9   22 43 30 35 62  3 54 11
   44 23 34 29  2 61 12 55   44  *  * 29  * 61 12  *   42 23 34 31  2 63 10 55
   17 46 27 40 59  8 49 14   17  *  * 40  *  8 49  *   19 46 27 38 59  6 51 14
   47 20 37 26  5 58 15 52    * 20 37  *  5  *  * 52   47 18 39 26  7 58 15 50
[ 4]                142R/64      <-- Unmatched -->      91/62
    1 62 11 56 27 40 17 46    * 62 11  * 27  *  * 46    1 64  9 56 25 40 17 48
   63  4 53 10 37 26 47 20    *  4 53  * 37  *  * 20   63  2 55 10 39 26 47 18
    6 57 16 51 32 35 22 41    6  *  * 51  * 35 22  *    8 57 16 49 32 33 24 41
   60  7 50 13 34 29 44 23   60  *  * 13  * 29 44  *   58  7 50 15 34 31 42 23
   38 25 48 19 64  3 54  9    * 25 48  * 64  *  *  9   38 27 46 19 62  3 54 11
   28 39 18 45  2 61 12 55    * 39 18  *  2  *  * 55   28 37 20 45  4 61 12 53
   33 30 43 24 59  8 49 14   33  *  * 24  *  8 49  *   35 30 43 22 59  6 51 14
   31 36 21 42  5 58 15 52   31  *  * 42  * 58 15  *   29 36 21 44  5 60 13 52
[ 5]                142R/64      <-- Unmatched -->      93/62
    1 62 11 56 27 40 17 46    * 62  * 56 27  * 17  *    1 64 11 54 25 40 19 46
   63  4 53 10 37 26 47 20    *  4  * 10 37  * 47  *   63  2 53 12 39 26 45 20
    6 57 16 51 32 35 22 41    * 57  * 51 32  * 22  *    6 59 16 49 30 35 24 41
   60  7 50 13 34 29 44 23    *  7  * 13 34  * 44  *   60  5 50 15 36 29 42 23
   38 25 48 19 64  3 54  9    * 25  * 19 64  * 54  *   38 27 48 17 62  3 56  9
   28 39 18 45  2 61 12 55    * 39  * 45  2  * 12  *   28 37 18 47  4 61 10 55
   33 30 43 24 59  8 49 14    * 30  * 24 59  * 49  *   33 32 43 22 57  8 51 14
   31 36 21 42  5 58 15 52    * 36  * 42  5  * 15  *   31 34 21 44  7 58 13 52
[ 6]                142R/64      <-- Unmatched -->     100/62
    1 62 11 56 27 40 17 46    * 62 11  * 27  *  * 46    1 64  9 56 25 40 17 48
   63  4 53 10 37 26 47 20   63  *  * 10  * 26 47  *   61  4 53 12 37 28 45 20
    6 57 16 51 32 35 22 41    6  *  * 51  * 35 22  *    8 57 16 49 32 33 24 41
   60  7 50 13 34 29 44 23    *  7 50  * 34  *  * 23   60  5 52 13 36 29 44 21
   38 25 48 19 64  3 54  9    * 25 48  * 64  *  *  9   38 27 46 19 62  3 54 11
   28 39 18 45  2 61 12 55   28  *  * 45  * 61 12  *   26 39 18 47  2 63 10 55
   33 30 43 24 59  8 49 14   33  *  * 24  *  8 49  *   35 30 43 22 59  6 51 14
   31 36 21 42  5 58 15 52    * 36 21  *  5  *  * 52   31 34 23 42  7 58 15 50
  . . . . . .
[21]                149R/64      <-- Unmatched -->      43/60
    1 60 21 48 53 16 33 28    * 60  * 48 53  * 33  *    1 64 21 44 49 16 37 28
   63  6 43 18 11 50 31 38    *  6  * 18 11  * 31  *   63  2 43 22 15 50 27 38
    4 57 24 45 56 13 36 25    * 57  * 45 56  * 36  *    4 61 24 41 52 13 40 25
   62  7 42 19 10 51 30 39    *  7  * 19 10  * 30  *   62  3 42 23 14 51 26 39
   12 49 32 37 64  5 44 17    * 49  * 37 64  * 44  *   12 53 32 33 60  5 48 17
   54 15 34 27  2 59 22 47    * 15  * 27  2  * 22  *   54 11 34 31  6 59 18 47
    9 52 29 40 61  8 41 20    * 52  * 40 61  * 41  *    9 56 29 36 57  8 45 20
   55 14 35 26  3 58 23 46    * 14  * 26  3  * 23  *   55 10 35 30  7 58 19 46
[22]                150R/64      <-- Unmatched -->      11/56
    1 56 25 48 57 16 33 24    * 56  * 48 57  * 33  *    1 64 25 40 49 16 41 24
   63 10 39 18  7 50 31 42    * 10  * 18  7  * 31  *   63  2 39 26 15 50 23 42
    4 53 28 45 60 13 36 21    * 53  * 45 60  * 36  *    4 61 28 37 52 13 44 21
   62 11 38 19  6 51 30 43    * 11  * 19  6  * 30  *   62  3 38 27 14 51 22 43
    8 49 32 41 64  9 40 17    * 49  * 41 64  * 40  *    8 57 32 33 56  9 48 17
   58 15 34 23  2 55 26 47    * 15  * 23  2  * 26  *   58  7 34 31 10 55 18 47
    5 52 29 44 61 12 37 20    * 52  * 44 61  * 37  *    5 60 29 36 53 12 45 20
   59 14 35 22  3 54 27 46    * 14  * 22  3  * 27  *   59  6 35 30 11 54 19 46

   1/56   3/60  18/62  [Count = 22]
```

**** Collect Similar Pairs whose 'Unmatched' Patterns ****
**** are the same and Put them into each Group ****

Type R+2:

```
[31]                141/R2      <-- Unmatched -->      94/62
   1 62 11 56 43 24 33 30    * 62  * 56 43  * 33  *    1 64 11 54 41 24 35 30
  63  4 53 10 21 42 31 36    *  4  * 10 21  * 31  *   63  2 53 12 23 42 29 36
   6 57 16 51 48 19 38 25    * 57  * 51 48  * 38  *    6 59 16 49 46 19 40 25
  60  7 50 13 18 45 28 39    *  7  * 13 18  * 28  *   60  5 50 15 20 45 26 39
  22 41 32 35 64  3 54  9    * 41  * 35 64  * 54  *   22 43 32 33 62  3 56  9
  44 23 34 29  2 61 12 55    * 23  * 29  2  * 12  *   44 21 34 31  4 61 10 55
  17 46 27 40 59  8 49 14    * 46  * 40 59  * 49  *   17 48 27 38 57  8 51 14
  47 20 37 26  5 58 15 52    * 20  * 26  5  * 15  *   47 18 37 28  7 58 13 52
[32]                142/R2      <-- Unmatched -->      93/62
   1 62 11 56 27 40 17 46    * 62  * 56 27  * 17  *    1 64 11 54 25 40 19 46
  63  4 53 10 37 26 47 20    *  4  * 10 37  * 47  *   63  2 53 12 39 26 45 20
   6 57 16 51 32 35 22 41    * 57  * 51 32  * 22  *    6 59 16 49 30 35 24 41
  60  7 50 13 34 29 44 23    *  7  * 13 34  * 44  *   60  5 50 15 36 29 42 23
  38 25 48 19 64  3 54  9    * 25  * 19 64  * 54  *   38 27 48 17 62  3 56  9
  28 39 18 45  2 61 12 55    * 39  * 45  2  * 12  *   28 37 18 47  4 61 10 55
  33 30 43 24 59  8 49 14    * 30  * 24 59  * 49  *   33 32 43 22 57  8 51 14
  31 36 21 42  5 58 15 52    * 36  * 42  5  * 15  *   31 34 21 44  7 58 13 52
[33]                143/R2      <-- Unmatched -->      96/62
   1 62 19 48 51 16 33 30    * 62  * 48 51  * 33  *    1 64 19 46 49 16 35 30
  63  4 45 18 13 50 31 36    *  4  * 18 13  * 31  *   63  2 45 20 15 50 29 36
   6 57 24 43 56 11 38 25    * 57  * 43 56  * 38  *    6 59 24 41 54 11 40 25
  60  7 42 21 10 53 28 39    *  7  * 21 10  * 28  *   60  5 42 23 12 53 26 39
  14 49 32 35 64  3 46 17    * 49  * 35 64  * 46  *   14 51 32 33 62  3 48 17
  52 15 34 29  2 61 20 47    * 15  * 29  2  * 20  *   52 13 34 31  4 61 18 47
   9 54 27 40 59  8 41 22    * 54  * 40 59  * 41  *    9 56 27 38 57  8 43 22
  55 12 37 26  5 58 23 44    * 12  * 26  5  * 23  *   55 10 37 28  7 58 21 44
  . . . . . . .
```

**** Check if each Transformation Works All Right By Type R+2: ****

```
[31]                141/RT2   <-- Matched -->   94/62
   1 64 11 54 41 24 35 30    * * * * * * * *    1 64 11 54 41 24 35 30
  63  2 53 12 23 42 29 36    * * * * * * * *   63  2 53 12 23 42 29 36
   6 59 16 49 46 19 40 25    * * * * * * * *    6 59 16 49 46 19 40 25
  60  5 50 15 20 45 26 39    * * * * * * * *   60  5 50 15 20 45 26 39
  22 43 32 33 62  3 56  9    * * * * * * * *   22 43 32 33 62  3 56  9
  44 21 34 31  4 61 10 55    * * * * * * * *   44 21 34 31  4 61 10 55
  17 48 27 38 57  8 51 14    * * * * * * * *   17 48 27 38 57  8 51 14
  47 18 37 28  7 58 13 52    * * * * * * * *   47 18 37 28  7 58 13 52
[32]                142/RT2   <-- Matched -->   93/62
   1 64 11 54 25 40 19 46    * * * * * * * *    1 64 11 54 25 40 19 46
  63  2 53 12 39 26 45 20    * * * * * * * *   63  2 53 12 39 26 45 20
   6 59 16 49 30 35 24 41    * * * * * * * *    6 59 16 49 30 35 24 41
  60  5 50 15 36 29 42 23    * * * * * * * *   60  5 50 15 36 29 42 23
  38 27 48 17 62  3 56  9    * * * * * * * *   38 27 48 17 62  3 56  9
  28 37 18 47  4 61 10 55    * * * * * * * *   28 37 18 47  4 61 10 55
  33 32 43 22 57  8 51 14    * * * * * * * *   33 32 43 22 57  8 51 14
  31 34 21 44  7 58 13 52    * * * * * * * *   31 34 21 44  7 58 13 52
[33]                143/RT2   <-- Matched -->   96/62
   1 64 19 46 49 16 35 30    * * * * * * * *    1 64 19 46 49 16 35 30
  63  2 45 20 15 50 29 36    * * * * * * * *   63  2 45 20 15 50 29 36
   6 59 24 41 54 11 40 25    * * * * * * * *    6 59 24 41 54 11 40 25
  60  5 42 23 12 53 26 39    * * * * * * * *   60  5 42 23 12 53 26 39
  14 51 32 33 62  3 48 17    * * * * * * * *   14 51 32 33 62  3 48 17
  52 13 34 31  4 61 18 47    * * * * * * * *   52 13 34 31  4 61 18 47
   9 56 27 38 57  8 43 22    * * * * * * * *    9 56 27 38 57  8 43 22
  55 10 37 28  7 58 21 44    * * * * * * * *   55 10 37 28  7 58 21 44
```

```
[34]                144/RT2   <-- Matched -->   89/62
    1 64  7 58 37 28 35 30    * * * * * * * *    1 64  7 58 37 28 35 30
   63  2 57  8 27 38 29 36    * * * * * * * *   63  2 57  8 27 38 29 36
   10 55 16 49 46 19 44 21    * * * * * * * *   10 55 16 49 46 19 44 21
   56  9 50 15 20 45 22 43    * * * * * * * *   56  9 50 15 20 45 22 43
   26 39 32 33 62  3 60  5    * * * * * * * *   26 39 32 33 62  3 60  5
   40 25 34 31  4 61  6 59    * * * * * * * *   40 25 34 31  4 61  6 59
   17 48 23 42 53 12 51 14    * * * * * * * *   17 48 23 42 53 12 51 14
   47 18 41 24 11 54 13 52    * * * * * * * *   47 18 41 24 11 54 13 52
[35]                145/RT2   <-- Matched -->   88/62
    1 64  7 58 21 44 19 46    * * * * * * * *    1 64  7 58 21 44 19 46
   63  2 57  8 43 22 45 20    * * * * * * * *   63  2 57  8 43 22 45 20
   10 55 16 49 30 35 28 37    * * * * * * * *   10 55 16 49 30 35 28 37
   56  9 50 15 36 29 38 27    * * * * * * * *   56  9 50 15 36 29 38 27
   42 23 48 17 62  3 60  5    * * * * * * * *   42 23 48 17 62  3 60  5
   24 41 18 47  4 61  6 59    * * * * * * * *   24 41 18 47  4 61  6 59
   33 32 39 26 53 12 51 14    * * * * * * * *   33 32 39 26 53 12 51 14
   31 34 25 40 11 54 13 52    * * * * * * * *   31 34 25 40 11 54 13 52
  . . . . . .
```

Thus we have got 40 daughters of the second generation.

#9. We have to do the same thing to the third generation.
Take 40 of G2 for the next mother set, and compare each to the other 100 object solutions.
I found 60 similar daughters to the new 40 mothers.
I could finally made 60 solutions of G3 only by two types of transformation.
.....
I found and made 31 solutions of G4, and 9 of G5. That's all!
Let me skip all explanations of those to my recent conclusion.

#10. I found we must have 12 types of transformation and 'mirror reflection' in all.
The next list shows our total system of transformations.

```
/**/
/* Trnsformation Procedures for All */
/**/
void trnsf1(){
 short n;
    for(n=0;n<65;n++){d[n]=c[n];}
    d[2]=c[37]; d[3]=c[40]; d[5]=c[34]; d[8]=c[35];
         d[10]=c[45]; d[11]=c[48]; d[13]=c[42];   d[16]=c[43];
    d[17]=c[54]; d[20]=c[55]; d[22]=c[49];   d[23]=c[52];
         d[25]=c[62]; d[28]=c[63]; d[30]=c[57];   d[31]=c[60];
    d[34]=c[5]; d[35]=c[8]; d[37]=c[2]; d[40]=c[3];
         d[42]=c[13]; d[43]=c[16]; d[45]=c[10];   d[48]=c[11];
    d[49]=c[22]; d[52]=c[23]; d[54]=c[17];   d[55]=c[20];
         d[57]=c[30]; d[60]=c[31]; d[62]=c[25];   d[63]=c[28];
}
/**/
void trnsf2(){
 short n;
    for(n=0;n<65;n++){d[n]=c[n];}
    d[2]=c[37]; d[4]=c[39]; d[5]=c[34]; d[7]=c[36];
         d[10]=c[45]; d[12]=c[47]; d[13]=c[42];   d[15]=c[44];
```

```
    d[18]=c[53]; d[20]=c[55]; d[21]=c[50];   d[23]=c[52];
         d[26]=c[61]; d[28]=c[63]; d[29]=c[58];   d[31]=c[60];
    d[34]=c[5]; d[36]=c[7]; d[37]=c[2]; d[39]=c[4];
         d[42]=c[13]; d[44]=c[15]; d[45]=c[10];   d[47]=c[12];
    d[50]=c[21]; d[52]=c[23]; d[53]=c[18];   d[55]=c[20];
         d[58]=c[29]; d[60]=c[31]; d[61]=c[26];   d[63]=c[28];
}
/**/
void trnsf3(){
 short n;
    for(n=0;n<65;n++){d[n]=c[n];}
    d[2]=c[37]; d[3]=c[40]; d[5]=c[34]; d[8]=c[35];
         d[9]=c[46]; d[12]=c[47]; d[14]=c[41];    d[15]=c[44];
    d[17]=c[54]; d[20]=c[55]; d[22]=c[49];   d[23]=c[52];
         d[26]=c[61]; d[27]=c[64]; d[29]=c[58];   d[32]=c[59];
    d[34]=c[5]; d[35]=c[8]; d[37]=c[2]; d[40]=c[3];
         d[41]=c[14]; d[44]=c[15]; d[46]=c[9];    d[47]=c[12];
    d[49]=c[22]; d[52]=c[23]; d[54]=c[17];   d[55]=c[20];
         d[58]=c[29]; d[59]=c[32]; d[61]=c[26];   d[64]=c[27];
}
/**/
void trnsf4(){
 short n;
    for(n=0;n<65;n++){d[n]=c[n];}
    d[33]=c[42]; d[34]=c[41]; d[35]=c[44];   d[36]=c[43];
         d[37]=c[46]; d[38]=c[45]; d[39]=c[48];   d[40]=c[47];
    d[41]=c[34]; d[42]=c[33]; d[43]=c[36];   d[44]=c[35];
         d[45]=c[38]; d[46]=c[37]; d[47]=c[40];   d[48]=c[39];
    d[49]=c[58]; d[50]=c[57]; d[51]=c[60];   d[52]=c[59];
         d[53]=c[62]; d[54]=c[61]; d[55]=c[64];   d[56]=c[63];
    d[57]=c[50]; d[58]=c[49]; d[59]=c[52];   d[60]=c[51];
         d[61]=c[54]; d[62]=c[53]; d[63]=c[56];   d[64]=c[55];
}
/**/
void trnsf5(){
 short n;
    for(n=1;n<65;n++){d[n]=c[n];}
    d[9]=c[46]; d[10]=c[45]; d[11]=c[48];    d[12]=c[47];
         d[13]=c[42]; d[14]=c[41]; d[15]=c[44];   d[16]=c[43];
    d[25]=c[62]; d[26]=c[61]; d[27]=c[64];   d[28]=c[63];
         d[29]=c[58]; d[30]=c[57]; d[31]=c[60];   d[32]=c[59];
    d[41]=c[14]; d[42]=c[13]; d[43]=c[16];   d[44]=c[15];
         d[45]=c[10]; d[46]=c[9]; d[47]=c[12];    d[48]=c[11];
    d[57]=c[30]; d[58]=c[29]; d[59]=c[32];   d[60]=c[31];
         d[61]=c[26]; d[62]=c[25]; d[63]=c[28];   d[64]=c[27];
}
/**/
void trnsf6(){
 short n;
    for(n=1;n<65;n++){d[n]=c[n];}
    d[2]=c[46]; d[4]=c[48]; d[6]=c[42]; d[8]=c[44];
         d[10]=c[38]; d[12]=c[40]; d[14]=c[34];   d[16]=c[36];
    d[18]=c[62]; d[20]=c[64]; d[22]=c[58];   d[24]=c[60];
         d[26]=c[54]; d[28]=c[56]; d[30]=c[50];   d[32]=c[52];
    d[34]=c[14]; d[36]=c[16]; d[38]=c[10];   d[40]=c[12];
         d[42]=c[6]; d[44]=c[8]; d[46]=c[2]; d[48]=c[4];
    d[50]=c[30]; d[52]=c[32]; d[54]=c[26];   d[56]=c[28];
```

```
        d[58]=c[22]; d[60]=c[24]; d[62]=c[18];   d[64]=c[20];
}
/**/
void trnsf7(){
 short n;
    for(n=1;n<65;n++){d[n]=c[n];}
    d[3]=c[10]; d[4]=c[9]; d[5]=c[16];  d[6]=c[15];
        d[9]=c[4]; d[10]=c[3]; d[15]=c[6];  d[16]=c[5];
    d[17]=c[28]; d[18]=c[27]; d[23]=c[30];   d[24]=c[29];
        d[27]=c[18]; d[28]=c[17]; d[29]=c[24];   d[30]=c[23];
    d[35]=c[42]; d[36]=c[41]; d[37]=c[48];   d[38]=c[47];
        d[41]=c[36]; d[42]=c[35]; d[47]=c[38];   d[48]=c[37];
    d[49]=c[60]; d[50]=c[59]; d[55]=c[62];   d[56]=c[61];
        d[59]=c[50]; d[60]=c[49]; d[61]=c[56];   d[62]=c[55];
}
/**/
void trnsf8(){
 short n;
    for(n=1;n<65;n++){d[n]=c[n];}
    d[3]=c[45]; d[4]=c[46]; d[5]=c[43]; d[6]=c[44];
        d[9]=c[39]; d[10]=c[40]; d[15]=c[33];    d[16]=c[34];
    d[17]=c[63]; d[18]=c[64]; d[23]=c[57];   d[24]=c[58];
        d[27]=c[53]; d[28]=c[54]; d[29]=c[51];   d[30]=c[52];
    d[33]=c[15]; d[34]=c[16]; d[39]=c[9];    d[40]=c[10];
        d[43]=c[5]; d[44]=c[6]; d[45]=c[3]; d[46]=c[4];
    d[51]=c[29]; d[52]=c[30]; d[53]=c[27];   d[54]=c[28];
        d[57]=c[23]; d[58]=c[24]; d[63]=c[17];   d[64]=c[18];
}
/**/
void trnsf9(){
 short n;
    for(n=0;n<65;n++){d[n]=c[n];}
    d[3]=c[10]; d[4]=c[9]; d[5]=c[16];  d[6]=c[15];
        d[9]=c[4]; d[10]=c[3]; d[15]=c[6];  d[16]=c[5];
    d[17]=c[28]; d[18]=c[27]; d[23]=c[30];   d[24]=c[29];
        d[27]=c[18]; d[28]=c[17]; d[29]=c[24];   d[30]=c[23];
    d[33]=c[44]; d[34]=c[43]; d[39]=c[46];   d[40]=c[45];
        d[43]=c[34]; d[44]=c[33]; d[45]=c[40];   d[46]=c[39];
    d[51]=c[58]; d[52]=c[57]; d[53]=c[64];   d[54]=c[63];
        d[57]=c[52]; d[58]=c[51]; d[63]=c[54];   d[64]=c[53];
}
/**/
void trnsf10(){
 short n;
    for(n=1;n<65;n++){d[n]=c[n];}
    d[5]=c[12]; d[6]=c[11]; d[7]=c[10]; d[8]=c[9];
        d[9]=c[8]; d[10]=c[7]; d[11]=c[6];  d[12]=c[5];
    d[17]=c[32]; d[18]=c[31]; d[19]=c[30];   d[20]=c[29];
        d[29]=c[20]; d[30]=c[19]; d[31]=c[18];   d[32]=c[17];
    d[37]=c[44]; d[38]=c[43]; d[39]=c[42];   d[40]=c[41];
        d[41]=c[40]; d[42]=c[39]; d[43]=c[38];   d[44]=c[37];
    d[49]=c[64]; d[50]=c[63]; d[51]=c[62];   d[52]=c[61];
        d[61]=c[52]; d[62]=c[51]; d[63]=c[50];   d[64]=c[49];
}
/**/
void trnsf11(){
```

```
 short n;
    for(n=1;n<65;n++){d[n]=c[n];}
    d[5]=c[47]; d[6]=c[48]; d[7]=c[45]; d[8]=c[46];
         d[9]=c[35]; d[10]=c[36]; d[11]=c[33];    d[12]=c[34];
    d[17]=c[59]; d[18]=c[60]; d[19]=c[57];   d[20]=c[58];
         d[29]=c[55]; d[30]=c[56]; d[31]=c[53];   d[32]=c[54];
    d[33]=c[11]; d[34]=c[12]; d[35]=c[9];    d[36]=c[10];
         d[45]=c[7]; d[46]=c[8]; d[47]=c[5]; d[48]=c[6];
    d[53]=c[31]; d[54]=c[32]; d[55]=c[29];   d[56]=c[30];
         d[57]=c[19]; d[58]=c[20]; d[59]=c[17];   d[60]=c[18];
}
/**/
void trnsf12(){
 short n;
    for(n=1;n<65;n++){d[n]=c[n];}
    d[5]=c[12]; d[6]=c[11]; d[7]=c[10]; d[8]=c[9];
         d[9]=c[8]; d[10]=c[7]; d[11]=c[6];  d[12]=c[5];
    d[17]=c[32]; d[18]=c[31]; d[19]=c[30];   d[20]=c[29];
         d[29]=c[20]; d[30]=c[19]; d[31]=c[18];   d[32]=c[17];
    d[33]=c[48]; d[34]=c[47]; d[35]=c[46];   d[36]=c[45];
         d[45]=c[36]; d[46]=c[35]; d[47]=c[34];   d[48]=c[33];
    d[53]=c[60]; d[54]=c[59]; d[55]=c[58];   d[56]=c[57];
         d[57]=c[56]; d[58]=c[55]; d[59]=c[54];   d[60]=c[53];
}
/**/
/* Mirror Reflection */
void reflect(){
    d[1]=c[1]; d[2]=c[9]; d[3]=c[17]; d[4]=c[25];
         d[5]=c[33]; d[6]=c[41]; d[7]=c[49]; d[8]=c[57];
    d[9]=c[2]; d[10]=c[10]; d[11]=c[18]; d[12]=c[26];
         d[13]=c[34]; d[14]=c[42]; d[15]=c[50]; d[16]=c[58];
    d[17]=c[3]; d[18]=c[11]; d[19]=c[19]; d[20]=c[27];
         d[21]=c[35]; d[22]=c[43]; d[23]=c[51]; d[24]=c[59];
    d[25]=c[4]; d[26]=c[12]; d[27]=c[20]; d[28]=c[28];
         d[29]=c[36]; d[30]=c[44]; d[31]=c[52]; d[32]=c[60];
    d[33]=c[5]; d[34]=c[13]; d[35]=c[21]; d[36]=c[29];
         d[37]=c[37]; d[38]=c[45]; d[39]=c[53]; d[40]=c[61];
    d[41]=c[6]; d[42]=c[14]; d[43]=c[22]; d[44]=c[30];
         d[45]=c[38]; d[46]=c[46]; d[47]=c[54]; d[48]=c[62];
    d[49]=c[7]; d[50]=c[15]; d[51]=c[23]; d[52]=c[31];
         d[53]=c[39]; d[54]=c[47]; d[55]=c[55]; d[56]=c[63];
    d[57]=c[8]; d[58]=c[16]; d[59]=c[24]; d[60]=c[32];
         d[61]=c[40]; d[62]=c[48]; d[63]=c[56]; d[64]=c[64];
}
/**/
/* Diagrams: Concept of those Transformation Procedure */
/*
                      0/B      --- Transform -->      1/T
   1  2  3  4  5  6  7  8    *  2  3  *  5  *  *  8    1 37 40  4 34  6  7 35
   9 10 11 12 13 14 15 16    * 10 11  * 13  *  * 16    9 45 48 12 42 14 15 43
  17 18 19 20 21 22 23 24   17  *  * 20  * 22 23  *   54 18 19 55 21 49 52 24
  25 26 27 28 29 30 31 32   25  *  * 28  * 30 31  *   62 26 27 63 29 57 60 32
  33 34 35 36 37 38 39 40    * 34 35  * 37  *  * 40   33  5  8 36  2 38 39  3
  41 42 43 44 45 46 47 48    * 42 43  * 45  *  * 48   41 13 16 44 10 46 47 11
  49 50 51 52 53 54 55 56   49  *  * 52  * 54 55  *   22 50 51 23 53 17 20 56
  57 58 59 60 61 62 63 64   57  *  * 60  * 62 63  *   30 58 59 31 61 25 28 64
```

```
                    0/B      --- Transform -->      2/T
 1  2  3  4  5  6  7  8     *  2  *  4  5  *  7  *     1 37  3 39 34  6 36  8
 9 10 11 12 13 14 15 16     * 10  * 12 13  * 15  *     9 45 11 47 42 14 44 16
17 18 19 20 21 22 23 24     * 18  * 20 21  * 23  *    17 53 19 55 50 22 52 24
25 26 27 28 29 30 31 32     * 26  * 28 29  * 31  *    25 61 27 63 58 30 60 32
33 34 35 36 37 38 39 40     * 34  * 36 37  * 39  *    33  5 35  7  2 38  4 40
41 42 43 44 45 46 47 48     * 42  * 44 45  * 47  *    41 13 43 15 10 46 12 48
49 50 51 52 53 54 55 56     * 50  * 52 53  * 55  *    49 21 51 23 18 54 20 56
57 58 59 60 61 62 63 64     * 58  * 60 61  * 63  *    57 29 59 31 26 62 28 64
                    0/B      --- Transform -->      3/T
 1  2  3  4  5  6  7  8     *  2  3  *  5  *  *  8     1 37 40  4 34  6  7 35
 9 10 11 12 13 14 15 16     9  *  * 12  * 14 15  *    46 10 11 47 13 41 44 16
17 18 19 20 21 22 23 24    17  *  * 20  * 22 23  *    54 18 19 55 21 49 52 24
25 26 27 28 29 30 31 32     * 26 27  * 29  *  * 32    25 61 64 28 58 30 31 59
33 34 35 36 37 38 39 40     * 34 35  * 37  *  * 40    33  5  8 36  2 38 39  3
41 42 43 44 45 46 47 48    41  *  * 44  * 46 47  *    14 42 43 15 45  9 12 48
49 50 51 52 53 54 55 56    49  *  * 52  * 54 55  *    22 50 51 23 53 17 20 56
57 58 59 60 61 62 63 64     * 58 59  * 61  *  * 64    57 29 32 60 26 62 63 27
                    0/R      --- Transform -->      Reflect+2/T
 1  9 17 25 33 41 49 57     *  9  * 25 33  * 49  *     1 37 17 53 13 41 29 57
 2 10 18 26 34 42 50 58     * 10  * 26 34  * 50  *     2 38 18 54 14 42 30 58
 3 11 19 27 35 43 51 59     * 11  * 27 35  * 51  *     3 39 19 55 15 43 31 59
 4 12 20 28 36 44 52 60     * 12  * 28 36  * 52  *     4 40 20 56 16 44 32 60
 5 13 21 29 37 45 53 61     * 13  * 29 37  * 53  *     5 33 21 49  9 45 25 61
 6 14 22 30 38 46 54 62     * 14  * 30 38  * 54  *     6 34 22 50 10 46 26 62
 7 15 23 31 39 47 55 63     * 15  * 31 39  * 55  *     7 35 23 51 11 47 27 63
 8 16 24 32 40 48 56 64     * 16  * 32 40  * 56  *     8 36 24 52 12 48 28 64
                    0/B      --- Transform -->      4/T
 1  2  3  4  5  6  7  8     *  *  *  *  *  *  *  *     1  2  3  4  5  6  7  8
 9 10 11 12 13 14 15 16     *  *  *  *  *  *  *  *     9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24     *  *  *  *  *  *  *  *    17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32     *  *  *  *  *  *  *  *    25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40    33 34 35 36 37 38 39 40    42 41 44 43 46 45 48 47
41 42 43 44 45 46 47 48    41 42 43 44 45 46 47 48    34 33 36 35 38 37 40 39
49 50 51 52 53 54 55 56    49 50 51 52 53 54 55 56    58 57 60 59 62 61 64 63
57 58 59 60 61 62 63 64    57 58 59 60 61 62 63 64    50 49 52 51 54 53 56 55
                    0/B      --- Transform -->      5/T
 1  2  3  4  5  6  7  8     *  *  *  *  *  *  *  *     1  2  3  4  5  6  7  8
 9 10 11 12 13 14 15 16     9 10 11 12 13 14 15 16    46 45 48 47 42 41 44 43
17 18 19 20 21 22 23 24     *  *  *  *  *  *  *  *    17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32    25 26 27 28 29 30 31 32    62 61 64 63 58 57 60 59
33 34 35 36 37 38 39 40     *  *  *  *  *  *  *  *    33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48    41 42 43 44 45 46 47 48    14 13 16 15 10  9 12 11
49 50 51 52 53 54 55 56     *  *  *  *  *  *  *  *    49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64    57 58 59 60 61 62 63 64    30 29 32 31 26 25 28 27
                    0/B      --- Transform -->      6/T
 1  2  3  4  5  6  7  8     *  2  *  4  *  6  *  8     1 46  3 48  5 42  7 44
 9 10 11 12 13 14 15 16     * 10  * 12  * 14  * 16     9 38 11 40 13 34 15 36
17 18 19 20 21 22 23 24     * 18  * 20  * 22  * 24    17 62 19 64 21 58 23 60
25 26 27 28 29 30 31 32     * 26  * 28  * 30  * 32    25 54 27 56 29 50 31 52
33 34 35 36 37 38 39 40     * 34  * 36  * 38  * 40    33 14 35 16 37 10 39 12
41 42 43 44 45 46 47 48     * 42  * 44  * 46  * 48    41  6 43  8 45  2 47  4
49 50 51 52 53 54 55 56     * 50  * 52  * 54  * 56    49 30 51 32 53 26 55 28
57 58 59 60 61 62 63 64     * 58  * 60  * 62  * 64    57 22 59 24 61 18 63 20
                    0/B      --- Transform -->      7/T
 1  2  3  4  5  6  7  8     *  *  3  4  5  6  *  *     1  2 10  9 16 15  7  8
 9 10 11 12 13 14 15 16     9 10  *  *  *  * 15 16     4  3 11 12 13 14  6  5
17 18 19 20 21 22 23 24    17 18  *  *  *  * 23 24    28 27 19 20 21 22 30 29
25 26 27 28 29 30 31 32     *  * 27 28 29 30  *  *    25 26 18 17 24 23 31 32
33 34 35 36 37 38 39 40     *  * 35 36 37 38  *  *    33 34 42 41 48 47 39 40
41 42 43 44 45 46 47 48    41 42  *  *  *  * 47 48    36 35 43 44 45 46 38 37
49 50 51 52 53 54 55 56    49 50  *  *  *  * 55 56    60 59 51 52 53 54 62 61
57 58 59 60 61 62 63 64     *  * 59 60 61 62  *  *    57 58 50 49 56 55 63 64
```

```
                    0/B      --- Transform -->      8/T
 1  2  3  4  5  6  7  8    *  *  3  4  5  6  *  *    1  2 45 46 43 44  7  8
 9 10 11 12 13 14 15 16    9 10  *  *  *  * 15 16   39 40 11 12 13 14 33 34
17 18 19 20 21 22 23 24   17 18  *  *  *  * 23 24   63 64 19 20 21 22 57 58
25 26 27 28 29 30 31 32    *  * 27 28 29 30  *  *   25 26 53 54 51 52 31 32
33 34 35 36 37 38 39 40   33 34  *  *  *  * 39 40   15 16 35 36 37 38  9 10
41 42 43 44 45 46 47 48    *  * 43 44 45 46  *  *   41 42  5  6  3  4 47 48
49 50 51 52 53 54 55 56    *  * 51 52 53 54  *  *   49 50 29 30 27 28 55 56
57 58 59 60 61 62 63 64   57 58  *  *  *  * 63 64   23 24 59 60 61 62 17 18
                    0/B      --- Transform -->      9/T
 1  2  3  4  5  6  7  8    *  *  3  4  5  6  *  *    1  2 10  9 16 15  7  8
 9 10 11 12 13 14 15 16    9 10  *  *  *  * 15 16    4  3 11 12 13 14  6  5
17 18 19 20 21 22 23 24   17 18  *  *  *  * 23 24   28 27 19 20 21 22 30 29
25 26 27 28 29 30 31 32    *  * 27 28 29 30  *  *   25 26 18 17 24 23 31 32
33 34 35 36 37 38 39 40   33 34  *  *  *  * 39 40   44 43 35 36 37 38 46 45
41 42 43 44 45 46 47 48    *  * 43 44 45 46  *  *   41 42 34 33 40 39 47 48
49 50 51 52 53 54 55 56    *  * 51 52 53 54  *  *   49 50 58 57 64 63 55 56
57 58 59 60 61 62 63 64   57 58  *  *  *  * 63 64   52 51 59 60 61 62 54 53
                    0/B      --- Transform -->     10/T
 1  2  3  4  5  6  7  8    *  *  *  *  5  6  7  8    1  2  3  4 12 11 10  9
 9 10 11 12 13 14 15 16    9 10 11 12  *  *  *  *    8  7  6  5 13 14 15 16
17 18 19 20 21 22 23 24   17 18 19 20  *  *  *  *   32 31 30 29 21 22 23 24
25 26 27 28 29 30 31 32    *  *  *  * 29 30 31 32   25 26 27 28 20 19 18 17
33 34 35 36 37 38 39 40    *  *  *  * 37 38 39 40   33 34 35 36 44 43 42 41
41 42 43 44 45 46 47 48   41 42 43 44  *  *  *  *   40 39 38 37 45 46 47 48
49 50 51 52 53 54 55 56   49 50 51 52  *  *  *  *   64 63 62 61 53 54 55 56
57 58 59 60 61 62 63 64    *  *  *  * 61 62 63 64   57 58 59 60 52 51 50 49
                    0/B      --- Transform -->     11/T
 1  2  3  4  5  6  7  8    *  *  *  *  5  6  7  8    1  2  3  4 47 48 45 46
 9 10 11 12 13 14 15 16    9 10 11 12  *  *  *  *   35 36 33 34 13 14 15 16
17 18 19 20 21 22 23 24   17 18 19 20  *  *  *  *   59 60 57 58 21 22 23 24
25 26 27 28 29 30 31 32    *  *  *  * 29 30 31 32   25 26 27 28 55 56 53 54
33 34 35 36 37 38 39 40   33 34 35 36  *  *  *  *   11 12  9 10 37 38 39 40
41 42 43 44 45 46 47 48    *  *  *  * 45 46 47 48   41 42 43 44  7  8  5  6
49 50 51 52 53 54 55 56    *  *  *  * 53 54 55 56   49 50 51 52 31 32 29 30
57 58 59 60 61 62 63 64   57 58 59 60  *  *  *  *   19 20 17 18 61 62 63 64
                    0/B      --- Transform -->     12/T
 1  2  3  4  5  6  7  8    *  *  *  *  5  6  7  8    1  2  3  4 12 11 10  9
 9 10 11 12 13 14 15 16    9 10 11 12  *  *  *  *    8  7  6  5 13 14 15 16
17 18 19 20 21 22 23 24   17 18 19 20  *  *  *  *   32 31 30 29 21 22 23 24
25 26 27 28 29 30 31 32    *  *  *  * 29 30 31 32   25 26 27 28 20 19 18 17
33 34 35 36 37 38 39 40   33 34 35 36  *  *  *  *   48 47 46 45 37 38 39 40
41 42 43 44 45 46 47 48    *  *  *  * 45 46 47 48   41 42 43 44 36 35 34 33
49 50 51 52 53 54 55 56    *  *  *  * 53 54 55 56   49 50 51 52 60 59 58 57
57 58 59 60 61 62 63 64   57 58 59 60  *  *  *  *   56 55 54 53 61 62 63 64
                    0/B      --- Transform -->     Reflect/
 1  2  3  4  5  6  7  8    *  2  3  4  5  6  7  8    1  9 17 25 33 41 49 57
 9 10 11 12 13 14 15 16    9  * 11 12 13 14 15 16    2 10 18 26 34 42 50 58
17 18 19 20 21 22 23 24   17 18  * 20 21 22 23 24    3 11 19 27 35 43 51 59
25 26 27 28 29 30 31 32   25 26 27  * 29 30 31 32    4 12 20 28 36 44 52 60
33 34 35 36 37 38 39 40   33 34 35 36  * 38 39 40    5 13 21 29 37 45 53 61
41 42 43 44 45 46 47 48   41 42 43 44 45  * 47 48    6 14 22 30 38 46 54 62
49 50 51 52 53 54 55 56   49 50 51 52 53 54  * 56    7 15 23 31 39 47 55 63
57 58 59 60 61 62 63 64   57 58 59 60 61 62 63  *    8 16 24 32 40 48 56 64
*/
/**/
```

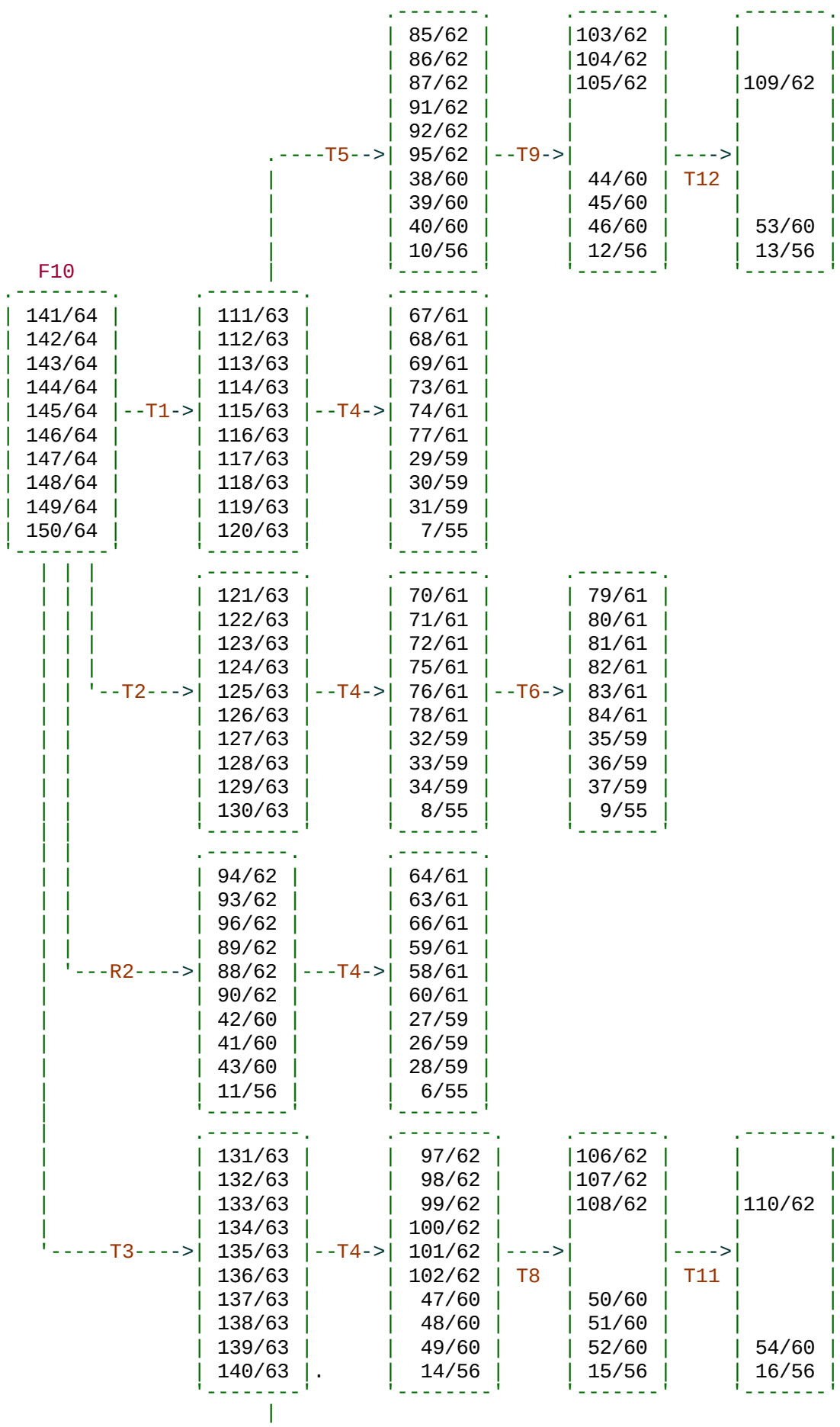
The next list shows all the recent result of my elaborate reconstruction.

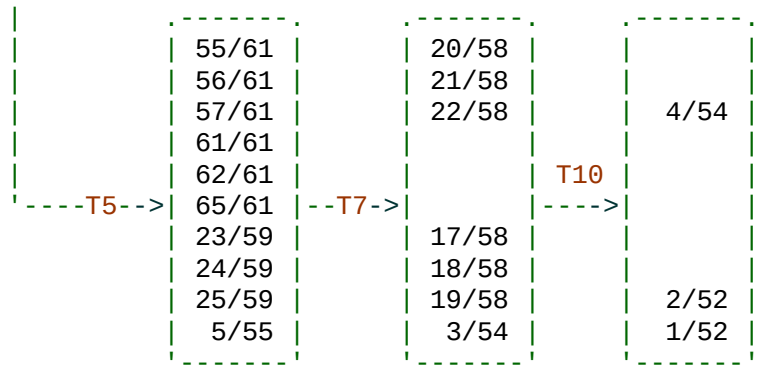
**** How to Make All 150 fron F10 of G1? ****

```
      [G1]          [G2]          [G3]         [G4]          [G5]
No.1
[41] 141/64+--T1-> 111/63+--T5->  85/62 --T9-> 103/62
[ 1]       |               '--T4->  67/61
[11]       |--T2-> 121/63 --T4->  70/61 --T6->  79/61
[31]       |--R2->  94/62 --T4->  64/61
[21]       '--T3-> 131/63+--T4->  97/62 --T8-> 106/62
[51]                       '--T5->  55/61 --T7->  20/58
No.2
[42] 142/64+--T1-> 112/63+--T5->  86/62 --T9-> 104/62
[ 2]       |               '--T4->  68/61
[12]       |--T2-> 122/63 --T4->  71/61 --T6->  80/61
[32]       |--R2->  93/62 --T4->  63/61
[22]       '--T3-> 132/63+--T4->  98/62 --T8-> 107/62
[52]                       '--T5->  56/61 --T7->  21/58
No.3
[43] 143/64+--T1-> 113/63+--T5->  87/62 --T9-> 105/62 --T12-> 109/62
[ 3]       |               '--T4->  69/61
[13]       |--T2-> 123/63 --T4->  72/61 --T6->  81/61
[33]       |--R2->  96/62 --T4->  66/61
[23]       '--T3-> 133/63+--T4->  99/62 --T8-> 108/62 --T11-> 110/62
[53]                       '--T5->  57/61 --T7->  22/58 --T10->   4/54
No.4
[44] 144/64+--T1-> 114/63+--T5->  91/62
[ 4]       |               '--T4->  73/61
[14]       |--T2-> 124/63 --T4->  75/61 --T6->  82/61
[34]       |--R2->  89/62 --T4->  59/61
[24]       '--T3-> 134/63+--T4-> 100/62
[54]                       '--T5->  61/61
NO.5
[45] 145/64+--T1-> 115/63+--T5->  92/62
[ 5]       |               '--T4->  74/61
[15]       |--T2-> 125/63 --T4->  76/61 --T6->  83/61
[35]       |--R2->  88/62 --T4->  58/61
[25]       '--T3-> 135/63+--T4-> 101/62
[55]                       '--T5->  62/61
No.6
[46] 146/64+--T1-> 116/63+--T5->  95/62
[ 6]       |               '--T4->  77/61
[16]       |--T2-> 126/63 --T4->  78/61 --T6->  84/61
[36]       |--R2->  90/62 --T4->  60/61
[26]       '--T3-> 136/63+--T4-> 102/62
[56]                       '--T5->  65/61
No.7
[47] 147/64+--T1-> 117/63+--T5->  38/60 --T9->  44/60
[ 7]       |               '--T4->  29/59
[17]       |--T2-> 127/63 --T4->  32/59 --T6->  35/59
[37]       |--R2->  42/60 --T4->  27/59
[27]       '--T3-> 137/63+--T4->  47/60 --T8->  50/60
[57]                       '--T5->  23/59 --T7->  17/58
No.8
[48] 148/64+--T1-> 118/63+--T5->  39/60 --T9->  45/60
[ 8]       |               '--T4->  30/59
[18]       |--T2-> 128/63 --T4->  33/59 --T6->  36/59
[38]       |--R2->  41/60 --T4->  26/59
[28]       '--T3-> 138/63+--T4->  48/60 --T8->  51/60
[58]                       '--T5->  24/59 --T7->  18/58
No.9
[49] 149/64+--T1-> 119/63+--T5->  40/60 --T9->  46/60 --T12->  53/60
[ 9]       |               '--T4->  31/59
[19]       |--T2-> 129/63 --T4->  34/59 --T6->  37/59
[39]       |--R2->  43/60 --T4->  28/59
[29]       '--T3-> 139/63+--T4->  49/60 --T8->  52/60 --T11->  54/60
[59]                       '--T5->  25/59 --T7->  19/58 --T10->   2/52
No.10
[50] 150/64+--T1-> 120/63+--T5->  10/56 --T9->  12/56 --T12->  13/56
[10]       |               '--T4->   7/55
[20]       |--T2-> 130/63 --T4->   8/55 --T6->   9/55
[40]       |--R2->  11/56 --T4->   6/55
[30]       '--T3-> 140/63+--T4->  14/56 --T8->  15/56 --T11->  16/56
[60]                       '--T5->   5/55 --T7->   3/54 --T10->   1/52
```

Let me show you another beautiful diagram illustrating how to make F10 into all 150:

**** Process Transition Diagram for Our System of Transformations ****





[**Count:** 10 40 60 31 9]

[**This diagram was composed by Kanji Setsuda on January 19, 2003**]

Anyhow I found we could reconstruct all 150 solutions of M&C&P magic squares of order 8 from the Fundamental Ten.
They proved to be really the children of the F10. They belong to the same noble family.

But I should say it is more important for us to have known of the inner structures of those squares.
Each reactivity to any specific transformation is one the most interesting properties. Any solution has its personality, but it also has the common character with its brothers and sisters.
This new knowledge might give us some certain basis for our ‘Fundamental Study’.

(Written in English on January 24, 2003 by Kanji Setsuda)

*** **E-Mail Address: jag12001@nifty.ne.jp**