

**Chapter 6: Fundamental Studies of Higher-Dimensional Extra-Cubic Magic Objects and their Decompositions: Kanji Setsuda**

**Section 3-4: The Most Fundamental Solution and the Transformation System of 'C&C' MC444 and MS88**

#1. In the former sections we studied about 6-Dimensional Extra-Cubic magic Objects of order 2 and their decompositions. We have obtained the complete lists of Fundamental 15 Solutions of 'Composite & Complete' Magic Cubes of order 4 and the Fundamental 10 Solutions of 'C&C' Magic Squares of order 8 by down-converting dimensions from the original 6-D object into 3-D or 2-D ones.

And in the last sections we studied about such the transformation systems as we could reconstruct the Fundamental 90 Solutions of 'C&C' Magic Squares of order 8 from the Most Fundamental 10, and have recently got the Fundamental 120 Solutions of C&C MC444 from the Most Fundamental 15.

We have thus known of the true meaning of the 'fundamental solutions'.

But I confess I am not yet satisfied with it. My intellectual request is growing wilder. Can't we find the 'Most Fundamental Solution', the only one of all?

Can't we invent any transformation methods for the only Fundamental Solution that makes into all 15 solutions of 'C&C' MC444 and 10 solutions of C&C MS88?

#2. This request arises from the discovery that there is essentially only one solution for the 6-Dimensional Extra-Cubic magic Object of order 2.

		Classic/		/N2i	
1-----2	3-----4	000000	000001	000010	000011
17-----+18	19-----+20	010000	---010001	010010	010011
33-----+34	35-----+36	100000	---100001	100010	100011
49-----+50	51-----+52	110000	---110001	110010	110011
5-- --- ---6	7-- --- ---8	000100	---000101	000110	000111
21-- --- ---22	23-- --- ---24	010100	---010101	010110	010111
37-- --- ---38	39-- --- ---40	100100	---100101	100110	100111
53-----54	55-----56	110100	---110101	110110	110111
9-----10	11-----12	001000	001001	001010	001011
25-----+26	27-----+28	011000	011001	011010	011011
41-----+42	43-----+44	101000	101001	101010	101011
57-----+58	59-----+60	111000	111001	111010	111011
13-- --- ---14	15-- --- ---16	001100	001101	001110	001111
29-- --- ---30	31-- --- ---32	011100	011101	011110	011111
45-- --- ---46	47-- --- ---48	101100	101101	101110	101111
61-----62	63-----64	111100	111101	111110	111111

In the 6-D world there are 6 axes which meet on the Origin  $n1(0,0,0,0,0,0)$ , and there are 6 adjacent positions next to  $n1$ :  $n2(0,0,0,0,0,1)$ ,  $n3(0,0,0,0,1,0)$ ,  $n5(0,0,0,1,0,0)$ ,  $n9(0,0,1,0,0,0)$ ,  $n17(0,1,0,0,0,0)$  and  $n33(1,0,0,0,0,0)$ .

We can see no difference among those six positions at all. We can find neither any rank nor any determined order among them in advance.

				SC4^3/8^2					
1-----2	3-----4	1	2	3	4	17	18	19	20
17	18	5	6	7	8	21	22	23	24
33	6	34	7	35	8	36	9	10	11
49	22	50	23	51	24	52	12	13	14
9	37	10	38	11	39	12	40	15	16
25	53	26	54	27	55	28	56	33	34
13	41	14	42	15	43	16	44	37	38
29	57	30	58	31	59	32	60	41	42
45	57	30	58	31	59	32	60	45	46
61	61	62	63	64	64	64	64	47	48
61	61	62	63	64	64	64	64	45	46

```

1-----2-----4-----3          CC4^3/8^2
| 17          18          20          | 19          1 2 4 3 29 30 32 31
5 49          6 50          8 52          | 51          5 6 8 7 25 26 28 27
| 21          33-----22-----34-----24-----36-----+23-----35 13 14 16 15 17 18 20 19
| 29          37          30          38          32          40          | 31          55          | 9 10 12 11 21 22 24 23
9-----61-----10-----62-----12-----64-----11 63          | 36 35 33 34 64 63 61 62
 25          45          26          46          28          48          | 27          47          | 40 39 37 38 60 59 57 58
 57          |          58          60          59          | 44 43 41 42 56 55 53 54
 41-----42-----44-----43
['C&C' Type Converted from the 'SC' Type above]

```

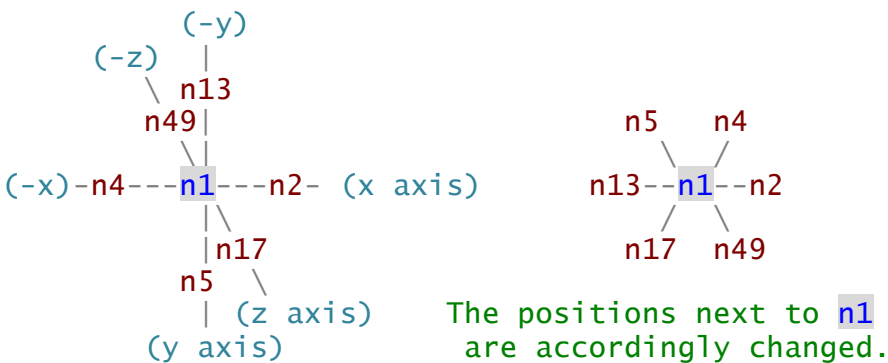
In our 'C&C' MC444 type of objects converted from ECO2<sup>6</sup> each one of these six positions could only take any one of the values {63, 62, 60, 56, 48, 32} and nothing else, as we have found before.

There is only one meaning for the fact that the 6 positions {n2, n3, n5, n9, n17, n33} next to n1 could only take those six values {63, 62, 60, 56, 48, 32}.

The permutations of both positions and values are essentially meaningless in the 6-Dimensional world, because their own high symmetry allows changing their faces in 5760 ways in total (n1=1) and proving they express the same substance.

#3. But each solution of C&C MC444 obtains a different face from others in the 3-D world we live in. When we convert the dimensions from 6 down to 3, we have only 3 axes and the 6 positions next to n1 sitting coupled on both sides of n1.

3-Dimensional Magic Cube <-> 6-Dimensional Magic Object  
in Extended Space below



\*\* Basic Forms for Complete Type of Magic Cubes of Order 4 \*\*

```

13
49 |          Basic/          /Solution
4-----n1-----n2-----n3-----n4          1-----63-----4-----62
| 17          18          19          | 20          | 48          18          45          | 19
n5 33          6 34          7 35          8 36          60 49          6 15          57 52          7 14
| 21          49-----22-----50-----23-----51-----+24-----52 21 32-----43-----34-----24-----29-----+42-----35
n9 37          | 10          38          11          39          12          40          | 13 12          | 51          54          16          9          50          55          |
| 25          53          26          54          27          55          | 28          56          | 36          37          30          27          33          40          | 31          26
13-----41-----14-----42-----15-----43-----16 44          | 56-----61-----10-----3-----53-----64-----11 2          |
 29          57          30          58          31          59          32          60          25 20          39          46          28          17          38          47
 45          |          46          47          48          | 8          |          58          5          59          |
 61-----62-----63-----64          41-----23-----44-----22

```

When we see this visible object, we cannot but notice the different faces of each object and count them one by one separately as many as 120 when n1=1.

There are 15 Most Fundamental Solutions for 'C&C' type of MC444 (n1=1).

It is because there are 15 different ways of making combinations of 6 positions and 6 values, and because there is only one way of rearranging all the rest. As a result this count 15 becomes equal to the count of our F15 solutions. Let me show you here the list of our objects as follows.

**\*\* 'Composite & Complete' Magic Cubes 4x4x4 \*\***  
**\*\* List of the Fundamental 15 Solutions \*\***

1/

```

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

```

2/

```

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

```

3/

```

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

```

4/

```

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

```

5/

```

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

```

6/

```

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

```

7/

```

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

```

8/

```

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

```

9/

```

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

```

10/

```

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

```

11/

```

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

```

12/

```

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

```

13/

```

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

```

14/

```

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

```

15/

```

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

```

[Count = 15] OK!

\*\* Calculated and Listed by Kanji Setsuda on Oct. 2, 2013 with MacOSX 10.8.5 & Xcode 5.0 \*\*

<p>[ 1 ]</p> <pre> 32 56   \ \ 60--1--63   \ \  62 48 </pre>	<p>[ 2 ]</p> <pre> 32 48   \ \ 60--1--63   \ \  62 56 </pre>	<p>[ 3 ]</p> <pre> 48 32   \ \ 60--1--63   \ \  62 56 </pre>	<p>[ 4 ]</p> <pre> 32 60   \ \ 56--1--63   \ \  62 48 </pre>	<p>[ 5 ]</p> <pre> 32 48   \ \ 56--1--63   \ \  62 60 </pre>
<p>[ 6 ]</p> <pre> 48 32   \ \ 56--1--63   \ \  62 60 </pre>	<p>[ 7 ]</p> <pre> 32 60   \ \ 48--1--63   \ \  62 56 </pre>	<p>[ 8 ]</p> <pre> 32 56   \ \ 48--1--63   \ \  62 60 </pre>	<p>[ 9 ]</p> <pre> 56 32   \ \ 48--1--63   \ \  62 60 </pre>	<p>[10]</p> <pre> 48 60   \ \ 32--1--63   \ \  62 56 </pre>
<p>[11]</p> <pre> 48 56   \ \ 32--1--63   \ \  62 60 </pre>	<p>[12]</p> <pre> 56 48   \ \ 32--1--63   \ \  62 60 </pre>	<p>[13]</p> <pre> 32 56   \ \ 62--1--63   \ \  60 48 </pre>	<p>[14]</p> <pre> 32 48   \ \ 62--1--63   \ \  60 56 </pre>	<p>[15]</p> <pre> 48 32   \ \ 62--1--63   \ \  60 56 </pre>

I presented the list of 'Adjacent Numbers next to n1' above, so that you may easily understand what I mean.

Since these 15 solutions are originally taken out of the 6-Dimensional Extra Cubic Object of order 2, each of them must have inherited some important property from the original existence, as you guess.

What is it like, then?

#4. These F15 must naturally have their common memory of 'brothers and sisters', I would say. It is because they have been born from the same Father.

Now let's invent our Inner Transformation System making from one into all 15. Our theoretical goal is the discovery of only one 'Most Fundamental Solution'.

First of all, let's take any two similar solutions out of the list of F15 and compare them precisely just in the same way as we have done before.

- (1) Find and collect similar pairs half of whose elements are the same.
- (2) Find and collect any possible transformation methods among those pairs.
- (3) Invent our total Transformation System for making one into 15 Solutions.

Watch the next lists of comparisons and the results of my collections.  
Let me show you only a part of my long investigation here as follows.

\*\* Find Possible Transformations among the F15 Solutions \*\*  
\*\* of 'Composite & Complete' Magic Cubes of Order 4 \*\*

[1]

<p style="color: blue;">1/F</p> <pre> 1-----63-----6-----60  48   18   43    21 62 49  4 15  57 54  7 12  19 32--45-34--24-27--42-37 11 14   53 52 16 9  50 55    38 35 28 29 33 40  31 26 56-59- -10--5--51-64--13 2    25 22 39 44 30 17 36 47    8     58   3   61    41-----23-----46-----20                 </pre>	<p style="color: blue;">&lt;-- Transform: Type 1 --&gt;</p> <pre> *-----*-----10-----56   *   *   *    39   25 * * * * 53 58 11 8   * *--*--*--*--28-23--38-41 7 *   57 * * 5 * 59    42 * 24 * * 44   * 22 60-55- -6--9--*-*--* * *    21 26 43 40 * * * *    12   54 * * *    37-----27-----*-----*                 </pre>	<p style="color: blue;">4/F</p> <pre> 1-----63-----10-----56  48   18   39    25 62 49  4 15  53 58 11 8  19 32--45-34--28-23--38-41 7 14   57 52 16 5  50 59    42 35 24 29 33 44  31 22 60-55- -6--9--51-64--13 2    21 26 43 40 30 17 36 47    12   54   3   61    37-----27-----46-----20                 </pre>
---	--	--

<p style="color: blue;">1/F</p> <pre> 1-----63-----6-----60  48   18   43    21 62 49  4 15  57 54  7 12  19 32--45-34--24-27--42-37 11 14   53 52 16 9  50 55    38 35 28 29 33 40  31 26 56-59- -10--5--51-64--13 2    25 22 39 44 30 17 36 47    8     58   3   61    41-----23-----46-----20                 </pre>	<p style="color: blue;">&lt;-- Transform: Type 4 --&gt;</p> <pre> *-----*-----*-----*  56   10   51    13 * 41 * 23 * 46 * 20  11 *--53-*--16-*--50-* 19 22   45 44 24 17 42 47     * * * * * * *   * * 48-* -18--*-43-*--21 *    * 14 * 52 * 9 * 55  *   * * * * *    49-----15-----54-----12                 </pre>	<p style="color: blue;">2/F</p> <pre> 1-----63-----6-----60  56   10   51    13 62 41  4 23  57 46  7 20  11 32--53-34--16-27--50-37 19 22   45 44 24 17 42 47    38 35 28 29 33 40  31 26 48-59- -18--5--43-64--21 2    25 14 39 52 30 9 36 55    8     58   3   61    49-----15-----54-----12                 </pre>
---	---	--

<p style="color: blue;">1/F</p> <pre> 1-----63-----6-----60  48   18   43    21 62 49  4 15  57 54  7 12  19 32--45-34--24-27--42-37 11 14   53 52 16 9  50 55    38 35 28 29 33 40  31 26 56-59- -10--5--51-64--13 2    25 22 39 44 30 17 36 47    8     58   3   61    41-----23-----46-----20                 </pre>	<p style="color: blue;">&lt;-- Transform: Type 5 --&gt;</p> <pre> *-----*-----18-----48  60   6   *   *     * * 37 * 27 45 * 19 *   7 *--57-*--*-15--*-49 * 26   * 40 28 * 38 *    50 * 16 * * 52   * 14 *-47- -*-17--39-*--25 *    13 * 51 * * 5 * 59  20   46 * * *    *-----*-----58-----8                 </pre>	<p style="color: blue;">8/F</p> <pre> 1-----63-----18-----48  60   6   43    21 62 37  4 27  45 54 19 12   7 32--57-34--24-15--42-49 11 26   53 40 28 9  38 55    50 35 16 29 33 52  31 14 56-47- -10--17--39-64--25 2    13 22 51 44 30 5 36 59  20   46   3   61    41-----23-----58-----8                 </pre>
---	--	---

<p style="color: blue;">1/F</p> <pre> 1-----63-----6-----60  48   18   43    21 62 49  4 15  57 54  7 12  19 32--45-34--24-27--42-37 11 14   53 52 16 9  50 55    38 35 28 29 33 40  31 26 56-59- -10--5--51-64--13 2    25 22 39 44 30 17 36 47    8     58   3   61    41-----23-----46-----20                 </pre>	<p style="color: blue;">&lt;-- Transform: Type 6 --&gt;</p> <pre> *-----*-----4-----62   *   *   *    45   19 60 * 6 * * 52 * 14  21 *--43-*--*-29--*-35 13 12   51 54 * * * *    36 37 30 27 * *   * * *-61- -*-3--53-*--11 *    * 20 * 46 28 * 38 *  *   * * 5 59    *-----*-----44-----22                 </pre>	<p style="color: blue;">13/F</p> <pre> 1-----63-----4-----62  48   18   45    19 60 49  6 15  57 52  7 14  21 32--43-34--24-29--42-35 13 12   51 54 16 9  50 55    36 37 30 27 33 40  31 26 56-61- -10--3--53-64--11 2    25 20 39 46 28 17 38 47    8     58   5   59    41-----23-----44-----22                 </pre>
---	---	--

[2]

2/F

```

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

```

<-- Transform: Type 1 -->

```

*-----*-----18-----48
| *      *      39      |25
* * * * 45 58  19  8
| * *--*--*--28-15--38-49
7 * | 57 * * * 5 * 59 |
|50 * 16 * * * 52 | * 14
60--47-|6-17--*--*--* * |
 13 26 51 40 * * * * *
   20 | 46      *      * |
 37-----27-----*-----*

```

7/F

```

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

```

2/F

```

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

```

<-- Transform: Type 2 -->

```

*-----*-----*-----*
| *      *      *      | *
* 25 * 39 * 30 * 36
| * 48--*--18--*--43--*--21
35 38 | 29 28  40 33  26 31 |
|22 19 44 45  17 24  47 42
32--*--|34--*--27--*--37 * |
 41 * 23 * 46 * 20 *
   * | *      *      * |
   *-----*-----*-----*

```

3/F

```

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

```

2/F

```

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

```

<-- Transform: Type 4 -->

```

*-----*-----*-----*
|48      18      43      |21
* 49 * 15 * 54 * 12
|19 *--45--*--24--*--42--*
11 14 | 53 52  16 9  50 55 |
| * * * * * * * | * *
56--*--|10--*--51--*--13 * |
 * 22 * 44 * 17 * 47
   * | *      *      * |
 41-----23-----46-----20

```

1/F

```

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

```

2/F

```

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

```

<-- Transform: Type 5 -->

```

*-----*-----10-----56
|60      6      *      | *
* 37 * 27 53 * 11 *
| 7 *--57--*--*--23--*--41
* 26 | * 40 28 * 38 * |
|42 * 24 * * 44 | * 22
*--55-|*--9--39--*--25 * |
 21 * 43 * * 5 * 59
   12 | 54      *      * |
   *-----*-----58-----8

```

5/F

```

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

```

2/F

```

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

```

<-- Transform: Type 6 -->

```

*-----*-----4-----62
| *      *      53      |11
60 * 6 * * * 44 * 22
|13 *--51--*--*--29--*--35
21 20 | 43 46 * * * * |
|36 37 30 27 * * * * |
*--61-|*--3--45--*--19 * |
 * 12 * 54 28 * 38 *
   * | *      5      59 |
   *-----*-----52-----14

```

14/F

```

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

```

...

[5]

5/F

```

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

```

<-- Transform: Type 1 -->

```

*-----*-----18-----48
| *      *      43      |21
* * * * 45 54  19  12
| * *--*--*--24-15--42-49
11 * | 53 * * * 9 * 55 |
|50 * 16 * * * 52 | * 14
56--47-|10-17--*--*--* * |
 13 22 51 44 * * * * *
   20 | 46      *      * |
 41-----23-----*-----*

```

8/F

```

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

```

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

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

Let's come up to our conclusion quickly.

**\*\* Collect the Transformation Methods for the Fundamental 15 Solutions of 'Composite & Complete' Magic Cubes of Order 4 \*\***

[Transform: Type 1]

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

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

[Transform: Type 2]

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

[Transform: Type 3]

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

[Transform: Type 4]

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

[Transform: Type 5]

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

[Transform: Type 6]

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

Type 1~3 above do the same thing in three different directions: up-down, front-back and left-right. Type 4~6 do another transformation in the same way.

#5. I found the 6 types of transformation methods in all. Let me show you the lists of Conceptual Diagrams, Program Codes and Process Transition now.

\*\* Conceptual Diagrams for each Transformation System \*\*  
 \*\* of 'Composite & Complete' Magic Cubes of Order 4: \*\*

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



```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

//
/** Transformation Programs *
void trans1(){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
dnm[3]=cnm[14]; dnm[19]=cnm[30]; dnm[35]=cnm[46]; dnm[51]=cnm[62];
dnm[4]=cnm[13]; dnm[20]=cnm[29]; dnm[36]=cnm[45]; dnm[52]=cnm[61];
dnm[7]=cnm[10]; dnm[23]=cnm[26]; dnm[39]=cnm[42]; dnm[55]=cnm[58];
dnm[8]=cnm[9]; dnm[24]=cnm[25]; dnm[40]=cnm[41]; dnm[56]=cnm[57];

```

```

    dnm[9]=cnm[8];   dnm[25]=cnm[24]; dnm[41]=cnm[40];   dnm[57]=cnm[56];
    dnm[10]=cnm[7];  dnm[26]=cnm[23]; dnm[42]=cnm[39];   dnm[58]=cnm[55];
    dnm[13]=cnm[4];  dnm[29]=cnm[20]; dnm[45]=cnm[36];   dnm[61]=cnm[52];
    dnm[14]=cnm[3];  dnm[30]=cnm[19]; dnm[46]=cnm[35];   dnm[62]=cnm[51];
}
//
void trans2(){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
    dnm[9]=cnm[53];   dnm[10]=cnm[54]; dnm[11]=cnm[55];   dnm[12]=cnm[56];
    dnm[13]=cnm[49];  dnm[14]=cnm[50]; dnm[15]=cnm[51];   dnm[16]=cnm[52];
    dnm[25]=cnm[37];  dnm[26]=cnm[38]; dnm[27]=cnm[39];   dnm[28]=cnm[40];
    dnm[29]=cnm[33];  dnm[30]=cnm[34]; dnm[31]=cnm[35];   dnm[32]=cnm[36];
    dnm[33]=cnm[29];  dnm[34]=cnm[30]; dnm[35]=cnm[31];   dnm[36]=cnm[32];
    dnm[37]=cnm[25];  dnm[38]=cnm[26]; dnm[39]=cnm[27];   dnm[40]=cnm[28];
    dnm[49]=cnm[13];  dnm[50]=cnm[14]; dnm[51]=cnm[15];   dnm[52]=cnm[16];
    dnm[53]=cnm[9];   dnm[54]=cnm[10]; dnm[55]=cnm[11];   dnm[56]=cnm[12];
}
//
void trans3(){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
    dnm[3]=cnm[50];   dnm[7]=cnm[54];   dnm[11]=cnm[58];   dnm[15]=cnm[62];
    dnm[4]=cnm[49];   dnm[8]=cnm[53];   dnm[12]=cnm[57];   dnm[16]=cnm[61];
    dnm[19]=cnm[34];  dnm[23]=cnm[38];  dnm[27]=cnm[42];   dnm[31]=cnm[46];
    dnm[20]=cnm[33];  dnm[24]=cnm[37];  dnm[28]=cnm[41];   dnm[32]=cnm[45];
    dnm[33]=cnm[20];  dnm[37]=cnm[24];  dnm[41]=cnm[28];   dnm[45]=cnm[32];
    dnm[34]=cnm[19];  dnm[38]=cnm[23];  dnm[42]=cnm[27];   dnm[46]=cnm[31];
    dnm[49]=cnm[4];   dnm[53]=cnm[8];   dnm[57]=cnm[12];   dnm[61]=cnm[16];
    dnm[50]=cnm[3];   dnm[54]=cnm[7];   dnm[58]=cnm[11];   dnm[62]=cnm[15];
}
//
void trans4(){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
    dnm[9]=cnm[21];   dnm[10]=cnm[22];   dnm[11]=cnm[23];   dnm[12]=cnm[24];
    dnm[13]=cnm[17];  dnm[14]=cnm[18];   dnm[15]=cnm[19];   dnm[16]=cnm[20];
    dnm[17]=cnm[13];  dnm[18]=cnm[14];   dnm[19]=cnm[15];   dnm[20]=cnm[16];
    dnm[21]=cnm[9];   dnm[22]=cnm[10];   dnm[23]=cnm[11];   dnm[24]=cnm[12];
    dnm[33]=cnm[61];  dnm[34]=cnm[62];   dnm[35]=cnm[63];   dnm[36]=cnm[64];
    dnm[37]=cnm[57];  dnm[38]=cnm[58];   dnm[39]=cnm[59];   dnm[40]=cnm[60];
    dnm[57]=cnm[37];  dnm[58]=cnm[38];   dnm[59]=cnm[39];   dnm[60]=cnm[40];
    dnm[61]=cnm[33];  dnm[62]=cnm[34];   dnm[63]=cnm[35];   dnm[64]=cnm[36];
}
//
void trans5(){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
    dnm[3]=cnm[18];   dnm[7]=cnm[22];   dnm[11]=cnm[26];   dnm[15]=cnm[30];
    dnm[4]=cnm[17];   dnm[8]=cnm[21];   dnm[12]=cnm[25];   dnm[16]=cnm[29];
    dnm[17]=cnm[4];   dnm[21]=cnm[8];   dnm[25]=cnm[12];   dnm[29]=cnm[16];
    dnm[18]=cnm[3];   dnm[22]=cnm[7];   dnm[26]=cnm[11];   dnm[30]=cnm[15];
    dnm[33]=cnm[52];  dnm[37]=cnm[56];  dnm[41]=cnm[60];   dnm[45]=cnm[64];
    dnm[34]=cnm[51];  dnm[38]=cnm[55];  dnm[42]=cnm[59];   dnm[46]=cnm[63];
    dnm[51]=cnm[34];  dnm[55]=cnm[38];  dnm[59]=cnm[42];   dnm[63]=cnm[46];
    dnm[52]=cnm[33];  dnm[56]=cnm[37];  dnm[60]=cnm[41];   dnm[64]=cnm[45];
}
//
void trans6(){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
    dnm[3]=cnm[6];   dnm[19]=cnm[22];   dnm[35]=cnm[38];   dnm[51]=cnm[54];
    dnm[4]=cnm[5];   dnm[20]=cnm[21];   dnm[36]=cnm[37];   dnm[52]=cnm[53];
    dnm[5]=cnm[4];   dnm[21]=cnm[20];   dnm[37]=cnm[36];   dnm[53]=cnm[52];
    dnm[6]=cnm[3];   dnm[22]=cnm[19];   dnm[38]=cnm[35];   dnm[54]=cnm[51];
    dnm[9]=cnm[16];  dnm[25]=cnm[32];   dnm[41]=cnm[48];   dnm[57]=cnm[64];
}

```

```

dnm[10]=cnm[15]; dnm[26]=cnm[31]; dnm[42]=cnm[47]; dnm[58]=cnm[63];
dnm[15]=cnm[10]; dnm[31]=cnm[26]; dnm[47]=cnm[42]; dnm[63]=cnm[58];
dnm[16]=cnm[9]; dnm[32]=cnm[25]; dnm[48]=cnm[41]; dnm[64]=cnm[57];
}

```

**\*\* Basic Forms for Complete Type of Magic Cubes of Order 4 \*\***

```

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

```

**#6. How do you use these transformation methods for making one into 15?**

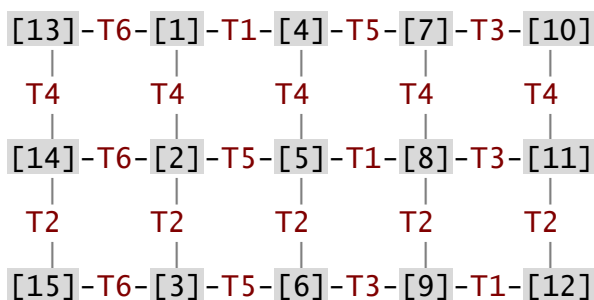
I found there are several good ways to make our job, but let me introduce you of my recent invention. I presented it in the diagrams as follows:

```

[ 1] 01-T1-04,           01-T4-02,  01-T5-08,  01-T6-13,
[ 2] 02-T1-07,  02-T2-03,  02-T4-01,  02-T5-05,  02-T6-14,
[ 3] 03-T1-10,  03-T2-02,           03-T5-06,  03-T6-15,
[ 4] 04-T1-01,           04-T4-05,  04-T5-07,
[ 5] 05-T1-08,  05-T2-06,  05-T3-12,  05-T4-04,  05-T5-02,  05-T6-15,
[ 6] 06-T1-11,  06-T2-05,  06-T3-09,           06-T5-03,  06-T6-14,
[ 7] 07-T1-02,           07-T3-10,  07-T4-08,  07-T5-04,
[ 8] 08-T1-05,  08-T2-09,  08-T3-11,  08-T4-07,  08-T5-01,
[ 9] 09-T1-12,  09-T2-08,  09-T3-06,           09-T6-13,
[10] 10-T1-03,           10-T3-07,  10-T4-11,
[11] 11-T1-06,  11-T2-12,  11-T3-08,  11-T4-10,
[12] 12-T1-09,  12-T2-11,  12-T3-05,
[13]           13-T4-14,  13-T5-09,  13-T6-01,
[14]           14-T2-15,  14-T4-13,  14-T5-06,  14-T6-02,
[15]           15-T2-14,  15-T5-05,  15-T6-03,

```

**\* Process Transition Diagram of Transformation System \***



You may start your reconstruction with any solution.

For instance, if you put No.1 solution onto the [1] position above, you may transform it into No.4 on [4] by the transformation T1. You may also transform No.1 into No.2 on [2] by T4, and transform No.1 into No.13 on [13] by T6.

For the second step begin with the answer No.2, and you can transform it into No.3 by T2, No.5 by T5 and No.14 by T6. You can also transform No.4 into No.7 by T5, transform No.14 into No.15 by T2, and transform No.5 into No.6 by T2. ...

Of course, you can begin your job with No.13 on [13] above instead of No.1. You can get No.1 from No.13 by T6, and No.14 by T4. ...

Since every types of transformation are 'reversible', you can begin your job with

any solution you personally choose.

#7. Let's come up to the final goal: Reconstruction of F15 from the representative No.1 solution. Here you are at the result list of my reconstruction as below.

\*\* Reconstruction of the Fundamental 15 Solutions of 'C&C' \*\*  
 \*\* Magic Cubes of Order 4 from the Representative One \*\*  
 \*\* Solution directly by our Transformation System \*\*

1/

```

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

2/

```

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

3/

```

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

4/

```

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

5/

```

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

6/

```

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

7/

```

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

8/

```

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

9/

```

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

10/

```

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

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

**\*\* Check any Duplication or Missing of Reconstructions \*\***

```

Err: 0
Nmb: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
-----
Cnt: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[OK!]

```

**\*\* Calculated and Listed by Kanji Setsuda on  
Sep.18, 2013 with MacOSX 10.8.5 & Xcode 4.6.3 \*\***

The check-list above tells us that we have really succeeded in reconstructing the F15 solutions from the only one representative solution No.1.

The next list of compact program is what I dictated and used for that purpose.

```

/** Transform the Only One Representative Solution into the **
/** Fundamental 15 Solutions of 'C&C' Magic Cubes of Order 4 **
/** File: 'TransCCMC4F15.c' Built by Kanji Setsuda **
/** in 2002, 2006 and On Oct.25, 2013 **
/** with MacOSX 10.9 and Xcode 5.0.1 **
//
#include <stdio.h>
//
/** Global Variables *
short cnt;
short nm[65], cnm[65], dnm[65];
short tnm[16][65];
short p[2][65];
//
/** Sub-Routiens *

```

```

void ansprint(void);
void pr2sol(short x);
//
/** Transformation System *
void recnstrct(void);
void trans1(void), trans2(void), trans3(void);
void trans4(void), trans5(void), trans6(void);
//
/** The Representative Solution Nominated *
short repr[65]={1,
    1, 63, 6, 60, 62, 4, 57, 7,
    11, 53, 16, 50, 56, 10, 51, 13,
    48, 18, 43, 21, 19, 45, 24, 42,
    38, 28, 33, 31, 25, 39, 30, 36,
    49, 15, 54, 12, 14, 52, 9, 55,
    59, 5, 64, 2, 8, 58, 3, 61,
    32, 34, 27, 37, 35, 29, 40, 26,
    22, 44, 17, 47, 41, 23, 46, 20};
//
/** Main Program **
int main(){
    short n;
    printf("\n");
    printf("*** Reconstruction of the Fundamental 15 Solutions of ***\n");
    printf("*** 'C&C' Magic Cubes of Order 4 from the Representative **\n");
    printf("*** One Solution directly by our Transformation System **\n");
    for(n=0;n<65;n++){tnm[0][n]=n;}
    printf("\n");
    printf("*** Basic Form and the Representative Solution **\n");
    for(n=0;n<65;n++){p[0][n]=tnm[0][n]; p[1][n]=repr[n];}
    pr2sol(2);
    recnstrct();    /** Reconstruction Process *
    printf("\n");
    printf("*** Result of Reconstructions **\n");
    ansprint();
    printf("\n");
    printf(" [Count = %d]\n",15);
    printf(" [OK!]\n");
    printf("\n");
    printf("*** Calculated and Listed by Kanji Setsuda on\n");
    printf("    Oct.25, 2013 with MacOSX 10.9 & Xcode 5.0.1 **\n");
    printf("\n");
    return 0;
}
//
/** Print the Result Solutions Reconstructed *
void ansprint(){
    short m,n;
    cnt=0;
    for(m=1;m<16;m++){
        for(n=0;n<65;n++){p[cnt][n]=tnm[m][n];}
        cnt++;
        if(cnt==2){pr2sol(cnt); cnt=0;}}
    if(cnt>0){pr2sol(cnt);}
}
//
/** Print 2 Solutions in a line *
void pr2sol(short x){
    short n;
    for(n=0;n<x;n++){
        printf("%28d/",p[n][0]);
        if(n==0){printf(" ");}}
    printf("\n");
    for(n=0;n<x;n++){
        printf(" %2d-----%2d-----%2d-----%2d",p[n][1],p[n][2],p[n][3],p[n][4]);
        if((n+1)<x){printf(" ");}}
}

```

```

printf("\n");
for(n=0;n<x;n++){
    printf("  l%2d      %2d      %2d      l%2d",p[n][17],p[n][18],p[n][19],p[n][20]);
    if((n+1)<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf(" %2d %2d %2d %2d %2d %2d %2d %2d",
        p[n][5],p[n][33],p[n][6],p[n][34],p[n][7],p[n][35],p[n][8],p[n][36]);
    if((n+1)<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf("  l%2d %2d---%2d--%2d---%2d--%2d---%2d--%2d",
        p[n][21],p[n][49],p[n][22],p[n][50],p[n][23],p[n][51],p[n][24],p[n][52]);
    if((n+1)<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf(" %2d %2d l %2d %2d %2d %2d %2d %2d l",
        p[n][9],p[n][37],p[n][10],p[n][38],p[n][11],p[n][39],p[n][12],p[n][40]);
    if((n+1)<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf("  l%2d %2d %2d %2d %2d %2d l%2d %2d",
        p[n][25],p[n][53],p[n][26],p[n][54],p[n][27],p[n][55],p[n][28],p[n][56]);
    if((n+1)<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf(" %2d--%2d-l-%2d--%2d---%2d--%2d---%2d %2d l",
        p[n][13],p[n][41],p[n][14],p[n][42],p[n][15],p[n][43],p[n][16],p[n][44]);
    if((n+1)<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf(" %2d %2d %2d %2d %2d %2d %2d %2d",
        p[n][29],p[n][57],p[n][30],p[n][58],p[n][31],p[n][59],p[n][32],p[n][60]);
    if((n+1)<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf(" %2d l %2d %2d %2d l",p[n][45],p[n][46],p[n][47],p[n][48]);
    if((n+1)<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf(" %2d-----%2d-----%2d-----%2d",p[n][61],p[n][62],p[n][63],p[n][64]);
    if((n+1)<x){printf(" ");}}
printf("\n");
}
//
/* Reconstruction Programs *
void recnstrct(){
short n;
for(n=1;n<65;n++){tnm[1][n]=repr[n]; cnm[n]=repr[n];}
tnm[1][0]=1; trans4();
for(n=1;n<65;n++){tnm[2][n]=dnm[n]; cnm[n]=dnm[n];}
tnm[2][0]=2; trans2();
for(n=1;n<65;n++){tnm[3][n]=dnm[n]; cnm[n]=tnm[1][n];}
tnm[3][0]=3; trans1();
for(n=1;n<65;n++){tnm[4][n]=dnm[n]; cnm[n]=dnm[n];}
tnm[4][0]=4; trans4();
for(n=1;n<65;n++){tnm[5][n]=dnm[n]; cnm[n]=dnm[n];}
tnm[5][0]=5; trans2();
for(n=1;n<65;n++){tnm[6][n]=dnm[n]; cnm[n]=tnm[4][n];}
tnm[6][0]=6; trans5();
for(n=1;n<65;n++){tnm[7][n]=dnm[n]; cnm[n]=dnm[n];}
tnm[7][0]=7; trans4();
for(n=1;n<65;n++){tnm[8][n]=dnm[n]; cnm[n]=dnm[n];}
tnm[8][0]=8; trans2();
for(n=1;n<65;n++){tnm[9][n]=dnm[n]; cnm[n]=tnm[7][n];}
tnm[9][0]=9; trans3();

```

```

for(n=1;n<65;n++){tnm[10][n]=dnm[n]; cnm[n]=dnm[n];}
tnm[10][0]=10; trans4();
for(n=1;n<65;n++){tnm[11][n]=dnm[n]; cnm[n]=dnm[n];}
tnm[11][0]=11; trans2();
for(n=1;n<65;n++){tnm[12][n]=dnm[n]; cnm[n]=tnm[1][n];}
tnm[12][0]=12; trans6();
for(n=1;n<65;n++){tnm[13][n]=dnm[n]; cnm[n]=dnm[n];}
tnm[13][0]=13; trans4();
for(n=1;n<65;n++){tnm[14][n]=dnm[n]; cnm[n]=dnm[n];}
tnm[14][0]=14; trans2();
for(n=1;n<65;n++){tnm[15][n]=dnm[n]; cnm[n]=dnm[n];}
tnm[15][0]=15;
}
//
/** Transition Diagram *
//
// [13]-T6-[1]-T1-[4]-T5-[7]-T3-[10]
// | | | | |
// T4 T4 T4 T4 T4
// | | | | |
// [14]-T6-[2]-T5-[5]-T1-[8]-T3-[11]
// | | | | |
// T2 T2 T2 T2 T2
// | | | | |
// [15]-T6-[3]-T5-[6]-T3-[9]-T1-[12]
//
/** Transformation Programs *
void trans1(){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
dnm[3]=cnm[14]; dnm[19]=cnm[30]; dnm[35]=cnm[46]; dnm[51]=cnm[62];
dnm[4]=cnm[13]; dnm[20]=cnm[29]; dnm[36]=cnm[45]; dnm[52]=cnm[61];
dnm[7]=cnm[10]; dnm[23]=cnm[26]; dnm[39]=cnm[42]; dnm[55]=cnm[58];
dnm[8]=cnm[9]; dnm[24]=cnm[25]; dnm[40]=cnm[41]; dnm[56]=cnm[57];
dnm[9]=cnm[8]; dnm[25]=cnm[24]; dnm[41]=cnm[40]; dnm[57]=cnm[56];
dnm[10]=cnm[7]; dnm[26]=cnm[23]; dnm[42]=cnm[39]; dnm[58]=cnm[55];
dnm[13]=cnm[4]; dnm[29]=cnm[20]; dnm[45]=cnm[36]; dnm[61]=cnm[52];
dnm[14]=cnm[3]; dnm[30]=cnm[19]; dnm[46]=cnm[35]; dnm[62]=cnm[51];
}
//
void trans2(){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
dnm[9]=cnm[53]; dnm[10]=cnm[54]; dnm[11]=cnm[55]; dnm[12]=cnm[56];
dnm[13]=cnm[49]; dnm[14]=cnm[50]; dnm[15]=cnm[51]; dnm[16]=cnm[52];
dnm[25]=cnm[37]; dnm[26]=cnm[38]; dnm[27]=cnm[39]; dnm[28]=cnm[40];
dnm[29]=cnm[33]; dnm[30]=cnm[34]; dnm[31]=cnm[35]; dnm[32]=cnm[36];
dnm[33]=cnm[29]; dnm[34]=cnm[30]; dnm[35]=cnm[31]; dnm[36]=cnm[32];
dnm[37]=cnm[25]; dnm[38]=cnm[26]; dnm[39]=cnm[27]; dnm[40]=cnm[28];
dnm[49]=cnm[13]; dnm[50]=cnm[14]; dnm[51]=cnm[15]; dnm[52]=cnm[16];
dnm[53]=cnm[9]; dnm[54]=cnm[10]; dnm[55]=cnm[11]; dnm[56]=cnm[12];
}
//
void trans3(){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
dnm[3]=cnm[50]; dnm[7]=cnm[54]; dnm[11]=cnm[58]; dnm[15]=cnm[62];
dnm[4]=cnm[49]; dnm[8]=cnm[53]; dnm[12]=cnm[57]; dnm[16]=cnm[61];
dnm[19]=cnm[34]; dnm[23]=cnm[38]; dnm[27]=cnm[42]; dnm[31]=cnm[46];
dnm[20]=cnm[33]; dnm[24]=cnm[37]; dnm[28]=cnm[41]; dnm[32]=cnm[45];
dnm[33]=cnm[20]; dnm[37]=cnm[24]; dnm[41]=cnm[28]; dnm[45]=cnm[32];
dnm[34]=cnm[19]; dnm[38]=cnm[23]; dnm[42]=cnm[27]; dnm[46]=cnm[31];
dnm[49]=cnm[4]; dnm[53]=cnm[8]; dnm[57]=cnm[12]; dnm[61]=cnm[16];
dnm[50]=cnm[3]; dnm[54]=cnm[7]; dnm[58]=cnm[11]; dnm[62]=cnm[15];
}
//

```



```

void trans4(){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
dnm[9]=cnm[21]; dnm[10]=cnm[22]; dnm[11]=cnm[23]; dnm[12]=cnm[24];
dnm[13]=cnm[17]; dnm[14]=cnm[18]; dnm[15]=cnm[19]; dnm[16]=cnm[20];
dnm[17]=cnm[13]; dnm[18]=cnm[14]; dnm[19]=cnm[15]; dnm[20]=cnm[16];
dnm[21]=cnm[9]; dnm[22]=cnm[10]; dnm[23]=cnm[11]; dnm[24]=cnm[12];
dnm[33]=cnm[61]; dnm[34]=cnm[62]; dnm[35]=cnm[63]; dnm[36]=cnm[64];
dnm[37]=cnm[57]; dnm[38]=cnm[58]; dnm[39]=cnm[59]; dnm[40]=cnm[60];
dnm[57]=cnm[37]; dnm[58]=cnm[38]; dnm[59]=cnm[39]; dnm[60]=cnm[40];
dnm[61]=cnm[33]; dnm[62]=cnm[34]; dnm[63]=cnm[35]; dnm[64]=cnm[36];
}
//
void trans5(){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
dnm[3]=cnm[18]; dnm[7]=cnm[22]; dnm[11]=cnm[26]; dnm[15]=cnm[30];
dnm[4]=cnm[17]; dnm[8]=cnm[21]; dnm[12]=cnm[25]; dnm[16]=cnm[29];
dnm[17]=cnm[4]; dnm[21]=cnm[8]; dnm[25]=cnm[12]; dnm[29]=cnm[16];
dnm[18]=cnm[3]; dnm[22]=cnm[7]; dnm[26]=cnm[11]; dnm[30]=cnm[15];
dnm[33]=cnm[52]; dnm[37]=cnm[56]; dnm[41]=cnm[60]; dnm[45]=cnm[64];
dnm[34]=cnm[51]; dnm[38]=cnm[55]; dnm[42]=cnm[59]; dnm[46]=cnm[63];
dnm[51]=cnm[34]; dnm[55]=cnm[38]; dnm[59]=cnm[42]; dnm[63]=cnm[46];
dnm[52]=cnm[33]; dnm[56]=cnm[37]; dnm[60]=cnm[41]; dnm[64]=cnm[45];
}
//
void trans6(){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
dnm[3]=cnm[6]; dnm[19]=cnm[22]; dnm[35]=cnm[38]; dnm[51]=cnm[54];
dnm[4]=cnm[5]; dnm[20]=cnm[21]; dnm[36]=cnm[37]; dnm[52]=cnm[53];
dnm[5]=cnm[4]; dnm[21]=cnm[20]; dnm[37]=cnm[36]; dnm[53]=cnm[52];
dnm[6]=cnm[3]; dnm[22]=cnm[19]; dnm[38]=cnm[35]; dnm[54]=cnm[51];
dnm[9]=cnm[16]; dnm[25]=cnm[32]; dnm[41]=cnm[48]; dnm[57]=cnm[64];
dnm[10]=cnm[15]; dnm[26]=cnm[31]; dnm[42]=cnm[47]; dnm[58]=cnm[63];
dnm[15]=cnm[10]; dnm[31]=cnm[26]; dnm[47]=cnm[42]; dnm[63]=cnm[58];
dnm[16]=cnm[9]; dnm[32]=cnm[25]; dnm[48]=cnm[41]; dnm[64]=cnm[57];
}
}

```

#### #8. What does it mean by this success of our reconstruction, then?

Now that we have demonstrated we can really make all the F15 from the only one representative solution No.1, we can say No.1 is just the "Most Fundamental Solution" for all F15, can't we?

But is this solution No.1 the only "Most Fundamental Solution"? No, it isn't.

Anyone of No. 2~No. 15 solutions can also become the "MFS" for all.

Watch the transition diagram above again, and you will know you can start with any solution and can get everyone of No. 1~15 whichever you might start with.

We should say anyone of F15 could be the "MFS" as a representative of all.

What do these 'equal' qualifications of every F15 Solutions mean?

It must be just the inheritance given by the 6-D Extra Cubic Object.

The F15 are 'brothers and sisters', the children of the same Father.

I would say they are the actual 'phenomena' that have come into our real world, and what the original ECO2<sup>6</sup> means must be the paternal Divine Concept.

#### #9. How about the Fundamental 10 Solutions of 'C&C' Magic Squares of Order 8?

Can we also discover the same kind of nature in them again?

F10 of C&C MS88 are taken out of the 6-Dimensional Extra Cubic Magic Objects

of order 2 by dimensional down-conversion. This origin has essentially the only one solution. But, it appears to be with 10 faces in the 2-dimensional MS world.

F10 are all children of the same Origin. They must have some common memory of their "Father", or of "brothers and sisters" for themselves.

I want to find its proof.

First of all let's find and collect any similar pairs of solutions among MF10. We want those pairs which have the same common 32 (half of all) elements.

**\*\* Find Similar Pairs among The Fundamental Ten \*\***  
**\*\* of 'Composite & Complete' Magic Squares 8x8 \*\***

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

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

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

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

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

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

```

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

```

#10. The next job we have to do is to find any transformation methods.

[Transform: T1]

```

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

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

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

```

[Transform: T2]

```

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

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

```

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

[Transform: T3]

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

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

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

[Transform: T4]

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

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

### [Transform: T5]

```

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

### [Transform: T6]

```

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

There are 6 types of possible transformation methods, I found.  
Let me show you part of the program codes for them I dictated as follows.  
The complete list of program is placed at the end of this article.

```
//
/* Transformations *
void trans1(){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
dnm[3]=cnm[10]; dnm[4]=cnm[9]; dnm[5]=cnm[16]; dnm[6]=cnm[15];
dnm[9]=cnm[4]; dnm[10]=cnm[3]; dnm[15]=cnm[6]; dnm[16]=cnm[5];
dnm[17]=cnm[28]; dnm[18]=cnm[27]; dnm[23]=cnm[30]; dnm[24]=cnm[29];
dnm[27]=cnm[18]; dnm[28]=cnm[17]; dnm[29]=cnm[24]; dnm[30]=cnm[23];
dnm[33]=cnm[44]; dnm[34]=cnm[43]; dnm[39]=cnm[46]; dnm[40]=cnm[45];
dnm[43]=cnm[34]; dnm[44]=cnm[33]; dnm[45]=cnm[40]; dnm[46]=cnm[39];
dnm[51]=cnm[58]; dnm[52]=cnm[57]; dnm[53]=cnm[64]; dnm[54]=cnm[63];
dnm[57]=cnm[52]; dnm[58]=cnm[51]; dnm[63]=cnm[54]; dnm[64]=cnm[53];
}
//
void trans2(){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
dnm[5]=cnm[51]; dnm[6]=cnm[52]; dnm[7]=cnm[49]; dnm[8]=cnm[50];
dnm[13]=cnm[59]; dnm[14]=cnm[60]; dnm[15]=cnm[57]; dnm[16]=cnm[58];
dnm[21]=cnm[35]; dnm[22]=cnm[36]; dnm[23]=cnm[33]; dnm[24]=cnm[34];
dnm[29]=cnm[43]; dnm[30]=cnm[44]; dnm[31]=cnm[41]; dnm[32]=cnm[42];
dnm[33]=cnm[23]; dnm[34]=cnm[24]; dnm[35]=cnm[21]; dnm[36]=cnm[22];
dnm[41]=cnm[31]; dnm[42]=cnm[32]; dnm[43]=cnm[29]; dnm[44]=cnm[30];
dnm[49]=cnm[7]; dnm[50]=cnm[8]; dnm[51]=cnm[5]; dnm[52]=cnm[6];
dnm[57]=cnm[15]; dnm[58]=cnm[16]; dnm[59]=cnm[13]; dnm[60]=cnm[14];
}
//
void trans3(void){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
dnm[5]=cnm[63]; dnm[6]=cnm[64]; dnm[7]=cnm[61]; dnm[8]=cnm[62];
dnm[13]=cnm[55]; dnm[14]=cnm[56]; dnm[15]=cnm[53]; dnm[16]=cnm[54];
dnm[17]=cnm[43]; dnm[18]=cnm[44]; dnm[19]=cnm[41]; dnm[20]=cnm[42];
dnm[25]=cnm[35]; dnm[26]=cnm[36]; dnm[27]=cnm[33]; dnm[28]=cnm[34];
dnm[33]=cnm[27]; dnm[34]=cnm[28]; dnm[35]=cnm[25]; dnm[36]=cnm[26];
```

```

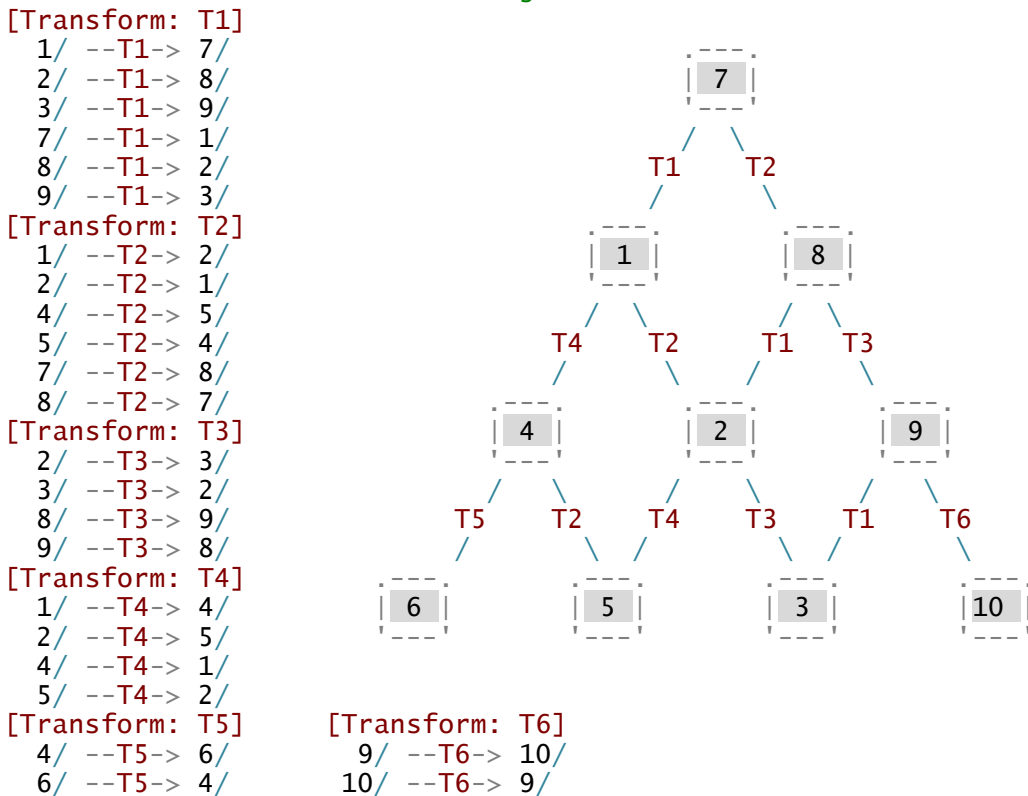
dnm[41]=cnm[19]; dnm[42]=cnm[20]; dnm[43]=cnm[17]; dnm[44]=cnm[18];
dnm[53]=cnm[15]; dnm[54]=cnm[16]; dnm[55]=cnm[13]; dnm[56]=cnm[14];
dnm[61]=cnm[7]; dnm[62]=cnm[8]; dnm[63]=cnm[5]; dnm[64]=cnm[6];
}
. . . . .

```

#11. How do we use these transformation procedures to make the only one into the other 9 Fundamental Solutions?

It is always useful for us to make such a quick-view diagram as follows:

\* Diagram for Process Transition \*



You can start with the solution No.7 at the top of this diagram. Transform it by T1, and you will have No.1. You can also transform No.7 into No.8 by T2.

For the second step you can transform these new results No.1 into No.4 by T4, into No.2 by T2 and another No. 8 into No.9 by T3. ...

Do the same thing one after another. You will finally have all F10 solutions.

And you will know No.7 is really the "Most Fundamental Solution" of all.

\*\*\* Reconstruct the Fundamental Ten Solutions of 'C&C' \*\*\*  
 \*\* Magic Squares of Order 8 by the Inner Transformation \*\*  
 \*\* from the Only One Representative Solution No.7 \*\*  
 \*\* List of the Result with precise Check-Sums \*\*

1#	Rw1+Rw2	C11+C12	Pd1/Pd2	Composite	Fours													
1	63	6	60	38	28	33	31	130+130	130+130	260/260	130	130	130	130	130	130	130	130
62	4	57	7	25	39	30	36	130+130	130+130	260/260	130	130	130	130	130	130	130	130
11	53	16	50	48	18	43	21	130+130	130+130	260/260	130	130	130	130	130	130	130	130
56	10	51	13	19	45	24	42	130+130	130+130	260/260	130	130	130	130	130	130	130	130
27	37	32	34	64	2	59	5	130+130	130+130	260/260	130	130	130	130	130	130	130	130
40	26	35	29	3	61	8	58	130+130	130+130	260/260	130	130	130	130	130	130	130	130
17	47	22	44	54	12	49	15	130+130	130+130	260/260	130	130	130	130	130	130	130	130
46	20	41	23	9	55	14	52	130+130	130+130	260/260	130	130	130	130	130	130	130	130

2# |Rw1+Rw2|C11+C12\Pd1/Pd2|Composite Fours

1	63	6	60	22	44	17	47	130+130	130+130\260/260	130	130	130	130	130	130	130	130
62	4	57	7	41	23	46	20	130+130	130+130\260/260	130	130	130	130	130	130	130	130
11	53	16	50	32	34	27	37	130+130	130+130\260/260	130	130	130	130	130	130	130	130
56	10	51	13	35	29	40	26	130+130	130+130\260/260	130	130	130	130	130	130	130	130
43	21	48	18	64	2	59	5	130+130	130+130\260/260	130	130	130	130	130	130	130	130
24	42	19	45	3	61	8	58	130+130	130+130\260/260	130	130	130	130	130	130	130	130
33	31	38	28	54	12	49	15	130+130	130+130\260/260	130	130	130	130	130	130	130	130
30	36	25	39	9	55	14	52	130+130	130+130\260/260	130	130	130	130	130	130	130	130

3# |Rw1+Rw2|C11+C12\Pd1/Pd2|Composite Fours

1	63	6	60	14	52	9	55	130+130	130+130\260/260	130	130	130	130	130	130	130	130
62	4	57	7	49	15	54	12	130+130	130+130\260/260	130	130	130	130	130	130	130	130
19	45	24	42	32	34	27	37	130+130	130+130\260/260	130	130	130	130	130	130	130	130
48	18	43	21	35	29	40	26	130+130	130+130\260/260	130	130	130	130	130	130	130	130
51	13	56	10	64	2	59	5	130+130	130+130\260/260	130	130	130	130	130	130	130	130
16	50	11	53	3	61	8	58	130+130	130+130\260/260	130	130	130	130	130	130	130	130
33	31	38	28	46	20	41	23	130+130	130+130\260/260	130	130	130	130	130	130	130	130
30	36	25	39	17	47	22	44	130+130	130+130\260/260	130	130	130	130	130	130	130	130

4# |Rw1+Rw2|C11+C12\Pd1/Pd2|Composite Fours

1	63	10	56	42	24	33	31	130+130	130+130\260/260	130	130	130	130	130	130	130	130
62	4	53	11	21	43	30	36	130+130	130+130\260/260	130	130	130	130	130	130	130	130
7	57	16	50	48	18	39	25	130+130	130+130\260/260	130	130	130	130	130	130	130	130
60	6	51	13	19	45	28	38	130+130	130+130\260/260	130	130	130	130	130	130	130	130
23	41	32	34	64	2	55	9	130+130	130+130\260/260	130	130	130	130	130	130	130	130
44	22	35	29	3	61	12	54	130+130	130+130\260/260	130	130	130	130	130	130	130	130
17	47	26	40	58	8	49	15	130+130	130+130\260/260	130	130	130	130	130	130	130	130
46	20	37	27	5	59	14	52	130+130	130+130\260/260	130	130	130	130	130	130	130	130

5# |Rw1+Rw2|C11+C12\Pd1/Pd2|Composite Fours

1	63	10	56	26	40	17	47	130+130	130+130\260/260	130	130	130	130	130	130	130	130
62	4	53	11	37	27	46	20	130+130	130+130\260/260	130	130	130	130	130	130	130	130
7	57	16	50	32	34	23	41	130+130	130+130\260/260	130	130	130	130	130	130	130	130
60	6	51	13	35	29	44	22	130+130	130+130\260/260	130	130	130	130	130	130	130	130
39	25	48	18	64	2	55	9	130+130	130+130\260/260	130	130	130	130	130	130	130	130
28	38	19	45	3	61	12	54	130+130	130+130\260/260	130	130	130	130	130	130	130	130
33	31	42	24	58	8	49	15	130+130	130+130\260/260	130	130	130	130	130	130	130	130
30	36	21	43	5	59	14	52	130+130	130+130\260/260	130	130	130	130	130	130	130	130

6# |Rw1+Rw2|C11+C12\Pd1/Pd2|Composite Fours

1	63	18	48	50	16	33	31	130+130	130+130\260/260	130	130	130	130	130	130	130	130
62	4	45	19	13	51	30	36	130+130	130+130\260/260	130	130	130	130	130	130	130	130
7	57	24	42	56	10	39	25	130+130	130+130\260/260	130	130	130	130	130	130	130	130
60	6	43	21	11	53	28	38	130+130	130+130\260/260	130	130	130	130	130	130	130	130
15	49	32	34	64	2	47	17	130+130	130+130\260/260	130	130	130	130	130	130	130	130
52	14	35	29	3	61	20	46	130+130	130+130\260/260	130	130	130	130	130	130	130	130
9	55	26	40	58	8	41	23	130+130	130+130\260/260	130	130	130	130	130	130	130	130
54	12	37	27	5	59	22	44	130+130	130+130\260/260	130	130	130	130	130	130	130	130

7# |Rw1+Rw2|C11+C12\Pd1/Pd2|Composite Fours

1	63	4	62	36	30	33	31	130+130	130+130\260/260	130	130	130	130	130	130	130	130
60	6	57	7	25	39	28	38	130+130	130+130\260/260	130	130	130	130	130	130	130	130
13	51	16	50	48	18	45	19	130+130	130+130\260/260	130	130	130	130	130	130	130	130
56	10	53	11	21	43	24	42	130+130	130+130\260/260	130	130	130	130	130	130	130	130
29	35	32	34	64	2	61	3	130+130	130+130\260/260	130	130	130	130	130	130	130	130
40	26	37	27	5	59	8	58	130+130	130+130\260/260	130	130	130	130	130	130	130	130
17	47	20	46	52	14	49	15	130+130	130+130\260/260	130	130	130	130	130	130	130	130
44	22	41	23	9	55	12	54	130+130	130+130\260/260	130	130	130	130	130	130	130	130

8# |Rw1+Rw2|C11+C12\Pd1/Pd2|Composite Fours

1	63	4	62	20	46	17	47	130+130	130+130\260/260	130	130	130	130	130	130	130	130
60	6	57	7	41	23	44	22	130+130	130+130\260/260	130	130	130	130	130	130	130	130
13	51	16	50	32	34	29	35	130+130	130+130\260/260	130	130	130	130	130	130	130	130
56	10	53	11	37	27	40	26	130+130	130+130\260/260	130	130	130	130	130	130	130	130
45	19	48	18	64	2	61	3	130+130	130+130\260/260	130	130	130	130	130	130	130	130
24	42	21	43	5	59	8	58	130+130	130+130\260/260	130	130	130	130	130	130	130	130



```

33 31 36 30 52 14 49 15|130+130|130+130\260/260|130 130 130 130 130 130 130 130
28 38 25 39 9 55 12 54|130+130|130+130\260/260|130 130 130 130 130 130 130 130

          9#|Rw1+Rw2|C11+C12\Pd1/Pd2|Composite Fours
  1 63 4 62 12 54 9 55|130+130|130+130\260/260|130 130 130 130 130 130 130 130
60 6 57 7 49 15 52 14|130+130|130+130\260/260|130 130 130 130 130 130 130 130
21 43 24 42 32 34 29 35|130+130|130+130\260/260|130 130 130 130 130 130 130 130
48 18 45 19 37 27 40 26|130+130|130+130\260/260|130 130 130 130 130 130 130 130
53 11 56 10 64 2 61 3|130+130|130+130\260/260|130 130 130 130 130 130 130 130
16 50 13 51 5 59 8 58|130+130|130+130\260/260|130 130 130 130 130 130 130 130
33 31 36 30 44 22 41 23|130+130|130+130\260/260|130 130 130 130 130 130 130 130
28 38 25 39 17 47 20 46|130+130|130+130\260/260|130 130 130 130 130 130 130 130

```

```

          10#|Rw1+Rw2|C11+C12\Pd1/Pd2|Composite Fours
  1 63 4 62 8 58 5 59|130+130|130+130\260/260|130 130 130 130 130 130 130 130
56 10 53 11 49 15 52 14|130+130|130+130\260/260|130 130 130 130 130 130 130 130
25 39 28 38 32 34 29 35|130+130|130+130\260/260|130 130 130 130 130 130 130 130
48 18 45 19 41 23 44 22|130+130|130+130\260/260|130 130 130 130 130 130 130 130
57 7 60 6 64 2 61 3|130+130|130+130\260/260|130 130 130 130 130 130 130 130
16 50 13 51 9 55 12 54|130+130|130+130\260/260|130 130 130 130 130 130 130 130
33 31 36 30 40 26 37 27|130+130|130+130\260/260|130 130 130 130 130 130 130 130
24 42 21 43 17 47 20 46|130+130|130+130\260/260|130 130 130 130 130 130 130 130

```

[Solutions Count = 10] OK!

\*\* Calculated and Listed by Kanji Setsuda on  
Oct.26, 2013 with MacOSX 10.9 & Xcode 5.0.1 \*\*

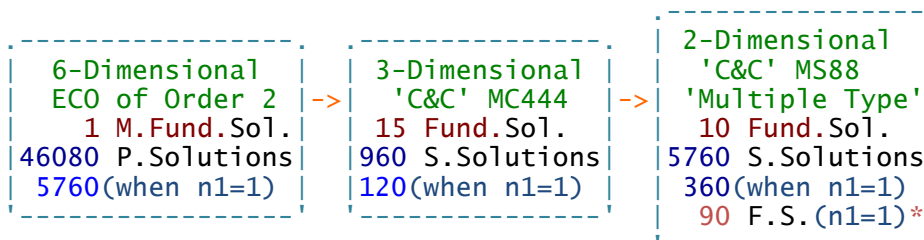
#12. But is this solution No.7 the only one "Most Fundamental Solutions" of all?

No, it isn't. You can nominate any solution of F10 for the "MFS".

Any solution can really become the "MFS" for all the rest.

Please watch the transition diagram again carefully. You can surely start with anyone anywhere. You can get to anyone elsewhere, since every transformation methods are essentially reversible.

We thus found that each F10 has inherited the common memory of its "family". It is the only one family. We could have found nothing else but this one below.



(\* F90 is not always of the 'Multiple Type' but of the 'Special Fundamental Type')

(Original Edition was written in 2000; First Revision in 2002;  
Newest Revision by Kanji Setsuda on October 30, 2013;  
Working with MacOSX 10.8.5 & Xcode 5.0.1)

---

E-Mail Address: jag12001@nifty.com

### 【Program List】

```

/** Reconstruct the Fundamental Ten Solutions of 'C&C' MS88 **
/** by transforming the Only One Representative Solution **
/** File: 'TransCC88F10.c' Built by Kanji Setsuda **
/** in 2002, 2006 and Oct.26, 2013 **
/** with MacOSX 10.9 and Xcode 5.0.1 **

```

```

//
#include <stdio.h>
//
/** Global Variables */
short cnt;
short nm[65], cnm[65], dnm[65];
short tnm[11][65];
short cs[33], csp[17], csm[65];
//
/** Sub-Routiens */
void prans(short x);
void chksums(short x), chksms(short x);
//
/** Transformation System */
void trans1(void), trans2(void), trans3(void);
void trans4(void), trans5(void), trans6(void);
//
/** The Representative Solution Nominated */
short orgnl[65]={7,
    1, 63, 4, 62, 36, 30, 33, 31,
    60, 6, 57, 7, 25, 39, 28, 38,
    13, 51, 16, 50, 48, 18, 45, 19,
    56, 10, 53, 11, 21, 43, 24, 42,
    29, 35, 32, 34, 64, 2, 61, 3,
    40, 26, 37, 27, 5, 59, 8, 58,
    17, 47, 20, 46, 52, 14, 49, 15,
    44, 22, 41, 23, 9, 55, 12, 54};
//
// Transition Diagram:
//
//      [ 7 ]
//      /  \
//     T1   T2
//    /    \
//   [ 1 ]  [ 8 ]
//  /  \   /  \
// T4  T2 T1  T3
// /  \ /  \ /  \
// [ 4 ] [ 2 ] [ 9 ]
// /  \ /  \ /  \
// T5 T2 T4 T3 T1 T6
// /  \ /  \ /  \
// [ 6 ] [ 5 ] [ 3 ] [ 10 ]
//
/** Main Program */
int main(){
short m, n;
printf("\n");
printf("*** Reconstruct the Fundamental Ten Solutions of 'C&C' ***\n");
printf("*** Magic Squares of Order 8 by the Inner Transformation **\n");
printf("*** from the Only One Representative Solution No.7 **\n");
printf("*** List of the Result with precise Check-Sums **\n");
cnt=0;
for(n=0;n<11;n++){tnm[n][0]=n;}
//
cnt++;
for(n=0;n<65;n++){tnm[7][n]=orgnl[n]; cnm[n]=orgnl[n];}
trans2(); cnt++;
for(n=1;n<65;n++){tnm[8][n]=dnm[n]; cnm[n]=orgnl[n];}
trans1(); cnt++;
for(n=1;n<65;n++){tnm[1][n]=dnm[n]; cnm[n]=dnm[n];}

```

```

trans2(); cnt++;
for(n=1;n<65;n++){tnm[2][n]=dnm[n]; cnm[n]=dnm[n];}
trans3(); cnt++;
for(n=1;n<65;n++){tnm[3][n]=dnm[n]; cnm[n]=tnm[1][n];}
trans4(); cnt++;
for(n=1;n<65;n++){tnm[4][n]=dnm[n]; cnm[n]=dnm[n];}
trans2(); cnt++;
for(n=1;n<65;n++){tnm[5][n]=dnm[n]; cnm[n]=tnm[4][n];}
trans5(); cnt++;
for(n=1;n<65;n++){tnm[6][n]=dnm[n]; cnm[n]=tnm[8][n];}
trans3(); cnt++;
for(n=1;n<65;n++){tnm[9][n]=dnm[n]; cnm[n]=dnm[n];}
trans6(); cnt++;
for(n=1;n<65;n++){tnm[10][n]=dnm[n];}
//
for(m=1; m<=cnt; m++){
    for(n=0;n<65;n++){nm[n]=tnm[m][n];}
    chksums(m); chksums(m);
    prans(m);
}
printf("\n");
printf(" [Solutions Count = %d] OK!\n",cnt);
printf("\n");
printf("*** Calculated and Listed by Kanji Setsuda on\n");
printf("    Oct.26, 2013 with MacOSX 10.9 & Xcode 5.0.1 **\n");
printf("\n");
return 0;
}
//
/** Print the Answers *
void prans(short x){
short n,m,m2,m8;
printf("%24d#|Rw1+Rw2|C11+C12\\Pd1/Pd2|Composite Fours",nm[0]);
printf("\n");
for(m=0;m<8;m++){m2=m*2; m8=m*8;
printf(" ");
for(n=1;n<9;n++){printf("%3d",nm[m8+n]);}
printf("|%3d+%3d|%3d+%3d\\%3d/%3d|",
    cs[m2+1],cs[m2+2],cs[m2+17],cs[m2+18],csp[m+1],csp[m+9]);
for(n=1;n<9;n++){
printf("%3d",csm[m8+n]);
if(n<8){printf(" ");}
}
printf("\n");
}
}
//
/** Check the Line-Sums *
void chksums(short x){
cs[1]=nm[1]+nm[2]+nm[3]+nm[4]; // rw11;
cs[2]=nm[5]+nm[6]+nm[7]+nm[8]; // rw12;
cs[3]=nm[9]+nm[10]+nm[11]+nm[12]; // rw21;
cs[4]=nm[13]+nm[14]+nm[15]+nm[16]; // rw22;
cs[5]=nm[17]+nm[18]+nm[19]+nm[20]; // rw31;
cs[6]=nm[21]+nm[22]+nm[23]+nm[24]; // rw32;
cs[7]=nm[25]+nm[26]+nm[27]+nm[28]; // rw41;
cs[8]=nm[29]+nm[30]+nm[31]+nm[32]; // rw42;
cs[9]=nm[33]+nm[34]+nm[35]+nm[36]; // rw51;
cs[10]=nm[37]+nm[38]+nm[39]+nm[40]; // rw52;
cs[11]=nm[41]+nm[42]+nm[43]+nm[44]; // rw61;
cs[12]=nm[45]+nm[46]+nm[47]+nm[48]; // rw62;
cs[13]=nm[49]+nm[50]+nm[51]+nm[52]; // rw71;
cs[14]=nm[53]+nm[54]+nm[55]+nm[56]; // rw72;
cs[15]=nm[57]+nm[58]+nm[59]+nm[60]; // rw81;
cs[16]=nm[61]+nm[62]+nm[63]+nm[64]; // rw82;
//

```

```

cs[17]=nm[1]+nm[9]+nm[17]+nm[25]; // c111;
cs[18]=nm[33]+nm[41]+nm[49]+nm[57]; // c112;
cs[19]=nm[2]+nm[10]+nm[18]+nm[26]; // c121;
cs[20]=nm[34]+nm[42]+nm[50]+nm[58]; // c122;
cs[21]=nm[3]+nm[11]+nm[19]+nm[27]; // c131;
cs[22]=nm[35]+nm[43]+nm[51]+nm[59]; // c132;
cs[23]=nm[4]+nm[12]+nm[20]+nm[28]; // c141;
cs[24]=nm[36]+nm[44]+nm[52]+nm[60]; // c142;
cs[25]=nm[5]+nm[13]+nm[21]+nm[29]; // c151;
cs[26]=nm[37]+nm[45]+nm[53]+nm[61]; // c152;
cs[27]=nm[6]+nm[14]+nm[22]+nm[30]; // c161;
cs[28]=nm[38]+nm[46]+nm[54]+nm[62]; // c162;
cs[29]=nm[7]+nm[15]+nm[23]+nm[31]; // c171;
cs[30]=nm[39]+nm[47]+nm[55]+nm[63]; // c172;
cs[31]=nm[8]+nm[16]+nm[24]+nm[32]; // c181;
cs[32]=nm[40]+nm[48]+nm[56]+nm[64]; // c182;
//
csp[1]=nm[1]+nm[10]+nm[19]+nm[28]+nm[37]+nm[46]+nm[55]+nm[64]; // pd1;
csp[2]=nm[2]+nm[11]+nm[20]+nm[29]+nm[38]+nm[47]+nm[56]+nm[65]; // pd2;
csp[3]=nm[3]+nm[12]+nm[21]+nm[30]+nm[39]+nm[48]+nm[49]+nm[58]; // pd3;
csp[4]=nm[4]+nm[13]+nm[22]+nm[31]+nm[40]+nm[41]+nm[50]+nm[59]; // pd4;
csp[5]=nm[5]+nm[14]+nm[23]+nm[32]+nm[33]+nm[42]+nm[51]+nm[60]; // pd5;
csp[6]=nm[6]+nm[15]+nm[24]+nm[25]+nm[34]+nm[43]+nm[52]+nm[61]; // pd6;
csp[7]=nm[7]+nm[16]+nm[17]+nm[26]+nm[35]+nm[44]+nm[53]+nm[62]; // pd7;
csp[8]=nm[8]+nm[9]+nm[18]+nm[27]+nm[36]+nm[45]+nm[54]+nm[63]; // pd8;
//
csp[9]=nm[1]+nm[16]+nm[23]+nm[30]+nm[37]+nm[44]+nm[51]+nm[58]; // pb1;
csp[10]=nm[2]+nm[9]+nm[24]+nm[31]+nm[38]+nm[45]+nm[52]+nm[59]; // pb2;
csp[11]=nm[3]+nm[10]+nm[17]+nm[32]+nm[39]+nm[46]+nm[53]+nm[60]; // pb3;
csp[12]=nm[4]+nm[11]+nm[18]+nm[25]+nm[40]+nm[47]+nm[54]+nm[61]; // pb4;
csp[13]=nm[5]+nm[12]+nm[19]+nm[26]+nm[33]+nm[48]+nm[55]+nm[62]; // pb5;
csp[14]=nm[6]+nm[13]+nm[20]+nm[27]+nm[34]+nm[41]+nm[56]+nm[63]; // pb6;
csp[15]=nm[7]+nm[14]+nm[21]+nm[28]+nm[35]+nm[42]+nm[49]+nm[64]; // pb7;
csp[16]=nm[8]+nm[15]+nm[22]+nm[29]+nm[36]+nm[43]+nm[50]+nm[57]; // pb8;
}
//
/** Check Sums of Composite Fours *
void chksms(short x){
csm[1]=nm[1]+nm[2]+nm[9]+nm[10];
csm[2]=nm[2]+nm[3]+nm[10]+nm[11];
csm[3]=nm[3]+nm[4]+nm[11]+nm[12];
csm[4]=nm[4]+nm[5]+nm[12]+nm[13];
csm[5]=nm[5]+nm[6]+nm[13]+nm[14];
csm[6]=nm[6]+nm[7]+nm[14]+nm[15];
csm[7]=nm[7]+nm[8]+nm[15]+nm[16];
csm[8]=nm[8]+nm[1]+nm[16]+nm[9];
csm[9]=nm[9]+nm[10]+nm[17]+nm[18];
csm[10]=nm[10]+nm[11]+nm[18]+nm[19];
csm[11]=nm[11]+nm[12]+nm[19]+nm[20];
csm[12]=nm[12]+nm[13]+nm[20]+nm[21];
csm[13]=nm[13]+nm[14]+nm[21]+nm[22];
csm[14]=nm[14]+nm[15]+nm[22]+nm[23];
csm[15]=nm[15]+nm[16]+nm[23]+nm[24];
csm[16]=nm[16]+nm[9]+nm[24]+nm[17];
csm[17]=nm[17]+nm[18]+nm[25]+nm[26];
csm[18]=nm[18]+nm[19]+nm[26]+nm[27];
csm[19]=nm[19]+nm[20]+nm[27]+nm[28];
csm[20]=nm[20]+nm[21]+nm[28]+nm[29];
csm[21]=nm[21]+nm[22]+nm[29]+nm[30];
csm[22]=nm[22]+nm[23]+nm[30]+nm[31];
csm[23]=nm[23]+nm[24]+nm[31]+nm[32];
csm[24]=nm[24]+nm[17]+nm[32]+nm[25];
csm[25]=nm[25]+nm[26]+nm[33]+nm[34];
csm[26]=nm[26]+nm[27]+nm[34]+nm[35];
csm[27]=nm[27]+nm[28]+nm[35]+nm[36];
csm[28]=nm[28]+nm[29]+nm[36]+nm[37];

```

```

csm[29]=nm[29]+nm[30]+nm[37]+nm[38];
csm[30]=nm[30]+nm[31]+nm[38]+nm[39];
csm[31]=nm[31]+nm[32]+nm[39]+nm[40];
csm[32]=nm[32]+nm[25]+nm[40]+nm[33];
csm[33]=nm[33]+nm[34]+nm[41]+nm[42];
csm[34]=nm[34]+nm[35]+nm[42]+nm[43];
csm[35]=nm[35]+nm[36]+nm[43]+nm[44];
csm[36]=nm[36]+nm[37]+nm[44]+nm[45];
csm[37]=nm[37]+nm[38]+nm[45]+nm[46];
csm[38]=nm[38]+nm[39]+nm[46]+nm[47];
csm[39]=nm[39]+nm[40]+nm[47]+nm[48];
csm[40]=nm[40]+nm[33]+nm[48]+nm[41];
csm[41]=nm[41]+nm[42]+nm[49]+nm[50];
csm[42]=nm[42]+nm[43]+nm[50]+nm[51];
csm[43]=nm[43]+nm[44]+nm[51]+nm[52];
csm[44]=nm[44]+nm[45]+nm[52]+nm[53];
csm[45]=nm[45]+nm[46]+nm[53]+nm[54];
csm[46]=nm[46]+nm[47]+nm[54]+nm[55];
csm[47]=nm[47]+nm[48]+nm[55]+nm[56];
csm[48]=nm[48]+nm[41]+nm[56]+nm[49];
csm[49]=nm[49]+nm[50]+nm[57]+nm[58];
csm[50]=nm[50]+nm[51]+nm[58]+nm[59];
csm[51]=nm[51]+nm[52]+nm[59]+nm[60];
csm[52]=nm[52]+nm[53]+nm[60]+nm[61];
csm[53]=nm[53]+nm[54]+nm[61]+nm[62];
csm[54]=nm[54]+nm[55]+nm[62]+nm[63];
csm[55]=nm[55]+nm[56]+nm[63]+nm[64];
csm[56]=nm[56]+nm[49]+nm[64]+nm[57];
csm[57]=nm[57]+nm[58]+nm[1]+nm[2];
csm[58]=nm[58]+nm[59]+nm[2]+nm[3];
csm[59]=nm[59]+nm[60]+nm[3]+nm[4];
csm[60]=nm[60]+nm[61]+nm[4]+nm[5];
csm[61]=nm[61]+nm[62]+nm[5]+nm[6];
csm[62]=nm[62]+nm[63]+nm[6]+nm[7];
csm[63]=nm[63]+nm[64]+nm[7]+nm[8];
csm[64]=nm[64]+nm[57]+nm[8]+nm[1];
}
//
/** Transformations: Type 1~6 *
void trans1(){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
dnm[3]=cnm[10]; dnm[4]=cnm[9]; dnm[5]=cnm[16]; dnm[6]=cnm[15];
dnm[9]=cnm[4]; dnm[10]=cnm[3]; dnm[15]=cnm[6]; dnm[16]=cnm[5];
dnm[17]=cnm[28]; dnm[18]=cnm[27]; dnm[23]=cnm[30]; dnm[24]=cnm[29];
dnm[27]=cnm[18]; dnm[28]=cnm[17]; dnm[29]=cnm[24]; dnm[30]=cnm[23];
dnm[33]=cnm[44]; dnm[34]=cnm[43]; dnm[39]=cnm[46]; dnm[40]=cnm[45];
dnm[43]=cnm[34]; dnm[44]=cnm[33]; dnm[45]=cnm[40]; dnm[46]=cnm[39];
dnm[51]=cnm[58]; dnm[52]=cnm[57]; dnm[53]=cnm[64]; dnm[54]=cnm[63];
dnm[57]=cnm[52]; dnm[58]=cnm[51]; dnm[63]=cnm[54]; dnm[64]=cnm[53];
}
//
void trans2(){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
dnm[5]=cnm[51]; dnm[6]=cnm[52]; dnm[7]=cnm[49]; dnm[8]=cnm[50];
dnm[13]=cnm[59]; dnm[14]=cnm[60]; dnm[15]=cnm[57]; dnm[16]=cnm[58];
dnm[21]=cnm[35]; dnm[22]=cnm[36]; dnm[23]=cnm[33]; dnm[24]=cnm[34];
dnm[29]=cnm[43]; dnm[30]=cnm[44]; dnm[31]=cnm[41]; dnm[32]=cnm[42];
dnm[33]=cnm[23]; dnm[34]=cnm[24]; dnm[35]=cnm[21]; dnm[36]=cnm[22];
dnm[41]=cnm[31]; dnm[42]=cnm[32]; dnm[43]=cnm[29]; dnm[44]=cnm[30];
dnm[49]=cnm[7]; dnm[50]=cnm[8]; dnm[51]=cnm[5]; dnm[52]=cnm[6];
dnm[57]=cnm[15]; dnm[58]=cnm[16]; dnm[59]=cnm[13]; dnm[60]=cnm[14];
}
//
void trans3(void){

```

```

short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
dnm[5]=cnm[63]; dnm[6]=cnm[64]; dnm[7]=cnm[61]; dnm[8]=cnm[62];
dnm[13]=cnm[55]; dnm[14]=cnm[56]; dnm[15]=cnm[53]; dnm[16]=cnm[54];
dnm[17]=cnm[43]; dnm[18]=cnm[44]; dnm[19]=cnm[41]; dnm[20]=cnm[42];
dnm[25]=cnm[35]; dnm[26]=cnm[36]; dnm[27]=cnm[33]; dnm[28]=cnm[34];
dnm[33]=cnm[27]; dnm[34]=cnm[28]; dnm[35]=cnm[25]; dnm[36]=cnm[26];
dnm[41]=cnm[19]; dnm[42]=cnm[20]; dnm[43]=cnm[17]; dnm[44]=cnm[18];
dnm[53]=cnm[15]; dnm[54]=cnm[16]; dnm[55]=cnm[13]; dnm[56]=cnm[14];
dnm[61]=cnm[7]; dnm[62]=cnm[8]; dnm[63]=cnm[5]; dnm[64]=cnm[6];
}
//
void trans4(void){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
dnm[3]=cnm[26]; dnm[4]=cnm[25]; dnm[5]=cnm[32]; dnm[6]=cnm[31];
dnm[11]=cnm[18]; dnm[12]=cnm[17]; dnm[13]=cnm[24]; dnm[14]=cnm[23];
dnm[17]=cnm[12]; dnm[18]=cnm[11]; dnm[23]=cnm[14]; dnm[24]=cnm[13];
dnm[25]=cnm[4]; dnm[26]=cnm[3]; dnm[31]=cnm[6]; dnm[32]=cnm[5];
dnm[33]=cnm[60]; dnm[34]=cnm[59]; dnm[39]=cnm[62]; dnm[40]=cnm[61];
dnm[41]=cnm[52]; dnm[42]=cnm[51]; dnm[47]=cnm[54]; dnm[48]=cnm[53];
dnm[51]=cnm[42]; dnm[52]=cnm[41]; dnm[53]=cnm[48]; dnm[54]=cnm[47];
dnm[59]=cnm[34]; dnm[60]=cnm[33]; dnm[61]=cnm[40]; dnm[62]=cnm[39];
}
//
void trans5(void){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
dnm[3]=cnm[22]; dnm[4]=cnm[21]; dnm[5]=cnm[20]; dnm[6]=cnm[19];
dnm[11]=cnm[30]; dnm[12]=cnm[29]; dnm[13]=cnm[28]; dnm[14]=cnm[27];
dnm[19]=cnm[6]; dnm[20]=cnm[5]; dnm[21]=cnm[4]; dnm[22]=cnm[3];
dnm[27]=cnm[14]; dnm[28]=cnm[13]; dnm[29]=cnm[12]; dnm[30]=cnm[11];
dnm[33]=cnm[56]; dnm[34]=cnm[55]; dnm[39]=cnm[50]; dnm[40]=cnm[49];
dnm[41]=cnm[64]; dnm[42]=cnm[63]; dnm[47]=cnm[58]; dnm[48]=cnm[57];
dnm[49]=cnm[40]; dnm[50]=cnm[39]; dnm[55]=cnm[34]; dnm[56]=cnm[33];
dnm[57]=cnm[48]; dnm[58]=cnm[47]; dnm[63]=cnm[42]; dnm[64]=cnm[41];
}
//
void trans6(void){
short n;
for(n=0;n<65;n++){dnm[n]=cnm[n];}
dnm[5]=cnm[47]; dnm[6]=cnm[48]; dnm[7]=cnm[45]; dnm[8]=cnm[46];
dnm[9]=cnm[35]; dnm[10]=cnm[36]; dnm[11]=cnm[33]; dnm[12]=cnm[34];
dnm[17]=cnm[59]; dnm[18]=cnm[60]; dnm[19]=cnm[57]; dnm[20]=cnm[58];
dnm[29]=cnm[55]; dnm[30]=cnm[56]; dnm[31]=cnm[53]; dnm[32]=cnm[54];
dnm[33]=cnm[11]; dnm[34]=cnm[12]; dnm[35]=cnm[9]; dnm[36]=cnm[10];
dnm[45]=cnm[7]; dnm[46]=cnm[8]; dnm[47]=cnm[5]; dnm[48]=cnm[6];
dnm[53]=cnm[31]; dnm[54]=cnm[32]; dnm[55]=cnm[29]; dnm[56]=cnm[30];
dnm[57]=cnm[19]; dnm[58]=cnm[20]; dnm[59]=cnm[17]; dnm[60]=cnm[18];
}
//

```