Part 4: "New Advanced Study of Magic Squares and Cubes"

Chapter 7: New Method of Composing High-Dimensional Extra-Cubic Objects and their Developed Forms: **Kanji Setsuda**

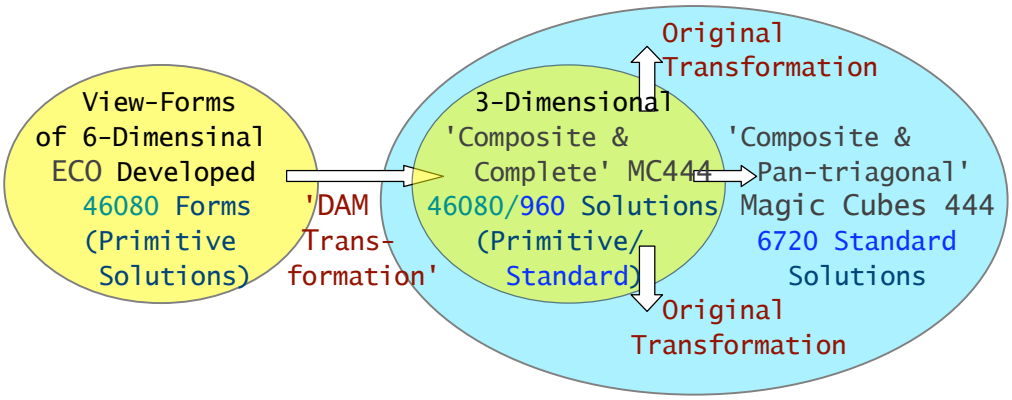
Section 6-2: New Method of Making 6720 'C&P' Magic Cubes 4^3 with All View-Forms of ECO2^6 and Original Transformation System

#1. Our New Object in this Section

Let's compose another common types of magic cubes and squares of order 4 and 8, with all possible 'view-forms' of six dimensional Extra-Cubic Objects developed and our 'DAM Transformation' and with some new additional transformation systems.

I would like to report about these newest methods and some actual results of their applications right here in this section and in another.

Let's make the 6720 'Composite & Pan-triagonal' Magic Cubes of Order 4 here.

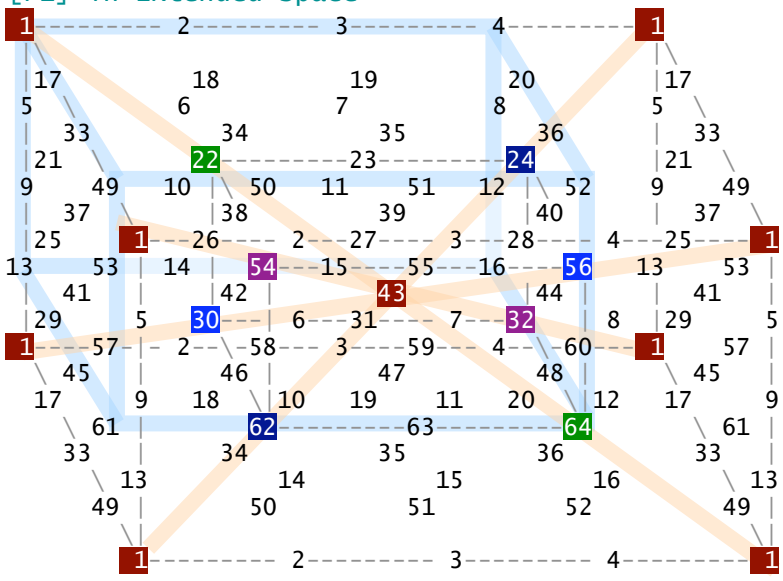


#2. What is the Difference between 'C&C' MC444 and 'C&P' MC444?

What are the 'Composite & Pan-triagonal' Magic Cubes of Order 4, then? It is known they have the larger set of 6720 standard solutions.

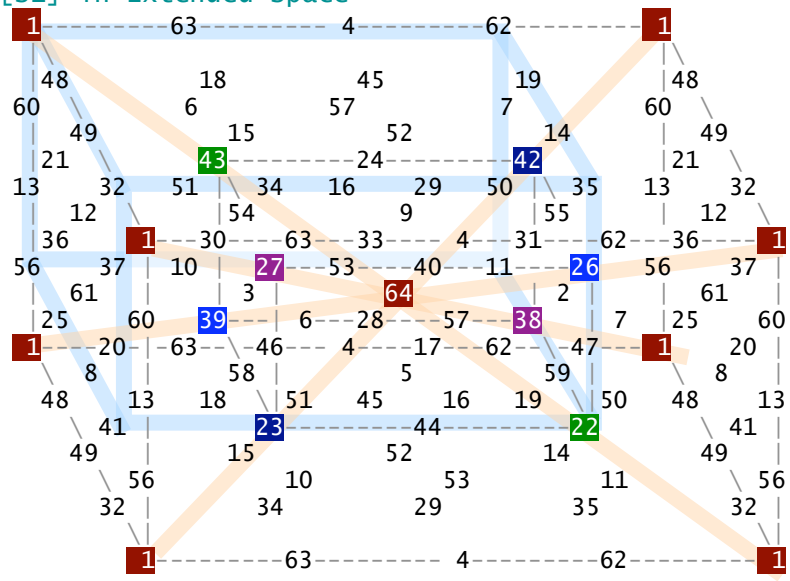
What do they look like? Which part is different from 'Composite & Complete' type?

```
** Conceptual Diagram for Pantriagonals of Pan-Magic Cubes 4x4x4 **
 [P1] in Extended Space
```



```
n1+n22+n43+n64=K  ...pt1-1;     n1+n24+n43+n62=K  ...pt1-2;
n1+n30+n43+n56=K  ...pt1-3;     n1+n32+n43+n54=K  ...pt1-4;
```

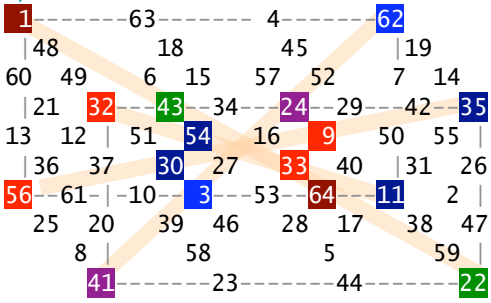
** Diagram for Four Pan-triagonals of PMC444 **
[S1] in Extended Space



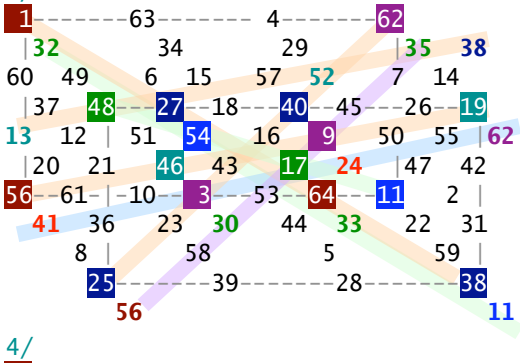
1+43+64+22=K ...pt1-1; 1+42+64+23=K ...pt1-2;
1+39+64+26=K ...pt1-3; 1+38+64+27=K ...pt1-4;

** List of Example Solutions: 'Composite & Complete' MC444 **

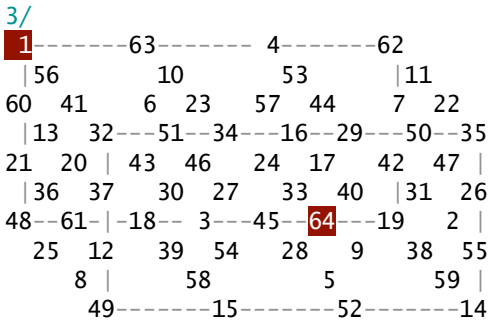
1/



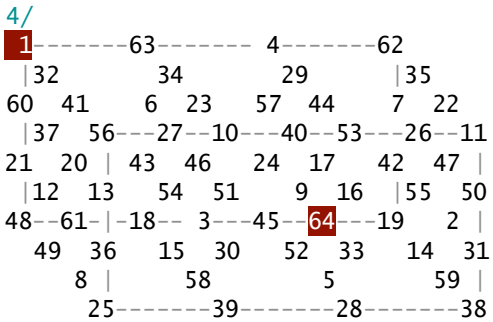
2/



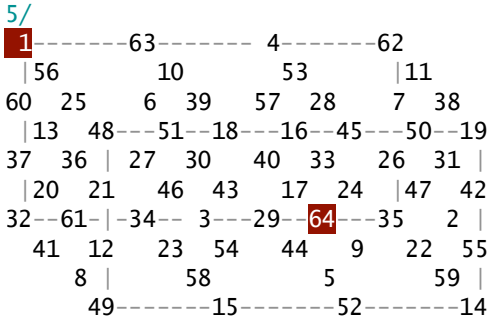
3/



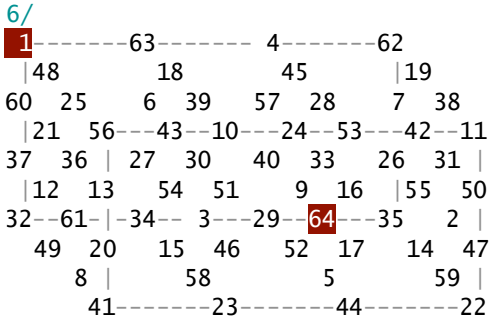
4/



5/



6/



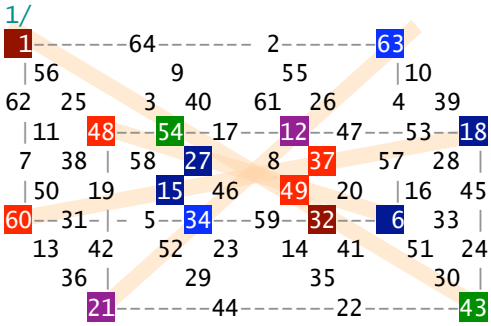
. . . .

They differ from each other mainly in the combination of Complementary Pairs on every pantriagonal: Each 'C&C' MC444 has only the 'Complementary Pairs of 65' on all pantriagonals, while all 'C&P' MC444 do not always obey such a strict rule as this but

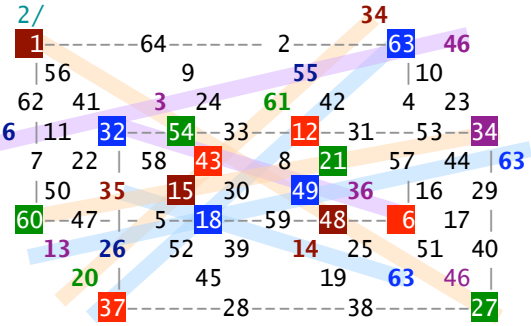
only keep the Constant Sum of all pantriagonals. Watch both N1 on the left top and N43 near the center in any example solution above and below, and you know whenever N1 takes the value 1, N43 should always take 64 in any 'C&C' MC444, but in any 'C&P' N43 does not always take 64 constantly.

** List of Example Solutions: 'Composite & Pan-triagonal' MC444 **

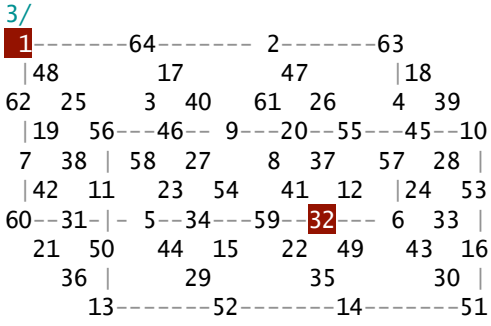
1/



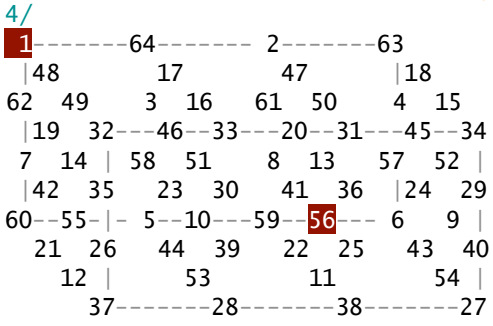
2/



3/

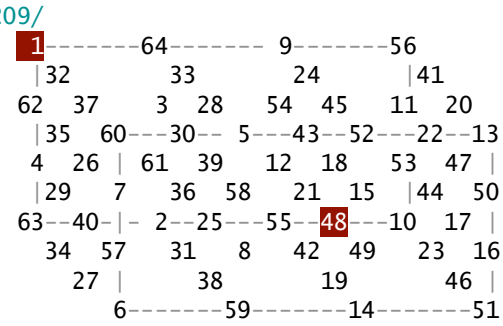


4/

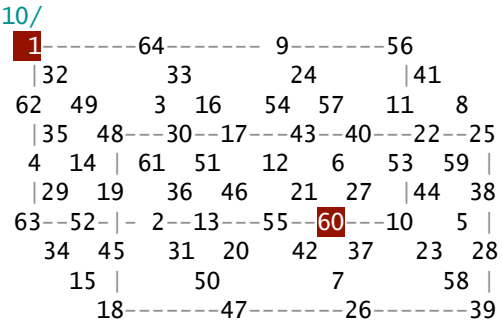


. . . .

209/

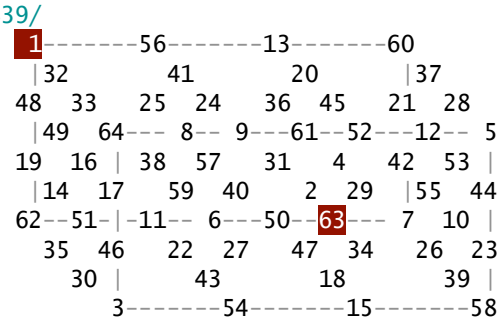


210/

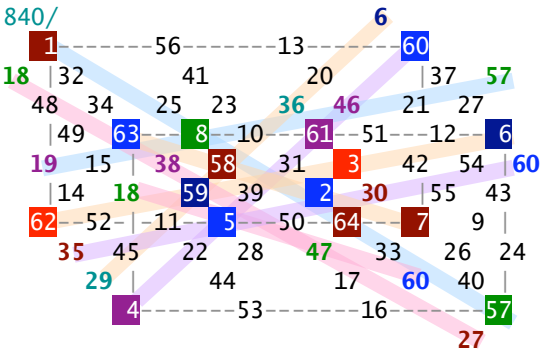


. . . .

839/



840/



Of course, just as the last example 840/, any 'C&C' MC444 is always included in the solution set of 'C&P' MC444.

The solution set of 960 'C&C' certainly makes the sub-set of the one of 6720 'C&P'.

The count 6720 of standard solutions of 'C&P' is 7 times as many as 960 of 'C&C'.

The ratio 7 comes from the necessity of real combinations of Complementary Pairs of pan-magic cubes. It is the fact whenever N1 takes 1, N43 should always take each of seven values {32, 48, 56, 60, 62, 63 or 64} and would take none of other values.

```
** Solution Counts according to the Values of N43, When N1==1: **
( 1:   0) ( 2:   0) ( 3:   0) ( 4:   0) ( 5:   0) ( 6:   0) ( 7:   0) ( 8:   0)
( 9:   0) (10:   0) (11:   0) (12:   0) (13:   0) (14:   0) (15:   0) (16:   0)
(17:   0) (18:   0) (19:   0) (20:   0) (21:   0) (22:   0) (23:   0) (24:   0)
(25:   0) (26:   0) (27:   0) (28:   0) (29:   0) (30:   0) (31:   0) (32: 120)
(33:   0) (34:   0) (35:   0) (36:   0) (37:   0) (38:   0) (39:   0) (40:   0)
(41:   0) (42:   0) (43:   0) (44:   0) (45:   0) (46:   0) (47:   0) (48: 120)
(49:   0) (50:   0) (51:   0) (52:   0) (53:   0) (54:   0) (55:   0) (56: 120)
(57:   0) (58:   0) (59:   0) (60: 120) (61:   0) (62: 120) (63: 120) (64: 120)
 Sum = 840
```

You may possibly be interested in this fact, but let me skip explaining about it here. If you want to know the secret reason for that, go to Part 1, Section 4 in this web site and find to read several old articles of mine.

#3. How to compose those 6720 'C&P' Magic Cubes 444

I want to make them from the solution set of 'C&C' MC444 and transform each into 'C&P' MC444 directly.

We already know how to compose the 960 'C&C' MC444 by our newest method using all possible view-forms of 6-dimensional ECO developed and 'Do-it-After-the-Model Transformation', just in the way I explained about in the previous section.

I want to know how to make these solutions into so many counts of 'C&P' solutions as 6720, seven times as many as 960 of 'C&C'.

The problem is how to find such a good transformation system as to get the result.

I wanted to avoid making myself develop any independent 960 ways of transformation for each solution, but I wanted to have got a single transformation system to deal with them totally. I have tried and tried to find such a kind of magic for a long time, but always failed in vain. I had to know any single transformation system could not afford to reach the goal.

I decided to change my tactics. How about taking the combination of several possible ways to get the result, then? How about making so many answers at first to check and choose the true ones out of them afterward? How about trying to find some realistic methods instead of looking for the idealistic only one.

I examined my new idea by making various experiments as hard as I could.

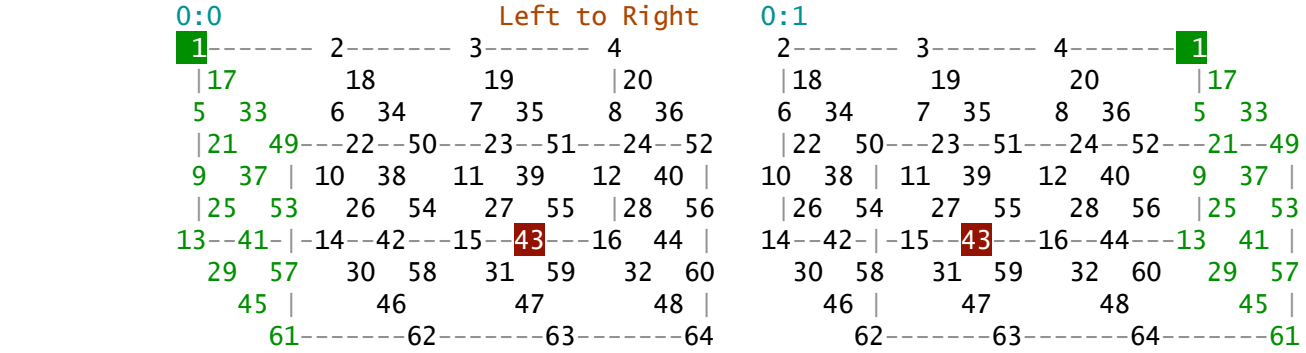
I have finally found how to combine several possible transformation methods and have really got the complete solution set of 6720 'C&P' Magic Cubes 444.

#4. About the Way using Unique Pan-magic Plane Shifting

Before all, let me explain about the unique method of transformation that repeats shifting each plane of pan-magic cubes. It is regarded as the most essential system of transformation. It can produce 4x4x4 different forms from the only origin.

The next three sets of diagrams express the concept of pan-magic plane shifting.

Three types of Plane Shifting: Left to Right, Top to Bottom and Back to Front



0:2

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

0:3

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

1:0 Top to Bottom

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

1:1

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

1:2

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

1:3

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

2:0 Back to Front

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

2:1

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

2:2

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

2:3

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

Any of the solutions transformed by this method still remain a pantriagonal magic cube. It should not break any Composite Conditions at all, either.

It is certified by the very conditions of Constant Sum that every 4 numbers on all the 64 pantriagonals must strictly add up to.

But I have to say it is impossible for us to use this method here for our object.

This way of pan-magic plane shifting can never change any 'C&C' MC444 into 'C&P'. The solution set of 'C&C' is 'closed' with respect to this operation and its result would never go out of it. It is because this operation could not change any combination of 4 numbers on any 4 pantriagonals into another, and N1 should always take N43 as its 'partner' and would not change its partnership there at all.

#5. About Our Newest Transformation System

We have to find another kind of transformation system. Let me introduce you such a combination of two groups of three transformation methods as listed below.

*** Conceptual Diagrams for Transformation Methods to make into 'C&P' MC444 ***

T0/Original

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

T1/

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

T2/

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

T3/

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

T4/

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

T5/

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

T6/

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

T0/Original Again

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

Diagram T0/ illustrates the concept of original solution, and those T1/~T6/ express how to make their transformations actually.

Each of T1/~T3/ expresses the same kind of transformation in three directions.

T4/~T6/ also express another kind of transformation in the same way.

All of them keep half unchanged and change the rest half of the number elements. Watch them carefully and know how 32 don't move and how the rest 32 move.

Seven positions colored red in T0/ diagram are the critical points of 'C&P' MC444.

Watch the position exchange carefully in T1/~T6/, especially about N43, and know how to exchange the two positions for N43 and its pair half.

After taking those 6 transformations, each value of {N4, N13, N49, N2, N5, N17} could come to appear at the N43 position. Thus N43 could take each of 7 values {32, 48, 56, 60, 62, 63, 64} in itself.

I have found and formed those 6 transformation methods from the solution set of our object goal 'C&P' MC444, by comparing and analyzing them carefully.

Studying the real object solutions is the only way to know the effective methods.

Let me demonstrate a sample process of making the transformation Type 1/.

```
** Making Transformation Method T1 to change 'C&C' into 'C&P' MC444 **
  1/Solution                                2/Solution
   1-------64------- 2-------63              1-------64------- 2-------63
    |56        9        55       |10         |56        9        55       |10
  62  25     3  40    61  26     4  39       62  41     3  24    61  42     4  23
    |11  48---54--17---12--47---53--18        |11  32---54--33---12--31---53--34
    7  38 | 58  27     8  37    57  28 |       7  22 | 58  43     8  21    57  44 |
    |50  19    15  46    49  20  |16  45       |50  35    15  30    49  36  |16  29
  60--31-|- 5--34---59--32--- 6  33 |       60--47-|- 5--18---59--48--- 6  17 |
     13  42    52  23    14  41    51  24        13  26    52  39    14  25    51  40
       36 |      29        35       30 |          20 |      45        19       46 |
         21-------44-------22-------43            37-------28-------38-------27
  1/Basic Form                              2/Basic Form Transformed
   1------- 2------- 3------- 4              1------- 2------- 3------- 4
    |17        18        19       |20         |17        18        19       |20
    5  33     6  34     7  35     8  36        5  59     6  60     7  57     8  58
    |21  49---22--50---23--51---24--52        |21  43---22--44---23--41---24--42
    9  37 | 10  38    11  39    12  40 |       9  63 | 10  64    11  61    12  62 |
    |25  53    26  54    27  55  |28  56       |25  47    26  48    27  45  |28  46
  13--41-|-14--42---15--43---16  44 |       13--51-|-14--52---15--49---16  50 |
    29  57    30  58    31  59    32  60        29  35    30  36    31  33    32  34
       45 |      46        47       48 |          55 |      56        53       54 |
         61-------62-------63-------64            39-------40-------37-------38
/**/
/* Dictated Program for the Transformation #1 */
void convrt1(short x){
 short n;
 short p[65],q[65];
  for(n=1;n<65;n++){q[n]=trn[0][n];}
  p[1]=q[1];    p[2]=q[2];    p[3]=q[3];    p[4]=q[4];
  p[5]=q[5];    p[6]=q[6];    p[7]=q[7];    p[8]=q[8];
  p[9]=q[9];    p[10]=q[10];  p[11]=q[11];  p[12]=q[12];
  p[13]=q[13];  p[14]=q[14];  p[15]=q[15];  p[16]=q[16];
  p[17]=q[17];  p[18]=q[18];  p[19]=q[19];  p[20]=q[20];
  p[21]=q[21];  p[22]=q[22];  p[23]=q[23];  p[24]=q[24];
  p[25]=q[25];  p[26]=q[26];  p[27]=q[27];  p[28]=q[28];
  p[29]=q[29];  p[30]=q[30];  p[31]=q[31];  p[32]=q[32];
  p[33]=q[59];  p[34]=q[60];  p[35]=q[57];  p[36]=q[58];
  p[37]=q[63];  p[38]=q[64];  p[39]=q[61];  p[40]=q[62];
  p[41]=q[51];  p[42]=q[52];  p[43]=q[49];  p[44]=q[50];
  p[45]=q[55];  p[46]=q[56];  p[47]=q[53];  p[48]=q[54];
```

```
  p[49]=q[43];  p[50]=q[44];  p[51]=q[41];  p[52]=q[42];
  p[53]=q[47];  p[54]=q[48];  p[55]=q[45];  p[56]=q[46];
  p[57]=q[35];  p[58]=q[36];  p[59]=q[33];  p[60]=q[34];
  p[61]=q[39];  p[62]=q[40];  p[63]=q[37];  p[64]=q[38];
  for(n=1;n<65;n++){trn[x][n]=p[n];}
}
/**/
```

You have to dictate the other 5 programs in the same way to make the complete transformation system.

#6. Job List: What are we going to do step by step?

(1) Make the developed diagrams of 6-dimensional Extra-Cubic Object of Order 2 as many as possible, just in the way as explained before in the previous section.

We have to have so many diagrams as 46080 for the 'primitive solutions'.

We now regard these diagrams as 'Prototype Cubes of Order 4', and are going to apply 'DAM Transformation' to each Prototype Cube.

The 'Model Solution' must be prepared by selecting out of the 'C&C' MC444.

(2) Make the 46080 primitive solutions of 'C&C' MC444 by applying 'DAMT' to the 46080 Prototype Cubes. Save all the results to our table to re-use afterward.

It is usual we select the 960 'standard solutions' by those 'List-forming Inequality Conditions', but now we prefer not to use these 960 here.

I fear those inequality conditions may sometimes conflict with any transformation process and make serious troubles. And actually 960 solutions are not enough.

```
/* Definitions of Basic Diagrams for ECO2^6 */
/* BasicForm/2^6
   1---- 2      3---- 4      33----34     35----36
   | 17--+-18   | 19--+-20    | 49--+-50   | 51--+-52
   5--|- 6  |   7--|- 8  |   37--|-38  |  39--|-40  |
     21----22     23----24      53----54     55----56

   9----10     11----12      41----42     43----44
   | 25--+-26  | 27--+-28    | 57--+-58   | 59--+-60
  13--|-14  |  15--|-16  |   45--|-46  |  47--|-48  |
     29----30     31----32      61----62     63----64

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

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

```
/* Dictated Program for the 'DAM Transformation' making 'C&C' MC444 *
void damtrnsf(){
 long int m;
 short p[65],q[65];
 short n;
  for(m=0;m<cntr;m++){
    for(n=0;n<65;n++){p[n]=snm[m][n];}
    q[0]=m+1;
    q[1]=p[1];    q[2]=p[63];   q[3]=p[4];    q[4]=p[62];
    q[5]=p[60];   q[6]=p[6];    q[7]=p[57];   q[8]=p[7];
    q[9]=p[13];   q[10]=p[51];  q[11]=p[16];  q[12]=p[50];
    q[13]=p[56];  q[14]=p[10];  q[15]=p[53];  q[16]=p[11];
    q[17]=p[48];  q[18]=p[18];  q[19]=p[45];  q[20]=p[19];
    q[21]=p[21];  q[22]=p[43];  q[23]=p[24];  q[24]=p[42];
    q[25]=p[36];  q[26]=p[30];  q[27]=p[33];  q[28]=p[31];
    q[29]=p[25];  q[30]=p[39];  q[31]=p[28];  q[32]=p[38];
    q[33]=p[49];  q[34]=p[15];  q[35]=p[52];  q[36]=p[14];
    q[37]=p[12];  q[38]=p[54];  q[39]=p[9];   q[40]=p[55];
    q[41]=p[61];  q[42]=p[3];   q[43]=p[64];  q[44]=p[2];
    q[45]=p[8];   q[46]=p[58];  q[47]=p[5];   q[48]=p[59];
    q[49]=p[32];  q[50]=p[34];  q[51]=p[29];  q[52]=p[35];
    q[53]=p[37];  q[54]=p[27];  q[55]=p[40];  q[56]=p[26];
    q[57]=p[20];  q[58]=p[46];  q[59]=p[17];  q[60]=p[47];
    q[61]=p[41];  q[62]=p[23];  q[63]=p[44];  q[64]=p[22];
    for(n=0;n<65;n++){tnm[m][n]=q[n];}
  }
}
*/
```

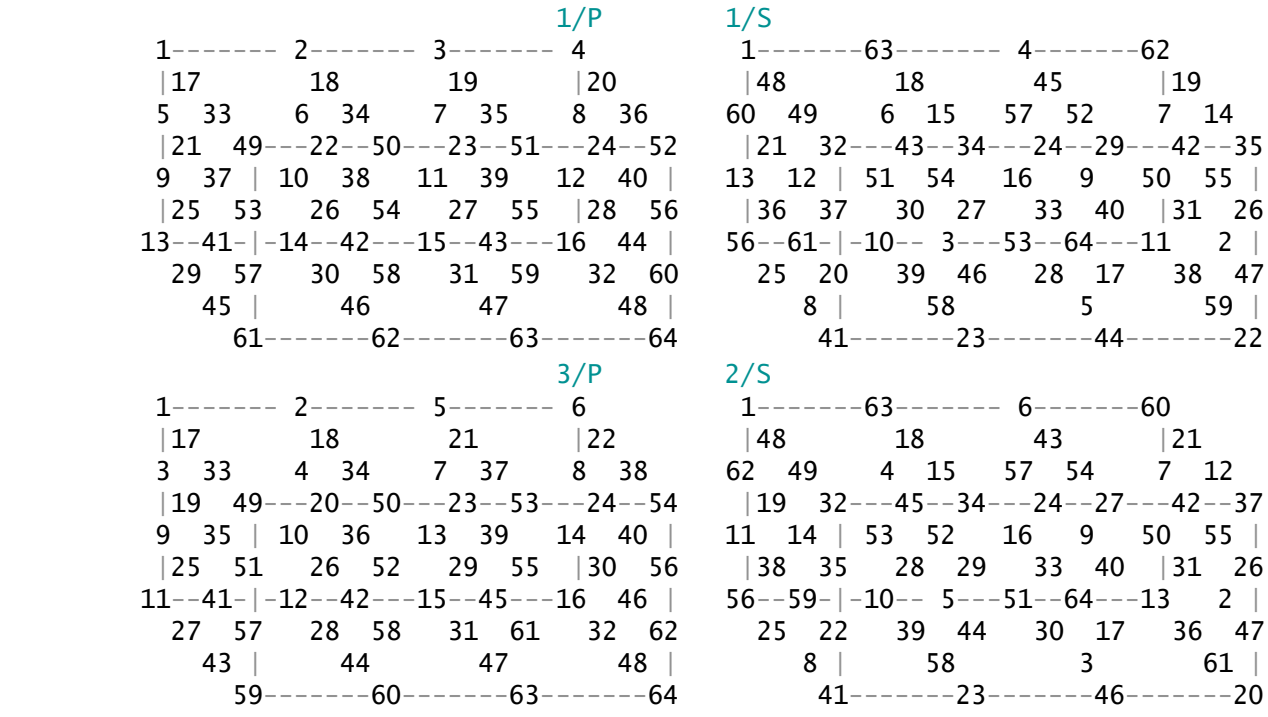
(3) We take this solution above for our 'Model' as we like and define the 'Do-it-After-the-Model Transformation' as listed there.

(4) We have to have the 46080 primitive solutions of 'C&C' type by this process, using the single 'DAMT' and 46080 primitive Prototype Cubes.

Apply the 'DAMT' to all Prototype Cubes and make them into the 'C&C' MC444 so as to mediate for our goal object 'C&P' MC444.

Let me illustrate this step right here by showing some sample pairs on the way.

** Process Samples: Prototype Cubes and 'C&C' MC444 Transformed **



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

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

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

```
                        4/S
 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
```

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

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

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

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

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

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

The last 5 pieces of 'C&C' are essentially the same solutions with the first one 1/. They are only the reflected patterns around the primary triagonal {1, 43, 64, 22}.

Among the primitive solutions, there are such kinds of 48 derivative forms for each original one: 6 types of 'Umbrella Reflections' and 8 types of 'Ball Rotations'.

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

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

We are going to do nothing about it but save all primitive answers to our table.

It is the last job for us to select the standard solutions, to be done in the end.

(5) We use our newest transformation system combining 6 methods to make those 'C&C' into our last goal 'C&P' MC444. We will have finally got the smart list of 6720 objects, under the list-forming inequality conditions.

#7. Program List

Let me show you the list of compact program I wrote for this experiment.

The first half of this program is very similar to, say, the same as the one I put in the previous section. I really added the last half to it, yes, my transformation process.

```
/** Compose 'Composite & Pan-triagonal' Magic Cubes 4^3 by our New Method  **/
/** 'NewCPMC4TT.c' Dictated by Kanji Setsuda **/
/** on Apr.26, '05; Aug.16, '06; Jan.10, '09 **/
/**  Working with MacOSX 10.5 and Xcode 3.0  **/
/**/
#include <stdio.h>
/**/
/* Global Variables */
long int cntr, cnt2;
/**/
```

```
short vc, pc;
short tv[65][6];
short td[721][6];
short tnm[46081][65], snm[46081][65];
short strn[6721][65];
short trn[8][65];
short cn1[65], cn43[65];
/**/
/* List of Sub-Procedures for our New Method */
void mkfv26(void), prmt26(void);
void d26(short x, short y);
void mksol26(void);
/* Programs for List Printing */
void pr2cpmc4(long x, short y);
/* Sorting and Transformation */
void srt43(long x), exc43(long x);
void cntsol(short x);
void prcn1(long x, short y);
void prcn43(long x, short y);
void trnsfmtn(void);
void convrt1(short x), convrt2(short x);
void convrt3(short x), convrt4(short x);
void convrt5(short x), convrt6(short x);
/**/
/* MAIN PROGRAM */
int main(){
 long int m,m1;
 short n,m2;
  printf("\n/**/\n");
  printf("/** RESULT REPORT **/\n");
  printf("/**\n");
  printf("** Make Standard type of 'Composite & Complete' Magic Cubes 444 by our **\n");
  printf("**  New Method Using All Possible View-Forms of Developed Extra-Cubic  **\n");
  printf("**  Objects of Order 2^6 and 'Do-it-After-the-Model' Transformation, **\n");
  printf("** And Transform 'C&C' type into 'Composite & Pan-triagonal' MC444 **\n");
  for(n=0;n<65;n++){cn1[n]=0;}
  vc=0; mkfv26();
  pc=0; prmt26();
  cntr=vc*pc;
  for(m=0;m<vc;m++){for(n=0;n<pc;n++){d26(m,n);}}
  mksol26();
  printf("\n** Solution Counts according to the Values of N1: **\n");
  prcn1(cntr,64);
/**/
  cnt2=cntr/2;
  cntr=0;
  for(n=0;n<65;n++){cn1[n]=0; cn43[n]=0;}
  printf("\n** Transforming ...\n");
  printf("  by the Special Transformation System **\n");
  printf(" ... Monitoring Total Solution Counts in Every Step:\n");
  for(m=0;m<=6720;m++){strn[m][0]=0; strn[m][1]=0;}
  for(m=0;m<cnt2;m++){
    for(n=1;n<65;n++){trn[0][n]=tnm[m][n];}
    trnsfmtn();
    m1=m+1;
    if(m1%720==0){m2=m1/720;
      printf("%4d:%4d,",m2,cntr);
      if(m2%8==0){printf("\n");}}
  }
  printf("\n ... Sorting Solution Data for the smart List. Wait for a while, please.\n");
  srt43(cntr);
  printf("\n** Standard Solutions of 'Composite & Pan-triagonal' Magic Cubes 4^3 **\n");
  pr2cpmc4(1,24); printf(" . . . .\n");
  pr2cpmc4(cn1[1]/4-1,6); printf(" . . . .\n");
  pr2cpmc4(cn1[1]-1,6); printf(" . . . .\n");
  pr2cpmc4((cn1[1]+cn1[2])-1,6); printf(" . . . .\n");
```

```
   pr2cpmc4((cn1[1]+cn1[2]+cn1[3])-1,6); printf(" . . . .\n");
   pr2cpmc4(cntr-7,8);
   printf(" [Total Count = %d]\n",cntr);
   printf("\n** Solution Counts according to the Values of N1: **\n");
   prcn1(cntr,32);
   printf("** Solution Counts according to the Values of N43, When N1==1: **\n");
   prcn43(cn1[1],64);
   printf("\n [OK!]\n");
 return 0;
}
/**/
/* New Method of Composing 'C&C' MC444 */
/* Sub-Procedures for Extra-Cubic Objects of Order 2^6 */
/**/
void mkfv26(){
 short d0,d1,d2,d3,d4,d5;
  for(d0=0;d0<2;d0++){
   for(d1=0;d1<2;d1++){
    for(d2=0;d2<2;d2++){
     for(d3=0;d3<2;d3++){
      for(d4=0;d4<2;d4++){
       for(d5=0;d5<2;d5++){
        tv[vc][0]=d0; tv[vc][1]=d1; tv[vc][2]=d2;
        tv[vc][3]=d3; tv[vc][4]=d4; tv[vc][5]=d5;
        vc++;
  }}}}}}
}
/**/
void prmt26(void){
   short d0,d1,d2,d3,d4,d5,n;
   short uflg[6];
     for(n=0;n<6;n++){uflg[n]=0;}
     for(d0=0;d0<6;d0++){
      uflg[d0]=1;
       for(d1=0;d1<6;d1++){
        if(uflg[d1]==0){uflg[d1]=1;
         for(d2=0;d2<6;d2++){
          if(uflg[d2]==0){uflg[d2]=1;
           for(d3=0;d3<6;d3++){
            if(uflg[d3]==0){uflg[d3]=1;
             for(d4=0;d4<6;d4++){
              if(uflg[d4]==0){uflg[d4]=1;
               for(d5=0;d5<6;d5++){
                if(uflg[d5]==0){uflg[d5]=1;
                 td[pc][0]=d0; td[pc][1]=d1; td[pc][2]=d2;
                 td[pc][3]=d3; td[pc][4]=d4; td[pc][5]=d5;
                 pc++;
                 uflg[d5]=0;
               }}
               uflg[d4]=0;
             }}
             uflg[d3]=0;
           }}
           uflg[d2]=0;
         }}
         uflg[d1]=0;
       }}
       uflg[d0]=0;
     }
}
/**/
void d26(short x, short y){
 short d0,d1,d2,d3,d4,d5;
 short t0,t1,t2,t3,t4,t5;
 short c;
 short s[6], cd[2][2];
```

```
  cd[0][0]=0; cd[0][1]=1; cd[1][0]=1; cd[1][1]=0;
  c=0;
  for(d0=0;d0<2;d0++){
   for(d1=0;d1<2;d1++){
    for(d2=0;d2<2;d2++){
     for(d3=0;d3<2;d3++){
      for(d4=0;d4<2;d4++){
       for(d5=0;d5<2;d5++){c++;
        s[0]=cd[tv[x][0]][d0]; s[1]=cd[tv[x][1]][d1]; s[2]=cd[tv[x][2]][d2];
        s[3]=cd[tv[x][3]][d3]; s[4]=cd[tv[x][4]][d4]; s[5]=cd[tv[x][5]][d5];
        t0=td[y][0]; t1=td[y][1]; t2=td[y][2]; t3=td[y][3]; t4=td[y][4]; t5=td[y][5];
        snm[x*pc+y][c]=((((s[t0]*2+s[t1])*2+s[t2])*2+s[t3])*2+s[t4])*2+s[t5]+1;
      }}
    }}
  }}
}
/**/
/* Transform each Basic Diagram of 2^6 into 'C&C' MC4^3 Solution */
void mksol26(){
 long int m,mc;
 short p[65],q[65];
 short n,q1;
  for(m=0;m<cntr;m++){
    for(n=0;n<65;n++){p[n]=snm[m][n];}
    q[0]=m+1;
    q[1]=p[1];    q[2]=p[63];   q[3]=p[4];    q[4]=p[62];
    q[5]=p[60];   q[6]=p[6];    q[7]=p[57];   q[8]=p[7];
    q[9]=p[13];   q[10]=p[51];  q[11]=p[16];  q[12]=p[50];
    q[13]=p[56];  q[14]=p[10];  q[15]=p[53];  q[16]=p[11];
    q[17]=p[48];  q[18]=p[18];  q[19]=p[45];  q[20]=p[19];
    q[21]=p[21];  q[22]=p[43];  q[23]=p[24];  q[24]=p[42];
    q[25]=p[36];  q[26]=p[30];  q[27]=p[33];  q[28]=p[31];
    q[29]=p[25];  q[30]=p[39];  q[31]=p[28];  q[32]=p[38];
    q[33]=p[49];  q[34]=p[15];  q[35]=p[52];  q[36]=p[14];
    q[37]=p[12];  q[38]=p[54];  q[39]=p[9];   q[40]=p[55];
    q[41]=p[61];  q[42]=p[3];   q[43]=p[64];  q[44]=p[2];
    q[45]=p[8];   q[46]=p[58];  q[47]=p[5];   q[48]=p[59];
    q[49]=p[32];  q[50]=p[34];  q[51]=p[29];  q[52]=p[35];
    q[53]=p[37];  q[54]=p[27];  q[55]=p[40];  q[56]=p[26];
    q[57]=p[20];  q[58]=p[46];  q[59]=p[17];  q[60]=p[47];
    q[61]=p[41];  q[62]=p[23];  q[63]=p[44];  q[64]=p[22];
    q1=q[1]; mc=(q1-1)*720+cn1[q1];
    for(n=0;n<65;n++){tnm[mc][n]=q[n];}
    cn1[q1]++;
  }
}
/**/
/* Print the Two Answers of 'C&C' MC4^3 */
void pr2cpmc4(long x, short y){
 long m;
 short n;
 short p[65],q[65];
  for(m=x-1;m<(x+y-1);m=m+2){
    for(n=0;n<65;n++){p[n]=strn[m][n]; q[n]=strn[m+1][n];}
    if(m<99){printf("%3d/%37d/\n",m+1,m+2);}
     else if((98<m)&&(m<999)){printf("%4d/%37d/\n",m+1,m+2);}
     else{printf("%5d/%37d/\n",m+1,m+2);}
    printf("  %2d-------%2d-------%2d-------%2d       ",p[1],p[2],p[3],p[4]);
    printf("  %2d-------%2d-------%2d-------%2d\n",q[1],q[2],q[3],q[4]);
    printf("    |%2d       %2d       %2d       |%2d     ",p[17],p[18],p[19],p[20]);
    printf("    |%2d       %2d       %2d       |%2d\n",q[17],q[18],q[19],q[20]);
    printf("  %2d  %2d   %2d  %2d   %2d  %2d   %2d  %2d   ",
      p[5],p[33],p[6],p[34],p[7],p[35],p[8],p[36]);
    printf("  %2d  %2d   %2d  %2d   %2d  %2d   %2d  %2d\n",
      q[5],q[33],q[6],q[34],q[7],q[35],q[8],q[36]);
    printf("    |%2d  %2d---%2d--%2d---%2d--%2d---%2d--%2d ",
```

```
       p[21],p[49],p[22],p[50],p[23],p[51],p[24],p[52]);
     printf("    |%2d  %2d--%2d--%2d--%2d--%2d--%2d--%2d\n",
       q[21],q[49],q[22],q[50],q[23],q[51],q[24],q[52]);
     printf("  %2d  %2d | %2d  %2d   %2d  %2d   %2d  %2d | ",
       p[9],p[37],p[10],p[38],p[11],p[39],p[12],p[40]);
     printf("  %2d  %2d | %2d  %2d   %2d  %2d   %2d  %2d |\n",
       q[9],q[37],q[10],q[38],q[11],q[39],q[12],q[40]);
     printf("    |%2d  %2d   %2d  %2d   %2d  %2d  |%2d  %2d ",
       p[25],p[53],p[26],p[54],p[27],p[55],p[28],p[56]);
     printf("    |%2d  %2d   %2d  %2d   %2d  %2d  |%2d  %2d\n",
       q[25],q[53],q[26],q[54],q[27],q[55],q[28],q[56]);
     printf("  %2d--%2d-|-%2d--%2d---%2d--%2d---%2d  %2d | ",
       p[13],p[41],p[14],p[42],p[15],p[43],p[16],p[44]);
     printf("  %2d--%2d-|-%2d--%2d---%2d--%2d---%2d  %2d |\n",
       q[13],q[41],q[14],q[42],q[15],q[43],q[16],q[44]);
     printf("     %2d  %2d   %2d  %2d   %2d  %2d   %2d  %2d ",
       p[29],p[57],p[30],p[58],p[31],p[59],p[32],p[60]);
     printf("     %2d  %2d   %2d  %2d   %2d  %2d   %2d  %2d\n",
       q[29],q[57],q[30],q[58],q[31],q[59],q[32],q[60]);
     printf("       %2d |     %2d       %2d       %2d | ",p[45],p[46],p[47],p[48]);
     printf("       %2d |     %2d       %2d       %2d |\n",q[45],q[46],q[47],q[48]);
     printf("         %2d-------%2d-------%2d-------%2d ",p[61],p[62],p[63],p[64]);
     printf("         %2d-------%2d-------%2d-------%2d\n",q[61],q[62],q[63],q[64]);
   }
}
/**/
/** Solution Counts according to the Values of N1: **/
void prcn1(long x, short y){
 short n;
   for(n=1;n<=y;n++){
     printf(" (%2d:%4d)",n,cn1[n]);
     if(n%8==0){printf("\n");}
   }
   printf(" Sum = %d\n",x);
}
/**/
/** Solution Counts according to the Values of N43: **/
void prcn43(long x, short y){
 short n;
   for(n=1;n<=y;n++){
     printf(" (%2d:%4d)",n,cn43[n]);
     if(n%8==0){printf("\n");}
   }
   printf(" Sum = %d\n",x);
}
/**/
/* Sort the Solution Data for the smart List of 'C&P' Standard Set */
void srt43(long x){
 short f;
 long m,n,d1,d2,d3,d4;
   m=x;
   do{f=0; m--;
     for(n=0;n<m;n++){
      if(strn[n][1]>strn[n+1][1]){exc43(n); f=1;}
      d1=(strn[n][2]*65+strn[n][4])*65+strn[n][5];
      d2=(strn[n+1][2]*65+strn[n+1][4])*65+strn[n+1][5];
      if((strn[n][1]==strn[n+1][1])&&(d1<d2)){exc43(n); f=1;}
      d3=(strn[n][13]*65+strn[n][17])*65+strn[n][49];
      d4=(strn[n+1][13]*65+strn[n+1][17])*65+strn[n+1][49];
      if((strn[n][1]==strn[n+1][1])&&(d1==d2)&&(d3<d4)){exc43(n); f=1;}
     }
    }while(f>0);
}
/**/
/* Exchange solution pairs according to their own values */
void exc43(long x){
```

```
 short p,sv;
  for(p=0;p<65;p++){sv=strn[x][p]; strn[x][p]=strn[x+1][p]; strn[x+1][p]=sv;}
}
/**/
void cntsol(short x){
 long int nn;
 short m,n,p,mtc,c;
  for(m=0;m<=x;m++){
   if((trn[m][1]<trn[m][4])&&(trn[m][1]<trn[m][13])&&(trn[m][1]<trn[m][16])){
    if((trn[m][1]<trn[m][49])&&(trn[m][1]<trn[m][52])&&(trn[m][1]<trn[m][61])){
     if((trn[m][1]<trn[m][64])&&(trn[m][17]<trn[m][5])&&(trn[m][5]<trn[m][2])){
      mtc=0;
      for(nn=0;nn<=cntr;nn++){
        if(strn[nn][1]==0){break;}
        for(p=1;p<65;p++){if(trn[m][p]==strn[nn][p]){mtc++;}else{mtc=0; break;}}
        if(mtc==64){break;}
      }
      if(mtc<64){c=trn[m][1];
        for(n=1;n<65;n++){strn[cntr][n]=trn[m][n];}
        cn1[c]++; if(c==1){cn43[trn[m][43]]++;}
        cntr++;
      }
   }}}
  }
}
/**/
/** Transformation System from 'C&C' type into 'C&P' */
void trnsfmtn(void){
  convrt1(1);
  convrt2(2);
  convrt3(3);
  convrt4(4);
  convrt5(5);
  convrt6(6);
  cntsol(6);
}
/**/
/* Convert it from 'C&C' to 'C&P' */
void convrt1(short x){
 short n;
 short p[65],q[65];
  for(n=1;n<65;n++){q[n]=trn[0][n];}
  p[1]=q[1];    p[2]=q[2];    p[3]=q[3];    p[4]=q[4];
  p[5]=q[5];    p[6]=q[6];    p[7]=q[7];    p[8]=q[8];
  p[9]=q[9];    p[10]=q[10];  p[11]=q[11];  p[12]=q[12];
  p[13]=q[13];  p[14]=q[14];  p[15]=q[15];  p[16]=q[16];
  p[17]=q[17];  p[18]=q[18];  p[19]=q[19];  p[20]=q[20];
  p[21]=q[21];  p[22]=q[22];  p[23]=q[23];  p[24]=q[24];
  p[25]=q[25];  p[26]=q[26];  p[27]=q[27];  p[28]=q[28];
  p[29]=q[29];  p[30]=q[30];  p[31]=q[31];  p[32]=q[32];
  p[33]=q[59];  p[34]=q[60];  p[35]=q[57];  p[36]=q[58];
  p[37]=q[63];  p[38]=q[64];  p[39]=q[61];  p[40]=q[62];
  p[41]=q[51];  p[42]=q[52];  p[43]=q[49];  p[44]=q[50];
  p[45]=q[55];  p[46]=q[56];  p[47]=q[53];  p[48]=q[54];
  p[49]=q[43];  p[50]=q[44];  p[51]=q[41];  p[52]=q[42];
  p[53]=q[47];  p[54]=q[48];  p[55]=q[45];  p[56]=q[46];
  p[57]=q[35];  p[58]=q[36];  p[59]=q[33];  p[60]=q[34];
  p[61]=q[39];  p[62]=q[40];  p[63]=q[37];  p[64]=q[38];
  for(n=1;n<65;n++){trn[x][n]=p[n];}
}
/**/
void convrt2(short x){
 short n;
 short p[65],q[65];
  for(n=1;n<65;n++){q[n]=trn[0][n];}
  p[1]=q[1];    p[2]=q[2];    p[3]=q[3];    p[4]=q[4];
```

```
   p[5]=q[5];    p[6]=q[6];    p[7]=q[7];    p[8]=q[8];
   p[9]=q[47];   p[10]=q[48];  p[11]=q[45];  p[12]=q[46];
   p[13]=q[43];  p[14]=q[44];  p[15]=q[41];  p[16]=q[42];
   p[17]=q[17];  p[18]=q[18];  p[19]=q[19];  p[20]=q[20];
   p[21]=q[21];  p[22]=q[22];  p[23]=q[23];  p[24]=q[24];
   p[25]=q[63];  p[26]=q[64];  p[27]=q[61];  p[28]=q[62];
   p[29]=q[59];  p[30]=q[60];  p[31]=q[57];  p[32]=q[58];
   p[33]=q[33];  p[34]=q[34];  p[35]=q[35];  p[36]=q[36];
   p[37]=q[37];  p[38]=q[38];  p[39]=q[39];  p[40]=q[40];
   p[41]=q[15];  p[42]=q[16];  p[43]=q[13];  p[44]=q[14];
   p[45]=q[11];  p[46]=q[12];  p[47]=q[9];   p[48]=q[10];
   p[49]=q[49];  p[50]=q[50];  p[51]=q[51];  p[52]=q[52];
   p[53]=q[53];  p[54]=q[54];  p[55]=q[55];  p[56]=q[56];
   p[57]=q[31];  p[58]=q[32];  p[59]=q[29];  p[60]=q[30];
   p[61]=q[27];  p[62]=q[28];  p[63]=q[25];  p[64]=q[26];
   for(n=1;n<65;n++){trn[x][n]=p[n];}
}
/**/
void convrt3(short x){
 short n;
 short p[65],q[65];
   for(n=1;n<65;n++){q[n]=trn[0][n];}
   p[1]=q[1];    p[2]=q[2];    p[3]=q[44];   p[4]=q[43];
   p[5]=q[5];    p[6]=q[6];    p[7]=q[48];   p[8]=q[47];
   p[9]=q[9];    p[10]=q[10];  p[11]=q[36];  p[12]=q[35];
   p[13]=q[13];  p[14]=q[14];  p[15]=q[40];  p[16]=q[39];
   p[17]=q[17];  p[18]=q[18];  p[19]=q[60];  p[20]=q[59];
   p[21]=q[21];  p[22]=q[22];  p[23]=q[64];  p[24]=q[63];
   p[25]=q[25];  p[26]=q[26];  p[27]=q[52];  p[28]=q[51];
   p[29]=q[29];  p[30]=q[30];  p[31]=q[56];  p[32]=q[55];
   p[33]=q[33];  p[34]=q[34];  p[35]=q[12];  p[36]=q[11];
   p[37]=q[37];  p[38]=q[38];  p[39]=q[16];  p[40]=q[15];
   p[41]=q[41];  p[42]=q[42];  p[43]=q[4];   p[44]=q[3];
   p[45]=q[45];  p[46]=q[46];  p[47]=q[8];   p[48]=q[7];
   p[49]=q[49];  p[50]=q[50];  p[51]=q[28];  p[52]=q[27];
   p[53]=q[53];  p[54]=q[54];  p[55]=q[32];  p[56]=q[31];
   p[57]=q[57];  p[58]=q[58];  p[59]=q[20];  p[60]=q[19];
   p[61]=q[61];  p[62]=q[62];  p[63]=q[24];  p[64]=q[23];
   for(n=1;n<65;n++){trn[x][n]=p[n];}
}
/**/
void convrt4(short x){
 short n;
 short p[65],q[65];
   for(n=1;n<65;n++){q[n]=trn[0][n];}
   p[1]=q[1];    p[2]=q[2];    p[3]=q[3];    p[4]=q[4];
   p[5]=q[5];    p[6]=q[6];    p[7]=q[7];    p[8]=q[8];
   p[9]=q[9];    p[10]=q[10];  p[11]=q[11];  p[12]=q[12];
   p[13]=q[13];  p[14]=q[14];  p[15]=q[15];  p[16]=q[16];
   p[17]=q[43];  p[18]=q[44];  p[19]=q[41];  p[20]=q[42];
   p[21]=q[47];  p[22]=q[48];  p[23]=q[45];  p[24]=q[46];
   p[25]=q[35];  p[26]=q[36];  p[27]=q[33];  p[28]=q[34];
   p[29]=q[39];  p[30]=q[40];  p[31]=q[37];  p[32]=q[38];
   p[33]=q[27];  p[34]=q[28];  p[35]=q[25];  p[36]=q[26];
   p[37]=q[31];  p[38]=q[32];  p[39]=q[29];  p[40]=q[30];
   p[41]=q[19];  p[42]=q[20];  p[43]=q[17];  p[44]=q[18];
   p[45]=q[23];  p[46]=q[24];  p[47]=q[21];  p[48]=q[22];
   p[49]=q[49];  p[50]=q[50];  p[51]=q[51];  p[52]=q[52];
   p[53]=q[53];  p[54]=q[54];  p[55]=q[55];  p[56]=q[56];
   p[57]=q[57];  p[58]=q[58];  p[59]=q[59];  p[60]=q[60];
   p[61]=q[61];  p[62]=q[62];  p[63]=q[63];  p[64]=q[64];
   for(n=1;n<65;n++){trn[x][n]=p[n];}
}
/**/
void convrt5(short x){
 short n;
```

```
 short p[65],q[65];
  for(n=1;n<65;n++){q[n]=trn[0][n];}
  p[1]=q[1];    p[2]=q[2];    p[3]=q[3];    p[4]=q[4];
  p[5]=q[43];   p[6]=q[44];   p[7]=q[41];   p[8]=q[42];
  p[9]=q[47];   p[10]=q[48];  p[11]=q[45];  p[12]=q[46];
  p[13]=q[5];   p[14]=q[6];   p[15]=q[7];   p[16]=q[8];
  p[17]=q[13];  p[18]=q[14];  p[19]=q[15];  p[20]=q[16];
  p[21]=q[39];  p[22]=q[40];  p[23]=q[37];  p[24]=q[38];
  p[25]=q[35];  p[26]=q[36];  p[27]=q[33];  p[28]=q[34];
  p[29]=q[9];   p[30]=q[10];  p[31]=q[11];  p[32]=q[12];
  p[33]=q[61];  p[34]=q[62];  p[35]=q[63];  p[36]=q[64];
  p[37]=q[23];  p[38]=q[24];  p[39]=q[21];  p[40]=q[22];
  p[41]=q[19];  p[42]=q[20];  p[43]=q[17];  p[44]=q[18];
  p[45]=q[57];  p[46]=q[58];  p[47]=q[59];  p[48]=q[60];
  p[49]=q[49];  p[50]=q[50];  p[51]=q[51];  p[52]=q[52];
  p[53]=q[27];  p[54]=q[28];  p[55]=q[25];  p[56]=q[26];
  p[57]=q[31];  p[58]=q[32];  p[59]=q[29];  p[60]=q[30];
  p[61]=q[53];  p[62]=q[54];  p[63]=q[55];  p[64]=q[56];
  for(n=1;n<65;n++){trn[x][n]=p[n];}
}
/**/
void convrt6(short x){
 short n;
 short p[65],q[65];
  for(n=1;n<65;n++){q[n]=trn[0][n];}
  p[1]=q[1];    p[2]=q[43];   p[3]=q[42];   p[4]=q[4];
  p[5]=q[5];    p[6]=q[47];   p[7]=q[46];   p[8]=q[8];
  p[9]=q[9];    p[10]=q[35];  p[11]=q[34];  p[12]=q[12];
  p[13]=q[13];  p[14]=q[39];  p[15]=q[38];  p[16]=q[16];
  p[17]=q[17];  p[18]=q[59];  p[19]=q[58];  p[20]=q[20];
  p[21]=q[21];  p[22]=q[63];  p[23]=q[62];  p[24]=q[24];
  p[25]=q[25];  p[26]=q[51];  p[27]=q[50];  p[28]=q[28];
  p[29]=q[29];  p[30]=q[55];  p[31]=q[54];  p[32]=q[32];
  p[33]=q[33];  p[34]=q[11];  p[35]=q[10];  p[36]=q[36];
  p[37]=q[37];  p[38]=q[15];  p[39]=q[14];  p[40]=q[40];
  p[41]=q[41];  p[42]=q[3];   p[43]=q[2];   p[44]=q[44];
  p[45]=q[45];  p[46]=q[7];   p[47]=q[6];   p[48]=q[48];
  p[49]=q[49];  p[50]=q[27];  p[51]=q[26];  p[52]=q[52];
  p[53]=q[53];  p[54]=q[31];  p[55]=q[30];  p[56]=q[56];
  p[57]=q[57];  p[58]=q[19];  p[59]=q[18];  p[60]=q[60];
  p[61]=q[61];  p[62]=q[23];  p[63]=q[22];  p[64]=q[64];
  for(n=1;n<65;n++){trn[x][n]=p[n];}
}
/**/
/* End_Of_File */
```

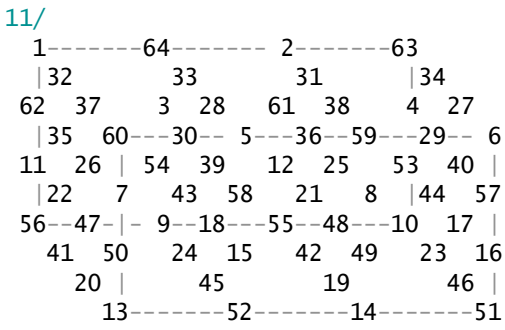
#8. Result Report

You will surely have the same report as follows, if you make the same experiment with the program dictated by me as above.

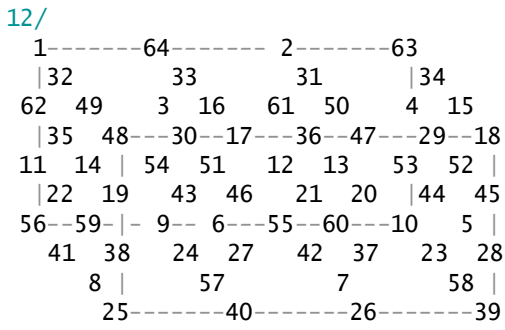
```
** Make Standard type of 'Composite & Complete' Magic Cubes 444 by our **
**  New Method Using All Possible View-Forms of Developed Extra-Cubic  **
**  Objects of Order 2^6 and 'Do-it-After-the-Model' Transformation, **
** And Transform 'C&C' type into 'Composite & Pan-triagonal' MC444 **

** Solution Counts according to the Values of N1: **
 ( 1: 720) ( 2: 720) ( 3: 720) ( 4: 720) ( 5: 720) ( 6: 720) ( 7: 720) ( 8: 720)
 ( 9: 720) (10: 720) (11: 720) (12: 720) (13: 720) (14: 720) (15: 720) (16: 720)
 (17: 720) (18: 720) (19: 720) (20: 720) (21: 720) (22: 720) (23: 720) (24: 720)
 (25: 720) (26: 720) (27: 720) (28: 720) (29: 720) (30: 720) (31: 720) (32: 720)
 (33: 720) (34: 720) (35: 720) (36: 720) (37: 720) (38: 720) (39: 720) (40: 720)
 (41: 720) (42: 720) (43: 720) (44: 720) (45: 720) (46: 720) (47: 720) (48: 720)
 (49: 720) (50: 720) (51: 720) (52: 720) (53: 720) (54: 720) (55: 720) (56: 720)
 (57: 720) (58: 720) (59: 720) (60: 720) (61: 720) (62: 720) (63: 720) (64: 720)
 Sum = 46080
```

```
** Transforming ...
 by the Special Transformation System **
 ... Monitoring Total Solution Counts in Every Step:
   1: 840,   2:1560,   3:2184,   4:2712,   5:3240,   6:3672,   7:4032,   8:4320,
   9:4728,  10:5040,  11:5280,  12:5448,  13:5616,  14:5712,  15:5760,  16:5760,
  17:6000,  18:6168,  19:6288,  20:6360,  21:6456,  22:6504,  23:6528,  24:6528,
  25:6624,  26:6672,  27:6696,  28:6696,  29:6720,  30:6720,  31:6720,  32:6720,

 ... Sorting Solution Data for the smart List. Wait for a while, please.

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

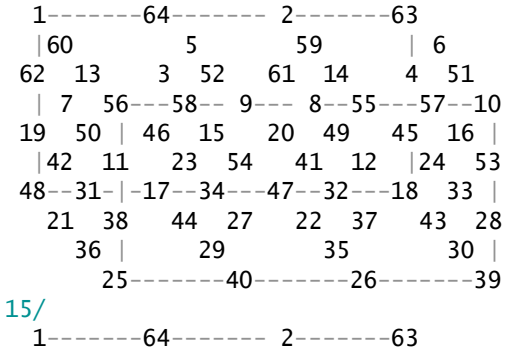
11/



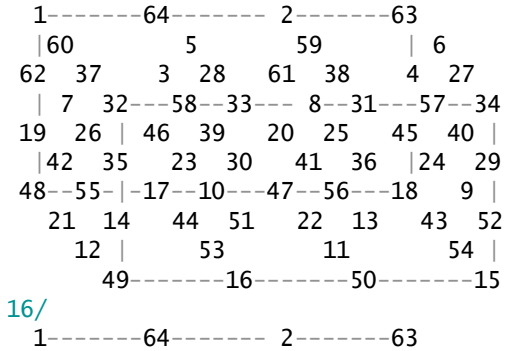
12/



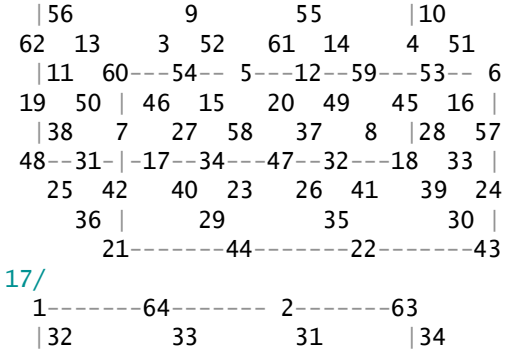
13/



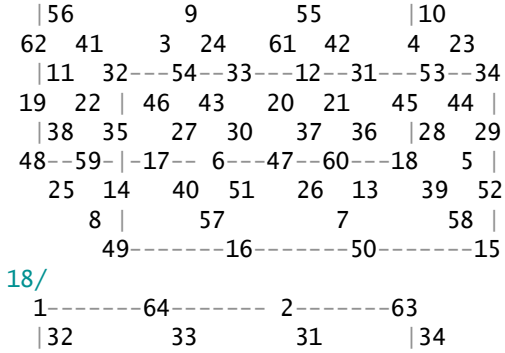
14/



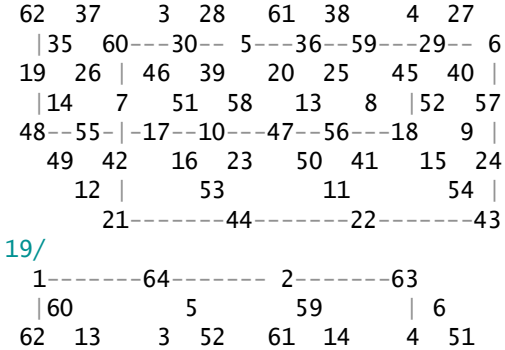
15/



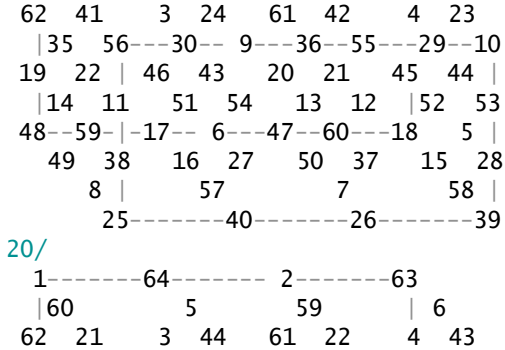
16/



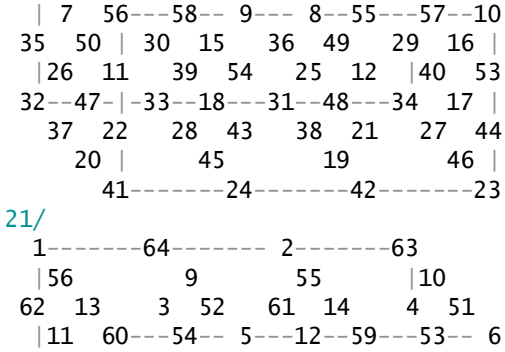
17/



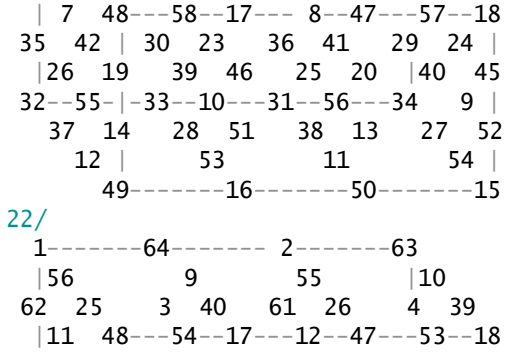
18/



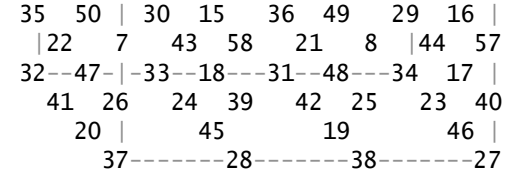
19/



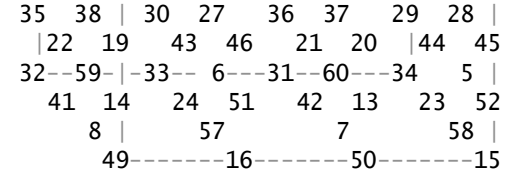
20/



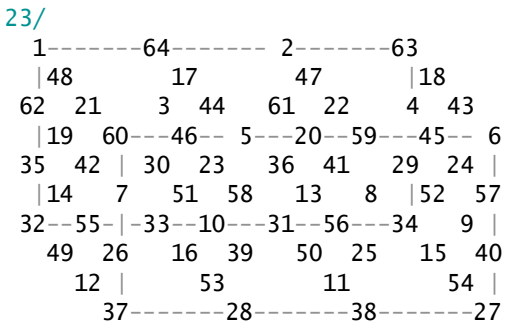
21/



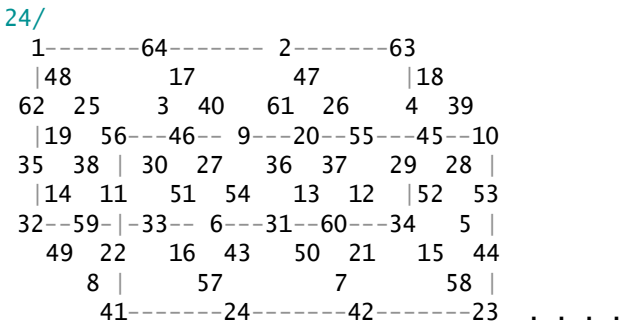
22/



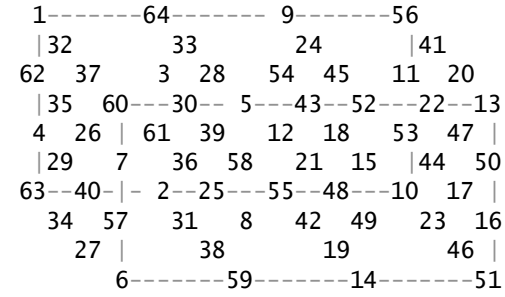
23/



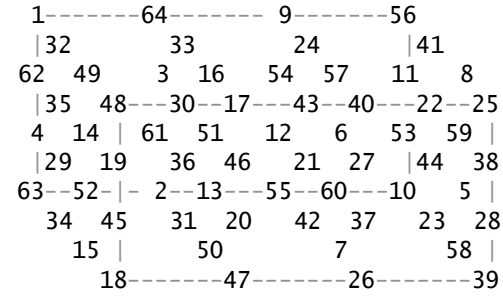
24/



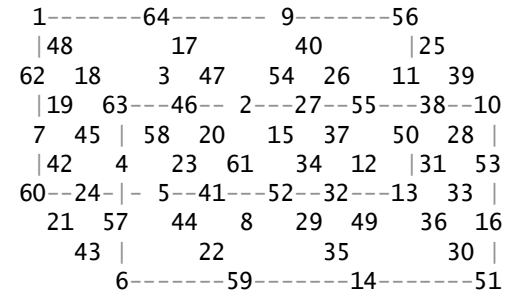
209/



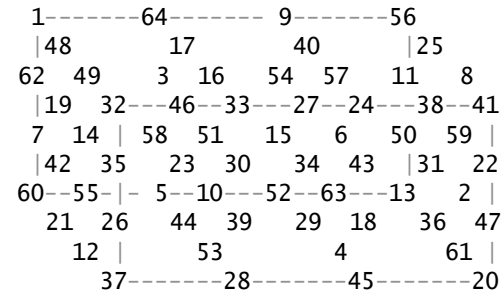
210/



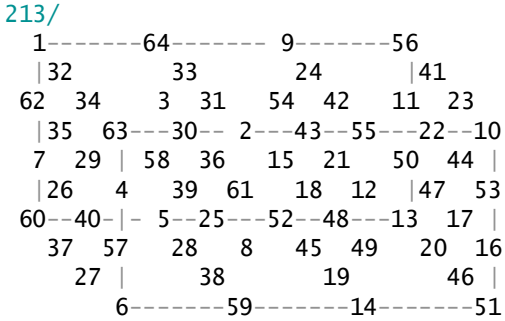
211/



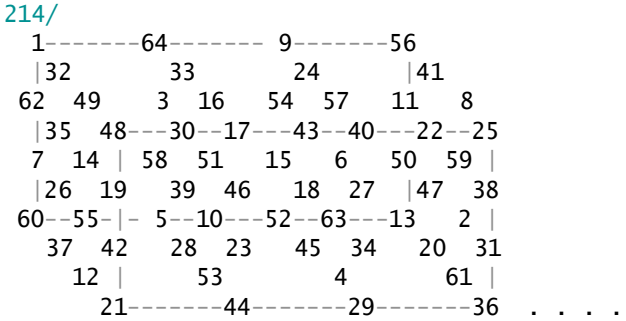
212/



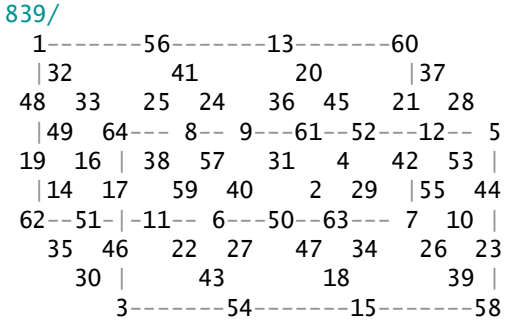
213/



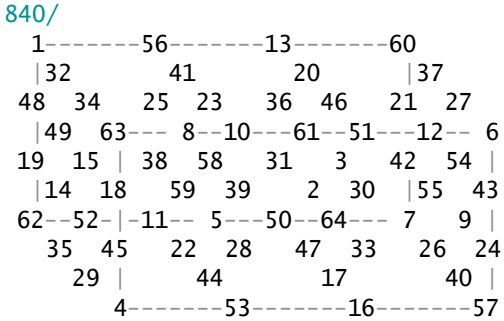
214/



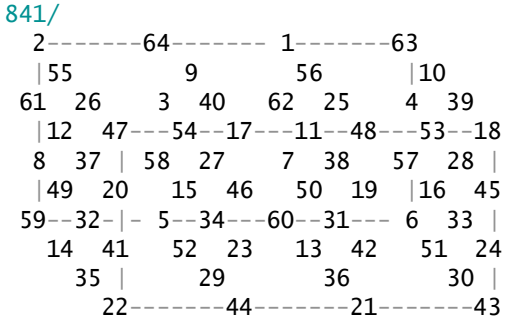
839/



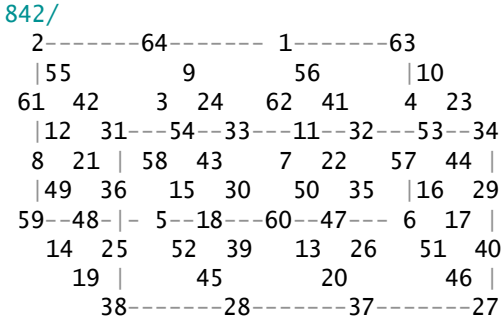
840/



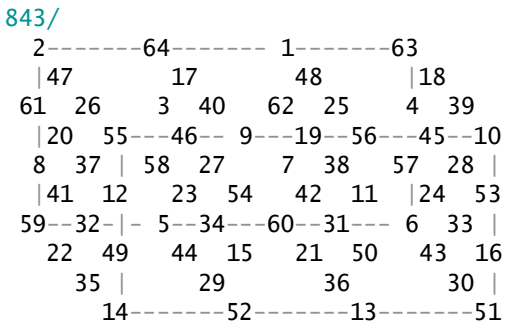
841/



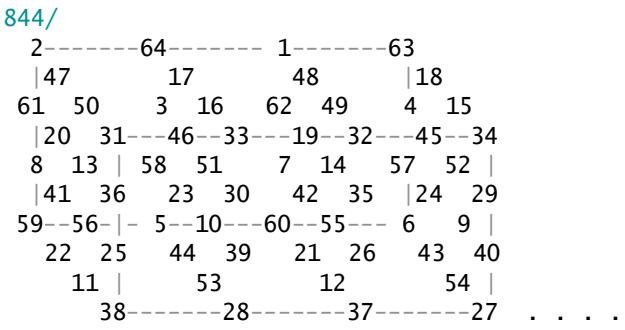
842/



843/

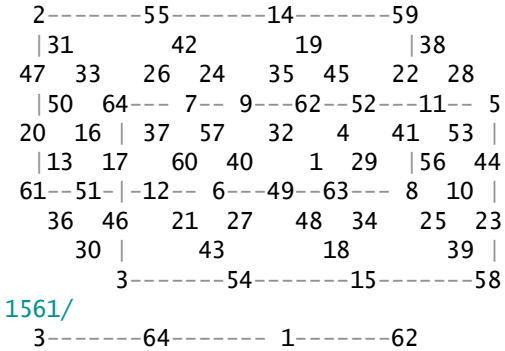


844/

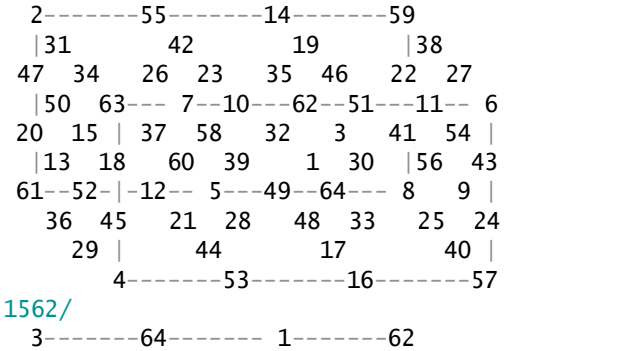


. . . .

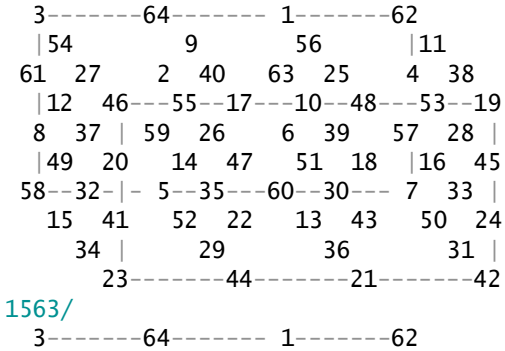
1559/



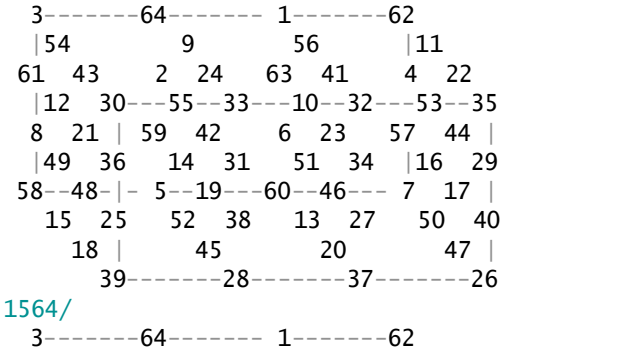
1560/



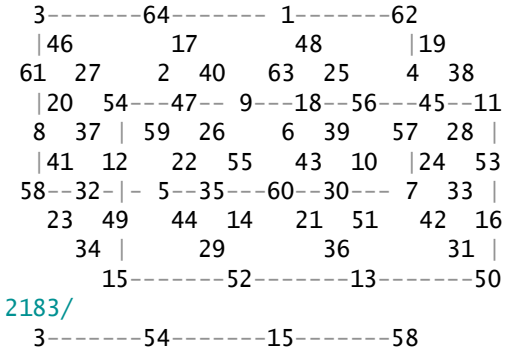
1561/



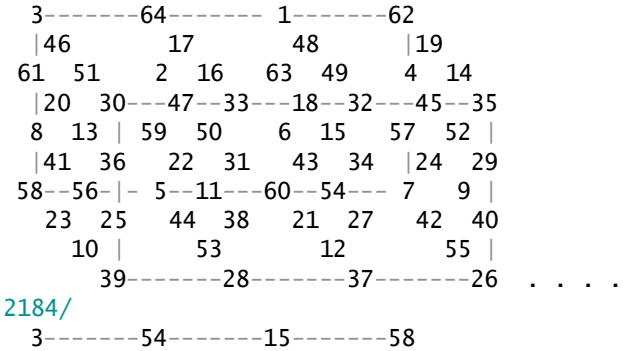
1562/



1563/

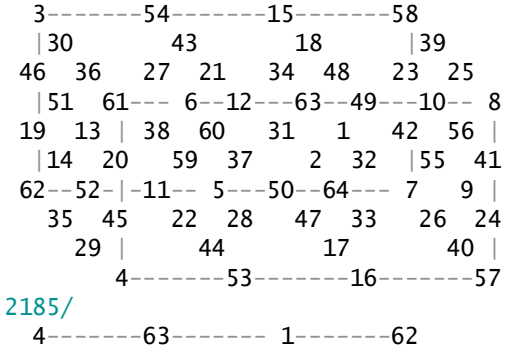


1564/

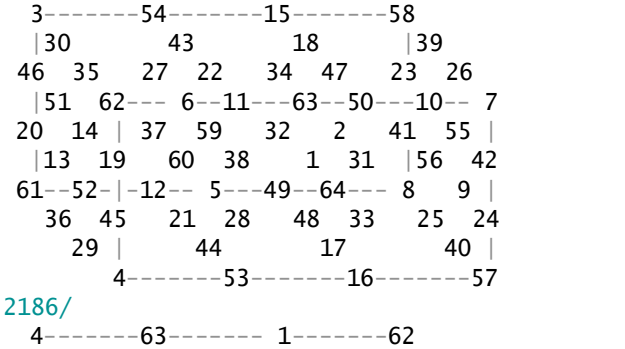


. . . .

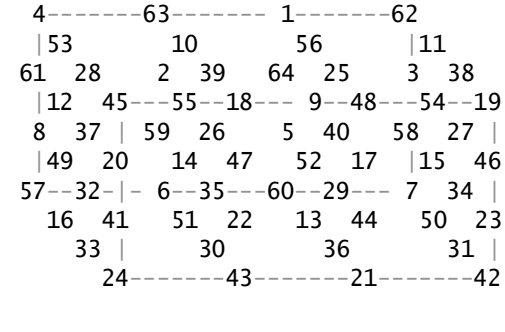
2183/



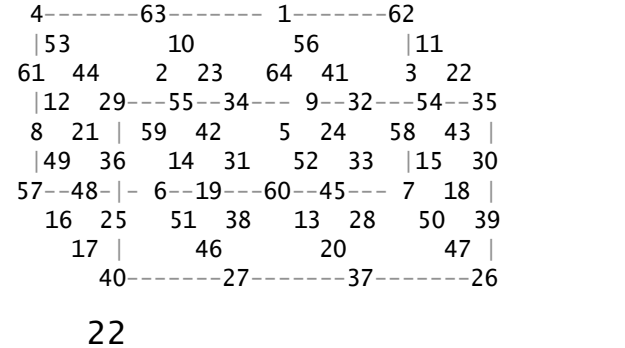
2184/



2185/



2186/



```
2187/
  4-------63------- 1-------62
  |45       18       48      |19
 61  28    2  39   64  25    3  38
  |20  53---47--10---17--56---46--11
  8  37 | 59  26    5  40   58  27 |
  |41  12   22  55   44   9  |23  54
 57--32-|- 6--35---60--29--- 7  34 |
   24  49   43  14   21  52   42  15
     33 |     30       36       31 |
       16-------51-------13-------50
2188/
  4-------63------- 1-------62
  |45       18       48      |19
 61  52    2  15   64  49    3  14
  |20  29---47--34---17--32---46--35
  8  13 | 59  50    5  16   58  51 |
  |41  36   22  31   44  33  |23  30
 57--56-|- 6--11---60--53--- 7  10 |
   24  25   43  38   21  28   42  39
     9 |     54       12       55 |
       40-------27-------37-------26  . . . .
6713/
 29-------52-------15-------34
  | 4       45       18      |63
 40  62    9  19   54  48   27   1
  |57  35---24--14---43--49--- 6--32
 25   7 | 56  42   11  21   38  60 |
  | 8  26   41  55   22  12  |59  37
 36--58-|-13--23---50--44---31   5 |
   61  39   20  10   47  53    2  28
     3 |     46       17       64 |
       30-------51-------16-------33
6714/
 29-------52-------15-------34
  | 4       45       18      |63
 40  61    9  20   54  47   27   2
  |57  36---24--13---43--50--- 6--31
 26   8 | 55  41   12  22   37  59 |
  | 7  25   42  56   21  11  |60  38
 35--58-|-14--23---49--44---32   5 |
   62  39   19  10   48  53    1  28
     3 |     46       17       64 |
       30-------51-------16-------33
6715/
 29-------44-------21-------36
  | 4       53       12      |61
 40  63   17  10   48  55   25   2
  |57  34---16--23---49--42--- 8--31
 26   6 | 47  51   18  14   39  59 |
  | 7  27   50  46   15  19  |58  38
 35--60-|-22--13---43--52---30   5 |
   62  37   11  20   54  45    3  28
     1 |     56       9       64 |
       32-------41-------24-------33
6716/
 29-------44-------21-------36
  | 4       53       12      |61
 40  62   17  11   48  54   25   3
  |57  35---16--22---49--43--- 8--30
 27   7 | 46  50   19  15   38  58 |
  | 6  26   51  47   14  18  |59  39
 34--60-|-23--13---42--52---31   5 |
   63  37   10  20   55  45    2  28
     1 |     56       9       64 |
       32-------41-------24-------33
6717/
 29-------44-------22-------35
  | 4       53       11      |62
 40  63   17  10   47  56   26   1
  |57  34---16--23---50--41--- 7--32
 25   6 | 48  51   18  13   39  60 |
  | 8  27   49  46   15  20  |58  37
 36--59-|-21--14---43--52---30   5 |
   61  38   12  19   54  45    3  28
     2 |     55       9       64 |
       31-------42-------24-------33
6718/
 29-------44-------22-------35
  | 4       53       11      |62
 40  61   17  12   47  54   26   3
  |57  36---16--21---50--43--- 7--30
 27   8 | 46  49   20  15   37  58 |
  | 6  25   51  48   13  18  |60  39
 34--59-|-23--14---41--52---32   5 |
   63  38   10  19   56  45    1  28
     2 |     55       9       64 |
       31-------42-------24-------33
6719/
 29-------44-------23-------34
  | 4       53       10      |63
 40  62   17  11   46  56   27   1
  |57  35---16--22---51--41--- 6--32
 25   7 | 48  50   19  13   38  60 |
  | 8  26   49  47   14  20  |59  37
 36--58-|-21--15---42--52---31   5 |
   61  39   12  18   55  45    2  28
     3 |     54       9       64 |
       30-------43-------24-------33
6720/
 29-------44-------23-------34
  | 4       53       10      |63
 40  61   17  12   46  55   27   2
  |57  36---16--21---51--42--- 6--31
 26   8 | 47  49   20  14   37  59 |
  | 7  25   50  48   13  19  |60  38
 35--58-|-22--15---41--52---32   5 |
   62  39   11  18   56  45    1  28
     3 |     54       9       64 |
       30-------43-------24-------33
```

[Total Count = 6720]

** Solution Counts according to the Values of N1: **

```
( 1: 840) ( 2: 720) ( 3: 624) ( 4: 528) ( 5: 528) ( 6: 432) ( 7: 360) ( 8: 288)
( 9: 408) (10: 312) (11: 240) (12: 168) (13: 168) (14:  96) (15:  48) (16:   0)
(17: 240) (18: 168) (19: 120) (20:  72) (21:  96) (22:  48) (23:  24) (24:   0)
(25:  96) (26:  48) (27:  24) (28:   0) (29:  24) (30:   0) (31:   0) (32:   0)
```

Sum = 6720

** Solution Counts according to the Values of N43, When N1==1: **

```
( 1:   0) ( 2:   0) ( 3:   0) ( 4:   0) ( 5:   0) ( 6:   0) ( 7:   0) ( 8:   0)
```

```
( 9:   0) (10:   0) (11:   0) (12:   0) (13:   0) (14:   0) (15:   0) (16:   0)
(17:   0) (18:   0) (19:   0) (20:   0) (21:   0) (22:   0) (23:   0) (24:   0)
(25:   0) (26:   0) (27:   0) (28:   0) (29:   0) (30:   0) (31:   0) (32: 120)
(33:   0) (34:   0) (35:   0) (36:   0) (37:   0) (38:   0) (39:   0) (40:   0)
(41:   0) (42:   0) (43:   0) (44:   0) (45:   0) (46:   0) (47:   0) (48: 120)
(49:   0) (50:   0) (51:   0) (52:   0) (53:   0) (54:   0) (55:   0) (56: 120)
(57:   0) (58:   0) (59:   0) (60: 120) (61:   0) (62: 120) (63: 120) (64: 120)
Sum = 840
```

```
/** Appendix **/
* Monitor of Solution Correspondences between the Old Data and New *
  ??: 0
   1: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
  81: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
 161: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
 241: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
 321: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
 401: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
 481: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
 561: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
 641: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
 721: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
 801: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
 881: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
 961: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
1041: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
1121: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
1201: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
1281: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
1361: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
1441: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
1521: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
1601: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
1681: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
1761: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
1841: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
1921: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
2001: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
2081: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
2161: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
2241: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
2321: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
2401: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
2481: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
2561: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
2641: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
2721: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
2801: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
2881: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
2961: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
3041: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
3121: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
3201: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
3281: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
3361: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
3441: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
3521: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
3601: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
3681: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
3761: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
3841: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
3921: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
4001: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
4081: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
4161: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
4241: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
4321: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
4401: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
4481: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
4561: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
4641: 11111111111111111111111111111111111111111111111111111111111111111111111111111111
```

```
4721: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
4801: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
4881: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
4961: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5041: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5121: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5201: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5281: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5361: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5441: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5521: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5601: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5681: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5761: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5841: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5921: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
6001: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
6081: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
6161: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
6241: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
6321: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
6401: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
6481: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
6561: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
6641: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
 OK!
```

The last monitor list is an appendix, though I did not present the procedure itself there in the program list above.

It shows our success. We have got just what we have wanted. We could neither have any piece of wrong answers, nor could miss any piece of correct answers at all.

#9. Discussion

But what does our success mean in this experiment?

We used the set of possible view-forms of 6-dimensional ECO developed and the 'DAM Transformation' to get the solution set of 46080 'C&C' MC444 in our process.

Then we transformed them into 322560 'C&P' MC444 by our original method and picked up the 6720 standard solutions of object in the end.

We used the only 'Model Solution' as a material at first, and then we made all objects only by using those 'Methods' all the time, without using anything else.

You can simply say that we could have made all the answers from the only Model using our Method. Why?

Does it mean the only Model knows everything as many as 6720?

Of course, it is we who know everything. We know everything standing behind: definitions, circumstances and various reliable methods. But why could we do all with them, then? What does the whole process mean in our experiment?

I am sure there might be a 'noble family', whose father is the only Model defined with the high dimensional ECO, and whose children are 'C&C' and 'C&P' MC444.

I also think that the only model we took this time might be regarded as the most fundamental solution for all 'Composite and Pantriagonal' Magic Cubes of Order 4.

How amazing it is! Don't you think so?

(Written by Kanji Setsuda on Jan.18, 2009; Working with MacOSX(10.5) & Xcode 3.0)

E-Mail Address: <jag12001@nifty.com>