

**Part 4 : New Advanced Study of Magic Squares and Cubes**  
**Chapter 4: Commentary Articles No.2 by Kanji Setsuda:**

“Various Arts and Tools for Studying Magic Squares”

**Section 11-2 : Compose Self-Complementary MS55 by the 5-th Increment Number System and our “New Euler’s Method”**

**0.** I would like to report here about my new challenge and its result: How I tried to reconstruct all the standard solutions of Self-complementary Magic Squares of Order 5 by the 5-th Increment Number System and our New Euler’s Method.

**1.** We know this type has the 48544 standard solutions in all originally, but we could really have composed only the 32 solutions of the ‘Complete Euler Type’ by the ‘Greco-Latinian Method’ with the 5-th Increment Number System.

\*\* 'Complete Euler Squares' of Order 5 for the Self-Complementary Type \*\*  
 \*\* Made by the 5-th Increment Number System and Greco-Latinian Method \*\*  
 (Used\_Unit:H/L/; Sol\_Numb#; Sum\_Checks:\Diag1/Diag2|Row|Column|)

3/	5/	1#	65/65	4/	6/	2#	65/65	
04321	02413	1	23 20 12	9 65 65	03421	02143	1 18 22 15	9 65 65
21043	41302	15	7 4 21	18 65 65	21340	31402	14 7 20 21	3 65 65
43210	30241	24	16 13 10	2 65 65	14203	43210	10 24 13 2	16 65 65
10432	24130	8	5 22 19	11 65 65	40132	24031	23 5 6 19	12 65 65
32104	13024	17	14 6 3	25 65 65	32014	10324	17 11 4 8	25 65 65
3/	9/	3#	65/65	4/	10/	4#	65/65	
04321	02431	1	23 20 14	7 65 65	03421	02341	1 18 24 15	7 65 65
21043	43102	15	9 2 21	18 65 65	21340	13402	12 9 20 21	3 65 65
43210	10243	22	16 13 10	4 65 65	14203	41230	10 22 13 4	16 65 65
10432	24310	8	5 24 17	11 65 65	40132	24013	23 5 6 17	14 65 65
32104	31024	19	12 6 3	25 65 65	32014	30124	19 11 2 8	25 65 65
9/	3/	5#	65/65	10/	4/	6#	65/65	
02431	04321	1	15 24 18	7 65 65	02341	03421	1 14 20 23	7 65 65
43102	21043	23	17 6 5	14 65 65	13402	21340	8 17 24 5	11 65 65
10243	43210	10	4 13 22	16 65 65	41230	14203	22 10 13 16	4 65 65
24310	10432	12	21 20 9	3 65 65	24013	40132	15 21 2 9	18 65 65
31024	32104	19	8 2 11	25 65 65	30124	32014	19 3 6 12	25 65 65
9/	15/	7#	65/65	10/	16/	8#	65/65	
02431	04123	1	15 22 18	9 65 65	02341	01423	1 12 20 23	9 65 65
43102	23041	23	19 6 5	12 65 65	13402	23140	8 19 22 5	11 65 65
10243	41230	10	2 13 24	16 65 65	41230	34201	24 10 13 16	2 65 65
24310	30412	14	21 20 7	3 65 65	24013	40312	15 21 4 7	18 65 65
31024	12304	17	8 4 11	25 65 65	30124	12034	17 3 6 14	25 65 65
3/ 21/	9#	4/ 22/	10#	3/ 26/	11#	4/ 25/	12#	
2 23 19 11 10		2 18 21 14 10		2 23 19 15 6		2 18 25 14 6		
14 6 5 22 18		15 6 19 22 3		14 10 1 22 18		11 10 19 22 3		
25 17 13 9 1		9 25 13 1 17		21 17 13 9 5		9 21 13 5 17		
8 4 21 20 12		23 4 7 20 11		8 4 25 16 12		23 4 7 16 15		
16 15 7 3 24		16 12 5 8 24		20 11 7 3 24		20 12 1 8 24		
10/ 19/	13#	9/ 20/	14#	9/ 31/	15#	10/ 32/	16#	
2 15 19 23 6		2 14 25 18 6		2 14 21 18 10		2 11 19 23 10		
8 16 25 4 12		23 16 7 4 15		23 20 7 4 11		8 20 21 4 12		
21 9 13 17 5		9 5 13 21 17		9 1 13 25 17		25 9 13 17 1		
14 22 1 10 18		11 22 19 10 3		15 22 19 6 3		14 22 5 6 18		
20 3 7 11 24		20 8 1 12 24		16 8 5 12 24		16 3 7 15 24		

3/ 39/ 17#	4/ 40/ 18#	3/ 44/ 19#	4/ 43/ 20#
4 23 17 11 10	4 18 21 12 10	4 23 17 15 6	4 18 25 12 6
12 6 5 24 18	15 6 17 24 3	12 10 1 24 18	11 10 17 24 3
25 19 13 7 1	7 25 13 1 19	21 19 13 7 5	7 21 13 5 19
8 2 21 20 14	23 2 9 20 11	8 2 25 16 14	23 2 9 16 15
16 15 9 3 22	16 14 5 8 22	20 11 9 3 22	20 14 1 8 22
10/ 33/ 21#	9/ 34/ 22#	9/ 45/ 23#	10/ 46/ 24#
4 15 17 23 6	4 12 25 18 6	4 12 21 18 10	4 11 17 23 10
8 16 25 2 14	23 16 9 2 15	23 20 9 2 11	8 20 21 2 14
21 7 13 19 5	7 5 13 21 19	7 1 13 25 19	25 7 13 19 1
12 24 1 10 18	11 24 17 10 3	15 24 17 6 3	12 24 5 6 18
20 3 9 11 22	20 8 1 14 22	16 8 5 14 22	16 3 9 15 22
3/ 56/ 25#	4/ 55/ 26#	3/ 60/ 27#	4/ 59/ 28#
5 23 16 12 9	5 18 22 11 9	5 23 16 14 7	5 18 24 11 7
11 7 4 25 18	14 7 16 25 3	11 9 2 25 18	12 9 16 25 3
24 20 13 6 2	6 24 13 2 20	22 20 13 6 4	6 22 13 4 20
8 1 22 19 15	23 1 10 19 12	8 1 24 17 15	23 1 10 17 14
17 14 10 3 21	17 15 4 8 21	19 12 10 3 21	19 15 2 8 21
10/ 49/ 29#	9/ 50/ 30#	10/ 61/ 31#	9/ 62/ 32#
5 14 16 23 7	5 11 24 18 7	5 12 16 23 9	5 11 22 18 9
8 17 24 1 15	23 17 10 1 14	8 19 22 1 15	23 19 10 1 12
22 6 13 20 4	6 4 13 22 20	24 6 13 20 2	6 2 13 24 20
11 25 2 9 18	12 25 16 9 3	11 25 4 7 18	14 25 16 7 3
19 3 10 12 21	19 8 2 15 21	17 3 10 14 21	17 8 4 15 21

[Total Count of Solutions = 32] [OK!]

I didn't know why we could have made so few solutions as 32 among 48544.

I was once afraid if this was one of the mysteries of Magic Square World.

But our **Greco-Latinian Method**, on the other hand, shows such the perfect power that it can reconstruct all the 3600 solutions of the Pan-diagonal MS55 and all the 16 of the Simultaneous Type: Self-complementary and Pan-diagonal.

We should find the true reason in the **Latin Condition** itself: Every number of (0, 1, 2, 3 and 4) must be used strictly once in any essential parts of Latin Units such as rows, columns or two primary diagonals. No duplications are allowed.

\*\* List of 'Latin Units' of Order 5 for the Self-Complementary Type \*\*  
 \*\* using the 5-th Increment Number System and Greco-Latinian Method \*\*

1/	2/	3/	4/	5/	6/	7/	8/
03421	02341	04321	03421	02413	02143	04123	02413
41302	21403	21043	21340	41302	31402	31042	21340
10243	43210	43210	14203	30241	43210	43210	34201
24130	14032	10432	40132	24130	24031	20431	40132
32014	30124	32104	32014	13024	10324	12304	13024
9/	10/	11/	12/	13/	14/	15/	16/
02431	02341	04321	02431	02143	01423	04123	01423
43102	13402	13042	23140	23401	43102	23041	23140
10243	41230	41230	14203	41230	30241	41230	34201
24310	24013	20413	40312	34012	24310	30412	40312
31024	30124	32104	31024	10324	12034	12304	12034
17/	18/	19/	20/	21/	22/	23/	24/
14320	12430	14320	13420	12304	12034	13024	12304
30412	20314	20431	20134	30412	40312	40132	20431
01234	34201	03214	34201	41230	34201	34201	43210
23041	03142	31042	01342	23041	23140	21340	31042
42103	41023	42103	42013	04123	01423	02413	04123

25/	26/	27/	28/	29/	30/	31/	32/
12430	12340	13420	12340	12034	10324	13024	10324
04312	34012	04132	24031	24310	34012	24130	24031
30241	01234	30241	03214	30241	41230	30241	43210
23104	23401	21304	31402	43102	23401	41302	31402
41023	40123	42013	40123	01423	02143	02413	02143
33/	34/	35/	36/	37/	38/	39/	40/
34120	31420	34120	32410	32104	31024	32104	32014
20413	20314	10432	20134	20413	40312	10432	40132
01234	14203	03214	14203	41230	14203	43210	14203
13042	03142	21043	01342	13042	23140	21043	21340
42301	42031	42301	43021	04321	02431	04321	03421
41/	42/	43/	44/	45/	46/	47/	48/
32140	31420	32410	32140	31024	30124	32014	30124
24013	04312	04132	14032	24310	24013	24130	14032
01234	10243	10243	03214	10243	41230	10243	43210
13402	23104	21304	21403	43102	13402	41302	21403
40321	42031	43021	40321	02431	02341	03421	02341
49/	50/	51/	52/	53/	54/	55/	56/
43021	40321	43021	42301	42013	40123	42103	42013
21304	21403	01342	21043	21304	31402	31042	01342
10243	03214	14203	03214	30241	03214	03214	34201
04132	14032	20134	10432	04132	24031	20431	20134
32410	32140	32410	34120	13420	12340	14320	13420
57/	58/	59/	60/	61/	62/	63/	64/
42031	40321	42301	42031	41023	40123	42103	41023
23104	13402	13042	03142	23104	23401	23041	03142
10243	01234	01234	14203	30241	01234	01234	34201
04312	24013	20413	20314	04312	34012	30412	20314
31420	32140	34120	31420	12430	12340	14320	12430

[Count of Latin Units = 64]

Only these 64 Latin Units could not really make all the 48544 standard solutions of Self-complementary type. It means we must take some other new method than **Greco-Latinian Method** to compose them by the 5-th Increment Number System.

2. How about taking the same method here as we took to reconstruct various types of magic squares of order 4? If you have not read my article about it yet, find it in the [Section 7-4, Chapter 4, Part 4](#) and read it, please.

Let's take the 5-th Increment Number System, without using Greco-Latinian Method. But let's not stop using our New Euler's method, whose conceptual diagrams I presented below.

**\*\* Conceptual Diagrams of 'New Euler's Method' for Self-complementary MS55:**

**(1) Decomposition of the Object by the 5th Increment Number System]**

[Classical]	#\65/65]	[Mathematical]	\60/60]	/D5i	High\10/10]	Low\10/10]													
1	20	19	22	3 65 65]	0	19	18	21	2 60 60]	0	3	3	4	0 10 10]	0	4	3	1	2 10 10]
5	17	11	18	14 65 65]	4	16	10	17	13 60 60]	0	3	2	3	2 10 10]	4	1	0	2	3 10 10]
24	16	13	10	2 65 65]	23	15	12	9	1 60 60]	4	3	2	1	0 10 10]	3	0	2	4	1 10 10]
12	8	15	9	21 65 65]	11	7	14	8	20 60 60]	2	1	2	1	4 10 10]	1	2	4	3	0 10 10]
23	4	7	6	25 65 65]	22	3	6	5	24 60 60]	4	0	1	1	4 10 10]	2	3	1	0	4 10 10]

[N#Classical - 1 = N#Math; Divide N#Math by 5 and Put the Integer Part of Quotient to High Unit and the Remainder to Low]

**(2) Combine Two Euler Units and Compose Object by the 5th Increment System \*\***

```
(Euler Units: High/Low/ [Math] [Classic] Check Sums:\D1/D2|Row|Column|)
EU: High\10/10|          Low\10/10| [Mathematical] \60/60| [Classical] #\65/65|
0 3 3 4 0|10|10| 0 4 3 1 2|10|10| 0 19 18 21 2|60|60| 1 20 19 22 3|65|65|
0 3 2 3 2|10|10| 4 1 0 2 3|10|10| 4 16 10 17 13|60|60| 5 17 11 18 14|65|65|
4 3 2 1 0|10|10| 3 0 2 4 1|10|10| 23 15 12 9 1|60|60| 24 16 13 10 2|65|65|
2 1 2 1 4|10|10| 1 2 4 3 0|10|10| 11 7 14 8 20|60|60| 12 8 15 9 21|65|65|
4 0 1 1 4|10|10| 2 3 1 0 4|10|10| 22 3 6 5 24|60|60| 23 4 7 6 25|65|65|
[(Every Number in H/) x 5 + (Corresponding Number in L/) = N#Math;
and N#Math + 1 = N#Classical;]
```

What conditions do we have to presuppose when we make our new Euler Units?  
Let's learn all from the original definitions of Self-complementary MS55. For we know the natures of original object are decomposed into its Euler Units faithfully.

**\*\* Original Definitions for Self-complementary Magic Squares of Order 5 \*\***

```
... Basic Form and Equations: C=65; *
| 1| 2| 3| 4| 5| n1+n2+n3+n4+n5=C ...sm1;
|---+---+---+---+---| n6+n7+n8+n9+n10=C ...sm2;
| 6| 7| 8| 9|10| n11+n12+n13+n14+n15=C ...sm3;
|---+---+---+---+---| n16+n17+n18+n19+n20=C ...sm4;
|11|12|13|14|15| n21+n22+n23+n24+n25=C ...sm5;
|---+---+---+---+---| n1+n6+n11+n16+n21=C ...sm6;
|16|17|18|19|20| n2+n7+n12+n17+n22=C ...sm7;
|---+---+---+---+---| n3+n8+n13+n18+n23=C ...sm8;
|21|22|23|24|25| n4+n9+n14+n19+n24=C ...sm9;
|---'---'---'---'---' n5+n10+n15+n20+n25=C ...sm10;
```

```
* Self-complementary Conditions: SC=26; *
n1+n25=n2+n24=n3+n23=n4+n22=n5+n21=n6+n20=n7+n19
=n8+n18=n9+n17=n10+n16=n11+n15=n12+n14=n13+n13=SC
```

\* We must use any number of (1,2,3,4,5,6,...,24,25) strictly once there.

Let's rewrite them into the next list of definitions for our first hypothesis.

**\* New Definitions for Euler Units of SCMS55 \***

```
... Basic Form and Equations: C=10; *
| 1| 2| 3| 4| 5| n1+n2+n3+n4+n5=C ...sm1;
|---+---+---+---+---| n6+n7+n8+n9+n10=C ...sm2;
| 6| 7| 8| 9|10| n11+n12+n13+n14+n15=C ...sm3;
|---+---+---+---+---| n16+n17+n18+n19+n20=C ...sm4;
|11|12|13|14|15| n21+n22+n23+n24+n25=C ...sm5;
|---+---+---+---+---| n1+n6+n11+n16+n21=C ...sm6;
|16|17|18|19|20| n2+n7+n12+n17+n22=C ...sm7;
|---+---+---+---+---| n3+n8+n13+n18+n23=C ...sm8;
|21|22|23|24|25| n4+n9+n14+n19+n24=C ...sm9;
|---'---'---'---'---' n5+n10+n15+n20+n25=C ...sm10;
```

```
* Self-complementary Conditions: SC=4; *
n1+n25=n2+n24=n3+n23=n4+n22=n5+n21=n6+n20=n7+n19
=n8+n18=n9+n17=n10+n16=n11+n15=n12+n14=n13+n13=SC
```

\* We must use any number of (0,1,2,3 and 4) just 5 times in each unit.

The next list shows the 'C' program I wrote recently, only the core part.

```
/** Self-complementary Magic Squares of Order 5 **
/** Reconstructed by "New Euler's Method" and /D5i **
/** 'NewEulerMS55SC.c': Dictated by Kanji Setsuda **
/** on Dec.18, 2011, Jan.25 and Feb. 7, 2012 **
/** Working with MacOSX 10.7.3 & Xcode 4.2.1 **
//
#include <stdio.h>
//
```

```

short int cnt, cnt2;
short eucnt[8];
short u1,u2;
short SC, LSM;
short ecnt, cnt3;
short nm[30], uflg[26];
short anm[8][76];
char tlu[2625][26];
char mtc[2625][2625];
short solt[7137][26];
short cntr[7137][3];
//
void estp01(void), estp02(void), estp03(void), estp04(void), estp05(void);
void estp06(void), estp07(void), estp08(void), estp09(void), estp10(void);
void estp11(void), estp12(void);
void eansrecord(void), preunit(short x);
//
void tblmtc(void);
void cmbcmp(void), solsave(void);
void prans(short x), pr2ans(void), pr1ans(void);
//
/* Main Program *
int main(void){
short n;
printf("\n");
printf("** Reconstruction of Self-complementary Magic Squares 5x5: **\n");
printf(" ** by the 5-th Increment System and 'New Euler's Method' **\n");
for(n=0;n<5;n++){uflg[n]=0;}
nm[13]=2; uflg[2]=1;
SC=4; LSM=10; cnt=0;
printf("\n");
printf(" ** List of Euler Units made by /D5i **\n");
estp01(); // Make the Euler Units by /D5i
if(cnt3>0){preunit(cnt3);} // Print the Rest One
ecnt=cnt; printf(" [Count of Euler Units = %d]\n",ecnt);
tblmtc();
printf("\n");
printf("** Standard Solutions of Self-complementary Magic Squares 5x5\n");
printf(" Reconstructed by the 5th Increment System and 'New Euler's Method' **\n");
printf(" (Euler_Units: High/Low/ Sol_Number# Check_Sums:\\Diagonals/Rows|Columns|)\n");
cnt=0; cnt3=0;
cmbcmp(); // Combine and Compose
prans(500); // Print the Solutions with that Interval
if(cnt3==1){pr1ans();} // Print the Rest One
printf(" [Solution Counts = %d] [OK!]\n",cnt);
printf("\n");
printf("** Calculated and listed by Kanji Setsuda\n");
printf(" on Jan.25, 2012 with MacOSX 10.7.2 & Xcode 4.2.1 **\n");
printf("\n");
return 0;
}
//
/** Make the Euler Units with /D5i **
// Level #1:
// Set N1 & n25
void estp01(){
short i,j;
for(i=0;i<5;i++){j=SC-i;
if((uflg[i]<5)&&(uflg[j]<5)){
nm[1]=i; nm[25]=j;
uflg[i]++; uflg[j]++;
estp02();
uflg[i]--; uflg[j]--;}
}
}
}

```

```

// Set N5 & n21
void estp02(){
  short i,j;
  for(i=0;i<5;i++){j=SC-i;
    if((uflg[i]<5)&&(uflg[j]<5)){cnt2=0;
      nm[5]=i; nm[21]=j;
      uflg[i]++; uflg[j]++;
      estp03();
      uflg[i]--; uflg[j]--;}
  }
}
// Set N7 & n19
void estp03(){
  short i,j,sm;
  for(i=0;i<5;i++){j=SC-i;
    if((uflg[i]<5)&&(uflg[j]<5)){if(nm[1]==0){cnt2=0;}
      nm[7]=i; nm[19]=j;
      uflg[i]++; uflg[j]++;
      sm=nm[1]+nm[7]+nm[13]+nm[19]+nm[25];
      if(sm==LSM){estp04();}
      uflg[i]--; uflg[j]--;}
  }
}
// Set N9 & n17
void estp04(){
  short i,j,sm;
  for(i=0;i<5;i++){j=SC-i;
    if((uflg[i]<5)&&(uflg[j]<5)){
      nm[9]=i; nm[17]=j;
      uflg[i]++; uflg[j]++;
      sm=nm[5]+nm[9]+nm[13]+nm[17]+nm[21];
      if(sm==LSM){estp05();}
      uflg[i]--; uflg[j]--;}
  }
}
// Level #2:
// Set N2 & n24
void estp05(){
  short i,j;
  for(i=4;i>=0;i--){j=SC-i;
    if((uflg[i]<5)&&(uflg[j]<5)){
      nm[2]=i; nm[24]=j;
      uflg[i]++; uflg[j]++;
      estp06();
      uflg[i]--; uflg[j]--;}
  }
}
// Set N4 & n22
void estp06(){
  short i,j;
  for(i=0;i<5;i++){j=SC-i;
    if((uflg[i]<5)&&(uflg[j]<5)){
      nm[4]=i; nm[22]=j;
      uflg[i]++; uflg[j]++;
      estp07();
      uflg[i]--; uflg[j]--;}
  }
}
// Set N3 & n23
void estp07(){
  short i,j,sm1,sm2;
  for(i=0;i<5;i++){j=SC-i;
    if((uflg[i]<5)&&(uflg[j]<5)){
      nm[3]=i; nm[23]=j;
      uflg[i]++; uflg[j]++;

```

```

        sm1=nm[1]+nm[2]+nm[3]+nm[4]+nm[5];
        sm2=nm[25]+nm[24]+nm[23]+nm[22]+nm[21];
        if((sm1==LSM)&&(sm2==LSM)){estp08();}
        uflg[i]--; uflg[j]--;}
    }
}
// Set N8 & n18
void estp08(){
    short i,j,sm;
    for(i=0;i<5;i++){j=SC-i;
        if((uflg[i]<5)&&(uflg[j]<5)){
            nm[8]=i; nm[18]=j;
            uflg[i]++; uflg[j]++;
            sm=nm[3]+nm[8]+nm[13]+nm[18]+nm[23];
            if(sm==LSM){estp09();}
            uflg[i]--; uflg[j]--;}
        }
    }
}
// Set N12 & n14
void estp09(){
    short i,j,sm1,sm2;
    for(i=0;i<5;i++){j=SC-i;
        if((uflg[i]<5)&&(uflg[j]<5)){
            nm[12]=i; nm[14]=j;
            uflg[i]++; uflg[j]++;
            sm1=nm[2]+nm[7]+nm[12]+nm[17]+nm[22];
            sm2=nm[24]+nm[19]+nm[14]+nm[9]+nm[4];
            if((sm1==LSM)&&(sm2==LSM)){estp10();}
            uflg[i]--; uflg[j]--;}
        }
    }
}
// Set N6 & n20
void estp10(){
    short i,j;
    for(i=0;i<5;i++){j=SC-i;
        if((uflg[i]<5)&&(uflg[j]<5)){
            nm[6]=i; nm[20]=j;
            uflg[i]++; uflg[j]++;
            estp11();
            uflg[i]--; uflg[j]--;}
        }
    }
}
// Set N10 & n16
void estp11(){
    short i,j,sm1,sm2;
    for(i=0;i<5;i++){j=SC-i;
        if((uflg[i]<5)&&(uflg[j]<5)){
            nm[10]=i; nm[16]=j;
            uflg[i]++; uflg[j]++;
            sm1=nm[6]+nm[7]+nm[8]+nm[9]+nm[10];
            sm2=nm[20]+nm[19]+nm[18]+nm[17]+nm[16];
            if((sm1==LSM)&&(sm2==LSM)){estp12();}
            uflg[i]--; uflg[j]--;}
        }
    }
}
// Set N11 & n15
void estp12(){
    short i,j,sm1,sm2,sm3;
    for(i=0;i<5;i++){j=SC-i;
        if((uflg[i]<5)&&(uflg[j]<5)){
            nm[11]=i; nm[15]=j;
            uflg[i]++; uflg[j]++;
            sm1=nm[1]+nm[6]+nm[11]+nm[16]+nm[21];
            sm2=nm[25]+nm[20]+nm[15]+nm[10]+nm[5];
            sm3=nm[11]+nm[12]+nm[13]+nm[14]+nm[15];

```

```

        if((sm1==LSM)&&(sm2==LSM)&&(sm3==LSM)){eansrecord();}
        uflg[i]--; uflg[j]--;}
    }
}
//
// Record Euler Units
void eansrecord(){
    short n;
    for(n=0;n<25;n++){tlu[cnt][n]=nm[n+1];}
    cnt++; cnt2++;
    if(cnt2==1){eucnt[cnt3]=cnt;
        for(n=0;n<25;n++){anm[cnt3][n]=nm[n+1];}
        cnt3++; if(cnt3==7){preunit(cnt3); cnt3=0;}
    }
}
//
// Print Euler Units
void preunit(short x){
    short l,m,m5,n;
    for(l=0;l<x;l++){printf(" %9d/",eucnt[l]);}
    printf("\n");
    for(m=0;m<5;m++){m5=m*5;
        for(l=0;l<x;l++){printf(" ");
            for(n=0;n<5;n++){printf("%2d",anm[l][m5+n]);}}
        printf("\n");
    }
}
//
// Check how well they match and Save the Data to the Table
void tblmtc(){
    short h,l;
    short n,d1,d2;
    for(h=0;h<ecnt;h++){
        for(l=0;l<ecnt;l++){
            d1=0; for(n=1;n<26;n++){if(teu[h][n]==teu[l][n]){d1++;}}
            d2=0; for(n=1;n<26;n++){if(teu[h][n]+teu[l][n]==SC){d2++;}}
            if(d1<d2){d1=d2;}
            mtc[h][l]=d1;}}
}
//
// * Combine and Compose *
void cmbcmp(){
    short n,d,fc;
    for(u1=0;u1<(ecnt/2);u1++){cnt2=0;
        for(u2=0;u2<ecnt;u2++){
            if(mtc[u1][u2]==5){
                for(n=1;n<26;n++){uflg[n]=0;}
                for(n=0;n<25;n++){
                    d=tlu[u1][n]*5+tlu[u2][n]+1;
                    nm[n+1]=d; uflg[d]++;
                }
                fc=0;
                for(n=1;n<26;n++){if(uflg[n]==0){fc++;}}
                if(fc==0){
                    if((nm[1]<nm[5])&&(nm[5]<nm[21])&&(nm[1]<nm[25])){solsave();}}
            }
        }
    }
}
//
// Save Answers to the Solution List
void solsave(){
    short n;
    for(n=1;n<26;n++){solt[cnt][n]=nm[n];}
    cntr[cnt][0]=cnt+1; cntr[cnt][1]=u1; cntr[cnt][2]=u2;
}

```



```

    cnt++;
}
//
// Print the Solution List
void prans(short x){
    short s;
    short m,n,sm1,sm2;
    for(s=0;s<cnt;s=s+x){
        eucnt[cnt3*3]=s+1;
        for(n=1;n<26;n++){nm[n]=solt[s][n]; anm[cnt3][n]=nm[n]; anm[cnt3+1][n]=0;}
        for(m=0;m<5;m++){sm1=0; sm2=0;
            for(n=0;n<5;n++){sm1=sm1+nm[m*5+n+1]; sm2=sm2+nm[n*5+m+1];}
            anm[cnt3][m+26]=sm1; anm[cnt3][m+31]=sm2;}
        anm[cnt3][36]=nm[1]+nm[7]+nm[13]+nm[19]+nm[25];
        anm[cnt3][37]=nm[5]+nm[9]+nm[13]+nm[17]+nm[21];
        eucnt[cnt3*3+1]=cntr[s][1]+1; eucnt[cnt3*3+2]=cntr[s][2]+1;
        cnt3++;
        if(cnt3==2){pr2ans(); cnt3=0;}
    }
}
//
// Print 2 Answers with Sum Checks
void pr2ans(){
    short lu1,lu2;
    short l,l5,m,n;
    printf("%5d/H%5d/L%15d# \\%2d/%2d| %5d/H%5d/L%15d# \\%2d/%2d|\n",
        eucnt[1],eucnt[2],eucnt[0],anm[0][36],anm[0][37],
        eucnt[4],eucnt[5],eucnt[3],anm[1][36],anm[1][37]);
    for(l=0;l<5;l++){l5=l*5;
        for(m=0;m<2;m++){
            lu1=eucnt[m*3+1]-1; lu2=eucnt[m*3+2]-1;
            printf(" ");
            for(n=0;n<5;n++){printf("%d",tlu[lu1][l5+n]);}
            printf(" ");
            for(n=0;n<5;n++){printf("%d",tlu[lu2][l5+n]);}
            printf(" ");
            for(n=1;n<6;n++){printf("%3d",anm[m][l5+n]);}
            printf(" |%2d|%2d|",anm[m][l+26],anm[m][l+31]);
            if(m<1){printf(" ");}
        }
        printf("\n");
    }
}
//
// Print the Rest One
void pr1ans(){
    short lu1,lu2;
    short l,l5,n;
    lu1=eucnt[1]-1; lu2=eucnt[2]-1;
    printf("%5d/H%5d/L%15d# \\%2d/%2d|\n",
        lu1+1,lu2+1,eucnt[0],anm[0][36],anm[0][37]);
    for(l=0;l<5;l++){l5=l*5;
        printf(" ");
        for(n=0;n<5;n++){printf("%d",tlu[lu1][l5+n]);}
        printf(" ");
        for(n=0;n<5;n++){printf("%d",tlu[lu2][l5+n]);}
        printf(" ");
        for(n=1;n<6;n++){printf("%3d",anm[0][l5+n]);}
        printf(" |%2d|%2d|\n",anm[0][l+26],anm[0][l+31]);
    }
}
//

```

The following lists show the results of its execution.

**\*\* Reconstruction of Self-complementary Magic Squares 5x5: \*\***  
**\*\* by the 5-th Increment System and 'New Euler's Method' \*\***

**\*\* List of Euler Units made by /D5i \*\***

1/ 0 4 4 2 0 3 0 3 1 3 2 1 2 3 2 1 3 1 4 1 4 2 0 0 4	12/ 0 4 2 4 0 3 1 3 0 3 2 1 2 3 2 1 4 1 3 1 4 0 2 0 4	45/ 0 3 4 3 0 2 2 3 0 3 3 0 2 4 1 1 4 1 2 2 4 1 0 1 4	69/ 0 3 3 4 0 2 3 2 0 3 3 0 2 4 1 1 4 2 1 2 4 0 1 1 4	102/ 0 3 3 4 0 1 4 2 1 2 3 0 2 4 1 2 3 2 0 3 4 0 1 1 4	113/ 0 4 2 3 1 4 0 3 0 3 2 1 2 3 2 1 4 1 4 0 3 1 2 0 4	143/ 0 4 2 3 1 4 1 3 0 2 1 0 2 4 3 2 4 1 3 0 3 1 2 0 4
163/ 0 4 1 4 1 3 2 2 0 3 3 0 2 4 1 1 4 2 2 1 3 0 3 0 4	189/ 0 3 2 4 1 4 3 1 0 2 1 0 2 4 3 2 4 3 1 0 3 0 2 1 4	209/ 0 2 3 4 1 3 4 2 0 1 1 0 2 4 3 3 4 2 0 1 3 0 1 2 4	239/ 0 3 3 2 2 3 0 4 0 3 4 1 2 3 0 1 4 0 4 1 2 2 1 1 4	257/ 0 4 1 3 2 4 1 2 0 3 3 0 2 4 1 1 4 2 3 0 2 1 3 0 4	289/ 0 4 1 3 2 4 2 0 1 3 3 0 2 4 1 1 3 4 2 0 2 1 3 0 4	301/ 0 3 1 4 2 4 3 2 0 1 1 0 2 4 3 3 4 2 1 0 2 0 3 1 4
333/ 0 1 4 3 2 3 4 1 0 2 3 0 2 4 1 2 4 3 0 1 2 1 0 3 4	351/ 0 3 2 2 3 4 0 3 0 3 4 1 2 3 0 1 4 1 4 0 1 2 2 1 4	381/ 0 3 0 4 3 4 1 3 0 2 3 2 2 2 1 2 4 1 3 0 1 0 4 1 4	401/ 0 3 0 4 3 3 2 3 0 2 4 1 2 3 0 2 4 1 2 1 1 0 4 1 4	427/ 0 2 2 3 3 4 3 3 0 0 1 0 2 4 3 4 4 1 1 0 1 1 2 2 4	447/ 0 2 1 4 3 3 4 2 0 1 3 0 2 4 1 3 4 2 0 1 1 0 3 2 4	477/ 0 3 2 1 4 4 0 3 1 2 4 1 2 3 0 2 3 1 4 0 0 3 2 1 4
488/ 0 2 1 3 4 4 1 4 0 1 3 2 2 2 1 3 4 0 3 0 0 1 3 2 4	521/ 0 2 1 3 4 3 2 4 0 1 4 1 2 3 0 3 4 0 2 1 0 1 3 2 4	545/ 0 2 1 3 4 4 3 2 0 1 3 0 2 4 1 3 4 2 1 0 0 1 3 2 4	578/ 0 2 1 3 4 3 4 2 1 0 3 0 2 4 1 4 3 2 0 1 0 1 3 2 4	589/ 1 4 2 3 0 3 0 3 0 4 2 1 2 3 2 0 4 1 4 1 4 1 2 0 3	715/ 1 4 2 2 1 3 0 3 0 4 3 0 2 4 1 0 4 1 4 1 3 2 2 0 3	775/ 1 4 1 2 2 3 0 4 0 3 3 0 2 4 1 1 4 0 4 1 2 2 3 0 3
895/ 1 3 1 2 3 4 0 2 0 4 4 1 2 3 0 0 4 2 4 0 1 2 3 1 3	955/ 1 2 2 1 4 4 0 3 0 3 4 1 2 3 0 1 4 1 4 0 0 3 2 2 3	1081/ 2 3 3 2 0 3 0 4 0 3 0 1 2 3 4 1 4 0 4 1 4 2 1 1 2	1193/ 2 4 1 2 1 3 0 4 0 3 1 0 2 4 3 1 4 0 4 1 3 2 3 0 2	1313/ 2 3 1 1 3 4 0 2 0 4 3 0 2 4 1 0 4 2 4 0 1 3 3 1 2	1433/ 2 2 1 1 4 3 0 4 0 3 4 1 2 3 0 1 4 0 4 1 0 3 3 2 2	1545/ 3 3 2 2 0 3 0 3 0 4 0 1 2 3 4 0 4 1 4 1 4 2 2 1 1
1671/ 3 3 1 2 1 4 0 2 0 4 0 1 2 3 4 0 4 2 4 0 3 2 3 1 1	1731/ 3 3 1 1 2 4 0 2 0 4 1 0 2 4 3 0 4 2 4 0 2 3 3 1 1	1851/ 3 2 2 0 3 3 0 3 0 4 3 0 2 4 1 0 4 1 4 1 1 4 2 2 1	1911/ 3 2 0 1 4 4 0 3 0 3 2 1 2 3 2 1 4 1 4 0 0 3 4 2 1	2037/ 4 3 2 1 0 2 0 3 1 4 0 1 2 3 4 0 3 1 4 2 4 3 2 1 0	2149/ 4 2 2 1 1 3 0 3 0 4 0 1 2 3 4 0 4 1 4 1 3 3 2 2 0	2275/ 4 2 1 1 2 3 0 4 0 3 0 1 2 3 4 1 4 0 4 1 2 3 3 2 0
2387/ 4 2 0 1 3 3 0 3 0 4 2 1 2 3 2 0 4 1 4 1 1 3 4 2 0	2513/ 4 2 0 0 4 3 0 3 1 3 2 1 2 3 2 1 3 1 4 1 0 4 4 2 0	[Count of Euler Units = 2624]				

**\*\* Euler Type of Self-complementary Magic Squares of Order 5**  
**Reconstructed by the 5th Increment System and New Euler's Method \*\***

[List of Euler Units and the Composed Objects with Check Sums]

(Euler Units: High/Low/ Sol Number# Check Sums of Each:\D1/D2|Row|Column|)

/EU	1/H\10/10	339/L\10/10	[Math]	\60/60	1#\65/65		
0 4 4 2 0	10 10	0 1 3 4 2	10 10	0 21 23 14 2	60 60	1 22 24 15 3	65 65
3 0 3 1 3	10 10	4 4 0 1 1	10 10	19 4 15 6 16	60 60	20 5 16 7 17	65 65
2 1 2 3 2	10 10	1 2 2 2 3	10 10	11 7 12 17 13	60 60	12 8 13 18 14	65 65
1 3 1 4 1	10 10	3 3 4 0 0	10 10	8 18 9 20 5	60 60	9 19 10 21 6	65 65
4 2 0 0 4	10 10	2 0 1 3 4	10 10	22 10 1 3 24	60 60	23 11 2 4 25	65 65
/EU	10/H\10/10	1935/L\10/10	[Math]	\60/60	101#\65/65		
0 4 3 3 0	10 10	3 2 1 0 4	10 10	3 22 16 15 4	60 60	4 23 17 16 5	65 65
2 0 2 3 3	10 10	1 0 4 3 2	10 10	11 0 14 18 17	60 60	12 1 15 19 18	65 65
3 4 2 0 1	10 10	4 3 2 1 0	10 10	19 23 12 1 5	60 60	20 24 13 2 6	65 65
1 1 2 4 2	10 10	2 1 0 4 3	10 10	7 6 10 24 13	60 60	8 7 11 25 14	65 65
4 1 1 0 4	10 10	0 4 3 2 1	10 10	20 9 8 2 21	60 60	21 10 9 3 22	65 65

/EU	19/H\10/10	580/L\10/10	[Math]	\60/60	201#\65/65
	0 4 4 2 0 10 10	0 1 2 3 4 10 10		0 21 22 13 4 60 60	1 22 23 14 5 65 65
	3 1 3 1 2 10 10	4 4 1 1 0 10 10		19 9 16 6 10 60 60	20 10 17 7 11 65 65
	1 0 2 4 3 10 10	2 1 2 3 2 10 10		7 1 12 23 17 60 60	8 2 13 24 18 65 65
	2 3 1 3 1 10 10	4 3 3 0 0 10 10		14 18 8 15 5 60 60	15 19 9 16 6 65 65
	4 2 0 0 4 10 10	0 1 2 3 4 10 10		20 11 2 3 24 60 60	21 12 3 4 25 65 65
/EU	26/H\10/10	1469/L\10/10	[Math]	\60/60	301#\65/65
	0 2 4 4 0 10 10	2 0 3 1 4 10 10		2 10 23 21 4 60 60	3 11 24 22 5 65 65
	2 1 3 1 3 10 10	3 1 4 2 0 10 10		13 6 19 7 15 60 60	14 7 20 8 16 65 65
	3 4 2 0 1 10 10	1 4 2 0 3 10 10		16 24 12 0 8 60 60	17 25 13 1 9 65 65
	1 3 1 3 2 10 10	4 2 0 3 1 10 10		9 17 5 18 11 60 60	10 18 6 19 12 65 65
	4 0 0 2 4 10 10	0 3 1 4 2 10 10		20 3 1 14 22 60 60	21 4 2 15 23 65 65
/EU	42/H\10/10	262/L\10/10	[Math]	\60/60	401#\65/65
	0 4 4 2 0 10 10	0 1 3 4 2 10 10		0 21 23 14 2 60 60	1 22 24 15 3 65 65
	1 1 3 4 1 10 10	4 1 2 0 3 10 10		9 6 17 20 8 60 60	10 7 18 21 9 65 65
	2 3 2 1 2 10 10	3 4 2 0 1 10 10		13 19 12 5 11 60 60	14 20 13 6 12 65 65
	3 0 1 3 3 10 10	1 4 2 3 0 10 10		16 4 7 18 15 60 60	17 5 8 19 16 65 65
	4 2 0 0 4 10 10	2 0 1 3 4 10 10		22 10 1 3 24 60 60	23 11 2 4 25 65 65
/EU	53/H\10/10	474/L\10/10	[Math]	\60/60	501#\65/65
	0 3 3 4 0 10 10	0 3 2 2 3 10 10		0 18 17 22 3 60 60	1 19 18 23 4 65 65
	3 2 0 1 4 10 10	1 4 1 4 0 10 10		16 14 1 9 20 60 60	17 15 2 10 21 65 65
	3 2 2 2 1 10 10	4 1 2 3 0 10 10		19 11 12 13 5 60 60	20 12 13 14 6 65 65
	0 3 4 2 1 10 10	4 0 3 0 3 10 10		4 15 23 10 8 60 60	5 16 24 11 9 65 65
	4 0 1 1 4 10 10	1 2 2 1 4 10 10		21 2 7 6 24 60 60	22 3 8 7 25 65 65
/EU	107/H\10/10	509/L\10/10	[Math]	\60/60	1001#\65/65
	0 4 2 4 0 10 10	0 3 1 2 4 10 10		0 23 11 22 4 60 60	1 24 12 23 5 65 65
	1 4 1 3 1 10 10	4 1 0 3 2 10 10		9 21 5 18 7 60 60	10 22 6 19 8 65 65
	2 1 2 3 2 10 10	4 3 2 1 0 10 10		14 8 12 16 10 60 60	15 9 13 17 11 65 65
	3 1 3 0 3 10 10	2 1 4 3 0 10 10		17 6 19 3 15 60 60	18 7 20 4 16 65 65
	4 0 2 0 4 10 10	0 2 3 1 4 10 10		20 2 13 1 24 60 60	21 3 14 2 25 65 65
/EU	136/H\10/10	1213/L\10/10	[Math]	\60/60	1501#\65/65
	0 4 3 2 1 10 10	2 3 4 0 1 10 10		2 23 19 10 6 60 60	3 24 20 11 7 65 65
	4 0 2 3 1 10 10	1 0 3 2 4 10 10		21 0 13 17 9 60 60	22 1 14 18 10 65 65
	0 3 2 1 4 10 10	4 1 2 3 0 10 10		4 16 12 8 20 60 60	5 17 13 9 21 65 65
	3 1 2 4 0 10 10	0 2 1 4 3 10 10		15 7 11 24 3 60 60	16 8 12 25 4 65 65
	3 2 1 0 4 10 10	3 4 0 1 2 10 10		18 14 5 1 22 60 60	19 15 6 2 23 65 65
/EU	154/H\10/10	1354/L\10/10	[Math]	\60/60	2001#\65/65
	0 4 1 4 1 10 10	2 0 4 1 3 10 10		2 20 9 21 8 60 60	3 21 10 22 9 65 65
	4 1 0 2 3 10 10	4 1 1 0 4 10 10		24 6 1 10 19 60 60	25 7 2 11 20 65 65
	2 3 2 1 2 10 10	3 2 2 2 1 10 10		13 17 12 7 11 60 60	14 18 13 8 12 65 65
	1 2 4 3 0 10 10	0 4 3 3 0 10 10		5 14 23 18 0 60 60	6 15 24 19 1 65 65
	3 0 3 0 4 10 10	1 3 0 4 2 10 10		16 3 15 4 22 60 60	17 4 16 5 23 65 65
/EU	178/H\10/10	863/L\10/10	[Math]	\60/60	2501#\65/65
	0 4 3 2 1 10 10	1 0 4 3 2 10 10		1 20 19 13 7 60 60	2 21 20 14 8 65 65
	0 2 4 3 1 10 10	2 4 1 0 3 10 10		2 14 21 15 8 60 60	3 15 22 16 9 65 65
	4 1 2 3 0 10 10	4 1 2 3 0 10 10		24 6 12 18 0 60 60	25 7 13 19 1 65 65
	3 1 0 2 4 10 10	1 4 3 0 2 10 10		16 9 3 10 22 60 60	17 10 4 11 23 65 65
	3 2 1 0 4 10 10	2 1 0 4 3 10 10		17 11 5 4 23 60 60	18 12 6 5 24 65 65
/EU	201/H\10/10	1228/L\10/10	[Math]	\60/60	3001#\65/65
	0 4 3 2 1 10 10	2 3 4 0 1 10 10		2 23 19 10 6 60 60	3 24 20 11 7 65 65
	1 3 0 4 2 10 10	2 0 3 4 1 10 10		7 15 3 24 11 60 60	8 16 4 25 12 65 65
	4 1 2 3 0 10 10	0 3 2 1 4 10 10		20 8 12 16 4 60 60	21 9 13 17 5 65 65
	2 0 4 1 3 10 10	3 0 1 4 2 10 10		13 0 21 9 17 60 60	14 1 22 10 18 65 65
	3 2 1 0 4 10 10	3 4 0 1 2 10 10		18 14 5 1 22 60 60	19 15 6 2 23 65 65
/EU	221/H\10/10	543/L\10/10	[Math]	\60/60	3501#\65/65
	0 2 4 3 1 10 10	0 3 1 2 4 10 10		0 13 21 17 9 60 60	1 14 22 18 10 65 65
	3 4 1 2 0 10 10	3 2 0 4 1 10 10		18 22 5 14 1 60 60	19 23 6 15 2 65 65
	0 1 2 3 4 10 10	4 3 2 1 0 10 10		4 8 12 16 20 60 60	5 9 13 17 21 65 65
	4 2 3 0 1 10 10	3 0 4 2 1 10 10		23 10 19 2 6 60 60	24 11 20 3 7 65 65
	3 1 0 2 4 10 10	0 2 3 1 4 10 10		15 7 3 11 24 60 60	16 8 4 12 25 65 65

```

/EU 254/H\10/10|      147/L\10/10| [Math]      \60/60|      4001#\65/65|
0 3 3 2 2|10|10| 0 3 2 4 1|10|10| 0 18 17 14 11|60|60| 1 19 18 15 12|65|65|
3 0 4 3 0|10|10| 4 1 1 0 4|10|10| 19 1 21 15 4|60|60| 20 2 22 16 5|65|65|
1 4 2 0 3|10|10| 3 2 2 2 1|10|10| 8 22 12 2 16|60|60| 9 23 13 3 17|65|65|
4 1 0 4 1|10|10| 0 4 3 3 0|10|10| 20 9 3 23 5|60|60| 21 10 4 24 6|65|65|
2 2 1 1 4|10|10| 3 0 2 1 4|10|10| 13 10 7 6 24|60|60| 14 11 8 7 25|65|65|

/EU 294/H\10/10|      135/L\10/10| [Math]      \60/60|      4501#\65/65|
0 1 3 4 2|10|10| 0 4 3 2 1|10|10| 0 9 18 22 11|60|60| 1 10 19 23 12|65|65|
3 2 4 1 0|10|10| 4 0 0 3 3|10|10| 19 10 20 8 3|60|60| 20 11 21 9 4|65|65|
1 4 2 0 3|10|10| 2 3 2 1 2|10|10| 7 23 12 1 17|60|60| 8 24 13 2 18|65|65|
4 3 0 2 1|10|10| 1 1 4 4 0|10|10| 21 16 4 14 5|60|60| 22 17 5 15 6|65|65|
2 0 1 3 4|10|10| 3 2 1 0 4|10|10| 13 2 6 15 24|60|60| 14 3 7 16 25|65|65|

/EU 332/H\10/10|      1585/L\10/10| [Math]      \60/60|      5001#\65/65|
0 2 3 3 2|10|10| 3 3 0 4 0|10|10| 3 13 15 19 10|60|60| 4 14 16 20 11|65|65|
3 3 0 4 0|10|10| 2 1 1 2 4|10|10| 17 16 1 22 4|60|60| 18 17 2 23 5|65|65|
1 4 2 0 3|10|10| 1 4 2 0 3|10|10| 6 24 12 0 18|60|60| 7 25 13 1 19|65|65|
4 0 4 1 1|10|10| 0 2 3 3 2|10|10| 20 2 23 8 7|60|60| 21 3 24 9 8|65|65|
2 1 1 2 4|10|10| 4 0 4 1 1|10|10| 14 5 9 11 21|60|60| 15 6 10 12 22|65|65|

/EU 753/H\10/10|      266/L\10/10| [Math]      \60/60|      5501#\65/65|
1 4 0 4 1|10|10| 0 3 3 2 2|10|10| 5 23 3 22 7|60|60| 6 24 4 23 8|65|65|
0 3 3 2 2|10|10| 4 1 0 1 4|10|10| 4 16 15 11 14|60|60| 5 17 16 12 15|65|65|
4 1 2 3 0|10|10| 4 1 2 3 0|10|10| 24 6 12 18 0|60|60| 25 7 13 19 1|65|65|
2 2 1 1 4|10|10| 0 3 4 3 0|10|10| 10 13 9 8 20|60|60| 11 14 10 9 21|65|65|
3 0 4 0 3|10|10| 2 2 1 1 4|10|10| 17 2 21 1 19|60|60| 18 3 22 2 20|65|65|

/EU 797/H\10/10|      165/L\10/10| [Math]      \60/60|      6001#\65/65|
1 4 3 0 2|10|10| 0 3 2 4 1|10|10| 5 23 17 4 11|60|60| 6 24 18 5 12|65|65|
4 0 0 3 3|10|10| 4 2 3 0 1|10|10| 24 2 3 15 16|60|60| 25 3 4 16 17|65|65|
2 1 2 3 2|10|10| 0 1 2 3 4|10|10| 10 6 12 18 14|60|60| 11 7 13 19 15|65|65|
1 1 4 4 0|10|10| 3 4 1 2 0|10|10| 8 9 21 22 0|60|60| 9 10 22 23 1|65|65|
2 4 1 0 3|10|10| 3 0 2 1 4|10|10| 13 20 7 1 19|60|60| 14 21 8 2 20|65|65|

/EU 837/H\10/10|      1217/L\10/10| [Math]      \60/60|      6501#\65/65|
1 4 0 3 2|10|10| 2 4 3 0 1|10|10| 7 24 3 15 11|60|60| 8 25 4 16 12|65|65|
0 2 3 4 1|10|10| 2 0 4 3 1|10|10| 2 10 19 23 6|60|60| 3 11 20 24 7|65|65|
4 3 2 1 0|10|10| 0 1 2 3 4|10|10| 20 16 12 8 4|60|60| 21 17 13 9 5|65|65|
3 0 1 2 4|10|10| 3 1 0 4 2|10|10| 18 1 5 14 22|60|60| 19 2 6 15 23|65|65|
2 1 4 0 3|10|10| 3 4 1 0 2|10|10| 13 9 21 0 17|60|60| 14 10 22 1 18|65|65|

/EU 877/H\10/10|      2243/L\10/10| [Math]      \60/60|      7001#\65/65|
1 0 4 3 2|10|10| 4 2 0 3 1|10|10| 9 2 20 18 11|60|60| 10 3 21 19 12|65|65|
0 4 3 2 1|10|10| 0 3 1 4 2|10|10| 0 23 16 14 7|60|60| 1 24 17 15 8|65|65|
4 3 2 1 0|10|10| 1 4 2 0 3|10|10| 21 19 12 5 3|60|60| 22 20 13 6 4|65|65|
3 2 1 0 4|10|10| 2 0 3 1 4|10|10| 17 10 8 1 24|60|60| 18 11 9 2 25|65|65|
2 1 0 4 3|10|10| 3 1 4 2 0|10|10| 13 6 4 22 15|60|60| 14 7 5 23 16|65|65|

```

[Solution Counts = 7136] [OK!]

\*\* Calculated and Listed by Kanji Setsuda  
on Apr. 3, 2012 with MacOSX 10.7.2 & Xcode 4.2.1 \*\*

Why! Only 7136 solutions among 48544! We could really have made far more than the 32 solutions, but they are not enough and too few to reach the goal.

What did we fail to notice?

(Note: This list above proves to be the new, important solution set of "Euler Type" of objects, according to our new definition in 2012.)

3. We must step back to the starting point, the original fact.

Before all let's have got the list of ordinary solutions with the decomposed units by the 5-th Increment System. We must observe them very carefully.

\*\* Self-complementary Magic Squares of Order 5 made by Ordinary Method: \*\*  
\*\* Solution List with **Decompositions** by the 5th Increment Number System \*\*

1#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10																			
1	22	21	19	2	65	65	0	21	20	18	1	60	60	0	4	4	3	0	11	10	0	1	0	3	1	5	10
20	3	15	9	18	65	65	19	2	14	8	17	60	60	3	0	2	1	3	9	11	4	2	4	3	2	15	5
12	16	13	10	14	65	65	11	15	12	9	13	60	60	2	3	2	1	2	10	10	1	0	2	4	3	10	10
8	17	11	23	6	65	65	7	16	10	22	5	60	60	1	3	2	4	1	11	9	2	1	0	2	0	5	15
24	7	5	4	25	65	65	23	6	4	3	24	60	60	4	1	0	0	4	9	10	3	1	4	3	4	15	10
397#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10																			
1	22	21	18	3	65	65	0	21	20	17	2	60	60	0	4	4	3	0	11	10	0	1	0	2	2	5	10
20	2	15	9	19	65	65	19	1	14	8	18	60	60	3	0	2	1	3	9	11	4	1	4	3	3	15	5
14	16	13	10	12	65	65	13	15	12	9	11	60	60	2	3	2	1	2	10	10	3	0	2	4	1	10	10
7	17	11	24	6	65	65	6	16	10	23	5	60	60	1	3	2	4	1	11	9	1	1	0	3	0	5	15
23	8	5	4	25	65	65	22	7	4	3	24	60	60	4	1	0	0	4	9	10	2	2	4	3	4	15	10
1241#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10																			
1	23	17	20	4	65	65	0	22	16	19	3	60	60	0	4	3	3	0	10	11	0	2	1	4	3	10	5
21	2	15	8	19	65	65	20	1	14	7	18	60	60	4	0	2	1	3	10	11	0	1	4	2	3	10	5
14	16	13	10	12	65	65	13	15	12	9	11	60	60	2	3	2	1	2	10	10	3	0	2	4	1	10	10
7	18	11	24	5	65	65	6	17	10	23	4	60	60	1	3	2	4	0	10	9	1	2	0	3	4	10	15
22	6	9	3	25	65	65	21	5	8	2	24	60	60	4	1	1	0	4	10	9	1	0	3	2	4	10	15
1757#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10																			
1	22	17	20	5	65	65	0	21	16	19	4	60	60	0	4	3	3	0	10	11	0	1	1	4	4	10	5
23	2	15	7	18	65	65	22	1	14	6	17	60	60	4	0	2	1	3	10	11	2	1	4	1	2	10	5
12	16	13	10	14	65	65	11	15	12	9	13	60	60	2	3	2	1	2	10	10	1	0	2	4	3	10	10
8	19	11	24	3	65	65	7	18	10	23	2	60	60	1	3	2	4	0	10	9	2	3	0	3	2	10	15
21	6	9	4	25	65	65	20	5	8	3	24	60	60	4	1	1	0	4	10	9	0	0	3	3	4	10	15
2585#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10																			
1	23	16	19	6	65	65	0	22	15	18	5	60	60	0	4	3	3	1	11	10	0	2	0	3	0	5	10
22	2	12	8	21	65	65	21	1	11	7	20	60	60	4	0	2	1	4	11	10	1	1	1	2	0	5	10
17	15	13	11	9	65	65	16	14	12	10	8	60	60	3	2	2	2	1	10	10	1	4	2	0	10	10	
5	18	14	24	4	65	65	4	17	13	23	3	60	60	0	3	2	4	0	9	10	4	2	3	3	3	15	10
20	7	10	3	25	65	65	19	6	9	2	24	60	60	3	1	1	0	4	9	10	4	1	4	2	4	15	10
3117#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10																			
1	22	17	18	7	65	65	0	21	16	17	6	60	60	0	4	3	3	1	11	11	0	1	1	2	1	5	5
23	2	15	5	20	65	65	22	1	14	4	19	60	60	4	0	2	0	3	9	11	2	1	4	4	4	15	5
16	12	13	14	10	65	65	15	11	12	13	9	60	60	3	2	2	2	1	10	10	0	1	2	3	4	10	10
6	21	11	24	3	65	65	5	20	10	23	2	60	60	1	4	2	4	0	11	9	0	0	0	3	2	5	15
19	8	9	4	25	65	65	18	7	8	3	24	60	60	3	1	1	0	4	9	9	3	2	3	3	4	15	15
3977#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10																			
1	21	15	20	8	65	65	0	20	14	19	7	60	60	0	4	2	3	1	10	11	0	0	4	4	2	10	5
23	2	17	4	19	65	65	22	1	16	3	18	60	60	4	0	3	0	3	10	11	2	1	1	3	3	10	5
16	14	13	12	10	65	65	15	13	12	11	9	60	60	3	2	2	2	1	10	10	0	3	2	1	4	10	10
7	22	9	24	3	65	65	6	21	8	23	2	60	60	1	4	1	4	0	10	9	1	1	3	3	2	10	15
18	6	11	5	25	65	65	17	5	10	4	24	60	60	3	1	2	0	4	10	9	2	0	0	4	4	10	15
4629#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10																			
1	23	14	18	9	65	65	0	22	13	17	8	60	60	0	4	2	3	1	10	11	0	2	3	2	3	10	5
22	2	16	5	20	65	65	21	1	15	4	19	60	60	4	0	3	0	3	10	11	1	1	0	4	4	10	5
19	11	13	15	7	65	65	18	10	12	14	6	60	60	3	2	2	2	1	10	10	3	0	2	4	1	10	10
6	21	10	24	4	65	65	5	20	9	23	3	60	60	1	4	1	4	0	10	9	0	0	4	3	3	10	15
17	8	12	3	25	65	65	16	7	11	2	24	60	60	3	1	2	0	4	10	9	1	2	1	2	4	10	15
5689#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10																			
1	21	15	18	10	65	65	0	20	14	17	9	60	60	0	4	2	3	1	10	10	0	0	4	2	4	10	10
19	2	23	4	17	65	65	18	1	22	3	16	60	60	3	0	4	0	3	10	11	3	1	2	3	1	10	5
20	12	13	14	6	65	65	19	11	12	13	5	60	60	3	2	2	2	1	10	10	4	1	2	3	0	10	10
9	22	3	24	7	65	65	8	21	2	23	6	60	60	1	4	0	4	1	10	9	3	1	2	3	1	10	15
16	8	11	5	25	65	65	15	7	10	4	24	60	60	3	1	2	0	4	10	10	0	2	0	4	4	10	10
6221#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10																			
1	22	17	14	11	65	65	0	21	16	13	10	60	60	0	4	3	2	2	11	9	0	1	1	3	0	5	15
19	2	23	5	16	65	65	18	1	22	4	15	60	60	3	0	4	0	3	10	11	3	1	2	4	0	10	5
20	8	13	18	6	65	65	19	7	12	17	5	60	60	3	1	2	3	1	10	10	4	2	2	2	0	10	10
10	21	3	24	7	65	65	9	20	2	23	6	60	60	1	4	0	4	1	10	9	4	0	2	3	1	10	15
15	12	9	4	25	65	65	14	11	8	3	24	60	60	2	2	1	0	4	9	11	4	1	3	3	4	15	5

7157#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10											
1	19	18	15	12 65 65	0	18	17	14	11 60 60	0	3	3	2	2 10 10	0	3	2	4	1 10 10
21	2	22	3	17 65 65	20	1	21	2	16 60 60	4	0	4	0	3 11 10	0	1	1	2	1 5 10
20	10	13	16	6 65 65	19	9	12	15	5 60 60	3	1	2	3	1 10 10	4	4	2	0	0 10 10
9	23	4	24	5 65 65	8	22	3	23	4 60 60	1	4	0	4	0 9 10	3	2	3	3	4 15 10
14	11	8	7	25 65 65	13	10	7	6	24 60 60	2	2	1	1	4 10 10	3	0	2	1	4 10 10
7793#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10											
2	21	22	17	3 65 65	1	20	21	16	2 60 60	0	4	4	3	0 11 10	1	0	1	1	2 5 10
19	1	18	12	15 65 65	18	0	17	11	14 60 60	3	0	3	2	2 10 10	3	0	2	1	4 10 10
10	20	13	6	16 65 65	9	19	12	5	15 60 60	1	3	2	1	3 10 10	4	4	2	0	0 10 10
11	14	8	25	7 65 65	10	13	7	24	6 60 60	2	2	1	4	1 10 10	0	3	2	4	1 10 10
23	9	4	5	24 65 65	22	8	3	4	23 60 60	4	1	0	0	4 9 10	2	3	3	4	3 15 10
14601#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10											
3	24	15	19	4 65 65	2	23	14	18	3 60 60	0	4	2	3	0 9 10	2	3	4	3	3 15 10
20	1	21	9	14 65 65	19	0	20	8	13 60 60	3	0	4	1	2 10 11	4	0	0	3	3 10 5
8	16	13	10	18 65 65	7	15	12	9	17 60 60	1	3	2	1	3 10 10	2	0	2	4	2 10 10
12	17	5	25	6 65 65	11	16	4	24	5 60 60	2	3	0	4	1 10 9	1	1	4	4	0 10 15
22	7	11	2	23 65 65	21	6	10	1	22 60 60	4	1	2	0	4 11 10	1	1	0	1	2 5 10
21805#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10											
4	23	16	17	5 65 65	3	22	15	16	4 60 60	0	4	3	3	0 10 10	3	2	0	1	4 10 10
15	1	24	6	19 65 65	14	0	23	5	18 60 60	2	0	4	1	3 10 10	4	0	3	0	3 10 10
18	12	13	14	8 65 65	17	11	12	13	7 60 60	3	2	2	2	1 10 10	2	1	2	3	2 10 10
7	20	2	25	11 65 65	6	19	1	24	10 60 60	1	3	0	4	2 10 10	1	4	1	4	0 10 10
21	9	10	3	22 65 65	20	8	9	2	21 60 60	4	1	1	0	4 10 10	0	3	4	2	1 10 10
27597#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10											
5	19	17	18	6 65 65	4	18	16	17	5 60 60	0	3	3	3	1 10 10	4	3	1	2	0 10 10
22	1	15	3	24 65 65	21	0	14	2	23 60 60	4	0	2	0	4 10 10	1	0	4	2	3 10 10
16	14	13	12	10 65 65	15	13	12	11	9 60 60	3	2	2	2	1 10 10	0	3	2	1	4 10 10
2	23	11	25	4 65 65	1	22	10	24	3 60 60	0	4	2	4	0 10 10	1	2	0	4	3 10 10
20	8	9	7	21 65 65	19	7	8	6	20 60 60	3	1	1	1	4 10 10	4	2	3	1	0 10 10
33193#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10											
6	22	12	18	7 65 65	5	21	11	17	6 60 60	1	4	2	3	1 11 11	0	1	1	2	1 5 5
21	1	16	3	24 65 65	20	0	15	2	23 60 60	4	0	3	0	4 11 11	0	0	0	2	3 5 5
17	11	13	15	9 65 65	16	10	12	14	8 60 60	3	2	2	2	1 10 10	1	0	2	4	3 10 10
2	23	10	25	5 65 65	1	22	9	24	4 60 60	0	4	1	4	0 9 9	1	2	4	4	4 15 15
19	8	14	4	20 65 65	18	7	13	3	19 60 60	3	1	2	0	3 9 9	3	2	3	3	4 15 15
37313#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10											
7	22	16	12	8 65 65	6	21	15	11	7 60 60	1	4	3	2	1 11 10	1	1	0	1	2 5 10
20	1	24	3	17 65 65	19	0	23	2	16 60 60	3	0	4	0	3 10 10	4	0	3	2	1 10 10
11	5	13	21	15 65 65	10	4	12	20	14 60 60	2	0	2	4	2 10 10	0	4	2	0	4 10 10
9	23	2	25	6 65 65	8	22	1	24	5 60 60	1	4	0	4	1 10 10	3	2	1	4	0 10 10
18	14	10	4	19 65 65	17	13	9	3	18 60 60	3	2	1	0	3 9 10	2	3	4	3	3 15 10
41081#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10											
8	23	10	15	9 65 65	7	22	9	14	8 60 60	1	4	1	2	1 9 11	2	2	4	4	3 15 5
21	1	22	2	19 65 65	20	0	21	1	18 60 60	4	0	4	0	3 11 11	0	0	1	1	3 5 5
12	6	13	20	14 65 65	11	5	12	19	13 60 60	2	1	2	3	2 10 10	1	0	2	4	3 10 10
7	24	4	25	5 65 65	6	23	3	24	4 60 60	1	4	0	4	0 9 9	1	3	3	4	4 15 15
17	11	16	3	18 65 65	16	10	15	2	17 60 60	3	2	3	0	3 11 9	1	0	0	2	2 5 15
44129#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10											
9	23	8	15	10 65 65	8	22	7	14	9 60 60	1	4	1	2	1 9 11	3	2	2	4	4 15 5
21	1	22	2	19 65 65	20	0	21	1	18 60 60	4	0	4	0	3 11 11	0	0	1	1	3 5 5
12	6	13	20	14 65 65	11	5	12	19	13 60 60	2	1	2	3	2 10 10	1	0	2	4	3 10 10
7	24	4	25	5 65 65	6	23	3	24	4 60 60	1	4	0	4	0 9 9	1	3	3	4	4 15 15
16	11	18	3	17 65 65	15	10	17	2	16 60 60	3	2	3	0	3 11 9	0	0	2	2	1 5 15
46533#\65/65				[Math]	\60/60	/D5i	H\10/10	L\10/10											
10	17	20	7	11 65 65	9	16	19	6	10 60 60	1	3	3	1	2 10 9	4	1	4	1	0 10 15
23	1	18	2	21 65 65	22	0	17	1	20 60 60	4	0	3	0	4 11 10	2	0	2	1	0 5 10
12	4	13	22	14 65 65	11	3	12	21	13 60 60	2	0	2	4	2 10 10	1	3	2	1	3 10 10
5	24	8	25	3 65 65	4	23	7	24	2 60 60	0	4	1	4	0 9 10	4	3	2	4	2 15 10
15	19	6	9	16 65 65	14	18	5	8	15 60 60	2	3	1	1	3 10 11	4	3	0	3	0 10 5

```

48065#\65/65| [Math]          \60/60| /D5i          H\10/10|          L\10/10|
11 21  4 17 12|65/65| 10 20  3 16 11|60/60|  2 4 0 3 2|11|10|  0 0 3 1 1| 5|10|
19  1 20  2 23|65/65| 18  0 19  1 22|60/60|  3 0 3 0 4|10|10|  3 0 4 1 2|10|10|
18 10 13 16  8|65/65| 17  9 12 15  7|60/60|  3 1 2 3 1|10|10|  2 4 2 0 2|10|10|
 3 24  6 25  7|65/65|  2 23  5 24  6|60/60|  0 4 1 4 1|10|10|  2 3 0 4 1|10|10|
14  9 22  5 15|65/65| 13  8 21  4 14|60/60|  2 1 4 0 2| 9|10|  3 3 1 4 4|15|10|
[Total Count of Solutions = 48544]

```

What can we find among those decomposed units?

A lot of irregular patterns are found in their rows and columns. The sums of four numbers on them are not always equal to the constant 10. They look like even the “usual” cases rather than the “exceptional” ones.

The next list explains what I have found about our decomposed units.

- (1) Watch the two primary diagonals and the third row and third column in the center, and find the sums of every four numbers on them are always equal to the constant 10. On top of that, the central number n13 of each unit always takes 2. No exceptions.

$$n1+n7+n13+n19+n25=10; \quad n5+n9+n13+n17+n21=10; \quad n3+n8+n13+n18+n23=10;$$

$$n11+n12+n13+n14+n15=10; \quad \text{and } n13=2;$$

- (2) Watch every pair of numbers in the symmetrical positions, and find that they always add up to the constant value 4. Since they make the ‘Complementary Pairs’ of 4, they are named as the “Self-complementary”. This is the same as the original objects.

Since n13 is placed symmetrically to itself,  $n13+n13=4$ .  $n13=2$  is always true.

The four equations of (1) consist of two pairs of complements and n13 as follows:

$$(n1+n25)+n13+(n7+n19)=4+2+4=10; \quad (n5+n21)+n13+(n9+n17)=4+2+4=10;$$

$$(n3+n23)+n13+(n8+n18)=4+2+4=10; \quad (n11+n15)+n13+(n12+n14)=4+2+4=10;$$

- (3) In the other 4 rows and 4 columns, the sums of 4 numbers are not always equal to the constant 10. They are sometimes equal to 9 or 11 as well as 10 in higher units, and they are also equal to 5 or 15 as well as 10 in lower units.

It seems there are some clear correspondences between the higher units and lower: When the sum is equal to 9 in high unit, the sum in low is always equal to 15. When the sum is equal to 10 in high, the sum in low is equal to 10. And when the sum is equal to 11 in high unit, the sum in low is equal to 5. But the other value combinations such as (8, 20) or (12, 0) cannot be found. What do they mean?

Watch the last example of the solution list above, and find the fifth row contains “21402” in high unit and “33144” in low.

$$(2+1+4+0+2) \times 5 + (3+3+1+4+4) = 9 \times 5 + 15 = 60;$$

This equation means the data structure of our object. The 5 numbers of the fifth row must add up to the constant sum 60 on the whole in our “Mathematical Expression”.

The value combination (9, 15) is all right. They don’t care about each unit, whether the sums of rows and columns are regularly equal to the constant value 10 or not.

What combinations of values are all right on the whole, then?

**\*\* Possible Combinations of Values between High and Low \*\***

$$[1] \quad 8 \times 5 + 20 = 60; \quad [2] \quad 9 \times 5 + 15 = 60;$$

$$[3] \quad 10 \times 5 + 10 = 60; \quad [4] \quad 11 \times 5 + 5 = 60;$$

$$[5] \quad 12 \times 5 + 0 = 60;$$

Theoretically speaking all the 5 cases above are possible, but you don’t have to care about the combinations (8, 20) and (12, 0). Both of them are practically impossible, since they cannot deny any duplication of number usage at all.

- (4) Any number of (0, 1, 2, 3 and 4) comes just 5 times as often as the others do in each unit. This is the same nature as the one of MS44, so called the ‘third rule’.

4. The next problem is how to apply these results of our observation. How can we rewrite our program with these new rules in order to reconstruct all the solutions of our object?

We can apply the rules (1), (2) and (4) in the same way just as we have done

before. Because there are no definite differences between higher unit and lower.

But how do we have to write the new conditions of the rule (3) in our program?

In high unit:  $8 < \text{Sum} < 12$ ; and in low:  $\text{Sum} = 5, 10, 15$ ;

We must divide the whole process into two separate procedures making higher units and lower, though only several steps actually need separation.

But it is a relief that we have to examine only the sums of 2 rows and 2 columns instead of 4 and 4, because the rest half are automatically determined by the 'Self-complementary Conditions' among 4 rows and 4 columns.

Read the next list of 'C' program I wrote for that purpose, please.

```
/** Self-complementary Magic Squares of Order 5 Reconstructed **
/** by the 5-th Increment System and "New Euler's Method" **
/** 'NEMS55SC.c': Dictated by Kanji Setsuda **
/** on Dec.18, 2011, Jan.25 and Feb. 7, 2012 **
/** Working with MacOSX 10.7.3 & Xcode 4.2.1 **
//
#include <stdio.h>
//
long int cnt, hcnt, lcnt, cnt2;
long lucnt[8];
long u1,u2;
short cnt3;
short nm[30], uflg[26];
short anm[8][76];
char htlu[39917][29];
char ltlu[30017][29];
char mtc[39917][30017];
short solt[48545][26];
long cntr[48545][3];
//
void hstp01(void), hstp02(void), hstp03(void), hstp04(void), hstp05(void);
void hstp06(void), hstp07(void), hstp08(void), hstp09(void), hstp10(void);
void hstp11(void), hstp12(void);
void hansrecord(void), prhunit(short x);
//
void lstp01(void), lstp02(void), lstp03(void), lstp04(void), lstp05(void);
void lstp06(void), lstp07(void), lstp08(void), lstp09(void), lstp10(void);
void lstp11(void), lstp12(void);
void lansrecord(void), prlunit(short x);
//
void tblmtc(void);
void cmbcmp(void), solsave(void);
void prans(short x), pr2ans(void), pr1ans(void);
//
/** Main Program *
int main(void){
short n;
printf("\n");
printf("** Self-complementary Magic Squares of Order 5 Reconstructed **\n");
printf(" ** by the 5-th Increment System and 'New Euler's Method' **\n");
for(n=0;n<5;n++){uflg[n]=0;}
nm[13]=2; uflg[2]=1;
cnt=0;
printf("\n");
printf(" ** List of Euler Units for High Positions by /D5i **\n");
hstp01(); // Make the Euler Units for High Positions
if(cnt3>0){prlunit(cnt3);} // Print the Rest One
hcnt=cnt; printf(" [Count of Euler Units for High Positions = %ld]\n",hcnt);
for(n=0;n<5;n++){uflg[n]=0;}
nm[13]=2; uflg[2]=1;
cnt=0; cnt3=0;
printf("\n");
```



```

printf(" ** List of Euler Units for Low Positions by /D5i **\n");
lstp01(); // Make the Euler Units for Low Positions
if(cnt3>0){prlunit(cnt3);} // Print the Rest One
lcnt=cnt; printf(" [Count of Euler Units for Low Positions = %ld]\n",lcnt);
tblmtc();
printf("\n");
printf("*** Standard Solutions of Self-complementary Magic Squares 5x5:\n");
printf(" Reconstructed by the 5th Increment System and 'New Euler's Method' **\n");
printf(" (Used_Units:High/Low/ Sol_Numb:[New]Old# Check_Sums:\\Diag/Row|Colmnl)\n");
cnt=0; cnt3=0;
cmbcmp(); // Combine and Compose
prans(1000); // Print the Solutions with that Interval
if(cnt3==1){pr1ans();} // Print the Rest One
printf(" [Solution Counts = %ld] [OK!]\n",cnt);
printf("\n");
printf("*** Calculated and listed by Kanji Setsuda\n");
printf(" on Jan.25, 2012 with MacOSX 10.7.2 & Xcode 4.2.1 **\n");
printf("\n");
return 0;
}
//
// Make the Euler Units for High Positions
// Level #1:
// Set N1 & n25 & N1<=n25
void hstp01(){
short i,j;
for(i=0;i<5;i++){j=4-i;
if((j>=i)&&(uflg[i]<5)&&(uflg[j]<5)){
nm[1]=i; nm[25]=j;
uflg[i]++; uflg[j]++;
hstp02();
uflg[i]--; uflg[j]--;}
}
}
// Set N5 & n21 & N1<=N5<=n21
void hstp02(){
short i,j;
for(i=nm[1];i<5;i++){j=4-i;
if((j>=i)&&(uflg[i]<5)&&(uflg[j]<5)){
nm[5]=i; nm[21]=j;
uflg[i]++; uflg[j]++;
hstp03();
uflg[i]--; uflg[j]--;}
}
}
// Set N7 & n19
void hstp03(){
short i,j,sm;
for(i=0;i<5;i++){j=4-i;
if((uflg[i]<5)&&(uflg[j]<5)){cnt2=0;
nm[7]=i; nm[19]=j;
uflg[i]++; uflg[j]++;
sm=nm[1]+nm[7]+nm[13]+nm[19]+nm[25];
if(sm==10){hstp04();}
uflg[i]--; uflg[j]--;}
}
}
// Set N9 & n17
void hstp04(){
short i,j,sm;
for(i=0;i<5;i++){j=4-i;
if((uflg[i]<5)&&(uflg[j]<5)){
nm[9]=i; nm[17]=j;
uflg[i]++; uflg[j]++;
sm=nm[5]+nm[9]+nm[13]+nm[17]+nm[21];
}
}
}

```

```

        if(sm==10){hstp05();}
        uflg[i]--; uflg[j]--;}
    }
}
// Level #2:
// Set N2 & n24
void hstp05(){
    short i,j;
    for(i=4;i>=0;i--){j=4-i;
        if((uflg[i]<5)&&(uflg[j]<5)){
            nm[2]=i; nm[24]=j;
            uflg[i]++; uflg[j]++;
            hstp06();
            uflg[i]--; uflg[j]--;}
        }
    }
// Set N4 & n22
void hstp06(){
    short i,j;
    for(i=0;i<5;i++){j=4-i;
        if((uflg[i]<5)&&(uflg[j]<5)){
            nm[4]=i; nm[22]=j;
            uflg[i]++; uflg[j]++;
            hstp07();
            uflg[i]--; uflg[j]--;}
        }
    }
// Set N3 & n23
void hstp07(){
    short i,j,sm1,sm2;
    for(i=0;i<5;i++){j=4-i;
        if((uflg[i]<5)&&(uflg[j]<5)){
            nm[3]=i; nm[23]=j;
            uflg[i]++; uflg[j]++;
            sm1=nm[1]+nm[2]+nm[3]+nm[4]+nm[5];
            sm2=nm[25]+nm[24]+nm[23]+nm[22]+nm[21];
            if((8<sm1)&&(sm1<12)&&(sm1+sm2==20)){nm[26]=sm1; hstp08();}
            uflg[i]--; uflg[j]--;}
        }
    }
// Set N8 & n18
void hstp08(){
    short i,j,sm;
    for(i=0;i<5;i++){j=4-i;
        if((uflg[i]<5)&&(uflg[j]<5)){
            nm[8]=i; nm[18]=j;
            uflg[i]++; uflg[j]++;
            sm=nm[3]+nm[8]+nm[13]+nm[18]+nm[23];
            if(sm==10){hstp09();}
            uflg[i]--; uflg[j]--;}
        }
    }
// Set N12 & n14
void hstp09(){
    short i,j,sm1,sm2;
    for(i=0;i<5;i++){j=4-i;
        if((uflg[i]<5)&&(uflg[j]<5)){
            nm[12]=i; nm[14]=j;
            uflg[i]++; uflg[j]++;
            sm1=nm[2]+nm[7]+nm[12]+nm[17]+nm[22];
            sm2=nm[24]+nm[19]+nm[14]+nm[9]+nm[4];
            if((8<sm1)&&(sm1<12)&&(sm1+sm2==20)){nm[28]=sm1; hstp10();}
            uflg[i]--; uflg[j]--;}
        }
    }
}

```

```

// Set N6 & n20
void hstp10(){
  short i,j;
  for(i=0;i<5;i++){j=4-i;
    if((uflg[i]<5)&&(uflg[j]<5)){
      nm[6]=i; nm[20]=j;
      uflg[i]++; uflg[j]++;
      hstp11();
      uflg[i]--; uflg[j]--;}
  }
}
// Set N10 & n16
void hstp11(){
  short i,j,sm1,sm2;
  for(i=0;i<5;i++){j=4-i;
    if((uflg[i]<5)&&(uflg[j]<5)){
      nm[10]=i; nm[16]=j;
      uflg[i]++; uflg[j]++;
      sm1=nm[6]+nm[7]+nm[8]+nm[9]+nm[10];
      sm2=nm[20]+nm[19]+nm[18]+nm[17]+nm[16];
      if((8<sm1)&&(sm1<12)&&(sm1+sm2==20)){nm[29]=sm1; hstp12();}
      uflg[i]--; uflg[j]--;}
  }
}
// Set N11 & n15
void hstp12(){
  short i,j,sm1,sm2,sm3;
  for(i=0;i<5;i++){j=4-i;
    if((uflg[i]<5)&&(uflg[j]<5)){
      nm[11]=i; nm[15]=j;
      uflg[i]++; uflg[j]++;
      sm1=nm[1]+nm[6]+nm[11]+nm[16]+nm[21];
      sm2=nm[25]+nm[20]+nm[15]+nm[10]+nm[5];
      sm3=nm[11]+nm[12]+nm[13]+nm[14]+nm[15];
      if((8<sm1)&&(sm1<12)&&(sm1+sm2==20)){
        if(sm3==10){nm[27]=sm1; hansrecord();}}
      uflg[i]--; uflg[j]--;}
  }
}
//
// Record the High Units
void hansrecord(){
  short n;
  for(n=0;n<29;n++){htlu[cnt][n]=nm[n+1];}
  cnt++; cnt2++;
  if(cnt2==1){lucnt[cnt3]=cnt;
    for(n=0;n<25;n++){anm[cnt3][n]=nm[n+1];}
    cnt3++; if(cnt3==7){prlunit(cnt3); cnt3=0;}
  }
}
//
// Make the Euler Units for Low Positions
// Level #1:
// Set N1 & n25
void lstp01(){
  short i,j;
  for(i=0;i<5;i++){j=4-i;
    if((uflg[i]<5)&&(uflg[j]<5)){
      nm[1]=i; nm[25]=j;
      uflg[i]++; uflg[j]++;
      lstp02();
      uflg[i]--; uflg[j]--;}
  }
}
// Set N5 & n21

```

```

void lstp02(){
  short i,j;
  for(i=0;i<5;i++){j=4-i;
    if((uflg[i]<5)&&(uflg[j]<5)){cnt2=0;
      nm[5]=i; nm[21]=j;
      uflg[i]++; uflg[j]++;
      lstp03();
      uflg[i]--; uflg[j]--;}
  }
}
// Set N7 & n19
void lstp03(){
  short i,j,sm;
  for(i=0;i<5;i++){j=4-i;
    if((uflg[i]<5)&&(uflg[j]<5)){if(nm[1]==0){cnt2=0;}
      nm[7]=i; nm[19]=j;
      uflg[i]++; uflg[j]++;
      sm=nm[1]+nm[7]+nm[13]+nm[19]+nm[25];
      if(sm==10){lstp04();}
      uflg[i]--; uflg[j]--;}
  }
}
// Set N9 & n17
void lstp04(){
  short i,j,sm;
  for(i=0;i<5;i++){j=4-i;
    if((uflg[i]<5)&&(uflg[j]<5)){
      nm[9]=i; nm[17]=j;
      uflg[i]++; uflg[j]++;
      sm=nm[5]+nm[9]+nm[13]+nm[17]+nm[21];
      if(sm==10){lstp05();}
      uflg[i]--; uflg[j]--;}
  }
}
// Level #2:
// Set N2 & n24
void lstp05(){
  short i,j;
  for(i=4;i>=0;i--){j=4-i;
    if((uflg[i]<5)&&(uflg[j]<5)){
      nm[2]=i; nm[24]=j;
      uflg[i]++; uflg[j]++;
      lstp06();
      uflg[i]--; uflg[j]--;}
  }
}
// Set N4 & n22
void lstp06(){
  short i,j;
  for(i=0;i<5;i++){j=4-i;
    if((uflg[i]<5)&&(uflg[j]<5)){
      nm[4]=i; nm[22]=j;
      uflg[i]++; uflg[j]++;
      lstp07();
      uflg[i]--; uflg[j]--;}
  }
}
// Set N3 & n23
void lstp07(){
  short i,j,sm1,sm2;
  for(i=0;i<5;i++){j=4-i;
    if((uflg[i]<5)&&(uflg[j]<5)){
      nm[3]=i; nm[23]=j;
      uflg[i]++; uflg[j]++;
      sm1=nm[1]+nm[2]+nm[3]+nm[4]+nm[5];

```

```

        sm2=nm[25]+nm[24]+nm[23]+nm[22]+nm[21];
        if((sm1==5)|| (sm1==10)|| (sm1==15)){
            if(sm1+sm2==20){nm[26]=sm1; lstp08();}}
        uflg[i]--; uflg[j]--;}
    }
}
// Set N8 & n18
void lstp08(){
    short i,j,sm;
    for(i=0;i<5;i++){j=4-i;
        if((uflg[i]<5)&&(uflg[j]<5)){
            nm[8]=i; nm[18]=j;
            uflg[i]++; uflg[j]++;
            sm=nm[3]+nm[8]+nm[13]+nm[18]+nm[23];
            if(sm==10){lstp09();}
            uflg[i]--; uflg[j]--;}
        }
}
// Set N12 & n14
void lstp09(){
    short i,j,sm1,sm2;
    for(i=0;i<5;i++){j=4-i;
        if((uflg[i]<5)&&(uflg[j]<5)){
            nm[12]=i; nm[14]=j;
            uflg[i]++; uflg[j]++;
            sm1=nm[2]+nm[7]+nm[12]+nm[17]+nm[22];
            sm2=nm[24]+nm[19]+nm[14]+nm[9]+nm[4];
            if((sm1==5)|| (sm1==10)|| (sm1==15)){
                if(sm1+sm2==20){nm[28]=sm1; lstp10();}}
            uflg[i]--; uflg[j]--;}
        }
}
// Set N6 & n20
void lstp10(){
    short i,j;
    for(i=0;i<5;i++){j=4-i;
        if((uflg[i]<5)&&(uflg[j]<5)){
            nm[6]=i; nm[20]=j;
            uflg[i]++; uflg[j]++;
            lstp11();
            uflg[i]--; uflg[j]--;}
        }
}
// Set N10 & n16
void lstp11(){
    short i,j,sm1,sm2;
    for(i=0;i<5;i++){j=4-i;
        if((uflg[i]<5)&&(uflg[j]<5)){
            nm[10]=i; nm[16]=j;
            uflg[i]++; uflg[j]++;
            sm1=nm[6]+nm[7]+nm[8]+nm[9]+nm[10];
            sm2=nm[20]+nm[19]+nm[18]+nm[17]+nm[16];
            if((sm1==5)|| (sm1==10)|| (sm1==15)){
                if(sm1+sm2==20){nm[29]=sm1; lstp12();}}
            uflg[i]--; uflg[j]--;}
        }
}
// Set N11 & n15
void lstp12(){
    short i,j,sm1,sm2,sm3;
    for(i=0;i<5;i++){j=4-i;
        if((uflg[i]<5)&&(uflg[j]<5)){
            nm[11]=i; nm[15]=j;
            uflg[i]++; uflg[j]++;
            sm1=nm[1]+nm[6]+nm[11]+nm[16]+nm[21];

```

```

        sm2=nm[25]+nm[20]+nm[15]+nm[10]+nm[5];
        sm3=nm[11]+nm[12]+nm[13]+nm[14]+nm[15];
        if((sm1==5)|| (sm1==10)|| (sm1==15)){
            if((sm1+sm2==20)&&(sm3==10)){nm[27]=sm1; lansrecord();}}
        uflg[i]--; uflg[j]--;}
    }
}
//
// Record Low Units
void lansrecord(){
    short n;
    for(n=0;n<29;n++){ltlu[cnt][n]=nm[n+1];}
    cnt++; cnt2++;
    if(cnt2==1){lucnt[cnt3]=cnt;
        for(n=0;n<25;n++){anm[cnt3][n]=nm[n+1];}
        cnt3++; if(cnt3==7){prlunit(cnt3); cnt3=0;}
    }
}
//
// Print Euler Units
void prlunit(short x){
    short l,m,m5,n;
    for(l=0;l<x;l++){printf(" %9ld/", lucnt[l]);}
    printf("\n");
    for(m=0;m<5;m++){m5=m*5;
        for(l=0;l<x;l++){printf(" ");
            for(n=0;n<5;n++){printf("%2d", anm[l][m5+n]);}}
        printf("\n");
    }
}
//
// Check how well they match and Save the Data to the Table
void tblmtc(){
    short h,l;
    short n,d1,d2;
    for(h=0;h<ecnt;h++){
        for(l=0;l<ecnt;l++){
            d1=0; for(n=1;n<26;n++){if(teu[h][n]==teu[l][n]){d1++;}}
            d2=0; for(n=1;n<26;n++){if(teu[h][n]+teu[l][n]==SC){d2++;}}
            if(d1<d2){d1=d2;}
            mtc[h][l]=d1;}}
}
//
// * Combine, Compose and Print *
void cmbcmp(){
    short n,d,fc;
    short s1,s2,s3,s4,s5,s6,s7,s8;
    for(u1=0;u1<hcnt;u1++){
        s1=htlu[u1][25]; s2=htlu[u1][26]; s3=htlu[u1][27]; s4=htlu[u1][28];
        if((8<s1)&&(s1<12)&&(8<s2)&&(s2<12)&&(8<s3)&&(s3<12)&&(8<s4)&&(s4<12)){cnt2=0;
            for(u2=0;u2<lcnt;u2++){
                s5=ltlu[u2][25]; s6=ltlu[u2][26]; s7=ltlu[u2][27]; s8=ltlu[u2][28];
                if((s1*5+s5==60)&&(s2*5+s6==60)&&(s3*5+s7==60)&&(s4*5+s8==60)){
                    if(mtc[u1][u2]==5){
                        for(n=1;n<26;n++){uflg[n]=0;}
                        for(n=0;n<25;n++){
                            d=htlu[u1][n]*5+ltlu[u2][n]+1;
                            nm[n+1]=d; uflg[d]++;
                        }
                        fc=0;
                        for(n=1;n<26;n++){if(uflg[n]==0){fc++;}}
                        if(fc==0){
                            if((nm[1]<nm[5])&&(nm[5]<nm[21])&&(nm[1]<nm[25])){solsave();}}
                    }
                }
            }
        }
    }
}
}

```

```

    }}
}
//
// Save Answers to the Solution List
void solsave(){
    short n;
    for(n=1;n<26;n++){solt[cnt][n]=nm[n];}
    cntr[cnt][1]=u1;  cntr[cnt][2]=u2;
    cnt++;
}
//
// Print the Solution List
void prans(short x){
    long s;
    short m,n,sm1,sm2;
    for(s=0;s<cnt;s=s+x){
        lucnt[cnt3*4]=s+1;
        for(n=1;n<26;n++){nm[n]=solt[s][n]; anm[cnt3][n]=nm[n]; anm[cnt3+1][n]=0;}
        for(m=0;m<5;m++){sm1=0; sm2=0;
            for(n=0;n<5;n++){sm1=sm1+nm[m*5+n+1]; sm2=sm2+nm[n*5+m+1];}
            anm[cnt3][m+26]=sm1; anm[cnt3][m+31]=sm2;}
        anm[cnt3][36]=nm[1]+nm[7]+nm[13]+nm[19]+nm[25];
        anm[cnt3][37]=nm[5]+nm[9]+nm[13]+nm[17]+nm[21];
        lucnt[cnt3*4+1]=cntr[s][1]+1; lucnt[cnt3*4+2]=cntr[s][2]+1;
        cnt3++;
        if(cnt3==2){pr2ans(); cnt3=0;}
    }
}
//
// Print 2 Answers with Sum Checks
void pr2ans(){
    long lu1,lu2;
    short l,l5,m,n;
    printf("%6ld/H%6ld/L%15ld#  \\%2d/%2d| %6ld/H%6ld/L%15ld#  \\%2d/%2d|\n",
        lucnt[1],lucnt[2],lucnt[0],anm[0][36],anm[0][37],
        lucnt[5],lucnt[6],lucnt[4],anm[1][36],anm[1][37]);
    for(l=0;l<5;l++){l5=l*5;
        for(m=0;m<2;m++){
            lu1=lucnt[m*4+1]-1; lu2=lucnt[m*4+2]-1;
            printf(" ");
            for(n=0;n<5;n++){printf("%d",htlu[lu1][l5+n]);}
            printf(" ");
            for(n=0;n<5;n++){printf("%d",ltlu[lu2][l5+n]);}
            printf(" ");
            for(n=1;n<6;n++){printf("%3d",anm[m][l5+n]);}
            printf(" |%2d|%2d|",anm[m][l+26],anm[m][l+31]);
            if(m<1){printf(" ");}
        }
        printf("\n");
    }
}
//
// Print the Rest One
void pr1ans(){
    long lu1,lu2;
    short l,l5,n;
    lu1=lucnt[1]-1; lu2=lucnt[2]-1;
    printf("%6ld/H%6ld/L%15ld#  \\%2d/%2d|\n",
        lu1+1,lu2+1,lucnt[0],anm[0][36],anm[0][37]);
    for(l=0;l<5;l++){l5=l*5;
        printf(" ");
        for(n=0;n<5;n++){printf("%d",htlu[lu1][l5+n]);}
        printf(" ");
        for(n=0;n<5;n++){printf("%d",ltlu[lu2][l5+n]);}
        printf(" ");
    }
}

```

```

for(n=1;n<6;n++){printf("%3d",anm[0][l5+n]);}
printf("  l%2dl%2dl\n",anm[0][l+26],anm[0][l+31]);
}
}

```

5. The next lists show the execution results of our program.

**\*\* Self-complementary Magic Squares of Order 5 Reconstructed \*\***  
**\*\* by the 5-th Increment System and 'New Euler's Method' \*\***

**\*\* List of Euler Units for High Positions by /D5i \*\***

1/ 0 4 3 2 0 3 0 3 0 3 2 1 2 3 2 1 4 1 4 1 4 2 1 0 4	682/ 0 4 3 2 0 3 1 2 0 3 1 0 2 4 3 1 4 2 3 1 4 2 1 0 4	2513/ 0 4 2 3 0 3 2 1 0 3 1 0 2 4 3 1 4 3 2 1 4 1 2 0 4	3929/ 0 3 3 3 0 2 3 2 0 3 4 0 2 4 0 1 4 2 1 2 4 1 1 1 4	5760/ 0 2 3 4 0 2 4 1 0 3 3 1 2 3 1 1 4 3 0 2 4 0 1 2 4	6441/ 0 4 2 2 1 3 0 3 0 3 3 0 2 4 1 1 4 1 4 1 3 2 2 0 4	8048/ 0 4 2 2 1 3 1 1 0 4 3 0 2 4 1 0 4 3 3 1 3 2 2 0 4
9774/ 0 4 1 3 1 4 2 1 0 2 1 0 2 4 3 2 4 3 2 0 3 1 3 0 4	11592/ 0 4 1 4 1 3 3 1 0 2 2 0 2 4 2 2 4 3 1 1 3 0 3 0 4	13318/ 0 3 1 4 1 2 4 1 0 2 3 0 2 4 1 2 4 3 0 2 3 0 3 1 4	14925/ 0 4 2 1 2 3 0 3 0 3 3 0 2 4 1 1 4 1 4 1 2 3 2 0 4	16299/ 0 4 1 2 2 4 1 1 0 3 3 0 2 4 1 1 4 3 3 0 2 2 3 0 4	18291/ 0 4 0 3 2 3 2 1 0 3 3 0 2 4 1 1 4 3 2 1 2 1 4 0 4	19051/ 0 4 1 4 2 3 3 1 0 2 3 0 2 4 1 2 4 3 1 1 2 0 3 0 4
21043/ 0 3 1 4 2 3 4 1 0 1 2 0 2 4 2 3 4 3 0 1 2 0 3 1 4	22417/ 1 4 2 1 1 3 0 2 0 4 3 0 2 4 1 0 4 2 4 1 3 3 2 0 3	24419/ 1 4 1 2 1 4 1 1 0 4 2 0 2 4 2 0 4 3 3 0 3 2 3 0 3	25389/ 1 4 0 3 1 2 2 1 0 4 3 0 2 4 1 0 4 3 2 2 3 1 4 0 3	26909/ 1 4 0 4 1 2 3 1 0 3 2 0 2 4 2 1 4 3 1 2 3 0 4 0 3	27879/ 1 3 0 4 1 2 4 1 0 2 1 0 2 4 3 2 4 3 0 2 3 0 4 1 3	29881/ 1 4 1 1 2 3 0 2 0 4 3 0 2 4 1 0 4 2 4 1 2 3 3 0 3
31769/ 1 4 0 2 2 3 1 1 0 4 3 0 2 4 1 0 4 3 3 1 2 2 4 0 3	33127/ 1 4 0 3 2 3 2 1 0 3 4 0 2 4 0 1 4 3 2 1 2 1 4 0 3	33855/ 1 4 0 4 2 2 3 1 0 3 3 0 2 4 1 1 4 3 1 2 2 0 4 0 3	35213/ 1 3 0 4 2 2 4 1 0 3 3 0 2 4 1 1 4 3 0 2 2 0 4 1 3	37101/ 2 4 0 1 2 3 0 3 0 3 1 0 2 4 3 1 4 1 4 1 2 3 4 0 2	37805/ 2 4 0 3 2 3 1 1 0 4 3 0 2 4 1 0 4 3 3 1 2 1 4 0 2	38509/ 2 3 0 3 2 3 3 0 0 4 3 0 2 4 1 0 4 4 1 1 2 1 4 1 2
39213/ 2 3 0 4 2 1 4 1 0 3 3 0 2 4 1 1 4 3 0 3 2 0 4 1 2	[Count of Euler Units for High Positions = 39916]					

**\*\* List of Euler Units for Low Positions by /D5i \*\***

1/ 0 4 3 3 0 1 0 2 0 2 3 1 2 3 1 2 4 2 4 3 4 1 1 0 4	165/ 0 4 1 0 0 1 1 1 0 2 3 2 2 2 1 2 4 3 3 3 4 4 3 0 4	438/ 0 4 1 0 0 1 2 1 0 1 2 1 2 3 2 3 4 3 2 3 4 4 3 0 4	668/ 0 4 0 1 0 2 3 2 0 3 3 1 2 3 1 1 4 2 1 2 4 3 4 0 4	941/ 0 4 3 3 0 2 4 1 0 3 3 2 2 2 1 1 4 3 0 2 4 1 1 0 4	1105/ 0 4 2 3 1 0 0 2 0 3 1 1 2 3 3 1 4 2 4 4 3 1 2 0 4	1401/ 0 4 0 0 1 1 1 2 0 1 3 2 2 2 1 3 4 2 3 3 3 4 4 0 4
1697/ 0 4 0 0 1 3 2 2 0 3 3 1 2 3 1 1 4 2 2 1 3 4 4 0 4	1985/ 0 4 4 1 1 2 3 2 0 3 4 1 2 3 0 1 4 2 1 2 3 3 0 0 4	2281/ 0 4 3 2 1 2 4 1 0 3 4 1 2 3 0 1 4 3 0 2 3 2 1 0 4	2577/ 0 4 3 1 2 3 0 1 0 1 2 4 2 0 2 3 4 3 4 1 2 3 1 0 4	2791/ 0 4 3 1 2 3 1 2 0 4 0 3 2 1 4 0 4 2 3 1 2 3 1 0 4	3080/ 0 4 1 3 2 1 2 1 0 1 4 4 2 0 0 3 4 3 2 3 2 1 3 0 4	3182/ 0 4 3 1 2 0 3 0 0 2 1 1 2 3 3 2 4 4 1 4 2 3 1 0 4
3471/ 0 4 3 1 2 3 4 1 0 2 3 0 2 4 1 2 4 3 0 1 2 3 1 0 4	3685/ 0 4 3 0 3 1 0 2 0 2 1 3 2 1 3 2 4 2 4 3 1 4 1 0 4	3981/ 0 4 3 0 3 1 1 0 0 3 2 2 2 2 2 1 4 4 3 3 1 4 1 0 4	4277/ 0 4 3 0 3 1 2 0 0 2 1 1 2 3 3 2 4 4 2 3 1 4 1 0 4	4565/ 0 4 3 0 3 2 3 2 0 3 1 0 2 4 3 1 4 2 1 2 1 4 1 0 4	4861/ 0 4 2 1 3 2 4 1 0 3 1 0 2 4 3 1 4 3 0 2 1 3 2 0 4	5157/ 0 3 1 2 4 3 0 2 0 0 3 1 2 3 1 4 4 2 4 1 0 2 3 1 4



5321/ 0 4 2 0 4 3 1 3 0 3 1 2 2 2 3 1 4 1 3 1 0 4 2 0 4	5594/ 0 4 2 0 4 1 2 1 0 1 1 1 2 3 3 3 4 3 2 3 0 4 2 0 4	5824/ 0 4 1 1 4 3 3 2 0 2 0 1 2 3 4 2 4 2 1 1 0 3 3 0 4	6097/ 0 3 2 1 4 3 4 1 0 2 0 1 2 3 4 2 4 3 0 1 0 3 2 1 4	6261/ 1 4 2 3 0 3 0 2 0 0 3 1 2 3 1 4 4 2 4 1 4 1 2 0 3	7733/ 1 4 4 0 1 1 0 2 0 2 3 3 2 1 1 2 4 2 4 3 3 4 0 0 3	8841/ 1 4 3 0 2 0 0 2 0 3 1 3 2 1 3 1 4 2 4 4 2 4 1 0 3
9945/ 1 4 2 0 3 3 0 0 0 2 3 3 2 1 1 2 4 4 4 1 1 4 2 0 3	11053/ 1 4 1 0 4 1 0 2 0 2 1 3 2 1 3 2 4 2 4 3 0 4 3 0 3	12525/ 2 4 3 1 0 1 0 1 0 3 2 4 2 0 2 1 4 3 4 3 4 3 1 0 2	13633/ 2 4 3 0 1 3 0 2 0 0 3 3 2 1 1 4 4 2 4 1 3 4 1 0 2	14737/ 2 4 1 1 2 1 0 1 0 3 4 4 2 0 0 1 4 3 4 3 2 3 3 0 2	15281/ 2 4 1 0 3 0 0 2 0 3 1 3 2 1 3 1 4 2 4 4 1 4 3 0 2	16385/ 2 4 3 2 4 1 0 1 0 3 1 0 2 4 3 1 4 3 4 3 0 2 1 0 2
17493/ 3 4 3 0 0 2 0 2 0 1 3 3 2 1 1 3 4 2 4 2 4 4 1 0 1	18965/ 3 4 2 0 1 2 0 0 0 3 1 3 2 1 3 1 4 4 4 2 3 4 2 0 1	20073/ 3 4 1 0 2 3 0 2 0 0 3 3 2 1 1 4 4 2 4 1 2 4 3 0 1	21177/ 3 4 0 0 3 1 0 2 0 2 3 3 2 1 1 2 4 2 4 3 1 4 4 0 1	22285/ 3 4 3 1 4 2 0 1 0 2 3 4 2 0 1 2 4 3 4 2 0 3 1 0 1	23757/ 4 3 1 2 0 0 0 2 0 3 1 1 2 3 3 1 4 2 4 4 4 2 3 1 0	24861/ 4 4 1 0 1 2 0 2 0 1 3 3 2 1 1 3 4 2 4 2 3 4 3 0 0
26333/ 4 4 3 2 2 3 0 1 0 1 3 0 2 4 1 3 4 3 4 1 2 2 1 0 0	27441/ 4 4 3 1 3 2 0 1 0 2 1 4 2 0 3 2 4 3 4 2 1 3 1 0 0	28913/ 4 3 3 1 4 1 0 2 0 2 3 0 2 4 1 2 4 2 4 3 0 3 1 1 0	[Count of Euler Units for Low = 30016]			

Here you are at the reconstruction results of solutions, only the abstract list.

\*\* Standard Solutions of Self-complementary Magic Squares of Order 5  
Reconstructed by the 5th Increment System and 'New Euler's Method' \*\*

(Used\_Units: High/Low/ Sol\_Number# Check\_Sums:\Diag/Row|Colmn|)

35/H 16810/L	1# \65/65	996/H 8970/L	1001# \65/65
04410 21444 3 22 25 10 5  65 65	04420 14032 2 25 21 14 3  65 65	30213 41433 20 2 15 9 19  65 65	31312 30124 19 6 17 8 15  65 65
21232 32221 14 8 13 18 12  65 65	10243 43210 10 4 13 22 16  65 65	13241 11030 7 17 11 24 6  65 65	23131 02341 11 18 9 20 7  65 65
43004 00032 21 16 1 4 23  65 65	42004 21403 23 12 5 1 24  65 65	1967/H 9144/L	2001# \65/65
04320 14442 2 25 20 15 3  65 65	03340 04123 1 20 17 23 4  65 65	31034 11300 17 7 4 16 21  65 65	42211 03421 21 14 15 8 7  65 65
32221 21232 18 12 13 14 8  65 65	01234 14203 2 10 13 16 24  65 65	01431 44133 5 10 22 19 9  65 65	33220 32014 19 18 11 12 5  65 65
42104 20003 23 11 6 1 24  65 65	40114 12304 22 3 9 6 25  65 65	3985/H 3382/L	4001# \65/65
03340 01432 1 17 20 24 3  65 65	04230 32334 4 23 14 19 5  65 65	23301 33234 14 19 18 4 10  65 65	13430 00320 6 16 24 18 1  65 65
20242 04204 11 5 13 21 15  65 65	12223 30241 9 11 13 15 17  65 65	34112 01211 16 22 8 7 12  65 65	41013 42144 25 8 2 10 20  65 65
40114 21034 23 2 6 9 25  65 65	41204 01121 21 7 12 3 22  65 65	6030/H 4562/L	6001# \65/65
03430 00343 1 16 24 20 4  65 65	04221 44034 5 25 11 14 10  65 65	24120 32244 14 23 8 15 5  65 65	40114 12133 22 3 7 9 24  65 65
11233 13213 7 9 13 17 19  65 65	31231 42220 20 8 13 18 6  65 65	42302 00221 21 11 18 3 12  65 65	03340 11323 2 17 19 23 4  65 65
41014 10144 22 6 2 10 25  65 65	32204 01400 16 12 15 1 21  65 65	7081/H 3137/L	8001# \65/65
02431 04042 1 15 21 20 8  65 65	04401 43431 5 24 25 4 7  65 65	40312 12331 22 3 19 9 12  65 65	20332 02143 11 3 17 20 14  65 65
14203 43210 10 24 13 2 16  65 65	31231 24202 18 10 13 16 8  65 65	23140 31123 14 17 7 23 4  65 65	21142 10324 12 6 9 23 15  65 65
31024 20404 18 6 5 11 25  65 65	34004 31010 19 22 1 2 21  65 65		

7917/H	29691/L		10001#	\65 65	8329/H	21468/L		11001#	\65 65
04221	44304	5 25 14 11 10		65 65	02331	33423	4 14 20 18 9		65 65
30043	13123	17 4 2 23 19		65 65	41401	21304	23 7 24 1 10		65 65
33211	42220	20 18 13 8 6		65 65	02224	40240	5 11 13 15 21		65 65
10441	12313	7 3 24 22 9		65 65	34030	04132	16 25 2 19 3		65 65
32204	04100	16 15 12 1 21		65 65	31124	12011	17 8 6 12 22		65 65
8811/H	3039/L		12001#	\65 65	9666/H	14433/L		14001#	\65 65
04411	03232	1 24 23 9 8		65 65	04331	20201	3 21 18 16 7		65 65
41023	11440	22 7 5 15 16		65 65	21240	43143	15 9 12 25 4		65 65
21232	30241	14 6 13 20 12		65 65	14203	03214	6 24 13 2 20		65 65
12430	40033	10 11 21 19 4		65 65	40232	10310	22 1 14 17 11		65 65
33004	21214	18 17 3 2 25		65 65	31104	34242	19 10 8 5 23		65 65
10563/H	9520/L		16001#	\65 65	11547/H	29635/L		18001#	\65 65
01441	13042	2 9 21 25 8		65 65	03141	44124	5 20 7 23 10		65 65
22411	43210	15 14 23 7 6		65 65	22340	03110	11 14 17 22 1		65 65
14203	41230	10 22 13 4 16		65 65	14203	23212	8 24 13 2 18		65 65
33022	43210	20 19 3 12 11		65 65	40122	43314	25 4 9 12 15		65 65
30034	20413	18 1 5 17 24		65 65	30314	02300	16 3 19 6 21		65 65
12444/H	3186/L		20001#	\65 65	13276/H	6377/L		22001#	\65 65
03241	04132	1 20 12 24 8		65 65	02431	14410	2 15 25 17 6		65 65
43120	03403	21 19 10 11 4		65 65	23140	30124	14 16 7 23 5		65 65
01234	23212	3 9 13 17 23		65 65	14203	21232	8 22 13 4 18		65 65
42310	14014	22 15 16 7 5		65 65	40312	02341	21 3 19 10 12		65 65
30214	21304	18 2 14 6 25		65 65	31024	43003	20 9 1 11 24		65 65
14159/H	27831/L		24001#	\65 65	14969/H	19851/L		26001#	\65 65
01431	42243	5 8 23 20 9		65 65	03322	34201	4 20 18 11 12		65 65
04321	11314	2 22 19 12 10		65 65	40303	24013	23 5 16 2 19		65 65
42220	44200	25 15 13 11 1		65 65	30241	10243	17 1 13 25 9		65 65
32104	03133	16 14 7 4 24		65 65	14140	13402	7 24 10 21 3		65 65
31034	10220	17 6 3 18 21		65 65	22114	34201	14 15 8 6 22		65 65
17023/H	19603/L		28001#	\65 65	19030/H	670/L		30001#	\65 65
03242	31001	4 17 11 21 12		65 65	03332	03200	1 19 18 16 11		65 65
31014	42243	20 8 3 10 24		65 65	42140	13001	22 14 6 21 2		65 65
43210	43210	25 19 13 7 1		65 65	03214	21232	3 17 13 9 23		65 65
03431	10220	2 16 23 18 6		65 65	40320	34413	24 5 20 12 4		65 65
20214	34431	14 5 15 9 22		65 65	21114	44214	15 10 8 7 25		65 65
21250/H	12664/L		32001#	\65 65	23196/H	16658/L		34001#	\65 65
03232	23140	3 19 12 20 11		65 65	12331	20134	8 11 17 19 10		65 65
34310	20030	18 21 16 9 1		65 65	40412	11201	22 2 23 6 12		65 65
00244	31231	4 2 13 24 22		65 65	04204	44200	5 25 13 1 21		65 65
43101	41442	25 17 10 5 8		65 65	23040	34233	14 20 3 24 4		65 65
21214	40312	15 6 14 7 23		65 65	31123	01342	16 7 9 15 18		65 65
25616/H	9803/L		36001#	\65 65	27975/H	12121/L		38001#	\65 65
12431	11102	7 12 22 16 8		65 65	12141	14244	7 15 8 25 10		65 65
12403	04321	6 15 24 3 17		65 65	34201	13330	17 24 14 4 6		65 65
40240	44200	25 5 13 21 1		65 65	00244	42220	5 3 13 23 21		65 65
14023	32104	9 23 2 11 20		65 65	34201	41113	20 22 12 2 9		65 65
31023	24333	18 10 4 14 19		65 65	30323	00203	16 1 18 11 19		65 65
30109/H	7473/L		40001#	\65 65	31815/H	17520/L		42001#	\65 65
13142	11210	7 17 8 22 11		65 65	14132	31100	9 22 7 16 11		65 65
30304	04402	16 5 20 1 23		65 65	41402	00203	21 6 23 1 14		65 65
42220	33211	24 14 13 12 2		65 65	10243	21232	8 2 13 24 18		65 65
04141	24004	3 25 6 21 10		65 65	24030	14244	12 25 3 20 5		65 65
20313	43233	15 4 18 9 19		65 65	21303	44331	15 10 19 4 17		65 65
33610/H	6633/L		44001#	\65 65	35364/H	1092/L		46001#	\65 65
13042	14320	7 20 4 23 11		65 65	11432	02210	6 8 23 17 11		65 65
32330	21011	18 12 16 17 2		65 65	24103	14140	12 25 7 5 16		65 65
04204	00244	1 21 13 5 25		65 65	40240	11233	22 2 13 24 4		65 65
41121	33432	24 9 10 14 8		65 65	14302	40303	10 21 19 1 14		65 65
20413	42103	15 3 22 6 19		65 65	21033	43224	15 9 3 18 20		65 65

```

37037/H 24834/L          48001# \65/65|
11332   41410   10  7 20 17 11 |65|65|
04140   44232    5 25  8 24  3 |65|65|
24202   11233   12 22 13  4 14 |65|65|
40304   21200   23  2 18  1 21 |65|65|
21133   43030   15  9  6 19 16 |65|65|
[Solution Counts = 48544] [OK!]

```

Let me show you another list with the data sorted and referred to the old list.

\*\* Smart List of Standard Solutions of Self-complementary MS55:  
 Reconstructed by the 5th Increment System and New Euler's Method \*\*

(Euler Units:High/Low/ Sol\_Numb:[New]Old# Check\_Sums:\D1/D2|Row|Column|)

```

EU: 137H\10/10|      1901L\10/10| [Math]      \60/60| [  1]      1#\65/65|
0 4 4 3 0|11|10|  0 1 0 3 1| 5|10|  0 21 20 18  1|60|60|  1 22 21 19  2|65|65|
3 0 2 1 3| 9|11|  4 2 4 3 2|15| 5| 19  2 14  8 17|60|60| 20  3 15  9 18|65|65|
2 3 2 1 2|10|10|  1 0 2 4 3|10|10| 11 15 12  9 13|60|60| 12 16 13 10 14|65|65|
1 3 2 4 1|11| 9|  2 1 0 2 0| 5|15|  7 16 10 22  5|60|60|  8 17 11 23  6|65|65|
4 1 0 0 4| 9|10|  3 1 4 3 4|15|10| 23  6  4  3 24|60|60| 24  7  5  4 25|65|65|

EU: 1162H\10/10|     2810L\10/10| [Math]      \60/60| [ 501]     501#\65/65|
0 4 2 4 0|10|11|  0 3 4 1 2|10| 5|  0 23 14 21  2|60|60|  1 24 15 22  3|65|65|
4 1 3 1 2|11| 9|  0 1 1 0 3| 5|15| 20  6 16  5 13|60|60| 21  7 17  6 14|65|65|
1 1 2 3 3|10|10|  2 4 2 0 2|10|10|  7  9 12 15 17|60|60|  8 10 13 16 18|65|65|
2 3 1 3 0| 9|11|  1 4 3 3 4|15| 5| 11 19  8 18  4|60|60| 12 20  9 19  5|65|65|
4 0 2 0 4|10| 9|  2 3 0 1 4|10|15| 22  3 10  1 24|60|60| 23  4 11  2 25|65|65|

EU: 5351H\10/10|     2920L\10/10| [Math]      \60/60| [1001]    1001#\65/65|
0 3 3 4 0|10|10|  0 4 3 1 2|10|10|  0 19 18 21  2|60|60|  1 20 19 22  3|65|65|
0 3 2 3 2|10|10|  4 1 0 2 3|10|10|  4 16 10 17 13|60|60|  5 17 11 18 14|65|65|
4 3 2 1 0|10|10|  3 0 2 4 1|10|10| 23 15 12  9  1|60|60| 24 16 13 10  2|65|65|
2 1 2 1 4|10|10|  1 2 4 3 0|10|10| 11  7 14  8 20|60|60| 12  8 15  9 21|65|65|
4 0 1 1 4|10|10|  2 3 1 0 4|10|10| 22  3  6  5 24|60|60| 23  4  7  6 25|65|65|

EU: 2631H\10/10|     4836L\10/10| [Math]      \60/60| [1501]    1501#\65/65|
0 3 3 4 0|10|10|  0 1 4 2 3|10|10|  0 16 19 22  3|60|60|  1 17 20 23  4|65|65|
2 2 4 0 1| 9|10|  4 3 3 4 1|15|10| 14 13 23  4  6|60|60| 15 14 24  5  7|65|65|
1 1 2 3 3|10|10|  2 4 2 0 2|10|10|  7  9 12 15 17|60|60|  8 10 13 16 18|65|65|
3 4 0 2 2|11|10|  3 0 1 1 0| 5|10| 18 20  1 11 10|60|60| 19 21  2 12 11|65|65|
4 0 1 1 4|10|10|  1 2 0 3 4|10|10| 21  2  5  8 24|60|60| 22  3  6  9 25|65|65|

EU: 1459H\10/10|     5911L\10/10| [Math]      \60/60| [2001]    2001#\65/65|
0 2 4 4 0|10|10|  0 1 3 2 4|10|10|  0 11 23 22  4|60|60|  1 12 24 23  5|65|65|
3 1 3 1 2|10|10|  2 3 4 1 0|10|10| 17  8 19  6 10|60|60| 18  9 20  7 11|65|65|
1 4 2 0 3|10|10|  4 1 2 3 0|10|10|  9 21 12  3 15|60|60| 10 22 13  4 16|65|65|
2 3 1 3 1|10|10|  4 3 0 1 2|10|10| 14 18  5 16  7|60|60| 15 19  6 17  8|65|65|
4 0 0 2 4|10|10|  0 2 1 3 4|10|10| 20  2  1 13 24|60|60| 21  3  2 14 25|65|65|

EU: 5786H\10/10|     5429L\10/10| [Math]      \60/60| [2501]    2501#\65/65|
0 3 4 3 0|10|11|  0 0 2 4 4|10| 5|  0 15 22 19  4|60|60|  1 16 23 20  5|65|65|
2 4 3 1 1|11|11|  0 1 1 1 2| 5| 5| 10 21 16  6  7|60|60| 11 22 17  7  8|65|65|
2 0 2 4 2|10|10|  3 1 2 3 1|10|10| 13  1 12 23 11|60|60| 14  2 13 24 12|65|65|
3 3 1 0 2| 9| 9|  2 3 3 3 4|15|15| 17 18  8  3 14|60|60| 18 19  9  4 15|65|65|
4 1 0 1 4|10| 9|  0 0 2 4 4|10|15| 20  5  2  9 24|60|60| 21  6  3 10 25|65|65|

EU:13045H\10/10|     604L\10/10| [Math]      \60/60| [3001]    3001#\65/65|
0 4 3 3 1|11|10|  0 2 0 3 0| 5|10|  0 22 15 18  5|60|60|  1 23 16 19  6|65|65|
0 3 2 4 0| 9|10|  4 2 3 3 3|15|10|  4 17 13 23  3|60|60|  5 18 14 24  4|65|65|
3 2 2 2 1|10|10|  1 4 2 0 3|10|10| 16 14 12 10  8|60|60| 17 15 13 11  9|65|65|
4 0 2 1 4|11|10|  1 1 1 2 0| 5|10| 21  1 11  7 20|60|60| 22  2 12  8 21|65|65|
3 1 1 0 4| 9|10|  4 1 4 2 4|15|10| 19  6  9  2 24|60|60| 20  7 10  3 25|65|65|

EU:10833H\10/10|     1364L\10/10| [Math]      \60/60| [3501]    3501#\65/65|
0 4 4 1 1|10|10|  0 3 2 4 1|10|10|  0 23 22  9  6|60|60|  1 24 23 10  7|65|65|
1 2 4 3 0|10|11|  3 0 0 4 3|10| 5|  8 10 20 19  3|60|60|  9 11 21 20  4|65|65|
2 1 2 3 2|10|10|  3 2 2 2 1|10|10| 13  7 12 17 11|60|60| 14  8 13 18 12|65|65|
4 1 0 2 3|10| 9|  1 0 4 4 1|10|15| 21  5  4 14 16|60|60| 22  6  5 15 17|65|65|
3 3 0 0 4|10|10|  3 0 2 1 4|10|10| 18 15  2  1 24|60|60| 19 16  3  2 25|65|65|

```

EU: 6484H\10/10	3175L\10/10	[Math] \60/60	[ 4001] 4001#\65/65
0 4 2 3 1 10 10	0 3 1 4 2 10 10	0 23 11 19 7 60 60	1 24 12 20 8 65 65
3 0 4 0 3 10 11	3 2 1 4 0 10  5	18 2 21 4 15 60 60	19 3 22 5 16 65 65
3 2 2 2 1 10 10	1 0 2 4 3 10 10	16 10 12 14 8 60 60	17 11 13 15 9 65 65
1 4 0 4 1 10  9	4 0 3 2 1 10 15	9 20 3 22 6 60 60	10 21 4 23 7 65 65
3 1 2 0 4 10 10	2 0 3 1 4 10 10	17 5 13 1 24 60 60	18 6 14 2 25 65 65
EU:13031H\10/10	3536L\10/10	[Math] \60/60	[ 4501] 4501#\65/65
0 4 2 3 1 10 10	0 2 3 3 2 10 10	0 22 13 18 7 60 60	1 23 14 19 8 65 65
0 3 3 4 0 10 10	4 4 0 1 1 10 10	4 19 15 21 1 60 60	5 20 16 22 2 65 65
3 2 2 2 1 10 10	1 0 2 4 3 10 10	16 10 12 14 8 60 60	17 11 13 15 9 65 65
4 0 1 1 4 10 10	3 3 4 0 0 10 10	23 3 9 5 20 60 60	24 4 10 6 21 65 65
3 1 2 0 4 10 10	2 1 1 2 4 10 10	17 6 11 2 24 60 60	18 7 12 3 25 65 65
EU: 9127H\10/10	4292L\10/10	[Math] \60/60	[ 5001] 5001#\65/65
0 2 4 3 1 10 10	0 3 1 3 3 10 10	0 13 21 18 8 60 60	1 14 22 19 9 65 65
4 1 3 2 1 11 10	3 2 0 0 0  5 10	23 7 15 10 5 60 60	24 8 16 11 6 65 65
0 4 2 0 4 10 10	2 0 2 4 2 10 10	2 20 12 4 22 60 60	3 21 13 5 23 65 65
3 2 1 3 0  9 10	4 4 4 2 1 15 10	19 14 9 17 1 60 60	20 15 10 18 2 65 65
3 1 0 2 4 10 10	1 1 3 1 4 10 10	16 6 3 11 24 60 60	17 7 4 12 25 65 65
EU:10260H\10/10	6754L\10/10	[Math] \60/60	[10001] 10001#\65/65
0 4 3 3 1 11 10	1 2 0 2 0  5 10	1 22 15 17 5 60 60	2 23 16 18 6 65 65
1 2 4 1 2 10 10	1 1 1 3 4 10 10	6 11 21 8 14 60 60	7 12 22 9 15 65 65
4 0 2 4 0 10 10	4 4 2 0 0 10 10	24 4 12 20 0 60 60	25 5 13 21 1 65 65
2 3 0 2 3 10 10	0 1 3 3 3 10 10	10 16 3 13 18 60 60	11 17 4 14 19 65 65
3 1 1 0 4  9 10	4 2 4 2 3 15 10	19 7 9 2 23 60 60	20 8 10 3 24 65 65
EU: 5479H\10/10	15706L\10/10	[Math] \60/60	[15001] 15001#\65/65
0 2 4 4 0 10 11	2 1 4 0 3 10  5	2 11 24 20 3 60 60	3 12 25 21 4 65 65
1 3 2 3 1 10 10	0 1 4 3 2 10 10	5 16 14 18 7 60 60	6 17 15 19 8 65 65
3 4 2 0 1 10 10	0 3 2 1 4 10 10	15 23 12 1 9 60 60	16 24 13 2 10 65 65
3 1 2 1 3 10 10	2 1 0 3 4 10 10	17 6 10 8 19 60 60	18 7 11 9 20 65 65
4 0 0 2 4 10  9	1 4 0 3 2 10 15	21 4 0 13 22 60 60	22 5 1 14 23 65 65
EU:13116H\10/10	16599L\10/10	[Math] \60/60	[20001] 20001#\65/65
0 3 4 1 1  9 10	2 4 4 1 4 15 10	2 19 24 6 9 60 60	3 20 25 7 10 65 65
2 3 2 4 0 11  9	3 1 0 0 1  5 15	13 16 10 20 1 60 60	14 17 11 21 2 65 65
1 0 2 4 3 10 10	2 3 2 1 2 10 10	7 3 12 21 17 60 60	8 4 13 22 18 65 65
4 0 2 1 2  9 11	3 4 4 3 1 15  5	23 4 14 8 11 60 60	24 5 15 9 12 65 65
3 3 0 1 4 11 10	0 3 0 0 2  5 10	15 18 0 5 22 60 60	16 19 1 6 23 65 65
EU: 9656H\10/10	21520L\10/10	[Math] \60/60	[25001] 25001#\65/65
0 4 2 3 1 10 10	3 4 0 0 3 10 10	3 24 10 15 8 60 60	4 25 11 16 9 65 65
2 1 4 4 0 11  9	1 1 0 2 1  5 15	11 6 20 22 1 60 60	12 7 21 23 2 65 65
1 3 2 1 3 10 10	2 4 2 0 2 10 10	7 19 12 5 17 60 60	8 20 13 6 18 65 65
4 0 0 3 2  9 11	3 2 4 3 3 15  5	23 2 4 18 13 60 60	24 3 5 19 14 65 65
3 1 2 0 4 10 10	1 4 4 0 1 10 10	16 9 14 0 21 60 60	17 10 15 1 22 65 65
EU:11830H\10/10	26558L\10/10	[Math] \60/60	[30001] 30000#\65/65
0 1 4 3 1  9  9	4 4 2 3 2 15 15	4 9 22 18 7 60 60	5 10 23 19 8 65 65
4 3 2 0 2 11 10	3 1 0 0 1  5 10	23 16 10 0 11 60 60	24 17 11 1 12 65 65
0 1 2 3 4 10 10	3 0 2 4 1 10 10	3 5 12 19 21 60 60	4 6 13 20 22 65 65
2 4 2 1 0  9 10	3 4 4 3 1 15 10	13 24 14 8 1 60 60	14 25 15 9 2 65 65
3 1 0 3 4 11 11	2 1 2 0 0  5  5	17 6 2 15 20 60 60	18 7 3 16 21 65 65
EU:28633H\10/10	4854L\10/10	[Math] \60/60	[35001] 35001#\65/65
1 4 0 4 1 10 11	0 0 3 4 3 10  5	5 20 3 24 8 60 60	6 21 4 25 9 65 65
1 4 3 2 0 10 11	1 3 0 4 2 10  5	6 23 15 14 2 60 60	7 24 16 15 3 65 65
2 1 2 3 2 10 10	1 2 2 2 3 10 10	11 7 12 17 13 60 60	12 8 13 18 14 65 65
4 2 1 0 3 10  9	2 0 4 1 3 10 15	22 10 9 1 18 60 60	23 11 10 2 19 65 65
3 0 4 0 3 10  9	1 0 1 4 4 10 15	16 0 21 4 19 60 60	17 1 22 5 20 65 65
EU:34396H\10/10	7061L\10/10	[Math] \60/60	[40001] 40001#\65/65
1 0 4 3 2 10 10	1 1 4 4 0 10 10	6 1 24 19 10 60 60	7 2 25 20 11 65 65
1 3 4 1 0  9 11	4 2 2 3 4 15  5	9 17 22 8 4 60 60	10 18 23 9 5 65 65
2 4 2 0 2 10 10	1 1 2 3 3 10 10	11 21 12 3 13 60 60	12 22 13 4 14 65 65
4 3 0 1 3 11  9	0 1 2 2 0  5 15	20 16 2 7 15 60 60	21 17 3 8 16 65 65
2 1 0 4 3 10 10	4 0 0 3 3 10 10	14 5 0 23 18 60 60	15 6 1 24 19 65 65

```

EU:30683H\10/10|      18841L\10/10| [Math]      \60/60| [45001] 45001#\65/65|
1 1 4 2 2|10|10|  3 0 4 3 0|10|10|    8 5 24 13 10|60|60|    9 6 25 14 11|65|65|
4 0 1 1 4|10|10|  1 4 1 2 2|10|10|    21 4 6 7 22|60|60|    22 5 7 8 23|65|65|
3 4 2 0 1|10|10|  0 3 2 1 4|10|10|    15 23 12 1 9|60|60|    16 24 13 2 10|65|65|
0 3 3 4 0|10|10|  2 2 3 0 3|10|10|    2 17 18 20 3|60|60|    3 18 19 21 4|65|65|
2 2 0 3 3|10|10|  4 1 0 4 1|10|10|    14 11 0 19 16|60|60|    15 12 1 20 17|65|65|

EU:39606H\10/10|      2128L\10/10| [Math]      \60/60| [48501] 48501#\65/65|
2 3 1 3 2|11|11|  0 3 0 1 1| 5| 5|    10 18 5 16 11|60|60|    11 19 6 17 12|65|65|
3 4 0 3 0|10| 9|  0 3 3 2 2|10|15|    15 23 3 17 2|60|60|    16 24 4 18 3|65|65|
0 0 2 4 4|10|10|  0 4 2 0 4|10|10|    0 4 12 20 24|60|60|    1 5 13 21 25|65|65|
4 1 4 0 1|10|11|  2 2 1 1 4|10| 5|    22 7 21 1 9|60|60|    23 8 22 2 10|65|65|
2 1 3 1 2| 9| 9|  3 3 4 1 4|15|15|    13 8 19 6 14|60|60|    14 9 20 7 15|65|65|

```

[Total Count of Solutions = 48544] [OK!]

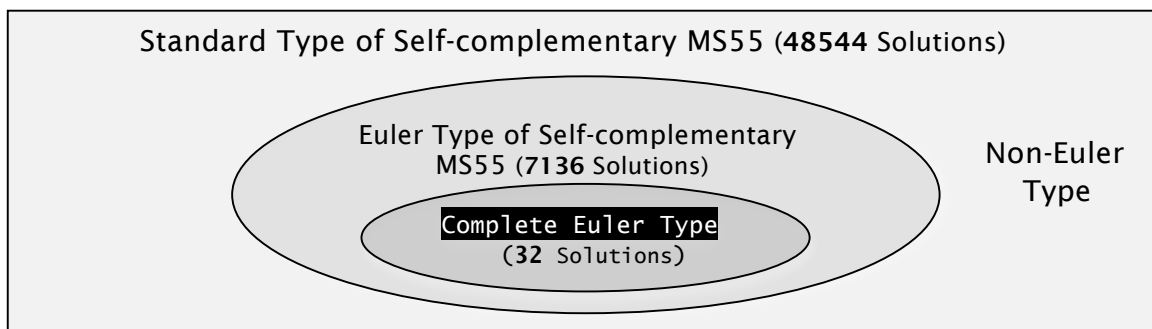
\*\* Calculated and Listed by Kanji Setsuda  
on Apr. 5, 2012 with MacOSX 10.7.3 & Xcode 4.2.1 \*\*

6. They prove our successful result of reconstructing all the standard solutions of Self-complementary Magic Squares of order 5 as many as 48544.

We could have neither made any other type of solutions, nor missed any one of necessary solutions to be reconstructed.

I am happy I can report about that here.

I hope the whole process of our experiment could clarify the inner structure of our object and might possibly inspire you to study more.



The set diagram above shows the relations among the three solution sets of Self-complementary Magic Squares of Order 5 listed in this article.

‘Complete Euler Type’ consists of those beautiful 32 solutions composed by the Greco-Latinian Method. Essential parts of their decomposed units such as rows, columns and two primary diagonals have every number of (0, 1, 2, 3 and 4) in each content, used strictly once and no duplication should be found there.

The ‘Euler Type’ is recently re-defined as that it should contain its decomposed Euler Units very similar to the whole object, such as the constant sum of every row, column, primary diagonal and every complementary pair in their symmetrical positions. But we don’t care about the content patterns of them at all, and pass every part simply when its sum is equal to the constant value.

I am afraid you might be surprised at the vast counts of Euler Units for high and low. In fact we had to test terribly lots of cases as many as 39916 x 30016, taking a lot of time longer than a quarter of an hour even by my newest PC.

It looks we might have had lots of waste working in our application, but they were so necessary that we had to take them all, to tell the truth.

It was indeed such a hard job that I would not like to try it again, I confess.

The next job for us might be to reconstruct all the Standard type of MS55, I know, but I myself am still hesitating to accept to do that.

As you know there are too many solutions to be analyzed. See the following list. How many Euler Units do we have to compose for them? How many cases do we have to examine? How many hours will it take? You can imagine, can't you?

**\*\* Examples of the Standard Type of Magic Squares of Order 5: \*\***

1 22 21 18 3 65 65/	41093 65\65/	86874 65\65/	154894 65\65/
1 22 21 18 3 65 65	1 23 21 18 2 65 65	1 22 21 19 2 65 65	1 23 21 18 2 65 65
20 2 15 9 19 65 65	22 3 15 14 11 65 65	14 4 20 12 15 65 65	15 5 17 12 16 65 65
14 16 13 10 12 65 65	13 16 12 5 19 65 65	18 16 11 7 13 65 65	20 14 10 8 13 65 65
7 17 11 24 6 65 65	9 17 7 24 8 65 65	9 17 5 24 10 65 65	7 19 6 24 9 65 65
23 8 5 4 25 65 65	20 6 10 4 25 65 65	23 6 8 3 25 65 65	22 4 11 3 25 65 65
221635 65\65/	312570 65\65/	401482 65\65/	502999 65\65/
1 22 23 17 2 65 65	1 22 20 19 3 65 65	1 22 21 18 3 65 65	1 23 21 16 4 65 65
15 6 18 14 12 65 65	13 7 18 15 12 65 65	12 8 16 15 14 65 65	17 9 14 15 10 65 65
20 13 9 7 16 65 65	17 14 8 5 21 65 65	20 13 7 6 19 65 65	20 12 6 8 19 65 65
8 19 4 24 10 65 65	11 16 10 24 4 65 65	9 17 11 24 4 65 65	5 18 11 24 7 65 65
21 5 11 3 25 65 65	23 6 9 2 25 65 65	23 5 10 2 25 65 65	22 3 13 2 25 65 65
598305 65\65/	710010 65\65/	811401 65\65/	938219 65\65/
1 19 21 18 6 65 65	1 21 22 15 6 65 65	1 19 23 15 7 65 65	1 14 23 17 10 65 65
12 10 20 14 9 65 65	16 11 17 14 7 65 65	10 12 17 18 8 65 65	15 13 20 12 5 65 65
16 15 5 7 22 65 65	20 13 4 9 19 65 65	22 14 3 6 20 65 65	21 16 2 8 18 65 65
13 17 8 24 3 65 65	5 18 10 24 8 65 65	11 16 9 24 5 65 65	6 19 9 24 7 65 65
23 4 11 2 25 65 65	23 2 12 3 25 65 65	21 4 13 2 25 65 65	22 3 11 4 25 65 65
1075807 65\65/	1202625 65\65/	1304016 65\65/	1415721 65\65/
1 18 21 16 9 65 65	1 18 24 14 8 65 65	1 17 24 15 8 65 65	1 15 24 18 7 65 65
12 14 19 13 7 65 65	9 15 17 19 5 65 65	14 16 18 13 4 65 65	12 17 16 14 6 65 65
22 11 2 10 20 65 65	21 16 2 6 20 65 65	20 10 2 11 22 65 65	21 11 2 9 22 65 65
6 17 15 23 4 65 65	11 13 12 22 7 65 65	7 19 12 21 6 65 65	8 19 13 20 5 65 65
24 5 8 3 25 65 65	23 3 10 4 25 65 65	23 3 9 5 25 65 65	23 3 10 4 25 65 65
1511027 65\65/	1612544 65\65/	1701456 65\65/	1792391 65\65/
1 20 24 13 7 65 65	1 13 24 20 7 65 65	1 15 24 18 7 65 65	1 14 22 20 8 65 65
9 18 15 17 6 65 65	9 19 15 16 6 65 65	9 20 16 14 6 65 65	5 21 18 15 6 65 65
21 8 2 12 22 65 65	21 12 2 8 22 65 65	21 8 2 12 22 65 65	24 9 2 11 19 65 65
11 16 14 19 5 65 65	11 17 14 18 5 65 65	11 19 13 17 5 65 65	12 17 13 16 7 65 65
23 3 10 4 25 65 65	23 4 10 3 25 65 65	23 3 10 4 25 65 65	23 4 10 3 25 65 65
1859132 65\65/	1927152 65\65/	1972933 65\65/	2014025 65\65/
1 14 24 18 8 65 65	1 16 21 18 9 65 65	1 17 23 14 10 65 65	1 21 22 18 3 65 65
11 22 12 13 7 65 65	6 23 15 17 4 65 65	6 24 9 19 7 65 65	20 2 14 10 19 65 65
21 6 2 16 20 65 65	22 10 2 11 20 65 65	21 8 2 16 18 65 65	15 17 13 8 12 65 65
9 19 17 15 5 65 65	12 13 19 14 7 65 65	15 12 20 13 5 65 65	6 16 11 25 7 65 65
23 4 10 3 25 65 65	24 3 8 5 25 65 65	22 4 11 3 25 65 65	23 9 5 4 24 65 65
2047980 65\65/	2098871 65\65/	2157680 65\65/	2228857 65\65/
1 23 22 17 2 65 65	1 23 21 18 2 65 65	1 23 22 17 2 65 65	1 23 21 17 3 65 65
20 3 16 11 15 65 65	15 4 20 14 12 65 65	15 5 18 13 14 65 65	19 6 15 11 14 65 65
18 13 12 8 14 65 65	19 13 11 5 17 65 65	20 12 10 7 16 65 65	18 12 9 10 16 65 65
5 19 6 25 10 65 65	8 16 6 25 10 65 65	8 19 4 25 9 65 65	5 20 7 25 8 65 65
21 7 9 4 24 65 65	22 9 7 3 24 65 65	21 6 11 3 24 65 65	22 4 13 2 24 65 65
2307901 65\65/	2399576 65\65/	2505443 65\65/	2602430 65\65/
1 23 20 18 3 65 65	1 21 22 17 4 65 65	1 20 23 17 4 65 65	1 19 21 18 6 65 65
14 7 16 15 13 65 65	11 8 18 12 16 65 65	13 9 14 18 11 65 65	15 10 17 11 12 65 65
19 12 8 5 21 65 65	20 14 7 9 15 65 65	21 16 6 3 19 65 65	22 14 5 8 16 65 65
9 17 10 25 4 65 65	10 19 5 25 6 65 65	8 15 10 25 7 65 65	4 20 9 25 7 65 65
22 6 11 2 24 65 65	23 3 13 2 24 65 65	22 5 12 2 24 65 65	23 2 13 3 24 65 65
2697682 65\65/	2805714 65\65/	2911721 65\65/	3020377 65\65/
1 20 22 16 6 65 65	1 16 23 17 8 65 65	1 18 21 14 11 65 65	1 20 22 15 7 65 65
12 11 18 15 9 65 65	10 12 20 14 9 65 65	9 13 19 17 7 65 65	11 14 18 17 5 65 65
19 14 4 7 21 65 65	21 15 3 7 19 65 65	22 16 2 5 20 65 65	19 16 3 6 21 65 65
10 17 8 25 5 65 65	11 18 6 25 5 65 65	10 12 15 25 3 65 65	9 13 12 23 8 65 65
23 3 13 2 24 65 65	22 4 13 2 24 65 65	23 6 8 4 24 65 65	25 2 10 4 24 65 65

3115151 65\65/	3243010 65\65/	3347862 65\65/	3461217 65\65/
1 17 22 16 9 65 65	1 19 25 12 8 65 65	1 18 23 15 8 65 65	1 13 25 19 7 65 65
13 15 19 11 7 65 65	15 16 10 18 6 65 65	7 17 20 16 5 65 65	10 18 15 17 5 65 65
20 12 2 10 21 65 65	21 13 2 9 20 65 65	19 12 2 10 22 65 65	22 14 2 6 21 65 65
6 18 14 23 4 65 65	5 14 17 22 7 65 65	13 14 11 21 6 65 65	9 16 12 20 8 65 65
25 3 8 5 24 65 65	23 3 11 4 24 65 65	25 4 9 3 24 65 65	23 4 11 3 24 65 65
3559391 65\65/	3658055 65\65/	3746462 65\65/	3838740 65\65/
1 22 23 13 6 65 65	1 19 25 13 7 65 65	1 15 23 18 8 65 65	1 12 25 19 8 65 65
8 19 17 16 5 65 65	9 20 15 16 5 65 65	13 21 11 14 6 65 65	5 22 17 14 7 65 65
20 7 3 14 21 65 65	22 6 2 14 21 65 65	19 10 2 12 22 65 65	21 9 2 13 20 65 65
11 15 12 18 9 65 65	10 17 12 18 8 65 65	7 16 20 17 5 65 65	15 18 10 16 6 65 65
25 2 10 4 24 65 65	23 3 11 4 24 65 65	25 3 9 4 24 65 65	23 4 11 3 24 65 65
3906462 65\65/	3973038 65\65/	4013925 65\65/	4051842 65\65/
1 16 22 17 9 65 65	1 14 21 18 11 65 65	1 24 20 17 3 65 65	1 24 20 18 2 65 65
6 23 14 18 4 65 65	10 25 15 12 3 65 65	19 2 21 8 15 65 65	15 3 14 12 21 65 65
20 12 2 10 21 65 65	22 5 2 16 20 65 65	16 12 14 10 13 65 65	19 17 13 6 10 65 65
13 11 19 15 7 65 65	9 17 19 13 7 65 65	7 18 4 25 11 65 65	8 16 7 25 9 65 65
25 3 8 5 24 65 65	23 4 8 6 24 65 65	22 9 6 5 23 65 65	22 5 11 4 23 65 65
4092654 65\65/	4154963 65\65/	4215776 65\65/	4295864 65\65/
1 21 22 19 2 65 65	1 20 24 18 2 65 65	1 21 22 19 2 65 65	1 24 21 17 2 65 65
15 4 20 10 16 65 65	16 5 10 13 21 65 65	13 6 20 14 12 65 65	13 7 15 14 16 65 65
18 14 12 8 13 65 65	19 14 11 6 15 65 65	18 16 10 4 17 65 65	19 11 9 6 20 65 65
7 17 5 25 11 65 65	7 17 12 25 4 65 65	9 15 5 25 11 65 65	10 18 8 25 4 65 65
24 9 6 3 23 65 65	22 9 8 3 23 65 65	24 7 8 3 23 65 65	22 5 12 3 23 65 65
4372535 65\65/	4462269 65\65/	4562877 65\65/	4665674 65\65/
1 19 25 18 2 65 65	1 22 21 18 3 65 65	1 22 20 18 4 65 65	1 19 22 17 6 65 65
17 8 13 11 16 65 65	16 9 15 14 11 65 65	12 10 17 15 11 65 65	20 11 15 12 7 65 65
15 14 9 7 20 65 65	19 13 7 6 20 65 65	21 14 6 5 19 65 65	16 14 5 9 21 65 65
10 21 6 24 4 65 65	5 17 10 25 8 65 65	7 16 9 25 8 65 65	4 18 10 25 8 65 65
22 3 12 5 23 65 65	24 4 12 2 23 65 65	24 3 13 2 23 65 65	24 3 13 2 23 65 65
4756380 65\65/	4866805 65\65/	4961309 65\65/	5076952 65\65/
1 19 24 14 7 65 65	1 17 21 18 8 65 65	1 12 24 18 10 65 65	1 16 21 18 9 65 65
16 12 17 11 9 65 65	11 13 20 14 7 65 65	15 14 17 11 8 65 65	11 15 19 12 8 65 65
20 10 4 13 18 65 65	19 15 3 6 22 65 65	21 16 2 7 19 65 65	22 14 2 7 20 65 65
6 21 5 25 8 65 65	10 16 9 25 5 65 65	6 20 9 25 5 65 65	6 17 13 24 5 65 65
22 3 15 2 23 65 65	24 4 12 2 23 65 65	22 3 13 4 23 65 65	25 3 10 4 23 65 65
5164282 65\65/	5273407 65\65/	5380809 65\65/	5493905 65\65/
1 15 24 19 6 65 65	1 18 24 14 8 65 65	1 20 22 14 8 65 65	1 16 24 17 7 65 65
12 16 18 14 5 65 65	12 17 16 11 9 65 65	10 18 13 15 9 65 65	12 19 15 13 6 65 65
20 13 3 8 21 65 65	21 7 2 15 20 65 65	25 7 2 12 19 65 65	22 9 2 11 21 65 65
7 17 9 22 10 65 65	6 19 13 22 5 65 65	5 16 17 21 6 65 65	5 18 14 20 8 65 65
25 4 11 2 23 65 65	25 4 10 3 23 65 65	24 4 11 3 23 65 65	25 3 10 4 23 65 65
5586540 65\65/	5709075 65\65/	5793000 65\65/	5884795 65\65/
1 15 24 18 7 65 65	1 14 25 17 8 65 65	1 14 24 18 8 65 65	1 14 22 19 9 65 65
12 20 13 14 6 65 65	13 21 7 15 9 65 65	6 22 13 19 5 65 65	11 24 13 12 5 65 65
22 9 2 11 21 65 65	22 10 2 12 19 65 65	21 15 2 7 20 65 65	21 6 2 16 20 65 65
5 17 16 19 8 65 65	5 16 20 18 6 65 65	12 11 16 17 9 65 65	7 17 18 15 8 65 65
25 4 10 3 23 65 65	24 4 11 3 23 65 65	25 3 10 4 23 65 65	25 4 10 3 23 65 65
5950928 65\65/	5996665 65\65/	6026373 65\65/	6070262 65\65/
1 18 24 12 10 65 65	1 23 17 21 3 65 65	1 23 20 19 2 65 65	1 20 23 19 2 65 65
5 25 9 20 6 65 65	16 2 20 9 18 65 65	21 3 15 9 17 65 65	17 4 15 8 21 65 65
21 8 2 15 19 65 65	13 19 15 6 12 65 65	12 18 14 8 13 65 65	12 16 13 10 14 65 65
16 11 17 14 7 65 65	11 14 5 25 10 65 65	7 16 6 25 11 65 65	11 18 5 25 6 65 65
22 3 13 4 23 65 65	24 7 8 4 22 65 65	24 5 10 4 22 65 65	24 7 9 3 22 65 65
6118886 65\65/	6183660 65\65/	6253859 65\65/	6336462 65\65/
1 19 23 20 2 65 65	1 18 23 21 2 65 65	1 23 19 20 2 65 65	1 21 23 18 2 65 65
14 5 16 9 21 65 65	19 6 13 12 15 65 65	14 7 16 11 17 65 65	19 8 14 13 11 65 65
15 17 12 8 13 65 65	14 20 11 3 17 65 65	21 13 10 6 15 65 65	15 16 9 5 20 65 65
11 18 4 25 7 65 65	7 16 8 25 9 65 65	5 18 8 25 9 65 65	6 17 7 25 10 65 65
24 6 10 3 22 65 65	24 5 10 4 22 65 65	24 4 12 3 22 65 65	24 3 12 4 22 65 65

6429800 65\65/	6521744 65\65/	6635682 65\65/	6733620 65\65/
1 20 24 18 2 65 65	1 21 24 16 3 65 65	1 20 23 17 4 65 65	1 20 23 15 6 65 65
13 9 16 15 12 65 65	15 10 17 14 9 65 65	10 11 14 18 12 65 65	17 12 16 9 11 65 65
21 14 8 3 19 65 65	20 11 7 8 19 65 65	24 13 6 3 19 65 65	19 10 5 13 18 65 65
7 17 6 25 10 65 65	6 18 4 25 12 65 65	9 16 7 25 8 65 65	4 21 7 25 8 65 65
23 5 11 4 22 65 65	23 5 13 2 22 65 65	21 5 15 2 22 65 65	24 2 14 3 22 65 65
6828983 65\65/	6927921 65\65/	7051499 65\65/	7166162 65\65/
1 18 24 16 6 65 65	1 16 23 17 8 65 65	1 19 21 14 10 65 65	1 14 23 18 9 65 65
14 13 17 12 9 65 65	10 14 20 12 9 65 65	9 15 18 17 6 65 65	13 16 19 12 5 65 65
19 11 4 10 21 65 65	19 15 3 7 21 65 65	23 16 2 4 20 65 65	20 15 2 7 21 65 65
8 20 5 25 7 65 65	11 18 6 25 5 65 65	8 12 13 25 7 65 65	6 17 10 24 8 65 65
23 3 15 2 22 65 65	24 2 13 4 22 65 65	24 3 11 5 22 65 65	25 3 11 4 22 65 65
7267139 65\65/	7380559 65\65/	7486831 65\65/	7612813 65\65/
1 19 24 12 9 65 65	1 20 24 14 6 65 65	1 17 24 15 8 65 65	1 16 24 17 7 65 65
15 17 16 11 6 65 65	5 18 17 15 10 65 65	7 19 16 12 11 65 65	12 20 14 9 10 65 65
20 8 2 14 21 65 65	23 7 3 13 19 65 65	23 6 2 14 20 65 65	23 6 3 15 18 65 65
4 18 13 23 7 65 65	11 16 9 21 8 65 65	9 18 13 21 4 65 65	4 21 13 19 8 65 65
25 3 10 5 22 65 65	25 4 12 2 22 65 65	25 5 10 3 22 65 65	25 2 11 5 22 65 65
7711706 65\65/	7822064 65\65/	7917353 65\65/	7988031 65\65/
1 15 23 18 8 65 65	1 19 24 12 9 65 65	1 14 23 18 9 65 65	1 14 21 19 10 65 65
11 21 14 13 6 65 65	8 23 15 13 6 65 65	13 24 10 12 6 65 65	9 25 13 12 6 65 65
24 7 2 12 20 65 65	20 4 2 18 21 65 65	21 7 2 15 20 65 65	23 4 2 16 20 65 65
4 17 16 19 9 65 65	11 16 14 17 7 65 65	5 17 19 16 8 65 65	8 17 18 15 7 65 65
25 5 10 3 22 65 65	25 3 10 5 22 65 65	25 3 11 4 22 65 65	24 5 11 3 22 65 65
8055937 65\65/	9916517 65\65/	11927511 65\65/	13840403 65\65/
1 22 20 19 3 65 65	1 22 16 23 3 65 65	1 23 21 17 3 65 65	1 23 17 21 3 65 65
23 2 15 8 17 65 65	15 2 19 8 21 65 65	20 2 15 6 22 65 65	20 2 16 5 22 65 65
12 18 16 6 13 65 65	14 18 17 4 12 65 65	11 16 18 12 8 65 65	13 15 19 6 12 65 65
5 14 10 25 11 65 65	11 13 7 25 9 65 65	9 14 4 25 13 65 65	7 14 9 25 10 65 65
24 9 4 7 21 65 65	24 10 6 5 20 65 65	24 10 7 5 19 65 65	24 11 4 8 18 65 65
15729139 65\65/	17498665 65\65/	19262329 65\65/	20945487 65\65/
1 21 18 22 3 65 65	1 22 19 20 3 65 65	1 24 17 20 3 65 65	1 24 18 19 3 65 65
19 2 14 7 23 65 65	17 2 15 8 23 65 65	21 2 13 10 19 65 65	21 2 15 7 20 65 65
13 16 20 6 10 65 65	11 18 21 5 10 65 65	11 14 22 6 12 65 65	12 13 23 6 11 65 65
8 11 9 25 12 65 65	12 9 6 25 13 65 65	9 7 8 25 16 65 65	9 10 4 25 17 65 65
24 15 4 5 17 65 65	24 14 4 7 16 65 65	23 18 5 4 15 65 65	22 16 5 8 14 65 65
22650789 65\65/	24256015 65\65/	25849695 65\65/	27246415 65\65/
1 23 16 21 4 65 65	1 23 19 18 4 65 65	1 23 18 20 3 65 65	1 23 20 18 3 65 65
19 2 12 10 22 65 65	20 3 14 7 21 65 65	17 4 15 7 22 65 65	19 5 13 7 21 65 65
14 15 24 3 9 65 65	9 16 24 5 11 65 65	14 9 24 5 13 65 65	8 12 24 4 17 65 65
11 7 5 25 17 65 65	13 8 2 25 17 65 65	12 10 2 25 16 65 65	15 9 2 25 14 65 65
20 18 8 6 13 65 65	22 15 6 10 12 65 65	21 19 6 8 11 65 65	22 16 6 11 10 65 65
28752703 65\65/	30012441 65\65/	31280461 65\65/	32357217 65\65/
1 22 19 20 3 65 65	1 22 18 20 4 65 65	1 20 19 21 4 65 65	1 21 20 19 4 65 65
18 6 13 5 23 65 65	17 7 15 5 21 65 65	16 8 14 5 22 65 65	16 9 7 10 23 65 65
15 8 24 4 14 65 65	14 11 24 3 13 65 65	13 10 24 3 15 65 65	11 13 24 3 14 65 65
10 12 2 25 16 65 65	10 9 2 25 19 65 65	12 9 2 25 17 65 65	15 5 2 25 18 65 65
21 17 7 11 9 65 65	23 16 6 12 8 65 65	23 18 6 11 7 65 65	22 17 12 8 6 65 65
33313773 65\65/	34050313 65\65/	34727611 65\65/	35163051 65\65/
1 23 19 18 4 65 65	1 23 16 20 5 65 65	1 21 19 18 6 65 65	1 22 15 20 7 65 65
17 10 11 6 21 65 65	18 11 6 8 22 65 65	16 12 11 4 22 65 65	18 13 10 3 21 65 65
12 7 24 2 20 65 65	13 10 24 3 15 65 65	15 7 24 5 14 65 65	14 5 24 6 16 65 65
13 9 3 25 15 65 65	12 7 2 25 19 65 65	10 8 2 25 20 65 65	9 8 4 25 19 65 65
22 16 8 14 5 65 65	21 14 17 9 4 65 65	23 17 9 13 3 65 65	23 17 12 11 2 65 65
35472327 65\65/	72448255 65\65/	109794755 65\65/	143903959 65\65/
2 22 21 17 3 65 65	3 21 20 17 4 65 65	4 21 19 16 5 65 65	5 22 15 17 6 65 65
20 1 15 10 19 65 65	19 1 22 8 15 65 65	14 1 22 8 20 65 65	21 1 16 8 19 65 65
14 18 13 9 11 65 65	13 16 12 14 10 65 65	17 13 11 15 9 65 65	9 20 10 14 12 65 65
6 16 12 24 7 65 65	7 18 5 24 11 65 65	7 18 10 24 6 65 65	7 18 13 24 3 65 65
23 8 4 5 25 65 65	23 9 6 2 25 65 65	23 12 3 2 25 65 65	23 4 11 2 25 65 65



174813433 65\65/	201422209 65\65/	223435493 65\65/	240804015 65\65/
6 15 17 20 7 65 65	7 14 18 17 9 65 65	8 18 20 10 9 65 65	9 20 14 12 10 65 65
21 1 22 8 13 65 65	20 1 22 6 16 65 65	13 1 17 12 22 65 65	13 1 19 11 21 65 65
10 19 9 11 16 65 65	12 21 8 13 11 65 65	16 21 7 15 6 65 65	17 22 6 16 4 65 65
5 18 14 24 4 65 65	3 19 15 24 4 65 65	5 14 19 24 3 65 65	3 15 18 24 5 65 65
23 12 3 2 25 65 65	23 10 2 5 25 65 65	23 11 2 4 25 65 65	23 7 8 2 25 65 65
253537291 65\65/	262175289 65\65/	267791141 65\65/	271388851 65\65/
10 20 17 7 11 65 65	11 18 14 10 12 65 65	12 19 16 5 13 65 65	13 19 15 4 14 65 65
15 1 19 12 18 65 65	17 1 23 8 16 65 65	17 1 23 10 14 65 65	17 1 22 16 9 65 65
13 22 5 16 9 65 65	13 21 4 20 7 65 65	9 20 3 22 11 65 65	6 24 3 20 12 65 65
4 14 21 24 2 65 65	2 19 15 24 5 65 65	6 18 15 24 2 65 65	8 11 18 23 5 65 65
23 8 3 6 25 65 65	22 6 9 3 25 65 65	21 7 8 4 25 65 65	21 10 7 2 25 65 65
273683093 65\65/	274792651 65\65/	275181411 65\65/	275279597 65\65/
14 18 13 5 15 65 65	15 14 13 6 17 65 65	16 14 12 6 17 65 65	17 15 9 6 18 65 65
20 1 24 12 8 65 65	20 1 23 12 9 65 65	19 1 23 13 9 65 65	14 2 25 11 13 65 65
9 22 2 21 11 65 65	7 24 2 21 11 65 65	7 24 2 22 10 65 65	12 24 1 21 7 65 65
3 17 16 23 6 65 65	5 16 19 22 3 65 65	5 15 20 21 4 65 65	3 16 20 22 4 65 65
19 7 10 4 25 65 65	18 10 8 4 25 65 65	18 11 8 3 25 65 65	19 8 10 5 23 65 65
275301177 65\65/			
18 10 5 13 19 65 65			
15 6 23 7 14 65 65			
11 25 3 24 2 65 65			
1 16 22 17 9 65 65			
20 8 12 4 21 65 65			

[Total Count of Standard Solutions = 275305224] [OK!]

\*\* Calculated and Listed by Kanji Setsuda \*\*

The 5-th order is already 'high' by itself.

It is always a hard job for us to study about the standard type of objects of high orders. For instance, it is almost impossible to compose all of them of order 6, and it is practically hopeless to make all those of orders more than 6.

And I admit that our new method mentioned here is not very quick.

But you know the 'Greco-Latinian Method' is such a quick application that it can show its wonderful power to compose every 'beautiful' object in a short time.

Although it can compose nothing but the beautiful, rare and precious objects, even of high orders, it surely gives us any easy chance to study them actually.

We now want to take those beautiful objects instead of knowing all the facts.

I would say, you had better take the 'Greco-Latinian Method' to make and study anything of high orders.

(Original Japanese Version was written on Jan. 8 and Mar.31, 2012;  
English one on Feb.10 and Apr. 6, 2012 with MacOSX 10.7.3 & Xcode 4.2.1)

---

by Kanji Setsuda: E-Mail Address:<jag12001@nifty.com>  
<http://homepage2.nifty.com/KanjiSetsuda/pages/EnglishP1.html>