

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 3-2: How to make 'Complete Euler Squares' of Order 7

#1. What is the 'Complete Euler Square' like?

It is sometimes called "Greco-Latin Square". Basic idea of that type was mentioned first by Legendary Leonhard Euler(1707-1783). What does it look like?

[Pan-diagonal MS77]														[Decomposed]																								
11	2	49	40	31	22	20	11	2	49	40	31	22	1	0	6	5	4	3	2	1	0	6	5	4	3	3	1	6	4	2	0	5	3	1	6	4	2	0
28	19	10	1 48 39 30 28 19 10							1	48	39	3	2	1	0 6 5 4 3 2 1							0	6	5	6	4	2	0 5 3 1 6 4 2							0	5	3
38	29	27	18  9  7 47 38 29 27							18	9	7	5	4	3	2 1 0 6 5 4 3							2	1	0	2	0	5	3 1 6 4 2 0 5							3	1	6
6	46	37	35 26 17  8  6 46 37							35	26	17	0	6	5	4 3 2 1 0 6 5							4	3	2	5	3	1	6 4 2 0 5 3 1							6	4	2
16	14	5	45 36 34 25 16 14  5							45	36	34	2	1	0	6 5 4 3 2 1 0							6	5	4	1	6	4	2 0 5 3 1 6 4							2	0	5
33	24	15	13  4 44 42 33 24 15							13	4	44	4	3	2	1 0 6 5 4 3 2							1	0	6	4	2	0	5 3 1 6 4 2 0							5	3	1
43	41	32	23 21 12  3 43 41 32							23	21	12	6	5	4	3 2 1 0 6 5 4							3	2	1	0	5	3	1 6 4 2 0 5 3							1	6	4
11	2	49	40 31 22 20 11  2 49							40	31	22	1	0	6	5 4 3 2 1 0 6							5	4	3	3	1	6	4 2 0 5 3 1 6							4	2	0
28	19	10	1	48	39	30	28	19	10	1	48	39	3	2	1	0	6	5	4	3	2	1	0	6	5	6	4	2	0	5	3	1	6	4	2	0	5	3

Let me take a system of base 7 for instance. Watch and study the figures above, especially the last two decomposed patterns of 'Complete Euler Square' 7x7.

Two positional layers for high values and low are drawn separately as shown above. At first you should always subtract 1 from any value of 49 numbers in the left form for object and then divide it by 7. Put the integer part of the quotient in the high layer and the remedy modulo(7) in the low one separately.

According to this system each sum of every column, of every row and even of every pan-diagonal in this example is always calculated by such the same form of equation:

$$(0+1+2+3+4+5+6) \times 7 + (0+1+2+3+4+5+6) = 21 \times 7 + 21 = 168 \text{ (Decimal)}$$

The equal sum means magic constant.

Yes. It is certainly an example of 'Complete Euler Square'.

The constant sum 168 of this example is equivalent to 175 in our classical notation, since modern system uses integers 0~48, instead of using natural numbers 1~49.

#2. Unique Properties of 'Complete Euler Squares'

The 'Complete Euler Square' has such beautiful properties as follows:

- (1) In each layer every column, every row and every pan-diagonal should consist of {0, 1, 2, 3, 4, 5 and 6} and have neither repetition, nor drop-off of any number.
- (2) Any value of {0, 1, 2, 3, 4, 5 or 6} should appear always seven times in each layer.
- (3) The value combination of high layer and low one in any location should be any of {00, 01, 02, 03, 04, 05, 06, 10, 11, 12, ..., 56, 60, 61, 62, 63, 64, 65 and 66(N7i)}.

Neither repetition nor drop-off of a certain combination should be found.

This property is logically equivalent to the basic promise of classic magic squares, in which you must use any natural number of {1, 2, 3, 4, 5, 6, 7, 8, ..., 45, 46, 47, 48 and 49} strictly once and you must not use any number twice or more often.

Why is the property(1) above always so important?

Watch the next equation again and try to think of its meaning, please.

$$(0+1+2+3+4+5+6) \times 7 + (0+1+2+3+4+5+6) = 21 \times 7 + 21 = 168 \text{ (Decimal)}$$

When the value pattern of any column, row and pan-diagonal is the same as the property(1), you will constantly have the same sum there. It means you will surely have got what you want, the Complete Euler type of pan-diagonal magic square.

Check and find that it has such beautiful properties in the example put above.

They say Euler did not mention about pan-diagonals, but I want all pan-diagonals also to have this property(1). I want to have those which have all the miraculous properties above and want to call them "Complete Euler Squares" from now on.

### #3. How to construct such 'Complete Euler Squares'

Can we make such a beautiful square as 'Complete Euler Square 7x7' directly?

Why don't you try to dictate our program simply expressing the idea literally after these beautiful definitions themselves?

[1] We usually want the variable N to take any value of 0~48 as dictated below.

But why don't you use two variables N\_high and N\_low, each of which takes any value of {0, 1, 2, 3, 4, 5 or 6}? Each variable is expected to stand for any value of its own positional layer.

How about dictating such a number generator program with double loops of for(...){...} sentences as follows?

```
/* Old */
for(n=0; n<49; n++){
    nm[1]=n;
    // ...;
}
/**/

/* New */
for(a=0; a<7; a++){           // a==d/7: for high
    for(b=0; b<7; b++){       // b==d%7: for low
        d=a*7+b;              // 'd' stands for N itself
        n[1]=d;
        // ...;
    }
}
/**/
```

[2] How do we realize that we must take any value of {0, 1, 2, 3, 4, 5 or 6} strictly once and must not use it twice or more often in any row, any column and any pan-diagonal of our object?

We have usually used the so-called "used\_flag" for this purpose.

When you use the value N, you should set the used\_flag for it as 'true', meaning 'used'. And when you finish using N, you should reset the flag as 'false', meaning 'not-used', and may return to the former step.

Whenever you make a new job, you should always check if the used\_flag is 'true' or 'false' before all.

```
/* Old */
for(n=0; n<49; n++){
    if(uflg[n]==0){           // Check if the value N is used or not
        nm[3]=n;             // If it is not used, then ...
        uflg[n]=1;           // Set Used_flag for the value N
        nextproc();          // Go to the next procedure
        uflg[n]=0;           // Reset Used_flag for the value N
    }
}
/**/
```

Why don't you remake it into the new style below to meet our double structure?

```

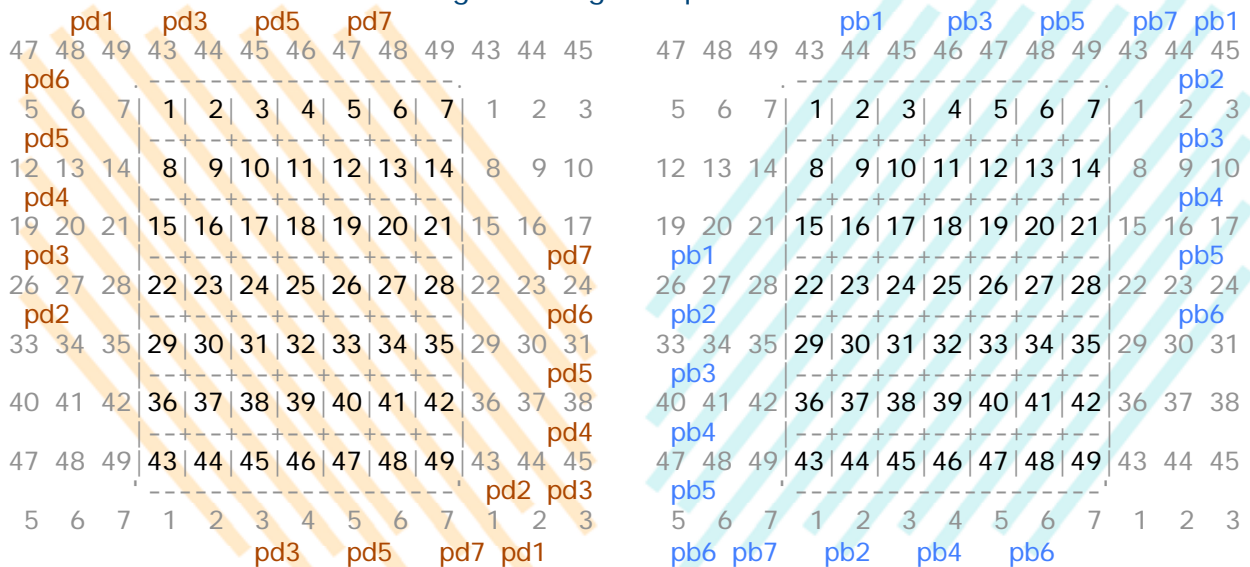
/* New */
for(a=0; a<7; a++){
  if(ufl g_high[a]==0){ // Check if it is used or not
    for(b=0; b<7; b++){
      if(ufl g_low[b]==0){ // Check if it is used or not
        n=a*7+b; nm[3]=n; // Produce N from a and b ...
        if(ufl g[n]==0){ // Check if the value N is used or not
          ufl g[n]=1; // Set Used_flag for the value N
          ufl g_high[a]=1; // Set Used_flag_high for Value a
          ufl g_low[b]=1; // Set Used_flag_low for Value b
          nextproc(); // Go to the next procedure
          ufl g_high[a]=0; // Reset Used_flag_high for Value a
          ufl g_low[b]=0; // Reset Used_flag_low for Value b
          ufl g[n]=0; // Reset Used_flag for Value N
        }
      }
    }
  }
}
/**/

```

[3] How many sets of used\_flags in all do we have to prepare, then?

We have to prepare 14 sets of used\_flags for 7 columns high and low, 14 for 7 rows high and low, and 28 for 14 pan-diagonals high and low: We need 56 sets in all!

**\*\* Basic Forms for Pan-diagonal Magic Squares of Order 7 \*\***



**\* Basic Conditions: C=175; \***

$n_1+n_2+n_3+n_4+n_5+n_6+n_7=C$	... rw1	$n_1+n_8+n_{15}+n_{22}+n_{29}+n_{36}+n_{43}=C$	... cl 1
$n_8+n_9+n_{10}+n_{11}+n_{12}+n_{13}+n_{14}=C$	... rw2	$n_2+n_9+n_{16}+n_{23}+n_{30}+n_{37}+n_{44}=C$	... cl 2
$n_{15}+n_{16}+n_{17}+n_{18}+n_{19}+n_{20}+n_{21}=C$	... rw3	$n_3+n_{10}+n_{17}+n_{24}+n_{31}+n_{38}+n_{45}=C$	... cl 3
$n_{22}+n_{23}+n_{24}+n_{25}+n_{26}+n_{27}+n_{28}=C$	... rw4	$n_4+n_{11}+n_{18}+n_{25}+n_{32}+n_{39}+n_{46}=C$	... cl 4
$n_{29}+n_{30}+n_{31}+n_{32}+n_{33}+n_{34}+n_{35}=C$	... rw5	$n_5+n_{12}+n_{19}+n_{26}+n_{33}+n_{40}+n_{47}=C$	... cl 5
$n_{36}+n_{37}+n_{38}+n_{39}+n_{40}+n_{41}+n_{42}=C$	... rw6	$n_6+n_{13}+n_{20}+n_{27}+n_{34}+n_{41}+n_{48}=C$	... cl 6
$n_{43}+n_{44}+n_{45}+n_{46}+n_{47}+n_{48}+n_{49}=C$	... rw7	$n_7+n_{14}+n_{21}+n_{28}+n_{35}+n_{42}+n_{49}=C$	... cl 7

**\* Pandiagonal Conditions: C=175; \***

$n_1+n_9+n_{17}+n_{25}+n_{33}+n_{41}+n_{49}=C$	... pd1	$n_1+n_{14}+n_{20}+n_{26}+n_{32}+n_{38}+n_{44}=C$	... pb1
$n_2+n_{10}+n_{18}+n_{26}+n_{34}+n_{42}+n_{43}=C$	... pd2	$n_2+n_8+n_{21}+n_{27}+n_{33}+n_{39}+n_{45}=C$	... pb2
$n_3+n_{11}+n_{19}+n_{27}+n_{35}+n_{36}+n_{44}=C$	... pd3	$n_3+n_9+n_{15}+n_{28}+n_{34}+n_{40}+n_{46}=C$	... pb3
$n_4+n_{12}+n_{20}+n_{28}+n_{29}+n_{37}+n_{45}=C$	... pd4	$n_4+n_{10}+n_{16}+n_{22}+n_{35}+n_{41}+n_{47}=C$	... pb4
$n_5+n_{13}+n_{21}+n_{22}+n_{30}+n_{38}+n_{46}=C$	... pd5	$n_5+n_{11}+n_{17}+n_{23}+n_{29}+n_{42}+n_{48}=C$	... pb5



```

short nm[50], uflg[50];
short anm[5][50];
short cl1h[8], cl1l[8], rw1h[8], rw1l[8], pd1h[8], pd1l[8], pb1h[8], pb1l[8];
short cl2h[8], cl2l[8], rw2h[8], rw2l[8], pd2h[8], pd2l[8], pb2h[8], pb2l[8];
short cl3h[8], cl3l[8], rw3h[8], rw3l[8], pd3h[8], pd3l[8], pb3h[8], pb3l[8];
short cl4h[8], cl4l[8], rw4h[8], rw4l[8], pd4h[8], pd4l[8], pb4h[8], pb4l[8];
short cl5h[8], cl5l[8], rw5h[8], rw5l[8], pd5h[8], pd5l[8], pb5h[8], pb5l[8];
short cl6h[8], cl6l[8], rw6h[8], rw6l[8], pd6h[8], pd6l[8], pb6h[8], pb6l[8];
short cl7h[8], cl7l[8], rw7h[8], rw7l[8], pd7h[8], pd7l[8], pb7h[8], pb7l[8];
/**/
/* Sub-Routines */
void stp01(void), stp02(void), stp03(void), stp04(void);
void stp05(void), stp06(void), stp07(void), stp08(void);
void stp09(void), stp10(void), stp11(void), stp12(void);
void stp13(void), stp14(void), stp15(void), stp16(void);
void stp17(void), stp18(void), stp19(void), stp20(void);
void stp21(void), stp22(void), stp23(void), stp24(void);
void stp25(void), stp26(void), stp27(void), stp28(void);
void stp29(void), stp30(void), stp31(void);
void printans(void), print2p(void);
/**/
/* Main Program */
/* Set n25=24 */
int main(){
short n;
printf("\n** 'Complete Euler Squares' of Simultaneous MS77: **\n");
printf("** Both Self-Complementary and Pan-Diagonal **\n");
for(n=0; n<50; n++){nm[n]=0; uflg[n]=0;}
for(n=0; n<7; n++){
cl1h[n]=0; rw1h[n]=0; pd1h[n]=0; pb1h[n]=0;
cl1l[n]=0; rw1l[n]=0; pd1l[n]=0; pb1l[n]=0;
cl2h[n]=0; rw2h[n]=0; pd2h[n]=0; pb2h[n]=0;
cl2l[n]=0; rw2l[n]=0; pd2l[n]=0; pb2l[n]=0;
cl3h[n]=0; rw3h[n]=0; pd3h[n]=0; pb3h[n]=0;
cl3l[n]=0; rw3l[n]=0; pd3l[n]=0; pb3l[n]=0;
cl4h[n]=0; rw4h[n]=0; pd4h[n]=0; pb4h[n]=0;
cl4l[n]=0; rw4l[n]=0; pd4l[n]=0; pb4l[n]=0;
cl5h[n]=0; rw5h[n]=0; pd5h[n]=0; pb5h[n]=0;
cl5l[n]=0; rw5l[n]=0; pd5l[n]=0; pb5l[n]=0;
cl6h[n]=0; rw6h[n]=0; pd6h[n]=0; pb6h[n]=0;
cl6l[n]=0; rw6l[n]=0; pd6l[n]=0; pb6l[n]=0;
cl7h[n]=0; rw7h[n]=0; pd7h[n]=0; pb7h[n]=0;
cl7l[n]=0; rw7l[n]=0; pd7l[n]=0; pb7l[n]=0;
}
nm[25]=24; uflg[24]=1;
cl4h[3]=1; cl4l[3]=1; rw4h[3]=1; rw4l[3]=1;
pd1h[3]=1; pd1l[3]=1; pb7h[3]=1; pb7l[3]=1;
LSM=168; CC=48; S6N=144; cnt=0; cnt3=0;
stp01();
if(cnt3>0){print2p();}
printf("** [Count = %d] OK! *\n", cnt);
return 0;
}
/* Begin the Search */
/* Set n1 & n49 & n1<n49 */
void stp01(){
short a, na, b, nb, d, nd;
for(a=0; a<7; a++){na=6-a;
if((cl1h[a]==0)&&(rw1h[a]==0)&&(pd1h[a]==0)&&(pb1h[a]==0)){
if((cl7h[na]==0)&&(rw7h[na]==0)&&(pd1h[na]==0)&&(pb6h[na]==0)){
for(b=0; b<7; b++){nb=6-b; d=7*a+b; nd=CC-d;
if((uflg[d]==0)&&(uflg[nd]==0)&&(d<nd)){
if((cl1l[b]==0)&&(rw1l[b]==0)&&(pd1l[b]==0)&&(pb1l[b]==0)){
if((cl7l[nb]==0)&&(rw7l[nb]==0)&&(pd1l[nb]==0)&&(pb6l[nb]==0)){

```

```

nm[1]=d; nm[49]=nd; uflg[d]=1; uflg[nd]=1; cnt2=0;
cl1h[a]=1; rw1h[a]=1; pd1h[a]=1; pb1h[a]=1;
cl1l[b]=1; rw1l[b]=1; pd1l[b]=1; pb1l[b]=1;
cl7h[na]=1; rw7h[na]=1; pd1h[na]=1; pb6h[na]=1;
cl7l[nb]=1; rw7l[nb]=1; pd1l[nb]=1; pb6l[nb]=1;
stp02();
cl1h[a]=0; rw1h[a]=0; pd1h[a]=0; pb1h[a]=0;
cl1l[b]=0; rw1l[b]=0; pd1l[b]=0; pb1l[b]=0;
cl7h[na]=0; rw7h[na]=0; pd1h[na]=0; pb6h[na]=0;
cl7l[nb]=0; rw7l[nb]=0; pd1l[nb]=0; pb6l[nb]=0;
uflg[d]=0; uflg[nd]=0; }}}
}
}
}
/* Set n9 & n41 */
void stp02(){
short a, na, b, nb, d, nd;
for(a=0; a<7; a++){na=6-a;
if((cl2h[a]==0)&&(rw2h[a]==0)&&(pd1h[a]==0)&&(pb3h[a]==0)){
if((cl6h[na]==0)&&(rw6h[na]==0)&&(pd1h[na]==0)&&(pb4h[na]==0)){
for(b=0; b<7; b++){nb=6-b; d=7*a+b; nd=CC-d;
if((uflg[d]==0)&&(uflg[nd]==0)){
if((cl2l[b]==0)&&(rw2l[b]==0)&&(pd1l[b]==0)&&(pb3l[b]==0)){
if((cl6l[nb]==0)&&(rw6l[nb]==0)&&(pd1l[nb]==0)&&(pb4l[nb]==0)){
nm[9]=d; nm[41]=nd; uflg[d]=1; uflg[nd]=1;
cl2h[a]=1; rw2h[a]=1; pd1h[a]=1; pb3h[a]=1;
cl2l[b]=1; rw2l[b]=1; pd1l[b]=1; pb3l[b]=1;
cl6h[na]=1; rw6h[na]=1; pd1h[na]=1; pb4h[na]=1;
cl6l[nb]=1; rw6l[nb]=1; pd1l[nb]=1; pb4l[nb]=1;
stp03();
cl2h[a]=0; rw2h[a]=0; pd1h[a]=0; pb3h[a]=0;
cl2l[b]=0; rw2l[b]=0; pd1l[b]=0; pb3l[b]=0;
cl6h[na]=0; rw6h[na]=0; pd1h[na]=0; pb4h[na]=0;
cl6l[nb]=0; rw6l[nb]=0; pd1l[nb]=0; pb4l[nb]=0;
uflg[d]=0; uflg[nd]=0; }}}
}}}
}
}
}
/* Set n17 & n33 */
void stp03(){
short a, na, b, nb, d, nd;
for(a=0; a<7; a++){na=6-a;
if((cl3h[a]==0)&&(rw3h[a]==0)&&(pd1h[a]==0)&&(pb5h[a]==0)){
if((cl5h[na]==0)&&(rw5h[na]==0)&&(pd1h[na]==0)&&(pb2h[na]==0)){
for(b=0; b<7; b++){nb=6-b; d=7*a+b; nd=CC-d;
if((uflg[d]==0)&&(uflg[nd]==0)){
if((cl3l[b]==0)&&(rw3l[b]==0)&&(pd1l[b]==0)&&(pb5l[b]==0)){
if((cl5l[nb]==0)&&(rw5l[nb]==0)&&(pd1l[nb]==0)&&(pb2l[nb]==0)){
nm[17]=d; nm[33]=nd; uflg[d]=1; uflg[nd]=1;
cl3h[a]=1; rw3h[a]=1; pd1h[a]=1; pb5h[a]=1;
cl3l[b]=1; rw3l[b]=1; pd1l[b]=1; pb5l[b]=1;
cl5h[na]=1; rw5h[na]=1; pd1h[na]=1; pb2h[na]=1;
cl5l[nb]=1; rw5l[nb]=1; pd1l[nb]=1; pb2l[nb]=1;
stp04();
cl3h[a]=0; rw3h[a]=0; pd1h[a]=0; pb5h[a]=0;
cl3l[b]=0; rw3l[b]=0; pd1l[b]=0; pb5l[b]=0;
cl5h[na]=0; rw5h[na]=0; pd1h[na]=0; pb2h[na]=0;
cl5l[nb]=0; rw5l[nb]=0; pd1l[nb]=0; pb2l[nb]=0;
uflg[d]=0; uflg[nd]=0; }}}
}}}
}
}
}
/* Set n7 & n43 & (n1<n7)&(n1<n43) */
void stp04(){

```

```

short a, na, b, nb, d, nd;
for(a=0; a<7; a++){na=6-a;
  if((cl 7h[a]==0)&&(rw1h[a]==0)&&(pd7h[a]==0)&&(pb7h[a]==0)){
    if((cl 1h[na]==0)&&(rw7h[na]==0)&&(pd2h[na]==0)&&(pb7h[na]==0)){
      for(b=0; b<7; b++){nb=6-b; d=7*a+b; nd=CC-d;
        if((ufl g[d]==0)&&(ufl g[nd]==0)&&(nm[1]<d)&&(nm[1]<nd)){
          if((cl 7l [b]==0)&&(rw7l [b]==0)&&(pd7l [b]==0)&&(pb7l [b]==0)){
            if((cl 1l [nb]==0)&&(rw7l [nb]==0)&&(pd2l [nb]==0)&&(pb7l [nb]==0)){
              nm[7]=d; nm[43]=nd; ufl g[d]=1; ufl g[nd]=1;
              cl 7h[a]=1; rw1h[a]=1; pd7h[a]=1; pb7h[a]=1;
              cl 7l [b]=1; rw1l [b]=1; pd7l [b]=1; pb7l [b]=1;
              cl 1h[na]=1; rw7h[na]=1; pd2h[na]=1; pb7h[na]=1;
              cl 1l [nb]=1; rw7l [nb]=1; pd2l [nb]=1; pb7l [nb]=1;
              stp05();
              cl 7h[a]=0; rw1h[a]=0; pd7h[a]=0; pb7h[a]=0;
              cl 7l [b]=0; rw1l [b]=0; pd7l [b]=0; pb7l [b]=0;
              cl 1h[na]=0; rw7h[na]=0; pd2h[na]=0; pb7h[na]=0;
              cl 1l [nb]=0; rw7l [nb]=0; pd2l [nb]=0; pb7l [nb]=0;
              ufl g[d]=0; ufl g[nd]=0; }}}
            }}}
          }
        }
      }
    }
  }
}
/* Set n13 & n37 */
void stp05(){
short a, na, b, nb, d, nd;
for(a=0; a<7; a++){na=6-a;
  if((cl 6h[a]==0)&&(rw2h[a]==0)&&(pd5h[a]==0)&&(pb7h[a]==0)){
    if((cl 2h[na]==0)&&(rw6h[na]==0)&&(pd4h[na]==0)&&(pb7h[na]==0)){
      for(b=0; b<7; b++){nb=6-b; d=7*a+b; nd=CC-d;
        if((ufl g[d]==0)&&(ufl g[nd]==0)){
          if((cl 6l [b]==0)&&(rw2l [b]==0)&&(pd5l [b]==0)&&(pb7l [b]==0)){
            if((cl 2l [nb]==0)&&(rw6l [nb]==0)&&(pd4l [nb]==0)&&(pb7l [nb]==0)){
              nm[13]=d; nm[37]=nd; ufl g[d]=1; ufl g[nd]=1;
              cl 6h[a]=1; rw2h[a]=1; pd5h[a]=1; pb7h[a]=1;
              cl 6l [b]=1; rw2l [b]=1; pd5l [b]=1; pb7l [b]=1;
              cl 2h[na]=1; rw6h[na]=1; pd4h[na]=1; pb7h[na]=1;
              cl 2l [nb]=1; rw6l [nb]=1; pd4l [nb]=1; pb7l [nb]=1;
              stp06();
              cl 6h[a]=0; rw2h[a]=0; pd5h[a]=0; pb7h[a]=0;
              cl 6l [b]=0; rw2l [b]=0; pd5l [b]=0; pb7l [b]=0;
              cl 2h[na]=0; rw6h[na]=0; pd4h[na]=0; pb7h[na]=0;
              cl 2l [nb]=0; rw6l [nb]=0; pd4l [nb]=0; pb7l [nb]=0;
              ufl g[d]=0; ufl g[nd]=0; }}}
            }}}
          }
        }
      }
    }
  }
}
/* Set n19 & n31 */
void stp06(){
short a, na, b, nb, d, nd;
for(a=0; a<7; a++){na=6-a;
  if((cl 5h[a]==0)&&(rw3h[a]==0)&&(pd3h[a]==0)&&(pb7h[a]==0)){
    if((cl 3h[na]==0)&&(rw5h[na]==0)&&(pd6h[na]==0)&&(pb7h[na]==0)){
      for(b=0; b<7; b++){nb=6-b; d=7*a+b; nd=CC-d;
        if((ufl g[d]==0)&&(ufl g[nd]==0)){
          if((cl 5l [b]==0)&&(rw3l [b]==0)&&(pd3l [b]==0)&&(pb7l [b]==0)){
            if((cl 3l [nb]==0)&&(rw5l [nb]==0)&&(pd6l [nb]==0)&&(pb7l [nb]==0)){
              nm[19]=d; nm[31]=nd; ufl g[d]=1; ufl g[nd]=1;
              cl 5h[a]=1; rw3h[a]=1; pd3h[a]=1; pb7h[a]=1;
              cl 5l [b]=1; rw3l [b]=1; pd3l [b]=1; pb7l [b]=1;
              cl 3h[na]=1; rw5h[na]=1; pd6h[na]=1; pb7h[na]=1;
              cl 3l [nb]=1; rw5l [nb]=1; pd6l [nb]=1; pb7l [nb]=1;
              stp07();
              cl 5h[a]=0; rw3h[a]=0; pd3h[a]=0; pb7h[a]=0;
              cl 5l [b]=0; rw3l [b]=0; pd3l [b]=0; pb7l [b]=0;
            }
          }
        }
      }
    }
  }
}

```

```

        cl 3h[na]=0; rw5h[na]=0; pd6h[na]=0; pb7h[na]=0;
        cl 3l[nb]=0; rw5l[nb]=0; pd6l[nb]=0; pb7l[nb]=0;
        uflg[d]=0; uflg[nd]=0; }}}
    }
}
/* Search Level 2: */
/* Set n2 & n48 */
void stp07(){
    short a, na, b, nb, d, nd;
    for(a=6; a>=0; a--){na=6-a;
        if((cl 2h[a]==0)&&(rw1h[a]==0)&&(pd2h[a]==0)&&(pb2h[a]==0)){
            if((cl 6h[na]==0)&&(rw7h[na]==0)&&(pd7h[na]==0)&&(pb5h[na]==0)){
                for(b=6; b>=0; b--){nb=6-b; d=7*a+b; nd=CC-d;
                    if((uflg[d]==0)&&(uflg[nd]==0)){
                        if((cl 2l[b]==0)&&(rw1l[b]==0)&&(pd2l[b]==0)&&(pb2l[b]==0)){
                            if((cl 6l[nb]==0)&&(rw7l[nb]==0)&&(pd7l[nb]==0)&&(pb5l[nb]==0)){
                                nm[2]=d; nm[48]=nd; uflg[d]=1; uflg[nd]=1;
                                cl 2h[a]=1; rw1h[a]=1; pd2h[a]=1; pb2h[a]=1;
                                cl 2l[b]=1; rw1l[b]=1; pd2l[b]=1; pb2l[b]=1;
                                cl 6h[na]=1; rw7h[na]=1; pd7h[na]=1; pb5h[na]=1;
                                cl 6l[nb]=1; rw7l[nb]=1; pd7l[nb]=1; pb5l[nb]=1;
                                stp08();
                                cl 2h[a]=0; rw1h[a]=0; pd2h[a]=0; pb2h[a]=0;
                                cl 2l[b]=0; rw1l[b]=0; pd2l[b]=0; pb2l[b]=0;
                                cl 6h[na]=0; rw7h[na]=0; pd7h[na]=0; pb5h[na]=0;
                                cl 6l[nb]=0; rw7l[nb]=0; pd7l[nb]=0; pb5l[nb]=0;
                                uflg[d]=0; uflg[nd]=0; }}}
                            }}}
                    }
                }
            }
        }
    }
}
/* Set n3 & n47 */
void stp08(){
    short a, na, b, nb, d, nd;
    for(a=6; a>=0; a--){na=6-a;
        if((cl 3h[a]==0)&&(rw1h[a]==0)&&(pd3h[a]==0)&&(pb3h[a]==0)){
            if((cl 5h[na]==0)&&(rw7h[na]==0)&&(pd6h[na]==0)&&(pb4h[na]==0)){
                for(b=6; b>=0; b--){nb=6-b; d=7*a+b; nd=CC-d;
                    if((uflg[d]==0)&&(uflg[nd]==0)){
                        if((cl 3l[b]==0)&&(rw1l[b]==0)&&(pd3l[b]==0)&&(pb3l[b]==0)){
                            if((cl 5l[nb]==0)&&(rw7l[nb]==0)&&(pd6l[nb]==0)&&(pb4l[nb]==0)){
                                nm[3]=d; nm[47]=nd; uflg[d]=1; uflg[nd]=1;
                                cl 3h[a]=1; rw1h[a]=1; pd3h[a]=1; pb3h[a]=1;
                                cl 3l[b]=1; rw1l[b]=1; pd3l[b]=1; pb3l[b]=1;
                                cl 5h[na]=1; rw7h[na]=1; pd6h[na]=1; pb4h[na]=1;
                                cl 5l[nb]=1; rw7l[nb]=1; pd6l[nb]=1; pb4l[nb]=1;
                                stp09();
                                cl 3h[a]=0; rw1h[a]=0; pd3h[a]=0; pb3h[a]=0;
                                cl 3l[b]=0; rw1l[b]=0; pd3l[b]=0; pb3l[b]=0;
                                cl 5h[na]=0; rw7h[na]=0; pd6h[na]=0; pb4h[na]=0;
                                cl 5l[nb]=0; rw7l[nb]=0; pd6l[nb]=0; pb4l[nb]=0;
                                uflg[d]=0; uflg[nd]=0; }}}
                            }}}
                    }
                }
            }
        }
    }
}
/* Set n6 & n44 */
void stp09(){
    short a, na, b, nb, d, nd;
    for(a=0; a<7; a++){na=6-a;
        if((cl 6h[a]==0)&&(rw1h[a]==0)&&(pd6h[a]==0)&&(pb6h[a]==0)){
            if((cl 2h[na]==0)&&(rw7h[na]==0)&&(pd3h[na]==0)&&(pb1h[na]==0)){
                for(b=0; b<7; b++){nb=6-b; d=7*a+b; nd=CC-d;
                    if((uflg[d]==0)&&(uflg[nd]==0)){
                        if((cl 6l[b]==0)&&(rw1l[b]==0)&&(pd6l[b]==0)&&(pb6l[b]==0)){

```

```

    i f((c1 2l [nb]==0)&&(rw7l [nb]==0)&&(pd3l [nb]==0)&&(pb1l [nb]==0)){
        nm[6]=d; nm[44]=nd; ufl g[d]=1; ufl g[nd]=1;
        cl 6h[a]=1; rw1h[a]=1; pd6h[a]=1; pb6h[a]=1;
        cl 6l [b]=1; rw1l [b]=1; pd6l [b]=1; pb6l [b]=1;
        cl 2h[na]=1; rw7h[na]=1; pd3h[na]=1; pb1h[na]=1;
        cl 2l [nb]=1; rw7l [nb]=1; pd3l [nb]=1; pb1l [nb]=1;
        stp10();
        cl 6h[a]=0; rw1h[a]=0; pd6h[a]=0; pb6h[a]=0;
        cl 6l [b]=0; rw1l [b]=0; pd6l [b]=0; pb6l [b]=0;
        cl 2h[na]=0; rw7h[na]=0; pd3h[na]=0; pb1h[na]=0;
        cl 2l [nb]=0; rw7l [nb]=0; pd3l [nb]=0; pb1l [nb]=0;
        ufl g[d]=0; ufl g[nd]=0; }}
    }}}
}
}
/* Set n5 & n45 */
void stp10(){
    short a, na, b, nb, d, nd;
    for(a=0; a<7; a++){na=6-a;
        i f((c1 5h[a]==0)&&(rw1h[a]==0)&&(pd5h[a]==0)&&(pb5h[a]==0)){
            i f((c1 3h[na]==0)&&(rw7h[na]==0)&&(pd4h[na]==0)&&(pb2h[na]==0)){
                for(b=0; b<7; b++){nb=6-b; d=7*a+b; nd=CC-d;
                    i f((ufl g[d]==0)&&(ufl g[nd]==0)){
                        i f((c1 5l [b]==0)&&(rw1l [b]==0)&&(pd5l [b]==0)&&(pb5l [b]==0)){
                            i f((c1 3l [nb]==0)&&(rw7l [nb]==0)&&(pd4l [nb]==0)&&(pb2l [nb]==0)){
                                nm[5]=d; nm[45]=nd; ufl g[d]=1; ufl g[nd]=1;
                                cl 5h[a]=1; rw1h[a]=1; pd5h[a]=1; pb5h[a]=1;
                                cl 5l [b]=1; rw1l [b]=1; pd5l [b]=1; pb5l [b]=1;
                                cl 3h[na]=1; rw7h[na]=1; pd4h[na]=1; pb2h[na]=1;
                                cl 3l [nb]=1; rw7l [nb]=1; pd4l [nb]=1; pb2l [nb]=1;
                                stp11();
                                cl 5h[a]=0; rw1h[a]=0; pd5h[a]=0; pb5h[a]=0;
                                cl 5l [b]=0; rw1l [b]=0; pd5l [b]=0; pb5l [b]=0;
                                cl 3h[na]=0; rw7h[na]=0; pd4h[na]=0; pb2h[na]=0;
                                cl 3l [nb]=0; rw7l [nb]=0; pd4l [nb]=0; pb2l [nb]=0;
                                ufl g[d]=0; ufl g[nd]=0; }}}
                            }}}
                        }
                    }
                }
            }
        }
    }
}
/* Set n4 & n46 */
void stp11(){
    short a, na, b, nb, d, nd;
    for(a=0; a<7; a++){na=6-a;
        i f((c1 4h[a]==0)&&(rw1h[a]==0)&&(pd4h[a]==0)&&(pb4h[a]==0)){
            i f((c1 4h[na]==0)&&(rw7h[na]==0)&&(pd5h[na]==0)&&(pb3h[na]==0)){
                for(b=0; b<7; b++){nb=6-b; d=7*a+b; nd=CC-d;
                    i f((ufl g[d]==0)&&(ufl g[nd]==0)){
                        i f((c1 4l [b]==0)&&(rw1l [b]==0)&&(pd4l [b]==0)&&(pb4l [b]==0)){
                            i f((c1 4l [nb]==0)&&(rw7l [nb]==0)&&(pd5l [nb]==0)&&(pb3l [nb]==0)){
                                nm[4]=d; nm[46]=nd; ufl g[d]=1; ufl g[nd]=1;
                                cl 4h[a]=1; rw1h[a]=1; pd4h[a]=1; pb4h[a]=1;
                                cl 4l [b]=1; rw1l [b]=1; pd4l [b]=1; pb4l [b]=1;
                                cl 4h[na]=1; rw7h[na]=1; pd5h[na]=1; pb3h[na]=1;
                                cl 4l [nb]=1; rw7l [nb]=1; pd5l [nb]=1; pb3l [nb]=1;
                                stp12();
                                cl 4h[a]=0; rw1h[a]=0; pd4h[a]=0; pb4h[a]=0;
                                cl 4l [b]=0; rw1l [b]=0; pd4l [b]=0; pb4l [b]=0;
                                cl 4h[na]=0; rw7h[na]=0; pd5h[na]=0; pb3h[na]=0;
                                cl 4l [nb]=0; rw7l [nb]=0; pd5l [nb]=0; pb3l [nb]=0;
                                ufl g[d]=0; ufl g[nd]=0; }}}
                            }}}
                        }
                    }
                }
            }
        }
    }
}
}
}
/* Set n8(<n2) & n42 */

```

```

void stp12(){
short a, na, b, nb, d, nd;
for(a=0; a<7; a++){na=6-a;
if((cl1h[a]==0)&&(rw2h[a]==0)&&(pd7h[a]==0)&&(pb2h[a]==0)){
if((cl7h[na]==0)&&(rw6h[na]==0)&&(pd2h[na]==0)&&(pb5h[na]==0)){
for(b=0; b<7; b++){nb=6-b; d=7*a+b; nd=CC-d;
if((uflg[d]==0)&&(uflg[nd]==0)&&(d<nm[2])){
if((cl1l[b]==0)&&(rw2l[b]==0)&&(pd7l[b]==0)&&(pb2l[b]==0)){
if((cl7l[nb]==0)&&(rw6l[nb]==0)&&(pd2l[nb]==0)&&(pb5l[nb]==0)){
nm[8]=d; nm[42]=nd; uflg[d]=1; uflg[nd]=1;
cl1h[a]=1; rw2h[a]=1; pd7h[a]=1; pb2h[a]=1;
cl1l[b]=1; rw2l[b]=1; pd7l[b]=1; pb2l[b]=1;
cl7h[na]=1; rw6h[na]=1; pd2h[na]=1; pb5h[na]=1;
cl7l[nb]=1; rw6l[nb]=1; pd2l[nb]=1; pb5l[nb]=1;
stp13();
cl1h[a]=0; rw2h[a]=0; pd7h[a]=0; pb2h[a]=0;
cl1l[b]=0; rw2l[b]=0; pd7l[b]=0; pb2l[b]=0;
cl7h[na]=0; rw6h[na]=0; pd2h[na]=0; pb5h[na]=0;
cl7l[nb]=0; rw6l[nb]=0; pd2l[nb]=0; pb5l[nb]=0;
uflg[d]=0; uflg[nd]=0; }}}
}}}
}
}
/* Set n15 & n35 & n1+n2+n3+n8+n9+n15=S6N? */
void stp13(){
short a, na, b, nb, d, nd;
for(a=0; a<7; a++){na=6-a;
if((cl1h[a]==0)&&(rw3h[a]==0)&&(pd6h[a]==0)&&(pb3h[a]==0)){
if((cl7h[na]==0)&&(rw5h[na]==0)&&(pd3h[na]==0)&&(pb4h[na]==0)){
for(b=0; b<7; b++){nb=6-b; d=7*a+b; nd=CC-d;
if((uflg[d]==0)&&(uflg[nd]==0)){
if((cl1l[b]==0)&&(rw3l[b]==0)&&(pd6l[b]==0)&&(pb3l[b]==0)){
if((cl7l[nb]==0)&&(rw5l[nb]==0)&&(pd3l[nb]==0)&&(pb4l[nb]==0)){
if(nm[1]+nm[2]+nm[3]+nm[8]+nm[9]+d==S6N){
nm[15]=d; nm[35]=nd; uflg[d]=1; uflg[nd]=1;
cl1h[a]=1; rw3h[a]=1; pd6h[a]=1; pb3h[a]=1;
cl1l[b]=1; rw3l[b]=1; pd6l[b]=1; pb3l[b]=1;
cl7h[na]=1; rw5h[na]=1; pd3h[na]=1; pb4h[na]=1;
cl7l[nb]=1; rw5l[nb]=1; pd3l[nb]=1; pb4l[nb]=1;
stp14();
cl1h[a]=0; rw3h[a]=0; pd6h[a]=0; pb3h[a]=0;
cl1l[b]=0; rw3l[b]=0; pd6l[b]=0; pb3l[b]=0;
cl7h[na]=0; rw5h[na]=0; pd3h[na]=0; pb4h[na]=0;
cl7l[nb]=0; rw5l[nb]=0; pd3l[nb]=0; pb4l[nb]=0;
uflg[d]=0; uflg[nd]=0; }}}}}
}}}
}
}
/* Set n14 & n36 */
void stp14(){
short a, na, b, nb, d, nd;
for(a=0; a<7; a++){na=6-a;
if((cl7h[a]==0)&&(rw2h[a]==0)&&(pd6h[a]==0)&&(pb1h[a]==0)){
if((cl1h[na]==0)&&(rw6h[na]==0)&&(pd3h[na]==0)&&(pb6h[na]==0)){
for(b=0; b<7; b++){nb=6-b; d=7*a+b; nd=CC-d;
if((uflg[d]==0)&&(uflg[nd]==0)){
if((cl7l[b]==0)&&(rw2l[b]==0)&&(pd6l[b]==0)&&(pb1l[b]==0)){
if((cl1l[nb]==0)&&(rw6l[nb]==0)&&(pd3l[nb]==0)&&(pb6l[nb]==0)){
nm[14]=d; nm[36]=nd; uflg[d]=1; uflg[nd]=1;
cl7h[a]=1; rw2h[a]=1; pd6h[a]=1; pb1h[a]=1;
cl7l[b]=1; rw2l[b]=1; pd6l[b]=1; pb1l[b]=1;
cl1h[na]=1; rw6h[na]=1; pd3h[na]=1; pb6h[na]=1;
cl1l[nb]=1; rw6l[nb]=1; pd3l[nb]=1; pb6l[nb]=1;
stp15();

```

```

        cl 7h[a]=0; rw2h[a]=0; pd6h[a]=0; pb1h[a]=0;
        cl 7l [b]=0; rw2l [b]=0; pd6l [b]=0; pb1l [b]=0;
        cl 1h[na]=0; rw6h[na]=0; pd3h[na]=0; pb6h[na]=0;
        cl 1l [nb]=0; rw6l [nb]=0; pd3l [nb]=0; pb6l [nb]=0;
        ufl g[d]=0; ufl g[nd]=0; }}}
    }}}
}
}
/* Set n21 & n29 & n5+n6+n7+n13+n14+n21=S6N? */
void stp15(){
    short a, na, b, nb, d, nd;
    for(a=0; a<7; a++){na=6-a;
        if((cl 7h[a]==0)&&(rw3h[a]==0)&&(pd5h[a]==0)&&(pb2h[a]==0)){
            if((cl 1h[na]==0)&&(rw5h[na]==0)&&(pd4h[na]==0)&&(pb5h[na]==0)){
                for(b=0; b<7; b++){nb=6-b; d=7*a+b; nd=CC-d;
                    if((ufl g[d]==0)&&(ufl g[nd]==0)){
                        if((cl 7l [b]==0)&&(rw3l [b]==0)&&(pd5l [b]==0)&&(pb2l [b]==0)){
                            if((cl 1l [nb]==0)&&(rw5l [nb]==0)&&(pd4l [nb]==0)&&(pb5l [nb]==0)){
                                if(nm[5]+nm[6]+nm[7]+nm[13]+nm[14]+d==S6N){
                                    nm[21]=d; nm[29]=nd; ufl g[d]=1; ufl g[nd]=1;
                                    cl 7h[a]=1; rw3h[a]=1; pd5h[a]=1; pb2h[a]=1;
                                    cl 7l [b]=1; rw3l [b]=1; pd5l [b]=1; pb2l [b]=1;
                                    cl 1h[na]=1; rw5h[na]=1; pd4h[na]=1; pb5h[na]=1;
                                    cl 1l [nb]=1; rw5l [nb]=1; pd4l [nb]=1; pb5l [nb]=1;
                                    stp16();
                                    cl 7h[a]=0; rw3h[a]=0; pd5h[a]=0; pb2h[a]=0;
                                    cl 7l [b]=0; rw3l [b]=0; pd5l [b]=0; pb2l [b]=0;
                                    cl 1h[na]=0; rw5h[na]=0; pd4h[na]=0; pb5h[na]=0;
                                    cl 1l [nb]=0; rw5l [nb]=0; pd4l [nb]=0; pb5l [nb]=0;
                                    ufl g[d]=0; ufl g[nd]=0; }}}}}
                    }}}
                }
            }
        }
    }
}
/* Set n22 & n28 */
void stp16(){
    short a, na, b, nb, d, nd;
    for(a=6; a>=0; a--){na=6-a;
        if((cl 1h[a]==0)&&(rw4h[a]==0)&&(pd5h[a]==0)&&(pb4h[a]==0)){
            if((cl 7h[na]==0)&&(rw4h[na]==0)&&(pd4h[na]==0)&&(pb3h[na]==0)){
                for(b=6; b>=0; b--){nb=6-b; d=7*a+b; nd=CC-d;
                    if((ufl g[d]==0)&&(ufl g[nd]==0)){
                        if((cl 1l [b]==0)&&(rw4l [b]==0)&&(pd5l [b]==0)&&(pb4l [b]==0)){
                            if((cl 7l [nb]==0)&&(rw4l [nb]==0)&&(pd4l [nb]==0)&&(pb3l [nb]==0)){
                                nm[22]=d; nm[28]=nd; ufl g[d]=1; ufl g[nd]=1;
                                cl 1h[a]=1; rw4h[a]=1; pd5h[a]=1; pb4h[a]=1;
                                cl 1l [b]=1; rw4l [b]=1; pd5l [b]=1; pb4l [b]=1;
                                cl 7h[na]=1; rw4h[na]=1; pd4h[na]=1; pb3h[na]=1;
                                cl 7l [nb]=1; rw4l [nb]=1; pd4l [nb]=1; pb3l [nb]=1;
                                stp17();
                                cl 1h[a]=0; rw4h[a]=0; pd5h[a]=0; pb4h[a]=0;
                                cl 1l [b]=0; rw4l [b]=0; pd5l [b]=0; pb4l [b]=0;
                                cl 7h[na]=0; rw4h[na]=0; pd4h[na]=0; pb3h[na]=0;
                                cl 7l [nb]=0; rw4l [nb]=0; pd4l [nb]=0; pb3l [nb]=0;
                                ufl g[d]=0; ufl g[nd]=0; }}}}}
                    }}}
                }
            }
        }
    }
}
/* Search Level 3: */
/* Set n10 & n40 */
void stp17(){
    short a, na, b, nb, d, nd;
    for(a=0; a<7; a++){na=6-a;
        if((cl 3h[a]==0)&&(rw2h[a]==0)&&(pd2h[a]==0)&&(pb4h[a]==0)){
            if((cl 5h[na]==0)&&(rw6h[na]==0)&&(pd7h[na]==0)&&(pb3h[na]==0)){

```



```

}
}
/* Set n11 & n39 */
void stp20(){
short a, na, b, nb, d, nd;
for(a=6; a>=0; a--){na=6-a;
if((cl4h[a]==0)&&(rw2h[a]==0)&&(pd3h[a]==0)&&(pb5h[a]==0)){
if((cl4h[na]==0)&&(rw6h[na]==0)&&(pd6h[na]==0)&&(pb2h[na]==0)){
for(b=6; b>=0; b--){nb=6-b; d=7*a+b; nd=CC-d;
if((uflg[d]==0)&&(uflg[nd]==0)){
if((cl4l[b]==0)&&(rw2l[b]==0)&&(pd3l[b]==0)&&(pb5l[b]==0)){
if((cl4l[nb]==0)&&(rw6l[nb]==0)&&(pd6l[nb]==0)&&(pb2l[nb]==0)){
nm[11]=d; nm[39]=nd; uflg[d]=1; uflg[nd]=1;
cl4h[a]=1; rw2h[a]=1; pd3h[a]=1; pb5h[a]=1;
cl4l[b]=1; rw2l[b]=1; pd3l[b]=1; pb5l[b]=1;
cl4h[na]=1; rw6h[na]=1; pd6h[na]=1; pb2h[na]=1;
cl4l[nb]=1; rw6l[nb]=1; pd6l[nb]=1; pb2l[nb]=1;
stp21();
cl4h[a]=0; rw2h[a]=0; pd3h[a]=0; pb5h[a]=0;
cl4l[b]=0; rw2l[b]=0; pd3l[b]=0; pb5l[b]=0;
cl4h[na]=0; rw6h[na]=0; pd6h[na]=0; pb2h[na]=0;
cl4l[nb]=0; rw6l[nb]=0; pd6l[nb]=0; pb2l[nb]=0;
uflg[d]=0; uflg[nd]=0; }}}
}}}
}
}
/* Set n20 & n30 */
void stp21(){
short a, na, b, nb, d, nd;
for(a=0; a<7; a++){na=6-a;
if((cl6h[a]==0)&&(rw3h[a]==0)&&(pd4h[a]==0)&&(pb1h[a]==0)){
if((cl2h[na]==0)&&(rw5h[na]==0)&&(pd5h[na]==0)&&(pb6h[na]==0)){
for(b=0; b<7; b++){nb=6-b; d=7*a+b; nd=CC-d;
if((uflg[d]==0)&&(uflg[nd]==0)){
if((cl6l[b]==0)&&(rw3l[b]==0)&&(pd4l[b]==0)&&(pb1l[b]==0)){
if((cl2l[nb]==0)&&(rw5l[nb]==0)&&(pd5l[nb]==0)&&(pb6l[nb]==0)){
nm[20]=d; nm[30]=nd; uflg[d]=1; uflg[nd]=1;
cl6h[a]=1; rw3h[a]=1; pd4h[a]=1; pb1h[a]=1;
cl6l[b]=1; rw3l[b]=1; pd4l[b]=1; pb1l[b]=1;
cl2h[na]=1; rw5h[na]=1; pd5h[na]=1; pb6h[na]=1;
cl2l[nb]=1; rw5l[nb]=1; pd5l[nb]=1; pb6l[nb]=1;
stp22();
cl6h[a]=0; rw3h[a]=0; pd4h[a]=0; pb1h[a]=0;
cl6l[b]=0; rw3l[b]=0; pd4l[b]=0; pb1l[b]=0;
cl2h[na]=0; rw5h[na]=0; pd5h[na]=0; pb6h[na]=0;
cl2l[nb]=0; rw5l[nb]=0; pd5l[nb]=0; pb6l[nb]=0;
uflg[d]=0; uflg[nd]=0; }}}
}}}
}
}
/* Set n23 & n27 */
void stp22(){
short a, na, b, nb, d, nd;
for(a=0; a<7; a++){na=6-a;
if((cl2h[a]==0)&&(rw4h[a]==0)&&(pd6h[a]==0)&&(pb5h[a]==0)){
if((cl6h[na]==0)&&(rw4h[na]==0)&&(pd3h[na]==0)&&(pb2h[na]==0)){
for(b=0; b<7; b++){nb=6-b; d=7*a+b; nd=CC-d;
if((uflg[d]==0)&&(uflg[nd]==0)){
if((cl2l[b]==0)&&(rw4l[b]==0)&&(pd6l[b]==0)&&(pb5l[b]==0)){
if((cl6l[nb]==0)&&(rw4l[nb]==0)&&(pd3l[nb]==0)&&(pb2l[nb]==0)){
nm[23]=d; nm[27]=nd; uflg[d]=1; uflg[nd]=1;
cl2h[a]=1; rw4h[a]=1; pd6h[a]=1; pb5h[a]=1;
cl2l[b]=1; rw4l[b]=1; pd6l[b]=1; pb5l[b]=1;
cl6h[na]=1; rw4h[na]=1; pd3h[na]=1; pb2h[na]=1;

```

```

        cl 6l [nb]=1; rw4l [nb]=1; pd3l [nb]=1; pb2l [nb]=1;
        stp23();
        cl 2h[a]=0; rw4h[a]=0; pd6h[a]=0; pb5h[a]=0;
        cl 2l [b]=0; rw4l [b]=0; pd6l [b]=0; pb5l [b]=0;
        cl 6h[na]=0; rw4h[na]=0; pd3h[na]=0; pb2h[na]=0;
        cl 6l [nb]=0; rw4l [nb]=0; pd3l [nb]=0; pb2l [nb]=0;
        ufl g[d]=0; ufl g[nd]=0; }}}
    }
}
/* Set n18 & n32 */
void stp23(){
    short a, na, b, nb, d, nd;
    for(a=0; a<7; a++){na=6-a;
        if((cl 4h[a]==0)&&(rw3h[a]==0)&&(pd2h[a]==0)&&(pb6h[a]==0)){
            if((cl 4h[na]==0)&&(rw5h[na]==0)&&(pd7h[na]==0)&&(pb1h[na]==0)){
                for(b=0; b<7; b++){nb=6-b; d=7*a+b; nd=CC-d;
                    if((ufl g[d]==0)&&(ufl g[nd]==0)){
                        if((cl 4l [b]==0)&&(rw3l [b]==0)&&(pd2l [b]==0)&&(pb6l [b]==0)){
                            if((cl 4l [nb]==0)&&(rw5l [nb]==0)&&(pd7l [nb]==0)&&(pb1l [nb]==0)){
                                nm[18]=d; nm[32]=nd; ufl g[d]=1; ufl g[nd]=1;
                                cl 4h[a]=1; rw3h[a]=1; pd2h[a]=1; pb6h[a]=1;
                                cl 4l [b]=1; rw3l [b]=1; pd2l [b]=1; pb6l [b]=1;
                                cl 4h[na]=1; rw5h[na]=1; pd7h[na]=1; pb1h[na]=1;
                                cl 4l [nb]=1; rw5l [nb]=1; pd7l [nb]=1; pb1l [nb]=1;
                                stp24();
                                cl 4h[a]=0; rw3h[a]=0; pd2h[a]=0; pb6h[a]=0;
                                cl 4l [b]=0; rw3l [b]=0; pd2l [b]=0; pb6l [b]=0;
                                cl 4h[na]=0; rw5h[na]=0; pd7h[na]=0; pb1h[na]=0;
                                cl 4l [nb]=0; rw5l [nb]=0; pd7l [nb]=0; pb1l [nb]=0;
                                ufl g[d]=0; ufl g[nd]=0; }}}
                            }}}
                    }
                }
            }
        }
    }
}
/* Set n24 & n26 */
void stp24(){
    short a, na, b, nb, d, nd;
    for(a=0; a<7; a++){na=6-a;
        if((cl 3h[a]==0)&&(rw4h[a]==0)&&(pd7h[a]==0)&&(pb6h[a]==0)){
            if((cl 5h[na]==0)&&(rw4h[na]==0)&&(pd2h[na]==0)&&(pb1h[na]==0)){
                for(b=0; b<7; b++){nb=6-b; d=7*a+b; nd=CC-d;
                    if((ufl g[d]==0)&&(ufl g[nd]==0)){
                        if((cl 3l [b]==0)&&(rw4l [b]==0)&&(pd7l [b]==0)&&(pb6l [b]==0)){
                            if((cl 5l [nb]==0)&&(rw4l [nb]==0)&&(pd2l [nb]==0)&&(pb1l [nb]==0)){
                                nm[24]=d; nm[26]=nd; ufl g[d]=1; ufl g[nd]=1;
                                cl 3h[a]=1; rw4h[a]=1; pd7h[a]=1; pb6h[a]=1;
                                cl 3l [b]=1; rw4l [b]=1; pd7l [b]=1; pb6l [b]=1;
                                cl 5h[na]=1; rw4h[na]=1; pd2h[na]=1; pb1h[na]=1;
                                cl 5l [nb]=1; rw4l [nb]=1; pd2l [nb]=1; pb1l [nb]=1;
                                printans();
                                cl 3h[a]=0; rw4h[a]=0; pd7h[a]=0; pb6h[a]=0;
                                cl 3l [b]=0; rw4l [b]=0; pd7l [b]=0; pb6l [b]=0;
                                cl 5h[na]=0; rw4h[na]=0; pd2h[na]=0; pb1h[na]=0;
                                cl 5l [nb]=0; rw4l [nb]=0; pd2l [nb]=0; pb1l [nb]=0;
                                ufl g[d]=0; ufl g[nd]=0; }}}
                            }}}
                    }
                }
            }
        }
    }
}
/**/
/* Print New Answer */
void printans(){
    short n;
    cnt++; cnt2++;
    if(cnt2==1){

```

```

    anm[cnt3*2][0]=cnt;
    for(n=1;n<50;n++){anm[cnt3*2][n]=nm[n]+1; anm[cnt3*2+1][n]=nm[n]; }
    cnt3++; if(cnt3==2){print2p(); anm[2][0]=0; cnt3=0;}
}
/**/
/* Print 2 Answers with their Decompositions */
void print2p(){
short l,l7,n;
printf(" /D7i H/      L/%21d#   /D7i H/      L/%21d#\n", anm[0][0], anm[2][0]);
for(l=0;l<7;l++){l7=l*7;
printf(" ");
for(n=1;n<8;n++){printf("%d", anm[1][l7+n]/7); }
printf(" ");
for(n=1;n<8;n++){printf("%d", anm[1][l7+n]%7); }
printf(" ");
for(n=1;n<8;n++){printf("%3d", anm[0][l7+n]); }
printf(" ");
for(n=1;n<8;n++){printf("%d", anm[3][l7+n]/7); }
printf(" ");
for(n=1;n<8;n++){printf("%d", anm[3][l7+n]%7); }
printf(" ");
for(n=1;n<8;n++){printf("%3d", anm[2][l7+n]); }
printf("\n");
}
}
/**/

```

### #5. Result list of My Recent Computation

The next list shows one of the recent results of my computation for our new type of Simultaneous magic squares 7x7: both self-complementary and pan-diagonal.

It was a big surprise to me that we could take only a few minutes to make all.

#### \*\* List of Standard Solutions of 'Complete Euler Type' of Simultaneous MS77: Both Self-complementary and Pan-diagonal (Part) \*\*

/D7i H/	L/	1#	/D7i H/	L/	7#
0654321	0531642	1 48 39 30 28 19 10	0654321	0364512	1 46 42 33 27 16 10
2106543	3164205	18 9 7 47 38 29 27	2106543	5120364	20 9 3 43 39 35 26
4321065	6420531	35 26 17 8 6 46 37	4321065	3645120	32 28 19 13 2 45 36
6543210	2053164	45 36 34 25 16 14 5	6543210	1203645	44 38 29 25 21 12 6
1065432	5316420	13 4 44 42 33 24 15	1065432	6451203	14 5 48 37 31 22 18
3210654	1642053	23 21 12 3 43 41 32	3210654	2036451	24 15 11 7 47 41 30
5432106	4205316	40 31 22 20 11 2 49	5432106	4512036	40 34 23 17 8 4 49
/D7i H/	L/	13#	/D7i H/	L/	19#
0652341	0531642	1 48 39 16 28 33 10	0652341	0364512	1 46 42 19 27 30 10
4106523	3164205	32 9 7 47 38 15 27	4106523	5120364	34 9 3 43 39 21 26
2341065	6420531	21 26 31 8 6 46 37	2341065	3645120	18 28 33 13 2 45 36
6523410	2053164	45 36 20 25 30 14 5	6523410	1203645	44 38 15 25 35 12 6
1065234	5316420	13 4 44 42 19 24 29	1065234	6451203	14 5 48 37 17 22 32
3410652	1642053	23 35 12 3 43 41 18	3410652	2036451	24 29 11 7 47 41 16
5234106	4205316	40 17 22 34 11 2 49	5234106	4512036	40 20 23 31 8 4 49
/D7i H/	L/	25#	/D7i H/	L/	31#
0654321	0432651	1 47 39 31 28 20 9	0654321	0365421	1 46 42 34 26 17 9
2106543	3265104	18 10 7 48 37 29 26	2106543	4210365	19 10 2 43 39 35 27
4321065	6510432	35 27 16 8 5 46 38	4321065	3654210	32 28 20 12 3 44 36
6543210	1043265	44 36 33 25 17 14 6	6543210	2103654	45 37 29 25 21 13 5
1065432	4326510	12 4 45 42 34 23 15	1065432	6542103	14 6 47 38 30 22 18
3210654	2651043	24 21 13 2 43 40 32	3210654	1036542	23 15 11 7 48 40 31
5432106	5104326	41 30 22 19 11 3 49	5432106	5421036	41 33 24 16 8 4 49

/D7i H/	L/		37#	/D7i H/	L/		43#
0652341	0432651	1 47 39 17 28 34 9		0652341	0365421	1 46 42 20 26 31 9	
4106523	3265104	32 10 7 48 37 15 26		4106523	4210365	33 10 2 43 39 21 27	
2341065	6510432	21 27 30 8 5 46 38		2341065	3654210	18 28 34 12 3 44 36	
6523410	1043265	44 36 19 25 31 14 6		6523410	2103654	45 37 15 25 35 13 5	
1065234	4326510	12 4 45 42 20 23 29		1065234	6542103	14 6 47 38 16 22 32	
3410652	2651043	24 35 13 2 43 40 18		3410652	1036542	23 29 11 7 48 40 17	
5234106	5104326	41 16 22 33 11 3 49		5234106	5421036	41 19 24 30 8 4 49	
/D7i H/	L/		49#	/D7i H/	L/		73#
0654321	0234651	1 45 39 33 28 20 9		0654321	0246531	1 45 40 35 27 18 9	
2106543	3465102	18 12 7 48 37 29 24		2106543	6531024	21 13 4 44 36 31 26	
4321065	6510234	35 27 16 8 3 46 40		4321065	1024653	30 22 17 12 7 48 39	
6543210	1023465	44 36 31 25 19 14 6		6543210	4653102	47 42 34 25 16 8 3	
1065432	2346510	10 4 47 42 34 23 15		1065432	3102465	11 2 43 38 33 28 20	
3210654	4651023	26 21 13 2 43 38 32		3210654	2465310	24 19 14 6 46 37 29	
5432106	5102346	41 30 22 17 11 5 49		5432106	5310246	41 32 23 15 10 5 49	
/D7i H/	L/		97#	/D7i H/	L/		121#
0432651	0654321	1 35 27 19 46 38 9		0432651	0645312	1 35 26 20 46 37 10	
3265104	2106543	24 16 43 42 13 5 32		3265104	1206453	23 17 43 42 12 6 32	
6510432	4321065	47 39 10 2 29 28 20		6510432	5312064	48 39 9 3 29 28 19	
1043265	6543210	14 6 33 25 17 44 36		1043265	6453120	14 5 34 25 16 45 36	
4326510	1065432	30 22 21 48 40 11 3		4326510	2064531	31 22 21 47 41 11 2	
2651043	3210654	18 45 37 8 7 34 26		2651043	3120645	18 44 38 8 7 33 27	
5104326	5432106	41 12 4 31 23 15 49		5104326	4531206	40 13 4 30 24 15 49	
/D7i H/	L/		145#	/D7i H/	L/		169#
0432651	0156432	1 30 27 21 47 39 10		0432651	0246531	1 31 26 21 48 39 9	
3265104	6432015	28 19 46 38 8 2 34		3265104	6531024	28 20 46 37 8 3 33	
6510432	2015643	45 36 9 6 35 26 18		6510432	1024653	44 36 10 5 35 27 18	
1043265	5643201	13 7 33 25 17 43 37		1043265	4653102	12 7 34 25 16 43 38	
4326510	3201564	32 24 15 44 41 14 5		4326510	3102465	32 23 15 45 40 14 6	
2651043	1564320	16 48 42 12 4 31 22		2651043	2465310	17 47 42 13 4 30 22	
5104326	4320156	40 11 3 29 23 20 49		5104326	5310246	41 11 2 29 24 19 49	
/D7i H/	L/		193#	/D7i H/	L/		217#
0625314	0531642	1 48 18 37 28 12 31		0625314	0432651	1 47 18 38 28 13 30	
1406253	3164205	11 30 7 47 17 36 27		1406253	3265104	11 31 7 48 16 36 26	
5314062	6420531	42 26 10 29 6 46 16		5314062	6510432	42 27 9 29 5 46 17	
6253140	2053164	45 15 41 25 9 35 5		6253140	1043265	44 15 40 25 10 35 6	
4062531	5316420	34 4 44 21 40 24 8		4062531	4326510	33 4 45 21 41 23 8	
3140625	1642053	23 14 33 3 43 20 39		3140625	2651043	24 14 34 2 43 19 39	
2531406	4205316	19 38 22 13 32 2 49		2531406	5104326	20 37 22 12 32 3 49	
/D7i H/	L/		241#	/D7i H/	L/		265#
0625314	0234651	1 45 18 40 28 13 30		0625314	0246531	1 45 19 42 27 11 30	
1406253	3465102	11 33 7 48 16 36 24		1406253	6531024	14 34 4 44 15 38 26	
5314062	6510234	42 27 9 29 3 46 19		5314062	1024653	37 22 10 33 7 48 18	
6253140	1023465	44 15 38 25 12 35 6		6253140	4653102	47 21 41 25 9 29 3	
4062531	2346510	31 4 47 21 41 23 8		4062531	3102465	32 2 43 17 40 28 13	
3140625	4651023	26 14 34 2 43 17 39		3140625	2465310	24 12 35 6 46 16 36	
2531406	5102346	20 37 22 10 32 5 49		2531406	5310246	20 39 23 8 31 5 49	
/D7i H/	L/		289#	/D7i H/	L/		313#
0362154	0654321	1 28 48 19 11 38 30		0362154	0432651	1 26 46 17 14 41 30	
1540362	2106543	10 37 29 7 27 47 18		1540362	3265104	11 38 35 6 23 43 19	
3621540	4321065	26 46 17 9 36 35 6		3621540	6510432	28 48 16 8 40 32 3	
5403621	6543210	42 34 5 25 45 16 8		5403621	1043265	37 29 5 25 45 21 13	
6215403	1065432	44 15 14 41 33 4 24		6215403	4326510	47 18 10 42 34 2 22	
4036215	3210654	32 3 23 43 21 13 40		4036215	2651043	31 7 27 44 15 12 39	
2154036	5432106	20 12 39 31 2 22 49		2154036	5104326	20 9 36 33 4 24 49	

/D7i H/	L/		337#	/D7i H/	L/		361#
0362154	0234651	1 24 46 19 14 41 30		0362154	0246531	1 24 47 21 13 39 30	
1540362	3465102	11 40 35 6 23 43 17		1540362	6531024	14 41 32 2 22 45 19	
3621540	6510234	28 48 16 8 38 32 5		3621540	1024653	23 43 17 12 42 34 4	
5403621	1023465	37 29 3 25 47 21 13		5403621	4653102	40 35 6 25 44 15 10	
6215403	2346510	45 18 12 42 34 2 22		6215403	3102465	46 16 8 38 33 7 27	
4036215	4651023	33 7 27 44 15 10 39		4036215	2465310	31 5 28 48 18 9 36	
2154036	5102346	20 9 36 31 4 26 49		2154036	5310246	20 11 37 29 3 26 49	
/D7i H/	L/		385#	/D7i H/	L/		409#
0654321	1630542	2 49 39 29 27 19 10		0654321	1432560	2 47 39 31 27 21 8	
2106543	3054216	18 8 6 47 38 30 28		2106543	3256014	18 10 6 49 36 30 26	
4321065	5421630	34 26 17 9 7 46 36		4321065	5601432	34 28 15 9 5 46 38	
6543210	2163054	45 37 35 25 15 13 5		6543210	0143256	43 37 33 25 17 13 7	
1065432	6305421	14 4 43 41 33 24 16		1065432	4325601	12 4 45 41 35 22 16	
3210654	0542163	22 20 12 3 44 42 32		3210654	2560143	24 20 14 1 44 40 32	
5432106	4216305	40 31 23 21 11 1 48		5432106	6014325	42 29 23 19 11 3 48	
/D7i H/	L/		433#	/D7i H/	L/		457#
0654321	1234560	2 45 39 33 27 21 8		0654321	1245630	2 45 40 34 28 18 8	
2106543	3456012	18 12 6 49 36 30 24		2106543	5630124	20 14 4 43 37 31 26	
4321065	5601234	34 28 15 9 3 46 40		4321065	0124563	29 23 17 12 6 49 39	
6543210	0123456	43 37 31 25 19 13 7		6543210	4563012	47 41 35 25 15 9 3	
1065432	2345601	10 4 47 41 35 22 16		1065432	3012456	11 1 44 38 33 27 21	
3210654	4560123	26 20 14 1 44 38 32		3210654	2456301	24 19 13 7 46 36 30	
5432106	6012345	42 29 23 17 11 5 48		5432106	6301245	42 32 22 16 10 5 48	
/D7i H/	L/		481#	/D7i H/	L/		505#
0432651	1564320	2 34 28 19 46 38 8		0432651	1546302	2 34 26 21 46 36 10	
3265104	2015643	24 15 44 41 14 5 32		3265104	0215463	22 17 44 41 12 7 32	
6510432	4320156	47 39 10 1 30 27 21		6510432	6302154	49 39 8 3 30 27 19	
1043265	5643201	13 7 33 25 17 43 37		1043265	5463021	13 5 35 25 15 45 37	
4326510	0156432	29 23 20 49 40 11 3		4326510	2154630	31 23 20 47 42 11 1	
2651043	3201564	18 45 36 9 6 35 26		2651043	3021546	18 43 38 9 6 33 28	
5104326	6432015	42 12 4 31 22 16 48		5104326	4630215	40 14 4 29 24 16 48	
/D7i H/	L/		529#	/D7i H/	L/		553#
0432651	1065432	2 29 28 20 47 39 10		0432651	1245630	2 31 26 20 49 39 8	
3265104	5432106	27 19 46 38 9 1 35		3265104	5630124	27 21 46 36 9 3 33	
6510432	2106543	45 37 8 7 34 26 18		6510432	0124563	43 37 10 5 34 28 18	
1043265	6543210	14 6 33 25 17 44 36		1043265	4563012	12 6 35 25 15 44 38	
4326510	3210654	32 24 16 43 42 13 5		4326510	3012456	32 22 16 45 40 13 7	
2651043	0654321	15 49 41 12 4 31 23		2651043	2456301	17 47 41 14 4 29 23	
5104326	4321065	40 11 3 30 22 21 48		5104326	6301245	42 11 1 30 24 19 48	
/D7i H/	L/		577#	/D7i H/	L/		601#
0625314	1630542	2 49 18 36 27 12 31		0625314	1432560	2 47 18 38 27 14 29	
1406253	3054216	11 29 6 47 17 37 28		1406253	3256014	11 31 6 49 15 37 26	
5314062	5421630	41 26 10 30 7 46 15		5314062	5601432	41 28 8 30 5 46 17	
6253140	2163054	45 16 42 25 8 34 5		6253140	0143256	43 16 40 25 10 34 7	
4062531	6305421	35 4 43 20 40 24 9		4062531	4325601	33 4 45 20 42 22 9	
3140625	0542163	22 13 33 3 44 21 39		3140625	2560143	24 13 35 1 44 19 39	
2531406	4216305	19 38 23 14 32 1 48		2531406	6014325	21 36 23 12 32 3 48	
/D7i H/	L/		625#	/D7i H/	L/		649#
0625314	1234560	2 45 18 40 27 14 29		0625314	1245630	2 45 19 41 28 11 29	
1406253	3456012	11 33 6 49 15 37 24		1406253	5630124	13 35 4 43 16 38 26	
5314062	5601234	41 28 8 30 3 46 19		5314062	0124563	36 23 10 33 6 49 18	
6253140	0123456	43 16 38 25 12 34 7		6253140	4563012	47 20 42 25 8 30 3	
4062531	2345601	31 4 47 20 42 22 9		4062531	3012456	32 1 44 17 40 27 14	
3140625	4560123	26 13 35 1 44 17 39		3140625	2456301	24 12 34 7 46 15 37	
2531406	6012345	21 36 23 10 32 5 48		2531406	6301245	21 39 22 9 31 5 48	

/D7i H/	L/		673#	/D7i H/	L/		697#
0362154	1564320	2 27 49 19 11 38 29		0362154	1432560	2 26 46 17 13 42 29	
1540362	2015643	10 36 30 6 28 47 18		1540362	3256014	11 38 34 7 22 44 19	
3621540	4320156	26 46 17 8 37 34 7		3621540	5601432	27 49 15 9 40 32 3	
5403621	5643201	41 35 5 25 45 15 9		5403621	0143256	36 30 5 25 45 20 14	
6215403	0156432	43 16 13 42 33 4 24		6215403	4325601	47 18 10 41 35 1 23	
4036215	3201564	32 3 22 44 20 14 40		4036215	2560143	31 6 28 43 16 12 39	
2154036	6432015	21 12 39 31 1 23 48		2154036	6014325	21 8 37 33 4 24 48	
/D7i H/	L/		721#	/D7i H/	L/		745#
0362154	1234560	2 24 46 19 13 42 29		0362154	1245630	2 24 47 20 14 39 29	
1540362	3456012	11 40 34 7 22 44 17		1540362	5630124	13 42 32 1 23 45 19	
3621540	5601234	27 49 15 9 38 32 5		3621540	0124563	22 44 17 12 41 35 4	
5403621	0123456	36 30 3 25 47 20 14		5403621	4563012	40 34 7 25 43 16 10	
6215403	2345601	45 18 12 41 35 1 23		6215403	3012456	46 15 9 38 33 6 28	
4036215	4560123	33 6 28 43 16 10 39		4036215	2456301	31 5 27 49 18 8 37	
2154036	6012345	21 8 37 31 4 26 48		2154036	6301245	21 11 36 30 3 26 48	
/D7i H/	L/		769#	/D7i H/	L/		1153#
0654321	2630451	3 49 39 29 26 20 9		0654321	4630251	5 49 39 29 24 20 9	
2106543	3045126	18 8 5 48 37 31 28		2106543	3025146	18 8 3 48 37 33 28	
4321065	4512630	33 27 16 10 7 46 36		4321065	2514630	31 27 16 12 7 46 36	
6543210	1263045	44 38 35 25 15 12 6		6543210	1463025	44 40 35 25 15 10 6	
1065432	6304512	14 4 43 40 34 23 17		1065432	6302514	14 4 43 38 34 23 19	
3210654	0451263	22 19 13 2 45 42 32		3210654	0251463	22 17 13 2 47 42 32	
5432106	5126304	41 30 24 21 11 1 47		5432106	5146302	41 30 26 21 11 1 45	
/D7i H/	L/		1537#	/D7i H/	L/		1921#
0654321	5630142	6 49 39 29 23 19 10		0654321	6531042	7 48 39 30 22 19 10	
2106543	3014256	18 8 2 47 38 34 28		2106543	3104265	18 9 1 47 38 35 27	
4321065	1425630	30 26 17 13 7 46 36		4321065	0426531	29 26 17 14 6 46 37	
6543210	2563014	45 41 35 25 15 9 5		6543210	2653104	45 42 34 25 16 8 5	
1065432	6301425	14 4 43 37 33 24 20		1065432	5310426	13 4 44 36 33 24 21	
3210654	0142563	22 16 12 3 48 42 32		3210654	1042653	23 15 12 3 49 41 32	
5432106	4256301	40 31 27 21 11 1 44		5432106	4265310	40 31 28 20 11 2 43	
/D7i H/	L/		2305#	/D7i H/	L/		2497#
1630542	0654321	8 49 27 5 39 31 16		1630542	1564320	9 48 28 5 39 31 15	
3054216	2106543	24 2 36 35 20 12 46		3054216	2015643	24 1 37 34 21 12 46	
5421630	4321065	40 32 17 9 43 28 6		5421630	4320156	40 32 17 8 44 27 7	
2163054	6543210	21 13 47 25 3 37 29		2163054	5643201	20 14 47 25 3 36 30	
6305421	1065432	44 22 7 41 33 18 10		6305421	0156432	43 23 6 42 33 18 10	
0542163	3210654	4 38 30 15 14 48 26		0542163	3201564	4 38 29 16 13 49 26	
4216305	5432106	34 19 11 45 23 1 42		4216305	6432015	35 19 11 45 22 2 41	
/D7i H/	L/		2689#	/D7i H/	L/		2881#
1630542	2465310	10 47 28 6 39 30 15		1630542	4265310	12 45 28 6 39 30 15	
3054216	1024653	23 1 38 33 21 13 46		3054216	1042653	23 1 40 31 21 13 46	
5421630	5310246	41 32 16 8 45 26 7		5421630	5310426	41 32 16 8 47 24 7	
2163054	4653102	19 14 48 25 2 36 31		2163054	2653104	17 14 48 25 2 36 33	
6305421	0246531	43 24 5 42 34 18 9		6305421	0426531	43 26 3 42 34 18 9	
0542163	3102465	4 37 29 17 12 49 27		0542163	3104265	4 37 29 19 10 49 27	
4216305	6531024	35 20 11 44 22 3 40		4216305	6531042	35 20 11 44 22 5 38	
/D7i H/	L/		3073#	/D7i H/	L/		3265#
1630542	5164320	13 44 28 5 39 31 15		1630542	6054321	14 43 27 5 39 31 16	
3054216	2051643	24 1 41 30 21 12 46		3054216	2160543	24 2 42 29 20 12 46	
5421630	4320516	40 32 17 8 48 23 7		5421630	4321605	40 32 17 9 49 22 6	
2163054	1643205	16 14 47 25 3 36 34		2163054	0543216	15 13 47 25 3 37 35	
6305421	0516432	43 27 2 42 33 18 10		6305421	1605432	44 28 1 41 33 18 10	
0542163	3205164	4 38 29 20 9 49 26		0542163	3216054	4 38 30 21 8 48 26	
4216305	6432051	35 19 11 45 22 6 37		4216305	5432160	34 19 11 45 23 7 36	

\* [Count = 3456] OK! \*

How happy I am to have known about the total counts of 3456 solutions!

But I was afraid for the first time if it might be too small, though it might look so many to you. When I tried to count up the simultaneous type of MS77 through before, I often gave it up, because it always took too long a time for me to go on. I remember I once stopped when I counted them more than 50000.

Why could we have such a small number of solutions as many as 3456 here?

Watch the next list of example solutions of general Simultaneous MS77 with /D7i and check if they are 'Complete Euler Type' or not, please.

**\*\* Examples of 'Non-Euler Type' of Simultaneous \*\***

**Magic Squares 7x7: Both Self-Complementary and Pan-Diagonal \*\***

														1/ /D7i																2/ /D7i	
1	48	47	32	27	16	4	0664320	0543513	1	48	47	31	32	12	4	0664410	0542343	21	9	5	16	44	43	37	2102665	6141101					
12	6	5	29	39	41	43	1004556	4540350	24	27	33	10	20	39	22	3341253	2542530	42	35	36	25	14	15	8	5453121	6603600					
36	35	22	10	20	33	19	5431242	0602544	28	11	30	40	17	23	26	3145233	6314214	13	7	6	34	45	41	29	1004654	5655250					
42	26	37	25	13	24	8	5353131	6413520	46	38	18	19	3	2	49	6522006	3234216														
31	17	30	40	28	15	14	4245321	2214606																							
7	9	11	21	45	44	38	0112665	6136212																							
46	34	23	18	3	2	49	6432006	3513216																							
														3/ /D7i																4/ /D7i	
1	48	47	31	23	12	13	0664311	0542145	1	48	47	31	21	18	9	0664221	0542631	13	6	10	22	39	42	43	1013556	5520360					
15	5	8	29	40	46	32	2014564	0400433	35	27	26	4	36	30	17	4330542	6543012	45	34	38	25	12	16	5	6453120	2523414					
34	28	17	11	20	41	24	4321253	5623552	33	20	14	46	24	23	15	4216332	4563210	7	8	11	28	40	44	37	0113565	6036411					
44	43	36	25	14	7	6	6653100	1003665	41	32	29	19	3	2	49	5442006	5304216														
26	9	30	39	33	22	16	3145432	4113401																							
18	4	10	21	42	45	35	2012564	3326626																							
37	38	27	19	3	2	49	5532006	1254216																							
														5/ /D7i																6/ /D7i	
1	48	47	31	18	23	7	0664230	0542316	1	48	47	29	34	9	7	0664410	0540516	23	5	6	14	46	42	39	3001655	1456363					
14	6	11	24	33	45	42	1013465	6532426	26	28	37	12	18	35	19	3351242	4614364	40	30	33	25	17	20	10	5443221	4143252					
34	30	22	4	41	29	15	4430542	5103500	31	15	32	38	13	22	24	4245133	2032502	11	8	4	36	44	45	27	1105663	3030125					
40	38	37	25	13	12	10	5553111	4213542	43	41	16	21	3	2	49	6522006	0516216														
35	21	9	46	28	20	16	4216322	6613651																							
8	5	17	26	39	44	36	1023565	0424310																							
43	27	32	19	3	2	49	6342006	0534216																							
														7/ /D7i																8/ /D7i	
1	48	47	29	33	12	5	0664410	0540444	1	48	47	29	33	12	5	0664410	0540444	15	7	9	22	46	40	36	2013655	0610340					
28	6	4	18	43	39	37	3002655	6533031	32	31	27	6	16	39	24	4430253	3255132	42	30	37	25	13	20	8	5453121	6113550					
20	23	34	14	19	41	24	2341253	5156452	26	11	34	44	23	19	18	3146322	4351143	14	10	4	28	41	43	35	1103564	6236506					
42	40	35	25	15	10	8	5543211	6463020	45	38	17	21	3	2	49	6522006	2226216														
26	9	31	36	16	27	30	3145234	4120151																							
13	11	7	32	46	44	22	1104663	5363310																							
45	38	17	21	3	2	49	6522006	2226216																							
														9/ /D7i																10/ /D7i	
1	48	47	29	30	16	4	0664420	0540113	1	48	47	29	26	15	9	0664321	0540401	18	5	11	12	43	44	42	2011665	3434016					
17	5	9	31	37	36	40	2014555	2412104	31	28	37	4	34	27	14	4350431	2613556	40	30	33	25	17	20	10	5443221	4143252					
32	28	15	11	23	42	24	4321353	3603162	36	23	16	46	13	22	19	5326132	0113504	8	6	7	38	39	45	32	1005564	0562323					
43	38	44	25	6	12	7	6563010	0213546	41	35	24	21	3	2	49	5432006	5626216														
26	8	27	39	35	22	18	3135432	4053603																							
10	14	13	19	41	45	33	1112564	2654524																							
46	34	20	21	3	2	49	6422006	3556216																							

In the case of order 7, there exist so many "Non-Euler Type" of solutions, say, far

more than Complete Euler Type. Why! How embarrassing it is!

But we must accept the fact we have got all the solutions of Complete Euler Type of Simultaneous MS77: both self-complementary and pan-diagonal as many as 3456.

In the case of higher order than six, "Greco-Latinian Method" seems to be capable of composing only the rare and precious jewels among many kinds of magic squares.

## #6. How about the Case of Pan-diagonal Magic Squares 7x7?

Can we compose 'Complete Euler Type' of Pan-diagonal MS77 in the same way?

Making any pan-diagonal MS77 is one of the most difficult problems we have ever had. It really takes too long a time for us to count them up.

We have not yet known how many solutions of any type of order 7 do really exist.

I myself used to try counting them up many times, but I gave it up every time, for it took too much energy to go on through. My computer seemed almost sleeping down.

I made up my mind to try counting them up again, since we have just got a new art of programming. Let's have a real challenge to make the pan-diagonal magic type.

Let's make and count only the fundamental type of solutions with  $n_1=1$  then, for we can guess we may have too many solutions. Fortunately we know how to reconstruct all the standard solutions of pan-diagonal MS77 from those fundamental ones.

## #7. Part of My Program List

```
/** 'Complete Euler Squares' for Pan-diagonal MS77: **/  
/** 'CEuler7PDF.c' Built by Kanji Setsuda on **/  
/** Mar. 19, 2003; Revised on Feb. 17, 2007 **/  
/** Working with MacOSX and Xcode 2.2 **/  
/**/  
#include <stdio.h>  
/**/  
/* Global Variables */  
long int cnt, cnt2, cnt3;  
short cnt3;  
short S6N, S62N;  
short nm[50], uf1g[50];  
short anm[5][50];  
short cl1h[8], cl1l[8], rw1h[8], rw1l[8], pd1h[8], pd1l[8], pb1h[8], pb1l[8];  
short cl2h[8], cl2l[8], rw2h[8], rw2l[8], pd2h[8], pd2l[8], pb2h[8], pb2l[8];  
short cl3h[8], cl3l[8], rw3h[8], rw3l[8], pd3h[8], pd3l[8], pb3h[8], pb3l[8];  
short cl4h[8], cl4l[8], rw4h[8], rw4l[8], pd4h[8], pd4l[8], pb4h[8], pb4l[8];  
short cl5h[8], cl5l[8], rw5h[8], rw5l[8], pd5h[8], pd5l[8], pb5h[8], pb5l[8];  
short cl6h[8], cl6l[8], rw6h[8], rw6l[8], pd6h[8], pd6l[8], pb6h[8], pb6l[8];  
short cl7h[8], cl7l[8], rw7h[8], rw7l[8], pd7h[8], pd7l[8], pb7h[8], pb7l[8];  
/**/  
/* Sub-Routines */  
void stp01(void), stp02(void), stp03(void), stp04(void);  
void stp05(void), stp06(void), stp07(void), stp08(void);  
void stp09(void), stp10(void), stp11(void), stp12(void);  
void stp13(void), stp14(void), stp15(void), stp16(void);  
void stp17(void), stp18(void), stp19(void), stp20(void);  
void stp21(void), stp22(void), stp23(void), stp24(void);  
void stp25(void), stp26(void), stp27(void), stp28(void);  
void stp29(void), stp30(void), stp31(void), stp32(void);  
void stp33(void), stp34(void), stp35(void), stp36(void);  
void stp37(void), stp38(void), stp39(void), stp40(void);  
void stp41(void), stp42(void), stp43(void), stp44(void);  
void stp45(void), stp46(void), stp47(void), stp48(void);  
void stp49(void), stp50(void), stp51(void), stp52(void);  
void printans(void), print2p(void);  
/**/
```

```

/* Main Program */
/* Reset All Memories and flags */
int main(){
short m, n;
printf("\n** 'Complete Euler Squares' for Pan-Diagonal Magic Type 7x7: **\n");
for(n=0; n<50; n++){nm[n]=0; uflg[n]=0;}
for(m=0; m<5; m++){for(n=0; n<50; n++){anm[m][n]=0;}}
for(n=0; n<7; n++){
cl 1h[n]=0; rw1h[n]=0; pd1h[n]=0; pb1h[n]=0;
cl 1l [n]=0; rw1l [n]=0; pd1l [n]=0; pb1l [n]=0;
cl 2h[n]=0; rw2h[n]=0; pd2h[n]=0; pb2h[n]=0;
cl 2l [n]=0; rw2l [n]=0; pd2l [n]=0; pb2l [n]=0;
cl 3h[n]=0; rw3h[n]=0; pd3h[n]=0; pb3h[n]=0;
cl 3l [n]=0; rw3l [n]=0; pd3l [n]=0; pb3l [n]=0;
cl 4h[n]=0; rw4h[n]=0; pd4h[n]=0; pb4h[n]=0;
cl 4l [n]=0; rw4l [n]=0; pd4l [n]=0; pb4l [n]=0;
cl 5h[n]=0; rw5h[n]=0; pd5h[n]=0; pb5h[n]=0;
cl 5l [n]=0; rw5l [n]=0; pd5l [n]=0; pb5l [n]=0;
cl 6h[n]=0; rw6h[n]=0; pd6h[n]=0; pb6h[n]=0;
cl 6l [n]=0; rw6l [n]=0; pd6l [n]=0; pb6l [n]=0;
cl 7h[n]=0; rw7h[n]=0; pd7h[n]=0; pb7h[n]=0;
cl 7l [n]=0; rw7l [n]=0; pd7l [n]=0; pb7l [n]=0;
}
cnt=0; cnt3=0;
stp01();
if(cnt3>0){print2p();}
printf("** [Count = %d] OK! *\n", cnt);
return 0;
}
/**/
/* Begin the Calculation */
/* Set n1=0 */
void stp01(){
short a, b, d;
for(a=0; a<1; a++){
if((cl 1h[a]==0)&&(rw1h[a]==0)&&(pd1h[a]==0)&&(pb1h[a]==0)){
for(b=0; b<1; b++){d=7*a+b;
if(uflg[d]==0){
if((cl 1l [b]==0)&&(rw1l [b]==0)&&(pd1l [b]==0)&&(pb1l [b]==0)){
nm[1]=d; uflg[d]=1;
cl 1h[a]=1; rw1h[a]=1; pd1h[a]=1; pb1h[a]=1;
cl 1l [b]=1; rw1l [b]=1; pd1l [b]=1; pb1l [b]=1;
stp02();
cl 1h[a]=0; rw1h[a]=0; pd1h[a]=0; pb1h[a]=0;
cl 1l [b]=0; rw1l [b]=0; pd1l [b]=0; pb1l [b]=0;
uflg[d]=0;}}
}}
}
}
/* Set n7 & n1<n7 */
void stp02(){
short a, b, d;
for(a=0; a<7; a++){
if((cl 7h[a]==0)&&(rw7h[a]==0)&&(pd7h[a]==0)&&(pb7h[a]==0)){
for(b=0; b<7; b++){d=7*a+b;
if((nm[1]<d)&&(uflg[d]==0)){
if((cl 7l [b]==0)&&(rw7l [b]==0)&&(pd7l [b]==0)&&(pb7l [b]==0)){
nm[7]=d; uflg[d]=1; cnt2=0;
cl 7h[a]=1; rw7h[a]=1; pd7h[a]=1; pb7h[a]=1;
cl 7l [b]=1; rw7l [b]=1; pd7l [b]=1; pb7l [b]=1;
stp03();
cl 7h[a]=0; rw7h[a]=0; pd7h[a]=0; pb7h[a]=0;
}
}
}
}
}
}
}

```

```

        cl 7l [b]=0; rw1l [b]=0; pd7l [b]=0; pb7l [b]=0;
        ufl g[d]=0; }}
    }}
}
}
/* Set n2 & n2>n7 */
void stp03(){
short a, b, d;
for(a=6; a>=0; a--){
if((cl 2h[a]==0)&&(rw1h[a]==0)&&(pd2h[a]==0)&&(pb2h[a]==0)){
for(b=6; b>=0; b--){d=7*a+b;
if((d>nm[7])&&(ufl g[d]==0)){
if((cl 2l [b]==0)&&(rw1l [b]==0)&&(pd2l [b]==0)&&(pb2l [b]==0)){
nm[2]=d; ufl g[d]=1;
cl 2h[a]=1; rw1h[a]=1; pd2h[a]=1; pb2h[a]=1;
cl 2l [b]=1; rw1l [b]=1; pd2l [b]=1; pb2l [b]=1;
stp04();
cl 2h[a]=0; rw1h[a]=0; pd2h[a]=0; pb2h[a]=0;
cl 2l [b]=0; rw1l [b]=0; pd2l [b]=0; pb2l [b]=0;
ufl g[d]=0; }}
}}
}
}
/* Set n3 */
void stp04(){
short a, b, d;
for(a=6; a>=0; a--){
if((cl 3h[a]==0)&&(rw1h[a]==0)&&(pd3h[a]==0)&&(pb3h[a]==0)){
for(b=6; b>=0; b--){d=7*a+b;
if(ufl g[d]==0){
if((cl 3l [b]==0)&&(rw1l [b]==0)&&(pd3l [b]==0)&&(pb3l [b]==0)){
nm[3]=d; ufl g[d]=1;
cl 3h[a]=1; rw1h[a]=1; pd3h[a]=1; pb3h[a]=1;
cl 3l [b]=1; rw1l [b]=1; pd3l [b]=1; pb3l [b]=1;
stp05();
cl 3h[a]=0; rw1h[a]=0; pd3h[a]=0; pb3h[a]=0;
cl 3l [b]=0; rw1l [b]=0; pd3l [b]=0; pb3l [b]=0;
ufl g[d]=0; }}
}}
}
}
/* Set n6 */
void stp05(){
short a, b, d;
for(a=0; a<7; a++){
if((cl 6h[a]==0)&&(rw1h[a]==0)&&(pd6h[a]==0)&&(pb6h[a]==0)){
for(b=0; b<7; b++){d=7*a+b;
if(ufl g[d]==0){
if((cl 6l [b]==0)&&(rw1l [b]==0)&&(pd6l [b]==0)&&(pb6l [b]==0)){
nm[6]=d; ufl g[d]=1;
cl 6h[a]=1; rw1h[a]=1; pd6h[a]=1; pb6h[a]=1;
cl 6l [b]=1; rw1l [b]=1; pd6l [b]=1; pb6l [b]=1;
stp06();
cl 6h[a]=0; rw1h[a]=0; pd6h[a]=0; pb6h[a]=0;
cl 6l [b]=0; rw1l [b]=0; pd6l [b]=0; pb6l [b]=0;
ufl g[d]=0; }}
}}
}
}
/* Set n5 */
void stp06(){
short a, b, d;
for(a=0; a<7; a++){
if((cl 5h[a]==0)&&(rw1h[a]==0)&&(pd5h[a]==0)&&(pb5h[a]==0)){

```

```

for(b=0; b<7; b++){d=7*a+b;
  if(uflg[d]==0){
    if((cl5l[b]==0)&&(rw1l[b]==0)&&(pd5l[b]==0)&&(pb5l[b]==0)){
      nm[5]=d; uflg[d]=1;
      cl5h[a]=1; rw1h[a]=1; pd5h[a]=1; pb5h[a]=1;
      cl5l[b]=1; rw1l[b]=1; pd5l[b]=1; pb5l[b]=1;
      stp07();
      cl5h[a]=0; rw1h[a]=0; pd5h[a]=0; pb5h[a]=0;
      cl5l[b]=0; rw1l[b]=0; pd5l[b]=0; pb5l[b]=0;
      uflg[d]=0; }}
  }}
}
}
/* Set n4 */
void stp07(){
  short a, b, d;
  for(a=6; a>=0; a--){
    if((cl4h[a]==0)&&(rw1h[a]==0)&&(pd4h[a]==0)&&(pb4h[a]==0)){
      for(b=6; b>=0; b--){d=7*a+b;
        if((cl4l[b]==0)&&(rw1l[b]==0)&&(pd4l[b]==0)&&(pb4l[b]==0)){
          if(uflg[d]==0){
            nm[4]=d; uflg[d]=1;
            cl4h[a]=1; rw1h[a]=1; pd4h[a]=1; pb4h[a]=1;
            cl4l[b]=1; rw1l[b]=1; pd4l[b]=1; pb4l[b]=1;
            stp08();
            cl4h[a]=0; rw1h[a]=0; pd4h[a]=0; pb4h[a]=0;
            cl4l[b]=0; rw1l[b]=0; pd4l[b]=0; pb4l[b]=0;
            uflg[d]=0; }}
          }}
        }
      }
}
/* Set n43 & n1<n43 */
void stp08(){
  short a, b, d;
  for(a=0; a<7; a++){
    if((cl1h[a]==0)&&(rw7h[a]==0)&&(pd2h[a]==0)&&(pb7h[a]==0)){
      for(b=0; b<7; b++){d=7*a+b;
        if((nm[1]<d)&&(uflg[d]==0)){
          if((cl1l[b]==0)&&(rw7l[b]==0)&&(pd2l[b]==0)&&(pb7l[b]==0)){
            nm[43]=d; uflg[d]=1;
            cl1h[a]=1; rw7h[a]=1; pd2h[a]=1; pb7h[a]=1;
            cl1l[b]=1; rw7l[b]=1; pd2l[b]=1; pb7l[b]=1;
            stp09();
            cl1h[a]=0; rw7h[a]=0; pd2h[a]=0; pb7h[a]=0;
            cl1l[b]=0; rw7l[b]=0; pd2l[b]=0; pb7l[b]=0;
            uflg[d]=0; }}
          }}
        }
      }
}
}
/* Set n8 & (n2>n8)&(n8>n43) */
void stp09(){
  short a, b, d;
  for(a=6; a>=0; a--){
    if((cl1h[a]==0)&&(rw2h[a]==0)&&(pd7h[a]==0)&&(pb2h[a]==0)){
      for(b=6; b>=0; b--){d=7*a+b;
        if((nm[2]>d)&&(d>nm[43])&&(uflg[d]==0)){
          if((cl1l[b]==0)&&(rw2l[b]==0)&&(pd7l[b]==0)&&(pb2l[b]==0)){
            nm[8]=d; uflg[d]=1;
            cl1h[a]=1; rw2h[a]=1; pd7h[a]=1; pb2h[a]=1;
            cl1l[b]=1; rw2l[b]=1; pd7l[b]=1; pb2l[b]=1;
            stp10();
            cl1h[a]=0; rw2h[a]=0; pd7h[a]=0; pb2h[a]=0;
            cl1l[b]=0; rw2l[b]=0; pd7l[b]=0; pb2l[b]=0;
            uflg[d]=0; }}
          }}
        }
      }
}
}

```

```

    }}
}
}
/* Set n15 */
void stp10(){
short a, b, d;
for(a=0; a<7; a++){
if((cl1h[a]==0)&&(rw3h[a]==0)&&(pd6h[a]==0)&&(pb3h[a]==0)){
for(b=0; b<7; b++){d=7*a+b;
if(uflg[d]==0){
if((cl1l[b]==0)&&(rw3l[b]==0)&&(pd6l[b]==0)&&(pb3l[b]==0)){
nm[15]=d; uflg[d]=1;
cl1h[a]=1; rw3h[a]=1; pd6h[a]=1; pb3h[a]=1;
cl1l[b]=1; rw3l[b]=1; pd6l[b]=1; pb3l[b]=1;
stp11();
cl1h[a]=0; rw3h[a]=0; pd6h[a]=0; pb3h[a]=0;
cl1l[b]=0; rw3l[b]=0; pd6l[b]=0; pb3l[b]=0;
uflg[d]=0;}}
}}
}
}
/* Set n36 */
void stp11(){
short a, b, d;
for(a=6; a>=0; a--){
if((cl1h[a]==0)&&(rw6h[a]==0)&&(pd3h[a]==0)&&(pb6h[a]==0)){
for(b=6; b>=0; b--){d=7*a+b;
if(uflg[d]==0){
if((cl1l[b]==0)&&(rw6l[b]==0)&&(pd3l[b]==0)&&(pb6l[b]==0)){
nm[36]=d; uflg[d]=1;
cl1h[a]=1; rw6h[a]=1; pd3h[a]=1; pb6h[a]=1;
cl1l[b]=1; rw6l[b]=1; pd3l[b]=1; pb6l[b]=1;
stp12();
cl1h[a]=0; rw6h[a]=0; pd3h[a]=0; pb6h[a]=0;
cl1l[b]=0; rw6l[b]=0; pd3l[b]=0; pb6l[b]=0;
uflg[d]=0;}}
}}
}
}
/* Set n29 */
void stp12(){
short a, b, d;
for(a=6; a>=0; a--){
if((cl1h[a]==0)&&(rw5h[a]==0)&&(pd4h[a]==0)&&(pb5h[a]==0)){
for(b=6; b>=0; b--){d=7*a+b;
if(uflg[d]==0){
if((cl1l[b]==0)&&(rw5l[b]==0)&&(pd4l[b]==0)&&(pb5l[b]==0)){
nm[29]=d; uflg[d]=1;
cl1h[a]=1; rw5h[a]=1; pd4h[a]=1; pb5h[a]=1;
cl1l[b]=1; rw5l[b]=1; pd4l[b]=1; pb5l[b]=1;
stp13();
cl1h[a]=0; rw5h[a]=0; pd4h[a]=0; pb5h[a]=0;
cl1l[b]=0; rw5l[b]=0; pd4l[b]=0; pb5l[b]=0;
uflg[d]=0;}}
}}
}
}
}
/* Set n22 */
void stp13(){
short a, b, d;
for(a=0; a<7; a++){
if((cl1h[a]==0)&&(rw4h[a]==0)&&(pd5h[a]==0)&&(pb4h[a]==0)){
for(b=0; b<7; b++){d=7*a+b;
if((cl1l[b]==0)&&(rw4l[b]==0)&&(pd5l[b]==0)&&(pb4l[b]==0)){

```

```

    if(uflg[d]==0){
        nm[22]=d; uflg[d]=1;
        cl1h[a]=1; rw4h[a]=1; pd5h[a]=1; pb4h[a]=1;
        cl1l[b]=1; rw4l[b]=1; pd5l[b]=1; pb4l[b]=1;
        stp14();
        cl1h[a]=0; rw4h[a]=0; pd5h[a]=0; pb4h[a]=0;
        cl1l[b]=0; rw4l[b]=0; pd5l[b]=0; pb4l[b]=0;
        uflg[d]=0;}}
    }}
}
}
/* Set n49 & n1<n49 */
void stp14(){
    short a, b, d;
    for(a=6; a>=0; a--){
        if((cl7h[a]==0)&&(rw7h[a]==0)&&(pd1h[a]==0)&&(pb6h[a]==0)){
            for(b=6; b>=0; b--){d=7*a+b;
                if((nm[1]<d)&&(uflg[d]==0)){
                    if((cl7l[b]==0)&&(rw7l[b]==0)&&(pd1l[b]==0)&&(pb6l[b]==0)){
                        nm[49]=d; uflg[d]=1;
                        cl7h[a]=1; rw7h[a]=1; pd1h[a]=1; pb6h[a]=1;
                        cl7l[b]=1; rw7l[b]=1; pd1l[b]=1; pb6l[b]=1;
                        stp15();
                        cl7h[a]=0; rw7h[a]=0; pd1h[a]=0; pb6h[a]=0;
                        cl7l[b]=0; rw7l[b]=0; pd1l[b]=0; pb6l[b]=0;
                        uflg[d]=0;}}
                    }}
            }
        }
}
/* Set n14 */
void stp15(){
    short a, b, d;
    for(a=6; a>=0; a--){
        if((cl7h[a]==0)&&(rw2h[a]==0)&&(pd6h[a]==0)&&(pb1h[a]==0)){
            for(b=6; b>=0; b--){d=7*a+b;
                if(uflg[d]==0){
                    if((cl7l[b]==0)&&(rw2l[b]==0)&&(pd6l[b]==0)&&(pb1l[b]==0)){
                        nm[14]=d; uflg[d]=1;
                        cl7h[a]=1; rw2h[a]=1; pd6h[a]=1; pb1h[a]=1;
                        cl7l[b]=1; rw2l[b]=1; pd6l[b]=1; pb1l[b]=1;
                        stp16();
                        cl7h[a]=0; rw2h[a]=0; pd6h[a]=0; pb1h[a]=0;
                        cl7l[b]=0; rw2l[b]=0; pd6l[b]=0; pb1l[b]=0;
                        uflg[d]=0;}}
                    }}
            }
        }
}
/* Set n21 */
void stp16(){
    short a, b, d;
    for(a=0; a<7; a++){
        if((cl7h[a]==0)&&(rw3h[a]==0)&&(pd5h[a]==0)&&(pb2h[a]==0)){
            for(b=0; b<7; b++){d=7*a+b;
                if(uflg[d]==0){
                    if((cl7l[b]==0)&&(rw3l[b]==0)&&(pd5l[b]==0)&&(pb2l[b]==0)){
                        nm[21]=d; uflg[d]=1;
                        cl7h[a]=1; rw3h[a]=1; pd5h[a]=1; pb2h[a]=1;
                        cl7l[b]=1; rw3l[b]=1; pd5l[b]=1; pb2l[b]=1;
                        stp17();
                        cl7h[a]=0; rw3h[a]=0; pd5h[a]=0; pb2h[a]=0;
                        cl7l[b]=0; rw3l[b]=0; pd5l[b]=0; pb2l[b]=0;
                        uflg[d]=0;}}
                    }}
            }
        }
}
}

```

```

}
/* Set n42 */
void stp17(){
short a, b, d;
for(a=0; a<7; a++){
if((cl7h[a]==0)&&(rw6h[a]==0)&&(pd2h[a]==0)&&(pb5h[a]==0)){
for(b=0; b<7; b++){d=7*a+b;
if(uflg[d]==0){
if((cl7l[b]==0)&&(rw6l[b]==0)&&(pd2l[b]==0)&&(pb5l[b]==0)){
nm[42]=d; uflg[d]=1;
cl7h[a]=1; rw6h[a]=1; pd2h[a]=1; pb5h[a]=1;
cl7l[b]=1; rw6l[b]=1; pd2l[b]=1; pb5l[b]=1;
stp18();
cl7h[a]=0; rw6h[a]=0; pd2h[a]=0; pb5h[a]=0;
cl7l[b]=0; rw6l[b]=0; pd2l[b]=0; pb5l[b]=0;
uflg[d]=0; }}
}}
}
}
/* Set n35 */
void stp18(){
short a, b, d;
for(a=0; a<7; a++){
if((cl7h[a]==0)&&(rw5h[a]==0)&&(pd3h[a]==0)&&(pb4h[a]==0)){
for(b=0; b<7; b++){d=7*a+b;
if(uflg[d]==0){
if((cl7l[b]==0)&&(rw5l[b]==0)&&(pd3l[b]==0)&&(pb4l[b]==0)){
nm[35]=d; uflg[d]=1;
cl7h[a]=1; rw5h[a]=1; pd3h[a]=1; pb4h[a]=1;
cl7l[b]=1; rw5l[b]=1; pd3l[b]=1; pb4l[b]=1;
stp19();
cl7h[a]=0; rw5h[a]=0; pd3h[a]=0; pb4h[a]=0;
cl7l[b]=0; rw5l[b]=0; pd3l[b]=0; pb4l[b]=0;
uflg[d]=0; }}
}}
}
}
/* Set n28 */
void stp19(){
short a, b, d;
for(a=0; a<7; a++){
if((cl7h[a]==0)&&(rw4h[a]==0)&&(pd4h[a]==0)&&(pb3h[a]==0)){
for(b=0; b<7; b++){d=7*a+b;
if((cl7l[b]==0)&&(rw4l[b]==0)&&(pd4l[b]==0)&&(pb3l[b]==0)){
if(uflg[d]==0){
nm[28]=d; uflg[d]=1;
cl7h[a]=1; rw4h[a]=1; pd4h[a]=1; pb3h[a]=1;
cl7l[b]=1; rw4l[b]=1; pd4l[b]=1; pb3l[b]=1;
stp20();
cl7h[a]=0; rw4h[a]=0; pd4h[a]=0; pb3h[a]=0;
cl7l[b]=0; rw4l[b]=0; pd4l[b]=0; pb3l[b]=0;
uflg[d]=0; }}
}}
}
}
}
/* Set n44 */
void stp20(){
short a, b, d;
for(a=0; a<7; a++){
if((cl2h[a]==0)&&(rw7h[a]==0)&&(pd3h[a]==0)&&(pb1h[a]==0)){
for(b=0; b<7; b++){d=7*a+b;
if(uflg[d]==0){
if((cl2l[b]==0)&&(rw7l[b]==0)&&(pd3l[b]==0)&&(pb1l[b]==0)){
nm[44]=d; uflg[d]=1;

```

```

        cl 2h[a]=1; rw7h[a]=1; pd3h[a]=1; pb1h[a]=1;
        cl 2l [b]=1; rw7l [b]=1; pd3l [b]=1; pb1l [b]=1;
        stp21();
        cl 2h[a]=0; rw7h[a]=0; pd3h[a]=0; pb1h[a]=0;
        cl 2l [b]=0; rw7l [b]=0; pd3l [b]=0; pb1l [b]=0;
        ufl g[d]=0; }}
    }}
}
}
/* Set n45 */
void stp21(){
    short a, b, d;
    for(a=0; a<7; a++){
        if((cl 3h[a]==0)&&(rw7h[a]==0)&&(pd4h[a]==0)&&(pb2h[a]==0)){
            for(b=0; b<7; b++){d=7*a+b;
                if(ufl g[d]==0){
                    if((cl 3l [b]==0)&&(rw7l [b]==0)&&(pd4l [b]==0)&&(pb2l [b]==0)){
                        nm[45]=d; ufl g[d]=1;
                        cl 3h[a]=1; rw7h[a]=1; pd4h[a]=1; pb2h[a]=1;
                        cl 3l [b]=1; rw7l [b]=1; pd4l [b]=1; pb2l [b]=1;
                        stp22();
                        cl 3h[a]=0; rw7h[a]=0; pd4h[a]=0; pb2h[a]=0;
                        cl 3l [b]=0; rw7l [b]=0; pd4l [b]=0; pb2l [b]=0;
                        ufl g[d]=0; }}
                    }}
            }
        }
}
/* Set n48 */
void stp22(){
    short a, b, d;
    for(a=0; a<7; a++){
        if((cl 6h[a]==0)&&(rw7h[a]==0)&&(pd7h[a]==0)&&(pb5h[a]==0)){
            for(b=0; b<7; b++){d=7*a+b;
                if(ufl g[d]==0){
                    if((cl 6l [b]==0)&&(rw7l [b]==0)&&(pd7l [b]==0)&&(pb5l [b]==0)){
                        nm[48]=d; ufl g[d]=1;
                        cl 6h[a]=1; rw7h[a]=1; pd7h[a]=1; pb5h[a]=1;
                        cl 6l [b]=1; rw7l [b]=1; pd7l [b]=1; pb5l [b]=1;
                        stp23();
                        cl 6h[a]=0; rw7h[a]=0; pd7h[a]=0; pb5h[a]=0;
                        cl 6l [b]=0; rw7l [b]=0; pd7l [b]=0; pb5l [b]=0;
                        ufl g[d]=0; }}
                    }}
            }
        }
}
}
}
/* Set n47 */
void stp23(){
    short a, b, d;
    for(a=0; a<7; a++){
        if((cl 5h[a]==0)&&(rw7h[a]==0)&&(pd6h[a]==0)&&(pb4h[a]==0)){
            for(b=0; b<7; b++){d=7*a+b;
                if((cl 5l [b]==0)&&(rw7l [b]==0)&&(pd6l [b]==0)&&(pb4l [b]==0)){
                    if(ufl g[d]==0){
                        nm[47]=d; ufl g[d]=1;
                        cl 5h[a]=1; rw7h[a]=1; pd6h[a]=1; pb4h[a]=1;
                        cl 5l [b]=1; rw7l [b]=1; pd6l [b]=1; pb4l [b]=1;
                        stp24();
                        cl 5h[a]=0; rw7h[a]=0; pd6h[a]=0; pb4h[a]=0;
                        cl 5l [b]=0; rw7l [b]=0; pd6l [b]=0; pb4l [b]=0;
                        ufl g[d]=0; }}
                    }}
            }
        }
}
}
}
/* Set n46 */

```

```

void stp24(){
short a, b, d;
for(a=0; a<7; a++){
if((cl 4h[a]==0)&&(rw7h[a]==0)&&(pd5h[a]==0)&&(pb3h[a]==0)){
for(b=0; b<7; b++){d=7*a+b;
if((cl 4l [b]==0)&&(rw7l [b]==0)&&(pd5l [b]==0)&&(pb3l [b]==0)){
if(uflg[d]==0){
nm[46]=d; uflg[d]=1;
cl 4h[a]=1; rw7h[a]=1; pd5h[a]=1; pb3h[a]=1;
cl 4l [b]=1; rw7l [b]=1; pd5l [b]=1; pb3l [b]=1;
stp25();
cl 4h[a]=0; rw7h[a]=0; pd5h[a]=0; pb3h[a]=0;
cl 4l [b]=0; rw7l [b]=0; pd5l [b]=0; pb3l [b]=0;
uflg[d]=0; }}
}}}
}
}
/**/
/* ... (Ski p)... */
/**/
/* Set n24 */
void stp40(){
short a, b, d;
for(a=0; a<7; a++){
if((cl 3h[a]==0)&&(rw4h[a]==0)&&(pd7h[a]==0)&&(pb6h[a]==0)){
for(b=0; b<7; b++){d=7*a+b;
if((cl 3l [b]==0)&&(rw4l [b]==0)&&(pd7l [b]==0)&&(pb6l [b]==0)){
if(uflg[d]==0){
nm[24]=d; uflg[d]=1;
cl 3h[a]=1; rw4h[a]=1; pd7h[a]=1; pb6h[a]=1;
cl 3l [b]=1; rw4l [b]=1; pd7l [b]=1; pb6l [b]=1;
stp41();
cl 3h[a]=0; rw4h[a]=0; pd7h[a]=0; pb6h[a]=0;
cl 3l [b]=0; rw4l [b]=0; pd7l [b]=0; pb6l [b]=0;
uflg[d]=0; }}
}}}
}
}
/* Set n27 */
void stp41(){
short a, b, d;
for(a=0; a<7; a++){
if((cl 6h[a]==0)&&(rw4h[a]==0)&&(pd3h[a]==0)&&(pb2h[a]==0)){
for(b=0; b<7; b++){d=7*a+b;
if((cl 6l [b]==0)&&(rw4l [b]==0)&&(pd3l [b]==0)&&(pb2l [b]==0)){
if(uflg[d]==0){
nm[27]=d; uflg[d]=1;
cl 6h[a]=1; rw4h[a]=1; pd3h[a]=1; pb2h[a]=1;
cl 6l [b]=1; rw4l [b]=1; pd3l [b]=1; pb2l [b]=1;
stp42();
cl 6h[a]=0; rw4h[a]=0; pd3h[a]=0; pb2h[a]=0;
cl 6l [b]=0; rw4l [b]=0; pd3l [b]=0; pb2l [b]=0;
uflg[d]=0; }}
}}}
}
}
/* Set n31 */
void stp42(){
short a, b, d;
for(a=0; a<7; a++){
if((cl 3h[a]==0)&&(rw5h[a]==0)&&(pd6h[a]==0)&&(pb7h[a]==0)){
for(b=0; b<7; b++){d=7*a+b;
if((cl 3l [b]==0)&&(rw5l [b]==0)&&(pd6l [b]==0)&&(pb7l [b]==0)){
if(uflg[d]==0){

```

```

        nm[31]=d; ufl g[d]=1;
        cl 3h[a]=1; rw5h[a]=1; pd6h[a]=1; pb7h[a]=1;
        cl 3l [b]=1; rw5l [b]=1; pd6l [b]=1; pb7l [b]=1;
        stp43();
        cl 3h[a]=0; rw5h[a]=0; pd6h[a]=0; pb7h[a]=0;
        cl 3l [b]=0; rw5l [b]=0; pd6l [b]=0; pb7l [b]=0;
        ufl g[d]=0; }}
    }}
}
}
/* Set n25 */
void stp43(){
    short a, b, d;
    for(a=0; a<7; a++){
        if((cl 4h[a]==0)&&(rw4h[a]==0)&&(pd1h[a]==0)&&(pb7h[a]==0)){
            for(b=0; b<7; b++){d=7*a+b;
                if((cl 4l [b]==0)&&(rw4l [b]==0)&&(pd1l [b]==0)&&(pb7l [b]==0)){
                    if(ufl g[d]==0){
                        nm[25]=d; ufl g[d]=1;
                        cl 4h[a]=1; rw4h[a]=1; pd1h[a]=1; pb7h[a]=1;
                        cl 4l [b]=1; rw4l [b]=1; pd1l [b]=1; pb7l [b]=1;
                        stp44();
                        cl 4h[a]=0; rw4h[a]=0; pd1h[a]=0; pb7h[a]=0;
                        cl 4l [b]=0; rw4l [b]=0; pd1l [b]=0; pb7l [b]=0;
                        ufl g[d]=0; }}
                }}
            }
        }
}
/* Set n26 */
void stp44(){
    short a, b, d;
    for(a=0; a<7; a++){
        if((cl 5h[a]==0)&&(rw4h[a]==0)&&(pd2h[a]==0)&&(pb1h[a]==0)){
            for(b=0; b<7; b++){d=7*a+b;
                if((cl 5l [b]==0)&&(rw4l [b]==0)&&(pd2l [b]==0)&&(pb1l [b]==0)){
                    if(ufl g[d]==0){
                        nm[26]=d; ufl g[d]=1;
                        cl 5h[a]=1; rw4h[a]=1; pd2h[a]=1; pb1h[a]=1;
                        cl 5l [b]=1; rw4l [b]=1; pd2l [b]=1; pb1l [b]=1;
                        stp45();
                        cl 5h[a]=0; rw4h[a]=0; pd2h[a]=0; pb1h[a]=0;
                        cl 5l [b]=0; rw4l [b]=0; pd2l [b]=0; pb1l [b]=0;
                        ufl g[d]=0; }}
                }}
            }
        }
}
/* Set n32 */
void stp45(){
    short a, b, d;
    for(a=0; a<7; a++){
        if((cl 4h[a]==0)&&(rw5h[a]==0)&&(pd7h[a]==0)&&(pb1h[a]==0)){
            for(b=0; b<7; b++){d=7*a+b;
                if((cl 4l [b]==0)&&(rw5l [b]==0)&&(pd7l [b]==0)&&(pb1l [b]==0)){
                    if(ufl g[d]==0){
                        nm[32]=d; ufl g[d]=1;
                        cl 4h[a]=1; rw5h[a]=1; pd7h[a]=1; pb1h[a]=1;
                        cl 4l [b]=1; rw5l [b]=1; pd7l [b]=1; pb1l [b]=1;
                        stp46();
                        cl 4h[a]=0; rw5h[a]=0; pd7h[a]=0; pb1h[a]=0;
                        cl 4l [b]=0; rw5l [b]=0; pd7l [b]=0; pb1l [b]=0;
                        ufl g[d]=0; }}
                }}
            }
        }
}
}
}

```

```

/* Set n34 */
void stp46(){
short a, b, d;
for(a=0; a<7; a++){
if((cl 6h[a]==0)&&(rw5h[a]==0)&&(pd2h[a]==0)&&(pb3h[a]==0)){
for(b=0; b<7; b++){d=7*a+b;
if((cl 6l [b]==0)&&(rw5l [b]==0)&&(pd2l [b]==0)&&(pb3l [b]==0)){
if(uf1g[d]==0){
nm[34]=d; uf1g[d]=1;
cl 6h[a]=1; rw5h[a]=1; pd2h[a]=1; pb3h[a]=1;
cl 6l [b]=1; rw5l [b]=1; pd2l [b]=1; pb3l [b]=1;
stp47();
cl 6h[a]=0; rw5h[a]=0; pd2h[a]=0; pb3h[a]=0;
cl 6l [b]=0; rw5l [b]=0; pd2l [b]=0; pb3l [b]=0;
uf1g[d]=0; }}
}}
}
}
/* Set n33 */
void stp47(){
short a, b, d;
for(a=6; a>=0; a--){
if((cl 5h[a]==0)&&(rw5h[a]==0)&&(pd1h[a]==0)&&(pb2h[a]==0)){
for(b=6; b>=0; b--){d=7*a+b;
if((cl 5l [b]==0)&&(rw5l [b]==0)&&(pd1l [b]==0)&&(pb2l [b]==0)){
if(uf1g[d]==0){
nm[33]=d; uf1g[d]=1;
cl 5h[a]=1; rw5h[a]=1; pd1h[a]=1; pb2h[a]=1;
cl 5l [b]=1; rw5l [b]=1; pd1l [b]=1; pb2l [b]=1;
stp48();
cl 5h[a]=0; rw5h[a]=0; pd1h[a]=0; pb2h[a]=0;
cl 5l [b]=0; rw5l [b]=0; pd1l [b]=0; pb2l [b]=0;
uf1g[d]=0; }}
}}
}
}
/* Set n39 */
void stp48(){
short a, b, d;
for(a=0; a<7; a++){
if((cl 4h[a]==0)&&(rw6h[a]==0)&&(pd6h[a]==0)&&(pb2h[a]==0)){
for(b=0; b<7; b++){d=7*a+b;
if((cl 4l [b]==0)&&(rw6l [b]==0)&&(pd6l [b]==0)&&(pb2l [b]==0)){
if(uf1g[d]==0){
nm[39]=d; uf1g[d]=1;
cl 4h[a]=1; rw6h[a]=1; pd6h[a]=1; pb2h[a]=1;
cl 4l [b]=1; rw6l [b]=1; pd6l [b]=1; pb2l [b]=1;
stp49();
cl 4h[a]=0; rw6h[a]=0; pd6h[a]=0; pb2h[a]=0;
cl 4l [b]=0; rw6l [b]=0; pd6l [b]=0; pb2l [b]=0;
uf1g[d]=0; }}
}}
}
}
/* Set n40 */
void stp49(){
short a, b, d;
for(a=0; a<7; a++){
if((cl 5h[a]==0)&&(rw6h[a]==0)&&(pd7h[a]==0)&&(pb3h[a]==0)){
for(b=0; b<7; b++){d=7*a+b;
if((cl 5l [b]==0)&&(rw6l [b]==0)&&(pd7l [b]==0)&&(pb3l [b]==0)){
if(uf1g[d]==0){
nm[40]=d; uf1g[d]=1;
cl 5h[a]=1; rw6h[a]=1; pd7h[a]=1; pb3h[a]=1;

```

```

        cl5l [b]=1; rw6l [b]=1; pd7l [b]=1; pb3l [b]=1;
        printans();
        cl5h[a]=0; rw6h[a]=0; pd7h[a]=0; pb3h[a]=0;
        cl5l [b]=0; rw6l [b]=0; pd7l [b]=0; pb3l [b]=0;
        uflg[d]=0; }}
    }}
}
}
/**/
/* Print New Answer */
void printans(){
    short n;
    cnt++; cnt2++;
    if(cnt2==1){
        cntr[cnt3]=cnt;
        for(n=1; n<50; n++){anm[cnt3*2][n]=nm[n]+1; anm[cnt3*2+1][n]=nm[n]; }
        cnt3++; if(cnt3==2){print2p(); cntr[1]=0; cnt3=0; }
    }
}
/**/
/* Print 2 Answers with their Decompositions */
void print2p(){
    short l, l7, n;
    printf(" /D7i H/      L/%21d#      /D7i H/      L/%21d#\n", cntr[0], cntr[1]);
    for(l=0; l<7; l++){l7=l*7;
        printf(" ");
        for(n=1; n<8; n++){printf("%d", anm[1][l7+n]/7); }
        printf(" ");
        for(n=1; n<8; n++){printf("%d", anm[1][l7+n]%7); }
        printf(" ");
        for(n=1; n<8; n++){printf("%3d", anm[0][l7+n]); }
        printf(" ");
        for(n=1; n<8; n++){printf("%d", anm[3][l7+n]/7); }
        printf(" ");
        for(n=1; n<8; n++){printf("%d", anm[3][l7+n]%7); }
        printf(" ");
        for(n=1; n<8; n++){printf("%3d", anm[2][l7+n]); }
        printf("\n");
    }
}
/**/

```

I put the next list-forming inequality condition complex to this program:

{  $n_1 < n_{49}$ ;  $n_1 < n_7$ ;  $n_1 < n_{43}$ ;  $n_2 > n_8$ ;  $n_2 > n_7$ ; and  $n_8 > n_{43}$ ; },

because we have to make and count only the fundamental solutions of that type.

It took about four hours to count them up to the last by my old machine.

I was really surprised at this extreme speed. I have never imagined I could really count up the whole solutions of that type through while I am living.

The extract data of my result is listed as below:

## #8. Result List of My Recent Computation

\* Fundamental Solutions with  $n_1=1$  of 'Complete Euler Type' of PDMS77 \*

/D7i H/	L/	1#	/D7i H/	L/	3457#
0654321	0654321	1 49 41 33 25 17 9	0654321	0564321	1 48 42 33 25 17 9
5432106	4321065	40 32 24 16 8 7 48	5432106	4321056	40 32 24 16 8 6 49
3210654	1065432	23 15 14 6 47 39 31	3210654	1056432	23 15 13 7 47 39 31
1065432	5432106	13 5 46 38 30 22 21	1065432	6432105	14 5 46 38 30 22 20
6543210	2106543	45 37 29 28 20 12 4	6543210	2105643	45 37 29 27 21 12 4
4321065	6543210	35 27 19 11 3 44 36	4321065	5643210	34 28 19 11 3 44 36
2106543	3210654	18 10 2 43 42 34 26	2106543	3210564	18 10 2 43 41 35 26

/D7i H/	L/		6913#	/D7i H/	L/		10369#
0654321	0465321	1 47 42 34 25 17 9		0654321	0365421	1 46 42 34 26 17 9	
5432106	5321046	41 32 24 16 8 5 49		5432106	5421036	41 33 24 16 8 4 49	
3210654	1046532	23 15 12 7 48 39 31		3210654	1036542	23 15 11 7 48 40 31	
1065432	6532104	14 6 46 38 30 22 19		1065432	6542103	14 6 47 38 30 22 18	
6543210	2104653	45 37 29 26 21 13 4		6543210	2103654	45 37 29 25 21 13 5	
4321065	4653210	33 28 20 11 3 44 36		4321065	3654210	32 28 20 12 3 44 36	
2106543	3210465	18 10 2 43 40 35 27		2106543	4210365	19 10 2 43 39 35 27	
/D7i H/	L/		13825#	/D7i H/	L/		17281#
0654321	0265431	1 45 42 34 26 18 9		0564321	0654321	1 42 48 33 25 17 9	
5432106	5431026	41 33 25 16 8 3 49		4321056	5432106	34 26 18 10 2 36 49	
3210654	1026543	23 15 10 7 48 40 32		1056432	3210654	11 3 37 43 35 27 19	
1065432	6543102	14 6 47 39 30 22 17		6432105	1065432	44 29 28 20 12 4 38	
6543210	3102654	46 37 29 24 21 13 5		2105643	6543210	21 13 5 39 45 30 22	
4321065	2654310	31 28 20 12 4 44 36		5643210	4321065	40 46 31 23 15 14 6	
2106543	4310265	19 11 2 43 38 35 27		3210564	2106543	24 16 8 7 41 47 32	
/D7i H/	L/		19009#	/D7i H/	L/		20737#
0564321	0564321	1 41 49 33 25 17 9		0564321	0465321	1 40 49 34 25 17 9	
4321056	6432105	35 26 18 10 2 36 48		4321056	6532104	35 27 18 10 2 36 47	
1056432	3210564	11 3 37 43 34 28 19		1056432	3210465	11 3 37 43 33 28 20	
6432105	1056432	44 29 27 21 12 4 38		6432105	1046532	44 29 26 21 13 4 38	
2105643	5643210	20 14 5 39 45 30 22		2105643	4653210	19 14 6 39 45 30 22	
5643210	4321056	40 46 31 23 15 13 7		5643210	5321046	41 46 31 23 15 12 7	
3210564	2105643	24 16 8 6 42 47 32		3210564	2104653	24 16 8 5 42 48 32	
/D7i H/	L/		22465#	/D7i H/	L/		24193#
0564321	0365421	1 39 49 34 26 17 9		0564321	0265431	1 38 49 34 26 18 9	
4321056	6542103	35 27 19 10 2 36 46		4321056	6543102	35 27 19 11 2 36 45	
1056432	4210365	12 3 37 43 32 28 20		1056432	4310265	12 4 37 43 31 28 20	
6432105	1036542	44 29 25 21 13 5 38		6432105	1026543	44 29 24 21 13 5 39	
2105643	3654210	18 14 6 40 45 30 22		2105643	2654310	17 14 6 40 46 30 22	
5643210	5421036	41 47 31 23 15 11 7		5643210	5431026	41 47 32 23 15 10 7	
3210564	2103654	24 16 8 4 42 48 33		3210564	3102654	25 16 8 3 42 48 33	
/D7i H/	L/		25921#	/D7i H/	L/		26497#
0463251	0654321	1 35 48 26 18 38 9		0463251	0564321	1 34 49 26 18 38 9	
3251046	5432106	27 19 39 10 2 29 49		3251046	6432105	28 19 39 10 2 29 48	
1046325	3210654	11 3 30 43 28 20 40		1046325	3210564	11 3 30 43 27 21 40	
6325104	1065432	44 22 21 41 12 4 31		6325104	1056432	44 22 20 42 12 4 31	
5104632	6543210	42 13 5 32 45 23 15		5104632	5643210	41 14 5 32 45 23 15	
4632510	4321065	33 46 24 16 36 14 6		4632510	4321056	33 46 24 16 36 13 7	
2510463	2106543	17 37 8 7 34 47 25		2510463	2105643	17 37 8 6 35 47 25	
/D7i H/	L/		27073#	/D7i H/	L/		27649#
0463251	0465321	1 33 49 27 18 38 9		0463251	0365421	1 32 49 27 19 38 9	
3251046	6532104	28 20 39 10 2 29 47		3251046	6542103	28 20 40 10 2 29 46	
1046325	3210465	11 3 30 43 26 21 41		1046325	4210365	12 3 30 43 25 21 41	
6325104	1046532	44 22 19 42 13 4 31		6325104	1036542	44 22 18 42 13 5 31	
5104632	4653210	40 14 6 32 45 23 15		5104632	3654210	39 14 6 33 45 23 15	
4632510	5321046	34 46 24 16 36 12 7		4632510	5421036	34 47 24 16 36 11 7	
2510463	2104653	17 37 8 5 35 48 25		2510463	2103654	17 37 8 4 35 48 26	
/D7i H/	L/		28225#	/D7i H/	L/		28801#
0463251	0265431	1 31 49 27 19 39 9		0654321	0654312	1 49 41 33 25 16 10	
3251046	6543102	28 20 40 11 2 29 45		5432106	4312065	40 32 23 17 8 7 48	
1046325	4310265	12 4 30 43 24 21 41		3210654	2065431	24 15 14 6 47 39 30	
6325104	1026543	44 22 17 42 13 5 32		1065432	5431206	13 5 46 37 31 22 21	
5104632	2654310	38 14 6 33 46 23 15		6543210	1206543	44 38 29 28 20 12 4	
4632510	5431026	34 47 25 16 36 10 7		4321065	6543120	35 27 19 11 2 45 36	
2510463	3102654	18 37 8 3 35 48 26		2106543	3120654	18 9 3 43 42 34 26	

/D7i H/	L/	57601#	/D7i H/	L/	86401#
0654321	0654213	1 49 41 33 24 16 11	0654321	0653214	1 49 41 32 24 16 12
5432106	4213065	40 31 23 18 8 7 48	5432106	3214065	39 31 23 19 8 7 48
3210654	3065421	25 15 14 6 47 38 30	3210654	4065321	26 15 14 6 46 38 30
1065432	5421306	13 5 45 37 32 22 21	1065432	5321406	13 4 45 37 33 22 21
6543210	1306542	44 39 29 28 20 12 3	6543210	1406532	44 40 29 28 20 11 3
4321065	6542130	35 27 19 10 2 46 36	4321065	6532140	35 27 18 10 2 47 36
2106543	2130654	17 9 4 43 42 34 26	2106543	2140653	17 9 5 43 42 34 25
/D7i H/	L/	115201#	/D7i H/	L/	144001#
0654321	0643215	1 49 40 32 24 16 13	0654321	0543216	1 48 40 32 24 16 14
5432106	3215064	39 31 23 20 8 7 47	5432106	3216054	39 31 23 21 8 6 47
3210654	5064321	27 15 14 5 46 38 30	3210654	6054321	28 15 13 5 46 38 30
1065432	4321506	12 4 45 37 34 22 21	1065432	4321605	12 4 45 37 35 22 20
6543210	1506432	44 41 29 28 19 11 3	6543210	1605432	44 42 29 27 19 11 3
4321065	6432150	35 26 18 10 2 48 36	4321065	5432160	34 26 18 10 2 49 36
2106543	2150643	17 9 6 43 42 33 25	2106543	2160543	17 9 7 43 41 33 25
/D7i H/	L/	172801#	/D7i H/	L/	201601#
0654312	0654321	1 49 41 33 25 10 16	0654312	0654312	1 49 41 33 25 9 17
5431206	4321065	40 32 24 9 15 7 48	5431206	4312065	40 32 23 10 15 7 48
3120654	1065432	23 8 21 6 47 39 31	3120654	2065431	24 8 21 6 47 39 30
2065431	5432106	20 5 46 38 30 22 14	2065431	5431206	20 5 46 37 31 22 14
6543120	2106543	45 37 29 28 13 19 4	6543120	1206543	44 38 29 28 13 19 4
4312065	6543210	35 27 12 18 3 44 36	4312065	6543120	35 27 12 18 2 45 36
1206543	3210654	11 17 2 43 42 34 26	1206543	3120654	11 16 3 43 42 34 26
/D7i H/	L/	230401#	/D7i H/	L/	259201#
0654312	0654213	1 49 41 33 24 9 18	0654312	0653214	1 49 41 32 24 9 19
5431206	4213065	40 31 23 11 15 7 48	5431206	3214065	39 31 23 12 15 7 48
3120654	3065421	25 8 21 6 47 38 30	3120654	4065321	26 8 21 6 46 38 30
2065431	5421306	20 5 45 37 32 22 14	2065431	5321406	20 4 45 37 33 22 14
6543120	1306542	44 39 29 28 13 19 3	6543120	1406532	44 40 29 28 13 18 3
4312065	6542130	35 27 12 17 2 46 36	4312065	6532140	35 27 11 17 2 47 36
1206543	2130654	10 16 4 43 42 34 26	1206543	2140653	10 16 5 43 42 34 25
/D7i H/	L/	288001#	/D7i H/	L/	316801#
0654312	0643215	1 49 40 32 24 9 20	0654312	0543216	1 48 40 32 24 9 21
5431206	3215064	39 31 23 13 15 7 47	5431206	3216054	39 31 23 14 15 6 47
3120654	5064321	27 8 21 5 46 38 30	3120654	6054321	28 8 20 5 46 38 30
2065431	4321506	19 4 45 37 34 22 14	2065431	4321605	19 4 45 37 35 22 13
6543120	1506432	44 41 29 28 12 18 3	6543120	1605432	44 42 29 27 12 18 3
4312065	6432150	35 26 11 17 2 48 36	4312065	5432160	34 26 11 17 2 49 36
1206543	2150643	10 16 6 43 42 33 25	1206543	2160543	10 16 7 43 41 33 25
/D7i H/	L/	345601#	/D7i H/	L/	374401#
0654213	0654321	1 49 41 33 18 10 23	0654213	0654312	1 49 41 33 18 9 24
5421306	4321065	40 32 17 9 22 7 48	5421306	4312065	40 32 16 10 22 7 48
2130654	1065432	16 8 28 6 47 39 31	2130654	2065431	17 8 28 6 47 39 30
3065421	5432106	27 5 46 38 30 15 14	3065421	5431206	27 5 46 37 31 15 14
6542130	2106543	45 37 29 21 13 26 4	6542130	1206543	44 38 29 21 13 26 4
4213065	6543210	35 20 12 25 3 44 36	4213065	6543120	35 20 12 25 2 45 36
1306542	3210654	11 24 2 43 42 34 19	1306542	3120654	11 23 3 43 42 34 19
/D7i H/	L/	403201#	/D7i H/	L/	432001#
0654213	0654213	1 49 41 33 17 9 25	0654213	0653214	1 49 41 32 17 9 26
5421306	4213065	40 31 16 11 22 7 48	5421306	3214065	39 31 16 12 22 7 48
2130654	3065421	18 8 28 6 47 38 30	2130654	4065321	19 8 28 6 46 38 30
3065421	5421306	27 5 45 37 32 15 14	3065421	5321406	27 4 45 37 33 15 14
6542130	1306542	44 39 29 21 13 26 3	6542130	1406532	44 40 29 21 13 25 3
4213065	6542130	35 20 12 24 2 46 36	4213065	6532140	35 20 11 24 2 47 36
1306542	2130654	10 23 4 43 42 34 19	1306542	2140653	10 23 5 43 42 34 18

/D7i H/	L/		460801#	/D7i H/	L/		489601#
0654213	0643215	1 49 40 32 17 9 27		0654213	0543216	1 48 40 32 17 9 28	
5421306	3215064	39 31 16 13 22 7 47		5421306	3216054	39 31 16 14 22 6 47	
2130654	5064321	20 8 28 5 46 38 30		2130654	6054321	21 8 27 5 46 38 30	
3065421	4321506	26 4 45 37 34 15 14		3065421	4321605	26 4 45 37 35 15 13	
6542130	1506432	44 41 29 21 12 25 3		6542130	1605432	44 42 29 20 12 25 3	
4213065	6432150	35 19 11 24 2 48 36		4213065	5432160	34 19 11 24 2 49 36	
1306542	2150643	10 23 6 43 42 33 18		1306542	2160543	10 23 7 43 41 33 18	
/D7i H/	L/		518401#	/D7i H/	L/		544321#
0653214	0654321	1 49 41 26 18 10 30		0653214	0654312	1 49 41 26 18 9 31	
5321406	4321065	40 25 17 9 29 7 48		5321406	4312065	40 25 16 10 29 7 48	
2140653	1065432	16 8 35 6 47 39 24		2140653	2065431	17 8 35 6 47 39 23	
4065321	5432106	34 5 46 38 23 15 14		4065321	5431206	34 5 46 37 24 15 14	
6532140	2106543	45 37 22 21 13 33 4		6532140	1206543	44 38 22 21 13 33 4	
3214065	6543210	28 20 12 32 3 44 36		3214065	6543120	28 20 12 32 2 45 36	
1406532	3210654	11 31 2 43 42 27 19		1406532	3120654	11 30 3 43 42 27 19	
/D7i H/	L/		570241#	/D7i H/	L/		596161#
0653214	0654213	1 49 41 26 17 9 32		0653214	0653214	1 49 41 25 17 9 33	
5321406	4213065	40 24 16 11 29 7 48		5321406	3214065	39 24 16 12 29 7 48	
2140653	3065421	18 8 35 6 47 38 23		2140653	4065321	19 8 35 6 46 38 23	
4065321	5421306	34 5 45 37 25 15 14		4065321	5321406	34 4 45 37 26 15 14	
6532140	1306542	44 39 22 21 13 33 3		6532140	1406532	44 40 22 21 13 32 3	
3214065	6542130	28 20 12 31 2 46 36		3214065	6532140	28 20 11 31 2 47 36	
1406532	2130654	10 30 4 43 42 27 19		1406532	2140653	10 30 5 43 42 27 18	
/D7i H/	L/		622081#	/D7i H/	L/		648001#
0653214	0643215	1 49 40 25 17 9 34		0653214	0543216	1 48 40 25 17 9 35	
5321406	3215064	39 24 16 13 29 7 47		5321406	3216054	39 24 16 14 29 6 47	
2140653	5064321	20 8 35 5 46 38 23		2140653	6054321	21 8 34 5 46 38 23	
4065321	4321506	33 4 45 37 27 15 14		4065321	4321605	33 4 45 37 28 15 13	
6532140	1506432	44 41 22 21 12 32 3		6532140	1605432	44 42 22 20 12 32 3	
3214065	6432150	28 19 11 31 2 48 36		3214065	5432160	27 19 11 31 2 49 36	
1406532	2150643	10 30 6 43 42 26 18		1406532	2160543	10 30 7 43 41 26 18	
/D7i H/	L/		673921#	/D7i H/	L/		691201#
0643215	0654321	1 49 34 26 18 10 37		0643215	0654312	1 49 34 26 18 9 38	
4321506	4321065	33 25 17 9 36 7 48		4321506	4312065	33 25 16 10 36 7 48	
2150643	1065432	16 8 42 6 47 32 24		2150643	2065431	17 8 42 6 47 32 23	
5064321	5432106	41 5 46 31 23 15 14		5064321	5431206	41 5 46 30 24 15 14	
6432150	2106543	45 30 22 21 13 40 4		6432150	1206543	44 31 22 21 13 40 4	
3215064	6543210	28 20 12 39 3 44 29		3215064	6543120	28 20 12 39 2 45 29	
1506432	3210654	11 38 2 43 35 27 19		1506432	3120654	11 37 3 43 35 27 19	
/D7i H/	L/		708481#	/D7i H/	L/		725761#
0643215	0654213	1 49 34 26 17 9 39		0643215	0653214	1 49 34 25 17 9 40	
4321506	4213065	33 24 16 11 36 7 48		4321506	3214065	32 24 16 12 36 7 48	
2150643	3065421	18 8 42 6 47 31 23		2150643	4065321	19 8 42 6 46 31 23	
5064321	5421306	41 5 45 30 25 15 14		5064321	5321406	41 4 45 30 26 15 14	
6432150	1306542	44 32 22 21 13 40 3		6432150	1406532	44 33 22 21 13 39 3	
3215064	6542130	28 20 12 38 2 46 29		3215064	6532140	28 20 11 38 2 47 29	
1506432	2130654	10 37 4 43 35 27 19		1506432	2140653	10 37 5 43 35 27 18	
/D7i H/	L/		743041#	/D7i H/	L/		760321#
0643215	0643215	1 49 33 25 17 9 41		0643215	0543216	1 48 33 25 17 9 42	
4321506	3215064	32 24 16 13 36 7 47		4321506	3216054	32 24 16 14 36 6 47	
2150643	5064321	20 8 42 5 46 31 23		2150643	6054321	21 8 41 5 46 31 23	
5064321	4321506	40 4 45 30 27 15 14		5064321	4321605	40 4 45 30 28 15 13	
6432150	1506432	44 34 22 21 12 39 3		6432150	1605432	44 35 22 20 12 39 3	
3215064	6432150	28 19 11 38 2 48 29		3215064	5432160	27 19 11 38 2 49 29	
1506432	2150643	10 37 6 43 35 26 18		1506432	2160543	10 37 7 43 34 26 18	

\* [Count = 777600] OK! \*

The total count of standard solutions of that type is logically calculated to be  $777600 \times 7 \times 7 = 38102400$ . You can reconstruct all of them by the fundamental solutions we have just got, using the typical transformations of pandiagonal rotation and shift of every row and column of them.

Of course, there exist lots of "Non-Euler Type" of PDMS77, say, far more than our Complete Euler Type. We can not even guess how many of them might really exist.

Anyhow it makes me really happy that we have come to know how many solutions of 'Complete Euler Squares' of order 7 for pan-diagonal type do really exist.

#### #9. Comment

This is the first successful result of my elaborate computation to try to make any Complete Euler type of pan-diagonal magic squares of order 7. I am proud of my experience I could surely have got some skills to solve the typical problems and could have any special key in my hand to open the door for the new world.

On top of that I could have fortunately invented another two ways of composing Complete Euler Squares: "Compositions by Knight's Tour", and "New Euler's Method".

I want you to read my articles I explained about them in the following sections.

You can do your job very fast by these new methods, faster than anything else.

#### #10. How about the Cases of 'Complete Euler Squares' of any other Orders?

You may want to go on up to order 9, but you may not try to make 'Complete Euler Squares' 9x9 of any pan-diagonal type, even of the simultaneous one.

Number system of base 9 does not seem to work very well for this object. Anything like 'Complete Euler Square' has not yet been found.

Base 3 might work well instead. But four positional layers of base 3 may force you too many conditions in your program, and I am sure you cannot help feeling this method is too heavy to hold on.

Number system of base 4 does not work perfect for all pan-diagonal magic squares of order 4. But base 2, that means, binary number system works perfect instead.

You cannot make any 'Complete Euler Square' 6x6 at all. It is logically proved that any pan-diagonal type of magic square is impossible to be constructed.

You can make all Complete Euler Squares of order 8 only by binary number system.

Only "Composite and Pan-diagonal" type of magic squares of order 8 (including "Composite and Complete" type) are all 'Complete Euler Squares.'

"Greco-Latinian Method" will be able to work well for order 11, 13, 17, .... , I guess.

(Original Written in English on March 27, 2003 by Kanji Setsuda  
Retyped Edition on February 22, 2007 with MacOSX and Xcode2.2)

---

E-Mail Address <jag12001@ni fty. com>