

### 第3部：『魔方陣の新研究』： 撰田 寛二

#### 第4章： 六方陣の代数的研究（新）

#### 第4節： 六次の「普通の魔方陣」について（最新）

##### 0. 目標

前節までの研究で，六次の場合，補数対称型も汎対角線型も構成が不可能とわかった。正方連結型の場合でも，主対角線2本さえもが一義的には定和にならない。

また四次や八次では作れた「完全オイラー方陣」が，六次では作れない。三進法や二進法でやってもだめである。六次のこの不自由さは，いったいどうしたことだろう。

では，どんなタイプならば作れるというのだろうか。珍しいタイプの構成は諦めて，ごく普通の魔方陣を試してみるしかあるまい。行列の定和と主対角線2本の定和だけを求め，それ以外は要請しない。これならばどうであろうか。まさか作れないはずはあるまい。

主対角線2本を定和にしないでは，もはや魔方陣とは言い難い。何も出来ないとわかってからならともかく，最初から「半・魔方陣」を作るのは，意気が上がらない。解の総数が多過ぎるのも，求める価値が半減する。ここは，やはり，最低でも「魔方陣」を作ろう。

##### 1. 定義の基本図と基本条件式

5	6	1	2	3	4	5	6	1	2	$n1+n2+n3+n4+n5+n6=C$ .... (rw1)
11	12	7	8	9	10	11	12	7	8	$n7+n8+n9+n10+n11+n12=C$ .... (rw2)
17	18	13	14	15	16	17	18	13	14	$n13+n14+n15+n16+n17+n18=C$ .... (rw3)
23	24	19	20	21	22	23	24	19	20	$n19+n20+n21+n22+n23+n24=C$ .... (rw4)
29	30	25	26	27	28	29	30	25	26	$n25+n26+n27+n28+n29+n30=C$ .... (rw5)
35	36	31	32	33	34	35	36	31	32	$n31+n32+n33+n34+n35+n36=C$ .... (rw6)
										$n1+n7+n13+n19+n25+n31=C$ .... (cl1)
										$n2+n8+n14+n20+n26+n32=C$ .... (cl2)
										$n3+n9+n15+n21+n27+n33=C$ .... (cl3)
										$n4+n10+n16+n22+n28+n34=C$ .... (cl4)
										$n5+n11+n17+n23+n29+n35=C$ .... (cl5)
										$n6+n12+n18+n24+n30+n36=C$ .... (cl6)
										$n1+n8+n15+n22+n29+n36=C$ .... (pd1)
										$n6+n11+n16+n21+n26+n31=C$ .... (pd2)

##### 2. 解のリスト

以上の条件下で普通にプログラムを組んで，解を求めさせたところ，下のようなリストを得た。解の出方は，非常に遅い。とても最後まで行き着かず，短時間には解の総数を決定できなかった。あらためて六次が既に「高次」であることを実感させられた。

##### \*\* 六次の標準型魔方陣の解のリスト(抜粋) \*\*

1/	39/	75/	131/
1 34 9 33 32 2	1 34 9 33 32 2	1 34 9 33 32 2	1 34 9 33 32 2
31 14 25 7 13 21	31 16 25 7 20 12	31 16 26 7 18 13	31 16 25 10 17 12
6 20 26 16 24 19	6 18 27 14 22 24	6 20 24 15 25 21	6 23 27 15 21 19
30 11 18 12 17 23	30 15 17 10 13 26	30 14 19 11 10 27	30 11 20 7 14 29
8 27 29 15 22 10	8 23 29 19 21 11	8 22 29 17 23 12	8 22 26 18 24 13
35 5 4 28 3 36	35 5 4 28 3 36	35 5 4 28 3 36	35 5 4 28 3 36
284/	374/	545/	745/
1 34 9 33 32 2	1 34 9 33 32 2	1 34 9 33 32 2	1 34 9 33 32 2
31 16 26 10 17 11	31 18 24 12 16 10	31 20 23 10 14 13	31 14 29 7 20 10
6 21 27 15 23 19	6 20 27 14 25 19	6 19 25 15 29 17	6 24 25 15 22 19
30 13 20 7 12 29	30 11 21 7 13 29	30 12 24 7 11 27	30 13 18 12 11 27
8 22 25 18 24 14	8 23 26 17 22 15	8 21 26 18 22 16	8 21 26 16 23 17
35 5 4 28 3 36	35 5 4 28 3 36	35 5 4 28 3 36	35 5 4 28 3 36

941/ 1 34 9 33 32 2 31 12 24 14 19 11 6 25 29 13 21 17 30 15 22 7 10 27 8 20 23 16 26 18 35 5 4 28 3 36	1167/ 1 34 9 33 32 2 31 17 22 10 15 16 6 25 26 20 23 11 30 12 21 7 14 27 8 18 29 13 24 19 35 5 4 28 3 36	1459/ 1 34 9 33 32 2 31 18 21 10 16 15 6 25 26 19 24 11 30 12 22 7 13 27 8 17 29 14 23 20 35 5 4 28 3 36	1654/ 1 34 9 33 32 2 31 16 25 7 20 12 6 24 26 19 23 13 30 15 18 10 11 27 8 17 29 14 22 21 35 5 4 28 3 36
2036/ 1 34 9 33 32 2 31 14 26 10 19 11 6 25 29 17 21 13 30 15 20 7 12 27 8 18 23 16 24 22 35 5 4 28 3 36	2275/ 1 34 9 33 32 2 31 14 27 10 17 12 6 21 29 22 20 13 30 18 16 7 15 25 8 19 26 11 24 23 35 5 4 28 3 36	2573/ 1 34 9 33 32 2 31 15 26 12 16 11 6 27 29 17 19 13 30 10 21 7 18 25 8 20 22 14 23 24 35 5 4 28 3 36	2784/ 1 34 9 33 32 2 31 15 22 13 16 14 6 27 29 19 20 10 30 12 21 7 17 24 8 18 26 11 23 25 35 5 4 28 3 36
3103/ 1 34 9 33 32 2 31 19 22 11 21 7 6 24 27 16 23 15 30 12 20 10 14 25 8 17 29 13 18 26 35 5 4 28 3 36	3303/ 1 34 9 33 32 2 31 15 25 7 21 12 6 24 29 19 22 11 30 17 18 10 13 23 8 16 26 14 20 27 35 5 4 28 3 36	3599/ 1 34 9 33 32 2 31 18 27 7 16 12 6 24 26 19 25 11 30 13 22 10 15 21 8 17 23 14 20 29 35 5 4 28 3 36	3850/ 1 34 9 33 32 2 31 12 22 10 15 21 6 23 29 18 16 19 27 13 17 8 20 26 11 24 30 14 25 7 35 5 4 28 3 36
3875/ 1 34 9 33 32 2 31 12 22 10 16 20 6 24 30 18 14 19 27 13 17 7 21 26 11 23 29 15 25 8 35 5 4 28 3 36	3928/ 1 34 9 33 32 2 31 16 29 8 13 14 6 21 25 18 22 19 27 12 20 7 15 30 11 23 24 17 26 10 35 5 4 28 3 36	4098/ 1 34 9 33 32 2 31 17 20 10 15 18 6 25 29 14 24 13 27 8 23 7 16 30 11 22 26 19 21 12 35 5 4 28 3 36	4247/ 1 34 9 33 32 2 31 8 23 16 14 19 6 24 29 17 20 15 27 18 21 7 12 26 11 22 25 10 30 13 35 5 4 28 3 36
4335/ 1 34 9 33 32 2 31 15 26 13 16 10 6 25 29 12 20 19 27 8 22 7 17 30 11 24 21 18 23 14 35 5 4 28 3 36	4605/ 1 34 9 33 32 2 31 16 25 8 24 7 6 23 26 13 22 21 27 14 18 12 10 30 11 19 29 17 20 15 35 5 4 28 3 36	4722/ 1 34 9 33 32 2 31 18 29 7 12 14 6 21 26 20 25 13 27 10 19 8 17 30 11 23 24 15 22 16 35 5 4 28 3 36	4963/ 1 34 9 33 32 2 31 18 20 14 21 7 6 26 25 13 22 19 27 12 24 8 10 30 11 16 29 15 23 17 35 5 4 28 3 36

.....

### 3. 解の検定試験

解の1つを選んで，検定試験を施そう。n進法分解図を作ったり，行列・汎対角線の和や対称型の補数の和を求めたりして，どの定和が成り立つかを実際に調べてみよう。

1/古典的表記

1	34	9	33	32	2
31	16	20	12	14	18
7	21	27	13	24	19
26	10	22	8	15	30
11	25	29	17	23	6
35	5	4	28	3	36

/数学的表記 ( 拡張空間 )

32	31	1	0	33	8	32	31	1	0	33	8
11	13	17	30	15	19	11	13	17	30	15	19
12	23	18	6	20	26	12	23	18	6	20	26
7	14	29	25	9	21	7	14	29	25	9	21
16	22	5	10	24	28	16	22	5	10	24	28
27	2	35	34	4	3	27	2	35	34	4	3

\* 六進法による表記図・分解図 \*

/N6i	/D6i	*6/	*1/
00 53 12 52 51 01	0 5 1 5 5 0	0 3 2 2 1 1	
50 23 31 15 21 25	5 2 3 1 2 2	0 3 1 5 1 5	
10 32 42 20 35 30	1 3 4 2 3 3	0 2 2 0 5 0	
41 13 33 11 22 45	4 1 3 1 2 4	1 3 3 1 2 5	
14 40 44 24 34 05	1 4 4 2 3 0	4 0 4 4 4 5	
54 04 03 43 02 55	5 0 0 4 0 5	4 4 3 3 2 5	

\* 行列・汎対角線の定和を調べる \*

0+33+8+32+31+1=	(0+5+1+5+5+0)*6+(0+3+2+2+1+1)=	105 ... rw1
30+15+19+11+13+17=	(5+2+3+1+2+2)*6+(0+3+1+5+1+5)=	105 ... rw2
6+20+26+12+23+18=	(1+3+4+2+3+3)*6+(0+2+2+0+5+0)=	105 ... rw3
25+9+21+7+14+29=	(4+1+3+1+2+4)*6+(1+3+3+1+2+5)=	105 ... rw4
10+24+28+16+22+5=	(1+4+4+2+3+0)*6+(4+0+4+4+4+5)=	105 ... rw5
34+4+3+27+2+35=	(5+0+0+4+0+5)*6+(4+4+3+3+2+5)=	105 ... rw6
0+30+6+25+10+34=	(0+5+1+4+1+5)*6+(0+0+0+1+4+4)=	105 ... cl1
33+15+20+9+24+4=	(5+2+3+1+4+0)*6+(3+3+2+3+0+4)=	105 ... cl2
8+19+26+21+28+3=	(1+3+4+3+4+0)*6+(2+1+2+3+4+3)=	105 ... cl3
32+11+12+7+16+27=	(5+1+2+1+2+4)*6+(2+5+0+1+4+3)=	105 ... cl4
31+13+23+14+22+2=	(5+2+3+2+3+0)*6+(1+1+5+2+4+2)=	105 ... cl5
1+17+18+29+5+35=	(0+2+3+4+0+5)*6+(1+5+0+5+5+5)=	105 ... cl6
0+15+26+7+22+35=	(0+2+4+1+3+5)*6+(0+3+2+1+4+5)=	105 ... pd1
1+13+12+21+24+34=	(0+2+2+3+4+5)*6+(1+1+0+3+0+4)=	105 ... pb6

... 確かに，これは基本的な標準型魔方陣である！

... だが，汎対角線型六方陣ではない！

0+35=35 ... cp1;	33+2=35 ... cp2;	8+27=35 ... cp3;	32+3=35 ... cp4;
31+4=35 ... cp5;	1+34=35 ... cp6;	30+5=35 ... cp7;	15+22=37 ... cp8;
19+16=35 ... cp9;	11+28=39 ... cp10;	13+24=37 ... cp11;	17+10=27 ... cp12;
6+29=35 ... cp13;	20+14=34 ... cp14;	26+7=33 ... cp15;	12+21=33 ... cp16;
23+9=32 ... cp17;	18+25=43 ... cp18;		

... これは，補数対称型六方陣ではない！

\* 三進法による表記図・分解図 \*

/N3i	/D3i
0000 1020 0022 1012 1011 0001	010110 000000 022110 002211
1010 0120 0201 0102 0111 0122	100000 012111 120012 001212
0020 0202 0222 0110 0212 0200	000000 022122 202110 022020
0221 0100 0210 0021 0112 1002	000001 212010 201210 100122
0101 0220 1001 0121 0211 0012	001000 120120 020211 101112
1021 0011 0010 1000 0002 1022	100101 000000 211002 110022

\* 二進法による表記図・分解図 \*

/N2i
000000 100001 001000 100000 011111 000001
011110 001111 010011 001011 001101 010001
000110 010100 011010 001100 010111 010010
011001 001001 010101 000111 001110 011101
001010 011000 011100 010000 010110 000101
100010 000100 000011 011011 000010 100011
/D2i
010100 000010 001010 000010 000010 010011
000000 101001 110110 110010 111100 011111
000000 011011 001100 110110 101011 000010
000000 101001 110011 001111 000110 111101
000000 011110 111000 001011 100010 000001
100001 000100 000100 010000 101111 001101

確かに目的のものは、作れている。だが、六・三・二進法分解図のいずれを見ても、これは「オイラー型」にはなっていない。

「オイラー型」というのは、分解図の各面ですべての行列・主対角線の定和が、次の各式で表されるような形になっているものを言う。(…)内の要素の並び方は問題にしない。要素が使われている度数だけを問題にするのである。

\* オイラー方陣の条件 \*

$$(0+1+2+3+4+5) \times 6 + (0+1+2+3+4+5) = 105 \quad \dots /D6i$$

$$(0+0+1+1+2+2) \times 27 + (0+0+1+1+2+2) \times 9 + (0+0+1+1+2+2) \times 3 + (0+0+1+1+2+2) = 105 \quad \dots /D3i$$

$$(0+0+0+1+1+1) \times 32 + (0+0+0+1+1+1) \times 16 + (0+0+0+1+1+1) \times 8 + (0+0+0+1+1+1) \times 4 + (0+0+0+1+1+1) \times 2 + (0+0+0+1+1+1) = 105 \quad \dots /D2i$$

オイラー型になっていないものを「非オイラー型」もしくは「ノン・オイラー型」ということにしよう。上の例は、それである。

4. 六進法分解

では、六方阵は全て「ノン・オイラー型」なのだろうか。これはいささか想像し難い。何しろ普通の六方阵の解は無数にある。その中に少しぐらい「オイラー型」があっても良いのではないか。ここは、調べてみるしかあるまい。三進法・二進法によるものは最初から望みが無い(最上位が埋まらない)から、六進法分解図によるものを調べてみよう。

まずは、解のサンプルを小数個選んで、それを六進法分解してみる。

\*\* 六方阵の解の例と六進法分解図 \*\*

1/ /D6i						3/ /D6i									
1	34	9	33	32	2	051550	032211	1	34	9	33	32	2	051550	032211
31	14	25	7	13	21	524123	010002	31	11	25	7	18	19	514123	040050
6	20	26	16	24	19	034233	511350	6	20	27	13	24	21	034233	512052
30	11	18	12	17	23	412123	545544	30	15	17	14	12	23	422213	524154
8	27	29	15	22	10	144231	124233	8	26	29	16	22	10	144231	114333
35	5	4	28	3	36	500405	443325	35	5	4	28	3	36	500405	443325
8/ /D6i						9/ /D6i									
1	34	9	33	32	2	051550	032211	1	34	9	33	32	2	051550	032211
31	12	27	7	19	15	514132	052002	31	11	25	7	18	19	514123	040050
6	20	24	14	21	26	033234	515121	6	27	24	14	17	23	043223	525144
30	17	18	13	11	22	422213	545043	30	12	20	13	15	21	413223	551022
8	23	29	16	25	10	134241	144303	8	22	29	16	26	10	134241	134313
35	5	4	28	3	36	500405	443325	35	5	4	28	3	36	500405	443325
13/ /D6i						16/ /D6i									
1	34	9	33	32	2	051550	032211	1	34	9	33	32	2	051550	032211
31	12	24	13	16	15	513222	055032	31	14	25	11	17	13	524122	010440
6	18	29	11	20	27	024134	554412	6	19	27	12	18	29	034124	502554
30	17	22	7	14	21	423123	543012	30	16	22	7	15	21	423123	533022
8	25	23	19	26	10	143341	104013	8	23	24	20	26	10	133341	145113
35	5	4	28	3	36	500405	443325	35	5	4	28	3	36	500405	443325
17/ /D6i						19/ /D6i									
1	34	9	33	32	2	051550	032211	1	34	9	33	32	2	051550	032211
31	11	26	7	17	19	514123	041040	31	14	18	12	13	23	522123	015504
6	21	25	12	23	24	034133	520545	6	25	27	15	17	21	044223	502242
30	13	18	16	14	20	422223	505311	30	11	24	7	20	19	413133	545010
8	27	29	15	22	10	144231	124233	8	22	29	16	26	10	134241	134313
35	5	4	28	3	36	500405	443325	35	5	4	28	3	36	500405	443325

20/ /D6i	21/ /D6i
1 34 9 33 32 2 051550 032211	1 34 9 33 32 2 051550 032211
31 16 22 13 14 15 523222 033012	31 11 24 12 14 19 513123 045510
6 20 27 12 17 29 034124 512544	6 17 27 15 20 26 024234 542211
30 11 23 7 21 19 413133 544020	30 21 22 7 13 18 433122 523005
8 25 26 18 24 10 144231 101553	8 23 25 16 29 10 134241 140343
35 5 4 28 3 36 500405 443325	35 5 4 28 3 36 500405 443325
25/ /D6i	27/ /D6i
1 34 9 33 32 2 051550 032211	1 34 9 33 32 2 051550 032211
31 14 23 7 11 25 523114 014040	31 11 22 14 12 21 513213 043152
6 18 26 16 24 21 024233 551352	6 20 29 13 18 25 034224 514050
30 13 20 12 19 17 423132 501504	30 15 23 7 19 17 423132 524004
8 27 29 15 22 10 144231 124233	8 26 24 16 27 10 143241 115323
35 5 4 28 3 36 500405 443325	35 5 4 28 3 36 500405 443325
32/ /D6i	33/ /D6i
1 34 9 33 32 2 051550 032211	1 34 9 33 32 2 051550 032211
31 16 23 7 15 19 523123 034020	31 12 19 7 15 27 513124 050022
6 24 20 12 22 27 033134 551532	6 22 26 14 23 20 034233 531141
30 11 26 13 14 17 414222 541014	30 17 24 11 13 16 423122 545403
8 21 29 18 25 10 134241 124503	8 21 29 18 25 10 134241 124503
35 5 4 28 3 36 500405 443325	35 5 4 28 3 36 500405 443325
34/ /D6i	38/ /D6i
1 34 9 33 32 2 051550 032211	1 34 9 33 32 2 051550 032211
31 11 22 7 19 21 513133 043002	31 12 19 7 15 27 513124 050022
6 25 24 12 17 27 043124 505542	6 22 26 14 20 23 034233 531114
30 16 23 13 14 15 423222 534012	30 17 24 11 16 13 423122 545430
8 20 29 18 26 10 134241 114513	8 21 29 18 25 10 134241 124503
35 5 4 28 3 36 500405 443325	35 5 4 28 3 36 500405 443325
39/ /D6i	40/ /D6i
1 34 9 33 32 2 051550 032211	1 34 9 33 32 2 051550 032211
31 16 25 7 20 12 524131 030015	31 15 21 14 12 18 523212 022155
6 18 27 14 22 24 024233 552135	6 20 29 13 24 19 034233 514050
30 15 17 10 13 26 422124 524301	30 10 22 7 17 25 413124 533040
8 23 29 19 21 11 134331 144024	8 27 26 16 23 11 144231 121344
35 5 4 28 3 36 500405 443325	35 5 4 28 3 36 500405 443325

.....

ざっと調べたところ、「オイラー型」は見つからない。どうやっても、「ノン・オイラー型」しか出て来ない。どうも全部がノン・オイラー型らしい。

だが、オイラー型は本当に無いのだろうか。本当に無いなら、その証明法を考えてやらなければならない。しかし、実はあるのではあるまいか。本当はあるのに、見つけ方が悪いので、見つからないのではあるまいか？ あれこれ想像しては悩まされ続けた。

そうして、ある実験をすることを思い付いた。存在すると仮定して、『新オイラー法』で実際に「オイラー六方陣」を作ってみるのである。何しろこの方法は、実行速度が速い。もしかしたら「オイラー方陣」をたちまちに見つけ出してくれるのではあるまいか。

5. オイラー方陣を「新オイラー法」で作る

早速にも書き下ろしたのが、次のプログラムである。条件は「普通の魔方陣」である。最初に諸条件を満たす「ラテン方陣」を設計して構成する。任意に2面を選んで、それら

を分解図の上位 / 下位に見立てて組み合わせ , 解を計算によって合成するのである。

```
/** Composing 'Euler Squares' of Order 6 **/  
/** by "New Euler's Method" with /D6i **/  
/** 'EulerMS6.c' built by Kanji Setsuda **/  
/** on Nov. 5, 2005; Apr.11, 2006 **/  
/** Working on MacOSX & Xcode 2.1 **/  
/**/  
#include <stdio.h>  
/**/  
long int cnt, bcnt;  
long cntr[4], cfc[37];  
short hcnt, lcnt, cnt2, cnt3;  
short u1,u2;  
short nm[37], uflg[37];  
short anm[3][39];  
char thlu[7681][37], tllu[92161][37];  
char mtc[7681][92161];  
/**/  
short rw[7][7], cl[7][7];  
short pd[7], pb[7];  
/**/  
void stp01(void), stp02(void), stp03(void), stp04(void), stp05(void);  
void stp06(void), stp07(void), stp08(void), stp09(void), stp10(void);  
void stp11(void), stp12(void), stp13(void), stp14(void), stp15(void);  
void stp16(void), stp17(void), stp18(void), stp19(void), stp20(void);  
void stp21(void), stp22(void), stp23(void), stp24(void), stp25(void);  
void stp26(void), stp27(void), stp28(void), stp29(void), stp30(void);  
void stp31(void), stp32(void), stp33(void), stp34(void), stp35(void);  
void stp36(void), ansrecord(void);  
void prlunit(void);  
void cmbcmp(void);  
void prans(void), pr3ans(void), pr2ans(void), pr1ans(void);  
void cmissn(void);  
/**/  
int main(){  
    short m,n;  
    printf("\n** Composing 'Euler Squares' of Order 6 **\n");  
    printf("*** by 'New Euler's Method' with /D6i ***\n");  
    for(m=0;m<7;m++){  
        for(n=0;n<7;n++){  
            rw[m][n]=0; cl[m][n]=0;  
        }  
    }  
    for(n=0;n<7;n++){pd[n]=0; pb[n]=0;}  
    for(n=0;n<37;n++){nm[n]=0;}  
    hcnt=0; cnt=0; bcnt=0; cnt3=0;  
    stp01(); /* Make the Latin Squares */  
    printf("\n [List of Latin Squares]\n");  
    prlunit(); /* Print the Latin Squares */  
    lcnt=cnt; cnt=0; cnt3=0;  
    printf("\n [Compositions of 'Euler' MS66]\n");  
    //printf("\n [Examples of Wrong Compositions of 'Euler' MS66]\n");  
    cmbcmp(); /* Combine, Compose and Print */  
    if(cnt3==1){pr1ans();} /* Print the Rest One */  
    printf(" [Count = %d]\n",cnt);  
    cmissn();  
    printf(" [OK!]\n");  
    return 0;  
}
```

```

/* Make the Latin Squares */
/* Line #1 & #6 */
/* Set n1 */
void stp01(){
  short i;
  for(i=0;i<6;i++){
    if((rw[1][i]==0)&&(cl[1][i]==0)&&(pd[i]==0)){nm[1]=i;
      rw[1][i]=1; cl[1][i]=1; pd[i]=1;
      stp02();
      rw[1][i]=0; cl[1][i]=0; pd[i]=0;}
  }
}
/**/
/* Set n36 */
void stp02(){
  short i;
  for(i=5;i>=0;i--){
    if((rw[6][i]==0)&&(cl[6][i]==0)&&(pd[i]==0)){nm[36]=i;
      rw[6][i]=1; cl[6][i]=1; pd[i]=1;
      stp03();
      rw[6][i]=0; cl[6][i]=0; pd[i]=0;}
  }
}
/**/
/* Set n2 */
void stp03(){
  short i;
  for(i=0;i<6;i++){
    if((rw[1][i]==0)&&(cl[2][i]==0)){nm[2]=i;
      rw[1][i]=1; cl[2][i]=1;
      stp04();
      rw[1][i]=0; cl[2][i]=0;}
  }
}
/**/
/* Set n35 */
void stp04(){
  short i;
  for(i=5;i>=0;i--){
    if((rw[6][i]==0)&&(cl[5][i]==0)){nm[35]=i;
      rw[6][i]=1; cl[5][i]=1;
      stp05();
      rw[6][i]=0; cl[5][i]=0;}
  }
}
/**/
/* Set n3 */
void stp05(){
  short i;
  for(i=0;i<6;i++){
    if((rw[1][i]==0)&&(cl[3][i]==0)){nm[3]=i;
      rw[1][i]=1; cl[3][i]=1;
      stp06();
      rw[1][i]=0; cl[3][i]=0;}
  }
}
/**/
/* Set n34 */
void stp06(){

```

```

short i;
for(i=5;i>=0;i--){
    if((rw[6][i]==0)&&(cl[4][i]==0)){nm[34]=i;
        rw[6][i]=1; cl[4][i]=1;
        stp07();
        rw[6][i]=0; cl[4][i]=0;}
}
}
/**/
/* Set n4 */
void stp07(){
short i;
for(i=0;i<6;i++){
    if((rw[1][i]==0)&&(cl[4][i]==0)){nm[4]=i;
        rw[1][i]=1; cl[4][i]=1;
        stp08();
        rw[1][i]=0; cl[4][i]=0;}
}
}
/**/
/* Set n33 */
void stp08(){
short i;
for(i=5;i>=0;i--){
    if((rw[6][i]==0)&&(cl[3][i]==0)){nm[33]=i;
        rw[6][i]=1; cl[3][i]=1;
        stp09();
        rw[6][i]=0; cl[3][i]=0;}
}
}
}
/**/
/* Set n5 */
void stp09(){
short i;
for(i=0;i<6;i++){
    if((rw[1][i]==0)&&(cl[5][i]==0)){nm[5]=i;
        rw[1][i]=1; cl[5][i]=1;
        stp10();
        rw[1][i]=0; cl[5][i]=0;}
}
}
}
/**/
/* Set n32 */
void stp10(){
short i;
for(i=5;i>=0;i--){
    if((rw[6][i]==0)&&(cl[2][i]==0)){nm[32]=i;
        rw[6][i]=1; cl[2][i]=1;
        stp11();
        rw[6][i]=0; cl[2][i]=0;}
}
}
}
/**/
/* Set n6 */
void stp11(){
short i;
for(i=0;i<6;i++){
    if((rw[1][i]==0)&&(cl[6][i]==0)&&(pb[i]==0)){nm[6]=i;
        rw[1][i]=1; cl[6][i]=1; pb[i]=1;

```



```

        stp12();
        rw[1][i]=0; cl[6][i]=0; pb[i]=0;}
    }
}
/**/
/* Set n31 */
void stp12(){
    short i;
    for(i=5;i>=0;i--){
        if((rw[6][i]==0)&&(cl[1][i]==0)&&(pb[i]==0)){nm[31]=i;
            rw[6][i]=1; cl[1][i]=1; pb[i]=1;
            stp13();
            rw[6][i]=0; cl[1][i]=0; pb[i]=0;}
    }
}
/**/
/* Line #2 & #5 */
/* Set n7 */
void stp13(){
    short i;
    for(i=0;i<6;i++){
        if((rw[2][i]==0)&&(cl[1][i]==0)){nm[7]=i;
            rw[2][i]=1; cl[1][i]=1;
            stp14();
            rw[2][i]=0; cl[1][i]=0;}
    }
}
/* Set n30 */
void stp14(){
    short i;
    for(i=5;i>=0;i--){
        if((rw[5][i]==0)&&(cl[6][i]==0)){nm[30]=i;
            rw[5][i]=1; cl[6][i]=1;
            stp15();
            rw[5][i]=0; cl[6][i]=0;}
    }
}
/**/
/* Set n8 */
void stp15(){
    short i;
    for(i=0;i<6;i++){
        if((rw[2][i]==0)&&(cl[2][i]==0)&&(pd[i]==0)){nm[8]=i;
            rw[2][i]=1; cl[2][i]=1; pd[i]=1;
            stp16();
            rw[2][i]=0; cl[2][i]=0; pd[i]=0;}
    }
}
/**/
/* Set n29 */
void stp16(){
    short i;
    for(i=5;i>=0;i--){
        if((rw[5][i]==0)&&(cl[5][i]==0)&&(pd[i]==0)){nm[29]=i;
            rw[5][i]=1; cl[5][i]=1; pd[i]=1;
            stp17();
            rw[5][i]=0; cl[5][i]=0; pd[i]=0;}
    }
}
}

```

```

/**/
/* Set n9 */
void stp17(){
  short i;
  for(i=0;i<6;i++){
    if((rw[2][i]==0)&&(cl[3][i]==0)){nm[9]=i;
      rw[2][i]=1; cl[3][i]=1;
      stp18();
      rw[2][i]=0; cl[3][i]=0;}
  }
}
/**/
/* Set n28 */
void stp18(){
  short i;
  for(i=5;i>=0;i--){
    if((rw[5][i]==0)&&(cl[4][i]==0)){nm[28]=i;
      rw[5][i]=1; cl[4][i]=1;
      stp19();
      rw[5][i]=0; cl[4][i]=0;}
  }
}
/**/
/* Set n10 */
void stp19(){
  short i;
  for(i=0;i<6;i++){
    if((rw[2][i]==0)&&(cl[4][i]==0)){nm[10]=i;
      rw[2][i]=1; cl[4][i]=1;
      stp20();
      rw[2][i]=0; cl[4][i]=0;}
  }
}
/**/
/* Set n27 */
void stp20(){
  short i;
  for(i=5;i>=0;i--){
    if((rw[5][i]==0)&&(cl[3][i]==0)){nm[27]=i;
      rw[5][i]=1; cl[3][i]=1;
      stp21();
      rw[5][i]=0; cl[3][i]=0;}
  }
}
/**/
/* Set n11 */
void stp21(){
  short i;
  for(i=0;i<6;i++){
    if((rw[2][i]==0)&&(cl[5][i]==0)&&(pb[i]==0)){nm[11]=i;
      rw[2][i]=1; cl[5][i]=1; pb[i]=1;
      stp22();
      rw[2][i]=0; cl[5][i]=0; pb[i]=0;}
  }
}
/**/
/* Set n26 */
void stp22(){
  short i;

```

```

    for(i=5;i>=0;i--){
        if((rw[5][i]==0)&&(cl[2][i]==0)&&(pb[i]==0)){nm[26]=i;
            rw[5][i]=1; cl[2][i]=1; pb[i]=1;
            stp23();
            rw[5][i]=0; cl[2][i]=0; pb[i]=0;}
    }
}
/**/
/* Set n12 */
void stp23(){
    short i;
    for(i=0;i<6;i++){
        if((rw[2][i]==0)&&(cl[6][i]==0)){nm[12]=i;
            rw[2][i]=1; cl[6][i]=1;
            stp24();
            rw[2][i]=0; cl[6][i]=0;}
    }
}
/**/
/* Set n25 */
void stp24(){
    short i;
    for(i=5;i>=0;i--){
        if((rw[5][i]==0)&&(cl[1][i]==0)){nm[25]=i;
            rw[5][i]=1; cl[1][i]=1;
            stp25();
            rw[5][i]=0; cl[1][i]=0;}
    }
}
/**/
/* Line #3 & #4 */
/* Set n13 */
void stp25(){
    short i;
    for(i=0;i<6;i++){
        if((rw[3][i]==0)&&(cl[1][i]==0)){nm[13]=i;
            rw[3][i]=1; cl[1][i]=1;
            stp26();
            rw[3][i]=0; cl[1][i]=0;}
    }
}
/**/
/* Set n24 */
void stp26(){
    short i;
    for(i=5;i>=0;i--){
        if((rw[4][i]==0)&&(cl[6][i]==0)){nm[24]=i;
            rw[4][i]=1; cl[6][i]=1;
            stp27();
            rw[4][i]=0; cl[6][i]=0;}
    }
}
/**/
/* Set n14 */
void stp27(){
    short i;
    for(i=0;i<6;i++){
        if((rw[3][i]==0)&&(cl[2][i]==0)){nm[14]=i;
            rw[3][i]=1; cl[2][i]=1;

```

```

        stp28();
        rw[3][i]=0; cl[2][i]=0;}
    }
}
/**/
/* Set n23 */
void stp28(){
    short i;
    for(i=5;i>=0;i--){
        if((rw[4][i]==0)&&(cl[5][i]==0)){nm[23]=i;
            rw[4][i]=1; cl[5][i]=1;
            stp29();
            rw[4][i]=0; cl[5][i]=0;}
    }
}
/**/
/* Set n15 */
void stp29(){
    short i;
    for(i=0;i<6;i++){
        if((rw[3][i]==0)&&(cl[3][i]==0)&&(pd[i]==0)){nm[15]=i;
            rw[3][i]=1; cl[3][i]=1; pd[i]=1;
            stp30();
            rw[3][i]=0; cl[3][i]=0; pd[i]=0;}
    }
}
/**/
/* Set n22 */
void stp30(){
    short i;
    for(i=5;i>=0;i--){
        if((rw[4][i]==0)&&(cl[4][i]==0)&&(pd[i]==0)){nm[22]=i;
            rw[4][i]=1; cl[4][i]=1; pd[i]=1;
            stp31();
            rw[4][i]=0; cl[4][i]=0; pd[i]=0;}
    }
}
/**/
/* Set n16 */
void stp31(){
    short i;
    for(i=0;i<6;i++){
        if((rw[3][i]==0)&&(cl[4][i]==0)&&(pb[i]==0)){nm[16]=i;
            rw[3][i]=1; cl[4][i]=1; pb[i]=1;
            stp32();
            rw[3][i]=0; cl[4][i]=0; pb[i]=0;}
    }
}
/**/
/* Set n21 */
void stp32(){
    short i;
    for(i=5;i>=0;i--){
        if((rw[4][i]==0)&&(cl[3][i]==0)&&(pb[i]==0)){nm[21]=i;
            rw[4][i]=1; cl[3][i]=1; pb[i]=1;
            stp33();
            rw[4][i]=0; cl[3][i]=0; pb[i]=0;}
    }
}
}

```

```

/**/
/* Set n17 */
void stp33(){
    short i;
    for(i=0;i<6;i++){
        if((rw[3][i]==0)&&(cl[5][i]==0)){nm[17]=i;
            rw[3][i]=1; cl[5][i]=1;
            stp34();
            rw[3][i]=0; cl[5][i]=0;}
    }
}
/**/
/* Set n20 */
void stp34(){
    short i;
    for(i=5;i>=0;i--){
        if((rw[4][i]==0)&&(cl[2][i]==0)){nm[20]=i;
            rw[4][i]=1; cl[2][i]=1;
            stp35();
            rw[4][i]=0; cl[2][i]=0;}
    }
}
/**/
/* Set n18 */
void stp35(){
    short i;
    for(i=0;i<6;i++){
        if((rw[3][i]==0)&&(cl[6][i]==0)){nm[18]=i;
            rw[3][i]=1; cl[6][i]=1;
            stp36();
            rw[3][i]=0; cl[6][i]=0;}
    }
}
/**/
/* Set n19 */
void stp36(){
    short i;
    for(i=5;i>=0;i--){
        if((rw[4][i]==0)&&(cl[1][i]==0)){nm[19]=i;
            rw[4][i]=1; cl[1][i]=1;
            ansrecord();
            rw[4][i]=0; cl[1][i]=0;}
    }
}
/**/
/* Record the Answers */
void ansrecord(){
    short n;
    tllu[cnt][0]=cnt+1;
    for(n=1;n<37;n++){tllu[cnt][n]=nm[n];}
    if((nm[1]==0)&&(nm[6]<nm[31])){
        thlu[hcnt][0]=hcnt+1;
        for(n=1;n<37;n++){thlu[hcnt][n]=nm[n];}
        hcnt++;
    }
    cnt++;
}
/**/
/* Classify and Print the Latin Squares */

```

```

void prlunit(){
int t,m,n;
short l,l6;
printf(" [for High]");
for(t=0;t<hcnt;t=t+768){
printf("\n%7d/%7d/%7d/%7d/%7d/%7d/%7d/%7d\n",
t+1,t+2,t+3,t+4,t+5,t+6,t+7,t+8);
for(l=0;l<6;l++){l6=l*6;
for(m=t;m<(t+8);m++){
printf(" ");
for(n=1;n<7;n++){printf("%d",thlu[m][l6+n]);}
}
printf("\n");
}
}
printf("\n [for Low]");
for(t=0;t<cnt;t=t+9216){
printf("\n%7d/%7d/%7d/%7d/%7d/%7d/%7d/%7d\n",
t+1,t+2,t+3,t+4,t+5,t+6,t+7,t+8);
for(l=0;l<6;l++){l6=l*6;
for(m=t;m<(t+8);m++){
printf(" ");
for(n=1;n<7;n++){printf("%d",tllu[m][l6+n]);}
}
printf("\n");
}
}
printf("\n [Counts of Latin Squares = %dH/%dL]\n",hcnt,cnt);
for(m=0;m<hcnt;m++){
for(n=0;n<cnt;n++){
t=0;
for(l=1;l<37;l++){if(thlu[m][l]==tllu[n][l]){t++;}}
mtc[m][n]=t;
if(mtc[m][n]==6){bcnt++;}
}
}
printf("\n [Reference Table for the Best Combinations = %d]\n",bcnt);
printf("\n T1|",t);
for(n=0;n<20;n++){printf("%3d",n+1);}
printf("\n---+-----");
for(m=0;m<20;m++){
printf("\n%3d|",m+1);
for(n=0;n<20;n++){printf("%3d",mtc[m][n]);}
}
printf("\n\n T2|",t);
for(n=20;n<40;n++){printf("%3d",n+1);}
printf("\n---+-----");
for(m=20;m<40;m++){
printf("\n%3d|",m+1);
for(n=20;n<40;n++){printf("%3d",mtc[m][n]);}
}
printf("\n");
}
/**/
/* Combine, Compose and Print */
void cmbcmp(){
short n,d,fc;
for(u1=0;u1<hcnt;u1++){cnt2=0;
for(u2=0;u2<lcnt;u2++){

```

```

    if(mtc[u1][u2]==6){
        for(n=1;n<37;n++){uflg[n]=0;}
        for(n=1;n<37;n++){
            d=thlu[u1][n]*6+tllu[u2][n]+1;
            nm[n]=d; uflg[d]++;
        }
        fc=0;
        for(n=1;n<37;n++){if(uflg[n]==0){fc++;}}
        cfc[fc]++;
        if(fc==0){//if(fc<=6){
            if((nm[1]==1)&&(nm[6]<nm[31])){prans();}
        }
    }
}
}
}
}
/**/
/* Print the Solution List */
void prans(){
    short n;
    cnt++; cnt2++;
    if(cnt2==1){
        cntr[cnt3]=cnt; cntr[cnt3+1]=0;
        for(n=1;n<37;n++){anm[cnt3][n]=nm[n]; anm[cnt3+1][n]=0;}
        anm[cnt3][37]=u1+1; anm[cnt3][38]=u2+1;
        cnt3++;
        if(cnt3==2){pr2ans(); cnt3=0;}
    }
}
/**/
/* Print 2 Answers */
void pr2ans(){
    short lu1,lu2,l,l6,m,n;
    printf("%6d/%6d/%18d#%8d/%6d/%18d#\n",
        anm[0][37],anm[0][38],cntr[0],anm[1][37],anm[1][38],cntr[1]);
    for(l=0;l<6;l++){l6=l*6;
        for(m=0;m<2;m++){
            lu1=anm[m][37]-1; lu2=anm[m][38]-1;
            printf(" ");
            for(n=1;n<7;n++){printf("%d",thlu[lu1][l6+n]);}
            printf(" ");
            for(n=1;n<7;n++){printf("%d",tllu[lu2][l6+n]);}
            printf(" ");
            for(n=1;n<7;n++){printf("%3d",anm[m][l6+n]);}
            if(m<1){printf(" ");}
        }
        printf("\n");
    }
}
}
/**/
/* Print the Rest One */
void pr1ans(){
    short lu1,lu2,l,l6,n;
    lu1=anm[0][37]; lu2=anm[0][38];
    printf("%6d/%6d/ /N6i%31d#\n", lu1, lu2, cntr[0]);
    for(l=0;l<6;l++){l6=l*6;
        printf(" ");
        for(n=1;n<7;n++){printf("%d",thlu[lu1-1][l6+n]);}
        printf(" ");
    }
}

```

```

        for(n=1;n<7;n++){printf("%d",tllu[lu2-1][l6+n]);}
        printf(" ");
        for(n=1;n<7;n++){printf(" %d%d",thlu[lu1-1][l6+n],tllu[lu2-1][l6+n]);}
        printf(" ");
        for(n=1;n<7;n++){printf("%3d",anm[0][l6+n]);}
        printf("\n");
    }
}
/**/
void cmissn(){
    short n;
    printf("\n [Count of Missing Numbers]");
    for(n=0;n<36;n++){
        if(n%6==0){printf("\n");}
        printf("%2d:%7d,",36-n,cfc[36-n]);
    }
    printf("\n [No Missing: %d]\n",cfc[0]);
}
/**/

```

テストしてみて、まず多数のラテン方陣が出来ることに驚かされた。

**\*\* オイラー型の六方陣を「新オイラー法」で作る \*\***

[ラテン方陣のリスト]

[上位面用]

1/	2/	3/	4/	5/	6/	7/	8/
012453	012453	012453	012453	012453	012453	012453	012453
340521	340521	340512	340512	540321	540321	540312	540312
253140	253104	153240	153204	253140	253104	153240	153204
135204	135240	235104	235140	135204	135240	235104	235140
504312	504312	504321	504321	304512	304512	304521	304521
421035	421035	421035	421035	421035	421035	421035	421035
769/	770/	771/	772/	773/	774/	775/	776/
041352	041352	041352	041352	041352	041352	041352	041352
124503	324501	532014	532014	124503	324501	530214	530214
253140	253140	324501	124503	253140	253140	324501	124503
305421	105423	253140	253140	305421	105423	253140	253140
532014	532014	105423	305421	530214	530214	105423	305421
410235	410235	410235	410235	412035	412035	412035	412035
1537/	1538/	1539/	1540/	1541/	1542/	1543/	1544/
012543	012543	012543	012543	012543	012543	012543	012543
350421	350421	350412	350412	450321	450321	450312	450312
243150	243105	143250	143205	243150	243105	143250	143205
134205	134250	234105	234150	134205	134250	234105	234150
405312	405312	405321	405321	305412	305412	305421	305421
521034	521034	521034	521034	521034	521034	521034	521034
2305/	2306/	2307/	2308/	2309/	2310/	2311/	2312/
041532	041532	041532	041532	041532	041532	041532	041532
123405	452013	452013	523401	123405	450213	450213	523401
235140	523401	123405	235140	235140	523401	123405	235140
504321	235140	235140	104325	504321	235140	235140	104325
452013	104325	504321	452013	450213	104325	504321	450213
310254	310254	310254	310254	312054	312054	312054	312054



3073/	3074/	3075/	3076/	3077/	3078/	3079/	3080/
012534	012534	012534	012534	012534	012534	012534	012534
350421	350421	350412	350412	450321	450321	450312	450312
234150	234105	134250	134205	234150	234105	134250	134205
143205	143250	243105	243150	143205	143250	243105	243150
405312	405312	405321	405321	305412	305412	305421	305421
521043	521043	521043	521043	521043	521043	521043	521043

3841/	3842/	3843/	3844/	3845/	3846/	3847/	3848/
031542	031542	031542	031542	031524	031524	031524	031524
345201	345201	315204	315204	325401	325401	315402	315402
152430	102435	452130	402135	154230	104235	254130	204135
403125	453120	103425	153420	203145	253140	103245	153240
210354	210354	240351	240351	410352	410352	420351	420351
524013	524013	524013	524013	542013	542013	542013	542013

4609/	4610/	4611/	4612/	4613/	4614/	4615/	4616/
012534	012534	012534	012534	012534	012534	012534	012534
154023	354021	435210	435210	150423	350421	435210	435210
243105	243105	354021	154023	324051	124053	241305	241305
320451	120453	243105	243105	241305	241305	350421	150423
435210	435210	120453	320451	435210	435210	124053	324051
501342	501342	501342	501342	503142	503142	503142	503142

5377/	5378/	5379/	5380/	5381/	5382/	5383/	5384/
032451	032451	032451	032451	032541	032541	032541	032541
415230	415230	410235	410235	145230	145230	350124	450123
153024	253014	153024	253014	203415	203415	421053	321054
204513	104523	204513	104523	450123	350124	203415	203415
320145	320145	325140	325140	321054	421053	145230	145230
541302	541302	541302	541302	514302	514302	514302	514302

6145/	6146/	6147/	6148/	6149/	6150/	6151/	6152/
012543	012543	012543	012543	012543	012543	012543	012543
153024	153024	234105	534102	150324	150324	234105	534102
534102	234105	345210	345210	534102	234105	345210	345210
345210	345210	501432	201435	345210	345210	501432	201435
201435	501432	153024	153024	201435	501432	150324	150324
420351	420351	420351	420351	423051	423051	423051	423051

6913/	6914/	6915/	6916/	6917/	6918/	6919/	6920/
031524	031524	031524	031524	031542	031542	031542	031542
250413	350412	425130	425130	324150	324150	320154	320154
314052	214053	103245	103245	145023	245013	145023	245013
103245	103245	350412	250413	203415	103425	203415	103425
425130	425130	214053	314052	510234	510234	514230	514230
542301	542301	542301	542301	452301	452301	452301	452301

[Counts of Latin Squares = 7680H]

[下位面用]

1/	2/	3/	4/	5/	6/	7/	8/
012453	012453	012453	012453	012534	012534	012534	012534
231504	431502	543021	543021	124053	324051	435210	435210
354210	354210	431502	231504	543102	543102	324051	124053
405132	205134	354210	354210	350421	150423	543102	543102
543021	543021	205134	405132	435210	435210	150423	350421
120345	120345	120345	120345	201345	201345	201345	201345

9217/	9218/	9219/	9220/	9221/	9222/	9223/	9224/
012345	012345	012345	012345	012543	012543	012543	012543
135024	435021	543210	543210	451320	451320	450321	450321
254103	254103	435021	135024	234015	324015	234015	324015
420531	120534	254103	254103	325104	235104	325104	235104
543210	543210	120534	420531	540231	540231	541230	541230
301452	301452	301452	301452	103452	103452	103452	103452

18433/	18434/	18435/	18436/	18437/	18438/	18439/	18440/
102435	102435	102435	102435	102543	102543	102543	102543
025143	325140	534201	534201	230415	453120	453120	530412
453012	453012	325140	025143	345201	530412	230415	345201
341520	041523	453012	453012	514032	345201	345201	214035
534201	534201	041523	341520	453120	214035	514032	453120
210354	210354	210354	210354	021354	021354	021354	021354

27649/	27650/	27651/	27652/	27653/	27654/	27655/	27656/
102534	102534	102534	102534	102543	102543	102543	102543
054123	054123	243015	543012	053214	053214	043215	043215
543012	243015	435201	435201	524031	514032	425031	415032
435201	435201	510342	210345	415302	425301	514302	524301
210345	510342	054123	054123	340125	340125	350124	350124
321450	321450	321450	321450	231450	231450	231450	231450

36865/	36866/	36867/	36868/	36869/	36870/	36871/	36872/
201345	201345	201345	201345	201534	201534	201534	201534
015234	415230	543102	543102	140325	354210	354210	540321
354021	354021	415230	015234	435102	540321	140325	435102
432510	032514	354021	354021	523041	435102	435102	123045
543102	543102	032514	432510	354210	123045	523041	354210
120453	120453	120453	120453	012453	012453	012453	012453

46081/	46082/	46083/	46084/	46085/	46086/	46087/	46088/
301254	301254	301254	301254	301254	301254	301254	301254
423510	423510	423501	423501	523410	523410	523401	523401
154032	154023	054132	054123	154032	154023	054132	054123
045123	045132	145023	145032	045123	045132	145023	145032
532401	532401	532410	532410	432501	432501	432510	432510
210345	210345	210345	210345	210345	210345	210345	210345

55297/	55298/	55299/	55300/	55301/	55302/	55303/	55304/
301245	301245	301245	301245	301542	301542	301542	301542
025314	425310	542103	542103	453210	453210	450213	450213
154032	154032	425310	025314	124305	214305	124305	214305
413520	013524	154032	154032	215034	125034	215034	125034
542103	542103	013524	413520	540123	540123	543120	543120
230451	230451	230451	230451	032451	032451	032451	032451

64513/	64514/	64515/	64516/	64517/	64518/	64519/	64520/
401235	401235	401235	401235	401235	401235	401235	401235
324510	324510	324501	324501	524310	524310	524301	524301
135042	135024	035142	035124	135042	135024	035142	035124
053124	053142	153024	153042	053124	053142	153024	153042
542301	542301	542310	542310	342501	342501	342510	342510
210453	210453	210453	210453	210453	210453	210453	210453

73729/	73730/	73731/	73732/	73733/	73734/	73735/	73736/
401523	401523	401523	401523	401532	401532	401532	401532
054312	054312	132045	532041	054213	054213	123045	523041

532041 132045 325104 325104 523041 123045 235104 235104  
 325104 325104 540231 140235 235104 235104 540321 140325  
 140235 540231 054312 054312 140325 540321 054213 054213  
 213450 213450 213450 213450 312450 312450 312450 312450

82945/ 82946/ 82947/ 82948/ 82949/ 82950/ 82951/ 82952/  
 501234 501234 501234 501234 501234 501234 501234 501234  
 042513 042513 042315 042315 142503 142503 142305 142305  
 253041 453021 253041 453021 253140 453120 253140 453120  
 435120 235140 435120 235140 435021 235041 435021 235041  
 124305 124305 124503 124503 024315 024315 024513 024513  
 310452 310452 310452 310452 310452 310452 310452 310452

[Counts of Latin Squares = 92160L]

[良い組み合わせを得るための参照表 = 13561920]

T1: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

1	13	14	13	15	8	11	6	6	13	11	11	9	11	9	9	7	8	10	5	6
2	12	12	15	18	9	12	5	5	11	13	9	11	9	11	7	9	7	9	6	7
3	17	20	10	10	6	7	9	11	12	10	12	10	10	8	10	8	7	9	7	8
4	16	18	12	13	7	8	8	10	10	12	10	12	8	10	8	10	6	8	8	9
5	11	12	14	16	8	10	6	7	12	10	10	8	12	10	10	8	8	9	5	7
6	10	10	16	19	9	11	5	6	10	12	8	10	10	12	8	10	7	8	6	8
7	15	18	11	11	6	6	9	12	11	9	11	9	11	9	11	9	7	8	7	9
8	14	16	13	14	7	7	8	11	9	11	9	11	9	11	9	11	6	7	8	10
9	4	6	11	12	18	16	7	8	6	7	5	6	7	8	6	7	11	10	9	9
10	6	4	12	13	16	18	8	9	8	8	7	7	9	9	8	8	11	12	8	10
11	5	6	4	6	9	10	18	16	8	7	8	7	7	6	7	6	11	11	11	10
12	6	7	6	4	10	11	16	18	9	9	9	9	8	8	8	8	10	12	11	12
13	10	12	15	16	6	8	6	7	18	16	14	12	16	14	12	10	8	11	5	5
14	9	10	17	19	7	9	5	6	16	18	12	14	14	16	10	12	7	10	6	6
15	12	14	14	15	5	7	8	9	14	12	18	16	12	10	16	14	6	7	8	10
16	11	12	16	18	6	8	7	8	12	14	16	18	10	12	14	16	5	6	9	11
17	8	10	18	19	7	8	5	7	16	14	12	10	18	16	14	12	9	11	4	5
18	7	8	20	22	8	9	4	6	14	16	10	12	16	18	12	14	8	10	5	6
19	10	12	17	18	6	7	7	9	12	10	16	14	14	12	18	16	7	7	7	10
20	9	10	19	21	7	8	6	8	10	12	14	16	12	14	16	18	6	6	8	11

T2: 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40

21	10	9	8	7	9	8	7	6	5	7	9	10	5	6	6	5	13	11	13	11
22	10	9	8	7	9	8	7	6	6	6	11	10	4	5	7	6	11	13	11	13
23	6	7	8	9	7	8	9	10	5	6	7	9	6	5	7	8	11	9	15	13
24	6	7	8	9	7	8	9	10	7	6	8	8	7	6	6	7	9	11	13	15
25	6	7	6	7	7	8	7	8	5	7	9	10	5	5	12	12	22	18	8	8
26	8	9	4	5	9	10	5	6	7	5	10	11	5	5	12	12	18	22	8	8
27	8	7	8	7	7	6	7	6	9	10	5	7	10	10	5	5	10	10	22	18
28	6	5	10	9	5	4	9	8	10	11	7	5	10	10	5	5	10	10	18	22
29	9	8	7	6	10	9	8	7	6	8	9	10	6	7	6	5	11	9	11	9
30	9	8	7	6	10	9	8	7	7	7	11	10	5	6	7	6	9	11	9	11
31	7	8	9	10	6	7	8	9	7	8	8	10	8	7	8	9	11	9	13	11
32	7	8	9	10	6	7	8	9	9	8	9	9	9	8	7	8	9	11	11	13
33	10	13	7	10	10	13	7	10	10	8	8	7	9	7	12	11	24	22	9	10
34	7	5	9	7	7	5	9	7	6	5	10	8	8	7	9	7	9	10	24	22
35	7	5	11	9	8	6	12	10	7	6	8	10	9	8	7	9	10	11	22	24
36	13	16	8	11	12	15	7	10	8	10	9	8	7	9	13	12	22	24	10	11
37	16	12	14	10	15	11	13	9	12	10	11	10	14	12	15	14	10	12	9	10
38	12	16	10	14	11	15	9	13	12	10	13	12	12	11	18	16	13	15	7	8
39	14	10	14	10	13	9	13	9	12	10	12	11	14	12	16	15	9	11	9	10
40	10	14	10	14	9	13	9	13	12	10	14	13	12	11	19	17	12	14	7	8

.....

\* 合成した解のリストと総数 \*

[カウント: 0]

合成した結果は、悩ましいものだった。多数のラテン方陣が全て「オイラー型」なのに、その内の2面を任意に組み合わせで計算・合成してみても、「ギリシャ・ラテン六方陣」(別名『オイラー方陣』)が1つも出来なかったからである。そんな結果になり得るとは、最初全然想像もしなかった。

プログラムのミスの子かと思つて何度も見直したが、どうもそうではないらしい。事実がそうになっているらしい。わざわざ「誤答」を強制出力したりして調べた結果、7680×92160の全ての場合で、合成解は次の「オイラー条件」に抵触・違反することがわかつた。

・上位面と下位面の対応する数字の組み合わせに、重複や欠落が無いこと。すなわち六進法で見て、{00, 01, 02, 03, 04, 05, 10, 11, 12, ..., 53, 54, 55(N6i)}の全数が各1回使われていること。この条件は、古典的表記における「自然数列 {1, 2, 3, 4, 5, 6, 7, 8, ..., 34, 35, 36} の全数字を各1回ずつ重複無く欠落無く使うこと」という魔方陣の基本約束の1つに論理的に対応している。これを満たさずには、魔方陣は作れない。

### 6. 結果の分析

繰り返し使う数があると、当然未使用の数が出る(全体で36個しか使わないから)。だから、重複と欠落は、表裏の関係にある。そこで、使っていない数を数えたら、次のような表を得た。未使用の数は0個が条件だが、そういう合成解は1個も無かつた。

最低でも、6個の数が失われている。最高で30個。つまり、6個の数だけを繰り返し何度も使用している合成解があるということになる。ゆえに「誤答」が生じるのである。

[Count of Missing Numbers]

36: 0,	35: 0,	34: 0,	33: 0,	32: 0,	31: 0,
30: 585568,	29: 0,	28: 628176,	27: 0,	26: 725728,	25: 0,
24: 232656,	23: 0,	22: 0,	21: 120954,	20: 112112,	19: 0,
18: 1062664,	17: 1028688,	16: 1424014,	15: 1912124,	14: 5295216,	13: 1384918,
12: 5973378,	11: 1640468,	10: 7582804,	9: 988896,	8: 0,	7: 0,
6: 1236860,	5: 0,	4: 0,	3: 0,	2: 0,	1: 0,

[No Missing: 0]

合成した誤答の実例を見ていただく。オイラー条件 によってフルイに掛ける前の解である。同じ数の繰り返し使用と未使用の数字の比較的小さい例を、リストにしてみた。

プログラムで言うと、void cmbcmp(){ という手続きの中の if(fc==0){ // if(fc<=6){ とあるところを変えて出力した。

[誤つた解を無理矢理出力させたリスト: 研究用]

1/	51/	1#	2/	82/	199#
012453 013524	1 8 16 30 33	23	012453 014532	1 8 17 30	34 21
340521 425310	23 27 6 34 14	7	340521 342150	22 29 3 32 18	7
253140 541203	18 35 20 9 25	4	253104 521403	18 33 20 11 1	28
135204 230451	9 22 31 17 6	26	135240 250314	9 24 31 16 26	5
504312 154032	32 6 29 19 10	15	504312 435021	35 4 30 19 9	14
421035 302145	28 13 9 2 23	36	421035 103245	26 13 10 3 23	36
3/	2314/	259#	4/	506/	315#
012453 051432	1 12 14 29 34	21	012453 015234	1 8 18 27 34	23
340512 542103	24 29 3 32 7	16	340512 143520	20 29 4 36 9	13
153240 403521	11 31 22 18 27	2	153204 532041	12 34 21 13 5	26
235104 314250	16 20 35 9 6	25	235140 401352	17 19 32 10 30	3
504321 235014	33 4 30 19 14	11	504321 250413	33 6 25 23 14	10
421035 120345	26 15 7 4 23	36	421035 324105	28 15 11 2 19	36

5/ 511/	377#	6/ 2311/	439#
012453 015234	1 8 18 27 34 23	012453 051432	1 12 14 29 34 21
540321 240513	33 29 1 24 14 10	540321 235104	33 28 6 20 13 11
253140 431052	17 34 20 7 30 3	253104 314520	16 32 23 12 3 25
135204 502341	12 19 33 16 5 26	135240 403251	11 19 34 15 30 2
304512 153420	20 6 28 35 9 13	304512 542013	24 5 27 31 8 16
421035 324105	28 15 11 2 19 36	421035 120345	26 15 7 4 23 36
7/ 87/	493#	8/ 172/	557#
012453 014532	1 8 17 30 34 21	012453 013524	1 8 16 30 33 23
540312 432051	35 28 3 19 12 14	540312 324051	34 27 5 19 12 14
153240 251304	9 36 20 16 25 5	153204 541203	12 35 20 15 1 28
235104 520413	18 21 31 11 2 28	235140 450312	17 24 31 10 26 3
304521 345120	22 5 30 32 15 7	304521 235140	21 4 30 32 17 7
421035 103245	26 13 10 3 23 36	421035 102435	26 13 9 5 22 36
9/ 95/	753#	10/ 77/	815#
014523 014352	1 8 29 34 18 21	014523 014352	1 8 29 34 18 21
143052 540231	12 29 19 3 34 14	243051 325104	16 27 24 2 31 11
532104 253410	33 24 16 11 2 25	532104 401523	35 19 14 12 3 28
250341 425103	17 33 6 20 25 10	150342 253410	9 36 4 23 26 13
325410 301524	22 13 32 30 9 5	325410 542031	24 17 33 25 10 2
401235 132045	26 4 9 13 23 36	401235 130245	26 4 7 15 23 36
11/ 161/	879#	12/ 655/	941#
014523 013254	1 8 28 33 18 23	014523 024351	1 9 29 34 18 20
325410 245103	21 17 36 26 7 4	325410 541032	24 17 32 25 10 3
243051 301542	16 25 20 6 35 9	143052 153420	8 30 22 5 33 13
532104 452310	35 24 15 10 2 25	532104 415203	35 20 18 9 1 28
150342 534021	12 34 5 19 27 14	250341 302514	16 31 3 24 26 11
401235 120435	26 3 7 17 22 36	401235 230145	27 4 7 14 23 36
13/ 101/	1001#	14/ 61/	1061#
014253 015432	1 8 30 17 34 21	014253 013452	1 8 28 17 36 21
345021 341250	22 29 32 3 18 7	345021 321504	22 27 32 6 13 11
132540 253014	9 24 16 31 26 5	132504 254310	9 24 17 34 2 25
251304 524103	18 33 11 20 1 28	251340 405123	17 31 12 20 27 4
503412 430521	35 4 19 30 9 14	503412 540231	36 5 19 27 10 14
420135 102345	26 13 3 10 23 36	420135 132045	26 16 3 7 23 36
15/ 385/	1259#	16/ 2433/	1321#
014253 012534	1 8 27 18 34 23	014253 054132	1 12 29 14 34 21
345012 145320	20 29 36 4 9 13	345012 541203	24 29 32 3 7 16
231540 403152	17 19 10 32 30 3	231504 312450	16 20 9 35 6 25
152304 530241	12 34 13 21 5 26	152340 405321	11 31 18 22 27 2
503421 254013	33 6 23 25 14 10	503421 230514	33 4 19 30 14 11
420135 321405	28 15 2 11 19 36	420135 123045	26 15 4 7 23 36
17/ 2432/	1377#	18/ 392/	1431#
014253 054132	1 12 29 14 34 21	014253 012534	1 8 27 18 34 23
543021 231504	33 28 20 6 13 11	543021 245013	33 29 24 1 14 10
132540 402351	11 19 15 34 30 2	132504 503241	12 19 16 33 5 26
251304 315420	16 32 12 23 3 25	251340 430152	17 34 7 20 30 3
305412 540213	24 5 31 27 8 16	305412 154320	20 6 35 28 9 13
420135 123045	26 15 4 7 23 36	420135 321405	28 15 2 11 19 36
19/ 243/	1493#	20/ 107/	1689#
014253 015324	1 8 30 16 33 23	014253 015432	1 8 30 17 34 21
543012 320451	34 27 19 5 12 14	543012 430251	35 28 19 3 12 14
231540 453012	17 24 10 31 26 3	231504 254013	15 24 11 31 2 28
152304 542103	12 35 15 20 1 28	152340 523104	12 33 16 20 25 5
305421 231540	21 4 32 30 17 7	305421 341520	22 5 32 30 15 7
420135 104235	26 13 5 9 22 36	420135 102345	26 13 3 10 23 36

• • • • •

1001/	513/		94001#	1002/	1/		94055#
043521	015243	1 26 24 33 17 10		043521	012453	1 26 21 35 18 10	
125340	120354	8 15 31 22 30 5		125340	231504	9 16 32 24 25 5	
514203	453021	35 12 28 13 3 20		514203	354210	34 12 29 15 2 19	
430152	501432	30 19 2 11 34 15		230154	405132	17 19 6 8 34 27	
251034	342510	16 35 9 6 20 25		451032	543021	30 35 10 1 21 14	
302415	234105	21 4 17 26 7 36		302415	120345	20 3 13 28 11 36	
1003/	514/		94251#	1004/	2/		94307#
043521	015243	1 26 24 33 17 10		043521	012453	1 26 21 35 18 10	
230154	342510	16 23 3 12 32 25		430152	431502	29 22 2 12 31 15	
451032	501432	30 31 8 5 22 15		251034	354210	16 36 11 3 20 25	
514203	420351	35 9 25 16 6 20		514203	205134	33 7 30 14 4 23	
125340	153024	8 18 34 19 27 5		125340	543021	12 17 34 19 27 2	
302415	234105	21 4 17 26 7 36		302415	120345	20 3 13 28 11 36	
1005/	17/		94505#	1006/	19/		94567#
043152	012534	1 26 21 12 34 17		043152	012534	1 26 21 12 34 17	
124503	120453	8 15 25 35 6 22		324501	435210	23 16 30 33 2 7	
251340	354021	16 36 11 19 27 2		251340	541302	18 35 8 22 25 3	
305421	541302	24 5 32 28 13 9		105423	320451	10 3 31 29 18 20	
512034	435210	35 10 18 3 20 25		512034	154023	32 12 17 1 21 28	
430215	203145	27 19 4 14 11 36		430215	203145	27 19 4 14 11 36	
1007/	19/		94627#	1008/	17/		94689#
043152	012534	1 26 21 12 34 17		043152	012534	1 26 21 12 34 17	
512034	435210	35 10 18 3 20 25		512034	120453	32 9 13 5 24 28	
324501	541302	24 17 26 34 1 9		124503	354021	10 18 29 31 3 20	
251340	320451	16 33 7 23 30 2		251340	541302	18 35 8 22 25 3	
105423	154023	8 6 35 25 15 22		305421	435210	23 4 36 27 14 7	
430215	203145	27 19 4 14 11 36		430215	203145	27 19 4 14 11 36	
1009/	67/		94753#	1010/	65/		94813#
043521	014253	1 26 23 33 18 10		043521	014253	1 26 23 33 18 10	
125340	543021	12 17 34 19 27 2		125340	235104	9 16 36 20 25 5	
431052	352410	28 24 9 5 32 13		231054	401532	17 19 8 6 34 27	
512403	435102	35 10 18 26 1 21		512403	352410	34 12 15 29 2 19	
250134	201534	15 31 2 12 22 29		450132	543021	30 35 4 7 21 14	
304215	120345	20 3 25 16 11 36		304215	120345	20 3 25 16 11 36	

.....

## 7. 結果の解釈

かくして、普通の六方陣には「オイラー型」が皆無であることがわかった。それは、すなわち普通の六方陣はすべてが「ノン・オイラー型」であることを意味する。

予想外の結論になったが、これをどう考えたら良いのだろう。

とにかく、これは一大事である。

なぜなら、我々は「ノン・オイラー型」という存在に六次で初めて出逢うからである。四次・五次では、オイラー型しか存在しなかった。七次・八次・九次では、オイラー型が少数とノン・オイラー型が無数にあって、混在する。その逆転現象の起点が六次にあるということになる。しかも、六次にはノン・オイラー型しか存在しない。

いったい何故なのか。その理由も含めて、更なる研究の必要を痛感する。

1つ山を越すと、その向こうにまた大きな山が見えて来る。魔方陣研究には、終着点とかゴールてえものがおよそ無いらしい。また眠れぬ夜が続く（その分昼寝はするけれども ... と、落語にある通りだが）。真に頭の痛い話である。

(Written on Nov.7, 2005; Revised on Apr.11, 2006; Working on MacOSX and Xcode 2.1)

Kanji Setsuda: E-Mail Address: <jag12001@nifty.ne.jp>