

第4部 : 『魔方陣の最新研究』 : 摂田 寛二
 第4章 : 解説『魔方陣研究の諸技法』II
 第8-3節 : 四進法で各種四方陣の全数を作る

0. 今節では、補数対称型四方陣 48 個・汎対角線型四方陣 48 個・標準型四方陣 880 個の全数をいかにして四進法で作るかというテーマで書く。前節の補完が目的だ。

一部に前節の訂正が含まれる。最重要ポイントは、ラテン方陣の放棄だ。

読者から寄せられる質問で多いのは、四進法で四方陣全数を作れないか。二進法でなどと言わず、N方陣はN進法で解くという一番自然な方法で作れないか、というものだ。

今までは「それは不可能です」と答えるのを常としていたのだが、それだけでは応対が少し不親切な気がする。その願望がいかにももっともだと、私自身も内心思うからだ。

何故に四進法で四方陣を作れないのか。それはただ、各行・各列・各対角線の内容を (0, 1, 2, 3) の構成で作ろうとするからだ。(0, 3, 1, 2) でも (1, 0, 3, 2) でもよい、順不同で良いから、各数字を必ず一度ずつ全数使う。そういう昔からの約束を「ラテン方陣」と言う。その約束のせいで全数の構成が不可能になるのだ。これは既に説明した。

では、このラテン条件を放棄して、別の条件を考えてやってはどうか。外見が少しくらい野暮ったくなくても構わない。とにかく全数を作ることはできないか。もし作れたならば、それはそれで「構造の説明」になるのではないか？

ものは試しにやってみるべし。… 試行錯誤の末に、何とか結果を出すことができた。

1. 四進法を使う。今回はこれで行く。その根拠を示す。次図の如く「分解」と「構成」が正逆一対の手順になっている。その対称の手順を、手法として採用しよう。

* 汎対角線型完全四方陣の四進法分解図 *

古典的表記/	数学的表記/	→ 高位/D4i	低位/D4i
$\begin{array}{ c c c c } \hline 1 15 4 14 \\ \hline 12 6 9 7 \\ \hline 13 3 16 2 \\ \hline 8 10 5 11 \\ \hline \end{array}$	$\begin{array}{ c c c c } \hline 0 14 3 13 \\ \hline 11 5 8 6 \\ \hline 12 2 15 1 \\ \hline 7 9 4 10 \\ \hline \end{array}$	$\begin{array}{ c c c c } \hline 0 3 0 3 \\ \hline 2 1 2 1 \\ \hline 3 0 3 0 \\ \hline 1 2 1 2 \\ \hline \end{array}$	$\begin{array}{ c c c c } \hline 0 2 3 1 \\ \hline 3 1 0 2 \\ \hline 0 2 3 1 \\ \hline 3 1 0 2 \\ \hline \end{array}$

(4で割ったときの整商が高位に、剰余が低位に入る)

* 汎対角線型完全四方陣の四進法構成図 *

高位/D4i	低位/D4i	→ 数学的表記/	古典的表記/
$\begin{array}{ c c c c } \hline 0 3 0 3 \\ \hline 2 1 2 1 \\ \hline 3 0 3 0 \\ \hline 1 2 1 2 \\ \hline \end{array}$	$\begin{array}{ c c c c } \hline 0 2 3 1 \\ \hline 3 1 0 2 \\ \hline 0 2 3 1 \\ \hline 3 1 0 2 \\ \hline \end{array}$	$\begin{array}{ c c c c } \hline 0 14 3 13 \\ \hline 11 5 8 6 \\ \hline 12 2 15 1 \\ \hline 7 9 4 10 \\ \hline \end{array}$	$\begin{array}{ c c c c } \hline 1 15 4 14 \\ \hline 12 6 9 7 \\ \hline 13 3 16 2 \\ \hline 8 10 5 11 \\ \hline \end{array}$

(高位×4 + 低位 = 数学的表記 ; 1を加えて古典的表記に)

2. 分解図の高位と低位には、同じような図が出現する。全体的に (0, 1, 2, 3) の4数しか使われていない。これは、「4で割ったときの整商が高位に、剰余が低位に入る」という約束から必然的にそうなる。しかも、元の魔方陣自体が0から15までの16数によって作られる(古典的表記では1から16までの16個の自然数が使われる)。これもまた昔からの約束だ。それ故に、高位面にも低位面にも同じような数字が並んでしまうのである。

この相似性を根拠として、高位・低位によらず、同一のカテゴリーで扱うことにする。今まではこれで「ラテン方陣」を作ってきたのだが、ラテン条件を放棄する手前、これからは「単位方陣」または「オイラー・ユニット」と呼ぶ。引続いて統一的に扱う。

3. 早速にも「補数対称型四方陣」を対象に選んで、その分析に取りかかる。

まずは、目前の事実を観察する。成功は、そこから何を学び取るかにかかっている。

次表の解の実例を見てほしい。その分解図の単位方陣から、我々はどんな特徴を見つけることができるだろうか。まずは、気が付いたことを羅列してみよう。

* 補数対称型四方陣の解の実例：四進法分解図付き *

1# /D4i				2# /D4i				3# /D4i									
1	15	8	10	0312	0231	1	15	8	10	0312	0231	1	15	12	6	0321	0231
14	4	11	5	3021	1320	12	6	13	3	2130	3102	14	4	7	9	3012	1320
12	6	13	3	2130	3102	14	4	11	5	3021	1320	8	10	13	3	1230	3102
7	9	2	16	1203	2013	7	9	2	16	1203	2013	11	5	2	16	2103	2013
4# /D4i				5# /D4i				6# /D4i									
1	15	12	6	0321	0231	1	15	14	4	0330	0213	1	15	14	4	0330	0213
8	10	13	3	1230	3102	12	6	7	9	2112	3120	8	10	11	5	1221	3120
14	4	7	9	3012	1320	8	10	11	5	1221	3120	12	6	7	9	2112	3120
11	5	2	16	2103	2013	13	3	2	16	3003	0213	13	3	2	16	3003	0213
7# /D4i				8# /D4i				9# /D4i									
1	14	8	11	0312	0132	1	14	12	7	0321	0132	1	14	15	4	0330	0123
12	7	13	2	2130	3201	8	11	13	2	1230	3201	12	7	6	9	2112	3210
15	4	10	5	3021	2310	15	4	6	9	3012	2310	8	11	10	5	1221	3210
6	9	3	16	1203	1023	10	5	3	16	2103	1023	13	2	3	16	3003	0123
10# /D4i				11# /D4i				12# /D4i									
1	14	15	4	0330	0123	1	12	14	7	0231	0312	1	12	15	6	0231	0321
8	11	10	5	1221	3210	8	13	11	2	1320	3021	8	13	10	3	1320	3012
12	7	6	9	2112	3210	15	6	4	9	3102	2130	14	7	4	9	3102	1230
13	2	3	16	3003	0123	10	3	5	16	2013	1203	11	2	5	16	2013	2103
13# /D4i				14# /D4i				15# /D4i									
2	16	7	9	0312	1320	2	16	7	9	0312	1320	2	16	11	5	0321	1320
13	3	12	6	3021	0231	11	5	14	4	2130	2013	13	3	8	10	3012	0231
11	5	14	4	2130	2013	13	3	12	6	3021	0231	7	9	14	4	1230	2013
8	10	1	15	1203	3102	8	10	1	15	1203	3102	12	6	1	15	2103	3102
.....																	
25# /D4i				26# /D4i				27# /D4i									
3	16	6	9	0312	2310	3	16	6	9	0312	2310	3	16	10	5	0321	2310
13	2	12	7	3021	0132	10	5	15	4	2130	1023	13	2	8	11	3012	0132
10	5	15	4	2130	1023	13	2	12	7	3021	0132	6	9	15	4	1230	1023
8	11	1	14	1203	3201	8	11	1	14	1203	3201	12	7	1	14	2103	3201
.....																	
31# /D4i				32# /D4i				33# /D4i									
3	10	13	8	0231	2103	3	10	16	5	0231	2130	4	15	5	10	0312	3201
6	15	12	1	1320	1230	6	15	9	4	1320	1203	14	1	11	8	3021	1023
16	5	2	11	3102	3012	13	8	2	11	3102	0312	9	6	16	3	2130	0132
9	4	7	14	2013	0321	12	1	7	14	2013	3021	7	12	2	13	1203	2310
34# /D4i				35# /D4i				36# /D4i									
4	15	5	10	0312	3201	4	15	9	6	0321	3201	4	15	9	6	0321	3201
9	6	16	3	2130	0132	14	1	7	12	3012	1023	5	10	16	3	1230	0132
14	1	11	8	3021	1023	5	10	16	3	1230	0132	14	1	7	12	3012	1023
7	12	2	13	1203	2310	11	8	2	13	2103	2310	11	8	2	13	2103	2310

40# /D4i	41# /D4i	42# /D4i
4 9 15 6 0231 3021	5 16 4 9 1302 0330	5 16 4 9 1302 0330
5 16 10 3 1320 0312	11 2 14 7 2031 2112	10 3 15 6 2031 1221
14 7 1 12 3102 1203	10 3 15 6 2031 1221	11 2 14 7 2031 2112
11 2 8 13 2013 2130	8 13 1 12 1302 3003	8 13 1 12 1302 3003

46# /D4i	47# /D4i	48# /D4i
6 15 3 10 1302 1221	6 12 9 7 1221 1302	6 9 12 7 1221 1032
9 4 16 5 2031 0330	3 13 16 2 0330 2031	3 16 13 2 0330 2301
12 1 13 8 2031 3003	15 1 4 14 3003 2031	15 4 1 14 3003 2301
7 14 2 11 1302 2112	10 8 5 11 2112 1302	10 5 8 11 2112 1032

[標準解の総数 = 48]

気付いたことを箇条書きにしてみると、

- (1) 相互に対称の位置にあるどの2数の組み合わせを取っても、(0, 3), (1, 2), (2, 1) あるいは (3, 0) のいずれかであり、他の組み合わせはない。つまり $a+b=3$ が成り立つ。
(a, b) が「3の補数」を成し、対称的な配置を守る。これは目的方陣と同じ性質だ。
- (2) 各行・各列・主対角線の等和は、 $i+j+k+l=6$ という形で成り立つ。その構成は、単に (0, 1, 2, 3) の組み合わせに限らない。1+2+1+2=6 もあれば、0+3+0+3=6 もある。
- (3) そして、これが一番重要な特徴だが、各面とも 0, 1, 2, 3 の各4数が、4度ずつ平等に使われていて、例外が無い。例えば、0 が3つとか1 が5つあったりはしない。

これらの特徴が、どの単位方陣においても必ず成立している。

であるならば、これらを「構成条件」として使うことができるのではないか？

4. いよいよ対称四方陣 48 解の構成に取りかかろう。

今までにやって来たのと同じ「新オイラー法」で、最初に我々は「単位方陣」を作る。上に見つけた条件下で (0, 1, 2, 3) の4数だけで構成する。

そして、出来た単位方陣の中から任意の2個を選ぶ。それを「高位面」, 「低位面」と見立てて、組み合わせる。高位×4 + 低位 + 1 という計算によって目的方陣を合成する。

プログラミングは、次のようにする。要点だけを示す (英語版で、ごめん!)。

```

/** Basic Form for Self-complementary Magic Square of Order 4 **
//      .---.---.---.---.      * Rows and Columns: C=6; *
//      4| 1| 2| 3| 4| 1      n1+ n2+ n3+ n4 = C ... rw1
//      |---+---+---+---|      n5+ n6+ n7+ n8 = C ... rw2
//      8| 5| 6| 7| 8| 5      n9+n10+n11+n12 = C ... rw3
//      |---+---+---+---|      n13+n14+n15+n16 = C ... rw4
//      12| 9|10|11|12| 9      n1+ n5+ n9+n13 = C ... cl1
//      |---+---+---+---|      n2+ n6+n10+n14 = C ... cl2
//      16|13|14|15|16|13      n3+ n7+n11+n15 = C ... cl3
//      |---.---.---.---|      n4+ n8+n12+n16 = C ... cl4
/** Contant Sums of Self-complementary Pairs *
// n1+n16 = n2+n15 = n3+n14 = n4+n13 = n5+n12
// = n6+n11 = n7+n10 = n8+n9 = n9+n8 = n10+n7 = ... = 3;
//
#include <stdio.h>
//
short int cnt, ecnt;
short cnt2, cnt3;
short luc[13];
short u1,u2;
short nm[17], uflg[17];

```

```

short anm[13][20];
short teun[49][16];
short mtc[49][49];
//
void estp01(void), estp02(void), estp03(void), estp04(void), estp05(void);
void estp06(void), estp07(void), estp08(void), estp09(void), estp10(void);
void estp11(void), estp12(void), estp13(void), estp14(void), estp15(void);
void estp16(void);
void eansrecord(void);
void prlunt(short x);
//
void tblmtc(void);
void cmbcmp(void);
void recprans(void), prans(short x);
//
int main(void){
short n;
printf("\n");
printf("*** Make Self-complementary Magic Squares 4x4 by /D4i ***\n");
for(n=0;n<17;n++){nm[n]=0;}
for(n=0;n<4;n++){uflg[n]=0;}
cnt=0; cnt3=0;
printf(" ** Make Euler Units by /D4i **\n");
estp01(); // Make the Euler Units
if(cnt3>0){prlunt(cnt3);}
ecnt=cnt; printf(" [Total Count of Euler Units = %d]\n",ecnt);
tblmtc();
printf("\n");
printf("*** List of Self-complementary Magic Squares 4x4 made by /D4i **\n");
cnt=0; cnt3=0;
cmbcmp(); // Combine, Compose and Print
if(cnt3>0){prans(cnt3);} // Print the Rest One
printf(" [Solution Counts = %d]\n",cnt);
printf("\n");
return 0;
}
//
// Make the Euler Units
// Level #1
// Set n1 & n16
void estp01(){
short i,j;
for(i=0;i<4;i++){j=3-i;
if((uflg[i]<4)&&(uflg[j]<4)){
nm[1]=i; nm[16]=j;
uflg[i]++; uflg[j]++;
estp02();
uflg[j]--; uflg[i]--;}
}
}
// Set n2 & n15
void estp02(){
short i,j;
for(i=0;i<4;i++){j=3-i;
if((uflg[i]<4)&&(uflg[j]<4)){
nm[2]=i; nm[15]=j;
uflg[i]++; uflg[j]++;
estp03();
uflg[j]--; uflg[i]--;}
}
}

```

```

}
// Set n3 & n14
void estp03(){
  short i,j;
  for(i=0;i<4;i++){j=3-i;
    if((uflg[i]<4)&&(uflg[j]<4)){
      nm[3]=i; nm[14]=j;
      uflg[i]++; uflg[j]++;
      estp04();
      uflg[j]--; uflg[i]--;}
  }
}
// Set n4=6-n1-n2-n3 & n13=6-n16-n15-n14
void estp04(){
  short i,j;
  i=6-nm[1]-nm[2]-nm[3];
  j=6-nm[16]-nm[15]-nm[14];
  if((0<=i)&&(i<4)&&(i+j==3)){
    if((uflg[i]<4)&&(uflg[j]<4)){
      nm[4]=i; nm[13]=j;
      uflg[i]++; uflg[j]++;
      estp05();
      uflg[j]--; uflg[i]--;}
  }
}
// Level #2
// Set n5 & n12
void estp05(){
  short i,j;
  for(i=0;i<4;i++){j=3-i;
    if((uflg[i]<4)&&(uflg[j]<4)){
      nm[5]=i; nm[12]=j;
      uflg[i]++; uflg[j]++;
      estp06();
      uflg[j]--; uflg[i]--;}
  }
}
// Set n9=6-n1-n5-n13 & n8=6-n16-n12-n4
void estp06(){
  short i,j;
  i=6-nm[1]-nm[5]-nm[13];
  j=6-nm[16]-nm[12]-nm[4];
  if((0<=i)&&(i<4)&&(i+j==3)){
    if((uflg[i]<4)&&(uflg[j]<4)){
      nm[9]=i; nm[8]=j;
      uflg[i]++; uflg[j]++;
      estp07();
      uflg[j]--; uflg[i]--;}
  }
}
// Level #3
// Set n6 & n11
void estp07(){
  short i,j;
  for(i=0;i<4;i++){j=3-i;
    if((uflg[i]<4)&&(uflg[j]<4)){cnt2=0;
      nm[6]=i; nm[11]=j;
      uflg[i]++; uflg[j]++;
      estp08();
      uflg[j]--; uflg[i]--;}
  }
}

```

```

}
}
// Set n7=6-n5-n6-n8 & n10=6-n2-n6-n14
void estp08(){
short i,j;
i=6-nm[5]-nm[6]-nm[8];
j=6-nm[2]-nm[6]-nm[14];
if((0<=i)&&(i<4)&&(i+j==3)){
if((uflg[i]<4)&&(uflg[j]<4)){
nm[7]=i; nm[10]=j;
uflg[i]++; uflg[j]++;
estp09();
uflg[j]--; uflg[i]--;}
}
}
// Check n3+n7+n11+n15=34 & n9+n10+n11+n12=34
void estp09(){
short i,j;
i=nm[3]+nm[7]+nm[11]+nm[15];
j=nm[9]+nm[10]+nm[11]+nm[12];
if((i==6)&&(j==6)){eansrecord();}
}
//
// Record the Answers
void eansrecord(){
short n;
for(n=0;n<16;n++){teun[cnt][n]=nm[n+1];}
cnt++; cnt2++;
if(cnt2==1){luc[cnt3]=cnt; luc[cnt3+1]=0;
for(n=0;n<16;n++){anm[cnt3][n]=nm[n+1]; anm[cnt3+1][n]=0;}
cnt3++; if(cnt3==12){prlunt(cnt3); cnt3=0;}
}
}
//
// Print the List of Euler Units
void prlunt(short x){
short l,l4,m,n;
for(m=0;m<x;m++){printf("%5d/",luc[m]);}
printf("\n");
for(l=0;l<4;l++){l4=l*4;
for(m=0;m<x;m++){printf(" ");
for(n=0;n<4;n++){printf("%d",anm[m][l4+n]);}}
printf("\n");
}
}
//
// Check how well they match and Save the Data to the Table
void tblmtc(){
short h,l,n,d;
for(h=0;h<ecnt;h++){
for(l=0;l<ecnt;l++){d=0;
for(n=0;n<16;n++){if(teun[h][n]==teun[l][n]){d++;}}
mtc[h][l]=d;}
}
}
//
// * Combine, Compose and Print *
void cmbcmp(){
short m,n,d,fc;
for(u1=0;u1<(ecnt/2);u1++){

```

```

for(u2=0;u2<ecnt;u2++){cnt2=0;
  if(mtc[u1][u2]==4){
    for(n=1;n<17;n++){uflg[n]=0;}
    for(n=0;n<16;n++){
      d=teun[u1][n]*4+teun[u2][n]+1;
      nm[n+1]=d; uflg[d]++;
    }
    fc=0;
    for(n=1;n<17;n++){if(uflg[n]==0){fc++;}}
    if(fc==0){
      if((nm[1]<nm[4])&&(nm[1]<nm[13])&&(nm[1]<nm[16])&&(nm[5]<nm[2])){
        recprans();}
    }
  }
}
}
//
// Record and Print the Solution List
void recprans(){
  short n;
  cnt++; cnt2++;
  if(cnt2==1){
    luc[cnt3*3]=cnt;
    for(n=1;n<17;n++){anm[cnt3][n]=nm[n]; anm[cnt3+1][n]=0;}
    luc[cnt3*3+1]=u1+1; luc[cnt3*3+2]=u2+1;
    cnt3++; if(cnt3==3){prans(cnt3); cnt3=0;}
  }
}
//
// Print each Answer
void prans(short x){
  short lu1,lu2;
  short l,l4,m,n;
  for(m=0;m<x;m++){
    printf("%5d/%5d/%12d#", luc[m*3+1], luc[m*3+2], luc[m*3]);
    if((m+1)<x){printf(" ");}
    printf("\n");
    for(l=0;l<4;l++){l4=l*4;
      for(m=0;m<x;m++){
        lu1=luc[m*3+1]-1; lu2=luc[m*3+2]-1;
        printf(" ");
        for(n=0;n<4;n++){printf("%d", teun[lu1][l4+n]);}
        printf(" ");
        for(n=0;n<4;n++){printf("%d", teun[lu2][l4+n]);}
        printf(" ");
        for(n=1;n<5;n++){printf("%3d", anm[m][l4+n]);}
        if((m+1)<x){printf(" ");}
      }
    }
  }
}
}

```

単位方陣全体で (0, 1, 2, 3) の各数を 4 度ずつ使うという制限の監視は, `uflg[i]` というフラグ・カウンターにさせている。後半の `void cmbcmp()`; の中では, 同じカウンターに (1, 2, 3, 4, ..., 15, 16) の各数字を 1 度ずつしか使えないという条件を監視させている。早速にもプログラムを実行させてみよう。

* 四進法と新オイラー法によって補数対称型四方陣を構成する *

* 単位方阵の構成結果 *

1/	2/	3/	4/	5/	6/	7/	8/	9/	10/	11/	12/
0123	0132	0132	0213	0231	0231	0312	0312	0321	0321	0330	0330
3210	2310	3201	3120	1320	3102	2130	3021	1230	3012	1221	2112
3210	3201	2310	3120	3102	1320	3021	2130	3012	1230	2112	1221
0123	1023	1023	0213	2013	2013	1203	1203	2103	2103	3003	3003
13/	14/	15/	16/	17/	18/	19/	20/	21/	22/	23/	24/
1023	1023	1032	1203	1203	1221	1221	1230	1230	1302	1320	1320
2310	3201	2301	2130	3021	0330	3003	0321	2103	2031	0231	2013
3201	2310	2301	3021	2130	3003	0330	2103	0321	2031	2013	0231
0132	0132	1032	0312	0312	2112	2112	3012	3012	1302	3102	3102
25/	26/	27/	28/	29/	30/	31/	32/	33/	34/	35/	36/
2013	2013	2031	2103	2103	2112	2112	2130	2130	2301	2310	2310
1320	3102	1302	1230	3012	0330	3003	0312	1203	1032	0132	1023
3102	1320	1302	3012	1230	3003	0330	1203	0312	1032	1023	0132
0231	0231	2031	0321	0321	1221	1221	3021	3021	2301	3201	3201
37/	38/	39/	40/	41/	42/	43/	44/	45/	46/	47/	48/
3003	3003	3012	3012	3021	3021	3102	3102	3120	3201	3201	3210
1221	2112	0321	2103	0312	1203	0231	2013	0213	0132	1023	0123
2112	1221	2103	0321	1203	0312	2013	0231	0213	1023	0132	0123
0330	0330	1230	1230	2130	2130	1320	1320	3120	2310	2310	3210

[単位方阵の総数 = 48]

48個もの単位方阵が出来た。かなり多い（多いか少ないかは、見方にもよる。私には多い）。というのは、組み合わせで $48 \times 48 = 2304$ 通りもの場合を検査して調べなければならぬからだ。その中から48個の標準解を選び出す。その選別が一苦労だ。

調べる場合の数を減らすべく対策を講じよう。高位方阵の数を予め24個に減らしたり、任意の2個の「相似度」を前もって測っておいて、相性がベストの相手と組み合わせるなどの工夫をしよう。時間節約の役に立つ。かつてそれで苦労した経験が活きる。

* 単位方阵を組み合わせる目的方阵を計算合成・選別した結果 *

5/	8/	1#				5/	10/	2#				5/	16/	3#			
0231	0312	1	12	14	7	0231	0321	1	12	15	6	0231	1203	2	11	13	8
1320	3021	8	13	11	2	1320	3012	8	13	10	3	1320	2130	7	14	12	1
3102	2130	15	6	4	9	3102	1230	14	7	4	9	3102	3021	16	5	3	10
2013	1203	10	3	5	16	2013	2103	11	2	5	16	2013	0312	9	4	6	15
5/	21/	4#				5/	28/	5#				5/	33/	6#			
0231	1230	2	11	16	5	0231	2103	3	10	13	8	0231	2130	3	10	16	5
1320	2103	7	14	9	4	1320	1230	6	15	12	1	1320	1203	6	15	9	4
3102	0321	13	8	3	10	3102	3012	16	5	2	11	3102	0312	13	8	2	11
2013	3012	12	1	6	15	2013	0321	9	4	7	14	2013	3021	12	1	7	14
5/	39/	7#				5/	41/	8#				7/	3/	9#			
0231	3012	4	9	14	7	0231	3021	4	9	15	6	0312	0132	1	14	8	11
1320	0321	5	16	11	2	1320	0312	5	16	10	3	2130	3201	12	7	13	2
3102	2103	15	6	1	12	3102	1203	14	7	1	12	3021	2310	15	4	10	5
2013	1230	10	3	8	13	2013	2130	11	2	8	13	1203	1023	6	9	3	16
7/	6/	10#				7/	13/	11#				7/	24/	12#			
0312	0231	1	15	8	10	0312	1023	2	13	7	12	0312	1320	2	16	7	9
2130	3102	12	6	13	3	2130	2310	11	8	14	1	2130	2013	11	5	14	4
3021	1320	14	4	11	5	3021	3201	16	3	9	6	3021	0231	13	3	12	6
1203	2013	7	9	2	16	1203	0132	5	10	4	15	1203	3102	8	10	1	15

7/ 25/ 13#	7/ 36/ 14#	7/ 43/ 15#
0312 2013 3 13 6 12	0312 2310 3 16 6 9	0312 3102 4 14 5 11
2130 1320 10 8 15 1	2130 1023 10 5 15 4	2130 0231 9 7 16 2
3021 3102 16 2 9 7	3021 0132 13 2 12 7	3021 2013 15 1 10 8
1203 0231 5 11 4 14	1203 3201 8 11 1 14	1203 1320 6 12 3 13
9/ 3/ 21#	10/ 5/ 29#	11/ 1/ 33#
0321 0132 1 14 12 7	0321 0231 1 15 12 6	0330 0123 1 14 15 4
1230 3201 8 11 13 2	3012 1320 14 4 7 9	1221 3210 8 11 10 5
3012 2310 15 4 6 9	1230 3102 8 10 13 3	2112 3210 12 7 6 9
2103 1023 10 5 3 16	2103 2013 11 5 2 16	3003 0123 13 2 3 16
12/ 1/ 37#	18/ 1/ 41#	22/ 11/ 45#
0330 0123 1 14 15 4	1221 0123 5 10 11 8	1302 0330 5 16 4 9
2112 3210 12 7 6 9	0330 3210 4 15 14 1	2031 1221 10 3 15 6
1221 3210 8 11 10 5	3003 3210 16 3 2 13	2031 2112 11 2 14 7
3003 0123 13 2 3 16	2112 0123 9 6 7 12	1302 3003 8 13 1 12

[合成解の総数 = 48]

次のリストは清書版。解をきれいに並べ替えた上で、昔のリストと比較対照した。

* 四進法で構成した補数対称型四方陣の標準解のリスト (清書版) *

* (単位方陣の番号 : 高位/低位/ 解の番号 : [新番号] 旧番号#) *

8/ 5/ [1] 1#	7/ 6/ [2] 2#	10/ 5/ [3] 3#
0312 0231 1 15 8 10	0312 0231 1 15 8 10	0321 0231 1 15 12 6
3021 1320 14 4 11 5	2130 3102 12 6 13 3	3012 1320 14 4 7 9
2130 3102 12 6 13 3	3021 1320 14 4 11 5	1230 3102 8 10 13 3
1203 2013 7 9 2 16	1203 2013 7 9 2 16	2103 2013 11 5 2 16
9/ 6/ [4] 4#	12/ 4/ [5] 5#	11/ 4/ [6] 6#
0321 0231 1 15 12 6	0330 0213 1 15 14 4	0330 0213 1 15 14 4
1230 3102 8 10 13 3	2112 3120 12 6 7 9	1221 3120 8 10 11 5
3012 1320 14 4 7 9	1221 3120 8 10 11 5	2112 3120 12 6 7 9
2103 2013 11 5 2 16	3003 0213 13 3 2 16	3003 0213 13 3 2 16
7/ 3/ [7] 7#	9/ 3/ [8] 8#	12/ 1/ [9] 9#
0312 0132 1 14 8 11	0321 0132 1 14 12 7	0330 0123 1 14 15 4
2130 3201 12 7 13 2	1230 3201 8 11 13 2	2112 3210 12 7 6 9
3021 2310 15 4 10 5	3012 2310 15 4 6 9	1221 3210 8 11 10 5
1203 1023 6 9 3 16	2103 1023 10 5 3 16	3003 0123 13 2 3 16
11/ 1/ [10] 10#	5/ 8/ [11] 11#	5/ 10/ [12] 12#
0330 0123 1 14 15 4	0231 0312 1 12 14 7	0231 0321 1 12 15 6
1221 3210 8 11 10 5	1320 3021 8 13 11 2	1320 3012 8 13 10 3
2112 3210 12 7 6 9	3102 2130 15 6 4 9	3102 1230 14 7 4 9
3003 0123 13 2 3 16	2013 1203 10 3 5 16	2013 2103 11 2 5 16
8/ 23/ [13] 13#	7/ 24/ [14] 14#	10/ 23/ [15] 15#
0312 1320 2 16 7 9	0312 1320 2 16 7 9	0321 1320 2 16 11 5
3021 0231 13 3 12 6	2130 2013 11 5 14 4	3012 0231 13 3 8 10
2130 2013 11 5 14 4	3021 0231 13 3 12 6	1230 2013 7 9 14 4
1203 3102 8 10 1 15	1203 3102 8 10 1 15	2103 3102 12 6 1 15
9/ 24/ [16] 16#	12/ 22/ [17] 17#	11/ 22/ [18] 18#
0321 1320 2 16 11 5	0330 1302 2 16 13 3	0330 1302 2 16 13 3
1230 2013 7 9 14 4	2112 2031 11 5 8 10	1221 2031 7 9 12 6
3012 0231 13 3 8 10	1221 2031 7 9 12 6	2112 2031 11 5 8 10
2103 3102 12 6 1 15	3003 1302 14 4 1 15	3003 1302 14 4 1 15

7/ 13/ [19] 19#	9/ 13/ [20] 20#	12/ 15/ [21] 21#
0312 1023 2 13 7 12	0321 1023 2 13 11 8	0330 1032 2 13 16 3
2130 2310 11 8 14 1	1230 2310 7 12 14 1	2112 2301 11 8 5 10
3021 3201 16 3 9 6	3012 3201 16 3 5 10	1221 2301 7 12 9 6
1203 0132 5 10 4 15	2103 0132 9 6 4 15	3003 1032 14 1 4 15
11/ 15/ [22] 22#	5/ 16/ [23] 23#	5/ 21/ [24] 24#
0330 1032 2 13 16 3	0231 1203 2 11 13 8	0231 1230 2 11 16 5
1221 2301 7 12 9 6	1320 2130 7 14 12 1	1320 2103 7 14 9 4
2112 2301 11 8 5 10	3102 3021 16 5 3 10	3102 0321 13 8 3 10
3003 1032 14 1 4 15	2013 0312 9 4 6 15	2013 3012 12 1 6 15
8/ 35/ [25] 25#	7/ 36/ [26] 26#	10/ 35/ [27] 27#
0312 2310 3 16 6 9	0312 2310 3 16 6 9	0321 2310 3 16 10 5
3021 0132 13 2 12 7	2130 1023 10 5 15 4	3012 0132 13 2 8 11
2130 1023 10 5 15 4	3021 0132 13 2 12 7	1230 1023 6 9 15 4
1203 3201 8 11 1 14	1203 3201 8 11 1 14	2103 3201 12 7 1 14
9/ 36/ [28] 28#	7/ 25/ [29] 29#	9/ 25/ [30] 30#
0321 2310 3 16 10 5	0312 2013 3 13 6 12	0321 2013 3 13 10 8
1230 1023 6 9 15 4	2130 1320 10 8 15 1	1230 1320 6 12 15 1
3012 0132 13 2 8 11	3021 3102 16 2 9 7	3012 3102 16 2 5 11
2103 3201 12 7 1 14	1203 0231 5 11 4 14	2103 0231 9 7 4 14
5/ 28/ [31] 31#	5/ 33/ [32] 32#	8/ 47/ [33] 33#
0231 2103 3 10 13 8	0231 2130 3 10 16 5	0312 3201 4 15 5 10
1320 1230 6 15 12 1	1320 1203 6 15 9 4	3021 1023 14 1 11 8
3102 3012 16 5 2 11	3102 0312 13 8 2 11	2130 0132 9 6 16 3
2013 0321 9 4 7 14	2013 3021 12 1 7 14	1203 2310 7 12 2 13
7/ 46/ [34] 34#	10/ 47/ [35] 35#	9/ 46/ [36] 36#
0312 3201 4 15 5 10	0321 3201 4 15 9 6	0321 3201 4 15 9 6
2130 0132 9 6 16 3	3012 1023 14 1 7 12	1230 0132 5 10 16 3
3021 1023 14 1 11 8	1230 0132 5 10 16 3	3012 1023 14 1 7 12
1203 2310 7 12 2 13	2103 2310 11 8 2 13	2103 2310 11 8 2 13
7/ 43/ [37] 37#	9/ 43/ [38] 38#	5/ 39/ [39] 39#
0312 3102 4 14 5 11	0321 3102 4 14 9 7	0231 3012 4 9 14 7
2130 0231 9 7 16 2	1230 0231 5 11 16 2	1320 0321 5 16 11 2
3021 2013 15 1 10 8	3012 2013 15 1 6 12	3102 2103 15 6 1 12
1203 1320 6 12 3 13	2103 1320 10 8 3 13	2013 1230 10 3 8 13
5/ 41/ [40] 40#	22/ 12/ [41] 41#	22/ 11/ [42] 42#
0231 3021 4 9 15 6	1302 0330 5 16 4 9	1302 0330 5 16 4 9
1320 0312 5 16 10 3	2031 2112 11 2 14 7	2031 1221 10 3 15 6
3102 1203 14 7 1 12	2031 1221 10 3 15 6	2031 2112 11 2 14 7
2013 2130 11 2 8 13	1302 3003 8 13 1 12	1302 3003 8 13 1 12
18/ 4/ [43] 43#	18/ 1/ [44] 44#	22/ 19/ [45] 45#
1221 0213 5 11 10 8	1221 0123 5 10 11 8	1302 1221 6 15 3 10
0330 3120 4 14 15 1	0330 3210 4 15 14 1	2031 3003 12 1 13 8
3003 3120 16 2 3 13	3003 3210 16 3 2 13	2031 0330 9 4 16 5
2112 0213 9 7 6 12	2112 0123 9 6 7 12	1302 2112 7 14 2 11
22/ 18/ [46] 46#	18/ 22/ [47] 47#	18/ 15/ [48] 48#
1302 1221 6 15 3 10	1221 1302 6 12 9 7	1221 1032 6 9 12 7
2031 0330 9 4 16 5	0330 2031 3 13 16 2	0330 2301 3 16 13 2
2031 3003 12 1 13 8	3003 2031 15 1 4 14	3003 2301 15 4 1 14
1302 2112 7 14 2 11	2112 1302 10 8 5 11	2112 1032 10 5 8 11

[解の総数 : 新リスト/旧リスト] = 48/48] [OK!]

** Calculated and Listed by Kanji Setsuda on Dec.13, 2011 **

標準解 48 個を過不足なく再現できた。誤答も無い。成功したと考えてよい。

単位方陣に与えた上記 3 条件が、この型の魔方陣の「内部構造」を示していると考えられる。四進法分解と構成の双方向で、単位方陣と全体方陣との相似性を確認できた。

5. 次に「**正方連結完全型**」の汎四方陣を、四進法と新オイラー法で構成してみよう。

単位方陣に与える条件は、次のように変わる。「対称型の補数」の代わりに「**完全型の補数**」が登場する。他方あちこちに四数定和が成り立つ。

(1) 汎対角線上の一つおきに並ぶどの 2 数の間にも $a+c=3$ の等式が成り立つ。いわゆる「完全型の補数」だ。この存在の成立をどの単位方陣にも要求する。

(2) 第一行・第一列の定和および 2×2 の正方連結四数の定和(=6)成立を要求する。

(3) 単位方陣全面で、(0, 1, 2, 3) の 4 数はおのおの平等に 4 度ずつ現れなければならない。例外的な過不足は許されない。

以上を表現するプログラムは、次の通り。全体として前の補数対称型の時と良く似ている。変わったところだけを抜粋して以下に示す。その他の部分は、全く同じで良い。

```
/** Basic Form and Equations for 'Composite & Complete' Pan-Magic Squares 4x4 **
// 14 15 16 13 14 15 16 13 14 15
//
//      .---.---.---.---.
//  2  3  4 | 1| 2| 3| 4| 1  2  3
//      ---+---+---+---+
//  6  7  8 | 5| 6| 7| 8| 5  6  7
//      ---+---+---+---+
// 10 11 12| 9|10|11|12| 9 10 11
//      ---+---+---+---+
// 14 15 16|13|14|15|16|13 14 15
//      .---.---.---.---.
//  2  3  4  1  2  3  4  1  2  3
/** Basic Conditions: C=6 **
// n1+n2+n3+n4=C    ...rw1; | n1+n5+n9+n13=C    ...c11;
/** Composite Conditions: **
// n1+n2+n5+n6=C    ...c1;   | n2+n3+n6+n7=C    ...c2;
// n3+n4+n7+n8=C    ...c3;   | n4+n1+n8+n5=C    ...c4;
// n5+n6+n9+n10=C   ...c5;   | n6+n7+n10+n11=C   ...c6;
// n7+n8+n11+n12=C  ...c7;   | n8+n5+n12+n9=C    ...c8;
// n9+n10+n13+n14=C ...c9;   | n10+n11+n14+n15=C ...c10;
// n11+n12+n15+n16=C...c11;  | n12+n9+n16+n13=C  ...c12;
// n13+n14+n1+n2=C  ...c13;  | n14+n15+n2+n3=C   ...c14;
// n15+n16+n3+n4=C  ...c15;  | n16+n13+n4+n1=C   ...c16;
/** Conditions for 'Complete' Type: CC=3 **
// n1+n11=n2+n12=n3+n9=n4+n10=n5+n15=n6+n16=n7+n13=n8+n14=CC ...cc
//
// Make the Euler Units
// Level #1
// Set n1 & n11
void estp01(){
  short i,j;
  for(i=0;i<4;i++){j=3-i;
    if((uflg[i]<4)&&(uflg[j]<4)){
      nm[1]=i; nm[11]=j;
      uflg[i]++; uflg[j]++;
      estp02();
      uflg[j]--; uflg[i]--;}
  }
}
```

```

// Set n2 & n12
void estp02(){
  short i,j;
  for(i=0;i<4;i++){j=3-i;
    if((uflg[i]<4)&&(uflg[j]<4)){
      nm[2]=i; nm[12]=j;
      uflg[i]++; uflg[j]++;
      estp03();
      uflg[j]--; uflg[i]--;}
  }
}
// Set n3 & n9
void estp03(){
  short i,j;
  for(i=0;i<4;i++){j=3-i;
    if((uflg[i]<4)&&(uflg[j]<4)){
      nm[3]=i; nm[9]=j;
      uflg[i]++; uflg[j]++;
      estp04();
      uflg[j]--; uflg[i]--;}
  }
}
// Set n4=6-n1-n2-n3 & n10=6-n11-n12-n9
void estp04(){
  short i,j;
  i=6-nm[1]-nm[2]-nm[3];
  j=6-nm[11]-nm[12]-nm[9];
  if((0<=i)&&(i<4)&&(i+j==3)){
    if((uflg[i]<4)&&(uflg[j]<4)){
      nm[4]=i; nm[10]=j;
      uflg[i]++; uflg[j]++;
      estp05();
      uflg[j]--; uflg[i]--;}
  }
}
// Level #2
// Set n5 & n15
void estp05(){
  short i,j;
  for(i=0;i<4;i++){j=3-i;
    if((uflg[i]<4)&&(uflg[j]<4)){cnt2=0;
      nm[5]=i; nm[15]=j;
      uflg[i]++; uflg[j]++;
      estp06();
      uflg[j]--; uflg[i]--;}
  }
}
// Set n13=6-n1-n5-n9 & n7=6-n11-n15-n3
void estp06(){
  short i,j;
  i=6-nm[1]-nm[5]-nm[9];
  j=6-nm[11]-nm[15]-nm[3];
  if((0<=i)&&(i<4)&&(i+j==3)){
    if((uflg[i]<4)&&(uflg[j]<4)){
      nm[13]=i; nm[7]=j;
      uflg[i]++; uflg[j]++;
      estp07();
      uflg[j]--; uflg[i]--;}
  }
}
}

```

```

// Level #3:
// Set n6=6-n1-n2-n5 & n16=6-n11-n12-n15
void estp07(){
  short i,j;
  i=6-nm[1]-nm[2]-nm[5];
  j=6-nm[11]-nm[12]-nm[15];
  if((0<=i)&&(i<4)&&(i+j==3)){
    if((uflg[i]<4)&&(uflg[j]<4)){
      nm[6]=i; nm[16]=j;
      uflg[i]++; uflg[j]++;
      estp08();
      uflg[j]--; uflg[i]--;}
    }
  }
// Check n2+n3+n6+n7=6 & n5+n6+n9+n10=6 & n6+n7+n10+n11=6
void estp08(){
  short s1,s2,s3;
  s1=nm[2]+nm[3]+nm[6]+nm[7];
  s2=nm[5]+nm[6]+nm[9]+nm[10];
  s3=nm[6]+nm[7]+nm[10]+nm[11];
  if((s1==6)&&(s2==6)&&(s3==6)){estp09();}
}
// Set n8=6-n3-n4-n7 & n14=6-n9-n10-n13
void estp09(){
  short i,j;
  i=6-nm[3]-nm[4]-nm[7];
  j=6-nm[9]-nm[10]-nm[13];
  if((0<=i)&&(i<4)&&(i+j==3)){
    if((uflg[i]<4)&&(uflg[j]<4)){
      nm[8]=i; nm[14]=j;
      uflg[i]++; uflg[j]++;
      estp10();
      uflg[j]--; uflg[i]--;}
    }
  }
// Check n7+n8+n11+n12=6 & n10+n11+n14+n15=6
void estp10(){
  short s1,s2;
  s1=nm[7]+nm[8]+nm[11]+nm[12];
  s2=nm[10]+nm[11]+nm[14]+nm[15];
  if((s1==6)&&(s2==6)){eansrecord();}
}
//
// Record the Answers
void eansrecord(){
  short n;
  for(n=0;n<16;n++){teun[cnt][n]=nm[n+1];}
  cnt++; cnt2++;
  if(cnt2==1){luc[cnt3]=cnt; luc[cnt3+1]=0;
  for(n=0;n<16;n++){anm[cnt3][n]=nm[n+1]; anm[cnt3+1][n]=0;}
  cnt3++; if(cnt3==12){prlunt(cnt3); cnt3=0;}
}
}
//

```

早速にも結果を示そう。まずは「単位方陣」48個のリスト。続けて、組み合わせて計算合成した「正方連結完全型汎四方陣」の標準解のリストを、直接に清書版で示そう。

* 四進法と新オイラー法によって正方連結完全型汎四方陣を構成する *

* 単位方阵の構成結果 *

1/	2/	3/	4/	5/	6/	7/	8/	9/	10/	11/	12/
0123	0123	0132	0213	0213	0231	0303	0303	0312	0312	0321	0321
2301	3210	3201	1302	3120	3102	1212	2121	1203	3021	2103	3012
1032	1032	0132	2031	2031	0231	3030	3030	2130	2130	1230	1230
3210	2301	3201	3120	1302	3102	2121	1212	3021	1203	3012	2103
13/	14/	15/	16/	17/	18/	19/	20/	21/	22/	23/	24/
1023	1032	1032	1203	1203	1212	1212	1230	1230	1302	1302	1320
2310	2301	3210	0312	2130	0303	3030	2103	3012	0213	2031	2013
1023	0123	0123	3021	3021	2121	2121	0321	0321	3120	3120	1320
2310	3210	2301	2130	0312	3030	0303	3012	2103	2031	0213	2013
25/	26/	27/	28/	29/	30/	31/	32/	33/	34/	35/	36/
2013	2031	2031	2103	2103	2121	2121	2130	2130	2301	2301	2310
1320	1302	3120	0321	1230	0303	3030	1203	3021	0123	1032	1023
2013	0213	0213	3012	3012	1212	1212	0312	0312	3210	3210	2310
1320	3120	1302	1230	0321	3030	0303	3021	1203	1032	0123	1023
37/	38/	39/	40/	41/	42/	43/	44/	45/	46/	47/	48/
3012	3012	3021	3021	3030	3030	3102	3120	3120	3201	3210	3210
0321	1230	0312	2130	1212	2121	0231	0213	2031	0132	0123	1032
2103	2103	1203	1203	0303	0303	3102	1302	1302	3201	2301	2301
1230	0321	2130	0312	2121	1212	0231	2031	0213	0132	1032	0123

[単位方阵の総数 = 48]

* 四進法で構成した正方連結完全型汎四方陣の標準解のリスト (清書版) *

* (単位方阵の番号 : 高位/低位/ 解の番号 : [新番号] 旧番号#) *

8/	6/	[1]	1#	7/	6/	[2]	2#	10/	4/	[3]	3#
0303	0231	1 15 4 14		0303	0231	1 15 4 14		0312	0213	1 15 6 12	
2121	3102	12 6 9 7		1212	3102	8 10 5 11		3021	1302	14 4 9 7	
3030	0231	13 3 16 2		3030	0231	13 3 16 2		2130	2031	11 5 16 2	
1212	3102	8 10 5 11		2121	3102	12 6 9 7		1203	3120	8 10 3 13	
9/	5/	[4]	4#	12/	4/	[5]	5#	11/	5/	[6]	6#
0312	0213	1 15 6 12		0321	0213	1 15 10 8		0321	0213	1 15 10 8	
1203	3120	8 10 3 13		3012	1302	14 4 5 11		2103	3120	12 6 3 13	
2130	2031	11 5 16 2		1230	2031	7 9 16 2		1230	2031	7 9 16 2	
3021	1302	14 4 9 7		2103	3120	12 6 3 13		3012	1302	14 4 5 11	
8/	3/	[7]	7#	7/	3/	[8]	8#	9/	2/	[9]	9#
0303	0132	1 14 4 15		0303	0132	1 14 4 15		0312	0123	1 14 7 12	
2121	3201	12 7 9 6		1212	3201	8 11 5 10		1203	3210	8 11 2 13	
3030	0132	13 2 16 3		3030	0132	13 2 16 3		2130	1032	10 5 16 3	
1212	3201	8 11 5 10		2121	3201	12 7 9 6		3021	2301	15 4 9 6	
11/	2/	[10]	10#	4/	10/	[11]	11#	4/	12/	[12]	12#
0321	0123	1 14 11 8		0213	0312	1 12 6 15		0213	0321	1 12 7 14	
2103	3210	12 7 2 13		1302	3021	8 13 3 10		1302	3012	8 13 2 11	
1230	1032	6 9 16 3		2031	2130	11 2 16 5		2031	1230	10 3 16 5	
3012	2301	15 4 5 10		3120	1203	14 7 9 4		3120	2103	15 6 9 4	
8/	24/	[13]	13#	7/	24/	[14]	14#	10/	22/	[15]	15#
0303	1320	2 16 3 13		0303	1320	2 16 3 13		0312	1302	2 16 5 11	
2121	2013	11 5 10 8		1212	2013	7 9 6 12		3021	0213	13 3 10 8	
3030	1320	14 4 15 1		3030	1320	14 4 15 1		2130	3120	12 6 15 1	
1212	2013	7 9 6 12		2121	2013	11 5 10 8		1203	2031	7 9 4 14	

9/	23/	[16]	16#	12/	22/	[17]	17#	11/	23/	[18]	18#
0312	1302	2	16	5	11	0321	1302	2	16	9	7
1203	2031	7	9	4	14	3012	0213	13	3	6	12
2130	3120	12	6	15	1	1230	3120	8	10	15	1
3021	0213	13	3	10	8	2103	2031	11	5	4	14
8/	13/	[19]	19#	7/	13/	[20]	20#	9/	14/	[21]	21#
0303	1023	2	13	3	16	0303	1023	2	13	3	16
2121	2310	11	8	10	5	1212	2310	7	12	6	9
3030	1023	14	1	15	4	3030	1023	14	1	15	4
1212	2310	7	12	6	9	2121	2310	11	8	10	5
11/	14/	[22]	22#	4/	17/	[23]	23#	4/	20/	[24]	24#
0321	1032	2	13	12	7	0213	1203	2	11	5	16
2103	2301	11	8	1	14	1302	2130	7	14	4	9
1230	0123	5	10	15	4	2031	3021	12	1	15	6
3012	3210	16	3	6	9	3120	0312	13	8	10	3
8/	36/	[25]	25#	7/	36/	[26]	26#	10/	34/	[27]	27#
0303	2310	3	16	2	13	0303	2310	3	16	2	13
2121	1023	10	5	11	8	1212	1023	6	9	7	12
3030	2310	15	4	14	1	3030	2310	15	4	14	1
1212	1023	6	9	7	12	2121	1023	10	5	11	8
9/	35/	[28]	28#	12/	34/	[29]	29#	11/	35/	[30]	30#
0312	2301	3	16	5	10	0321	2301	3	16	9	6
1203	1032	6	9	4	15	3012	0123	13	2	7	12
2130	3210	12	7	14	1	1230	3210	8	11	14	1
3021	0123	13	2	11	8	2103	1032	10	5	4	15
8/	25/	[31]	31#	7/	25/	[32]	32#	9/	26/	[33]	33#
0303	2013	3	13	2	16	0303	2013	3	13	2	16
2121	1320	10	8	11	5	1212	1320	6	12	7	9
3030	2013	15	1	14	4	3030	2013	15	1	14	4
1212	1320	6	12	7	9	2121	1320	10	8	11	5
11/	26/	[34]	34#	8/	46/	[35]	35#	7/	46/	[36]	36#
0321	2031	3	13	12	6	0303	3201	4	15	1	14
2103	1302	10	8	1	15	2121	0132	9	6	12	7
1230	0213	5	11	14	4	3030	3201	16	3	13	2
3012	3120	16	2	7	9	1212	0132	5	10	8	11
10/	48/	[37]	37#	9/	47/	[38]	38#	12/	48/	[39]	39#
0312	3210	4	15	6	9	0312	3210	4	15	6	9
3021	1032	14	1	12	7	1203	0123	5	10	3	16
2130	2301	11	8	13	2	2130	2301	11	8	13	2
1203	0123	5	10	3	16	3021	1032	14	1	12	7
11/	47/	[40]	40#	8/	43/	[41]	41#	7/	43/	[42]	42#
0321	3210	4	15	10	5	0303	3102	4	14	1	15
2103	0123	9	6	3	16	2121	0231	9	7	12	6
1230	2301	7	12	13	2	3030	3102	16	2	13	3
3012	1032	14	1	8	11	1212	0231	5	11	8	10
9/	44/	[43]	43#	11/	44/	[44]	44#	22/	10/	[45]	45#
0312	3120	4	14	7	9	0321	3120	4	14	11	5
1203	0213	5	11	2	16	2103	0213	9	7	2	16
2130	1302	10	8	13	3	1230	1302	6	12	13	3
3021	2031	15	1	12	6	3012	2031	15	1	8	10
0321	1302	2	16	9	7	0321	1302	2	16	9	7
2103	2031	11	5	4	14	2103	2031	11	5	4	14
1230	3120	8	10	15	1	1230	3120	8	10	15	1
3012	0213	13	3	6	12	3012	0213	13	3	6	12
0312	1032	2	13	8	11	0312	1032	2	13	8	11
1203	2301	7	12	1	14	1203	2301	7	12	1	14
2130	0123	9	6	15	4	2130	0123	9	6	15	4
3021	3210	16	3	10	5	3021	3210	16	3	10	5
0213	1230	2	11	8	13	0213	1230	2	11	8	13
1302	2103	7	14	1	12	1302	2103	7	14	1	12
2031	0321	9	4	15	6	2031	0321	9	4	15	6
3120	3012	16	5	10	3	3120	3012	16	5	10	3
0312	2301	3	16	5	10	0312	2301	3	16	5	10
3021	0123	13	2	11	8	3021	0123	13	2	11	8
2130	3210	12	7	14	1	2130	3210	12	7	14	1
1203	1032	6	9	4	15	1203	1032	6	9	4	15
0312	2301	3	16	9	6	0312	2301	3	16	9	6
3021	0123	13	2	11	8	3021	0123	13	2	11	8
2130	3210	12	7	14	1	2130	3210	12	7	14	1
1203	1032	6	9	4	15	1203	1032	6	9	4	15
0312	2301	3	16	5	10	0312	2301	3	16	5	10
3021	0123	13	2	11	8	3021	0123	13	2	11	8
2130	3210	12	7	14	1	2130	3210	12	7	14	1
1203	1032	6	9	4	15	1203	1032	6	9	4	15
0312	2301	3	16	9	6	0312	2301	3	16	9	6
3021	0123	13	2	11	8	3021	0123	13	2	11	8
2130	3210	12	7	14	1	2130	3210	12	7	14	1
1203	1032	6	9	4	15	1203	1032	6	9	4	15
0312	2301	3	16	5	10	0312	2301	3	16	5	10
3021	0123	13	2	11	8	3021	0123	13	2	11	8
2130	3210	12	7	14	1	2130	3210	12	7	14	1
1203	1032	6	9	4	15	1203	1032	6	9	4	15
0312	2301	3	16	9	6	0312	2301	3	16	9	6
3021	0123	13	2	11	8	3021	0123	13	2	11	8
2130	3210	12	7	14	1	2130	3210	12	7	14	1
1203	1032	6	9	4	15	1203	1032	6	9	4	15
0312	2301	3	16	5	10	0312	2301	3	16	5	10
3021	0123	13	2	11	8	3021	0123	13	2	11	8
2130	3210	12	7	14	1	2130	3210	12	7	14	1
1203	1032	6	9	4	15	1203	1032	6	9	4	15
0312	2301	3	16	9	6	0312	2301	3	16	9	6
3021	0123	13	2	11	8	3021	0123	13	2	11	8
2130	3210	12	7	14	1	2130	3210	12	7	14	1
1203	1032	6	9	4	15	1203	1032	6	9	4	15
0312	2301	3	16	9	6	0312	2301	3	16	9	6
3021	0123	13	2	11	8	3021	0123	13	2	11	8
2130	3210	12	7	14	1	2130	3210	12	7	14	1
1203	1032	6	9	4	15	1203	1032	6	9	4	15
0312	2301	3	16	9	6	0312	2301	3	16	9	6
3021	0123	13	2	11	8	3021	0123	13	2	11	8
2130	3210	12	7	14	1	2130	3210	12	7	14	1
1203	1032	6	9	4	15	1203	1032	6	9	4	15
0312	2301	3	16	9	6	0312	2301	3	16	9	6
3021	0123	13	2	11	8	3021	0123	13	2	11	8
2130	3210	12	7	14	1	2130	3210	12	7	14	1
1203	1032	6	9	4	15	1203	1032	6	9	4	15
0312	2301	3	16	9	6	0312	2301	3	16	9	6
3021	0123	13	2	11	8	3021	0123	13	2	11	8
2130	3210	12	7	14	1	2130	3210	12	7	14	1
1203	1032	6	9	4	15	1203	1032	6	9	4	15
0312	2301	3	16	9	6	0312	2301	3	16	9	6
3021	0123	13	2	11	8	3021	0123	13	2	11	8
2130	3210	12	7	14	1	2130	3210	12	7	14	1
1203	1032	6	9	4	15	1203	1032	6	9	4	15
0312	2301	3	16	9	6	0312	2301	3	16	9	6
3021	0123	13	2	11	8	3021	0123	13	2	11	8
2130	3210	12	7	14	1	2130	3210	12	7	14	1
1203	1032	6	9	4	15	1203	1032	6	9	4	15
0312	2301	3	16	9	6	0312	2301	3	16	9	6
3021	0123	13	2	11	8	3021	0123	13	2	11	8
2130	3210	12	7	14	1	2130	3210	12	7	14	1
1203	1032	6	9	4	15	1203	1032	6	9	4	15
0312	2301	3	16	9	6	0312	2301	3	16	9	6
3021	0123	13	2	11	8	3021	0123	13	2	11	8
2130	3210	12	7	14	1	2130	3210	12	7	14	1
1203	1032	6	9	4	15	1203	1032	6	9	4	15
0312	2301	3	16	9	6	0312	2301	3	16	9	6
3021	0123	13	2	11	8	3021	0123	13	2	11	8
2130	3210	12	7	14	1	2130	3210	12	7	14	1
1203	10										

22/	12/	[46]	46#	22/	17/	[47]	47#	22/	20/	[48]	48#
1302	0321	5 16	3 10	1302	1203	6 15	1 12	1302	1230	6 15	4 9
0213	3012	4 9	6 15	0213	2130	3 10	8 13	0213	2103	3 10	5 16
3120	1230	14 7	12 1	3120	3021	16 5	11 2	3120	0321	13 8	11 2
2031	2103	11 2	13 8	2031	0312	9 4	14 7	2031	3012	12 1	14 7

[解の総数 : 新リスト/旧リスト] = 48/48 [OK!]

** Calculated and Listed by Kanji Setsuda on Dec. 13, 2011 **

かくして目的方陣の標準解 48 個を過不足なく再現構成できた。今度もまた成功した。四進法で分解・構成した単位方陣にも目的方陣と同じ性質が成り立っていることを、逆にそのように仮定して目的方陣を再構成することによって、確かめることができた。

正逆双方向で成り立っているからこそ、単位方陣に与えた 3 条件が目的方陣の「内部構造」になっていることを、示せたのだと思う。

それにしても、第 3 番目の (0, 1, 2, 3) の各数字を平等に 4 度ずつ使うという条件は、いったい何を意味しているのか？ そうでなければならぬ理由は、何なのだろうか？

6. その解明の前に、次は「標準型四方陣」880 個の構成に取りかかろう。

何しろ相手は普通の魔方陣だ。対称型や汎対角線型のような個性があるわけではない。総数も多い。全数を同一の基準で作ろうという我々には、かえって作り難い事情がある。

```

.---.---.---.---. * 基本図と定義式群: C=6 *
| 1| 2| 3| 4| n1+n2+n3+n4=C ... rw1;
---+---+---+---+ n5+n6+n7+n8=C ... rw2;
| 5| 6| 7| 8| n9+n10+n11+n12=C ... rw3;
---+---+---+---+ n13+n14+n15+n16=C ... rw4;
| 9|10|11|12| n1+n5+n9+n13=C ... cl1;
---+---+---+---+ n2+n6+n10+n14=C ... cl2;
|13|14|15|16| n3+n7+n11+n15=C ... cl3;
---'---'---'---' n4+n8+n12+n16=C ... cl4;
n1+n6+n11+n16=C ... Pd; n4+n7+n10+n13=C ... Pb;

```

単位方陣に与える条件だが、まずは目的方陣の定義と相似のものを仮設しよう。

(1) 各行・各列・主対角線 2 本の等和は、 $i+j+k+l=6$ という形で成り立つ。その構成内容には、注文を付けない。 $0+1+2+3=6$ や $1+2+1+2=6$ もあれば、 $0+3+0+3=6$ も可。

(2) そして、各面ともに 0, 1, 2, 3 の各 4 数が、4 度ずつ平等に使われる。例外は認めない。例えば、0 が 3 つとか 1 が 5 つあったりしてはいけない。

今度は、この 2 つの条件しかない。与条件の数が少ない分、解の総数が多くなる。単位方陣を作成するプログラムは、次のようにする。

```

/** Make the New Euler Units *
// Level #1:
// Set n1
void stp01(){
short i;
for(i=0;i<4;i++){
if(uflg[i]<4){nm[1]=i;
uflg[i]++; stp02(); uflg[i]--;}
}
}
// Set n2
void stp02(){
short i;
for(i=3;i>=0;i--){
if(uflg[i]<4){nm[2]=i;

```

```

        uflg[i]++; stp03(); uflg[i]--;}
    }
}
// Set n4
void stp03(){
    short i;
    for(i=0;i<4;i++){
        if(uflg[i]<4){nm[4]=i; cnt2=0;
            uflg[i]++; stp04(); uflg[i]--;}
    }
}
// Set n3
void stp04(){
    short i,sm;
    for(i=0;i<4;i++){
        if(uflg[i]<4){nm[3]=i;
            sm=nm[1]+nm[2]+nm[3]+nm[4];
            if(sm==LSM){uflg[i]++; stp05(); uflg[i]--;}}
    }
}
// Set n5
void stp05(){
    short i;
    for(i=3;i>=0;i--){
        if(uflg[i]<4){nm[5]=i;
            uflg[i]++; stp06(); uflg[i]--;}
    }
}
// Set n13
void stp06(){
    short i;
    for(i=0;i<4;i++){
        if(uflg[i]<4){nm[13]=i;
            uflg[i]++; stp07(); uflg[i]--;}
    }
}
// Set n9
void stp07(){
    short i,sm;
    for(i=0;i<4;i++){
        if(uflg[i]<4){nm[9]=i;
            sm=nm[1]+nm[5]+nm[9]+nm[13];
            if(sm==LSM){uflg[i]++; stp08(); uflg[i]--;}}
    }
}
// Level #2:
// Set n6
void stp08(){
    short i;
    for(i=0;i<4;i++){
        if(uflg[i]<4){nm[6]=i;
            uflg[i]++; stp09(); uflg[i]--;}
    }
}
// Set n7
void stp09(){
    short i;
    for(i=0;i<4;i++){
        if(uflg[i]<4){nm[7]=i;
            uflg[i]++; stp10(); uflg[i]--;}
    }
}

```

```

}
}
// Set n10
void stp10(){
    short i,sm;
    for(i=0;i<4;i++){
        if(uflg[i]<4){nm[10]=i;
            sm=nm[4]+nm[7]+nm[10]+nm[13];
            if(sm==LSM){uflg[i]++; stp11(); uflg[i]--;}}
    }
}
// Set n8
void stp11(){
    short i,sm;
    for(i=0;i<4;i++){
        if(uflg[i]<4){nm[8]=i;
            sm=nm[5]+nm[6]+nm[7]+nm[8];
            if(sm==LSM){uflg[i]++; stp12(); uflg[i]--;}}
    }
}
// Set n14
void stp12(){
    short i,sm;
    for(i=0;i<4;i++){
        if(uflg[i]<4){nm[14]=i;
            sm=nm[2]+nm[6]+nm[10]+nm[14];
            if(sm==LSM){uflg[i]++; stp13(); uflg[i]--;}}
    }
}
// Level #3:
// Set n11
void stp13(){
    short i;
    for(i=0;i<4;i++){
        if(uflg[i]<4){nm[11]=i;
            uflg[i]++; stp14(); uflg[i]--;}
    }
}
// Set n12
void stp14(){
    short i,sm;
    for(i=0;i<4;i++){
        if(uflg[i]<4){nm[12]=i;
            sm=nm[9]+nm[10]+nm[11]+nm[12];
            if(sm==LSM){uflg[i]++; stp15(); uflg[i]--;}}
    }
}
// Set n15
void stp15(){
    short i,sm;
    for(i=0;i<4;i++){
        if(uflg[i]<4){nm[15]=i;
            sm=nm[3]+nm[7]+nm[11]+nm[15];
            if(sm==LSM){uflg[i]++; stp16(); uflg[i]--;}}
    }
}
// Set n16
void stp16(){
    short i,sm1,sm2,sm3;
    for(i=0;i<4;i++){

```

```

    if(uflg[i]<4){nm[16]=i;
        sm1=nm[13]+nm[14]+nm[15]+nm[16];
        sm2=nm[4]+nm[8]+nm[12]+nm[16];
        sm3=nm[1]+nm[6]+nm[11]+nm[16];
        if((sm1==LSM)&&(sm2==LSM)&&(sm3==LSM)){
            uflg[i]++; ansrecord(); uflg[i]--;}
    }
}
/**/
/* Record the Answers */
void ansrecord(){
    short n;
    for(n=0;n<16;n++){tlu[cnt][n]=nm[n+1];}
    cnt++; cnt2++;
    if(cnt2==1){luc[cnt3]=cnt; luc[cnt3+1]=0;
        for(n=0;n<16;n++){anm[cnt3][n]=nm[n+1]; anm[cnt3+1][n]=0;}
        cnt3++; if(cnt3==12){pr12lun(cnt3); cnt3=0;}
    }
}
/**/
/* Print Every 12 Euler Units */
void pr12lun(short x){
    short l,l4,m,n;
    printf("//");
    for(m=0;m<x;m++){printf("%5d/",luc[m]);}
    printf("\n");
    for(l=0;l<4;l++){l4=l*4;
        printf("//");
        for(m=0;m<x;m++){printf(" ");
            for(n=0;n<4;n++){printf("%d",anm[m][l4+n]);}}
        printf("\n");
    }
    printf("//\n");
}
}

```

これで出力させたところ、次のような結果を得た。先ず「単位方陣」のリスト。

**** 四進法と新オイラー法によって作った「単位方陣」のリスト（抜粋） ****

1/	3/	11/	17/	23/	31/	38/	44/	51/	57/	65/	80/
0330	0321	0312	0303	0231	0213	0132	0123	0033	1320	1302	1230
2112	3012	3021	3120	3102	3120	3201	3210	3210	3012	3030	3012
1221	1230	2130	2031	1320	3120	2310	3210	2121	0321	2121	0321
3003	2103	1203	1212	2013	0213	1023	0123	1302	2013	0213	2103
88/	90/	96/	102/	108/	123/	129/	135/	150/	156/	162/	168/
1221	1212	1203	1122	1032	1023	2310	2301	2211	2130	2121	2112
3003	3030	3021	3210	3120	3201	3021	3030	3120	3021	3030	3003
0330	2121	2130	2301	2211	2310	0312	1212	1302	0312	1212	0330
2112	0303	0312	0033	0303	0132	1023	0123	0033	1203	0303	1221
170/	178/	193/	201/	207/	214/	220/	227/	235/	241/	247/	255/
2103	2031	2013	3300	3210	3201	3120	3102	3030	3021	3012	3003
3012	3120	3102	2031	2031	3021	2031	3012	2121	3021	3012	2112
1230	1302	1320	0213	1302	0312	1302	0321	0303	0312	0321	1221
0321	0213	0231	1122	0123	0132	0213	0231	1212	0312	0321	0330

[単位方陣の解の総数 = 256]

**** 四進法と新オイラー法によって構成した標準型四方陣のリスト（抜粋） ****

1/	20/		1#	2/	20/		21#	3/	10/		41#
0330	0303	1	16 13 4	0330	0303	1	16 13 4	0321	0321	1	16 11 6
2112	2121	11	6 7 10	1221	2121	7	10 11 6	3012	0321	13	4 7 10
1221	3030	8	9 12 5	2112	3030	12	5 8 9	1230	3012	8	9 14 3
3003	1212	14	3 2 15	3003	1212	14	3 2 15	2103	3012	12	5 2 15
4/	30/		73#	5/	20/		85#	6/	7/		93#
0321	0231	1	15 12 6	0321	0303	1	16 9 8	0321	0321	1	16 11 6
3102	0321	13	8 3 10	3210	2121	15	10 7 2	2103	1230	10	7 4 13
0231	3012	4	9 14 7	0123	3030	4	5 12 13	1230	3012	8	9 14 3
3012	3102	16	2 5 11	3012	1212	14	3 6 11	3012	2103	15	2 5 12
7/	6/		117#	8/	26/		141#	9/	17/		153#
0321	0321	1	16 11 6	0321	0231	1	15 12 6	0321	0303	1	16 9 8
1230	2103	7	10 13 4	1320	2103	7	14 9 4	0213	3120	4	10 7 13
3012	1230	14	3 8 9	2013	1230	10	3 8 13	3120	2031	15	5 12 2
2103	3012	12	5 2 15	3012	3102	16	2 5 11	3012	1212	14	3 6 11
10/	3/		161#	12/	30/		201#	14/	26/		213#
0321	0321	1	16 11 6	0312	0231	1	15 8 10	0312	0231	1	15 8 10
0321	3012	4	13 10 7	3201	0321	13	12 3 6	2310	2103	11	14 5 4
3012	1230	14	3 8 9	0132	3012	4	5 14 11	1023	1230	6	3 12 13
3012	2103	15	2 5 12	3021	3102	16	2 9 7	3021	3102	16	2 9 7
15/	7/		225#	16/	3/		249#	23/	7/		289#
0312	0321	1	16 7 10	0312	0321	1	16 7 10	0231	0321	1	12 15 6
1203	1230	6	11 4 13	0312	3012	4	13 6 11	3102	1230	14	7 4 9
2130	3012	12	5 14 3	3021	1230	14	3 12 5	1320	3012	8	13 10 3
3021	2103	15	2 9 8	3021	2103	15	2 9 8	2013	2103	11	2 5 16
24/	10/		313#	25/	55/		353#	26/	8/		361#
0231	0321	1	12 15 6	0231	0033	1	9 16 8	0231	0321	1	12 15 6
3102	0321	13	8 3 10	3210	2301	15	12 5 2	2103	1320	10	8 3 13
0231	3012	4	9 14 7	0123	3210	4	7 10 13	1230	2013	7	9 14 4
3102	3012	16	5 2 11	3102	1122	14	6 3 11	3102	3012	16	5 2 11
27/	3/		373#	28/	6/		405#	29/	52/		429#
0231	0321	1	12 15 6	0231	0321	1	12 15 6	0231	0033	1	9 16 8
1320	3012	8	13 10 3	1320	2103	7	14 9 4	0213	3300	4	12 5 13
3102	1230	14	7 4 9	2013	1230	10	3 8 13	3120	2211	15	7 10 2
2013	2103	11	2 5 16	3102	3012	16	5 2 11	3102	1122	14	6 3 11
30/	4/		437#	35/	1/		449#	37/	1/		453#
0231	0321	1	12 15 6	0213	0330	1	12 8 13	0213	0330	1	12 8 13
0321	3102	4	14 9 7	2301	2112	11	14 2 7	1302	2112	7	14 2 11
3012	0231	13	3 8 10	1032	1221	6	3 15 10	2031	1221	10	3 15 6
3102	3012	16	5 2 11	3120	3003	16	5 9 4	3120	3003	16	5 9 4
39/	10/		461#	41/	6/		501#	42/	8/		525#
0132	0321	1	8 15 10	0132	0321	1	8 15 10	0132	0321	1	8 15 10
3201	0321	13	12 3 6	2310	2103	11	14 5 4	1203	1320	6	12 3 13
0132	3012	4	5 14 11	1023	1230	6	3 12 13	2130	2013	11	5 14 4
3201	3012	16	9 2 7	3201	3012	16	9 2 7	3201	3012	16	9 2 7
43/	4/		537#	48/	1/		549#	50/	1/		557#
0132	0321	1	8 15 10	0123	0330	1	8 12 13	0123	0330	1	8 12 13
0312	3102	4	14 5 11	2301	2112	11	14 2 7	1302	2112	7	14 2 11
3021	0231	13	3 12 6	1032	1221	6	3 15 10	2031	1221	10	3 15 6
3201	3012	16	9 2 7	3210	3003	16	9 5 4	3210	3003	16	9 5 4

68/	10/	561#	69/	1/	565#	76/	8/	569#
1302	0321	5 16 3 10	1302	0330	5 16 4 9	1302	0321	5 16 3 10
3120	0321	13 8 11 2	3210	2112	15 10 6 3	0123	1320	2 8 11 13
0033	3012	4 1 14 15	0123	1221	2 7 11 14	3030	2013	15 1 14 4
2211	3012	12 9 6 7	2031	3003	12 1 13 8	2211	3012	12 9 6 7
77/	1/	573#	88/	20/	581#	89/	20/	601#
1302	0330	5 16 4 9	1221	0303	5 12 9 8	1221	0303	5 12 9 8
0213	2112	3 10 6 15	3003	2121	15 2 3 14	0330	2121	3 14 15 2
3120	1221	14 7 11 2	0330	3030	4 13 16 1	3003	3030	16 1 4 13
2031	3003	12 1 13 8	2112	1212	10 7 6 11	2112	1212	10 7 6 11
91/	26/	621#	94/	24/	625#	103/	10/	629#
1212	0231	5 11 8 10	1212	0231	5 11 8 10	1122	0321	5 8 11 10
3120	2103	15 6 9 4	0123	3102	4 6 9 15	3300	0321	13 16 3 2
0033	1230	2 3 16 13	3030	0231	13 3 16 2	0213	3012	4 9 6 15
2301	3102	12 14 1 7	2301	3102	12 14 1 7	2031	3012	12 1 14 7
106/	8/	633#	111/	1/	637#	112/	26/	645#
1122	0321	5 8 11 10	1032	0330	5 4 16 9	1032	0231	5 3 16 10
0303	1320	2 16 3 13	3210	2112	15 10 6 3	3300	2103	15 14 1 4
3210	2013	15 9 6 4	0123	1221	2 7 11 14	0213	1230	2 11 8 13
2031	3012	12 1 14 7	2301	3003	12 13 1 8	2121	3102	12 6 9 7

[解の総数 = 656]

続いて、計算・合成して得た目的方陣の構成解のリストを上を示した。

何と、全部で 880 解ある中の 656 解しか作れなかった。これはこれで、なかなかに見事な「オイラー型」の方陣群なのだが、数的に不満足だ。

7. どうしてなのか。何が足りないというのだろうか。

ここは原点に戻って、次の四進法分解図をじっくり観察しなければならない。

* 標準型四方陣の解の実例 (n1=1): 四進法分解図付き *

[古典的表記#, 数学的表記, 四進法分解図: 高位/ 低位/, サム・チェック\対角線/行|列|]

1#	34/34	/Math	\30/30	/D4i	H\7/5	L\2/10
1	16 13	4 34 34	0 15 12	3 30 30	0 3 3 0 6 6	0 3 0 3 6 6
12	9 8	5 34 34	11 8 7	4 30 30	2 2 1 1 6 6	3 0 3 0 6 6
6	7 10 11	34 34	5 6 9 10	30 30	1 1 2 2 6 6	1 2 1 2 6 6
15	2 3 14	34 34	14 1 2 13	30 30	3 0 0 3 6 6	2 1 2 1 6 6
24#	34/34	/Math	\30/30	/D4i	H\7/5	L\2/10
1	16 5 12	34 34	0 15 4 11	30 30	0 3 1 2 6 6	0 3 0 3 6 6
10	13 4 7	34 34	9 12 3 6	30 30	2 3 0 1 6 6	1 0 3 2 6 6
8	3 14 9	34 34	7 2 13 8	30 30	1 0 3 2 6 6	3 2 1 0 6 6
15	2 11 6	34 34	14 1 10 5	30 30	3 0 2 1 6 6	2 1 2 1 6 6
26#	34/34	/Math	\30/30	/D4i	H\6/6	L\6/6
1	15 14	4 34 34	0 14 13	3 30 30	0 3 3 0 6 6	0 2 1 3 6 6
12	6 7 9	34 34	11 5 6 8	30 30	2 1 1 2 6 6	3 1 2 0 6 6
8	10 11 5	34 34	7 9 10 4	30 30	1 2 2 1 6 6	3 1 2 0 6 6
13	3 2 16	34 34	12 2 1 15	30 30	3 0 0 3 6 6	0 2 1 3 6 6
46#	34/34	/Math	\30/30	/D4i	H\7/5	L\2/10
1	15 10 8	34 34	0 14 9 7	30 30	0 3 2 1 6 6	0 2 1 3 6 6
16	6 3 9	34 34	15 5 2 8	30 30	3 1 0 2 6 6	3 1 2 0 6 6
5	11 14 4	34 34	4 10 13 3	30 30	1 2 3 0 6 6	0 2 1 3 6 6
12	2 7 13	34 34	11 1 6 12	30 30	2 0 1 3 6 6	3 1 2 0 6 6

62#\34/34				/Math	\30/30	/D4i	H\6/6	L\6/6							
1	14	15	4 34 34	0	13	14	3 30 30	0	3	3	0 6 6	0	1	2	3 6 6
12	7	6	9 34 34	11	6	5	8 30 30	2	1	1	2 6 6	3	2	1	0 6 6
8	11	10	5 34 34	7	10	9	4 30 30	1	2	2	1 6 6	3	2	1	0 6 6
13	2	3	16 34 34	12	1	2	15 30 30	3	0	0	3 6 6	0	1	2	3 6 6
78#\34/34				/Math	\30/30	/D4i	H\7/5	L\2/10							
1	14	11	8 34 34	0	13	10	7 30 30	0	3	2	1 6 6	0	1	2	3 6 6
16	5	4	9 34 34	15	4	3	8 30 30	3	1	0	2 6 6	3	0	3	0 6 6
7	12	13	2 34 34	6	11	12	1 30 30	1	2	3	0 6 6	2	3	0	1 6 6
10	3	6	15 34 34	9	2	5	14 30 30	2	0	1	3 6 6	1	2	1	2 6 6
94#\34/34				/Math	\30/30	/D4i	H\6/6	L\6/6							
1	13	16	4 34 34	0	12	15	3 30 30	0	3	3	0 6 6	0	0	3	3 6 6
12	8	5	9 34 34	11	7	4	8 30 30	2	1	1	2 6 6	3	3	0	0 6 6
7	11	10	6 34 34	6	10	9	5 30 30	1	2	2	1 6 6	2	2	1	1 6 6
14	2	3	15 34 34	13	1	2	14 30 30	3	0	0	3 6 6	1	1	2	2 6 6
115#\34/34				/Math	\30/30	/D4i	H\6/6	L\6/6							
1	12	15	6 34 34	0	11	14	5 30 30	0	2	3	1 6 6	0	3	2	1 6 6
14	7	4	9 34 34	13	6	3	8 30 30	3	1	0	2 6 6	1	2	3	0 6 6
8	13	10	3 34 34	7	12	9	2 30 30	1	3	2	0 6 6	3	0	1	2 6 6
11	2	5	16 34 34	10	1	4	15 30 30	2	0	1	3 6 6	2	1	0	3 6 6
147#\34/34				/Math	\30/30	/D4i	H\6/6	L\6/6							
1	11	16	6 34 34	0	10	15	5 30 30	0	2	3	1 6 6	0	2	3	1 6 6
14	8	3	9 34 34	13	7	2	8 30 30	3	1	0	2 6 6	1	3	2	0 6 6
7	13	10	4 34 34	6	12	9	3 30 30	1	3	2	0 6 6	2	0	1	3 6 6
12	2	5	15 34 34	11	1	4	14 30 30	2	0	1	3 6 6	3	1	0	2 6 6
151#\34/34				/Math	\30/30	/D4i	H\7/5	L\2/10							
1	11	14	8 34 34	0	10	13	7 30 30	0	2	3	1 6 6	0	2	1	3 6 6
16	5	4	9 34 34	15	4	3	8 30 30	3	1	0	2 6 6	3	0	3	0 6 6
7	12	13	2 34 34	6	11	12	1 30 30	1	2	3	0 6 6	2	3	0	1 6 6
10	6	3	15 34 34	9	5	2	14 30 30	2	1	0	3 6 6	1	1	2	2 6 6
158#\34/34				/Math	\30/30	/D4i	H\6/6	L\6/6							
1	10	16	7 34 34	0	9	15	6 30 30	0	2	3	1 6 6	0	1	3	2 6 6
15	8	2	9 34 34	14	7	1	8 30 30	3	1	0	2 6 6	2	3	1	0 6 6
6	13	11	4 34 34	5	12	10	3 30 30	1	3	2	0 6 6	1	0	2	3 6 6
12	3	5	14 34 34	11	2	4	13 30 30	2	0	1	3 6 6	3	2	0	1 6 6
172#\34/34				/Math	\30/30	/D4i	H\6/6	L\6/6							
1	9	16	8 34 34	0	8	15	7 30 30	0	2	3	1 6 6	0	0	3	3 6 6
15	12	5	2 34 34	14	11	4	1 30 30	3	2	1	0 6 6	2	3	0	1 6 6
4	7	10	13 34 34	3	6	9	12 30 30	0	1	2	3 6 6	3	2	1	0 6 6
14	6	3	11 34 34	13	5	2	10 30 30	3	1	0	2 6 6	1	1	2	2 6 6
177#\34/34				/Math	\30/30	/D4i	H\6/6	L\6/6							
1	8	15	10 34 34	0	7	14	9 30 30	0	1	3	2 6 6	0	3	2	1 6 6
13	12	3	6 34 34	12	11	2	5 30 30	3	2	0	1 6 6	0	3	2	1 6 6
4	5	14	11 34 34	3	4	13	10 30 30	0	1	3	2 6 6	3	0	1	2 6 6
16	9	2	7 34 34	15	8	1	6 30 30	3	2	0	1 6 6	3	0	1	2 6 6
194#\34/34				/Math	\30/30	/D4i	H\6/6	L\6/6							
1	7	16	10 34 34	0	6	15	9 30 30	0	1	3	2 6 6	0	2	3	1 6 6
14	12	3	5 34 34	13	11	2	4 30 30	3	2	0	1 6 6	1	3	2	0 6 6
4	6	13	11 34 34	3	5	12	10 30 30	0	1	3	2 6 6	3	1	0	2 6 6
15	9	2	8 34 34	14	8	1	7 30 30	3	2	0	1 6 6	2	0	1	3 6 6
203#\34/34				/Math	\30/30	/D4i	H\6/6	L\6/6							
1	6	16	11 34 34	0	5	15	10 30 30	0	1	3	2 6 6	0	1	3	2 6 6


```

15 12 2 5|34|34| 14 11 1 4|30|30| 3 2 0 1|6|6| 2 3 1 0|6|6|
 4 7 13 10|34|34| 3 6 12 9|30|30| 0 1 3 2|6|6| 3 2 0 1|6|6|
14 9 3 8|34|34| 13 8 2 7|30|30| 3 2 0 1|6|6| 1 0 2 3|6|6|
      207#|34|34| /Math \30|30| /D4i H\7|5| L\2|10|
 1 5 16 12|34|34| 0 4 15 11|30|30| 0 1 3 2|6|6| 0 0 3 3|6|6|
10 14 3 7|34|34| 9 13 2 6|30|30| 2 3 0 1|6|6| 1 1 2 2|6|6|
 8 4 13 9|34|34| 7 3 12 8|30|30| 1 0 3 2|6|6| 3 3 0 0|6|6|
15 11 2 6|34|34| 14 10 1 5|30|30| 3 2 0 1|6|6| 2 2 1 1|6|6|
. . . . .

```

一見今までと同じように見える。どの単位方陣にも、(0, 1, 2, 3) の各数字が4度ずつ平等に現れている。どの行列をとっても、その和は6である。定和が成り立っている。

しかし、主対角線の様子が少し違うようだ。もちろん $a+b+c+d=6$ のものもあるが、和が5になるものや10になるものさえある。決して一律ではない。

注意深く観察すると、高位の主対角線4数の和は、5, 6, 7のものしか見当たらない。その時に対応する低位の主対角線4数の和は、10, 6, 2となっている。

これはどういうことか。

$$5 \times 4 + 10 = 30 ; \quad 6 \times 4 + 6 = 30 ; \quad 7 \times 4 + 2 = 30 ;$$

高位の和 $\times 4$ + 低位の和 = 4数の定和 (数学的表記で30) という構造になっている。

目的方陣たる標準型四方陣の側から見れば、単位方陣の中身に一々顧慮することなく、とにかく全体として定和になっていけばよい。それに応じるのが基本的要請なのだ。

8. では単位方陣を作る段階で、与える条件をどう仮定したらよいものだろうか。

- (1) 単位面全体で (0, 1, 2, 3) の各数字を4度ずつ平等に使う。過不足は許さない。
- (2) 行列の和は一律に6として、定和を実現する。
- (3) 問題は、2本の主対角線の扱いだ。

高位面では、 $4 < \text{主対角線の和} < 8$; そして低位面では、和 = 2, 6, 10 という条件を、そのまま課するしかあるまい。

そうすると、高位面と低位面とを別々に作る必要が生じてしまう。今までのように「統一的」に扱っていたのでは、上の条件を簡単には盛り込み難い。

プログラムは、結局、次のように組んだ。

```

//
/** Definitions for Standard Magic Squares of Order 4 **
//.---.---.---.--- * Basic Form and Equations: C=6 *
//| 1| 2| 3| 4| n1+n2+n3+n4=C ... rw1;
//|---+---+---+---| n5+n6+n7+n8=C ... rw2;
//| 5| 6| 7| 8| n9+n10+n11+n12=C ... rw3;
//|---+---+---+---| n13+n14+n15+n16=C ... rw4;
//| 9|10|11|12| n1+n5+n9+n13=C ... cl1;
//|---+---+---+---| n2+n6+n10+n14=C ... cl2;
//|13|14|15|16| n3+n7+n11+n15=C ... cl3;
//|---'---'---'---' n4+n8+n12+n16=C ... cl4;
// n1+n6+n11+n16=C1 ... Pd; n4+n7+n10+n13=C2 ... Pb;
//
#include <stdio.h>
//
short int cnt, hcnt, lcnt;
short cnt2, cnt3;
short luc[12];
short u1,u2;
short nm[17], uflg[17];
short anm[12][20];

```

```

short thlu[184][17], tllu[769][17];
short mtc[184][769];
//
void hstp01(void), hstp02(void), hstp03(void), hstp04(void), hstp05(void);
void hstp06(void), hstp07(void), hstp08(void), hstp09(void), hstp10(void);
void hstp11(void), hstp12(void), hstp13(void), hstp14(void), hstp15(void);
void hstp16(void);
void hansrecord(void);
//
void prlunt(short x);
//
void lstp01(void), lstp02(void), lstp03(void), lstp04(void), lstp05(void);
void lstp06(void), lstp07(void), lstp08(void), lstp09(void), lstp10(void);
void lstp11(void), lstp12(void), lstp13(void), lstp14(void), lstp15(void);
void lstp16(void);
void lansrecord(void);
//
void tblmtc(void);
void cmbcmp(void);
void recprans(void), prans(short x);
//
/* Main Program *
int main(void){
short n;
printf("\n");
printf("*** The 'Standard' Magic Squares 4x4 made by /D4i ***\n");
for(n=0;n<17;n++){nm[n]=0;}
for(n=0;n<4;n++){uflg[n]=0;}
cnt=0; cnt3=0;
printf(" ** Make Euler Units for High Positions by /D4i **\n");
hstp01(); // Make the Euler Units
if(cnt3>0){prlunt(cnt3);}
hcnt=cnt; printf(" [Total Count of Euler Units for High = %d]\n",hcnt);
for(n=0;n<17;n++){nm[n]=0;}
for(n=0;n<4;n++){uflg[n]=0;}
cnt=0; cnt3=0;
printf("\n");
printf(" ** Make Euler Units for Low Positions by /D4i **\n");
lstp01(); // Make the Euler Units
if(cnt3>0){prlunt(cnt3);}
lcnt=cnt; printf(" [Total Count of Euler Units for Low = %d]\n",lcnt);
tblmtc();
printf("\n");
printf("*** List of 'Standard' Magic Squares 4x4 made by /D4i **\n");
lcnt=cnt; cnt=0; cnt3=0;
cmbcmp(); // Combine, Compose and Print
if(cnt3>0){prans(cnt3);} // Print the Rest One
printf(" [Solution Counts = %d]\n",cnt);
printf(" [OK!] ");
printf("** Calculated and Listed by Kanji Setsuda on Dec.13, 2011 **\n");
printf("\n");
return 0;
}
//
// Make the Euler Units for High Positions
// Level #1:
// Set n1
void hstp01(){
short i;
for(i=0;i<4;i++){

```

```

        if(uflg[i]<4){nm[1]=i;
            uflg[i]++; hstp02(); uflg[i]--;}
    }
}
// Set n2
void hstp02(){
    short i;
    for(i=3;i>=0;i--){
        if(uflg[i]<4){nm[2]=i;
            uflg[i]++; hstp03(); uflg[i]--;}
    }
}
// Set n3
void hstp03(){
    short i;
    for(i=0;i<4;i++){cnt2=0;
        if(uflg[i]<4){nm[3]=i;
            uflg[i]++; hstp04(); uflg[i]--;}
    }
}
// Set n4=6-n1-n2-n3 & n4>=n1
void hstp04(){
    short i;
    i=6-nm[1]-nm[2]-nm[3];
    if((nm[1]<=i)&&(i<=3)){
        if(uflg[i]<4){nm[4]=i;
            uflg[i]++; hstp05(); uflg[i]--;}}
}
// Set n5(<n2)
void hstp05(){
    short i;
    for(i=nm[2];i>=0;i--){
        if(uflg[i]<4){nm[5]=i;
            uflg[i]++; hstp06(); uflg[i]--;}
    }
}
// Set n9
void hstp06(){
    short i;
    for(i=0;i<4;i++){
        if(uflg[i]<4){nm[9]=i;
            uflg[i]++; hstp07(); uflg[i]--;}
    }
}
// Set n13=6-n1-n5-n9 & n13>=n1
void hstp07(){
    short i;
    i=6-nm[1]-nm[5]-nm[9];
    if((nm[1]<=i)&&(i<=3)){
        if(uflg[i]<4){nm[13]=i;
            uflg[i]++; hstp08(); uflg[i]--;}}
}
// Level #2:
// Set n6
void hstp08(){
    short i;
    for(i=0;i<4;i++){
        if(uflg[i]<4){nm[6]=i;
            uflg[i]++; hstp09(); uflg[i]--;}
    }
}

```

```

}
// Set n7
void hstp09(){
  short i;
  for(i=0;i<4;i++){
    if(uflg[i]<4){nm[7]=i;
      uflg[i]++; hstp10(); uflg[i]--;}
  }
}
// Set n8=6-n5-n6-n7
void hstp10(){
  short i;
  i=6-nm[5]-nm[6]-nm[7];
  if((0<=i)&&(i<=3)){
    if(uflg[i]<4){nm[8]=i;
      uflg[i]++; hstp11(); uflg[i]--;}}
}
// Set n10
void hstp11(){
  short i,sm;
  for(i=0;i<4;i++){
    if(uflg[i]<4){nm[10]=i;
      sm=nm[4]+nm[7]+nm[10]+nm[13];
      if((4<sm)&&(sm<8)){
        uflg[i]++; hstp12(); uflg[i]--;}}
  }
}
// Set n14=6-n2-n6-n10
void hstp12(){
  short i;
  i=6-nm[2]-nm[6]-nm[10];
  if((0<=i)&&(i<=3)){
    if(uflg[i]<4){nm[14]=i;
      uflg[i]++; hstp13(); uflg[i]--;}}
}
// Level #3:
// Set n11
void hstp13(){
  short i;
  for(i=0;i<4;i++){
    if(uflg[i]<4){nm[11]=i;
      uflg[i]++; hstp14(); uflg[i]--;}
  }
}
// Set n12=6-n9-n10-n11
void hstp14(){
  short i;
  i=6-nm[9]-nm[10]-nm[11];
  if((0<=i)&&(i<=3)){
    if(uflg[i]<4){nm[12]=i;
      uflg[i]++; hstp15(); uflg[i]--;}}
}
// Set n15=6-n3-n7-n11
void hstp15(){
  short i;
  i=6-nm[3]-nm[7]-nm[11];
  if((0<=i)&&(i<=3)){
    if(uflg[i]<4){nm[15]=i;
      uflg[i]++; hstp16(); uflg[i]--;}}
}

```

```

// Set n16=6-n4-n8-n12 & n16=6-n13-n14-n15 & n16>=n1
void hstp16(){
short i,j,sm;
i=6-nm[4]-nm[8]-nm[12];
j=6-nm[13]-nm[14]-nm[15];
if(i==j){
if((nm[1]<=i)&&(i<=3)){
if(uflg[i]<4){nm[16]=i;
sm=nm[1]+nm[6]+nm[11]+nm[16];
if((4<sm)&&(sm<8)){
uflg[i]++; hansrecord(); uflg[i]--;}}}}
}
//
// Record the Answers
void hansrecord(){
short n;
for(n=0;n<16;n++){thlu[cnt][n]=nm[n+1];}
cnt++; cnt2++;
if(cnt2==1){luc[cnt3]=cnt; luc[cnt3+1]=0;
for(n=0;n<16;n++){anm[cnt3][n]=nm[n+1]; anm[cnt3+1][n]=0;}
cnt3++; if(cnt3==11){prlunt(cnt3); cnt3=0;}
}
}
//
// Print the List of Euler Units
void prlunt(short x){
short l,l4,m,n;
for(m=0;m<x;m++){printf("%5d/",luc[m]);}
printf("\n");
for(l=0;l<4;l++){l4=l*4;
for(m=0;m<x;m++){printf(" ");
for(n=0;n<4;n++){printf("%d",anm[m][l4+n]);}}
printf("\n");
}
}
//
// Make the Euler Units for Low Positions
// Level #1:
// Set n1:
void lstp01(){
short i;
for(i=0;i<4;i++){
if(uflg[i]<4){nm[1]=i;
uflg[i]++; lstp02(); uflg[i]--;}
}
}
// Set n2
void lstp02(){
short i;
for(i=0;i<4;i++){
if(uflg[i]<4){nm[2]=i;
uflg[i]++; lstp03(); uflg[i]--;}
}
}
// Set n3
void lstp03(){
short i;
for(i=0;i<4;i++){cnt2=0;
if(uflg[i]<4){nm[3]=i;
uflg[i]++; lstp04(); uflg[i]--;}
}
}

```

```

}
}
// Set n4=6-n1-n2-n3
void lstp04(){
short i;
i=6-nm[1]-nm[2]-nm[3];
if((0<=i)&&(i<=3)){
if(uflg[i]<4){nm[4]=i;
uflg[i]++; lstp05(); uflg[i]--;}}
}
// Set n5
void lstp05(){
short i;
for(i=0;i<4;i++){
if(uflg[i]<4){nm[5]=i;
uflg[i]++; lstp06(); uflg[i]--;}
}
}
// Set n9
void lstp06(){
short i;
for(i=0;i<4;i++){
if(uflg[i]<4){nm[9]=i;
uflg[i]++; lstp07(); uflg[i]--;}
}
}
// Set n13=6-n1-n5-n9
void lstp07(){
short i;
i=6-nm[1]-nm[5]-nm[9];
if((0<=i)&&(i<=3)){
if(uflg[i]<4){nm[13]=i;
uflg[i]++; lstp08(); uflg[i]--;}}
}
// Level #2:
// Set n6
void lstp08(){
short i;
for(i=0;i<4;i++){
if(uflg[i]<4){nm[6]=i;
uflg[i]++; lstp09(); uflg[i]--;}
}
}
// Set n7
void lstp09(){
short i;
for(i=0;i<4;i++){
if(uflg[i]<4){nm[7]=i;
uflg[i]++; lstp10(); uflg[i]--;}
}
}
// Set n8=6-n5-n6-n7
void lstp10(){
short i;
i=6-nm[5]-nm[6]-nm[7];
if((0<=i)&&(i<=3)){
if(uflg[i]<4){nm[8]=i;
uflg[i]++; lstp11(); uflg[i]--;}}
}
// Set n10

```

```

void lstp11(){
    short i,sm;
    for(i=0;i<4;i++){
        if(uflg[i]<4){nm[10]=i;
            sm=nm[4]+nm[7]+nm[10]+nm[13];
            if((sm%4)==2){
                uflg[i]++; lstp12(); uflg[i]--;}}
    }
}
// Set n14=6-n2-n6-n10
void lstp12(){
    short i;
    i=6-nm[2]-nm[6]-nm[10];
    if((0<=i)&&(i<=3)){
        if(uflg[i]<4){nm[14]=i;
            uflg[i]++; lstp13(); uflg[i]--;}}
}
// Level #3:
// Set n11
void lstp13(){
    short i;
    for(i=0;i<4;i++){
        if(uflg[i]<4){nm[11]=i;
            uflg[i]++; lstp14(); uflg[i]--;}}
}
// Set n12=6-n9-n10-n11
void lstp14(){
    short i;
    i=6-nm[9]-nm[10]-nm[11];
    if((0<=i)&&(i<=3)){
        if(uflg[i]<4){nm[12]=i;
            uflg[i]++; lstp15(); uflg[i]--;}}
}
// Set n15=6-n3-n7-n11
void lstp15(){
    short i;
    i=6-nm[3]-nm[7]-nm[11];
    if((0<=i)&&(i<=3)){
        if(uflg[i]<4){nm[15]=i;
            uflg[i]++; lstp16(); uflg[i]--;}}
}
// Set n16=6-n4-n8-n12
void lstp16(){
    short i,j,sm;
    i=6-nm[4]-nm[8]-nm[12];
    j=6-nm[13]-nm[14]-nm[15];
    if(i==j){
        if((0<=i)&&(i<=3)){
            if(uflg[i]<4){nm[16]=i;
                sm=nm[1]+nm[6]+nm[11]+nm[16];
                if((sm%4)==2){
                    uflg[i]++; lansrecord(); uflg[i]--;}}}}
}
//
// Record the Answers
void lansrecord(){
    short n;
    for(n=0;n<16;n++){tllu[cnt][n]=nm[n+1];}
    cnt++; cnt2++;
}

```



```

    if(cnt2==1){luc[cnt3]=cnt; luc[cnt3+1]=0;
        for(n=0;n<16;n++){anm[cnt3][n]=nm[n+1]; anm[cnt3+1][n]=0;}
        cnt3++; if(cnt3==11){prlunt(cnt3); cnt3=0;}
    }
}
//
// Check how well they match and Save the Data to the Table
void tblmtc(){
    short h,l,n,d;
    for(h=0;h<hcnt;h++){
        for(l=0;l<lcnt;l++){d=0;
            for(n=0;n<16;n++){if(thlu[h][n]==tllu[l][n]){d++;}}
            mtc[h][l]=d;}
    }
}
//
/* Combine, Compose and Print *
void cmbcmp(){
    short m,n,d,fc;
    short s1,s2,s3,s4;
    for(u1=0;u1<hcnt;u1++){cnt2=0;
        s1=thlu[u1][0]+thlu[u1][5]+thlu[u1][10]+thlu[u1][15];
        s2=thlu[u1][3]+thlu[u1][6]+thlu[u1][9]+thlu[u1][12];
        for(u2=0;u2<lcnt;u2++){
            s3=tllu[u2][0]+tllu[u2][5]+tllu[u2][10]+tllu[u2][15];
            s4=tllu[u2][3]+tllu[u2][6]+tllu[u2][9]+tllu[u2][12];
            if((s1*4+s3==30)&&(s2*4+s4==30)){
                if(mtc[u1][u2]==4){
                    for(n=1;n<17;n++){uflg[n]=0;}
                    for(n=0;n<16;n++){
                        d=thlu[u1][n]*4+tllu[u2][n]+1;
                        nm[n+1]=d; uflg[d]++;
                    }
                    fc=0;
                    for(n=1;n<17;n++){if(uflg[n]==0){fc++;}}
                    if(fc==0){
                        if((nm[1]<nm[4])&&(nm[1]<nm[13])&&(nm[1]<nm[16])&&(nm[5]<nm[2])){
                            recprans();}}
                }}
        }
    }
}
//
// Record and Print the Solution List
void recprans(){
    short n;
    cnt++; cnt2++;
    if(cnt2==1){
        luc[cnt3*3]=cnt;
        for(n=1;n<17;n++){anm[cnt3][n]=nm[n]; anm[cnt3+1][n]=0;}
        luc[cnt3*3+1]=u1+1; luc[cnt3*3+2]=u2+1;
        cnt3++; if(cnt3==3){prans(cnt3); cnt3=0;}
    }
}
//
// Print each Answer
void prans(short x){
    short lu1,lu2;
    short l,l4,m,n;
    for(m=0;m<x;m++){

```

```

printf("%5d/%5d/%12d#", luc[m*3+1], luc[m*3+2], luc[m*3]);
if((m+1)<x){printf(" ");}}
printf("\n");
for(l=0; l<4; l++){l4=l*4;
for(m=0; m<x; m++){
lu1=luc[m*3+1]-1; lu2=luc[m*3+2]-1;
printf(" ");
for(n=0; n<4; n++){printf("%d", thlu[lu1][l4+n]);}
printf(" ");
for(n=0; n<4; n++){printf("%d", tllu[lu2][l4+n]);}
printf(" ");
for(n=1; n<5; n++){printf("%3d", anm[m][l4+n]);}
if((m+1)<x){printf(" ");}}
}
printf("\n");
}
}
}

```

内容は前とほとんど同じだが、単位方陣作成用のプログラムを高位用と低位用に分けている。step11の n10 および step16 の n16 を定義している箇所を注意深く見てほしい。低位用では、主対角線の和を4で割ったときの剰余が2となるものを可としている。

実行させたところ、単位方陣が多数出て来た。低位用は、768個もあった。

* 標準型四方陣を四進法で構成する *

* 高位用の単位方陣を構成する *

1/	23/	49/	76/	86/	101/	102/	119/	126/	133/	158/
0303	0312	0321	0330	0213	0222	0231	0123	0132	1302	1311
3120	3102	3012	3012	2301	1113	2103	1203	1203	3102	3030
1032	0231	0231	1221	1032	3030	1230	2130	2130	0231	0123
2211	3021	3102	2103	3120	2301	3102	3210	3201	2031	2202
162/	164/	167/	171/	172/	175/	177/	178/			
1203	1212	1221	1113	1122	1131	1023	1032			
2130	0123	2103	0222	0213	0213	0321	0303			
2031	3030	0330	3030	3030	3300	3021	2220			
1302	2301	3012	2301	2301	2022	2301	3111			

[高位用単位方陣の総数 = 183]

* 低位用の単位方陣を構成する *

1/	17/	44/	63/	90/	92/	117/	133/	152/	177/	193/
0033	0123	0132	0213	0222	0231	0303	0312	0321	0330	1023
1122	0123	0132	0123	2130	0213	1032	0132	0123	1023	0312
2211	3210	3201	3210	3003	3102	2121	3201	3012	2112	2310
3300	3210	3201	3120	1311	3120	3210	3021	3210	3201	3021
212/	239/	241/	257/	259/	278/	294/	306/	331/	358/	360/
1032	1113	1122	1131	1203	1212	1221	1230	1302	1311	1320
0123	0330	0033	2310	0132	0123	0132	0123	0033	0132	0123
2301	3201	2211	3003	2130	2301	2310	2103	2211	3003	2103
3210	2022	3300	0222	3201	3030	3003	3210	3120	2220	3120
385/	410/	412/	439/	464/	476/	492/	511/	513/	529/	531/
2013	2022	2031	2103	2112	2121	2130	2202	2211	2220	2301
0213	0330	0213	0123	0330	0033	0312	0330	0033	0312	0123
1320	1203	1122	1230	1203	1302	1203	3021	1302	3003	1032
3120	3111	3300	3210	3021	3210	3021	1113	3120	1131	3210

558/ 577/ 593/ 618/ 637/ 653/ 678/ 680/ 707/ 726/ 753/
 2310 3003 3012 3021 3030 3102 3111 3120 3201 3210 3300
 0132 0132 0123 0132 0123 0123 0330 0213 0132 0123 0033
 1023 1221 0321 0312 1212 0321 1023 0213 0312 0123 1122
 3201 2310 3210 3201 2301 3120 2202 3120 3021 3210 2211

[高位用単位方阵の総数= 768]

高位と低位を組み合わせると $183 \times 768 = \text{約}14\text{万個}$ の場合の数を調べて、検査・検定をしなくてはならない。強力なコンピューターにやらせるにしても、大変な数だ。

これでも少ない方で、主対角線の定和を全然定義しないで作ると、**4464**個もの単位方阵が出来てしまう。それを組み合わせて2千万近くある場合の数を調べるのは、恐怖だ。

*** 単位方阵を組み合わせて計算合成した標準型四方陣のリスト (抜粋) ***

1/ 0303 3120 1032 2211	740/ 3210 1302 0033 2121	1# 4 15 2 13 14 8 9 3 5 1 16 12 11 10 7 6	2/ 0303 3210 1032 2121	153/ 0321 0213 3120 3012	3# 1 16 3 14 13 11 6 4 8 2 15 9 12 5 10 7	3/ 0303 3210 2031	567/ 2310 1203 2130 1023	9# 3 16 2 13 14 11 5 4 7 6 12 9 10 1 15 8
5/ 0303 3021 2130 1212	518/ 2211 0303 3030 1122	12# 3 15 2 14 13 4 9 8 12 5 16 1 6 10 7 11	7/ 0303 3120 2031 1212	153/ 0321 0213 3120 3012	14# 1 16 3 14 13 7 10 4 12 2 15 5 8 9 6 11	9/ 0303 3210 2031 1122	63/ 0213 0123 3210 3120	20# 1 15 2 16 13 10 7 4 12 3 14 5 8 6 11 9
10/ 0303 3210 2121 1032	371/ 1320 2013 1320 2013	22# 2 16 3 13 15 9 6 4 10 8 11 5 7 1 14 12	11/ 0303 2211 1032 3120	250/ 1122 2301 3030 0213	26# 2 14 3 15 11 12 5 6 8 1 16 9 13 7 10 4	13/ 0303 2121 3030 1212	13/ 0033 3300 1122 2211	28# 1 13 4 16 12 8 9 5 14 2 15 3 7 11 6 10
14/ 0303 2130 3021 1212	518/ 2211 0303 3030 1122	68# 3 15 2 14 9 8 13 4 16 1 12 5 6 10 7 11	15/ 0303 2211 3030 1122	20/ 0123 1212 2301 3030	71# 1 14 3 16 10 11 6 7 15 4 13 2 8 5 12 9	16/ 0303 2211 3120 1032	45/ 0132 0312 3021 3201	77# 1 14 4 15 9 12 6 7 16 5 11 2 8 3 13 10
18/ 0303 1212 2031 3120	256/ 1122 3300 2031 0213	85# 2 14 3 15 8 12 5 9 11 1 16 6 13 7 10 4	19/ 0303 1122 3030 2211	425/ 2031 1302 2031 1302	87# 3 13 4 14 6 8 9 11 15 1 16 2 10 12 5 7	20/ 0303 1212 3030 2121	13/ 0033 3300 1122 2211	93# 1 13 4 16 8 12 5 9 14 2 15 3 11 7 10 6
21/ 0303 1212 3120 2031	571/ 2310 2310 1023 1023	133# 3 16 2 13 7 12 6 9 14 5 11 4 10 1 15 8	23/ 0312 3102 0231 3021	518/ 2211 0303 3030 1122	137# 3 15 6 10 13 8 1 12 4 9 16 5 14 2 11 7	24/ 0312 3201 0132 3021	45/ 0132 0312 3021 3201	139# 1 14 8 11 13 12 2 7 4 5 15 10 16 3 9 6
25/ 0312 3210 0123 3021	743/ 3210 1320 2013 0123	145# 4 15 6 9 14 12 7 1 3 5 10 16 13 2 11 8	28/ 0312 3021 2130 1203	68/ 0213 1302 2031 3120	146# 1 15 6 12 14 4 9 7 11 5 16 2 8 10 3 13	29/ 0312 3030 2121 1203	313/ 1230 1302 2031 2103	162# 2 15 8 9 14 4 13 3 11 5 12 6 7 10 1 16
30/ 0312 3120	123/ 0303 2121	163# 1 16 5 12 15 6 11 2	32/ 0312 2103	526/ 2211 3300	165# 3 15 6 10 12 8 1 13	33/ 0312 2301	18/ 0123 0231	169# 1 14 7 12 9 15 4 6

2031	1212	10	3	14	7	1230	0033	5	9	16	4	1032	3102	8	2	13	11
1203	3030	8	9	4	13	3021	1122	14	2	11	7	3021	3210	16	3	10	5
34/	47/			173#		36/	319/			185#		37/	118/			189#	
0312	0132	1	14	8	11	0312	1230	2	15	8	9	0312	0303	1	16	5	12
2310	1203	10	15	5	4	2031	2301	11	4	13	6	2121	1122	10	6	11	7
1023	2130	7	2	12	13	3120	1032	14	5	12	3	3030	2211	15	3	14	2
3021	3201	16	3	9	6	1203	2103	7	10	1	16	1203	3030	8	9	4	13
38/	31/			192#		40/	37/			216#		41/	21/			240#	
0312	0123	1	14	7	12	0312	0123	1	14	7	12	0312	0123	1	14	7	12
2130	2301	11	8	13	2	1203	3210	8	11	2	13	1302	1230	6	15	4	9
3021	3210	16	3	10	5	2130	1032	10	5	16	3	2031	2103	11	2	13	8
1203	1032	6	9	4	15	3021	2301	15	4	9	6	3021	3210	16	3	10	5
. . .																	
65/	550/			367#		66/	47/			371#		67/	31/			383#	
0321	2301	3	16	9	6	0321	0132	1	14	12	7	0321	0123	1	14	11	8
1230	2301	7	12	13	2	1320	1203	6	15	9	4	1230	2301	7	12	13	2
2013	1032	10	1	8	15	2013	2130	11	2	8	13	3012	3210	16	3	6	9
3102	1032	14	5	4	11	3012	3201	16	3	5	10	2103	1032	10	5	4	15
68/	256/			407#		69/	29/			411#		71/	129/			414#	
0321	1122	2	14	11	7	0321	0123	1	14	11	8	0321	0303	1	16	9	8
1230	3300	8	12	13	1	1320	2031	7	13	12	2	0213	3210	4	11	6	13
3102	2031	15	5	4	10	3012	3300	16	4	5	9	3120	1032	14	5	12	3
2013	0213	9	3	6	16	2013	1212	10	3	6	15	3012	2121	15	2	7	10
72/	545/			422#		74/	1/			425#		75/	10/			431#	
0321	2301	3	16	9	6	0321	0033	1	13	12	8	0321	0033	1	13	12	8
0231	1302	2	12	13	7	0312	1122	2	14	7	11	0321	2211	3	15	10	6
3012	2031	15	1	8	10	3021	2211	15	3	10	6	3012	3120	16	2	7	9
3102	1032	14	5	4	11	3012	3300	16	4	5	9	3012	1302	14	4	5	11
. . .																	
157/	24/			827#		164/	56/			831#		165/	44/			835#	
1302	0123	5	14	3	12	1212	0132	5	10	8	11	1212	0132	5	10	8	11
0321	1212	2	15	10	7	0123	3201	4	7	9	14	0303	0132	1	14	4	15
3012	3030	16	1	8	9	3030	0132	13	2	16	3	3120	3201	16	7	9	2
2031	2301	11	4	13	6	2301	3201	12	15	1	6	2031	3201	12	3	13	6
166/	367/			837#		169/	13/			841#		172/	29/			861#	
1212	1320	6	12	7	9	1221	0033	5	9	12	8	1122	0123	5	6	11	12
0330	1302	2	16	13	3	0330	3300	4	16	13	1	0213	2031	3	9	8	14
3003	2031	15	1	4	14	3003	1122	14	2	3	15	3030	3300	16	4	13	1
2121	2013	11	5	10	8	2112	2211	11	7	6	10	2301	1212	10	15	2	7
173/	142/			863#		174/	313/			867#		179/	179/			871#	
1122	0312	5	8	10	11	1122	1230	6	7	12	9	1032	0330	5	4	16	9
0303	2310	3	16	2	13	0330	1302	2	16	13	3	0213	1221	2	11	7	14
3210	1023	14	9	7	4	3003	2031	15	1	4	14	3120	2112	15	6	10	3
2031	3021	12	1	15	6	2211	2103	11	10	5	8	2301	3003	12	13	1	8
180/	135/			875#		181/	371/			876#		182/	95/			878#	
1032	0312	5	4	14	11	1032	1320	6	4	15	9	1032	0231	5	3	16	10
0303	1023	2	13	3	16	0303	2013	3	13	2	16	0312	1032	2	13	8	11
3120	2310	15	8	10	1	3210	1320	14	12	7	1	3021	2301	15	4	9	6
2211	3021	12	9	7	6	2121	2013	11	5	10	8	2301	3102	12	14	1	7
183/	121/			879#													
1032	0303	5	4	13	12												

0321 1212 2 15 10 7
 3012 3030 16 1 8 9
 2301 2121 11 14 3 6 [解の総数 = 880]

[OK!] ** Calculated and Listed by Kanji Setsuda on Dec. 13, 2011 **

解の順序をきれいに並べ替えた次の清書版には、比較対照して得た旧番号も記した。

** 四進法で構成した標準型四方陣の標準解のリスト (清書版) **

* (単位方阵の番号 : 高位/低位/ 解の番号 : [新番号] 旧番号#) *

15/	145/	[1]	1#	33/	120/	[11]	11#	38/	162/	[21]	21#
0303	0312	1 16	2 15	0312	0303	1 16	5 12	0312	0321	1 16	7 10
2211	3021	12 9	7 6	2301	1032	10 13	4 7	2130	2103	11 6	13 4
3030	0312	13 4	14 3	1032	3210	8 3	14 9	3021	1230	14 3	12 5
1122	3021	8 5	11 10	3021	2121	15 2	11 6	1203	3012	8 9	2 15
67/	137/	[31]	31#	82/	120/	[41]	41#	38/	71/	[51]	51#
0321	0312	1 16	10 7	0330	0303	1 16	13 4	0312	0213	1 15	6 12
1230	1203	6 11	13 4	1212	1032	6 9	8 11	2130	1302	10 8	13 3
3012	2130	15 2	8 9	2121	3210	12 7	10 5	3021	3120	16 2	11 5
2103	3021	12 5	3 14	3003	2121	15 2	3 14	1203	2031	7 9	4 14
56/	68/	[61]	61#	50/	94/	[71]	71#	78/	74/	[81]	81#
0321	0213	1 15	10 8	0321	0231	1 15	12 6	0330	0213	1 15	14 4
3012	1302	14 4	5 11	3102	0321	13 8	3 10	2112	2301	11 8	5 10
1230	2031	7 9	16 2	0231	3012	4 9	14 7	1221	1032	6 9	12 7
2103	3120	12 6	3 13	3012	3102	16 2	5 11	3003	3120	16 2	3 13
13/	56/	[91]	91#	41/	21/	[101]	101#	75/	43/	[111]	111#
0303	0132	1 14	4 15	0312	0123	1 14	7 12	0321	0123	1 14	11 8
2121	3201	12 7	9 6	1302	1230	6 15	4 9	0321	3210	4 15	10 5
3030	0132	13 2	16 3	2031	2103	11 2	13 8	3012	3210	16 3	6 9
1212	3201	8 11	5 10	3021	3210	16 3	10 5	3012	0123	13 2	7 12
80/	21/	[121]	121#	46/	13/	[131]	131#	78/	13/	[141]	141#
0330	0123	1 14	15 4	0312	0033	1 13	8 12	0330	0033	1 13	16 4
2211	1230	10 11	8 5	0312	3300	4 16	5 9	2112	3300	12 8	5 9
1122	2103	7 6	9 12	3021	1122	14 2	11 7	1221	1122	6 10	11 7
3003	3210	16 3	2 13	3021	2211	15 3	10 6	3003	2211	15 3	2 14
99/	159/	[151]	151#	116/	118/	[161]	161#	117/	168/	[171]	171#
0213	0321	1 12	7 14	0231	0303	1 12	13 8	0231	0321	1 12	15 6
1302	1230	6 15	4 9	0312	1122	2 14	7 11	0321	3102	4 14	9 7
3120	3012	16 5	10 3	3021	2211	15 3	10 6	3012	0231	13 3	8 10
2031	2103	11 2	13 8	3102	3030	16 5	4 9	3102	3012	16 5	2 11
106/	110/	[181]	181#	111/	17/	[191]	191#	121/	184/	[201]	201#
0231	0231	1 11	16 6	0231	0123	1 10	15 8	0123	0330	1 8	12 13
1320	3102	8 14	9 3	1320	0123	5 14	11 4	1302	2112	7 14	2 11
2013	1320	10 4	7 13	3012	3210	16 3	6 9	2031	1221	10 3	15 6
3102	2013	15 5	2 12	2103	3210	12 7	2 13	3210	3003	16 9	5 4
10/	371/	[211]	211#	40/	350/	[221]	221#	50/	351/	[231]	231#
0303	1320	2 16	3 13	0312	1302	2 16	5 11	0321	1302	2 16	9 7
3210	2013	15 9	6 4	1203	2031	7 9	4 14	3102	2121	15 6	3 10
2121	1320	10 8	11 5	2130	3120	12 6	15 1	0231	3210	4 11	14 5
1032	2013	7 1	14 12	3021	0213	13 3	10 8	3012	0033	13 1	8 12
50/	365/	[241]	241#	84/	348/	[251]	251#	38/	262/	[261]	261#
0321	1320	2 16	11 5	0330	1302	2 16	13 3	0312	1203	2 15	5 12

3102	1230	14	7	4	9	1221	2031	7	9	12	6	2130	0312	9	8	14	3
0231	2103	3	10	13	8	2112	2031	11	5	8	10	3021	3021	16	1	11	6
3012	2013	15	1	6	12	3003	1302	14	4	1	15	1203	2130	7	10	4	13
40/	307/	[271]	271#	56/	312/	[281]	281#	2/	242/	[291]	291#						
0312	1230	2	15	8	9	0321	1230	2	15	12	5	0303	1122	2	14	3	15
1203	0321	5	12	3	14	3012	1230	14	3	8	9	3210	0213	13	11	6	4
2130	2103	11	6	13	4	1230	2103	7	10	13	4	1032	2031	7	1	16	10
3021	3012	16	1	10	7	2103	2103	11	6	1	16	2121	3300	12	8	9	5
...	...																
106/	207/	[391]	391#	119/	372/	[401]	401#	20/	552/	[411]	411#						
0231	1023	2	9	15	8	0123	1320	2	8	11	13	0303	2301	3	16	1	14
1320	2310	7	16	10	1	1203	2301	7	12	1	14	1212	3012	8	9	6	11
2013	3201	12	3	5	14	2130	1032	10	5	16	3	3030	0321	13	4	15	2
3102	0132	13	6	4	11	3210	2013	15	9	6	4	2121	1032	10	5	12	7
20/	566/	[421]	421#	40/	560/	[431]	431#	50/	570/	[441]	441#						
0303	2310	3	16	2	13	0312	2310	3	16	6	9	0321	2310	3	16	10	5
1212	1023	6	9	7	12	1203	0132	5	10	4	15	3102	2130	15	6	4	9
3030	2310	15	4	14	1	2130	3201	12	7	13	2	0231	1203	2	11	13	8
2121	1023	10	5	11	8	3021	1023	14	1	11	8	3012	1023	14	1	7	12
13/	522/	[451]	451#	47/	526/	[461]	461#	75/	517/	[471]	471#						
0303	2211	3	15	2	14	0312	2211	3	15	6	10	0321	2211	3	15	10	6
2121	1122	10	6	11	7	0321	3300	4	16	9	5	0321	0033	1	13	12	8
3030	3300	16	4	13	1	3012	0033	13	1	8	12	3012	3120	16	2	7	9
1212	0033	5	9	8	12	3021	1122	14	2	11	7	3012	1302	14	4	5	11
38/	443/	[481]	481#	75/	453/	[491]	491#	20/	408/	[501]	501#						
0312	2103	3	14	5	12	0321	2103	3	14	9	8	0303	2013	3	13	2	16
2130	0321	9	8	15	2	0321	1230	2	15	12	5	1212	3102	8	10	5	11
3021	3012	16	1	10	7	3012	3012	16	1	6	11	3030	1320	14	4	15	1
1203	1230	6	11	4	13	3012	0321	13	4	7	10	2121	0231	9	7	12	6
...	...																
128/	481/	[571]	571#	14/	763/	[581]	581#	32/	768/	[591]	591#						
0132	2121	3	6	15	10	0303	3300	4	16	1	13	0312	3300	4	16	5	9
1320	0303	5	16	9	4	2130	1212	10	7	14	3	2103	2211	11	7	2	14
2013	3030	12	1	8	13	3021	2121	15	2	11	6	1230	1122	6	10	15	3
3201	1212	14	11	2	7	1212	0033	5	9	8	12	3021	0033	13	1	12	8
65/	762/	[601]	601#	21/	725/	[611]	611#	40/	716/	[621]	621#						
0321	3300	4	16	9	5	0303	3201	4	15	1	14	0312	3201	4	15	5	10
1230	1122	6	10	15	3	1212	3201	8	11	5	10	1203	1023	6	9	3	16
2013	2031	11	1	8	14	3120	0132	13	6	12	3	2130	2310	11	8	14	1
3102	0213	13	7	2	12	2031	0132	9	2	16	7	3021	0132	13	2	12	7
56/	715/	[631]	631#	65/	752/	[641]	641#	20/	662/	[651]	651#						
0321	3201	4	15	9	6	0321	3210	4	15	10	5	0303	3102	4	14	1	15
3012	1023	14	1	7	12	1230	3210	8	11	14	1	1212	0231	5	11	8	10
1230	0132	5	10	16	3	2013	0123	9	2	7	16	3030	3102	16	2	13	3
2103	2310	11	8	2	13	3102	0123	13	6	3	12	2121	0231	9	7	12	6
46/	662/	[661]	661#	75/	662/	[671]	671#	20/	585/	[681]	681#						
0312	3102	4	14	5	11	0321	3102	4	14	9	7	0303	3003	4	13	1	16
0312	0231	1	15	8	10	0321	0231	1	15	12	6	1212	1221	6	11	7	10
3021	3102	16	2	9	7	3012	3102	16	2	5	11	3030	2112	15	2	14	3
3021	0231	13	3	12	6	3012	0231	13	3	8	10	2121	0330	9	8	12	5
...	...																
144/	107/	[781]	781#	149/	13/	[791]	791#	165/	44/	[801]	801#						

1302	0231	5	15	4	10	1302	0033	5	13	4	12	1212	0132	5	10	8	11
2031	3102	12	2	13	7	1320	3300	8	16	9	1	0303	0132	1	14	4	15
2031	0231	9	3	16	6	2013	1122	10	2	7	15	3120	3201	16	7	9	2
1302	3102	8	14	1	11	2031	2211	11	3	14	6	2031	3201	12	3	13	6
183/	126/	[811]	811#	157/	355/	[821]	821#	144/	297/	[831]	831#						
1032	0303	5	4	13	12	1302	1302	6	16	1	11	1302	1221	6	15	3	10
0321	2121	3	14	11	6	0321	3030	4	13	12	5	2031	0330	9	4	16	5
3012	3030	16	1	8	9	3012	2121	15	2	7	10	2031	3003	12	1	13	8
2301	1212	10	15	2	7	2031	0213	9	3	14	8	1302	2112	7	14	2	11
144/	203/	[841]	841#	169/	334/	[851]	851#	173/	309/	[861]	861#						
1302	1023	6	13	3	12	1221	1302	6	12	9	7	1122	1230	6	7	12	9
2031	2310	11	4	14	5	0330	0213	1	15	14	4	0303	0231	1	15	4	14
2031	1023	10	1	15	8	3003	3120	16	2	3	13	3210	3102	16	10	5	3
1302	2310	7	16	2	9	2112	2031	11	5	8	10	2031	2103	11	2	13	8
141/	508/	[871]	871#														
1302	2130	7	14	4	9												
2031	3201	12	3	13	6												
1032	0132	5	2	16	11												
2301	1203	10	15	1	8												

[解の総数：新リスト/旧リスト= 880/880] [OK!]

** Calculated and Listed by Kanji Setsuda on Dec. 13, 2011

抜粋のリストではあるが、880 個の標準解全数個を完全に再現したことが容易に想像できよう。かくして、今度もまた、四進法で目的方陣全数を構成することに成功した。

9. かくのごとく、「ラテン方陣」にとらわれさえしなければ、四進法と新オイラー法で各種の四方陣を全数個作ることができるのである。

では、ラテン方陣とはいったい何だったのか、という反省が次に当然来よう。

一つには、審美的な選択が含まれていたと思う。誰でも、なるべくなら、美しい方法で作りたい。全ての定和条件を (0, 1, 2, 3) で構成し、 $0+1+2+3=6$ の一本で説明したい。

元々は、中世の修道院あたりで考え出され、近世に秘技として伝えられたようだが、確かに美しい方法だ。特に汎五方陣、連立型対称汎五方陣の場合には、見事に全数個を再現構成する。何度接しても、その都度「神品」の印象を感得せざるを得ない。

ラテン方陣は、本来的に優雅で希有のものを作る秘法のような。それに速い。短い時間に多数の成果を上げる。高次方陣を手がける時には、これがずいぶんと助けになる。

ただし、その型の魔方陣全数個を説明することは滅多にない。高次方陣では、ノン・オイラー型（非グレコ・ラテン方陣）の方が圧倒的多数存在することがわかっている。

現実には全部が全部美しいとは限らない、という教訓だとでも言うのであろうか。

しかし、理論は本来現実を、つまり全数個を説明すべきである。そのためにどんなに面倒な分析が必要になろうとも、それを回避してはならないと、私は考える。

10. 三タイプ全部の魔方陣を構成する際、我々は常に次の条件を単位方陣に課した：

「単位面全体で (0, 1, 2, 3) の各数字が 4 度ずつ平等に使われなければならない。」

この条件は、どんな場合でも変わらない。例外が無かった。

五次・六次の時でも、それは変わらない。どんなタイプの魔方陣についても、いつでも同じ条件を課する。例外が無い。

この恒常性は、いったい何を意味するのか。

私は、この条件が「1 から 16 までの連続する自然数をそれぞれ各 1 度ずつ用いて作らなければならない」という魔方陣全般に固有の定義に直接関係していると考ええる。

魔方陣作成時にはどんな重複も欠落も許さないというのが、大昔からの約束だ。

この条件を数学的表記に基づき四進数で表すと、「00, 01, 02, 03, 10, 11, 12, 13, 20, 21, 22, 23, 30, 31, 32, 33^(四)」の各数を各一度ずつ用いて魔方陣を作らなければならない」となる。

そのまま高位と低位の単位方陣に分けてみよう。「(0, 1, 2, 3) という4つの数字をそれぞれ4度ずつ平等に使わなければならない」という条件になる。おわかりだろうか。

もし高位の単位方陣内で0が5度使われていると仮定する。低位の単位方陣内ではそれに対応して何が起きるか。低位面には(0, 1, 2, 3)しか数字がないので、5つ目の0に対応する時にいずれか1つを重複使用して組み合わせなければならなくなる。その結果、例えば(00, 01, 02, 03, 01)となる。いかんせん重複が避けられない。

これは最初の約束に反するので、認められない。

この不合理を避けるためには、高位面で0が5度使われてはならない。

0が3度しか使われないと仮定すると、例えば、(00, 01, 02)で終わって03が不可能になる。つまり、欠落が生じる。また他の数字が5度以上使われていると推理することができることから、この場合も重複使用の禁則に当然反することになる。

0は(他の1, 2, 3も)、いつでも4度ずつ使われなければならないのだ。

魔方陣作成時の昔からの約束が、こんなふうに単位方陣を制約しているのだ。

これが、我々が常に作成プログラムに組み込んでやらなければならない理由だ。

11. その後、次の如き疑問が生じ私の心を離れなかった。「標準型四方陣の場合、二進法による『完全オイラー方陣』は528個だった。それが四進法によると、規則的なオイラー型は656個もある。『ノン・オイラー型』は、果たして $880-528=352$ 個あるのか？ それとも $880-656=224$ 個あるのか？」

多くの場合、二進法と四進法は「同値」の働きをする。例えば、補数対称型や正方連結完全型の四方陣の場合は、二進法で作っても四進法で作っても、規則的なものは同数の48個しか作れない。N進法によらない通常の方法で作っても、48個だ。つまり、この場合はノン・オイラー型が、 $48-48=0$ 個だ。

ところが、標準型の場合は、いささか様子が違う。

先ず分解図の各行列・各対角線が定和に等しい規則的なオイラー型の解の総数が、二進法と四進法とで528個対656個と、相異なる。一体全体、何故に異なってしまうのだろうか？ どちらかが間違っているのでも言うのであろうか？

そんなはずはない。どちらも正しい。各々がきっと別のものを数えているはずだ。

ここで、最初の原点にまた戻る。通常の方法で作った標準型四方陣のリストに返って、二進法と四進法の分解図をあらためて比較研究することにしよう。

次のリストを見てほしい。古典的表記、数学的表記、四進法分解図、二進法分解図と並べてみた。そして、規則に従わない非オイラー型の主対角線に黄色い色を付けてみた。緑色もしくは無色の主対角線は、その和のそれぞれが規則通り定和に等しい。

**** 通常の方法で作った標準型四方陣の解のリスト：N進法分解図付き ****

**** 比較のために四進法分解図と二進法分解図の両方を載せた ****

1#	/Math	\30/30	/D4i: Irregular	/D2i: Irregular
1 16 13 4	0 15 12 3	30 30	0 3 3 0	0 3 0 3
12 9 8 5	11 8 7 4	30 30	2 2 1 1	3 0 3 0
6 7 10 11	5 6 9 10	30 30	1 1 2 2	1 2 1 2
15 2 3 14	14 1 2 13	30 30	3 0 0 3	2 1 2 1
7#	/Math	\30/30	/D4i	/D2i
1 16 11 6	0 15 10 5	30 30	0 3 2 1	0 3 2 1
13 4 7 10	12 3 6 9	30 30	3 0 1 2	0 3 2 1
8 9 14 3	7 8 13 2	30 30	1 2 3 0	3 0 1 2
12 5 2 15	11 4 1 14	30 30	2 1 0 3	3 0 1 2
11#	/Math	\30/30	/D4i	/D2i
1 16 10 7	0 15 9 6	30 30	0 3 2 1	0 3 1 2
13 4 6 11	12 3 5 10	30 30	3 0 1 2	0 3 1 2
8 9 15 2	7 8 14 1	30 30	1 2 3 0	3 0 2 1
12 5 3 14	11 4 2 13	30 30	2 1 0 3	3 0 2 1

[Solution#(Classical) /Mathematical with Sum Checks: \d1/d2|rw|c| /D4i /D2i]

15# /Math \30/30				/D4i				/D2i: Irregular			
1 16 9 8	0 15 8 7	30 30	30 30	0 3 2 1	0 3 0 3	0110	0101	0101	0101		
15 10 7 2	14 9 6 1	30 30	30 30	3 2 1 0	2 1 2 1	1100	1010	1010	0101		
4 5 12 13	3 4 11 12	30 30	30 30	0 1 2 3	3 0 3 0	0011	0101	1010	1010		
14 3 6 11	13 2 5 10	30 30	30 30	3 0 1 2	1 2 1 2	1001	1010	0101	1010		
20# /Math \30/30				/D4i				/D2i			
1 16 7 10	0 15 6 9	30 30	30 30	0 3 1 2	0 3 2 1	0101	0110	0110	0101		
6 11 4 13	5 10 3 12	30 30	30 30	1 2 0 3	1 2 3 0	0101	1001	0110	1010		
12 5 14 3	11 4 13 2	30 30	30 30	2 1 3 0	3 0 1 2	1010	0110	1001	1010		
15 2 9 8	14 1 8 7	30 30	30 30	3 0 2 1	2 1 0 3	1010	1001	1001	0101		
22# /Math \30/30				/D4i				/D2i			
1 16 6 11	0 15 5 10	30 30	30 30	0 3 1 2	0 3 1 2	0101	0110	0101	0110		
7 10 4 13	6 9 3 12	30 30	30 30	1 2 0 3	2 1 3 0	0101	1001	1010	0110		
12 5 15 2	11 4 14 1	30 30	30 30	2 1 3 0	3 0 2 1	1010	0110	1010	1001		
14 3 9 8	13 2 8 7	30 30	30 30	3 0 2 1	1 2 0 3	1010	1001	0101	1001		
24# /Math \30/30				/D4i: Irregular				/D2i: Irregular			
1 16 5 12	0 15 4 11	30 30	30 30	0 3 1 2	0 3 0 3	0101	0110	0101	0101		
10 13 4 7	9 12 3 6	30 30	30 30	2 3 0 1	1 0 3 2	1100	0101	0011	1010		
8 3 14 9	7 2 13 8	30 30	30 30	1 0 3 2	3 2 1 0	0011	1010	1100	1010		
15 2 11 6	14 1 10 5	30 30	30 30	3 0 2 1	2 1 2 1	1010	1001	1010	0101		
26# /Math \30/30				/D4i				/D2i			
1 15 14 4	0 14 13 3	30 30	30 30	0 3 3 0	0 2 1 3	0110	0110	0101	0011		
12 6 7 9	11 5 6 8	30 30	30 30	2 1 1 2	3 1 2 0	1001	0110	1010	1100		
8 10 11 5	7 9 10 4	30 30	30 30	1 2 2 1	3 1 2 0	0110	1001	1010	1100		
13 3 2 16	12 2 1 15	30 30	30 30	3 0 0 3	0 2 1 3	1001	1001	0101	0011		
36# /Math \30/30				/D4i				/D2i			
1 15 12 6	0 14 11 5	30 30	30 30	0 3 2 1	0 2 3 1	0110	0101	0110	0011		
14 4 7 9	13 3 6 8	30 30	30 30	3 0 1 2	1 3 2 0	1001	1010	0110	1100		
8 10 13 3	7 9 12 2	30 30	30 30	1 2 3 0	3 1 0 2	0110	1010	1001	1100		
11 5 2 16	10 4 1 15	30 30	30 30	2 1 0 3	2 0 1 3	1001	0101	1001	0011		
46# /Math \30/30				/D4i: Irregular				/D2i: Irregular			
1 15 10 8	0 14 9 7	30 30	30 30	0 3 2 1	0 2 1 3	0110	0101	0101	0011		
16 6 3 9	15 5 2 8	30 30	30 30	3 1 0 2	3 1 2 0	1001	1100	1010	1100		
5 11 14 4	4 10 13 3	30 30	30 30	1 2 3 0	0 2 1 3	0110	1010	0101	0011		
12 2 7 13	11 1 6 12	30 30	30 30	2 0 1 3	3 1 2 0	1001	0011	1010	1100		
56# /Math \30/30				/D4i				/D2i			
1 15 8 10	0 14 7 9	30 30	30 30	0 3 1 2	0 2 3 1	0101	0110	0110	0011		
13 12 3 6	12 11 2 5	30 30	30 30	3 2 0 1	0 3 2 1	1100	1001	0110	0101		
4 5 14 11	3 4 13 10	30 30	30 30	0 1 3 2	3 0 1 2	0011	0110	1001	1010		
16 2 9 7	15 1 8 6	30 30	30 30	3 0 2 1	3 1 0 2	1010	1001	1001	1100		
60# /Math \30/30				/D4i				/D2i			
1 15 6 12	0 14 5 11	30 30	30 30	0 3 1 2	0 2 1 3	0101	0110	0101	0011		
8 10 3 13	7 9 2 12	30 30	30 30	1 2 0 3	3 1 2 0	0101	1001	1010	1100		
11 5 16 2	10 4 15 1	30 30	30 30	2 1 3 0	2 0 3 1	1010	0110	1010	0011		
14 4 9 7	13 3 8 6	30 30	30 30	3 0 2 1	1 3 0 2	1010	1001	0101	1100		
62# /Math \30/30				/D4i				/D2i			
1 14 15 4	0 13 14 3	30 30	30 30	0 3 3 0	0 1 2 3	0110	0110	0011	0101		
12 7 6 9	11 6 5 8	30 30	30 30	2 1 1 2	3 2 1 0	1001	0110	1100	1010		
8 11 10 5	7 10 9 4	30 30	30 30	1 2 2 1	3 2 1 0	0110	1001	1100	1010		
13 2 3 16	12 1 2 15	30 30	30 30	3 0 0 3	0 1 2 3	1001	1001	0011	0101		
72# /Math \30/30				/D4i				/D2i			
1 14 12 7	0 13 11 6	30 30	30 30	0 3 2 1	0 1 3 2	0110	0101	0011	0110		
15 4 6 9	14 3 5 8	30 30	30 30	3 0 1 2	2 3 1 0	1001	1010	1100	0110		
8 11 13 2	7 10 12 1	30 30	30 30	1 2 3 0	3 2 0 1	0110	1010	1100	1001		
10 5 3 16	9 4 2 15	30 30	30 30	2 1 0 3	1 0 2 3	1001	0101	0011	1001		
78# /Math \30/30				/D4i: Irregular				/D2i: Irregular			
1 14 11 8	0 13 10 7	30 30	30 30	0 3 2 1	0 1 2 3	0110	0101	0011	0101		
16 5 4 9	15 4 3 8	30 30	30 30	3 1 0 2	3 0 3 0	1001	1100	1010	1010		
7 12 13 2	6 11 12 1	30 30	30 30	1 2 3 0	2 3 0 1	0110	1010	1100	0101		
10 3 6 15	9 2 5 14	30 30	30 30	2 0 1 3	1 2 1 2	1001	0011	0101	1010		

[Solution#(Classical) /Mathematical with Sum Checks: \d1/d2|rw|c| /D4i /D2i]

84# /Math \30/30				/D4i		/D2i															
1	14	8	11	0	13	7	10	30	30	0	3	1	2	0	1	3	2	0101	0110	0011	0110
13	12	2	7	12	11	1	6	30	30	3	2	0	1	0	3	1	2	1100	1001	0101	0110
4	5	15	10	3	4	14	9	30	30	0	1	3	2	3	0	2	1	0011	0110	1010	1001
16	3	9	6	15	2	8	5	30	30	3	0	2	1	3	2	0	1	1010	1001	1100	1001
88# /Math \30/30				/D4i: Irregular		/D2i: Irregular															
1	14	7	12	0	13	6	11	30	30	0	3	1	2	0	1	2	3	0101	0110	0011	0101
11	13	2	8	10	12	1	7	30	30	2	3	0	1	2	0	1	3	1100	0101	1001	0011
6	4	15	9	5	3	14	8	30	30	1	0	3	2	1	3	2	0	0011	1010	0110	1100
16	3	10	5	15	2	9	4	30	30	3	0	2	1	3	2	1	0	1010	1001	1100	1010
94# /Math \30/30				/D4i		/D2i															
1	13	16	4	0	12	15	3	30	30	0	3	3	0	0	0	3	3	0110	0110	0011	0011
12	8	5	9	11	7	4	8	30	30	2	1	1	2	3	3	0	0	1001	0110	1100	1100
7	11	10	6	6	10	9	5	30	30	1	2	2	1	2	2	1	1	0110	1001	1100	0011
14	2	3	15	13	1	2	14	30	30	3	0	0	3	1	1	2	2	1001	1001	0011	1100
100# /Math \30/30				/D4i		/D2i															
1	13	12	8	0	12	11	7	30	30	0	3	2	1	0	0	3	3	0110	0101	0011	0011
16	4	5	9	15	3	4	8	30	30	3	0	1	2	3	3	0	0	1001	1010	1100	1100
7	11	14	2	6	10	13	1	30	30	1	2	3	0	2	2	1	1	0110	1010	1100	0011
10	6	3	15	9	5	2	14	30	30	2	1	0	3	1	1	2	2	1001	0101	0011	1100
111# /Math \30/30				/D4i		/D2i: Irregular															
1	13	8	12	0	12	7	11	30	30	0	3	1	2	0	0	3	3	0101	0110	0011	0011
16	11	2	5	15	10	1	4	30	30	3	2	0	1	3	2	1	0	1100	1001	1100	1010
3	6	15	10	2	5	14	9	30	30	0	1	3	2	2	1	2	1	0011	0110	1010	0101
14	4	9	7	13	3	8	6	30	30	3	0	2	1	1	3	0	2	1010	1001	0101	1100
115# /Math \30/30				/D4i		/D2i															
1	12	15	6	0	11	14	5	30	30	0	2	3	1	0	3	2	1	0110	0011	0110	0101
14	7	4	9	13	6	3	8	30	30	3	1	0	2	1	2	3	0	1001	1100	0110	1010
8	13	10	3	7	12	9	2	30	30	1	3	2	0	3	0	1	2	0110	1100	1001	1010
11	2	5	16	10	1	4	15	30	30	2	0	1	3	2	1	0	3	1001	0011	1001	0101
125# /Math \30/30				/D4i		/D2i															
1	12	14	7	0	11	13	6	30	30	0	2	3	1	0	3	1	2	0110	0011	0101	0110
15	6	4	9	14	5	3	8	30	30	3	1	0	2	2	1	3	0	1001	1100	1010	0110
8	13	11	2	7	12	10	1	30	30	1	3	2	0	3	0	2	1	0110	1100	1010	1001
10	3	5	16	9	2	4	15	30	30	2	0	1	3	1	2	0	3	1001	0011	0101	1001
131# /Math \30/30				/D4i		/D2i: Irregular															
1	12	13	8	0	11	12	7	30	30	0	2	3	1	0	3	0	3	0110	0011	0101	0101
16	7	2	9	15	6	1	8	30	30	3	1	0	2	3	2	1	0	1001	1100	1100	1010
3	10	15	6	2	9	14	5	30	30	0	2	3	1	2	1	2	1	0110	0011	1010	0101
14	5	4	11	13	4	3	10	30	30	3	1	0	2	1	0	3	2	1001	1100	0011	1010
147# /Math \30/30				/D4i		/D2i															
1	11	16	6	0	10	15	5	30	30	0	2	3	1	0	2	3	1	0110	0011	0110	0011
14	8	3	9	13	7	2	8	30	30	3	1	0	2	1	3	2	0	1001	1100	0110	1100
7	13	10	4	6	12	9	3	30	30	1	3	2	0	2	0	1	3	0110	1100	1001	0011
12	2	5	15	11	1	4	14	30	30	2	0	1	3	3	1	0	2	1001	0011	1001	1100
158# /Math \30/30				/D4i		/D2i															
1	10	16	7	0	9	15	6	30	30	0	2	3	1	0	1	3	2	0110	0011	0011	0110
15	8	2	9	14	7	1	8	30	30	3	1	0	2	2	3	1	0	1001	1100	1100	0110
6	13	11	4	5	12	10	3	30	30	1	3	2	0	1	0	2	3	0110	1100	0011	1001
12	3	5	14	11	2	4	13	30	30	2	0	1	3	3	2	0	1	1001	0011	1100	1001
172# /Math \30/30				/D4i		/D2i: Irregular															
1	9	16	8	0	8	15	7	30	30	0	2	3	1	0	0	3	3	0110	0011	0011	0011
15	12	5	2	14	11	4	1	30	30	3	2	1	0	2	3	0	1	1100	1010	1100	0101
4	7	10	13	3	6	9	12	30	30	0	1	2	3	3	2	1	0	0011	0101	1100	1010
14	6	3	11	13	5	2	10	30	30	3	1	0	2	1	1	2	2	1001	1100	0011	1100
177# /Math \30/30				/D4i		/D2i															
1	8	15	10	0	7	14	9	30	30	0	1	3	2	0	3	2	1	0011	0110	0110	0101
13	12	3	6	12	11	2	5	30	30	3	2	0	1	0	3	2	1	1100	1001	0110	0101
4	5	14	11	3	4	13	10	30	30	0	1	3	2	3	0	1	2	0011	0110	1001	1010
16	9	2	7	15	8	1	6	30	30	3	2	0	1	3	0	1	2	1100	1001	1001	1010

	194#	/Math	\30/30	/D4i	/D2i
1	7	16	10	0 1 3 2	0 2 3 1
14	12	3	5	3 2 0 1	1 3 2 0
4	6	13	11	0 1 3 2	3 1 0 2
15	9	2	8	3 2 0 1	2 0 1 3
	203#	/Math	\30/30	/D4i	/D2i
1	6	16	11	0 1 3 2	0 1 3 2
15	12	2	5	3 2 0 1	2 3 1 0
4	7	13	10	0 1 3 2	3 2 0 1
14	9	3	8	3 2 0 1	1 0 2 3
	207#	/Math	\30/30	/D4i: Irregular	/D2i: Irregular
1	5	16	12	0 1 3 2	0 0 3 3
10	14	3	7	2 3 0 1	1 1 2 2
8	4	13	9	1 0 3 2	3 3 0 0
15	11	2	6	3 2 0 1	2 2 1 1

[Solution#(Classical) /Mathematical with Sum Checks: \d1/d2|rw|cl| /D4i /D2i]
[標準解の総数 = 880]

解 1#は、四進法分解図で見ても二進法のそれで見ても、不規則な「ノン・オイラー型」と言える。24# も 46# も、同様の「ノン・オイラー型」だ。

しかし、解 15# や 111# は、四進法分解図では規則通りなのに、二進法のそれでは不規則的で、必ずしも全ての主対角線の和が定和 2 に等しいとは限らない。

131# や 172# も、同じタイプだ。こういうケースもあるのだ。

かかる異色の存在が、四進法では「規則に合っている」と数えられる一方、二進法では「規則に反している」としてカウントされない。656 個対 528 個の相違の原因がここにあったのだ。一般に、二進法の方が個数が少なくなる。より細かく調べているからだ。

我々はこの結果をどう扱うべきか？ オイラー型は何個あり、ノン・オイラー型は何個あるとみなすべきか。

私は、今ここで見解を無理に統一する必要は無いと考える。あるがままに受け入れるのが良いと考えている。「構造研究」の立場からも、そう要請される。

他方では、二進法によるより少ない構成解 528 個の集合を『完全オイラー型』と呼んで特別に珍重しても良いのではないかと、とも考えている。

ここで、おなじみの学術用語を新たに定義し直すことにしよう（「2012 年定義」）。

- (1) N進法分解図・構成図においても、各行列・各対角線の和が常に等しくなるなど、元の方陣の特性を相似に持っているタイプに、『オイラー型』あるいは『オイラー方陣』の名を冠する。そして、構成図の単位方陣と元の方陣とが相似でないものを『非オイラー型』と呼ぶ。

オイラー型の補数対称四方陣は、分解図でも対称の位置にあるどの 2 数の和も一定だ。また汎対角線四方陣は、分解図でも汎対角線の定和が成り立っている。反例は無い。

各行列・各対角線の構成内容については、特に定義しない。

- (2) もし、分解図の各行列・各対角線に (0, 1, 2, 3) が各 1 回ずつ使われるようにしたければ、それを予め明示しなければならない。そして、そのタイプを『ラテン方陣』と呼び、方陣全体を『ギリシャ・ラテン方陣』と呼ぼう。以前はこの別名「グレコ・ラテン方陣」のことを「オイラー方陣」と呼んでいたが、その習慣を廃し、今後は別々の意味を持たせたい。

ギリシャ・ラテン方陣の解集合は、オイラー方陣の解集合に含まれる下位集合である。二者が重なる場合もあるが、多くの場合前者の方がより小さい集合になる。

- (3) 標準型四方陣の時のように、四進法と二進法でオイラー方陣の解集合が異なる場合がある。リストを示す時は、そのことを明示することにしよう。二進法分解による方がより精細で純粋な形だと思う人は、そのタイプを特に『完全オイラー型』と呼んでもかまわない。

例えば四進法と二進法は、多くの場合に「同値」の働きをする。

(Written on Dec. 18, 2011 and Mar. 16, 2012 with MacOSX 10.7.3 & Xcode 4.2.1)

by 撰田 寛二(Kanji Setsuda) : E-Mail Address:<jag12001@nifty.com>

<http://homepage2.nifty.com/KanjiSetsuda/>