

## 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10-2節：「新オイラー法」による汎対称立体四方陣の構成

0. 外国のホームページを見ていると、驚かされる情報に時々出あう。

汎対角線型と補数対称型の性質を併せ持った連立型の立体四方陣が存在するという。この記述には大層驚いた。小生は、かような連立型は不可能だと思い込んでいたから。

完全型と対称型の相互書き換えの方法を専ら追究していた小生には、にわかには信じ難い話だった。書き換えるのだから、その時はもちろん両者は違った形になる。何もしない同一解の中で既に両方の性質が同時に成り立っているなどということはありません。

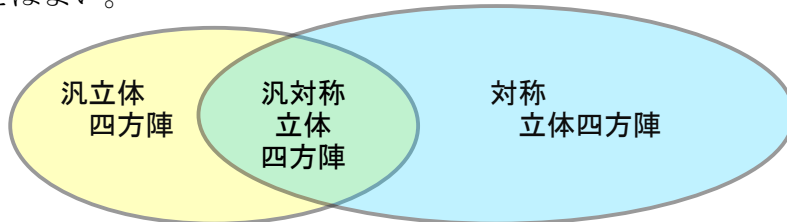
だが、完全型でない汎対角線型ならば、あり得るかも知れない。そう思い直してちょっと調べてみたが、これがどうして解が中々出て来ない。解無しとまで確かめられなかったが、とてもありそうにない感触だったので、この問題はずっと敬遠していた。

ところが、「完全オイラー方陣」を研究していて、『新オイラー法』という非常に高速な方陣構成法をたまたま発見・発明した。これを用いて試してみたら、もしかしたら連立型を作れるのではないか。完全型には無くても、汎対角線型にならあるかも知れないから。

では、どうするか。

「連立型」というのは、両方の性質を併せ持つものを言う。集合論で言えば、汎型の解集合と対称型の解集合の「積集合」に属するような解を求める。

だから、両方の解集合が既に得られている場合には、両方に重複して存在している解を探し出せばよい。



こう言うと、簡単そうに聞こえるかも知れない。だが、これは「言うは易く行うは難い」大変な作業だ。汎立体四方陣が  $n1=1$  の標準解だけでも 240 万余個、対称立体四方陣にいたっては一千万超個もあって、いまだに全数が確定していない。比較しようにもデータが多過ぎて、とても扱いきれない。無駄な探索をしているうちに時間ばかりが経ってしまう。

したがって、手早く調べるには、これはもう最初から直接に連立型を作るしか無い。

1. いつものように連立方程式を並べて「連立型汎対称立体四方陣」を定義しよう。

汎立体四方陣の基本条件である 48+64 本の定和式を全部使い、さらに下のような対称型の補数定和式を加えてプログラムを作り、解を探索するのである。

$n1+n2+n3+n4=C$ ... d1;	$n1+n17+n33+n49=C$ ... d17;	$n1+n5+n9+n13=C$ ... d33;
$n5+n6+n7+n8=C$ ... d2;	$n2+n18+n34+n50=C$ ... d18;	$n2+n6+n10+n14=C$ ... d34;
$n9+n10+n11+n12=C$ ... d3;	$n3+n19+n35+n51=C$ ... d19;	$n3+n7+n11+n15=C$ ... d35;
$n13+n14+n15+n16=C$ ... d4;	$n4+n20+n36+n52=C$ ... d20;	$n4+n8+n12+n16=C$ ... d36;
$n17+n18+n19+n20=C$ ... d5;	$n5+n21+n37+n53=C$ ... d21;	$n17+n21+n25+n29=C$ ... d37;
$n21+n22+n23+n24=C$ ... d6;	$n6+n22+n38+n54=C$ ... d22;	$n18+n22+n26+n30=C$ ... d38;
$n25+n26+n27+n28=C$ ... d7;	$n7+n23+n39+n55=C$ ... d23;	$n19+n23+n27+n31=C$ ... d39;
$n29+n30+n31+n32=C$ ... d8;	$n8+n24+n40+n56=C$ ... d24;	$n20+n24+n28+n32=C$ ... d40;
$n33+n34+n35+n36=C$ ... d9;	$n9+n25+n41+n57=C$ ... d25;	$n33+n37+n41+n45=C$ ... d41;
$n37+n38+n39+n40=C$ ... d10;	$n10+n26+n42+n58=C$ ... d26;	$n34+n38+n42+n46=C$ ... d42;
$n41+n42+n43+n44=C$ ... d11;	$n11+n27+n43+n59=C$ ... d27;	$n35+n39+n43+n47=C$ ... d43;
$n45+n46+n47+n48=C$ ... d12;	$n12+n28+n44+n60=C$ ... d28;	$n36+n40+n44+n48=C$ ... d44;
$n49+n50+n51+n52=C$ ... d13;	$n13+n29+n45+n61=C$ ... d29;	$n49+n53+n57+n61=C$ ... d45;
$n53+n54+n55+n56=C$ ... d14;	$n14+n30+n46+n62=C$ ... d30;	$n50+n54+n58+n62=C$ ... d46;
$n57+n58+n59+n60=C$ ... d15;	$n15+n31+n47+n63=C$ ... d31;	$n51+n55+n59+n63=C$ ... d47;
$n61+n62+n63+n64=C$ ... d16;	$n16+n32+n48+n64=C$ ... d32;	$n52+n56+n60+n64=C$ ... d48;



い。非常に厳しい条件だが、それにパスしたもののだけが「完全オイラー方陣」になれる。

完全オイラー方陣でかつ連立型の汎対称立体四方陣は、本当にあるものだろうか。

3. ラテン・ユニットの設計の仕方だが、1つ1つの変数の値を決めて行くときに、いつも7方向から定義する。行・列・柱3本と汎対角線4本だ。どの場合でも、0を3回も4回も使っていないか、1を3回も4回も使っていないかを監視する。最終的に{0, 0, 1, 1}の組み合わせになる可能性がある場合にのみ、次のステップに進むようにする。

それにもう一つ、対称型の補数ペアも、同時に決定して行く。つまり常に8方向に気を配らなければならないわけだ。1つ1つの作業が、複雑きわまりない。

我がプログラムの実際は、次のリストを見ていただくしかない。わかりにくいかも知れないが、辛抱強く読んでほしい。これで何通りのユニットが出来るだろうか。

## [Program List]

```
/** 'Complete Euler Cubes' of Order 4 for the Simultaneous Type: **
/** Both Self-Complementary and Pan-Triagonal;: **
/** Composed by 'New Euler's Method' of Binary System **
/** 'NEulerMC4SmlF.c': Dictated by Kanji Setsuda in 2010; **
/** Recompiled and listed on July 23, 2017 **
/** with macOS Sierra 10.12.6 and Xcode 7.3; **
/**
#include <stdio.h>
/* Global Var. *
short int cnt;
short cntr[2];
short cnt2, cnt3;
short CC;
short u1,u2,u3,u4,u5,u6;
short nm[65], uflg[65];
short d[7][72];
short df[49][2], pd[65][2];
short tlu[33][65];
short mtc[33][33];
short cn1[65], cn43[65];
/* Functions(Sub-Routines) *
void stp01(void), stp02(void), stp03(void), stp04(void);
void stp05(void), stp06(void), stp07(void), stp08(void);
void stp09(void), stp10(void), stp11(void), stp12(void);
void stp13(void), stp14(void), stp15(void), stp16(void);
void stp17(void), stp18(void), stp19(void), stp20(void);
void stp21(void), stp22(void), stp23(void), stp24(void);
void stp25(void), stp26(void), stp27(void), stp28(void);
void stp29(void), stp30(void), stp31(void), stp32(void);
void recordans(void), prc4lu(void);
void pr4lu(short x);
void cmbcmp(void);
void pransc2(void), pr2ans(void);
void pransc1(void), pra4lu(void), pra2lu(void);
void prcn1(void), prcn43(void);
//
/* Main Program *
int main(){
short m,n;
printf("//\n");
printf("/** 'Complete Euler Cubes' of Order 4 for the Simultaneous Type: **\n");
printf("/** Self-Complementary and Pan-Triagonal: Solutions with n1=1;: **\n");
printf("/** Composed by the 'New Euler's Method' of Binary System; **\n");
CC=1; cnt=0;
for(n=0;n<65;n++){nm[n]=0; uflg[n]=0;}
for(m=0;m<48;m++){for(n=0;n<2;n++){df[m][n]=0;}}
for(m=0;m<64;m++){for(n=0;n<2;n++){pd[m][n]=0;}}
```

```

stp01(); /* Begin the Search *
prc4lu();
printf("//\n");
printf("/** List of Compositions:(Part) **\n");
cnt=0; cnt3=0;
cmbcmp();
printf("//\n");
printf("// [Count of Compositions = %d]\n",cnt);
printf("//\n");
printf("// [Counts according to n1]\n");
prcn1();
printf("\n");
printf("// [Counts according to n43]\n");
prcn43();
printf("\n");
printf("// OK!\n");
printf("/** Originally dictated in 2010; Recompiled on July 23, 2017\n");
printf("// by Kanji Setsuda with macOS Sierra 10.12.6 and Xcode 7.3 **\n");
printf("//\n");
return 0;
}
//
/* Begin the Search *
/* Set n1 & n64 *
void stp01(){
short a,b;
for(a=0;a<1;a++){b=CC-a;
if((df[1][a]<2)&&(df[17][a]<2)&&(df[33][a]<2)){
if((pd[1][a]<2)&&(pd[2][a]<2)&&(pd[3][a]<2)&&(pd[4][a]<2)){
if((df[16][b]<2)&&(df[32][b]<2)&&(df[48][b]<2)){
if((pd[1][b]<2)&&(pd[10][b]<2)&&(pd[35][b]<2)&&(pd[44][b]<2)){
nm[1]=a; nm[64]=b;
df[1][a]++; df[17][a]++; df[33][a]++;
pd[1][a]++; pd[2][a]++; pd[3][a]++; pd[4][a]++;
df[16][b]++; df[32][b]++; df[48][b]++;
pd[1][b]++; pd[10][b]++; pd[35][b]++; pd[44][b]++;
stp02();
df[1][a]--; df[17][a]--; df[33][a]--;
pd[1][a]--; pd[2][a]--; pd[3][a]--; pd[4][a]--;
df[16][b]--; df[32][b]--; df[48][b]--;
pd[1][b]--; pd[10][b]--; pd[35][b]--; pd[44][b]--;
}}}}
}
}
/* Set n2 & n63 *
void stp02(){
short a,b;
for(a=1;a>=0;a--){b=CC-a;
if((df[1][a]<2)&&(df[18][a]<2)&&(df[34][a]<2)){
if((pd[5][a]<2)&&(pd[6][a]<2)&&(pd[7][a]<2)&&(pd[8][a]<2)){
if((df[16][b]<2)&&(df[31][b]<2)&&(df[47][b]<2)){
if((pd[6][b]<2)&&(pd[13][b]<2)&&(pd[40][b]<2)&&(pd[47][b]<2)){
nm[2]=a; nm[63]=b;
df[1][a]++; df[18][a]++; df[34][a]++;
pd[5][a]++; pd[6][a]++; pd[7][a]++; pd[8][a]++;
df[16][b]++; df[31][b]++; df[47][b]++;
pd[6][b]++; pd[13][b]++; pd[40][b]++; pd[47][b]++;
stp03();
df[1][a]--; df[18][a]--; df[34][a]--;
pd[5][a]--; pd[6][a]--; pd[7][a]--; pd[8][a]--;
df[16][b]--; df[31][b]--; df[47][b]--;
pd[6][b]--; pd[13][b]--; pd[40][b]--; pd[47][b]--;
}}}}
}
}
}

```

```

/** Set n3 & n62 *
void stp03(){
short a,b;
for(a=0;a<2;a++){b=CC-a;
if((df[1][a]<2)&&(df[19][a]<2)&&(df[35][a]<2)){
if((pd[9][a]<2)&&(pd[10][a]<2)&&(pd[11][a]<2)&&(pd[12][a]<2)){
if((df[16][b]<2)&&(df[30][b]<2)&&(df[46][b]<2)){
if((pd[2][b]<2)&&(pd[9][b]<2)&&(pd[36][b]<2)&&(pd[43][b]<2)){
nm[3]=a; nm[62]=b;
df[1][a]++; df[19][a]++; df[35][a]++;
pd[9][a]++; pd[10][a]++; pd[11][a]++; pd[12][a]++;
df[16][b]++; df[30][b]++; df[46][b]++;
pd[2][b]++; pd[9][b]++; pd[36][b]++; pd[43][b]++;
stp04();
df[1][a]--; df[19][a]--; df[35][a]--;
pd[9][a]--; pd[10][a]--; pd[11][a]--; pd[12][a]--;
df[16][b]--; df[30][b]--; df[46][b]--;
pd[2][b]--; pd[9][b]--; pd[36][b]--; pd[43][b]--;
}}}}
}
}
/** Set n4 & n61 *
void stp04(){
short a,b;
for(a=1;a>=0;a--){b=CC-a;
if((df[1][a]<2)&&(df[20][a]<2)&&(df[36][a]<2)){
if((pd[13][a]<2)&&(pd[14][a]<2)&&(pd[15][a]<2)&&(pd[16][a]<2)){
if((df[16][b]<2)&&(df[29][b]<2)&&(df[45][b]<2)){
if((pd[5][b]<2)&&(pd[14][b]<2)&&(pd[39][b]<2)&&(pd[48][b]<2)){
nm[4]=a; nm[61]=b; cnt2=0;
df[1][a]++; df[20][a]++; df[36][a]++;
pd[13][a]++; pd[14][a]++; pd[15][a]++; pd[16][a]++;
df[16][b]++; df[29][b]++; df[45][b]++;
pd[5][b]++; pd[14][b]++; pd[39][b]++; pd[48][b]++;
stp05();
df[1][a]--; df[20][a]--; df[36][a]--;
pd[13][a]--; pd[14][a]--; pd[15][a]--; pd[16][a]--;
df[16][b]--; df[29][b]--; df[45][b]--;
pd[5][b]--; pd[14][b]--; pd[39][b]--; pd[48][b]--;
}}}}
}
}
/** Set n5 & n60 *
void stp05(){
short a,b;
for(a=1;a>=0;a--){b=CC-a;
if((df[2][a]<2)&&(df[21][a]<2)&&(df[33][a]<2)){
if((pd[17][a]<2)&&(pd[18][a]<2)&&(pd[19][a]<2)&&(pd[20][a]<2)){
if((df[15][b]<2)&&(df[28][b]<2)&&(df[48][b]<2)){
if((pd[19][b]<2)&&(pd[28][b]<2)&&(pd[49][b]<2)&&(pd[58][b]<2)){
nm[5]=a; nm[60]=b;
df[2][a]++; df[21][a]++; df[33][a]++;
pd[17][a]++; pd[18][a]++; pd[19][a]++; pd[20][a]++;
df[15][b]++; df[28][b]++; df[48][b]++;
pd[19][b]++; pd[28][b]++; pd[49][b]++; pd[58][b]++;
stp06();
df[2][a]--; df[21][a]--; df[33][a]--;
pd[17][a]--; pd[18][a]--; pd[19][a]--; pd[20][a]--;
df[15][b]--; df[28][b]--; df[48][b]--;
pd[19][b]--; pd[28][b]--; pd[49][b]--; pd[58][b]--;
}}}}
}
}
/** Set n6 & n59 *

```

```

void stp06(){
short a,b;
for(a=0;a<2;a++){b=CC-a;
if((df[2][a]<2)&&(df[22][a]<2)&&(df[34][a]<2)){
if((pd[21][a]<2)&&(pd[22][a]<2)&&(pd[23][a]<2)&&(pd[24][a]<2)){
if((df[15][b]<2)&&(df[27][b]<2)&&(df[47][b]<2)){
if((pd[24][b]<2)&&(pd[31][b]<2)&&(pd[54][b]<2)&&(pd[61][b]<2)){
nm[6]=a; nm[59]=b;
df[2][a]++; df[22][a]++; df[34][a]++;
pd[21][a]++; pd[22][a]++; pd[23][a]++; pd[24][a]++;
df[15][b]++; df[27][b]++; df[47][b]++;
pd[24][b]++; pd[31][b]++; pd[54][b]++; pd[61][b]++;
stp07();
df[2][a]--; df[22][a]--; df[34][a]--;
pd[21][a]--; pd[22][a]--; pd[23][a]--; pd[24][a]--;
df[15][b]--; df[27][b]--; df[47][b]--;
pd[24][b]--; pd[31][b]--; pd[54][b]--; pd[61][b]--;
}}}}
}
}
/* Set n7 & n58 *
void stp07(){
short a,b;
for(a=1;a>=0;a--){b=CC-a;
if((df[2][a]<2)&&(df[23][a]<2)&&(df[35][a]<2)){
if((pd[25][a]<2)&&(pd[26][a]<2)&&(pd[27][a]<2)&&(pd[28][a]<2)){
if((df[15][b]<2)&&(df[26][b]<2)&&(df[46][b]<2)){
if((pd[20][b]<2)&&(pd[27][b]<2)&&(pd[50][b]<2)&&(pd[57][b]<2)){
nm[7]=a; nm[58]=b;
df[2][a]++; df[23][a]++; df[35][a]++;
pd[25][a]++; pd[26][a]++; pd[27][a]++; pd[28][a]++;
df[15][b]++; df[26][b]++; df[46][b]++;
pd[20][b]++; pd[27][b]++; pd[50][b]++; pd[57][b]++;
stp08();
df[2][a]--; df[23][a]--; df[35][a]--;
pd[25][a]--; pd[26][a]--; pd[27][a]--; pd[28][a]--;
df[15][b]--; df[26][b]--; df[46][b]--;
pd[20][b]--; pd[27][b]--; pd[50][b]--; pd[57][b]--;
}}}}
}
}
/* Set n8 & n57 *
void stp08(){
short a,b;
for(a=0;a<2;a++){b=CC-a;
if((df[2][a]<2)&&(df[24][a]<2)&&(df[36][a]<2)){
if((pd[29][a]<2)&&(pd[30][a]<2)&&(pd[31][a]<2)&&(pd[32][a]<2)){
if((df[15][b]<2)&&(df[25][b]<2)&&(df[45][b]<2)){
if((pd[23][b]<2)&&(pd[32][b]<2)&&(pd[53][b]<2)&&(pd[62][b]<2)){
nm[8]=a; nm[57]=b;
df[2][a]++; df[24][a]++; df[36][a]++;
pd[29][a]++; pd[30][a]++; pd[31][a]++; pd[32][a]++;
df[15][b]++; df[25][b]++; df[45][b]++;
pd[23][b]++; pd[32][b]++; pd[53][b]++; pd[62][b]++;
stp09();
df[2][a]--; df[24][a]--; df[36][a]--;
pd[29][a]--; pd[30][a]--; pd[31][a]--; pd[32][a]--;
df[15][b]--; df[25][b]--; df[45][b]--;
pd[23][b]--; pd[32][b]--; pd[53][b]--; pd[62][b]--;
}}}}
}
}
/* Search Level 2: *
/* Set n9 & n56 *
void stp09(){

```

```

short a,b;
for(a=0;a<2;a++){b=CC-a;
if((df[3][a]<2)&&(df[25][a]<2)&&(df[33][a]<2)){
if((pd[33][a]<2)&&(pd[34][a]<2)&&(pd[35][a]<2)&&(pd[36][a]<2)){
if((df[14][b]<2)&&(df[24][b]<2)&&(df[48][b]<2)){
if((pd[3][b]<2)&&(pd[12][b]<2)&&(pd[33][b]<2)&&(pd[42][b]<2)){
nm[9]=a; nm[56]=b;
df[3][a]++; df[25][a]++; df[33][a]++;
pd[33][a]++; pd[34][a]++; pd[35][a]++; pd[36][a]++;
df[14][b]++; df[24][b]++; df[48][b]++;
pd[3][b]++; pd[12][b]++; pd[33][b]++; pd[42][b]++;
stp10();
df[3][a]--; df[25][a]--; df[33][a]--;
pd[33][a]--; pd[34][a]--; pd[35][a]--; pd[36][a]--;
df[14][b]--; df[24][b]--; df[48][b]--;
pd[3][b]--; pd[12][b]--; pd[33][b]--; pd[42][b]--;
}}}}
}
}
/* Set n10 & n55 *
void stp10(){
short a,b;
for(a=1;a>=0;a--){b=CC-a;
if((df[3][a]<2)&&(df[26][a]<2)&&(df[34][a]<2)){
if((pd[37][a]<2)&&(pd[38][a]<2)&&(pd[39][a]<2)&&(pd[40][a]<2)){
if((df[14][b]<2)&&(df[23][b]<2)&&(df[47][b]<2)){
if((pd[8][b]<2)&&(pd[15][b]<2)&&(pd[38][b]<2)&&(pd[45][b]<2)){
nm[10]=a; nm[55]=b;
df[3][a]++; df[26][a]++; df[34][a]++;
pd[37][a]++; pd[38][a]++; pd[39][a]++; pd[40][a]++;
df[14][b]++; df[23][b]++; df[47][b]++;
pd[8][b]++; pd[15][b]++; pd[38][b]++; pd[45][b]++;
stp11();
df[3][a]--; df[26][a]--; df[34][a]--;
pd[37][a]--; pd[38][a]--; pd[39][a]--; pd[40][a]--;
df[14][b]--; df[23][b]--; df[47][b]--;
pd[8][b]--; pd[15][b]--; pd[38][b]--; pd[45][b]--;
}}}}
}
}
/* Set n11 & n54 *
void stp11(){
short a,b;
for(a=0;a<2;a++){b=CC-a;
if((df[3][a]<2)&&(df[27][a]<2)&&(df[35][a]<2)){
if((pd[41][a]<2)&&(pd[42][a]<2)&&(pd[43][a]<2)&&(pd[44][a]<2)){
if((df[14][b]<2)&&(df[22][b]<2)&&(df[46][b]<2)){
if((pd[4][b]<2)&&(pd[11][b]<2)&&(pd[34][b]<2)&&(pd[41][b]<2)){
nm[11]=a; nm[54]=b;
df[3][a]++; df[27][a]++; df[35][a]++;
pd[41][a]++; pd[42][a]++; pd[43][a]++; pd[44][a]++;
df[14][b]++; df[22][b]++; df[46][b]++;
pd[4][b]++; pd[11][b]++; pd[34][b]++; pd[41][b]++;
stp12();
df[3][a]--; df[27][a]--; df[35][a]--;
pd[41][a]--; pd[42][a]--; pd[43][a]--; pd[44][a]--;
df[14][b]--; df[22][b]--; df[46][b]--;
pd[4][b]--; pd[11][b]--; pd[34][b]--; pd[41][b]--;
}}}}
}
}
/* Set n12 & n53 *
void stp12(){
short a,b;

```



```

for(a=1;a>=0;a--){b=CC-a;
  if((df[3][a]<2)&&(df[28][a]<2)&&(df[36][a]<2)){
    if((pd[45][a]<2)&&(pd[46][a]<2)&&(pd[47][a]<2)&&(pd[48][a]<2)){
      if((df[14][b]<2)&&(df[21][b]<2)&&(df[45][b]<2)){
        if((pd[7][b]<2)&&(pd[16][b]<2)&&(pd[37][b]<2)&&(pd[46][b]<2)){
          nm[12]=a; nm[53]=b;
          df[3][a]++; df[28][a]++; df[36][a]++;
          pd[45][a]++; pd[46][a]++; pd[47][a]++; pd[48][a]++;
          df[14][b]++; df[21][b]++; df[45][b]++;
          pd[7][b]++; pd[16][b]++; pd[37][b]++; pd[46][b]++;
          stp13();
          df[3][a]--; df[28][a]--; df[36][a]--;
          pd[45][a]--; pd[46][a]--; pd[47][a]--; pd[48][a]--;
          df[14][b]--; df[21][b]--; df[45][b]--;
          pd[7][b]--; pd[16][b]--; pd[37][b]--; pd[46][b]--;
        }}}
    }
  }
}
/* Set n13 & n52 *
void stp13(){
  short a,b;
  for(a=1;a>=0;a--){b=CC-a;
    if((df[4][a]<2)&&(df[29][a]<2)&&(df[33][a]<2)){
      if((pd[49][a]<2)&&(pd[50][a]<2)&&(pd[51][a]<2)&&(pd[52][a]<2)){
        if((df[13][b]<2)&&(df[20][b]<2)&&(df[48][b]<2)){
          if((pd[17][b]<2)&&(pd[26][b]<2)&&(pd[51][b]<2)&&(pd[60][b]<2)){
            nm[13]=a; nm[52]=b;
            df[4][a]++; df[29][a]++; df[33][a]++;
            pd[49][a]++; pd[50][a]++; pd[51][a]++; pd[52][a]++;
            df[13][b]++; df[20][b]++; df[48][b]++;
            pd[17][b]++; pd[26][b]++; pd[51][b]++; pd[60][b]++;
            stp14();
            df[4][a]--; df[29][a]--; df[33][a]--;
            pd[49][a]--; pd[50][a]--; pd[51][a]--; pd[52][a]--;
            df[13][b]--; df[20][b]--; df[48][b]--;
            pd[17][b]--; pd[26][b]--; pd[51][b]--; pd[60][b]--;
          }}}
        }
      }
    }
  }
}
/* Set n14 & n51 *
void stp14(){
  short a,b;
  for(a=0;a<2;a++){b=CC-a;
    if((df[4][a]<2)&&(df[30][a]<2)&&(df[34][a]<2)){
      if((pd[53][a]<2)&&(pd[54][a]<2)&&(pd[55][a]<2)&&(pd[56][a]<2)){
        if((df[13][b]<2)&&(df[19][b]<2)&&(df[47][b]<2)){
          if((pd[22][b]<2)&&(pd[29][b]<2)&&(pd[56][b]<2)&&(pd[63][b]<2)){
            nm[14]=a; nm[51]=b;
            df[4][a]++; df[30][a]++; df[34][a]++;
            pd[53][a]++; pd[54][a]++; pd[55][a]++; pd[56][a]++;
            df[13][b]++; df[19][b]++; df[47][b]++;
            pd[22][b]++; pd[29][b]++; pd[56][b]++; pd[63][b]++;
            stp15();
            df[4][a]--; df[30][a]--; df[34][a]--;
            pd[53][a]--; pd[54][a]--; pd[55][a]--; pd[56][a]--;
            df[13][b]--; df[19][b]--; df[47][b]--;
            pd[22][b]--; pd[29][b]--; pd[56][b]--; pd[63][b]--;
          }}}
        }
      }
    }
  }
}
/* Set n15 & n50 *
void stp15(){
  short a,b;
  for(a=1;a>=0;a--){b=CC-a;

```



```

if((df[4][a]<2)&&(df[31][a]<2)&&(df[35][a]<2)){
if((pd[57][a]<2)&&(pd[58][a]<2)&&(pd[59][a]<2)&&(pd[60][a]<2)){
if((df[13][b]<2)&&(df[18][b]<2)&&(df[46][b]<2)){
if((pd[18][b]<2)&&(pd[25][b]<2)&&(pd[52][b]<2)&&(pd[59][b]<2)){
nm[15]=a; nm[50]=b;
df[4][a]++; df[31][a]++; df[35][a]++;
pd[57][a]++; pd[58][a]++; pd[59][a]++; pd[60][a]++;
df[13][b]++; df[18][b]++; df[46][b]++;
pd[18][b]++; pd[25][b]++; pd[52][b]++; pd[59][b]++;
stp16();
df[4][a]--; df[31][a]--; df[35][a]--;
pd[57][a]--; pd[58][a]--; pd[59][a]--; pd[60][a]--;
df[13][b]--; df[18][b]--; df[46][b]--;
pd[18][b]--; pd[25][b]--; pd[52][b]--; pd[59][b]--;
}}}}
}
}
/* Set n16 & n49 *
void stp16(){
short a,b;
for(a=0;a<2;a++){b=CC-a;
if((df[4][a]<2)&&(df[32][a]<2)&&(df[36][a]<2)){
if((pd[61][a]<2)&&(pd[62][a]<2)&&(pd[63][a]<2)&&(pd[64][a]<2)){
if((df[13][b]<2)&&(df[17][b]<2)&&(df[45][b]<2)){
if((pd[21][b]<2)&&(pd[30][b]<2)&&(pd[55][b]<2)&&(pd[64][b]<2)){
nm[16]=a; nm[49]=b;
df[4][a]++; df[32][a]++; df[36][a]++;
pd[61][a]++; pd[62][a]++; pd[63][a]++; pd[64][a]++;
df[13][b]++; df[17][b]++; df[45][b]++;
pd[21][b]++; pd[30][b]++; pd[55][b]++; pd[64][b]++;
stp17();
df[4][a]--; df[32][a]--; df[36][a]--;
pd[61][a]--; pd[62][a]--; pd[63][a]--; pd[64][a]--;
df[13][b]--; df[17][b]--; df[45][b]--;
pd[21][b]--; pd[30][b]--; pd[55][b]--; pd[64][b]--;
}}}}
}
}
/* Search Level 3: *
/* Set n17 & n48 *
void stp17(){
short a,b;
for(a=1;a>=0;a--){b=CC-a;
if((df[5][a]<2)&&(df[17][a]<2)&&(df[37][a]<2)){
if((pd[24][a]<2)&&(pd[31][a]<2)&&(pd[54][a]<2)&&(pd[61][a]<2)){
if((df[12][b]<2)&&(df[32][b]<2)&&(df[44][b]<2)){
if((pd[21][b]<2)&&(pd[22][b]<2)&&(pd[23][b]<2)&&(pd[24][b]<2)){
nm[17]=a; nm[48]=b;
df[5][a]++; df[17][a]++; df[37][a]++;
pd[24][a]++; pd[31][a]++; pd[54][a]++; pd[61][a]++;
df[12][b]++; df[32][b]++; df[44][b]++;
pd[21][b]++; pd[22][b]++; pd[23][b]++; pd[24][b]++;
stp18();
df[5][a]--; df[17][a]--; df[37][a]--;
pd[24][a]--; pd[31][a]--; pd[54][a]--; pd[61][a]--;
df[12][b]--; df[32][b]--; df[44][b]--;
pd[21][b]--; pd[22][b]--; pd[23][b]--; pd[24][b]--;
}}}}
}
}
/* Set n18 & n47 *
void stp18(){
short a,b;
for(a=0;a<2;a++){b=CC-a;
if((df[5][a]<2)&&(df[18][a]<2)&&(df[38][a]<2)){

```

```

if((pd[19][a]<2)&&(pd[28][a]<2)&&(pd[49][a]<2)&&(pd[58][a]<2)){
  if((df[12][b]<2)&&(df[31][b]<2)&&(df[43][b]<2)){
    if((pd[17][b]<2)&&(pd[18][b]<2)&&(pd[19][b]<2)&&(pd[20][b]<2)){
      nm[18]=a; nm[47]=b;
      df[5][a]++; df[18][a]++; df[38][a]++;
      pd[19][a]++; pd[28][a]++; pd[49][a]++; pd[58][a]++;
      df[12][b]++; df[31][b]++; df[43][b]++;
      pd[17][b]++; pd[18][b]++; pd[19][b]++; pd[20][b]++;
      stp19();
      df[5][a]--; df[18][a]--; df[38][a]--;
      pd[19][a]--; pd[28][a]--; pd[49][a]--; pd[58][a]--;
      df[12][b]--; df[31][b]--; df[43][b]--;
      pd[17][b]--; pd[18][b]--; pd[19][b]--; pd[20][b]--;
    }
  }
}
}
/* Set n19 & n46 *
void stp19(){
  short a,b;
  for(a=1;a>=0;a--){b=CC-a;
    if((df[5][a]<2)&&(df[19][a]<2)&&(df[39][a]<2)){
      if((pd[23][a]<2)&&(pd[32][a]<2)&&(pd[53][a]<2)&&(pd[62][a]<2)){
        if((df[12][b]<2)&&(df[30][b]<2)&&(df[42][b]<2)){
          if((pd[29][b]<2)&&(pd[30][b]<2)&&(pd[31][b]<2)&&(pd[32][b]<2)){
            nm[19]=a; nm[46]=b;
            df[5][a]++; df[19][a]++; df[39][a]++;
            pd[23][a]++; pd[32][a]++; pd[53][a]++; pd[62][a]++;
            df[12][b]++; df[30][b]++; df[42][b]++;
            pd[29][b]++; pd[30][b]++; pd[31][b]++; pd[32][b]++;
            stp20();
            df[5][a]--; df[19][a]--; df[39][a]--;
            pd[23][a]--; pd[32][a]--; pd[53][a]--; pd[62][a]--;
            df[12][b]--; df[30][b]--; df[42][b]--;
            pd[29][b]--; pd[30][b]--; pd[31][b]--; pd[32][b]--;
          }
        }
      }
    }
  }
}
/* Set n20 & n45 *
void stp20(){
  short a,b;
  for(a=0;a<2;a++){b=CC-a;
    if((df[5][a]<2)&&(df[20][a]<2)&&(df[40][a]<2)){
      if((pd[20][a]<2)&&(pd[27][a]<2)&&(pd[50][a]<2)&&(pd[57][a]<2)){
        if((df[12][b]<2)&&(df[29][b]<2)&&(df[41][b]<2)){
          if((pd[25][b]<2)&&(pd[26][b]<2)&&(pd[27][b]<2)&&(pd[28][b]<2)){
            nm[20]=a; nm[45]=b;
            df[5][a]++; df[20][a]++; df[40][a]++;
            pd[20][a]++; pd[27][a]++; pd[50][a]++; pd[57][a]++;
            df[12][b]++; df[29][b]++; df[41][b]++;
            pd[25][b]++; pd[26][b]++; pd[27][b]++; pd[28][b]++;
            stp21();
            df[5][a]--; df[20][a]--; df[40][a]--;
            pd[20][a]--; pd[27][a]--; pd[50][a]--; pd[57][a]--;
            df[12][b]--; df[29][b]--; df[41][b]--;
            pd[25][b]--; pd[26][b]--; pd[27][b]--; pd[28][b]--;
          }
        }
      }
    }
  }
}
/* Set n21 & n44 *
void stp21(){
  short a,b;
  for(a=0;a<2;a++){b=CC-a;
    if((df[6][a]<2)&&(df[21][a]<2)&&(df[37][a]<2)){
      if((pd[6][a]<2)&&(pd[13][a]<2)&&(pd[40][a]<2)&&(pd[47][a]<2)){

```

```

if((df[11][b]<2)&&(df[28][b]<2)&&(df[44][b]<2)){
  if((pd[5][b]<2)&&(pd[6][b]<2)&&(pd[7][b]<2)&&(pd[8][b]<2)){
    nm[21]=a; nm[44]=b;
    df[6][a]++; df[21][a]++; df[37][a]++;
    pd[6][a]++; pd[13][a]++; pd[40][a]++; pd[47][a]++;
    df[11][b]++; df[28][b]++; df[44][b]++;
    pd[5][b]++; pd[6][b]++; pd[7][b]++; pd[8][b]++;
    stp22();
    df[6][a]--; df[21][a]--; df[37][a]--;
    pd[6][a]--; pd[13][a]--; pd[40][a]--; pd[47][a]--;
    df[11][b]--; df[28][b]--; df[44][b]--;
    pd[5][b]--; pd[6][b]--; pd[7][b]--; pd[8][b]--;
  }
}
}
/* Set n22 & n43 *
void stp22(){
  short a,b;
  for(a=1;a>=0;a--){b=CC-a;
    if((df[6][a]<2)&&(df[22][a]<2)&&(df[38][a]<2)){
      if((pd[1][a]<2)&&(pd[10][a]<2)&&(pd[35][a]<2)&&(pd[44][a]<2)){
        if((df[11][b]<2)&&(df[27][b]<2)&&(df[43][b]<2)){
          if((pd[1][b]<2)&&(pd[2][b]<2)&&(pd[3][b]<2)&&(pd[4][b]<2)){
            nm[22]=a; nm[43]=b;
            df[6][a]++; df[22][a]++; df[38][a]++;
            pd[1][a]++; pd[10][a]++; pd[35][a]++; pd[44][a]++;
            df[11][b]++; df[27][b]++; df[43][b]++;
            pd[1][b]++; pd[2][b]++; pd[3][b]++; pd[4][b]++;
            stp23();
            df[6][a]--; df[22][a]--; df[38][a]--;
            pd[1][a]--; pd[10][a]--; pd[35][a]--; pd[44][a]--;
            df[11][b]--; df[27][b]--; df[43][b]--;
            pd[1][b]--; pd[2][b]--; pd[3][b]--; pd[4][b]--;
          }
        }
      }
    }
  }
}
/* Set n23 & n42 *
void stp23(){
  short a,b;
  for(a=0;a<2;a++){b=CC-a;
    if((df[6][a]<2)&&(df[23][a]<2)&&(df[39][a]<2)){
      if((pd[5][a]<2)&&(pd[14][a]<2)&&(pd[39][a]<2)&&(pd[48][a]<2)){
        if((df[11][b]<2)&&(df[26][b]<2)&&(df[42][b]<2)){
          if((pd[13][b]<2)&&(pd[14][b]<2)&&(pd[15][b]<2)&&(pd[16][b]<2)){
            nm[23]=a; nm[42]=b;
            df[6][a]++; df[23][a]++; df[39][a]++;
            pd[5][a]++; pd[14][a]++; pd[39][a]++; pd[48][a]++;
            df[11][b]++; df[26][b]++; df[42][b]++;
            pd[13][b]++; pd[14][b]++; pd[15][b]++; pd[16][b]++;
            stp24();
            df[6][a]--; df[23][a]--; df[39][a]--;
            pd[5][a]--; pd[14][a]--; pd[39][a]--; pd[48][a]--;
            df[11][b]--; df[26][b]--; df[42][b]--;
            pd[13][b]--; pd[14][b]--; pd[15][b]--; pd[16][b]--;
          }
        }
      }
    }
  }
}
/* Set n24 & n41 *
void stp24(){
  short a,b;
  for(a=1;a>=0;a--){b=CC-a;
    if((df[6][a]<2)&&(df[24][a]<2)&&(df[40][a]<2)){
      if((pd[2][a]<2)&&(pd[9][a]<2)&&(pd[36][a]<2)&&(pd[43][a]<2)){
        if((df[11][b]<2)&&(df[25][b]<2)&&(df[41][b]<2)){

```

```

        if((pd[9][b]<2)&&(pd[10][b]<2)&&(pd[11][b]<2)&&(pd[12][b]<2)){
            nm[24]=a; nm[41]=b;
            df[6][a]++; df[24][a]++; df[40][a]++;
            pd[2][a]++; pd[9][a]++; pd[36][a]++; pd[43][a]++;
            df[11][b]++; df[25][b]++; df[41][b]++;
            pd[9][b]++; pd[10][b]++; pd[11][b]++; pd[12][b]++;
            stp25();
            df[6][a]--; df[24][a]--; df[40][a]--;
            pd[2][a]--; pd[9][a]--; pd[36][a]--; pd[43][a]--;
            df[11][b]--; df[25][b]--; df[41][b]--;
            pd[9][b]--; pd[10][b]--; pd[11][b]--; pd[12][b]--;
        }}}
    }
}
/* Search Level 4: *
/* Set n25 & n40 *
void stp25(){
    short a,b;
    for(a=1;a>=0;a--){b=CC-a;
        if((df[7][a]<2)&&(df[25][a]<2)&&(df[37][a]<2)){
            if((pd[22][a]<2)&&(pd[29][a]<2)&&(pd[56][a]<2)&&(pd[63][a]<2)){
                if((df[10][b]<2)&&(df[24][b]<2)&&(df[44][b]<2)){
                    if((pd[53][b]<2)&&(pd[54][b]<2)&&(pd[55][b]<2)&&(pd[56][b]<2)){
                        nm[25]=a; nm[40]=b;
                        df[7][a]++; df[25][a]++; df[37][a]++;
                        pd[22][a]++; pd[29][a]++; pd[56][a]++; pd[63][a]++;
                        df[10][b]++; df[24][b]++; df[44][b]++;
                        pd[53][b]++; pd[54][b]++; pd[55][b]++; pd[56][b]++;
                        stp26();
                        df[7][a]--; df[25][a]--; df[37][a]--;
                        pd[22][a]--; pd[29][a]--; pd[56][a]--; pd[63][a]--;
                        df[10][b]--; df[24][b]--; df[44][b]--;
                        pd[53][b]--; pd[54][b]--; pd[55][b]--; pd[56][b]--;
                    }}}
                }
            }
        }
}
/* Set n26 & n39 *
void stp26(){
    short a,b;
    for(a=0;a<2;a++){b=CC-a;
        if((df[7][a]<2)&&(df[26][a]<2)&&(df[38][a]<2)){
            if((pd[17][a]<2)&&(pd[26][a]<2)&&(pd[51][a]<2)&&(pd[60][a]<2)){
                if((df[10][b]<2)&&(df[23][b]<2)&&(df[43][b]<2)){
                    if((pd[49][b]<2)&&(pd[50][b]<2)&&(pd[51][b]<2)&&(pd[52][b]<2)){
                        nm[26]=a; nm[39]=b;
                        df[7][a]++; df[26][a]++; df[38][a]++;
                        pd[17][a]++; pd[26][a]++; pd[51][a]++; pd[60][a]++;
                        df[10][b]++; df[23][b]++; df[43][b]++;
                        pd[49][b]++; pd[50][b]++; pd[51][b]++; pd[52][b]++;
                        stp27();
                        df[7][a]--; df[26][a]--; df[38][a]--;
                        pd[17][a]--; pd[26][a]--; pd[51][a]--; pd[60][a]--;
                        df[10][b]--; df[23][b]--; df[43][b]--;
                        pd[49][b]--; pd[50][b]--; pd[51][b]--; pd[52][b]--;
                    }}}
                }
            }
        }
}
}
/* Set n27 & n38 *
void stp27(){
    short a,b;
    for(a=1;a>=0;a--){b=CC-a;
        if((df[7][a]<2)&&(df[27][a]<2)&&(df[39][a]<2)){
            if((pd[21][a]<2)&&(pd[30][a]<2)&&(pd[55][a]<2)&&(pd[64][a]<2)){
                if((df[10][b]<2)&&(df[22][b]<2)&&(df[42][b]<2)){
                    if((pd[61][b]<2)&&(pd[62][b]<2)&&(pd[63][b]<2)&&(pd[64][b]<2)){

```

```

        nm[27]=a; nm[38]=b;
        df[7][a]++; df[27][a]++; df[39][a]++;
        pd[21][a]++; pd[30][a]++; pd[55][a]++; pd[64][a]++;
        df[10][b]++; df[22][b]++; df[42][b]++;
        pd[61][b]++; pd[62][b]++; pd[63][b]++; pd[64][b]++;
        stp28();
        df[7][a]--; df[27][a]--; df[39][a]--;
        pd[21][a]--; pd[30][a]--; pd[55][a]--; pd[64][a]--;
        df[10][b]--; df[22][b]--; df[42][b]--;
        pd[61][b]--; pd[62][b]--; pd[63][b]--; pd[64][b]--;
    }
}
}
/* Set n28 & n37 *
void stp28(){
    short a,b;
    for(a=0;a<2;a++){b=CC-a;
        if((df[7][a]<2)&&(df[28][a]<2)&&(df[40][a]<2)){
            if((pd[18][a]<2)&&(pd[25][a]<2)&&(pd[52][a]<2)&&(pd[59][a]<2)){
                if((df[10][b]<2)&&(df[21][b]<2)&&(df[41][b]<2)){
                    if((pd[57][b]<2)&&(pd[58][b]<2)&&(pd[59][b]<2)&&(pd[60][b]<2)){
                        nm[28]=a; nm[37]=b;
                        df[7][a]++; df[28][a]++; df[40][a]++;
                        pd[18][a]++; pd[25][a]++; pd[52][a]++; pd[59][a]++;
                        df[10][b]++; df[21][b]++; df[41][b]++;
                        pd[57][b]++; pd[58][b]++; pd[59][b]++; pd[60][b]++;
                        stp29();
                        df[7][a]--; df[28][a]--; df[40][a]--;
                        pd[18][a]--; pd[25][a]--; pd[52][a]--; pd[59][a]--;
                        df[10][b]--; df[21][b]--; df[41][b]--;
                        pd[57][b]--; pd[58][b]--; pd[59][b]--; pd[60][b]--;
                    }
                }
            }
        }
    }
}
}
/* Set n29 & n36 *
void stp29(){
    short a,b;
    for(a=0;a<2;a++){b=CC-a;
        if((df[8][a]<2)&&(df[29][a]<2)&&(df[37][a]<2)){
            if((pd[8][a]<2)&&(pd[15][a]<2)&&(pd[38][a]<2)&&(pd[45][a]<2)){
                if((df[9][b]<2)&&(df[20][b]<2)&&(df[44][b]<2)){
                    if((pd[37][b]<2)&&(pd[38][b]<2)&&(pd[39][b]<2)&&(pd[40][b]<2)){
                        nm[29]=a; nm[36]=b;
                        df[8][a]++; df[29][a]++; df[37][a]++;
                        pd[8][a]++; pd[15][a]++; pd[38][a]++; pd[45][a]++;
                        df[9][b]++; df[20][b]++; df[44][b]++;
                        pd[37][b]++; pd[38][b]++; pd[39][b]++; pd[40][b]++;
                        stp30();
                        df[8][a]--; df[29][a]--; df[37][a]--;
                        pd[8][a]--; pd[15][a]--; pd[38][a]--; pd[45][a]--;
                        df[9][b]--; df[20][b]--; df[44][b]--;
                        pd[37][b]--; pd[38][b]--; pd[39][b]--; pd[40][b]--;
                    }
                }
            }
        }
    }
}
}
/* Set n30 & n35 *
void stp30(){
    short a,b;
    for(a=1;a>=0;a--){b=CC-a;
        if((df[8][a]<2)&&(df[30][a]<2)&&(df[38][a]<2)){
            if((pd[3][a]<2)&&(pd[12][a]<2)&&(pd[33][a]<2)&&(pd[42][a]<2)){
                if((df[9][b]<2)&&(df[19][b]<2)&&(df[43][b]<2)){
                    if((pd[33][b]<2)&&(pd[34][b]<2)&&(pd[35][b]<2)&&(pd[36][b]<2)){
                        nm[30]=a; nm[35]=b;
                    }
                }
            }
        }
    }
}
}

```

```

        df[8][a]++; df[30][a]++; df[38][a]++;
        pd[3][a]++; pd[12][a]++; pd[33][a]++; pd[42][a]++;
        df[9][b]++; df[19][b]++; df[43][b]++;
        pd[33][b]++; pd[34][b]++; pd[35][b]++; pd[36][b]++;
        stp31();
        df[8][a]--; df[30][a]--; df[38][a]--;
        pd[3][a]--; pd[12][a]--; pd[33][a]--; pd[42][a]--;
        df[9][b]--; df[19][b]--; df[43][b]--;
        pd[33][b]--; pd[34][b]--; pd[35][b]--; pd[36][b]--;
    }
}
}
/* Set n31 & n34 *
void stp31(){
    short a,b;
    for(a=0;a<2;a++){b=CC-a;
        if((df[8][a]<2)&&(df[31][a]<2)&&(df[39][a]<2)){
            if((pd[7][a]<2)&&(pd[16][a]<2)&&(pd[37][a]<2)&&(pd[46][a]<2)){
                if((df[9][b]<2)&&(df[18][b]<2)&&(df[42][b]<2)){
                    if((pd[45][b]<2)&&(pd[46][b]<2)&&(pd[47][b]<2)&&(pd[48][b]<2)){
                        nm[31]=a; nm[34]=b;
                        df[8][a]++; df[31][a]++; df[39][a]++;
                        pd[7][a]++; pd[16][a]++; pd[37][a]++; pd[46][a]++;
                        df[9][b]++; df[18][b]++; df[42][b]++;
                        pd[45][b]++; pd[46][b]++; pd[47][b]++; pd[48][b]++;
                        stp32();
                        df[8][a]--; df[31][a]--; df[39][a]--;
                        pd[7][a]--; pd[16][a]--; pd[37][a]--; pd[46][a]--;
                        df[9][b]--; df[18][b]--; df[42][b]--;
                        pd[45][b]--; pd[46][b]--; pd[47][b]--; pd[48][b]--;
                    }
                }
            }
        }
    }
}
}
/* Set n32 & n33 *
void stp32(){
    short a,b;
    for(a=1;a>=0;a--){b=CC-a;
        if((df[8][a]<2)&&(df[32][a]<2)&&(df[40][a]<2)){
            if((pd[4][a]<2)&&(pd[11][a]<2)&&(pd[34][a]<2)&&(pd[41][a]<2)){
                if((df[9][b]<2)&&(df[17][b]<2)&&(df[41][b]<2)){
                    if((pd[41][b]<2)&&(pd[42][b]<2)&&(pd[43][b]<2)&&(pd[44][b]<2)){
                        nm[32]=a; nm[33]=b;
                        df[8][a]++; df[32][a]++; df[40][a]++;
                        pd[4][a]++; pd[11][a]++; pd[34][a]++; pd[41][a]++;
                        df[9][b]++; df[17][b]++; df[41][b]++;
                        pd[41][b]++; pd[42][b]++; pd[43][b]++; pd[44][b]++;
                        recordans();
                        df[8][a]--; df[32][a]--; df[40][a]--;
                        pd[4][a]--; pd[11][a]--; pd[34][a]--; pd[41][a]--;
                        df[9][b]--; df[17][b]--; df[41][b]--;
                        pd[41][b]--; pd[42][b]--; pd[43][b]--; pd[44][b]--;
                    }
                }
            }
        }
    }
}
}
//
/* Record the Answer *
void recordans(){
    short n;
    cnt++;
    tlu[cnt-1][0]=cnt;
    for(n=1;n<65;n++){tlu[cnt-1][n]=nm[n];}
}
//
/* Print the Latin Units *
void prc4lu(){

```

```

short m,n,t,l;
t=4;
printf("//\n");
printf("// [Latin Units]\n");
for(m=0;m<cnt;m=m+t){
    for(n=0;n<t;n++){d[n][0]=m+n+1;}
    for(n=1;n<65;n++){for(l=0;l<t;l++){d[l][n]=tlu[m+l][n];}}
    if(m<(cnt-2)){pr4lu(t);}else{pr4lu(2);}
}
printf("// [Count of Latin Units = %d]\n",cnt);
for(m=0;m<cnt;m++){
    for(n=0;n<cnt;n++){
        t=0;
        for(l=1;l<65;l++){if(tlu[m][l]==tlu[n][l]){t++;}}
        mtc[m][n]=t;
    }
}
printf("//\n");
printf("// [Reference Table for Matching Combinations]\n");
printf("// *|");
for(n=0;n<cnt;n++){printf("%3d",n+1);}
printf("\n");
printf("// -+-----\n");
for(m=0;m<cnt;m++){
    printf("//%3d|",m+1);
    for(n=0;n<cnt;n++){
        t=mtc[m][n];
        printf("%3d",t);
    }
    printf("\n");
}
}
//
/* Print 4 Forms *
void pr4lu(short x){
short n;
printf("//%17d/",d[0][0]);
for(n=1;n<x;n++){printf("%20d/",d[n][0]);}
printf("\n//");
printf("%3d----%d----%d----%d",d[0][1],d[0][17],d[0][33],d[0][49]);
for(n=1;n<x;n++){
    printf("%6d----%d----%d----%d",d[n][1],d[n][17],d[n][33],d[n][49]);}
printf("\n//");
printf(" |%d%5d%5d |%d",d[0][2],d[0][18],d[0][34],d[0][50]);
for(n=1;n<x;n++){
    printf(" |%d%5d%5d |%d",d[n][2],d[n][18],d[n][34],d[n][50]);}
printf("\n//");
printf("%3d%2d%3d%2d%3d%2d%3d%2d",
    d[0][5],d[0][3],d[0][21],d[0][19],d[0][37],d[0][35],d[0][53],d[0][51]);
for(n=1;n<x;n++){
    printf("%4d%2d%3d%2d%3d%2d%3d%2d",
    d[n][5],d[n][3],d[n][21],d[n][19],d[n][37],d[n][35],d[n][53],d[n][51]);}
printf("\n//");
printf(" |%d%2d--%d-%d--%d-%d-+%d-%d",
    d[0][6],d[0][4],d[0][22],d[0][20],d[0][38],d[0][36],d[0][54],d[0][52]);
for(n=1;n<x;n++){
    printf(" |%d%2d--%d-%d--%d-%d-+%d-%d",
    d[n][6],d[n][4],d[n][22],d[n][20],d[n][38],d[n][36],d[n][54],d[n][52]);}
printf("\n//");
printf("%3d%2d|%2d%2d%3d%2d%3d%2d|",
    d[0][9],d[0][7],d[0][25],d[0][23],d[0][41],d[0][39],d[0][57],d[0][55]);
for(n=1;n<x;n++){
    printf("%3d%2d|%2d%2d%3d%2d%3d%2d|",
    d[n][9],d[n][7],d[n][25],d[n][23],d[n][41],d[n][39],d[n][57],d[n][55]);}
printf("\n//");

```





```

    }
    }
    }
    }
    }
}
//
/** Print the Compositions *
void pransc2(){
short n;
cnt++; cnt2++;
cn1[nm[1]]++; cn43[nm[43]]++;
if(cnt2==1){
    cnt3[cnt3]=cnt;
    for(n=1;n<65;n++){d[cnt3][n]=nm[n];}
    d[cnt3][65]=u1+1; d[cnt3][66]=u2+1; d[cnt3][67]=u3+1;
    d[cnt3][68]=u4+1; d[cnt3][69]=u5+1; d[cnt3][70]=u6+1;
    cnt3++; if(cnt3==2){pr2ans(); cnt3=0;}
}
}
//
/** Print 2 Answers *
void pr2ans(){
printf("//%5d/%3d/%3d/%3d/%3d/%3d/%8d#",
        d[0][65],d[0][66],d[0][67],d[0][68],d[0][69],d[0][70],cnt3[0]);
printf("%12d/%3d/%3d/%3d/%3d/%3d/%8d#\n//",
        d[1][65],d[1][66],d[1][67],d[1][68],d[1][69],d[1][70],cnt3[1]);
printf("%4d-----%2d-----%2d-----%2d",d[0][1],d[0][17],d[0][33],d[0][49]);
printf("%12d-----%2d-----%2d-----%2d\n//",d[1][1],d[1][17],d[1][33],d[1][49]);
printf("    l%2d %9d %9d    l%2d",
        d[0][2],d[0][18],d[0][34],d[0][50]);
printf("        l%2d %9d %9d        l%2d\n//",
        d[1][2],d[1][18],d[1][34],d[1][50]);
printf("%4d %3d %5d %3d %5d %3d %5d %3d",
        d[0][5],d[0][3],d[0][21],d[0][19],d[0][37],d[0][35],d[0][53],d[0][51]);
printf("%8d %3d %5d %3d %5d %3d %5d %3d\n//",
        d[1][5],d[1][3],d[1][21],d[1][19],d[1][37],d[1][35],d[1][53],d[1][51]);
printf("    l%2d %2d---%2d--%2d---%2d--%2d---%2d---%2d--%2d",
        d[0][6],d[0][4],d[0][22],d[0][20],d[0][38],d[0][36],d[0][54],d[0][52]);
printf("        l%2d %2d---%2d--%2d---%2d--%2d---%2d---%2d--%2d\n//",
        d[1][6],d[1][4],d[1][22],d[1][20],d[1][38],d[1][36],d[1][54],d[1][52]);
printf("%4d %3d l%4d %3d %5d %3d %5d %3d l",
        d[0][9],d[0][7],d[0][25],d[0][23],d[0][41],d[0][39],d[0][57],d[0][55]);
printf("%6d %3d l%4d %3d %5d %3d %5d %3d l\n//",
        d[1][9],d[1][7],d[1][25],d[1][23],d[1][41],d[1][39],d[1][57],d[1][55]);
printf("    l%2d %3d %5d %3d %5d %3d    l%2d %3d",
        d[0][10],d[0][8],d[0][26],d[0][24],d[0][42],d[0][40],d[0][58],d[0][56]);
printf("        l%2d %3d %5d %3d %5d %3d    l%2d %3d\n//",
        d[1][10],d[1][8],d[1][26],d[1][24],d[1][42],d[1][40],d[1][58],d[1][56]);
printf("%4d--%2d-l--%2d--%2d---%2d--%2d---%2d %3d l",
        d[0][13],d[0][11],d[0][29],d[0][27],d[0][45],d[0][43],d[0][61],d[0][59]);
printf("%6d--%2d-l--%2d--%2d---%2d--%2d---%2d %3d l\n//",
        d[1][13],d[1][11],d[1][29],d[1][27],d[1][45],d[1][43],d[1][61],d[1][59]);
printf("%6d %3d %5d %3d %5d %3d %5d %3d",
        d[0][14],d[0][12],d[0][30],d[0][28],d[0][46],d[0][44],d[0][62],d[0][60]);
printf("    %4d %3d %5d %3d %5d %3d %5d %3d\n//",
        d[1][14],d[1][12],d[1][30],d[1][28],d[1][46],d[1][44],d[1][62],d[1][60]);
printf("%8d l%8d %9d %9d l",
        d[0][15],d[0][31],d[0][47],d[0][63]);
printf("%10d l%8d %9d %9d l\n//",
        d[1][15],d[1][31],d[1][47],d[1][63]);
printf("%10d-----%2d-----%2d-----%2d",
        d[0][16],d[0][32],d[0][48],d[0][64]);

```

```

printf("%12d-----%2d-----%2d-----%2d\n",
       d[1][16],d[1][32],d[1][48],d[1][64]);
}
//
/** Print the Compositions *
void pransc1(){
short n;
cnt++; cnt2++;
cn1[nm[1]]++; cn43[nm[43]]++;
if(cnt2==1){
  cntr[0]=cnt;
  d[1][0]=u1+1; d[2][0]=u2+1; d[3][0]=u3+1;
  d[4][0]=u4+1; d[5][0]=u5+1; d[6][0]=u6+1;
  for(n=1;n<65;n++){d[0][n]=nm[n];
    d[1][n]=tlu[u1][n]; d[2][n]=tlu[u2][n];
    d[3][n]=tlu[u3][n]; d[4][n]=tlu[u4][n];
    d[5][n]=tlu[u5][n]; d[6][n]=tlu[u6][n];}
  pra4lu(); pra2lu();
}
}
//
/** Print 4 latin units *
void pra4lu(){
short n;
printf("//\n");
printf("//[Latin Units]%3d/",d[1][0]);
for(n=2;n<5;n++){printf("%20d/",d[n][0]);}
printf("\n//");
printf("%2d----%d----%d----%d",d[1][1],d[1][17],d[1][33],d[1][49]);
for(n=2;n<5;n++){
  printf("%6d----%d----%d----%d",d[n][1],d[n][17],d[n][33],d[n][49]);}
printf("\n//");
printf(" |%d%5d%5d |%d",d[1][2],d[1][18],d[1][34],d[1][50]);
for(n=2;n<5;n++){
  printf(" |%d%5d%5d |%d",d[n][2],d[n][18],d[n][34],d[n][50]);}
printf("\n//");
printf("%2d%2d%3d%2d%3d%2d%3d%2d",
       d[1][5],d[1][3],d[1][21],d[1][19],d[1][37],d[1][35],d[1][53],d[1][51]);
for(n=2;n<5;n++){
  printf("%4d%2d%3d%2d%3d%2d%3d%2d",
       d[n][5],d[n][3],d[n][21],d[n][19],d[n][37],d[n][35],d[n][53],d[n][51]);}
printf("\n//");
printf(" |%d%2d--%d-%d--%d-%d--%d-%d",
       d[1][6],d[1][4],d[1][22],d[1][20],d[1][38],d[1][36],d[1][54],d[1][52]);
for(n=2;n<5;n++){
  printf(" |%d%2d--%d-%d--%d-%d--%d-%d",
       d[n][6],d[n][4],d[n][22],d[n][20],d[n][38],d[n][36],d[n][54],d[n][52]);}
printf("\n//");
printf("%2d%2d|%2d%2d%3d%2d%3d%2d|",
       d[1][9],d[1][7],d[1][25],d[1][23],d[1][41],d[1][39],d[1][57],d[1][55]);
for(n=2;n<5;n++){
  printf("%3d%2d|%2d%2d%3d%2d%3d%2d|",
       d[n][9],d[n][7],d[n][25],d[n][23],d[n][41],d[n][39],d[n][57],d[n][55]);}
printf("\n//");
printf(" |%d%2d%3d%2d%3d%2d |%d%2d",
       d[1][10],d[1][8],d[1][26],d[1][24],d[1][42],d[1][40],d[1][58],d[1][56]);
for(n=2;n<5;n++){
  printf(" |%d%2d%3d%2d%3d%2d |%d%2d",
       d[n][10],d[n][8],d[n][26],d[n][24],d[n][42],d[n][40],d[n][58],d[n][56]);}
printf("\n//");
printf("%2d-%d|-%d-%d--%d-%d--%d%2d|",
       d[1][13],d[1][11],d[1][29],d[1][27],d[1][45],d[1][43],d[1][61],d[1][59]);
for(n=2;n<5;n++){
  printf("%3d-%d|-%d-%d--%d-%d--%d%2d|",
       d[n][13],d[n][11],d[n][29],d[n][27],d[n][45],d[n][43],d[n][61],d[n][59]);}

```

```

printf("\n//");
printf("%3d%2d%3d%2d%3d%2d%3d%2d",
    d[1][14],d[1][12],d[1][30],d[1][28],d[1][46],d[1][44],d[1][62],d[1][60]);
for(n=2;n<5;n++){
    printf(" %3d%2d%3d%2d%3d%2d%3d%2d",
        d[n][14],d[n][12],d[n][30],d[n][28],d[n][46],d[n][44],d[n][62],d[n][60]);}
printf("\n//");
printf("%4d|%4d%5d%5d|",d[1][15],d[1][31],d[1][47],d[1][63]);
for(n=2;n<5;n++){
    printf("%5d|%4d%5d%5d|",d[n][15],d[n][31],d[n][47],d[n][63]);}
printf("\n//");
printf("%5d----%d----%d----%d",d[1][16],d[1][32],d[1][48],d[1][64]);
for(n=2;n<5;n++){
    printf("%6d----%d----%d----%d",d[n][16],d[n][32],d[n][48],d[n][64]);}
printf("\n");
}
//
/* Print Answer with 2 latin units *
void pra2lu(){
    printf("//%16d/%20d/      [Composition]%19d#\n",d[5][0],d[6][0],cntr[0]);
    printf("//%2d----%d----%d----%d",d[5][1],d[5][17],d[5][33],d[5][49]);
    printf("%6d----%d----%d----%d",d[6][1],d[6][17],d[6][33],d[6][49]);
    printf("%8d-----%2d-----%2d-----%2d\n",d[0][1],d[0][17],d[0][33],d[0][49]);
    printf("// %1d%5d%5d %1d",d[5][2],d[5][18],d[5][34],d[5][50]);
    printf(" %1d%5d%5d %1d",d[6][2],d[6][18],d[6][34],d[6][50]);
    printf(" %1d%2d %9d %9d %1d\n",d[0][2],d[0][18],d[0][34],d[0][50]);
    printf("//%2d%2d%3d%2d%3d%2d%3d%2d",
        d[5][5],d[5][3],d[5][21],d[5][19],d[5][37],d[5][35],d[5][53],d[5][51]);
    printf("%4d%2d%3d%2d%3d%2d%3d%2d",
        d[6][5],d[6][3],d[6][21],d[6][19],d[6][37],d[6][35],d[6][53],d[6][51]);
    printf("%6d %3d %5d %3d %5d %3d %5d %3d\n",
        d[0][5],d[0][3],d[0][21],d[0][19],d[0][37],d[0][35],d[0][53],d[0][51]);
    printf("// %1d%2d--%d-%d--%d-%d--%d-%d",
        d[5][6],d[5][4],d[5][22],d[5][20],d[5][38],d[5][36],d[5][54],d[5][52]);
    printf(" %1d%2d--%d-%d--%d-%d--%d-%d",
        d[6][6],d[6][4],d[6][22],d[6][20],d[6][38],d[6][36],d[6][54],d[6][52]);
    printf(" %1d%2d %2d---%2d--%2d---%2d--%2d---%2d--%2d\n",
        d[0][6],d[0][4],d[0][22],d[0][20],d[0][38],d[0][36],d[0][54],d[0][52]);
    printf("//%2d%2d|%2d%2d%3d%2d%3d%2d|",
        d[5][9],d[5][7],d[5][25],d[5][23],d[5][41],d[5][39],d[5][57],d[5][55]);
    printf("%3d%2d|%2d%2d%3d%2d%3d%2d|",
        d[6][9],d[6][7],d[6][25],d[6][23],d[6][41],d[6][39],d[6][57],d[6][55]);
    printf("%5d %3d |%4d %3d %5d %3d %5d %3d |%1d",
        d[0][9],d[0][7],d[0][25],d[0][23],d[0][41],d[0][39],d[0][57],d[0][55]);
    printf("// %1d%2d%3d%2d%3d%2d %1d%2d",
        d[5][10],d[5][8],d[5][26],d[5][24],d[5][42],d[5][40],d[5][58],d[5][56]);
    printf(" %1d%2d%3d%2d%3d%2d %1d%2d",
        d[6][10],d[6][8],d[6][26],d[6][24],d[6][42],d[6][40],d[6][58],d[6][56]);
    printf(" %1d%2d %3d %5d %3d %5d %3d %1d%2d %3d\n",
        d[0][10],d[0][8],d[0][26],d[0][24],d[0][42],d[0][40],d[0][58],d[0][56]);
    printf("//%2d-%d|-%d-%d--%d-%d--%d%2d|",
        d[5][13],d[5][11],d[5][29],d[5][27],d[5][45],d[5][43],d[5][61],d[5][59]);
    printf("%3d-%d|-%d-%d--%d-%d--%d%2d|",
        d[6][13],d[6][11],d[6][29],d[6][27],d[6][45],d[6][43],d[6][61],d[6][59]);
    printf("%5d--%2d-|-%2d--%2d---%2d--%2d---%2d %3d |%1d",
        d[0][13],d[0][11],d[0][29],d[0][27],d[0][45],d[0][43],d[0][61],d[0][59]);
    printf("//%3d%2d%3d%2d%3d%2d%3d%2d",
        d[5][14],d[5][12],d[5][30],d[5][28],d[5][46],d[5][44],d[5][62],d[5][60]);
    printf("%4d%2d%3d%2d%3d%2d%3d%2d",
        d[6][14],d[6][12],d[6][30],d[6][28],d[6][46],d[6][44],d[6][62],d[6][60]);
    printf("%7d %3d %5d %3d %5d %3d %5d %3d\n",
        d[0][14],d[0][12],d[0][30],d[0][28],d[0][46],d[0][44],d[0][62],d[0][60]);
    printf("//%4d|%4d%5d%5d|",d[5][15],d[5][31],d[5][47],d[5][63]);
    printf("%5d|%4d%5d%5d|",d[6][15],d[6][31],d[6][47],d[6][63]);
    printf("%9d %8d %9d %9d |%1d",d[0][15],d[0][31],d[0][47],d[0][63]);
}

```

```

printf("//%5d----%d----%d----%d", d[5][16], d[5][32], d[5][48], d[5][64]);
printf("%6d----%d----%d----%d", d[6][16], d[6][32], d[6][48], d[6][64]);
printf("%11d-----%2d-----%2d-----%2d\n", d[0][16], d[0][32], d[0][48], d[0][64]);
}
//
/** Print the Counts according to n1 *
void prcn1(){
short d,n;
printf("//");
for(n=1;n<41;n++){d=cn1[n];
if(d>0){printf("%4d:%4d,",n,d);}
else{printf("%4d: ...,",n);}
if(n%8==0){printf("\n//");}
}
}
//
/** Print the Counts according to n43 *
void prcn43(){
short d,n;
printf("//");
for(n=1;n<65;n++){d=cn43[n];
if(d>0){printf("%4d:%4d,",n,d);}
else{printf("%4d: ...,",n);}
if(n%8==0){printf("\n//");}
}
}
//

```

リストを見やすくするために、 $n1 = 1$  の標準解のみを出力するようにした。そうすると、ユニットの数も半分になる。解の総数も激減して、扱いやすくなる。

4. 以下のように 16個のユニットが出て来た。この 16個から任意に6個を取って組み合わせ、求める解の二進法分解図とするのである。そして、対応するポジションの数値から次の式で、解を計算して合成する。

$$(((a*2+b)*2+c)*2+d)*2+e)*2+f+1 = nN \quad (\text{古典的表記にしたいので、} 1 \text{ を加える})$$

**\* 新オイラー法による連立型汎対称立体四方陣 \***

\* 専用のラテン・ユニット( $n1=0$ のもの) \*

[Latin Units]

1/		2/		3/		4/	
0---1---0---1	0---0---1---1	0---1---0---1	0---0---1---1	0---1---0---1	0---0---1---1	0---1---0---1	0---0---1---1
1 0 1  0	1 1 0  0	1 1 0  0	1 1 0  0	1 1 0  0	1 0 1  0	1 1 0  0	1 0 1  0
1 0 0 1 1 0 0 1	1 0 1 0 0 1 0 1	1 0 1 0 0 1 0 1	0 0 1 1 0 0 1 1	0 0 1 1 0 0 1 1	0 0 1 1 0 0 1 1	0 0 1 1 0 0 1 1	0 0 1 1 0 0 1 1
0 1-1-0-0-1+1-0	0 1-0-1-1-0+1-0	1 1-0-0-1-1+0-0	0 1-0-1-1-0+1-0	1 1-0-0-1-1+0-0	0 1-0-1-1-0+1-0	1 1-0-0-1-1+0-0	0 1-0-1-1-0+1-0
0 1  1 0 0 1 1 0	0 1  0 1 1 0 1 0	0 0  0 1 1 0 1 1	1 1  0 1 1 0 0 0	0 0  0 1 1 0 1 1	1 1  0 1 1 0 0 0	1 1  0 1 1 0 0 0	1 1  0 1 1 0 0 0
1 0 0 1 1 0  0 1	1 0 1 0 0 1  0 1	0 0 1 0 0 1  1 1	1 1 1 0 0 0  0 0	0 0 1 0 0 1  1 1	1 1 1 0 0 0  0 0	1 1 1 0 0 0  0 0	1 1 1 0 0 0  0 0
1-0 -0-1-1-0-0 1	1-0 -1-0-0-1-0 1	1-1 -0-0-1-1-0 0	1-0 -1-0-0-1-0 1	1-1 -0-0-1-1-0 0	1-0 -1-0-0-1-0 1	1-1 -0-0-1-1-0 0	1-0 -1-0-0-1-0 1
0 1 1 0 0 1 1 0	0 1 0 1 1 0 1 0	0 1 0 1 1 0 1 0	0 0 1 1 0 0 1 1	0 1 0 1 1 0 1 0	0 0 1 1 0 0 1 1	0 1 0 1 1 0 1 0	0 0 1 1 0 0 1 1
1  0 1 0	1  1 0 0	1  1 0 0	1  0 1 0	1  1 0 0	1  0 1 0	1  1 0 0	1  0 1 0
0---1---0---1	0---0---1---1	0---1---0---1	0---0---1---1	0---1---0---1	0---0---1---1	0---1---0---1	0---0---1---1

5/		6/		7/		8/	
0---1---0---1	0---0---1---1	0---1---1---0	0---0---1---1	0---1---1---0	0---0---1---1	0---1---1---0	0---0---1---1
1 0 1  0	1 1 0  0	1 1 0  0	1 1 0  0	1 1 0  0	1 0 0  1	1 1 0  0	1 0 0  1
0 0 1 1 0 0 1 1	1 1 1 1 0 0 0 0	1 1 1 0 0 0 0 1	0 1 1 1 0 0 0 0	1 1 1 0 0 0 0 1	0 1 1 1 0 0 0 0	1 1 1 0 0 0 0 1	0 1 1 1 0 0 0 0
1 1-0-0-1-1+0-0	0 0-0-0-1-1+1-1	1 0-0-0-0-1+1-1	0 0-0-0-1-1+1-1	1 0-0-0-0-1+1-1	0 0-0-0-1-1+1-1	1 0-0-0-0-1+1-1	0 0-0-0-1-1+1-1
1 0  0 1 1 0 0 1	1 0  1 0 0 1 0 1	1 0  0 0 0 1 1 1	1 1  1 0 0 0 0 1	1 0  0 0 0 1 1 1	1 1  1 0 0 0 0 1	1 1  1 0 0 0 0 1	1 1  1 0 0 0 0 1
0 1 1 0 0 1  1 0	0 1 0 1 1 0  1 0	0 0 0 1 1 1  1 0	0 1 1 1 1 0  0 0	0 0 0 1 1 1  1 0	0 1 1 1 1 0  0 0	0 1 1 1 1 0  0 0	0 1 1 1 1 0  0 0
1-1 -0-0-1-1-0 0	0-0 -0-0-1-1-1 1	0-0 -0-1-1-1-1 0	1-0 -0-0-0-1-1 1	0-0 -0-1-1-1-1 0	1-0 -0-0-0-1-1 1	1-0 -0-0-0-1-1 1	1-0 -0-0-0-1-1 1
0 0 1 1 0 0 1 1	1 1 1 1 0 0 0 0	0 1 1 1 1 0 0 0	1 1 1 1 0 0 0 0	0 1 1 1 1 0 0 0	1 1 1 0 0 0 0 1	1 1 1 0 0 0 0 1	1 1 1 0 0 0 0 1
1  0 1 0	1  1 0 0	1  1 0 0	1  1 0 0	1  1 0 0	0  1 1 0	0  1 1 0	0  1 1 0
0---1---0---1	0---0---1---1	1---0---0---1	0---0---1---1	1---0---0---1	0---0---1---1	1---0---0---1	0---0---1---1

9/ 0---1---1---0  1 0 0  1 0 1 1 0 1 0 0 1  1 0-0-1-0-1+1-0 1 1  0 0 0 1 1   0 0 1 1 1 1 0 0 1-0 -0-1-0-1-1 0  0 1 1 0 1 0 0 1 0  1 1 0  1---0---0---1	10/ 0---0---1---1  0 1 0  1 1 1 0 0 1 1 0 0  0 1-0-1-1-0+1-0 0 1  1 1 0 0 1 0   1 0 1 1 0 0 0 1 1-0 -1-0-0-1-0 1  1 1 0 0 1 1 0 0 0  1 0 1  0---0---1---1	11/ 0---0---1---1  0 1 1  0 1 1 0 1 0 0 1 0  0 1-0-0-1-0+1-1 1 0  1 1 0 1 0 0   1 1 0 1 0 0 1 0 0-0 -1-0-1-1-0 1  1 0 1 1 0 1 0 0 1  0 0 1  0---0---1---1	12/ 0---1---0---1  0 1 0  1 1 1 0 0 1 1 0 0  1 1-0-0-1-1+0-0 1 1  0 0 1 1 1 1   0 0 1 1 0 0 1 1 1-1 -0-0-1-1-0 0  1 1 0 0 1 1 0 0 0  1 0 1  0---1---0---1
13/ 0---1---1---0  0 1 1  0 1 1 0 0 0 0 1 1  1 1-0-0-0-0+1-1 1 0  0 1 0 1 1 0   1 0 0 1 0 1 1 0 0-0 -1-1-1-1-0 0  0 0 1 1 1 1 0 0 1  0 0 1  1---0---0---1	14/ 0---0---1---1  0 0 1  1 0 1 0 1 1 0 1 0  0 1-0-1-1-0+1-0 1 1  1 1 0 0 0 0   1 1 1 1 0 0 0 0 1-0 -1-0-0-1-0 1  1 0 1 0 0 1 0 1 0  0 0 1 1  0---0---1---1	15/ 0---1---1---0  0 0 1  1 0 1 0 0 1 0 1 1  1 1-0-1-0-0+1-0 1 1  0 1 0 0 1 0   1 0 1 1 0 1 0 0 1-0 -1-1-0-1-0 0  0 0 1 0 1 1 0 1 0  0 1 1  1---0---0---1	16/ 0---1---0---1  0 0 1  1 0 1 0 1 1 0 1 0  1 1-0-0-1-1+0-0 1 0  1 1 0 0 0 1   0 1 1 1 0 0 1 0 1-1 -0-0-1-1-0 0  1 0 1 0 0 1 0 1 0  0 1 1  0---1---0---1

[Count of Latin Units = 16]

[最良の組み合わせを見つけるための参照表]

*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	64	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32
2	32	64	48	48	32	32	32	32	32	48	32	32	32	32	32	16
3	32	48	64	32	48	32	40	24	32	32	24	48	32	16	24	32
4	32	48	32	64	48	32	24	40	32	32	40	16	32	48	40	32
5	32	32	48	48	64	32	32	32	32	16	32	32	32	32	32	48
6	32	32	32	32	32	64	48	48	32	32	48	32	32	32	16	32
7	32	32	40	24	32	48	64	32	48	24	32	32	48	16	32	24
8	32	32	24	40	32	48	32	64	48	40	32	32	16	48	32	40
9	32	32	32	32	32	32	48	48	64	32	16	32	32	32	48	32
10	32	48	32	32	16	32	24	40	32	64	40	48	32	48	40	32
11	32	32	24	40	32	48	32	32	16	40	64	32	48	48	32	40
12	32	32	48	16	32	32	32	32	32	48	32	64	32	32	32	48
13	32	32	32	32	32	32	48	16	32	32	48	32	64	32	48	32
14	32	32	16	48	32	32	16	48	32	48	48	32	32	64	48	48
15	32	32	24	40	32	16	32	32	48	40	32	32	48	48	64	40
16	32	16	32	32	48	32	24	40	32	32	40	48	32	48	40	64

\* 新オイラー法によって構成した連立型汎対称立体四方陣の解 (n1=1) \*

\*\* 'Complete Euler Cubes' of Order 4 for the Simultaneous Type: \*\*  
 \*\* Self-Complementary and Pan-Triagonal: Solutions with n1=1: \*\*  
 \*\* Composed by our 'New Euler's Method' of Binary System; \*\*  
 \*\* List of Compositions: (Part) \*\*

[Latin Units] 1/ 0---1---0---1  1 0 1  0 1 0 0 1 1 0 0 1  0 1-1-0-0-1-1-0 0 1  1 0 0 1 1 0   1 0 0 1 1 0 0 1 1-0 -0-1-1-0-0 1  0 1 1 0 0 1 1 0 1  0 1 0  0---1---0---1	2/ 0---0---1---1  1 1 0  0 1 0 1 0 0 1 0 1  0 1-0-1-1-0-1-0 0 1  0 1 1 0 1 0   1 0 1 0 0 1 0 1 1-0 -1-0-0-1-0 1  0 1 0 1 1 0 1 0 1  1 1 0 0  0---0---1---1	5/ 0---1---0---1  1 0 1  0 0 0 1 1 0 0 1 1  1 1-0-0-1-1-0-0 1 0  0 1 1 0 0 1   0 1 1 0 0 1 1 0 1-1 -0-0-1-1-0 0  0 0 1 1 0 0 1 1 1  0 1 0  0---1---0---1	7/ 0---1---1---0  1 1 0  0 1 1 1 0 0 0 0 1  1 0-0-0-0-1-1-1 1 0  0 0 0 1 1 1   0 0 0 1 1 1 1 0 0-0 -0-1-1-1-1 0  0 1 1 1 1 0 0 0 1  1 0 0 0  1---0---0---1
--	--	--	--

15/ 0---1---1---0  0 0 1  1 0 1 0 0 1 0 1 1  1 1-0-1-0-0-1-0 1 1  0 1 0 0 1 0   1 0 1 1 0 1  0 0 1-0 -1-1-0-1-0 0  0 0 1 0 1 1 0 1 0  0 1 1   1---0---0---1	11/ 0---0---1---1  0 1 1  0 1 1 0 1 0 0 1 0  0 1-0-0-1-0-1-1 1 0  1 1 0 1 0 0   1 1 0 1 0 0  1 0 0-0 -1-0-1-1-0 1  1 0 1 1 0 1 0 0 1  0 0 1   0---0---1---1
---	---

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

[Latin Units] 1/ 0---1---0---1  1 0 1  0 1 0 0 1 1 0 0 1  0 1-1-0-0-1-1-0 0 1  1 0 0 1 1 0   1 0 0 1 1 0  0 1 1-0 -0-1-1-0-0 1  0 1 1 0 0 1 1 0 1  0 1 0   0---1---0---1	2/ 0---0---1---1  1 1 0  0 1 0 1 0 0 1 0 1  0 1-0-1-1-0-1-0 0 1  0 1 1 0 1 0   1 0 1 0 0 1  0 1 1-0 -1-0-0-1-0 1  0 1 0 1 1 0 1 0 1  1 0 0   0---0---1---1
--	--

7/ 0---1---1---0  1 1 0  0 1 1 1 0 0 0 0 1  1 0-0-0-0-1-1-1 1 0  0 0 0 1 1 1   0 0 0 1 1 1  1 0 0-0 -0-1-1-1-1 0  0 1 1 1 1 0 0 0 1  1 0 0   1---0---0---1	5/ 0---1---0---1  1 0 1  0 0 0 1 1 0 0 1 1  1 1-0-0-1-1-0-0 1 0  0 1 1 0 0 1   0 1 1 0 0 1  1 0 1-1 -0-0-1-1-0 0  0 0 1 1 0 0 1 1 1  0 1 0   0---1---0---1
--	--

15/ 0---1---1---0  0 0 1  1 0 1 0 0 1 0 1 1  1 1-0-1-0-0-1-0 1 1  0 1 0 0 1 0   1 0 1 1 0 1  0 0 1-0 -1-1-0-1-0 0  0 0 1 0 1 1 0 1 0  0 1 1   1---0---0---1	11/ 0---0---1---1  0 1 1  0 1 1 0 1 0 0 1 0  0 1-0-0-1-0-1-1 1 0  1 1 0 1 0 0   1 1 0 1 0 0  1 0 0-0 -1-0-1-1-0 1  1 0 1 1 0 1 0 0 1  0 0 1   0---0---1---1
---	---

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

[Latin Units] 1/ 0---1---0---1  1 0 1  0 1 0 0 1 1 0 0 1  0 1-1-0-0-1-1-0 0 1  1 0 0 1 1 0   1 0 0 1 1 0  0 1 1-0 -0-1-1-0-0 1  0 1 1 0 0 1 1 0 1  0 1 0   0---1---0---1	2/ 0---0---1---1  1 1 0  0 1 0 1 0 0 1 0 1  0 1-0-1-1-0-1-0 0 1  0 1 1 0 1 0   1 0 1 0 0 1  0 1 1-0 -1-0-0-1-0 1  0 1 0 1 1 0 1 0 1  1 0 0   0---0---1---1
--	--

12/ 0---1---0---1  0 1 0  1 1 1 0 0 1 1 0 0  1 1-0-0-1-1-0-0 0 0  1 1 0 0 1 1   0 0 1 1 0 0  1 1 1-1 -0-0-1-1-0 0  1 1 0 0 1 1 0 0 0  1 0 1   0---1---0---1	5/ 0---1---0---1  1 0 1  0 0 0 1 1 0 0 1 1  1 1-0-0-1-1-0-0 1 0  0 1 1 0 0 1   0 1 1 0 0 1  1 0 1-1 -0-0-1-1-0 0  0 0 1 1 0 0 1 1 1  0 1 0   0---1---0---1
---	--

13/ 0---1---1---0  0 1 1  0 1 1 0 0 0 0 1 1  1 1-0-0-0-0-1-1 1 0  0 1 0 1 1 0   1 0 0 1 0 1  1 0 0-0 -1-1-1-1-0 0  0 0 1 1 1 1 0 0 1  0 0 1   1---0---0---1	9/ 0---1---1---0  1 0 0  1 0 1 1 0 1 0 0 1  1 0-0-1-0-1-1-0 1 1  0 0 0 0 1 1   0 0 1 1 1 1  0 0 1-0 -0-1-0-1-1 0  0 1 1 0 1 0 0 1 0  1 1 0   1---0---0---1
---	--

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

[Latin Units] 1/ 0---1---0---1  1 0 1  0 1 0 0 1 1 0 0 1  0 1-1-0-0-1-1-0	2/ 0---0---1---1  1 1 0  0 1 0 1 0 0 1 0 1  0 1-0-1-1-0-1-0
---	---

13/ 0---1---1---0  0 1 1  0 1 1 0 0 0 0 1 1  1 1-0-0-0-0-1-1	5/ 0---1---0---1  1 0 1  0 0 0 1 1 0 0 1 1  1 1-0-0-1-1-0-0
--	---



0 1  1 0 0 1 1 0	0 1  0 1 1 0 1 0	1 0  0 1 0 1 1 0	1 0  0 1 1 0 0 1
1 0 0 1 1 0  0 1	1 0 1 0 0 1  0 1	1 0 0 1 0 1  1 0	0 1 1 0 0 1  1 0
1-0 -0-1-1-0-0-1	1-0 -1-0-0-1-0-1	0-0 -1-1-1-1-0-0	1-1 -0-0-1-1-0-0
0 1 1 0 0 1 1 0	0 1 0 1 1 0 1 0	0 0 1 1 1 1 0 0	0 0 1 1 0 0 1 1
1  0 1 0	1  1 0 0	1  0 0 1	1  0 1 0
0-1-0-1	0-0-1-1	1-0-0-1	0-1-0-1

9/

6/

[Composition]

29#

0-1-1-0	0-0-1-1
1 0 0 1	1 1 0 0
0 1 1 0 0 1	1 1 1 1 0 0 0 0
1 0-0-1-0-1-1-0	0 0-0-0-1-1-1-1
1 1  0 0 0 0 1 1	1 0  1 0 0 1 0 1
0 0 1 1 1 1  0 0	0 1 0 1 1 0  1 0
1-0 -0-1-0-1-1-0	0-0 -0-0-1-1-1-1
0 1 1 0 1 0 0 1	1 1 1 1 0 0 0 0
0  1 1 0	1  1 0 0
1-0-0-1	0-0-1-1

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

[Latin Units] 1/

2/

14/

5/

0-1-0-1	0-0-1-1
1 0 1 0	1 1 0 0
1 0 0 1 1 0 0 1	1 0 1 0 0 1 0 1
0 1-1-0-0-1-1-0	0 1-0-1-1-0-1-0
0 1  1 0 0 1 1 0	0 1  0 1 1 0 1 0
1 0 0 1 1 0  0 1	1 0 1 0 0 1  0 1
1-0 -0-1-1-0-0-1	1-0 -1-0-0-1-0-1
0 1 1 0 0 1 1 0	0 1 0 1 1 0 1 0
1  0 1 0	1  1 0 0
0-1-0-1	0-0-1-1

0-0-1-1	0-1-0-1
0 0 1 1	1 0 1 0
0 1 0 1 1 0 1 0	0 0 1 1 0 0 1 1
0 1-0-1-1-0-1-0	1 1-0-0-1-1-0-0
1 1  1 1 0 0 0 0	1 0  0 1 1 0 0 1
1 1 1 1 0 0  0 0	0 1 1 0 0 1  1 0
1-0 -1-0-0-1-0-1	1-1 -0-0-1-1-0-0
1 0 1 0 0 1 0 1	0 0 1 1 0 0 1 1
0  0 1 1	1  0 1 0
0-0-1-1	0-1-0-1

13/

9/

[Composition]

53#

0-1-1-0	0-1-1-0
0 1 1 0	1 0 0 1
1 1 0 0 0 0 1 1	0 1 1 0 1 0 0 1
1 1-0-0-0-0-1-1	1 0-0-1-0-1-1-0
1 0  0 1 0 1 1 0	1 1  0 0 0 0 1 1
1 0 0 1 0 1  1 0	0 0 1 1 1 1  0 0
0-0 -1-1-1-1-0-0	1-0 -0-1-0-1-1-0
0 0 1 1 1 1 0 0	0 1 1 0 1 0 0 1
1  0 0 1	0  1 1 0
1-0-0-1	1-0-0-1

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

[Latin Units] 1/

2/

15/

5/

0-1-0-1	0-0-1-1
1 0 1 0	1 1 0 0
1 0 0 1 1 0 0 1	1 0 1 0 0 1 0 1
0 1-1-0-0-1-1-0	0 1-0-1-1-0-1-0
0 1  1 0 0 1 1 0	0 1  0 1 1 0 1 0
1 0 0 1 1 0  0 1	1 0 1 0 0 1  0 1
1-0 -0-1-1-0-0-1	1-0 -1-0-0-1-0-1
0 1 1 0 0 1 1 0	0 1 0 1 1 0 1 0
1  0 1 0	1  1 0 0
0-1-0-1	0-0-1-1

0-1-1-0	0-1-0-1
0 0 1 1	1 0 1 0
0 1 0 0 1 0 1 1	0 0 1 1 0 0 1 1
1 1-0-1-0-0-1-0	1 1-0-0-1-1-0-0
1 1  0 1 0 0 1 0	1 0  0 1 1 0 0 1
1 0 1 1 0 1  0 0	0 1 1 0 0 1  1 0
1-0 -1-1-0-1-0-0	1-1 -0-0-1-1-0-0
0 0 1 0 1 1 0 1	0 0 1 1 0 0 1 1
0  0 1 1	1  0 1 0
1-0-0-1	0-1-0-1

7/

11/

[Composition]

59#

0-1-1-0	0-0-1-1
1 1 0 0	0 1 1 0
1 1 1 0 0 0 0 1	1 1 0 1 0 0 1 0
1 0-0-0-0-1-1-1	0 1-0-0-1-0-1-1
1 0  0 0 0 1 1 1	1 0  1 1 0 1 0 0
0 0 0 1 1 1  1 0	1 1 0 1 0 0  1 0
0-0 -0-1-1-1-1-0	0-0 -1-0-1-1-0-1
0 1 1 1 1 0 0 0	1 0 1 1 0 1 0 0
1  1 0 0	1  0 0 1
1-0-0-1	0-0-1-1

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





13/ 9/  
 0—1—1—0 0—1—1—0  
 |0 1 1 |0 |1 0 0 |1  
 1 1 0 0 0 0 1 1 0 1 0 0 0 1  
 |1 1—0 0—0 0—1 1 |1 0—0 1—0 1—1 0  
 |1 0 |0 1 0 1 |1 0 |1 1 |0 0 0 0 1 1  
 |1 0 0 1 0 1 |1 0 |0 0 1 1 1 |0 0  
 0 0—1 1—1 1—0 0 |1 0—0 1—0 1—1 0 |  
 0 0 1 1 1 1 0 0 0 1 1 0 1 0 0 1  
 |1 |0 0 1 |0 |1 1 1 0 |  
 1—0—0—1 1—0—0—1

[Composition] 119#  
 1—40—28—61  
 |54 15 51 |10  
 47 12 22 49 42 13 19 56  
 |8 63—33—26—29—38—60—3  
 20 58 |41 31 21 35 48 6 |  
 |59 17 30 44 34 24 |7 45  
 62—5—27—36—39—32—2 57 |  
 9 46 52 23 16 43 53 18  
 55 |14 50 11 |  
 4—37—25—64

1/ 4/ 13/ 3/ 9/ 10/ 125#  
 1—47—28—54  
 |55 14 57 |4  
 46 12 23 49 36 6 25 63  
 |15 62—33—20—22—39—60—9  
 27 52 |34 30 21 41 48 7 |  
 |58 17 24 44 35 31 |13 38  
 56—5—26—43—45—32—3 50 |  
 2 40 59 29 16 42 53 19  
 61 |8 51 10 |  
 11—37—18—64

1/ 4/ 16/ 3/ 13/ 9/ 143#  
 1—48—20—61  
 |54 7 59 |10  
 39 12 22 57 42 5 27 56  
 |16 63—33—18—29—46—52—3  
 28 50 |41 31 21 35 40 14 |  
 |51 25 30 44 34 24 |15 37  
 62—13—19—36—47—32—2 49 |  
 9 38 60 23 8 43 53 26  
 55 |6 58 11 |  
 4—45—17—64

1/ 5/ 2/ 7/ 15/ 11/ 149#  
 1—55—16—58  
 |61 14 52 |3  
 46 8 29 50 35 9 20 63  
 |23 60—33—11—26—53—48—6  
 24 43 |34 28 25 38 47 21 |  
 |44 18 27 40 37 31 |22 41  
 59—17—12—39—54—32—5 42 |  
 2 45 56 30 15 36 57 19  
 62 |13 51 4 |  
 7—49—10—64

1/ 5/ 7/ 2/ 15/ 11/ 160#  
 1—59—16—54  
 |61 14 52 |3  
 46 12 29 50 35 5 20 63  
 |27 56—33—7—22—57—48—10  
 28 39 |34 24 21 42 47 25 |  
 |40 18 23 44 41 31 |26 37  
 55—17—8—43—58—32—9 38 |  
 2 45 60 30 15 36 53 19  
 62 |13 51 4 |  
 11—49—6—64

1/ 5/ 12/ 2/ 13/ 9/ 166#  
 1—60—8—61  
 |54 15 51 |10  
 47 12 22 49 42 13 19 56  
 |28 63—33—6—29—58—40—3  
 20 38 |41 31 21 35 48 26 |  
 |39 17 30 44 34 24 |27 45  
 62—25—7—36—59—32—2 37 |  
 9 46 52 23 16 43 53 18  
 55 |14 50 11 |  
 4—57—5—64

1/ 5/ 13/ 2/ 9/ 6/ 177#  
 1—59—16—54  
 |56 14 57 |3  
 46 12 24 50 35 5 25 63  
 |27 61—33—7—22—52—48—10  
 28 39 |34 29 21 42 47 20 |  
 |45 18 23 44 36 31 |26 37  
 55—17—13—43—58—32—4 38 |  
 2 40 60 30 15 41 53 19  
 62 |8 51 9 |  
 11—49—6—64

1/ 5/ 14/ 2/ 13/ 9/ 201#  
 1—52—16—61  
 |54 7 59 |10  
 39 12 22 57 42 5 27 56  
 |20 63—33—14—29—50—48—3  
 28 46 |41 31 21 35 40 18 |  
 |47 25 30 44 34 24 |19 37  
 62—17—15—36—51—32—2 45 |  
 9 38 60 23 8 43 53 26  
 55 |6 58 11 |  
 4—49—13—64

1/ 5/ 15/ 2/ 7/ 11/ 207#  
 1—59—16—54  
 |55 8 58 |9  
 40 12 23 50 41 5 26 63  
 |27 62—33—13—22—51—48—4  
 28 45 |34 30 21 36 47 19 |  
 |46 18 29 44 35 31 |20 37  
 61—17—14—43—52—32—3 38 |  
 2 39 60 24 15 42 53 25  
 56 |7 57 10 |  
 11—49—6—64

1/ 7/ 2/ 5/ 15/ 11/ 225#  
 1—55—28—46  
 |61 26 40 |3  
 58 20 29 38 35 9 8 63

1/ 7/ 5/ 2/ 15/ 11/ 231#  
 1—59—24—46  
 |61 22 44 |3  
 54 20 29 42 35 5 12 63

|23 48—33—11—14—53—60—18  
 24 43 | 34 16 13 50 59 21 |  
 |44 6 15 52 49 31 |22 41  
 47—5—|—12—51—54—32—17 42 |  
 2 57 56 30 27 36 45 7  
 62 | 25 39 4 |  
 19—37—10—64

|27 48—33—7—14—57—56—18  
 28 39 | 34 16 13 50 55 25 |  
 |40 10 15 52 49 31 |26 37  
 47—9—|—8—51—58—32—17 38 |  
 2 53 60 30 23 36 45 11  
 62 | 21 43 4 |  
 19—41—6—64

1/ 7/ 12/ 2/ 15/ 11/ 237#  
 1—59—24—46  
 |53 30 36 |11  
 62 28 21 34 43 13 4 55  
 |27 48—33—7—14—57—56—18  
 20 39 | 42 16 5 50 63 25 |  
 |40 2 15 60 49 23 |26 45  
 47—9—|—8—51—58—32—17 38 |  
 10 61 52 22 31 44 37 3  
 54 | 29 35 12 |  
 19—41—6—64

1/ 7/ 15/ 2/ 5/ 11/ 243#  
 1—59—30—40  
 |55 22 44 |9  
 54 26 23 36 41 5 12 63  
 |27 48—33—13—8—51—62—18  
 28 45 | 34 16 7 50 61 19 |  
 |46 4 15 58 49 31 |20 37  
 47—3—|—14—57—52—32—17 38 |  
 2 53 60 24 29 42 39 11  
 56 | 21 43 10 |  
 25—35—6—64

1/ 10/ 3/ 4/ 13/ 9/ 261#  
 1—44—24—61  
 |46 27 39 |18  
 59 20 14 37 50 25 7 48  
 |12 63—33—22—29—42—56—3  
 8 54 | 49 31 13 35 60 10 |  
 |55 5 30 52 34 16 |11 57  
 62—9—|—23—36—43—32—2 53 |  
 17 58 40 15 28 51 45 6  
 47 | 26 38 19 |  
 4—41—21—64

1/ 10/ 4/ 3/ 13/ 9/ 267#  
 1—40—28—61  
 |46 23 43 |18  
 55 20 14 41 50 21 11 48  
 |8 63—33—26—29—38—60—3  
 12 58 | 49 31 13 35 56 6 |  
 |59 9 30 52 34 16 |7 53  
 62—5—|—27—36—39—32—2 57 |  
 17 54 44 15 24 51 45 10  
 47 | 22 42 19 |  
 4—37—25—64

1/ 10/ 13/ 3/ 4/ 9/ 273#  
 1—46—28—55  
 |40 29 43 |18  
 61 26 8 35 50 21 11 48  
 |14 63—33—20—23—38—60—9  
 12 52 | 49 31 7 41 62 6 |  
 |59 3 24 58 34 16 |13 53  
 56—5—|—27—42—45—32—2 51 |  
 17 54 44 15 30 57 39 4  
 47 | 22 36 25 |  
 10—37—19—64

1/ 10/ 16/ 3/ 13/ 9/ 291#  
 1—48—20—61  
 |38 23 43 |26  
 55 28 6 41 58 21 11 40  
 |16 63—33—18—29—46—52—3  
 12 50 | 57 31 5 35 56 14 |  
 |51 9 30 60 34 8 |15 53  
 62—13—|—19—36—47—32—2 49 |  
 25 54 44 7 24 59 37 10  
 39 | 22 42 27 |  
 4—45—17—64

1/ 12/ 2/ 5/ 13/ 9/ 297#  
 1—56—12—61  
 |46 27 39 |18  
 59 20 14 37 50 25 7 48  
 |24 63—33—10—29—54—44—3  
 8 42 | 49 31 13 35 60 22 |  
 |43 5 30 52 34 16 |23 57  
 62—21—|—11—36—55—32—2 41 |  
 17 58 40 15 28 51 45 6  
 47 | 26 38 19 |  
 4—53—9—64

1/ 12/ 5/ 2/ 13/ 9/ 308#  
 1—60—8—61  
 |46 23 43 |18  
 55 20 14 41 50 21 11 48  
 |28 63—33—6—29—58—40—3  
 12 38 | 49 31 13 35 56 26 |  
 |39 9 30 52 34 16 |27 53  
 62—25—|—7—36—59—32—2 37 |  
 17 54 44 15 24 51 45 10  
 47 | 22 42 19 |  
 4—57—5—64

1/ 12/ 7/ 2/ 15/ 11/ 319#  
 1—59—16—54  
 |45 30 36 |19  
 62 28 13 34 51 21 4 47  
 |27 56—33—7—22—57—48—10  
 12 39 | 50 24 5 42 63 25 |  
 |40 2 23 60 41 15 |26 53  
 55—17—|—8—43—58—32—9 38 |

1/ 12/ 13/ 2/ 5/ 9/ 325#  
 1—60—14—55  
 |40 29 43 |18  
 61 26 8 35 50 21 11 48  
 |28 63—33—6—23—52—46—9  
 12 38 | 49 31 7 41 62 20 |  
 |45 3 24 58 34 16 |27 53  
 56—19—|—13—42—59—32—2 37 |

18 61 44 14 31 52 37 3  
 46 | 29 35 20 |  
 11-----49-----6-----64

17 54 44 15 30 57 39 4  
 47 | 22 36 25 |  
 10-----51-----5-----64

1/ 12/ 14/ 2/ 13/ 9/ 349#  
 1-----52-----16-----61  
 |38 23 43 |26  
 55 28 6 41 58 21 11 40  
 |20 63-----33-14-----29-50-----48-3  
 12 46 | 57 31 5 35 56 18 |  
 |47 9 30 60 34 8 |19 53  
 62-17-|15-36-51-32-2 45 |  
 25 54 44 7 24 59 37 10  
 39 | 22 42 27 |  
 4-----49-----13-----64

1/ 12/ 15/ 2/ 7/ 11/ 355#  
 1-----59-----16-----54  
 |39 24 42 |25  
 56 28 7 34 57 21 10 47  
 |27 62-----33-13-----22-51-----48-4  
 12 45 | 50 30 5 36 63 19 |  
 |46 2 29 60 35 15 |20 53  
 61-17-|14-43-52-32-3 38 |  
 18 55 44 8 31 58 37 9  
 40 | 23 41 26 |  
 11-----49-----6-----64

1/ 13/ 2/ 5/ 9/ 6/ 373#  
 1-----55-----28-----46  
 |48 26 53 |3  
 58 20 16 38 35 9 21 63  
 |23 61-----33-11-----14-40-----60-18  
 24 43 | 34 29 13 50 59 8 |  
 |57 6 15 52 36 31 |22 41  
 47-5-|25-51-54-32-4 42 |  
 2 44 56 30 27 49 45 7  
 62 | 12 39 17 |  
 19-----37-----10-----64

1/ 13/ 3/ 4/ 9/ 10/ 397#  
 1-----59-----24-----46  
 |47 26 53 |4  
 58 20 15 37 36 10 21 63  
 |27 62-----33-8-----14-43-----56-17  
 23 40 | 34 30 13 49 60 11 |  
 |54 5 16 52 35 31 |25 42  
 48-9-|22-51-57-32-3 38 |  
 2 44 55 29 28 50 45 7  
 61 | 12 39 18 |  
 19-----41-----6-----64

1/ 13/ 4/ 3/ 9/ 10/ 415#  
 1-----55-----28-----46  
 |47 22 57 |4  
 54 20 15 41 36 6 25 63  
 |23 62-----33-12-----14-39-----60-17  
 27 44 | 34 30 13 49 56 7 |  
 |58 9 16 52 35 31 |21 38  
 48-5-|26-51-53-32-3 42 |  
 2 40 59 29 24 50 45 11  
 61 | 8 43 18 |  
 19-----37-----10-----64

1/ 13/ 5/ 2/ 9/ 6/ 433#  
 1-----59-----24-----46  
 |48 22 57 |3  
 54 20 16 42 35 5 25 63  
 |27 61-----33-7-----14-44-----56-18  
 28 39 | 34 29 13 50 55 12 |  
 |53 10 15 52 36 31 |26 37  
 47-9-|21-51-58-32-4 38 |  
 2 40 60 30 23 49 45 11  
 62 | 8 43 17 |  
 19-----41-----6-----64

... (skip) ...

16/ 3/ 13/ 1/ 4/ 9/ 4549#  
 1-----62-----12-----55  
 |24 25 47 |34  
 29 42 20 39 38 17 43 32  
 |58 63-----5-4-----51-54-----16-9  
 44 8 | 37 59 19 13 30 50 |  
 |15 35 52 46 6 28 |57 21  
 56-49-|11-14-----61-60-----2 7 |  
 33 22 48 27 26 45 23 36  
 31 | 18 40 41 |  
 10-----53-----3-----64

16/ 4/ 1/ 3/ 13/ 9/ 4561#  
 1-----48-----20-----61  
 |30 7 59 |34  
 15 36 22 57 42 5 51 32  
 |40 63-----9-18-----53-46-----28-3  
 52 26 | 41 55 21 11 16 38 |  
 |27 49 54 44 10 24 |39 13  
 62-37-|19-12-----47-56-----2 25 |  
 33 14 60 23 8 43 29 50  
 31 | 6 58 35 |  
 4-----45-----17-----64

16/ 4/ 3/ 1/ 13/ 9/ 4567#  
 1-----48-----20-----61  
 |30 11 55 |34  
 15 36 26 53 38 9 51 32  
 |44 63-----5-18-----57-46-----24-3  
 52 22 | 37 59 25 7 16 42 |  
 |23 49 58 40 6 28 |43 13  
 62-41-|19-8-----47-60-----2 21 |  
 33 14 56 27 12 39 29 50  
 31 | 10 54 35 |  
 4-----45-----17-----64

16/ 4/ 10/ 1/ 13/ 9/ 4570#  
 1-----40-----28-----61  
 |22 11 55 |42  
 15 44 18 53 46 9 51 24  
 |36 63-----5-26-----57-38-----32-3  
 52 30 | 45 59 17 7 16 34 |  
 |31 49 58 48 6 20 |35 13  
 62-33-|27-8-----39-60-----2 29 |  
 41 14 56 19 12 47 21 50  
 23 | 10 54 43 |  
 4-----37-----25-----64

16/ 4/ 13/ 1/ 3/ 9/ 4573#  
 1-----48-----26-----55  
 |24 11 61 |34  
 15 42 20 53 38 3 57 32  
 |44 63-----5-18-----51-40-----30-9  
 58 22 | 37 59 19 13 16 36 |  
 |29 49 52 46 6 28 |43 7  
 56-35-|25-14-47-60-2 21 |  
 33 8 62 27 12 45 23 50  
 31 | 4 54 41 |  
 10-----39-----17-----64

16/ 10/ 1/ 3/ 13/ 9/ 4585#  
 1-----48-----20-----61  
 |14 23 43 |50  
 31 52 6 41 58 21 35 16  
 |40 63-----9-18-----53-46-----28-3  
 36 26 | 57 55 5 11 32 38 |  
 |27 33 54 60 10 8 |39 29  
 62-37-|19-12-47-56-2 25 |  
 49 30 44 7 24 59 13 34  
 15 | 22 42 51 |  
 4-----45-----17-----64

16/ 10/ 3/ 1/ 13/ 9/ 4591#  
 1-----48-----20-----61  
 |14 27 39 |50  
 31 52 10 37 54 25 35 16  
 |44 63-----5-18-----57-46-----24-3  
 36 22 | 53 59 9 7 32 42 |  
 |23 33 58 56 6 12 |43 29  
 62-41-|19-8-47-60-2 21 |  
 49 30 40 11 28 55 13 34  
 15 | 26 38 51 |  
 4-----45-----17-----64

16/ 10/ 4/ 1/ 13/ 9/ 4594#  
 1-----40-----28-----61  
 |14 19 47 |50  
 23 52 10 45 54 17 43 16  
 |36 63-----5-26-----57-38-----32-3  
 44 30 | 53 59 9 7 24 34 |  
 |31 41 58 56 6 12 |35 21  
 62-33-|27-8-39-60-2 29 |  
 49 22 48 11 20 55 13 42  
 15 | 18 46 51 |  
 4-----37-----25-----64

16/ 10/ 13/ 1/ 3/ 9/ 4597#  
 1-----48-----26-----55  
 | 8 27 45 |50  
 31 58 4 37 54 19 41 16  
 |44 63-----5-18-----51-40-----30-9  
 42 22 | 53 59 3 13 32 36 |  
 |29 33 52 62 6 12 |43 23  
 56-35-|25-14-47-60-2 21 |  
 49 24 46 11 28 61 7 34  
 15 | 20 38 57 |  
 10-----39-----17-----64

16/ 13/ 1/ 3/ 4/ 9/ 4609#  
 1-----62-----20-----47  
 |16 21 59 |34  
 29 50 8 43 42 5 51 32  
 |54 63-----9-4-39-46-----28-17  
 52 12 | 41 55 7 25 30 38 |  
 |27 35 40 58 10 24 |53 13  
 48-37-|19-26-61-56-2 11 |  
 33 14 60 23 22 57 15 36  
 31 | 6 44 49 |  
 18-----45-----3-----64

16/ 13/ 3/ 1/ 4/ 9/ 4627#  
 1-----62-----20-----47  
 |16 25 55 |34  
 29 50 12 39 38 9 51 32  
 |58 63-----5-4-43-46-----24-17  
 52 8 | 37 59 11 21 30 42 |  
 |23 35 44 54 6 28 |57 13  
 48-41-|19-22-61-60-2 7 |  
 33 14 56 27 26 53 15 36  
 31 | 10 40 49 |  
 18-----45-----3-----64

16/ 13/ 4/ 1/ 3/ 9/ 4639#  
 1-----56-----26-----47  
 |16 19 61 |34  
 23 50 12 45 38 3 57 32  
 |52 63-----5-10-43-40-----30-17  
 58 14 | 37 59 11 21 24 36 |  
 |29 41 44 54 6 28 |51 7  
 48-35-|25-22-55-60-2 13 |  
 33 8 62 27 20 53 15 42  
 31 | 4 46 49 |  
 18-----39-----9-----64

16/ 13/ 9/ 1/ 3/ 4/ 4651#  
 1-----63-----26-----40  
 |16 19 54 |41  
 23 57 12 38 45 3 50 32  
 |59 56-----5-10-36-47-----30-17  
 58 14 | 37 52 4 21 31 43 |  
 |22 34 44 61 13 28 |51 7  
 48-35-|18-29-55-60-9 6 |  
 33 15 62 20 27 53 8 42  
 24 | 11 46 49 |  
 25-----39-----2-----64

16/ 13/ 10/ 1/ 3/ 9/ 4669#  
 1-----56-----26-----47  
 | 8 27 53 |42  
 31 58 4 37 46 11 49 24  
 |52 63-----5-10-43-40-----30-17  
 50 14 | 45 59 3 21 32 36 |  
 |29 33 44 62 6 20 |51 15  
 48-35-|25-22-55-60-2 13 |  
 41 16 54 19 28 61 7 34  
 23 | 12 38 57 |  
 18-----39-----9-----64

[解の計数 : Count of Compositions = 4680]

[n1の値による解の計数 : Counts according to n1]



```

1: 4680, 2: ..., 3: ..., 4: ..., 5: ..., 6: ..., 7: ..., 8: ...,
9: ..., 10: ..., 11: ..., 12: ..., 13: ..., 14: ..., 15: ..., 16: ...,
17: ..., 18: ..., 19: ..., 20: ..., 21: ..., 22: ..., 23: ..., 24: ...,
25: ..., 26: ..., 27: ..., 28: ..., 29: ..., 30: ..., 31: ..., 32: ...,
33: ..., 34: ..., 35: ..., 36: ..., 37: ..., 38: ..., 39: ..., 40: ...

```

[n43の値による解の計数 : Counts according to n1]

```

1: ..., 2: ..., 3: ..., 4: ..., 5: ..., 6: ..., 7: ..., 8: ...,
9: ..., 10: ..., 11: ..., 12: ..., 13: ..., 14: ..., 15: ..., 16: ...,
17: ..., 18: ..., 19: ..., 20: ..., 21: ..., 22: ..., 23: ..., 24: ...,
25: ..., 26: ..., 27: ..., 28: ..., 29: ..., 30: ..., 31: ..., 32: 780,
33: ..., 34: ..., 35: ..., 36: ..., 37: ..., 38: ..., 39: ..., 40: ...,
41: ..., 42: ..., 43: ..., 44: ..., 45: ..., 46: ..., 47: ..., 48: 780,
49: ..., 50: ..., 51: ..., 52: ..., 53: ..., 54: ..., 55: ..., 56: 780,
57: ..., 58: ..., 59: ..., 60: 780, 61: ..., 62: 780, 63: 780, 64: ...

```

OK!

\*\* Originally dictated in 2010; Recompiled on July 23, 2017  
by Kanji Setsuda with macOS Sierra 10.12.6 and Xcode 7.3 \*\*

上に出力結果 (部分) を示した。

またたく間に 4680個もの解を合成して出力した。ずいぶん多い感じがすると思うが、専門家の眼から見ると、これはむしろ異常なくらいに少ない数字である。

何しろ、240万余個の汎型と1千万超個の対称型の「積集合」である。もっとあってもよいのに、実はこれだけしかなかった。片手間に探したくらいではもともと見つかるはずもない、極めて希少価値の高い宝物であった。

しかも、我々が作ったのは、単なる連立型の汎対称立体四方陣ではない。あくまで「完全オイラー方陣の・・・」である。そのために特別に珍しかったのかも知れない。

**5.** では、単なる連立型の汎対称立体四方陣は、一体いくつあるのか。べつに「完全オイラー方陣」でなくてもよい。そうであってもよいが、そうでなくてもよい。要するに総数はいくつあるのかを、次に知りたくなる。好奇心でえものには際限が無い。

二進法を使わずに連立型を作る工夫は無いのか。今までは、普通に探してことごとく失敗している。1千万超個の対称型を作って、後から汎対角線条件でふるいに掛ける方法は、効率が全然悪い。実行中に時間ばかり食って、いつ終わるのか皆目見当が付かない。

ここは、汎型から攻めるしかない。そして対称型の補数条件を最初から絡ませて行く。幸い汎型の研究は進んでいる。第1章の上級編でわかったことを積極的に活用しよう。

(1)  $n1 = 1$  の標準解の場合、 $n43$  の取り得る値は {64, 63, 62, 60, 56, 48, 32} のいずれかに限られる。 $n43 = 64$  の時は例外だ(「完全型立体四方陣」になる)が、C型補数16ペアとD型補数16ペアが定義される。それらが全て組んで汎対角線上に並び、 $C + D = 130$  の定和を実現する。しかも、 $C = n1 + n43$ ;  $D = 130 - C$  である。

(2) 一方、対称型では、例えば  $n1 + n64 = 65$ ;  $n22 + n43 = 65$ ; である。 $n1$ ,  $n22$ ,  $n43$ ,  $n64$  は1本の汎対角線上にあり、結局その上の4数は必ず同時に決まる。 $n43$  の値を決めた時点で、 $n22$ ,  $n64$  も一義的に決まるのである。例えば、 $n1 = 1$ ;  $n43 = 32$ ; と選択した場合には、 $n64 = 64$ ;  $n22 = 33$ ; と決まる。他の値になることはない。

(3)  $n1$  が乗る汎対角線は1本だけではない。他に3本もある。だが、 $n1$  の補数はいつも決まって  $n43$  だ。そして、{ $n24$ ,  $n62$ }, { $n30$ ,  $n56$ }, { $n32$ ,  $n54$ } などの補数ペアは、必ずD型になり、和が  $130 - C = 130 - (n1 + n43)$  に決まる。

{ $n2$ ,  $n44$ } も、D型の補数ペアだ。よって、 $n2 + n44 = 130 - (n1 + n43) = 130 - 33 = 97$  と決まる。 $n63 + n21 = 130 - 97 = 33$  と決まるから、{ $n2$ ,  $n21$ ,  $n44$ ,  $n63$ } の4数の取り得る値は自ずから限られてくる。・・・

このような内在的関係を利用して、汎対角線上の4数を一度に決めるという工夫を何とか講じよう。そうすれば、無駄な探索をいろいろしないで済むかも知れない。

次に我がプログラムをリストにして示そう。4数ずつをまとめて見てほしい。

```
/** The Simultaneous Type of Magic Cubes of Order 4: **
/** Both 'Pan-Triagonal' and 'Self-Complementary': **
/** Starting with n1=1 & n43=32~63; **
/** 'MC444Sm12.c': Originally dictated by Kanji Setsuda in 2005, 2011; **
/** Recompiled on July 23, 2017 with macOS Sierra 10.12.6 & Xcode 7.3 **
//
#include <stdio.h>
/** Global Variables *
short int cnt, cnt1;
short cnt2, cnt3;
short CC, DD, LSM;
short nm[65], flg[65];
short nc[7];
short d[4][65];
short ctbl[33], dtbl[33];
short cn1[65], cn43[65];
//
/** Sub-Routines(Functions) *
short mktbl(void), ctyp(short x), dtyp(short x);
void tblprint(void);
void stp01(void), stp011(void), stp012(void), stp013(void);
void stp02(void), stp021(void), stp022(void), stp023(void);
void stp03(void), stp031(void), stp032(void), stp033(void);
void stp04(void), stp041(void), stp042(void), stp043(void);
void stp05(void), stp051(void), stp052(void), stp053(void);
void stp06(void), stp061(void), stp062(void), stp063(void);
void stp07(void), stp071(void), stp072(void), stp073(void);
void stp08(void), stp081(void), stp082(void), stp083(void);
void stp09(void), stp091(void), stp092(void), stp093(void);
void stp10(void), stp101(void), stp102(void), stp103(void);
void stp11(void), stp111(void), stp112(void), stp113(void);
void stp12(void), stp121(void), stp122(void), stp123(void);
void stp13(void), stp131(void), stp132(void), stp133(void);
void stp14(void);
void stp15(void), stp151(void), stp152(void), stp153(void);
void stp16(void);
void stp17(void), stp171(void), stp172(void), stp173(void);
void stp18(void), stp19(void), stp20(void);
void stp21(void), stp22(void);
void ansprint(void);
void pr3sol(void);
void prcn1(void), prcn43(void);
//
/** Main Program: *
/** Set n1=1 & n64=64 & n43 & n22=65-n43 *
int main(){
short m, n;
printf("//\n");
printf("/** Simultaneous Type of Magic Cubes of Order 4:\n");
printf("// Both 'Pan-triagonal' and 'Self-Complementary'; **\n");
nc[0]=32; nc[1]=48; nc[2]=56; nc[3]=60; nc[4]=62; nc[5]=63;
for(m=0;m<6;m++){
for(n=0;n<65;n++){nm[n]=0;} // Reset the Basic Positions
nm[1]=1; nm[43]=nc[m]; LSM=130; // C1: Set n1 & n43
CC=nm[1]+nm[43]; DD=LSM-CC; // Set C & D
for(n=0;n<33;n++){ctbl[n]=0; dtbl[n]=0;} // Reset the Tables
for(n=0;n<65;n++){flg[n]=0;} // Reset the Flags
n=mktbl(); // Make the CD-Tables
if(n>0){tblprint(); printf("// No!\n"); return 0;} // Error Return
//
printf("//\n");
```

```

printf("// [Abstract List of Solutions when n1=1 & n64=64 & n43=%2d]\n",nm[43]);
for(n=0;n<65;n++){flg[n]=0;}; // Reset the Flags
nm[64]=65-nm[1]; nm[22]=65-nm[43]; // Define Self-complementary Pairs
flg[nm[1]]=1; flg[nm[22]]=1; flg[nm[43]]=1; flg[nm[64]]=1;
//
cnt=0; cnt3=0; // Reset the Counter
stp01(); // Begin the Search
if(cnt3>0){pr3sol();}
printf("// [Solution Counts = %d]\n",cnt); // Final Report
}
printf("//\n");
printf("// [Solution Counts in Total when n1==1: %d]\n",cn1[1]);
prcn1();
printf("\n");
printf("// [Counts according to the Value of n43]\n");
prcn43();
printf("\n");
printf("// OK!\n");
printf("//\n");
printf("//** Dictated and Listed by Kanji Setsuda originally in 2005 and 2011;\n");
printf("// Recompiled on July 23, 2017 with macOS Sierra 10.12.6 & Xcode 7.3 **\n");
printf("//\n");
return 0;
}
//
// Make the CD-Tables
short mktbl(){
short a,b,n,m1,m2,ans;
m1=1; m2=1; ans=0;
for(n=1;n<33;n++){
a=n; b=CC-a;
if((a<b)&&(flg[a]==0)&&(flg[b]==0)){
ctbl[m1]=a; ctbl[33-m1]=b;
flg[a]=1; flg[b]=1; m1++;
a=66-n; b=DD-a;
do{a--; b++;}
while((a==b)|| (flg[a]>0)|| (flg[b]>0));
if(a>b){dtbl[m2]=a; dtbl[33-m2]=b;
flg[a]=1; flg[b]=1; m2++;}}
}
if((ctbl[16]==0)|| (dtbl[16]==0)){ans=1;}
return ans;
}
// Print the CD-Tables
void tblprint(){
short n;
printf("C=%d: D=%d\n",CC,DD);
for(n=1;n<17;n++){
printf("(%d,%d)",ctbl[n],ctbl[33-n]);
printf("{%d,%d}\n",dtbl[n],dtbl[33-n]);
}
}
// Is it C-Type?
short ctyp(short x){
short n,ans;
ans=1;
if((1<x)&&(x<CC)){
for(n=1;n<33;n++){if(x==ctbl[n]){ans=0; break;}}
}
return ans;
}
// Is it D-Type?
short dtyp(short x){
short n,ans;

```

```

ans=1;
if((1<x)&&(x<65)){
    for(n=1;n<33;n++){if(x==dtbl[n]){ans=0; break;}}
}
return ans;
}
//
// C: (n1,n43),(n3,n41),(n6,n48),(n8,n46),(n9,n35),(n11,n33),(n14,n40),(n16,n38),
// (n18,n60),(n20,n58),(n21,n63),(n23,n61),(n26,n52),(n28,n50),(n29,n55),(n31,n53)
// D: (n2,n44),(n4,n42),(n5,n47),(n7,n45),(n10,n36),(n12,n34),(n13,n39),(n15,n37),
// (n17,n59),(n19,n57),(n22,n64),(n24,n62),(n25,n51),(n27,n49),(n30,n56),(n32,n54)
//
/** Begin the Search *
/** Set n4(>n1) *
void stp010(){
    short a;
    for(a=63;a>nm[1];a--){
        if(flg[a]==0){
            flg[a]=1; nm[4]=a;
            stp011();
            flg[a]=0;}
    }
}
/** Set n42 *
void stp011(){
    short a;
    a=DD-nm[4];
    if((1<a)&&(a<64)&&(flg[a]==0)){
        flg[a]=1; nm[42]=a;
        stp012();
        flg[a]=0;}
}
/** Set n61(>n1) *
void stp012(){
    short a;
    a=65-nm[4];
    if((nm[1]<a)&&(a<64)&&(flg[a]==0)){
        flg[a]=1; nm[61]=a;
        stp013();
        flg[a]=0;}
}
/** Set n23 *
void stp013(){
    short a;
    a=65-nm[42];
    if((1<a)&&(a<64)&&(flg[a]==0)){
        flg[a]=1; nm[23]=a; cnt2=0;
        stp02();
        flg[a]=0;}
}
//
/** Set n13: (n1<n13<n4) *
void stp02(){
    short a;
    for(a=nm[4]-1;a>nm[1];a--){
        if(flg[a]==0){
            flg[a]=1; nm[13]=a;
            if(cnt<390){cnt2=0;}
            stp021();
            flg[a]=0;}
    }
}
/** Set n39 *
void stp021(){
    short a;
    a=DD-nm[13];
    if((1<a)&&(a<64)&&(flg[a]==0)){
        flg[a]=1; nm[39]=a;
        stp022();
        flg[a]=0;}
}
/** Set n52(>n1) *
void stp022(){
    short a;
    a=65-nm[13];
    if((nm[1]<a)&&(a<64)&&(flg[a]==0)){
        flg[a]=1; nm[52]=a;
        stp023();
        flg[a]=0;}
}
/** Set n26 *
void stp023(){
    short a;
    a=65-nm[39];
    if((1<a)&&(a<64)&&(flg[a]==0)){
        flg[a]=1; nm[26]=a;
        stp03();
        flg[a]=0;}
}
//
/** Set n49: (n1<n49<n13) *
void stp03(){
    short a;
    for(a=nm[13]-1;a>nm[1];a--){
        if(flg[a]==0){
            flg[a]=1; nm[49]=a;
            stp031();
            flg[a]=0;}
    }
}
/** Set n27 *
void stp031(){
    short a;
    a=DD-nm[49];
    if((1<a)&&(a<64)&&(flg[a]==0)){
        flg[a]=1; nm[27]=a;
        stp032();
        flg[a]=0;}
}
/** Set n16(>n1) *
void stp032(){
    short a;
    a=65-nm[49];
    if((nm[1]<a)&&(a<64)&&(flg[a]==0)){
        flg[a]=1; nm[16]=a;

```

```

    stp033();
    flg[a]=0;}
}
/** Set n38 *
void stp033(){
short a;
a=65-nm[27];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[38]=a;
    stp04();
    flg[a]=0;}
}
/** Search Level 1: *
/** Set n2 *
void stp04(){
short a;
for(a=63;a>1;a--){
    if(flg[a]==0){
        flg[a]=1; nm[2]=a;
        stp041();
        flg[a]=0;}
    }
}
/** Set n44 *
void stp041(){
short a;
a=DD-nm[2];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[44]=a;
    stp042();
    flg[a]=0;}
}
/** Set n63 *
void stp042(){
short a;
a=65-nm[2];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[63]=a;
    stp043();
    flg[a]=0;}
}
/** Set n21 *
void stp043(){
short a;
a=65-nm[44];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[21]=a;
    stp05();
    flg[a]=0;}
}
/**
/** Set n3=130-n1-n2-n4 *
void stp05(){
short a;
a=LSM-nm[1]-nm[2]-nm[4];
if(flg[a]==0){
    flg[a]=1; nm[3]=a;
    stp051();
    flg[a]=0;}
}
/** Set n41 *
void stp051(){
short a;
a=CC-nm[3];

```

```

    ↑ if((1<a)&&(a<64)&&(flg[a]==0)){
        flg[a]=1; nm[41]=a;
        stp052();
        flg[a]=0;}
    }
/** Set n62 *
void stp052(){
short a;
a=65-nm[3];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[62]=a;
    stp053();
    flg[a]=0;}
}
/** Set n24 *
void stp053(){
short a;
a=65-nm[41];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[24]=a;
    stp06();
    flg[a]=0;}
}
/**
/** Set n5 *
void stp06(){
short a;
for(a=63;a>1;a--){
    if(flg[a]==0){
        flg[a]=1; nm[5]=a;
        stp061();
        flg[a]=0;}
    }
}
/** Set n47 *
void stp061(){
short a;
a=DD-nm[5];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[47]=a;
    stp062();
    flg[a]=0;}
}
/** Set n60 *
void stp062(){
short a;
a=65-nm[5];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[60]=a;
    stp063();
    flg[a]=0;}
}
/** Set n18 *
void stp063(){
short a;
a=65-nm[47];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[18]=a;
    stp07();
    flg[a]=0;}
}
/**
/** Set n9=130-n1-n5-n13 *
void stp07(){

```

```

short a;
a=LSM-nm[1]-nm[5]-nm[13];
if(flag[a]==0){
    flg[a]=1; nm[9]=a;
    stp071();
    flg[a]=0;}
}
/* Set n35 *
void stp071(){
short a;
a=CC-nm[9];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[35]=a;
    stp072();
    flg[a]=0;}
}
/* Set n56 *
void stp072(){
short a;
a=65-nm[9];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[56]=a;
    stp073();
    flg[a]=0;}
}
/* Set n30 *
void stp073(){
short a;
a=65-nm[35];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[30]=a;
    stp08();
    flg[a]=0;}
}
//
/* Set n17 *
void stp08(){
short a;
for(a=63;a>1;a--){
    if(flag[a]==0){
        flg[a]=1; nm[17]=a;
        stp081();
        flg[a]=0;}
    }
}
/* Set n59 *
void stp081(){
short a;
a=DD-nm[17];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[59]=a;
    stp082();
    flg[a]=0;}
}
/* Set n48 *
void stp082(){
short a;
a=65-nm[17];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[48]=a;
    stp083();
    flg[a]=0;}
}
/* Set n6 *

```

```

void stp083(){
short a;
a=65-nm[59];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[6]=a;
    stp09();
    flg[a]=0;}
}
//
/* Set n33=130-n1-n17-n49 *
void stp09(){
short a;
a=LSM-nm[1]-nm[17]-nm[49];
if(flag[a]==0){
    flg[a]=1; nm[33]=a;
    stp091();
    flg[a]=0;}
}
/* Set n11 *
void stp091(){
short a;
a=CC-nm[33];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[11]=a;
    stp092();
    flg[a]=0;}
}
/* Set n32 *
void stp092(){
short a;
a=65-nm[33];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[32]=a;
    stp093();
    flg[a]=0;}
}
/* Set n54 *
void stp093(){
short a;
a=65-nm[11];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[54]=a;
    stp10();
    flg[a]=0;}
}
/* Search Level 2: *
/* Set n8 *
void stp10(){
short a;
for(a=63;a>1;a--){
    if(flag[a]==0){
        flg[a]=1; nm[8]=a;
        stp101();
        flg[a]=0;}
    }
}
/* Set n46 *
void stp101(){
short a;
a=CC-nm[8];
if((1<a)&&(a<64)&&(flg[a]==0)){
    flg[a]=1; nm[46]=a;
    stp102();
    flg[a]=0;}
}

```

```

}
/** Set n57 *
void stp102(){
short a;
a=65-nm[8];
if((1<a)&&(a<64)&&(flg[a]==0)){
flg[a]=1; nm[57]=a;
stp103();
flg[a]=0;}
}
/** Set n19 *
void stp103(){
short a;
a=65-nm[46];
if((1<a)&&(a<64)&&(flg[a]==0)){
flg[a]=1; nm[19]=a;
stp11();
flg[a]=0;}
}
//
/** Set n7=130-n5-n6-n8 *
void stp11(){
short a;
a=LSM-nm[5]-nm[6]-nm[8];
if(flg[a]==0){
flg[a]=1; nm[7]=a;
stp111();
flg[a]=0;}
}
/** Set n45 *
void stp111(){
short a;
a=DD-nm[7];
if((1<a)&&(a<64)&&(flg[a]==0)){
flg[a]=1; nm[45]=a;
stp112();
flg[a]=0;}
}
/** Set n58 *
void stp112(){
short a;
a=65-nm[7];
if((1<a)&&(a<64)&&(flg[a]==0)){
flg[a]=1; nm[58]=a;
stp113();
flg[a]=0;}
}
/** Set n20 *
void stp113(){
short a;
a=65-nm[45];
if((1<a)&&(a<64)&&(flg[a]==0)){
flg[a]=1; nm[20]=a;
stp12();
flg[a]=0;}
}
//
/** Set n12=130-n4-n8-n16 *
void stp12(){
short a;
a=LSM-nm[4]-nm[8]-nm[16];
if(flg[a]==0){
flg[a]=1; nm[12]=a;
stp121();

```

```

↑ flg[a]=0;}
}
/** Set n34 *
void stp121(){
short a;
a=DD-nm[12];
if((1<a)&&(a<64)&&(flg[a]==0)){
flg[a]=1; nm[34]=a;
stp122();
flg[a]=0;}
}
/** Set n53 *
void stp122(){
short a;
a=65-nm[12];
if((1<a)&&(a<64)&&(flg[a]==0)){
flg[a]=1; nm[53]=a;
stp123();
flg[a]=0;}
}
/** Set n31 *
void stp123(){
short a;
a=65-nm[34];
if((1<a)&&(a<64)&&(flg[a]==0)){
flg[a]=1; nm[31]=a;
stp13();
flg[a]=0;}
}
//
/** Set n10=130-n9-n11-n12 *
void stp13(){
short a;
a=LSM-nm[9]-nm[11]-nm[12];
if(flg[a]==0){
flg[a]=1; nm[10]=a;
stp131();
flg[a]=0;}
}
/** Set n36 *
void stp131(){
short a;
a=DD-nm[10];
if((1<a)&&(a<64)&&(flg[a]==0)){
flg[a]=1; nm[36]=a;
stp132();
flg[a]=0;}
}
/** Set n55 *
void stp132(){
short a;
a=65-nm[10];
if((1<a)&&(a<64)&&(flg[a]==0)){
flg[a]=1; nm[55]=a;
stp133();
flg[a]=0;}
}
/** Set n29 *
void stp133(){
short a;
a=65-nm[36];
if((1<a)&&(a<64)&&(flg[a]==0)){
flg[a]=1; nm[29]=a;
stp14();

```



```

    flg[a]=0;}
}
//
/** Check (n4+n20+n36+n52=130) *
void stp14(){
    if(nm[4]+nm[20]+nm[36]+nm[52]==LSM){stp15
();}
}
//
/** Set n14=130-n2-n6-n10 *
void stp15(){
    short a;
    a=LSM-nm[2]-nm[6]-nm[10];
    if(flga==0){
        flg[a]=1; nm[14]=a;
        stp151();
        flg[a]=0;}
}
/** Set n40 *
void stp151(){
    short a;
    a=CC-nm[14];
    if((1<a)&&(a<64)&&(flga==0)){

```

```

    flg[a]=1; nm[40]=a;
    stp152();
    flg[a]=0;}
}
/** Set n51 *
void stp152(){
    short a;
    a=65-nm[14];
    if((1<a)&&(a<64)&&(flga==0)){
        flg[a]=1; nm[51]=a;
        stp153();
        flg[a]=0;}
}
/** Set n25 *
void stp153(){
    short a;
    a=65-nm[40];
    if((1<a)&&(a<64)&&(flga==0)){
        flg[a]=1; nm[25]=a;
        stp16();
        flg[a]=0;}
}
//

```

```

/** Check (n14+n30+n46+n62=130) *
void stp16(){
    if(nm[14]+nm[30]+nm[46]+nm[62]==LSM){stp17();}
}
//
/** Set n15=130-n13-n14-n16 & n15=130-n3-n7-n11 *
void stp17(){
    short a,b;
    a=LSM-nm[13]-nm[14]-nm[16];
    b=LSM-nm[3]-nm[7]-nm[11];
    if((a==b)&&(flga==0)){
        flg[a]=1; nm[15]=a;
        stp171();
        flg[a]=0;}
}
/** Set n37 *
void stp171(){
    short a;
    a=DD-nm[15];
    if((1<a)&&(a<64)&&(flga==0)){
        flg[a]=1; nm[37]=a;
        stp172();
        flg[a]=0;}
}
/** Set n50 *
void stp172(){
    short a;
    a=65-nm[15];
    if((1<a)&&(a<64)&&(flga==0)){
        flg[a]=1; nm[50]=a;
        stp173();
        flg[a]=0;}
}
/** Set n28 *
void stp173(){
    short a;
    a=65-nm[37];
    if((1<a)&&(a<64)&&(flga==0)){
        flg[a]=1; nm[28]=a;
        stp18();
        flg[a]=0;}
}

```

```

}
//
/** Check (n15+n31+n47+n63=130) *
void stp18(){
    if(nm[15]+nm[31]+nm[47]+nm[63]==LSM){stp19();}
}
/** Pantriagonal Checks 1 *
void stp19(){
    short sm1,sm2,sm3;
    sm1=nm[1]+nm[24]+nm[43]+nm[62];
    sm2=nm[1]+nm[30]+nm[43]+nm[56];
    sm3=nm[1]+nm[32]+nm[43]+nm[54];
    if((sm1==LSM)&&(sm2==LSM)&&(sm3==LSM)){stp20();}
}
/** Pantriagonal Checks 2 *
void stp20(){
    short sm1,sm2,sm3;
    sm1=nm[2]+nm[23]+nm[44]+nm[61];
    sm2=nm[2]+nm[31]+nm[44]+nm[53];
    sm3=nm[2]+nm[29]+nm[44]+nm[55];
    if((sm1==LSM)&&(sm2==LSM)&&(sm3==LSM)){stp21();}
}
/** Pantriagonal Checks 3 *
void stp21(){
    short sm1,sm2,sm3;
    sm1=nm[3]+nm[22]+nm[41]+nm[64];
    sm2=nm[3]+nm[32]+nm[41]+nm[54];
    sm3=nm[3]+nm[30]+nm[41]+nm[56];
    if((sm1==LSM)&&(sm2==LSM)&&(sm3==LSM)){stp22();}
}
/** Pantriagonal Checks 4 *
void stp22(){
    short sm1,sm2,sm3;
    sm1=nm[4]+nm[21]+nm[42]+nm[63];
    sm2=nm[4]+nm[29]+nm[42]+nm[55];
    sm3=nm[4]+nm[31]+nm[42]+nm[53];
    if((sm1==LSM)&&(sm2==LSM)&&(sm3==LSM)){cnt++; cnt2++;
        cn1[nm[1]]++; cn43[nm[43]]++;
        if(cnt2==1){ansprint();}
    }
}
//
/** Print the Result *
void ansprint(){
    short n,m;
    d[cnt3][0]=cnt;
    for(n=1;n<65;n++){d[cnt3][n]=nm[n];}
    cnt3++;
    if(cnt3==3){pr3sol(); cnt3=0;
        for(m=0;m<4;m++){for(n=0;n<65;n++){d[m][n]=0;}
        }
    }
}
//
/** Print 3 Answers in a line *
void pr3sol(){
    printf("%%4d#%%41d#%%41d#\n",d[0][0],d[1][0],d[2][0]);
    printf("%%4d-----%2d-----%2d-----%2d",d[0][1],d[0][17],d[0][33],d[0][49]);
    printf("%%12d-----%2d-----%2d-----%2d",d[1][1],d[1][17],d[1][33],d[1][49]);
    printf("%%12d-----%2d-----%2d-----%2d\n",d[2][1],d[2][17],d[2][33],d[2][49]);
    printf("//    %2d %9d %9d    %2d",d[0][2],d[0][18],d[0][34],d[0][50]);
    printf("        %2d %9d %9d    %2d",d[1][2],d[1][18],d[1][34],d[1][50]);
    printf("        %2d %9d %9d    %2d\n",d[2][2],d[2][18],d[2][34],d[2][50]);
    printf("%%4d %3d %5d %3d %5d %3d %5d %3d",
        d[0][5],d[0][3],d[0][21],d[0][19],d[0][37],d[0][35],d[0][53],d[0][51]);
}

```

```

printf("%8d %3d %5d %3d %5d %3d %5d %3d",
    d[1][5],d[1][3],d[1][21],d[1][19],d[1][37],d[1][35],d[1][53],d[1][51]);
printf("%8d %3d %5d %3d %5d %3d %5d %3d\n",
    d[2][5],d[2][3],d[2][21],d[2][19],d[2][37],d[2][35],d[2][53],d[2][51]);
printf("// %2d %2d---%2d--%2d---%2d--%2d---%2d--%2d",
    d[0][6],d[0][4],d[0][22],d[0][20],d[0][38],d[0][36],d[0][54],d[0][52]);
printf(" %2d %2d---%2d--%2d---%2d--%2d---%2d--%2d",
    d[1][6],d[1][4],d[1][22],d[1][20],d[1][38],d[1][36],d[1][54],d[1][52]);
printf(" %2d %2d---%2d--%2d---%2d--%2d---%2d--%2d\n",
    d[2][6],d[2][4],d[2][22],d[2][20],d[2][38],d[2][36],d[2][54],d[2][52]);
printf("//%4d %3d %4d %3d %5d %3d %5d %3d |",
    d[0][9],d[0][7],d[0][25],d[0][23],d[0][41],d[0][39],d[0][57],d[0][55]);
printf("%6d %3d %4d %3d %5d %3d %5d %3d |",
    d[1][9],d[1][7],d[1][25],d[1][23],d[1][41],d[1][39],d[1][57],d[1][55]);
printf("%6d %3d %4d %3d %5d %3d %5d %3d | \n",
    d[2][9],d[2][7],d[2][25],d[2][23],d[2][41],d[2][39],d[2][57],d[2][55]);
printf("// %2d %3d %5d %3d %5d %3d %5d %3d |%2d %3d",
    d[0][10],d[0][8],d[0][26],d[0][24],d[0][42],d[0][40],d[0][58],d[0][56]);
printf(" %2d %3d %5d %3d %5d %3d %5d %3d |%2d %3d",
    d[1][10],d[1][8],d[1][26],d[1][24],d[1][42],d[1][40],d[1][58],d[1][56]);
printf(" %2d %3d %5d %3d %5d %3d %5d %3d |%2d %3d\n",
    d[2][10],d[2][8],d[2][26],d[2][24],d[2][42],d[2][40],d[2][58],d[2][56]);
printf("//%4d--%2d-|-%2d--%2d---%2d--%2d---%2d %3d |",
    d[0][13],d[0][11],d[0][29],d[0][27],d[0][45],d[0][43],d[0][61],d[0][59]);
printf("%6d--%2d-|-%2d--%2d---%2d--%2d---%2d %3d |",
    d[1][13],d[1][11],d[1][29],d[1][27],d[1][45],d[1][43],d[1][61],d[1][59]);
printf("%6d--%2d-|-%2d--%2d---%2d--%2d---%2d %3d | \n",
    d[2][13],d[2][11],d[2][29],d[2][27],d[2][45],d[2][43],d[2][61],d[2][59]);
printf("//%6d %3d %5d %3d %5d %3d %5d %3d",
    d[0][14],d[0][12],d[0][30],d[0][28],d[0][46],d[0][44],d[0][62],d[0][60]);
printf(" %4d %3d %5d %3d %5d %3d %5d %3d",
    d[1][14],d[1][12],d[1][30],d[1][28],d[1][46],d[1][44],d[1][62],d[1][60]);
printf(" %4d %3d %5d %3d %5d %3d %5d %3d\n",
    d[2][14],d[2][12],d[2][30],d[2][28],d[2][46],d[2][44],d[2][62],d[2][60]);
printf("//%8d %8d %9d %9d |",
    d[0][15],d[0][31],d[0][47],d[0][63]);
printf("%10d %8d %9d %9d |",
    d[1][15],d[1][31],d[1][47],d[1][63]);
printf("%10d %8d %9d %9d | \n",
    d[2][15],d[2][31],d[2][47],d[2][63]);
printf("//%10d-----%2d-----%2d-----%2d",
    d[0][16],d[0][32],d[0][48],d[0][64]);
printf("%12d-----%2d-----%2d-----%2d",
    d[1][16],d[1][32],d[1][48],d[1][64]);
printf("%12d-----%2d-----%2d-----%2d\n",
    d[2][16],d[2][32],d[2][48],d[2][64]);
}
//
/* Print the Counts according to n1 *
void prcn1(){
short d,n;
printf("//");
for(n=1;n<36;n++){d=cn1[n];
if(d>0){printf("%4d:%5d," ,n,d);}
else{printf("%4d: ....," ,n);}
if(n%7==0){printf("\n//");}
}
}
//
/* Print the Counts according to n43 *
void prcn43(){
short d,n;
printf("//");
for(n=1;n<65;n++){d=cn43[n];
if(d>0){printf("%4d:%4d," ,n,d);}
}
}

```

```

else{printf("%4d: ...",n);}
if(n%8==0){printf("\n//");}
}
}
//

```

n43=64 の場合を全然調べていないのはおかしいとお思いかもしいかもしれませんが、これは心配要らない。n1+n43=65; n1+n64=65; が同時に成り立つことはあり得ないからだ。

興味深いのは、計数結果だ。次のリストを見てほしい。抜粋表ではあるが、全体の特徴が良くわかるように工夫してみた。

**\* 連立型の汎対称立体四方陣 (n1=1 のタイプ) \***

[Abstract List of Solutions when n1=1 & n64=64 & n43=32]

<p>1#</p> <pre> 1-----52-----16-----61  58      23      43        6 55 8 26 41 38 21 11 60  20 63---33---14---29---50---48---3 12 46   37 31 25 35 56 18    47 9 30 40 34 28  19 53 62-17- 15-36-51-32- 2 45    5 54 44 27 24 39 57 10  59   22 42 7    4-----49-----13-----64 </pre>	<p>67#</p> <pre> 1-----46-----24-----59  58      20      47        5 52 8 26 43 37 17 15 62  14 63---33---21---27---42---56---4 16 53   35 31 25 36 54 10    55 11 29 40 34 30  12 49 61-9- 23-38-44-32- 2 51    3 50 48 28 22 39 57 13  60   18 45 7    6-----41-----19-----64 </pre>	<p>85#</p> <pre> 1-----54-----16-----59  58      23      45        4 55 8 26 41 36 19 13 62  22 63---33---12---27---50---48---5 14 44   35 31 25 37 56 18    47 9 28 40 34 30  21 51 60-17- 15-38-53-32- 2 43    3 52 46 29 24 39 57 10  61   20 42 7    6-----49-----11-----64 </pre>
<p>151#</p> <pre> 1-----44-----30-----55  52      22      47        9 54 14 20 39 41 17 15 60  12 63---33---25---23---36---62---6 16 57   37 31 19 38 58 4    61 7 27 46 34 28   8 49 59-3- 29-42-40-32- 2 53    5 50 48 24 26 45 51 11  56   18 43 13    10-----35-----21-----64 </pre>	<p>163#</p> <pre> 1-----58-----16-----55  54      27      45        4 59 12 22 37 36 19 13 62  26 63---33---8---23---50---48---9 14 40   35 31 21 41 60 18    47 5 24 44 34 30  25 51 56-17- 15-42-57-32- 2 39    3 52 46 29 28 43 53 6  61   20 38 11    10-----49-----7-----64 </pre>	<p>217#</p> <pre> 1-----52-----30-----47  40      14      59       17 46 26 8 43 49 5 27 56  20 63---33---21---15---36---62---10 28 53   41 31 7 42 54 4    61 11 23 58 34 24  12 37 55-3- 29-50-44-32- 2 45    9 38 60 16 22 57 39 19  48   6 51 25    18-----35-----13-----64 </pre>
<p>223#</p> <pre> 1-----58-----24-----47  46      27      53        4 59 20 14 37 36 11 21 62  26 63---33---8---15---42---56---17 22 40   35 31 13 49 60 10    55 5 16 52 34 30  25 43 48-9- 23-50-57-32- 2 39    3 44 54 29 28 51 45 6  61   12 38 19    18-----41-----7-----64 </pre>	<p>271#</p> <pre> 1-----47-----24-----58  59      20      46        5 52 8 27 42 37 17 14 63  15 62---33---21---26---43---56---4 16 53   34 30 25 36 55 11    54 10 29 40 35 31  12 49 61-9- 22-39-44-32- 3 50    2 51 48 28 23 38 57 13  60   19 45 6    7-----41-----18-----64 </pre>	<p>289#</p> <pre> 1-----55-----16-----58  59      22      45        4 54 8 27 41 36 18 13 63  23 62---33---12---26---51---48---5 15 44   34 30 25 37 56 19    46 9 28 40 35 31  21 50 60-17- 14-39-53-32- 3 42    2 52 47 29 24 38 57 11  61   20 43 6    7-----49-----10-----64 </pre>
<p>349#</p> <pre> 1-----44-----31-----54  52      23      46        9 55 15 20 38 41 17 14 60  12 62---33---25---22---36---63---7 16 57   37 30 18 39 59 4    61 6 26 47 35 28   8 49 58-2- 29-43-40-32- 3 53    5 51 48 24 27 45 50 10  56   19 42 13    11-----34-----21-----64 </pre>	<p>361#</p> <pre> 1-----59-----16-----54  55      26      45        4 58 12 23 37 36 18 13 63  27 62---33---8---22---51---48---9 15 40   34 30 21 41 60 19    46 5 24 44 35 31  25 50 56-17- 14-43-57-32- 3 38    2 52 47 29 28 42 53 7  61   20 39 10    11-----49-----6-----64 </pre>	<p>475#</p> <pre> 1-----47-----24-----58  60      22      45        3 54 8 28 42 35 17 13 63  15 61---33---19---26---44---56---6 16 51   34 29 25 38 55 12    53 10 27 40 36 31  14 49 59-9- 21-39-46-32- 4 50    2 52 48 30 23 37 57 11  62   20 43 5    7-----41-----18-----64 </pre>
<p>493#</p> <pre> 1-----47-----24-----58  61      22      44        3 54 8 29 42 35 17 12 63  15 60---33---19---26---45---56---6 16 51   34 28 25 38 55 13    52 10 27 40 37 31  14 49 59-9- 20-39-46-32- 5 50    2 53 48 30 23 36 57 11  62   21 43 4    7-----41-----18-----64 </pre>	<p>631#</p> <pre> 1-----50-----25-----54  58      28      41        3 60 12 26 36 35 19 9 63  18 59---33---16---22---45---57---10 14 48   34 27 21 42 61 13    52 4 23 44 38 31  17 51 55-8- 20-43-49-32- 6 47    2 56 46 30 29 39 53 5  62   24 37 7    11-----40-----15-----64 </pre>	<p>649#</p> <pre> 1-----46-----31-----52  56      27      42        5 59 15 24 36 37 17 10 62  14 58---33---21---20---40---63---11 16 53   35 26 18 43 61 8    57 4 22 47 39 30  12 49 54-2- 25-45-44-32- 7 51    3 55 48 28 29 41 50 6  60   23 38 9    13-----34-----19-----64 </pre>

661#

```

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

```

667#

```

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

```

751#

```

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

```

757#

```

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

```

763#

```

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

```

769#

```

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

```

[Solution Counts = 804]

[Abstract List of Solutions when n1=1 & n64=64 & n43=48]

1#

```

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

```

67#

```

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

```

85#

```

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

```

145#

```

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

```

157#

```

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

```

211#

```

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

```

217#

```

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

```

265#

```

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

```

283#

```

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

```

343#

```

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

```

355#

```

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

```

463#

```

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

```

481#

```

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

```

619#

```

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

```

631#

```

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

```

643#

```

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

```

727#

```

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

```

733#

```

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

```

739#

```

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

```

745#

```

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

```

[Solution Counts = 780]

[Abstract List of Solutions when n1=1 & n64=64 & n43=56]

1#

```

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

```

67#

```

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

```

85#

```

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

```

145#

```

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

```

157#

```

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

```

211#

```

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

```

217#

```

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

```

265#

```

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

```

283#

```

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

```

343#

```

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

```

355#

```

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

```

463#

```

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

```

481#

```

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

```

619#

```

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

```

631#

```

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

```

643#

```

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

```

727#

```

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

```

733#

```

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

```

739#

```

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

```

745#

```

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

```

[Solution Counts = 780]

[Abstract List of Solutions when n1=1 & n64=64 & n43=60]

1#

```

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

```

67#

```

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

```

85#

```

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

```

145#

```

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

```

157#

```

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

```

211#

```

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

```

217#

```

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

```

265#

```

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

```

283#

```

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

```

343#

```

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

```

355#

```

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

```

463#

```

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

```

481#

```

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

```

619#

```

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

```

631#

```

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

```

643#

```

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

```

727#

```

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

```

733#

```

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

```

739#

```

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

```

745#

```

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

```

[Solution Counts = 780]

[Abstract List of Solutions when n1=1 & n64=64 & n43=62]

1#

```

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

```

67#

```

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

```

85#

```

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

```



145#

```

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

```

157#

```

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

```

211#

```

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

```

217#

```

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

```

265#

```

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

```

283#

```

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

```

343#

```

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

```

355#

```

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

```

463#

```

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

```

481#

```

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

```

619#

```

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

```

631#

```

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

```

643#

```

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

```

727#

```

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

```

733#

```

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

```

739#

```

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

```

745#

```

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

```

[Solution Counts = 780]

[Abstract List of Solutions when n1=1 & n64=64 & n43=63]

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

67#  
 1-----28-----47-----54  
 |52-----39-----30-----9  
 40 15 51 22 10 33 29 60  
 |27 62-----2-41-----53-20-----48-7  
 31 42 | 6 61 49 8 44 19 |  
 |46 21 57 16 4 59 |23 34  
 58-17-|45-12-24-63-3 38 |  
 5 36 32 55 43 14 50 25  
 56 | 35 26 13 |  
 11-----18-----37-----64

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

151#  
 1-----24-----59-----46  
 |40-----43-----30-----17  
 44 27 39 14 18 33 29 56  
 |23 62-----2-49-----45-8-----60-11  
 31 50 | 10 61 37 12 52 7 |  
 |58 13 53 28 4 55 |15 34  
 54-5-|57-20-16-63-3 42 |  
 9 36 32 47 51 26 38 21  
 48 | 35 22 25 |  
 19-----6-----41-----64

163#  
 1-----52-----31-----46  
 |44-----53-----26-----7  
 54 23 43 10 8 37 25 60  
 |51 62-----2-15-----45-36-----32-17  
 27 16 | 6 61 41 18 56 35 |  
 |30 9 47 24 4 59 |49 38  
 48-33-|29-20-50-63-3 14 |  
 5 40 28 57 55 22 42 11  
 58 | 39 12 21 |  
 19-----34-----13-----64

217#  
 1-----40-----59-----30  
 |16-----27-----54-----33  
 28 51 15 22 34 9 53 48  
 |39 62-----2-41-----29-8-----60-19  
 55 42 | 18 61 13 20 44 7 |  
 |58 21 45 52 4 47 |23 10  
 46-5-|57-36-24-63-3 26 |  
 17 12 56 31 43 50 14 37  
 32 | 11 38 49 |  
 35-----6-----25-----64

223#  
 1-----52-----47-----30  
 |28-----53-----42-----7  
 54 39 27 10 8 21 41 60  
 |51 62-----2-15-----29-20-----48-33  
 43 16 | 6 61 25 34 56 19 |  
 |46 9 31 40 4 59 |49 22  
 32-17-|45-36-50-63-3 14 |  
 5 24 44 57 55 38 26 11  
 58 | 23 12 37 |  
 35-----18-----13-----64

271#  
 1-----30-----47-----52  
 |54-----39-----28-----9  
 40 15 53 20 10 33 27 62  
 |29 60-----2-41-----51-22-----48-7  
 31 42 | 4 59 49 8 46 21 |  
 |44 19 57 16 6 61 |23 34  
 58-17-|43-14-24-63-5 36 |  
 3 38 32 55 45 12 50 25  
 56 | 37 26 11 |  
 13-----18-----35-----64

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

349#  
 1-----24-----61-----44  
 |40-----45-----28-----17  
 46 29 39 12 18 33 27 56  
 |23 60-----2-49-----43-8-----62-13  
 31 50 | 10 59 35 14 54 7 |  
 |58 11 51 30 6 55 |15 34  
 52-3-|57-22-16-63-5 42 |  
 9 38 32 47 53 26 36 19  
 48 | 37 20 25 |  
 21-----4-----41-----64

361#  
 1-----54-----31-----44  
 |46-----51-----26-----7  
 52 23 45 10 8 35 25 62  
 |53 60-----2-15-----43-38-----32-17  
 29 16 | 4 59 41 18 56 37 |  
 |28 9 47 24 6 61 |49 36  
 48-33-|27-22-50-63-5 12 |  
 3 40 30 57 55 20 42 13  
 58 | 39 14 19 |  
 21-----34-----11-----64

475#  
 1-----30-----47-----52  
 |56-----43-----26-----5  
 44 15 55 20 6 33 25 62  
 |29 58-----2-37-----51-24-----48-11  
 31 38 | 4 57 49 12 46 23 |  
 |42 19 53 16 8 61 |27 34  
 54-17-|41-14-28-63-7 36 |  
 3 40 32 59 45 10 50 21  
 60 | 39 22 9 |  
 13-----18-----35-----64

493#  
 1-----30-----47-----52  
 |58-----43-----24-----5  
 44 15 57 20 6 33 23 62  
 |29 56-----2-37-----51-26-----48-11  
 31 38 | 4 55 49 12 46 25 |  
 |40 19 53 16 10 61 |27 34  
 54-17-|39-14-28-63-9 36 |  
 3 42 32 59 45 8 50 21  
 60 | 41 22 7 |  
 13-----18-----35-----64

631#  
 1-----36-----49-----44  
 |52-----55-----18-----5  
 56 23 51 8 6 37 17 62  
 |35 54-----2-31-----43-26-----50-19  
 27 32 | 4 53 41 20 58 25 |  
 |40 7 45 24 12 61 |33 38  
 46-15-|39-22-34-63-11 30 |  
 3 48 28 59 57 14 42 9  
 60 | 47 10 13 |  
 21-----16-----29-----64

649#  
 1-----28-----61-----40  
 |48-----53-----20-----9  
 54 29 47 8 10 33 19 60  
 |27 52-----2-41-----39-16-----62-21  
 31 42 | 6 51 35 22 58 15 |  
 |50 7 43 30 14 59 |23 34  
 44-3-|49-26-24-63-13 38 |  
 5 46 32 55 57 18 36 11  
 56 | 45 12 17 |  
 25-----4-----37-----64

661#  
 1-----52-----53-----24  
 |36-----55-----32-----7  
 56 43 35 6 8 19 31 62  
 |51 50-----2-17-----23-26-----54-37  
 45 18 | 4 49 21 38 60 25 |  
 |40 5 27 44 16 61 |47 20  
 28-11-|39-42-48-63-15 14 |  
 3 34 46 57 59 30 22 9  
 58 | 33 10 29 |  
 41-----12-----13-----64

667#  
 1-----30-----55-----44  
 |58-----51-----16-----5  
 52 23 57 12 6 33 15 62  
 |29 48-----2-37-----43-26-----56-19  
 31 38 | 4 47 41 20 54 25 |  
 |40 11 45 24 18 61 |27 34  
 46-9-|39-22-28-63-17 36 |  
 3 50 32 59 53 8 42 13  
 60 | 49 14 7 |  
 21-----10-----35-----64

751#  
 1-----54-----59-----16  
 |32-----43-----38-----17  
 44 51 31 8 18 9 37 62  
 |53 46-----2-25-----15-24-----60-35  
 55 26 | 4 45 13 36 58 23 |  
 |42 7 29 52 20 61 |39 10  
 30-5-|41-50-40-63-19 12 |  
 3 28 56 47 57 34 14 21  
 48 | 27 22 33 |  
 49-----6-----11-----64

757#  
 1-----52-----61-----16  
 |32-----45-----36-----17  
 46 53 31 8 18 9 35 60  
 |51 44-----2-25-----15-24-----62-37  
 55 26 | 6 43 11 38 58 23 |  
 |42 7 27 54 22 59 |39 10  
 28-3-|41-50-40-63-21 14 |

763#  
 1-----52-----61-----16  
 |32-----45-----36-----17  
 46 57 31 12 18 5 35 56  
 |51 40-----2-21-----15-28-----62-41  
 59 22 | 10 39 7 42 54 27 |  
 |38 11 23 58 26 55 |43 6  
 24-3-|37-50-44-63-25 14 |

769#  
 1-----46-----55-----28  
 |58-----51-----16-----5  
 52 39 57 12 6 17 15 62  
 |45 32-----2-21-----27-42-----56-35  
 47 22 | 4 31 25 36 54 41 |  
 |24 11 29 40 34 61 |43 18  
 30-9-|23-38-44-63-33 20 |



<pre> 1-----44-----35-----50  42      13      58      17 14 25   41 46   18 5   57 54  43 62---2-27---49-32---36-9 59 28   12 61   39 10   20 31    34 45   55 26   4 53    37 6 56-29- 33-16-38-63-3 22    11 8   60 47   19 24   40 51    48     7     52     23      15-----30-----21-----64 </pre>	<p style="text-align: center;">2#</p> <p style="text-align: center;">[二進法分解図] 1L</p> <pre> 0-1-1-1-1  1 0 1 0 0 0 1 1 0 0 1 1  1 1-0-0-1-0-1-0 1 0  0 1 1 0 0 0   1 1 1 0 0 1 1 0 1-0 -1-0-1-1-0 0  0 0 1 1 0 0 1 1 1  0 1 0 0  0-0-0-0-1 </pre>	<p style="text-align: center;">2L</p> <pre> 0-0-0-0-1  0 0 1 1 0 1 0 0 1 0 1 1  0 1-0-1-1-1-0-0 1 1  0 1 0 0 1 1   0 0 1 1 0 1 0 0 1-1 -0-0-0-1-0 1  0 0 1 0 1 1 0 1 0  0 1 1 1  0-1-1-1-1 </pre>	
<p style="text-align: center;">3L</p> <pre> 0-1-0-0  1 1 1 0 1 1 1 1 0 0 1 0  1 1-0-1-0-1-0-1 1 1  1 1 0 1 0 1   0 1 0 1 0 0 0 0 0-1 -0-1-0-1-0 0  1 0 1 1 0 0 0 0 1  0 0 0 0  1-1-0-1 </pre>	<p style="text-align: center;">4L</p> <pre> 0-0-0-0  0 1 0 0 1 0 0 1 0 1 0 1  0 1-0-0-0-1-0-0 0 0  0 1 1 0 0 1   0 1 1 0 0 1 1 1 1-1 -0-1-1-1-0 1  0 1 0 1 0 1 1 0 1  1 0 1 1  1-1-1-1-1 </pre>	<p style="text-align: center;">5L</p> <pre> 0-1-1-0  0 0 0 0 0 0 0 0 0 0 0 0  1 0-0-1-0-1-1-0 1 1  1 0 1 0 1 1   0 0 1 0 1 0 0 0 1-0 -0-1-0-1-1 0  1 1 1 1 1 1 1 1 1  1 1 1 1  1-0-0-0-1 </pre>	<p style="text-align: center;">6L</p> <pre> 0-1-0-1  1 0 1 0 1 0 0 1 1 0 0 1  0 1-1-0-0-1-1-0 0 1  1 0 0 1 1 0   1 0 0 1 1 0 0 1 1-0 -0-1-1-0-0 1  0 1 1 0 0 1 1 0 1  0 1 1 0  0-1-0-1-1 </pre>

<pre> 1-----20-----59-----50  32      47      24      27 48 35   31 12   28 39   23 44  19 62---2-51---49-8---60-9 25 52   22 61   29 10   54 7    58 11   55 36   4 43    13 40 56-5- 57-16-14-63-3 46    21 42   26 37   53 34   30 17    38     41     18     33      15-----6-----45-----64 </pre>	<p style="text-align: center;">3#</p> <p style="text-align: center;">[二進法分解図] 1L</p> <pre> 0-0-1-1  0 1 0 0 1 1 0 0 0 1 0 1  0 1-0-1-1-0-1-0 0 1  0 1 0 0 1 0   1 0 1 1 0 1 0 1 1-0 -1-0-0-1-0 1  0 1 0 1 1 1 0 0 1  1 0 1 1  0-0-1-1 </pre>	<p style="text-align: center;">2L</p> <pre> 0-1-1-1  1 0 1 1 0 0 1 0 1 0 1 0  1 1-0-1-1-0-1-0 1 1  1 1 1 0 1 0   1 0 1 0 0 0 0 0 1-0 -1-0-0-1-0 0  1 0 1 0 1 0 1 1 0  0 1 1 0  0-0-0-0-1 </pre>	
<p style="text-align: center;">3L</p> <pre> 0-0-1-0  1 1 0 1 1 0 1 1 1 0 0 1  0 1-0-0-0-0-1-1 1 0  0 1 1 1 0 0   1 1 0 0 0 1 1 0 0-0 -1-1-1-1-0 1  0 1 1 0 0 0 1 0 0  1 0 0 0  1-0-1-1-1 </pre>	<p style="text-align: center;">4L</p> <pre> 0-0-0-0  1 1 1 0 1 0 1 0 0 1 1 0  0 1-0-0-0-1-0-0 0 0  1 1 1 0 1 1   0 0 1 0 0 0 1 1 1-1 -0-1-1-1-0 1  1 0 0 1 1 0 1 0 1  0 0 1 0  1-1-1-1-1 </pre>	<p style="text-align: center;">5L</p> <pre> 0-1-1-0  1 1 1 1 1 1 1 1 1 1 1 1  1 0-0-1-0-1-1-0 0 1  0 0 0 0 0 1   0 1 1 1 1 1 0 1 1-0 -0-1-0-1-1 0  0 0 0 0 0 0 0 0 0  0 0 0 0  1-0-0-0-1 </pre>	<p style="text-align: center;">6L</p> <pre> 0-1-0-1  1 0 1 0 1 0 0 1 1 0 0 1  0 1-1-0-0-1-1-0 0 1  1 0 0 1 1 0   1 0 0 1 1 0 0 1 1-0 -0-1-1-0-0 1  0 1 1 0 0 1 1 0 1  0 1 1 0  0-1-0-1-1 </pre>

<pre> 1-----54-----25-----50  32      13      58      27 14 35   31 46   28 5   57 44  53 62---2-17---49-42---26-9 59 18   22 61   29 10   20 41    24 45   55 36   4 43    47 6 56-39- 23-16-48-63-3 12    21 8   60 37   19 34   30 51    38     7     52     33      15-----40-----11-----64 </pre>	<p style="text-align: center;">4#</p> <p style="text-align: center;">[二進法分解図] 1L</p> <pre> 0-1-0-1  0 0 1 0 0 1 0 1 0 0 1 1  1 1-0-0-1-1-0-0 1 0  0 1 0 0 0 1   0 1 1 1 0 1 1 0 1-1 -0-0-1-1-0 0  0 0 1 1 0 1 0 1 1  0 1 1 1  0-1-0-1 </pre>	<p style="text-align: center;">2L</p> <pre> 0-1-1-1  1 0 1 1 0 0 1 0 1 0 1 0  1 1-0-1-1-0-1-0 1 1  1 1 1 0 1 0   1 0 1 0 0 0 0 0 1-0 -1-0-0-1-0 0  1 0 1 0 1 0 1 1 0  0 1 1 0  0-0-0-0-1 </pre>	
<p style="text-align: center;">3L</p> <pre> 0-0-1-0  1 1 1 1 1 0 1 1 1 0 1 1  0 1-0-0-0-1-1-1 1 0  0 1 1 1 0 1   0 1 0 0 0 1 1 0 0-1 -0-1-1-1-0 1  1 1 1 0 0 0 1 1 1  1 1 1 0  1-1-1-1-1 </pre>	<p style="text-align: center;">4L</p> <pre> 0-1-0-0  1 1 0 0 1 0 1 1 0 1 0 0  1 1-0-0-0-0-0-0 0 0  1 1 1 0 0 0   1 1 1 0 0 0 1 1 1-1 -0-1-1-1-0 1  1 0 1 1 0 0 1 1 1  1 1 1 0  1-1-1-1-1 </pre>	<p style="text-align: center;">5L</p> <pre> 0-0-0-0  1 0 0 1 0 1 1 0 1 0 0 1  0 0-0-0-0-0-0-0 1 0  0 0 0 0 1 0   1 0 1 1 1 1 1 0 1-0 -0-1-1-1-1 0  1 0 1 1 1 1 1 0 1  0 1 1 1  1-0-0-0-1 </pre>	<p style="text-align: center;">6L</p> <pre> 0-1-0-1  1 0 1 0 1 0 0 1 1 0 0 1  0 1-1-0-0-1-1-0 0 1  1 0 0 1 1 0   1 0 0 1 1 0 0 1 1-0 -0-1-1-0-0 1  1 0 0 1 1 0 1 0 1  0 1 1 0  0-1-0-1-1 </pre>

```

0-0|-0-1--1-1-0 1| 1-1|-1-1--1-1-0 0| 1-1|-1-1--1-1-1 1| 1-0|-0-1--1-0-0 1|
0 0 1 0 0 0 1 0 1 1 0 1 0 0 1 0 0 1 1 0 0 1 1 0
0| 0 0 0 0| 1| 1 0 0 0| 0| 1 1 1 0| 1| 0 1 1 0|
1---0---1---1 1---1---0---1 1---1---1---1 0---1---0---1

```

```

. . . . .
                                24#           [二進法分解図] 1L           2L
1-----62-----43-----24           0---1---1---0           0---1---0---1
|26      55      32      |17           |0  1  0  |0           |1  1  1  |1
56 53  25  6  18 19  31 52           1 1 0 0 0 0 0 1 1 1 1 0 1 1 1 1
|61 50-- 2-- 7--23-36--44-37           |1 1-0-0-0-1-1-1| |1 1-0-0-1-0-0-0
45  8 | 14 49  11 38  60 35 |           1 0| 0 1 0 1 1 1| 0 0| 0 1 0 0 1 0|
|30  5 | 27 54  16 51  |57 20           |0 0 0 1 0 1 |1 0| |1 0 1 1 0 1 |1 1
28-21-|-29-42--58-63--15  4 |           0-0|-0-1--1-1-0 0| 1-1|-1-0--1-1-0 0|
13 34  46 47  59 40  12  9           0 1 1 1 1 1 0 0 0 0 0 0 1 0 0 0
 48 |   33      10      39 |           1|  1  0  1|  0|  0  0  0|
 41-----22----- 3-----64           1---0---0---1           0---1---0---1

                                3L           4L           5L           6L
0---1---1---0           0---1---0---1           0---0---1---1           0---1---0---1
|1  0  1  |0  |0  1  1  |0           |0  1  1  |0           |1  0  1  |0
0 0 1 0 0 0 1 0 1 1 0 1 0 0 1 0           1 0 0 0 0 1 1 1 1 0 0 1 1 0 0 1
|1 0-0-0-0-0-1-0| |1 0-0-1-1-0-0-1| |0 0-0-1-1-1-1-0| |0 1-1-0-0-1-1-0|
1 0| 1 0 1 0 1 0| 1 1| 1 0 0 1 0 0| 0 1| 0 0 1 0 1 1| 0 1| 1 0 0 1 1 0|
|1 0 1 0 1 0 |1 0| |1 1 0 1 1 0 |0 0| |0 0 1 0 1 1 |0 1| |1 0 0 1 1 0 |0 1
1-0|-1-1--1-1-1 0| 0-1|-1-0--0-1-1 0| 1-0|-0-0--0-1-1 1| 1-0|-0-1--1-0-0 1|
1 0 1 1 1 0 1 1 1 0 1 1 0 1 0 0           0 0 0 1 1 1 1 0 0 1 1 0 0 1 1 0
1|  0  1  0| 1|  0  0  1| 1|  0  0  1| 1|  0  0  1| 1|  0  1  1  0|
1---0---0---1 0---1---0---1 0---0---1---1 0---1---0---1

```

上図の**二進法分解図**を見ると、行列柱のあちこちに  $\{0, 1, 0, 0\}$  だとか  $\{1, 1, 0, 1\}$  だとかがある。 $\{1, 1, 1, 1\}$  さえもが見られる。「完全オイラー方陣」の厳しい条件に必ずしも従っていないことが、見て取れよう。48個もこういう異端児が居たのである。

ただし、割合にして言えば、わずかに1%だ。あとの99%もが「完全オイラー方陣」だったのだ。この調査結果には、驚いた。まったく予想外であった。

平面八方陣なみの規模を持ちながら、あくまで立体**四方陣**であるせいかも知れない。

ちなみに平面方陣では、四次と五次には「非オイラー型」の汎魔方陣は存在しない。全てが「完全オイラー方陣」なのだ。七次・八次以上の汎魔方陣になると、逆転して非オイラー型の方が圧倒的に多い。完全オイラー方陣は極めて珍しい存在になってしまう。

かくして我が**連立型の汎対称立体四方陣**は、極めて珍しい存在ながら、ほとんど全てが(約99%が)**完全オイラー型**であることがわかった。

特に珍重してよい方陣だと思う。

(初版: Jan. 17, 2005: MacOSX & Xcode 1.5; 改訂版: Jul. 24, 2010 with Xcode 3.1.2;

最新改訂清書版: Jul. 24, 2017 with macOS Sierra 10.12.6 and Xcode 7.3;

著者: 摂田 寛二(Kanji Setsuda): jag12001@nifty.com)