第７章： 多次元超立体魔方陣の最新研究：摂田 寛二
第４節： 三連立型正方連結対称汎八方陣の「根源方陣」

六次元超立体二方陣の構成とそこからの変換によって，はたして三連立型の正方連結対称汎八方陣を 5760 個得ることができるであろうか。二者間の一対一対応を確認することができるであろうか。前者は後者の「根源方陣」であろうか。前節の宿題を解決したい。

１．六次元超立体二方陣の構成は，今までの成果をそのまま活用する。
例えば，定義条件式 50 本などは，前節のままを踏襲する。
ただ，基本図を下図のように変える。変換の仕方が，シンプルに見えるからだ。

```
BD/2^6
   1---- 2       3---- 4       33----34      35----36
   |  5--+- 6    |  7--+- 8    | 37--+-38    | 39--+-40
   9--|-10  |   11--|-12  |    41--|-42  |   43--|-44  |
     13----14      15----16      45----46      47----48

  17----18      19----20      49----50      51----52
   | 21--+-22   | 23--+-24   | 53--+-54   | 55--+-56
  25--|-26  |   27--|-28  |   57--|-58  |   59--|-60  |
     29----30      31----32      61----62      63----64

  1---------- 2---------- 3---------- 4        SC4^3/8^2
  |  5           6           7          |  8          1  2  3  4  5  6  7  8
  9     33     10     34     11     35     12     36       9 10 11 12 13 14 15 16
  | 13     37----14----38----15----39--+-16----40    17 18 19 20 21 22 23 24
 17     41   | 18     42     19     43     20     44  |   25 26 27 28 29 30 31 32
  | 21     45     22     46     23     47   | 24     48   33 34 35 36 37 38 39 40
 25----49--|-26----50----27----51----28     52  |    41 42 43 44 45 46 47 48
    29     53     30     54     31     55     32     56   49 50 51 52 53 54 55 56
       57   |        58           59           60  |    57 58 59 60 61 62 63 64
            61----------62----------63----------64
```

そして懸案の「変換法」を次のようにする．左図の各ポジション(位置）を右図のように変えながら数値を置き換える。具体的には下のようにする。

```
** 三連立型正方連結対称汎八方陣に変換する方法 **
                        1/SC                        1/Sml
   1  2  3  4  5  6  7  8     1  2  4  3  6  5  7  8
   9 10 11 12 13 14 15 16     9 10 12 11 14 13 15 16
  17 18 19 20 21 22 23 24    25 26 28 27 30 29 31 32
  25 26 27 28 29 30 31 32    17 18 20 19 22 21 23 24
  33 34 35 36 37 38 39 40    41 42 44 43 46 45 47 48
  41 42 43 44 45 46 47 48    33 34 36 35 38 37 39 40
  49 50 51 52 53 54 55 56    49 50 52 51 54 53 55 56
  57 58 59 60 61 62 63 64    57 58 60 59 62 61 63 64

mm[1]=nm[1]; mm[2]=nm[2]; mm[3]=nm[4]; mm[4]=nm[3]; mm[5]=nm[6]; mm[6]=nm[5];
  mm[7]=nm[7]; mm[8]=nm[8]; mm[9]=nm[9]; mm[10]=nm[10]; mm[11]=nm[12]; mm[12]=nm[11];
    mm[13]=nm[14]; mm[14]=nm[13]; mm[15]=nm[15]; mm[16]=nm[16]; ....
```

根源方陣の最良の「ひな形」をさがす要領で，いろいろテストした結果，こうするのが良いとわかったからである。

早速にもプログラムを書こう。
今回は，原始解 46080 個を求めるのでなく，標準解 5760 個を求めることにする。リスト出力用の正規化不等式をプログラムの途中にどんどん入れて，実行スピードを速くする。解

のリストも，よりコンパクトにできる。
　変換図で見て，さいわいクリティカル・ポジションの n2, n8, n9, n57, n64 などに異動が無いので，立て方は簡単に n1＜n8; n1＜n57; n1＜n64; n2＞n9; として，問題は無かろう。

```
//** Compose the Extra-Cubic Magic Objects of Order 2^6 and **
//**  Transform them into '4-Type Simultaneous' Magic Squares 8^2: **
//**  Multiple, 'Composite', Self-Complementary and Pan-Diagonal; **
//**  List of the Standard Solutions with Check-sum Monitors  **
//** File: 'ECO26Sml4T.c' Dictated by Kanji Setsuda on **
//**  Apr.24, 2005 and Feb.28, 2006 with MacOSX & Xcode 1.5; **
//**  Recompiled on Mar. 4, 2015 with MacOSX 10.10.2 & Xcode 6.1.1 **
//
//* Definitions of Basic Diagrams for ECO2^6 */
// Type 0/2^6
//    1---- 2      3---- 4      33----34     35----36
//    |  5--+- 6   |  7--+- 8    | 37--+-38   | 39--+-40
//    9--|-10  |  11--|-12  |   41--|-42  |  43--|-44  |
//       13----14     15----16      45----46     47----48
//
//   17----18     19----20      49----50     51----52
//    | 21--+-22   | 23--+-24    | 53--+-54   | 55--+-56
//   25--|-26  |  27--|-28  |   57--|-58  |  59--|-60  |
//      29----30     31----32      61----62     63----64
//
//    1---------- 2---------- 3---------- 4       SC4^3/8^2
//    |  5          6          7        |  8          1  2  3  4  5  6  7  8
//    9    33    10    34    11    35    12    36       9 10 11 12 13 14 15 16
//    | 13    37----14----38----15----39--+-16----40   17 18 19 20 21 22 23 24
//   17    41  | 18    42    19    43    20    44  |   25 26 27 28 29 30 31 32
//    | 21    45    22    46    23    47  | 24    48   33 34 35 36 37 38 39 40
//   25----49--|-26----50----27----51----28    52  |   41 42 43 44 45 46 47 48
//      29    53    30    54    31    55    32    56   49 50 51 52 53 54 55 56
//         57  |       58          59          60  |   57 58 59 60 61 62 63 64
//           61----------62----------63----------64
//
#include <stdio.h>
//
//* Variables *
short int cnt, cnt2, cnt3;
short CC, SSM, LSM;
short nm[65], uflg[65], mm[65];
short tn[5][65];
short cs[33], csp[17], csm[65];
//
//* Sub-Routines *
void stp01(void), stp02(void), stp03(void), stp04(void), stp05(void);
void stp06(void), stp07(void), stp08(void), stp09(void), stp10(void);
void stp11(void), stp12(void), stp13(void), stp14(void), stp15(void);
void stp16(void), stp17(void), stp18(void), stp19(void), stp20(void);
void stp21(void), stp22(void), stp23(void), stp24(void), stp25(void);
void stp26(void), stp27(void), stp28(void), stp29(void), stp30(void);
void stp31(void), stp32(void), stp33(void), stp34(void), stp35(void);
void stp36(void), stp37(void), stp38(void), stp39(void), stp40(void);
void recrdans(void);
void trnsf(void);
void prms8(short x);
void ansprint(void), prpat8(void);
void chksums(void), chksms(void);
//
//* Main Program *
int main(){
 short n;
  printf("\n");
  printf("** Compose the Extra-Cubic Magic Objects of Order 2^6 and **\n");
```

```
  printf("**  Transform them into '4-Type Simultaneous' Magic Squares 8^2: **\n");
  printf("**  Multiple, 'Composite', Self-Complementary and Pan-Diagonal; **\n");
  printf("** List of the Standard Solutions with Check-sum Monitors **\n");
  //printf("** Compact List of the 5760 Standard Solutions **\n");
  for(n=0;n<65;n++){nm[n]=0; uflg[n]=0;}
  CC=65; SSM=130; cnt=0; cnt3=0;
  stp01();      //* Begin the Calculations *
  if(cnt3>0){prms8(cnt3);}
  printf("\n");
  printf(" [Count = %d]  OK!\n",cnt);
  printf("** Recompiled and Listed by Kanji Setsuda on\n");
  printf("    Mar. 4, 2015 with MacOSX 10.10.2 & Xcode 6.1.1 **\n");
  printf("\n");
 return 0;
}
//
//* Calculations: *
//* Set n1 & n64 & n1<n64 *
void stp01(){
 short a,b;
  for(a=1;a<33;a++){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){cnt2=0;
      nm[1]=a; nm[64]=b;
      uflg[a]=1; uflg[b]=1;
      stp02();
      uflg[b]=0; uflg[a]=0;}
  }
}
//* Set n2 & n63 *
void stp02(){
 short a,b;
  for(a=64;a>0;a--){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[2]=a; nm[63]=b;
      uflg[a]=1; uflg[b]=1;
      stp03();
      uflg[b]=0; uflg[a]=0;}
  }
}
//* Set n3 & n62 *
void stp03(){
 short a,b;
  for(a=64;a>0;a--){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[3]=a; nm[62]=b;
      uflg[a]=1; uflg[b]=1;
      stp04();
      uflg[b]=0; uflg[a]=0;}
  }
}
//* Set n4=SSM-n1-n2-n3 & n61 *
void stp04(){
 short a,b;
  a=SSM-nm[1]-nm[2]-nm[3];
  if((0<a)&&(a<65)){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){if(nm[1]==1){cnt2=0;}
      nm[4]=a; nm[61]=b;
      uflg[a]=1; uflg[b]=1;
      stp05();
      uflg[b]=0; uflg[a]=0;}}
}
//* Set n5 & n60 *
void stp05(){
 short a,b;
  for(a=64;a>0;a--){b=CC-a;
```

```
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[5]=a; nm[60]=b;
      uflg[a]=1; uflg[b]=1;
      stp06();
      uflg[b]=0; uflg[a]=0;}
  }
}
//* Set n6=SSM-n1-n2-n5 & n59 *
void stp06(){
 short a,b;
  a=SSM-nm[1]-nm[2]-nm[5];
  if((0<a)&&(a<65)){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[6]=a; nm[59]=b;
      uflg[a]=1; uflg[b]=1;
      stp07();
      uflg[b]=0; uflg[a]=0;}}
}
//* Set n7=SSM-n1-n3-n5 & n58 */
void stp07(){
 short a,b;
  a=SSM-nm[1]-nm[3]-nm[5];
  if((0<a)&&(a<65)){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[7]=a; nm[58]=b;
      uflg[a]=1; uflg[b]=1;
      stp08();
      uflg[b]=0; uflg[a]=0;}}
}
//* Set n8=SSM-n2-n4-n6 & n57 & n1<n8 & n1<n57 */
void stp08(){
 short a,b;
  a=SSM-nm[2]-nm[4]-nm[6];
  if((nm[1]<a)&&(a<65)){b=CC-a;
    if((b>nm[1])&&(uflg[a]==0)&&(uflg[b]==0)){
      nm[8]=a; nm[57]=b;
      uflg[a]=1; uflg[b]=1;
      stp09();
      uflg[b]=0; uflg[a]=0;}}
}
//* Level #2: *
//* Set n9 & n56 & n9<n2 *
void stp09(){
 short a,b;
  for(a=nm[2]-1;a>0;a--){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[9]=a; nm[56]=b;
      uflg[a]=1; uflg[b]=1;
      stp10();
      uflg[b]=0; uflg[a]=0;}
  }
}
//* Set n10=SSM-n1-n2-n9 & n55 *
void stp10(){
 short a,b;
  a=SSM-nm[1]-nm[2]-nm[9];
  if((0<a)&&(a<65)){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[10]=a; nm[55]=b;
      uflg[a]=1; uflg[b]=1;
      stp11();
      uflg[b]=0; uflg[a]=0;}}
}
//* Set n11=SSM-n1-n3-n9 & n54 *
void stp11(){
```

```
 short a,b;
   a=SSM-nm[1]-nm[3]-nm[9];
   if((0<a)&&(a<65)){b=CC-a;
     if((uflg[a]==0)&&(uflg[b]==0)){
       nm[11]=a; nm[54]=b;
       uflg[a]=1; uflg[b]=1;
       stp12();
       uflg[b]=0; uflg[a]=0;}}
}
//* Set n12=SSM-n2-n4-n10 & n53 *
void stp12(){
 short a,b;
   a=SSM-nm[2]-nm[4]-nm[10];
   if((0<a)&&(a<65)){b=CC-a;
     if((uflg[a]==0)&&(uflg[b]==0)){
       nm[12]=a; nm[53]=b;
       uflg[a]=1; uflg[b]=1;
       stp13();
       uflg[b]=0; uflg[a]=0;}}
}
//* Set n13=SSM-n1-n5-n9 & n52 *
void stp13(){
 short a,b;
   a=SSM-nm[1]-nm[5]-nm[9];
   if((0<a)&&(a<65)){b=CC-a;
     if((uflg[a]==0)&&(uflg[b]==0)){
       nm[13]=a; nm[52]=b;
       uflg[a]=1; uflg[b]=1;
       stp14();
       uflg[b]=0; uflg[a]=0;}}
}
//* Set n14=SSM-n2-n6-n10 & n51 *
void stp14(){
 short a,b;
   a=SSM-nm[2]-nm[6]-nm[10];
   if((0<a)&&(a<65)){b=CC-a;
     if((uflg[a]==0)&&(uflg[b]==0)){
       nm[14]=a; nm[51]=b;
       uflg[a]=1; uflg[b]=1;
       stp15();
       uflg[b]=0; uflg[a]=0;}}
}
//* Set n15=SSM-n3-n7-n11 & n50 *
void stp15(){
 short a,b;
   a=SSM-nm[3]-nm[7]-nm[11];
   if((0<a)&&(a<65)){b=CC-a;
     if((uflg[a]==0)&&(uflg[b]==0)){
       nm[15]=a; nm[50]=b;
       uflg[a]=1; uflg[b]=1;
       stp16();
       uflg[b]=0; uflg[a]=0;}}
}
//* Set n16=SSM-n4-n8-n12 & n49 *
void stp16(){
 short a,b;
   a=SSM-nm[4]-nm[8]-nm[12];
   if((0<a)&&(a<65)){b=CC-a;
     if((uflg[a]==0)&&(uflg[b]==0)){
       nm[16]=a; nm[49]=b;
       uflg[a]=1; uflg[b]=1;
       stp17();
       uflg[b]=0; uflg[a]=0;}}
}
//* Level #3: *
```

```
//* Set n17 & n48 *
void stp17(){
 short a,b;
  for(a=64;a>0;a--){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[17]=a; nm[48]=b;
      uflg[a]=1; uflg[b]=1;
      stp18();
      uflg[b]=0; uflg[a]=0;}
  }
}
//* Set n18=SSM-n1-n2-n17 & n47 *
void stp18(){
 short a,b;
  a=SSM-nm[1]-nm[2]-nm[17];
  if((0<a)&&(a<65)){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[18]=a; nm[47]=b;
      uflg[a]=1; uflg[b]=1;
      stp19();
      uflg[b]=0; uflg[a]=0;}}
}
//* Set n19=SSM-n1-n3-n17 & n46 *
void stp19(){
 short a,b;
  a=SSM-nm[1]-nm[3]-nm[17];
  if((0<a)&&(a<65)){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[19]=a; nm[46]=b;
      uflg[a]=1; uflg[b]=1;
      stp20();
      uflg[b]=0; uflg[a]=0;}}
}
//* Set n20=SSM-n2-n4-n18 & n45 *
void stp20(){
 short a,b;
  a=SSM-nm[2]-nm[4]-nm[18];
  if((0<a)&&(a<65)){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[20]=a; nm[45]=b;
      uflg[a]=1; uflg[b]=1;
      stp21();
      uflg[b]=0; uflg[a]=0;}}
}
//* Set n21=SSM-n1-n5-n17 & n44 *
void stp21(){
 short a,b;
  a=SSM-nm[1]-nm[5]-nm[17];
  if((0<a)&&(a<65)){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[21]=a; nm[44]=b;
      uflg[a]=1; uflg[b]=1;
      stp22();
      uflg[b]=0; uflg[a]=0;}}
}
//* Set n22=SSM-n2-n6-n18 & n43 *
void stp22(){
 short a,b;
  a=SSM-nm[2]-nm[6]-nm[18];
  if((0<a)&&(a<65)){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[22]=a; nm[43]=b;
      uflg[a]=1; uflg[b]=1;
      stp23();
      uflg[b]=0; uflg[a]=0;}}
```

```
}
//* Set n23=SSM-n3-n7-n19 & n42 *
void stp23(){
 short a,b;
   a=SSM-nm[3]-nm[7]-nm[19];
   if((0<a)&&(a<65)){b=CC-a;
     if((uflg[a]==0)&&(uflg[b]==0)){
       nm[23]=a; nm[42]=b;
       uflg[a]=1; uflg[b]=1;
       stp24();
       uflg[b]=0; uflg[a]=0;}}
}
//* Set n24=SSM-n4-n8-n20 & n41 *
void stp24(){
 short a,b;
   a=SSM-nm[4]-nm[8]-nm[20];
   if((0<a)&&(a<65)){b=CC-a;
     if((uflg[a]==0)&&(uflg[b]==0)){
       nm[24]=a; nm[41]=b;
       uflg[a]=1; uflg[b]=1;
       stp25();
       uflg[b]=0; uflg[a]=0;}}
}
//* Level #4 *
//* Set n25=SSM-n1-n9-n17 & n40 *
void stp25(){
 short a,b;
   a=SSM-nm[1]-nm[9]-nm[17];
   if((0<a)&&(a<65)){b=CC-a;
     if((uflg[a]==0)&&(uflg[b]==0)){
       nm[25]=a; nm[40]=b;
       uflg[a]=1; uflg[b]=1;
       stp26();
       uflg[b]=0; uflg[a]=0;}}
}
//* Set n26=SSM-n2-n10-n18 & n39 *
void stp26(){
 short a,b;
   a=SSM-nm[2]-nm[10]-nm[18];
   if((0<a)&&(a<65)){b=CC-a;
     if((uflg[a]==0)&&(uflg[b]==0)){
       nm[26]=a; nm[39]=b;
       uflg[a]=1; uflg[b]=1;
       stp27();
       uflg[b]=0; uflg[a]=0;}}
}
//* Set n27=SSM-n3-n11-n19 & n38 *
void stp27(){
 short a,b;
   a=SSM-nm[3]-nm[11]-nm[19];
   if((0<a)&&(a<65)){b=CC-a;
     if((uflg[a]==0)&&(uflg[b]==0)){
       nm[27]=a; nm[38]=b;
       uflg[a]=1; uflg[b]=1;
       stp28();
       uflg[b]=0; uflg[a]=0;}}
}
//* Set n28=SSM-n4-n12-n20 & n37 *
void stp28(){
 short a,b;
   a=SSM-nm[4]-nm[12]-nm[20];
   if((0<a)&&(a<65)){b=CC-a;
     if((uflg[a]==0)&&(uflg[b]==0)){
       nm[28]=a; nm[37]=b;
       uflg[a]=1; uflg[b]=1;
```

```
       stp29();
       uflg[b]=0; uflg[a]=0;}}
}
//* Set n29=SSM-n5-n13-n21 & n36 *
void stp29(){
 short a,b;
  a=SSM-nm[5]-nm[13]-nm[21];
  if((0<a)&&(a<65)){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[29]=a; nm[36]=b;
      uflg[a]=1; uflg[b]=1;
      stp30();
      uflg[b]=0; uflg[a]=0;}}
}
//* Set n30=SSM-n6-n14-n22 & n35 *
void stp30(){
 short a,b;
  a=SSM-nm[6]-nm[14]-nm[22];
  if((0<a)&&(a<65)){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[30]=a; nm[35]=b;
      uflg[a]=1; uflg[b]=1;
      stp31();
      uflg[b]=0; uflg[a]=0;}}
}
//* Set n31=SSM-n7-n15-n23 & n34 *
void stp31(){
 short a,b;
  a=SSM-nm[7]-nm[15]-nm[23];
  if((0<a)&&(a<65)){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[31]=a; nm[34]=b;
      uflg[a]=1; uflg[b]=1;
      stp32();
      uflg[b]=0; uflg[a]=0;}}
}
//* Set n32=SSM-n8-n16-n24 & n33 *
void stp32(){
 short a,b;
  a=SSM-nm[8]-nm[16]-nm[24];
  b=SSM-nm[1]-nm[2]-nm[34];
  if((0<a)&&(a<65)&&(a+b==CC)){
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[32]=a; nm[33]=b;
      uflg[a]=1; uflg[b]=1;
      stp33();
      uflg[b]=0; uflg[a]=0;}}
}
//
//* Check some Composite Sums *
void stp33(){
 short sm1,sm2,sm3;
  sm1=nm[1]+nm[3]+nm[33]+nm[35];
  sm2=nm[1]+nm[5]+nm[33]+nm[37];
  sm3=nm[1]+nm[9]+nm[33]+nm[41];
  if((sm1==SSM)&&(sm2==SSM)&&(sm3==SSM)){stp34();}
}
//
//* Check some Composite Sums *
void stp34(){
 short sm1,sm2,sm3;
  sm1=nm[1]+nm[17]+nm[33]+nm[49];
  sm2=nm[2]+nm[4]+nm[34]+nm[36];
  sm3=nm[2]+nm[6]+nm[34]+nm[38];
  if((sm1==SSM)&&(sm2==SSM)&&(sm3==SSM)){stp35();}
```

```
}
//
//* Check some Composite Sums *
void stp35(){
 short sm1,sm2,sm3;
   sm1=nm[2]+nm[10]+nm[34]+nm[42];
   sm2=nm[2]+nm[18]+nm[34]+nm[50];
   sm3=nm[3]+nm[7]+nm[35]+nm[39];
   if((sm1==SSM)&&(sm2==SSM)&&(sm3==SSM)){stp36();}
}
//
//* Check some Composite Sums *
void stp36(){
 short sm1,sm2,sm3;
   sm1=nm[3]+nm[11]+nm[35]+nm[43];
   sm2=nm[3]+nm[19]+nm[35]+nm[51];
   sm3=nm[4]+nm[8]+nm[36]+nm[40];
   if((sm1==SSM)&&(sm2==SSM)&&(sm3==SSM)){stp37();}
}
//
//* Check some Composite Sums *
void stp37(){
 short sm1,sm2,sm3;
   sm1=nm[4]+nm[12]+nm[36]+nm[44];
   sm2=nm[4]+nm[20]+nm[36]+nm[52];
   sm3=nm[5]+nm[13]+nm[37]+nm[45];
   if((sm1==SSM)&&(sm2==SSM)&&(sm3==SSM)){stp38();}
}
//
//* Check some Composite Sums *
void stp38(){
 short sm1,sm2,sm3;
   sm1=nm[5]+nm[21]+nm[37]+nm[53];
   sm2=nm[6]+nm[14]+nm[38]+nm[46];
   sm3=nm[6]+nm[22]+nm[38]+nm[54];
   if((sm1==SSM)&&(sm2==SSM)&&(sm3==SSM)){stp39();}
}
//
//* Check some Composite Sums *
void stp39(){
 short sm1,sm2,sm3;
   sm1=nm[7]+nm[15]+nm[39]+nm[47];
   sm2=nm[7]+nm[23]+nm[39]+nm[55];
   sm3=nm[8]+nm[16]+nm[40]+nm[48];
   if((sm1==SSM)&&(sm2==SSM)&&(sm3==SSM)){stp40();}
}
//
//* Last Check of Composite Sums *
void stp40(){
 short sm1,sm2,sm3;
   sm1=nm[8]+nm[24]+nm[40]+nm[56];
   sm2=nm[17]+nm[18]+nm[49]+nm[50];
   sm3=nm[31]+nm[32]+nm[63]+nm[64];
   if((sm1==SSM)&&(sm2==SSM)&&(sm3==SSM)){
     recrdans();}
}
//
//** Concept of Transformation from SC-type to 4T Simul-type **
//   0/SC                    0/Sml
//    1  2  3  4  5  6  7  8    1  2  4  3  6  5  7  8
//    9 10 11 12 13 14 15 16    9 10 12 11 14 13 15 16
//   17 18 19 20 21 22 23 24   25 26 28 27 30 29 31 32
//   25 26 27 28 29 30 31 32   17 18 20 19 22 21 23 24
//   33 34 35 36 37 38 39 40   41 42 44 43 46 45 47 48
//   41 42 43 44 45 46 47 48   33 34 36 35 38 37 39 40
```

```
//   49 50 51 52 53 54 55 56    49 50 52 51 54 53 55 56
//   57 58 59 60 61 62 63 64    57 58 60 59 62 61 63 64
//
//* Transform these Answers into Another Type *
void trnsf(){
  mm[1]=nm[1];   mm[2]=nm[2];   mm[3]=nm[4];   mm[4]=nm[3];
  mm[5]=nm[6];   mm[6]=nm[5];   mm[7]=nm[7];   mm[8]=nm[8];
  mm[9]=nm[9];   mm[10]=nm[10]; mm[11]=nm[12]; mm[12]=nm[11];
  mm[13]=nm[14]; mm[14]=nm[13]; mm[15]=nm[15]; mm[16]=nm[16];
  mm[17]=nm[25]; mm[18]=nm[26]; mm[19]=nm[28]; mm[20]=nm[27];
  mm[21]=nm[30]; mm[22]=nm[29]; mm[23]=nm[31]; mm[24]=nm[32];
  mm[25]=nm[17]; mm[26]=nm[18]; mm[27]=nm[20]; mm[28]=nm[19];
  mm[29]=nm[22]; mm[30]=nm[21]; mm[31]=nm[23]; mm[32]=nm[24];
  mm[33]=nm[41]; mm[34]=nm[42]; mm[35]=nm[44]; mm[36]=nm[43];
  mm[37]=nm[46]; mm[38]=nm[45]; mm[39]=nm[47]; mm[40]=nm[48];
  mm[41]=nm[33]; mm[42]=nm[34]; mm[43]=nm[36]; mm[44]=nm[35];
  mm[45]=nm[38]; mm[46]=nm[37]; mm[47]=nm[39]; mm[48]=nm[40];
  mm[49]=nm[49]; mm[50]=nm[50]; mm[51]=nm[52]; mm[52]=nm[51];
  mm[53]=nm[54]; mm[54]=nm[53]; mm[55]=nm[55]; mm[56]=nm[56];
  mm[57]=nm[57]; mm[58]=nm[58]; mm[59]=nm[60]; mm[60]=nm[59];
  mm[61]=nm[62]; mm[62]=nm[61]; mm[63]=nm[63]; mm[64]=nm[64];
}
//* Record them *
void recrdans(void){
 short n;
  cnt++; cnt2++;
  if(cnt2==1){
    trnsf();
    tn[cnt3][0]=cnt;
    for(n=1;n<65;n++){tn[cnt3][n]=mm[n];}
    cnt3++;
    //if(cnt3==4){prms8(cnt3); cnt3=0;}
    if(cnt3==1){prpat8(); cnt3=0;}
  }
}
//
//* Print 4 Pieces of 'C&C' MS88 *
void prms8(short x){
 short l,m,m8,n;
  for(n=0;n<x;n++){
    printf("%23d#",tn[n][0]);
    if(n+1<x){printf("  ");}
  }
  printf("\n");
  for(m=0;m<8;m++){m8=m*8;
    for(l=0;l<x;l++){
      for(n=1;n<9;n++){printf("%3d",tn[l][m8+n]);}
      if(l+1<x){printf("  ");}
    }
    printf("\n");
  }
}
//
//* Print 'C&C' MS88 with Check-sums *
void prpat8(){
 short n,m,m2,m8;
  chksums(); chksms();
  printf("%24d#|Rw1+Rw2|Cl1+Cl2\\Pd1/Pd2|Composite Fours\n",cnt);
  for(m=0;m<8;m++){m2=m*2; m8=m*8;
    printf(" ");
    for(n=1;n<9;n++){printf("%3d",mm[m8+n]);}
    printf("|%3d+%3d|%3d+%3d\\%3d/%3d|",
           cs[m2+1],cs[m2+2],cs[m2+17],cs[m2+18],csp[m+1],csp[m+9]);
    for(n=1;n<9;n++){
      printf("%3d",csm[m8+n]);
```

```
      if(n<8){printf(" ");}
    }
    printf("\n");
  }
}
//
//* Check the Line-Sums *
void chksums(){
  cs[1]=mm[1]+mm[2]+mm[3]+mm[4];         // rw11;
  cs[2]=mm[5]+mm[6]+mm[7]+mm[8];         // rw12;
  cs[3]=mm[9]+mm[10]+mm[11]+mm[12];      // rw21;
  cs[4]=mm[13]+mm[14]+mm[15]+mm[16];     // rw22;
  cs[5]=mm[17]+mm[18]+mm[19]+mm[20];     // rw31;
  cs[6]=mm[21]+mm[22]+mm[23]+mm[24];     // rw32;
  cs[7]=mm[25]+mm[26]+mm[27]+mm[28];     // rw41;
  cs[8]=mm[29]+mm[30]+mm[31]+mm[32];     // rw42;
  cs[9]=mm[33]+mm[34]+mm[35]+mm[36];     // rw51;
  cs[10]=mm[37]+mm[38]+mm[39]+mm[40];    // rw52;
  cs[11]=mm[41]+mm[42]+mm[43]+mm[44];    // rw61;
  cs[12]=mm[45]+mm[46]+mm[47]+mm[48];    // rw62;
  cs[13]=mm[49]+mm[50]+mm[51]+mm[52];    // rw71;
  cs[14]=mm[53]+mm[54]+mm[55]+mm[56];    // rw72;
  cs[15]=mm[57]+mm[58]+mm[59]+mm[60];    // rw81;
  cs[16]=mm[61]+mm[62]+mm[63]+mm[64];    // rw82;
  //
  cs[17]=mm[1]+mm[9]+mm[17]+mm[25];      // cl11;
  cs[18]=mm[33]+mm[41]+mm[49]+mm[57];    // cl12;
  cs[19]=mm[2]+mm[10]+mm[18]+mm[26];     // cl21;
  cs[20]=mm[34]+mm[42]+mm[50]+mm[58];    // cl22;
  cs[21]=mm[3]+mm[11]+mm[19]+mm[27];     // cl31;
  cs[22]=mm[35]+mm[43]+mm[51]+mm[59];    // cl32;
  cs[23]=mm[4]+mm[12]+mm[20]+mm[28];     // cl41;
  cs[24]=mm[36]+mm[44]+mm[52]+mm[60];    // cl42;
  cs[25]=mm[5]+mm[13]+mm[21]+mm[29];     // cl51;
  cs[26]=mm[37]+mm[45]+mm[53]+mm[61];    // cl52;
  cs[27]=mm[6]+mm[14]+mm[22]+mm[30];     // cl61;
  cs[28]=mm[38]+mm[46]+mm[54]+mm[62];    // cl62;
  cs[29]=mm[7]+mm[15]+mm[23]+mm[31];     // cl71;
  cs[30]=mm[39]+mm[47]+mm[55]+mm[63];    // cl72;
  cs[31]=mm[8]+mm[16]+mm[24]+mm[32];     // cl81;
  cs[32]=mm[40]+mm[48]+mm[56]+mm[64];    // cl82;
  //
  csp[1]=mm[1]+mm[10]+mm[19]+mm[28]+mm[37]+mm[46]+mm[55]+mm[64];   // pd1;
  csp[2]=mm[2]+mm[11]+mm[20]+mm[29]+mm[38]+mm[47]+mm[56]+mm[57];   // pd2;
  csp[3]=mm[3]+mm[12]+mm[21]+mm[30]+mm[39]+mm[48]+mm[49]+mm[58];   // pd3;
  csp[4]=mm[4]+mm[13]+mm[22]+mm[31]+mm[40]+mm[41]+mm[50]+mm[59];   // pd4;
  csp[5]=mm[5]+mm[14]+mm[23]+mm[32]+mm[33]+mm[42]+mm[51]+mm[60];   // pd5;
  csp[6]=mm[6]+mm[15]+mm[24]+mm[25]+mm[34]+mm[43]+mm[52]+mm[61];   // pd6;
  csp[7]=mm[7]+mm[16]+mm[17]+mm[26]+mm[35]+mm[44]+mm[53]+mm[62];   // pd7;
  csp[8]=mm[8]+mm[9]+mm[18]+mm[27]+mm[36]+mm[45]+mm[54]+mm[63];    // pd8;
  //
  csp[9]=mm[1]+mm[16]+mm[23]+mm[30]+mm[37]+mm[44]+mm[51]+mm[58];   // pb1;
  csp[10]=mm[2]+mm[9]+mm[24]+mm[31]+mm[38]+mm[45]+mm[52]+mm[59];   // pb2;
  csp[11]=mm[3]+mm[10]+mm[17]+mm[32]+mm[39]+mm[46]+mm[53]+mm[60];  // pb3;
  csp[12]=mm[4]+mm[11]+mm[18]+mm[25]+mm[40]+mm[47]+mm[54]+mm[61];  // pb4;
  csp[13]=mm[5]+mm[12]+mm[19]+mm[26]+mm[33]+mm[48]+mm[55]+mm[62];  // pb5;
  csp[14]=mm[6]+mm[13]+mm[20]+mm[27]+mm[34]+mm[41]+mm[56]+mm[63];  // pb6;
  csp[15]=mm[7]+mm[14]+mm[21]+mm[28]+mm[35]+mm[42]+mm[49]+mm[64];  // pb7;
  csp[16]=mm[8]+mm[15]+mm[22]+mm[29]+mm[36]+mm[43]+mm[50]+mm[57];  // pb8;
}
//
//* Check Sums of Composite Fours *
void chksms(){
  csm[1]=mm[1]+mm[2]+mm[9]+mm[10];
  csm[2]=mm[2]+mm[3]+mm[10]+mm[11];
```

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

２． 本来ならば，これに三連立型正方連結対称汎八方陣のいつもの構成プログラムと，新旧比較対照チェック用のモニタ・プログラムを追加するのだが，リストが長くなり過ぎる。これを割愛して，モニターは実行結果だけを示す。

```
** 六次元超立体二方陣から三連立型正方連結対称汎八方陣を取り出した結果 **
   ** 標準解のリスト：チェック・サム・モニター付き **
                          1#|Rw1+Rw2|Cl1+Cl2\Pd1/Pd2|Composite Fours
 1 63  4 62  6 60  7 57|130+130|130+130\260/260|130 130 130 130 130 130 130 130
56 10 53 11 51 13 50 16|130+130|130+130\260/260|130 130 130 130 130 130 130 130
25 39 28 38 30 36 31 33|130+130|130+130\260/260|130 130 130 130 130 130 130 130
48 18 45 19 43 21 42 24|130+130|130+130\260/260|130 130 130 130 130 130 130 130
41 23 44 22 46 20 47 17|130+130|130+130\260/260|130 130 130 130 130 130 130 130
32 34 29 35 27 37 26 40|130+130|130+130\260/260|130 130 130 130 130 130 130 130
49 15 52 14 54 12 55  9|130+130|130+130\260/260|130 130 130 130 130 130 130 130
 8 58  5 59  3 61  2 64|130+130|130+130\260/260|130 130 130 130 130 130 130 130
                         25#|Rw1+Rw2|Cl1+Cl2\Pd1/Pd2|Composite Fours
 1 63  6 60  4 62  7 57|130+130|130+130\260/260|130 130 130 130 130 130 130 130
56 10 51 13 53 11 50 16|130+130|130+130\260/260|130 130 130 130 130 130 130 130
25 39 30 36 28 38 31 33|130+130|130+130\260/260|130 130 130 130 130 130 130 130
48 18 43 21 45 19 42 24|130+130|130+130\260/260|130 130 130 130 130 130 130 130
41 23 46 20 44 22 47 17|130+130|130+130\260/260|130 130 130 130 130 130 130 130
32 34 27 37 29 35 26 40|130+130|130+130\260/260|130 130 130 130 130 130 130 130
49 15 54 12 52 14 55  9|130+130|130+130\260/260|130 130 130 130 130 130 130 130
 8 58  3 61  5 59  2 64|130+130|130+130\260/260|130 130 130 130 130 130 130 130
                         49#|Rw1+Rw2|Cl1+Cl2\Pd1/Pd2|Composite Fours
 1 63 10 56  4 62 11 53|130+130|130+130\260/260|130 130 130 130 130 130 130 130
60  6 51 13 57  7 50 16|130+130|130+130\260/260|130 130 130 130 130 130 130 130
21 43 30 36 24 42 31 33|130+130|130+130\260/260|130 130 130 130 130 130 130 130
48 18 39 25 45 19 38 28|130+130|130+130\260/260|130 130 130 130 130 130 130 130
37 27 46 20 40 26 47 17|130+130|130+130\260/260|130 130 130 130 130 130 130 130
32 34 23 41 29 35 22 44|130+130|130+130\260/260|130 130 130 130 130 130 130 130
49 15 58  8 52 14 59  5|130+130|130+130\260/260|130 130 130 130 130 130 130 130
12 54  3 61  9 55  2 64|130+130|130+130\260/260|130 130 130 130 130 130 130 130
                         73#|Rw1+Rw2|Cl1+Cl2\Pd1/Pd2|Composite Fours
 1 63 18 48  4 62 19 45|130+130|130+130\260/260|130 130 130 130 130 130 130 130
60  6 43 21 57  7 42 24|130+130|130+130\260/260|130 130 130 130 130 130 130 130
13 51 30 36 16 50 31 33|130+130|130+130\260/260|130 130 130 130 130 130 130 130
56 10 39 25 53 11 38 28|130+130|130+130\260/260|130 130 130 130 130 130 130 130
37 27 54 12 40 26 55  9|130+130|130+130\260/260|130 130 130 130 130 130 130 130
32 34 15 49 29 35 14 52|130+130|130+130\260/260|130 130 130 130 130 130 130 130
41 23 58  8 44 22 59  5|130+130|130+130\260/260|130 130 130 130 130 130 130 130
20 46  3 61 17 47  2 64|130+130|130+130\260/260|130 130 130 130 130 130 130 130
                         97#|Rw1+Rw2|Cl1+Cl2\Pd1/Pd2|Composite Fours
 1 63 34 32  4 62 35 29|130+130|130+130\260/260|130 130 130 130 130 130 130 130
60  6 27 37 57  7 26 40|130+130|130+130\260/260|130 130 130 130 130 130 130 130
13 51 46 20 16 50 47 17|130+130|130+130\260/260|130 130 130 130 130 130 130 130
56 10 23 41 53 11 22 44|130+130|130+130\260/260|130 130 130 130 130 130 130 130
21 43 54 12 24 42 55  9|130+130|130+130\260/260|130 130 130 130 130 130 130 130
48 18 15 49 45 19 14 52|130+130|130+130\260/260|130 130 130 130 130 130 130 130
25 39 58  8 28 38 59  5|130+130|130+130\260/260|130 130 130 130 130 130 130 130
36 30  3 61 33 31  2 64|130+130|130+130\260/260|130 130 130 130 130 130 130 130
                        121#|Rw1+Rw2|Cl1+Cl2\Pd1/Pd2|Composite Fours
 1 62  4 63  7 60  6 57|130+130|130+130\260/260|130 130 130 130 130 130 130 130
56 11 53 10 50 13 51 16|130+130|130+130\260/260|130 130 130 130 130 130 130 130
25 38 28 39 31 36 30 33|130+130|130+130\260/260|130 130 130 130 130 130 130 130
48 19 45 18 42 21 43 24|130+130|130+130\260/260|130 130 130 130 130 130 130 130
41 22 44 23 47 20 46 17|130+130|130+130\260/260|130 130 130 130 130 130 130 130
```

```
32 35 29 34 26 37 27 40|130+130|130+130\260/260|130 130 130 130 130 130 130 130
49 14 52 15 55 12 54  9|130+130|130+130\260/260|130 130 130 130 130 130 130 130
 8 59  5 58  2 61  3 64|130+130|130+130\260/260|130 130 130 130 130 130 130 130
                    145#|Rw1+Rw2|Cl1+Cl2\Pd1/Pd2|Composite Fours
 1 62  7 60  4 63  6 57|130+130|130+130\260/260|130 130 130 130 130 130 130 130
56 11 50 13 53 10 51 16|130+130|130+130\260/260|130 130 130 130 130 130 130 130
25 38 31 36 28 39 30 33|130+130|130+130\260/260|130 130 130 130 130 130 130 130
48 19 42 21 45 18 43 24|130+130|130+130\260/260|130 130 130 130 130 130 130 130
41 22 47 20 44 23 46 17|130+130|130+130\260/260|130 130 130 130 130 130 130 130
32 35 26 37 29 34 27 40|130+130|130+130\260/260|130 130 130 130 130 130 130 130
49 14 55 12 52 15 54  9|130+130|130+130\260/260|130 130 130 130 130 130 130 130
 8 59  2 61  5 58  3 64|130+130|130+130\260/260|130 130 130 130 130 130 130 130
                    163#|Rw1+Rw2|Cl1+Cl2\Pd1/Pd2|Composite Fours
 1 62 11 56  4 63 10 53|130+130|130+130\260/260|130 130 130 130 130 130 130 130
60  7 50 13 57  6 51 16|130+130|130+130\260/260|130 130 130 130 130 130 130 130
21 42 31 36 24 43 30 33|130+130|130+130\260/260|130 130 130 130 130 130 130 130
48 19 38 25 45 18 39 28|130+130|130+130\260/260|130 130 130 130 130 130 130 130
37 26 47 20 40 27 46 17|130+130|130+130\260/260|130 130 130 130 130 130 130 130
32 35 22 41 29 34 23 44|130+130|130+130\260/260|130 130 130 130 130 130 130 130
49 14 59  8 52 15 58  5|130+130|130+130\260/260|130 130 130 130 130 130 130 130
12 55  2 61  9 54  3 64|130+130|130+130\260/260|130 130 130 130 130 130 130 130
                    181#|Rw1+Rw2|Cl1+Cl2\Pd1/Pd2|Composite Fours
 1 62 19 48  4 63 18 45|130+130|130+130\260/260|130 130 130 130 130 130 130 130
60  7 42 21 57  6 43 24|130+130|130+130\260/260|130 130 130 130 130 130 130 130
13 50 31 36 16 51 30 33|130+130|130+130\260/260|130 130 130 130 130 130 130 130
56 11 38 25 53 10 39 28|130+130|130+130\260/260|130 130 130 130 130 130 130 130
37 26 55 12 40 27 54  9|130+130|130+130\260/260|130 130 130 130 130 130 130 130
32 35 14 49 29 34 15 52|130+130|130+130\260/260|130 130 130 130 130 130 130 130
41 22 59  8 44 23 58  5|130+130|130+130\260/260|130 130 130 130 130 130 130 130
20 47  2 61 17 46  3 64|130+130|130+130\260/260|130 130 130 130 130 130 130 130
                    199#|Rw1+Rw2|Cl1+Cl2\Pd1/Pd2|Composite Fours
 1 62 35 32  4 63 34 29|130+130|130+130\260/260|130 130 130 130 130 130 130 130
60  7 26 37 57  6 27 40|130+130|130+130\260/260|130 130 130 130 130 130 130 130
13 50 47 20 16 51 46 17|130+130|130+130\260/260|130 130 130 130 130 130 130 130
56 11 22 41 53 10 23 44|130+130|130+130\260/260|130 130 130 130 130 130 130 130
21 42 55 12 24 43 54  9|130+130|130+130\260/260|130 130 130 130 130 130 130 130
48 19 14 49 45 18 15 52|130+130|130+130\260/260|130 130 130 130 130 130 130 130
25 38 59  8 28 39 58  5|130+130|130+130\260/260|130 130 130 130 130 130 130 130
36 31  2 61 33 30  3 64|130+130|130+130\260/260|130 130 130 130 130 130 130 130
                    217#|Rw1+Rw2|Cl1+Cl2\Pd1/Pd2|Composite Fours
 1 60  6 63  7 62  4 57|130+130|130+130\260/260|130 130 130 130 130 130 130 130
56 13 51 10 50 11 53 16|130+130|130+130\260/260|130 130 130 130 130 130 130 130
25 36 30 39 31 38 28 33|130+130|130+130\260/260|130 130 130 130 130 130 130 130
48 21 43 18 42 19 45 24|130+130|130+130\260/260|130 130 130 130 130 130 130 130
41 20 46 23 47 22 44 17|130+130|130+130\260/260|130 130 130 130 130 130 130 130
32 37 27 34 26 35 29 40|130+130|130+130\260/260|130 130 130 130 130 130 130 130
49 12 54 15 55 14 52  9|130+130|130+130\260/260|130 130 130 130 130 130 130 130
 8 61  3 58  2 59  5 64|130+130|130+130\260/260|130 130 130 130 130 130 130 130
                    235#|Rw1+Rw2|Cl1+Cl2\Pd1/Pd2|Composite Fours
 1 60  7 62  6 63  4 57|130+130|130+130\260/260|130 130 130 130 130 130 130 130
56 13 50 11 51 10 53 16|130+130|130+130\260/260|130 130 130 130 130 130 130 130
25 36 31 38 30 39 28 33|130+130|130+130\260/260|130 130 130 130 130 130 130 130
48 21 42 19 43 18 45 24|130+130|130+130\260/260|130 130 130 130 130 130 130 130
41 20 47 22 46 23 44 17|130+130|130+130\260/260|130 130 130 130 130 130 130 130
32 37 26 35 27 34 29 40|130+130|130+130\260/260|130 130 130 130 130 130 130 130
49 12 55 14 54 15 52  9|130+130|130+130\260/260|130 130 130 130 130 130 130 130
 8 61  2 59  3 58  5 64|130+130|130+130\260/260|130 130 130 130 130 130 130 130
 . . . . .
```

253#

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

265#

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

277#

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

289#

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

301#

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

313#

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

325#

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

331#

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

337#

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

343#

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

349#

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

355#

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

361#

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

721#

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

1081#

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

1441#

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

1765#

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

2089#

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

2413#

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

2737#

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

2989#

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

3241#

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

3493#

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

3745#

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

3961#

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

4177#

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

4393#

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

4609#

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

```
                    4753#                    4897#                    5041#                    5185#
18 63  1 48 34 15 49 32  19 62  1 48 35 14 49 32  20 61  2 47 36 13 50 31  21 60  1 48 37 12 49 32
45  4 62 19 29 52 14 35  45  4 63 18 29 52 15 34  46  3 64 17 30 51 16 33  43  6 63 18 27 54 15 34
24 57  7 42 40  9 55 26  24 57  6 43 40  9 54 27  23 58  5 44 39 10 53 28  24 57  4 45 40  9 52 29
43  6 60 21 27 54 12 37  42  7 60 21 26 55 12 37  41  8 59 22 25 56 11 38  42  7 62 19 26 55 14 35
28 53 11 38 44  5 59 22  28 53 10 39 44  5 58 23  27 54  9 40 43  6 57 24  30 51 10 39 46  3 58 23
39 10 56 25 23 58  8 41  38 11 56 25 22 59  8 41  37 12 55 26 21 60  7 42  36 13 56 25 20 61  8 41
30 51 13 36 46  3 61 20  31 50 13 36 47  2 61 20  32 49 14 35 48  1 62 19  31 50 11 38 47  2 59 22
33 16 50 31 17 64  2 47  33 16 51 30 17 64  3 46  34 15 52 29 18 63  4 45  33 16 53 28 17 64  5 44
                    5293#                    5401#                    5509#                    5617#
22 59  2 47 38 11 50 31  23 58  3 46 39 10 51 30  24 57  4 45 40  9 52 29  25 56  1 48 41  8 49 32
44  5 64 17 28 53 16 33  44  5 64 17 28 53 16 33  43  6 63 18 27 54 15 34  39 10 63 18 23 58 15 34
23 58  3 46 39 10 51 30  22 59  2 47 38 11 50 31  21 60  1 48 37 12 49 32  28 53  4 45 44  5 52 29
41  8 61 20 25 56 13 36  41  8 61 20 25 56 13 36  42  7 62 19 26 55 14 35  38 11 62 19 22 59 14 35
29 52  9 40 45  4 57 24  29 52  9 40 45  4 57 24  30 51 10 39 46  3 58 23  30 51  6 43 46  3 54 27
35 14 55 26 19 62  7 42  34 15 54 27 18 63  6 43  33 16 53 28 17 64  5 44  36 13 60 21 20 61 12 37
32 49 12 37 48  1 60 21  32 49 12 37 48  1 60 21  31 50 11 38 47  2 59 22  31 50  7 42 47  2 55 26
34 15 54 27 18 63  6 43  35 14 55 26 19 62  7 42  36 13 56 25 20 61  8 41  33 16 57 24 17 64  9 40
                    5653#                    5689#                    5725#
26 55  2 47 42  7 50 31  27 54  3 46 43  6 51 30  28 53  4 45 44  5 52 29
40  9 64 17 24 57 16 33  40  9 64 17 24 57 16 33  39 10 63 18 23 58 15 34
27 54  3 46 43  6 51 30  26 55  2 47 42  7 50 31  25 56  1 48 41  8 49 32
37 12 61 20 21 60 13 36  37 12 61 20 21 60 13 36  38 11 62 19 22 59 14 35
29 52  5 44 45  4 53 28  29 52  5 44 45  4 53 28  30 51  6 43 46  3 54 27
35 14 59 22 19 62 11 38  34 15 58 23 18 63 10 39  33 16 57 24 17 64  9 40
32 49  8 41 48  1 56 25  32 49  8 41 48  1 56 25  31 50  7 42 47  2 55 26
34 15 58 23 18 63 10 39  35 14 59 22 19 62 11 38  36 13 60 21 20 61 12 37

 [Count = 5760]  OK!
** Recompiled and Listed by Kanji Setsuda on
    Mar. 4, 2015 with MacOSX 10.10.2 & Xcode 6.1.1 **

*  新旧のリストで解の対応関係をモニターした結果  *
  ??: 0
   1: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
 101: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
 201: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
 301: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
 401: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
 501: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
 601: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
 701: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
 801: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
 901: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1001: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1101: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1201: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1301: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1401: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1501: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1601: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1701: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1801: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1901: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
2001: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
2101: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
2201: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
2301: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
2401: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
2501: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
2601: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
2701: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
2801: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
2901: 11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
```

```
3001: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
3101: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
3201: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
3301: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
3401: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
3501: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
3601: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
3701: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
3801: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
3901: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
4001: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
4101: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
4201: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
4301: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
4401: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
4501: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
4601: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
4701: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
4801: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
4901: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5001: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5101: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5201: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5301: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5401: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5501: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5601: 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
5701: 1111111111111111111111111111111111111111111111111111111111111111
[Count = 5760:5760] OK!
```

３． 実行結果は，以上の通り，全部が正解だ。解のリストには一部省略があるが，「お見合いモニター」の方は省略せずに全部を掲げておいた。見事な**一対一対応**が見て取れる。

三連立型正方連結対称汎八方陣の解集合が，六次元超立体二方陣からめでたく取り出せた。ただ１通りの変換法で。しかも，高速に(従来の十分の一程度の時間で)。次元の壁を越え，タイプの区別を超えて，２つの現象を結び付けている**内的連関**が明らかになった。

この結果によって，六次元超立体二方陣が三連立型正方連結対称汎八方陣の**根源方陣**であると，主張してよいように思う。

「根源方陣」は，いろいろな意味で「原初方陣」に似ている。たった一つの変換法というのが「ひな形変換法」そっくりである。ただし，根源方陣は，既に解そのものであって，原初方陣のように「基本配列」の模倣方陣ではない。

事実今回の場合は，途中で**四方陣多重型**の**補数対称八方陣**から，変換法によって**三連立型**正方連結対称汎八方陣（四方陣多重型も勘定に入れれば，**四連立型**）を得たのである。

これは，途中で四方陣多重型の**正方連結完全八方陣**を経由しても得られる。その場合には，別の変換法が必要になるが，小生は既にそれを得ている。

かくして，わが六次元超立体二方陣は，様々なタイプの魔方陣を１つに結び付ける「根源方陣」なのである。

では，六次元超立体二方陣は，何種類の魔方陣の「根源方陣」たり得るのであろうか。この親には何人の子が居るのであろうか。また何故にこのような不思議な現象が可能なのか。

一つの不思議がまた新たな不思議を呼び起こす。いくら研究しても，課題は減るどころか，逆に増えるばかりである。

(著者：摂田 寛二(Kanji Setsuda); jag12001@nifty.com;
執筆・計数・作表 in 2005 and 2006 with MacOSX & Xcode 2.1;
再計算清書版 on March 4, 2015 with MacOSX 10.10.2 & Xcode 6.1.1;)