

Chapter 4 : New Algebraic Study of MS66 : Kanji Setsuda

Section 6 : 'Semi-Composite' Type of Magic Squares 6x6

#0. What I would like to report about

As you have known, the world of Magic Squares of Order 6 looks quite different from those of orders 4, 8, 12 and 16.

You can surely build the Standard Type of MS66, but you can make no solution of Self-complementary Type of Order 6. You can build neither Pan-diagonal Type nor 'Composite' MS66 at all.

If you try to make them, you will find 'No Answers' for them. You cannot make all the sums of rows, columns and two primary diagonals equal to the Constant Value at the same time along with all the characteristic conditions of those types.

This is the fatal limitation to the world of magic squares of such the 'Singly Even' orders as 6, 10, 14, 18 and 22.

But we, earnest Magic Square makers, feel so lonely here that we have long wanted to have those beautiful objects and tried to build them by any means.

We even tried to change our definitions only a little in order to have the 'similar' results to them. We would never mind even when we should have only the 'Semi' types of our true objects. We are almost satisfied with them after all.

Don't you think we are irrational enough to be called 'ridiculous', do you?

Now I am going to make those 'Semi-Composite' type of MS66 here with the reformed definitions, by reducing some Composite Conditions but only a few.

I would like to report about the 3 cases of these new objects as follows.

#1. Experiment No.1: with 30 Composite Conditions and 6 ones Un-defined

The first type of our 'Semi-Composite' MS66 is designed under the following definitions as listed below.

**** Definition-1: Composite Conditions: S=74 ****

$\begin{array}{ c c c c c c } \hline n6 & n1 & n2 & n3 & n4 & n5 & n6 & n1 \\ \hline 12 & n7 & n8 & n9 & 10 & 11 & 12 & n7 \\ \hline 18 & 13 & 14 & 15 & 16 & 17 & 18 & 13 \\ \hline 24 & 19 & 20 & 21 & 22 & 23 & 24 & 19 \\ \hline 30 & 25 & 26 & 27 & 28 & 29 & 30 & 25 \\ \hline 36 & 31 & 32 & 33 & 34 & 35 & 36 & 31 \\ \hline n6 & n1 & n2 & n3 & n4 & n5 & n6 & n1 \\ \hline \end{array}$	$\begin{array}{l} n1+ n2+ n7 +n8 = S \dots (1) \\ n2+ n3+ n8+ n9 = S \dots (2) \\ n3+ n4+ n9+n10 = S \dots (3) \\ n4+ n5+n10+n11 = S \dots (4) \\ n5+ n6+n11+n12 = S \dots (5) \\ n6+ n1+n12+ n7 = S \dots (6) \\ *n7+ n8+n13+n14 = S \dots (7)* \\ n8+ n9+n14+n15 = S \dots (8) \\ n9+n10+n15+n16 = S \dots (9) \\ n10+n11+n16+n17 = S \dots (10) \\ *n11+n12+n17+n18 = S \dots (11)* \\ n12+ n7+n18+n13 = S \dots (12) \\ n13+n14+n19+n20 = S \dots (13) \\ n14+n15+n20+n21 = S \dots (14) \\ \hline n15+n16+n21+n22 = S \dots (15) \quad \quad n26+n27+n32+n33 = S \dots (26) \\ n16+n17+n22+n23 = S \dots (16) \quad \quad n27+n28+n33+n34 = S \dots (27) \\ n17+n18+n23+n24 = S \dots (17) \quad \quad n28+n29+n34+n35 = S \dots (28) \\ n18+n13+n24+n19 = S \dots (18) \quad \quad n29+n30+n35+n36 = S \dots (29) \\ *n19+n20+n25+n26 = S \dots (19)* \quad \quad n30+n25+n36+n31 = S \dots (30) \\ n20+n21+n26+n27 = S \dots (20) \quad \quad *n31+n32+ n1+ n2 = S \dots (31)* \\ n21+n22+n27+n28 = S \dots (21) \quad \quad n32+n33+ n2+ n3 = S \dots (32) \\ n22+n23+n28+n29 = S \dots (22) \quad \quad n33+n34+ n3+ n4 = S \dots (33) \\ *n23+n24+n29+n30 = S \dots (23)* \quad \quad n34+n35+ n4+ n5 = S \dots (34) \\ n24+n19+n30+n25 = S \dots (24) \quad \quad *n35+n36+ n5+ n6 = S \dots (35)* \\ n25+n26+n31+n32 = S \dots (25) \quad \quad n36+n31+ n6+ n1 = S \dots (36) \end{array}$
---	---

Those 6 conditions (7), (11), (19), (23), (31) and (35) are reduced from the list of 36 above to be treated as 'Un-defined', and the rest 30 conditions are dictated definitely to be always fulfilled.

Of course, the other definitions as below must be fulfilled at the same time.

```

** Sums of Rows and Columns: K=111; Two Primary Diagonals: K=111 **
  n1+n2+n3+n4+n5+n6=K      ... (b1) | n1+n8+n15+n22+n29+n36=K ... (p1)
  n7+n8+n9+n10+n11+n12=K   ... (b2) |
  n13+n14+n15+n16+n17+n18=K ... (b3) |
  n19+n20+n21+n22+n23+n24=K ... (b4) |
  n25+n26+n27+n28+n29+n30=K ... (b5) |
  n31+n32+n33+n34+n35+n36=K ... (b6) |
  n1+n7+n13+n19+n25+n31=K   ... (b7) |
  n2+n8+n14+n20+n26+n32=K   ... (b8) |
  n3+n9+n15+n21+n27+n33=K   ... (b9) |
  n4+n10+n16+n22+n28+n34=K  ... (b10)|
  n5+n11+n17+n23+n29+n35=K  ... (b11)|
  n6+n12+n18+n24+n30+n36=K  ... (b12)| n6+n11+n16+n21+n26+n31=K ... (p2)

```

I have written such the 'C' program as listed below for this case.

```

/** 'Semi-Composite' Type 1 of Standard Magic Squares 6x6: **
/** Standard Solutions with Check-Sum Monitor of Composite 2x2 **
/** 'MS66SCmpst4ST.c': Dictated by Kanji Setsuda **
/** on Nov.1, 2005, June 26, 2013 and July 20, 2013; **
/** with MacOSX 10.8.4 and Xcode 4.6.3; **
//
#include <stdio.h>
//
/** Global Variables *
short int cnt;
short cnt2, cnt3, stlpr;
short LSM, SSM;
short nm[37], uflg[37];
short anm[5][37];
short csum[5][25];
short csm[5][37];
//
/** Sub-Routines *
void stp01(void), stp02(void), stp03(void);
void stp04(void), stp05(void), stp06(void);
void stp07(void), stp08(void), stp09(void);
void stp10(void), stp11(void), stp12(void);
void stp13(void), stp14(void), stp15(void);
void stp16(void), stp17(void), stp18(void);
void stp19(void), stp20(void), stp21(void);
void stp22(void), stp23(void), stp24(void);
void stp25(void), stp26(void), stp27(void);
void stp28(void), stp29(void), stp30(void);
void stp31(void), stp32(void), stp33(void);
void stp34(void), stp35(void), stp36(void);
void stp37(void);
void ansprint(void);
void printsol(short x);
//
void csums(short x);
void chksms(short x);
//
/** Main Program *
int main(){
short n;
printf("//\n");
printf("//** 'Semi-Composite' Type 1 of Standard Magic Squares 6x6:\n");
printf("//      Abstract List of Standard Solutions with precise Check-Sums: **\n");

```

```

printf("//\n");
LSM=111; SSM=74; cnt=0; cnt3=0;
for(n=0;n<37;n++){nm[n]=0; uflg[n]=0;}
stp01(); /* Begin the Calculations *
if(cnt3>0){printsol(cnt3);}
printf("// [Count = %d]\n",cnt);
printf("//\n");
printf("/* OK! *\n");
printf("/** Calculated and Listed by Kanji Setsuda\n");
printf("// on July 20, 2013 with MacOSX 10.8.4 & Xcode 4.6.3 **\n");
printf("//\n");
return 0;
}
//
/** Begin the Calculations *
/** Set N1 *
void stp01(){
short i;
for(i=1;i<18;i++){
if(uflg[i]==0){cnt2=0; /* Terminal Point *
nm[1]=i; uflg[i]=1;
stp02();
uflg[i]=0;}
}
}
//
/** Set N2 *
void stp02(){
short i;
for(i=1;i<37;i++){
if(uflg[i]==0){if(nm[1]==1){cnt2=0;} /* Terminal Point *
nm[2]=i; uflg[i]=1;
stp03();
uflg[i]=0;}
}
}
//
/** Set N7(>N2) *
void stp03(){
short i;
for(i=nm[2]+1;i<37;i++){
if(uflg[i]==0){
nm[7]=i; uflg[i]=1;
stp04();
uflg[i]=0;}
}
}
//
/** Set n8=74-n1-n2-n7 *
void stp04(){
short i;
i=SSM-nm[1]-nm[2]-nm[7];
if((0<i)&&(i<37)){
if(uflg[i]==0){
nm[8]=i; uflg[i]=1;
stp05();
uflg[i]=0;}}
}
//
/** Set N3 *
void stp05(){
short i;
for(i=1;i<37;i++){
if(uflg[i]==0){
nm[3]=i; uflg[i]=1;
stp06();

```

```

        uflg[i]=0;}
    }
}
//
/** Set n9=74-n2-n3-n8 *
void stp06(){
    short i;
    i=SSM-nm[2]-nm[3]-nm[8];
    if((0<i)&&(i<37)){
        if(uflg[i]==0){
            nm[9]=i; uflg[i]=1;
            stp07();
            uflg[i]=0;}}
}
//
/** Set N4 *
void stp07(){
    short i;
    for(i=1;i<37;i++){
        if(uflg[i]==0){
            nm[4]=i; uflg[i]=1;
            stp08();
            uflg[i]=0;}
    }
}
//
/** Set n10=74-n3-n4-n9 *
void stp08(){
    short i;
    i=SSM-nm[3]-nm[4]-nm[9];
    if((0<i)&&(i<37)){
        if(uflg[i]==0){
            nm[10]=i; uflg[i]=1;
            stp09();
            uflg[i]=0;}}
}
//
/** Set N5 *
void stp09(){
    short i;
    for(i=1;i<37;i++){
        if(uflg[i]==0){
            nm[5]=i; uflg[i]=1;
            stp10();
            uflg[i]=0;}
    }
}
//
/** Set n11=74-n4-n5-n10 *
void stp10(){
    short i;
    i=SSM-nm[4]-nm[5]-nm[10];
    if((0<i)&&(i<37)){
        if(uflg[i]==0){
            nm[11]=i; uflg[i]=1;
            stp11();
            uflg[i]=0;}}
}
//
/** Set n6=LSM-n1-n2-n3-n4-n5 & n6>N1 *
void stp11(){
    short i;
    i=LSM-nm[1]-nm[2]-nm[3]-nm[4]-nm[5];
    if((nm[1]<i)&&(i<37)){
        if(uflg[i]==0){
            nm[6]=i; uflg[i]=1;

```

```

        stp12();
        uflg[i]=0;}
    }
}
//
/** Set n12=111-n7-n8-n9-n10-n11 *
void stp12(){
short i,j,k;
i=LSM-nm[7]-nm[8]-nm[9]-nm[10]-nm[11];
j=SSM-nm[5]-nm[6]-nm[11];
k=SSM-nm[6]-nm[1]-nm[7];
if((0<i)&&(i<37)){
    if((i==j)&&(i==k)){
        if(uflg[i]==0){
            nm[12]=i; uflg[i]=1;
            stp13();
            uflg[i]=0;}}}}
}
//
/** Search Level 2: *
/** Set N13 *
void stp13(){
short i;
for(i=1;i<37;i++){
    if(uflg[i]==0){
        nm[13]=i; uflg[i]=1;
        stp14();
        uflg[i]=0;}
}
}
//
/** Set N14 *
void stp14(){
short i;
for(i=1;i<37;i++){
    if(uflg[i]==0){
        nm[14]=i; uflg[i]=1;
        stp15();
        uflg[i]=0;}
}
}
//
/** Set n15=74-n8-n9-n14 *
void stp15(){
short i;
i=SSM-nm[8]-nm[9]-nm[14];
if((0<i)&&(i<37)){
    if(uflg[i]==0){
        nm[15]=i; uflg[i]=1;
        stp16();
        uflg[i]=0;}}
}
//
/** Set n16=74-n9-n10-n15 *
void stp16(){
short i;
i=SSM-nm[9]-nm[10]-nm[15];
if((0<i)&&(i<37)){
    if(uflg[i]==0){
        nm[16]=i; uflg[i]=1;
        stp17();
        uflg[i]=0;}}
}
//
/** Set n17=74-n10-n11-n16 *
void stp17(){

```

```

short i;
i=SSM-nm[10]-nm[11]-nm[16];
if((0<i)&&(i<37)){
    if(uflg[i]==0){
        nm[17]=i; uflg[i]=1;
        stp18();
        uflg[i]=0;}}
}
//
/** Set n18=111-n13-n14-n15-n16-n17 *
void stp18(){
short i,j;
i=LSM-nm[13]-nm[14]-nm[15]-nm[16]-nm[17];
j=SSM-nm[12]-nm[7]-nm[13];
if((0<i)&&(i<37)){
    if(i==j){
        if(uflg[i]==0){
            nm[18]=i; uflg[i]=1;
            stp19();
            uflg[i]=0;}}}
}
//
/** Search Level 3: *
/** Set N19 *
void stp19(){
short i;
for(i=1;i<37;i++){
    if(uflg[i]==0){
        nm[19]=i; uflg[i]=1;
        stp20();
        uflg[i]=0;}
}
}
//
/** Set n20=74-n13-n14-n19 *
void stp20(){
short i;
i=SSM-nm[13]-nm[14]-nm[19];
if((0<i)&&(i<37)){
    if(uflg[i]==0){
        nm[20]=i; uflg[i]=1;
        stp21();
        uflg[i]=0;}}
}
/** Set n21=74-n14-n15-n20 *
void stp21(){
short i;
i=SSM-nm[14]-nm[15]-nm[20];
if((0<i)&&(i<37)){
    if(uflg[i]==0){
        nm[21]=i; uflg[i]=1;
        stp22();
        uflg[i]=0;}}
}
/** Set n22=74-n15-n16-n21 *
void stp22(){
short i;
i=SSM-nm[15]-nm[16]-nm[21];
if((0<i)&&(i<37)){
    if(uflg[i]==0){
        nm[22]=i; uflg[i]=1;
        stp23();
        uflg[i]=0;}}
}
/** Set n23=74-n16-n17-n22 *
void stp23(){

```

```

short i;
i=SSM-nm[16]-nm[17]-nm[22];
if((0<i)&&(i<37)){
    if(uflg[i]==0){
        nm[23]=i; uflg[i]=1;
        stp24();
        uflg[i]=0;}}
}
//
/** Set n24=111-n19-n20-n21-n22-n23 *
void stp24(){
short i,j,k;
i=LSM-nm[19]-nm[20]-nm[21]-nm[22]-nm[23];
j=SSM-nm[17]-nm[18]-nm[23];
k=SSM-nm[18]-nm[13]-nm[19];
if((0<i)&&(i<37)){
    if((i==j)&&(i==k)){
        if(uflg[i]==0){
            nm[24]=i; uflg[i]=1;
            stp25();
            uflg[i]=0;}}}
}
//
/** Search Level 4: *
/** Set N25 *
void stp25(){
short i;
for(i=1;i<37;i++){
    if(uflg[i]==0){
        nm[25]=i; uflg[i]=1;
        stp26();
        uflg[i]=0;}}
}
//
/** Set N26 *
void stp26(){
short i;
for(i=1;i<37;i++){
    if(uflg[i]==0){
        nm[26]=i; uflg[i]=1;
        stp27();
        uflg[i]=0;}}
}
//
/** Set n27=74-n20-n21-n26 *
void stp27(){
short i;
i=SSM-nm[20]-nm[21]-nm[26];
if((0<i)&&(i<37)){
    if(uflg[i]==0){
        nm[27]=i; uflg[i]=1;
        stp28();
        uflg[i]=0;}}
}
//
/** Set n28=74-n21-n22-n27 *
void stp28(){
short i;
i=SSM-nm[21]-nm[22]-nm[27];
if((0<i)&&(i<37)){
    if(uflg[i]==0){
        nm[28]=i; uflg[i]=1;
        stp29();
        uflg[i]=0;}}
}
/** Set n29=74-n22-n23-n28 *

```

```

void stp29(){
short i;
i=SSM-nm[22]-nm[23]-nm[28];
if((0<i)&&(i<37)){
if(uflg[i]==0){
nm[29]=i; uflg[i]=1;
stp30();
uflg[i]=0;}}
}
/* Set n30=111-n25-n26-n27-n28-n29 *
void stp30(){
short i,j;
i=LSM-nm[25]-nm[26]-nm[27]-nm[28]-nm[29];
j=SSM-nm[24]-nm[19]-nm[25];
if((0<i)&&(i<37)){
if((i==j)&&(uflg[i]==0)){
nm[30]=i; uflg[i]=1;
stp31();
uflg[i]=0;}}
}
//
/* Search Level 5: *
/* Set n31=111-n1-n7-n13-n19-n25 & n31>N1 *
void stp31(){
short i,j;
i=LSM-nm[1]-nm[7]-nm[13]-nm[19]-nm[25];
j=LSM-nm[6]-nm[11]-nm[16]-nm[21]-nm[26];
if((nm[1]<i)&&(i<37)){
if((i==j)&&(uflg[i]==0)){
nm[31]=i; uflg[i]=1;
stp32();
uflg[i]=0;}}
}
/* Set n32=111-n2-n8-n14-n20-n26 *
void stp32(){
short i,j;
i=LSM-nm[2]-nm[8]-nm[14]-nm[20]-nm[26];
j=SSM-nm[25]-nm[26]-nm[31];
if((0<i)&&(i<37)&&(i==j)){
if(uflg[i]==0){
nm[32]=i; uflg[i]=1;
stp33();
uflg[i]=0;}}
}
/* Set n33=111-n3-n9-n15-n21-n27 *
void stp33(){
short i,j,k;
i=LSM-nm[3]-nm[9]-nm[15]-nm[21]-nm[27];
j=SSM-nm[26]-nm[27]-nm[32];
k=SSM-nm[32]-nm[2]-nm[3];
if((0<i)&&(i<37)){
if((i==j)&&(i==k)){
if(uflg[i]==0){
nm[33]=i; uflg[i]=1;
stp34();
uflg[i]=0;}}}}
}
/* Set n34=111-n4-n10-n16-n22-n28 *
void stp34(){
short i,j,k;
i=LSM-nm[4]-nm[10]-nm[16]-nm[22]-nm[28];
j=SSM-nm[27]-nm[28]-nm[33];
k=SSM-nm[33]-nm[3]-nm[4];
if((0<i)&&(i<37)){
if((i==j)&&(i==k)){
if(uflg[i]==0){

```



```

        nm[34]=i; uflg[i]=1;
        stp35();
        uflg[i]=0;}}
}
/** Set n35=111-n5-n11-n17-n23-n29 *
void stp35(){
short i,j,k;
i=LSM-nm[5]-nm[11]-nm[17]-nm[23]-nm[29];
j=SSM-nm[28]-nm[29]-nm[34];
k=SSM-nm[34]-nm[4]-nm[5];
if((0<i)&&(i<37)){
    if((i==j)&&(i==k)){
        if(uflg[i]==0){
            nm[35]=i; uflg[i]=1;
            stp36();
            uflg[i]=0;}}}}
}
//
/** Set n36=111-n1-n8-n15-n22-n29 & N1<n36 *
void stp36(){
short i,j,k,l;
i=LSM-nm[1]-nm[8]-nm[15]-nm[22]-nm[29];
j=SSM-nm[29]-nm[30]-nm[35];
k=SSM-nm[30]-nm[25]-nm[31];
l=SSM-nm[31]-nm[6]-nm[1];
if((nm[1]<i)&&(i<37)){
    if((i==j)&&(i==k)&&(i==l)){
        if(uflg[i]==0){
            nm[36]=i; uflg[i]=1;
            stp37();
            uflg[i]=0;}}}}
}
//
/** The Last Checks *
void stp37(){
short sm1,sm2;
sm1=nm[6]+nm[12]+nm[18]+nm[24]+nm[30]+nm[36];
sm2=nm[31]+nm[32]+nm[33]+nm[34]+nm[35]+nm[36];
if((sm1==LSM)&&(sm2==LSM)){ansprint();}
}
//
/** Print The Answers *
void ansprint(){
short n;
cnt++; cnt2++;
if(cnt2==1){
    anm[cnt3][0]=cnt;
    for(n=1;n<37;n++){anm[cnt3][n]=nm[n];}
    csums(cnt3);
    chksms(cnt3);
    cnt3++; if(cnt3==2){printsol(cnt3); cnt3=0;}
}
}
//
/** Print the Answers with precise Check-Sums *
void printsol(short x){
short l,m,n;
printf("//");
for(m=0;m<x;m++){
    printf("%17d#|Row|Clm\\Pd1/Pd2|Cp4",anm[m][0]);
    if(m+1<x){printf("
");}
}
printf("\n");
for(l=0;l<6;l++){
    printf("//");
    for(m=0;m<x;m++){

```

```

        for(n=1;n<7;n++){printf("%3d", anm[m][l*6+n]);}
        printf(" |%3d|%3d\\", csum[m][l+1], csum[m][l+7]);
        printf("%3d/%3d|", csum[m][l+13], csum[m][l+19]);
        for(n=1;n<7;n++){printf("%3d", csm[m][l*6+n]);}
        printf("|");
        if(m+1<x){printf(" ");}
    }
    printf("\\n");
}
printf("//\\n");
}
//
/** Check Sums of Rows, Columns and Pan-Diagonals *
void csums(short x){
    csum[x][1]=nm[1]+nm[2]+nm[3]+nm[4]+nm[5]+nm[6];
    csum[x][2]=nm[7]+nm[8]+nm[9]+nm[10]+nm[11]+nm[12];
    csum[x][3]=nm[13]+nm[14]+nm[15]+nm[16]+nm[17]+nm[18];
    csum[x][4]=nm[19]+nm[20]+nm[21]+nm[22]+nm[23]+nm[24];
    csum[x][5]=nm[25]+nm[26]+nm[27]+nm[28]+nm[29]+nm[30];
    csum[x][6]=nm[31]+nm[32]+nm[33]+nm[34]+nm[35]+nm[36];
    csum[x][7]=nm[1]+nm[7]+nm[13]+nm[19]+nm[25]+nm[31];
    csum[x][8]=nm[2]+nm[8]+nm[14]+nm[20]+nm[26]+nm[32];
    csum[x][9]=nm[3]+nm[9]+nm[15]+nm[21]+nm[27]+nm[33];
    csum[x][10]=nm[4]+nm[10]+nm[16]+nm[22]+nm[28]+nm[34];
    csum[x][11]=nm[5]+nm[11]+nm[17]+nm[23]+nm[29]+nm[35];
    csum[x][12]=nm[6]+nm[12]+nm[18]+nm[24]+nm[30]+nm[36];
    /** Pan-diagonal Check: **
    csum[x][13]=nm[1]+nm[8]+nm[15]+nm[22]+nm[29]+nm[36];
    csum[x][14]=nm[2]+nm[9]+nm[16]+nm[23]+nm[30]+nm[31];
    csum[x][15]=nm[3]+nm[10]+nm[17]+nm[24]+nm[25]+nm[32];
    csum[x][16]=nm[4]+nm[11]+nm[18]+nm[19]+nm[26]+nm[33];
    csum[x][17]=nm[5]+nm[12]+nm[13]+nm[20]+nm[27]+nm[34];
    csum[x][18]=nm[6]+nm[7]+nm[14]+nm[21]+nm[28]+nm[35];
    csum[x][19]=nm[1]+nm[12]+nm[17]+nm[22]+nm[27]+nm[32];
    csum[x][20]=nm[2]+nm[7]+nm[18]+nm[23]+nm[28]+nm[33];
    csum[x][21]=nm[3]+nm[8]+nm[13]+nm[24]+nm[29]+nm[34];
    csum[x][22]=nm[4]+nm[9]+nm[14]+nm[19]+nm[30]+nm[35];
    csum[x][23]=nm[5]+nm[10]+nm[15]+nm[20]+nm[25]+nm[36];
    csum[x][24]=nm[6]+nm[11]+nm[16]+nm[21]+nm[26]+nm[31];
}
//
/** Check Sums of Composite Fours *
void chksms(short x){
    csm[x][1]=nm[1]+ nm[2]+ nm[7 ]+nm[8];
    csm[x][2]=nm[2]+ nm[3]+ nm[8]+ nm[9];
    csm[x][3]=nm[3]+ nm[4]+ nm[9]+nm[10];
    csm[x][4]=nm[4]+ nm[5]+nm[10]+nm[11];
    csm[x][5]=nm[5]+ nm[6]+nm[11]+nm[12];
    csm[x][6]=nm[6]+ nm[1]+nm[12]+ nm[7];
    csm[x][7]=nm[7]+ nm[8]+nm[13]+nm[14];
    csm[x][8]=nm[8]+ nm[9]+nm[14]+nm[15];
    csm[x][9]=nm[9]+nm[10]+nm[15]+nm[16];
    csm[x][10]=nm[10]+nm[11]+nm[16]+nm[17];
    csm[x][11]=nm[11]+nm[12]+nm[17]+nm[18];
    csm[x][12]=nm[12]+ nm[7]+nm[18]+nm[13];
    csm[x][13]=nm[13]+nm[14]+nm[19]+nm[20];
    csm[x][14]=nm[14]+nm[15]+nm[20]+nm[21];
    csm[x][15]=nm[15]+nm[16]+nm[21]+nm[22];
    csm[x][16]=nm[16]+nm[17]+nm[22]+nm[23];
    csm[x][17]=nm[17]+nm[18]+nm[23]+nm[24];
    csm[x][18]=nm[18]+nm[13]+nm[24]+nm[19];
    csm[x][19]=nm[19]+nm[20]+nm[25]+nm[26];
    csm[x][20]=nm[20]+nm[21]+nm[26]+nm[27];
    csm[x][21]=nm[21]+nm[22]+nm[27]+nm[28];
    csm[x][22]=nm[22]+nm[23]+nm[28]+nm[29];
    csm[x][23]=nm[23]+nm[24]+nm[29]+nm[30];
}

```

```

csm[x][24]=nm[24]+nm[19]+nm[30]+nm[25];
csm[x][25]=nm[25]+nm[26]+nm[31]+nm[32];
csm[x][26]=nm[26]+nm[27]+nm[32]+nm[33];
csm[x][27]=nm[27]+nm[28]+nm[33]+nm[34];
csm[x][28]=nm[28]+nm[29]+nm[34]+nm[35];
csm[x][29]=nm[29]+nm[30]+nm[35]+nm[36];
csm[x][30]=nm[30]+nm[25]+nm[36]+nm[31];
csm[x][31]=nm[31]+nm[32]+ nm[1]+ nm[2];
csm[x][32]=nm[32]+nm[33]+ nm[2]+ nm[3];
csm[x][33]=nm[33]+nm[34]+ nm[3]+ nm[4];
csm[x][34]=nm[34]+nm[35]+ nm[4]+ nm[5];
csm[x][35]=nm[35]+nm[36]+ nm[5]+ nm[6];
csm[x][36]=nm[36]+nm[31]+ nm[6]+ nm[1];
}
//

```

This is one of the smallest programs I have written. But it can make the perfect list of solutions displaying the check-sum monitors of every conditions.

**** 'Semi-Composite' Type 1 of Standard Magic Squares 6x6:
Abstract List of Standard Solutions with precise Check-Sums: ****

```

1#|Row|C|I|Pd1|Pd2|Cp4      13#|Row|C|I|Pd1|Pd2|Cp4
1 2 20 26 32 30|111|111\111/ 90| 74 74 74 74 74 74| 1 3 18 23 35 31|111|111\111/ 90| 74 74 74 74 74 74|
36 35 17 11 5 7|111|111\ 90/124| 82 74 74 74 66 74| 36 34 19 14 2 6|111|111\ 90/123| 83 74 74 74 65 74|
8 3 19 27 31 23|111|111\119/106| 74 74 74 74 74 74| 8 5 16 25 33 24|111|111\120/108| 74 74 74 74 74 74|
29 34 18 10 6 14|111|111\116/103| 87 74 74 74 61 74| 29 32 21 12 4 13|111|111\114/102| 86 74 74 74 62 74|
15 9 13 33 25 16|111|111\ 98/132| 74 74 74 74 74 74| 15 10 11 30 28 17|111|111\ 99/132| 74 74 74 74 74 74|
22 28 24 4 12 21|111|111\132/111| 53 74 74 74 95 74| 22 27 26 7 9 20|111|111\132/111| 53 74 74 74 95 74|

53#|Row|C|I|Pd1|Pd2|Cp4      101#|Row|C|I|Pd1|Pd2|Cp4
1 4 15 22 33 36|111|111\111/ 90| 74 74 74 74 74 74| 1 5 13 25 31 36|111|111\111/ 92| 74 74 74 74 74 74|
34 35 20 17 2 3|111|111\ 90/121| 85 74 74 74 63 74| 33 35 21 15 3 4|111|111\ 92/120| 84 74 74 74 64 74|
9 7 12 25 30 28|111|111\122/112| 74 74 74 74 74 74| 14 2 16 22 34 23|111|111\121/112| 74 74 74 74 74 74|
27 31 24 13 6 10|111|111\110/100| 84 74 74 74 64 74| 28 30 26 10 8 9|111|111\110/101| 83 74 74 74 65 74|
21 5 14 23 32 16|111|111\101/132| 74 74 74 74 74 74| 18 7 11 27 29 19|111|111\102/130| 74 74 74 74 74 74|
19 29 26 11 8 18|111|111\132/111| 53 74 74 74 95 74| 17 32 24 12 6 20|111|111\130/111| 55 74 74 74 93 74|

185#|Row|C|I|Pd1|Pd2|Cp4      281#|Row|C|I|Pd1|Pd2|Cp4
1 6 12 24 32 36|111|111\111/ 92| 74 74 74 74 74 74| 1 7 11 27 29 36|111|111\111/ 98| 74 74 74 74 74 74|
33 34 22 16 2 4|111|111\ 92/120| 84 74 74 74 64 74| 31 35 21 15 3 6|111|111\ 98/116| 82 74 74 74 66 74|
14 3 15 21 35 23|111|111\121/112| 74 74 74 74 74 74| 12 4 14 24 32 25|111|111\119/114| 74 74 74 74 74 74|
28 29 27 11 7 9|111|111\110/101| 83 74 74 74 65 74| 28 30 26 10 8 9|111|111\108/103| 79 74 74 74 69 74|
18 8 10 26 30 19|111|111\102/130| 74 74 74 74 74 74| 19 2 16 22 34 18|111|111\106/124| 74 74 74 74 74 74|
17 31 25 13 5 20|111|111\130/111| 55 74 74 74 93 74| 20 33 23 13 5 17|111|111\124/111| 61 74 74 74 87 74|

381#|Row|C|I|Pd1|Pd2|Cp4      479#|Row|C|I|Pd1|Pd2|Cp4
1 8 10 26 30 36|111|111\111/ 98| 74 74 74 74 74 74| 1 9 11 26 28 36|111|111\111/ 94| 74 74 74 74 74 74|
31 34 22 16 2 6|111|111\ 98/116| 82 74 74 74 66 74| 29 35 19 18 2 8|111|111\ 94/125| 77 74 74 74 71 74|
12 5 13 23 33 25|111|111\119/114| 74 74 74 74 74 74| 6 7 13 24 30 31|111|111\114/100| 74 74 74 74 74 74|
28 29 27 11 7 9|111|111\108/103| 79 74 74 74 69 74| 27 34 20 17 3 10|111|111\122/108| 88 74 74 74 60 74|
19 3 15 21 35 18|111|111\106/124| 74 74 74 74 74 74| 23 4 16 21 33 14|111|111\ 97/128| 74 74 74 74 74 74|
20 32 24 14 4 17|111|111\124/111| 61 74 74 74 87 74| 25 22 32 5 15 12|111|111\128/111| 57 74 74 74 91 74|

595#|Row|C|I|Pd1|Pd2|Cp4      693#|Row|C|I|Pd1|Pd2|Cp4
1 10 14 23 27 36|111|111\111/ 72| 74 74 74 74 74 74| 1 11 8 31 24 36|111|111\111/ 92| 74 74 74 74 74 74|
28 35 15 22 2 9|111|111\ 72/143| 81 74 74 74 67 74| 27 35 20 15 4 10|111|111\ 92/120| 84 74 74 74 64 74|
12 6 18 19 31 25|111|111\118/ 86| 74 74 74 74 74 74| 19 3 16 23 32 18|111|111\121/112| 74 74 74 74 74 74|
32 24 26 11 13 5|111|111\136/104|106 74 74 74 42 74| 30 22 33 2 17 7|111|111\110/101| 83 74 74 74 65 74|
30 20 4 33 17 7|111|111\ 79/150| 74 74 74 74 74 74| 25 6 13 26 29 12|111|111\102/130| 74 74 74 74 74 74|
8 16 34 3 21 29|111|111\150/111| 35 74 74 74 113 74| 9 34 21 14 5 28|111|111\130/111| 55 74 74 74 93 74|

773#|Row|C|I|Pd1|Pd2|Cp4      909#|Row|C|I|Pd1|Pd2|Cp4
1 12 7 33 22 36|111|111\111/ 94| 74 74 74 74 74 74| 1 13 6 29 26 36|111|111\111/ 92| 74 74 74 74 74 74|
27 34 21 13 6 10|111|111\ 94/119| 83 74 74 74 65 74| 27 33 22 17 2 10|111|111\ 92/120| 84 74 74 74 64 74|
19 3 16 24 31 18|111|111\120/112| 74 74 74 74 74 74| 19 5 14 21 34 18|111|111\121/112| 74 74 74 74 74 74|
29 23 32 2 17 8|111|111\110/102| 82 74 74 74 66 74| 30 20 35 4 15 7|111|111\110/101| 83 74 74 74 65 74|
26 4 15 25 30 11|111|111\103/128| 74 74 74 74 74 74| 25 8 11 24 31 12|111|111\102/130| 74 74 74 74 74 74|
9 35 20 14 5 28|111|111\128/111| 57 74 74 74 91 74| 9 32 23 16 3 28|111|111\130/111| 55 74 74 74 93 74|

```

<p>973# Row C l m\Pd1/Pd2 Cp4</p> <p>1 14 17 21 22 36 111 111\111/ 94 74 74 74 74 74 74 </p> <p>24 35 8 28 3 13 111 111\ 94/120 82 74 74 74 66 74 </p> <p>7 16 15 23 20 30 111 111\119/110 74 74 74 74 74 74 </p> <p>19 32 11 25 6 18 111 111\112/103 83 74 74 74 65 74 </p> <p>27 5 26 12 31 10 111 111\102/128 74 74 74 74 74 74 </p> <p>33 9 34 2 29 4 111 111\128/111 57 74 74 74 91 74 </p>	<p>1059# Row C l m\Pd1/Pd2 Cp4</p> <p>1 15 16 20 23 36 111 111\111/ 94 74 74 74 74 74 74 </p> <p>24 34 9 29 2 13 111 111\ 94/120 82 74 74 74 66 74 </p> <p>7 17 14 22 21 30 111 111\119/110 74 74 74 74 74 74 </p> <p>19 31 12 26 5 18 111 111\112/103 83 74 74 74 65 74 </p> <p>27 6 25 11 32 10 111 111\102/128 74 74 74 74 74 74 </p> <p>33 8 35 3 28 4 111 111\128/111 57 74 74 74 91 74 </p>
--	---

<p>1179# Row C l m\Pd1/Pd2 Cp4</p> <p>1 16 6 31 21 36 111 111\111/ 86 74 74 74 74 74 74 </p> <p>23 34 18 19 3 14 111 111\ 86/132 78 74 74 74 70 74 </p> <p>11 10 12 25 27 26 111 111\115/ 94 74 74 74 74 74 74 </p> <p>33 20 32 5 17 4 111 111\128/107 95 74 74 74 53 74 </p> <p>35 7 15 22 30 2 111 111\ 90/136 74 74 74 74 74 74 </p> <p>8 24 28 9 13 29 111 111\136/111 49 74 74 74 99 74 </p>	<p>1315# Row C l m\Pd1/Pd2 Cp4</p> <p>1 17 14 24 19 36 111 111\111/ 82 74 74 74 74 74 74 </p> <p>21 35 8 28 3 16 111 111\ 82/129 85 74 74 74 63 74 </p> <p>27 2 29 9 34 10 111 111\122/104 74 74 74 74 74 74 </p> <p>33 12 31 5 26 4 111 111\118/100 92 74 74 74 56 74 </p> <p>22 25 6 32 11 15 111 111\ 93/140 74 74 74 74 74 74 </p> <p>7 20 23 13 18 30 111 111\140/111 45 74 74 74 103 74 </p>
---	--

<p>1385# Row C l m\Pd1/Pd2 Cp4</p> <p>1 18 12 25 19 36 111 111\111/ 98 74 74 74 74 74 74 </p> <p>21 34 10 27 3 16 111 111\ 98/121 77 74 74 74 71 74 </p> <p>5 17 13 24 20 32 111 111\114/104 74 74 74 74 74 74 </p> <p>23 29 15 22 8 14 111 111\118/108 84 74 74 74 64 74 </p> <p>30 2 28 9 35 7 111 111\101/124 74 74 74 74 74 74 </p> <p>31 11 33 4 26 6 111 111\124/111 61 74 74 74 87 74 </p>	<p>1455# Row C l m\Pd1/Pd2 Cp4</p> <p>1 19 12 23 20 36 111 111\111/104 74 74 74 74 74 74 </p> <p>21 33 10 29 2 16 111 111\104/103 89 74 74 74 59 74 </p> <p>30 5 26 9 34 7 111 111\126/134 74 74 74 74 74 74 </p> <p>31 8 35 4 27 6 111 111\ 88/ 96 66 74 74 74 82 74 </p> <p>13 14 17 18 25 24 111 111\119/118 74 74 74 74 74 74 </p> <p>15 32 11 28 3 22 111 111\118/111 67 74 74 74 81 74 </p>
---	---

<p>1529# Row C l m\Pd1/Pd2 Cp4</p> <p>1 20 14 23 17 36 111 111\111/ 94 74 74 74 74 74 74 </p> <p>21 32 8 29 5 16 111 111\ 94/108 94 74 74 74 54 74 </p> <p>19 22 12 25 15 18 111 111\131/134 74 74 74 74 74 74 </p> <p>26 7 33 4 30 11 111 111\ 88/ 91 71 74 74 74 77 74 </p> <p>35 3 31 6 34 2 111 111\114/128 74 74 74 74 74 74 </p> <p>9 27 13 24 10 28 111 111\128/111 57 74 74 74 91 74 </p>	<p>1657# Row C l m\Pd1/Pd2 Cp4</p> <p>1 21 14 19 20 36 111 111\111/100 74 74 74 74 74 74 </p> <p>24 28 11 30 5 13 111 111\100/132 64 74 74 74 84 74 </p> <p>3 9 26 7 32 34 111 111\101/ 80 74 74 74 74 74 74 </p> <p>35 27 12 29 6 2 111 111\142/121 95 74 74 74 53 74 </p> <p>15 18 17 16 23 22 111 111\ 90/122 74 74 74 74 74 74 </p> <p>33 8 31 10 25 4 111 111\122/111 63 74 74 74 85 74 </p>
---	---

<p>1775# Row C l m\Pd1/Pd2 Cp4</p> <p>1 22 8 29 15 36 111 111\111/ 78 74 74 74 74 74 74 </p> <p>23 28 16 21 9 14 111 111\ 78/147 71 74 74 74 77 74 </p> <p>17 3 27 10 34 20 111 111\108/ 72 74 74 74 74 74 74 </p> <p>35 19 25 12 18 2 111 111\150/114 110 74 74 74 38 74 </p> <p>30 26 4 33 11 7 111 111\ 75/144 74 74 74 74 74 74 </p> <p>5 13 31 6 24 32 111 111\144/111 41 74 74 74 107 74 </p>	<p>1911# Row C l m\Pd1/Pd2 Cp4</p> <p>1 23 14 22 15 36 111 111\111/126 74 74 74 74 74 74 </p> <p>24 26 11 27 10 13 111 111\126/ 90 80 74 74 74 68 74 </p> <p>21 9 28 8 29 16 111 111\117/138 74 74 74 74 74 74 </p> <p>25 19 18 20 17 12 111 111\ 84/105 53 74 74 74 95 74 </p> <p>5 4 33 3 34 32 111 111\132/ 96 74 74 74 74 74 74 </p> <p>35 30 7 31 6 2 111 111\ 96/111 89 74 74 74 59 74 </p>
---	---

<p>2037# Row C l m\Pd1/Pd2 Cp4</p> <p>1 24 7 30 13 36 111 111\111/110 74 74 74 74 74 74 </p> <p>27 22 21 16 15 10 111 111\110/130 56 74 74 74 92 74 </p> <p>5 2 29 8 35 32 111 111\ 93/ 74 74 74 74 74 74 74 </p> <p>33 34 9 28 3 4 111 111\148/129 93 74 74 74 55 74 </p> <p>14 12 19 18 25 23 111 111\ 92/112 74 74 74 74 74 74 </p> <p>31 17 26 11 20 6 111 111\112/111 73 74 74 74 75 74 </p>	<p>2161# Row C l m\Pd1/Pd2 Cp4</p> <p>1 25 11 26 12 36 111 111\111/128 74 74 74 74 74 74 </p> <p>27 21 17 20 16 10 111 111\128/ 93 75 74 74 74 73 74 </p> <p>19 8 28 9 29 18 111 111\112/130 74 74 74 74 74 74 </p> <p>24 23 15 22 14 13 111 111\ 92/110 56 74 74 74 92 74 </p> <p>7 2 34 3 35 30 111 111\129/ 94 74 74 74 74 74 74 </p> <p>33 32 6 31 5 4 111 111\ 94/111 91 74 74 74 57 74 </p>
---	---

<p>2237# Row C l m\Pd1/Pd2 Cp4</p> <p>1 26 4 33 11 36 111 111\111/ 82 74 74 74 74 74 74 </p> <p>27 20 24 13 17 10 111 111\ 82/146 68 74 74 74 80 74 </p> <p>19 2 28 9 35 18 111 111\105/ 70 74 74 74 74 74 74 </p> <p>30 23 21 16 14 7 111 111\152/117 109 74 74 74 39 74 </p> <p>31 25 5 32 12 6 111 111\ 76/140 74 74 74 74 74 74 </p> <p>3 15 29 8 22 34 111 111\140/111 45 74 74 74 103 74 </p>	<p>2329# Row C l m\Pd1/Pd2 Cp4</p> <p>1 27 10 14 23 36 111 111\111/ 94 74 74 74 74 74 74 </p> <p>28 18 19 31 6 9 111 111\ 94/118 84 74 74 74 64 74 </p> <p>12 26 11 13 24 25 111 111\121/114 74 74 74 74 74 74 </p> <p>32 4 33 17 20 5 111 111\108/101 81 74 74 74 67 74 </p> <p>30 15 22 2 35 7 111 111\104/128 74 74 74 74 74 74 </p> <p>8 21 16 34 3 29 111 111\128/111 57 74 74 74 91 74 </p>
---	---

<p>2389# Row C l m\Pd1/Pd2 Cp4</p> <p>1 28 3 20 23 36 111 111\111/ 86 74 74 74 74 74 74 </p> <p>29 16 27 24 7 8 111 111\ 86/131 79 74 74 74 69 74 </p> <p>15 19 12 11 32 22 111 111\116/ 96 74 74 74 74 74 74 </p> <p>31 9 34 17 14 6 111 111\126/106 94 74 74 74 54 74 </p> <p>33 21 10 13 30 4 111 111\ 91/136 74 74 74 74 74 74 </p> <p>2 18 25 26 5 35 111 111\136/111 49 74 74 74 99 74 </p>	<p>2445# Row C l m\Pd1/Pd2 Cp4</p> <p>1 29 4 33 8 36 111 111\111/ 94 74 74 74 74 74 74 </p> <p>30 14 27 10 23 7 111 111\ 94/135 67 74 74 74 81 74 </p> <p>12 11 22 15 26 25 111 111\104/ 80 74 74 74 74 74 74 </p> <p>16 35 6 31 2 21 111 111\142/118 98 74 74 74 50 74 </p> <p>34 13 20 17 24 3 111 111\ 87/128 74 74 74 74 74 74 </p> <p>18 9 32 5 28 19 111 111\128/111 57 74 74 74 91 74 </p>
---	---

<p>2521# Row C l m\Pd1/Pd2 Cp4</p> <p>1 30 13 10 21 36 111 111\111/ 86 74 74 74 74 74 74 </p> <p>31 12 19 32 11 6 111 111\ 86/140 70 74 74 74 78 74 </p> <p>2 25 18 5 26 35 111 111\107/ 78 74 74 74 74 74 74 </p> <p>33 14 17 34 9 4 111 111\144/115 103 74 74 74 45 74 </p> <p>29 27 16 7 24 8 111 111\ 82/136 74 74 74 74 74 74 </p> <p>15 3 28 23 20 22 111 111\136/111 49 74 74 74 99 74 </p>	<p>2585# Row C l m\Pd1/Pd2 Cp4</p> <p>1 31 12 13 18 36 111 111\111/ 84 74 74 74 74 74 74 </p> <p>32 10 21 28 15 5 111 111\ 84/140 72 74 74 74 76 74 </p> <p>3 27 16 9 22 34 111 111\109/ 80 74 74 74 74 74 74 </p> <p>33 11 20 29 14 4 111 111\142/113 103 74 74 74 45 74 </p> <p>35 24 19 6 25 2 111 111\ 82/138 74 74 74 74 74 74 </p> <p>7 8 23 26 17 30 111 111\138/111 47 74 74 74 101 74 </p>
--	---

```

2635#|Row|Clm\Pd1/Pd2|Cp4          2687#|Row|Clm\Pd1/Pd2|Cp4
1 32 5 25 12 36|111|111\111/110| 74 74 74 74 74 74| 1 33 4 29 8 36|111|111\111/ 98| 74 74 74 74 74 74|
33 8 29 15 22 4|111|111\110/129| 57 74 74 74 91 74| 35 5 32 9 28 2|111|111\ 98/136| 62 74 74 74 86 74|
7 9 28 2 35 30|111|111\ 94/ 76| 74 74 74 74 74 74| 15 7 30 3 34 22|111|111\ 99/ 74| 74 74 74 74 74 74|
31 27 10 34 3 6|111|111\146/128| 92 74 74 74 56 74| 25 27 10 31 6 12|111|111\148/123| 99 74 74 74 49 74|
16 18 19 11 26 21|111|111\ 93/112| 74 74 74 74 74 74| 24 23 14 19 18 13|111|111\ 86/124| 74 74 74 74 74 74|
23 17 20 24 13 14|111|111\112/111| 73 74 74 74 75 74| 11 16 21 20 17 26|111|111\124/111| 61 74 74 74 87 74|

2711#|111\111/          2743#|111\111/          5607#|111\111/          7309#|111\111/
1 34 3 24 13 36|111|111| 2 1 24 27 25 32|111|111| 3 1 21 25 33 28|111|111| 4 1 21 19 35 31|111|111|
35 4 33 14 23 2|111|111| 35 36 13 10 12 5|111|111| 34 36 16 12 4 9|111|111| 33 36 16 18 2 6|111|111|
11 16 21 6 31 26|111|111| 11 8 17 34 18 23|111|111| 8 7 15 31 27 23|111|111| 3 8 14 26 28 32|111|111|
29 18 19 28 9 8|111|111| 26 29 20 3 19 14|111|111| 24 35 17 11 5 19|111|111| 34 29 23 11 9 5|111|111|
25 17 20 7 30 12|111|111| 28 4 21 30 22 6|111|111| 13 2 20 26 32 18|111|111| 20 12 10 30 24 15|111|111|
10 22 15 32 5 27|111|111| 9 33 16 7 15 31|111|111| 29 30 22 6 10 14|111|111| 17 25 27 7 13 22|111|111|

9003#|111\111/          10867#|111\111/          12185#|111\111/          13285#|111\111/
5 1 20 23 35 27|111|111| 6 1 27 10 36 31|111|111| 7 1 15 25 35 28|111|111| 8 1 20 23 35 24|111|111|
32 36 17 14 2 10|111|111| 32 35 11 26 2 5|111|111| 30 36 22 12 2 9|111|111| 29 36 17 14 2 13|111|111|
11 6 15 28 30 21|111|111| 4 13 15 22 24 33|111|111| 17 5 11 29 31 18|111|111| 11 6 15 28 30 21|111|111|
26 31 22 9 7 16|111|111| 28 29 17 20 8 9|111|111| 20 32 26 8 6 19|111|111| 26 31 22 9 7 16|111|111|
8 3 18 25 33 24|111|111| 23 21 7 30 16 14|111|111| 14 3 13 27 33 21|111|111| 5 3 18 25 33 27|111|111|
29 34 19 12 4 13|111|111| 18 12 34 3 25 19|111|111| 23 34 24 10 4 16|111|111| 32 34 19 12 4 10|111|111|

14549#|111\111/          15469#|111\111/          16231#|111\111/          16863#|111\111/
9 1 22 29 27 23|111|111| 10 1 29 3 35 33|111|111| 11 1 22 29 27 21|111|111| 12 1 32 10 34 22|111|111|
28 36 15 8 10 14|111|111| 27 36 8 34 2 4|111|111| 26 36 15 8 10 16|111|111| 25 36 5 27 3 15|111|111|
2 3 20 31 25 30|111|111| 19 21 9 23 15 24|111|111| 2 3 20 31 25 30|111|111| 16 26 7 35 9 18|111|111|
35 34 17 6 12 7|111|111| 18 16 28 14 22 13|111|111| 35 34 17 6 12 7|111|111| 21 11 30 2 28 19|111|111|
11 4 19 32 24 21|111|111| 17 5 25 7 31 26|111|111| 9 4 19 32 24 23|111|111| 20 4 29 13 31 14|111|111|
26 33 18 5 13 16|111|111| 20 32 12 30 6 11|111|111| 28 33 18 5 13 14|111|111| 17 33 8 24 6 23|111|111|

17193#|111\111/          17893#|111\111/          18569#|111\111/          19091#|111\111/
13 1 18 25 30 24|111|111| 14 1 15 31 29 21|111|111| 15 1 11 25 35 24|111|111| 16 1 11 25 35 23|111|111|
31 29 26 5 14 6|111|111| 23 36 22 6 8 16|111|111| 22 36 26 12 2 13|111|111| 21 36 26 12 2 14|111|111|
2 3 16 27 28 35|111|111| 7 3 13 33 27 28|111|111| 16 5 7 29 31 23|111|111| 15 5 7 29 31 24|111|111|
33 36 19 12 7 4|111|111| 30 34 24 4 10 9|111|111| 21 32 30 8 6 14|111|111| 22 32 30 8 6 13|111|111|
15 10 9 34 21 22|111|111| 17 5 11 35 25 18|111|111| 20 9 3 33 27 19|111|111| 19 9 3 33 27 20|111|111|
17 32 23 8 11 20|111|111| 20 32 26 2 12 19|111|111| 17 28 34 4 10 18|111|111| 18 28 34 4 10 17|111|111|

19421#|111\111/
17 2 11 32 29 20|111|111|
22 33 28 3 10 15|111|111|
14 1 12 31 30 23|111|111|
24 35 26 5 8 13|111|111|
16 6 7 36 25 21|111|111|
18 34 27 4 9 19|111|111| [Count = 19568] * OK! *

```

** Calculated and Listed by Kanji Setsuda
on July 20, 2013 with MacOSX 10.8.4 & Xcode 4.6.3 **

We could have counted all the solutions through up to the last. How wonderful!
You may think ‘So many as about 20 thousands’ counts of solutions, but it is really so few that we could actually have counted them through.

Though they are the ‘Semi-Composite’ type, they really belong to the Standard Type of MS66. As we don’t yet know how many solutions this ordinary Standard Type has, they say more than 10^{16} , our special type of MS66 should be regarded as ‘rare’ and ‘precious’ solutions, I think.

In fact, when I once made another type of Semi-Composite MS66 with half of all conditions as many as 18 reduced, I could not count them up to the last. It really took too long a time to go on counting all through.

#2. Experiment No.2: with 32 Composite Conditions and 4 Un-defined

For the next case I designed and made the new Semi-Composite MS66 with the 4 conditions reduced and the rest 32 ones kept definitely dictated as follows.

```

** Definition-2: Composite Conditions: S=74 **
-----
n6 | n1 | n2 | n3 | n4 | n5 | n6 | n1      n1+ n2+ n7 +n8 = S .... (1)
---+---+---+---+---+---+
12 | n7 | n8 | n9 | 10 | 11 | 12 | n7      n2+ n3+ n8+ n9 = S .... (2)
---+---+---+---+---+
18 | 13 | 14 | 15 | 16 | 17 | 18 | 13      n3+ n4+ n9+n10 = S .... (3)
---+---+---+---+---+
24 | 19 | 20 | 21 | 22 | 23 | 24 | 19      n4+ n5+n10+n11 = S .... (4)
---+---+---+---+---+
30 | 25 | 26 | 27 | 28 | 29 | 30 | 25      n5+ n6+n11+n12 = S .... (5)
---+---+---+---+---+
36 | 31 | 32 | 33 | 34 | 35 | 36 | 31      n6+ n1+n12+ n7 = S .... (6)
-----
n6 n1 n2 n3 n4 n5 n6 n1      *n7+ n8+n13+n14 = S .... (7)*
n15+n16+n21+n22 = S ... (15) | n26+n27+n32+n33 = S ... (26)
n16+n17+n22+n23 = S ... (16) | n27+n28+n33+n34 = S ... (27)
n17+n18+n23+n24 = S ... (17) | n28+n29+n34+n35 = S ... (28)
n18+n13+n24+n19 = S ... (18) | n29+n30+n35+n36 = S ... (29)
n19+n20+n25+n26 = S ... (19) | n30+n25+n36+n31 = S ... (30)
n20+n21+n26+n27 = S ... (20) | *n31+n32+ n1+ n2 = S ... (31)*
n21+n22+n27+n28 = S ... (21) | n32+n33+ n2+ n3 = S ... (32)
n22+n23+n28+n29 = S ... (22) | n33+n34+ n3+ n4 = S ... (33)
n23+n24+n29+n30 = S ... (23) | n34+n35+ n4+ n5 = S ... (34)
n24+n19+n30+n25 = S ... (24) | *n35+n36+ n5+ n6 = S ... (35)*
n25+n26+n31+n32 = S ... (25) | n36+n31+ n6+ n1 = S ... (36)

```

Only the Conditions (7), (11), (31) and (35) are reduced to be 'Un-defined'. The rest 32 must be dictated to be always true.

```

** Sums of Rows and Columns: K=111; Two Primary Diagonals: K=111 **
n1+n2+n3+n4+n5+n6=K ... (b1) | n1+n8+n15+n22+n29+n36=K ... (p1)
n7+n8+n9+n10+n11+n12=K ... (b2)
n13+n14+n15+n16+n17+n18=K ... (b3)
n19+n20+n21+n22+n23+n24=K ... (b4)
n25+n26+n27+n28+n29+n30=K ... (b5)
n31+n32+n33+n34+n35+n36=K ... (b6)
n1+n7+n13+n19+n25+n31=K ... (b7)
n2+n8+n14+n20+n26+n32=K ... (b8)
n3+n9+n15+n21+n27+n33=K ... (b9)
n4+n10+n16+n22+n28+n34=K ... (b10)
n5+n11+n17+n23+n29+n35=K ... (b11)
n6+n12+n18+n24+n30+n36=K ... (b12) | n6+n11+n16+n21+n26+n31=K ... (p2)

```

Under these definitions above, I have got the following list of solutions.

** 'Semi-Composite' Type 2 of Standard Magic Squares 6x6:

Abstract List of Standard Solutions with precise Check-Sums: **

```

Sol_Num# | Row | C1m\Pd1/Pd2 | Composite Fours
-----
1# | Row | C1m\Pd1/Pd2 | Cp4
1 3 31 9 35 32 | 111 | 111\111/ 74 | 74 74 74 74 74 74 | 1 3 31 9 35 32 | 111 | 111\111/ 74 | 74 74 74 74 74 74
36 34 6 28 2 5 | 111 | 111\ 74/111 | 111 74 74 74 37 74 | 36 34 6 28 2 5 | 111 | 111\ 74/111 | 111 74 74 74 37 74
17 24 10 30 14 16 | 111 | 111\148/148 | 74 74 74 74 74 74 | 17 24 10 30 14 16 | 111 | 111\148/148 | 74 74 74 74 74 74
8 25 15 19 11 33 | 111 | 111\ 74/ 74 | 74 74 74 74 74 74 | 20 13 27 7 23 21 | 111 | 111\ 74/ 74 | 74 74 74 74 74 74
29 12 22 18 26 4 | 111 | 111\111/148 | 74 74 74 74 74 74 | 29 12 22 18 26 4 | 111 | 111\111/148 | 74 74 74 74 74 74
20 13 27 7 23 21 | 111 | 111\148/111 | 37 74 74 74 111 74 | 8 25 15 19 11 33 | 111 | 111\148/111 | 37 74 74 74 111 74

2# | Row | C1m\Pd1/Pd2 | Cp4
3# | Row | C1m\Pd1/Pd2 | Cp4
1 3 31 9 35 32 | 111 | 111\111/ 74 | 74 74 74 74 74 74 | 1 3 31 9 35 32 | 111 | 111\111/ 74 | 74 74 74 74 74 74
36 34 6 28 2 5 | 111 | 111\ 74/111 | 111 74 74 74 37 74 | 36 34 6 28 2 5 | 111 | 111\ 74/111 | 111 74 74 74 37 74
29 12 22 18 26 4 | 111 | 111\148/148 | 74 74 74 74 74 74 | 29 12 22 18 26 4 | 111 | 111\148/148 | 74 74 74 74 74 74
8 25 15 19 11 33 | 111 | 111\ 74/ 74 | 74 74 74 74 74 74 | 20 13 27 7 23 21 | 111 | 111\ 74/ 74 | 74 74 74 74 74 74
17 24 10 30 14 16 | 111 | 111\111/148 | 74 74 74 74 74 74 | 17 24 10 30 14 16 | 111 | 111\111/148 | 74 74 74 74 74 74
20 13 27 7 23 21 | 111 | 111\148/111 | 37 74 74 74 111 74 | 8 25 15 19 11 33 | 111 | 111\148/111 | 37 74 74 74 111 74

```


<p>29# Row C1m\Pd1/Pd2 Cp4</p> <p>1 20 29 8 17 36 111 111\111/ 74 74 74 74 74 74 74 </p> <p>32 21 4 33 16 5 111 111\ 74/111 111 74 74 74 37 74 </p> <p>31 27 22 15 10 6 111 111\148/148 74 74 74 74 74 74 </p> <p>3 13 12 25 24 34 111 111\ 74/ 74 74 74 74 74 74 74 </p> <p>35 23 26 11 14 2 111 111\111/148 74 74 74 74 74 74 </p> <p>9 7 18 19 30 28 111 111\148/111 37 74 74 74 111 74 </p>	<p>31# Row C1m\Pd1/Pd2 Cp4</p> <p>1 20 29 8 17 36 111 111\111/ 74 74 74 74 74 74 74 </p> <p>32 21 4 33 16 5 111 111\ 74/111 111 74 74 74 37 74 </p> <p>35 23 26 11 14 2 111 111\148/148 74 74 74 74 74 74 </p> <p>3 13 12 25 24 34 111 111\ 74/ 74 74 74 74 74 74 74 </p> <p>31 27 22 15 10 6 111 111\111/148 74 74 74 74 74 74 </p> <p>9 7 18 19 30 28 111 111\148/111 37 74 74 74 111 74 </p>
--	--

<p>33# Row C1m\Pd1/Pd2 Cp4</p> <p>2 4 22 28 34 21 111 111\111/ 92 74 74 74 74 74 74 </p> <p>35 33 15 9 3 16 111 111\ 92/111 93 74 74 74 55 74 </p> <p>13 12 14 36 26 10 111 111\130/130 74 74 74 74 74 74 </p> <p>20 29 19 5 7 31 111 111\ 92/ 92 74 74 74 74 74 74 </p> <p>17 8 18 32 30 6 111 111\111/130 74 74 74 74 74 74 </p> <p>24 25 23 1 11 27 111 111\130/111 55 74 74 74 93 74 </p>	<p>41# Row C1m\Pd1/Pd2 Cp4</p> <p>2 16 11 34 30 18 111 111\111/104 74 74 74 74 74 74 </p> <p>35 21 26 3 7 19 111 111\104/111 81 74 74 74 67 74 </p> <p>12 13 14 31 33 8 111 111\118/118 74 74 74 74 74 74 </p> <p>22 27 20 9 1 32 111 111\104/104 74 74 74 74 74 74 </p> <p>15 10 17 28 36 5 111 111\111/118 74 74 74 74 74 74 </p> <p>25 24 23 6 4 29 111 111\118/111 67 74 74 74 81 74 </p>
---	---

<p>49# Row C1m\Pd1/Pd2 Cp4</p> <p>2 17 12 34 30 16 111 111\111/106 74 74 74 74 74 74 </p> <p>35 20 25 3 7 21 111 111\106/111 79 74 74 74 69 74 </p> <p>10 14 15 31 33 8 111 111\116/116 74 74 74 74 74 74 </p> <p>24 26 19 9 1 32 111 111\106/106 74 74 74 74 74 74 </p> <p>13 11 18 28 36 5 111 111\111/116 74 74 74 74 74 74 </p> <p>27 23 22 6 4 29 111 111\116/111 69 74 74 74 79 74 </p>	<p>57# Row C1m\Pd1/Pd2 Cp4</p> <p>2 20 13 24 17 35 111 111\111/ 92 74 74 74 74 74 74 </p> <p>21 31 10 27 6 16 111 111\ 92/111 93 74 74 74 55 74 </p> <p>22 19 14 23 18 15 111 111\130/130 74 74 74 74 74 74 </p> <p>4 29 12 25 8 33 111 111\ 92/ 92 74 74 74 74 74 74 </p> <p>34 7 26 11 30 3 111 111\111/130 74 74 74 74 74 74 </p> <p>28 5 36 1 32 9 111 111\130/111 55 74 74 74 93 74 </p>
---	---

<p>65# Row C1m\Pd1/Pd2 Cp4</p> <p>2 20 5 32 17 35 111 111\111/120 74 74 74 74 74 74 </p> <p>29 23 26 11 14 8 111 111\120/111 65 74 74 74 83 74 </p> <p>10 3 22 15 34 27 111 111\102/102 74 74 74 74 74 74 </p> <p>28 33 16 21 4 9 111 111\120/120 74 74 74 74 74 74 </p> <p>12 1 24 13 36 25 111 111\111/102 74 74 74 74 74 74 </p> <p>30 31 18 19 6 7 111 111\102/111 83 74 74 74 65 74 </p>	<p>73# Row C1m\Pd1/Pd2 Cp4</p> <p>2 28 10 30 12 29 111 111\111/120 74 74 74 74 74 74 </p> <p>35 9 27 7 25 8 111 111\120/111 65 74 74 74 83 74 </p> <p>5 16 22 18 24 26 111 111\102/102 74 74 74 74 74 74 </p> <p>20 33 3 31 1 23 111 111\120/120 74 74 74 74 74 74 </p> <p>17 4 34 6 36 14 111 111\111/102 74 74 74 74 74 74 </p> <p>32 21 15 19 13 11 111 111\102/111 83 74 74 74 65 74 </p>
---	---

<p>89# Row C1m\Pd1/Pd2 Cp4</p> <p>2 30 10 28 12 29 111 111\111/120 74 74 74 74 74 74 </p> <p>35 7 27 9 25 8 111 111\120/111 65 74 74 74 83 74 </p> <p>5 18 22 16 24 26 111 111\102/102 74 74 74 74 74 74 </p> <p>20 31 3 33 1 23 111 111\120/120 74 74 74 74 74 74 </p> <p>17 6 34 4 36 14 111 111\111/102 74 74 74 74 74 74 </p> <p>32 19 15 21 13 11 111 111\102/111 83 74 74 74 65 74 </p>	<p>97# Row C1m\Pd1/Pd2 Cp4</p> <p>2 34 10 36 12 17 111 111\111/126 74 74 74 74 74 74 </p> <p>35 3 27 1 25 20 111 111\126/111 59 74 74 74 89 74 </p> <p>5 16 28 18 30 14 111 111\ 96/ 96 74 74 74 74 74 74 </p> <p>29 24 6 22 4 26 111 111\126/126 74 74 74 74 74 74 </p> <p>8 13 31 15 33 11 111 111\111/ 96 74 74 74 74 74 74 </p> <p>32 21 9 19 7 23 111 111\ 96/111 89 74 74 74 59 74 </p>
---	---

<p>SoI_Num# D1\D2/ Row C1m </p> <p>121# 111\111/</p> <p>3 1 21 25 33 28 111 111 4 12 21 20 22 32 111 111 </p> <p>34 36 16 12 4 9 111 111 33 25 16 17 15 5 111 111 </p> <p>8 7 15 31 27 23 111 111 1 10 23 18 24 35 111 111 </p> <p>24 35 17 11 5 19 111 111 36 27 14 19 13 2 111 111 </p> <p>13 2 20 26 32 18 111 111 8 3 30 11 31 28 111 111 </p> <p>29 30 22 6 10 14 111 111 29 34 7 26 6 9 111 111 </p>	<p>189# 111\111/</p> <p>5 2 28 6 34 36 111 111 6 4 32 5 33 31 111 111 </p> <p>32 35 9 31 3 1 111 111 28 36 2 35 1 9 111 111 </p> <p>21 23 7 27 13 20 111 111 18 29 7 30 8 19 111 111 </p> <p>4 26 18 22 12 29 111 111 10 17 21 16 20 27 111 111 </p> <p>33 11 19 15 25 8 111 111 34 13 23 14 24 3 111 111 </p> <p>16 14 30 10 24 17 111 111 15 12 26 11 25 22 111 111 </p>	<p>205# 111\111/</p> <p>11 2 16 30 34 18 111 111 </p> <p>29 15 34 3 22 8 111 111 26 35 21 7 3 19 111 111 </p> <p>6 7 18 19 30 31 111 111 12 8 10 36 28 17 111 111 </p> <p>28 33 16 21 4 9 111 111 22 32 24 4 6 23 111 111 </p> <p>12 1 24 13 36 25 111 111 15 5 13 33 31 14 111 111 </p> <p>26 35 14 23 2 11 111 111 25 29 27 1 9 20 111 111 </p>	<p>341# 111\111/</p> <p>17 11 2 35 26 20 111 111 </p> <p>14 5 11 35 29 17 111 111 15 9 3 33 27 24 111 111 </p> <p>23 32 26 2 8 20 111 111 22 28 34 4 10 13 111 111 </p> <p>16 6 10 36 28 15 111 111 16 5 7 29 31 23 111 111 </p> <p>19 33 25 3 7 24 111 111 21 32 30 8 6 14 111 111 </p> <p>18 4 12 34 30 13 111 111 20 1 11 25 35 19 111 111 </p> <p>21 31 27 1 9 22 111 111 17 36 26 12 2 18 111 111 </p>
--	--	--	---

<p>365# 111\111/</p> <p>8 11 16 30 34 12 111 111 </p> <p>29 26 21 7 3 25 111 111 </p> <p>15 17 10 36 28 5 111 111 </p> <p>19 23 24 4 6 35 111 111 </p> <p>18 14 13 33 31 2 111 111 </p> <p>22 20 27 1 9 32 111 111 </p>	<p>413# 111\111/</p> <p>9 1 35 2 36 28 111 111 </p> <p>31 33 5 32 4 6 111 111 </p> <p>22 25 11 26 12 15 111 111 </p> <p>3 24 14 23 13 34 111 111 </p> <p>27 20 16 21 17 10 111 111 </p> <p>19 8 30 7 29 18 111 111 </p>	<p>437# 111\111/</p> <p>10 20 5 32 17 27 111 111 </p> <p>29 15 34 3 22 8 111 111 </p> <p>6 7 18 19 30 31 111 111 </p> <p>28 33 16 21 4 9 111 111 </p> <p>12 1 24 13 36 25 111 111 </p> <p>26 35 14 23 2 11 111 111 </p>	<p>453# 111\111/</p> <p>11 2 16 30 34 18 111 111 </p> <p>26 35 21 7 3 19 111 111 </p> <p>12 8 10 36 28 17 111 111 </p> <p>22 32 24 4 6 23 111 111 </p> <p>15 5 13 33 31 14 111 111 </p> <p>25 29 27 1 9 20 111 111 </p>
---	---	---	---

<p>481# 111\111/</p> <p>12 2 16 30 34 17 111 111 </p> <p>25 35 21 7 3 20 111 111 </p> <p>11 8 10 36 28 18 111 111 </p> <p>23 32 24 4 6 22 111 111 </p> <p>14 5 13 33 31 15 111 111 </p> <p>26 29 27 1 9 19 111 111 </p>	<p>489# 111\111/</p> <p>13 10 4 34 28 22 111 111 </p> <p>24 27 33 3 9 15 111 111 </p> <p>14 6 8 30 32 21 111 111 </p> <p>19 35 25 11 1 20 111 111 </p> <p>18 2 12 26 36 17 111 111 </p> <p>23 31 29 7 5 16 111 111 </p>	<p>505# 111\111/</p> <p>14 5 11 35 29 17 111 111 </p> <p>23 32 26 2 8 20 111 111 </p> <p>16 6 10 36 28 15 111 111 </p> <p>19 33 25 3 7 24 111 111 </p> <p>18 4 12 34 30 13 111 111 </p> <p>21 31 27 1 9 22 111 111 </p>	<p>541# 111\111/</p> <p>15 9 3 33 27 24 111 111 </p> <p>22 28 34 4 10 13 111 111 </p> <p>16 5 7 29 31 23 111 111 </p> <p>21 32 30 8 6 14 111 111 </p> <p>20 1 11 25 35 19 111 111 </p> <p>17 36 26 12 2 18 111 111 </p>
---	---	---	---

<p>545# 111\111/</p> <p>16 9 3 33 27 23 111 111 </p> <p>21 28 34 4 10 14 111 111 </p> <p>15 5 7 29 31 24 111 111 </p> <p>22 32 30 8 6 13 111 111 </p> <p>19 1 11 25 35 20 111 111 </p> <p>18 36 26 12 2 17 111 111 </p>	<p>549# 111\111/</p> <p>17 11 2 35 26 20 111 111 </p> <p>14 32 29 8 5 23 111 111 </p> <p>16 9 4 33 28 21 111 111 </p> <p>22 27 34 3 10 15 111 111 </p> <p>24 1 12 25 36 13 111 111 </p> <p>18 31 30 7 6 19 111 111 </p>
---	---

[Count = 552] * OK! *

** Calculated and Listed by Kanji Setsuda
on June 26, 2013 with MacOSX 10.8.4 & Xcode 4.6.3 **

Why! We could have got all the solutions of this type again, as few as 552.
Imagine how rare and how precious they might be.
It took only 5 minutes to count them through.
I really got excited, as they are the real solutions of Magic Squares of Order 6.

#3. Experiment No.3: Another Type with 32 Conditions and 4 Un-defined

Let me show you another case of Semi-Composite MS66 with the 4 conditions reduced and the rest 32 definitely dictated at the definition stage.

```

** Definition-3: Composite Conditions: S=74 **
-----
n6|n1|n2|n3|n4|n5|n6|n1      n1+ n2+ n7 +n8 = S .... (1)
|---+---+---+---+---+---|      n2+ n3+ n8+ n9 = S .... (2)
12|n7|n8|n9|10|11|12|n7      n3+ n4+ n9+n10 = S .... (3)
|---+---+---+---+---+---|      n4+ n5+n10+n11 = S .... (4)
18|13|14|15|16|17|18|13      n5+ n6+n11+n12 = S .... (5)
|---+---+---+---+---+---|      n6+ n1+n12+ n7 = S .... (6)
24|19|20|21|22|23|24|19      n7+ n8+n13+n14 = S .... (7)
|---+---+---+---+---+---|      n8+ n9+n14+n15 = S .... (8)
30|25|26|27|28|29|30|25      n9+n10+n15+n16 = S .... (9)
|---+---+---+---+---+---|      n10+n11+n16+n17 = S ... (10)
36|31|32|33|34|35|36|31      n11+n12+n17+n18 = S ... (11)
|---+---+---+---+---+---|      n12+ n7+n18+n13 = S ... (12)
n6 n1 n2 n3 n4 n5 n6 n1      n13+n14+n19+n20 = S ... (13)
                               n14+n15+n20+n21 = S ... (14)
-----
n15+n16+n21+n22 = S ... (15) | n26+n27+n32+n33 = S ... (26)
n16+n17+n22+n23 = S ... (16) | n27+n28+n33+n34 = S ... (27)
n17+n18+n23+n24 = S ... (17) | n28+n29+n34+n35 = S ... (28)
n18+n13+n24+n19 = S ... (18) | n29+n30+n35+n36 = S ... (29)
*n19+n20+n25+n26 = S ... (19)* | n30+n25+n36+n31 = S ... (30)
n20+n21+n26+n27 = S ... (20) | *n31+n32+ n1+ n2 = S ... (31)*
n21+n22+n27+n28 = S ... (21) | n32+n33+ n2+ n3 = S ... (32)
n22+n23+n28+n29 = S ... (22) | n33+n34+ n3+ n4 = S ... (33)
*n23+n24+n29+n30 = S ... (23)* | n34+n35+ n4+ n5 = S ... (34)
n24+n19+n30+n25 = S ... (24) | *n35+n36+ n5+ n6 = S ... (35)*
n25+n26+n31+n32 = S ... (25) | n36+n31+ n6+ n1 = S ... (36)

```

As you see, I changed only the positions of two conditions reduced.
But the result is wonderful again. Let me show it to you now.

** 'Semi-Composite' Type 3 of Standard Magic Squares 6x6:
Abstract List of Standard Solutions with precise Check-Sum-Errors: **

```

Sol_Num#|Row|C1m\Pd1/Pd2|Composite Fours
1#|R|C\D/d|Cp4      2#|R|C\D/d|Cp4      3#|R|C\D/d|Cp4
1 8 15 22 29 36|0|0\0/-|000000| 1 8 29 22 15 36|0|0\0/-|000000| 1 8 15 22 29 36|0|0\0/-|000000|
31 34 17 20 3 6|0|0\-/ +|000000| 31 34 3 20 17 6|0|0\-/ +|000000| 32 33 18 19 4 5|0|0\-/ +|000000|
7 2 21 16 35 30|0|0\0/-|000000| 7 2 35 16 21 30|0|0\0/-|000000| 7 2 21 16 35 30|0|0\0/-|000000|
32 33 18 19 4 5|0|0\+ /0|+000-0| 32 33 4 19 18 5|0|0\+ /0|+000-0| 31 34 17 20 3 6|0|0\+ /0|+000-0|
13 11 12 25 26 24|0|0\-/ +|000000| 13 11 26 25 12 24|0|0\-/ +|000000| 13 11 12 25 26 24|0|0\-/ +|000000|
27 23 28 9 14 10|0|0\+ /0|-000+0| 27 23 14 9 28 10|0|0\+ /0|-000+0| 27 23 28 9 14 10|0|0\+ /0|-000+0|

4#|R|C\D/d|Cp4      5#|R|C\D/d|Cp4      6#|R|C\D/d|Cp4
1 8 29 22 15 36|0|0\0/-|000000| 1 8 20 24 25 33|0|0\0/-|000000| 1 8 25 24 20 33|0|0\0/-|000000|
32 33 4 19 18 5|0|0\-/ +|000000| 35 30 16 14 11 5|0|0\-/ +|000000| 35 30 11 14 16 5|0|0\-/ +|000000|
7 2 35 16 21 30|0|0\0/-|000000| 2 7 21 23 26 32|0|0\0/-|000000| 2 7 26 23 21 32|0|0\0/-|000000|
31 34 3 20 17 6|0|0\+ /0|+000-0| 36 29 17 13 12 4|0|0\+ /0|+000-0| 36 29 12 13 17 4|0|0\+ /0|+000-0|
13 11 26 25 12 24|0|0\-/ +|000000| 28 18 10 34 15 6|0|0\-/ +|000000| 28 18 15 34 10 6|0|0\-/ +|000000|
27 23 14 9 28 10|0|0\+ /0|-000+0| 9 19 27 3 22 31|0|0\+ /0|-000+0| 9 19 22 3 27 31|0|0\+ /0|-000+0|

```

<p>7# R C\D/d Cp4</p> <p>1 8 20 24 25 33 0 0\0/- 000000 </p> <p>36 29 17 13 12 4 0 0\-/ + 000000 </p> <p>2 7 21 23 26 32 0 0\0/- 000000 </p> <p>35 30 16 14 11 5 0 0\+ /0 +000-0 </p> <p>28 18 10 34 15 6 0 0\-/ + 000000 </p> <p>9 19 27 3 22 31 0 0\+ /0 -000+0 </p>	<p>8# R C\D/d Cp4</p> <p>1 8 25 24 20 33 0 0\0/- 000000 </p> <p>36 29 12 13 17 4 0 0\-/ + 000000 </p> <p>2 7 26 23 21 32 0 0\0/- 000000 </p> <p>35 30 11 14 16 5 0 0\+ /0 +000-0 </p> <p>28 18 15 34 10 6 0 0\-/ + 000000 </p> <p>9 19 22 3 27 31 0 0\+ /0 -000+0 </p>	<p>9# R C\D/d Cp4</p> <p>1 10 23 18 24 35 0 0\0/- 000000 </p> <p>29 34 7 26 6 9 0 0\-/ + 000000 </p> <p>8 3 30 11 31 28 0 0\0/- 000000 </p> <p>36 27 14 19 13 2 0 0\+ /0 +000-0 </p> <p>4 12 21 20 22 32 0 0\-/ + 000000 </p> <p>33 25 16 17 15 5 0 0\+ /0 -000+0 </p>
--	--	--

<p>10# R C\D/d Cp4</p> <p>1 10 24 18 23 35 0 0\0/- 000000 </p> <p>29 34 6 26 7 9 0 0\-/ + 000000 </p> <p>8 3 31 11 30 28 0 0\0/- 000000 </p> <p>36 27 13 19 14 2 0 0\+ /0 +000-0 </p> <p>4 12 22 20 21 32 0 0\-/ + 000000 </p> <p>33 25 15 17 16 5 0 0\+ /0 -000+0 </p>	<p>11# R C\D/d Cp4</p> <p>1 10 23 18 24 35 0 0\0/- 000000 </p> <p>36 27 14 19 13 2 0 0\-/ + 000000 </p> <p>8 3 30 11 31 28 0 0\0/- 000000 </p> <p>29 34 7 26 6 9 0 0\+ /0 +000-0 </p> <p>4 12 21 20 22 32 0 0\-/ + 000000 </p> <p>33 25 16 17 15 5 0 0\+ /0 -000+0 </p>	<p>12# R C\D/d Cp4</p> <p>1 10 24 18 23 35 0 0\0/- 000000 </p> <p>36 27 13 19 14 2 0 0\-/ + 000000 </p> <p>8 3 31 11 30 28 0 0\0/- 000000 </p> <p>29 34 6 26 7 9 0 0\+ /0 +000-0 </p> <p>4 12 22 20 21 32 0 0\-/ + 000000 </p> <p>33 25 15 17 16 5 0 0\+ /0 -000+0 </p>
---	---	---

<p>13# R C\D/d Cp4</p> <p>1 15 19 18 22 36 0 0\0/- 000000 </p> <p>25 33 7 30 4 12 0 0\-/ + 000000 </p> <p>3 13 21 16 24 34 0 0\0/- 000000 </p> <p>27 31 9 28 6 10 0 0\+ /0 +000-0 </p> <p>20 11 23 14 26 17 0 0\-/ + 000000 </p> <p>35 8 32 5 29 2 0 0\+ /0 -000+0 </p>	<p>15# R C\D/d Cp4</p> <p>1 15 19 18 22 36 0 0\0/- 000000 </p> <p>27 31 9 28 6 10 0 0\-/ + 000000 </p> <p>3 13 21 16 24 34 0 0\0/- 000000 </p> <p>25 33 7 30 4 12 0 0\+ /0 +000-0 </p> <p>20 11 23 14 26 17 0 0\-/ + 000000 </p> <p>35 8 32 5 29 2 0 0\+ /0 -000+0 </p>	<p>17# R C\D/d Cp4</p> <p>1 18 19 15 22 36 0 0\0/- 000000 </p> <p>25 30 7 33 4 12 0 0\-/ + 000000 </p> <p>3 16 21 13 24 34 0 0\0/- 000000 </p> <p>27 28 9 31 6 10 0 0\+ /0 +000-0 </p> <p>20 14 23 11 26 17 0 0\-/ + 000000 </p> <p>35 5 32 8 29 2 0 0\+ /0 -000+0 </p>
---	---	---

<p>19# R C\D/d Cp4</p> <p>1 18 19 15 22 36 0 0\0/- 000000 </p> <p>27 28 9 31 6 10 0 0\-/ + 000000 </p> <p>3 16 21 13 24 34 0 0\0/- 000000 </p> <p>25 30 7 33 4 12 0 0\+ /0 +000-0 </p> <p>20 14 23 11 26 17 0 0\-/ + 000000 </p> <p>35 5 32 8 29 2 0 0\+ /0 -000+0 </p>	<p>21# R C\D/d Cp4</p> <p>1 18 23 10 24 35 0 0\0/- 000000 </p> <p>29 26 7 34 6 9 0 0\-/ + 000000 </p> <p>8 11 30 3 31 28 0 0\0/- 000000 </p> <p>36 19 14 27 13 2 0 0\+ /0 +000-0 </p> <p>4 20 21 12 22 32 0 0\-/ + 000000 </p> <p>33 17 16 25 15 5 0 0\+ /0 -000+0 </p>	<p>23# R C\D/d Cp4</p> <p>1 18 23 10 24 35 0 0\0/- 000000 </p> <p>36 19 14 27 13 2 0 0\-/ + 000000 </p> <p>8 11 30 3 31 28 0 0\0/- 000000 </p> <p>29 26 7 34 6 9 0 0\+ /0 +000-0 </p> <p>4 20 21 12 22 32 0 0\-/ + 000000 </p> <p>33 17 16 25 15 5 0 0\+ /0 -000+0 </p>
---	---	---

<p>25# R C\D/d Cp4</p> <p>1 22 15 8 29 36 0 0\0/- 000000 </p> <p>31 20 17 34 3 6 0 0\-/ + 000000 </p> <p>7 16 21 2 35 30 0 0\0/- 000000 </p> <p>32 19 18 33 4 5 0 0\+ /0 +000-0 </p> <p>13 25 12 11 26 24 0 0\-/ + 000000 </p> <p>27 9 28 23 14 10 0 0\+ /0 -000+0 </p>	<p>27# R C\D/d Cp4</p> <p>1 22 15 8 29 36 0 0\0/- 000000 </p> <p>32 19 18 33 4 5 0 0\-/ + 000000 </p> <p>7 16 21 2 35 30 0 0\0/- 000000 </p> <p>31 20 17 34 3 6 0 0\+ /0 +000-0 </p> <p>13 25 12 11 26 24 0 0\-/ + 000000 </p> <p>27 9 28 23 14 10 0 0\+ /0 -000+0 </p>	<p>29# R C\D/d Cp4</p> <p>1 24 20 8 25 33 0 0\0/- 000000 </p> <p>35 14 16 30 11 5 0 0\-/ + 000000 </p> <p>2 23 21 7 26 32 0 0\0/- 000000 </p> <p>36 13 17 29 12 4 0 0\+ /0 +000-0 </p> <p>28 34 10 18 15 6 0 0\-/ + 000000 </p> <p>9 3 27 19 22 31 0 0\+ /0 -000+0 </p>
---	---	---

<p>33# R C\D/d Cp4</p> <p>1 29 6 31 8 36 0 0\0/- 000000 </p> <p>30 14 25 12 23 7 0 0\-/ + 000000 </p> <p>28 2 33 4 35 9 0 0\0/- 000000 </p> <p>10 34 5 32 3 27 0 0\+ /0 +000-0 </p> <p>16 17 18 19 20 21 0 0\-/ + 000000 </p> <p>26 15 24 13 22 11 0 0\+ /0 -000+0 </p>	<p>37# R C\D/d Cp4</p> <p>1 30 2 35 7 36 0 0\0/- 000000 </p> <p>33 10 32 5 27 4 0 0\-/ + 000000 </p> <p>25 6 26 11 31 12 0 0\0/- 000000 </p> <p>21 22 20 17 15 16 0 0\+ /0 +000-0 </p> <p>28 14 18 19 23 9 0 0\-/ + 000000 </p> <p>3 29 13 24 8 34 0 0\+ /0 -000+0 </p>	<p>41# R C\D/d Cp4</p> <p>2 7 16 21 30 35 0 0\0/- 000000 </p> <p>33 32 19 18 5 4 0 0\-/ + 000000 </p> <p>8 1 22 15 36 29 0 0\0/- 000000 </p> <p>34 31 20 17 6 3 0 0\+ /0 +000-0 </p> <p>10 12 11 26 25 27 0 0\-/ + 000000 </p> <p>24 28 23 14 9 13 0 0\+ /0 -000+0 </p>
---	---	---

<p>49# R C\D/d Cp4</p> <p>2 8 13 24 29 35 0 0\0/- 000000 </p> <p>28 36 17 20 1 9 0 0\-/ + 000000 </p> <p>4 6 15 22 31 33 0 0\0/- 000000 </p> <p>34 30 23 14 7 3 0 0\+ /0 +000-0 </p> <p>18 5 16 21 32 19 0 0\-/ + 000000 </p> <p>25 26 27 10 11 12 0 0\+ /0 -000+0 </p>	<p>53# R C\D/d Cp4</p> <p>2 12 14 36 26 21 0 0\0/- 000000 </p> <p>31 29 19 5 7 20 0 0\-/ + 000000 </p> <p>6 8 18 32 30 17 0 0\0/- 000000 </p> <p>35 25 23 1 11 16 0 0\+ /0 +000-0 </p> <p>13 4 22 28 34 10 0 0\-/ + 000000 </p> <p>24 33 15 9 3 27 0 0\+ /0 -000+0 </p>	<p>57# R C\D/d Cp4</p> <p>2 13 19 14 27 36 0 0\0/- 000000 </p> <p>28 31 11 30 3 8 0 0\-/ + 000000 </p> <p>9 6 26 7 34 29 0 0\0/- 000000 </p> <p>35 24 18 23 10 1 0 0\+ /0 +000-0 </p> <p>5 15 17 16 25 33 0 0\-/ + 000000 </p> <p>32 22 20 21 12 4 0 0\+ /0 -000+0 </p>
---	---	---

<p>65# R C\D/d Cp4</p> <p>2 14 11 26 23 35 0 0\0/- 000000 </p> <p>28 30 19 18 7 9 0 0\-/ + 000000 </p> <p>6 10 15 22 27 31 0 0\0/- 000000 </p> <p>34 24 25 12 13 3 0 0\+ /0 +000-0 </p> <p>36 17 8 29 20 1 0 0\-/ + 000000 </p> <p>5 16 33 4 21 32 0 0\+ /0 -000+0 </p>	<p>77# R C\D/d Cp4</p> <p>2 16 22 18 24 29 0 0\0/- 000000 </p> <p>23 33 3 31 1 20 0 0\-/ + 000000 </p> <p>14 4 34 6 36 17 0 0\0/- 000000 </p> <p>35 21 15 19 13 8 0 0\+ /0 +000-0 </p> <p>5 28 10 30 12 26 0 0\-/ + 000000 </p> <p>32 9 27 7 25 11 0 0\+ /0 -000+0 </p>	<p>85# R C\D/d Cp4</p> <p>2 18 22 16 24 29 0 0\0/- 000000 </p> <p>23 31 3 33 1 20 0 0\-/ + 000000 </p> <p>14 6 34 4 36 17 0 0\0/- 000000 </p> <p>35 19 15 21 13 8 0 0\+ /0 +000-0 </p> <p>5 30 10 28 12 26 0 0\-/ + 000000 </p> <p>32 7 27 9 25 11 0 0\+ /0 -000+0 </p>
---	---	---

93# R C\D/d Cp4	97# R C\D/d Cp4	101# R C\D/d Cp4
2 21 16 7 30 35 0 0\0/- 000000	2 23 21 7 26 32 0 0\0/- 000000	2 24 13 8 29 35 0 0\0/- 000000
33 18 19 32 5 4 0 0\-/ + 000000	35 14 16 30 11 5 0 0\-/ + 000000	28 20 17 36 1 9 0 0\-/ + 000000
8 15 22 1 36 29 0 0\0/- 000000	1 24 20 8 25 33 0 0\0/- 000000	4 22 15 6 31 33 0 0\0/- 000000
34 17 20 31 6 3 0 0\+ /0 +000-0	36 13 17 29 12 4 0 0\+ /0 +000-0	34 14 23 30 7 3 0 0\+ /0 +000-0
10 26 11 12 25 27 0 0\-/ + 000000	28 34 10 18 15 6 0 0\-/ + 000000	18 21 16 5 32 19 0 0\-/ + 000000
24 14 23 28 9 13 0 0\+ /0 -000+0	9 3 27 19 22 31 0 0\+ /0 -000+0	25 10 27 26 11 12 0 0\+ /0 -000+0

105# R C\D/d Cp4	109# R C\D/d Cp4	111# R C\D/d Cp4
2 26 11 14 23 35 0 0\0/- 000000	2 28 6 36 10 29 0 0\0/- 000000	2 29 5 32 8 35 0 0\0/- 000000
28 18 19 30 7 9 0 0\-/ + 000000	35 9 31 1 27 8 0 0\-/ + 000000	34 9 31 6 28 3 0 0\-/ + 000000
6 22 15 10 27 31 0 0\0/- 000000	26 4 30 12 34 5 0 0\0/- 000000	30 1 33 4 36 7 0 0\0/- 000000
34 12 25 24 13 3 0 0\+ /0 +000-0	11 33 7 25 3 32 0 0\+ /0 +000-0	16 27 13 24 10 21 0 0\+ /0 +000-0
36 29 8 17 20 1 0 0\-/ + 000000	17 16 18 24 22 14 0 0\-/ + 000000	17 19 15 22 18 20 0 0\-/ + 000000
5 4 33 16 21 32 0 0\+ /0 -000+0	20 21 19 13 15 23 0 0\+ /0 -000+0	12 26 14 23 11 25 0 0\+ /0 -000+0

115# R C\D/d Cp4	117# R C\D/d Cp4	125# R C\D/d Cp4
2 30 4 34 12 29 0 0\0/- 000000	2 31 14 13 33 18 0 0\0/- 000000	2 32 5 29 8 35 0 0\0/- 000000
35 7 33 3 25 8 0 0\-/ + 000000	32 9 20 27 1 22 0 0\-/ + 000000	34 6 31 9 28 3 0 0\-/ + 000000
26 6 28 10 36 5 0 0\0/- 000000	5 28 17 10 36 15 0 0\0/- 000000	30 4 33 1 36 7 0 0\0/- 000000
11 31 9 27 1 32 0 0\+ /0 +000-0	35 6 23 24 4 19 0 0\+ /0 +000-0	16 24 13 27 10 21 0 0\+ /0 +000-0
17 18 16 22 24 14 0 0\-/ + 000000	12 34 11 16 30 8 0 0\-/ + 000000	17 22 15 19 18 20 0 0\-/ + 000000
20 19 21 15 13 23 0 0\+ /0 -000+0	25 3 26 21 7 29 0 0\+ /0 -000+0	12 23 14 26 11 25 0 0\+ /0 -000+0

Sol_Num#|D1\D2/|Row|C|Im|

133# 0\0/	185# 0\0/	253# 0\0/	345# 0\0/
3 6 17 20 31 34 0 0	4 5 18 19 32 33 0 0	5 4 19 18 33 32 0 0	6 3 20 17 34 31 0 0
29 36 15 22 1 8 0 0	29 36 15 22 1 8 0 0	30 35 16 21 2 7 0 0	30 35 16 21 2 7 0 0
4 5 18 19 32 33 0 0	3 6 17 20 31 34 0 0	6 3 20 17 34 31 0 0	5 4 19 18 33 32 0 0
35 30 21 16 7 2 0 0	35 30 21 16 7 2 0 0	36 29 22 15 8 1 0 0	36 29 22 15 8 1 0 0
13 9 14 23 28 24 0 0	13 9 14 23 28 24 0 0	10 14 9 28 23 27 0 0	10 14 9 28 23 27 0 0
27 25 26 11 12 10 0 0	27 25 26 11 12 10 0 0	24 26 25 12 11 13 0 0	24 26 25 12 11 13 0 0

393# 0\0/	421# 0\0/	483# 0\0/	503# 0\0/
7 2 21 16 35 30 0 0	8 1 22 15 36 29 0 0	9 1 20 17 36 28 0 0	10 3 22 15 34 27 0 0
31 34 17 20 3 6 0 0	33 32 19 18 5 4 0 0	33 31 22 15 6 4 0 0	28 33 16 21 4 9 0 0
1 8 15 22 29 36 0 0	2 7 16 21 30 35 0 0	3 7 14 23 30 34 0 0	12 1 24 13 36 25 0 0
32 33 18 19 4 5 0 0	34 31 20 17 6 3 0 0	35 29 24 13 8 2 0 0	30 31 18 19 6 7 0 0
13 11 12 25 26 24 0 0	10 12 11 26 25 27 0 0	12 11 10 27 26 25 0 0	5 23 2 35 14 32 0 0
27 23 28 9 14 10 0 0	24 28 23 14 9 13 0 0	19 32 21 16 5 18 0 0	26 20 29 8 17 11 0 0

531# 0\0/	557# 0\0/	577# 0\0/	621# 0\0/
11 8 10 36 28 18 0 0	12 4 30 7 33 25 0 0	13 4 12 31 32 19 0 0	14 5 13 33 31 15 0 0
23 32 24 4 6 22 0 0	34 24 16 21 13 3 0 0	22 35 23 8 3 20 0 0	23 32 24 4 6 22 0 0
14 5 13 33 31 15 0 0	10 6 28 9 31 27 0 0	15 2 14 29 34 17 0 0	11 8 10 36 28 18 0 0
26 29 27 1 9 19 0 0	36 22 18 19 15 1 0 0	24 33 25 6 5 18 0 0	26 29 27 1 9 19 0 0
12 2 16 30 34 17 0 0	2 29 5 32 8 35 0 0	11 9 7 36 27 21 0 0	12 2 16 30 34 17 0 0
25 35 21 7 3 20 0 0	17 26 14 23 11 20 0 0	26 28 30 1 10 16 0 0	25 35 21 7 3 20 0 0

653# 0\0/	693# 0\0/
15 2 14 29 34 17 0 0	16 9 3 35 25 23 0 0
22 35 23 8 3 20 0 0	13 36 26 10 4 22 0 0
13 4 12 31 32 19 0 0	24 1 11 27 33 15 0 0
24 33 25 6 5 18 0 0	21 28 34 2 12 14 0 0
11 9 7 36 27 21 0 0	19 5 7 31 29 20 0 0
26 28 30 1 10 16 0 0	18 32 30 6 8 17 0 0

[Count = 720] * OK! *

** Calculated and Listed by Kanji Setsuda
on June 26, 2013 with MacOSX 10.8.4 & Xcode 4.6.3 **

How interesting! These 720 counts of solutions are a little more than before, but they still keep very few. It took less than 5 minutes to count all through. We could really have got another rare and precious set of solutions again. How wonderful it is! I must confess I really got excited.

I tried to make some other experiments for this kind of Semi-Composite type, but I could not find any answers for the case of 3 conditions reduced, of 2 conditions reduced, and of the fewest 1 reduced, either.

It is the newest conclusion that 4 reductions may be the 'minimum sacrifice'.

One more thing I must mention is that those solution sets above of the 2nd case and the 3rd are included in the 1st set of about 20 thousands solutions in page 11~13. I have known that by checking every solutions precisely.

Why? You might know the reason if you should try any comparative study of the definition lists of Composite Conditions for those three types.

Which equations did we reduce to be 'Un-defined'? Did you notice the reduction sets of the 2nd and the 3rd are actually included in the 1st case? They are exactly corresponding to the relation of those solution sets themselves.

The true meaning of 'Un-defined' conditions is that they don't care about the present sums of equations whether they are equal to the constant value 74 or not. This 'tolerant' attitude can allow any accidental sums. They may be equal to 74!

#4. Conclusion

What significance does the existence of our Semi-Composite MS66 have?

Does it have the same to all the Composite magic squares of all orders?

Every Composite types could make us see the whole world of our objects in a simplest way. We could see the total image of object at a glance, because the Composite Conditions always make an effective sampling of the big data of object.

With no exceptions even our 'Semi-Composite' type could certainly make such the same thing for us to see the whole world of 'Standard' MS66 at a glance.

The solutions of our Semi-Composite types really prove to be the good samples of the Standard MS66 themselves. We can now easily see the compact image of that vast world by the sampling data presented in our short lists of solutions, though we can never catch the vast monster of 10^{16} standard solutions at a glance.

How great these effects are! Don't you think so?

I can never stop my studying about all the Composite types of magic squares.

(Written by Kanji Setsuda on June 27 and July 31, 2013
with MacOSX 10.8.4 and Xcode 4.6.3;
E-Mail Address: jag12001@nifty.com)
