

## Chapter 4 : Commentary Articles No.2: **Kanji Setsuda**

"Various Arts and Tools for Studying Magic Squares"

### Section 7-3 : **Standard Magic Squares of Order 4 and 'Euler Squares'**

#### 0. Why don't you make the Standard Magic Squares 4x4 as 'Euler Squares'?

This section includes the same content as I wrote in the last part of the preceding Section 7-2 of the old version. But, I have to revise that part now and prefer to give a new independent section to it right here.

Can we make any 'Euler Squares' of Order 4 for the Standard type of Magic Squares? How about taking the 4-th Increment Number System at first?

#### 1. We can make No Pan-diagonal Magic Squares 4x4 by 4-th Increment System

We already know that we can make no 'Complete Euler Squares' of Order 4 for any Pan-magic type by the 4-th Increment Number System.

I have written about it in the end of Section 6, Chapter 4, Part 4 as follows:

Watch the next diagrams for the "Knight's Tour Composition" in the case of order 4. If you try to prevent any number from sitting next to the same one in the neighborhood by that method, you can certainly have the true 'Latin Pattern' (0,1,2,3) in every pan-diagonals of each Latin Unit, although you should inevitably have such the 'Non-Latin Patterns' as (0,2,0,2) or (1,3,1,3) in either any row or any column.

[Order 4]	T1/	T2/	T3/	T4/
0 3 2 1 0 3 2 1	1 3 1 3 1 3 1 3	0 3 2 1 0 3 2 1	1 3 1 3 1 3 1 3	
2 1   0   3   2   1   0   3	0 2   0   2   0   2   0   2	2 1   0   3   2   1   0   3	0 2   0   2   0   2   0   2	
0 3   2   1   0   3   2   1	3 1   3   1   3   1   3   1	0 3   2   1   0   3   2   1	3 1   3   1   3   1   3   1	
2 1   0   3   2   1   0   3	2 0   2   0   2   0   2   0	2 1   0   3   2   1   0   3	2 0   2   0   2   0   2   0	
0 3   2   1   0   3   2   1	1 3   1   3   1   3   1   3	0 3   2   1   0   3   2   1	1 3   1   3   1   3   1   3	
2 1 0 3 2 1 0 3	0 2 0 2 0 2 0 2	2 1 0 3 2 1 0 3	0 2 0 2 0 2 0 2	

No such units can follow the Basic Condition that all rows and columns must add up to the same Magic Constant.

It means you can make no 'Complete Euler Squares' of Order 4 for any Pan-magic type composing "Greco-Latin Squares" by the 4-th Increment Number System.

#### 2. How about making the Standard MS44 by the 4-th Increment System?

Can't we make any "Non-Complete type" of Euler Squares 4x4, regardless of pan-diagonals, for those Standard Magic Squares 4x4, then? Isn't it possible, either?

It seems impossible for us to make **all** the 880 solutions, because the Standard MS44 includes those 'impossible' 48 Pan-diagonal MS44 above and also the 48 Self-complementary MS44 in the standard set of solutions themselves.

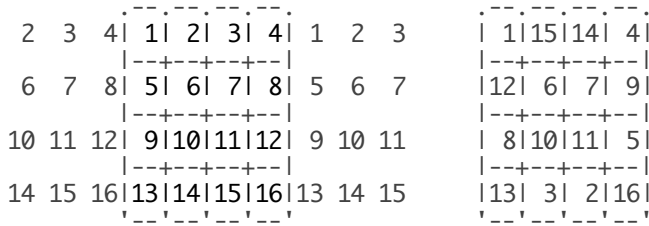
How many of them can we really make by the 4-th Increment System, then?

Let's make some experiments about it here, shall we?

**\*\* Basic Conditions for the Latin Units of Standard MS44: C=6 \*\***

n1+n2+n3+n4=C	...bc1;		n1+n5+n9+n13=C	...bc5;
n5+n6+n7+n8=C	...bc2;		n2+n6+n10+n14=C	...bc6;
n9+n10+n11+n12=C	...bc3;		n3+n7+n11+n15=C	...bc7;
n13+n14+n15+n16=C	...bc4;		n4+n8+n12+n16=C	...bc8;
n1+n6+n11+n16=C	...bc9;			
n4+n7+n10+n13=C	...bc10;			

**\*\* Basic Forms \*\***



**\*\* List-forming Inequality Conditions: \*\***

n1<n16, n1<n4, n1<n13, and n2>n5;

I have prepared the Diagram and Basic Conditions for our definitions as put above.

We must presuppose every 4 numbers of the ten Basic Conditions should take the 'Latin Pattern' (0,1,2,3) to make each Sum of them equal to the Magic Constant.

H/	L/	S#		0Hx4+0L+1=1S;	3Hx4+2L+1=15S;
0321	0231	1 15 12 6		2Hx4+3L+1=12S;	1Hx4+1L+1= 6S;
1230	3102	8 10 13 3		1Hx4+3L+1= 8S;	2Hx4+1L+1=10S;
3012	1320	14 4 7 9		3Hx4+0L+1=13S;	0Hx4+2L+1= 3S;
2103	2013	11 5 2 16		3Hx4+1L+1=14S;	0Hx4+3L+1= 4S;
				. . . . . :	mHx4+nL+1=pS ...cmp;

When we have got all the Latin Units by the "New Euler's Method", we pick up any two different units, combine them and compose our objective MS44 as shown above.

Let me show you the 'C' program I have dictated for that purpose as follows.

```

/** 'Standard' Magic Squares of Order 4: Composing the 'Euler Squares' **/
/** of Order 4 by "New Euler's Method" with 4-th Increment Number System **/
/** 'CES4Std4.c': Dictated by Kanji Setsuda on Oct.22,'04; Sep.15,'06; **/
/** Aug.10,'10; and Mar.17, 2011; with MacOSX 10.5.8 and Xcode 3.1.2 **/
/**/
#include <stdio.h>
/**/
short lcnt, cnt, cnt3;
short LSM;
short nm[17], uflg[17];
short tlu[49][17];
short mtc[49][49];
short ta[145][19];
short d[7][19];
short lnsm[2][9], pdsm[2][10];
/**/
short bcf[11][4];
short u1, u2;
/**/
void sstp01(void), sstp02(void), sstp03(void), sstp04(void);
void sstp05(void), sstp06(void), sstp07(void), sstp08(void);
void sstp09(void), sstp10(void), sstp11(void), sstp12(void);
void sstp13(void), sstp14(void), sstp15(void), sstp16(void);
void slurecord(void);
void prlunit(void);
void cmbcmp(void);
void recordsol(void);
void sortsol(void), exc(short x);
void chksum(short x);
void pr2sol(short x);
void pr6sol(short x, short y, short z);
/**/
/* Main Program */
int main(){

```

```

short m,n;
printf("\n** 'Euler Squares' of Order 4: Composed by the\n");
printf(" 'New Euler's Method' with the 4-th Increment Number System **\n");
for(n=0;n<17;n++){nm[n]=0;}
for(m=0;m<11;m++){for(n=0;n<2;n++){bcf[m][n]=0;}}
lcnt=0;
sstp01(); // Make Latin Units
printf("\n [List of Latin Units with Decompositions by 4-th Increment System]\n");
prlunit(); // Print Latin Units
cnt=0;
cmbcmp(); // Combine and Compose
sortsol(); // Sort the Solutions
printf("\n ** 'Standard' Magic Squares of Order 4: Composing the **\n");
printf(" ** 'Euler Squares' of Order 4 by 'New Euler's Method' **\n");
printf(" ** with the 4-th Increment Number System **\n");
printf("\n [List of Standard Solutions(Sorted):\n");
printf(" Used Units//Sol_N#($:Self-complementary; P:Pan-diagonal)|Sum-Checks|/]\n");
pr2sol(18);
pr6sol(19,36,1);
pr6sol(37,72,2);
pr6sol(73,cnt,4);
printf(" . . . . .\n");
printf(" [Solution Counts = %d]\n",cnt);
printf(" [OK!]\n");
printf("** Calculated and Listed by Kanji Setsuda on Mar.17, 2011 with MAC Xcode 3 **\n");
return 0;
}
/**/
/** Program Modules making the Latin Units of 'Euler Squares' of **/
/** Order 4 for the 'Standard' type of Magic Squares of Order 4 **/
/** Dictated by Kanji Setsuda on Mar.11, '11 with Mac Xcode 3 **/
/**/
/* Level 1: */
/* Set n1 */
void sstp01(){
short a;
for(a=0;a<4;a++){
if((bcf[1][a]==0)&&(bcf[5][a]==0)&&(bcf[9][a]==0)){
bcf[1][a]=1; bcf[5][a]=1; bcf[9][a]=1;
nm[1]=a; sstp02();
bcf[9][a]=0; bcf[5][a]=0; bcf[1][a]=0;
}
}
}
/**/
/* Set n16 */
void sstp02(){
short a;
for(a=3;a>=0;a--){
if((bcf[4][a]==0)&&(bcf[8][a]==0)&&(bcf[9][a]==0)){
bcf[4][a]=1; bcf[8][a]=1; bcf[9][a]=1;
nm[16]=a; sstp03();
bcf[9][a]=0; bcf[8][a]=0; bcf[4][a]=0;
}
}
}
/**/
/* Set n2 */
void sstp03(){
short a;
for(a=3;a>=0;a--){
if((bcf[1][a]==0)&&(bcf[6][a]==0)){
bcf[1][a]=1; bcf[6][a]=1;
nm[2]=a; sstp04();
}
}
}

```

```

        bcf[6][a]=0; bcf[1][a]=0;
    }
}
}
/**/
/* Set n15 */
void sstp04(){
    short a;
    for(a=0;a<4;a++){
        if((bcf[4][a]==0)&&(bcf[7][a]==0)){
            bcf[4][a]=1; bcf[7][a]=1;
            nm[15]=a; sstp05();
            bcf[7][a]=0; bcf[4][a]=0;
        }
    }
}
}
/**/
/* Set n3 */
void sstp05(){
    short a;
    for(a=3;a>=0;a--){
        if((bcf[1][a]==0)&&(bcf[7][a]==0)){
            bcf[1][a]=1; bcf[7][a]=1;
            nm[3]=a; sstp06();
            bcf[7][a]=0; bcf[1][a]=0;
        }
    }
}
}
/**/
/* Set n14 */
void sstp06(){
    short a;
    for(a=0;a<4;a++){
        if((bcf[4][a]==0)&&(bcf[6][a]==0)){
            bcf[4][a]=1; bcf[6][a]=1;
            nm[14]=a; sstp07();
            bcf[6][a]=0; bcf[4][a]=0;
        }
    }
}
}
}
/**/
/* Set n4 */
void sstp07(){
    short a;
    for(a=0;a<4;a++){
        if((bcf[1][a]==0)&&(bcf[8][a]==0)&&(bcf[10][a]==0)){
            bcf[1][a]=1; bcf[8][a]=1; bcf[10][a]=1;
            nm[4]=a; sstp08();
            bcf[10][a]=0; bcf[8][a]=0; bcf[1][a]=0;
        }
    }
}
}
}
/**/
/* Set n13 */
void sstp08(){
    short a;
    for(a=3;a>=0;a--){
        if((bcf[4][a]==0)&&(bcf[5][a]==0)&&(bcf[10][a]==0)){
            bcf[4][a]=1; bcf[5][a]=1; bcf[10][a]=1;
            nm[13]=a; sstp09();
            bcf[10][a]=0; bcf[5][a]=0; bcf[4][a]=0;
        }
    }
}
}
}

```

```

/**/
/* Level 2: */
/* Set n5 */
void sstp09(){
    short a;
    for(a=3;a>=0;a--){
        if((bcf[2][a]==0)&&(bcf[5][a]==0)){
            bcf[2][a]=1; bcf[5][a]=1;
            nm[5]=a; sstp10();
            bcf[5][a]=0; bcf[2][a]=0;
        }
    }
}
/**/
/* Set n12 */
void sstp10(){
    short a;
    for(a=0;a<4;a++){
        if((bcf[3][a]==0)&&(bcf[8][a]==0)){
            bcf[3][a]=1; bcf[8][a]=1;
            nm[12]=a; sstp11();
            bcf[8][a]=0; bcf[3][a]=0;
        }
    }
}
/**/
/* Set n6 */
void sstp11(){
    short a;
    for(a=0;a<4;a++){
        if((bcf[2][a]==0)&&(bcf[6][a]==0)&&(bcf[9][a]==0)){
            bcf[2][a]=1; bcf[6][a]=1; bcf[9][a]=1;
            nm[6]=a; sstp12();
            bcf[9][a]=0; bcf[6][a]=0; bcf[2][a]=0;
        }
    }
}
/**/
/* Set n11 */
void sstp12(){
    short a;
    for(a=3;a>=0;a--){
        if((bcf[3][a]==0)&&(bcf[7][a]==0)&&(bcf[9][a]==0)){
            bcf[3][a]=1; bcf[7][a]=1; bcf[9][a]=1;
            nm[11]=a; sstp13();
            bcf[9][a]=0; bcf[7][a]=0; bcf[3][a]=0;
        }
    }
}
/**/
/* Set n7 */
void sstp13(){
    short a;
    for(a=0;a<4;a++){
        if((bcf[2][a]==0)&&(bcf[7][a]==0)&&(bcf[10][a]==0)){
            bcf[2][a]=1; bcf[7][a]=1; bcf[10][a]=1;
            nm[7]=a; sstp14();
            bcf[10][a]=0; bcf[7][a]=0; bcf[2][a]=0;
        }
    }
}
/**/
/* Set n10 */
void sstp14(){

```

```

short a;
for(a=3;a>=0;a--){
    if((bcf[3][a]==0)&&(bcf[6][a]==0)&&(bcf[10][a]==0)){
        bcf[3][a]=1; bcf[6][a]=1; bcf[10][a]=1;
        nm[10]=a; sstp15();
        bcf[10][a]=0; bcf[6][a]=0; bcf[3][a]=0;
    }
}
}
/**/
/* Set n8 */
void sstp15(){
short a;
for(a=3;a>=0;a--){
    if((bcf[2][a]==0)&&(bcf[8][a]==0)){
        bcf[2][a]=1; bcf[8][a]=1;
        nm[8]=a; sstp16();
        bcf[8][a]=0; bcf[2][a]=0;
    }
}
}
/**/
/* Set n9 */
void sstp16(){
short a;
for(a=0;a<4;a++){
    if((bcf[3][a]==0)&&(bcf[5][a]==0)){
        bcf[3][a]=1; bcf[5][a]=1;
        nm[9]=a; slurecord();
        bcf[5][a]=0; bcf[3][a]=0;
    }
}
}
/**/
/* Save the Latin Units */
void slurecord(){
short n;
tlu[lcnt][0]=lcnt+1;
for(n=1;n<17;n++){tlu[lcnt][n]=nm[n];}
lcnt++;
}
/**/
/* Make Matching Table and Print the Latin Units */
void prlunit(){
short t,l,l4,m,n;
for(m=0;m<lcnt;m++){
    for(n=0;n<lcnt;n++){
        t=0;
        for(l=1;l<17;l++){if(tlu[m][l]==tlu[n][l]){t++;}}
        mtc[m][n]=t;
    }
}
for(t=0;t<lcnt;t=t+8){
    printf("%8d/%9d/%9d/%9d/%9d/%9d/%9d/%9d\n",t+1,t+2,t+3,t+4,t+5,t+6,t+7,t+8);
    for(l=0;l<4;l++){l4=l*4;
        for(m=t;m<(t+8);m++){
            printf(" ");
            for(n=1;n<5;n++){printf("%2d",tlu[m][l4+n]);}
            if(m<(t+7)){printf(" ");}
        }
        printf("\n");
    }
}
printf(" [Count of Latin Units = %d]\n",lcnt);

```

```

}
/**/
/* Combine and Compose */
void cmbcmp(){
short lu,luh,md,n,d,fc;
lu=lcnt; luh=lu/4; md=8;
for(u1=0;u1<luh;u1++){
for(u2=0;u2<lu;u2++){
for(n=1;n<17;n++){uflg[n]=0;}
for(n=1;n<17;n++){
d=tlu[u1][n]*4+tlu[u2][n]+1;
nm[n]=d; uflg[d]++;
}
fc=0;
for(n=1;n<17;n++){if(uflg[n]==1){fc++;}else{break;}}
if(fc==16){
if((nm[1]<nm[16])&&(nm[1]<nm[4])&&(nm[1]<nm[13])&&(nm[2]>nm[5])){recordsol();}
}
}}
}
/**/
/* Record the Solutions to the Table */
void recordsol(){
short n;
ta[cnt][0]=cnt+1;
for(n=1;n<17;n++){ta[cnt][n]=nm[n];}
ta[cnt][17]=u1; ta[cnt][18]=u2;
cnt++;
}
/**/
/* Solution Sorting for the Smart List */
void sortsol(){
short mx,m,f;
short d1,d2,d3,d4,d5,d6;
mx=cnt-1;
do{f=0;
for(m=0;m<mx;m++){d1=ta[m][1]; d2=ta[m+1][1];
if(d1>d2){exc(m); f=1;}
else{
d3=(ta[m][16]*17+ta[m][2])*17+ta[m][3];
d4=(ta[m+1][16]*17+ta[m+1][2])*17+ta[m+1][3];
if((d1==d2)&&(d3<d4)){exc(m); f=1;}
else{
d5=(ta[m][15]*17+ta[m][14])*17+ta[m][4];
d6=(ta[m+1][15]*17+ta[m+1][14])*17+ta[m+1][4];
if((d1==d2)&&(d3==d4)&&(d5>d6)){exc(m); f=1;}
}
}
}
mx--;
}while(f>0);
}
/**/
void exc(short x){
short n,d;
for(n=1;n<19;n++){d=ta[x][n]; ta[x][n]=ta[x+1][n]; ta[x+1][n]=d;}
}
/**/
/* Check every sums of rows and columns */
void chksum(short x){
short m,m4,n,s;
for(m=0;m<4;m++){m4=m*4; s=0;
for(n=1;n<5;n++){s=s+d[x][m4+n];}
lnsm[x][m]=s; s=0;
for(n=0;n<4;n++){s=s+d[x][n*4+m+1];}
}
}

```

```

    lnsms[x][m+4]=s;
}
pdsms[x][0]=d[x][1]+d[x][6]+d[x][11]+d[x][16];
pdsms[x][1]=d[x][2]+d[x][7]+d[x][12]+d[x][13];
pdsms[x][2]=d[x][3]+d[x][8]+d[x][9]+d[x][14];
pdsms[x][3]=d[x][4]+d[x][5]+d[x][10]+d[x][15];
pdsms[x][4]=d[x][1]+d[x][8]+d[x][11]+d[x][14];
pdsms[x][5]=d[x][2]+d[x][5]+d[x][12]+d[x][15];
pdsms[x][6]=d[x][3]+d[x][6]+d[x][9]+d[x][16];
pdsms[x][7]=d[x][4]+d[x][7]+d[x][10]+d[x][13];
s=0;
for(n=0;n<8;n++){if(pdsms[x][n]==LSM){s++;}else{break;}}
pdsms[x][8]=s;
s=0;
for(n=1;n<17;n++){if(d[x][n]+d[x][17-n]==17){s++;}else{break;}}
pdsms[x][9]=s;
}
/**/
/* Print 2 Answers */
void pr2sol(short x){
short t,l,l4,m,n,s,v;
for(t=0;t<x;t=t+2){
for(s=0;s<2;s++){
for(n=1;n<22;n++){d[s][n]=ta[t+s][n];}
chksum(s);
printf("%5d/%4d/%12d",d[s][17]+1,d[s][18]+1,t+s+1);
if(pdsms[s][8]==8){printf("PIRWICLIP1\\P2/");}
else if(pdsms[s][9]==16){printf("$IRWICLIP1\\P2/");}
else{printf("#IRWICLIP1\\P2/");}
if(s==0){printf(" ");}
}
printf("\n");
/**/
for(l=0;l<4;l++){l4=l*4;
for(s=0;s<2;s++){
printf(" ");
for(m=1;m<3;m++){
printf(" ");
for(n=1;n<5;n++){v=tlu[d[s][m+16]][l4+n];
printf("%d",v);}
}
printf(" ");
for(n=1;n<5;n++){printf("%3d",d[s][l4+n]);}
printf("|%2d|%2d|%2d\\%2d/",lnsm[s][l],lnsm[s][l+4],pdsms[s][l],pdsms[s][l+4]);
if(s==0){printf(" ");}
}
printf("\n");
}
}
}
/**/
/* Print 6 Answers */
void pr6sol(short x, short y, short z){
short t,l,l4,n,s,sz,z6;
z6=z*6;
for(t=(x-1);t<y;t=t+z6){
for(s=0;s<6;s++){sz=s*z;
if(t+sz>y){break;}
for(n=1;n<22;n++){d[s][n]=ta[t+sz][n];}
chksum(s);
printf("%3d/%2d/%4d",d[s][17]+1,d[s][18]+1,t+sz+1);
if(pdsms[s][8]==8){printf("P");}
else if(pdsms[s][9]==16){printf("$");}
else{printf("#");}
}
}
}
}

```



```

        if(s<5){printf(" ");}
    }
    printf("\n");
/**/
for(l=0;l<4;l++){l4=l*4;
    for(s=0;s<6;s++){sz=s*z;
        if(t+sz>y){break;}
        for(n=1;n<5;n++){printf("%3d",d[s][l4+n]);}
        if(s<5){printf(" ");}
    }
    printf("\n");
}
}
}
/**/

```

The next lists would show you the result of my recent calculations. We could have the 48 Latin Units and only the objective Solutions as many as 144.

**\*\* 'Euler Squares' of Order 4: Composed by the 'New Euler's Method' with the 4-th Increment Number System \*\***

**[List of Latin Units with Decompositions by 4-th Increment System]**

1/ 0 3 2 1	2/ 0 3 1 2	3/ 0 2 3 1	4/ 0 1 3 2	5/ 0 3 2 1	6/ 0 2 3 1	7/ 0 2 1 3	8/ 0 1 2 3
1 2 3 0	2 1 3 0	3 1 0 2	3 2 0 1	2 1 0 3	1 3 2 0	3 1 2 0	2 3 0 1
3 0 1 2	3 0 2 1	1 3 2 0	2 3 1 0	1 2 3 0	2 0 1 3	2 0 3 1	3 2 1 0
2 1 0 3	1 2 0 3	2 0 1 3	1 0 2 3	3 0 1 2	3 1 0 2	1 3 0 2	1 0 3 2
9/ 0 3 1 2	10/ 0 2 1 3	11/ 0 1 3 2	12/ 0 1 2 3	13/ 1 3 2 0	14/ 1 3 0 2	15/ 1 2 3 0	16/ 1 0 3 2
1 2 0 3	1 3 0 2	2 3 1 0	3 2 1 0	0 2 3 1	2 0 3 1	3 0 1 2	3 2 1 0
2 1 3 0	3 1 2 0	1 0 2 3	1 0 3 2	3 1 0 2	3 1 2 0	0 3 2 1	2 3 0 1
3 0 2 1	2 0 3 1	3 2 0 1	2 3 0 1	2 0 1 3	0 2 1 3	2 1 0 3	0 1 2 3
17/ 1 3 2 0	18/ 1 2 3 0	19/ 1 2 0 3	20/ 1 0 2 3	21/ 1 3 0 2	22/ 1 2 0 3	23/ 1 0 3 2	24/ 1 0 2 3
2 0 1 3	0 3 2 1	3 0 2 1	2 3 1 0	0 2 1 3	0 3 1 2	2 3 0 1	3 2 0 1
0 2 3 1	2 1 0 3	2 1 3 0	3 2 0 1	2 0 3 1	3 0 2 1	0 1 2 3	0 1 3 2
3 1 0 2	3 0 1 2	0 3 1 2	0 1 3 2	3 1 2 0	2 1 3 0	3 2 1 0	2 3 1 0
25/ 2 3 1 0	26/ 2 3 0 1	27/ 2 1 3 0	28/ 2 0 3 1	29/ 2 3 1 0	30/ 2 1 3 0	31/ 2 1 0 3	32/ 2 0 1 3
0 1 3 2	1 0 3 2	3 0 2 1	3 1 2 0	1 0 2 3	0 3 1 2	3 0 1 2	1 3 2 0
3 2 0 1	3 2 1 0	0 3 1 2	1 3 0 2	0 1 3 2	1 2 0 3	1 2 3 0	3 1 0 2
1 0 2 3	0 1 2 3	1 2 0 3	0 2 1 3	3 2 0 1	3 0 2 1	0 3 2 1	0 2 3 1
33/ 2 3 0 1	34/ 2 1 0 3	35/ 2 0 3 1	36/ 2 0 1 3	37/ 3 2 1 0	38/ 3 2 0 1	39/ 3 1 2 0	40/ 3 0 2 1
0 1 2 3	0 3 2 1	1 3 0 2	3 1 0 2	0 1 2 3	1 0 2 3	2 0 3 1	2 1 3 0
1 0 3 2	3 0 1 2	0 2 1 3	0 2 3 1	2 3 0 1	2 3 1 0	0 2 1 3	1 2 0 3
3 2 1 0	1 2 3 0	3 1 2 0	1 3 2 0	1 0 3 2	0 1 3 2	1 3 0 2	0 3 1 2
41/ 3 2 1 0	42/ 3 1 2 0	43/ 3 1 0 2	44/ 3 0 1 2	45/ 3 2 0 1	46/ 3 1 0 2	47/ 3 0 2 1	48/ 3 0 1 2
1 0 3 2	0 2 1 3	2 0 1 3	1 2 3 0	0 1 3 2	0 2 3 1	1 2 0 3	2 1 0 3
0 1 2 3	1 3 0 2	1 3 2 0	2 1 0 3	1 0 2 3	2 0 1 3	0 3 1 2	0 3 2 1
2 3 0 1	2 0 3 1	0 2 3 1	0 3 2 1	2 3 1 0	1 3 2 0	2 1 3 0	1 2 3 0

[Count of Latin Units = 48]

**[List of Standard Solutions(Sorted):**

Used Units//Sol\_N#(\$:Self-complementary; P:Pan-diagonal)|Sum-Checks|/]

1/ 3/	1\$ RW CL P1\P2/	2/ 3/	2\$ RW CL P1\P2/		
0321 0231 1 15 12 6 34 34 34\16/		0312 0231 1 15 8 10 34 34 34\24/			
1230 3102 8 10 13 3 34 34 48\34/		2130 3102 12 6 13 3 34 34 40\34/			
3012 1320 14 4 7 9 34 34 34\52/		3021 1320 14 4 11 5 34 34 34\44/			
2103 2013 11 5 2 16 34 34 20\34/		1203 2013 7 9 2 16 34 34 28\34/			
1/ 4/	3\$ RW CL P1\P2/	2/ 4/	4\$ RW CL P1\P2/		
0321 0132 1 14 12 7 34 34 34\14/		0312 0132 1 14 8 11 34 34 34\22/			
1230 3201 8 11 13 2 34 34 46\34/		2130 3201 12 7 13 2 34 34 38\34/			
3012 2310 15 4 6 9 34 34 34\54/		3021 2310 15 4 10 5 34 34 34\46/			
2103 1023 10 5 3 16 34 34 22\34/		1203 1023 6 9 3 16 34 34 30\34/			
1/ 5/	5\$ RW CL P1\P2/	2/ 5/	6\$ RW CL P1\P2/		
0321 0321 1 16 11 6 34 34 34\18/		0312 0321 1 16 7 10 34 34 34\26/			
1230 2103 7 10 13 4 34 34 50\34/		2130 2103 11 6 13 4 34 34 42\34/			
3012 1230 14 3 8 9 34 34 34\50/		3021 1230 14 3 12 5 34 34 34\42/			
2103 3012 12 5 2 15 34 34 18\34/		1203 3012 8 9 2 15 34 34 26\34/			
1/ 8/	7\$ RW CL P1\P2/	2/ 8/	8\$ RW CL P1\P2/		
0321 0123 1 14 11 8 34 34 34\14/		0312 0123 1 14 7 12 34 34 34\22/			
1230 2301 7 12 13 2 34 34 46\34/		2130 2301 11 8 13 2 34 34 38\34/			
3012 3210 16 3 6 9 34 34 34\54/		3021 3210 16 3 10 5 34 34 34\46/			
2103 1032 10 5 4 15 34 34 22\34/		1203 1032 6 9 4 15 34 34 30\34/			
1/ 9/	9\$ RW CL P1\P2/	2/ 9/	10\$ RW CL P1\P2/		
0321 0312 1 16 10 7 34 34 34\18/		0312 0312 1 16 6 11 34 34 34\26/			
1230 1203 6 11 13 4 34 34 50\34/		2130 1203 10 7 13 4 34 34 42\34/			
3012 2130 15 2 8 9 34 34 34\50/		3021 2130 15 2 12 5 34 34 34\42/			
2103 3021 12 5 3 14 34 34 18\34/		1203 3021 8 9 3 14 34 34 26\34/			
1/ 10/	11\$ RW CL P1\P2/	2/ 10/	12\$ RW CL P1\P2/		
0321 0213 1 15 10 8 34 34 34\16/		0312 0213 1 15 6 12 34 34 34\24/			
1230 1302 6 12 13 3 34 34 48\34/		2130 1302 10 8 13 3 34 34 40\34/			
3012 3120 16 2 7 9 34 34 34\52/		3021 3120 16 2 11 5 34 34 34\44/			
2103 2031 11 5 4 14 34 34 20\34/		1203 2031 7 9 4 14 34 34 28\34/			
5/ 1/	13\$ RW CL P1\P2/	5/ 2/	14\$ RW CL P1\P2/		
0321 0321 1 16 11 6 34 34 34\30/		0321 0312 1 16 10 7 34 34 34\32/			
2103 1230 10 7 4 13 34 34 38\34/		2103 2130 11 6 4 13 34 34 36\34/			
1230 3012 8 9 14 3 34 34 34\38/		1230 3021 8 9 15 2 34 34 34\36/			
3012 2103 15 2 5 12 34 34 30\34/		3012 1203 14 3 5 12 34 34 32\34/			
6/ 3/	15\$ RW CL P1\P2/	6/ 4/	16\$ RW CL P1\P2/		
0231 0231 1 11 16 6 34 34 34\16/		0231 0132 1 10 16 7 34 34 34\14/			
1320 3102 8 14 9 3 34 34 48\34/		1320 3201 8 15 9 2 34 34 46\34/			
2013 1320 10 4 7 13 34 34 34\52/		2013 2310 11 4 6 13 34 34 34\54/			
3102 2013 15 5 2 12 34 34 20\34/		3102 1023 14 5 3 12 34 34 22\34/			
5/ 6/	17\$ RW CL P1\P2/	5/ 7/	18P RW CL P1\P2/		
0321 0231 1 15 12 6 34 34 34\30/		0321 0213 1 15 10 8 34 34 34\34/			
2103 1320 10 8 3 13 34 34 38\34/		2103 3120 12 6 3 13 34 34 34\34/			
1230 2013 7 9 14 4 34 34 34\38/		1230 2031 7 9 16 2 34 34 34\34/			
3012 3102 16 2 5 11 34 34 30\34/		3012 1302 14 4 5 11 34 34 34\34/			
6/ 5/ 19#	6/ 8/ 20#	5/11/ 21#	5/12/ 22P	6/ 9/ 23#	6/10/ 24#
1 12 15 6	1 10 15 8	1 14 12 7	1 14 11 8	1 12 14 7	1 11 14 8
7 14 9 4	7 16 9 2	11 8 2 13	12 7 2 13	6 15 9 4	6 16 9 3
10 3 8 13	12 3 6 13	6 9 15 4	6 9 16 3	11 2 8 13	12 2 7 13
16 5 2 11	14 5 4 11	16 3 5 10	15 4 5 10	16 5 3 10	15 5 4 10
9/ 1/ 25#	9/ 2/ 26#	10/ 1/ 27#	10/ 2/ 28#	9/ 6/ 29#	9/ 7/ 30P
1 16 7 10	1 16 6 11	1 12 7 14	1 12 6 15	1 15 8 10	1 15 6 12
6 11 4 13	7 10 4 13	6 15 4 9	7 14 4 9	6 12 3 13	8 10 3 13
12 5 14 3	12 5 15 2	16 5 10 3	16 5 11 2	11 5 14 4	11 5 16 2
15 2 9 8	14 3 9 8	11 2 13 8	10 3 13 8	16 2 9 7	14 4 9 7

10/ 6/ 31#	10/ 7/ 32#	9/11/ 33#	9/12/ 34P	10/11/ 35#	10/12/ 36#
1 11 8 14	1 11 6 16	1 14 8 11	1 14 7 12	1 10 8 15	1 10 7 16
6 16 3 9	8 14 3 9	7 12 2 13	8 11 2 13	7 16 2 9	8 15 2 9
15 5 10 4	15 5 12 2	10 5 15 4	10 5 16 3	14 5 11 4	14 5 12 3
12 2 13 7	10 4 13 7	16 3 9 6	15 4 9 6	12 3 13 6	11 4 13 6
1/15/ 37#	1/16/ 39#	1/17/ 41\$	1/20/ 43\$	1/21/ 45#	1/22/ 47#
2 15 12 5	2 13 12 7	2 16 11 5	2 13 11 8	2 16 9 7	2 15 9 8
8 9 14 3	8 11 14 1	7 9 14 4	7 12 14 1	5 11 14 4	5 12 14 3
13 4 7 10	15 4 5 10	13 3 8 10	16 3 5 10	15 1 8 10	16 1 7 10
11 6 1 16	9 6 3 16	12 6 1 15	9 6 4 15	12 6 3 13	11 6 4 13
5/13/ 49#	6/15/ 51#	5/18/ 53#	6/17/ 55#	5/23/ 57P	6/21/ 59#
2 16 11 5	2 11 16 5	2 15 12 5	2 12 15 5	2 13 12 7	2 12 13 7
9 7 4 14	8 13 10 3	9 8 3 14	7 13 10 4	11 8 1 14	5 15 10 4
8 10 13 3	9 4 7 14	7 10 13 4	9 3 8 14	5 10 15 4	11 1 8 14
15 1 6 12	15 6 1 12	16 1 6 11	16 6 1 11	16 3 6 9	16 6 3 9
9/13/ 61#	10/13/ 63#	9/18/ 65#	10/18/ 67#	9/23/ 69P	10/23/ 71#
2 16 7 9	2 12 7 13	2 15 8 9	2 11 8 13	2 13 8 11	2 9 8 15
5 11 4 14	5 15 4 10	5 12 3 14	5 16 3 10	7 12 1 14	7 16 1 10
12 6 13 3	16 6 9 3	11 6 13 4	15 6 9 4	9 6 15 4	13 6 11 4
15 1 10 8	11 1 14 8	16 1 10 7	12 1 14 7	16 3 10 5	12 3 14 5
1/27/ 73#	1/29/ 77\$	1/33/ 81#	5/25/ 85#	5/30/ 89#	5/35/ 93P
3 14 12 5	3 16 10 5	3 16 9 6	3 16 10 5	3 14 12 5	3 13 12 6
8 9 15 2	6 9 15 4	5 10 15 4	9 6 4 15	9 8 2 15	10 8 1 15
13 4 6 11	13 2 8 11	14 1 8 11	8 11 13 2	6 11 13 4	5 11 14 4
10 7 1 16	12 7 1 14	12 7 2 13	14 1 7 12	16 1 7 10	16 2 7 9
9/25/ 97#	9/30/ 101#	9/35/ 105P	1/39/ 109#	1/41/ 113#	1/45/ 117\$
3 16 6 9	3 14 8 9	3 13 8 10	4 14 11 5	4 15 10 5	4 15 9 6
5 10 4 15	5 12 2 15	6 12 1 15	7 9 16 2	6 9 16 3	5 10 16 3
12 7 13 2	10 7 13 4	9 7 14 4	13 3 6 12	13 2 7 12	14 1 7 12
14 1 11 8	16 1 11 6	16 2 11 5	10 8 1 15	11 8 1 14	11 8 2 13
5/37/ 121P	5/42/ 125P	5/47/ 129#	9/37/ 133P	9/42/ 137P	9/47/ 141#
4 15 10 5	4 14 11 5	4 13 11 6	4 15 6 9	4 14 7 9	4 13 7 10
9 6 3 16	9 7 2 16	10 7 1 16	5 10 3 16	5 11 2 16	6 11 1 16
7 12 13 2	6 12 13 3	5 12 14 3	11 8 13 2	10 8 13 3	9 8 14 3
14 1 8 11	15 1 8 10	15 2 8 9	14 1 12 7	15 1 12 6	15 2 12 5

. . . . .

[Solution Counts = 144]

[OK!]

\*\* Calculated and Listed by Kanji Setsuda on Mar.14, 2011 with MacOSX & Xcode 3 \*\*

Only 144 'Euler Squares' of Order 4 could exist among the 880 standard solutions of our objects. So many or so few, how do you think? What does this mean?

It surely means those Euler Squares make the selected, precious solution set, but they could neither explain the total concept of Standard MS44 nor reveal the secret structure behind them. The ambitious dream of 'Euler School' has been broken.

The objective 'structures' of our 880 solutions might be presented as follows.

\*\* 'Standard' Magic Squares of Order 4 made by our Ordinary Method: \*\*

[Standard Solutions with Decompositions by 4-th Increment System]

[ 1 ]	D4i: H/ L/	[ 2 ]	D4i: H/ L/	[ 3 ]	D4i: H/ L/
1 15 14 4	0330 0213	1 15 14 4	0330 0213	1 15 12 6	0321 0231
12 6 7 9	2112 3120	8 10 11 5	1221 3120	14 4 7 9	3012 1320
8 10 11 5	1221 3120	12 6 7 9	2112 3120	8 10 13 3	1230 3102
13 3 2 16	3003 0213	13 3 2 16	3003 0213	11 5 2 16	2103 2013

[ 4]	D4i: H/ L/	[ 5]	D4i: H/ L/	[ 6]	D4i: H/ L/
1 15 12 6 0321 0231		1 15 8 10 0312 0231		1 15 8 10 0312 0231	
8 10 13 3 1230 3102		14 4 11 5 3021 1320		12 6 13 3 2130 3102	
14 4 7 9 3012 1320		12 6 13 3 2130 3102		14 4 11 5 3021 1320	
11 5 2 16 2103 2013		7 9 2 16 1203 2013		7 9 2 16 1203 2013	
[ 7]	D4i: H/ L/	[ 8]	D4i: H/ L/	[ 9]	D4i: H/ L/
1 14 15 4 0330 0123		1 14 15 4 0330 0123		1 14 12 7 0321 0132	
12 7 6 9 2112 3210		8 11 10 5 1221 3210		8 11 13 2 1230 3201	
8 11 10 5 1221 3210		12 7 6 9 2112 3210		15 4 6 9 3012 2310	
13 2 3 16 3003 0123		13 2 3 16 3003 0123		10 5 3 16 2103 1023	
[ 10]	D4i: H/ L/	[ 11]	D4i: H/ L/	[ 12]	D4i: H/ L/
1 14 8 11 0312 0132		1 12 15 6 0231 0321		1 12 14 7 0231 0312	
12 7 13 2 2130 3201		8 13 10 3 1320 3012		8 13 11 2 1320 3021	
15 4 10 5 3021 2310		14 7 4 9 3102 1230		15 6 4 9 3102 2130	
6 9 3 16 1203 1023		11 2 5 16 2013 2103		10 3 5 16 2013 1203	
[ 13]	D4i: H/ L/	[ 29]	D4i: H/ L/	[ 45]	D4i: H/ L/
1 16 13 4 0330 0303		1 16 13 4 0330 0303		1 16 5 12 0312 0303	
11 6 7 10 2112 2121		12 9 8 5 2211 3030		15 6 11 2 3120 2121	
8 9 12 5 1221 3030		6 7 10 11 1122 1212		10 3 14 7 2031 1212	
14 3 2 15 3003 1212		15 2 3 14 3003 2121		8 9 4 13 1203 3030	
[ 71]	D4i: H/ L/	[ 83]	D4i: H/ L/	[111]	D4i: H/ L/
1 16 11 6 0321 0321		1 16 9 8 0321 0303		1 16 9 8 0321 0303	
10 7 4 13 2103 1230		15 10 7 2 3210 2121		14 11 6 3 3210 1212	
8 9 14 3 1230 3012		4 5 12 13 0123 3030		4 5 12 13 0123 3030	
15 2 5 12 3012 2103		14 3 6 11 3012 1212		15 2 7 10 3012 2121	
[129]	D4i: H/ L/	[145]	D4i: H/ L/	[157]	D4i: H/ L/
1 15 10 8 0321 0213		1 16 7 10 0312 0321		1 16 3 14 0303 0321	
14 11 4 5 3201 1230		6 11 4 13 1203 1230		13 11 6 4 3210 0213	
3 6 13 12 0132 2103		12 5 14 3 2130 3012		8 2 15 9 1032 3120	
16 2 7 9 3012 3120		15 2 9 8 3021 2103		12 5 10 7 2121 3012	
[173]	D4i: H/ L/	[189]	D4i: H/ L/	[197]	D4i: H/ L/
1 16 5 12 0312 0303		1 14 7 12 0312 0123		1 12 8 13 0213 0330	
10 13 4 7 2301 1032		11 13 2 8 2301 2013		11 14 2 7 2301 2112	
8 3 14 9 1032 3210		6 4 15 9 1032 1320		6 3 15 10 1032 1221	
15 2 11 6 3021 2121		16 3 10 5 3021 3210		16 5 9 4 3120 3003	
[209]	D4i: H/ L/	[225]	D4i: H/ L/	[237]	D4i: H/ L/
2 15 14 3 0330 1212		2 16 13 3 0330 1302		2 16 13 3 0330 1302	
12 5 8 9 2112 3030		11 5 8 10 2112 2031		12 7 6 9 2112 3210	
7 10 11 6 1221 2121		7 9 12 6 1221 2031		5 10 11 8 1221 0123	
13 4 1 16 3003 0303		14 4 1 15 3003 1302		15 1 4 14 3003 2031	
[253]	D4i: H/ L/	[269]	D4i: H/ L/	[291]	D4i: H/ L/
2 16 9 7 0321 1302		2 16 11 5 0321 1320		2 16 9 7 0321 1302	
5 11 14 4 1230 0213		14 7 4 9 3102 1230		3 13 12 6 0321 2031	
15 1 8 10 3012 2031		3 10 13 8 0231 2103		15 1 8 10 3012 2031	
12 6 3 13 2103 3120		15 1 6 12 3012 2013		14 4 5 11 3012 1302	
[303]	D4i: H/ L/	[319]	D4i: H/ L/	[337]	D4i: H/ L/
2 16 9 7 0321 1302		2 15 10 7 0321 1212		2 16 7 9 0312 1320	
5 14 11 4 1320 0123		13 12 5 4 3210 0303		14 11 4 5 3201 1230	
12 1 8 13 2013 3030		3 6 11 14 0123 2121		3 6 13 12 0132 2103	
15 3 6 10 3012 2211		16 1 8 9 3012 3030		15 1 10 8 3021 2013	
[353]	D4i: H/ L/	[365]	D4i: H/ L/	[373]	D4i: H/ L/
2 16 5 11 0312 1302		2 16 5 11 0312 1302		2 15 6 11 0312 1212	
3 13 8 10 0312 2031		9 14 7 4 2310 0123		9 14 3 8 2301 0123	
15 1 12 6 3021 2031		8 1 12 13 1023 3030		7 4 13 10 1032 2301	
14 4 9 7 3021 1302		15 3 10 6 3021 2211		16 1 12 5 3021 3030	

```

[389]      D4i: H/  L/  [397]      D4i: H/  L/  [409]      D4i: H/  L/
  2 14  3 15 0303 1122   2 12  7 13 0213 1320   3 15 10  6 0321 2211
 11 12  5  6 2211 2301   5 15  4 10 1302 0231   14  2  7 11 3012 1122
  8  1 16  9 1032 3030   11  1 14  8 2031 2013   8 12 13  1 1230 3300
 13  7 10  4 3120 0213   16  6  9  3 3120 3102   9  5  4 16 2103 0033

[575]      D4i: H/  L/  [753]      D4i: H/  L/  [817]      D4i: H/  L/
  4 15 10  5 0321 3210   5 16  4  9 1302 0330   6 16  3  9 1302 1320
 14  1  8 11 3012 1032   11  2 14  7 2031 2112   11  1 14  8 2031 2013
  7 12 13  2 1230 2301   10  3 15  6 2031 1221   10  4 15  5 2031 1320
  9  6  3 16 2103 0123   8 13  1 12 1302 3003   7 13  2 12 1302 2013

[865]      D4i: H/  L/
  7 16  2  9 1302 2310
 11  4 14  5 2031 2310
  6  1 15 12 1032 1023
 10 13  3  8 2301 1023   . . . . . [Solution Counts = 880]

```

What consistent laws can we find among them? I have no idea about that now.

### 3. How about making those Standard MS44 by the Binary Number System?

Why don't you take the Binary Number System? Let's make another experiments with it here. Can we make any Euler Squares 4x4 for the Standard Magic Squares?

Diagram and Basic Conditions I must prepare are almost the same as before.

**\*\* Basic Forms \*\***

```

  2  3  4 | 1 | 2 | 3 | 4 | 1  2  3      | 1 |15|14| 4 |
  6  7  8 | 5 | 6 | 7 | 8 | 5  6  7      |12| 6 | 7 | 9 |
 10 11 12 | 9 |10|11|12 | 9 10 11      | 8 |10|11| 5 |
 14 15 16 |13|14|15|16 |13 14 15      |13| 3 | 2 |16 |

```

**\*\* Basic Conditions for the Latin Units of Standard MS44: C=2 \*\***

```

n1+n2+n3+n4=C   ...bc1; | n1+n5+n9+n13=C   ...bc5;
n5+n6+n7+n8=C   ...bc2; | n2+n6+n10+n14=C  ...bc6;
n9+n10+n11+n12=C ...bc3; | n3+n7+n11+n15=C  ...bc7;
n13+n14+n15+n16=C ...bc4; | n4+n8+n12+n16=C  ...bc8;
n1+n6+n11+n16=C   ...bc9;
n4+n7+n10+n13=C   ...bc10;

```

**\*\* List-forming Inequality Conditions: \*\***

$n1 < n16$ ,  $n1 < n4$ ,  $n1 < n13$ , and  $n2 > n5$ ;

Each of the 4 Latin Units should have such the same property in common as every row, every column and two primary diagonals must take the Latin Pattern: (0,0,1,1), (0,1,0,1), (0,1,1,0), (1,0,0,1), (1,0,1,0) or (1,1,0,0).

```

  1/   2/   3/   4/           S/ | ((0*2+0)*2+0)*2+0+1 = 1/S
0110 0110 0101 0011  1 15 14  4 | ((1*2+1)*2+1)*2+0+1 = 15/S
1001 0110 1010 1100  12  6  7  9 | ((1*2+1)*2+0)*2+1+1 = 14/S
0110 1001 1010 1100   8 10 11  5 | ((0*2+0)*2+1)*2+1+1 = 4/S
1001 1001 0101 0011  13  3  2 16 | ((1*2+0)*2+1)*2+1+1 = 12/S
... ((a*2+b)*2+c)*2+d+1 = N/S ... cmp

```

After we have got all the Latin Units by the "New Euler's Method", we pick up any four different units, combine them and compose our objective MS44 as shown above.

Let me show you the 'C' program I have dictated for that purpose as follows.

```

/** 'Standard' Magic Squares of Order 4: Composing 'Complete Euler Squares' **/
/** of Order 4 by the "New Euler's Method" with Binary Number System **/
/** 'CES44Std.c': Dictated by Kanji Setsuda on Oct.22,'04; Sep.15,'06; **/
/** Aug.10,'10; and Mar.18, 2011; with MacOSX 10.5.8 and Xcode 3.1.2 **/
/**/
#include <stdio.h>
/**/
short lcnt, cnt;
short LSM;
short nm[17], uflg[17];
short tlu[17][17];
short mtc[17][17];
short ta[530][21];
short d[7][21];
short lnsm[2][9], pdsm[2][10];
/**/
short bcf[11][2];
short u1, u2, u3, u4;
/**/
void sstp01(void), sstp02(void), sstp03(void), sstp04(void);
void sstp05(void), sstp06(void), sstp07(void), sstp08(void);
void sstp09(void), sstp10(void), sstp11(void), sstp12(void);
void sstp13(void), sstp14(void), sstp15(void), sstp16(void);
void slurecord(void);
void prlunit(void);
void cmbcmp(void);
void recordsol(void);
void sortsol(void), exc(short x);
void chksum(short x);
void pr2sol(short x);
void pr6sol(short x, short y, short z);
/**/
int main(){
short m,n;
printf("\n** Making 'Complete Euler Squares' of Order 4\n");
printf(" by the 'New Euler's Method' with Binary Number System **\n");
for(n=0;n<17;n++){nm[n]=0;}
for(m=0;m<11;m++){for(n=0;n<2;n++){bcf[m][n]=0;}}
lcnt=0;
sstp01(); // Make Latin Units
printf("\n [List of Latin Units with Decompositions by Binary System]\n");
prlunit(); // Print Latin Units
cnt=0; LSM=34;
cmbcmp(); // Combine and Compose
sortsol(); // Sort the Solutions
printf("\n ** 'Standard' Magic Squares of Order 4: Composing **\n");
printf(" ** the 'Complete Euler Squares' of Order 4 by the **\n");
printf(" ** 'New Euler's Method' with Binary Number System **\n");
printf("\n [List of Standard Solutions(Sorted):\n");
printf(" Used Units//Sol_N#($:Self-complementary; P:Pan-diagonal)|Sum-Checks|/]\n");
pr2sol(24);
pr6sol(25,36,1);
pr6sol(38,60,2);
pr6sol(64,132,4);
pr6sol(133,271,8);
pr6sol(277,cnt,16);
printf(" . . . . .\n");
printf(" [Solution Counts = %d]\n",cnt);
printf(" [OK!]\n");
printf("** Calculated and Listed by Kanji Setsuda on Mar.18, 2011 with MAC Xcode 3 **\n");
return 0;
}
/**/
/** Program Modules for the Latin Units of 'Complete Euler Squares' **/

```

```

/** of Order 4 for the 'Standard' type of Magic Squares of Order 4 */
/** Dictated by Kanji Setsuda on Mar.10, '11 with MacOSX & Xcode 3 */
/**/
/* Level 1: */
/* Set n1 */
void sstp01(){
  short a;
  for(a=0;a<2;a++){
    if((bcf[1][a]<2)&&(bcf[5][a]<2)&&(bcf[9][a]<2)){
      bcf[1][a]++; bcf[5][a]++; bcf[9][a]++;
      nm[1]=a; sstp02();
      bcf[9][a]--; bcf[5][a]--; bcf[1][a]--;
    }
  }
}
/**/
/* Set n16 */
void sstp02(){
  short a;
  for(a=1;a>=0;a--){
    if((bcf[4][a]<2)&&(bcf[8][a]<2)&&(bcf[9][a]<2)){
      bcf[4][a]++; bcf[8][a]++; bcf[9][a]++;
      nm[16]=a; sstp03();
      bcf[9][a]--; bcf[8][a]--; bcf[4][a]--;
    }
  }
}
/**/
/* Set n2 */
void sstp03(){
  short a;
  for(a=1;a>=0;a--){
    if((bcf[1][a]<2)&&(bcf[6][a]<2)){
      bcf[1][a]++; bcf[6][a]++;
      nm[2]=a; sstp04();
      bcf[6][a]--; bcf[1][a]--;
    }
  }
}
/**/
/* Set n15 */
void sstp04(){
  short a;
  for(a=0;a<2;a++){
    if((bcf[4][a]<2)&&(bcf[7][a]<2)){
      bcf[4][a]++; bcf[7][a]++;
      nm[15]=a; sstp05();
      bcf[7][a]--; bcf[4][a]--;
    }
  }
}
/**/
/* Set n3 */
void sstp05(){
  short a;
  for(a=1;a>=0;a--){
    if((bcf[1][a]<2)&&(bcf[7][a]<2)){
      bcf[1][a]++; bcf[7][a]++;
      nm[3]=a; sstp06();
      bcf[7][a]--; bcf[1][a]--;
    }
  }
}
/**/

```

```

/* Set n14 */
void sstp06(){
  short a;
  for(a=0;a<2;a++){
    if((bcf[4][a]<2)&&(bcf[6][a]<2)){
      bcf[4][a]++; bcf[6][a]++;
      nm[14]=a; sstp07();
      bcf[6][a]--; bcf[4][a]--;
    }
  }
}
/**/
/* Set n4 */
void sstp07(){
  short a;
  for(a=0;a<2;a++){
    if((bcf[1][a]<2)&&(bcf[8][a]<2)&&(bcf[10][a]<2)){
      bcf[1][a]++; bcf[8][a]++; bcf[10][a]++;
      nm[4]=a; sstp08();
      bcf[10][a]--; bcf[8][a]--; bcf[1][a]--;
    }
  }
}
/**/
/* Set n13 */
void sstp08(){
  short a;
  for(a=1;a>=0;a--){
    if((bcf[4][a]<2)&&(bcf[5][a]<2)&&(bcf[10][a]<2)){
      bcf[4][a]++; bcf[5][a]++; bcf[10][a]++;
      nm[13]=a; sstp09();
      bcf[10][a]--; bcf[5][a]--; bcf[4][a]--;
    }
  }
}
/**/
/* Level 2: */
/* Set n5 */
void sstp09(){
  short a;
  for(a=1;a>=0;a--){
    if((bcf[2][a]<2)&&(bcf[5][a]<2)){
      bcf[2][a]++; bcf[5][a]++;
      nm[5]=a; sstp10();
      bcf[5][a]--; bcf[2][a]--;
    }
  }
}
/**/
/* Set n12 */
void sstp10(){
  short a;
  for(a=0;a<2;a++){
    if((bcf[3][a]<2)&&(bcf[8][a]<2)){
      bcf[3][a]++; bcf[8][a]++;
      nm[12]=a; sstp11();
      bcf[8][a]--; bcf[3][a]--;
    }
  }
}
/**/
/* Set n6 */
void sstp11(){
  short a;

```



```

for(a=0;a<2;a++){
    if((bcf[2][a]<2)&&(bcf[6][a]<2)&&(bcf[9][a]<2)){
        bcf[2][a]++; bcf[6][a]++; bcf[9][a]++;
        nm[6]=a; sstp12();
        bcf[9][a]--; bcf[6][a]--; bcf[2][a]--;
    }
}
}
/**/
/* Set n11 */
void sstp12(){
    short a;
    for(a=1;a>=0;a--){
        if((bcf[3][a]<2)&&(bcf[7][a]<2)&&(bcf[9][a]<2)){
            bcf[3][a]++; bcf[7][a]++; bcf[9][a]++;
            nm[11]=a; sstp13();
            bcf[9][a]--; bcf[7][a]--; bcf[3][a]--;
        }
    }
}
/**/
/* Set n7 */
void sstp13(){
    short a;
    for(a=0;a<2;a++){
        if((bcf[2][a]<2)&&(bcf[7][a]<2)&&(bcf[10][a]<2)){
            bcf[2][a]++; bcf[7][a]++; bcf[10][a]++;
            nm[7]=a; sstp14();
            bcf[10][a]--; bcf[7][a]--; bcf[2][a]--;
        }
    }
}
/**/
/* Set n10 */
void sstp14(){
    short a;
    for(a=1;a>=0;a--){
        if((bcf[3][a]<2)&&(bcf[6][a]<2)&&(bcf[10][a]<2)){
            bcf[3][a]++; bcf[6][a]++; bcf[10][a]++;
            nm[10]=a; sstp15();
            bcf[10][a]--; bcf[6][a]--; bcf[3][a]--;
        }
    }
}
/**/
/* Set n8 */
void sstp15(){
    short a;
    for(a=1;a>=0;a--){
        if((bcf[2][a]<2)&&(bcf[8][a]<2)){
            bcf[2][a]++; bcf[8][a]++;
            nm[8]=a; sstp16();
            bcf[8][a]--; bcf[2][a]--;
        }
    }
}
/**/
/* Set n9 */
void sstp16(){
    short a;
    for(a=0;a<2;a++){
        if((bcf[3][a]<2)&&(bcf[5][a]<2)){
            bcf[3][a]++; bcf[5][a]++;
            nm[9]=a; slurecord();
        }
    }
}

```

```

        bcf[5][a]--; bcf[3][a]--;
    }
}
}
/**/
/* Record the Answers */
void slurecord(){
    short n;
    tlu[lcnt][0]=lcnt+1;
    for(n=1;n<17;n++){tlu[lcnt][n]=nm[n];}
    lcnt++;
}
/**/
/* Make Matching Table and Print the Latin Units */
void prlunit(){
    short t,l,l4,m,n;
    for(m=0;m<lcnt;m++){
        for(n=0;n<lcnt;n++){
            t=0;
            for(l=1;l<17;l++){if(tlu[m][l]==tlu[n][l]){t++;}}
            mtc[m][n]=t;
        }
    }
    for(t=0;t<lcnt;t=t+8){
        printf("%8d/%9d/%9d/%9d/%9d/%9d/%9d/%9d/\n", t+1,t+2,t+3,t+4,t+5,t+6,t+7,t+8);
        for(l=0;l<4;l++){l4=l*4;
            for(m=t;m<(t+8);m++){
                printf(" ");
                for(n=1;n<5;n++){printf("%2d",tlu[m][l4+n]);}
                if(m<(t+7)){printf(" ");}
            }
            printf("\n");
        }
    }
    printf(" [Count of Latin Units = %d]\n",lcnt);
    printf("\n[Reference Table for the Best Combination]\n");
    printf(" *|");
    for(n=0;n<lcnt;n++){printf("%3d",n+1);}
    printf("\n");
    printf(" ---");
    for(m=0;m<lcnt;m++){printf("----");}
    printf("\n");
    for(m=0;m<lcnt;m++){
        printf("%3d|",m+1);
        for(n=0;n<lcnt;n++){t=mtc[m][n];
            if(t>=0){printf("%3d",t);}else{printf(" -");}}
        printf("\n");
    }
}
/**/
/* Combine and Compose */
void cmbcmp(){
    short lu,luh,md,n,d,fc;
    lu=lcnt; luh=lu/2; md=8;
    for(u1=0;u1<luh;u1++){
        for(u2=0;u2<lu;u2++){
            if(mtc[u2][u1]==md){
                for(u3=0;u3<lu;u3++){
                    if((mtc[u3][u1]==md)&&(mtc[u3][u2]==md)){
                        for(u4=0;u4<lu;u4++){
                            if((mtc[u4][u1]==md)&&(mtc[u4][u2]==md)&&(mtc[u4][u3]==md)){
                                for(n=1;n<17;n++){uflg[n]=0;}
                                for(n=1;n<17;n++){
                                    d=tlu[u1][n]*8+tlu[u2][n]*4+tlu[u3][n]*2+tlu[u4][n]+1;

```

```

        nm[n]=d; uflg[d]++;
    }
    fc=0;
    for(n=1;n<17;n++){if(uflg[n]==1){fc++;}else{break;}}
    if(fc==16){
        if((nm[1]<nm[16])&&(nm[1]<nm[4])&&(nm[1]<nm[13])&&(nm[2]>nm[5])){
            recordsol();}
    }
    }}}}
}
}
/**/
/* Record the Answers to the Table */
void recordsol(){
    short n;
    ta[cnt][0]=cnt+1;
    for(n=1;n<17;n++){ta[cnt][n]=nm[n];}
    ta[cnt][17]=u1; ta[cnt][18]=u2;
    ta[cnt][19]=u3; ta[cnt][20]=u4;
    cnt++;
}
/**/
/* Solution Sorting for the Smart List */
void sortsol(){
    short mx,m,f;
    short d1,d2,d3,d4,d5,d6;
    mx=cnt-1;
    do{f=0;
        for(m=0;m<mx;m++){d1=ta[m][1]; d2=ta[m+1][1];
            if(d1>d2){exc(m); f=1;}
            else{
                d3=(ta[m][16]*17+ta[m][2])*17+ta[m][3];
                d4=(ta[m+1][16]*17+ta[m+1][2])*17+ta[m+1][3];
                if((d1==d2)&&(d3<d4)){exc(m); f=1;}
                else{
                    d5=(ta[m][15]*17+ta[m][14])*17+ta[m][4];
                    d6=(ta[m+1][15]*17+ta[m+1][14])*17+ta[m+1][4];
                    if((d1==d2)&&(d3==d4)&&(d5>d6)){exc(m); f=1;}
                }
            }
        }
        mx--;
    }while(f>0);
}
/**/
void exc(short x){
    short n,d;
    for(n=1;n<21;n++){d=ta[x][n]; ta[x][n]=ta[x+1][n]; ta[x+1][n]=d;}
}
/**/
/* Check every sums of rows and columns */
void chksum(short x){
    short m,m4,n,s;
    for(m=0;m<4;m++){m4=m*4; s=0;
        for(n=1;n<5;n++){s=s+d[x][m4+n];}
        lnsm[x][m]=s; s=0;
        for(n=0;n<4;n++){s=s+d[x][n*4+m+1];}
        lnsm[x][m+4]=s;
    }
    pdsm[x][0]=d[x][1]+d[x][6]+d[x][11]+d[x][16];
    pdsm[x][1]=d[x][2]+d[x][7]+d[x][12]+d[x][13];
    pdsm[x][2]=d[x][3]+d[x][8]+d[x][9]+d[x][14];
    pdsm[x][3]=d[x][4]+d[x][5]+d[x][10]+d[x][15];
    pdsm[x][4]=d[x][1]+d[x][8]+d[x][11]+d[x][14];
    pdsm[x][5]=d[x][2]+d[x][5]+d[x][12]+d[x][15];
}

```

```

pdsm[x][6]=d[x][3]+d[x][6]+d[x][9]+d[x][16];
pdsm[x][7]=d[x][4]+d[x][7]+d[x][10]+d[x][13];
s=0;
for(n=0;n<8;n++){if(pdsm[x][n]==LSM){s++;}else{break;}}
pdsm[x][8]=s;
s=0;
for(n=1;n<17;n++){if(d[x][n]+d[x][17-n]==17){s++;}else{break;}}
pdsm[x][9]=s;
}
/**/
/* Print 2 Answers */
void pr2sol(short x){
short t,l,l4,m,n,s,v;
for(t=0;t<x;t=t+2){
for(s=0;s<2;s++){
for(n=1;n<22;n++){d[s][n]=ta[t+s][n];}
chksum(s);
printf("%4d/%4d/%4d/%4d/%12d",d[s][17]+1,d[s][18]+1,d[s][19]+1,d[s][20]+1,t+s+1);
if(pdsm[s][8]==8){printf("PIRWICLIP1\\P2/");}
else if(pdsm[s][9]==16){printf("$IRWICLIP1\\P2/");}
else{printf("#IRWICLIP1\\P2/");}
if(s==0){printf(" ");}
}
printf("\n");
/**/
for(l=0;l<4;l++){l4=l*4;
for(s=0;s<2;s++){
for(m=1;m<5;m++){
printf(" ");
for(n=1;n<5;n++){v=tlu[d[s][m+16]][l4+n];
printf("%d",v);}
}
printf(" ");
for(n=1;n<5;n++){printf("%3d",d[s][l4+n]);}
printf("|%2d|%2d|%2d\\%2d/",lnsm[s][l],lnsm[s][l+4],pdsm[s][l],pdsm[s][l+4]);
if(s==0){printf(" ");}
}
printf("\n");
}
}
}
/**/
/* Print 6 Answers */
void pr6sol(short x, short y, short z){
short t,l,l4,n,s,sz,z6;
z6=z*6;
for(t=(x-1);t<y;t=t+z6){
for(s=0;s<6;s++){sz=s*z;
if(t+sz>y){break;}
for(n=1;n<22;n++){d[s][n]=ta[t+sz][n];}
chksum(s);
if(y<133){printf("%2d/%d/%d/%d/",d[s][17]+1,d[s][18]+1,d[s][19]+1,d[s][20]+1);
printf("%3d",t+sz+1);}
else{printf("%12d",t+sz+1);}
if(pdsm[s][8]==8){printf("P");}
else if(pdsm[s][9]==16){printf("$");}
else{printf("#");}
if(s<5){printf(" ");}
}
printf("\n");
/**/
for(l=0;l<4;l++){l4=l*4;
for(s=0;s<6;s++){sz=s*z;
if(t+sz>y){break;}

```

```

printf(" ");
for(n=1;n<5;n++){printf("%3d",d[s][14+n]);}
if(s<5){printf(" ");}
}
printf("\n");
}
}
}
}
/**/

```

I would like to show you the result of my recent calculation as follows.

**\*\* Making the 'Euler Squares' of Order 4  
by the 'New Euler's Method' with Binary Number System \*\***

**[List 1 of Latin Units with Decompositions by Binary System]**

1/	2/	3/	4/	5/	6/	7/	8/
0 1 1 0	0 1 1 0	0 1 0 1	0 0 1 1	0 1 0 1	0 1 0 1	0 0 1 1	0 0 1 1
1 0 0 1	0 1 1 0	1 0 1 0	1 1 0 0	1 1 0 0	0 1 0 1	1 1 0 0	0 1 0 1
0 1 1 0	1 0 0 1	1 0 1 0	1 1 0 0	0 0 1 1	1 0 1 0	0 0 1 1	1 0 1 0
1 0 0 1	1 0 0 1	0 1 0 1	0 0 1 1	1 0 1 0	1 0 1 0	1 1 0 0	1 1 0 0
9/	10/	11/	12/	13/	14/	15/	16/
1 1 0 0	1 1 0 0	1 0 1 0	1 0 1 0	1 1 0 0	1 0 1 0	1 0 0 1	1 0 0 1
1 0 1 0	0 0 1 1	1 0 1 0	0 0 1 1	0 0 1 1	0 1 0 1	1 0 0 1	0 1 1 0
0 1 0 1	1 1 0 0	0 1 0 1	1 1 0 0	0 0 1 1	0 1 0 1	0 1 1 0	1 0 0 1
0 0 1 1	0 0 1 1	0 1 0 1	0 1 0 1	1 1 0 0	1 0 1 0	0 1 1 0	0 1 1 0

[Count of Latin Units = 16]

**[Reference Table 1 for the Best Combinations]**

*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
-----																
1	16	8	8	8	8	8	8	8	8	8	8	8	8	8	8	0
2	8	16	8	8	8	8	8	8	8	8	8	8	8	8	0	8
3	8	8	16	8	8	8	8	8	8	8	8	8	8	8	0	8
4	8	8	8	16	8	8	8	8	8	8	8	8	0	8	8	8
5	8	8	8	8	16	12	12	8	8	4	4	0	8	8	8	8
6	8	8	8	8	12	16	8	12	4	8	0	4	8	8	8	8
7	8	8	8	8	12	8	16	12	4	0	8	4	8	8	8	8
8	8	8	8	8	8	12	12	16	0	4	4	8	8	8	8	8
9	8	8	8	8	8	4	4	0	16	12	12	8	8	8	8	8
10	8	8	8	8	4	8	0	4	12	16	8	12	8	8	8	8
11	8	8	8	8	4	0	8	4	12	8	16	12	8	8	8	8
12	8	8	8	8	0	4	4	8	8	12	12	16	8	8	8	8
13	8	8	8	0	8	8	8	8	8	8	8	8	16	8	8	8
14	8	8	0	8	8	8	8	8	8	8	8	8	8	8	16	8
15	8	0	8	8	8	8	8	8	8	8	8	8	8	8	8	16
16	0	8	8	8	8	8	8	8	8	8	8	8	8	8	8	16

**\*\* 'Standard' Magic Squares of Order 4: Composing the 'Euler Squares' \*\*  
\*\* of Order 4 by the 'New Euler's Method' with Binary Number System \*\***

**[List of Standard Solutions(Sorted):**

**Used Units//Sol\_N#(\$:Self-complementary; P:Pan-diagonal)|Sum-Checks|/]**

1/	2/	3/	4/	1\$ RW CL P1\P2/	2/	1/	3/	4/	2\$ RW CL P1\P2/
0110	0110	0101	0011	1 15 14 4 34 34 34\24/	0110	0110	0101	0011	1 15 14 4 34 34 34\16/
1001	0110	1010	1100	12 6 7 9 34 34 40\34/	0110	1001	1010	1100	8 10 11 5 34 34 48\34/
0110	1001	1010	1100	8 10 11 5 34 34 34\44/	1001	0110	1010	1100	12 6 7 9 34 34 34\52/
1001	1001	0101	0011	13 3 2 16 34 34 28\34/	1001	1001	0101	0011	13 3 2 16 34 34 20\34/
1/	3/	2/	4/	3\$ RW CL P1\P2/	2/	3/	1/	4/	4\$ RW CL P1\P2/
0110	0101	0110	0011	1 15 12 6 34 34 34\28/	0110	0101	0110	0011	1 15 12 6 34 34 34\16/
1001	1010	0110	1100	14 4 7 9 34 34 36\34/	0110	1010	1001	1100	8 10 13 3 34 34 48\34/
0110	1010	1001	1100	8 10 13 3 34 34 34\40/	1001	1010	0110	1100	14 4 7 9 34 34 34\52/
1001	0101	1001	0011	11 5 2 16 34 34 32\34/	1001	0101	1001	0011	11 5 2 16 34 34 20\34/

3/ 1/ 2/ 4/	5\$ RW CL P1\P2/	3/ 2/ 1/ 4/	6\$ RW CL P1\P2/		
0101 0110 0110 0011	1 15 8 10 34 34 34\28/	0101 0110 0110 0011	1 15 8 10 34 34 34\24/		
1010 1001 0110 1100	14 4 11 5 34 34 36\34/	1010 0110 1001 1100	12 6 13 3 34 34 40\34/		
1010 0110 1001 1100	12 6 13 3 34 34 34\40/	1010 1001 0110 1100	14 4 11 5 34 34 34\44/		
0101 1001 1001 0011	7 9 2 16 34 34 32\34/	0101 1001 1001 0011	7 9 2 16 34 34 28\34/		
1/ 2/ 4/ 3/	7\$ RW CL P1\P2/	2/ 1/ 4/ 3/	8\$ RW CL P1\P2/		
0110 0110 0011 0101	1 14 15 4 34 34 34\22/	0110 0110 0011 0101	1 14 15 4 34 34 34\14/		
1001 0110 1100 1010	12 7 6 9 34 34 38\34/	0110 1001 1100 1010	8 11 10 5 34 34 46\34/		
0110 1001 1100 1010	8 11 10 5 34 34 34\46/	1001 0110 1100 1010	12 7 6 9 34 34 34\54/		
1001 1001 0011 0101	13 2 3 16 34 34 30\34/	1001 1001 0011 0101	13 2 3 16 34 34 22\34/		
2/ 3/ 4/ 1/	9\$ RW CL P1\P2/	3/ 2/ 4/ 1/	10\$ RW CL P1\P2/		
0110 0101 0011 0110	1 14 12 7 34 34 34\14/	0101 0110 0011 0110	1 14 8 11 34 34 34\22/		
0110 1010 1100 1001	8 11 13 2 34 34 46\34/	1010 0110 1100 1001	12 7 13 2 34 34 38\34/		
1001 1010 1100 0110	15 4 6 9 34 34 34\54/	1010 1001 1100 0110	15 4 10 5 34 34 34\46/		
1001 0101 0011 1001	10 5 3 16 34 34 22\34/	0101 1001 0011 1001	6 9 3 16 34 34 30\34/		
2/ 4/ 1/ 3/	11\$ RW CL P1\P2/	2/ 4/ 3/ 1/	12\$ RW CL P1\P2/		
0110 0011 0110 0101	1 12 15 6 34 34 34\10/	0110 0011 0101 0110	1 12 14 7 34 34 34\10/		
0110 1100 1001 1010	8 13 10 3 34 34 42\34/	0110 1100 1010 1001	8 13 11 2 34 34 42\34/		
1001 1100 0110 1010	14 7 4 9 34 34 34\58/	1001 1100 1010 0110	15 6 4 9 34 34 34\58/		
1001 0011 1001 0101	11 2 5 16 34 34 26\34/	1001 0011 0101 1001	10 3 5 16 34 34 26\34/		
1/ 2/ 3/ 6/	13# RW CL P1\P2/	2/ 1/ 3/ 6/	14# RW CL P1\P2/		
0110 0110 0101 0101	1 16 13 4 34 34 34\26/	0110 0110 0101 0101	1 16 13 4 34 34 34\18/		
1001 0110 1010 0101	11 6 7 10 34 34 42\34/	0110 1001 1010 0101	7 10 11 6 34 34 50\34/		
0110 1001 1010 1010	8 9 12 5 34 34 34\42/	1001 0110 1010 1010	12 5 8 9 34 34 34\50/		
1001 1001 0101 1010	14 3 2 15 34 34 26\34/	1001 1001 0101 1010	14 3 2 15 34 34 18\34/		
1/ 3/ 2/ 6/	15# RW CL P1\P2/	2/ 3/ 1/ 6/	16# RW CL P1\P2/		
0110 0101 0110 0101	1 16 11 6 34 34 34\30/	0110 0101 0110 0101	1 16 11 6 34 34 34\18/		
1001 1010 0110 0101	13 4 7 10 34 34 38\34/	0110 1010 1001 0101	7 10 13 4 34 34 50\34/		
0110 1010 1001 1010	8 9 14 3 34 34 34\38/	1001 1010 0110 1010	14 3 8 9 34 34 34\50/		
1001 0101 1001 1010	12 5 2 15 34 34 30\34/	1001 0101 1001 1010	12 5 2 15 34 34 18\34/		
3/ 1/ 2/ 6/	17# RW CL P1\P2/	3/ 2/ 1/ 6/	18# RW CL P1\P2/		
0101 0110 0110 0101	1 16 7 10 34 34 34\30/	0101 0110 0110 0101	1 16 7 10 34 34 34\26/		
1010 1001 0110 0101	13 4 11 6 34 34 38\34/	1010 0110 1001 0101	11 6 13 4 34 34 42\34/		
1010 0110 1001 1010	12 5 14 3 34 34 34\38/	1010 1001 0110 1010	14 3 12 5 34 34 34\42/		
0101 1001 1001 1010	8 9 2 15 34 34 30\34/	0101 1001 1001 1010	8 9 2 15 34 34 26\34/		
2/ 3/ 4/ 6/	19# RW CL P1\P2/	3/ 2/ 4/ 6/	20# RW CL P1\P2/		
0110 0101 0011 0101	1 14 11 8 34 34 34\14/	0101 0110 0011 0101	1 14 7 12 34 34 34\22/		
0110 1010 1100 0101	7 12 13 2 34 34 46\34/	1010 0110 1100 0101	11 8 13 2 34 34 38\34/		
1001 1010 1100 1010	16 3 6 9 34 34 34\54/	1010 1001 1100 1010	16 3 10 5 34 34 34\46/		
1001 0101 0011 1010	10 5 4 15 34 34 22\34/	0101 1001 0011 1010	6 9 4 15 34 34 30\34/		
1/ 2/ 4/ 7/	21# RW CL P1\P2/	2/ 1/ 4/ 7/	22# RW CL P1\P2/		
0110 0110 0011 0011	1 13 16 4 34 34 34\22/	0110 0110 0011 0011	1 13 16 4 34 34 34\14/		
1001 0110 1100 1100	12 8 5 9 34 34 38\34/	0110 1001 1100 1100	8 12 9 5 34 34 46\34/		
0110 1001 1100 0011	7 11 10 6 34 34 34\46/	1001 0110 1100 0011	11 7 6 10 34 34 34\54/		
1001 1001 0011 1100	14 2 3 15 34 34 30\34/	1001 1001 0011 1100	14 2 3 15 34 34 22\34/		
2/ 4/ 3/ 6/	23# RW CL P1\P2/	2/ 4/ 1/ 7/	24# RW CL P1\P2/		
0110 0011 0101 0101	1 12 13 8 34 34 34\10/	0110 0011 0110 0011	1 11 16 6 34 34 34\10/		
0110 1100 1010 0101	7 14 11 2 34 34 42\34/	0110 1100 1001 1100	8 14 9 3 34 34 42\34/		
1001 1100 1010 1010	16 5 4 9 34 34 34\58/	1001 1100 0110 0011	13 7 4 10 34 34 34\58/		
1001 0011 0101 1010	10 3 6 15 34 34 26\34/	1001 0011 1001 1100	12 2 5 15 34 34 26\34/		
1/2/6/3/ 25#	2/1/6/3/ 26#	1/3/6/2/ 27#	2/3/6/1/ 28#	3/1/6/2/ 29#	3/2/6/1/ 30#
1 16 13 4	1 16 13 4	1 16 10 7	1 16 10 7	1 16 6 11	1 16 6 11
10 7 6 11	6 11 10 7	13 4 6 11	6 11 13 4	13 4 10 7	10 7 13 4
8 9 12 5	12 5 8 9	8 9 15 2	15 2 8 9	12 5 15 2	15 2 12 5
15 2 3 14	15 2 3 14	12 5 3 14	12 5 3 14	8 9 3 14	8 9 3 14
2/3/6/4/ 31#	3/2/6/4/ 32#	1/2/7/4/ 33#	2/1/7/4/ 34#	2/4/6/3/ 35#	2/4/7/1/ 36#
1 15 10 8	1 15 6 12	1 13 16 4	1 13 16 4	1 12 13 8	1 10 16 7
6 12 13 3	10 8 13 3	12 8 5 9	8 12 9 5	6 15 10 3	8 15 9 2
16 2 7 9	16 2 11 5	6 10 11 7	10 6 7 11	16 5 4 9	13 6 4 11
11 5 4 14	7 9 4 14	15 3 2 14	15 3 2 14	11 2 7 14	12 3 5 14

1/2/6/7/ 38#	2/1/6/7/ 40#	3/1/6/7/ 42P	1/2/8/5/ 44#	2/1/8/5/ 46#	2/4/7/6/ 48#
1 15 14 4	1 15 14 4	1 15 6 12	1 14 15 4	1 14 15 4	1 10 15 8
10 8 5 11	6 12 9 7	14 4 9 7	10 8 5 11	6 12 9 7	7 16 9 2
7 9 12 6	11 5 8 10	11 5 16 2	7 9 12 6	11 5 8 10	14 5 4 11
16 2 3 13	16 2 3 13	8 10 3 13	16 3 2 13	16 3 2 13	12 3 6 13
2/6/1/3/ 50#	2/6/3/1/ 52#	3/6/2/1/ 54#	3/6/2/4/ 56#	3/6/4/2/ 58#	2/7/4/1/ 60#
1 16 11 6	1 16 10 7	1 16 4 13	1 15 4 14	1 14 4 15	1 10 16 7
4 13 10 7	4 13 11 6	10 7 11 6	10 8 11 5	11 8 10 5	8 15 9 2
14 3 8 9	15 2 8 9	15 2 14 3	16 2 13 3	16 3 13 2	11 4 6 13
15 2 5 12	14 3 5 12	8 9 5 12	7 9 6 12	6 9 7 12	14 5 3 12
2/6/1/7/ 64#	3/6/4/7/ 68#	2/7/4/6/ 72#	2/6/7/1/ 76#	3/6/7/4/ 80#	2/7/6/4/ 84#
1 15 12 6	1 13 4 16	1 10 15 8	1 14 12 7	1 13 4 16	1 11 14 8
4 14 9 7	12 8 9 5	7 16 9 2	4 15 9 6	12 8 9 5	6 16 9 3
13 3 8 10	15 3 14 2	12 3 6 13	13 2 8 11	14 2 15 3	12 2 7 13
16 2 5 11	6 10 7 11	14 5 4 11	16 3 5 10	7 11 6 10	15 5 4 10
6/2/3/1/ 88#	6/3/2/4/ 92#	6/4/3/2/ 96#	6/2/1/7/100#	6/3/4/7/104#	8/2/1/5/108#
1 16 6 11	1 15 4 14	1 12 6 15	1 15 8 10	1 13 4 16	1 8 15 10
4 13 7 10	6 12 7 9	7 14 4 9	4 14 5 11	8 12 5 9	4 14 5 11
15 2 12 5	16 2 13 3	16 5 11 2	13 3 12 6	15 3 14 2	13 3 12 6
14 3 9 8	11 5 10 8	10 3 13 8	16 2 9 7	10 6 11 7	16 9 2 7
6/2/7/1/112#	6/3/7/4/116#	8/2/5/1/120#	6/7/2/1/124#	6/7/3/4/128#	8/5/2/1/132#
1 14 8 11	1 13 4 16	1 8 14 11	1 12 8 13	1 11 6 16	1 8 12 13
4 15 5 10	8 12 5 9	4 15 5 10	6 15 3 10	8 14 3 9	6 15 3 10
13 2 12 7	14 2 15 3	13 2 12 7	11 2 14 7	12 2 15 5	11 2 14 7
16 3 9 6	11 7 10 6	16 9 3 6	16 5 9 4	13 7 10 4	16 9 5 4
133#	141#	149\$	157#	165#	173#
2 15 14 3	2 13 12 7	2 16 7 9	2 16 13 3	2 13 16 3	2 15 9 8
12 5 8 9	8 11 14 1	13 3 12 6	12 7 6 9	8 11 10 5	14 3 5 12
7 10 11 6	15 4 5 10	11 5 14 4	5 10 11 8	9 6 7 12	7 10 16 1
13 4 1 16	9 6 3 16	8 10 1 15	15 1 4 14	15 4 1 14	11 6 4 13
181#	189#	197#	205#	213#	221P
2 16 11 5	2 11 16 5	2 15 9 8	2 14 11 7	2 12 13 7	2 16 5 11
14 7 4 9	9 7 4 14	12 5 3 14	3 15 10 6	5 15 10 4	7 9 4 14
3 10 13 8	8 10 13 3	7 10 16 1	13 1 8 12	11 1 8 14	12 6 15 1
15 1 6 12	15 6 1 12	13 4 6 11	16 4 5 9	16 6 3 9	13 3 10 8
229#	237#	245#	253#	261#	269#
2 16 5 11	2 13 8 11	2 13 7 12	2 12 7 13	2 9 8 15	3 13 12 6
3 13 8 10	3 16 5 10	9 16 6 3	5 15 4 10	7 16 1 10	8 10 15 1
15 1 12 6	15 4 9 6	8 1 11 14	11 1 14 8	11 4 13 6	14 4 5 11
14 4 9 7	14 1 12 7	15 4 10 5	16 6 9 3	14 5 12 3	9 7 2 16
277\$	293#	309#	325#	341#	357#
3 16 10 5	3 16 10 5	3 14 9 8	3 12 13 6	3 16 5 10	3 13 6 12
13 2 8 11	15 6 4 9	12 5 2 15	5 14 11 4	2 13 8 11	9 16 7 2
6 9 15 4	2 11 13 8	6 11 16 1	10 1 8 15	14 1 12 7	8 1 10 15
12 7 1 14	14 1 7 12	13 4 7 10	16 7 2 9	15 4 9 6	14 4 11 5
373#	389\$	405#	421#	437#	453#
4 13 11 6	4 14 9 7	4 16 9 5	4 13 11 6	4 11 6 13	4 15 6 9
7 10 16 1	5 11 16 2	1 13 12 8	10 7 1 16	5 14 3 12	1 14 7 12
14 3 5 12	15 1 6 12	14 2 7 11	5 12 14 3	15 8 9 2	13 2 11 8
9 8 2 15	10 8 3 13	15 3 6 10	15 2 8 9	10 1 16 7	16 3 10 5
469#	485#	501\$	517#		
5 16 3 10	5 16 4 9	6 15 3 10	6 16 3 9		
9 4 15 6	15 10 6 3	12 1 13 8	1 11 8 14		
12 1 14 7	2 7 11 14	9 4 16 5	15 5 10 4		
8 13 2 11	12 1 13 8	7 14 2 11	12 2 13 7	. . . . .	

[Solution Counts = 528]

\* Monitor List of Solution Correspondences between Old and New \*

?: 0

```

1: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 0
41: 1 0 1 1 1 0 0 0 0 1 1 0 0 1 1 0 1 1 0 0 1 1 0 0 1 1 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1
81: 1 1 0 0 0 0 0 0 1 1 0 0 1 1 0 1 1 0 0 0 1 1 1 0 0 1 1 0 0 1 0 0 0 0 1 1 1 1 1 1 0
121: 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1
161: 1 1 1 1 0 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1
201: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
241: 1 1 0 0 0 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
281: 0 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
321: 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
361: 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1
401: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
441: 1 1 1 1 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
481: 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 0 0 1 1 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 0
521: 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1
561: 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
601: 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1
641: 1 1 1 0 0 0 1 1 1 1 1 0 0 0 0 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
681: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 1 1 0 1 1 1 1 1 0
721: 0 0 0 1 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
761: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
801: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
841: 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-----
+n: 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0

```

[OK!]

\*\* Calculated and Listed by Kanji Setsuda on Mar.18, 2011 with MacOSX & Xcode 3 \*\*

I was surprised at the result. The solutions, as many as 528 among 880, proved to be the "Euler Squares" even in their primary diagonals! How many!

It is really amazing. These 528 solutions should make the core of all 880 solutions.

I would recommend you to analyze the decomposed units of them directly and find anything beautiful about their structures and relations among solutions.

We have now come to the entrance to the fundamental study of magic squares.

### 3. How about neglecting the Latin Definitions for the 2 Primary Diagonals?

How many 'Euler Squares' of Order 4 can we make without the definition of two primary diagonals? It is 'all right', if we have the two primary diagonals adding up to the same Magic Constant 34. We don't mind whether each takes the Latin Pattern (0,0,1,1) or not. We only want all rows and columns to have the very Latin Pattern.

Let's make the 'Euler Squares' of Order 4 only with all rows and columns in Latin patterns at first. And in the last stage, let's pick up only those answers whose primary diagonals  $n_1+n_6+n_{11}+n_{16}$  and  $n_4+n_7+n_{10}+n_{13}$  add up to the constant sum 34.

The result of my recent experiment is shown below.

\*\* Make 'Euler Squares' of Order 4 by Binary Number System \*\*

[List 2 of Latin Units with Decompositions by Binary System]

1/	2/	3/	4/	5/	6/	7/	8/	9/
0 1 1 0	0 1 1 0	0 1 1 0	0 1 1 0	0 1 1 0	0 1 1 0	0 1 1 0	0 1 1 0	0 1 0 1
1 0 0 1	1 0 1 0	1 1 0 0	0 0 1 1	0 1 0 1	0 1 1 0	1 0 0 1	1 0 1 0	1 0 1 0
0 1 1 0	0 1 0 1	0 0 1 1	1 1 0 0	1 0 1 0	1 0 0 1	1 0 1 0	1 0 0 1	0 1 1 0
1 0 0 1	1 0 0 1	1 0 0 1	1 0 0 1	1 0 0 1	1 0 0 1	0 1 0 1	0 1 0 1	1 0 0 1
10/	11/	12/	13/	14/	15/	16/	17/	18/
0 1 0 1	0 1 0 1	0 1 1 0	0 1 1 0	0 1 0 1	0 1 0 1	0 0 1 1	0 0 1 1	0 0 1 1
0 1 1 0	1 0 1 0	1 0 0 1	1 1 0 0	1 0 1 0	1 1 0 0	1 1 0 0	0 1 1 0	1 0 1 0
1 0 1 0	1 0 1 0	1 1 0 0	1 0 0 1	1 1 0 0	1 0 1 0	0 1 1 0	1 1 0 0	1 1 0 0
1 0 0 1	0 1 0 1	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1	1 0 0 1	1 0 0 1	0 1 0 1
19/	20/	21/	22/	23/	24/	25/	26/	27/
0 0 1 1	0 0 1 1	0 1 1 0	0 1 1 0	0 1 0 1	0 1 0 1	0 1 1 0	0 1 1 0	0 1 1 0
1 1 0 0	1 1 0 0	1 0 0 1	0 0 1 1	1 0 1 0	0 0 1 1	1 0 0 1	0 1 0 1	1 0 0 1
1 0 1 0	1 1 0 0	0 0 1 1	1 0 0 1	0 0 1 1	1 0 1 0	0 1 0 1	1 0 0 1	1 0 0 1
0 1 0 1	0 0 1 1	1 1 0 0	1 1 0 0	1 1 0 0	1 1 0 0	1 0 1 0	1 0 1 0	0 1 1 0



28/	29/	30/	31/	32/	33/	34/	35/	36/
0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1	0 0 1 1
1 0 0 1	1 0 1 0	1 1 0 0	0 0 1 1	0 1 0 1	0 1 1 0	1 0 0 1	1 0 1 0	1 0 0 1
0 1 1 0	0 1 0 1	0 0 1 1	1 1 0 0	1 0 1 0	1 0 0 1	1 0 1 0	1 0 0 1	0 1 1 0
1 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0	0 1 1 0	0 1 1 0	1 1 0 0
37/	38/	39/	40/	41/	42/	43/	44/	45/
0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1
1 0 1 0	1 1 0 0	0 0 1 1	0 1 0 1	0 1 1 0	1 1 0 0	0 1 0 1	1 0 0 1	1 1 0 0
0 1 0 1	0 0 1 1	1 1 0 0	1 0 1 0	1 0 0 1	0 1 0 1	1 1 0 0	1 1 0 0	1 0 0 1
1 1 0 0	1 1 0 0	1 1 0 0	1 1 0 0	1 1 0 0	1 0 1 0	1 0 1 0	0 1 1 0	0 1 1 0
46/	47/	48/	49/	50/	51/	52/	53/	54/
1 1 0 0	1 1 0 0	1 1 0 0	1 1 0 0	1 1 0 0	1 1 0 0	1 1 0 0	1 1 0 0	1 1 0 0
0 0 1 1	0 1 1 0	1 0 1 0	0 0 1 1	1 0 0 1	1 0 1 0	1 1 0 0	0 0 1 1	0 1 0 1
0 1 1 0	0 0 1 1	0 0 1 1	1 0 1 0	0 1 1 0	0 1 0 1	0 0 1 1	1 1 0 0	1 0 1 0
1 0 0 1	1 0 0 1	0 1 0 1	0 1 0 1	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1
55/	56/	57/	58/	59/	60/	61/	62/	63/
1 1 0 0	1 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0
0 1 1 0	0 1 0 1	0 1 1 0	1 0 0 1	1 0 1 0	1 1 0 0	0 0 1 1	0 1 0 1	0 1 1 0
1 0 0 1	0 1 1 0	0 1 0 1	0 1 1 0	0 1 0 1	0 0 1 1	1 1 0 0	1 0 1 0	1 0 0 1
0 0 1 1	1 0 0 1	1 0 0 1	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1
64/	65/	66/	67/	68/	69/	70/	71/	72/
1 0 0 1	1 0 0 1	1 0 0 1	1 0 1 0	1 0 1 0	1 0 0 1	1 0 0 1	1 1 0 0	1 1 0 0
0 1 1 0	1 0 1 0	0 1 1 0	1 1 0 0	0 1 0 1	1 1 0 0	0 1 1 0	0 0 1 1	0 0 1 1
0 1 1 0	0 1 1 0	1 0 1 0	0 1 0 1	1 1 0 0	0 1 1 0	1 1 0 0	0 0 1 1	0 1 0 1
1 0 0 1	0 1 0 1	0 1 0 1	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1	1 1 0 0	1 0 1 0
73/	74/	75/	76/	77/	78/	79/	80/	81/
1 1 0 0	1 1 0 0	1 1 0 0	1 0 1 0	1 0 1 0	1 0 0 1	1 0 0 1	1 0 1 0	1 0 1 0
0 1 0 1	1 0 0 1	0 0 1 1	0 0 1 1	0 1 0 1	0 0 1 1	0 1 1 0	0 1 0 1	1 0 0 1
0 0 1 1	0 0 1 1	1 0 0 1	0 1 0 1	0 0 1 1	0 1 1 0	0 0 1 1	0 1 0 1	0 1 0 1
1 0 1 0	0 1 1 0	0 1 1 0	1 1 0 0	1 1 0 0	1 1 0 0	1 1 0 0	1 0 1 0	0 1 1 0
82/	83/	84/	85/	86/	87/	88/	89/	90/
1 0 1 0	1 0 0 1	1 0 0 1	1 0 0 1	1 0 0 1	1 0 0 1	1 0 0 1	1 0 0 1	1 0 0 1
0 1 0 1	0 1 0 1	0 1 1 0	1 0 0 1	1 0 1 0	1 1 0 0	0 0 1 1	0 1 0 1	0 1 1 0
1 0 0 1	0 1 1 0	0 1 0 1	0 1 1 0	0 1 0 1	0 0 1 1	1 1 0 0	1 0 1 0	1 0 0 1
0 1 1 0	1 0 1 0	1 0 1 0	0 1 1 0	0 1 1 0	0 1 1 0	0 1 1 0	0 1 1 0	0 1 1 0

[Count of Latin Units = 90]

[Reference Table 2 for the Best Combinations]

*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	..		
1	16	12	12	12	12	8	12	8	12	8	8	12	8	8	8	12	8	8	8	12	8	8	8	12	8	8	12	8	8	12	..
2	12	16	12	12	8	12	8	12	12	8	8	10	10	10	6	10	10	10	6	8	10	10	10	10	6	12	8	8	8	..	
3	12	12	16	8	12	12	10	10	10	10	8	8	12	6	10	12	8	6	10	8	12	8	10	6	10	10	8	8	8	..	
4	12	12	8	16	12	12	10	10	10	10	8	12	8	10	6	8	12	10	6	8	8	12	6	10	10	10	8	8	8	..	
5	12	8	12	12	16	12	12	8	8	12	8	10	10	6	10	10	10	6	10	8	10	10	6	10	8	12	8	8	8	..	
6	8	12	12	12	12	16	8	12	8	12	8	8	12	8	8	8	12	8	8	8	12	8	8	8	12	8	8	4	8	..	
7	12	8	10	10	12	8	16	12	8	8	12	12	10	8	10	8	6	10	12	8	12	10	8	10	8	8	12	8	8	..	
8	8	12	10	10	8	12	12	16	8	8	12	10	12	10	8	6	8	12	10	8	10	12	10	8	8	8	12	4	8	..	
9	12	12	10	10	8	8	8	8	16	12	12	8	6	12	10	12	10	10	8	8	8	6	12	10	8	4	4	12	8	..	
10	8	8	10	10	12	12	8	8	12	16	12	6	8	10	12	10	12	8	10	8	6	8	10	12	4	8	4	8	8	..	
11	8	8	8	8	8	8	12	12	12	12	16	8	8	12	12	8	8	12	12	8	8	8	12	12	4	4	8	8	8	..	
12	12	10	8	12	10	8	12	10	8	6	8	16	12	12	10	8	8	10	8	12	8	8	4	6	12	10	12	10	8	..	
13	8	10	12	8	10	12	10	12	6	8	8	12	16	10	12	8	8	8	10	12	8	8	6	4	10	12	12	6	8	..	
14	8	10	6	10	6	8	8	10	12	10	12	12	10	16	12	8	10	12	8	12	4	6	8	8	8	6	8	10	8	..	
15	8	6	10	6	10	8	10	8	10	12	12	10	12	12	16	10	8	8	12	12	6	4	8	8	6	8	8	10	8	..	
16	12	10	12	8	10	8	8	6	12	10	8	8	8	8	10	16	12	10	12	12	8	4	8	6	8	6	4	10	8	..	
17	8	10	8	12	10	12	6	8	10	12	8	8	8	10	8	12	16	12	10	12	4	8	6	8	6	8	4	6	8	..	
18	8	10	6	10	6	8	10	12	10	8	12	10	8	12	8	10	12	16	12	12	6	8	8	8	6	4	8	6	8	..	
19	8	6	10	6	10	8	12	10	8	10	12	8	10	8	12	12	10	12	16	12	8	6	8	8	4	6	8	6	8	..	
20	8	8	8	8	8	8	8	8	8	8	8	12	12	12	12	12	12	12	12	16	4	4	4	4	8	8	8	8	8	..	
21	12	10	12	8	10	8	12	10	8	6	8	8	8	4	6	8	4	6	8	4	16	12	12	10	12	10	12	10	8	..	

```

22| 8 10 8 12 10 12 10 12 6 8 8 8 8 6 4 4 8 8 6 4 12 16 10 12 10 12 12 6 ..
23| 8 10 10 6 6 8 8 10 12 10 12 4 6 8 8 8 6 8 8 4 12 10 16 12 8 6 8 10 ..
24| 8 6 6 10 10 8 10 8 10 12 12 6 4 8 8 6 8 8 8 4 10 12 12 16 6 8 8 10 ..
25| 12 12 10 10 8 8 8 8 4 4 12 10 8 6 8 6 6 4 8 12 10 8 6 16 12 12 12 ..
26| 8 8 10 10 12 12 8 8 4 8 4 10 12 6 8 6 8 4 6 8 10 12 6 8 12 16 12 8 ..
27| 8 8 8 8 8 8 12 12 4 4 8 12 12 8 8 4 4 8 8 8 12 12 8 8 12 12 16 8 ..
28| 12 8 8 8 8 4 8 4 12 8 8 10 6 10 10 10 6 6 6 8 10 6 10 10 12 8 8 16 ..
29| 8 12 8 8 4 8 4 8 12 8 8 8 8 12 8 8 8 8 4 8 8 8 12 8 12 8 8 12 ..
30| 8 8 12 4 8 8 6 6 10 10 8 6 10 8 12 10 6 4 8 8 10 6 12 8 10 10 8 12 ..
31| 8 8 4 12 8 8 6 6 10 10 8 10 6 12 8 6 10 8 4 8 6 10 8 12 10 10 8 12 ..
32| 8 4 8 8 12 8 8 4 8 12 8 8 8 8 12 8 8 4 8 8 8 8 8 12 8 12 8 12 ..
33| 4 8 8 8 8 12 4 8 8 12 8 6 10 10 10 6 10 6 6 8 6 10 10 10 8 12 8 8 ..
34| 8 4 6 6 8 4 12 8 8 8 12 10 8 10 12 6 4 8 10 8 10 8 10 12 8 8 12 12 ..
35| 4 8 6 6 4 8 8 12 8 8 12 8 10 12 10 4 6 10 8 8 8 10 12 10 8 8 12 8 ..
36| 12 8 8 8 8 4 10 6 10 6 8 8 4 6 6 12 8 10 10 8 12 8 10 10 10 6 8 12 ..
..| ... ..

```

\*\* 'Standard' Magic Squares of Order 4: Composing 'Euler Squares' \*\*  
 \*\* of Order 4 by 'New Euler's Method' with Binary Number System \*\*

[List 2 of Standard Solutions(Sorted):

Used Units//Sol\_N#(\$:Self-complementary; P:Pan-diagonal)|Sum-Checks|/]

1/	6/	11/	20/	1\$ RW CL P1\P2/	6/	1/	11/	20/	2\$ RW CL P1\P2/
0110	0110	0101	0011	1 15 14 4 34 34 34\24/	0110	0110	0101	0011	1 15 14 4 34 34 34\16/
1001	0110	1010	1100	12 6 7 9 34 34 40\34/	0110	1001	1010	1100	8 10 11 5 34 34 48\34/
0110	1001	1010	1100	8 10 11 5 34 34 34\44/	1001	0110	1010	1100	12 6 7 9 34 34 34\52/
1001	1001	0101	0011	13 3 2 16 34 34 28\34/	1001	1001	0101	0011	13 3 2 16 34 34 20\34/
1/	11/	6/	20/	3\$ RW CL P1\P2/	6/	11/	1/	20/	4\$ RW CL P1\P2/
0110	0101	0110	0011	1 15 12 6 34 34 34\28/	0110	0101	0110	0011	1 15 12 6 34 34 34\16/
1001	1010	0110	1100	14 4 7 9 34 34 36\34/	0110	1010	1001	1100	8 10 13 3 34 34 48\34/
0110	1010	1001	1100	8 10 13 3 34 34 34\40/	1001	1010	0110	1100	14 4 7 9 34 34 34\52/
1001	0101	1001	0011	11 5 2 16 34 34 32\34/	1001	0101	1001	0011	11 5 2 16 34 34 20\34/
11/	1/	6/	20/	5\$ RW CL P1\P2/	11/	6/	1/	20/	6\$ RW CL P1\P2/
0101	0110	0110	0011	1 15 8 10 34 34 34\28/	0101	0110	0110	0011	1 15 8 10 34 34 34\24/
1010	1001	0110	1100	14 4 11 5 34 34 36\34/	1010	0110	1001	1100	12 6 13 3 34 34 40\34/
1010	0110	1001	1100	12 6 13 3 34 34 34\40/	1010	1001	0110	1100	14 4 11 5 34 34 34\44/
0101	1001	1001	0011	7 9 2 16 34 34 32\34/	0101	1001	1001	0011	7 9 2 16 34 34 28\34/
1/	6/	20/	11/	7\$ RW CL P1\P2/	6/	1/	20/	11/	8\$ RW CL P1\P2/
0110	0110	0011	0101	1 14 15 4 34 34 34\22/	0110	0110	0011	0101	1 14 15 4 34 34 34\14/
1001	0110	1100	1010	12 7 6 9 34 34 38\34/	0110	1001	1100	1010	8 11 10 5 34 34 46\34/
0110	1001	1100	1010	8 11 10 5 34 34 34\46/	1001	0110	1100	1010	12 7 6 9 34 34 34\54/
1001	1001	0011	0101	13 2 3 16 34 34 30\34/	1001	1001	0011	0101	13 2 3 16 34 34 22\34/
6/	11/	20/	1/	9\$ RW CL P1\P2/	11/	6/	20/	1/	10\$ RW CL P1\P2/
0110	0101	0011	0110	1 14 12 7 34 34 34\14/	0101	0110	0011	0110	1 14 8 11 34 34 34\22/
0110	1010	1100	1001	8 11 13 2 34 34 46\34/	1010	0110	1100	1001	12 7 13 2 34 34 38\34/
1001	1010	1100	0110	15 4 6 9 34 34 34\54/	1010	1001	1100	0110	15 4 10 5 34 34 34\46/
1001	0101	0011	1001	10 5 3 16 34 34 22\34/	0101	1001	0011	1001	6 9 3 16 34 34 30\34/
6/	20/	1/	11/	11\$ RW CL P1\P2/	6/	20/	11/	1/	12\$ RW CL P1\P2/
0110	0011	0110	0101	1 12 15 6 34 34 34\10/	0110	0011	0101	0110	1 12 14 7 34 34 34\10/
0110	1100	1001	1010	8 13 10 3 34 34 42\34/	0110	1100	1010	1001	8 13 11 2 34 34 42\34/
1001	1100	0110	1010	14 7 4 9 34 34 34\58/	1001	1100	1010	0110	15 6 4 9 34 34 34\58/
1001	0011	1001	0101	11 2 5 16 34 34 26\34/	1001	0011	0101	1001	10 3 5 16 34 34 26\34/
1/	6/	11/	32/	13# RW CL P1\P2/	6/	1/	11/	32/	14# RW CL P1\P2/
0110	0110	0101	0101	1 16 13 4 34 34 34\26/	0110	0110	0101	0101	1 16 13 4 34 34 34\18/
1001	0110	1010	0101	11 6 7 10 34 34 42\34/	0110	1001	1010	0101	7 10 11 6 34 34 50\34/
0110	1001	1010	1010	8 9 12 5 34 34 34\42/	1001	0110	1010	1010	12 5 8 9 34 34 34\50/
1001	1001	0101	1010	14 3 2 15 34 34 26\34/	1001	1001	0101	1010	14 3 2 15 34 34 18\34/
1/	11/	6/	32/	15# RW CL P1\P2/	6/	11/	1/	32/	16# RW CL P1\P2/
0110	0101	0110	0101	1 16 11 6 34 34 34\30/	0110	0101	0110	0101	1 16 11 6 34 34 34\18/
1001	1010	0110	0101	13 4 7 10 34 34 38\34/	0110	1010	1001	0101	7 10 13 4 34 34 50\34/
0110	1010	1001	1010	8 9 14 3 34 34 34\38/	1001	1010	0110	1010	14 3 8 9 34 34 34\50/
1001	0101	1001	1010	12 5 2 15 34 34 30\34/	1001	0101	1001	1010	12 5 2 15 34 34 18\34/

11/ 1/ 6/ 32/	17# RW CL P1\P2/	11/ 3/ 4/ 29/	18# RW CL P1\P2/		
0101 0110 0110 0101 1 16 7 10 34 34 34\30/		0101 0110 0110 0101 1 16 7 10 34 34 34\26/			
1010 1001 0110 0101 13 4 11 6 34 34 38\34/		1010 1100 0011 1010 14 5 12 3 34 34 42\38/			
1010 0110 1001 1010 12 5 14 3 34 34 34\38/		1010 0011 1100 0101 11 4 13 6 34 34 30\38/			
0101 1001 1001 1010 8 9 2 15 34 34 30\34/		0101 1001 1001 1010 8 9 2 15 34 34 30\34/			
11/ 5/ 2/ 31/	19# RW CL P1\P2/	11/ 6/ 1/ 32/	20# RW CL P1\P2/		
0101 0110 0110 0101 1 16 7 10 34 34 34\29/		0101 0110 0110 0101 1 16 7 10 34 34 34\26/			
1010 0101 1010 0011 11 5 12 6 34 34 39\32/		1010 0110 1001 0101 11 6 13 4 34 34 42\34/			
1010 1010 0101 1100 14 4 13 3 34 34 36\41/		1010 1001 0110 1010 14 3 12 5 34 34 34\42/			
0101 1001 1001 1010 8 9 2 15 34 34 27\34/		0101 1001 1001 1010 8 9 2 15 34 34 26\34/			
6/ 11/ 20/ 32/	21# RW CL P1\P2/	6/ 15/ 18/ 31/	22# RW CL P1\P2/		
0110 0101 0011 0101 1 14 11 8 34 34 34\14/		0110 0101 0011 0101 1 14 11 8 34 34 34\11/			
0110 1010 1100 0101 7 12 13 2 34 34 46\34/		0110 1100 1010 0011 7 13 12 2 34 34 45\36/			
1001 1010 1100 1010 16 3 6 9 34 34 34\54/		1001 1010 1100 1100 16 4 5 9 34 34 32\55/			
1001 0101 0011 1010 10 5 4 15 34 34 22\34/		1001 0011 0101 1010 10 3 6 15 34 34 25\34/			
11/ 6/ 20/ 32/	23# RW CL P1\P2/	1/ 6/ 20/ 38/	24# RW CL P1\P2/		
0101 0110 0011 0101 1 14 7 12 34 34 34\22/		0110 0110 0011 0011 1 13 16 4 34 34 34\22/			
1010 0110 1100 0101 11 8 13 2 34 34 38\34/		1001 0110 1100 1100 12 8 5 9 34 34 38\34/			
1010 1001 1100 1010 16 3 10 5 34 34 34\46/		0110 1001 1100 0011 7 11 10 6 34 34 34\46/			
0101 1001 0011 1010 6 9 4 15 34 34 30\34/		1001 1001 0011 1100 14 2 3 15 34 34 30\34/			
6/ 1/ 20/ 38/	25# RW CL P1\P2/	6/ 20/ 11/ 32/	26# RW CL P1\P2/		
0110 0110 0011 0011 1 13 16 4 34 34 34\14/		0110 0011 0101 0101 1 12 13 8 34 34 34\10/			
0110 1001 1100 1100 8 12 9 5 34 34 46\34/		0110 1100 1010 0101 7 14 11 2 34 34 42\34/			
1001 0110 1100 0011 11 7 6 10 34 34 34\54/		1001 1100 1010 1010 16 5 4 9 34 34 34\58/			
1001 1001 0011 1100 14 2 3 15 34 34 22\34/		1001 0011 0101 1010 10 3 6 15 34 34 26\34/			
6/ 20/ 1/ 38/	27# RW CL P1\P2/	6/ 19/ 14/ 39/	28# RW CL P1\P2/		
0110 0011 0110 0011 1 11 16 6 34 34 34\10/		0110 0011 0101 0011 1 11 14 8 34 34 34\14/			
0110 1100 1001 1100 8 14 9 3 34 34 42\34/		0110 1100 1010 0011 7 13 12 2 34 34 42\30/			
1001 1100 0110 0011 13 7 4 10 34 34 34\58/		1001 1010 1100 1100 16 4 5 9 34 34 38\58/			
1001 0011 1001 1100 12 2 5 15 34 34 26\34/		1001 0101 0011 1100 10 6 3 15 34 34 22\34/			
1/ 6/ 32/ 11/	29# RW CL P1\P2/	3/ 4/ 29/ 11/	30# RW CL P1\P2/		
0110 0110 0101 0101 1 16 13 4 34 34 34\26/		0110 0110 0101 0101 1 16 13 4 34 34 34\18/			
1001 0110 0101 1010 10 7 6 11 34 34 42\34/		1100 0011 1010 1010 12 9 8 5 34 34 50\42/			
0110 1001 1010 1010 8 9 12 5 34 34 34\42/		0011 1100 0101 1010 6 7 10 11 34 34 26\42/			
1001 1001 1010 0101 15 2 3 14 34 34 26\34/		1001 1001 1010 0101 15 2 3 14 34 34 26\34/			
5/ 2/ 31/ 11/	31# RW CL P1\P2/	6/ 1/ 32/ 11/	32# RW CL P1\P2/		
0110 0110 0101 0101 1 16 13 4 34 34 34\24/		0110 0110 0101 0101 1 16 13 4 34 34 34\18/			
0101 1010 0011 1010 6 9 8 11 34 34 44\30/		0110 1001 0101 1010 6 11 10 7 34 34 50\34/			
1010 0101 1100 1010 12 7 10 5 34 34 38\48/		1001 0110 1010 1010 12 5 8 9 34 34 34\50/			
1001 1001 1010 0101 15 2 3 14 34 34 20\34/		1001 1001 1010 0101 15 2 3 14 34 34 18\34/			
1/ 11/ 32/ 6/	33# RW CL P1\P2/	6/ 11/ 32/ 1/	34# RW CL P1\P2/		
0110 0101 0101 0110 1 16 10 7 34 34 34\32/		0110 0101 0101 0110 1 16 10 7 34 34 34\18/			
1001 1010 0101 0110 13 4 6 11 34 34 36\34/		0110 1010 0101 1001 6 11 13 4 34 34 50\34/			
0110 1010 1010 1001 8 9 15 2 34 34 34\36/		1001 1010 1010 0110 15 2 8 9 34 34 34\50/			
1001 0101 1010 1001 12 5 3 14 34 34 32\34/		1001 0101 1010 1001 12 5 3 14 34 34 18\34/			
11/ 1/ 32/ 6/	35# RW CL P1\P2/	11/ 6/ 32/ 1/	36# RW CL P1\P2/		
0101 0110 0101 0110 1 16 6 11 34 34 34\32/		0101 0110 0101 0110 1 16 6 11 34 34 34\26/			
1010 1001 0101 0110 13 4 10 7 34 34 36\34/		1010 0110 0101 1001 10 7 13 4 34 34 42\34/			
1010 0110 1010 1001 12 5 15 2 34 34 34\36/		1010 1001 1010 0110 15 2 12 5 34 34 34\42/			
0101 1001 1010 1001 8 9 3 14 34 34 32\34/		0101 1001 1010 1001 8 9 3 14 34 34 26\34/			
37#	38#	39#	40#	41#	42#
1 15 10 8	1 15 6 12	1 13 16 4	1 13 16 4	1 13 16 4	1 13 16 4
6 12 13 3	10 8 13 3	12 8 5 9	12 10 7 5	6 10 7 11	8 12 9 5
16 2 7 9	16 2 11 5	6 10 11 7	6 8 9 11	12 8 9 5	10 6 7 11
11 5 4 14	7 9 4 14	15 3 2 14	15 3 2 14	15 3 2 14	15 3 2 14
43#	44#	45#	46#	47#	48#
1 12 13 8	1 10 16 7	1 16 9 8	1 16 5 12	1 16 5 12	1 15 14 4
6 15 10 3	8 15 9 2	14 5 4 11	15 6 11 2	10 6 11 7	11 8 5 10
16 5 4 9	13 6 4 11	7 10 15 2	10 3 14 7	15 3 14 2	6 9 12 7
11 2 7 14	12 3 5 14	12 3 6 13	8 9 4 13	8 9 4 13	16 2 3 13

50#	54#	58P	62#	66#	70#
1 15 14 4	1 15 14 4	1 15 6 12	1 14 15 4	1 14 15 4	1 9 16 8
12 9 6 7	7 12 9 6	14 4 9 7	10 11 8 5	6 12 9 7	7 15 10 2
5 8 11 10	10 5 8 11	11 5 16 2	7 6 9 12	11 5 8 10	14 4 5 11
16 2 3 13	16 2 3 13	8 10 3 13	16 3 2 13	16 3 2 13	12 6 3 13
74#	78#	82#	86#	90#	94#
1 16 10 7	1 15 4 14	1 10 16 7	1 16 3 14	1 15 12 6	1 15 12 6
4 13 11 6	10 8 11 5	8 15 9 2	12 7 10 5	10 13 8 3	4 14 9 7
15 2 8 9	16 2 13 3	11 4 6 13	13 2 15 4	7 4 9 14	13 3 8 10
14 3 5 12	7 9 6 12	14 5 3 12	8 9 6 11	16 2 5 11	16 2 5 11
103#	112#	121#	130#	139#	148#
1 12 15 6	1 16 9 8	1 13 12 8	1 15 10 8	1 12 13 8	1 16 6 11
10 13 8 3	4 11 6 13	2 14 7 11	12 13 6 3	11 14 7 2	4 13 7 10
7 4 9 14	14 5 12 3	16 4 9 5	5 4 11 14	6 3 10 15	15 2 12 5
16 5 2 11	15 2 7 10	15 3 6 10	16 2 7 9	16 5 4 9	14 3 9 8
157#	166#	175#	184#	193#	202P
1 16 3 14	1 13 8 12	1 14 8 11	1 10 8 15	1 11 6 16	1 12 6 15
13 11 6 4	4 16 5 9	13 12 2 7	7 16 2 9	7 13 4 10	8 13 3 10
8 2 15 9	15 3 10 6	4 5 15 10	14 5 11 4	14 2 15 3	11 2 16 5
12 5 10 7	14 2 11 7	16 3 9 6	12 3 13 6	12 8 9 5	14 7 9 4
209#	225\$	241P	257#	273P	289#
2 15 14 3	2 16 13 3	2 16 9 7	2 15 14 3	2 16 9 7	2 9 16 7
12 5 8 9	11 5 8 10	13 3 6 12	5 10 7 12	11 5 4 14	4 14 11 5
7 10 11 6	7 9 12 6	8 10 15 1	11 8 9 6	8 10 15 1	15 3 6 10
13 4 1 16	14 4 1 15	11 5 4 14	16 1 4 13	13 3 6 12	13 8 1 12
305#	321#	337#	353#	369#	385#
2 16 1 15	2 15 1 16	2 16 7 9	2 16 5 11	2 13 4 15	2 9 7 16
14 9 8 3	11 10 8 5	14 11 4 5	3 13 8 10	7 10 5 12	8 15 1 10
11 4 13 6	14 3 13 4	3 6 13 12	15 1 12 6	14 3 16 1	13 6 12 3
7 5 12 10	7 6 12 9	15 1 10 8	14 4 9 7	11 8 9 6	11 4 14 5
401#	417P	433#	449#	465#	481#
2 11 7 14	3 16 9 6	3 16 9 6	3 15 10 6	3 13 12 6	3 13 12 6
9 16 4 5	13 2 7 12	5 10 15 4	2 14 11 7	10 4 5 15	2 16 9 7
8 1 13 12	8 11 14 1	14 1 8 11	16 4 5 9	7 9 16 2	14 4 5 11
15 6 10 3	10 5 4 15	12 7 2 13	13 1 8 12	14 8 1 11	15 1 8 10
497P	513#	529#	545#	561P	577#
3 13 2 16	3 15 6 10	3 15 6 10	3 14 2 15	3 13 2 16	4 15 10 5
10 8 11 5	2 14 7 11	5 16 9 4	8 9 5 12	6 12 7 9	8 11 14 1
15 1 14 4	16 4 9 5	12 1 8 13	13 4 16 1	15 1 14 4	13 6 3 12
6 12 7 9	13 1 12 8	14 2 11 7	10 7 11 6	10 8 11 5	9 2 7 16
593#	609#	625#	641#	657#	673#
4 16 9 5	4 16 9 5	4 16 9 5	4 16 9 5	4 11 14 5	4 11 13 6
13 1 8 12	6 10 15 3	1 13 12 8	1 13 12 8	6 13 12 3	5 14 12 3
6 10 15 3	11 1 8 14	15 3 6 10	14 2 7 11	9 2 7 16	10 1 7 16
11 7 2 14	13 7 2 12	14 2 7 11	15 3 6 10	15 8 1 10	15 8 2 9
689#	705#	721#	737#	753\$	769#
4 14 3 13	4 15 5 10	4 16 1 13	4 7 10 13	5 16 4 9	5 11 10 8
5 7 10 12	6 9 3 16	5 9 8 12	5 16 1 12	11 2 14 7	3 16 13 2
16 2 15 1	11 8 14 1	14 2 15 3	14 9 8 3	10 3 15 6	14 1 4 15
9 11 6 8	13 2 12 7	11 7 10 6	11 2 15 6	8 13 1 12	12 6 7 9
785#	801#	817#	833#	849#	865#
5 16 3 10	5 16 2 11	6 16 3 9	6 15 1 12	6 7 12 9	7 16 2 9
13 8 11 2	13 8 10 3	11 1 14 8	10 3 13 8	3 16 13 2	11 4 14 5
4 1 14 15	4 1 15 14	10 4 15 5	11 2 16 5	14 1 4 15	6 1 15 12
12 9 6 7	12 9 7 6	7 13 2 12	7 14 4 9	11 10 5 8	10 13 3 8

[Solution Counts = 880/37608]

\* Monitor List of Solution Correspondences between Old and New \*

