

## Program Sheets for the 'New Euler's Method' of Order 5 and 7

(Appendix to the article of Section 11, Chapter 4, Part 4)

### Program #4:

```
//
/** Simultaneous MS55: Both Self-complementary and Pan-diagonal; Reconstructed **
/** by the 5-th Increment Number System and New Euler's Method **
/** File:'CES55Sml.c': Dictated by Kanji Setsuda **
/** in 2003, 2006, 2011; Revised on Apr.15, 2012 and **
/** Recompiled on Feb.13, 2015 with MacOSX 10.10.2 & Xcode 6.1.1 **
//
#include <stdio.h>
//
short int cnt, cnt2;
short lcnt, cnt3;
short nm[26], uf1g[26], tn[26];
short SC, LSM;
short u1,u2;
short anm[7][27];
char tlu[17][26];
char mtc[17][17];
short cs[5][21];
//
short rw1[5], cl1[5], pd1[5], pb1[5];
short rw2[5], cl2[5], pd2[5], pb2[5];
short rw3[5], cl3[5], pd3[5], pb3[5];
short rw4[5], cl4[5], pd4[5], pb4[5];
short rw5[5], cl5[5], pd5[5], pb5[5];
//
void stp00(void);
void stp01(void), stp02(void), stp03(void), stp04(void), stp05(void);
void stp06(void), stp07(void), stp08(void), stp09(void), stp10(void);
void stp11(void), stp12(void), stp13(void);
void lurecord(void), prlunit(short x);
//
void tblmtc(void);
void cmbcmp(void), solsavep(void);
void pr1ans(void);
void chksms(short x);
//
/** Main Program *
int main(void){
short n;
printf("\n");
printf("** Simultaneous Magic Squares of Order 5: Self-complementary and Pan-diagonal;\n");
printf("  'Complete Euler Type' Reconstructed by the 'Greco-Latinian Method' **\n");
for(n=0;n<26;n++){nm[n]=0;}
for(n=0;n<5;n++){uf1g[n]=0;
rw1[n]=0; cl1[n]=0; pd1[n]=0; pb1[n]=0;
rw2[n]=0; cl2[n]=0; pd2[n]=0; pb2[n]=0;
rw3[n]=0; cl3[n]=0; pd3[n]=0; pb3[n]=0;
rw4[n]=0; cl4[n]=0; pd4[n]=0; pb4[n]=0;
rw5[n]=0; cl5[n]=0; pd5[n]=0; pb5[n]=0;
}
SC=4; LSM=10; cnt=0;
printf("\n");
printf("** List of Latin Units by /D5i **\n");
stp00(); // Make the Latin Units by /D5i
lcnt=cnt;
prlunit(8);
printf("\n");
printf("** Complete Euler Type of Simultaneous MS55: Self-complementary and Pan-diagonal; **\n");
printf(" ** Primitive List of Reconstruction with Full Data: Check Sums **\n");
```

```

printf(" [Latin Units: High, Low; [Mathem]; Check Sums:|Row|Column\\Pd1/Pd2|; [S.Nmb]]\\n");
cnt=0;
cmbcmp(); // Combine and Compose
printf(" [Solution Counts = %d] [OK!]\\n",cnt);
printf("\\n");
printf("*** Recompiled and listed by Kanji Setsuda on\\n");
printf(" Feb.13, 2015 with MacOSX 10.10.2 & Xcode 6.1.1 **\\n");
printf("\\n");
return 0;
}
//
/** Make Latin Units for the Simultaneous MS55 **
/** Basic Form for Simultaneous MS55: SC & PD **
//
// 2 3 4 5| 1| 2| 3| 4| 5| 1 2 3 4
// |-----|
// 7 8 9 10| 6| 7| 8| 9|10| 6 7 8 9
// |-----|
// 12 13 14 15|11|12|13|14|15|11 12 13 14
// |-----|
// 17 18 19 20|16|17|18|19|20|16 17 18 19
// |-----|
// 22 23 24 25|21|22|23|24|25|21 22 23 24
// |-----|
/** Define Conditions for Rows and Columns: C=10 **
// n1+n2+n3+n4+n5=C ... rw1 | n1+n6+n11+n16+n21=C ... c1
// n6+n7+n8+n9+n10=C ... rw2 | n2+n7+n12+n17+n22=C ... c2
// n11+n12+n13+n14+n15=C ... rw3 | n3+n8+n13+n18+n23=C ... c3
// n16+n17+n18+n19+n20=C ... rw4 | n4+n9+n14+n19+n24=C ... c4
// n21+n22+n23+n24+n25=C ... rw5 | n5+n10+n15+n20+n25=C ... c5
/** Pan-diagonal Conditions of the Object: C=10 **
// n1+n7+n13+n19+n25=C ... pd1 | n1+n10+n14+n18+n22=C ... pd6
// n2+n8+n14+n20+n21=C ... pd2 | n2+n6+n15+n19+n23=C ... pd7
// n3+n9+n15+n16+n22=C ... pd3 | n3+n7+n11+n20+n24=C ... pd8
// n4+n10+n11+n17+n23=C ... pd4 | n4+n8+n12+n16+n25=C ... pd9
// n5+n6+n12+n18+n24=C ... pd5 | n5+n9+n13+n17+n21=C ... pd10
/** Self-complementary Conditions of the Object: SC=4 **
// n1+n25=n2+n24=n3+n23=n4+n22=n5+n21=n6+n20=n7+n19=
// n8+n18=n9+n17=n10+n16=n11+n15=n12+n14=n13+n13=SC ... scc
/** Normalizing Inequality Conditions for the Standard Solutions **
// n1<n25; and n1<n5<n21;
//
/** Level #0: *
// Set n13{3,3,1,5}
void stp00(){
short a;
a=2; nm[13]=a;
uflg[a]++; rw3[a]=1; c13[a]=1; pd1[a]=1; pb5[a]=1;
stp01();
uflg[a]--; rw3[a]=0; c13[a]=0; pd1[a]=0; pb5[a]=0;
}
/** Level #1: *
// Set n1{1,1,1,1} and n25{5,5,1,4}
void stp01(){
short a,b;
for(a=0;a<5;a++){b=SC-a;
if((uflg[a]<5)&&(uflg[b]<5)){
if((rw1[a]==0)&&(c11[a]==0)&&(pd1[a]==0)&&(pb1[a]==0)){
if((rw5[b]==0)&&(c15[b]==0)&&(pd1[b]==0)&&(pb4[b]==0)){
nm[1]=a; nm[25]=b;
uflg[a]++; rw1[a]=1; c11[a]=1; pd1[a]=1; pb1[a]=1;
uflg[b]++; rw5[b]=1; c15[b]=1; pd1[b]=1; pb4[b]=1;
stp02();
uflg[b]--; rw5[b]=0; c15[b]=0; pd1[b]=0; pb4[b]=0;
uflg[a]--; rw1[a]=0; c11[a]=0; pd1[a]=0; pb1[a]=0;}}}}

```

```

}
}
// Set n7{2,2,1,3} and n19{4,4,1,2}
void stp02(){
short a,b;
for(a=0;a<5;a++){b=SC-a;
if((uflg[a]<5)&&(uflg[b]<5)){
if((rw2[a]==0)&&(cl2[a]==0)&&(pd1[a]==0)&&(pb3[a]==0)){
if((rw4[b]==0)&&(cl4[b]==0)&&(pd1[b]==0)&&(pb2[b]==0)){
nm[7]=a; nm[19]=b;
uflg[a]++; rw2[a]=1; cl2[a]=1; pd1[a]=1; pb3[a]=1;
uflg[b]++; rw4[b]=1; cl4[b]=1; pd1[b]=1; pb2[b]=1;
stp03();
uflg[b]--; rw4[b]=0; cl4[b]=0; pd1[b]=0; pb2[b]=0;
uflg[a]--; rw2[a]=0; cl2[a]=0; pd1[a]=0; pb3[a]=0;}}}}
}
}
/* Level #2: *
// Set n2{1,2,2,2} and n24{5,4,5,3}
void stp03(){
short a,b;
for(a=4;a>=0;a--){b=SC-a;
if((uflg[a]<5)&&(uflg[b]<5)){
if((rw1[a]==0)&&(cl2[a]==0)&&(pd2[a]==0)&&(pb2[a]==0)){
if((rw5[b]==0)&&(cl4[b]==0)&&(pd5[b]==0)&&(pb3[b]==0)){
nm[2]=a; nm[24]=b;
uflg[a]++; rw1[a]=1; cl2[a]=1; pd2[a]=1; pb2[a]=1;
uflg[b]++; rw5[b]=1; cl4[b]=1; pd5[b]=1; pb3[b]=1;
stp04();
uflg[b]--; rw5[b]=0; cl4[b]=0; pd5[b]=0; pb3[b]=0;
uflg[a]--; rw1[a]=0; cl2[a]=0; pd2[a]=0; pb2[a]=0;}}}}
}
}
// Set n3{1,3,3,3} and n23{5,3,4,2}
void stp04(){
short a,b;
for(a=4;a>=0;a--){b=SC-a;
if((uflg[a]<5)&&(uflg[b]<5)){
if((rw1[a]==0)&&(cl3[a]==0)&&(pd3[a]==0)&&(pb3[a]==0)){
if((rw5[b]==0)&&(cl3[b]==0)&&(pd4[b]==0)&&(pb2[b]==0)){
nm[3]=a; nm[23]=b;
uflg[a]++; rw1[a]=1; cl3[a]=1; pd3[a]=1; pb3[a]=1;
uflg[b]++; rw5[b]=1; cl3[b]=1; pd4[b]=1; pb2[b]=1;
stp05();
uflg[b]--; rw5[b]=0; cl3[b]=0; pd4[b]=0; pb2[b]=0;
uflg[a]--; rw1[a]=0; cl3[a]=0; pd3[a]=0; pb3[a]=0;}}}}
}
}
// Set n4{1,4,4,4} and n22{5,2,3,1}
void stp05(){
short a,b;
for(a=4;a>=0;a--){b=SC-a;
if((uflg[a]<5)&&(uflg[b]<5)){
if((rw1[a]==0)&&(cl4[a]==0)&&(pd4[a]==0)&&(pb4[a]==0)){
if((rw5[b]==0)&&(cl2[b]==0)&&(pd3[b]==0)&&(pb1[b]==0)){
nm[4]=a; nm[22]=b;
uflg[a]++; rw1[a]=1; cl4[a]=1; pd4[a]=1; pb4[a]=1;
uflg[b]++; rw5[b]=1; cl2[b]=1; pd3[b]=1; pb1[b]=1;
stp06();
uflg[b]--; rw5[b]=0; cl2[b]=0; pd3[b]=0; pb1[b]=0;
uflg[a]--; rw1[a]=0; cl4[a]=0; pd4[a]=0; pb4[a]=0;}}}}
}
}
// Set n5{1,5,5,5} and n21{5,1,2,5}
void stp06(){

```

```

short a,b;
for(a=4;a>=0;a--){b=SC-a;
  if((uflg[a]<5)&&(uflg[b]<5)){
    if((rw1[a]==0)&&(c15[a]==0)&&(pd5[a]==0)&&(pb5[a]==0)){
      if((rw5[b]==0)&&(c11[b]==0)&&(pd2[b]==0)&&(pb5[b]==0)){
        nm[5]=a; nm[21]=b;
        uflg[a]++; rw1[a]=1; c15[a]=1; pd5[a]=1; pb5[a]=1;
        uflg[b]++; rw5[b]=1; c11[b]=1; pd2[b]=1; pb5[b]=1;
        stp07();
        uflg[b]--; rw5[b]=0; c11[b]=0; pd2[b]=0; pb5[b]=0;
        uflg[a]--; rw1[a]=0; c15[a]=0; pd5[a]=0; pb5[a]=0;}}}}
  }
}
/* Level #3: *
// Set n11{3,1,4,3} and n15{3,5,3,2}
void stp07(){
  short a,b;
  for(a=4;a>=0;a--){b=SC-a;
    if((uflg[a]<5)&&(uflg[b]<5)){
      if((rw3[a]==0)&&(c11[a]==0)&&(pd4[a]==0)&&(pb3[a]==0)){
        if((rw3[b]==0)&&(c15[b]==0)&&(pd3[b]==0)&&(pb2[b]==0)){
          nm[11]=a; nm[15]=b;
          uflg[a]++; rw3[a]=1; c11[a]=1; pd4[a]=1; pb3[a]=1;
          uflg[b]++; rw3[b]=1; c15[b]=1; pd3[b]=1; pb2[b]=1;
          stp08();
          uflg[b]--; rw3[b]=0; c15[b]=0; pd3[b]=0; pb2[b]=0;
          uflg[a]--; rw3[a]=0; c11[a]=0; pd4[a]=0; pb3[a]=0;}}}}
    }
  }
// Set n16{4,1,3,4} and n10{2,5,4,1}
void stp08(){
  short a,b;
  for(a=0;a<5;a++){b=SC-a;
    if((uflg[a]<5)&&(uflg[b]<5)){
      if((rw4[a]==0)&&(c11[a]==0)&&(pd3[a]==0)&&(pb4[a]==0)){
        if((rw2[b]==0)&&(c15[b]==0)&&(pd4[b]==0)&&(pb1[b]==0)){
          nm[16]=a; nm[10]=b;
          uflg[a]++; rw4[a]=1; c11[a]=1; pd3[a]=1; pb4[a]=1;
          uflg[b]++; rw2[b]=1; c15[b]=1; pd4[b]=1; pb1[b]=1;
          stp09();
          uflg[b]--; rw2[b]=0; c15[b]=0; pd4[b]=0; pb1[b]=0;
          uflg[a]--; rw4[a]=0; c11[a]=0; pd3[a]=0; pb4[a]=0;}}}}
    }
  }
// Set n6{2,1,5,2} and n20{4,5,2,3}
void stp09(){
  short a,b;
  for(a=4;a>=0;a--){b=SC-a;
    if((uflg[a]<5)&&(uflg[b]<5)){
      if((rw2[a]==0)&&(c11[a]==0)&&(pd5[a]==0)&&(pb2[a]==0)){
        if((rw4[b]==0)&&(c15[b]==0)&&(pd2[b]==0)&&(pb3[b]==0)){
          nm[6]=a; nm[20]=b;
          uflg[a]++; rw2[a]=1; c11[a]=1; pd5[a]=1; pb2[a]=1;
          uflg[b]++; rw4[b]=1; c15[b]=1; pd2[b]=1; pb3[b]=1;
          stp10();
          uflg[b]--; rw4[b]=0; c15[b]=0; pd2[b]=0; pb3[b]=0;
          uflg[a]--; rw2[a]=0; c11[a]=0; pd5[a]=0; pb2[a]=0;}}}}
    }
  }
/* Level #4: *
// Set n8{2,3,2,4} and n18{4,3,5,1}
void stp10(){
  short a,b;
  for(a=0;a<5;a++){b=SC-a;
    if((uflg[a]<5)&&(uflg[b]<5)){

```

```

    if((rw2[a]==0)&&(cl3[a]==0)&&(pd2[a]==0)&&(pb4[a]==0)){
        if((rw4[b]==0)&&(cl3[b]==0)&&(pd5[b]==0)&&(pb1[b]==0)){
            nm[8]=a; nm[18]=b;
            uflg[a]++; rw2[a]=1; cl3[a]=1; pd2[a]=1; pb4[a]=1;
            uflg[b]++; rw4[b]=1; cl3[b]=1; pd5[b]=1; pb1[b]=1;
            stp11();
            uflg[b]--; rw4[b]=0; cl3[b]=0; pd5[b]=0; pb1[b]=0;
            uflg[a]--; rw2[a]=0; cl3[a]=0; pd2[a]=0; pb4[a]=0;}}}}
    }
}
// Set n9{2,4,3,5} and n17{4,2,4,5}
void stp11(){
    short a,b;
    for(a=4;a>=0;a--){b=SC-a;
        if((uflg[a]<5)&&(uflg[b]<5)){
            if((rw2[a]==0)&&(cl4[a]==0)&&(pd3[a]==0)&&(pb5[a]==0)){
                if((rw4[b]==0)&&(cl2[b]==0)&&(pd4[b]==0)&&(pb5[b]==0)){
                    nm[9]=a; nm[17]=b;
                    uflg[a]++; rw2[a]=1; cl4[a]=1; pd3[a]=1; pb5[a]=1;
                    uflg[b]++; rw4[b]=1; cl2[b]=1; pd4[b]=1; pb5[b]=1;
                    stp12();
                    uflg[b]--; rw4[b]=0; cl2[b]=0; pd4[b]=0; pb5[b]=0;
                    uflg[a]--; rw2[a]=0; cl4[a]=0; pd3[a]=0; pb5[a]=0;}}}}
        }
    }
// Set n12{3,2,5,4} and n14{3,4,2,1}
void stp12(){
    short a,b;
    for(a=4;a>=0;a--){b=SC-a;
        if((uflg[a]<5)&&(uflg[b]<5)){
            if((rw3[a]==0)&&(cl2[a]==0)&&(pd5[a]==0)&&(pb4[a]==0)){
                if((rw3[b]==0)&&(cl4[b]==0)&&(pd2[b]==0)&&(pb1[b]==0)){
                    nm[12]=a; nm[14]=b;
                    uflg[a]++; rw3[a]=1; cl2[a]=1; pd5[a]=1; pb4[a]=1;
                    uflg[b]++; rw3[b]=1; cl4[b]=1; pd2[b]=1; pb1[b]=1;
                    lurecord();
                    uflg[b]--; rw3[b]=0; cl4[b]=0; pd2[b]=0; pb1[b]=0;
                    uflg[a]--; rw3[a]=0; cl2[a]=0; pd5[a]=0; pb4[a]=0;}}}}
        }
    }
//
/** Record The Latin Units *
void lurecord(){
    short n;
    tlu[cnt][0]=cnt+1;
    for(n=1;n<26;n++){tlu[cnt][n]=nm[n];}
    cnt++;
}
//
/** Print Latin Units *
void prlunit(short x){
    short t,l,l5,m,n,p;
    for(t=0;t<lcnt;t=t+x){
        p=x; if(t+x>lcnt){p=lcnt-t;}
        for(m=0;m<p;m++){printf(" %9d/",tlu[t+m][0]);}
        printf("\n");
        for(l=0;l<5;l++){l5=l*5;
            for(m=0;m<p;m++){
                printf(" ");
                for(n=1;n<6;n++){printf("%2d",tlu[t+m][l5+n]);}
            }
            printf("\n");
        }
    }
}
printf(" [Count of Latin Units = %d]\n",lcnt);

```

```

/* Measure How Similar or How Complementary each pair is *
for(m=0;m<cnt;m++){
  for(n=0;n<cnt;n++){
    t=0;
    for(l=1;l<26;l++){if(tlu[m][l]==tlu[n][l]){t++;}}
    mtc[m][n]=t;
    t=0;
    for(l=1;l<26;l++){if(tlu[m][l]+tlu[n][l]==4){t++;}}
    if(mtc[m][n]<t){mtc[m][n]=t;}
  }
}
printf("\n");
printf(" [Reference Table for the Best Combination]\n");
printf(" 01");
for(n=0;n<16;n++){printf("%3d",n+1);}
printf("\n");
printf(" -----\n");
for(m=0;m<16;m++){
  printf("%3d",m+1);
  for(n=0;n<16;n++){printf("%3d",mtc[m][n]);}
  printf("\n");
}
}
//
/* Combine and Compose *
void cmbcmp(){
  short n,d,fc;
  for(u1=0;u1<(lcnt/2);u1++){
    for(u2=0;u2<lcnt;u2++){
      if(mtc[u1][u2]==5){
        for(n=1;n<26;n++){uflg[n]=0;}
        for(n=1;n<26;n++){
          d=tlu[u1][n]*5+tlu[u2][n]+1;
          nm[n]=d; uflg[d]++;
        }
        fc=0;
        for(n=1;n<26;n++){if(uflg[n]==0){fc++;}}
        if(fc==0){
          if((nm[1]<nm[5])&&(nm[5]<nm[21])&&(nm[1]<nm[25])){solsavep();}}
        }
      }
    }
  }
}
//
/* Save Answers to the Solution List and Print *
void solsavep(){
  short n;
  cnt++;
  anm[0][0]=cnt;
  for(n=1;n<26;n++){anm[0][n]=nm[n];}
  for(n=1;n<26;n++){tn[n]=nm[n]-1; anm[1][n]=tn[n];}; chksms(1);
  for(n=1;n<26;n++){anm[2][n]=tlu[u1][n];}; anm[2][26]=u1;
  for(n=1;n<26;n++){anm[3][n]=tlu[u2][n];}; anm[3][26]=u2;
  pr1ans();
}
//
/* Print the Answer one by one *
void pr1ans(){
  short eu1,eu2;
  short l,l5,n;
  eu1=anm[2][26]; eu2=anm[3][26];
  printf(" /LU%5dH%11dL [Math] |Rw|C1\\P1/P2|",eu1+1,eu2+1);
  printf("%16d#\n",anm[0][0]);
  for(l=0;l<5;l++){l5=l*5;
    printf(" ");
  }
}

```

```

    for(n=1;n<6;n++){printf("%2d",tlu[eu1][l5+n]);}
    printf(" ");
    for(n=1;n<6;n++){printf("%2d",tlu[eu2][l5+n]);}
    printf(" ");
    for(n=1;n<6;n++){printf("%3d",anm[1][l5+n]);}
    printf("%2d|%2d|%2d|%2d| ",cs[1][l+1],cs[1][l+6],cs[1][l+11],cs[1][l+16]);
    for(n=1;n<6;n++){printf("%3d",anm[0][l5+n]);}
    printf("\n");
}
}
//
/** Check Sums *
void chksms(short x){
    cs[x][1]=tn[1]+tn[2]+tn[3]+tn[4]+tn[5];
    cs[x][2]=tn[6]+tn[7]+tn[8]+tn[9]+tn[10];
    cs[x][3]=tn[11]+tn[12]+tn[13]+tn[14]+tn[15];
    cs[x][4]=tn[16]+tn[17]+tn[18]+tn[19]+tn[20];
    cs[x][5]=tn[21]+tn[22]+tn[23]+tn[24]+tn[25];
    cs[x][6]=tn[1]+tn[6]+tn[11]+tn[16]+tn[21];
    cs[x][7]=tn[2]+tn[7]+tn[12]+tn[17]+tn[22];
    cs[x][8]=tn[3]+tn[8]+tn[13]+tn[18]+tn[23];
    cs[x][9]=tn[4]+tn[9]+tn[14]+tn[19]+tn[24];
    cs[x][10]=tn[5]+tn[10]+tn[15]+tn[20]+tn[25];
    //
    cs[x][11]=tn[1]+tn[7]+tn[13]+tn[19]+tn[25];
    cs[x][12]=tn[2]+tn[8]+tn[14]+tn[20]+tn[21];
    cs[x][13]=tn[3]+tn[9]+tn[15]+tn[16]+tn[22];
    cs[x][14]=tn[4]+tn[10]+tn[11]+tn[17]+tn[23];
    cs[x][15]=tn[5]+tn[6]+tn[12]+tn[18]+tn[24];
    cs[x][16]=tn[1]+tn[10]+tn[14]+tn[18]+tn[22];
    cs[x][17]=tn[2]+tn[6]+tn[15]+tn[19]+tn[23];
    cs[x][18]=tn[3]+tn[7]+tn[11]+tn[20]+tn[24];
    cs[x][19]=tn[4]+tn[8]+tn[12]+tn[16]+tn[25];
    cs[x][20]=tn[5]+tn[9]+tn[13]+tn[17]+tn[21];
}
//

```

## Program #5:

```

//
/** 'Complete Euler Squares' of Order 7 for the Simultaneous Type: **
/** Both Self-complementary and Pan-diagonal by 'New Euler's Method' **
/** 'CEulerS7Sml.c' Dictated by Kanji Setsuda in 2003, 2010, 2011, **
/** and on Mar.18, 2012; Revised on Apr.13, 2012 and Recompiled **
/** on Feb.15, 2015 with MacOSX 10.10.2 and Xcode 6.1.1 **
//
#include <stdio.h>
//
short int lcnt, cnt, cnt1, cnt3;
short SC, LSM;
short u1, u2;
short nm[50], uflg[50];
short dn[5][52];
short tlu[193][50];
short mtc[193][193];
short st[3457][52];
short tn[50];
short cs[4][29];
//
short rw1[7], cl1[7], pd1[7], pb1[7];
short rw2[7], cl2[7], pd2[7], pb2[7];
short rw3[7], cl3[7], pd3[7], pb3[7];
short rw4[7], cl4[7], pd4[7], pb4[7];
short rw5[7], cl5[7], pd5[7], pb5[7];

```

```

short rw6[7], cl6[7], pd6[7], pb6[7];
short rw7[7], cl7[7], pd7[7], pb7[7];
//
/** Sub-procedures *
void stp00(void);
void stp01(void), stp02(void), stp03(void), stp04(void);
void stp05(void), stp06(void), stp07(void), stp08(void);
void stp09(void), stp10(void), stp11(void), stp12(void);
void stp13(void), stp14(void), stp15(void), stp16(void);
void stp17(void), stp18(void), stp19(void), stp20(void);
void stp21(void), stp22(void), stp23(void), stp24(void);
void lurecord(void);
void slusort(void), excslu(short x);
void refmatch(void);
void prlunit(void);
void cmbcmp(void), recsol(void);
void solsort(void), excsol(short x);
void solprint(void);
void pr2sol(void);
void prsol(short x);
void chksms(short x);
//
/** Main Program *
int main(){
short n;
printf("\n");
printf("*** 'Complete Euler Type' of Simultaneous Magic Squares of Order 7: **\n");
printf("*** Self-complementary and Pan-diagonal; made by New Euler's Method **\n");
for(n=0;n<50;n++){nm[n]=0;}
for(n=0;n<7;n++){uflg[n]=0;
rw1[n]=0; cl1[n]=0; pd1[n]=0; pb1[n]=0;
rw2[n]=0; cl2[n]=0; pd2[n]=0; pb2[n]=0;
rw3[n]=0; cl3[n]=0; pd3[n]=0; pb3[n]=0;
rw4[n]=0; cl4[n]=0; pd4[n]=0; pb4[n]=0;
rw5[n]=0; cl5[n]=0; pd5[n]=0; pb5[n]=0;
rw6[n]=0; cl6[n]=0; pd6[n]=0; pb6[n]=0;
rw7[n]=0; cl7[n]=0; pd7[n]=0; pb7[n]=0;
}
SC=6; LSM=21; cnt=0; cnt3=0;
stp00(); // Make the Latin Units
slusort(); // Sort the Latin Units
lcnt=cnt;
refmatch(); // Make the Reference Table of Matching
cnt=0;
printf("\n");
printf(" [List of Latin Units]\n");
prlunit();
LSM=175; cnt=0; cnt1=0; cnt3=0;
printf("\n");
printf("*** Complete Euler Type of Simultaneous MS77: Self-Complementary and Pan-diagonal; **\n");
printf(" [Latin Units: /High /Low; Reconstruction: Sol_Number#; Sum_Errors:|Row|Clm\\Pd1/Pd2|\n");
cmbcmp(); // Combine and Compose
solsort(); // Sort the Solutions
solprint(); // Print the Solution List
if(cnt3>0){prsol(cnt3);} // Print The Rest One
printf(" [Total Counts of Solutions([n1=1] All#) = [%d] %d#] [OK!]\n",cnt1,cnt);
printf("\n");
printf("*** Recompiled and Listed by Kanji Setsuda on\n");
printf(" Feb.15, 2015 with MacOSX 10.10.2 and Xcode 6.1.1 **\n");
printf("\n");
return 0;
}
//
/**** Make Program Modules of 'Complete Euler Squares' of Order 7 ***
/**** for the Simultaneous Type: Self-complementary & Pan-diagonal **

```



```

/** Listed by Kanji Setsuda on Apr.13, 2012 with MacOSX & Xcode 4 **
/** Basic Form for the Latin Units of Order 7 **
//
//      .---.---.---.---.---.---.---.---.---.
//  2  3  4  5  6  7| 1| 2| 3| 4| 5| 6| 7| 1  2  3  4  5  6
//      |---+---+---+---+---+---+---|
//  9 10 11 12 13 14| 8| 9|10|11|12|13|14| 8  9 10 11 12 13
//      |---+---+---+---+---+---+---|
// 16 17 18 19 20 21|15|16|17|18|19|20|21|15 16 17 18 19 20
//      |---+---+---+---+---+---+---|
// 23 24 25 26 27 28|22|23|24|25|26|27|28|22 23 24 25 26 27
//      |---+---+---+---+---+---+---|
// 30 31 32 33 34 35|29|30|31|32|33|34|35|29 30 31 32 33 34
//      |---+---+---+---+---+---+---|
// 37 38 39 40 41 42|36|37|38|39|40|41|42|36 37 38 39 40 41
//      |---+---+---+---+---+---+---|
// 44 45 46 47 48 49|43|44|45|46|47|48|49|43 44 45 46 47 48
//      '---'---'---'---'---'---'---'
/** Define Conditions for Rows and Columns: C=21 **
//  n1+n2+n3+n4+n5+n6+n7=C      ... rw1 | n1+n8+n15+n22+n29+n36+n43=C      ... c1
//  n8+n9+n10+n11+n12+n13+n14=C ... rw2 | n2+n9+n16+n23+n30+n37+n44=C      ... c2
//  n15+n16+n17+n18+n19+n20+n21=C ... rw3 | n3+n10+n17+n24+n31+n38+n45=C      ... c3
//  n22+n23+n24+n25+n26+n27+n28=C ... rw4 | n4+n11+n18+n25+n32+n39+n46=C      ... c4
//  n29+n30+n31+n32+n33+n34+n35=C ... rw5 | n5+n12+n19+n26+n33+n40+n47=C      ... c5
//  n36+n37+n38+n39+n40+n41+n42=C ... rw6 | n6+n13+n20+n27+n34+n41+n48=C      ... c6
//  n43+n44+n45+n46+n47+n48+n49=C ... rw7 | n7+n14+n21+n28+n35+n42+n49=C      ... c7
/** Pan-diagonal Conditions of the Object: C=21 **
//  n1+n9+n17+n25+n33+n41+n49=C      ... pd1 | n1+n14+n20+n26+n32+n38+n44=C      ... pd8
//  n2+n10+n18+n26+n34+n42+n43=C      ... pd2 | n2+n8+n21+n27+n33+n39+n45=C      ... pd9
//  n3+n11+n19+n27+n35+n36+n44=C      ... pd3 | n3+n9+n15+n28+n34+n40+n46=C      ... pd10
//  n4+n12+n20+n28+n29+n37+n45=C      ... pd4 | n4+n10+n16+n22+n35+n41+n47=C      ... pd11
//  n5+n13+n21+n22+n30+n38+n46=C      ... pd5 | n5+n11+n17+n23+n29+n42+n48=C      ... pd12
//  n6+n14+n15+n23+n31+n39+n47=C      ... pd6 | n6+n12+n18+n24+n30+n36+n49=C      ... pd13
//  n7+n8+n16+n24+n32+n40+n48=C      ... pd7 | n7+n13+n19+n25+n31+n37+n43=C      ... pd14
/** Self-complementary Conditions of the Object: SC=6 **
//  n1+n49=n2+n48=n3+n47=n4+n46=n5+n45=n6+n44=n7+n43=n8+n42=n9+n41=
//  n10+n40=n11+n39=n12+n38=n13+n37=n14+n36=n15+n35=n16+n34=n17+n33=n18+n32=
//  n19+n31=n20+n30=n21+n29=n22+n28=n23+n27=n24+n26=n25+n25=SC ... scc
/** List-forming Inequality Conditions for the Standard Solutions **
//  n1<n43; n43<n7; and n1<n49;
//
// Level #0:
// Set n25{4,4,1,7}
void stp00(){
short a;
a=3; nm[25]=a;
uflg[a]++; rw4[a]=1; c14[a]=1; pd1[a]=1; pb7[a]=1;
stp01();
uflg[a]--; rw4[a]=0; c14[a]=0; pd1[a]=0; pb7[a]=0;
}
// Level #1:
// Set n1 & n49=SC-N1
void stp01(){
short a,b;
for(a=0;a<7;a++){b=SC-a;
if((uflg[a]<7)&&(uflg[b]<7)){
if((rw1[a]==0)&&(c11[a]==0)&&(pd1[a]==0)&&(pb1[a]==0)){
if((rw7[b]==0)&&(c17[b]==0)&&(pd1[b]==0)&&(pb6[b]==0)){
nm[1]=a; nm[49]=b;
uflg[a]++; rw1[a]=1; c11[a]=1; pd1[a]=1; pb1[a]=1;
uflg[b]++; rw7[b]=1; c17[b]=1; pd1[b]=1; pb6[b]=1;
stp02();
uflg[b]--; rw7[b]=0; c17[b]=0; pd1[b]=0; pb6[b]=0;
uflg[a]--; rw1[a]=0; c11[a]=0; pd1[a]=0; pb1[a]=0;}}}}
}
}

```

```

// Set n9 & n41=SC-N9
void stp02(){
  short a,b;
  for(a=0;a<7;a++){b=SC-a;
    if((uflg[a]<7)&&(uflg[b]<7)){
      if((rw2[a]==0)&&(cl2[a]==0)&&(pd1[a]==0)&&(pb3[a]==0)){
        if((rw6[b]==0)&&(cl6[b]==0)&&(pd1[b]==0)&&(pb4[b]==0)){
          nm[9]=a; nm[41]=b;
          uflg[a]++; rw2[a]=1; cl2[a]=1; pd1[a]=1; pb3[a]=1;
          uflg[b]++; rw6[b]=1; cl6[b]=1; pd1[b]=1; pb4[b]=1;
          stp03();
          uflg[b]--; rw6[b]=0; cl6[b]=0; pd1[b]=0; pb4[b]=0;
          uflg[a]--; rw2[a]=0; cl2[a]=0; pd1[a]=0; pb3[a]=0;}}}}
  }
}
// Set n17 & n33=SC-N17
void stp03(){
  short a,b;
  for(a=0;a<7;a++){b=SC-a;
    if((uflg[a]<7)&&(uflg[b]<7)){
      if((rw3[a]==0)&&(cl3[a]==0)&&(pd1[a]==0)&&(pb5[a]==0)){
        if((rw5[b]==0)&&(cl5[b]==0)&&(pd1[b]==0)&&(pb2[b]==0)){
          nm[17]=a; nm[33]=b;
          uflg[a]++; rw3[a]=1; cl3[a]=1; pd1[a]=1; pb5[a]=1;
          uflg[b]++; rw5[b]=1; cl5[b]=1; pd1[b]=1; pb2[b]=1;
          stp04();
          uflg[b]--; rw5[b]=0; cl5[b]=0; pd1[b]=0; pb2[b]=0;
          uflg[a]--; rw3[a]=0; cl3[a]=0; pd1[a]=0; pb5[a]=0;}}}}
  }
}
// Level #2:
// Set n2 & n48=SC-N2
void stp04(){
  short a,b;
  for(a=6;a>=0;a--){b=SC-a;
    if((uflg[a]<7)&&(uflg[b]<7)){
      if((rw1[a]==0)&&(cl2[a]==0)&&(pd2[a]==0)&&(pb2[a]==0)){
        if((rw7[b]==0)&&(cl6[b]==0)&&(pd7[b]==0)&&(pb5[b]==0)){
          nm[2]=a; nm[48]=b;
          uflg[a]++; rw1[a]=1; cl2[a]=1; pd2[a]=1; pb2[a]=1;
          uflg[b]++; rw7[b]=1; cl6[b]=1; pd7[b]=1; pb5[b]=1;
          stp05();
          uflg[b]--; rw7[b]=0; cl6[b]=0; pd7[b]=0; pb5[b]=0;
          uflg[a]--; rw1[a]=0; cl2[a]=0; pd2[a]=0; pb2[a]=0;}}}}
  }
}
// Set n3 & n47=SC-N3
void stp05(){
  short a,b;
  for(a=6;a>=0;a--){b=SC-a;
    if((uflg[a]<7)&&(uflg[b]<7)){
      if((rw1[a]==0)&&(cl3[a]==0)&&(pd3[a]==0)&&(pb3[a]==0)){
        if((rw7[b]==0)&&(cl5[b]==0)&&(pd6[b]==0)&&(pb4[b]==0)){
          nm[3]=a; nm[47]=b;
          uflg[a]++; rw1[a]=1; cl3[a]=1; pd3[a]=1; pb3[a]=1;
          uflg[b]++; rw7[b]=1; cl5[b]=1; pd6[b]=1; pb4[b]=1;
          stp06();
          uflg[b]--; rw7[b]=0; cl5[b]=0; pd6[b]=0; pb4[b]=0;
          uflg[a]--; rw1[a]=0; cl3[a]=0; pd3[a]=0; pb3[a]=0;}}}}
  }
}
// Set n4 & n46=SC-N4
void stp06(){
  short a,b;
  for(a=6;a>=0;a--){b=SC-a;

```

```

    if((uflg[a]<7)&&(uflg[b]<7)){
        if((rw1[a]==0)&&(cl4[a]==0)&&(pd4[a]==0)&&(pb4[a]==0)){
            if((rw7[b]==0)&&(cl4[b]==0)&&(pd5[b]==0)&&(pb3[b]==0)){
                nm[4]=a; nm[46]=b;
                uflg[a]++; rw1[a]=1; cl4[a]=1; pd4[a]=1; pb4[a]=1;
                uflg[b]++; rw7[b]=1; cl4[b]=1; pd5[b]=1; pb3[b]=1;
                stp07();
                uflg[b]--; rw7[b]=0; cl4[b]=0; pd5[b]=0; pb3[b]=0;
                uflg[a]--; rw1[a]=0; cl4[a]=0; pd4[a]=0; pb4[a]=0;}}}}
    }
}
// Set n5 & n45=SC-N5
void stp07(){
    short a,b;
    for(a=6;a>=0;a--){b=SC-a;
        if((uflg[a]<7)&&(uflg[b]<7)){
            if((rw1[a]==0)&&(cl5[a]==0)&&(pd5[a]==0)&&(pb5[a]==0)){
                if((rw7[b]==0)&&(cl3[b]==0)&&(pd4[b]==0)&&(pb2[b]==0)){
                    nm[5]=a; nm[45]=b;
                    uflg[a]++; rw1[a]=1; cl5[a]=1; pd5[a]=1; pb5[a]=1;
                    uflg[b]++; rw7[b]=1; cl3[b]=1; pd4[b]=1; pb2[b]=1;
                    stp08();
                    uflg[b]--; rw7[b]=0; cl3[b]=0; pd4[b]=0; pb2[b]=0;
                    uflg[a]--; rw1[a]=0; cl5[a]=0; pd5[a]=0; pb5[a]=0;}}}}
        }
    }
// Set n6 & n44=SC-N6
void stp08(){
    short a,b;
    for(a=6;a>=0;a--){b=SC-a;
        if((uflg[a]<7)&&(uflg[b]<7)){
            if((rw1[a]==0)&&(cl6[a]==0)&&(pd6[a]==0)&&(pb6[a]==0)){
                if((rw7[b]==0)&&(cl2[b]==0)&&(pd3[b]==0)&&(pb1[b]==0)){
                    nm[6]=a; nm[44]=b;
                    uflg[a]++; rw1[a]=1; cl6[a]=1; pd6[a]=1; pb6[a]=1;
                    uflg[b]++; rw7[b]=1; cl2[b]=1; pd3[b]=1; pb1[b]=1;
                    stp09();
                    uflg[b]--; rw7[b]=0; cl2[b]=0; pd3[b]=0; pb1[b]=0;
                    uflg[a]--; rw1[a]=0; cl6[a]=0; pd6[a]=0; pb6[a]=0;}}}}
        }
    }
// Set n7 & n43=SC-N7
void stp09(){
    short a,b;
    for(a=6;a>=0;a--){b=SC-a;
        if((uflg[a]<7)&&(uflg[b]<7)){
            if((rw1[a]==0)&&(cl7[a]==0)&&(pd7[a]==0)&&(pb7[a]==0)){
                if((rw7[b]==0)&&(cl1[b]==0)&&(pd2[b]==0)&&(pb7[b]==0)){
                    nm[7]=a; nm[43]=b;
                    uflg[a]++; rw1[a]=1; cl7[a]=1; pd7[a]=1; pb7[a]=1;
                    uflg[b]++; rw7[b]=1; cl1[b]=1; pd2[b]=1; pb7[b]=1;
                    stp10();
                    uflg[b]--; rw7[b]=0; cl1[b]=0; pd2[b]=0; pb7[b]=0;
                    uflg[a]--; rw1[a]=0; cl7[a]=0; pd7[a]=0; pb7[a]=0;}}}}
        }
    }
// Level #3:
// Set n8 & n42=SC-N8
void stp10(){
    short a,b;
    for(a=0;a<7;a++){b=SC-a;
        if((uflg[a]<7)&&(uflg[b]<7)){
            if((rw2[a]==0)&&(cl1[a]==0)&&(pd7[a]==0)&&(pb2[a]==0)){
                if((rw6[b]==0)&&(cl7[b]==0)&&(pd2[b]==0)&&(pb5[b]==0)){
                    nm[8]=a; nm[42]=b;

```

```

        uflg[a]++; rw2[a]=1; c11[a]=1; pd7[a]=1; pb2[a]=1;
        uflg[b]++; rw6[b]=1; c17[b]=1; pd2[b]=1; pb5[b]=1;
        stp11();
        uflg[b]--; rw6[b]=0; c17[b]=0; pd2[b]=0; pb5[b]=0;
        uflg[a]--; rw2[a]=0; c11[a]=0; pd7[a]=0; pb2[a]=0;}}}}
    }
}
// Set n15 & n35=SC-N15
void stp11(){
    short a,b;
    for(a=0;a<7;a++){b=SC-a;
        if((uflg[a]<7)&&(uflg[b]<7)){
            if((rw3[a]==0)&&(c11[a]==0)&&(pd6[a]==0)&&(pb3[a]==0)){
                if((rw5[b]==0)&&(c17[b]==0)&&(pd3[b]==0)&&(pb4[b]==0)){
                    nm[15]=a; nm[35]=b;
                    uflg[a]++; rw3[a]=1; c11[a]=1; pd6[a]=1; pb3[a]=1;
                    uflg[b]++; rw5[b]=1; c17[b]=1; pd3[b]=1; pb4[b]=1;
                    stp12();
                    uflg[b]--; rw5[b]=0; c17[b]=0; pd3[b]=0; pb4[b]=0;
                    uflg[a]--; rw3[a]=0; c11[a]=0; pd6[a]=0; pb3[a]=0;}}}}
        }
    }
}
// Set n22 & n28=SC-N22
void stp12(){
    short a,b;
    for(a=0;a<7;a++){b=SC-a;
        if((uflg[a]<7)&&(uflg[b]<7)){
            if((rw4[a]==0)&&(c11[a]==0)&&(pd5[a]==0)&&(pb4[a]==0)){
                if((rw4[b]==0)&&(c17[b]==0)&&(pd4[b]==0)&&(pb3[b]==0)){
                    nm[22]=a; nm[28]=b;
                    uflg[a]++; rw4[a]=1; c11[a]=1; pd5[a]=1; pb4[a]=1;
                    uflg[b]++; rw4[b]=1; c17[b]=1; pd4[b]=1; pb3[b]=1;
                    stp13();
                    uflg[b]--; rw4[b]=0; c17[b]=0; pd4[b]=0; pb3[b]=0;
                    uflg[a]--; rw4[a]=0; c11[a]=0; pd5[a]=0; pb4[a]=0;}}}}
        }
    }
}
// Set n29 & n21=SC-N29
void stp13(){
    short a,b;
    for(a=0;a<7;a++){b=SC-a;
        if((uflg[a]<7)&&(uflg[b]<7)){
            if((rw5[a]==0)&&(c11[a]==0)&&(pd4[a]==0)&&(pb5[a]==0)){
                if((rw3[b]==0)&&(c17[b]==0)&&(pd5[b]==0)&&(pb2[b]==0)){
                    nm[29]=a; nm[21]=b;
                    uflg[a]++; rw5[a]=1; c11[a]=1; pd4[a]=1; pb5[a]=1;
                    uflg[b]++; rw3[b]=1; c17[b]=1; pd5[b]=1; pb2[b]=1;
                    stp14();
                    uflg[b]--; rw3[b]=0; c17[b]=0; pd5[b]=0; pb2[b]=0;
                    uflg[a]--; rw5[a]=0; c11[a]=0; pd4[a]=0; pb5[a]=0;}}}}
        }
    }
}
// Set n36 & n14=SC-N36
void stp14(){
    short a,b;
    for(a=0;a<7;a++){b=SC-a;
        if((uflg[a]<7)&&(uflg[b]<7)){
            if((rw6[a]==0)&&(c11[a]==0)&&(pd3[a]==0)&&(pb6[a]==0)){
                if((rw2[b]==0)&&(c17[b]==0)&&(pd6[b]==0)&&(pb1[b]==0)){
                    nm[36]=a; nm[14]=b;
                    uflg[a]++; rw6[a]=1; c11[a]=1; pd3[a]=1; pb6[a]=1;
                    uflg[b]++; rw2[b]=1; c17[b]=1; pd6[b]=1; pb1[b]=1;
                    stp15();
                    uflg[b]--; rw2[b]=0; c17[b]=0; pd6[b]=0; pb1[b]=0;
                    uflg[a]--; rw6[a]=0; c11[a]=0; pd3[a]=0; pb6[a]=0;}}}}
        }
    }
}

```

```

}
}
// Level #4:
// Set n13 & n37=SC-N13
void stp15(){
short a,b;
for(a=0;a<7;a++){b=SC-a;
if((uflg[a]<7)&&(uflg[b]<7)){
if((rw2[a]==0)&&(cl6[a]==0)&&(pd5[a]==0)&&(pb7[a]==0)){
if((rw6[b]==0)&&(cl2[b]==0)&&(pd4[b]==0)&&(pb7[b]==0)){
nm[13]=a; nm[37]=b;
uflg[a]++; rw2[a]=1; cl6[a]=1; pd5[a]=1; pb7[a]=1;
uflg[b]++; rw6[b]=1; cl2[b]=1; pd4[b]=1; pb7[b]=1;
stp16();
uflg[b]--; rw6[b]=0; cl2[b]=0; pd4[b]=0; pb7[b]=0;
uflg[a]--; rw2[a]=0; cl6[a]=0; pd5[a]=0; pb7[a]=0;}}}}
}
}
// Set n19 & n31=SC-N19
void stp16(){
short a,b;
for(a=0;a<7;a++){b=SC-a;
if((uflg[a]<7)&&(uflg[b]<7)){
if((rw3[a]==0)&&(cl5[a]==0)&&(pd3[a]==0)&&(pb7[a]==0)){
if((rw5[b]==0)&&(cl3[b]==0)&&(pd6[b]==0)&&(pb7[b]==0)){
nm[19]=a; nm[31]=b;
uflg[a]++; rw3[a]=1; cl5[a]=1; pd3[a]=1; pb7[a]=1;
uflg[b]++; rw5[b]=1; cl3[b]=1; pd6[b]=1; pb7[b]=1;
stp17();
uflg[b]--; rw5[b]=0; cl3[b]=0; pd6[b]=0; pb7[b]=0;
uflg[a]--; rw3[a]=0; cl5[a]=0; pd3[a]=0; pb7[a]=0;}}}}
}
}
// Set n10 & n40=SC-N10
void stp17(){
short a,b;
for(a=0;a<7;a++){b=SC-a;
if((uflg[a]<7)&&(uflg[b]<7)){
if((rw2[a]==0)&&(cl3[a]==0)&&(pd2[a]==0)&&(pb4[a]==0)){
if((rw6[b]==0)&&(cl5[b]==0)&&(pd7[b]==0)&&(pb3[b]==0)){
nm[10]=a; nm[40]=b;
uflg[a]++; rw2[a]=1; cl3[a]=1; pd2[a]=1; pb4[a]=1;
uflg[b]++; rw6[b]=1; cl5[b]=1; pd7[b]=1; pb3[b]=1;
stp18();
uflg[b]--; rw6[b]=0; cl5[b]=0; pd7[b]=0; pb3[b]=0;
uflg[a]--; rw2[a]=0; cl3[a]=0; pd2[a]=0; pb4[a]=0;}}}}
}
}
// Set n11 & n39=SC-N11
void stp18(){
short a,b;
for(a=0;a<7;a++){b=SC-a;
if((uflg[a]<7)&&(uflg[b]<7)){
if((rw2[a]==0)&&(cl4[a]==0)&&(pd3[a]==0)&&(pb5[a]==0)){
if((rw6[b]==0)&&(cl4[b]==0)&&(pd6[b]==0)&&(pb2[b]==0)){
nm[11]=a; nm[39]=b;
uflg[a]++; rw2[a]=1; cl4[a]=1; pd3[a]=1; pb5[a]=1;
uflg[b]++; rw6[b]=1; cl4[b]=1; pd6[b]=1; pb2[b]=1;
stp19();
uflg[b]--; rw6[b]=0; cl4[b]=0; pd6[b]=0; pb2[b]=0;
uflg[a]--; rw2[a]=0; cl4[a]=0; pd3[a]=0; pb5[a]=0;}}}}
}
}
// Set n12 & n38=SC-N12
void stp19(){

```

```

short a,b;
for(a=0;a<7;a++){b=SC-a;
  if((uflg[a]<7)&&(uflg[b]<7)){
    if((rw2[a]==0)&&(cl5[a]==0)&&(pd4[a]==0)&&(pb6[a]==0)){
      if((rw6[b]==0)&&(cl3[b]==0)&&(pd5[b]==0)&&(pb1[b]==0)){
        nm[12]=a; nm[38]=b;
        uflg[a]++; rw2[a]=1; cl5[a]=1; pd4[a]=1; pb6[a]=1;
        uflg[b]++; rw6[b]=1; cl3[b]=1; pd5[b]=1; pb1[b]=1;
        stp20();
        uflg[b]--; rw6[b]=0; cl3[b]=0; pd5[b]=0; pb1[b]=0;
        uflg[a]--; rw2[a]=0; cl5[a]=0; pd4[a]=0; pb6[a]=0;}}}}
}
}
// Set n16 & n34=SC-N16
void stp20(){
  short a,b;
  for(a=0;a<7;a++){b=SC-a;
    if((uflg[a]<7)&&(uflg[b]<7)){
      if((rw3[a]==0)&&(cl2[a]==0)&&(pd7[a]==0)&&(pb4[a]==0)){
        if((rw5[b]==0)&&(cl6[b]==0)&&(pd2[b]==0)&&(pb3[b]==0)){
          nm[16]=a; nm[34]=b;
          uflg[a]++; rw3[a]=1; cl2[a]=1; pd7[a]=1; pb4[a]=1;
          uflg[b]++; rw5[b]=1; cl6[b]=1; pd2[b]=1; pb3[b]=1;
          stp21();
          uflg[b]--; rw5[b]=0; cl6[b]=0; pd2[b]=0; pb3[b]=0;
          uflg[a]--; rw3[a]=0; cl2[a]=0; pd7[a]=0; pb4[a]=0;}}}}
    }
  }
// Set n23 & n27=SC-N23
void stp21(){
  short a,b;
  for(a=0;a<7;a++){b=SC-a;
    if((uflg[a]<7)&&(uflg[b]<7)){
      if((rw4[a]==0)&&(cl2[a]==0)&&(pd6[a]==0)&&(pb5[a]==0)){
        if((rw4[b]==0)&&(cl6[b]==0)&&(pd3[b]==0)&&(pb2[b]==0)){
          nm[23]=a; nm[27]=b;
          uflg[a]++; rw4[a]=1; cl2[a]=1; pd6[a]=1; pb5[a]=1;
          uflg[b]++; rw4[b]=1; cl6[b]=1; pd3[b]=1; pb2[b]=1;
          stp22();
          uflg[b]--; rw4[b]=0; cl6[b]=0; pd3[b]=0; pb2[b]=0;
          uflg[a]--; rw4[a]=0; cl2[a]=0; pd6[a]=0; pb5[a]=0;}}}}
    }
  }
// Set n30 & n20=SC-N30
void stp22(){
  short a,b;
  for(a=0;a<7;a++){b=SC-a;
    if((uflg[a]<7)&&(uflg[b]<7)){
      if((rw5[a]==0)&&(cl2[a]==0)&&(pd5[a]==0)&&(pb6[a]==0)){
        if((rw3[b]==0)&&(cl6[b]==0)&&(pd4[b]==0)&&(pb1[b]==0)){
          nm[30]=a; nm[20]=b;
          uflg[a]++; rw5[a]=1; cl2[a]=1; pd5[a]=1; pb6[a]=1;
          uflg[b]++; rw3[b]=1; cl6[b]=1; pd4[b]=1; pb1[b]=1;
          stp23();
          uflg[b]--; rw3[b]=0; cl6[b]=0; pd4[b]=0; pb1[b]=0;
          uflg[a]--; rw5[a]=0; cl2[a]=0; pd5[a]=0; pb6[a]=0;}}}}
    }
  }
// Set n18 & n32=SC-N18
void stp23(){
  short a,b;
  for(a=0;a<7;a++){b=SC-a;
    if((uflg[a]<7)&&(uflg[b]<7)){
      if((rw3[a]==0)&&(cl4[a]==0)&&(pd2[a]==0)&&(pb6[a]==0)){
        if((rw5[b]==0)&&(cl4[b]==0)&&(pd7[b]==0)&&(pb1[b]==0)){

```

```

        nm[18]=a; nm[32]=b;
        uflg[a]++; rw3[a]=1; c14[a]=1; pd2[a]=1; pb6[a]=1;
        uflg[b]++; rw5[b]=1; c14[b]=1; pd7[b]=1; pb1[b]=1;
        stp24();
        uflg[b]--; rw5[b]=0; c14[b]=0; pd7[b]=0; pb1[b]=0;
        uflg[a]--; rw3[a]=0; c14[a]=0; pd2[a]=0; pb6[a]=0;}}}
    }
}
// Set n24 & n26=SC-N24
void stp24(){
    short a,b;
    for(a=0;a<7;a++){b=SC-a;
        if((uflg[a]<7)&&(uflg[b]<7)){
            if((rw4[a]==0)&&(c13[a]==0)&&(pd7[a]==0)&&(pb6[a]==0)){
                if((rw4[b]==0)&&(c15[b]==0)&&(pd2[b]==0)&&(pb1[b]==0)){
                    nm[24]=a; nm[26]=b;
                    uflg[a]++; rw4[a]=1; c13[a]=1; pd7[a]=1; pb6[a]=1;
                    uflg[b]++; rw4[b]=1; c15[b]=1; pd2[b]=1; pb1[b]=1;
                    lurecord();
                    uflg[b]--; rw4[b]=0; c15[b]=0; pd2[b]=0; pb1[b]=0;
                    uflg[a]--; rw4[a]=0; c13[a]=0; pd7[a]=0; pb6[a]=0;}}}
        }
    }
}
//
/* Record Each of the Latin Units *
void lurecord(){
    short n;
    tlu[cnt][0]=cnt+1;
    for(n=1;n<50;n++){tlu[cnt][n]=nm[n];}
    cnt++;
}
//
/* Sort the Latin Units *
void slusort(){
    short mx,n,f;
    short d1,d2;
    long d3,d4,d5,d6,d7,d8;
    mx=cnt-1;
    do{f=0;
        for(n=0;n<mx;n++){d1=tlu[n][1]; d2=tlu[n+1][1];
            if(d1>d2){excslu(n); f=1;}
            else{d3=tlu[n][9]*50+tlu[n][17]; d4=tlu[n+1][9]*50+tlu[n+1][17];
                if((d1==d2)&&(d3>d4)){excslu(n); f=1;}
                else{
                    d5=((tlu[n][2]*50+tlu[n][3])*50+tlu[n][4])*50+tlu[n][5];
                    d6=((tlu[n+1][2]*50+tlu[n+1][3])*50+tlu[n+1][4])*50+tlu[n+1][5];
                    if((d1==d2)&&(d3==d4)&&(d5<d6)){excslu(n); f=1;}
                    else{
                        d7=((tlu[n][8]*50+tlu[n][15])*50+tlu[n][22])*50+tlu[n][29];
                        d8=((tlu[n+1][8]*50+tlu[n+1][15])*50+tlu[n+1][22])*50+tlu[n+1][29];
                        if((d1==d2)&&(d3==d4)&&(d5==d6)&&(d7<d8)){excslu(n); f=1;}
                    }}}
            }
        mx--;
    }while(f>0);
}
//
void excslu(short x){
    short n,d;
    for(n=1;n<50;n++){d=tlu[x][n]; tlu[x][n]=tlu[x+1][n]; tlu[x+1][n]=d;}
}
//
/* Make the Reference Table of Matching *
void refmatch(){
    short t,l,m,n;

```

```

for(m=0;m<lcnt;m++){
  for(n=0;n<lcnt;n++){
    t=0;
    for(l=1;l<50;l++){if(tlu[m][l]==tlu[n][l]){t++;}}
    mtc[m][n]=t;
    t=0;
    for(l=1;l<50;l++){if(tlu[m][l]+tlu[n][l]==6){t++;}}
    if(mtc[m][n]<t){mtc[m][n]=t;}
  }
}
}
//
/* Print the Latin Units *
void prlunit(){
  short t,l,l7,m,n;
  for(t=0;t<lcnt;t=t+48){
    printf("%8d/%8d/%8d/%8d/%8d/%8d/%8d/%8d\n",
           t+1,t+2,t+3,t+4,t+5,t+6,t+7,t+8);
    for(l=0;l<7;l++){l7=l*7;
      for(m=t;m<(t+8);m++){
        printf(" ");
        for(n=1;n<8;n++){printf("%d",tlu[m][l7+n]);}
      }
      printf("\n");
    }
}
printf(" [Count of Latin Units = %d]\n",lcnt);
printf("\n");
printf(" [Reference Table for the Best Combination]\n");
printf("  0|");
for(n=0;n<24;n++){printf("%3d",n+1);}
printf("\n ---+-----");
for(m=0;m<24;m++){
  printf("\n%3d|",m+1);
  for(n=0;n<24;n++){printf("%3d",mtc[m][n]);}
}
printf("\n\n");
printf("100|");
for(n=168;n<192;n++){printf("%3d",n-99);}
printf("\n ---+-----");
for(m=168;m<192;m++){
  printf("\n%3d|",m-99);
  for(n=168;n<192;n++){printf("%3d",mtc[m][n]);}
}
printf("\n");
}
//
/* Combine and Compose *
void cmbcmp(){
  short n,d,fc;
  for(u1=0;u1<(lcnt/2);u1++){
    for(u2=0;u2<lcnt;u2++){
      if(mtc[u1][u2]==7){
        for(n=1;n<50;n++){uflg[n]=0;}
        for(n=1;n<50;n++){
          d=tlu[u1][n]*7+tlu[u2][n]+1;
          nm[n]=d; uflg[d]++;
        }
        fc=0;
        for(n=1;n<50;n++){if(uflg[n]==1){fc++;}else{break;}}
        if(fc==49){
          if((nm[1]<nm[49])&&(nm[1]<nm[43])&&(nm[43]<nm[7])){recsol();}
        }
      }
    }
  }
}
}

```



```

}
}
//
/** Record Each of the Solutions *
void recsol(){
short n;
cnt++;
st[cnt][0]=cnt;
for(n=1;n<50;n++){st[cnt][n]=nm[n];}
st[cnt][50]=u1+1; st[cnt][51]=u2+1;
if(nm[1]==1){cnt1++;}
}
//
/** Sort the Solutions *
void solsort(){
short mx,n,f;
short d1,d2;
long d3,d4,d5,d6,d7,d8;
mx=cnt-1;
do{f=0;
for(n=0;n<mx;n++){d1=st[n][1]; d2=st[n+1][1];
if(d1>d2){excsol(n); f=1;}
else{d3=st[n][9]*50+st[n][17]; d4=st[n+1][9]*50+st[n+1][17];
if((d1==d2)&&(d3>d4)){excsol(n); f=1;}
else{
d5=((st[n][2]*50+st[n][3])*50+st[n][4])*50+st[n][5];
d6=((st[n+1][2]*50+st[n+1][3])*50+st[n+1][4])*50+st[n+1][5];
if((d1==d2)&&(d3==d4)&&(d5<d6)){excsol(n); f=1;}
else{
d7=((st[n][8]*50+st[n][15])*50+st[n][22])*50+st[n][29];
d8=((st[n+1][8]*50+st[n+1][15])*50+st[n+1][22])*50+st[n+1][29];
if((d1==d2)&&(d3==d4)&&(d5==d6)&&(d7<d8)){excsol(n); f=1;}
}}}}
}
mx--;
}while(f>0);
}
//
void excsol(short x){
short n,d;
for(n=1;n<52;n++){d=st[x][n]; st[x][n]=st[x+1][n]; st[x+1][n]=d;}
}
//
/** Print X Pieces *
void prsol(short x){
short l,l7,m,n;
for(m=0;m<x;m++){
printf("%6d/H%5d/L%6d#",dn[m][50],dn[m][51],dn[m][0]);
if(m+1<x){printf(" ");}
}
printf("\n");
for(l=0;l<7;l++){l7=l*7;
for(m=0;m<x;m++){
printf(" ");
for(n=1;n<8;n++){printf("%3d",dn[m][l7+n]);}
if(m+1<x){printf(" ");}
}
printf("\n");
}
}
//
/** Print the Solution List *
void solprint(){
short mm,n;
short p1,p2,d1,d2;

```

```

d1=0; d2=1;
for(mm=1;mm<=cnt;mm++){
  dn[cnt3][0]=mm;
  for(n=1;n<52;n++){dn[cnt3][n]=st[mm][n];}
  p1=dn[cnt3][1]; p2=dn[cnt3][9];
  if(mm<25){cnt3++; if(cnt3==2){pr2sol(); cnt3=0;}}
  else if((24<mm)&&(mm<385)&&(p2!=d2)){d2=p2; cnt3++; if(cnt3==4){prsol(cnt3); cnt3=0;}}
  else if((mm>=385)&&(p1!=d1)){d1=p1; cnt3++; if(cnt3==4){prsol(cnt3); cnt3=0;}}
}
}
//
/* Print 2 Pieces with Full Information *
void pr2sol(){
short l,l7,m,n;
short lu1,lu2;
for(n=1;n<50;n++){tn[n]=dn[0][n];}; chksms(0);
for(n=1;n<50;n++){tn[n]=dn[1][n];}; chksms(1);
printf("%6d/H%6d/L%21d#IRIC\P/PI%8d/H%6d/L%21d#IRIC\P/PI\n",
        dn[0][50],dn[0][51],dn[0][0],dn[1][50],dn[1][51],dn[1][0]);
for(l=0;l<7;l++){l7=l*7;
  for(m=0;m<2;m++){
    lu1=dn[m][50]-1; lu2=dn[m][51]-1;
    printf(" ");
    for(n=1;n<8;n++){printf("%d",tlu[lu1][l7+n]);}
    printf(" ");
    for(n=1;n<8;n++){printf("%d",tlu[lu2][l7+n]);}
    printf(" ");
    for(n=1;n<8;n++){printf("%3d",dn[m][l7+n]);}
    printf(" |%d|%d|%d|%d",
            cs[m][l+1]-LSM,cs[m][l+8]-LSM,cs[m][l+15]-LSM,cs[m][l+22]-LSM);
    if(m<1){printf(" ");}
  }
  printf("\n");
}
}
//
/* Check Sums of Essential Lines *
void chksms(short x){
cs[x][ 1]=tn[ 1]+tn[ 2]+tn[ 3]+tn[ 4]+tn[ 5]+tn[ 6]+tn[ 7];
cs[x][ 2]=tn[ 8]+tn[ 9]+tn[10]+tn[11]+tn[12]+tn[13]+tn[14];
cs[x][ 3]=tn[15]+tn[16]+tn[17]+tn[18]+tn[19]+tn[20]+tn[21];
cs[x][ 4]=tn[22]+tn[23]+tn[24]+tn[25]+tn[26]+tn[27]+tn[28];
cs[x][ 5]=tn[29]+tn[30]+tn[31]+tn[32]+tn[33]+tn[34]+tn[35];
cs[x][ 6]=tn[36]+tn[37]+tn[38]+tn[39]+tn[40]+tn[41]+tn[42];
cs[x][ 7]=tn[43]+tn[44]+tn[45]+tn[46]+tn[47]+tn[48]+tn[49];
//
cs[x][ 8]=tn[ 1]+tn[ 8]+tn[15]+tn[22]+tn[29]+tn[36]+tn[43];
cs[x][ 9]=tn[ 2]+tn[ 9]+tn[16]+tn[23]+tn[30]+tn[37]+tn[44];
cs[x][10]=tn[ 3]+tn[10]+tn[17]+tn[24]+tn[31]+tn[38]+tn[45];
cs[x][11]=tn[ 4]+tn[11]+tn[18]+tn[25]+tn[32]+tn[39]+tn[46];
cs[x][12]=tn[ 5]+tn[12]+tn[19]+tn[26]+tn[33]+tn[40]+tn[47];
cs[x][13]=tn[ 6]+tn[13]+tn[20]+tn[27]+tn[34]+tn[41]+tn[48];
cs[x][14]=tn[ 7]+tn[14]+tn[21]+tn[28]+tn[35]+tn[42]+tn[49];
//
cs[x][15]=tn[ 1]+tn[ 9]+tn[17]+tn[25]+tn[33]+tn[41]+tn[49];
cs[x][16]=tn[ 2]+tn[10]+tn[18]+tn[26]+tn[34]+tn[42]+tn[43];
cs[x][17]=tn[ 3]+tn[11]+tn[19]+tn[27]+tn[35]+tn[36]+tn[44];
cs[x][18]=tn[ 4]+tn[12]+tn[20]+tn[28]+tn[29]+tn[37]+tn[45];
cs[x][19]=tn[ 5]+tn[13]+tn[21]+tn[22]+tn[30]+tn[38]+tn[46];
cs[x][20]=tn[ 6]+tn[14]+tn[15]+tn[23]+tn[31]+tn[39]+tn[47];
cs[x][21]=tn[ 7]+tn[ 8]+tn[16]+tn[24]+tn[32]+tn[40]+tn[48];
//
cs[x][22]=tn[ 1]+tn[14]+tn[20]+tn[26]+tn[32]+tn[38]+tn[44];
cs[x][23]=tn[ 2]+tn[ 8]+tn[21]+tn[27]+tn[33]+tn[39]+tn[45];
cs[x][24]=tn[ 3]+tn[ 9]+tn[15]+tn[28]+tn[34]+tn[40]+tn[46];

```

```

cs[x][25]=tn[ 4]+tn[10]+tn[16]+tn[22]+tn[35]+tn[41]+tn[47];
cs[x][26]=tn[ 5]+tn[11]+tn[17]+tn[23]+tn[29]+tn[42]+tn[48];
cs[x][27]=tn[ 6]+tn[12]+tn[18]+tn[24]+tn[30]+tn[36]+tn[49];
cs[x][28]=tn[ 7]+tn[13]+tn[19]+tn[25]+tn[31]+tn[37]+tn[43];
}
//

```

## Program #6:

```

//
/** Compose 'Complete Euler Squares' of Order 7 for **
/** the Pan-diagonal Type by the New Euler's Method **
/** 'CEulerS77PD.c': Dictated by Kanji Setsuda on Apr.4, 2010, **
/** Dec.11, 2010, Feb. 2, 2011, Apr.14, 2012 and Mar.25, 2013 **
/** Recompiled on Mar.14, 2015 with MacOSX 10.10.2 & Xcode 6.2 **
//
#include <stdio.h>
//
long int cnt, cnt1;
long cntn[5];
short lcnt, cnt2, cnt3;
short u1,u2;
short nm[50], uflg[50], tn[50];
short dn[5][52];
short LSM;
short tlu[20161][50];
short cs[3][29];
//
short rw1[7], cl1[7], pd1[7], pb1[7];
short rw2[7], cl2[7], pd2[7], pb2[7];
short rw3[7], cl3[7], pd3[7], pb3[7];
short rw4[7], cl4[7], pd4[7], pb4[7];
short rw5[7], cl5[7], pd5[7], pb5[7];
short rw6[7], cl6[7], pd6[7], pb6[7];
short rw7[7], cl7[7], pd7[7], pb7[7];
//
void stp01(void), stp02(void), stp03(void), stp04(void);
void stp05(void), stp06(void), stp07(void), stp08(void);
void stp09(void), stp10(void), stp11(void), stp12(void);
void stp13(void), stp14(void), stp15(void), stp16(void);
void stp17(void), stp18(void), stp19(void), stp20(void);
void stp21(void), stp22(void), stp23(void), stp24(void);
void stp25(void), stp26(void), stp27(void), stp28(void);
void stp29(void), stp30(void), stp31(void), stp32(void);
void stp33(void), stp34(void), stp35(void), stp36(void);
void stp37(void), stp38(void), stp39(void), stp40(void);
void stp41(void), stp42(void), stp43(void), stp44(void);
void stp45(void), stp46(void), stp47(void), stp48(void);
void stp49(void), lurecord(void);
void prlunit(void);
short chkmtc(short x, short y);
void cmbcmp(void);
void prans(void);
void pr2sol(void), prsol(short x);
void chksms(short x);
//
/** Main Program *
int main(){
short n;
printf("\n");
printf("** Compose 'Complete Euler Squares' of Order 7 for **\n");
printf("** the Pan-diagonal MS77 by the New Euler's Method **\n");
for(n=0;n<50;n++){nm[n]=0;}

```

```

for(n=0;n<7;n++){uflg[n]=0;
  rw1[n]=0; c11[n]=0; pd1[n]=0; pb1[n]=0;
  rw2[n]=0; c12[n]=0; pd2[n]=0; pb2[n]=0;
  rw3[n]=0; c13[n]=0; pd3[n]=0; pb3[n]=0;
  rw4[n]=0; c14[n]=0; pd4[n]=0; pb4[n]=0;
  rw5[n]=0; c15[n]=0; pd5[n]=0; pb5[n]=0;
  rw6[n]=0; c16[n]=0; pd6[n]=0; pb6[n]=0;
  rw7[n]=0; c17[n]=0; pd7[n]=0; pb7[n]=0;
}
lcnt=0; cnt=0; cnt3=0;
stp01(); // Make the Latin-Units
lcnt=cnt;
printf(" [List of Latin Units]\n");
prlunit(); // Print the Latin-Units
LSM=175; cnt=0; cnt1=0; cnt3=0;
printf("\n");
printf("** List of the 'Complete Euler Squares' of Order 7: Pan-Diagonal Type **\n");
printf(" [Solutions when n1=1: Used_Units/H/L Sol_Nmb# Check_Sum-Errors|I\|V|I]\n");
cmbcmp(); // Combine, Compose and Print
if(cnt3>0){prsol(cnt3);} // Print The Rest One
printf(" . . . . .\n");
printf(" [Count(n1=1/All) = %ld/%ld] OK!\n",cnt1,cnt);
printf("\n");
printf(" ** Recompiled and Listed by Kanji Setsuda on\n");
printf(" Mar.14, 2015 with MacOSX 10.10.2 and Xcode 6.2 **\n");
printf("\n");
return 0;
}
//
/** Program Modules of 'Complete Euler Squares' of Order 7 for the Pan-diagonal Type **
/** Dictated by Kanji Setsuda on Apr.14, 2012 with Mac Xcode 4.3.2 **
//
// Level #1:
// Set n1{1,1,1,1}
void stp01(){
  short a;
  for(a=0;a<7;a++){
    if(uflg[a]<7){
      if((rw1[a]==0)&&(c11[a]==0)&&(pd1[a]==0)&&(pb1[a]==0)){
        nm[1]=a;
        uflg[a]++; rw1[a]=1; c11[a]=1; pd1[a]=1; pb1[a]=1;
        stp02();
        rw1[a]=0; c11[a]=0; pd1[a]=0; pb1[a]=0; uflg[a]--;}}
  }
}
// Set n49{7,7,1,6}
void stp02(){
  short a;
  for(a=6;a>=0;a--){
    if(uflg[a]<7){
      if((rw7[a]==0)&&(c17[a]==0)&&(pd1[a]==0)&&(pb6[a]==0)){
        nm[49]=a;
        uflg[a]++; rw7[a]=1; c17[a]=1; pd1[a]=1; pb6[a]=1;
        stp03();
        rw7[a]=0; c17[a]=0; pd1[a]=0; pb6[a]=0; uflg[a]--;}}
  }
}
// Set n9{2,2,1,3}
void stp03(){
  short a;
  for(a=0;a<7;a++){
    if(uflg[a]<7){
      if((rw2[a]==0)&&(c12[a]==0)&&(pd1[a]==0)&&(pb3[a]==0)){
        nm[9]=a;
        uflg[a]++; rw2[a]=1; c12[a]=1; pd1[a]=1; pb3[a]=1;

```

```

        stp04();
        rw2[a]=0; cl2[a]=0; pd1[a]=0; pb3[a]=0; uflg[a]--;}}
    }
}
// Set n41{6,6,1,4}
void stp04(){
    short a;
    for(a=6;a>=0;a--){
        if(uflg[a]<7){
            if((rw6[a]==0)&&(cl6[a]==0)&&(pd1[a]==0)&&(pb4[a]==0)){
                nm[41]=a;
                uflg[a]++; rw6[a]=1; cl6[a]=1; pd1[a]=1; pb4[a]=1;
                stp05();
                rw6[a]=0; cl6[a]=0; pd1[a]=0; pb4[a]=0; uflg[a]--;}}
        }
    }
}
// Set n17{3,3,1,5}
void stp05(){
    short a;
    for(a=0;a<7;a++){
        if(uflg[a]<7){
            if((rw3[a]==0)&&(cl3[a]==0)&&(pd1[a]==0)&&(pb5[a]==0)){
                nm[17]=a;
                uflg[a]++; rw3[a]=1; cl3[a]=1; pd1[a]=1; pb5[a]=1;
                stp06();
                rw3[a]=0; cl3[a]=0; pd1[a]=0; pb5[a]=0; uflg[a]--;}}
        }
    }
}
// Set n33{5,5,1,2}
void stp06(){
    short a;
    for(a=6;a>=0;a--){
        if(uflg[a]<7){
            if((rw5[a]==0)&&(cl5[a]==0)&&(pd1[a]==0)&&(pb2[a]==0)){
                nm[33]=a;
                uflg[a]++; rw5[a]=1; cl5[a]=1; pd1[a]=1; pb2[a]=1;
                stp07();
                rw5[a]=0; cl5[a]=0; pd1[a]=0; pb2[a]=0; uflg[a]--;}}
        }
    }
}
// Set n25{4,4,1,7}
void stp07(){
    short a;
    for(a=0;a<7;a++){
        if(uflg[a]<7){
            if((rw4[a]==0)&&(cl4[a]==0)&&(pd1[a]==0)&&(pb7[a]==0)){
                nm[25]=a;
                uflg[a]++; rw4[a]=1; cl4[a]=1; pd1[a]=1; pb7[a]=1;
                stp08();
                rw4[a]=0; cl4[a]=0; pd1[a]=0; pb7[a]=0; uflg[a]--;}}
        }
    }
}
// Level #2:
// Set n2{1,2,2,2}
void stp08(){
    short a;
    for(a=6;a>=0;a--){
        if(uflg[a]<7){
            if((rw1[a]==0)&&(cl2[a]==0)&&(pd2[a]==0)&&(pb2[a]==0)){
                nm[2]=a;
                uflg[a]++; rw1[a]=1; cl2[a]=1; pd2[a]=1; pb2[a]=1;
                stp09();
                rw1[a]=0; cl2[a]=0; pd2[a]=0; pb2[a]=0; uflg[a]--;}}
        }
    }
}
}

```

```

// Set n48{7,6,7,5}
void stp09(){
  short a;
  for(a=0;a<7;a++){
    if(uflg[a]<7){
      if((rw7[a]==0)&&(cl6[a]==0)&&(pd7[a]==0)&&(pb5[a]==0)){
        nm[48]=a;
        uflg[a]++; rw7[a]=1; cl6[a]=1; pd7[a]=1; pb5[a]=1;
        stp10();
        rw7[a]=0; cl6[a]=0; pd7[a]=0; pb5[a]=0; uflg[a]--;}}
    }
}
// Set n3{1,3,3,3}
void stp10(){
  short a;
  for(a=6;a>=0;a--){
    if(uflg[a]<7){
      if((rw1[a]==0)&&(cl3[a]==0)&&(pd3[a]==0)&&(pb3[a]==0)){
        nm[3]=a;
        uflg[a]++; rw1[a]=1; cl3[a]=1; pd3[a]=1; pb3[a]=1;
        stp11();
        rw1[a]=0; cl3[a]=0; pd3[a]=0; pb3[a]=0; uflg[a]--;}}
    }
}
// Set n47{7,5,6,4}
void stp11(){
  short a;
  for(a=0;a<7;a++){
    if(uflg[a]<7){
      if((rw7[a]==0)&&(cl5[a]==0)&&(pd6[a]==0)&&(pb4[a]==0)){
        nm[47]=a;
        uflg[a]++; rw7[a]=1; cl5[a]=1; pd6[a]=1; pb4[a]=1;
        stp12();
        rw7[a]=0; cl5[a]=0; pd6[a]=0; pb4[a]=0; uflg[a]--;}}
    }
}
// Set n4{1,4,4,4}
void stp12(){
  short a;
  for(a=6;a>=0;a--){
    if(uflg[a]<7){
      if((rw1[a]==0)&&(cl4[a]==0)&&(pd4[a]==0)&&(pb4[a]==0)){
        nm[4]=a;
        uflg[a]++; rw1[a]=1; cl4[a]=1; pd4[a]=1; pb4[a]=1;
        stp13();
        rw1[a]=0; cl4[a]=0; pd4[a]=0; pb4[a]=0; uflg[a]--;}}
    }
}
// Set n46{7,4,5,3}
void stp13(){
  short a;
  for(a=0;a<7;a++){
    if(uflg[a]<7){
      if((rw7[a]==0)&&(cl4[a]==0)&&(pd5[a]==0)&&(pb3[a]==0)){
        nm[46]=a;
        uflg[a]++; rw7[a]=1; cl4[a]=1; pd5[a]=1; pb3[a]=1;
        stp14();
        rw7[a]=0; cl4[a]=0; pd5[a]=0; pb3[a]=0; uflg[a]--;}}
    }
}
// Set n5{1,5,5,5}
void stp14(){
  short a;
  for(a=6;a>=0;a--){
    if(uflg[a]<7){

```

```

        if((rw1[a]==0)&&(cl5[a]==0)&&(pd5[a]==0)&&(pb5[a]==0)){
            nm[5]=a;
            uflg[a]++; rw1[a]=1; cl5[a]=1; pd5[a]=1; pb5[a]=1;
            stp15();
            rw1[a]=0; cl5[a]=0; pd5[a]=0; pb5[a]=0; uflg[a]--;}}
    }
}
// Set n45{7,3,4,2}
void stp15(){
    short a;
    for(a=0;a<7;a++){
        if(uflg[a]<7){
            if((rw7[a]==0)&&(cl3[a]==0)&&(pd4[a]==0)&&(pb2[a]==0)){
                nm[45]=a;
                uflg[a]++; rw7[a]=1; cl3[a]=1; pd4[a]=1; pb2[a]=1;
                stp16();
                rw7[a]=0; cl3[a]=0; pd4[a]=0; pb2[a]=0; uflg[a]--;}}
        }
    }
}
// Set n6{1,6,6,6}
void stp16(){
    short a;
    for(a=6;a>=0;a--){
        if(uflg[a]<7){
            if((rw1[a]==0)&&(cl6[a]==0)&&(pd6[a]==0)&&(pb6[a]==0)){
                nm[6]=a;
                uflg[a]++; rw1[a]=1; cl6[a]=1; pd6[a]=1; pb6[a]=1;
                stp17();
                rw1[a]=0; cl6[a]=0; pd6[a]=0; pb6[a]=0; uflg[a]--;}}
        }
    }
}
// Set n44{7,2,3,1}
void stp17(){
    short a;
    for(a=0;a<7;a++){
        if(uflg[a]<7){
            if((rw7[a]==0)&&(cl2[a]==0)&&(pd3[a]==0)&&(pb1[a]==0)){
                nm[44]=a;
                uflg[a]++; rw7[a]=1; cl2[a]=1; pd3[a]=1; pb1[a]=1;
                stp18();
                rw7[a]=0; cl2[a]=0; pd3[a]=0; pb1[a]=0; uflg[a]--;}}
        }
    }
}
// Set n7{1,7,7,7}
void stp18(){
    short a;
    for(a=6;a>=0;a--){
        if(uflg[a]<7){
            if((rw1[a]==0)&&(cl7[a]==0)&&(pd7[a]==0)&&(pb7[a]==0)){
                nm[7]=a;
                uflg[a]++; rw1[a]=1; cl7[a]=1; pd7[a]=1; pb7[a]=1;
                stp19();
                rw1[a]=0; cl7[a]=0; pd7[a]=0; pb7[a]=0; uflg[a]--;}}
        }
    }
}
// Set n43{7,1,2,7}
void stp19(){
    short a;
    for(a=0;a<7;a++){
        if(uflg[a]<7){
            if((rw7[a]==0)&&(cl1[a]==0)&&(pd2[a]==0)&&(pb7[a]==0)){
                nm[43]=a;
                uflg[a]++; rw7[a]=1; cl1[a]=1; pd2[a]=1; pb7[a]=1;
                stp20();
                rw7[a]=0; cl1[a]=0; pd2[a]=0; pb7[a]=0; uflg[a]--;}}
        }
    }
}

```

```

}
}
// Level #3:
// Set n8{2,1,7,2}
void stp20(){
  short a;
  for(a=0;a<7;a++){
    if(uflg[a]<7){
      if((rw2[a]==0)&&(cl1[a]==0)&&(pd7[a]==0)&&(pb2[a]==0)){
        nm[8]=a;
        uflg[a]++; rw2[a]=1; cl1[a]=1; pd7[a]=1; pb2[a]=1;
        stp21();
        rw2[a]=0; cl1[a]=0; pd7[a]=0; pb2[a]=0; uflg[a]--;}}
    }
}
// Set n42{6,7,2,5}
void stp21(){
  short a;
  for(a=6;a>=0;a--){
    if(uflg[a]<7){
      if((rw6[a]==0)&&(cl7[a]==0)&&(pd2[a]==0)&&(pb5[a]==0)){
        nm[42]=a;
        uflg[a]++; rw6[a]=1; cl7[a]=1; pd2[a]=1; pb5[a]=1;
        stp22();
        rw6[a]=0; cl7[a]=0; pd2[a]=0; pb5[a]=0; uflg[a]--;}}
    }
}
// Set n15{3,1,6,3}
void stp22(){
  short a;
  for(a=0;a<7;a++){
    if(uflg[a]<7){
      if((rw3[a]==0)&&(cl1[a]==0)&&(pd6[a]==0)&&(pb3[a]==0)){
        nm[15]=a;
        uflg[a]++; rw3[a]=1; cl1[a]=1; pd6[a]=1; pb3[a]=1;
        stp23();
        rw3[a]=0; cl1[a]=0; pd6[a]=0; pb3[a]=0; uflg[a]--;}}
    }
}
// Set n35{5,7,3,4}
void stp23(){
  short a;
  for(a=6;a>=0;a--){
    if(uflg[a]<7){
      if((rw5[a]==0)&&(cl7[a]==0)&&(pd3[a]==0)&&(pb4[a]==0)){
        nm[35]=a;
        uflg[a]++; rw5[a]=1; cl7[a]=1; pd3[a]=1; pb4[a]=1;
        stp24();
        rw5[a]=0; cl7[a]=0; pd3[a]=0; pb4[a]=0; uflg[a]--;}}
    }
}
// Set n22{4,1,5,4}
void stp24(){
  short a;
  for(a=0;a<7;a++){
    if(uflg[a]<7){
      if((rw4[a]==0)&&(cl1[a]==0)&&(pd5[a]==0)&&(pb4[a]==0)){
        nm[22]=a;
        uflg[a]++; rw4[a]=1; cl1[a]=1; pd5[a]=1; pb4[a]=1;
        stp25();
        rw4[a]=0; cl1[a]=0; pd5[a]=0; pb4[a]=0; uflg[a]--;}}
    }
}
// Set n28{4,7,4,3}
void stp25(){

```



```

short a;
for(a=6;a>=0;a--){
    if(uflg[a]<7){
        if((rw4[a]==0)&&(cl7[a]==0)&&(pd4[a]==0)&&(pb3[a]==0)){
            nm[28]=a;
            uflg[a]++; rw4[a]=1; cl7[a]=1; pd4[a]=1; pb3[a]=1;
            stp26();
            rw4[a]=0; cl7[a]=0; pd4[a]=0; pb3[a]=0; uflg[a]--;}}
    }
}
// Set n29{5,1,4,5}
void stp26(){
    short a;
    for(a=0;a<7;a++){
        if(uflg[a]<7){
            if((rw5[a]==0)&&(cl1[a]==0)&&(pd4[a]==0)&&(pb5[a]==0)){
                nm[29]=a;
                uflg[a]++; rw5[a]=1; cl1[a]=1; pd4[a]=1; pb5[a]=1;
                stp27();
                rw5[a]=0; cl1[a]=0; pd4[a]=0; pb5[a]=0; uflg[a]--;}}
        }
    }
}
// Set n21{3,7,5,2}
void stp27(){
    short a;
    for(a=6;a>=0;a--){
        if(uflg[a]<7){
            if((rw3[a]==0)&&(cl7[a]==0)&&(pd5[a]==0)&&(pb2[a]==0)){
                nm[21]=a;
                uflg[a]++; rw3[a]=1; cl7[a]=1; pd5[a]=1; pb2[a]=1;
                stp28();
                rw3[a]=0; cl7[a]=0; pd5[a]=0; pb2[a]=0; uflg[a]--;}}
        }
    }
}
// Set n36{6,1,3,6}
void stp28(){
    short a;
    for(a=0;a<7;a++){
        if(uflg[a]<7){
            if((rw6[a]==0)&&(cl1[a]==0)&&(pd3[a]==0)&&(pb6[a]==0)){
                nm[36]=a;
                uflg[a]++; rw6[a]=1; cl1[a]=1; pd3[a]=1; pb6[a]=1;
                stp29();
                rw6[a]=0; cl1[a]=0; pd3[a]=0; pb6[a]=0; uflg[a]--;}}
        }
    }
}
// Set n14{2,7,6,1}
void stp29(){
    short a;
    for(a=6;a>=0;a--){
        if(uflg[a]<7){
            if((rw2[a]==0)&&(cl7[a]==0)&&(pd6[a]==0)&&(pb1[a]==0)){
                nm[14]=a;
                uflg[a]++; rw2[a]=1; cl7[a]=1; pd6[a]=1; pb1[a]=1;
                stp30();
                rw2[a]=0; cl7[a]=0; pd6[a]=0; pb1[a]=0; uflg[a]--;}}
        }
    }
}
// Level #4:
// Set n13{2,6,5,7}
void stp30(){
    short a;
    for(a=0;a<7;a++){
        if(uflg[a]<7){
            if((rw2[a]==0)&&(cl6[a]==0)&&(pd5[a]==0)&&(pb7[a]==0)){

```

```

        nm[13]=a;
        uflg[a]++; rw2[a]=1; c16[a]=1; pd5[a]=1; pb7[a]=1;
        stp31();
        rw2[a]=0; c16[a]=0; pd5[a]=0; pb7[a]=0; uflg[a]--;}}
    }
}
// Set n37{6,2,4,7}
void stp31(){
    short a;
    for(a=6;a>=0;a--){
        if(uflg[a]<7){
            if((rw6[a]==0)&&(c12[a]==0)&&(pd4[a]==0)&&(pb7[a]==0)){
                nm[37]=a;
                uflg[a]++; rw6[a]=1; c12[a]=1; pd4[a]=1; pb7[a]=1;
                stp32();
                rw6[a]=0; c12[a]=0; pd4[a]=0; pb7[a]=0; uflg[a]--;}}
        }
    }
}
// Set n19{3,5,3,7}
void stp32(){
    short a;
    for(a=0;a<7;a++){
        if(uflg[a]<7){
            if((rw3[a]==0)&&(c15[a]==0)&&(pd3[a]==0)&&(pb7[a]==0)){
                nm[19]=a;
                uflg[a]++; rw3[a]=1; c15[a]=1; pd3[a]=1; pb7[a]=1;
                stp33();
                rw3[a]=0; c15[a]=0; pd3[a]=0; pb7[a]=0; uflg[a]--;}}
        }
    }
}
// Set n31{5,3,6,7}
void stp33(){
    short a;
    for(a=6;a>=0;a--){
        if(uflg[a]<7){
            if((rw5[a]==0)&&(c13[a]==0)&&(pd6[a]==0)&&(pb7[a]==0)){
                nm[31]=a;
                uflg[a]++; rw5[a]=1; c13[a]=1; pd6[a]=1; pb7[a]=1;
                stp34();
                rw5[a]=0; c13[a]=0; pd6[a]=0; pb7[a]=0; uflg[a]--;}}
        }
    }
}
// Set n10{2,3,2,4}
void stp34(){
    short a;
    for(a=0;a<7;a++){
        if(uflg[a]<7){
            if((rw2[a]==0)&&(c13[a]==0)&&(pd2[a]==0)&&(pb4[a]==0)){
                nm[10]=a;
                uflg[a]++; rw2[a]=1; c13[a]=1; pd2[a]=1; pb4[a]=1;
                stp35();
                rw2[a]=0; c13[a]=0; pd2[a]=0; pb4[a]=0; uflg[a]--;}}
        }
    }
}
// Set n40{6,5,7,3}
void stp35(){
    short a;
    for(a=6;a>=0;a--){
        if(uflg[a]<7){
            if((rw6[a]==0)&&(c15[a]==0)&&(pd7[a]==0)&&(pb3[a]==0)){
                nm[40]=a;
                uflg[a]++; rw6[a]=1; c15[a]=1; pd7[a]=1; pb3[a]=1;
                stp36();
                rw6[a]=0; c15[a]=0; pd7[a]=0; pb3[a]=0; uflg[a]--;}}
        }
    }
}

```

```

}
// Set n11{2,4,3,5}
void stp36(){
  short a;
  for(a=0;a<7;a++){
    if(uflg[a]<7){
      if((rw2[a]==0)&&(cl4[a]==0)&&(pd3[a]==0)&&(pb5[a]==0)){
        nm[11]=a;
        uflg[a]++; rw2[a]=1; cl4[a]=1; pd3[a]=1; pb5[a]=1;
        stp37();
        rw2[a]=0; cl4[a]=0; pd3[a]=0; pb5[a]=0; uflg[a]--;}}
    }
}
// Set n39{6,4,6,2}
void stp37(){
  short a;
  for(a=6;a>=0;a--){
    if(uflg[a]<7){
      if((rw6[a]==0)&&(cl4[a]==0)&&(pd6[a]==0)&&(pb2[a]==0)){
        nm[39]=a;
        uflg[a]++; rw6[a]=1; cl4[a]=1; pd6[a]=1; pb2[a]=1;
        stp38();
        rw6[a]=0; cl4[a]=0; pd6[a]=0; pb2[a]=0; uflg[a]--;}}
    }
}
// Set n12{2,5,4,6}
void stp38(){
  short a;
  for(a=0;a<7;a++){
    if(uflg[a]<7){
      if((rw2[a]==0)&&(cl5[a]==0)&&(pd4[a]==0)&&(pb6[a]==0)){
        nm[12]=a;
        uflg[a]++; rw2[a]=1; cl5[a]=1; pd4[a]=1; pb6[a]=1;
        stp39();
        rw2[a]=0; cl5[a]=0; pd4[a]=0; pb6[a]=0; uflg[a]--;}}
    }
}
// Set n38{6,3,5,1}
void stp39(){
  short a;
  for(a=6;a>=0;a--){
    if(uflg[a]<7){
      if((rw6[a]==0)&&(cl3[a]==0)&&(pd5[a]==0)&&(pb1[a]==0)){
        nm[38]=a;
        uflg[a]++; rw6[a]=1; cl3[a]=1; pd5[a]=1; pb1[a]=1;
        stp40();
        rw6[a]=0; cl3[a]=0; pd5[a]=0; pb1[a]=0; uflg[a]--;}}
    }
}
// Level #5:
// Set n16{3,2,7,4}
void stp40(){
  short a;
  for(a=0;a<7;a++){
    if(uflg[a]<7){
      if((rw3[a]==0)&&(cl2[a]==0)&&(pd7[a]==0)&&(pb4[a]==0)){
        nm[16]=a;
        uflg[a]++; rw3[a]=1; cl2[a]=1; pd7[a]=1; pb4[a]=1;
        stp41();
        rw3[a]=0; cl2[a]=0; pd7[a]=0; pb4[a]=0; uflg[a]--;}}
    }
}
// Set n34{5,6,2,3}
void stp41(){
  short a;

```

```

for(a=6;a>=0;a--){
  if(uflg[a]<7){
    if((rw5[a]==0)&&(cl6[a]==0)&&(pd2[a]==0)&&(pb3[a]==0)){
      nm[34]=a;
      uflg[a]++; rw5[a]=1; cl6[a]=1; pd2[a]=1; pb3[a]=1;
      stp42();
      rw5[a]=0; cl6[a]=0; pd2[a]=0; pb3[a]=0; uflg[a]--;}
  }
}
// Set n23{4,2,6,5}
void stp42(){
  short a;
  for(a=0;a<7;a++){
    if(uflg[a]<7){
      if((rw4[a]==0)&&(cl2[a]==0)&&(pd6[a]==0)&&(pb5[a]==0)){
        nm[23]=a;
        uflg[a]++; rw4[a]=1; cl2[a]=1; pd6[a]=1; pb5[a]=1;
        stp43();
        rw4[a]=0; cl2[a]=0; pd6[a]=0; pb5[a]=0; uflg[a]--;}
    }
  }
}
// Set n27{4,6,3,2}
void stp43(){
  short a;
  for(a=6;a>=0;a--){
    if(uflg[a]<7){
      if((rw4[a]==0)&&(cl6[a]==0)&&(pd3[a]==0)&&(pb2[a]==0)){
        nm[27]=a;
        uflg[a]++; rw4[a]=1; cl6[a]=1; pd3[a]=1; pb2[a]=1;
        stp44();
        rw4[a]=0; cl6[a]=0; pd3[a]=0; pb2[a]=0; uflg[a]--;}
    }
  }
}
// Set n30{5,2,5,6}
void stp44(){
  short a;
  for(a=0;a<7;a++){
    if(uflg[a]<7){
      if((rw5[a]==0)&&(cl2[a]==0)&&(pd5[a]==0)&&(pb6[a]==0)){
        nm[30]=a;
        uflg[a]++; rw5[a]=1; cl2[a]=1; pd5[a]=1; pb6[a]=1;
        stp45();
        rw5[a]=0; cl2[a]=0; pd5[a]=0; pb6[a]=0; uflg[a]--;}
    }
  }
}
// Set n20{3,6,4,1}
void stp45(){
  short a;
  for(a=6;a>=0;a--){
    if(uflg[a]<7){
      if((rw3[a]==0)&&(cl6[a]==0)&&(pd4[a]==0)&&(pb1[a]==0)){
        nm[20]=a;
        uflg[a]++; rw3[a]=1; cl6[a]=1; pd4[a]=1; pb1[a]=1;
        stp46();
        rw3[a]=0; cl6[a]=0; pd4[a]=0; pb1[a]=0; uflg[a]--;}
    }
  }
}
// Set n18{3,4,2,6}
void stp46(){
  short a;
  for(a=0;a<7;a++){
    if(uflg[a]<7){
      if((rw3[a]==0)&&(cl4[a]==0)&&(pd2[a]==0)&&(pb6[a]==0)){
        nm[18]=a;
        uflg[a]++; rw3[a]=1; cl4[a]=1; pd2[a]=1; pb6[a]=1;

```

```

        stp47();
        rw3[a]=0; cl4[a]=0; pd2[a]=0; pb6[a]=0; uflg[a]--;}}
    }
}
// Set n32{5,4,7,1}
void stp47(){
    short a;
    for(a=6;a>=0;a--){
        if(uflg[a]<7){
            if((rw5[a]==0)&&(cl4[a]==0)&&(pd7[a]==0)&&(pb1[a]==0)){
                nm[32]=a;
                uflg[a]++; rw5[a]=1; cl4[a]=1; pd7[a]=1; pb1[a]=1;
                stp48();
                rw5[a]=0; cl4[a]=0; pd7[a]=0; pb1[a]=0; uflg[a]--;}}
        }
    }
}
// Set n24{4,3,7,6}
void stp48(){
    short a;
    for(a=0;a<7;a++){
        if(uflg[a]<7){
            if((rw4[a]==0)&&(cl3[a]==0)&&(pd7[a]==0)&&(pb6[a]==0)){
                nm[24]=a;
                uflg[a]++; rw4[a]=1; cl3[a]=1; pd7[a]=1; pb6[a]=1;
                stp49();
                rw4[a]=0; cl3[a]=0; pd7[a]=0; pb6[a]=0; uflg[a]--;}}
        }
    }
}
// Set n26{4,5,2,1}
void stp49(){
    short a;
    for(a=6;a>=0;a--){
        if(uflg[a]<7){
            if((rw4[a]==0)&&(cl5[a]==0)&&(pd2[a]==0)&&(pb1[a]==0)){
                nm[26]=a;
                uflg[a]++; rw4[a]=1; cl5[a]=1; pd2[a]=1; pb1[a]=1;
                lurecord();
                rw4[a]=0; cl5[a]=0; pd2[a]=0; pb1[a]=0; uflg[a]--;}}
        }
    }
}
//
/** Record the Answers *
void lurecord(){
    short n;
    tlu[cnt][0]=cnt+1;
    for(n=1;n<50;n++){tlu[cnt][n]=nm[n];}
    cnt++;
}
//
/** Classify and Print the Latin Units *
void prlunit(){
    short t,l,l7,m,n;
    for(t=0;t<lcnt;t=t+3360){
        printf("%8d/%8d/%8d/%8d/%8d/%8d/%8d/%8d/%8d/\n",
            t+1,t+2,t+3,t+4,t+5,t+6,t+7,t+8,t+9);
        for(l=0;l<7;l++){l7=l*7;
            for(m=t;m<(t+9);m++){
                printf(" ");
                for(n=1;n<8;n++){printf("%d",tlu[m][l7+n]);}
            }
            printf("\n");
        }
    }
}
printf(" [Counts of Latin Units = %d]\n",lcnt);
}

```

```

//
//Check How well they match
short chkmtc(short x, short y){
    short n,d1,d2;
    d1=0; for(n=1;n<50;n++){if(tlu[x][n]==tlu[y][n]){d1++;}}
    d2=0; for(n=1;n<50;n++){if(tlu[x][n]+tlu[x][n]==6){d2++;}}
    if(d2>d1){d1=d2;}
    return d1;
}
//
/* Combine, Compose and Print *
void cmbcmp(){
    short n,d,fc;
    for(u1=0;u1<(lcnt/2);u1++){cnt2=0;
        for(u2=0;u2<lcnt;u2++){
            if(chkmtc(u1,u2)==7){
                for(n=1;n<50;n++){uflg[n]=0;}
                for(n=1;n<50;n++){
                    d=tlu[u1][n]*7+tlu[u2][n]+1;
                    nm[n]=d; uflg[d]++;
                }
                fc=0;
                for(n=1;n<50;n++){if(uflg[n]==1){fc++;}else{break;}}
                if(fc==49){
                    if((nm[1]<nm[49])&&(nm[1]<nm[43])&&(nm[43]<nm[7])){prans();}
                }
            }
        }
    }
}
//
/* Print the Solution List *
void prans(){
    short n;
    cnt++; cnt2++;
    if(nm[1]==1){cnt1++;
        if(cnt2==1){
            cnt3[cnt3]=cnt1; cnt3[cnt3+1]=0;
            for(n=1;n<50;n++){dn[cnt3][n]=nm[n];}
            dn[cnt3][50]=u1+1; dn[cnt3][51]=u2+1;
            cnt3++;
            if(cnt1<60000){if(cnt3==2){pr2sol(); cnt3=0;}}
            else{if(cnt3==4){prsol(cnt3); cnt3=0;}}
        }
    }
}
//
/* Print 2 Pieces with Full Information *
void pr2sol(){
    short l,l7,m,n;
    short lu1,lu2;
    for(n=1;n<50;n++){tn[n]=dn[0][n];}; chksms(0);
    for(n=1;n<50;n++){tn[n]=dn[1][n];}; chksms(1);
    printf("%6d/H%6d/L%21d#IRIC\\P/PI%7d/H%6d/L%21d#IRIC\\P/PI\n",
        dn[0][50],dn[0][51],cnt3[0],dn[1][50],dn[1][51],cnt3[1]);
    for(l=0;l<7;l++){l7=l*7;
        for(m=0;m<2;m++){
            lu1=dn[m][50]-1; lu2=dn[m][51]-1;
            printf(" ");
            for(n=1;n<8;n++){printf("%d",tlu[lu1][l7+n]);}
            printf(" ");
            for(n=1;n<8;n++){printf("%d",tlu[lu2][l7+n]);}
            printf(" ");
            for(n=1;n<8;n++){printf("%3d",dn[m][l7+n]);}
            printf("%d\\%d\\%d\\%d\\",

```

```

        cs[m][l+1]-LSM,cs[m][l+8]-LSM,cs[m][l+15]-LSM,cs[m][l+22]-LSM);
    if(m<1){printf(" ");}
}
printf("\n");
}
}
//
/* Check Sums of Essential Lines *
void chksms(short x){
    cs[x][ 1]=tn[ 1]+tn[ 2]+tn[ 3]+tn[ 4]+tn[ 5]+tn[ 6]+tn[ 7];
    cs[x][ 2]=tn[ 8]+tn[ 9]+tn[10]+tn[11]+tn[12]+tn[13]+tn[14];
    cs[x][ 3]=tn[15]+tn[16]+tn[17]+tn[18]+tn[19]+tn[20]+tn[21];
    cs[x][ 4]=tn[22]+tn[23]+tn[24]+tn[25]+tn[26]+tn[27]+tn[28];
    cs[x][ 5]=tn[29]+tn[30]+tn[31]+tn[32]+tn[33]+tn[34]+tn[35];
    cs[x][ 6]=tn[36]+tn[37]+tn[38]+tn[39]+tn[40]+tn[41]+tn[42];
    cs[x][ 7]=tn[43]+tn[44]+tn[45]+tn[46]+tn[47]+tn[48]+tn[49];
    //
    cs[x][ 8]=tn[ 1]+tn[ 8]+tn[15]+tn[22]+tn[29]+tn[36]+tn[43];
    cs[x][ 9]=tn[ 2]+tn[ 9]+tn[16]+tn[23]+tn[30]+tn[37]+tn[44];
    cs[x][10]=tn[ 3]+tn[10]+tn[17]+tn[24]+tn[31]+tn[38]+tn[45];
    cs[x][11]=tn[ 4]+tn[11]+tn[18]+tn[25]+tn[32]+tn[39]+tn[46];
    cs[x][12]=tn[ 5]+tn[12]+tn[19]+tn[26]+tn[33]+tn[40]+tn[47];
    cs[x][13]=tn[ 6]+tn[13]+tn[20]+tn[27]+tn[34]+tn[41]+tn[48];
    cs[x][14]=tn[ 7]+tn[14]+tn[21]+tn[28]+tn[35]+tn[42]+tn[49];
    //
    cs[x][15]=tn[ 1]+tn[ 9]+tn[17]+tn[25]+tn[33]+tn[41]+tn[49];
    cs[x][16]=tn[ 2]+tn[10]+tn[18]+tn[26]+tn[34]+tn[42]+tn[43];
    cs[x][17]=tn[ 3]+tn[11]+tn[19]+tn[27]+tn[35]+tn[36]+tn[44];
    cs[x][18]=tn[ 4]+tn[12]+tn[20]+tn[28]+tn[29]+tn[37]+tn[45];
    cs[x][19]=tn[ 5]+tn[13]+tn[21]+tn[22]+tn[30]+tn[38]+tn[46];
    cs[x][20]=tn[ 6]+tn[14]+tn[15]+tn[23]+tn[31]+tn[39]+tn[47];
    cs[x][21]=tn[ 7]+tn[ 8]+tn[16]+tn[24]+tn[32]+tn[40]+tn[48];
    //
    cs[x][22]=tn[ 1]+tn[14]+tn[20]+tn[26]+tn[32]+tn[38]+tn[44];
    cs[x][23]=tn[ 2]+tn[ 8]+tn[21]+tn[27]+tn[33]+tn[39]+tn[45];
    cs[x][24]=tn[ 3]+tn[ 9]+tn[15]+tn[28]+tn[34]+tn[40]+tn[46];
    cs[x][25]=tn[ 4]+tn[10]+tn[16]+tn[22]+tn[35]+tn[41]+tn[47];
    cs[x][26]=tn[ 5]+tn[11]+tn[17]+tn[23]+tn[29]+tn[42]+tn[48];
    cs[x][27]=tn[ 6]+tn[12]+tn[18]+tn[24]+tn[30]+tn[36]+tn[49];
    cs[x][28]=tn[ 7]+tn[13]+tn[19]+tn[25]+tn[31]+tn[37]+tn[43];
}
//
/* Print X pieces of solutions *
void prsol(short x){
    short l,l7,m,n;
    for(m=0;m<x;m++){
        printf("%6d/%4d/%9ld#",dn[m][50],dn[m][51],cntr[m]);
        if(m+1<x){printf(" ");}
    }
    printf("\n");
    for(l=0;l<7;l++){l7=l*7;
        printf(" ");
        for(m=0;m<x;m++){
            for(n=1;n<8;n++){printf("%3d",dn[m][l7+n]);}
            if(m+1<x){printf(" ");}
        }
        printf("\n");
    }
}
//

```

## Execution Result of the Program #6:

\*\* Compose 'Complete Euler Squares' of Order 7 for \*\*  
 \*\* the Pan-diagonal MS77 by the New Euler's Method \*\*

[List of Latin Units]

1/	2/	3/	4/	5/	6/	7/	8/	9/
0654321	0531642	0362514	0246135	0653421	0541632	0462513	0236145	0654231
2106543	3164205	5140362	6135024	2106534	4163205	5130462	6145023	3106542
4321065	6420531	3625140	5024613	3421065	6320541	4625130	5023614	4231065
6543210	2053164	1403625	4613502	6534210	2054163	1304625	3614502	6542310
1065432	5316420	6251403	3502461	1065342	5416320	6251304	4502361	1065423
3210654	1642053	4036251	2461350	4210653	1632054	3046251	2361450	3210654
5432106	4205316	2514036	1350246	5342106	3205416	2513046	1450236	5423106
3361/	3362/	3363/	3364/	3365/	3366/	3367/	3368/	3369/
1630542	1564320	1352604	1245036	1640532	1563420	1452603	1235046	1620543
3054216	2015643	6041352	5036124	4053216	2015634	6031452	5046123	2054316
5421630	4320156	3526041	6124503	5321640	3420156	4526031	6123504	5431620
2163054	5643201	0413526	4503612	2164053	5634201	0314526	3504612	3162054
6305421	0156432	5260413	3612450	6405321	0156342	5260314	4612350	6205431
0542163	3201564	4135260	2450361	0532164	4201563	3145260	2350461	0543162
4216305	6432015	2604135	0361245	3216405	6342015	2603145	0461235	4316205
6721/	6722/	6723/	6724/	6725/	6726/	6727/	6728/	6729/
2630451	2465310	2341605	2154036	2650431	2541603	2463510	2134056	2610453
3045126	1024653	6052341	4036215	5043126	6032541	1024635	4056213	1045326
4512630	5310246	3416052	6215403	4312650	5416032	3510246	6213405	4532610
1263045	4653102	0523416	5403621	1265043	0325416	4635102	3405621	3261045
6304512	0246531	4160523	3621540	6504312	4160325	0246351	5621340	6104532
0451263	3102465	5234160	1540362	0431265	3254160	5102463	1340562	0453261
5126304	6531024	1605234	0362154	3126504	1603254	6351024	0562134	5326104
10081/	10082/	10083/	10084/	10085/	10086/	10087/	10088/	10089/
3640251	3421605	3265410	3152046	3650241	3521604	3264510	3142056	3610254
4025136	6053421	1032654	2046315	5024136	6043521	1032645	2056314	1025436
2513640	4216053	5410326	6315204	2413650	5216043	4510326	6314205	2543610
1364025	0534216	2654103	5204631	1365024	0435216	2645103	4205631	4361025
6402513	2160534	0326541	4631520	6502413	2160435	0326451	5631420	6102543
0251364	5342160	4103265	1520463	0241365	4352160	5103264	1420563	0254361
5136402	1605342	6541032	0463152	4136502	1604352	6451032	0563142	5436102
13441/	13442/	13443/	13444/	13445/	13446/	13447/	13448/	13449/
4630152	4312605	4251036	4165320	4650132	4512603	4231056	4163520	4620153
3015246	6054312	1036425	2041653	5013246	6034512	1056423	2041635	2015346
1524630	3126054	6425103	5320416	1324650	5126034	6423105	3520416	1534620
2463015	0543126	5103642	1653204	2465013	0345126	3105642	1635204	3462015
6301524	1260543	3642510	0416532	6501324	1260345	5642310	0416352	6201534
0152463	5431260	2510364	3204165	0132465	3451260	2310564	5204163	0153462
5246301	2605431	0364251	6532041	3246501	2603451	0564231	6352041	5346201
16801/	16802/	16803/	16804/	16805/	16806/	16807/	16808/	16809/
5631042	5302614	5240136	5064321	5641032	5402613	5230146	5063421	5621043
3104256	6145302	0136524	2150643	4103256	6135402	0146523	2150634	2104356
0425631	3026145	6524013	4321506	0325641	4026135	6523014	3421506	0435621
2563104	1453026	4013652	0643215	2564103	1354026	3014652	0634215	3562104
6310425	0261453	3652401	1506432	6410325	0261354	4652301	1506342	6210435
1042563	4530261	2401365	3215064	1032564	3540261	2301465	4215063	1043562
4256310	2614530	1365240	6432150	3256410	2613540	1465230	6342150	4356210

[Counts of Latin Units = 20160]

\*\* List of the 'Complete Euler Squares' of Order 7: Pan-Diagonal Type \*\*

[Solutions when n1=1: Used\_Units/H/L Sol\_Nmb# Check\_Sum-Errors||\|]

3/H	1/L	1# R C\P/P	4/H	1/L	2161# R C\P/P
0362514	0654321	1 28 48 19 39 10 30 0 0\0/0	0246135	0654321	1 21 34 47 11 24 37 0 0\0/0
5140362	2106543	38 9 29 7 27 47 18 0 0\0/0	6135024	2106543	45 9 22 42 6 19 32 0 0\0/0
3625140	4321065	26 46 17 37 8 35 6 0 0\0/0	5024613	4321065	40 4 17 30 43 14 27 0 0\0/0
1403625	6543210	14 34 5 25 45 16 36 0 0\0/0	4613502	6543210	35 48 12 25 38 2 15 0 0\0/0
6251403	1065432	44 15 42 13 33 4 24 0 0\0/0	3502461	1065432	23 36 7 20 33 46 10 0 0\0/0
4036251	3210654	32 3 23 43 21 41 12 0 0\0/0	2461350	3210654	18 31 44 8 28 41 5 0 0\0/0
2514036	5432106	20 40 11 31 2 22 49 0 0\0/0	1350246	5432106	13 26 39 3 16 29 49 0 0\0/0



7/H	1/L		4321# R C\P/P	8/H	1/L		6481# R C\P/P
0462513	0654321	1 35 48 19 39	10 23 0 0\0/0	0236145	0654321	1 21 27 47 11 31 37	0 0\0/0
5130462	2106543	38 9 22 7 34 47 18	0 0\0/0	6145023	2106543	45 9 29 42 6 19 25	0 0\0/0
4625130	4321065	33 46 17 37 8 28 6	0 0\0/0	5023614	4321065	40 4 17 23 43 14 34	0 0\0/0
1304625	6543210	14 27 5 32 45 16 36	0 0\0/0	3614502	6543210	28 48 12 32 38 2 15	0 0\0/0
6251304	1065432	44 15 42 13 26 4 31	0 0\0/0	4502361	1065432	30 36 7 20 26 46 10	0 0\0/0
3046251	3210654	25 3 30 43 21 41 12	0 0\0/0	2361450	3210654	18 24 44 8 35 41 5	0 0\0/0
2513046	5432106	20 40 11 24 2 29 49	0 0\0/0	1450236	5432106	13 33 39 3 16 22 49	0 0\0/0
11/H	1/L		8641# R C\P/P	12/H	1/L		10801# R C\P/P
0346125	0654321	1 28 34 47 11 17 37	0 0\0/0	0263514	0654321	1 21 48 26 39 10 30	0 0\0/0
6125034	2106543	45 9 15 42 6 26 32	0 0\0/0	5140263	2106543	38 9 29 7 20 47 25	0 0\0/0
5034612	4321065	40 4 24 30 43 14 20	0 0\0/0	2635140	4321065	19 46 24 37 8 35 6	0 0\0/0
4612503	6543210	35 48 12 18 38 2 22	0 0\0/0	1402635	6543210	14 34 5 18 45 23 36	0 0\0/0
2503461	1065432	16 36 7 27 33 46 10	0 0\0/0	6351402	1065432	44 22 42 13 33 4 17	0 0\0/0
3461250	3210654	25 31 44 8 21 41 5	0 0\0/0	4026351	3210654	32 3 16 43 28 41 12	0 0\0/0
1250346	5432106	13 19 39 3 23 29 49	0 0\0/0	3514026	5432106	27 40 11 31 2 15 49	0 0\0/0
14/H	1/L		12961# R C\P/P	16/H	1/L		15121# R C\P/P
0541623	0654321	1 42 34 12 46 17 23	0 0\0/0	0326145	0654321	1 28 20 47 11 31 37	0 0\0/0
4162305	2106543	31 9 43 21 27 5 39	0 0\0/0	6145032	2106543	45 9 29 42 6 26 18	0 0\0/0
6230541	4321065	47 18 24 2 36 35 13	0 0\0/0	5032614	4321065	40 4 24 16 43 14 34	0 0\0/0
3054162	6543210	28 6 40 32 10 44 15	0 0\0/0	2614503	6543210	21 48 12 32 38 2 22	0 0\0/0
5416230	1065432	37 29 14 48 19 25 3	0 0\0/0	4503261	1065432	30 36 7 27 19 46 10	0 0\0/0
1623054	3210654	11 45 16 22 7 41 33	0 0\0/0	3261450	3210654	25 17 44 8 35 41 5	0 0\0/0
2305416	5432106	20 26 4 38 30 8 49	0 0\0/0	1450326	5432106	13 33 39 3 23 15 49	0 0\0/0
18/H	1/L		17281# R C\P/P	19/H	1/L		19441# R C\P/P
0521634	0654321	1 42 20 12 46 24 30	0 0\0/0	0436125	0654321	1 35 27 47 11 17 37	0 0\0/0
2163405	2106543	17 9 43 28 34 5 39	0 0\0/0	6125043	2106543	45 9 15 42 6 33 25	0 0\0/0
6340521	4321065	47 25 31 2 36 21 13	0 0\0/0	5043612	4321065	40 4 31 23 43 14 20	0 0\0/0
4052163	6543210	35 6 40 18 10 44 22	0 0\0/0	3612504	6543210	28 48 12 18 38 2 29	0 0\0/0
5216340	1065432	37 15 14 48 26 32 3	0 0\0/0	2504361	1065432	16 36 7 34 26 46 10	0 0\0/0
1634052	3210654	11 45 23 29 7 41 19	0 0\0/0	4361250	3210654	32 24 44 8 21 41 5	0 0\0/0
3405216	5432106	27 33 4 38 16 8 49	0 0\0/0	1250436	5432106	13 19 39 3 30 22 49	0 0\0/0
22/H	1/L		21601# R C\P/P	23/H	1/L		23761# R C\P/P
0531624	0654321	1 42 27 12 46 17 30	0 0\0/0	0426135	0654321	1 35 20 47 11 24 37	0 0\0/0
3162405	2106543	24 9 43 21 34 5 39	0 0\0/0	6135042	2106543	45 9 22 42 6 33 18	0 0\0/0
6240531	4321065	47 18 31 2 36 28 13	0 0\0/0	5042613	4321065	40 4 31 16 43 14 27	0 0\0/0
4053162	6543210	35 6 40 25 10 44 15	0 0\0/0	2613504	6543210	21 48 12 25 38 2 29	0 0\0/0
5316240	1065432	37 22 14 48 19 32 3	0 0\0/0	3504261	1065432	23 36 7 34 19 46 10	0 0\0/0
1624053	3210654	11 45 16 29 7 41 26	0 0\0/0	4261350	3210654	32 17 44 8 28 41 5	0 0\0/0
2405316	5432106	20 33 4 38 23 8 49	0 0\0/0	1350426	5432106	13 26 39 3 30 15 49	0 0\0/0
27/H	1/L		25921# R C\P/P	28/H	1/L		28081# R C\P/P
0362415	0654321	1 28 48 19 32 10 37	0 0\0/0	0256134	0654321	1 21 41 47 11 24 30	0 0\0/0
4150362	2106543	31 9 36 7 27 47 18	0 0\0/0	6134025	2106543	45 9 22 35 6 19 39	0 0\0/0
3624150	4321065	26 46 17 30 8 42 6	0 0\0/0	4025613	4321065	33 4 17 37 43 14 27	0 0\0/0
1503624	6543210	14 41 5 25 45 16 29	0 0\0/0	5613402	6543210	42 48 12 25 31 2 15	0 0\0/0
6241503	1065432	44 15 35 13 40 4 24	0 0\0/0	3402561	1065432	23 29 7 20 40 46 10	0 0\0/0
5036241	3210654	39 3 23 43 21 34 12	0 0\0/0	2561340	3210654	18 38 44 8 28 34 5	0 0\0/0
2415036	5432106	20 33 11 38 2 22 49	0 0\0/0	1340256	5432106	13 26 32 3 16 36 49	0 0\0/0
30/H	1/L		30241# R C\P/P	32/H	1/L		32401# R C\P/P
0562413	0654321	1 42 48 19 32 10 23	0 0\0/0	0236154	0654321	1 21 27 47 11 38 30	0 0\0/0
4130562	2106543	31 9 22 7 41 47 18	0 0\0/0	6154023	2106543	45 9 36 35 6 19 25	0 0\0/0
5624130	4321065	40 46 17 30 8 28 6	0 0\0/0	4023615	4321065	33 4 17 23 43 14 41	0 0\0/0
1305624	6543210	14 27 5 39 45 16 29	0 0\0/0	3615402	6543210	28 48 12 39 31 2 15	0 0\0/0
6241305	1065432	44 15 35 13 26 4 38	0 0\0/0	5402361	1065432	37 29 7 20 26 46 10	0 0\0/0
3056241	3210654	25 3 37 43 21 34 12	0 0\0/0	2361540	3210654	18 24 44 8 42 34 5	0 0\0/0
2413056	5432106	20 33 11 24 2 36 49	0 0\0/0	1540236	5432106	13 40 32 3 16 22 49	0 0\0/0
35/H	1/L		34561# R C\P/P	36/H	1/L		36721# R C\P/P
0356124	0654321	1 28 41 47 11 17 30	0 0\0/0	0263415	0654321	1 21 48 26 32 10 37	0 0\0/0
6124035	2106543	45 9 15 35 6 26 39	0 0\0/0	4150263	2106543	31 9 36 7 20 47 25	0 0\0/0
4035612	4321065	33 4 24 37 43 14 20	0 0\0/0	2634150	4321065	19 46 24 30 8 42 6	0 0\0/0
5612403	6543210	42 48 12 18 31 2 22	0 0\0/0	1502634	6543210	14 41 5 18 45 23 29	0 0\0/0
2403561	1065432	16 29 7 27 40 46 10	0 0\0/0	6341502	1065432	44 22 35 13 40 4 17	0 0\0/0
3561240	3210654	25 38 44 8 21 34 5	0 0\0/0	5026341	3210654	39 3 16 43 28 34 12	0 0\0/0
1240356	5432106	13 19 32 3 23 36 49	0 0\0/0	3415026	5432106	27 33 11 38 2 15 49	0 0\0/0

39/H	1/L	38881# R C\P/P	40/H	1/L	41041# R C\P/P						
0451623	0654321	1 35 41 12 46 17 23 0 0\0/0	0326154	0654321	1 28 20 47 11 38 30 0 0\0/0						
5162304	2106543	38 9 43 21 27 5 32 0 0\0/0	6154032	2106543	45 9 36 35 6 26 18 0 0\0/0						
6230451	4321065	47 18 24 2 29 42 13 0 0\0/0	4032615	4321065	33 4 24 16 43 14 41 0 0\0/0						
3045162	6543210	28 6 33 39 10 44 15 0 0\0/0	2615403	6543210	21 48 12 39 31 2 22 0 0\0/0						
4516230	1065432	30 36 14 48 19 25 3 0 0\0/0	5403261	1065432	37 29 7 27 19 46 10 0 0\0/0						
1623045	3210654	11 45 16 22 7 34 40 0 0\0/0	3261540	3210654	25 17 44 8 42 34 5 0 0\0/0						
2304516	5432106	20 26 4 31 37 8 49 0 0\0/0	1540326	5432106	13 40 32 3 23 15 49 0 0\0/0						
42/H	1/L	43201# R C\P/P	43/H	1/L	45361# R C\P/P						
0536124	0654321	1 42 27 47 11 17 30 0 0\0/0	0421635	0654321	1 35 20 12 46 24 37 0 0\0/0						
6124053	2106543	45 9 15 35 6 40 25 0 0\0/0	2163504	2106543	17 9 43 28 41 5 32 0 0\0/0						
4053612	4321065	33 4 38 23 43 14 20 0 0\0/0	6350421	4321065	47 25 38 2 29 21 13 0 0\0/0						
3612405	6543210	28 48 12 18 31 2 36 0 0\0/0	5042163	6543210	42 6 33 18 10 44 22 0 0\0/0						
2405361	1065432	16 29 7 41 26 46 10 0 0\0/0	4216350	1065432	30 15 14 48 26 39 3 0 0\0/0						
5361240	3210654	39 24 44 8 21 34 5 0 0\0/0	1635042	3210654	11 45 23 36 7 34 19 0 0\0/0						
1240536	5432106	13 19 32 3 37 22 49 0 0\0/0	3504216	5432106	27 40 4 31 16 8 49 0 0\0/0						
46/H	1/L	47521# R C\P/P	47/H	1/L	49681# R C\P/P						
0526134	0654321	1 42 20 47 11 24 30 0 0\0/0	0431625	0654321	1 35 27 12 46 17 37 0 0\0/0						
6134052	2106543	45 9 22 35 6 40 18 0 0\0/0	3162504	2106543	24 9 43 21 41 5 32 0 0\0/0						
4052613	4321065	33 4 38 16 43 14 27 0 0\0/0	6250431	4321065	47 18 38 2 29 28 13 0 0\0/0						
2613405	6543210	21 48 12 25 31 2 36 0 0\0/0	5043162	6543210	42 6 33 25 10 44 15 0 0\0/0						
3405261	1065432	23 29 7 41 19 46 10 0 0\0/0	4316250	1065432	30 22 14 48 19 39 3 0 0\0/0						
5261340	3210654	39 17 44 8 28 34 5 0 0\0/0	1625043	3210654	11 45 16 36 7 34 26 0 0\0/0						
1340526	5432106	13 26 32 3 37 15 49 0 0\0/0	2504316	5432106	20 40 4 31 23 8 49 0 0\0/0						
50/H	1/L	51841# R C\P/P	52/H	1/L	54001# R C\P/P						
0462315	0654321	1 35 48 19 25 10 37 0 0\0/0	0256143	0654321	1 21 41 47 11 31 23 0 0\0/0						
3150462	2106543	24 9 36 7 34 47 18 0 0\0/0	6143025	2106543	45 9 29 28 6 19 39 0 0\0/0						
4623150	4321065	33 46 17 23 8 42 6 0 0\0/0	3025614	4321065	26 4 17 37 43 14 34 0 0\0/0						
1504623	6543210	14 41 5 32 45 16 22 0 0\0/0	5614302	6543210	42 48 12 32 24 2 15 0 0\0/0						
6231504	1065432	44 15 28 13 40 4 31 0 0\0/0	4302561	1065432	30 22 7 20 40 46 10 0 0\0/0						
5046231	3210654	39 3 30 43 21 27 12 0 0\0/0	2561430	3210654	18 38 44 8 35 27 5 0 0\0/0						
2315046	5432106	20 26 11 38 2 29 49 0 0\0/0	1430256	5432106	13 33 25 3 16 36 49 0 0\0/0						
54/H	1/L	56161# R C\P/P	56/H	1/L	58321# R C\P/P						
0562314	0654321	1 42 48 19 25 10 30 0 0\0/0	0246153	0654321	1 21 34 47 11 38 23 0 0\0/0						
3140562	2106543	24 9 29 7 41 47 18 0 0\0/0	6153024	2106543	45 9 36 28 6 19 32 0 0\0/0						
5623140	4321065	40 46 17 23 8 35 6 0 0\0/0	3024615	4321065	26 4 17 30 43 14 41 0 0\0/0						
1405623	6543210	14 34 5 39 45 16 22 0 0\0/0	4615302	6543210	35 48 12 39 24 2 15 0 0\0/0						
6231405	1065432	44 15 28 13 33 4 38 0 0\0/0	5302461	1065432	37 22 7 20 33 46 10 0 0\0/0						
4056231	3210654	32 3 37 43 21 27 12 0 0\0/0	2461530	3210654	18 31 44 8 42 27 5 0 0\0/0						
2314056	5432106	20 26 11 31 2 36 49 0 0\0/0	1530246	5432106	13 40 25 3 16 29 49 0 0\0/0						
58/	1/	60481#	60/	1/	62641#	63/	1/	64801#	64/	1/	66961#
1 35 41 47 11 17 23	1 21 48 33 25 10 37	1 35 20 47 11 38 23	1 28 41 12 46 17 30								
45 9 15 28 6 33 39	24 9 36 7 20 47 32	45 9 36 28 6 33 18	38 9 43 21 34 5 25								
26 4 31 37 43 14 20	19 46 31 23 8 42 6	26 4 31 16 43 14 41	47 18 31 2 22 42 13								
42 48 12 18 24 2 29	14 41 5 18 45 30 22	21 48 12 39 24 2 29	35 6 26 39 10 44 15								
16 22 7 34 40 46 10	44 29 28 13 40 4 17	37 22 7 34 19 46 10	23 36 14 48 19 32 3								
32 38 44 8 21 27 5	39 3 16 43 35 27 12	32 17 44 8 42 27 5	11 45 16 29 7 27 40								
13 19 25 3 30 36 49	34 26 11 38 2 15 49	13 40 25 3 30 15 49	20 33 4 24 37 8 49								
66/	1/	69121#	67/	1/	71281#	70/	1/	73441#	72/	1/	75601#
1 42 34 47 11 17 23	1 28 20 12 46 31 37	1 42 20 47 11 31 23	1 28 34 12 46 17 37								
45 9 15 28 6 40 32	17 9 43 35 41 5 25	45 9 29 28 6 40 18	31 9 43 21 41 5 25								
26 4 38 30 43 14 20	47 32 38 2 22 21 13	26 4 38 16 43 14 34	47 18 38 2 22 35 13								
35 48 12 18 24 2 36	42 6 26 18 10 44 29	21 48 12 32 24 2 36	42 6 26 32 10 44 15								
16 22 7 41 33 46 10	23 15 14 48 33 39 3	30 22 7 41 19 46 10	23 29 14 48 19 39 3								
39 31 44 8 21 27 5	11 45 30 36 7 27 19	39 17 44 8 35 27 5	11 45 16 36 7 27 33								
13 19 25 3 37 29 49	34 40 4 24 16 8 49	13 33 25 3 37 15 49	20 40 4 24 30 8 49								
74/	1/	77761#	75/	1/	79921#	78/	1/	82081#	79/	1/	84241#
1 35 48 26 18 10 37	1 28 41 47 11 31 16	1 42 48 26 18 10 30	1 28 34 47 11 38 16								
17 9 36 7 34 47 25	45 9 29 21 6 26 39	17 9 29 7 41 47 25	45 9 36 21 6 26 32								
33 46 24 16 8 42 6	19 4 24 37 43 14 34	40 46 24 16 8 35 6	19 4 24 30 43 14 41								
14 41 5 32 45 23 15	42 48 12 32 17 2 22	14 34 5 39 45 23 15	35 48 12 39 17 2 22								
44 22 21 13 40 4 31	30 15 7 27 40 46 10	44 22 21 13 33 4 38	37 15 7 27 33 46 10								
39 3 30 43 28 20 12	25 38 44 8 35 20 5	32 3 37 43 28 20 12	25 31 44 8 42 20 5								
27 19 11 38 2 29 49	13 33 18 3 23 36 49	27 19 11 31 2 36 49	13 40 18 3 23 29 49								

82/ 1/ 86401#	83/ 1/ 88561#	87/ 1/ 90721#	88/ 1/ 92881#
1 35 41 47 11 24 16	1 28 48 33 18 10 37	1 35 27 47 11 38 16	1 21 41 12 46 24 30
45 9 22 21 6 33 39	17 9 36 7 27 47 32	45 9 36 21 6 33 25	38 9 43 28 34 5 18
19 4 31 37 43 14 27	26 46 31 16 8 42 6	19 4 31 23 43 14 41	47 25 31 2 15 42 13
42 48 12 25 17 2 29	14 41 5 25 45 30 15	28 48 12 39 17 2 29	35 6 19 39 10 44 22
23 15 7 34 40 46 10	44 29 21 13 40 4 24	37 15 7 34 26 46 10	16 36 14 48 26 32 3
32 38 44 8 28 20 5	39 3 23 43 35 20 12	32 24 44 8 42 20 5	11 45 23 29 7 20 40
13 26 18 3 30 36 49	34 19 11 38 2 22 49	13 40 18 3 30 22 49	27 33 4 17 37 8 49
90/ 1/ 95041#	92/ 1/ 97201#	94/ 1/ 99361#	96/ 1/ 101521#
1 42 34 47 11 24 16	1 21 27 12 46 31 37	1 42 27 47 11 31 16	1 21 34 12 46 24 37
45 9 22 21 6 40 32	24 9 43 35 41 5 18	45 9 29 21 6 40 25	31 9 43 28 41 5 18
19 4 38 30 43 14 27	47 32 38 2 15 28 13	19 4 38 23 43 14 34	47 25 38 2 15 35 13
35 48 12 25 17 2 36	42 6 19 25 10 44 29	28 48 12 32 17 2 36	42 6 19 32 10 44 22
23 15 7 41 33 46 10	16 22 14 48 33 39 3	30 15 7 41 26 46 10	16 29 14 48 26 39 3
39 31 44 8 28 20 5	11 45 30 36 7 20 26	39 24 44 8 35 20 5	11 45 23 36 7 20 33
13 26 18 3 37 29 49	34 40 4 17 23 8 49	13 33 18 3 37 22 49	27 40 4 17 30 8 49
99/ 1/ 103681#	100/ 1/ 105841#	103/ 1/ 108001#	104/ 1/ 110161#
1 28 48 12 39 17 30	1 14 34 47 18 24 37	1 35 48 12 39 17 23	1 14 27 47 18 31 37
38 16 29 7 27 47 11	45 16 22 42 6 12 32	38 16 22 7 34 47 11	45 16 29 42 6 12 25
26 46 10 37 15 35 6	40 4 10 30 43 21 27	33 46 10 37 15 28 6	40 4 10 23 43 21 34
21 34 5 25 45 9 36	35 48 19 25 38 2 8	21 27 5 32 45 9 36	28 48 19 32 38 2 8
44 8 42 20 33 4 24	23 36 7 13 33 46 17	44 8 42 20 26 4 31	30 36 7 13 26 46 17
32 3 23 43 14 41 19	11 31 44 15 28 41 5	25 3 30 43 14 41 19	11 24 44 15 35 41 5
13 40 18 31 2 22 49	20 26 39 3 9 29 49	13 40 18 24 2 29 49	20 33 39 3 9 22 49
. . . . .			
497/ 1/ 535681#	499/ 1/ 537841#	501/ 1/ 540001#	503/ 1/ 542161#
1 49 20 12 39 24 30	1 35 27 40 11 17 44	1 49 27 12 39 17 30	1 35 20 40 11 24 44
17 9 36 28 34 5 46	38 9 15 49 6 33 25	24 9 36 21 34 5 46	38 9 22 49 6 33 18
40 25 31 2 43 21 13	47 4 31 23 36 14 20	40 18 31 2 43 28 13	47 4 31 16 36 14 27
35 6 47 18 10 37 22	28 41 12 18 45 2 29	35 6 47 25 10 37 15	21 41 12 25 45 2 29
44 15 14 41 26 32 3	16 43 7 34 26 39 10	44 22 14 41 19 32 3	23 43 7 34 19 39 10
11 38 23 29 7 48 19	32 24 37 8 21 48 5	11 38 16 29 7 48 26	32 17 37 8 28 48 5
27 33 4 45 16 8 42	13 19 46 3 30 22 42	20 33 4 45 23 8 42	13 26 46 3 30 15 42
507/ 1/ 544321#	508/ 1/ 546481#	509/ 1/ 548641#	512/ 1/ 550801#
1 28 41 19 32 10 44	1 21 48 40 11 24 30	1 49 41 19 32 10 23	1 21 27 40 11 45 30
31 9 43 7 27 40 18	38 9 22 35 6 19 46	31 9 22 7 48 40 18	38 9 43 35 6 19 25
26 39 17 30 8 49 6	33 4 17 44 36 14 27	47 39 17 30 8 28 6	33 4 17 23 36 14 48
14 48 5 25 38 16 29	49 41 12 25 31 2 15	14 27 5 46 38 16 29	28 41 12 46 31 2 15
37 15 35 13 47 4 24	23 29 7 20 47 39 10	37 15 35 13 26 4 45	44 29 7 20 26 39 10
46 3 23 36 21 34 12	18 45 37 8 28 34 5	25 3 44 36 21 34 12	18 24 37 8 49 34 5
20 33 11 45 2 22 42	13 26 32 3 16 43 42	20 33 11 24 2 43 42	13 47 32 3 16 22 42
515/ 1/ 552961#	516/ 1/ 555121#	519/ 1/ 557281#	520/ 1/ 559441#
1 28 48 40 11 17 30	1 21 41 26 32 10 44	1 35 48 12 39 17 23	1 28 20 40 11 45 30
38 9 15 35 6 26 46	31 9 43 7 20 40 25	45 9 36 21 27 5 32	38 9 43 35 6 26 18
33 4 24 44 36 14 20	19 39 24 30 8 49 6	40 18 24 2 29 49 13	33 4 24 16 36 14 48
49 41 12 18 31 2 22	14 48 5 18 38 23 29	28 6 33 46 10 37 15	21 41 12 46 31 2 22
16 29 7 27 47 39 10	37 22 35 13 47 4 17	30 43 14 41 19 25 3	44 29 7 27 19 39 10
25 45 37 8 21 34 5	46 3 16 36 28 34 12	11 38 16 22 7 34 47	25 17 37 8 49 34 5
13 19 32 3 23 43 42	27 33 11 45 2 15 42	20 26 4 31 44 8 42	13 47 32 3 23 15 42
521/ 1/ 561601#	523/ 1/ 563761#	525/ 1/ 565921#	527/ 1/ 568081#
1 49 27 40 11 17 30	1 35 20 12 39 24 44	1 49 20 40 11 24 30	1 35 27 12 39 17 44
38 9 15 35 6 47 25	17 9 36 28 48 5 32	38 9 22 35 6 47 18	24 9 36 21 48 5 32
33 4 45 23 36 14 20	40 25 45 2 29 21 13	33 4 45 16 36 14 27	40 18 45 2 29 28 13
28 41 12 18 31 2 43	49 6 33 18 10 37 22	21 41 12 25 31 2 43	49 6 33 25 10 37 15
16 29 7 48 26 39 10	30 15 14 41 26 46 3	23 29 7 48 19 39 10	30 22 14 41 19 46 3
46 24 37 8 21 34 5	11 38 23 43 7 34 19	46 17 37 8 28 34 5	11 38 16 43 7 34 26
13 19 32 3 44 22 42	27 47 4 31 16 8 42	13 26 32 3 44 15 42	20 47 4 31 23 8 42
530/ 1/ 570241#	532/ 1/ 572401#	533/ 1/ 574561#	536/ 1/ 576721#
1 35 41 19 25 10 44	1 21 48 40 11 31 23	1 49 41 19 25 10 30	1 21 34 40 11 45 23
24 9 43 7 34 40 18	38 9 29 28 6 19 46	24 9 29 7 48 40 18	38 9 43 28 6 19 32
33 39 17 23 8 49 6	26 4 17 44 36 14 34	47 39 17 23 8 35 6	26 4 17 30 36 14 48
14 48 5 32 38 16 22	49 41 12 32 24 2 15	14 34 5 46 38 16 22	35 41 12 46 24 2 15
37 15 28 13 47 4 31	30 22 7 20 47 39 10	37 15 28 13 33 4 45	44 22 7 20 33 39 10
46 3 30 36 21 27 12	18 45 37 8 35 27 5	32 3 44 36 21 27 12	18 31 37 8 49 27 5

20 26 11 45 2 29 42	13 33 25 3 16 43 42	20 26 11 31 2 43 42	13 47 25 3 16 29 42
. . . . .			
995/ 1/ 1071361#	996/ 1/ 1073521#	998/ 1/ 1075681#	1000/ 1/ 1077841#
1 28 48 33 11 17 37	1 21 34 26 39 10 44	1 42 48 12 32 17 23	1 28 20 33 11 45 37
31 9 15 42 6 26 46	38 9 43 7 20 33 25	45 9 29 21 27 5 39	31 9 43 42 6 26 18
40 4 24 44 29 14 20	19 32 24 37 8 49 6	33 18 24 2 36 49 13	40 4 24 16 29 14 48
49 34 12 18 38 2 22	14 48 5 18 31 23 36	28 6 40 46 10 30 15	21 34 12 46 38 2 22
16 36 7 27 47 32 10	30 22 42 13 47 4 17	37 43 14 34 19 25 3	44 36 7 27 19 32 10
25 45 30 8 21 41 5	46 3 16 29 28 41 12	11 31 16 22 7 41 47	25 17 30 8 49 41 5
13 19 39 3 23 43 35	27 40 11 45 2 15 35	20 26 4 38 44 8 35	13 47 39 3 23 15 35
1001/ 1/ 1080001#	1002/ 1/ 1082161#	1005/ 1/ 1084321#	1006/ 1/ 1086481#
1 49 27 33 11 17 37	1 42 20 12 32 24 44	1 49 20 33 11 24 37	1 42 27 12 32 17 44
31 9 15 42 6 47 25	17 9 29 28 48 5 39	31 9 22 42 6 47 18	24 9 29 21 48 5 39
40 4 45 23 29 14 20	33 25 45 2 36 21 13	40 4 45 16 29 14 27	33 18 45 2 36 28 13
28 34 12 18 38 2 43	49 6 40 18 10 30 22	21 34 12 25 38 2 43	49 6 40 25 10 30 15
16 36 7 48 26 32 10	37 15 14 34 26 46 3	23 36 7 48 19 32 10	37 22 14 34 19 46 3
46 24 30 8 21 41 5	11 31 23 43 7 41 19	46 17 30 8 28 41 5	11 31 16 43 7 41 26
13 19 39 3 44 22 35	27 47 4 38 16 8 35	13 26 39 3 44 15 35	20 47 4 38 23 8 35
1009/ 1/ 1088641#	1012/ 1/ 1090801#	1013/ 1/ 1092961#	1016/ 1/ 1095121#
1 42 34 19 25 10 44	1 21 48 33 11 38 23	1 49 34 19 25 10 37	1 21 41 33 11 45 23
24 9 43 7 41 33 18	31 9 36 28 6 19 46	24 9 36 7 48 33 18	31 9 43 28 6 19 39
40 32 17 23 8 49 6	26 4 17 44 29 14 41	47 32 17 23 8 42 6	26 4 17 37 29 14 48
14 48 5 39 31 16 22	49 34 12 39 24 2 15	14 41 5 46 31 16 22	42 34 12 46 24 2 15
30 15 28 13 47 4 38	37 22 7 20 47 32 10	30 15 28 13 40 4 45	44 22 7 20 40 32 10
46 3 37 29 21 27 12	18 45 30 8 42 27 5	39 3 44 29 21 27 12	18 38 30 8 49 27 5
20 26 11 45 2 36 35	13 40 25 3 16 43 35	20 26 11 38 2 43 35	13 47 25 3 16 36 35
1017/ 1/ 1097281#	1020/ 1/ 1099441#	1022/ 1/ 1101601#	1024/ 1/ 1103761#
1 42 48 33 11 17 23	1 21 34 40 25 10 44	1 42 20 33 11 45 23	1 28 48 12 32 17 37
31 9 15 28 6 40 46	24 9 43 7 20 33 39	31 9 43 28 6 40 18	45 9 29 21 41 5 25
26 4 38 44 29 14 20	19 32 38 23 8 49 6	26 4 38 16 29 14 48	33 18 38 2 22 49 13
49 34 12 18 24 2 36	14 48 5 18 31 37 22	21 34 12 46 24 2 36	42 6 26 46 10 30 15
16 22 7 41 47 32 10	30 36 28 13 47 4 17	44 22 7 41 19 32 10	23 43 14 34 19 39 3
39 45 30 8 21 27 5	46 3 16 29 42 27 12	39 17 30 8 49 27 5	11 31 16 36 7 27 47
13 19 25 3 37 43 35	41 26 11 45 2 15 35	13 47 25 3 37 15 35	20 40 4 24 44 8 35
1025/ 1/ 1105921#	1027/ 1/ 1108081#	1029/ 1/ 1110241#	1032/ 1/ 1112401#
1 49 41 33 11 17 23	1 28 20 12 32 38 44	1 49 20 33 11 38 23	1 28 41 12 32 17 44
31 9 15 28 6 47 39	17 9 29 42 48 5 25	31 9 36 28 6 47 18	38 9 29 21 48 5 25
26 4 45 37 29 14 20	33 39 45 2 22 21 13	26 4 45 16 29 14 41	33 18 45 2 22 42 13
42 34 12 18 24 2 43	49 6 26 18 10 30 36	21 34 12 39 24 2 43	49 6 26 39 10 30 15
16 22 7 48 40 32 10	23 15 14 34 40 46 3	37 22 7 48 19 32 10	23 36 14 34 19 46 3
46 38 30 8 21 27 5	11 31 37 43 7 27 19	46 17 30 8 42 27 5	11 31 16 43 7 27 40
13 19 25 3 44 36 35	41 47 4 24 16 8 35	13 40 25 3 44 15 35	20 47 4 24 37 8 35
. . . . .			
1497/ 1/ 1615681#	1500/ 1/ 1617841#	1502/ 1/ 1620001#	1503/ 1/ 1622161#
1 42 48 26 11 17 30	1 21 27 40 32 10 44	1 42 20 26 11 45 30	1 35 48 12 25 17 37
24 9 15 35 6 40 46	31 9 43 7 20 26 39	24 9 43 35 6 40 18	45 9 22 21 41 5 32
33 4 38 44 22 14 20	19 25 38 30 8 49 6	33 4 38 16 22 14 48	26 18 38 2 29 49 13
49 27 12 18 31 2 36	14 48 5 18 24 37 29	21 27 12 46 31 2 36	42 6 33 46 10 23 15
16 29 7 41 47 25 10	23 36 35 13 47 4 17	44 29 7 41 19 25 10	30 43 14 27 19 39 3
39 45 23 8 21 34 5	46 3 16 22 42 34 12	39 17 23 8 49 34 5	11 24 16 36 7 34 47
13 19 32 3 37 43 28	41 33 11 45 2 15 28	13 47 32 3 37 15 28	20 40 4 31 44 8 28
1505/ 1/ 1624321#	1506/ 1/ 1626481#	1509/ 1/ 1628641#	1511/ 1/ 1630801#
1 49 41 26 11 17 30	1 35 20 12 25 38 44	1 49 20 26 11 38 30	1 35 41 12 25 17 44
24 9 15 35 6 47 39	17 9 22 42 48 5 32	24 9 36 35 6 47 18	38 9 22 21 48 5 32
33 4 45 37 22 14 20	26 39 45 2 29 21 13	33 4 45 16 22 14 41	26 18 45 2 29 42 13
42 27 12 18 31 2 43	49 6 33 18 10 23 36	21 27 12 39 31 2 43	49 6 33 39 10 23 15
16 29 7 48 40 25 10	30 15 14 27 40 46 3	37 29 7 48 19 25 10	30 36 14 27 19 46 3
46 38 23 8 21 34 5	11 24 37 43 7 34 19	46 17 23 8 42 34 5	11 24 16 43 7 34 40
13 19 32 3 44 36 28	41 47 4 31 16 8 28	13 40 32 3 44 15 28	20 47 4 31 37 8 28
1513/ 1/ 1632961#	1514/ 1/ 1635121#	1517/ 1/ 1637281#	1518/ 1/ 1639441#
1 42 27 33 18 10 44	1 35 48 26 11 38 16	1 49 27 33 18 10 37	1 35 41 26 11 45 16
17 9 43 7 41 26 32	24 9 36 21 6 33 46	17 9 36 7 48 26 32	24 9 43 21 6 33 39
40 25 31 16 8 49 6	19 4 31 44 22 14 41	47 25 31 16 8 42 6	19 4 31 37 22 14 48
14 48 5 39 24 30 15	49 27 12 39 17 2 29	14 41 5 46 24 30 15	42 27 12 46 17 2 29

23 29 21 13 47 4 38	37 15 7 34 47 25 10	23 29 21 13 40 4 45	44 15 7 34 40 25 10
46 3 37 22 35 20 12	32 45 23 8 42 20 5	39 3 44 22 35 20 12	32 38 23 8 49 20 5
34 19 11 45 2 36 28	13 40 18 3 30 43 28	34 19 11 38 2 43 28	13 47 18 3 30 36 28
1521/ 1/ 1641601#	1522/ 1/ 1643761#	1526/ 1/ 1645921#	1528/ 1/ 1648081#
1 42 48 26 11 31 16	1 35 27 40 18 10 44	1 42 34 26 11 45 16	1 21 48 12 25 31 37
24 9 29 21 6 40 46	17 9 43 7 34 26 39	24 9 43 21 6 40 32	45 9 22 35 41 5 18
19 4 38 44 22 14 34	33 25 38 16 8 49 6	19 4 38 30 22 14 48	26 32 38 2 15 49 13
49 27 12 32 17 2 36	14 48 5 32 24 37 15	35 27 12 46 17 2 36	42 6 19 46 10 23 29
30 15 7 41 47 25 10	23 36 21 13 47 4 31	44 15 7 41 33 25 10	16 43 14 27 33 39 3
39 45 23 8 35 20 5	46 3 30 22 42 20 12	39 31 23 8 49 20 5	11 24 30 36 7 20 47
13 33 18 3 37 43 28	41 19 11 45 2 29 28	13 47 18 3 37 29 28	34 40 4 17 44 8 28
. . . . .			
1993/ 1/ 2151361#	1994/ 1/ 2153521#	1997/ 1/ 2155681#	1998/ 1/ 2157841#
1 42 20 33 25 10 44	1 35 48 19 11 38 23	1 49 20 33 25 10 37	1 35 41 19 11 45 23
24 9 43 7 41 19 32	17 9 36 28 6 33 46	24 9 36 7 48 19 32	17 9 43 28 6 33 39
40 18 31 23 8 49 6	26 4 31 44 15 14 41	47 18 31 23 8 42 6	26 4 31 37 15 14 48
14 48 5 39 17 30 22	49 20 12 39 24 2 29	14 41 5 46 17 30 22	42 20 12 46 24 2 29
16 29 28 13 47 4 38	37 22 7 34 47 18 10	16 29 28 13 40 4 45	44 22 7 34 40 18 10
46 3 37 15 35 27 12	32 45 16 8 42 27 5	39 3 44 15 35 27 12	32 38 16 8 49 27 5
34 26 11 45 2 36 21	13 40 25 3 30 43 21	34 26 11 38 2 43 21	13 47 25 3 30 36 21
2001/ 1/ 2160001#	2002/ 1/ 2162161#	2006/ 1/ 2164321#	2007/ 1/ 2166481#
1 42 48 19 11 31 23	1 35 20 40 25 10 44	1 42 34 19 11 45 23	1 28 48 12 18 31 37
17 9 29 28 6 40 46	24 9 43 7 34 19 39	17 9 43 28 6 40 32	45 9 15 35 41 5 25
26 4 38 44 15 14 34	33 18 38 23 8 49 6	26 4 38 30 15 14 48	19 32 38 2 22 49 13
49 20 12 32 24 2 36	14 48 5 32 17 37 22	35 20 12 46 24 2 36	42 6 26 46 10 16 29
30 22 7 41 47 18 10	16 36 28 13 47 4 31	44 22 7 41 33 18 10	23 43 14 20 33 39 3
39 45 16 8 35 27 5	46 3 30 15 42 27 12	39 31 16 8 49 27 5	11 17 30 36 7 27 47
13 33 25 3 37 43 21	41 26 11 45 2 29 21	13 47 25 3 37 29 21	34 40 4 24 44 8 21
2009/ 1/ 2168641#	2011/ 1/ 2170801#	2013/ 1/ 2172961#	2015/ 1/ 2175121#
1 49 41 19 11 31 23	1 28 34 12 18 38 44	1 49 34 19 11 38 23	1 28 41 12 18 31 44
17 9 29 28 6 47 39	31 9 15 42 48 5 25	17 9 36 28 6 47 32	38 9 15 35 48 5 25
26 4 45 37 15 14 34	19 39 45 2 22 35 13	26 4 45 30 15 14 41	19 32 45 2 22 42 13
42 20 12 32 24 2 43	49 6 26 32 10 16 36	35 20 12 39 24 2 43	49 6 26 39 10 16 29
30 22 7 48 40 18 10	23 29 14 20 40 46 3	37 22 7 48 33 18 10	23 36 14 20 33 46 3
46 38 16 8 35 27 5	11 17 37 43 7 27 33	46 31 16 8 42 27 5	11 17 30 43 7 27 40
13 33 25 3 44 36 21	41 47 4 24 30 8 21	13 40 25 3 44 29 21	34 47 4 24 37 8 21
2018/ 1/ 2177281#	2020/ 1/ 2179441#	2022/ 1/ 2181601#	2024/ 1/ 2183761#
1 35 20 12 46 24 37	1 14 41 19 25 31 44	1 42 20 12 46 24 30	1 14 34 19 25 38 44
45 23 36 7 34 19 11	17 23 29 49 6 12 39	45 23 29 7 41 19 11	17 23 36 49 6 12 32
33 18 10 44 22 42 6	47 4 10 37 15 28 34	40 18 10 44 22 35 6	47 4 10 30 15 28 41
28 41 5 32 17 9 43	42 20 26 32 45 2 8	28 34 5 39 17 9 43	35 20 26 39 45 2 8
16 8 49 27 40 4 31	30 43 7 13 40 18 24	16 8 49 27 33 4 38	37 43 7 13 33 18 24
39 3 30 15 14 48 26	11 38 16 22 35 48 5	32 3 37 15 14 48 26	11 31 16 22 42 48 5
13 47 25 38 2 29 21	27 33 46 3 9 36 21	13 47 25 31 2 36 21	27 40 46 3 9 29 21
2026/ 1/ 2185921#	2028/ 1/ 2188081#	2029/ 1/ 2190241#	2031/ 1/ 2192401#
1 35 41 19 25 10 44	1 14 20 33 46 24 37	1 49 41 26 18 10 30	1 35 13 19 25 38 44
17 23 8 49 6 33 39	45 23 36 7 13 19 32	38 23 15 14 34 5 46	17 23 36 49 6 33 11
47 4 31 37 15 28 13	12 18 31 44 22 42 6	19 11 31 2 43 42 27	47 4 31 9 15 28 41
42 20 26 11 45 2 29	28 41 5 11 17 30 43	35 6 47 39 24 16 8	14 20 26 39 45 2 29
9 43 7 34 40 18 24	16 29 49 27 40 4 10	44 36 28 20 12 32 3	37 43 7 34 12 18 24
32 38 16 22 14 48 5	39 3 9 15 35 48 26	25 17 9 29 7 48 40	32 10 16 22 42 48 5
27 12 46 3 30 36 21	34 47 25 38 2 8 21	13 33 4 45 37 22 21	27 40 46 3 30 8 21
. . . . .			
2498/ 1/ 2695681#	2499/ 1/ 2697841#	2502/ 1/ 2700001#	2503/ 1/ 2702161#
1 35 13 19 46 24 37	1 21 41 12 25 31 44	1 42 13 19 46 24 30	1 21 34 12 25 38 44
45 23 36 7 34 12 18	10 23 29 49 6 19 39	45 23 29 7 41 12 18	10 23 36 49 6 19 32
33 11 17 44 22 42 6	47 4 17 37 8 28 34	40 11 17 44 22 35 6	47 4 17 30 8 28 41
28 41 5 32 10 16 43	42 13 26 32 45 2 15	28 34 5 39 10 16 43	35 13 26 39 45 2 15
9 15 49 27 40 4 31	30 43 7 20 40 11 24	9 15 49 27 33 4 38	37 43 7 20 33 11 24
39 3 30 8 21 48 26	18 38 9 22 35 48 5	32 3 37 8 21 48 26	18 31 9 22 42 48 5
20 47 25 38 2 29 14	27 33 46 3 16 36 14	20 47 25 31 2 36 14	27 40 46 3 16 29 14
2506/ 1/ 2704321#	2507/ 1/ 2706481#	2509/ 1/ 2708641#	2511/ 1/ 2710801#
1 35 41 12 25 17 44	1 21 13 33 46 24 37	1 49 41 26 11 17 30	1 35 20 12 25 38 44
10 23 15 49 6 33 39	45 23 36 7 20 12 32	38 23 8 21 34 5 46	10 23 36 49 6 33 18

47 4 31 37 8 28 20 19 11 31 44 22 42 6 12 18 31 2 43 42 27 47 4 31 16 8 28 41  
 42 13 26 18 45 2 29 28 41 5 18 10 30 43 35 6 47 39 24 9 15 21 13 26 39 45 2 29  
 16 43 7 34 40 11 24 9 29 49 27 40 4 17 44 36 28 13 19 32 3 37 43 7 34 19 11 24  
 32 38 9 22 21 48 5 39 3 16 8 35 48 26 25 10 16 29 7 48 40 32 17 9 22 42 48 5  
 27 19 46 3 30 36 14 34 47 25 38 2 15 14 20 33 4 45 37 22 14 27 40 46 3 30 15 14  
 2513/ 1/ 2712961# 2514/ 1/ 2715121# 2517/ 1/ 2717281# 2518/ 1/ 2719441#  
 1 49 20 26 11 31 37 1 42 34 12 25 17 44 1 49 34 26 11 17 37 1 42 20 12 25 31 44  
 17 23 8 35 41 5 46 10 23 15 49 6 40 32 31 23 8 21 41 5 46 10 23 29 49 6 40 18  
 12 32 38 2 43 21 27 47 4 38 30 8 28 20 12 18 38 2 43 35 27 47 4 38 16 8 28 34  
 42 6 47 18 24 9 29 35 13 26 18 45 2 36 42 6 47 32 24 9 15 21 13 26 32 45 2 36  
 44 15 28 13 33 39 3 16 43 7 41 33 11 24 44 29 28 13 19 39 3 30 43 7 41 19 11 24  
 25 10 30 36 7 48 19 39 31 9 22 21 48 5 25 10 16 36 7 48 33 39 17 9 22 35 48 5  
 34 40 4 45 16 22 14 27 19 46 3 37 29 14 20 40 4 45 30 22 14 27 33 46 3 37 15 14  
 2522/ 1/ 2721601# 2523/ 1/ 2723761# 2525/ 1/ 2725921# 2527/ 1/ 2728081#  
 1 35 13 19 39 24 44 1 21 48 12 25 31 37 1 49 13 19 39 24 30 1 21 34 12 25 45 37  
 38 23 43 7 34 12 18 10 23 29 42 6 19 46 38 23 29 7 48 12 18 10 23 43 42 6 19 32  
 33 11 17 37 22 49 6 40 4 17 44 8 28 34 47 11 17 37 22 35 6 40 4 17 30 8 28 48  
 28 48 5 32 10 16 36 49 13 26 32 38 2 15 28 34 5 46 10 16 36 35 13 26 46 38 2 15  
 9 15 42 27 47 4 31 30 36 7 20 47 11 24 9 15 42 27 33 4 45 44 36 7 20 33 11 24  
 46 3 30 8 21 41 26 18 45 9 22 35 41 5 32 3 44 8 21 41 26 18 31 9 22 49 41 5  
 20 40 25 45 2 29 14 27 33 39 3 16 43 14 20 40 25 31 2 43 14 27 47 39 3 16 29 14  
 . . . . .  
 2833/ 1/ 3058561# 2836/ 2/ 3060721# 2837/ 1/ 3062881# 2840/ 2/ 3065041#  
 1 35 13 19 25 45 37 1 13 25 37 35 19 45 1 42 13 19 25 45 30 1 13 25 30 42 19 45  
 24 44 36 7 34 12 18 18 44 7 12 24 36 34 24 44 29 7 41 12 18 18 44 7 12 24 29 41  
 33 11 17 23 43 42 6 42 33 17 43 6 11 23 40 11 17 23 43 35 6 35 40 17 43 6 11 23  
 49 41 5 32 10 16 22 10 22 41 32 16 49 5 49 34 5 39 10 16 22 10 22 34 39 16 49 5  
 9 15 28 48 40 4 31 48 4 9 28 40 31 15 9 15 28 48 33 4 38 48 4 9 28 33 38 15  
 39 3 30 8 21 27 47 30 21 47 3 8 27 39 32 3 37 8 21 27 47 37 21 47 3 8 27 32  
 20 26 46 38 2 29 14 26 38 29 20 46 2 14 20 26 46 31 2 36 14 26 31 36 20 46 2 14  
 2843/ 1/ 3067201# 2844/ 2/ 3069361# 2847/ 1/ 3071521# 2848/ 2/ 3073681#  
 1 21 13 33 25 45 37 1 13 25 37 21 33 45 1 28 41 47 11 17 30 1 13 25 16 42 33 45  
 24 44 36 7 20 12 32 32 44 7 12 24 36 20 38 44 8 21 34 5 25 32 44 7 12 24 15 41  
 19 11 31 23 43 42 6 42 19 31 43 6 11 23 12 18 31 2 22 42 48 21 40 31 43 6 11 23  
 49 41 5 18 10 30 22 10 22 41 18 30 49 5 35 6 26 39 45 9 15 10 22 20 39 30 49 5  
 9 29 28 48 40 4 17 48 4 9 28 40 17 29 23 36 49 13 19 32 3 48 4 9 28 19 38 29  
 39 3 16 8 35 27 47 16 35 47 3 8 27 39 46 10 16 29 7 27 40 37 35 47 3 8 27 18  
 34 26 46 38 2 15 14 26 38 15 34 46 2 14 20 33 4 24 37 43 14 26 17 36 34 46 2 14  
 2850/ 1/ 3075841# 2852/ 2/ 3078001# 2855/ 1/ 3080161# 2856/ 2/ 3082321#  
 1 28 20 47 11 31 37 1 13 25 30 21 40 45 1 28 34 47 11 17 37 1 13 25 16 35 40 45  
 17 44 8 35 41 5 25 39 44 7 12 24 29 20 31 44 8 21 41 5 25 39 44 7 12 24 15 34  
 12 32 38 2 22 21 48 35 19 38 43 6 11 23 12 18 38 2 22 35 48 21 33 38 43 6 11 23  
 42 6 26 18 45 9 29 10 22 34 18 37 49 5 42 6 26 32 45 9 15 10 22 20 32 37 49 5  
 23 15 49 13 33 39 3 48 4 9 28 33 17 36 23 29 49 13 19 39 3 48 4 9 28 19 31 36  
 46 10 30 36 7 27 19 16 42 47 3 8 27 32 46 10 16 36 7 27 33 30 42 47 3 8 27 18  
 34 40 4 24 16 43 14 26 31 15 41 46 2 14 20 40 4 24 30 43 14 26 17 29 41 46 2 14  
 2857/ 1/ 3084481# 2860/ 2/ 3086641# 2861/ 1/ 3088801# 2864/ 2/ 3090961#  
 1 35 13 26 18 45 37 1 13 18 37 35 26 45 1 42 13 26 18 45 30 1 13 18 30 42 26 45  
 17 44 36 7 34 12 25 25 44 7 12 17 36 34 17 44 29 7 41 12 25 25 44 7 12 17 29 41  
 33 11 24 16 43 42 6 42 33 24 43 6 11 16 40 11 24 16 43 35 6 35 40 24 43 6 11 16  
 49 41 5 32 10 23 15 10 15 41 32 23 49 5 49 34 5 39 10 23 15 10 15 34 39 23 49 5  
 9 22 21 48 40 4 31 48 4 9 21 40 31 22 9 22 21 48 33 4 38 48 4 9 21 33 38 22  
 39 3 30 8 28 20 47 30 28 47 3 8 20 39 32 3 37 8 28 20 47 37 28 47 3 8 20 32  
 27 19 46 38 2 29 14 19 38 29 27 46 2 14 27 19 46 31 2 36 14 19 31 36 27 46 2 14  
 2866/ 1/ 3093121# 2868/ 2/ 3095281# 2871/ 1/ 3097441# 2872/ 2/ 3099601#  
 1 28 13 33 18 45 37 1 13 18 37 28 33 45 1 21 41 47 11 24 30 1 13 18 23 42 33 45  
 17 44 36 7 27 12 32 32 44 7 12 17 36 27 38 44 8 28 34 5 18 32 44 7 12 17 22 41  
 26 11 31 16 43 42 6 42 26 31 43 6 11 16 12 25 31 2 15 42 48 28 40 31 43 6 11 16  
 49 41 5 25 10 30 15 10 15 41 25 30 49 5 35 6 19 39 45 9 22 10 15 27 39 30 49 5  
 9 29 21 48 40 4 24 48 4 9 21 40 24 29 16 36 49 13 26 32 3 48 4 9 21 26 38 29  
 39 3 23 8 35 20 47 23 35 47 3 8 20 39 46 10 23 29 7 20 40 37 35 47 3 8 20 25  
 34 19 46 38 2 22 14 19 38 22 34 46 2 14 27 33 4 17 37 43 14 19 24 36 34 46 2 14

2875/ 1/ 3101761#	2876/ 2/ 3103921#	2879/ 1/ 3106081#	2880/ 2/ 3108241#
1 21 27 47 11 31 37	1 13 18 30 28 40 45	1 21 34 47 11 24 37	1 13 18 23 35 40 45
24 44 8 35 41 5 18	39 44 7 12 17 29 27	31 44 8 28 41 5 18	39 44 7 12 17 22 34
12 32 38 2 15 28 48	35 26 38 43 6 11 16	12 25 38 2 15 35 48	28 33 38 43 6 11 16
42 6 19 25 45 9 29	10 15 34 25 37 49 5	42 6 19 32 45 9 22	10 15 27 32 37 49 5
16 22 49 13 33 39 3	48 4 9 21 33 24 36	16 29 49 13 26 39 3	48 4 9 21 26 31 36
46 10 30 36 7 20 26	23 42 47 3 8 20 32	46 10 23 36 7 20 33	30 42 47 3 8 20 25
34 40 4 17 23 43 14	19 31 22 41 46 2 14	27 40 4 17 30 43 14	19 24 29 41 46 2 14

. . . . .  
 [Count(n1=1/All) = 3110400/38102400] OK!

\*\* Recompiled and Listed by Kanji Setsuda on  
 Mar.14, 2015 with MacOSX 10.10.2 and Xcode 6.2 \*\*

(Compiled and Listed by Kanji Setsuda in 2003, 2006, 2011, 2012;  
 Recompiled on Mar.14, 2015 with MacOSX 10.10.2 & Xcode 6.2)

---

E-mail Address <jag12001@nifty.com>  
<http://kanjisetuda.la.coocan.jp/>