Part 3: New Algebraic Study of Magic Squares: Kanji Setsuda
Chapter 9: Fundamental Study of **Composite** Magic Squares:
Section 7 : 'Semi-Composite' Type of Magic Squares 10x10

#0. Making the 'Semi-Composite' type of ordinary Magic Squares of Order 10

At first I must mention that this article continued from the "Semi-Composite type of Magic Squares 6x6" in the Section 6, Chapter 4, Part 3. If you have not read it yet, be kind enough to look it through before all, will you please?

I would like to make the same type of Order 10 here by the same method.

As you know, 10(=4x2+2) is 'Singly' even number, and the world of Magic Squares of Order 10 looks quite different from those of Order 4, 8, 12 and 16.

We can never make any Pan-diagonal MS10 here. We can build neither symmetric Self-complementary Type nor 'Composite' one of Order 10 at all.

While we can build the ordinary type of Standard MS10 here, it is very hard for us to build them when we actually want to have. The solution counts might be extremely big, far bigger than we have ever imagined, and we can hardly come up to the first solution in a short time even when we may use our fastest PC.

They really present only a few sample solutions in the recent Internet Sites.

We, earnest Magic Square builders, want to have any 'rare' and 'precious' set of objects of Order 10 and would even try to reform our definitions for that purpose.

Thus I built the Composite type of 'Semi-Pandiagonal' MS10 or the same type of 'Semi-Magic' Squares of Order 10 in the previous articles. But they were no longer the ordinary type of MS10 at all, since we could not make all the rows, columns and 2 primary diagonals add up to the same Constant Value at the same time.

How can we build the rare set of MS10 solutions with those essential conditions always adding up to the constant sum 505 at the same time, then?

What about making the 'Semi-Composite' type of MS10 by the same method with the case of Order 6? Let's make the experiment for them, shall we? We might expect to have the 'rare' solution set of ordinary type very quickly in a short time.

#1. New Experiment with New Definitions

** Basic Form for 'Composite' MS10 in the Extended Space **

```
 96 97 98 99 H0 91 92 93 94 95 96 97 98 99 H0 91 92 93 94 95
                .--.--.--.--.--.--.--.--.--.
  6  7  8  9 10| 1| 2| 3| 4| 5| 6| 7| 8| 9|10| 1  2  3  4  5
               |--+--+--+--+--+--+--+--+--+--|
 16 17 18 19 20|11|12|13|14|15|16|17|18|19|20|11 12 13 14 15
               |--+--+--+--+--+--+--+--+--+--|
 26 27 28 29 30|21|22|23|24|25|26|27|28|29|30|21 22 23 24 25
               |--+--+--+--+--+--+--+--+--+--|
 36 37 38 39 40|31|32|33|34|35|36|37|38|39|40|31 32 33 34 35
               |--+--+--+--+--+--+--+--+--+--|
 46 47 48 49 50|41|42|43|44|45|46|47|48|49|50|41 42 43 44 45
               |--+--+--+--+--+--+--+--+--+--|
 56 57 58 59 60|51|52|53|54|55|56|57|58|59|60|51 52 53 54 55
               |--+--+--+--+--+--+--+--+--+--|
 66 67 68 69 70|61|62|63|64|65|66|67|68|69|70|61 62 63 64 65
               |--+--+--+--+--+--+--+--+--+--|
 76 77 78 79 80|71|72|73|74|75|76|77|78|79|80|71 72 73 74 75
               |--+--+--+--+--+--+--+--+--+--|
 86 87 88 89 90|81|82|83|84|85|86|87|88|89|90|81 82 83 84 85
               |--+--+--+--+--+--+--+--+--+--|
 96 97 98 99 H0|91|92|93|94|95|96|97|98|99|H0|91 92 93 94 95
               '--'--'--'--'--'--'--'--'--'--'
  6  7  8  9 10  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5
                                    * 'H0'↑ above means '100' *
```

Inspired by my memorable success in the case of Order 6, I tried very hard to build the rare solution set of the 'Semi-Composite' Magic Squares 10x10.

After trial and error, I could have finally got the result I have wanted to have.

I would like to report about it here as precisely as possible.

I defined everything as usual by the Basic Diagram above and the Simultaneous Equations below for all the essential components: 'Composite Conditions', Basic Conditions for Rows, Columns and two Primary Diagonals; as presented as follows.

```
**  Composite Conditions: CC=202; **
   n1+n2+n11+n12=CC    ... cc1;  | n51+n52+n61+n62=CC   ... cc51;
   n2+n3+n12+n13=CC    ... cc2;  | n52+n53+n62+n63=CC   ... cc52;
   n3+n4+n13+n14=CC    ... cc3;  | n53+n54+n63+n64=CC   ... cc53;
   n4+n5+n14+n15=CC    ... cc4;  | n54+n55+n64+n65=CC   ... cc54;
   n5+n6+n15+n16=CC    ... cc5;  | n55+n56+n65+n66=CC   ... cc55;
  *n6+n7+n16+n17=CC    ... cc6;* | n56+n57+n66+n67=CC   ... cc56;
   n7+n8+n17+n18=CC    ... cc7;  | n57+n58+n67+n68=CC   ... cc57;
   n8+n9+n18+n19=CC    ... cc8;  | n58+n59+n68+n69=CC   ... cc58;
   n9+n10+n19+n20=CC   ... cc9;  | n59+n60+n69+n70=CC   ... cc59;
  *n10+n1+n20+n11=CC   ... cc10;*| n60+n51+n70+n61=CC   ... cc60;

  n11+n12+n21+n22=CC   ... cc11; | n61+n62+n71+n72=CC   ... cc61;
  n12+n13+n22+n23=CC   ... cc12; | n62+n63+n72+n73=CC   ... cc62;
  n13+n14+n23+n24=CC   ... cc13; | n63+n64+n73+n74=CC   ... cc63;
  n14+n15+n24+n25=CC   ... cc14; | n64+n65+n74+n75=CC   ... cc64;
  n15+n16+n25+n26=CC   ... cc15; | n65+n66+n75+n76=CC   ... cc65;
  n16+n17+n26+n27=CC   ... cc16; | n66+n67+n76+n77=CC   ... cc66;
  n17+n18+n27+n28=CC   ... cc17; | n67+n68+n77+n78=CC   ... cc67;
  n18+n19+n28+n29=CC   ... cc18; | n68+n69+n78+n79=CC   ... cc68;
  n19+n20+n29+n30=CC   ... cc19; | n69+n70+n79+n80=CC   ... cc69;
  n20+n11+n30+n21=CC   ... cc20; | n70+n61+n80+n71=CC   ... cc70;

  n21+n22+n31+n32=CC   ... cc21; | n71+n72+n81+n82=CC   ... cc71;
  n22+n23+n32+n33=CC   ... cc22; | n72+n73+n82+n83=CC   ... cc72;
  n23+n24+n33+n34=CC   ... cc23; | n73+n74+n83+n84=CC   ... cc73;
  n24+n25+n34+n35=CC   ... cc24; | n74+n75+n84+n85=CC   ... cc74;
  n25+n26+n35+n36=CC   ... cc25; | n75+n76+n85+n86=CC   ... cc75;
  n26+n27+n36+n37=CC   ... cc26; | n76+n77+n86+n87=CC   ... cc76;
  n27+n28+n37+n38=CC   ... cc27; | n77+n78+n87+n88=CC   ... cc77;
  n28+n29+n38+n39=CC   ... cc28; | n78+n79+n88+n89=CC   ... cc78;
  n29+n30+n39+n40=CC   ... cc29; | n79+n80+n89+n90=CC   ... cc79;
  n30+n21+n40+n31=CC   ... cc30; | n80+n71+n90+n81=CC   ... cc80;

  n31+n32+n41+n42=CC   ... cc31; | n81+n82+n91+n92=CC   ... cc81;
  n32+n33+n42+n43=CC   ... cc32; | n82+n83+n92+n93=CC   ... cc82;
  n33+n34+n43+n44=CC   ... cc33; | n83+n84+n93+n94=CC   ... cc83;
  n34+n35+n44+n45=CC   ... cc34; | n84+n85+n94+n95=CC   ... cc84;
  n35+n36+n45+n46=CC   ... cc35; | n85+n86+n95+n96=CC   ... cc85;
  n36+n37+n46+n47=CC   ... cc36; |*n86+n87+n96+n97=CC   ... cc86;*
  n37+n38+n47+n48=CC   ... cc37; | n87+n88+n97+n98=CC   ... cc87;
  n38+n39+n48+n49=CC   ... cc38; | n88+n89+n98+n99=CC   ... cc88;
  n39+n40+n49+n50=CC   ... cc39; | n89+n90+n99+n100=CC  ... cc89;
  n40+n31+n50+n41=CC   ... cc40; |*n90+n81+n100+n91=CC  ... cc90;*

  n41+n42+n51+n52=CC   ... cc41; | n91+n92+n1+n2=CC     ... cc91;
  n42+n43+n52+n53=CC   ... cc42; | n92+n93+n2+n3=CC     ... cc92;
  n43+n44+n53+n54=CC   ... cc43; | n93+n94+n3+n4=CC     ... cc93;
  n44+n45+n54+n55=CC   ... cc44; | n94+n95+n4+n5=CC     ... cc94;
  n45+n46+n55+n56=CC   ... cc45; | n95+n96+n5+n6=CC     ... cc95;
  n46+n47+n56+n57=CC   ... cc46; | n96+n97+n6+n7=CC     ... cc96;
  n47+n48+n57+n58=CC   ... cc47; | n97+n98+n7+n8=CC     ... cc97;
```

```
  n48+n49+n58+n59=CC   ... cc48; | n98+n99+n8+n9=CC     ... cc98;
  n49+n50+n59+n60=CC   ... cc49; | n99+n100+n9+n10=CC   ... cc99;
  n50+n41+n60+n51=CC   ... cc50; | n100+n91+n10+n1=CC   ... cc100;

**  Basic Conditions for Rows and Columns: C=505; **
   n1+n2+n3+n4+n5+n6+n7+n8+n9+n10=C             ...rw1
  n11+n12+n13+n14+n15+n16+n17+n18+n19+n20=C     ...rw2
  n21+n22+n23+n24+n25+n26+n27+n28+n29+n30=C     ...rw3
  n31+n32+n33+n34+n35+n36+n37+n38+n39+n40=C     ...rw4
  n41+n42+n43+n44+n45+n46+n47+n48+n49+n50=C     ...rw5
  n51+n52+n53+n54+n55+n56+n57+n58+n59+n60=C     ...rw6
  n61+n62+n63+n64+n65+n66+n67+n68+n69+n70=C     ...rw7
  n71+n72+n73+n74+n75+n76+n77+n78+n79+n80=C     ...rw8
  n81+n82+n83+n84+n85+n86+n87+n88+n89+n90=C     ...rw9
  n91+n92+n93+n94+n95+n96+n97+n98+n99+n100=C    ...rw10

  n1+n11+n21+n31+n41+n51+n61+n71+n81+n91=C      ...cl1
  n2+n12+n22+n32+n42+n52+n62+n72+n82+n92=C      ...cl2
  n3+n13+n23+n33+n43+n53+n63+n73+n83+n93=C      ...cl3
  n4+n14+n24+n34+n44+n54+n64+n74+n84+n94=C      ...cl4
  n5+n15+n25+n35+n45+n55+n65+n75+n85+n95=C      ...cl5
  n6+n16+n26+n36+n46+n56+n66+n76+n86+n96=C      ...cl6
  n7+n17+n27+n37+n47+n57+n67+n77+n87+n97=C      ...cl7
  n8+n18+n28+n38+n48+n58+n68+n78+n88+n98=C      ...cl8
  n9+n19+n29+n39+n49+n59+n69+n79+n89+n99=C      ...cl9
  n10+n20+n30+n40+n50+n60+n70+n80+n90+n100=C    ...cl10

**  Basic Conditions for 2 Primary Diagonals: C=505; **
  n1+n12+n23+n34+n45+n56+n67+n78+n89+n100=C     ...pd1;
  n10+n19+n28+n37+n46+n55+n64+n73+n82+n91=C     ...pb10;

**  List-forming Inequality Conditions for Standard Solutions **
  n1<n10; n1<n91; n1<n100 and n2>n11;
```

Under these definitions above we are going to build our object solutions.

You may wonder if we need so many equations as about 140 above. Yes, we do.

But we now reduce the following 4 equations to be treated as 'Un-defined': cc6, cc10, cc86 and cc90. It is because we cannot make any single solution for our object, if we have all the 100 Composite Conditions defined to be true.

We must sacrifice these 4 equations in order to realize the above 2 Conditions for Primary Diagonals to be true. But we expect that the rest 96 Composite Conditions could surely show their effective power to build the rare set of solutions very quickly.

In fact all the 20 Basic Conditions of Rows and Columns are not always necessary here, but only several ones need to be dictated to be true in our program.

But the last 2 Conditions for Primary Diagonals above are absolutely necessary to be dictated to be true there.

Let me show you a few examples here to present what our object is really like.

```
                                                  1#|Row|Clm\Pd1/Pd2|Composite 2x2
  1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
 90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
 21  79  23  78  25  77  22  80  24  76|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 55  47  53  48  51  49  54  46  52  50|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
 66  34  68  33  70  32  67  35  69  31|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 65  37  63  38  61  39  64  36  62  40|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
 71  29  73  28  75  27  72  30  74  26|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
 45  57  43  58  41  59  44  56  42  60|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 81  19  83  18  85  17  82  20  84  16|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
 10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

```
                                                13#|Row|Clm\Pd1/Pd2|Composite 2x2
  1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
 90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
 26  74  28  73  30  72  27  75  29  71|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 55  47  53  48  51  49  54  46  52  50|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
 56  44  58  43  60  42  57  45  59  41|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 70  32  68  33  66  34  69  31  67  35|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
 76  24  78  23  80  22  77  25  79  21|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
 40  62  38  63  36  64  39  61  37  65|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 81  19  83  18  85  17  82  20  84  16|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
 10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202

                                                25#|Row|Clm\Pd1/Pd2|Composite 2x2
  1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
 90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
 36  64  38  63  40  62  37  65  39  61|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 55  47  53  48  51  49  54  46  52  50|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
 56  44  58  43  60  42  57  45  59  41|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 80  22  78  23  76  24  79  21  77  25|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
 66  34  68  33  70  32  67  35  69  31|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
 30  72  28  73  26  74  29  71  27  75|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 81  19  83  18  85  17  82  20  84  16|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
 10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

#2. The Program List I have dictated

Before listing out the Program, I have to mention a few important things about it.

To tell the truth at first, I could not have actually got all the solutions of this type.

I had to stop my calculation on the way, because it might have taken too long a time to finish my counting. 10th order is already 'ultra high'.

I also had to wait for a long time before I could actually have got the first solution.

I am now hesitating to show you the slow, older list of my complete programs.

Let me present you the new one instead, the 'Fastest Version' of mine.

In the early steps of program I assigned certain constant values to some variables, as I was taught by my discovery of the first solution. I want you to find it quickly and to have many other sample solutions in a reasonably short time.

But I might have lost the possibility of finding any other groups of solutions by this decision. I would say I had to lose them to get another. I have wanted to have many good sample solutions, as quickly as possible. This is a matter of choice.

```
//** 'Semi-Composite' Type of Magic Squares of Order 10 **
//** 'SCmpstMS100.c': Dictated by Kanji Setsuda **
//**  on Jan.16, 2010, Jul.27 and Aug. 8, 2013  **
//**  working with MacOSX 10.8.4 & Xcode 4.6.3  **
//
#include <stdio.h>
//
//* Global Variables *
long int cnt, cnt2;
short cnt3;
short LSM, SMC;
short nm[101], uflg[101];
short csm[101], csm2[41];
//
//* Sub-Routiens: *
void stp01(void), stp02(void), stp03(void), stp04(void), stp05(void);
void stp06(void), stp07(void), stp08(void), stp09(void), stp10(void);
void stp11(void), stp12(void), stp13(void), stp14(void), stp15(void);
void stp16(void), stp17(void), stp18(void), stp19(void), stp20(void);
void stp21(void), stp22(void), stp23(void), stp24(void), stp25(void);
void stp26(void), stp27(void), stp28(void), stp29(void), stp30(void);
void stp31(void), stp32(void), stp33(void), stp34(void), stp35(void);
void stp36(void), stp37(void), stp38(void), stp39(void), stp40(void);
void stp41(void), stp42(void), stp43(void), stp44(void), stp45(void);
```

```
void stp46(void), stp47(void), stp48(void), stp49(void), stp50(void);
void stp51(void), stp52(void), stp53(void), stp54(void), stp55(void);
void stp56(void), stp57(void), stp58(void), stp59(void), stp60(void);
void stp61(void), stp62(void), stp63(void), stp64(void), stp65(void);
void stp66(void), stp67(void), stp68(void), stp69(void), stp70(void);
void stp71(void), stp72(void), stp73(void), stp74(void), stp75(void);
void stp76(void), stp77(void), stp78(void), stp79(void), stp80(void);
void stp81(void), stp82(void), stp83(void), stp84(void), stp85(void);
void stp86(void), stp87(void), stp88(void), stp89(void), stp90(void);
void stp91(void), stp92(void), stp93(void), stp94(void), stp95(void);
void stp96(void), stp97(void), stp98(void), stp99(void), stp100(void);
void ansprint(void);
void prans(void);
//
//* Sub-Routines 2: *
void chksms(void), chk2sms(void);
//
//* Main Program *
int main(){
 short n;
  LSM=505; SMC=202;
  printf("\n");
  printf("**  'Semi-Composite' Type of Magic Squares of Order 10:  **\n");
  printf("**  Abstract List of Standard Solutions with Check-Sums  **\n");
  for(n=0;n<101;n++){nm[n]=0; uflg[n]=0;}
  cnt=0; cnt3=0;
  stp01();   //* Begin the Calculations *
  printf(" [Count = %ld]  OK!\n",cnt);
  printf("\n");
  printf("** Calculated and Listed by Kanji Setsuda **\n");
  printf("**  on Aug. 8, 2013 with MacOSX & Xcode 4.6.3 **\n");
  printf("\n");
 return 0;
}
//
//* Begin the Calculations *
//* Level 1: *
//* Set N1=1 *
void stp01(){
 short a;
  for(a=1;a<2;a++){
    if(uflg[a]==0){
      nm[1]=a; uflg[a]=1;
      stp02();
      uflg[a]=0;}
  }
}
//* Set N100(>N1) *
void stp02(){
 short a;
  for(a=100;a>nm[1];a--){
    if(uflg[a]==0){
      nm[100]=a; uflg[a]=1;
      stp03();
      uflg[a]=0;}
  }
}
//* Set N10(>N1) *
void stp03(){
 short a;
  for(a=91;a>nm[1];a--){
    if(uflg[a]==0){
      nm[10]=a; uflg[a]=1;
      stp04();
      uflg[a]=0;}
  }
```

```
}
//* Set n91=SMC-n1-n10-n100 & n91>N1 *
void stp04(){
 short a;
  a=SMC-nm[1]-nm[10]-nm[100];
  if((nm[1]<a)&&(a<101)){
    if(uflg[a]==0){
      nm[91]=a; uflg[a]=1;
      stp05();
      uflg[a]=0;}}
}
//* Set N2 *
void stp05(){
 short a;
  for(a=99;a>97;a--){
    if(uflg[a]==0){
      nm[2]=a; uflg[a]=1;
      stp06();
      uflg[a]=0;}
  }
}
//* Set n92=SMC-n1-n2-n91 *
void stp06(){
 short a;
  a=SMC-nm[1]-nm[2]-nm[91];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[92]=a; uflg[a]=1;
      stp07();
      uflg[a]=0;}}
}
//* Set N3 *
void stp07(){
 short a;
  for(a=3;a<6;a++){
    if(uflg[a]==0){
      nm[3]=a; uflg[a]=1;
      stp08();
      uflg[a]=0;}
  }
}
//* Set n93=SMC-n2-n3-n92 *
void stp08(){
 short a;
  a=SMC-nm[2]-nm[3]-nm[92];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[93]=a; uflg[a]=1;
      stp09();
      uflg[a]=0;}}
}
//* Set N4 *
void stp09(){
 short a;
  for(a=100;a>0;a--){
    if(uflg[a]==0){
      nm[4]=a; uflg[a]=1;
      stp10();
      uflg[a]=0;}
  }
}
//* Set n94=SMC-n3-n4-n93 *
void stp10(){
 short a;
  a=SMC-nm[3]-nm[4]-nm[93];
  if((0<a)&&(a<101)){
```

```
     if(uflg[a]==0){
       nm[94]=a; uflg[a]=1;
       stp11();
       uflg[a]=0;}}
}
//* Set N5 *
void stp11(){
 short a;
  for(a=1;a<101;a++){
     if(uflg[a]==0){
       nm[5]=a; uflg[a]=1;
       stp12();
       uflg[a]=0;}
  }
}
//* Set n95=SMC-n4-n5-n94 *
void stp12(){
 short a;
  a=SMC-nm[4]-nm[5]-nm[94];
  if((0<a)&&(a<101)){
     if(uflg[a]==0){
       nm[95]=a; uflg[a]=1;
       stp13();
       uflg[a]=0;}}
}
//* Set N6 *
void stp13(){
 short a;
  for(a=100;a>0;a--){
     if(uflg[a]==0){
       nm[6]=a; uflg[a]=1;
       stp14();
       uflg[a]=0;}
  }
}
//* Set n96=SMC-n5-n6-n95 *
void stp14(){
 short a;
  a=SMC-nm[5]-nm[6]-nm[95];
  if((0<a)&&(a<101)){
     if(uflg[a]==0){
       nm[96]=a; uflg[a]=1;
       stp15();
       uflg[a]=0;}}
}
//* Set N7 *
void stp15(){
 short a;
  for(a=1;a<101;a++){
     if(uflg[a]==0){cnt2=0;
       nm[7]=a; uflg[a]=1;
       stp16();
       uflg[a]=0;}
  }
}
//* Set n97=SMC-n6-n7-n96 *
void stp16(){
 short a;
  a=SMC-nm[6]-nm[7]-nm[96];
  if((0<a)&&(a<101)){
     if(uflg[a]==0){
       nm[97]=a; uflg[a]=1;
       stp17();
       uflg[a]=0;}}
}
//* Set N8 *
```

```
void stp17(){
 short a;
  for(a=100;a>0;a--){
    if(uflg[a]==0){
      nm[8]=a; uflg[a]=1;
      stp18();
      uflg[a]=0;}
  }
}
//* Set n98=SMC-n7-n8-n97 *
void stp18(){
 short a;
  a=SMC-nm[7]-nm[8]-nm[97];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[98]=a; uflg[a]=1;
      stp19();
      uflg[a]=0;}}
}
//* Set n9=LSM-n1-n2-n3-n4-n5-n6-n7-n8-n10 *
void stp19(){
 short a;
  a=LSM-nm[1]-nm[2]-nm[3]-nm[4]-nm[5]-nm[6]-nm[7]-nm[8]-nm[10];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[9]=a; uflg[a]=1;
      stp20();
      uflg[a]=0;}}
}
//* Set n99=LSM-n91-n92-n93-n94-n95-n96-n97-n98-n100 *
void stp20(){
 short a,b,c;
  a=LSM-nm[91]-nm[92]-nm[93]-nm[94]-nm[95]-nm[96]-nm[97]-nm[98]-nm[100];
  b=SMC-nm[8]-nm[9]-nm[98];
  c=SMC-nm[9]-nm[10]-nm[100];
  if((0<a)&&(a<101)&&(a==b)){
    if((a==c)&&(uflg[a]==0)){
      nm[99]=a; uflg[a]=1;
      stp21();
      uflg[a]=0;}}
}
//
//* Level 2: *
//* Set N11(<N2) *
void stp21(){
 short a;
  for(a=nm[2]-1;a>0;a--){
    if(uflg[a]==0){if(cnt<8100){cnt2=0;}
      nm[11]=a; uflg[a]=1;
      stp22();
      uflg[a]=0;}
  }
}
//* Set n12=SMC-n1-n2-n11 *
void stp22(){
 short a;
  a=SMC-nm[1]-nm[2]-nm[11];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[12]=a; uflg[a]=1;
      stp23();
      uflg[a]=0;}}
}
//* Set n13=SMC-n2-n3-n12 *
void stp23(){
 short a;
```

```
   a=SMC-nm[2]-nm[3]-nm[12];
   if((0<a)&&(a<101)){
      if(uflg[a]==0){
        nm[13]=a; uflg[a]=1;
        stp24();
        uflg[a]=0;}}
}
//* Set n14=SMC-n3-n4-n13 *
void stp24(){
 short a;
   a=SMC-nm[3]-nm[4]-nm[13];
   if((0<a)&&(a<101)){
      if(uflg[a]==0){
        nm[14]=a; uflg[a]=1;
        stp25();
        uflg[a]=0;}}
}
//* Set n15=SMC-n4-n5-n14 *
void stp25(){
 short a;
   a=SMC-nm[4]-nm[5]-nm[14];
   if((0<a)&&(a<101)){
      if(uflg[a]==0){
        nm[15]=a; uflg[a]=1;
        stp26();
        uflg[a]=0;}}
}
//* Set n16=SMC-n5-n6-n15 *
void stp26(){
 short a;
   a=SMC-nm[5]-nm[6]-nm[15];
   if((0<a)&&(a<101)){
      if(uflg[a]==0){
        nm[16]=a; uflg[a]=1;
        stp27();
        uflg[a]=0;}}
}
//* Set n17=SMC+5-n6-n7-n16 *
void stp27(){
 short a;
   a=SMC+5-nm[6]-nm[7]-nm[16];
   if((0<a)&&(a<101)){
      if(uflg[a]==0){
        nm[17]=a; uflg[a]=1;
        stp28();
        uflg[a]=0;}}
}
//* Set n18=SMC-n7-n8-n17 *
void stp28(){
 short a;
   a=SMC-nm[7]-nm[8]-nm[17];
   if((0<a)&&(a<101)){
      if(uflg[a]==0){
        nm[18]=a; uflg[a]=1;
        stp29();
        uflg[a]=0;}}
}
//* Set n19=SMC-n8-n9-n18 *
void stp29(){
 short a;
   a=SMC-nm[8]-nm[9]-nm[18];
   if((0<a)&&(a<101)){
      if(uflg[a]==0){
        nm[19]=a; uflg[a]=1;
        stp30();
        uflg[a]=0;}}
```

```
}
//* Set n20=SMC-n9-n10-n19 *
void stp30(){
 short a,b,c;
  a=LSM-nm[11]-nm[12]-nm[13]-nm[14]-nm[15]-nm[16]-nm[17]-nm[18]-nm[19];
  b=SMC-nm[9]-nm[10]-nm[19];
  c=SMC-5-nm[10]-nm[1]-nm[11];
  if((0<a)&&(a<101)){
    if((a==b)&&(a==c)&&(uflg[a]==0)){
      nm[20]=a; uflg[a]=1;
      stp31();
      uflg[a]=0;}}
}
//
//* Set N90 *
void stp31(){
 short a;
  for(a=1;a<101;a++){
    if(uflg[a]==0){if(cnt<1000){cnt2=0;}
      nm[90]=a; uflg[a]=1;
      stp32();
      uflg[a]=0;}
  }
}
//* Set n89=SMC-n90-n99-n100 *
void stp32(){
 short a;
  a=SMC-nm[90]-nm[99]-nm[100];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[89]=a; uflg[a]=1;
      stp33();
      uflg[a]=0;}}
}
//* Set n88=SMC-n89-n98-n99 *
void stp33(){
 short a;
  a=SMC-nm[89]-nm[98]-nm[99];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[88]=a; uflg[a]=1;
      stp34();
      uflg[a]=0;}}
}
//* Set n87=SMC-n88-n97-n98 *
void stp34(){
 short a;
  a=SMC-nm[88]-nm[97]-nm[98];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[87]=a; uflg[a]=1;
      stp35();
      uflg[a]=0;}}
}
//* Set n86=SMC-5-n87-n96-n97 *
void stp35(){
 short a;
  a=SMC-5-nm[87]-nm[96]-nm[97];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[86]=a; uflg[a]=1;
      stp36();
      uflg[a]=0;}}
}
//* Set n85=SMC-n86-n95-n96 *
void stp36(){
```

```
 short a;
  a=SMC-nm[86]-nm[95]-nm[96];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[85]=a; uflg[a]=1;
      stp37();
      uflg[a]=0;}}
}
//* Set n84=SMC-n85-n94-n95 *
void stp37(){
 short a;
  a=SMC-nm[85]-nm[94]-nm[95];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[84]=a; uflg[a]=1;
      stp38();
      uflg[a]=0;}}
}
//* Set n83=SMC-n84-n93-n94 *
void stp38(){
 short a;
  a=SMC-nm[84]-nm[93]-nm[94];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[83]=a; uflg[a]=1;
      stp39();
      uflg[a]=0;}}
}
//* Set n82=SMC-n83-n92-n93 *
void stp39(){
 short a,b;
  a=SMC-nm[83]-nm[92]-nm[93];
  b=SMC-nm[12]-nm[19]-nm[89];
  if((0<a)&&(a<101)){
    if((a==b)&&(uflg[a]==0)){
      nm[82]=a; uflg[a]=1;
      stp40();
      uflg[a]=0;}}
}
//* Set n81=SMC-n82-n91-n92 *
void stp40(){
 short a,b,c;
  a=LSM-nm[82]-nm[83]-nm[84]-nm[85]-nm[86]-nm[87]-nm[88]-nm[89]-nm[90];
  b=SMC-nm[82]-nm[91]-nm[92];
  c=SMC+5-nm[90]-nm[91]-nm[100];
  if((0<a)&&(a<101)){
    if((a==b)&&(a==c)&&(uflg[a]==0)){
      nm[81]=a; uflg[a]=1;
      stp41();
      uflg[a]=0;}}
}
//* Level 3: *
//* Set N21 *
void stp41(){
 short a;
  for(a=1;a<101;a++){
    if(uflg[a]==0){
      nm[21]=a; uflg[a]=1;
      stp42();
      uflg[a]=0;}
  }
}
//* Set n22=SMC-n11-n12-n21 *
void stp42(){
 short a;
  a=SMC-nm[11]-nm[12]-nm[21];
```

```
   if((0<a)&&(a<101)){
     if(uflg[a]==0){
       nm[22]=a; uflg[a]=1;
       stp43();
       uflg[a]=0;}}
}
//* Set n23=SMC-n12-n13-n22 *
void stp43(){
 short a;
   a=SMC-nm[12]-nm[13]-nm[22];
   if((0<a)&&(a<101)){
     if(uflg[a]==0){
       nm[23]=a; uflg[a]=1;
       stp44();
       uflg[a]=0;}}
}
//* Set n24=SMC-n13-n14-n23 *
void stp44(){
 short a;
   a=SMC-nm[13]-nm[14]-nm[23];
   if((0<a)&&(a<101)){
     if(uflg[a]==0){
       nm[24]=a; uflg[a]=1;
       stp45();
       uflg[a]=0;}}
}
//* Set n25=SMC-n14-n15-n24 *
void stp45(){
 short a;
   a=SMC-nm[14]-nm[15]-nm[24];
   if((0<a)&&(a<101)){
     if(uflg[a]==0){
       nm[25]=a; uflg[a]=1;
       stp46();
       uflg[a]=0;}}
}
//* Set n26=SMC-n15-n16-n25 *
void stp46(){
 short a;
   a=SMC-nm[15]-nm[16]-nm[25];
   if((0<a)&&(a<101)){
     if(uflg[a]==0){
       nm[26]=a; uflg[a]=1;
       stp47();
       uflg[a]=0;}}
}
//* Set n27=SMC-n16-n17-n26 *
void stp47(){
 short a;
   a=SMC-nm[16]-nm[17]-nm[26];
   if((0<a)&&(a<101)){
     if(uflg[a]==0){
       nm[27]=a; uflg[a]=1;
       stp48();
       uflg[a]=0;}}
}
//* Set n28=SMC-n17-n18-n27 *
void stp48(){
 short a;
   a=SMC-nm[17]-nm[18]-nm[27];
   if((0<a)&&(a<101)){
     if(uflg[a]==0){
       nm[28]=a; uflg[a]=1;
       stp49();
       uflg[a]=0;}}
}
```

```
//* Set n29=SMC-n18-n19-n28 *
void stp49(){
 short a;
  a=SMC-nm[18]-nm[19]-nm[28];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[29]=a; uflg[a]=1;
      stp50();
      uflg[a]=0;}}
}
//* Set n30=SMC-n19-n20-n29 *
void stp50(){
 short a,b;
  a=SMC-nm[19]-nm[20]-nm[29];
  b=SMC-nm[20]-nm[21]-nm[11];
  if((0<a)&&(a<101)&&(a==b)){
    if(uflg[a]==0){
      nm[30]=a; uflg[a]=1;
      stp51();
      uflg[a]=0;}}
}
//
//* Set N80 *
void stp51(){
 short a;
  for(a=100;a>0;a--){
    if(uflg[a]==0){
      nm[80]=a; uflg[a]=1;
      stp52();
      uflg[a]=0;}
  }
}
//* Set n79=SMC-n80-n89-n90 *
void stp52(){
 short a;
  a=SMC-nm[80]-nm[89]-nm[90];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[79]=a; uflg[a]=1;
      stp53();
      uflg[a]=0;}}
}
//* Set n78=SMC-n79-n88-n89 *
void stp53(){
 short a;
  a=SMC-nm[79]-nm[88]-nm[89];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[78]=a; uflg[a]=1;
      stp54();
      uflg[a]=0;}}
}
//* Set n77=SMC-n78-n87-n88 *
void stp54(){
 short a;
  a=SMC-nm[78]-nm[87]-nm[88];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[77]=a; uflg[a]=1;
      stp55();
      uflg[a]=0;}}
}
//* Set n76=SMC-n77-n86-n87 *
void stp55(){
 short a;
  a=SMC-nm[77]-nm[86]-nm[87];
```

```
   if((0<a)&&(a<101)){
     if(uflg[a]==0){
       nm[76]=a; uflg[a]=1;
       stp56();
       uflg[a]=0;}}
}
//* Set n75=SMC-n76-n85-n86 *
void stp56(){
 short a;
   a=SMC-nm[76]-nm[85]-nm[86];
   if((0<a)&&(a<101)){
     if(uflg[a]==0){
       nm[75]=a; uflg[a]=1;
       stp57();
       uflg[a]=0;}}
}
//* Set n74=SMC-n75-n84-n85 *
void stp57(){
 short a;
   a=SMC-nm[75]-nm[84]-nm[85];
   if((0<a)&&(a<101)){
     if(uflg[a]==0){
       nm[74]=a; uflg[a]=1;
       stp58();
       uflg[a]=0;}}
}
//* Set n73=SMC-n74-n83-n84 *
void stp58(){
 short a,b;
   a=SMC-nm[74]-nm[83]-nm[84];
   b=SMC-nm[23]-nm[28]-nm[78];
   if((0<a)&&(a<101)){
     if((a==b)&&(uflg[a]==0)){
       nm[73]=a; uflg[a]=1;
       stp59();
       uflg[a]=0;}}
}
//* Set n72=SMC-n73-n82-n83 *
void stp59(){
 short a;
   a=SMC-nm[73]-nm[82]-nm[83];
   if((0<a)&&(a<101)){
     if(uflg[a]==0){
       nm[72]=a; uflg[a]=1;
       stp60();
       uflg[a]=0;}}
}
//* Set n71=SMC-n72-n81-n82 *
void stp60(){
 short a,b;
   a=SMC-nm[72]-nm[81]-nm[82];
   b=SMC-nm[80]-nm[81]-nm[90];
   if((0<a)&&(a<101)&&(a==b)){
     if(uflg[a]==0){
       nm[71]=a; uflg[a]=1;
       stp61();
       uflg[a]=0;}}
}
//* Level 4: *
//* Set N31 *
void stp61(){
 short a;
   for(a=1;a<101;a++){
     if(uflg[a]==0){
       nm[31]=a; uflg[a]=1;
       stp62();
```

```
        uflg[a]=0;}
   }
}
//* Set n32=SMC-n21-n22-n31 *
void stp62(){
 short a;
   a=SMC-nm[21]-nm[22]-nm[31];
   if((0<a)&&(a<101)){
      if(uflg[a]==0){
        nm[32]=a; uflg[a]=1;
        stp63();
        uflg[a]=0;}}
}
//* Set n33=SMC-n22-n23-n32 *
void stp63(){
 short a;
   a=SMC-nm[22]-nm[23]-nm[32];
   if((0<a)&&(a<101)){
      if(uflg[a]==0){
        nm[33]=a; uflg[a]=1;
        stp64();
        uflg[a]=0;}}
}
//* Set n34=SMC-n23-n24-n33 *
void stp64(){
 short a;
   a=SMC-nm[23]-nm[24]-nm[33];
   if((0<a)&&(a<101)){
      if(uflg[a]==0){
        nm[34]=a; uflg[a]=1;
        stp65();
        uflg[a]=0;}}
}
//* Set n35=SMC-n24-n25-n34 *
void stp65(){
 short a;
   a=SMC-nm[24]-nm[25]-nm[34];
   if((0<a)&&(a<101)){
      if(uflg[a]==0){
        nm[35]=a; uflg[a]=1;
        stp66();
        uflg[a]=0;}}
}
//* Set n36=SMC-n25-n26-n35 *
void stp66(){
 short a;
   a=SMC-nm[25]-nm[26]-nm[35];
   if((0<a)&&(a<101)){
      if(uflg[a]==0){
        nm[36]=a; uflg[a]=1;
        stp67();
        uflg[a]=0;}}
}
//* Set n37=SMC-n26-n27-n36 *
void stp67(){
 short a;
   a=SMC-nm[26]-nm[27]-nm[36];
   if((0<a)&&(a<101)){
      if(uflg[a]==0){
        nm[37]=a; uflg[a]=1;
        stp68();
        uflg[a]=0;}}
}
//* Set n38=SMC-n27-n28-n37 *
void stp68(){
 short a;
```

```
   a=SMC-nm[27]-nm[28]-nm[37];
   if((0<a)&&(a<101)){
      if(uflg[a]==0){
        nm[38]=a; uflg[a]=1;
        stp69();
        uflg[a]=0;}}
}
//* Set n39=SMC-n28-n29-n38 *
void stp69(){
 short a;
   a=SMC-nm[28]-nm[29]-nm[38];
   if((0<a)&&(a<101)){
      if(uflg[a]==0){
        nm[39]=a; uflg[a]=1;
        stp70();
        uflg[a]=0;}}
}
//* Set n40=SMC-n29-n30-n39 *
void stp70(){
 short a,b;
   a=SMC-nm[29]-nm[30]-nm[39];
   b=SMC-nm[21]-nm[30]-nm[31];
   if((0<a)&&(a<101)&&(a==b)){
      if(uflg[a]==0){
        nm[40]=a; uflg[a]=1;
        stp71();
        uflg[a]=0;}}
}
//
//* Set N70 *
void stp71(){
 short a;
   for(a=1;a<101;a++){
      if(uflg[a]==0){
        nm[70]=a; uflg[a]=1;
        stp72();
        uflg[a]=0;}
   }
}
//* Set n69=SMC-n70-n79-n80 *
void stp72(){
 short a;
   a=SMC-nm[70]-nm[79]-nm[80];
   if((0<a)&&(a<101)){
      if(uflg[a]==0){
        nm[69]=a; uflg[a]=1;
        stp73();
        uflg[a]=0;}}
}
//* Set n68=SMC-n69-n78-n79 *
void stp73(){
 short a;
   a=SMC-nm[69]-nm[78]-nm[79];
   if((0<a)&&(a<101)){
      if(uflg[a]==0){
        nm[68]=a; uflg[a]=1;
        stp74();
        uflg[a]=0;}}
}
//* Set n67=SMC-n68-n77-n78 *
void stp74(){
 short a;
   a=SMC-nm[68]-nm[77]-nm[78];
   if((0<a)&&(a<101)){
      if(uflg[a]==0){
        nm[67]=a; uflg[a]=1;
```

```
       stp75();
       uflg[a]=0;}}
}
//* Set n66=SMC-n67-n76-n77 *
void stp75(){
 short a;
  a=SMC-nm[67]-nm[76]-nm[77];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[66]=a; uflg[a]=1;
      stp76();
      uflg[a]=0;}}
}
//* Set n65=SMC-n66-n75-n76 *
void stp76(){
 short a;
  a=SMC-nm[66]-nm[75]-nm[76];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[65]=a; uflg[a]=1;
      stp77();
      uflg[a]=0;}}
}
//* Set n64=SMC-n65-n74-n75 *
void stp77(){
 short a,b;
  a=SMC-nm[65]-nm[74]-nm[75];
  b=SMC-nm[34]-nm[37]-nm[67];
  if((0<a)&&(a<101)){
    if((a==b)&&(uflg[a]==0)){
      nm[64]=a; uflg[a]=1;
      stp78();
      uflg[a]=0;}}
}
//* Set n63=SMC-n64-n73-n74 *
void stp78(){
 short a;
  a=SMC-nm[64]-nm[73]-nm[74];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[63]=a; uflg[a]=1;
      stp79();
      uflg[a]=0;}}
}
//* Set n62=SMC-n63-n72-n73 *
void stp79(){
 short a;
  a=SMC-nm[63]-nm[72]-nm[73];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[62]=a; uflg[a]=1;
      stp80();
      uflg[a]=0;}}
}
//* Set n61=SMC-n62-n71-n72 *
void stp80(){
 short a,b;
  a=SMC-nm[62]-nm[71]-nm[72];
  b=SMC-nm[70]-nm[71]-nm[80];
  if((0<a)&&(a<101)&&(a==b)){
    if(uflg[a]==0){
      nm[61]=a; uflg[a]=1;
      stp81();
      uflg[a]=0;}}
}
//* Level 5: *
```

```
//* Set N41 *
void stp81(){
 short a;
  for(a=100;a>0;a--){
    if(uflg[a]==0){
      nm[41]=a; uflg[a]=1;
      stp82();
      uflg[a]=0;}
  }
}
//* Set n51=LSM-n1-n11-n21-n31-n41-n61-n71-n81-n91 *
void stp82(){
 short a;
  a=LSM-nm[1]-nm[11]-nm[21]-nm[31]-nm[41]-nm[61]-nm[71]-nm[81]-nm[91];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[51]=a; uflg[a]=1;
      stp83();
      uflg[a]=0;}}
}
//* Set n42=SMC-n31-n32-n41 *
void stp83(){
 short a;
  a=SMC-nm[31]-nm[32]-nm[41];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[42]=a; uflg[a]=1;
      stp84();
      uflg[a]=0;}}
}
//* Set n52=SMC-n41-n42-n51 *
void stp84(){
 short a,b;
  a=SMC-nm[41]-nm[42]-nm[51];
  b=SMC-nm[51]-nm[61]-nm[62];
  if((0<a)&&(a<101)&&(a==b)){
    if(uflg[a]==0){
      nm[52]=a; uflg[a]=1;
      stp85();
      uflg[a]=0;}}
}
//* Set n43=SMC-n32-n33-n42 *
void stp85(){
 short a;
  a=SMC-nm[32]-nm[33]-nm[42];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[43]=a; uflg[a]=1;
      stp86();
      uflg[a]=0;}}
}
//* Set n53=SMC-n42-n43-n52 *
void stp86(){
 short a,b;
  a=SMC-nm[42]-nm[43]-nm[52];
  b=SMC-nm[52]-nm[62]-nm[63];
  if((0<a)&&(a<101)&&(a==b)){
    if(uflg[a]==0){
      nm[53]=a; uflg[a]=1;
      stp87();
      uflg[a]=0;}}
}
//* Set n44=SMC-n33-n34-n43 *
void stp87(){
 short a;
  a=SMC-nm[33]-nm[34]-nm[43];
```

```
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[44]=a; uflg[a]=1;
      stp88();
      uflg[a]=0;}}
}
//* Set n54=SMC-n43-n44-n53 *
void stp88(){
 short a,b;
  a=SMC-nm[43]-nm[44]-nm[53];
  b=SMC-nm[53]-nm[63]-nm[64];
  if((0<a)&&(a<101)&&(a==b)){
    if(uflg[a]==0){
      nm[54]=a; uflg[a]=1;
      stp89();
      uflg[a]=0;}}
}
//* Set n45=SMC-n34-n35-n44 *
void stp89(){
 short a;
  a=SMC-nm[34]-nm[35]-nm[44];
  if((0<a)&&(a<101)){
    if(uflg[a]==0){
      nm[45]=a; uflg[a]=1;
      stp90();
      uflg[a]=0;}}
}
//* Set n55=SMC-n44-n45-n54 *
void stp90(){
 short a,b;
  a=SMC-nm[44]-nm[45]-nm[54];
  b=SMC-nm[54]-nm[64]-nm[65];
  if((0<a)&&(a<101)&&(a==b)){
    if(uflg[a]==0){
      nm[55]=a; uflg[a]=1;
      stp91();
      uflg[a]=0;}}
}
//* Set n56=LSM-n1-n12-n23-n34-n45-n67-n78-n89-n100 *
void stp91(){
 short a,b;
  a=LSM-nm[1]-nm[12]-nm[23]-nm[34]-nm[45]-nm[67]-nm[78]-nm[89]-nm[100];
  b=SMC-nm[55]-nm[65]-nm[66];
  if((0<a)&&(a<101)&&(a==b)){
    if(uflg[a]==0){
      nm[56]=a; uflg[a]=1;
      stp92();
      uflg[a]=0;}}
}
//* Set n46=LSM-n10-n19-n28-n37-n55-n64-n73-n82-n91 *
void stp92(){
 short a,b,c;
  a=LSM-nm[10]-nm[19]-nm[28]-nm[37]-nm[55]-nm[64]-nm[73]-nm[82]-nm[91];
  b=SMC-nm[35]-nm[36]-nm[45];
  c=SMC-nm[45]-nm[55]-nm[56];
  if((0<a)&&(a<101)&&(a==b)){
    if((a==c)&&(uflg[a]==0)){
      nm[46]=a; uflg[a]=1;
      stp93();
      uflg[a]=0;}}
}
//* Set n47=SMC-n36-n37-n46 *
void stp93(){
 short a;
  a=SMC-nm[36]-nm[37]-nm[46];
  if((0<a)&&(a<101)){
```

```
     if(uflg[a]==0){
       nm[47]=a; uflg[a]=1;
       stp94();
       uflg[a]=0;}}
}
//* Set n57=SMC-n46-n47-n56 *
void stp94(){
 short a,b;
  a=SMC-nm[46]-nm[47]-nm[56];
  b=SMC-nm[56]-nm[66]-nm[67];
  if((0<a)&&(a<101)&&(a==b)){
     if(uflg[a]==0){
       nm[57]=a; uflg[a]=1;
       stp95();
       uflg[a]=0;}}
}
//* Set n48=SMC-n37-n38-n47 *
void stp95(){
 short a;
  a=SMC-nm[37]-nm[38]-nm[47];
  if((0<a)&&(a<101)){
     if(uflg[a]==0){
       nm[48]=a; uflg[a]=1;
       stp96();
       uflg[a]=0;}}
}
//* Set n58=SMC-n47-n48-n57 *
void stp96(){
 short a,b;
  a=SMC-nm[47]-nm[48]-nm[57];
  b=SMC-nm[57]-nm[67]-nm[68];
  if((0<a)&&(a<101)&&(a==b)){
     if(uflg[a]==0){
       nm[58]=a; uflg[a]=1;
       stp97();
       uflg[a]=0;}}
}
//* Set n49=SMC-n38-n39-n48 *
void stp97(){
 short a;
  a=SMC-nm[38]-nm[39]-nm[48];
  if((0<a)&&(a<101)){
     if(uflg[a]==0){
       nm[49]=a; uflg[a]=1;
       stp98();
       uflg[a]=0;}}
}
//* Set n59=SMC-n48-n49-n58 *
void stp98(){
 short a,b;
  a=SMC-nm[48]-nm[49]-nm[58];
  b=SMC-nm[58]-nm[68]-nm[69];
  if((0<a)&&(a<101)&&(a==b)){
     if(uflg[a]==0){
       nm[59]=a; uflg[a]=1;
       stp99();
       uflg[a]=0;}}
}
//* Set n50=LSM-n41-n42-n43-n44-n45-n46-n47-n48-n49 *
void stp99(){
 short a,b,c;
  a=LSM-nm[41]-nm[42]-nm[43]-nm[44]-nm[45]-nm[46]-nm[47]-nm[48]-nm[49];
  b=SMC-nm[39]-nm[40]-nm[49];
  c=SMC-nm[31]-nm[40]-nm[41];
  if((0<a)&&(a<101)&&(a==b)){
     if((a==c)&&(uflg[a]==0)){
```

```
        nm[50]=a; uflg[a]=1;
        stp100();
        uflg[a]=0;}}
}
//* Set n60=SMC-n49-n50-n59 *
void stp100(){
 short a,b,c,d;
   a=SMC-nm[49]-nm[50]-nm[59];
   b=SMC-nm[41]-nm[50]-nm[51];
   c=SMC-nm[59]-nm[69]-nm[70];
   d=SMC-nm[51]-nm[61]-nm[70];
   if((0<a)&&(a<101)&&(a==b)){
      if((a==c)&&(a==d)&&(uflg[a]==0)){
        nm[60]=a; uflg[a]=1;
        ansprint();
        uflg[a]=0;}}
}
//
//* Print The Answers *
void ansprint(){
   cnt++; cnt2++;
   if(cnt2==1){
      chk2sms();    //* Check Sums of Rows, Columns and Pan-diagonals
      chksms();     //* Check Sums of Composite 2x2 Little Squares
      prans();
   }
}
//* Print the Answers *
void prans(){
 short l,l10,n;
   printf("%40ld|Row|Clm\\Pd1/Pd2|Composite 2x2\n",cnt);
   for(l=0;l<10;l++){l10=l*10;
      for(n=1;n<11;n++){printf("%4d",nm[l10+n]);}
      printf("|%3d|%3d",csm2[l+1],csm2[l+11]);
      printf("\\%3d/%3d|",csm2[l+21],csm2[l+31]);
      for(n=1;n<11;n++){printf("%4d",csm[l10+n]);}
      printf("\n");
   }
   printf("\n");
}
//
//** Check Sums of Composite Fours **
void chksms(){
   csm[ 1]=nm[1]+nm[2]+nm[11]+nm[12];
   csm[ 2]=nm[2]+nm[3]+nm[12]+nm[13];
   csm[ 3]=nm[3]+nm[4]+nm[13]+nm[14];
   csm[ 4]=nm[4]+nm[5]+nm[14]+nm[15];
   csm[ 5]=nm[5]+nm[6]+nm[15]+nm[16];
   csm[ 6]=nm[6]+nm[7]+nm[16]+nm[17];
   csm[ 7]=nm[7]+nm[8]+nm[17]+nm[18];
   csm[ 8]=nm[8]+nm[9]+nm[18]+nm[19];
   csm[ 9]=nm[9]+nm[10]+nm[19]+nm[20];
   csm[10]=nm[10]+nm[1]+nm[20]+nm[11];
   csm[11]=nm[11]+nm[12]+nm[21]+nm[22];
   csm[12]=nm[12]+nm[13]+nm[22]+nm[23];
   csm[13]=nm[13]+nm[14]+nm[23]+nm[24];
   csm[14]=nm[14]+nm[15]+nm[24]+nm[25];
   csm[15]=nm[15]+nm[16]+nm[25]+nm[26];
   csm[16]=nm[16]+nm[17]+nm[26]+nm[27];
   csm[17]=nm[17]+nm[18]+nm[27]+nm[28];
   csm[18]=nm[18]+nm[19]+nm[28]+nm[29];
   csm[19]=nm[19]+nm[20]+nm[29]+nm[30];
   csm[20]=nm[20]+nm[11]+nm[30]+nm[21];
   csm[21]=nm[21]+nm[22]+nm[31]+nm[32];
   csm[22]=nm[22]+nm[23]+nm[32]+nm[33];
   csm[23]=nm[23]+nm[24]+nm[33]+nm[34];
```

```
csm[24]=nm[24]+nm[25]+nm[34]+nm[35];
csm[25]=nm[25]+nm[26]+nm[35]+nm[36];
csm[26]=nm[26]+nm[27]+nm[36]+nm[37];
csm[27]=nm[27]+nm[28]+nm[37]+nm[38];
csm[28]=nm[28]+nm[29]+nm[38]+nm[39];
csm[29]=nm[29]+nm[30]+nm[39]+nm[40];
csm[30]=nm[30]+nm[21]+nm[40]+nm[31];
csm[31]=nm[31]+nm[32]+nm[41]+nm[42];
csm[32]=nm[32]+nm[33]+nm[42]+nm[43];
csm[33]=nm[33]+nm[34]+nm[43]+nm[44];
csm[34]=nm[34]+nm[35]+nm[44]+nm[45];
csm[35]=nm[35]+nm[36]+nm[45]+nm[46];
csm[36]=nm[36]+nm[37]+nm[46]+nm[47];
csm[37]=nm[37]+nm[38]+nm[47]+nm[48];
csm[38]=nm[38]+nm[39]+nm[48]+nm[49];
csm[39]=nm[39]+nm[40]+nm[49]+nm[50];
csm[40]=nm[40]+nm[31]+nm[50]+nm[41];
csm[41]=nm[41]+nm[42]+nm[51]+nm[52];
csm[42]=nm[42]+nm[43]+nm[52]+nm[53];
csm[43]=nm[43]+nm[44]+nm[53]+nm[54];
csm[44]=nm[44]+nm[45]+nm[54]+nm[55];
csm[45]=nm[45]+nm[46]+nm[55]+nm[56];
csm[46]=nm[46]+nm[47]+nm[56]+nm[57];
csm[47]=nm[47]+nm[48]+nm[57]+nm[58];
csm[48]=nm[48]+nm[49]+nm[58]+nm[59];
csm[49]=nm[49]+nm[50]+nm[59]+nm[60];
csm[50]=nm[50]+nm[41]+nm[60]+nm[51];
csm[51]=nm[51]+nm[52]+nm[61]+nm[62];
csm[52]=nm[52]+nm[53]+nm[62]+nm[63];
csm[53]=nm[53]+nm[54]+nm[63]+nm[64];
csm[54]=nm[54]+nm[55]+nm[64]+nm[65];
csm[55]=nm[55]+nm[56]+nm[65]+nm[66];
csm[56]=nm[56]+nm[57]+nm[66]+nm[67];
csm[57]=nm[57]+nm[58]+nm[67]+nm[68];
csm[58]=nm[58]+nm[59]+nm[68]+nm[69];
csm[59]=nm[59]+nm[60]+nm[69]+nm[70];
csm[60]=nm[60]+nm[51]+nm[70]+nm[61];
csm[61]=nm[61]+nm[62]+nm[71]+nm[72];
csm[62]=nm[62]+nm[63]+nm[72]+nm[73];
csm[63]=nm[63]+nm[64]+nm[73]+nm[74];
csm[64]=nm[64]+nm[65]+nm[74]+nm[75];
csm[65]=nm[65]+nm[66]+nm[75]+nm[76];
csm[66]=nm[66]+nm[67]+nm[76]+nm[77];
csm[67]=nm[67]+nm[68]+nm[77]+nm[78];
csm[68]=nm[68]+nm[69]+nm[78]+nm[79];
csm[69]=nm[69]+nm[70]+nm[79]+nm[80];
csm[70]=nm[70]+nm[61]+nm[80]+nm[71];
csm[71]=nm[71]+nm[72]+nm[81]+nm[82];
csm[72]=nm[72]+nm[73]+nm[82]+nm[83];
csm[73]=nm[73]+nm[74]+nm[83]+nm[84];
csm[74]=nm[74]+nm[75]+nm[84]+nm[85];
csm[75]=nm[75]+nm[76]+nm[85]+nm[86];
csm[76]=nm[76]+nm[77]+nm[86]+nm[87];
csm[77]=nm[77]+nm[78]+nm[87]+nm[88];
csm[78]=nm[78]+nm[79]+nm[88]+nm[89];
csm[79]=nm[79]+nm[80]+nm[89]+nm[90];
csm[80]=nm[80]+nm[71]+nm[90]+nm[81];
csm[81]=nm[81]+nm[82]+nm[91]+nm[92];
csm[82]=nm[82]+nm[83]+nm[92]+nm[93];
csm[83]=nm[83]+nm[84]+nm[93]+nm[94];
csm[84]=nm[84]+nm[85]+nm[94]+nm[95];
csm[85]=nm[85]+nm[86]+nm[95]+nm[96];
csm[86]=nm[86]+nm[87]+nm[96]+nm[97];
csm[87]=nm[87]+nm[88]+nm[97]+nm[98];
csm[88]=nm[88]+nm[89]+nm[98]+nm[99];
csm[89]=nm[89]+nm[90]+nm[99]+nm[100];
```

```
   csm[90]=nm[90]+nm[81]+nm[100]+nm[91];
   csm[91]=nm[91]+nm[92]+nm[1]+nm[2];
   csm[92]=nm[92]+nm[93]+nm[2]+nm[3];
   csm[93]=nm[93]+nm[94]+nm[3]+nm[4];
   csm[94]=nm[94]+nm[95]+nm[4]+nm[5];
   csm[95]=nm[95]+nm[96]+nm[5]+nm[6];
   csm[96]=nm[96]+nm[97]+nm[6]+nm[7];
   csm[97]=nm[97]+nm[98]+nm[7]+nm[8];
   csm[98]=nm[98]+nm[99]+nm[8]+nm[9];
   csm[99]=nm[99]+nm[100]+nm[9]+nm[10];
   csm[100]=nm[100]+nm[91]+nm[10]+nm[1];
};
//
//** Check Sums of Rows and Columns **
void chk2sms(){
   csm2[ 1]=nm[1]+nm[2]+nm[3]+nm[4]+nm[5]+nm[6]+nm[7]+nm[8]+nm[9]+nm[10];
   csm2[ 2]=nm[11]+nm[12]+nm[13]+nm[14]+nm[15]+nm[16]+nm[17]+nm[18]+nm[19]+nm[20];
   csm2[ 3]=nm[21]+nm[22]+nm[23]+nm[24]+nm[25]+nm[26]+nm[27]+nm[28]+nm[29]+nm[30];
   csm2[ 4]=nm[31]+nm[32]+nm[33]+nm[34]+nm[35]+nm[36]+nm[37]+nm[38]+nm[39]+nm[40];
   csm2[ 5]=nm[41]+nm[42]+nm[43]+nm[44]+nm[45]+nm[46]+nm[47]+nm[48]+nm[49]+nm[50];
   csm2[ 6]=nm[51]+nm[52]+nm[53]+nm[54]+nm[55]+nm[56]+nm[57]+nm[58]+nm[59]+nm[60];
   csm2[ 7]=nm[61]+nm[62]+nm[63]+nm[64]+nm[65]+nm[66]+nm[67]+nm[68]+nm[69]+nm[70];
   csm2[ 8]=nm[71]+nm[72]+nm[73]+nm[74]+nm[75]+nm[76]+nm[77]+nm[78]+nm[79]+nm[80];
   csm2[ 9]=nm[81]+nm[82]+nm[83]+nm[84]+nm[85]+nm[86]+nm[87]+nm[88]+nm[89]+nm[90];
   csm2[10]=nm[91]+nm[92]+nm[93]+nm[94]+nm[95]+nm[96]+nm[97]+nm[98]+nm[99]+nm[100];
   //
   csm2[11]=nm[1]+nm[11]+nm[21]+nm[31]+nm[41]+nm[51]+nm[61]+nm[71]+nm[81]+nm[91];
   csm2[12]=nm[2]+nm[12]+nm[22]+nm[32]+nm[42]+nm[52]+nm[62]+nm[72]+nm[82]+nm[92];
   csm2[13]=nm[3]+nm[13]+nm[23]+nm[33]+nm[43]+nm[53]+nm[63]+nm[73]+nm[83]+nm[93];
   csm2[14]=nm[4]+nm[14]+nm[24]+nm[34]+nm[44]+nm[54]+nm[64]+nm[74]+nm[84]+nm[94];
   csm2[15]=nm[5]+nm[15]+nm[25]+nm[35]+nm[45]+nm[55]+nm[65]+nm[75]+nm[85]+nm[95];
   csm2[16]=nm[6]+nm[16]+nm[26]+nm[36]+nm[46]+nm[56]+nm[66]+nm[76]+nm[86]+nm[96];
   csm2[17]=nm[7]+nm[17]+nm[27]+nm[37]+nm[47]+nm[57]+nm[67]+nm[77]+nm[87]+nm[97];
   csm2[18]=nm[8]+nm[18]+nm[28]+nm[38]+nm[48]+nm[58]+nm[68]+nm[78]+nm[88]+nm[98];
   csm2[19]=nm[9]+nm[19]+nm[29]+nm[39]+nm[49]+nm[59]+nm[69]+nm[79]+nm[89]+nm[99];
   csm2[20]=nm[10]+nm[20]+nm[30]+nm[40]+nm[50]+nm[60]+nm[70]+nm[80]+nm[90]+nm[100];
 //
 //** Check Sums of Pan-Diagonals **
   csm2[21]=nm[1]+nm[12]+nm[23]+nm[34]+nm[45]+nm[56]+nm[67]+nm[78]+nm[89]+nm[100];
   csm2[22]=nm[2]+nm[13]+nm[24]+nm[35]+nm[46]+nm[57]+nm[68]+nm[79]+nm[90]+nm[91];
   csm2[23]=nm[3]+nm[14]+nm[25]+nm[36]+nm[47]+nm[58]+nm[69]+nm[80]+nm[81]+nm[92];
   csm2[24]=nm[4]+nm[15]+nm[26]+nm[37]+nm[48]+nm[59]+nm[70]+nm[71]+nm[82]+nm[93];
   csm2[25]=nm[5]+nm[16]+nm[27]+nm[38]+nm[49]+nm[60]+nm[61]+nm[72]+nm[83]+nm[94];
   csm2[26]=nm[6]+nm[17]+nm[28]+nm[39]+nm[50]+nm[51]+nm[62]+nm[73]+nm[84]+nm[95];
   csm2[27]=nm[7]+nm[18]+nm[29]+nm[40]+nm[41]+nm[52]+nm[63]+nm[74]+nm[85]+nm[96];
   csm2[28]=nm[8]+nm[19]+nm[30]+nm[31]+nm[42]+nm[53]+nm[64]+nm[75]+nm[86]+nm[97];
   csm2[29]=nm[9]+nm[20]+nm[21]+nm[32]+nm[43]+nm[54]+nm[65]+nm[76]+nm[87]+nm[98];
   csm2[30]=nm[10]+nm[11]+nm[22]+nm[33]+nm[44]+nm[55]+nm[66]+nm[77]+nm[88]+nm[99];
 //
   csm2[31]=nm[1]+nm[20]+nm[29]+nm[38]+nm[47]+nm[56]+nm[65]+nm[74]+nm[83]+nm[92];
   csm2[32]=nm[2]+nm[11]+nm[30]+nm[39]+nm[48]+nm[57]+nm[66]+nm[75]+nm[84]+nm[93];
   csm2[33]=nm[3]+nm[12]+nm[21]+nm[40]+nm[49]+nm[58]+nm[67]+nm[76]+nm[85]+nm[94];
   csm2[34]=nm[4]+nm[13]+nm[22]+nm[31]+nm[50]+nm[59]+nm[68]+nm[77]+nm[86]+nm[95];
   csm2[35]=nm[5]+nm[14]+nm[23]+nm[32]+nm[41]+nm[60]+nm[69]+nm[78]+nm[87]+nm[96];
   csm2[36]=nm[6]+nm[15]+nm[24]+nm[33]+nm[42]+nm[51]+nm[70]+nm[79]+nm[88]+nm[97];
   csm2[37]=nm[7]+nm[16]+nm[25]+nm[34]+nm[43]+nm[52]+nm[61]+nm[80]+nm[89]+nm[98];
   csm2[38]=nm[8]+nm[17]+nm[26]+nm[35]+nm[44]+nm[53]+nm[62]+nm[71]+nm[90]+nm[99];
   csm2[39]=nm[9]+nm[18]+nm[27]+nm[36]+nm[45]+nm[54]+nm[63]+nm[72]+nm[81]+nm[100];
   csm2[40]=nm[10]+nm[19]+nm[28]+nm[37]+nm[46]+nm[55]+nm[64]+nm[73]+nm[82]+nm[91];
};
//
```

#3. Result List of Execution of this Program

```
**  'Semi-Composite' Type of Magic Squares of Order 10:  **
```

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

1#|Row|Clm\Pd1/Pd2|Composite 2x2

```
  1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
 90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
 21  79  23  78  25  77  22  80  24  76|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 55  47  53  48  51  49  54  46  52  50|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
 66  34  68  33  70  32  67  35  69  31|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 65  37  63  38  61  39  64  36  62  40|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
 71  29  73  28  75  27  72  30  74  26|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
 45  57  43  58  41  59  44  56  42  60|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 81  19  83  18  85  17  82  20  84  16|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
 10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

13#|Row|Clm\Pd1/Pd2|Composite 2x2

```
  1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
 90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
 26  74  28  73  30  72  27  75  29  71|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 55  47  53  48  51  49  54  46  52  50|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
 56  44  58  43  60  42  57  45  59  41|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 70  32  68  33  66  34  69  31  67  35|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
 76  24  78  23  80  22  77  25  79  21|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
 40  62  38  63  36  64  39  61  37  65|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 81  19  83  18  85  17  82  20  84  16|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
 10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

25#|Row|Clm\Pd1/Pd2|Composite 2x2

```
  1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
 90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
 36  64  38  63  40  62  37  65  39  61|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 55  47  53  48  51  49  54  46  52  50|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
 56  44  58  43  60  42  57  45  59  41|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 80  22  78  23  76  24  79  21  77  25|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
 66  34  68  33  70  32  67  35  69  31|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
 30  72  28  73  26  74  29  71  27  75|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 81  19  83  18  85  17  82  20  84  16|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
 10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

37#|Row|Clm\Pd1/Pd2|Composite 2x2

```
  1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
 90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
 41  59  43  58  45  57  42  60  44  56|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 65  37  63  38  61  39  64  36  62  40|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
 51  49  53  48  55  47  52  50  54  46|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 75  27  73  28  71  29  74  26  72  30|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
 66  34  68  33  70  32  67  35  69  31|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
 25  77  23  78  21  79  24  76  22  80|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 81  19  83  18  85  17  82  20  84  16|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
 10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

49#|Row|Clm\Pd1/Pd2|Composite 2x2

```
  1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
 90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
 51  49  53  48  55  47  52  50  54  46|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 65  37  63  38  61  39  64  36  62  40|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
 41  59  43  58  45  57  42  60  44  56|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 75  27  73  28  71  29  74  26  72  30|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
 66  34  68  33  70  32  67  35  69  31|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
 25  77  23  78  21  79  24  76  22  80|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 81  19  83  18  85  17  82  20  84  16|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
 10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

61#|Row|Clm\Pd1/Pd2|Composite 2x2

```
  1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
 90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
 56  44  58  43  60  42  57  45  59  41|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 55  47  53  48  51  49  54  46  52  50|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
 36  64  38  63  40  62  37  65  39  61|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 80  22  78  23  76  24  79  21  77  25|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
 66  34  68  33  70  32  67  35  69  31|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
 30  72  28  73  26  74  29  71  27  75|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
 81  19  83  18  85  17  82  20  84  16|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
 10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

85#|Row|Clm\Pd1/Pd2|Composite 2x2

```
 1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
66  34  68  33  70  32  67  35  69  31|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
65  37  63  38  61  39  64  36  62  40|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
41  59  43  58  45  57  42  60  44  56|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
75  27  73  28  71  29  74  26  72  30|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
51  49  53  48  55  47  52  50  54  46|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
25  77  23  78  21  79  24  76  22  80|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
81  19  83  18  85  17  82  20  84  16|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

121#|Row|Clm\Pd1/Pd2|Composite 2x2

```
 1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
71  29  73  28  75  27  72  30  74  26|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
55  47  53  48  51  49  54  46  52  50|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
21  79  23  78  25  77  22  80  24  76|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
65  37  63  38  61  39  64  36  62  40|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
66  34  68  33  70  32  67  35  69  31|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
45  57  43  58  41  59  44  56  42  60|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
81  19  83  18  85  17  82  20  84  16|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

133#|Row|Clm\Pd1/Pd2|Composite 2x2

```
 1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
76  24  78  23  80  22  77  25  79  21|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
55  47  53  48  51  49  54  46  52  50|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
26  74  28  73  30  72  27  75  29  71|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
70  32  68  33  66  34  69  31  67  35|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
56  44  58  43  60  42  57  45  59  41|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
40  62  38  63  36  64  39  61  37  65|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
81  19  83  18  85  17  82  20  84  16|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

145#|Row|Clm\Pd1/Pd2|Composite 2x2

```
 1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
26  74  28  73  30  72  27  75  29  71|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
55  47  53  48  51  49  54  46  52  50|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
56  44  58  43  60  42  57  45  59  41|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
70  32  68  33  66  34  69  31  67  35|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
81  19  83  18  85  17  82  20  84  16|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
40  62  38  63  36  64  39  61  37  65|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
76  24  78  23  80  22  77  25  79  21|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

217#|Row|Clm\Pd1/Pd2|Composite 2x2

```
 1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
21  79  23  78  25  77  22  80  24  76|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
55  47  53  48  51  49  54  46  52  50|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
66  34  68  33  70  32  67  35  69  31|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
65  37  63  38  61  39  64  36  62  40|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
81  19  83  18  85  17  82  20  84  16|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
45  57  43  58  41  59  44  56  42  60|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
71  29  73  28  75  27  72  30  74  26|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

325#|Row|Clm\Pd1/Pd2|Composite 2x2

```
 1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
21  79  23  78  25  77  22  80  24  76|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
55  47  53  48  51  49  54  46  52  50|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
71  29  73  28  75  27  72  30  74  26|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
65  37  63  38  61  39  64  36  62  40|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
81  19  83  18  85  17  82  20  84  16|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
45  57  43  58  41  59  44  56  42  60|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
66  34  68  33  70  32  67  35  69  31|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

505#|Row|Clm\Pd1/Pd2|Composite 2x2

```
 1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
41  59  43  58  45  57  42  60  44  56|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
55  47  53  48  51  49  54  46  52  50|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
66  34  68  33  70  32  67  35  69  31|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
85  17  83  18  81  19  84  16  82  20|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
71  29  73  28  75  27  72  30  74  26|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
25  77  23  78  21  79  24  76  22  80|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
61  39  63  38  65  37  62  40  64  36|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

577#|Row|Clm\Pd1/Pd2|Composite 2x2

```
 1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
26  74  28  73  30  72  27  75  29  71|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
55  47  53  48  51  49  54  46  52  50|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
76  24  78  23  80  22  77  25  79  21|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
70  32  68  33  66  34  69  31  67  35|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
81  19  83  18  85  17  82  20  84  16|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
40  62  38  63  36  64  39  61  37  65|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
56  44  58  43  60  42  57  45  59  41|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

721#|Row|Clm\Pd1/Pd2|Composite 2x2

```
 1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
41  59  43  58  45  57  42  60  44  56|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
65  37  63  38  61  39  64  36  62  40|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
66  34  68  33  70  32  67  35  69  31|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
75  27  73  28  71  29  74  26  72  30|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
81  19  83  18  85  17  82  20  84  16|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
25  77  23  78  21  79  24  76  22  80|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
51  49  53  48  55  47  52  50  54  46|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

757#|Row|Clm\Pd1/Pd2|Composite 2x2

```
 1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
56  44  58  43  60  42  57  45  59  41|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
65  37  63  38  61  39  64  36  62  40|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
66  34  68  33  70  32  67  35  69  31|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
80  22  78  23  76  24  79  21  77  25|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
71  29  73  28  75  27  72  30  74  26|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
20  82  18  83  16  84  19  81  17  85|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
46  54  48  53  50  52  47  55  49  51|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

829#|Row|Clm\Pd1/Pd2|Composite 2x2

```
 1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
51  49  53  48  55  47  52  50  54  46|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
65  37  63  38  61  39  64  36  62  40|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
66  34  68  33  70  32  67  35  69  31|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
75  27  73  28  71  29  74  26  72  30|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
81  19  83  18  85  17  82  20  84  16|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
25  77  23  78  21  79  24  76  22  80|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
41  59  43  58  45  57  42  60  44  56|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

901#|Row|Clm\Pd1/Pd2|Composite 2x2

```
 1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
56  44  58  43  60  42  57  45  59  41|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
55  47  53  48  51  49  54  46  52  50|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
66  34  68  33  70  32  67  35  69  31|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
80  22  78  23  76  24  79  21  77  25|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
81  19  83  18  85  17  82  20  84  16|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
30  72  28  73  26  74  29  71  27  75|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
36  64  38  63  40  62  37  65  39  61|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202
```

```
                                                937#|Row|Clm\Pd1/Pd2|Composite 2x2
 1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
56  44  58  43  60  42  57  45  59  41|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
55  47  53  48  51  49  54  46  52  50|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
76  24  78  23  80  22  77  25  79  21|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
70  32  68  33  66  34  69  31  67  35|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
81  19  83  18  85  17  82  20  84  16|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
40  62  38  63  36  64  39  61  37  65|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
26  74  28  73  30  72  27  75  29  71|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202

                                                973#|Row|Clm\Pd1/Pd2|Composite 2x2
 1  99   3  98   5  97   7  95   9  91|505|505\505/500| 202 202 202 202 202 207 202 202 202 197
90  12  88  13  86  14  89  11  87  15|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
66  34  68  33  70  32  67  35  69  31|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
55  47  53  48  51  49  54  46  52  50|505|505\510/510| 202 202 202 202 202 202 202 202 202 202
71  29  73  28  75  27  72  30  74  26|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
65  37  63  38  61  39  64  36  62  40|505|505\510/505| 202 202 202 202 202 202 202 202 202 202
81  19  83  18  85  17  82  20  84  16|505|505\505/510| 202 202 202 202 202 202 202 202 202 202
45  57  43  58  41  59  44  56  42  60|505|505\500/500| 202 202 202 202 202 202 202 202 202 202
21  79  23  78  25  77  22  80  24  76|505|505\510/510| 202 202 202 202 202 197 202 202 202 207
10  92   8  93   6  94   4  96   2 100|505|505\500/505| 202 202 202 202 202 202 202 202 202 202

                                1009#\0/0|Composite                                1729#\0/0|Composite
 1 99  3 98  5 97  7 95  9 91|0|0|00000+000-     1 99  3 98  5 97  7 95  9 91|0|0|00000+000-
85 17 83 18 81 19 84 16 82 20|0|0|0000000000    80 22 78 23 76 24 79 21 77 25|0|0|0000000000
21 79 23 78 25 77 22 80 24 76|0|0|0000000000    16 84 18 83 20 82 17 85 19 81|0|0|0000000000
55 47 53 48 51 49 54 46 52 50|0|0|0000000000    60 42 58 43 56 44 59 41 57 45|0|0|0000000000
61 39 63 38 65 37 62 40 64 36|0|0|0000000000    66 34 68 33 70 32 67 35 69 31|0|0|0000000000
70 32 68 33 66 34 69 31 67 35|0|0|0000000000    65 37 63 38 61 39 64 36 62 40|0|0|0000000000
71 29 73 28 75 27 72 30 74 26|0|0|0000000000    71 29 73 28 75 27 72 30 74 26|0|0|0000000000
45 57 43 58 41 59 44 56 42 60|0|0|0000000000    50 52 48 53 46 54 49 51 47 55|0|0|0000000000
86 14 88 13 90 12 87 15 89 11|0|0|00000-000+    86 14 88 13 90 12 87 15 89 11|0|0|00000-000+
10 92  8 93  6 94  4 96  2 H0|0|0|0000000000    10 92  8 93  6 94  4 96  2 H0|0|0|0000000000

                                2305#\0/0|Composite                                3025#\0/0|Composite
 1 99  3 98  5 97  7 95  9 91|0|0|00000+000-     1 99  3 98  5 97  7 95  9 91|0|0|00000+000-
75 27 73 28 71 29 74 26 72 30|0|0|0000000000    70 32 68 33 66 34 69 31 67 35|0|0|0000000000
16 84 18 83 20 82 17 85 19 81|0|0|0000000000    16 84 18 83 20 82 17 85 19 81|0|0|0000000000
60 42 58 43 56 44 59 41 57 45|0|0|0000000000    60 42 58 43 56 44 59 41 57 45|0|0|0000000000
61 39 63 38 65 37 62 40 64 36|0|0|0000000000    61 39 63 38 65 37 62 40 64 36|0|0|0000000000
70 32 68 33 66 34 69 31 67 35|0|0|0000000000    75 27 73 28 71 29 74 26 72 30|0|0|0000000000
76 24 78 23 80 22 77 25 79 21|0|0|0000000000    76 24 78 23 80 22 77 25 79 21|0|0|0000000000
50 52 48 53 46 54 49 51 47 55|0|0|0000000000    50 52 48 53 46 54 49 51 47 55|0|0|0000000000
86 14 88 13 90 12 87 15 89 11|0|0|00000-000+    86 14 88 13 90 12 87 15 89 11|0|0|00000-000+
10 92  8 93  6 94  4 96  2 H0|0|0|0000000000    10 92  8 93  6 94  4 96  2 H0|0|0|0000000000

                                4033#\0/0|Composite                                4753#\0/0|Composite
 1 99  3 98  5 97  7 95  9 91|0|0|00000+000-     1 99  3 98  5 97  7 95  9 91|0|0|00000+000-
65 37 63 38 61 39 64 36 62 40|0|0|0000000000    60 42 58 43 56 44 59 41 57 45|0|0|0000000000
16 84 18 83 20 82 17 85 19 81|0|0|0000000000    16 84 18 83 20 82 17 85 19 81|0|0|0000000000
60 42 58 43 56 44 59 41 57 45|0|0|0000000000    65 37 63 38 61 39 64 36 62 40|0|0|0000000000
66 34 68 33 70 32 67 35 69 31|0|0|0000000000    66 34 68 33 70 32 67 35 69 31|0|0|0000000000
80 22 78 23 76 24 79 21 77 25|0|0|0000000000    80 22 78 23 76 24 79 21 77 25|0|0|0000000000
71 29 73 28 75 27 72 30 74 26|0|0|0000000000    71 29 73 28 75 27 72 30 74 26|0|0|0000000000
50 52 48 53 46 54 49 51 47 55|0|0|0000000000    50 52 48 53 46 54 49 51 47 55|0|0|0000000000
86 14 88 13 90 12 87 15 89 11|0|0|00000-000+    86 14 88 13 90 12 87 15 89 11|0|0|00000-000+
10 92  8 93  6 94  4 96  2 H0|0|0|0000000000    10 92  8 93  6 94  4 96  2 H0|0|0|0000000000

                                5329#\0/0|Composite                                6049#\0/0|Composite
 1 99  3 98  5 97  7 95  9 91|0|0|00000+000-     1 99  3 98  5 97  7 95  9 91|0|0|00000+000-
55 47 53 48 51 49 54 46 52 50|0|0|0000000000    50 52 48 53 46 54 49 51 47 55|0|0|0000000000
21 79 23 78 25 77 22 80 24 76|0|0|0000000000    16 84 18 83 20 82 17 85 19 81|0|0|0000000000
70 32 68 33 66 34 69 31 67 35|0|0|0000000000    65 37 63 38 61 39 64 36 62 40|0|0|0000000000
61 39 63 38 65 37 62 40 64 36|0|0|0000000000    66 34 68 33 70 32 67 35 69 31|0|0|0000000000
85 17 83 18 81 19 84 16 82 20|0|0|0000000000    80 22 78 23 76 24 79 21 77 25|0|0|0000000000
71 29 73 28 75 27 72 30 74 26|0|0|0000000000    71 29 73 28 75 27 72 30 74 26|0|0|0000000000
45 57 43 58 41 59 44 56 42 60|0|0|0000000000    60 42 58 43 56 44 59 41 57 45|0|0|0000000000
86 14 88 13 90 12 87 15 89 11|0|0|00000-000+    86 14 88 13 90 12 87 15 89 11|0|0|00000-000+
10 92  8 93  6 94  4 96  2 H0|0|0|0000000000    10 92  8 93  6 94  4 96  2 H0|0|0|0000000000
```

6337#\0/0|Composite

```
 1 99  3 98  5 97  7 95  9 91|0|0|00000+000-
45 57 43 58 41 59 44 56 42 60|0|0|0000000000
21 79 23 78 25 77 22 80 24 76|0|0|0000000000
70 32 68 33 66 34 69 31 67 35|0|0|0000000000
61 39 63 38 65 37 62 40 64 36|0|0|0000000000
85 17 83 18 81 19 84 16 82 20|0|0|0000000000
71 29 73 28 75 27 72 30 74 26|0|0|0000000000
55 47 53 48 51 49 54 46 52 50|0|0|0000000000
86 14 88 13 90 12 87 15 89 11|0|0|00000-000+
10 92  8 93  6 94  4 96  2 H0|0|0|0000000000
```

6769#\0/0|Composite

```
 1 99  3 98  5 97  7 95  9 91|0|0|00000+000-
40 62 38 63 36 64 39 61 37 65|0|0|0000000000
26 74 28 73 30 72 27 75 29 71|0|0|0000000000
70 32 68 33 66 34 69 31 67 35|0|0|0000000000
51 49 53 48 55 47 52 50 54 46|0|0|0000000000
85 17 83 18 81 19 84 16 82 20|0|0|0000000000
76 24 78 23 80 22 77 25 79 21|0|0|0000000000
60 42 58 43 56 44 59 41 57 45|0|0|0000000000
86 14 88 13 90 12 87 15 89 11|0|0|00000-000+
10 92  8 93  6 94  4 96  2 H0|0|0|0000000000
```

7057#\0/0|Composite

```
 1 99  3 98  5 97  7 95  9 91|0|0|00000+000-
30 72 28 73 26 74 29 71 27 75|0|0|0000000000
36 64 38 63 40 62 37 65 39 61|0|0|0000000000
80 22 78 23 76 24 79 21 77 25|0|0|0000000000
51 49 53 48 55 47 52 50 54 46|0|0|0000000000
85 17 83 18 81 19 84 16 82 20|0|0|0000000000
66 34 68 33 70 32 67 35 69 31|0|0|0000000000
60 42 58 43 56 44 59 41 57 45|0|0|0000000000
86 14 88 13 90 12 87 15 89 11|0|0|00000-000+
10 92  8 93  6 94  4 96  2 H0|0|0|0000000000
```

7345#\0/0|Composite

```
 1 99  3 98  5 97  7 95  9 91|0|0|00000+000-
25 77 23 78 21 79 24 76 22 80|0|0|0000000000
41 59 43 58 45 57 42 60 44 56|0|0|0000000000
75 27 73 28 71 29 74 26 72 30|0|0|0000000000
51 49 53 48 55 47 52 50 54 46|0|0|0000000000
85 17 83 18 81 19 84 16 82 20|0|0|0000000000
61 39 63 38 65 37 62 40 64 36|0|0|0000000000
70 32 68 33 66 34 69 31 67 35|0|0|0000000000
86 14 88 13 90 12 87 15 89 11|0|0|00000-000+
10 92  8 93  6 94  4 96  2 H0|0|0|0000000000
```

7777#\0/0|Composite

```
 1 99  3 98  5 97  7 95  9 91|0|0|00000+000-
20 82 18 83 16 84 19 81 17 85|0|0|0000000000
46 54 48 53 50 52 47 55 49 51|0|0|0000000000
75 27 73 28 71 29 74 26 72 30|0|0|0000000000
56 44 58 43 60 42 57 45 59 41|0|0|0000000000
90 12 88 13 86 14 89 11 87 15|0|0|0000000000
61 39 63 38 65 37 62 40 64 36|0|0|0000000000
70 32 68 33 66 34 69 31 67 35|0|0|0000000000
76 24 78 23 80 22 77 25 79 21|0|0|00000-000+
10 92  8 93  6 94  4 96  2 H0|0|0|0000000000
```

8065#\0/0|Composite

```
 1 99  3 98  5 97  9 95  7 91|0|0|00000+000-
90 12 88 13 86 14 87 11 89 15|0|0|0000000000
21 79 23 78 25 77 24 80 22 76|0|0|0000000000
55 47 53 48 51 49 52 46 54 50|0|0|0000000000
66 34 68 33 70 32 69 35 67 31|0|0|0000000000
65 37 63 38 61 39 62 36 64 40|0|0|0000000000
71 29 73 28 75 27 74 30 72 26|0|0|0000000000
45 57 43 58 41 59 42 56 44 60|0|0|0000000000
81 19 83 18 85 17 84 20 82 16|0|0|00000-000+
10 92  8 93  6 94  2 96  4 H0|0|0|0000000000
```

16129#

```
 1 99  3 97  5 98  7 95  9 91
90 12 88 14 86 13 89 11 87 15
21 79 23 77 25 78 22 80 24 76
55 47 53 49 51 48 54 46 52 50
66 34 68 32 70 33 67 35 69 31
65 37 63 39 61 38 64 36 62 40
71 29 73 27 75 28 72 30 74 26
45 57 43 59 41 58 44 56 42 60
81 19 83 17 85 18 82 20 84 16
10 92  8 94  6 93  4 96  2 H0
```

24193#

```
 1 99  3 97  5 98  9 95  7 91
90 12 88 14 86 13 87 11 89 15
21 79 23 77 25 78 24 80 22 76
55 47 53 49 51 48 52 46 54 50
66 34 68 32 70 33 69 35 67 31
65 37 63 39 61 38 62 36 64 40
71 29 73 27 75 28 74 30 72 26
45 57 43 59 41 58 42 56 44 60
81 19 83 17 85 18 84 20 82 16
10 92  8 94  6 93  2 96  4 H0
```

32257#

```
 1 99  5 98  3 97  7 95  9 91
90 12 86 13 88 14 89 11 87 15
21 79 25 78 23 77 22 80 24 76
55 47 51 48 53 49 54 46 52 50
66 34 70 33 68 32 67 35 69 31
65 37 61 38 63 39 64 36 62 40
71 29 75 28 73 27 72 30 74 26
45 57 41 58 43 59 44 56 42 60
81 19 85 18 83 17 82 20 84 16
10 92  6 93  8 94  4 96  2 H0
```

40321#

```
 1 99  5 98  3 97  9 95  7 91
90 12 86 13 88 14 87 11 89 15
21 79 25 78 23 77 24 80 22 76
55 47 51 48 53 49 52 46 54 50
66 34 70 33 68 32 69 35 67 31
65 37 61 38 63 39 62 36 64 40
71 29 75 28 73 27 74 30 72 26
45 57 41 58 43 59 42 56 44 60
81 19 85 18 83 17 84 20 82 16
10 92  6 93  8 94  2 96  4 H0
```

48385#

```
 1 99  5 97  3 98  7 95  9 91
90 12 86 14 88 13 89 11 87 15
21 79 25 77 23 78 22 80 24 76
55 47 51 49 53 48 54 46 52 50
66 34 70 32 68 33 67 35 69 31
65 37 61 39 63 38 64 36 62 40
71 29 75 27 73 28 72 30 74 26
45 57 41 59 43 58 44 56 42 60
81 19 85 17 83 18 82 20 84 16
10 92  6 94  8 93  4 96  2 H0
```

56449#

```
 1 99  5 97  3 98  9 95  7 91
90 12 86 14 88 13 87 11 89 15
21 79 25 77 23 78 24 80 22 76
55 47 51 49 53 48 52 46 54 50
66 34 70 32 68 33 69 35 67 31
65 37 61 39 63 38 62 36 64 40
71 29 75 27 73 28 74 30 72 26
45 57 41 59 43 58 42 56 44 60
81 19 85 17 83 18 84 20 82 16
10 92  6 94  8 93  2 96  4 H0
```

64513#

```
 1 98  3 99  5 97  7 95  9 91
90 13 88 12 86 14 89 11 87 15
21 78 23 79 25 77 22 80 24 76
55 48 53 47 51 49 54 46 52 50
66 33 68 34 70 32 67 35 69 31
65 38 63 37 61 39 64 36 62 40
71 28 73 29 75 27 72 30 74 26
45 58 43 57 41 59 44 56 42 60
81 18 83 19 85 17 82 20 84 16
10 93  8 92  6 94  4 96  2 H0
```

72577#

```
 1 98  3 99  5 97  9 95  7 91
90 13 88 12 86 14 87 11 89 15
21 78 23 79 25 77 24 80 22 76
55 48 53 47 51 49 52 46 54 50
66 33 68 34 70 32 69 35 67 31
65 38 63 37 61 39 62 36 64 40
71 28 73 29 75 27 74 30 72 26
45 58 43 57 41 59 42 56 44 60
81 18 83 19 85 17 84 20 82 16
10 93  8 92  6 94  2 96  4 H0
```

80641#

```
 1 98  3 97  5 99  7 95  9 91
90 13 88 14 86 12 89 11 87 15
21 78 23 77 25 79 22 80 24 76
55 48 53 49 51 47 54 46 52 50
66 33 68 32 70 34 67 35 69 31
65 38 63 39 61 37 64 36 62 40
71 28 73 27 75 29 72 30 74 26
45 58 43 59 41 57 44 56 42 60
81 18 83 17 85 19 82 20 84 16
10 93  8 94  6 92  4 96  2 H0
```

```
                                88705#
 1 98  3 97  5 99  9 95  7 91
90 13 88 14 86 12 87 11 89 15
21 78 23 77 25 79 24 80 22 76
55 48 53 49 51 47 52 46 54 50
66 33 68 32 70 34 69 35 67 31
65 38 63 39 61 37 62 36 64 40
71 28 73 27 75 29 74 30 72 26
45 58 43 59 41 57 42 56 44 60
81 18 83 17 85 19 84 20 82 16
10 93  8 94  6 92  2 96  4 H0

                                96769#
 1 98  5 99  3 97  7 95  9 91
90 13 86 12 88 14 89 11 87 15
21 78 25 79 23 77 22 80 24 76
55 48 51 47 53 49 54 46 52 50
66 33 70 34 68 32 67 35 69 31
65 38 61 37 63 39 64 36 62 40
71 28 75 29 73 27 72 30 74 26
45 58 41 57 43 59 44 56 42 60
81 18 85 19 83 17 82 20 84 16
10 93  6 92  8 94  4 96  2 H0

                               104833#
 1 98  5 99  3 97  9 95  7 91
90 13 86 12 88 14 87 11 89 15
21 78 25 79 23 77 24 80 22 76
55 48 51 47 53 49 52 46 54 50
66 33 70 34 68 32 69 35 67 31
65 38 61 37 63 39 62 36 64 40
71 28 75 29 73 27 74 30 72 26
45 58 41 57 43 59 42 56 44 60
81 18 85 19 83 17 84 20 82 16
10 93  6 92  8 94  2 96  4 H0

                               112897#
 1 98  5 97  3 99  7 95  9 91
90 13 86 14 88 12 89 11 87 15
21 78 25 77 23 79 22 80 24 76
55 48 51 49 53 47 54 46 52 50
66 33 70 32 68 34 67 35 69 31
65 38 61 39 63 37 64 36 62 40
71 28 75 27 73 29 72 30 74 26
45 58 41 59 43 57 44 56 42 60
81 18 85 17 83 19 82 20 84 16
10 93  6 94  8 92  4 96  2 H0

                               120961#
 1 98  5 97  3 99  9 95  7 91
90 13 86 14 88 12 87 11 89 15
21 78 25 77 23 79 24 80 22 76
55 48 51 49 53 47 52 46 54 50
66 33 70 32 68 34 69 35 67 31
65 38 61 39 63 37 62 36 64 40
71 28 75 27 73 29 74 30 72 26
45 58 41 59 43 57 42 56 44 60
81 18 85 17 83 19 84 20 82 16
10 93  6 94  8 92  2 96  4 H0

                               129025#
 1 97  3 99  5 98  7 95  9 91
90 14 88 12 86 13 89 11 87 15
21 77 23 79 25 78 22 80 24 76
55 49 53 47 51 48 54 46 52 50
66 32 68 34 70 33 67 35 69 31
65 39 63 37 61 38 64 36 62 40
71 27 73 29 75 28 72 30 74 26
45 59 43 57 41 58 44 56 42 60
81 17 83 19 85 18 82 20 84 16
10 94  8 92  6 93  4 96  2 H0

                               137089#
 1 97  3 99  5 98  9 95  7 91
90 14 88 12 86 13 87 11 89 15
21 77 23 79 25 78 24 80 22 76
55 49 53 47 51 48 52 46 54 50
66 32 68 34 70 33 69 35 67 31
65 39 63 37 61 38 62 36 64 40
71 27 73 29 75 28 74 30 72 26
45 59 43 57 41 58 42 56 44 60
81 17 83 19 85 18 84 20 82 16
10 94  8 92  6 93  2 96  4 H0

                               145153#
 1 97  3 98  5 99  7 95  9 91
90 14 88 13 86 12 89 11 87 15
21 77 23 78 25 79 22 80 24 76
55 49 53 48 51 47 54 46 52 50
66 32 68 33 70 34 67 35 69 31
65 39 63 38 61 37 64 36 62 40
71 27 73 28 75 29 72 30 74 26
45 59 43 58 41 57 44 56 42 60
81 17 83 18 85 19 82 20 84 16
10 94  8 93  6 92  4 96  2 H0

                               153217#
 1 97  3 98  5 99  9 95  7 91
90 14 88 13 86 12 87 11 89 15
21 77 23 78 25 79 24 80 22 76
55 49 53 48 51 47 52 46 54 50
66 32 68 33 70 34 69 35 67 31
65 39 63 38 61 37 62 36 64 40
71 27 73 28 75 29 74 30 72 26
45 59 43 58 41 57 42 56 44 60
81 17 83 18 85 19 84 20 82 16
10 94  8 93  6 92  2 96  4 H0

                               161281#
 1 97  5 99  3 98  7 95  9 91
90 14 86 12 88 13 89 11 87 15
21 77 25 79 23 78 22 80 24 76
55 49 51 47 53 48 54 46 52 50
66 32 70 34 68 33 67 35 69 31
65 39 61 37 63 38 64 36 62 40
71 27 75 29 73 28 72 30 74 26
45 59 41 57 43 58 44 56 42 60
81 17 85 19 83 18 82 20 84 16
10 94  6 92  8 93  4 96  2 H0

                               169345#
 1 97  5 99  3 98  9 95  7 91
90 14 86 12 88 13 87 11 89 15
21 77 25 79 23 78 24 80 22 76
55 49 51 47 53 48 52 46 54 50
66 32 70 34 68 33 69 35 67 31
65 39 61 37 63 38 62 36 64 40
71 27 75 29 73 28 74 30 72 26
45 59 41 57 43 58 42 56 44 60
81 17 85 19 83 18 84 20 82 16
10 94  6 92  8 93  2 96  4 H0

                               177409#
 1 97  5 98  3 99  7 95  9 91
90 14 86 13 88 12 89 11 87 15
21 77 25 78 23 79 22 80 24 76
55 49 51 48 53 47 54 46 52 50
66 32 70 33 68 34 67 35 69 31
65 39 61 38 63 37 64 36 62 40
71 27 75 28 73 29 72 30 74 26
45 59 41 58 43 57 44 56 42 60
81 17 85 18 83 19 82 20 84 16
10 94  6 93  8 92  4 96  2 H0

. . . . . .
```

#4. Conclusion

How long should it take the time for all? I had to stop my counting before it came to an end. I am sorry that I could not really know the total solution count of this type.

10th order is indeed 'ultra high'. I was strongly impressed with that again.

But I have to say it actually took only 4 hours to print out all those 177 thousands sample solutions as listed above, as a result of my fastest program.

Our method of making the 'Semi-Composite' type could really show its great effect on sampling the vast data of ordinary Standard MS10 in a reasonably short time.

I hope it might be the good news for those who have wanted to get lots of sample solutions of MS10 ready for their further study.

Written by Kanji Setsuda on July 21 and August 8, 2013;
Working with MacOSX 10.8.4 and Xcode 4.6.3;
E-Mail Address: jag12001@nifty.com
