

Chapter 9: Fundamental Study of Composite Magic Squares: Kanji Setsuda

Section 2: Composite Type of Pan-Magic Squares of Order 8

#1. 'Composite' Conditions

The meaning of 'Composite' is just the same with the one of order 6.
We define it with so many simultaneous equations as the next list.

[Figure 1: Basic Form and Composite Conditions]

| | |
|--|--|
| n1 n2 n3 n4 n5 n6 n7 n8 n1 | $n1 + n2 + n9 + n10 = S \dots (1)$ |
| n9 10 11 12 13 14 15 16 n9 | $n9 + n10 + n17 + n18 = S \dots (2)$ |
| 17 18 19 20 21 22 23 24 17 | $n17 + n18 + n25 + n26 = S \dots (3)$ |
| 25 26 27 28 29 30 31 32 25 | $n25 + n26 + n33 + n34 = S \dots (4)$ |
| 33 34 35 36 37 38 39 40 33 | $n33 + n34 + n41 + n42 = S \dots (5)$ |
| 41 42 43 44 45 46 47 48 41 | $n41 + n42 + n49 + n50 = S \dots (6)$ |
| 49 50 51 52 53 54 55 56 49 | $n49 + n50 + n57 + n58 = S \dots (7)$ |
| 57 58 59 60 61 62 63 64 57 | $n57 + n58 + n1 + n2 = S \dots (8)$ |
| n1 n2 n3 n4 n5 n6 n7 n8 n1 | $n2 + n3 + n10 + n11 = S \dots (9)$ |
| n19 + n20 + n27 + n28 = S ... (19) | $n10 + n11 + n18 + n19 = S \dots (10)$ |
| n27 + n28 + n35 + n36 = S ... (20) | $n18 + n19 + n26 + n27 = S \dots (11)$ |
| n35 + n36 + n43 + n44 = S ... (21) | $n26 + n27 + n34 + n35 = S \dots (12)$ |
| n43 + n44 + n51 + n52 = S ... (22) | $n34 + n35 + n42 + n43 = S \dots (13)$ |
| n51 + n52 + n59 + n60 = S ... (23) | $n42 + n43 + n50 + n51 = S \dots (14)$ |
| n59 + n60 + n3 + n4 = S ... (24) | $n50 + n51 + n58 + n59 = S \dots (15)$ |
| n4 + n5 + n12 + n13 = S ... (25) | $n58 + n59 + n2 + n3 = S \dots (16)$ |
| n12 + n13 + n20 + n21 = S ... (26) | $n3 + n4 + n11 + n12 = S \dots (17)$ |
| n20 + n21 + n28 + n29 = S ... (27) | $n11 + n12 + n19 + n20 = S \dots (18)$ |
| n28 + n29 + n36 + n37 = S ... (28) | $n22 + n23 + n30 + n31 = S \dots (43)$ |
| n36 + n37 + n44 + n45 = S ... (29) | $n30 + n31 + n38 + n39 = S \dots (44)$ |
| n44 + n45 + n52 + n53 = S ... (30) | $n38 + n39 + n46 + n47 = S \dots (45)$ |
| n52 + n53 + n60 + n61 = S ... (31) | $n46 + n47 + n54 + n55 = S \dots (46)$ |
| n60 + n61 + n4 + n5 = S ... (32) | $n54 + n55 + n62 + n63 = S \dots (47)$ |
| n5 + n6 + n13 + n14 = S ... (33) | $n54 + n55 + n62 + n63 = S \dots (47)$ |
| n13 + n14 + n21 + n22 = S ... (34) | $n62 + n63 + n6 + n7 = S \dots (48)$ |
| n21 + n22 + n29 + n30 = S ... (35) | $n7 + n8 + n15 + n16 = S \dots (49)$ |
| n29 + n30 + n37 + n38 = S ... (36) | $n15 + n16 + n23 + n24 = S \dots (50)$ |
| n37 + n38 + n45 + n46 = S ... (37) | $n23 + n24 + n31 + n32 = S \dots (51)$ |
| n45 + n46 + n53 + n54 = S ... (38) | $n31 + n32 + n39 + n40 = S \dots (52)$ |
| n53 + n54 + n61 + n62 = S ... (39) | $n39 + n40 + n47 + n48 = S \dots (53)$ |
| n61 + n62 + n5 + n6 = S ... (40) | $n47 + n48 + n55 + n56 = S \dots (54)$ |
| n6 + n7 + n14 + n15 = S ... (41) | $n55 + n56 + n63 + n64 = S \dots (55)$ |
| n14 + n15 + n22 + n23 = S ... (42) | $n63 + n64 + n7 + n8 = S \dots (56)$ |
| | $n8 + n1 + n16 + n9 = S \dots (57)$ |
| | $n16 + n9 + n24 + n17 = S \dots (58)$ |
| | $n24 + n17 + n32 + n25 = S \dots (59)$ |
| | $n32 + n25 + n40 + n33 = S \dots (60)$ |
| | $n40 + n33 + n48 + n41 = S \dots (61)$ |
| | $n48 + n41 + n56 + n49 = S \dots (62)$ |
| | $n56 + n49 + n64 + n57 = S \dots (63)$ |
| | $n64 + n57 + n8 + n1 = S \dots (64)$ |
| | \dots Composite Conditions |

What properties can you find if you accept only these 64 composite conditions at first? Are all rows, columns and pan-diagonals classified into two groups according to their sums again just as in the case of order 6?

Let's make some algebraic study in the same way as follows.

[1] (1)+(3)+(5)+(7)

$$n_1 + n_2 + n_9 + n_{10} = S \dots (1)$$

$$n_{17} + n_{18} + n_{25} + n_{26} = S \dots (3)$$

$$n_{33} + n_{34} + n_{41} + n_{42} = S \dots (5)$$

$$+) n_{49} + n_{50} + n_{57} + n_{58} = S \dots (7)$$

$$\begin{array}{r} (n_1+n_9+n_{17}+n_{25}+n_{33}+n_{41}+n_{49}+n_{57}) + (n_2+n_{10}+n_{18}+n_{26}+n_{34}+n_{42}+n_{50}+n_{58}) = 4*S \\ \text{L1} \qquad \qquad \qquad + \qquad \qquad \qquad \text{L2} \qquad \qquad \qquad = 4*S \end{array}$$

(9)+(11)+(13)+(15)

$$n_2 + n_3 + n_{10} + n_{11} = S \dots (9)$$

$$n_{18} + n_{19} + n_{26} + n_{27} = S \dots (11)$$

$$n_{34} + n_{35} + n_{42} + n_{43} = S \dots (13)$$

$$+) n_{50} + n_{51} + n_{58} + n_{59} = S \dots (15)$$

$$\begin{array}{r} (n_2+n_{10}+n_{18}+n_{26}+n_{34}+n_{42}+n_{50}+n_{58}) + (n_3+n_{11}+n_{19}+n_{27}+n_{35}+n_{43}+n_{51}+n_{59}) = 4*S \\ \text{L2} \qquad \qquad \qquad + \qquad \qquad \qquad \text{L3} \qquad \qquad \qquad = 4*S \end{array}$$

(17)+(19)+(21)+(23)

$$\begin{array}{r} (n_3+n_{11}+n_{19}+n_{27}+n_{35}+n_{43}+n_{51}+n_{59}) + (n_4+n_{12}+n_{20}+n_{28}+n_{36}+n_{44}+n_{52}+n_{60}) = 4*S \\ \text{L3} + \text{L4} = 4*S \end{array}$$

(25)+(27)+(29)+(31)

$$\begin{array}{r} (n_4+n_{12}+n_{20}+n_{28}+n_{36}+n_{44}+n_{52}+n_{60}) + (n_5+n_{13}+n_{21}+n_{29}+n_{37}+n_{45}+n_{53}+n_{61}) = 4*S \\ \text{L4} + \text{L5} = 4*S \end{array}$$

.....

Therefore,

$$\begin{array}{l} \text{L1} = \text{L3} = \text{L5} = \text{L7} = \text{LS1}; \\ \text{L2} = \text{L4} = \text{L6} = \text{L8} = \text{LS2}; \quad (\text{LS1} + \text{LS2} = 4*S) \end{array}$$

All columns are classified into two groups according to their sums.

But you cannot yet say $\text{LS1} = \text{LS2} = 2*S = 260$.

[2] In the same way

(1)+(17)+(33)+(49)

$$n_1 + n_2 + n_9 + n_{10} = S \dots (1)$$

$$n_3 + n_4 + n_{11} + n_{12} = S \dots (17)$$

$$n_5 + n_6 + n_{13} + n_{14} = S \dots (33)$$

$$+) n_7 + n_8 + n_{15} + n_{16} = S \dots (49)$$

$$\begin{array}{r} (n_1+n_2+n_3+n_4+n_5+n_6+n_7+n_8) + (n_9+n_{10}+n_{11}+n_{12}+n_{13}+n_{14}+n_{15}+n_{16}) = 4*S \\ \text{R1} \qquad \qquad \qquad + \qquad \qquad \qquad \text{R9} \qquad \qquad \qquad = 4*S \end{array}$$

(2)+(18)+(34)+(50)

$$\begin{array}{r} (n_9+n_{10}+n_{11}+n_{12}+n_{13}+n_{14}+n_{15}+n_{16}) + (n_{17}+n_{18}+n_{19}+n_{20}+n_{21}+n_{22}+n_{23}+n_{24}) = 4*S \\ \text{R9} + \text{R17} = 4*S \end{array}$$

(3)+(19)+(35)+(51)

$$\begin{array}{r} (n_{17}+n_{18}+n_{19}+n_{20}+n_{21}+n_{22}+n_{23}+n_{24}) + (n_{25}+n_{26}+n_{27}+n_{28}+n_{29}+n_{30}+n_{31}+n_{32}) = 4*S \\ \text{R17} + \text{R25} = 4*S \end{array}$$

.....

Therefore

$$\begin{array}{l} \text{R1} = \text{R17} = \text{R33} = \text{R49} = \text{RS1}; \\ \text{R9} = \text{R25} = \text{R41} = \text{R57} = \text{RS2}; \quad (\text{RS1} + \text{RS2} = 4*S) \end{array}$$

All rows are classified into two groups according to their sums

But you cannot yet say $\text{RS1} = \text{RS2} = 260$.

If you want all rows and columns to be equal to the magic constant 260, you must add at least $\text{L1} = \text{R1} = 260$ to the first definition set before all by yourself.

[3] Before going to study pan-diagonals, we must study some other interesting properties here.

[Figure 1(again): Basic Form and Composite Conditions]

| | | |
|---|--|--|
| n1 n2 n3 n4 n5 n6 n7 n8 | n1+ n2+ n9+n10 = S | |
| n9 10 11 12 13 14 15 16 | n9+n10+n17+n18 = S --> n1+n2=n17+n18=P1 | |
| 17 18 19 20 21 22 23 24 | n17+n18+n25+n26 = S --> n9+n10=n25+n26=Q1 | |
| 25 26 27 28 29 30 31 32 | n25+n26+n33+n34 = S --> n17+n18=n33+n34=P1 | |
| 33 34 35 36 37 38 39 40 | n33+n34+n41+n42 = S --> n25+n26=n41+n42=Q1 | |
| 41 42 43 44 45 46 47 48 | n41+n42+n49+n50 = S --> n33+n34=n49+n50=P1 | |
| 49 50 51 52 53 54 55 56 | n49+n50+n57+n58 = S --> n41+n42=n57+n58=Q1 | |
| 57 58 59 60 61 62 63 64 | n57+n58+ n1+ n2 = S (P1+Q1=S) | |
| | n2+ n3+n10+n11 = S | |
| | n10+n11+n18+n19 = S --> n2+n3=n18+n19=P2 | |
| | n18+n19+n26+n27 = S --> n10+n11=n26+n27=Q2 | |
| | n26+n27+n34+n35 = S --> n18+n19=n34+n35=P2 | |
| | n34+n35+n42+n43 = S --> n26+n27=n42+n43=Q2 | |
| | n42+n43+n50+n51 = S --> n34+n35=n50+n51=P2 | |
| | n50+n51+n58+n59 = S --> n42+n43=n58+n59=Q2 | |
| | n58+n59+ n2+ n3 = S (P2+Q2=S) | |
| | n3+ n4+n11+n12 = S | |
| | n11+n12+n19+n20 = S --> n3+n4=n19+n20=P3 | |
| | n19+n20+n27+n28 = S --> n11+n12=n27+n28=Q3 | |
| | | |
| n19+n20=n35+n36=P3; n27+n28=n43+n44=Q3; n35+n36=n41+n52=P3; | | |
| n43+n44=n59+n60=Q3; (P3+Q3=S) | | |
| n4+n5=n20+n21=P4; n12+n13=n28+n29=Q4; n20+n21=n36+n37=P4; | | |
| n28+n29=n44+n45=Q4; n36+n37=n52+n53=P4; n44+n45=n60+n61=Q4; (P4+Q4=S) | | |
| n5+n6=n21+n22=P5; n13+n14=n29+n30=Q5; n21+n22=n37+n38=P5; | | |
| n29+n30=n45+n46=Q5; n37+n38=n53+n54=P5; n45+n46=n61+n62=Q5; (P5+Q5=S) | | |
| n6+n7=n22+n23=P6; n14+n15=n30+n31=Q6; n22+n23=n38+n39=P6; | | |
| n30+n31=n46+n47=Q6; n38+n39=n54+n55=P6; n46+n47=n62+n63=Q6; (P6+Q6=S) | | |
| n7+n8=n23+n24=P7; n15+n16=n31+n32=Q7; n23+n24=n39+n40=P7; | | |
| n31+n32=n47+n48=Q7; n39+n40=n55+n56=P7; n47+n48=n63+n64=Q7; (P7+Q7=S) | | |
| | | |
| n1+ n9+ n2+n10 = S | | |
| n2+n10+ n3+n11 = S --> n1+n9=n3+n11=M1 | | |
| n3+n11+ n4+n12 = S --> n2+n10=n4+n12=N1 | | |
| n4+n12+ n5+n13 = S --> n3+n11=n5+n13=M1 | | |
| n5+n13+ n6+n14 = S --> n4+n12=n6+n14=N1 | | |
| n6+n14+ n7+n15 = S --> n5+n13=n7+n15=M1 | | |
| n7+n15+ n8+n16 = S --> n6+n14=n8+n16=N1 (M1+N1=S) | | |
| | | |
| n9+n17=n11+n19=M2; n10+n18=n12+n20=N2; n11+n19=n13+n21=M2; | | |
| n12+n20=n14+n22=N2; n13+n21=n15+n23=M2; n14+n22=n16+n24=N2; (M2+N2=S) | | |
| n17+n25=n19+n27=M3; n18+n26=n20+n28=N3; n19+n27=n21+n29=M3; | | |
| n20+n28=n22+n30=N3; n21+n29=n23+n31=M3; n22+n30=n24+n32=N3; (M3+N3=S) | | |
| | | |
| n49+n57=n51+n59=M7; n50+n58=n52+n60=N7; n51+n59=n53+n61=M7; | | |
| n52+n60=n54+n62=N7; n53+n61=n55+n63=M7; n54+n62+n56+n64=N7; (M7+N7=S) | | |

There are many types of complementary pairs such as $P_n+Q_n=S$; $M_n+N_n=S$.
But you cannot always expect $P_n=Q_n=C(=65)$, nor you can always say $M_n=N_n=C$.

What properties can you find among 4 tops of any blocks such as $\{n_1, n_3, n_{17}, n_{19}\}$, $\{n_1, n_4, n_{25}, n_{28}\}$, or $\{n_1, n_6, n_{41}, n_{46}\}$?

[Figure 1(again): Relationship among 4 tops of any Block]

What relationship is there among {n1, n3, n17, n19}?

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|----|----|----|----|----|----|----|----|-------|--|--|--|--|--|--|--|--|----|----|----|----|----|----|----|----|----|-------|--|--|--|--|--|--|--|--|----|----|----|----|----|----|----|----|----|-------|--|--|--|--|--|--|--|--|----|----|----|----|----|----|----|----|----|-------|--|--|--|--|--|--|--|--|----|----|----|----|----|----|----|----|----|-------|--|--|--|--|--|--|--|--|----|----|----|----|----|----|----|----|----|-------|--|--|--|--|--|--|--|--|----|----|----|----|----|----|----|----|----|-------|--|--|--|--|--|--|--|--|----|----|----|----|----|----|----|----|----|-------|--|--|--|--|--|--|--|--|----|----|----|----|----|----|----|----|----|---|
| <table style="border-collapse: collapse; width: 100%; text-align: center;"> <tr><td style="border-right: 1px dashed black;">n1</td><td style="border-right: 1px dashed black;">n2</td><td style="border-right: 1px dashed black;">n3</td><td style="border-right: 1px dashed black;">n4</td><td style="border-right: 1px dashed black;">n5</td><td style="border-right: 1px dashed black;">n6</td><td style="border-right: 1px dashed black;">n7</td><td style="border-right: 1px dashed black;">n8</td><td style="border-right: 1px dashed black;">n1</td></tr> <tr><td colspan="9">-----</td></tr> <tr><td style="border-right: 1px dashed black;">n9</td><td style="border-right: 1px dashed black;">10</td><td style="border-right: 1px dashed black;">11</td><td style="border-right: 1px dashed black;">12</td><td style="border-right: 1px dashed black;">13</td><td style="border-right: 1px dashed black;">14</td><td style="border-right: 1px dashed black;">15</td><td style="border-right: 1px dashed black;">16</td><td style="border-right: 1px dashed black;">n9</td></tr> <tr><td colspan="9">-----</td></tr> <tr><td style="border-right: 1px dashed black;">17</td><td style="border-right: 1px dashed black;">18</td><td style="border-right: 1px dashed black;">19</td><td style="border-right: 1px dashed black;">20</td><td style="border-right: 1px dashed black;">21</td><td style="border-right: 1px dashed black;">22</td><td style="border-right: 1px dashed black;">23</td><td style="border-right: 1px dashed black;">24</td><td style="border-right: 1px dashed black;">17</td></tr> <tr><td colspan="9">-----</td></tr> <tr><td style="border-right: 1px dashed black;">25</td><td style="border-right: 1px dashed black;">26</td><td style="border-right: 1px dashed black;">27</td><td style="border-right: 1px dashed black;">28</td><td style="border-right: 1px dashed black;">29</td><td style="border-right: 1px dashed black;">30</td><td style="border-right: 1px dashed black;">31</td><td style="border-right: 1px dashed black;">32</td><td style="border-right: 1px dashed black;">25</td></tr> <tr><td colspan="9">-----</td></tr> <tr><td style="border-right: 1px dashed black;">33</td><td style="border-right: 1px dashed black;">34</td><td style="border-right: 1px dashed black;">35</td><td style="border-right: 1px dashed black;">36</td><td style="border-right: 1px dashed black;">37</td><td style="border-right: 1px dashed black;">38</td><td style="border-right: 1px dashed black;">39</td><td style="border-right: 1px dashed black;">40</td><td style="border-right: 1px dashed black;">33</td></tr> <tr><td colspan="9">-----</td></tr> <tr><td style="border-right: 1px dashed black;">41</td><td style="border-right: 1px dashed black;">42</td><td style="border-right: 1px dashed black;">43</td><td style="border-right: 1px dashed black;">44</td><td style="border-right: 1px dashed black;">45</td><td style="border-right: 1px dashed black;">46</td><td style="border-right: 1px dashed black;">47</td><td style="border-right: 1px dashed black;">48</td><td style="border-right: 1px dashed black;">41</td></tr> <tr><td colspan="9">-----</td></tr> <tr><td style="border-right: 1px dashed black;">49</td><td style="border-right: 1px dashed black;">50</td><td style="border-right: 1px dashed black;">51</td><td style="border-right: 1px dashed black;">52</td><td style="border-right: 1px dashed black;">53</td><td style="border-right: 1px dashed black;">54</td><td style="border-right: 1px dashed black;">55</td><td style="border-right: 1px dashed black;">56</td><td style="border-right: 1px dashed black;">49</td></tr> <tr><td colspan="9">-----</td></tr> <tr><td style="border-right: 1px dashed black;">57</td><td style="border-right: 1px dashed black;">58</td><td style="border-right: 1px dashed black;">59</td><td style="border-right: 1px dashed black;">60</td><td style="border-right: 1px dashed black;">61</td><td style="border-right: 1px dashed black;">62</td><td style="border-right: 1px dashed black;">63</td><td style="border-right: 1px dashed black;">64</td><td style="border-right: 1px dashed black;">57</td></tr> <tr><td colspan="9">-----</td></tr> <tr><td style="border-right: 1px dashed black;">n1</td><td style="border-right: 1px dashed black;">n2</td><td style="border-right: 1px dashed black;">n3</td><td style="border-right: 1px dashed black;">n4</td><td style="border-right: 1px dashed black;">n5</td><td style="border-right: 1px dashed black;">n6</td><td style="border-right: 1px dashed black;">n7</td><td style="border-right: 1px dashed black;">n8</td><td style="border-right: 1px dashed black;">n1</td></tr> </table> | n1 | n2 | n3 | n4 | n5 | n6 | n7 | n8 | n1 | ----- | | | | | | | | | n9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | n9 | ----- | | | | | | | | | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 17 | ----- | | | | | | | | | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 25 | ----- | | | | | | | | | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 33 | ----- | | | | | | | | | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 41 | ----- | | | | | | | | | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 49 | ----- | | | | | | | | | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 57 | ----- | | | | | | | | | n1 | n2 | n3 | n4 | n5 | n6 | n7 | n8 | n1 | $n1+n2=n17+n18(=P1)$ $-)n2+n3=n18+n19(=P2)$ <hr style="border-top: 1px dashed black;"/> $n1-n3=n17-n19 \rightarrow n1+n19=n3+n17$ ('Cross-sums are equal.') |
| n1 | n2 | n3 | n4 | n5 | n6 | n7 | n8 | n1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| n9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | n9 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 17 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 25 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 33 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 41 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 49 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 57 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| n1 | n2 | n3 | n4 | n5 | n6 | n7 | n8 | n1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <p>How about {n1, n4, n25, n28}?</p> $n1+ n2+ n9+n10 = S$ $n17+n18+n25+n26 = S$ $n3+ n4+n11+n12 = S$ $+)n19+n20+n27+n28 = S$ <hr style="border-top: 1px dashed black;"/> $-) n9+n10+n17+n18 = S$ $-)n11+n12+n19+n20 = S$ <hr style="border-top: 1px dashed black;"/> $n1+n2+n3+n4+n25+n26+n27+n28=2*S$ $-) n2+ n3+n26+n27=P2+Q2 = S$ <hr style="border-top: 1px dashed black;"/> <p>Therefore $n1+ n4+n25+n28 = S$</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

{n1, n4, n25, n28} are the 4 tops of 4x4 block. Every 4 tops of any 4x4 block within add up to the same composite sum S(=130).

The 4 tops of 6x6 block {n1, n6, n41, n46} are just the same as the 4 tops of 4x4 block {n1, n6, n41, n46} in the Extended Space.

Therefore $n1+n6+n41+n46 = S$

{n1, n8, n57, n64} are not only the 4 tops of the largest 8x8 block, but also the 4 tops of the smallest 2x2 block on the right side bottom.

After all you can say every 4 tops of any block 2 x 2, 4 x 4, 6 x 6 or 8 x 8 add up to the composite sum 130.

How about {n1, n5, n33, n37}?

| | |
|--------------------------|---|
| $n1+ n2=n33+n34(=P1)$ | $n2+n20=n4+n18$ (Cross-sums of 3 x 3) |
| $+)n36+n37= n4+ n5(=P4)$ | $+)n18+n36=n20+n34$ (Cross-sums of 3 x 3) |

$n1+n2+n36+n37=n4+n5+n33+n34;$ $n2+n36=n4+n34;$ $\rightarrow n1+n37=n5+n33$

This proves cross-sums are equal to each other among 4 tops of any 5 x 5 block.

How about {n1, n7, n49, n55}?

They are not only the 4 tops of 7 x 7 block, but also the 4 tops of 3 x 3 block in the Extended Space on the right side bottom. Cross-sums are equal: $n1+n55=n7+n49;$

After all you can say every cross-sums are equal to each other among every 4 tops of any 3 x 3, 5 x 5 or 7 x 7 block.

#2. Pan-diagonals of Composite Square of Order 8

What properties do our Composite Conditions add to the pan-diagonals?

Let's have some algebraic study here. Let's not assume the constant sums of all pan-diagonals now at first and not yet define any equality of all sums.

[Figure 2: Basic Form & List of Pan-diagonal Equations]

| | |
|---------------------------------------|---------------------------------------|
| $n1+n10+n19+n28+n37+n46+n55+n64=pd1;$ | $n1+n16+n23+n30+n37+n44+n51+n58=pb1;$ |
| $n2+n11+n20+n29+n38+n47+n56+n57=pd2;$ | $n2+n9+n24+n31+n38+n45+n52+n59=pb2;$ |
| $n3+n12+n21+n30+n39+n48+n49+n58=pd3;$ | $n3+n10+n17+n32+n39+n46+n53+n60=pb3;$ |

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

$$\begin{aligned}
n4+n13+n22+n31+n40+n41+n50+n59 &= \text{pd4}; & n4+n11+n18+n25+n40+n47+n54+n61 &= \text{pb4}; \\
n5+n14+n23+n32+n33+n42+n51+n60 &= \text{pd5}; & n5+n12+n19+n26+n33+n48+n55+n62 &= \text{pb5}; \\
n6+n15+n24+n25+n34+n43+n52+n61 &= \text{pd6}; & n6+n13+n20+n27+n34+n41+n56+n63 &= \text{pb6}; \\
n7+n16+n17+n26+n35+n44+n53+n62 &= \text{pd7}; & n7+n14+n21+n28+n35+n42+n49+n64 &= \text{pb7}; \\
n8+n9+n18+n27+n36+n45+n54+n63 &= \text{pd8}; & n8+n15+n22+n29+n36+n43+n50+n57 &= \text{pb8};
\end{aligned}$$

Compare the **pd1** with **pb1**, you will find the common entries {n1, n37}.

$$\begin{aligned}
n1+n10+n19+n28+n37+n46+n55+n64 &= \text{pd1}; \\
n1+n16+n23+n30+n37+n44+n51+n58 &= \text{pb1}; \\
n10+n64 &= n16+n58 \quad (\text{Cross-sums of } 7 \times 7); \\
n19+n55 &= n23+n51 \quad (\text{Cross-sums of } 5 \times 5); \\
n28+n46 &= n30+n44 \quad (\text{Cross-sums of } 3 \times 3); \text{ Therefore } \text{pd1} = \text{pb1}
\end{aligned}$$

$$\begin{aligned}
n2+n11+n20+n29+n38+n47+n56+n57 &= \text{pd2}; \\
n2+n9+n24+n31+n38+n45+n52+n59 &= \text{pb2}; \quad \{n2+n38\} \text{ is common.} \\
n29+n47 &= n31+n45 \quad (\text{Cross-sums of } 3 \times 3); \\
n20+n56 &= n24+n52 \quad (\text{Cross-sums of } 5 \times 5); \\
n9+n59 &= n11+n57 \quad (\text{Cross-sums of } 7 \times 7); \text{ Therefore } \text{pd2} = \text{pb2} \\
\text{pd3} = \text{pb3}; \quad \text{pd4} = \text{pb4}; \quad \text{pd5} = \text{pb5}; \quad \text{pd6} = \text{pb6}; \quad \text{pd7} = \text{pb7}; \quad \text{pd8} = \text{pb8};
\end{aligned}$$

$$\begin{aligned}
n1+n10+n19+n28+n37+n46+n55+n64 &= \text{pd1}; \\
n3+n10+n17+n32+n39+n46+n53+n60 &= \text{pb3}; \quad \{n10+n46\} \text{ is common.} \\
n1+n19 &= n3+n17 \quad (\text{Cross-sums of } 3 \times 3); \\
n37+n55 &= n39+n53 \quad (\text{Cross-sums of } 3 \times 3); \\
n28+n64 &= n32+n60 \quad (\text{Cross-sums of } 5 \times 5); \text{ Therefore } \text{pd1} = \text{pb3}
\end{aligned}$$

$$\begin{aligned}
n1+n10+n19+n28+n37+n46+n55+n64 &= \text{pd1}; \\
n5+n12+n19+n26+n33+n48+n55+n62 &= \text{pb5}; \quad \{n19+n55\} \text{ is common.} \\
n10+n28 &= n12+n26 \quad (\text{Cross-sums of } 3 \times 3); \\
n1+n37 &= n5+n33 \quad (\text{Cross-sums of } 5 \times 5); \\
n46+n64 &= n48+n62 \quad (\text{Cross-sums of } 3 \times 3); \text{ Therefore } \text{pd1} = \text{pb5}
\end{aligned}$$

$$\begin{aligned}
n1+n10+n19+n28+n37+n46+n55+n64 &= \text{pd1}; \\
n7+n14+n21+n28+n35+n42+n49+n64 &= \text{pb7}; \quad \text{--> } \text{pd1} = \text{pb7}
\end{aligned}$$

After all $\text{pd1} = \text{pb1} = \text{pd3} = \text{pb3} = \text{pd5} = \text{pb5} = \text{pd7} = \text{pb7} = P$
In the same way $\text{pd2} = \text{pb2} = \text{pd4} = \text{pb4} = \text{pd6} = \text{pb6} = \text{pd8} = \text{pb8} = Q$
And $P + Q = 2 * K = 520$

All pan-diagonals are classified into 2 groups according to their line-sums:

- (1) The group of 8 pan-diagonals with n1, n3, n5 and n7, and
- (2) The one of 8 pan-diagonals with n2, n4, n6 and n8;

In each group they are all equal to one another. But you cannot always say all the sums are equal to 260, only under the Composite Conditions.

If you want all pan-diagonals to be equal to the magic constant, you must add at least one condition $pd1=260$ to the set of the first definitions before all by yourself.

#3. Making a 'Composite & Pan-diagonal' Magic Square of Order 8

Let's actually make some Composite & Pan-diagonal Magic Square of Order 8 here.

I made the 'Composite and Complete' type of magic squares before. They have all complementary pairs of 65 only on their pan-diagonals. But this time let's not add those strict conditions to our object.

We only presume (1) the Composite Conditions and (2) $RS1=LS1=pd1=K(=260)$.

$$n1+n2+n3+n4+n5+n6+n7+n8=K \quad \dots (RS1);$$

$$n1+n9+n17+n25+n33+n41+n49+n57=K \quad \dots (LS1);$$

$$n1+n10+n19+n28+n37+n46+n55+n64=K \quad \dots (pd1);$$

If you like, you may well add any other Pan-diagonal Condition such as

$$n1+n16+n23+n30+n37+n44+n51+n58=K \quad \dots (pb1);$$

$$n2+n11+n20+n29+n38+n47+n56+n57=K \quad \dots (pd2);$$

$$n2+n9+n24+n31+n38+n45+n52+n59=K \quad \dots (pb2);$$

...

You can never lose anything you want to have, even if you add these conditions.

** 'Composite' Type of Pan-diagonal Magic Squares of Order 8 **

** Compact List of the Standard Solutions (Only a Part) **

| 1# | 1729# | 2161# |
|-------------------------|-------------------------|-------------------------|
| 1 64 3 62 5 60 7 58 | 1 64 3 60 5 62 7 58 | 1 64 3 58 5 62 7 60 |
| 63 2 61 4 59 6 57 8 | 63 2 61 6 59 4 57 8 | 63 2 61 8 59 4 57 6 |
| 25 40 27 38 29 36 31 34 | 25 40 27 36 29 38 31 34 | 25 40 27 34 29 38 31 36 |
| 39 26 37 28 35 30 33 32 | 39 26 37 30 35 28 33 32 | 39 26 37 32 35 28 33 30 |
| 42 23 44 21 46 19 48 17 | 42 23 44 19 46 21 48 17 | 42 23 44 17 46 21 48 19 |
| 24 41 22 43 20 45 18 47 | 24 41 22 45 20 43 18 47 | 24 41 22 47 20 43 18 45 |
| 50 15 52 13 54 11 56 9 | 50 15 52 11 54 13 56 9 | 50 15 52 9 54 13 56 11 |
| 16 49 14 51 12 53 10 55 | 16 49 14 53 12 51 10 55 | 16 49 14 55 12 51 10 53 |
| 2593# | 3025# | 3457# |
| 1 64 3 56 9 62 11 54 | 1 64 3 54 9 62 11 56 | 1 64 3 48 17 62 19 46 |
| 63 2 61 10 55 4 53 12 | 63 2 61 12 55 4 53 10 | 63 2 61 18 47 4 45 20 |
| 21 44 23 36 29 42 31 34 | 21 44 23 34 29 42 31 36 | 13 52 15 36 29 50 31 34 |
| 43 22 41 30 35 24 33 32 | 43 22 41 32 35 24 33 30 | 51 14 49 30 35 16 33 32 |
| 38 27 40 19 46 25 48 17 | 38 27 40 17 46 25 48 19 | 38 27 40 11 54 25 56 9 |
| 28 37 26 45 20 39 18 47 | 28 37 26 47 20 39 18 45 | 28 37 26 53 12 39 10 55 |
| 50 15 52 7 58 13 60 5 | 50 15 52 5 58 13 60 7 | 42 23 44 7 58 21 60 5 |
| 16 49 14 57 8 51 6 59 | 16 49 14 59 8 51 6 57 | 24 41 22 57 8 43 6 59 |
| 3889# | 4321# | 4753# |
| 1 64 3 46 17 62 19 48 | 1 64 3 32 33 62 35 30 | 1 64 3 30 33 62 35 32 |
| 63 2 61 20 47 4 45 18 | 63 2 61 34 31 4 29 36 | 63 2 61 36 31 4 29 34 |
| 13 52 15 34 29 50 31 36 | 13 52 15 20 45 50 47 18 | 13 52 15 18 45 50 47 20 |
| 51 14 49 32 35 16 33 30 | 51 14 49 46 19 16 17 48 | 51 14 49 48 19 16 17 46 |
| 38 27 40 9 54 25 56 11 | 22 43 24 11 54 41 56 9 | 22 43 24 9 54 41 56 11 |
| 28 37 26 55 12 39 10 53 | 44 21 42 53 12 23 10 55 | 44 21 42 55 12 23 10 53 |
| 42 23 44 5 58 21 60 7 | 26 39 28 7 58 37 60 5 | 26 39 28 5 58 37 60 7 |
| 24 41 22 59 8 43 6 57 | 40 25 38 57 8 27 6 59 | 40 25 38 59 8 27 6 57 |
| 5185# | 5617# | 7345# |
| 1 64 5 62 3 60 7 58 | 1 64 5 60 3 62 7 58 | 1 64 5 58 3 62 7 60 |
| 63 2 59 4 61 6 57 8 | 63 2 59 6 61 4 57 8 | 63 2 59 8 61 4 57 6 |
| 25 40 29 38 27 36 31 34 | 25 40 29 36 27 38 31 34 | 25 40 29 34 27 38 31 36 |
| 39 26 35 28 37 30 33 32 | 39 26 35 30 37 28 33 32 | 39 26 35 32 37 28 33 30 |
| 42 23 46 21 44 19 48 17 | 42 23 46 19 44 21 48 17 | 42 23 46 17 44 21 48 19 |
| 24 41 20 43 22 45 18 47 | 24 41 20 45 22 43 18 47 | 24 41 20 47 22 43 18 45 |
| 50 15 54 13 52 11 56 9 | 50 15 54 11 52 13 56 9 | 50 15 54 9 52 13 56 11 |
| 16 49 12 51 14 53 10 55 | 16 49 12 53 14 51 10 55 | 16 49 12 55 14 51 10 53 |

| | | |
|---|---|---|
| <p>7777#</p> <p>1 64 5 56 9 60 13 52 63 2 59 10 55 6 51 14 19 46 23 38 27 42 31 34 45 20 41 28 37 24 33 32 36 29 40 21 44 25 48 17 30 35 26 43 22 39 18 47 50 15 54 7 58 11 62 3 16 49 12 57 8 53 4 61</p> <p>9073#</p> <p>1 64 5 44 17 60 21 48 63 2 59 22 47 6 43 18 11 54 15 34 27 50 31 38 53 12 49 32 37 16 33 28 36 29 40 9 52 25 56 13 30 35 26 55 14 39 10 51 42 23 46 3 58 19 62 7 24 41 20 61 8 45 4 57</p> <p>10369#</p> <p>1 64 7 62 3 60 5 58 63 2 57 4 61 6 59 8 25 40 31 38 27 36 29 34 39 26 33 28 37 30 35 32 42 23 48 21 44 19 46 17 24 41 18 43 22 45 20 47 50 15 56 13 52 11 54 9 16 49 10 51 14 53 12 55</p> <p>12961#</p> <p>1 64 7 54 11 58 13 52 63 2 57 12 53 8 51 14 17 48 23 38 27 42 29 36 47 18 41 28 37 24 35 30 34 31 40 21 44 25 46 19 32 33 26 43 22 39 20 45 50 15 56 5 60 9 62 3 16 49 10 59 6 55 4 61</p> <p>14257#</p> <p>1 64 7 44 19 58 21 46 63 2 57 22 45 8 43 20 9 56 15 36 27 50 29 38 55 10 49 30 37 16 35 28 34 31 40 11 52 25 54 13 32 33 26 53 14 39 12 51 42 23 48 3 60 17 62 5 24 41 18 61 6 47 4 59</p> <p>15553#</p> <p>1 64 9 62 3 56 11 54 63 2 55 4 61 10 53 12 21 44 29 42 23 36 31 34 43 22 35 24 41 30 33 32 38 27 46 25 40 19 48 17 28 37 20 39 26 45 18 47 50 15 58 13 52 7 60 5 16 49 8 51 14 57 6 59</p> <p>31105#</p> <p>1 64 17 62 3 48 19 46 63 2 47 4 61 18 45 20 13 52 29 50 15 36 31 34 51 14 35 16 49 30 33 32 38 27 54 25 40 11 56 9 28 37 12 39 26 53 10 55 42 23 58 21 44 7 60 5 24 41 8 43 22 57 6 59</p> | <p>8209#</p> <p>1 64 5 52 9 60 13 56 63 2 59 14 55 6 51 10 19 46 23 34 27 42 31 38 45 20 41 32 37 24 33 28 36 29 40 17 44 25 48 21 30 35 26 47 22 39 18 43 50 15 54 3 58 11 62 7 16 49 12 61 8 53 4 57</p> <p>9505#</p> <p>1 64 5 32 33 60 37 28 63 2 59 34 31 6 27 38 11 54 15 22 43 50 47 18 53 12 49 44 21 16 17 48 20 45 24 13 52 41 56 9 46 19 42 51 14 23 10 55 26 39 30 7 58 35 62 3 40 25 36 57 8 29 4 61</p> <p>10801#</p> <p>1 64 7 60 3 62 5 58 63 2 57 6 61 4 59 8 25 40 31 36 27 38 29 34 39 26 33 30 37 28 35 32 42 23 48 19 44 21 46 17 24 41 18 45 22 43 20 47 50 15 56 11 52 13 54 9 16 49 10 53 14 51 12 55</p> <p>13393#</p> <p>1 64 7 52 11 58 13 54 63 2 57 14 53 8 51 12 17 48 23 36 27 42 29 38 47 18 41 30 37 24 35 28 34 31 40 19 44 25 46 21 32 33 26 45 22 39 20 43 50 15 56 3 60 9 62 5 16 49 10 61 6 55 4 59</p> <p>14689#</p> <p>1 64 7 30 35 58 37 28 63 2 57 36 29 8 27 38 9 56 15 22 43 50 45 20 55 10 49 44 21 16 19 46 18 47 24 13 52 41 54 11 48 17 42 51 14 23 12 53 26 39 32 5 60 33 62 3 40 25 34 59 6 31 4 61</p> <p>20737#</p> <p>1 64 11 62 3 56 9 54 63 2 53 4 61 10 55 12 21 44 31 42 23 36 29 34 43 22 33 24 41 30 35 32 38 27 48 25 40 19 46 17 28 37 18 39 26 45 20 47 50 15 60 13 52 7 58 5 16 49 6 51 14 57 8 59</p> <p>36289#</p> <p>1 64 19 62 3 48 17 46 63 2 45 4 61 18 47 20 13 52 31 50 15 36 29 34 51 14 33 16 49 30 35 32 38 27 56 25 40 11 54 9 28 37 10 39 26 53 12 55 42 23 60 21 44 7 58 5 24 41 6 43 22 57 8 59</p> | <p>8641#</p> <p>1 64 5 48 17 60 21 44 63 2 59 18 47 6 43 22 11 54 15 38 27 50 31 34 53 12 49 28 37 16 33 32 36 29 40 13 52 25 56 9 30 35 26 51 14 39 10 55 42 23 46 7 58 19 62 3 24 41 20 57 8 45 4 61</p> <p>9937#</p> <p>1 64 5 28 33 60 37 32 63 2 59 38 31 6 27 34 11 54 15 18 43 50 47 22 53 12 49 48 21 16 17 44 20 45 24 9 52 41 56 13 46 19 42 55 14 23 10 51 26 39 30 3 58 35 62 7 40 25 36 61 8 29 4 57</p> <p>11233#</p> <p>1 64 7 58 3 62 5 60 63 2 57 8 61 4 59 6 25 40 31 34 27 38 29 36 39 26 33 32 37 28 35 30 42 23 48 17 44 21 46 19 24 41 18 47 22 43 20 45 50 15 56 9 52 13 54 11 16 49 10 55 14 51 12 53</p> <p>13825#</p> <p>1 64 7 46 19 58 21 44 63 2 57 20 45 8 43 22 9 56 15 38 27 50 29 36 55 10 49 28 37 16 35 30 34 31 40 13 52 25 54 11 32 33 26 51 14 39 12 53 42 23 48 5 60 17 62 3 24 41 18 59 6 47 4 61</p> <p>15121#</p> <p>1 64 7 28 35 58 37 30 63 2 57 38 29 8 27 36 9 56 15 20 43 50 45 22 55 10 49 46 21 16 19 44 18 47 24 11 52 41 54 13 48 17 42 53 14 23 12 51 26 39 32 3 60 33 62 5 40 25 34 61 6 31 4 59</p> <p>25921#</p> <p>1 64 13 60 5 56 9 52 63 2 51 6 59 10 55 14 19 46 31 42 23 38 27 34 45 20 33 24 41 28 37 32 36 29 48 25 40 21 44 17 30 35 18 39 26 43 22 47 50 15 62 11 54 7 58 3 16 49 4 53 12 57 8 61</p> <p>41473#</p> <p>1 64 21 60 5 48 17 44 63 2 43 6 59 18 47 22 11 54 31 50 15 38 27 34 53 12 33 16 49 28 37 32 36 29 56 25 40 13 52 9 30 35 10 39 26 51 14 55 42 23 62 19 46 7 58 3 24 41 4 45 20 57 8 61</p> |
|---|---|---|

| | | |
|--|--|---|
| <p>46657#</p> <p>1 64 25 56 9 48 17 40 63 2 39 10 55 18 47 26 7 58 31 50 15 42 23 34 57 8 33 16 49 24 41 32 36 29 60 21 44 13 52 5 30 35 6 43 22 51 14 59 38 27 62 19 46 11 54 3 28 37 4 45 20 53 12 61</p> <p>62209#</p> <p>1 64 37 60 5 32 33 28 63 2 27 6 59 34 31 38 11 54 47 50 15 22 43 18 53 12 17 16 49 44 21 48 20 45 56 41 24 13 52 9 46 19 10 23 42 51 14 55 26 39 62 35 30 7 58 3 40 25 4 29 36 57 8 61</p> <p>77761#</p> <p>1 64 2 63 5 60 6 59 62 3 61 4 58 7 57 8 25 40 26 39 29 36 30 35 38 27 37 28 34 31 33 32 43 22 44 21 47 18 48 17 24 41 23 42 20 45 19 46 51 14 52 13 55 10 56 9 16 49 15 50 12 53 11 54</p> <p>93313#</p> <p>1 64 9 63 2 56 10 55 62 3 54 4 61 11 53 12 21 44 29 43 22 36 30 35 42 23 34 24 41 31 33 32 39 26 47 25 40 18 48 17 28 37 20 38 27 45 19 46 51 14 59 13 52 6 60 5 16 49 8 50 15 57 7 58</p> <p>108865#</p> <p>1 64 17 63 2 48 18 47 62 3 46 4 61 19 45 20 13 52 29 51 14 36 30 35 50 15 34 16 49 31 33 32 39 26 55 25 40 10 56 9 28 37 12 38 27 53 11 54 43 22 59 21 44 6 60 5 24 41 8 42 23 57 7 58</p> <p>124417#</p> <p>1 64 25 56 9 48 17 40 62 3 38 11 54 19 46 27 6 59 30 51 14 43 22 35 57 8 33 16 49 24 41 32 36 29 60 21 44 13 52 5 31 34 7 42 23 50 15 58 39 26 63 18 47 10 55 2 28 37 4 45 20 53 12 61</p> <p>139969#</p> <p>1 64 37 60 5 32 33 28 62 3 26 7 58 35 30 39 10 55 46 51 14 23 42 19 53 12 17 16 49 44 21 48 20 45 56 41 24 13 52 9 47 18 11 22 43 50 15 54 27 38 63 34 31 6 59 2 40 25 4 29 36 57 8 61</p> | <p>51841#</p> <p>1 64 33 62 3 32 35 30 63 2 31 4 61 34 29 36 13 52 45 50 15 20 47 18 51 14 19 16 49 46 17 48 22 43 54 41 24 11 56 9 44 21 12 23 42 53 10 55 26 39 58 37 28 7 60 5 40 25 8 27 38 57 6 59</p> <p>67393#</p> <p>1 64 41 56 9 32 33 24 63 2 23 10 55 34 31 42 7 58 47 50 15 26 39 18 57 8 17 16 49 40 25 48 20 45 60 37 28 13 52 5 46 19 6 27 38 51 14 59 22 43 62 35 30 11 54 3 44 21 4 29 36 53 12 61</p> <p>82945#</p> <p>1 64 5 63 2 60 6 59 62 3 58 4 61 7 57 8 25 40 29 39 26 36 30 35 38 27 34 28 37 31 33 32 43 22 47 21 44 18 48 17 24 41 20 42 23 45 19 46 51 14 55 13 52 10 56 9 16 49 12 50 15 53 11 54</p> <p>98497#</p> <p>1 64 10 63 2 56 9 55 62 3 53 4 61 11 54 12 21 44 30 43 22 36 29 35 42 23 33 24 41 31 34 32 39 26 48 25 40 18 47 17 28 37 19 38 27 45 20 46 51 14 60 13 52 6 59 5 16 49 7 50 15 57 8 58</p> <p>114049#</p> <p>1 64 18 63 2 48 17 47 62 3 45 4 61 19 46 20 13 52 30 51 14 36 29 35 50 15 33 16 49 31 34 32 39 26 56 25 40 10 55 9 28 37 11 38 27 53 12 54 43 22 60 21 44 6 59 5 24 41 7 42 23 57 8 58</p> <p>129601#</p> <p>1 64 33 63 2 32 34 31 62 3 30 4 61 35 29 36 13 52 45 51 14 20 46 19 50 15 18 16 49 47 17 48 23 42 55 41 24 10 56 9 44 21 12 22 43 53 11 54 27 38 59 37 28 6 60 5 40 25 8 26 39 57 7 58</p> <p>145153#</p> <p>1 64 41 56 9 32 33 24 62 3 22 11 54 35 30 43 6 59 46 51 14 27 38 19 57 8 17 16 49 40 25 48 20 45 60 37 28 13 52 5 47 18 7 26 39 50 15 58 23 42 63 34 31 10 55 2 44 21 4 29 36 53 12 61</p> | <p>57025#</p> <p>1 64 35 62 3 32 33 30 63 2 29 4 61 34 31 36 13 52 47 50 15 20 45 18 51 14 17 16 49 46 19 48 22 43 56 41 24 11 54 9 44 21 10 23 42 53 12 55 26 39 60 37 28 7 58 5 40 25 6 27 38 57 8 59</p> <p>72577#</p> <p>1 64 49 48 17 32 33 16 63 2 15 18 47 34 31 50 7 58 55 42 23 26 39 10 57 8 9 24 41 40 25 56 12 53 60 37 28 21 44 5 54 11 6 27 38 43 22 59 14 51 62 35 30 19 46 3 52 13 4 29 36 45 20 61</p> <p>88129#</p> <p>1 64 6 63 2 60 5 59 62 3 57 4 61 7 58 8 25 40 30 39 26 36 29 35 38 27 33 28 37 31 34 32 43 22 48 21 44 18 47 17 24 41 19 42 23 45 20 46 51 14 56 13 52 10 55 9 16 49 11 50 15 53 12 54</p> <p>103681#</p> <p>1 64 13 60 5 56 9 52 62 3 50 7 58 11 54 15 18 47 30 43 22 39 26 35 45 20 33 24 41 28 37 32 36 29 48 25 40 21 44 17 31 34 19 38 27 42 23 46 51 14 63 10 55 6 59 2 16 49 4 53 12 57 8 61</p> <p>119233#</p> <p>1 64 21 60 5 48 17 44 62 3 42 7 58 19 46 23 10 55 30 51 14 39 26 35 53 12 33 16 49 28 37 32 36 29 56 25 40 13 52 9 31 34 11 38 27 50 15 54 43 22 63 18 47 6 59 2 24 41 4 45 20 57 8 61</p> <p>134785#</p> <p>1 64 34 63 2 32 33 31 62 3 29 4 61 35 30 36 13 52 46 51 14 20 45 19 50 15 17 16 49 47 18 48 23 42 56 41 24 10 55 9 44 21 11 22 43 53 12 54 27 38 60 37 28 6 59 5 40 25 7 26 39 57 8 58</p> <p>150337#</p> <p>1 64 49 48 17 32 33 16 62 3 14 19 46 35 30 51 6 59 54 43 22 27 38 11 57 8 9 24 41 40 25 56 12 53 60 37 28 21 44 5 55 10 7 26 39 42 23 58 15 50 63 34 31 18 47 2 52 13 4 29 36 45 20 61</p> |
|--|--|---|

| | | |
|--|---|---|
| <p style="text-align: center; color: blue;">155521#</p> <p>1 64 5 60 9 56 13 52 61 4 57 8 53 12 49 16 18 47 22 43 26 39 30 35 46 19 42 23 38 27 34 31 36 29 40 25 44 21 48 17 32 33 28 37 24 41 20 45 51 14 55 10 59 6 63 2 15 50 11 54 7 58 3 62</p> <p style="text-align: center; color: blue;">196993#</p> <p>1 64 2 63 3 62 4 61 60 5 59 6 58 7 57 8 25 40 26 39 27 38 28 37 36 29 35 30 34 31 33 32 45 20 46 19 47 18 48 17 24 41 23 42 22 43 21 44 53 12 54 11 55 10 56 9 16 49 15 50 14 51 13 52</p> <p style="text-align: center; color: blue;">357697#</p> <p>1 64 9 56 17 48 25 40 57 8 49 16 41 24 33 32 4 61 12 53 20 45 28 37 60 5 52 13 44 21 36 29 38 27 46 19 54 11 62 3 31 34 23 42 15 50 7 58 39 26 47 18 55 10 63 2 30 35 22 43 14 51 6 59</p> <p style="text-align: center; color: blue;">489889#</p> <p>1 64 2 63 5 60 6 59 54 11 53 12 50 15 49 16 19 46 20 45 23 42 24 41 40 25 39 26 36 29 35 30 43 22 44 21 47 18 48 17 32 33 31 34 28 37 27 38 57 8 58 7 61 4 62 3 14 51 13 52 10 55 9 56</p> <p style="text-align: center; color: blue;">585793#</p> <p>1 64 3 62 17 48 19 46 51 14 49 16 35 30 33 32 6 59 8 57 22 43 24 41 56 9 54 11 40 25 38 27 42 23 44 21 58 7 60 5 31 34 29 36 15 50 13 52 45 20 47 18 61 4 63 2 28 37 26 39 12 53 10 55</p> <p style="text-align: center; color: blue;">689473#</p> <p>1 64 3 62 5 60 7 58 47 18 45 20 43 22 41 24 10 55 12 53 14 51 16 49 40 25 38 27 36 29 34 31 50 15 52 13 54 11 56 9 32 33 30 35 28 37 26 39 57 8 59 6 61 4 63 2 23 42 21 44 19 46 17 48</p> <p style="text-align: center;">.</p> | <p style="text-align: center; color: blue;">160705#</p> <p>1 64 9 60 5 56 13 52 61 4 53 8 57 12 49 16 18 47 26 43 22 39 30 35 46 19 38 23 42 27 34 31 36 29 44 25 40 21 48 17 32 33 24 37 28 41 20 45 51 14 59 10 55 6 63 2 15 50 7 54 11 58 3 62</p> <p style="text-align: center; color: blue;">274753#</p> <p>1 64 3 62 9 56 11 54 59 6 57 8 51 14 49 16 18 47 20 45 26 39 28 37 44 21 42 23 36 29 34 31 38 27 40 25 46 19 48 17 32 33 30 35 24 41 22 43 53 12 55 10 61 4 63 2 15 50 13 52 7 58 5 60</p> <p style="text-align: center; color: blue;">370657#</p> <p>1 64 2 63 3 62 4 61 56 9 55 10 54 11 53 12 21 44 22 43 23 42 24 41 36 29 35 30 34 31 33 32 45 20 46 19 47 18 48 17 28 37 27 38 26 39 25 40 57 8 58 7 59 6 60 5 16 49 15 50 14 51 13 52</p> <p style="text-align: center; color: blue;">531361#</p> <p>1 64 5 60 17 48 21 44 53 12 49 16 37 28 33 32 4 61 8 57 20 45 24 41 56 9 52 13 40 25 36 29 42 23 46 19 58 7 62 3 31 34 27 38 15 50 11 54 43 22 47 18 59 6 63 2 30 35 26 39 14 51 10 55</p> <p style="text-align: center; color: blue;">598753#</p> <p>1 64 2 63 17 48 18 47 50 15 49 16 34 31 33 32 7 58 8 57 23 42 24 41 56 9 55 10 40 25 39 26 43 22 44 21 59 6 60 5 30 35 29 36 14 51 13 52 45 20 46 19 61 4 62 3 28 37 27 38 12 53 11 54</p> <p style="text-align: center; color: blue;">730945#</p> <p>1 64 2 63 5 60 6 59 46 19 45 20 42 23 41 24 11 54 12 53 15 50 16 49 40 25 39 26 36 29 35 30 51 14 52 13 55 10 56 9 32 33 31 34 28 37 27 38 57 8 58 7 61 4 62 3 22 43 21 44 18 47 17 48</p> | <p style="text-align: center; color: blue;">165889#</p> <p>1 64 13 60 5 56 9 52 61 4 49 8 57 12 53 16 18 47 30 43 22 39 26 35 46 19 34 23 42 27 38 31 36 29 48 25 40 21 44 17 32 33 20 37 28 41 24 45 51 14 63 10 55 6 59 2 15 50 3 54 11 58 7 62</p> <p style="text-align: center; color: blue;">316225#</p> <p>1 64 2 63 9 56 10 55 58 7 57 8 50 15 49 16 19 46 20 45 27 38 28 37 44 21 43 22 36 29 35 30 39 26 40 25 47 18 48 17 32 33 31 34 24 41 23 42 53 12 54 11 61 4 62 3 14 51 13 52 6 59 5 60</p> <p style="text-align: center; color: blue;">448417#</p> <p>1 64 3 62 5 60 7 58 55 10 53 12 51 14 49 16 18 47 20 45 22 43 24 41 40 25 38 27 36 29 34 31 42 23 44 21 46 19 48 17 32 33 30 35 28 37 26 39 57 8 59 6 61 4 63 2 15 50 13 52 11 54 9 56</p> <p style="text-align: center; color: blue;">544321#</p> <p>1 64 2 63 3 62 4 61 52 13 51 14 50 15 49 16 21 44 22 43 23 42 24 41 40 25 39 26 38 27 37 28 45 20 46 19 47 18 48 17 32 33 31 34 30 35 29 36 57 8 58 7 59 6 60 5 12 53 11 54 10 55 9 56</p> <p style="text-align: center; color: blue;">611713#</p> <p>1 64 2 63 3 62 4 61 48 17 47 18 46 19 45 20 13 52 14 51 15 50 16 49 36 29 35 30 34 31 33 32 53 12 54 11 55 10 56 9 28 37 27 38 26 39 25 40 57 8 58 7 59 6 60 5 24 41 23 42 22 43 21 44</p> <p style="text-align: center; color: blue;">772417#</p> <p>1 64 5 60 9 56 13 52 45 20 41 24 37 28 33 32 4 61 8 57 12 53 16 49 48 17 44 21 40 25 36 29 50 15 54 11 58 7 62 3 31 34 27 38 23 42 19 46 51 14 55 10 59 6 63 2 30 35 26 39 22 43 18 47</p> |
|--|---|---|

The long list above has a part of my computation result for the standard solutions of this type. The solutions are too many for me to count them up through. It took too long a time to do that job. I could not help giving it up on the way.

We must try to find anything different, such as the smaller set of solutions.

#4. Find the 'Fundamental Solutions' of the same type.

But it should be the next problem how we could give inequality conditions to the critical positions. If you give the more strict conditions, the less solutions you will get. It might be a relief, but you always have to do that as reasonably as possible.

I studied elaborately about the possibility of line-exchange in a square.

Which rows and columns can we exchange with each other?

I have found many ways to do that. But I skip to explain about the whole process here. I only mention about a few steps and the result directly.

[Figure 3: Basic Form and Possibility of Line Exchanges]

| | | | | | | | | OK (4-6) Line Exchange | | | | | | | |
|----|----|----|----|----|----|----|----|------------------------|----|----|----|----|----|----|-------|
| n1 | n2 | n3 | n4 | n5 | n6 | n7 | n8 | n1 | n2 | n3 | n4 | n5 | n6 | n7 | n8 |
| n9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | n9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 <- |
| 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 <- |
| 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 |
| 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 |

$n1+n10+n19+n28+n37+n46+n55+n64=K$; $n1+n10+n19+n44+n37+n30+n55+n64=NewPD1$
 $n28+n46=n30+n44$ (Cross-sums of 3 x 3 block within) Therefore $NwePD1=K$;
 $n8+n15+n22+n29+n36+n43+n50+n57=K$; $n8+n15+n22+n45+n36+n27+n50+n57=NewPD2$
 $n29+n43=n27+n45$ (Cross-sums of 3 x 3 block within) Therefore $NwePD2=K$;

 $n17+n18+n41+n42=n17+n18+n25+n26=S$ (Because $n25+n26=n41+n42=Q1$)
 $n31+n32+n55+n56=n31+n32+n39+n40=S$ (Because $n39+n40=n55+n56=Q7$)

| | | | | | | | | OK (3-7) Line Exchange | | | | | | | |
|----|----|----|----|----|----|----|----|------------------------|----|----|----|----|----|----|-------|
| n1 | n2 | n3 | n4 | n5 | n6 | n7 | n8 | n1 | n2 | n3 | n4 | n5 | n6 | n7 | n8 |
| n9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | n9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 <- |
| 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
| 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 <- |
| 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 |

$n1+n10+n19+n28+n37+n46+n55+n64=K$; $n1+n10+n51+n28+n37+n46+n23+n64=NewPD1$
 $n19+n55=n23+n55$ (Cross-sums of 5 x 5 block within) Therefore $NwePD3=K$;
 $n8+n15+n22+n29+n36+n43+n50+n57=K$; $n8+n15+n54+n29+n36+n43+n18+n57=NewPD4$
 $n18+n54=n22+n50$ (Cross-sums of 5 x 5 block within) Therefore $NwePD4=K$;

 $n9+n10+n49+n50=n9+n10+n17+n18=S$ (Because $n49+n50=n17+n18=P1$)
 $n23+n24+n63+n64=n55+n56+n63+n64=S$ (Because $n55+n56=n23+n27=P7$)

I decided to make the 'fundamental form' with $n1=1$, and rearrange all the other entries except $n1$, giving such inequality conditions as: $n1 < n3 < n7 < n5$, $n1 < n17 < n49 < n33$, $n2 > n4 > n8 > n6$, $n9 > n25 > n57 > n41$ and $n2 > n9$.

These conditions are so strict that they should make you have the smallest set of Fundamental Solutions and take the shortest time to calculate all.

I am glad to report I have finally found the 360 'Fundamental Solutions' of that type.

Of course, each row, each column and each pan-diagonal of our result adds up to the magic constant 260.

The next list below shows a part of the result of my recent computation.

** 'Composite' Type of Pan-diagonal Magic Squares of Order 8 **
 ** Compact List of the 360 Fundamental Solutions (when $n1=1$) **

| | | |
|---|--|---|
| <p>1#</p> <p>1 64 3 62 7 58 5 60 63 2 61 4 57 8 59 6 25 40 27 38 31 34 29 36 39 26 37 28 33 32 35 30 50 15 52 13 56 9 54 11 16 49 14 51 10 55 12 53 42 23 44 21 48 17 46 19 24 41 22 43 18 47 20 45</p> <p>137#</p> <p>1 64 2 63 4 61 3 62 60 5 59 6 57 8 58 7 25 40 26 39 28 37 27 38 36 29 35 30 33 32 34 31 53 12 54 11 56 9 55 10 16 49 15 50 13 52 14 51 45 20 46 19 48 17 47 18 24 41 23 42 21 44 22 43</p> <p>203#</p> <p>1 64 2 63 4 61 3 62 56 9 55 10 53 12 54 11 21 44 22 43 24 41 23 42 36 29 35 30 33 32 34 31 57 8 58 7 60 5 59 6 16 49 15 50 13 52 14 51 45 20 46 19 48 17 47 18 28 37 27 38 25 40 26 39</p> <p>241#</p> <p>1 64 2 63 4 61 3 62 52 13 51 14 49 16 50 15 21 44 22 43 24 41 23 42 40 25 39 26 37 28 38 27 57 8 58 7 60 5 59 6 12 53 11 54 9 56 10 55 45 20 46 19 48 17 47 18 32 33 31 34 29 36 30 35</p> <p>258#</p> <p>1 64 2 63 6 59 5 60 46 19 45 20 41 24 42 23 11 54 12 53 16 49 15 50 40 25 39 26 35 30 36 29 57 8 58 7 62 3 61 4 22 43 21 44 17 48 18 47 51 14 52 13 56 9 55 10 32 33 31 34 27 38 28 37</p> <p>291#</p> <p>1 63 3 61 12 54 10 56 60 6 58 8 49 15 51 13 17 47 19 45 28 38 26 40 44 22 42 24 33 31 35 29 53 11 55 9 64 2 62 4 16 50 14 52 5 59 7 57 37 27 39 25 48 18 46 20 32 34 30 36 21 43 23 41</p> <p>331#</p> <p>1 62 2 61 12 55 11 56 60 7 59 8 49 14 50 13 17 46 18 45 28 39 27 40 44 23 43 24 33 30 34 29 53 10 54 9 64 3 63 4 16 51 15 52 5 58 6 57 37 26 38 25 48 19 47 20 32 35 31 36 21 42 22 41</p> | <p>61#</p> <p>1 64 2 63 6 59 5 60 62 3 61 4 57 8 58 7 25 40 26 39 30 35 29 36 38 27 37 28 33 32 34 31 51 14 52 13 56 9 55 10 16 49 15 50 11 54 12 53 43 22 44 21 48 17 47 18 24 41 23 42 19 46 20 45</p> <p>176#</p> <p>1 64 3 62 11 54 9 56 59 6 57 8 49 16 51 14 18 47 20 45 28 37 26 39 44 21 42 23 34 31 36 29 53 12 55 10 63 2 61 4 15 50 13 52 5 60 7 58 38 27 40 25 48 17 46 19 32 33 30 35 22 43 24 41</p> <p>227#</p> <p>1 64 3 62 7 58 5 60 55 10 53 12 49 16 51 14 18 47 20 45 24 41 22 43 40 25 38 27 34 31 36 29 57 8 59 6 63 2 61 4 15 50 13 52 9 56 11 54 42 23 44 21 48 17 46 19 32 33 30 35 26 39 28 37</p> <p>245#</p> <p>1 64 2 63 4 61 3 62 48 17 47 18 45 20 46 19 13 52 14 51 16 49 15 50 36 29 35 30 33 32 34 31 57 8 58 7 60 5 59 6 24 41 23 42 21 44 22 43 53 12 54 11 56 9 55 10 28 37 27 38 25 40 26 39</p> <p>260#</p> <p>1 64 2 63 4 61 3 62 44 21 43 22 41 24 42 23 13 52 14 51 16 49 15 50 40 25 39 26 37 28 38 27 57 8 58 7 60 5 59 6 20 45 19 46 17 48 18 47 53 12 54 11 56 9 55 10 32 33 31 34 29 36 30 35</p> <p>312#</p> <p>1 63 3 61 8 58 6 60 56 10 54 12 49 15 51 13 17 47 19 45 24 42 22 44 40 26 38 28 33 31 35 29 57 7 59 5 64 2 62 4 16 50 14 52 9 55 11 53 41 23 43 21 48 18 46 20 32 34 30 36 25 39 27 37</p> <p>343#</p> <p>1 62 2 61 8 59 7 60 56 11 55 12 49 14 50 13 17 46 18 45 24 43 23 44 40 27 39 28 33 30 34 29 57 6 58 5 64 3 63 4 16 51 15 52 9 54 10 53 41 22 42 21 48 19 47 20 32 35 31 36 25 38 26 37</p> | <p>113#</p> <p>1 64 5 60 13 52 9 56 61 4 57 8 49 16 53 12 18 47 22 43 30 35 26 39 46 19 42 23 34 31 38 27 51 14 55 10 63 2 59 6 15 50 11 54 3 62 7 58 36 29 40 25 48 17 44 21 32 33 28 37 20 45 24 41</p> <p>191#</p> <p>1 64 2 63 10 55 9 56 58 7 57 8 49 16 50 15 19 46 20 45 28 37 27 38 44 21 43 22 35 30 36 29 53 12 54 11 62 3 61 4 14 51 13 52 5 60 6 59 39 26 40 25 48 17 47 18 32 33 31 34 23 42 24 41</p> <p>235#</p> <p>1 64 2 63 6 59 5 60 54 11 53 12 49 16 50 15 19 46 20 45 24 41 23 42 40 25 39 26 35 30 36 29 57 8 58 7 62 3 61 4 14 51 13 52 9 56 10 55 43 22 44 21 48 17 47 18 32 33 31 34 27 38 28 37</p> <p>255#</p> <p>1 64 3 62 7 58 5 60 47 18 45 20 41 24 43 22 10 55 12 53 16 49 14 51 40 25 38 27 34 31 36 29 57 8 59 6 63 2 61 4 23 42 21 44 17 48 19 46 50 15 52 13 56 9 54 11 32 33 30 35 26 39 28 37</p> <p>261#</p> <p>1 63 5 59 14 52 10 56 62 4 58 8 49 15 53 11 17 47 21 43 30 36 26 40 46 20 42 24 33 31 37 27 51 13 55 9 64 2 60 6 16 50 12 54 3 61 7 57 35 29 39 25 48 18 44 22 32 34 28 38 19 45 23 41</p> <p>325#</p> <p>1 63 3 61 8 58 6 60 48 18 46 20 41 23 43 21 9 55 11 53 16 50 14 52 40 26 38 28 33 31 35 29 57 7 59 5 64 2 62 4 24 42 22 44 17 47 19 45 49 15 51 13 56 10 54 12 32 34 30 36 25 39 27 37</p> <p>351#</p> <p>1 62 2 61 8 59 7 60 48 19 47 20 41 22 42 21 9 54 10 53 16 51 15 52 40 27 39 28 33 30 34 29 57 6 58 5 64 3 63 4 24 43 23 44 17 46 18 45 49 14 50 13 56 11 55 12 32 35 31 36 25 38 26 37</p> |
|---|--|---|

```

355#
1 60 2 59 22 47 21 48
56 13 55 14 35 26 36 25
3 58 4 57 24 45 23 46
54 15 53 16 33 28 34 27
43 18 44 17 64 5 63 6
30 39 29 40 9 52 10 51
41 20 42 19 62 7 61 8
32 37 31 38 11 50 12 49
[Count = 360] OK!

```

```

** Counts according to the Value of n37 **
[52] 2, [54] 2, [55] 5, [56] 15, [58] 6, [59] 15, [60] 39, [61] 30,
[62] 66, [63] 90, [64] 90;

```

```

** Recompiled and Listed by Kanji Setsuda on
Feb.22, 2015 with MacOSX 10.10.2 & Xcode 6.1.1 **

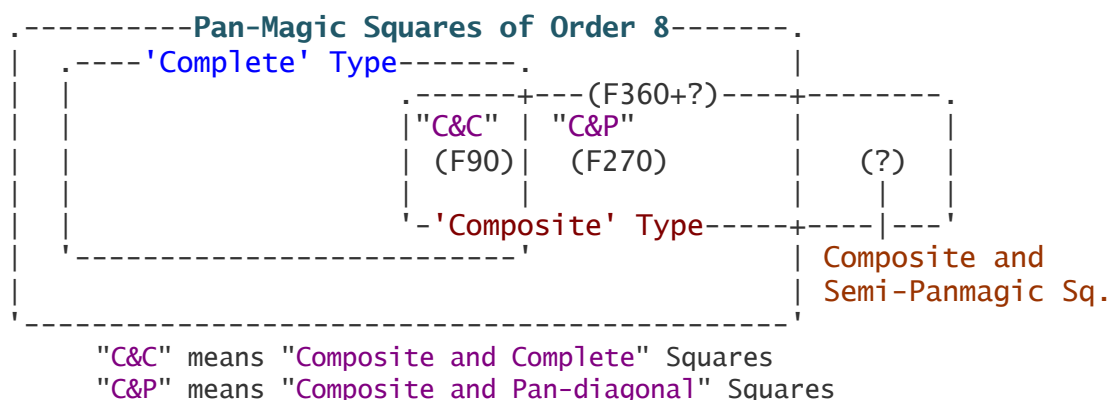
```

I found the 90 Fundamental Solutions of "C&C" magic squares before. They are really contained in this list. The C&C solutions make the subset of our pan-diagonal 360 solutions set. The count ratio is $90 : 360 = 1 : 4$.

I could also have found then how to reconstruct many other standard solutions by these fundamentals, though I skip to explain it now (I also found they have the "Ten Principal Solutions" in their list).

It is known the set of 'Pan-magic' type is usually bigger than that of 'Complete' one in higher orders, whether it is 'Composite' type or not. The 'Complete MS88' with all the complementary pairs of 65 situated only on their pandiagonals is getting rarer and more precious than the massive ordinary Pan-magic Squares of Order 8.

[Figure 4: Conceptual Diagram for the Set and its Sub-set]



(The original written in English on August 2, 2001;
Revised on June 2, 2006 with MacOSX & Xcode 2.2;
Recompiled on February 23, 2015 with MacOSX 10.10.2 & Xcode 6.1.1)

By Kanji Setsuda; E-Mail Address: jag12001@nifty.com;

<< Appendix: Program List for the 'Composite & Pan-diagonal' MS88 >>

```

/** 'Composite' and Pan-Diagonal Magic Squares of Order 8: **
/** Compact List of the 360 Fundamental Solutions(when n1=1) **
/** Files: 'CPMS88F.c' Dictated by Kanji Setsuda **

```

```

/** on Jan.28,'01; Jul.10,'06; Aug.12,'08 and Recompiled **
/** on Feb.22, 2015 with MacOSX 10.10.2 and Xcode 6.1.1; **
//
#include <stdio.h>
//
/** Global Variables *
short int cnt, cnt2;
short cntr[4];
short int cnt3;
short LSM, SSM, nc;
short nm[65], uflg[65];
short nnc[65];
short anm[4][65];
//
/** Sub-Routiens *
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), stp50(void), stp51(void), stp52(void);
void stp53(void), stp54(void), stp55(void), stp56(void);
void stp57(void), stp58(void), stp59(void), stp60(void);
void stp61(void), stp62(void), stp63(void), stp64(void);
void stp65(void), stp66(void), stp67(void);
void ansprint(void), prans(short x);
void prnnc(void);
//
/** Main Program *
int main(){
short n;
printf("\n");
printf("*** 'Composite' Type of Pan-diagonal Magic Squares of Order 8 **\n");
printf("*** Compact List of the 360 Fundamental Solutions (when n1=1) **\n");
for(n=0;n<65;n++){nm[n]=0; uflg[n]=0; nnc[n]=0;}
LSM=260; SSM=130; cnt=0; cnt3=0; nc=37;
stp01();      /** Begin the Calculations *
if(cnt3>0){prans(cnt3);}
printf(" [Count = %d] OK!\n",cnt);
printf("\n");
printf("*** Counts according to the Value of n%d **\n",nc);
prnnc();
printf("\n");
printf("*** Recompiled and Listed by Kanji Setsuda on\n");
printf(" Feb.22, 2015 with MacOSX 10.10.2 & Xcode 6.1.1 **\n");
printf("\n");
return 0;
}
//
/** Begin the Calculations *
/** 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 N2 *
void stp02(){
short a;
for(a=64;a>59;a--){
if(uflg[a]==0){
nm[2]=a; uflg[a]=1;
}
}
}

```

```

        stp03();
        uflg[a]=0;}
    }
}
/* Set N9(<N2) *
void stp03(){
    short a;
    for(a=nm[2]-1;a>43;a--){
        if(uflg[a]==0){
            nm[9]=a; uflg[a]=1;
            stp04();
            uflg[a]=0;}
        }
}
/* Set n10=SSM-n1-n2-n9 *
void stp04(){
    short a;
    a=SSM-nm[1]-nm[2]-nm[9];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){cnt2=0;
            nm[10]=a; uflg[a]=1;
            stp05();
            uflg[a]=0;}}
}
/* Set N3(>n1) *
void stp05(){
    short a;
    for(a=2;a<26;a++){
        if(uflg[a]==0){
            nm[3]=a; uflg[a]=1;
            stp06();
            uflg[a]=0;}
        }
}
/* Set n11=130-n2-n3-n10 *
void stp06(){
    short a;
    a=SSM-nm[2]-nm[3]-nm[10];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[11]=a; uflg[a]=1;
            stp07();
            uflg[a]=0;}}
}
/* Set N4(<n2) *
void stp07(){
    short a;
    for(a=nm[2]-1;a>39;a--){
        if(uflg[a]==0){cnt2=0;
            nm[4]=a; uflg[a]=1;
            stp08();
            uflg[a]=0;}
        }
}
/* Set n12=130-n3-n4-n11 *
void stp08(){
    short a;
    a=SSM-nm[3]-nm[4]-nm[11];

```

```

-----
/* Set n7=LSM-n1-n2-n3-n4-n5-n6-n8 & n3<n7<n5 *
void stp14(){
    short a;
    a=LSM-nm[1]-nm[2]-nm[3]-nm[4]-nm[5]-nm[6]-nm[8];
    if((nm[3]<a)&&(a<nm[5])){
        if(uflg[a]==0){

```

```

    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[12]=a; uflg[a]=1;
            stp09();
            uflg[a]=0;}}
    }
}
/* Level #2: *
/* Set N5(>n3) *
void stp09(){
    short a;
    for(a=nm[3]+1;a<52;a++){
        if(uflg[a]==0){
            nm[5]=a; uflg[a]=1;
            stp10();
            uflg[a]=0;}
        }
}
/* Set n13=130-n4-n5-n12 *
void stp10(){
    short a;
    a=SSM-nm[4]-nm[5]-nm[12];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[13]=a; uflg[a]=1;
            stp11();
            uflg[a]=0;}}
    }
}
/* Set N6(<n4) *
void stp11(){
    short a;
    for(a=nm[4]-1;a>14;a--){
        if(uflg[a]==0){
            nm[6]=a; uflg[a]=1;
            stp12();
            uflg[a]=0;}
        }
}
/* Set n14=SSM-n5-n6-n13 *
void stp12(){
    short a;
    a=SSM-nm[5]-nm[6]-nm[13];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[14]=a; uflg[a]=1;
            stp13();
            uflg[a]=0;}}
    }
}
/* Set N8 & n4>n8>n6 *
void stp13(){
    short a;
    for(a=nm[4]-1;a>nm[6];a--){
        if(uflg[a]==0){
            nm[8]=a; uflg[a]=1;
            stp14();
            uflg[a]=0;}
        }
    }
}

```

```

        nm[7]=a; uflg[a]=1;
        stp15();
        uflg[a]=0;}}
}
//
/** Set n15=130-n6-n7-n14 *
void stp15(){
    short a;
    a=SSM-nm[6]-nm[7]-nm[14];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[15]=a; uflg[a]=1;
            stp16();
            uflg[a]=0;}}
}
/** Set n16=130-n7-n8-n15 *
void stp16(){
    short a,b;
    a=SSM-nm[7]-nm[8]-nm[15];
    b=SSM-nm[1]-nm[8]-nm[9];
    if((1<a)&&(a<65)&&(a==b)){
        if(uflg[a]==0){
            nm[16]=a; uflg[a]=1;
            stp17();
            uflg[a]=0;}}
}
/** Level #3: *
/** Set N17(>n1) *
void stp17(){
    short a;
    for(a=nm[1]+1;a<30;a++){
        if(uflg[a]==0){
            nm[17]=a; uflg[a]=1;
            stp18();
            uflg[a]=0;}}
}
}
/** Set n18=130-n9-n10-n17 *
void stp18(){
    short a;
    a=SSM-nm[9]-nm[10]-nm[17];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[18]=a; uflg[a]=1;
            stp19();
            uflg[a]=0;}}
}
/** Set n19=130-n10-n11-n18 *
void stp19(){
    short a;
    a=SSM-nm[10]-nm[11]-nm[18];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[19]=a; uflg[a]=1;
            stp20();
            uflg[a]=0;}}
}
/** Set n20=130-n11-n12-n19 *
void stp20(){
    short a;
    a=SSM-nm[11]-nm[12]-nm[19];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[20]=a; uflg[a]=1;
            stp21();

```

```

        uflg[a]=0;}}
}
/** Set n21=130-n12-n13-n20 *
void stp21(){
    short a;
    a=SSM-nm[12]-nm[13]-nm[20];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[21]=a; uflg[a]=1;
            stp22();
            uflg[a]=0;}}
}
/** Set n22=SSM-n13-n14-n21 *
void stp22(){
    short a;
    a=SSM-nm[13]-nm[14]-nm[21];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[22]=a; uflg[a]=1;
            stp23();
            uflg[a]=0;}}
}
/** Set n23=130-n14-n15-n22 *
void stp23(){
    short a;
    a=SSM-nm[14]-nm[15]-nm[22];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[23]=a; uflg[a]=1;
            stp24();
            uflg[a]=0;}}
}
}
/** Set n24=130-n15-n16-n23 *
void stp24(){
    short a,b;
    a=SSM-nm[15]-nm[16]-nm[23];
    b=SSM-nm[9]-nm[16]-nm[17];
    if((1<a)&&(a<65)&&(a==b)){
        if(uflg[a]==0){
            nm[24]=a; uflg[a]=1;
            stp25();
            uflg[a]=0;}}
}
}
/** Level #4: *
/** Set N25(<n9) *
void stp25(){
    short a;
    for(a=nm[9]-1;a>35;a--){
        if(uflg[a]==0){
            nm[25]=a; uflg[a]=1;
            stp26();
            uflg[a]=0;}}
}
}
/** Set n26=130-n17-n18-n25 *
void stp26(){
    short a;
    a=SSM-nm[17]-nm[18]-nm[25];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){

```

```

        nm[26]=a; uflg[a]=1;
        stp27();
        uflg[a]=0;}}
}
/** Set n27=130-n18-n19-n26 *
void stp27(){
    short a;
    a=SSM-nm[18]-nm[19]-nm[26];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[27]=a; uflg[a]=1;
            stp28();
            uflg[a]=0;}}
}
/** Set n28=130-n19-n20-n27 *
void stp28(){
    short a;
    a=SSM-nm[19]-nm[20]-nm[27];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[28]=a; uflg[a]=1;
            stp29();
            uflg[a]=0;}}
}
/** Set n29=130-n20-n21-n28 *
void stp29(){
    short a;
    a=SSM-nm[20]-nm[21]-nm[28];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[29]=a; uflg[a]=1;
            stp30();
            uflg[a]=0;}}
}
/** Set n30=SSM-n21-n22-n29 *
void stp30(){
    short a;
    a=SSM-nm[21]-nm[22]-nm[29];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[30]=a; uflg[a]=1;
            stp31();
            uflg[a]=0;}}
}
/** Set n31=130-n22-n23-n30 *
void stp31(){
    short a;
    a=SSM-nm[22]-nm[23]-nm[30];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[31]=a; uflg[a]=1;
            stp32();
            uflg[a]=0;}}
}
/** Set n32=130-n23-n24-n31 *
void stp32(){
    short a,b;
    a=SSM-nm[23]-nm[24]-nm[31];
    b=SSM-nm[17]-nm[24]-nm[25];
    if((1<a)&&(a<65)&&(a==b)){
        if(uflg[a]==0){
            nm[32]=a; uflg[a]=1;
            stp33();
            uflg[a]=0;}}
}

```

```

↗/** Level #5: *
/** Set N33(>n17) *
void stp33(){
    short a;
    for(a=nm[17]+1;a<58;a++){
        if(uflg[a]==0){
            nm[33]=a; uflg[a]=1;
            stp34();
            uflg[a]=0;}}
    }
}
/** Set n34=130-n25-n26-n33 *
void stp34(){
    short a;
    a=SSM-nm[25]-nm[26]-nm[33];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[34]=a; uflg[a]=1;
            stp35();
            uflg[a]=0;}}
}
/** Set n35=130-n26-n27-n34 *
void stp35(){
    short a;
    a=SSM-nm[26]-nm[27]-nm[34];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[35]=a; uflg[a]=1;
            stp36();
            uflg[a]=0;}}
}
/** Set n36=130-n27-n28-n35 *
void stp36(){
    short a;
    a=SSM-nm[27]-nm[28]-nm[35];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[36]=a; uflg[a]=1;
            stp37();
            uflg[a]=0;}}
}
/** Set n37=130-n28-n29-n36 *
void stp37(){
    short a,b;
    a=SSM-nm[28]-nm[29]-nm[36];
    b=nm[5]+nm[33]-nm[1];
    if((1<a)&&(a<65)){
        if((uflg[a]==0)&&(a==b)){
            nm[37]=a; uflg[a]=1;
            stp38();
            uflg[a]=0;}}
}
/** Set n38=SSM-n29-n30-n37 *
void stp38(){
    short a;
    a=SSM-nm[29]-nm[30]-nm[37];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[38]=a; uflg[a]=1;
            stp39();
            uflg[a]=0;}}
}
/** Set n39=130-n30-n31-n38 *
void stp39(){
    ↗ short a;

```



```

a=SSM-nm[30]-nm[31]-nm[38];
if((1<a)&&(a<65)){
    if(uflg[a]==0){
        nm[39]=a; uflg[a]=1;
        stp40();
        uflg[a]=0;}}
}
/** Set n40=130-n31-n32-n39 *
void stp40(){
    short a,b;
    a=SSM-nm[31]-nm[32]-nm[39];
    b=SSM-nm[25]-nm[32]-nm[33];
    if((1<a)&&(a<65)&&(a==b)){
        if(uflg[a]==0){
            nm[40]=a; uflg[a]=1;
            stp41();
            uflg[a]=0;}}
}
/** Level #6: *
/** Set N41(<n25) *
void stp41(){
    short a;
    for(a=nm[25]-1;a>7;a--){
        if(uflg[a]==0){
            nm[41]=a; uflg[a]=1;
            stp42();
            uflg[a]=0;}}
}
}
/** Set n42=SSM-n33-n34-n41 *
void stp42(){
    short a;
    a=SSM-nm[33]-nm[34]-nm[41];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[42]=a; uflg[a]=1;
            stp43();
            uflg[a]=0;}}
}
}
/** Set n43=SSM-n34-n35-n42 *
void stp43(){
    short a;
    a=SSM-nm[34]-nm[35]-nm[42];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[43]=a; uflg[a]=1;
            stp44();
            uflg[a]=0;}}
}
}
/** Set n44=SSM-n35-n36-n43 *
void stp44(){
    short a;
    a=SSM-nm[35]-nm[36]-nm[43];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[44]=a; uflg[a]=1;

```

```

    stp45();
    uflg[a]=0;}}
}
}
/** Set n45=SSM-n36-n37-n44 *
void stp45(){
    short a;
    a=SSM-nm[36]-nm[37]-nm[44];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[45]=a; uflg[a]=1;
            stp46();
            uflg[a]=0;}}
}
}
/** Set n46=SSM-n37-n38-n45 *
void stp46(){
    short a;
    a=SSM-nm[37]-nm[38]-nm[45];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[46]=a; uflg[a]=1;
            stp47();
            uflg[a]=0;}}
}
}
/** Set n47=130-n38-n39-n46 *
void stp47(){
    short a;
    a=SSM-nm[38]-nm[39]-nm[46];
    if((1<a)&&(a<65)){
        if(uflg[a]==0){
            nm[47]=a; uflg[a]=1;
            stp48();
            uflg[a]=0;}}
}
}
/** Set n48=130-n39-n40-n47 *
void stp48(){
    short a,b;
    a=SSM-nm[39]-nm[40]-nm[47];
    b=SSM-nm[33]-nm[40]-nm[41];
    if((1<a)&&(a<65)&&(a==b)){
        if(uflg[a]==0){
            nm[48]=a; uflg[a]=1;
            stp49();
            uflg[a]=0;}}
}
}
/** Level #7: *
/** Set N57 & n25>n57>n41 *
void stp49(){
    short a;
    for(a=nm[25]-1;a>nm[41];a--){
        if(uflg[a]==0){
            nm[57]=a; uflg[a]=1;
            stp50();
            uflg[a]=0;}}
}
}

```

```

-----
/** Set n49=LSM-n1-n9-n17-n25-n33-n41-n57 & n17<n49<n33 *
void stp50(){
    short a;
    a=LSM-nm[1]-nm[9]-nm[17]-nm[25]-nm[33]-nm[41]-nm[57];
    if((nm[17]<a)&&(a<nm[33])){
        if(uflg[a]==0){
            nm[49]=a; uflg[a]=1;
            stp51();

```

```

        uflg[a]=0;}}
}
/** Set n50=LSM-n8-n15-n22-n29-n36-n43-n57 *
void stp51(){
short a,b;
a=LSM-nm[8]-nm[15]-nm[22]-nm[29]-nm[36]-nm[43]-nm[57];
b=SSM-nm[41]-nm[42]-nm[49];
if((1<a)&&(a<65)&&(a==b)){
    if(uflg[a]==0){
        nm[50]=a; uflg[a]=1;
        stp52();
        uflg[a]=0;}}
}
/** Set n58=LSM-n3-n12-n21-n30-n39-n48-n49 *
void stp52(){
short a,b,c;
a=LSM-nm[3]-nm[12]-nm[21]-nm[30]-nm[39]-nm[48]-nm[49];
b=SSM-nm[49]-nm[50]-nm[57];
c=SSM-nm[1]-nm[2]-nm[57];
if((1<a)&&(a<65)&&(a==b)&&(a==c)){
    if(uflg[a]==0){
        nm[58]=a; uflg[a]=1;
        stp53();
        uflg[a]=0;}}
}
/** Set n51=LSM-n1-n16-n23-n30-n37-n44-n58 *
void stp53(){
short a,b;
a=LSM-nm[1]-nm[16]-nm[23]-nm[30]-nm[37]-nm[44]-nm[58];
b=SSM-nm[42]-nm[43]-nm[50];
if((1<a)&&(a<65)&&(a==b)){
    if(uflg[a]==0){
        nm[51]=a; uflg[a]=1;
        stp54();
        uflg[a]=0;}}
}
/** Set n59=LSM-n4-n13-n22-n31-n40-n41-n50 *
void stp54(){
short a,b,c;
a=LSM-nm[4]-nm[13]-nm[22]-nm[31]-nm[40]-nm[41]-nm[50];
b=SSM-nm[50]-nm[51]-nm[58];
c=SSM-nm[2]-nm[3]-nm[58];
if((1<a)&&(a<65)&&(a==b)&&(a==c)){
    if(uflg[a]==0){
        nm[59]=a; uflg[a]=1;
        stp55();
        uflg[a]=0;}}
}
/** Set n52=LSM-n2-n9-n24-n31-n38-n45-n59 *
void stp55(){
short a,b;
a=LSM-nm[2]-nm[9]-nm[24]-nm[31]-nm[38]-nm[45]-nm[59];
b=SSM-nm[43]-nm[44]-nm[51];
if((1<a)&&(a<65)&&(a==b)){
    if(uflg[a]==0){
        nm[52]=a; uflg[a]=1;
        stp56();
        uflg[a]=0;}}
}
/** Set n60=LSM-n5-n14-n23-n32-n33-n42-n51 *
void stp56(){
short a,b,c;
a=LSM-nm[5]-nm[14]-nm[23]-nm[32]-nm[33]-nm[42]-nm[51];
b=SSM-nm[51]-nm[52]-nm[59];
c=SSM-nm[3]-nm[4]-nm[59];

```

```

    if((1<a)&&(a<65)&&(a==b)&&(a==c)){
        if(uflg[a]==0){
            nm[60]=a; uflg[a]=1;
            stp57();
            uflg[a]=0;}}
}
/** Set n53=LSM-n3-n10-n17-n32-n39-n46-n60 *
void stp57(){
    short a,b;
    a=LSM-nm[3]-nm[10]-nm[17]-nm[32]-nm[39]-nm[46]-nm[60];
    b=SSM-nm[44]-nm[45]-nm[52];
    if((1<a)&&(a<65)&&(a==b)){
        if(uflg[a]==0){
            nm[53]=a; uflg[a]=1;
            stp58();
            uflg[a]=0;}}
}
/** Set n61=LSM-n6-n15-n24-n25-n34-n43-n52 *
void stp58(){
    short a,b,c;
    a=LSM-nm[6]-nm[15]-nm[24]-nm[25]-nm[34]-nm[43]-nm[52];
    b=SSM-nm[52]-nm[53]-nm[60];
    c=SSM-nm[4]-nm[5]-nm[60];
    if((1<a)&&(a<65)&&(a==b)&&(a==c)){
        if(uflg[a]==0){
            nm[61]=a; uflg[a]=1;
            stp59();
            uflg[a]=0;}}
}
/** Set n54=LSM-n4-n11-n18-n25-n40-n47-n61 *
void stp59(){
    short a,b;
    a=LSM-nm[4]-nm[11]-nm[18]-nm[25]-nm[40]-nm[47]-nm[61];
    b=SSM-nm[45]-nm[46]-nm[53];
    if((1<a)&&(a<65)&&(a==b)){
        if(uflg[a]==0){
            nm[54]=a; uflg[a]=1;
            stp60();
            uflg[a]=0;}}
}
/** Set n62=LSM-n7-n16-n17-n26-n35-n44-n53 *
void stp60(){
    short a,b,c;
    a=LSM-nm[7]-nm[16]-nm[17]-nm[26]-nm[35]-nm[44]-nm[53];
    b=SSM-nm[53]-nm[54]-nm[61];
    c=SSM-nm[5]-nm[6]-nm[61];
    if((1<a)&&(a<65)&&(a==b)&&(a==c)){
        if(uflg[a]==0){
            nm[62]=a; uflg[a]=1;
            stp61();
            uflg[a]=0;}}
}
/** Set n55=LSM-n5-n12-n19-n26-n33-n48-n62 *
void stp61(){
    short a,b;
    a=LSM-nm[5]-nm[12]-nm[19]-nm[26]-nm[33]-nm[48]-nm[62];
    b=SSM-nm[46]-nm[47]-nm[54];
    if((1<a)&&(a<65)&&(a==b)){
        if(uflg[a]==0){
            nm[55]=a; uflg[a]=1;
            stp62();
            uflg[a]=0;}}
}
/** Set n63=LSM-n8-n9-n18-n27-n36-n45-n54 *
void stp62(){

```

```

short a,b,c;
a=LSM-nm[8]-nm[9]-nm[18]-nm[27]-nm[36]-nm[45]-nm[54];
b=SSM-nm[54]-nm[55]-nm[62];
c=SSM-nm[6]-nm[7]-nm[62];
if((1<a)&&(a<65)&&(a==b)&&(a==c)){
    if(uflg[a]==0){
        nm[63]=a; uflg[a]=1;
        stp63();
        uflg[a]=0;}}
}
/* Set n56=LSM-n2-n11-n20-n29-n38-n47-n57 *
void stp63(){
short a,b,c,d;
a=LSM-nm[2]-nm[11]-nm[20]-nm[29]-nm[38]-nm[47]-nm[57];
b=LSM-nm[6]-nm[13]-nm[20]-nm[27]-nm[34]-nm[41]-nm[63];
c=SSM-nm[47]-nm[48]-nm[55];
d=SSM-nm[48]-nm[41]-nm[49];
if((1<a)&&(a<65)&&(a==b)&&(a==c)&&(a==d)){
    if(uflg[a]==0){
        nm[56]=a; uflg[a]=1;
        stp64();
        uflg[a]=0;}}
}
/* Set n64=LSM-n1-n10-n19-n28-n37-n46-n55 & n1<n64 *
void stp64(){
short a,b,c;
a=LSM-nm[1]-nm[10]-nm[19]-nm[28]-nm[37]-nm[46]-nm[55];
b=LSM-nm[7]-nm[14]-nm[21]-nm[28]-nm[35]-nm[42]-nm[49];
c=SSM-nm[55]-nm[56]-nm[63];
if((1<a)&&(a<65)&&(a==b)&&(a==c)){
    if(uflg[a]==0){
        nm[64]=a; uflg[a]=1;
        stp65();
        uflg[a]=0;}}
}
/* Check the Composite Sums *
void stp65(){
short sm1,sm2,sm3;
sm1=nm[56]+nm[49]+nm[64]+nm[57];
sm2=nm[63]+nm[64]+nm[7]+nm[8];
sm3=nm[64]+nm[57]+nm[8]+nm[1];
if((sm1==SSM)&&(sm2==SSM)&&(sm3==SSM)){ansprint();}
}
//
/* Print The Answers *
void ansprint(){
short n;
cnt++;
n=nm[nc]; nnc[n]++;
cnt2++;
if(cnt2==1){
    cntr[cnt3]=cnt;
    for(n=1;n<65;n++){anm[cnt3][n]=nm[n];}
    cnt3++; if(cnt3==3){prans(cnt3); cnt3=0;}
}
}
//
/* Print 3 Answers *
void prans(short x){
short l,l8,m,n;
for(m=0;m<x;m++){
    printf("%24d#",cntr[m]);
    if(m+1<x){printf(" ");}
}
printf("\n");

```

```

for(l=0;l<8;l++){l8=l*8;
  for(m=0;m<x;m++){
    printf(" ");
    for(n=1;n<9;n++){printf("%3d",anm[m][l8+n]);}
    if(m+1<x){printf(" ");}
  }
  printf("\n");
}
}
//
/* Print the Counts according to the value of nm[nc] *
void prnnc(){
short m,n,c;
c=0;
for(m=1;m<65;m++){
  n=nnc[m];
  if(n>0){
    printf(" [%2d]%3d",m,n);
    c++; if(c%8==0){printf("\n");}
  }
}
printf("\n");
}
//

```