

第3部：『魔方陣の新研究』： 摂田 寛二
 第7章： 九方陣の代数的研究（新）
 第1節： 対称汎九方陣の代数的研究（2000年稿清書版）

0. 九方陣くらいにスケールが大きくなると、対称型も汎型も、一般解はなかなか求め難い。解が多過ぎる。時間がかかり過ぎる。解の例を幾つか得ることができても、1個ずつカウントアップして総数を知ることは現実に不可能である。

だから今後は、少しでも珍しい方陣を追求せざるを得ない。

「対称汎九方陣」は、汎九方陣でもあり対称九方陣でもある。両方の性質を併せもつ。つまり、汎九方陣の解の集合と対称九方陣の解の集合の「論理的積集合」に含まれる解のことである。大変に珍しい方陣だが、既に5次、7次の平面魔方陣で見つかっている。高次方陣では解の数も多そうだが、9次の場合はどうであろうか。

いつものように図と式によって、基本条件を次のように定義する。

1. 汎九方陣の拡張空間での基本ポジションと変数名：

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

2. 汎九方陣の連立方程式群：(19)～(36)式が汎対角線の定和を表す。

- $n_1 + n_2 + n_3 + n_4 + n_5 + n_6 + n_7 + n_8 + n_9 = K \dots (1)$
- $n_{10} + n_{11} + n_{12} + n_{13} + n_{14} + n_{15} + n_{16} + n_{17} + n_{18} = K \dots (2)$
- $n_{19} + n_{20} + n_{21} + n_{22} + n_{23} + n_{24} + n_{25} + n_{26} + n_{27} = K \dots (3)$
- $n_{28} + n_{29} + n_{30} + n_{31} + n_{32} + n_{33} + n_{34} + n_{35} + n_{36} = K \dots (4)$
- $n_{37} + n_{38} + n_{39} + n_{40} + n_{41} + n_{42} + n_{43} + n_{44} + n_{45} = K \dots (5)$
- $n_{46} + n_{47} + n_{48} + n_{49} + n_{50} + n_{51} + n_{52} + n_{53} + n_{54} = K \dots (6)$
- $n_{55} + n_{56} + n_{57} + n_{58} + n_{59} + n_{60} + n_{61} + n_{62} + n_{63} = K \dots (7)$
- $n_{64} + n_{65} + n_{66} + n_{67} + n_{68} + n_{69} + n_{70} + n_{71} + n_{72} = K \dots (8)$
- $n_{73} + n_{74} + n_{75} + n_{76} + n_{77} + n_{78} + n_{79} + n_{80} + n_{81} = K \dots (9)$
- $n_1 + n_{10} + n_{19} + n_{28} + n_{37} + n_{46} + n_{55} + n_{64} + n_{73} = K \dots (10)$
- $n_2 + n_{11} + n_{20} + n_{29} + n_{38} + n_{47} + n_{56} + n_{65} + n_{74} = K \dots (11)$
- $n_3 + n_{12} + n_{21} + n_{30} + n_{39} + n_{48} + n_{57} + n_{66} + n_{75} = K \dots (12)$
- $n_4 + n_{13} + n_{22} + n_{31} + n_{40} + n_{49} + n_{58} + n_{67} + n_{76} = K \dots (13)$
- $n_5 + n_{14} + n_{23} + n_{32} + n_{41} + n_{50} + n_{59} + n_{68} + n_{77} = K \dots (14)$
- $n_6 + n_{15} + n_{24} + n_{33} + n_{42} + n_{51} + n_{60} + n_{69} + n_{78} = K \dots (15)$
- $n_7 + n_{16} + n_{25} + n_{34} + n_{43} + n_{52} + n_{61} + n_{70} + n_{79} = K \dots (16)$
- $n_8 + n_{17} + n_{26} + n_{35} + n_{44} + n_{53} + n_{62} + n_{71} + n_{80} = K \dots (17)$
- $n_9 + n_{18} + n_{27} + n_{36} + n_{45} + n_{54} + n_{63} + n_{72} + n_{81} = K \dots (18)$

- $n_1 + n_{11} + n_{21} + n_{31} + n_{41} + n_{51} + n_{61} + n_{71} + n_{81} = K \dots (19)$
- $n_1 + n_{18} + n_{26} + n_{34} + n_{42} + n_{50} + n_{58} + n_{66} + n_{74} = K \dots (20)$
- $n_2 + n_{12} + n_{22} + n_{32} + n_{42} + n_{52} + n_{62} + n_{72} + n_{73} = K \dots (21)$
- $n_2 + n_{10} + n_{27} + n_{35} + n_{43} + n_{51} + n_{59} + n_{67} + n_{75} = K \dots (22)$
- $n_3 + n_{13} + n_{23} + n_{33} + n_{43} + n_{53} + n_{63} + n_{64} + n_{74} = K \dots (23)$
- $n_3 + n_{11} + n_{19} + n_{36} + n_{44} + n_{52} + n_{60} + n_{68} + n_{76} = K \dots (24)$
- $n_4 + n_{14} + n_{24} + n_{34} + n_{44} + n_{54} + n_{55} + n_{65} + n_{75} = K \dots (25)$
- $n_4 + n_{12} + n_{20} + n_{28} + n_{45} + n_{53} + n_{61} + n_{69} + n_{77} = K \dots (26)$
- $n_5 + n_{15} + n_{25} + n_{35} + n_{45} + n_{46} + n_{56} + n_{66} + n_{76} = K \dots (27)$
- $n_5 + n_{13} + n_{21} + n_{29} + n_{37} + n_{54} + n_{62} + n_{70} + n_{78} = K \dots (28)$
- $n_6 + n_{16} + n_{26} + n_{36} + n_{37} + n_{47} + n_{57} + n_{67} + n_{77} = K \dots (29)$
- $n_6 + n_{14} + n_{22} + n_{30} + n_{38} + n_{46} + n_{63} + n_{71} + n_{79} = K \dots (30)$
- $n_7 + n_{17} + n_{27} + n_{28} + n_{38} + n_{48} + n_{58} + n_{68} + n_{78} = K \dots (31)$
- $n_7 + n_{15} + n_{23} + n_{31} + n_{39} + n_{47} + n_{55} + n_{72} + n_{80} = K \dots (32)$
- $n_8 + n_{18} + n_{19} + n_{29} + n_{39} + n_{49} + n_{59} + n_{69} + n_{79} = K \dots (33)$
- $n_8 + n_{16} + n_{24} + n_{32} + n_{40} + n_{48} + n_{56} + n_{64} + n_{81} = K \dots (34)$
- $n_9 + n_{10} + n_{20} + n_{30} + n_{40} + n_{50} + n_{60} + n_{70} + n_{80} = K \dots (35)$
- $n_9 + n_{17} + n_{25} + n_{33} + n_{41} + n_{49} + n_{57} + n_{65} + n_{73} = K \dots (36)$

3. 対称九方阵の基本図 (補数表現型)

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

対称性を表す補数和の等式群：

- $n_1 + n_{81} = n_2 + n_{80} = n_3 + n_{79} =$
 - $n_4 + n_{78} = n_5 + n_{77} = n_6 + n_{76} =$
 - $n_7 + n_{75} = n_8 + n_{74} = n_9 + n_{73} =$
 - $n_{10} + n_{72} = n_{11} + n_{71} = n_{12} + n_{70} =$
 - $n_{13} + n_{69} = n_{14} + n_{68} = n_{15} + n_{67} =$
 - $n_{16} + n_{66} = n_{17} + n_{65} = n_{18} + n_{64} =$
 - $n_{19} + n_{63} = n_{20} + n_{62} = n_{21} + n_{61} =$
 - $n_{22} + n_{60} = n_{23} + n_{59} = n_{24} + n_{58} =$
 - $n_{25} + n_{57} = n_{26} + n_{56} = n_{27} + n_{55} =$
 - $n_{28} + n_{54} = n_{29} + n_{53} = n_{30} + n_{52} =$
 - $n_{31} + n_{51} = n_{32} + n_{50} = n_{33} + n_{49} =$
 - $n_{34} + n_{48} = n_{35} + n_{47} = n_{36} + n_{46} =$
 - $n_{37} + n_{45} = n_{38} + n_{44} = n_{39} + n_{43} =$
 - $n_{40} + n_{42} = n_{41} + n_{41} = C$
- 補数表現で言えば、
- $n_1 + n_1 = n_2 + n_2 = n_3 + n_3 =$
 - $n_4 + n_4 = n_5 + n_5 = \dots \dots \dots$
 - $= n_{40} + n_{40} = n_{41} + n_{41} = C$
- ということになる。

ここで、K、Cなどの定数値を予め求めておこう。

(1)～(9)式の辺々を加える。

$$n_1 + n_2 + n_3 + \dots + n_{80} + n_{81} = 9K$$

左辺には、1～81の自然数が重複なく入る。その総和は、

$$81 \times 82 \div 2 = 9K$$

ゆえに $K = 369$

Cの値を決めておこう。

上の表の主対角線を見て、 $n_1 + n_{11} + n_{21} + n_{31} + n_{41} + n_{51} + n_{61} + n_{71} + n_{81} = K$

これを補数で表現すれば、

$$n_1 + n_{11} + n_{21} + n_{31} + n_{41} + n_{31} + n_{21} + n_{11} + n_1 = K$$

$$(n_1 + n_1) + (n_{11} + n_{11}) + (n_{21} + n_{21}) + (n_{31} + n_{31}) + n_{41} = K$$

$$\text{すなわち } C + C + C + C + C/2 = K$$

$$\text{両辺を2倍して, } 9C = 2K = 369 \times 2 = 738$$

ゆえに $C = 82$

上の40ペアが全て82の補数を成すことがわかった。

なお 方阵の中心に位置する **n41** には、自分自身との和が82の補数を成す41が入る。

4. 先ず汎九方阵の性質を研究する。七方阵の例に習い、「三角連結隅(すみ)10数

1	2	3	4	5	6	7	8	9		
P							Q			
10	11	12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	27		
28	29	30	31	32	33	34	35	36		
37	38	39	40	41	42	43	44	45		
46	47	48	49	50	51	52	53	54		
55	56	57	58	59	60	61	62	63		
64	65	66	67	68	69	70	71	72		
R							S			
73	74	75	76	77	78	79	80	81		

の等和」に関する式を集めよう。

左の図で

$$n_1 + n_2 + n_3 + n_4 + n_{10} +$$

$$n_{11} + n_{12} + n_{19} + n_{20} + n_{28} = P$$

$$n_6 + n_7 + n_8 + n_9 + n_{16} +$$

$$n_{17} + n_{18} + n_{26} + n_{27} + n_{36} = Q$$

$$n_{46} + n_{55} + n_{56} + n_{64} + n_{65} +$$

$$n_{66} + n_{73} + n_{74} + n_{75} + n_{76} = R$$

$$n_{54} + n_{62} + n_{63} + n_{70} + n_{71} +$$

$$n_{72} + n_{78} + n_{79} + n_{80} + n_{81} = S$$

と、ブロック名を付ける。

$$P = S \quad \text{および} \quad Q = R$$

が成り立つことを示したいのだが、

そのために先ず何とか、

$$P + (\cdots A \cdots) = S + (\cdots B \cdots)$$

の形に持ち込み、

$$P = S \quad \text{と} \quad A = B$$

の両方が成り立つことを示したい。

一番簡単なのは、左図を見れば、

$$P + (n_5 + n_{13} + n_{14} + n_{21} + n_{22} + n_{23} + n_{29} + n_{30} + n_{31} + n_{32})$$

$$= 4K - (n_6 + n_7 + n_8 + n_9 + n_{15} + n_{16} + n_{17} + n_{18} +$$

$$n_{24} + n_{25} + n_{26} + n_{27} + n_{33} + n_{34} + n_{35} + n_{36})$$

$$S + (n_{42} + n_{43} + n_{44} + n_{45} + n_{51} + n_{52} + n_{53} + n_{60} + n_{61} + n_{69})$$

$$= 4K - (n_6 + n_7 + n_8 + n_9 + n_{15} + n_{16} + n_{17} + n_{18} +$$

$$n_{24} + n_{25} + n_{26} + n_{27} + n_{33} + n_{34} + n_{35} + n_{36})$$

$$n_5 + n_{13} + n_{14} + n_{21} + n_{22} + n_{23} + n_{29} + n_{30} + n_{31} + n_{32} = U$$

$$n_{42} + n_{43} + n_{44} + n_{45} + n_{51} + n_{52} + n_{53} + n_{60} + n_{61} + n_{69} = V$$

とおけば、上の2式は右辺が同じなので、 $P + U = S + V$ だ。

この (U, V) のブロックは、(P, S) と同様に汎九方陣拡張空間で「三角連結隅10数」のペアを成している。したがって、 $P = S$ が成り立てば、 $U = V$ が成り立たなければならない。 $U = V$ が成り立てば、 $P = S$ が成り立たなければならない。

別の方法も試してみよう。次の5式を加える。

$$n_1 + n_2 + n_3 + n_4 + n_5 + n_6 + n_7 + n_8 + n_9 = K \quad \dots (1)$$

$$n_3 + n_{11} + n_{19} + n_{36} + n_{44} + n_{52} + n_{60} + n_{68} + n_{76} = K \quad \dots (24)$$

$$n_6 + n_{16} + n_{26} + n_{36} + n_{37} + n_{47} + n_{57} + n_{67} + n_{77} = K \quad \dots (29)$$

$$n_7 + n_{17} + n_{27} + n_{28} + n_{38} + n_{48} + n_{58} + n_{68} + n_{78} = K \quad \dots (31)$$

$$+) n_4 + n_{12} + n_{20} + n_{28} + n_{45} + n_{53} + n_{61} + n_{69} + n_{77} = K \quad \dots (26)$$

$$(n_1 + n_2 + n_3 + n_4 + n_{11} + n_{12} + n_{19} + n_{20} + n_{28}) + (n_6 + n_7 + n_8 + n_9 + n_{16} + n_{17} + n_{26} + n_{36} + n_{27}) + (n_5 + n_{37} + n_{45} + n_{77}) + (n_3 + n_4 + n_6 + n_7 + n_{28} + n_{36} + n_{38} + n_{44} + n_{47} + n_{48} + n_{52} + n_{53} + n_{57} + n_{58} + n_{60} + n_{61} + n_{67} + n_{68} + n_{69} + n_{76} + n_{77} + n_{78}) = 5K$$

両辺に $(n_{10} + n_{18})$ を加える。

$$(n_1 + n_2 + n_3 + n_4 + n_{10} + n_{11} + n_{12} + n_{19} + n_{20} + n_{28}) + (n_6 + n_7 + n_8 + n_9 + n_{16} + n_{17} + n_{18} + n_{26} + n_{36} + n_{27}) + (n_5 + n_{37} + n_{45} + n_{77}) + (n_3 + n_4 + n_6 + n_7 + n_{28} + n_{36} + n_{38} + n_{44} + n_{47} + n_{48} + n_{52} + n_{53} + n_{57} + n_{58} + n_{60} + n_{61} + n_{67} + n_{68} + n_{69} + n_{76} + n_{77} + n_{78}) = 5K + (n_{10} + n_{18})$$

$$P + Q + W + (n_3 + n_4 + n_6 + n_7 + n_{28} + n_{36} + n_{38} + n_{44} + n_{47} + n_{48} + n_{52} + n_{53} + n_{57} + n_{58} + n_{60} + n_{61} + n_{67} + n_{68} + n_{69} + n_{76} + n_{77} + n_{78}) = 5K + (n_{10} + n_{18}) \quad \dots (40)$$

次の5式を加える。

$$n_9 + n_{18} + n_{27} + n_{36} + n_{45} + n_{54} + n_{63} + n_{72} + n_{81} = K \quad \dots (18)$$

$$n_6 + n_{16} + n_{26} + n_{36} + n_{37} + n_{47} + n_{57} + n_{67} + n_{77} = K \quad \dots (29)$$

$$n_7 + n_{17} + n_{27} + n_{28} + n_{38} + n_{48} + n_{58} + n_{68} + n_{78} = K \quad \dots (31)$$

$$n_5 + n_{13} + n_{21} + n_{29} + n_{37} + n_{54} + n_{62} + n_{70} + n_{78} = K \quad \dots (28)$$

$$+) n_6 + n_{14} + n_{22} + n_{30} + n_{38} + n_{46} + n_{63} + n_{71} + n_{79} = K \quad \dots (30)$$

$$(n_6 + n_7 + n_9 + n_{16} + n_{17} + n_{18} + n_{26} + n_{27} + n_{36}) + (n_{54} + n_{62} + n_{63} + n_{70} + n_{71} + n_{72} + n_{78} + n_{79} + n_{81}) + (n_5 + n_{37} + n_{45} + n_{77}) + (n_6 + n_{13} + n_{14} + n_{21} + n_{22} + n_{27} + n_{28} + n_{29} + n_{30} + n_{36} + n_{37} + n_{38} + n_{38} + n_{46} + n_{47} + n_{48} + n_{54} + n_{57} + n_{58} + n_{63} + n_{67} + n_{68} + n_{78}) = 5K$$

両辺に $(n8 + n80)$ を加える。

$$(n6+n7+n8+n9+n16+n17+n18+n26+n27+n36) + (n54+n62+n63+n70+n71+n72+n78+n79+n80+n81) + (n5+n37+n45+n77) + (n5+n13+n14+n21+n22+n27+n28+n29+n30+n36+n37+n38+n38+n46+n47+n48+n54+n57+n58+n63+n67+n68+n78) = 5K + (n8 + n80)$$

$$Q + S + W + (n6+n13+n14+n21+n22+n27+n28+n29+n30+n36+n37+n38+n38+n46+n47+n48+n54+n57+n58+n63+n67+n68+n78) = 5K + (n8+n80) \dots (41)$$

(40)-(41)

$$P + (n3+n4+n6+n7+n8+n28+n36+n38+n44+n47+n48+n52+n53+n57+n58+n60+n61+n67+n68+n68+n69+n76+n77+n78+n80) = S + (n6+n10+n13+n14+n18+n21+n22+n27+n28+n29+n30+n36+n37+n38+n38+n46+n47+n48+n54+n57+n58+n63+n67+n68+n78)$$

同類項を整理して

$$P + (n3+n4+n7+n8+n44+n52+n53+n60+n61+n68+n69+n76+n77+n80)$$

$$= S + (n10+n13+n14+n18+n21+n22+n27+n29+n30+n37+n38+n46+n54+n63)$$

項数を少なくするために工夫しよう。両辺に $(n11+n19+n36+n6+n71+n79)$ を加える。

$$P + (n3+n11+n19+n36+n44+n52+n60+n68+n76) + (n4+n6+n7+n8+n53+n61+n69+n71+n77+n79+n80)$$

$$= S + (n6+n14+n22+n30+n38+n46+n63+n71+n79) + (n10+n11+n13+n18+n19+n21+n27+n29+n37+n54+n36)$$

$$\text{ところが } n6+n14+n22+n30+n38+n46+n63+n71+n79 = K \dots (30)$$

$$n3+n11+n19+n36+n44+n52+n60+n68+n76 = K \dots (24)$$

両辺から K を引いて、

$$P + (n4+n6+n7+n8+n53+n61+n69+n71+n77+n79+n80)$$

$$= S + (n10+n11+n13+n18+n19+n21+n27+n29+n37+n54+n36) \dots (42)$$

今度は媒介項を R に変える。

$$n1 + n10 + n19 + n28 + n37 + n46 + n55 + n64 + n73 = K \dots (10)$$

$$n3 + n11 + n19 + n36 + n44 + n52 + n60 + n68 + n76 = K \dots (24)$$

$$n4 + n12 + n20 + n28 + n45 + n53 + n61 + n69 + n77 = K \dots (26)$$

$$n4 + n14 + n24 + n34 + n44 + n54 + n55 + n65 + n75 = K \dots (25)$$

$$+) n5 + n15 + n25 + n35 + n45 + n46 + n56 + n66 + n76 = K \dots (27)$$

$$(n1 + n3 + n4 + n10 + n11 + n12 + n19 + n20 + n28) + (n46 + n55 + n56 + n64 + n65 + n66 + n73 + n75 + n76) + (n5 + n37 + n45 + n77) + (n4 + n14 + n15 + n19 + n24 + n25 + n28 + n34 + n35 + n36 + n44 + n44 + n45 + n46 + n52 + n53 + n54 + n55 + n60 + n61 + n68 + n69 + n76) = 5K$$

両辺に $(n2 + n74)$ を加える。

$$(n1 + n2 + n3 + n4 + n10 + n11 + n12 + n19 + n20 + n28) + (n46 + n55 + n56 + n64 + n65 + n66 + n73 + n74 + n75 + n76) + (n5 + n37 + n45 + n77) + (n4 + n14 + n15 + n19 + n24 + n25 + n28 + n34 + n35 + n36 + n44 + n44 + n45 + n46 + n52 + n53 + n54 + n55 + n60 + n61 + n68 + n69 + n76) = 5K + (n2 + n74)$$

$$P + R + W + (n4 + n14 + n15 + n19 + n24 + n25 + n28 + n34 + n35 + n36 + n44 + n44 + n45 + n46 + n52 + n53 + n54 + n55 + n60 + n61 + n68 + n69 + n76) = 5K + (n2 + n74) \dots (43)$$

$$n73 + n74 + n75 + n76 + n77 + n78 + n79 + n80 + n81 = K \dots (9)$$

$$n5 + n15 + n25 + n35 + n45 + n46 + n56 + n66 + n76 = K \dots (27)$$

$$n4 + n14 + n24 + n34 + n44 + n54 + n55 + n65 + n75 = K \dots (25)$$

$$n5 + n13 + n21 + n29 + n37 + n54 + n62 + n70 + n78 = K \dots (28)$$

$$+) n6 + n14 + n22 + n30 + n38 + n46 + n63 + n71 + n79 = K \dots (30)$$

$$(n46 + n55 + n56 + n65 + n66 + n73 + n74 + n75 + n76) + (n54 + n62 + n63 + n70 + n71 + n78 + n79 + n80 + n81) + (n5 + n37 + n45 + n77) + (n4 + n5 + n6 + n13 + n14 + n14 + n15 + n21 + n22 + n24 + n25 + n29 + n30 + n34 + n35 + n38 + n44 + n46 + n54 + n75 + n76 + n78 + n79) = 5K$$

両辺に $(n64 + n72)$ を加える。

$$(n46 + n55 + n56 + n64 + n65 + n66 + n73 + n74 + n75 + n76) + (n54 + n62 + n63 + n70 + n71 + n72 + n78 + n79 + n80 + n81) + (n5 + n37 + n45 + n77) + (n4 + n5 + n6 + n13 + n14 + n14 + n15 + n21 + n22 + n24 + n25 + n29 + n30 + n34 + n35 + n38 + n44 + n46 + n54 + n75 + n76 + n78 + n79) = 5K + (n64 + n72)$$

$$R + S + W + (n4 + n5 + n6 + n13 + n14 + n14 + n15 + n21 + n22 + n24 + n25 + n29 + n30 + n34 + n35 + n38 + n44 + n46 + n54 + n75 + n76 + n78 + n79) = 5K + (n64 + n72) \dots (44)$$

(43)-(44)

$$P + (n19 + n28 + n36 + n44 + n45 + n52 + n53 + n55 + n60 + n61 + n64 + n68 + n69 + n72)$$

$$= S + (n2 + n5 + n6 + n13 + n14 + n21 + n22 + n29 + n30 + n38 + n74 + n75 + n78 + n79) \dots (45)$$

項数を少なくするために工夫を講じる。

$$P + (n19 + n36 + n44 + n52 + n60 + n68) + (n28 + n45 + n53 + n55 + n61 + n64 + n69 + n72)$$

$$= S + (n_6 + n_{14} + n_{22} + n_{30} + n_{38} + n_{79}) + (n_2 + n_5 + n_{13} + n_{21} + n_{29} + n_{74} + n_{75} + n_{78})$$

両辺に $(n_3 + n_{11} + n_{76} + n_{46} + n_{63} + n_{71})$ を加える。

$$P + (n_3 + n_{11} + n_{19} + n_{36} + n_{44} + n_{52} + n_{60} + n_{68} + n_{76}) + (n_{28} + n_{45} + n_{46} + n_{53} + n_{55} + n_{61} + n_{63} + n_{64} + n_{69} + n_{71} + n_{72})$$

$$= S + (n_6 + n_{14} + n_{22} + n_{30} + n_{38} + n_{46} + n_{63} + n_{71} + n_{79}) + (n_2 + n_3 + n_5 + n_{11} + n_{13} + n_{21} + n_{29} + n_{74} + n_{75} + n_{78} + n_{76})$$

ところが、 $n_3 + n_{11} + n_{19} + n_{36} + n_{44} + n_{52} + n_{60} + n_{68} + n_{76} = K$

$$n_6 + n_{14} + n_{22} + n_{30} + n_{38} + n_{46} + n_{63} + n_{71} + n_{79} = K$$

両辺から K を引いて、

$$P + (n_{28} + n_{45} + n_{46} + n_{53} + n_{55} + n_{61} + n_{63} + n_{64} + n_{69} + n_{71} + n_{72})$$

$$= S + (n_2 + n_3 + n_5 + n_{11} + n_{13} + n_{21} + n_{29} + n_{74} + n_{75} + n_{78} + n_{76}) \dots (46)$$

(42)式と似たような式が得られた。試みに両式を加えてみる。

(42) + (46)

$$2P + (n_4 + n_5 + n_7 + n_8 + n_{28} + n_{45} + n_{46} + n_{53} + n_{53} + n_{55} + n_{61} + n_{61} + n_{63} + n_{64} + n_{69} + n_{69} + n_{71} + n_{71} + n_{72} + n_{77} + n_{79} + n_{80})$$

$$= 2S + (n_2 + n_3 + n_5 + n_{10} + n_{11} + n_{11} + n_{13} + n_{13} + n_{18} + n_{19} + n_{21} + n_{21} + n_{27} + n_{29} + n_{29} + n_{36} + n_{37} + n_{54} + n_{74} + n_{75} + n_{76} + n_{78})$$

項数を少なくするために前と同様の工夫を講じる。

$$2P + (n_4 + n_{28} + n_{45} + n_{53} + n_{61} + n_{69} + n_{77}) + (n_6 + n_7 + n_8 + n_{46} + n_{53} + n_{55} + n_{61} + n_{63} + n_{64} + n_{69} + n_{71} + n_{71} + n_{72} + n_{79} + n_{80})$$

$$= 2S + (n_5 + n_{13} + n_{21} + n_{29} + n_{37} + n_{54} + n_{78}) + (n_2 + n_3 + n_{10} + n_{11} + n_{11} + n_{13} + n_{18} + n_{19} + n_{21} + n_{27} + n_{29} + n_{36} + n_{74} + n_{75} + n_{76})$$

両辺に $(n_{12} + n_{20} + n_{62} + n_{70})$ を加える。

$$2P + (n_4 + n_{12} + n_{20} + n_{28} + n_{45} + n_{53} + n_{61} + n_{69} + n_{77}) + (n_6 + n_7 + n_8 + n_{46} + n_{53} + n_{55} + n_{61} + n_{62} + n_{63} + n_{64} + n_{69} + n_{70} + n_{71} + n_{71} + n_{72} + n_{79} + n_{80})$$

$$= 2S + (n_5 + n_{13} + n_{21} + n_{29} + n_{37} + n_{54} + n_{62} + n_{70} + n_{78}) + (n_2 + n_3 + n_{10} + n_{11} + n_{11} + n_{12} + n_{13} + n_{18} + n_{19} + n_{20} + n_{21} + n_{27} + n_{29} + n_{36} + n_{74} + n_{75} + n_{76})$$

ところが、 $n_4 + n_{12} + n_{20} + n_{28} + n_{45} + n_{53} + n_{61} + n_{69} + n_{77} = K$

$$n_5 + n_{13} + n_{21} + n_{29} + n_{37} + n_{54} + n_{62} + n_{70} + n_{78} = K$$

$$2P + (n_6 + n_7 + n_8 + n_{46} + n_{53} + n_{55} + n_{61} + n_{62} + n_{63} + n_{64} + n_{69} + n_{70} + n_{71} + n_{71} + n_{72} + n_{79} + n_{80})$$

$$= 2S + (n_2 + n_3 + n_{10} + n_{11} + n_{11} + n_{12} + n_{13} + n_{18} + n_{19} + n_{20} + n_{21} + n_{27} + n_{29} + n_{36} + n_{74} + n_{75} + n_{76})$$

まだまだ追い出せる項がある。

$$2P + (n_{64} + n_{69} + n_{70} + n_{71} + n_{72}) + (n_6 + n_7 + n_8 + n_{46} + n_{53} + n_{55} + n_{61} + n_{62} + n_{63} + n_{71} + n_{79} + n_{80})$$

$$= 2S + (n_{10} + n_{11} + n_{12} + n_{13} + n_{18}) + (n_2 + n_3 + n_{11} + n_{19} + n_{20} + n_{21} + n_{27} + n_{29} + n_{36} + n_{74} + n_{75} + n_{76})$$

両辺に $(n_{65} + n_{66} + n_{67} + n_{68} + n_{14} + n_{15} + n_{16} + n_{17})$ を加える。

$$2P + (n_{64} + n_{65} + n_{66} + n_{67} + n_{68} + n_{69} + n_{70} + n_{71} + n_{72}) + (n_6 + n_7 + n_8 + n_{14} + n_{15} + n_{16} + n_{17} + n_{46} + n_{53} + n_{55} + n_{61} + n_{62} + n_{63} + n_{71} + n_{79} + n_{80})$$

$$= 2S + (n_{10} + n_{11} + n_{12} + n_{13} + n_{14} + n_{15} + n_{16} + n_{17} + n_{18}) + (n_2 + n_3 + n_{11} + n_{19} + n_{20} + n_{21} + n_{27} + n_{29} + n_{36} + n_{65} + n_{66} + n_{67} + n_{68} + n_{74} + n_{75} + n_{76})$$

ところが、 $n_{64} + n_{65} + n_{66} + n_{67} + n_{68} + n_{69} + n_{70} + n_{71} + n_{72} = K$

$$n_{10} + n_{11} + n_{12} + n_{13} + n_{14} + n_{15} + n_{16} + n_{17} + n_{18} = K$$

$$2P + (n_6 + n_7 + n_8 + n_{14} + n_{15} + n_{16} + n_{17} + n_{46} + n_{53} + n_{55} + n_{61} + n_{62} + n_{63} + n_{71} + n_{79} + n_{80})$$

$$= 2S + (n_2 + n_3 + n_{11} + n_{19} + n_{20} + n_{21} + n_{27} + n_{29} + n_{36} + n_{65} + n_{66} + n_{67} + n_{68} + n_{74} + n_{75} + n_{76})$$

もう1歩踏み込もう。

$$2P + (n_8 + n_{17} + n_{53} + n_{62} + n_{71} + n_{80}) + (n_6 + n_7 + n_{14} + n_{15} + n_{16} + n_{46} + n_{55} + n_{61} + n_{63} + n_{79})$$

$$= 2S + (n_2 + n_{11} + n_{20} + n_{29} + n_{65} + n_{74}) + (n_3 + n_{19} + n_{21} + n_{27} + n_{36} + n_{66} + n_{67} + n_{68} + n_{75} + n_{76})$$

両辺に $(n_{26} + n_{35} + n_{44} + n_{38} + n_{47} + n_{56})$ を加える。

$$2P + (n_8 + n_{17} + n_{26} + n_{35} + n_{44} + n_{53} + n_{62} + n_{71} + n_{80}) + (n_6 + n_7 + n_{14} + n_{15} + n_{16} + n_{38} + n_{46} + n_{47} + n_{55} + n_{56} + n_{61} + n_{63} + n_{79})$$

$$= 2S + (n_2 + n_{11} + n_{20} + n_{29} + n_{38} + n_{47} + n_{56} + n_{65} + n_{74}) + (n_3 + n_{19} + n_{21} + n_{26} + n_{27} + n_{35} + n_{36} + n_{44} + n_{66} + n_{67} + n_{68} + n_{75} + n_{76})$$

ところが、 $n_8 + n_{17} + n_{26} + n_{35} + n_{44} + n_{53} + n_{62} + n_{71} + n_{80} = K$

$$n_2 + n_{11} + n_{20} + n_{29} + n_{38} + n_{47} + n_{56} + n_{65} + n_{74} = K$$

$$2P + (n_6 + n_7 + n_{14} + n_{15} + n_{16} + n_{38} + n_{46} + n_{47} + n_{55} + n_{56} + n_{61} + n_{63} + n_{79})$$

$$= 2S + (n_3 + n_{19} + n_{21} + n_{26} + n_{27} + n_{35} + n_{36} + n_{44} + n_{66} + n_{67} + n_{68} + n_{75} + n_{76})$$

まだまだ。

$$2P + (n_6 + n_{14} + n_{38} + n_{46} + n_{63} + n_{79}) + (n_7 + n_{15} + n_{16} + n_{47} + n_{55} + n_{56} + n_{61})$$

$$= 2S + (n_3 + n_{19} + n_{36} + n_{44} + n_{68} + n_{76}) + (n_{21} + n_{26} + n_{27} + n_{35} + n_{66} + n_{67} + n_{75})$$

両辺に $(n_{22} + n_{30} + n_{71} + n_{11} + n_{52} + n_{60})$ を加える。

$$2P + (n_6 + n_{14} + n_{22} + n_{30} + n_{38} + n_{46} + n_{63} + n_{71} + n_{79}) + (n_7 + n_{11} + n_{15} + n_{16} + n_{47} + n_{52} + n_{55} + n_{56} + n_{60} + n_{61})$$

$$= 2S + (n_3 + n_{11} + n_{19} + n_{36} + n_{44} + n_{52} + n_{60} + n_{68} + n_{76}) + (n_{21} + n_{22} + n_{26} + n_{27} + n_{30} + n_{35} + n_{66} + n_{67} + n_{71} + n_{75})$$

ところが、 $n_6 + n_{14} + n_{22} + n_{30} + n_{38} + n_{46} + n_{63} + n_{71} + n_{79} = K$

$$n_3 + n_{11} + n_{19} + n_{36} + n_{44} + n_{52} + n_{60} + n_{68} + n_{76} = K$$

$$2P + (n_7 + n_{11} + n_{15} + n_{16} + n_{47} + n_{52} + n_{55} + n_{56} + n_{60} + n_{61})$$

$$=2S + (n_{21}+n_{22}+n_{26}+n_{27}+n_{30}+n_{35}+n_{66}+n_{67}+n_{71}+n_{75}) \dots (47)$$

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

(47)式の各項の空間的位置関係は、左図の通り。方陣内部に含まれる仮想七方陣の三角連結3数になりかけみたい配置だ。一角に欠員があり、 $3 \times 4 - 2 = 10$ 数で一組になっている。

対角の位置にある2組が等和になる。
対角を変えれば、

$$2Q + (n_{17}+n_{33}+n_{12}+n_{13}+n_{48}+n_{57}+n_{58}+n_{53}+n_{62}+n_{63})$$

$$= 2R + (n_{65}+n_{19}+n_{20}+n_{29}+n_{24}+n_{25}+n_{34}+n_{69}+n_{70}+n_{79})$$

.... (48)

が成り立つ。
「三角連結偶10数」といい、この(47),(48)式といい、九方陣なのに10項の和の式になってはいかにも効率が悪い気がするが、それでも何も無いよりはましだと考えることにしよう。
こういう関係式を用いて、1つでも独立変数を従属変数に変えると、効果が存外大きい。

5. 方程式をいろいろ捻くり回しても、上げる収穫は高々この程度のものである。そのものズバリ“P=S”と証明したいところだが、どうしてもそうならない。

($\dots + \dots + \dots + \dots$ $\dots + \dots + \dots$) という長い足し算の連鎖式が、なかなか短く切れない。得られる関係式は、大概が ($\dots + \dots + \dots + \dots$ $\dots + \dots + \dots$) + ($\dots + \dots + \dots + \dots$ $\dots + \dots + \dots$) というような和の形のままだ。こういう傾向は、六方陣の頃から始まっている。7次、8次、9次とますますひどくなる。どうも高次方陣の宿命らしい。

今回も、 $P+U=S+V$ や (42), (45), (47), (48)などの式は得られるが、それ以上は進まない。小生は、今までそれを自分の数学的非力のせいだとばかりずっと思って来た。自責の念にもかられていたが、もしかしたらそういうことではないのかも知れない。これは、高次方陣の1つの性質ではないか、と最近思うようになった(と言っても、決してロマンティストの開き直りではない)。

高次方陣で方程式が示すのは、1つの答ではなくて、1つの枠組みなのではないだろうか。「こういう枠組みさえ守ってくれば、汎魔方陣はいつでも作れるよ。その範囲内でなら自分でどんな条件を付けてもいいよ。」と言ってくれているのではないか。高次方陣は、本質的にそれほどまでに寛容なのではあるまいか。

大体が、81個の変数を扱うのに連立方程式が36本しかない。これでは、分が悪過ぎる。たったこれだけの条件では、せいぜいガイドラインか、あるいはフレームを示せるだけではないのか。足し算の長い連鎖式は、そのことを言っているのではないのか。

それに方陣の構成は、連立方程式だけで全てが決まるわけではない。素材の自然数によっても決まる。配列という構造によっても決まる。自分が与えた注文によっても左右される。だからこそ、何通りもの個性が生じる。何千万個もの解が生じる。

我々の目的は、あくまで魔方陣の構成にある。必ずしも数式の厳密解・唯一解を求めているわけではない。

厳密解があるにこしたことはないが、実際のところ我々の関心は魔方陣が出来るか出来ないのか、その1点に絞られている。数学的理想主義より目的合理主義なのである。

$P+A=S+B$, $P+M=S+N$, $P+U=S+V$, \dots という形の式は、いずれも $P=S$ とともに成り立つ条件を示している。 $A=B$, $M=N$, $U=V$ であれば、 $P=S$ が成り立つ。 $P=S$ であれば、 $A=B$, $M=N$, $U=V$ が成り立つ。

($P=S$) & ($A=B$)であっても良い。それで十分である。何の文句も無い。

$P=S$ であって $A \neq B$ ということはない。 $P \neq S$ であって $A=B$ ということもない。

($P \neq S$) & ($A \neq B$) でかつ $P + A = S + B$ という場合は、あり得る。しかし、そういうケースを一々調べていた日にゃ、何も出来ない内に陽が暮れる。解が何千万個もある場合には、全ての場合を調べ尽くすのは現実的に不可能である。

時間制限は、現実問題として存在する。

結局、選択の問題だ。第1に ($P = S$) & ($A = B$)；第2に ($P \neq S$) & ($A \neq B$) でかつ $P + A = S + B$ ；そして第3に、単に $P + A = S + B$

この際、第一の選択をしよう。その方が勝負が速い。それに何と言っても「美しい」。

そこで、我々は今後 $P = S$, $Q = R$ とともに以下の等式群をすべて仮定する。

$$n_5 + n_{13} + n_{14} + n_{21} + n_{22} + n_{23} + n_{29} + n_{30} + n_{31} + n_{32} \\ = n_{42} + n_{43} + n_{44} + n_{45} + n_{51} + n_{52} + n_{53} + n_{60} + n_{61} + n_{69} \dots (U = V)$$

$$n_7 + n_{11} + n_{15} + n_{16} + n_{47} + n_{52} + n_{55} + n_{56} + n_{60} + n_{61} \\ = n_{21} + n_{22} + n_{26} + n_{27} + n_{30} + n_{35} + n_{66} + n_{67} + n_{71} + n_{75} \dots (47)'$$

$$n_{17} + n_3 + n_{12} + n_{13} + n_{48} + n_{57} + n_{58} + n_{53} + n_{62} + n_{63} \\ = n_{65} + n_{19} + n_{20} + n_{29} + n_{24} + n_{25} + n_{34} + n_{69} + n_{70} + n_{79} \dots (48)'$$

また汎魔方陣だから、 $P = S$ や $Q = R$ も拡張空間の中で言う。

$$n_2 + n_3 + n_4 + n_5 + n_{11} + n_{12} + n_{13} + n_{20} + n_{21} + n_{29} \\ = n_{46} + n_{63} + n_{55} + n_{71} + n_{72} + n_{64} + n_{79} + n_{80} + n_{81} + n_{73} \dots (P_2 = S_2)$$

$$n_{10} + n_{11} + n_{12} + n_{13} + n_{19} + n_{20} + n_{21} + n_{28} + n_{29} + n_{37} \\ = n_{63} + n_{71} + n_{72} + n_{79} + n_{80} + n_{81} + n_6 + n_7 + n_8 + n_9 \dots (P_{10} = S_{10})$$

$$n_{11} + n_{12} + n_{13} + n_{14} + n_{20} + n_{21} + n_{22} + n_{29} + n_{30} + n_{38} \\ = n_{55} + n_{72} + n_{64} + n_{80} + n_{81} + n_{73} + n_7 + n_8 + n_9 + n_1 \dots (P_{11} = S_{11})$$

.....

などの等式が成り立つものとする。それは (47)' , (48)' 式でも同様である。百本も 2 百本も立つ。

これだけの条件を全て仮定しても、汎九方陣の枠組みの中におさまる。だから、方陣の構成は可能のはずだ。式が余計にある分、探索・計数の時間が短縮されると期待できる。

6. 加えて、対称方陣の性質も併せもつ。P, S 内部の各項は各々対称の位置にあるので、

$$P + S = (n_1 + n_2 + n_3 + n_4 + n_{10} + n_{11} + n_{12} + n_{19} + n_{20} + n_{28}) \\ + (n_{54} + n_{62} + n_{63} + n_{70} + n_{71} + n_{72} + n_{78} + n_{79} + n_{80} + n_{81}) \\ = (n_1 + n_{81}) + (n_2 + n_{80}) + (n_3 + n_{79}) + (n_4 + n_{78}) + (n_{10} + n_{72}) \\ + (n_{11} + n_{71}) + (n_{12} + n_{70}) + (n_{19} + n_{63}) + (n_{20} + n_{62}) + (n_{28} + n_{54}) \\ = 82 + 82 + 82 + 82 + 82 + 82 + 82 + 82 + 82 + 82 = 820$$

汎九方陣でかつ対称九方陣である場合には、

$$P = S, \quad P + S = 820 \text{ がともに成り立つ。}$$

ゆえに、 $P = S = 410$

同様に、 $Q = R = 410$

$P = Q = R = S = 410$ ならば、かなり強力な式になる。同様にして、

$$n_7 + n_{11} + n_{15} + n_{16} + n_{47} + n_{52} + n_{55} + n_{56} + n_{60} + n_{61} \\ = n_{21} + n_{22} + n_{26} + n_{27} + n_{30} + n_{35} + n_{66} + n_{67} + n_{71} + n_{75} = 410 \dots (47)''$$

$$n_{17} + n_3 + n_{12} + n_{13} + n_{48} + n_{57} + n_{58} + n_{53} + n_{62} + n_{63} \\ = n_{65} + n_{19} + n_{20} + n_{29} + n_{24} + n_{25} + n_{34} + n_{69} + n_{70} + n_{79} = 410 \dots (48)''$$

も成り立つ。だが、

$$n_5 + n_{13} + n_{14} + n_{21} + n_{22} + n_{23} + n_{29} + n_{30} + n_{31} + n_{32} \\ = n_{42} + n_{43} + n_{44} + n_{45} + n_{51} + n_{52} + n_{53} + n_{60} + n_{61} + n_{69} \dots (U = V) \text{ については、}$$

対応する各項が対称の位置にないので、必ずしも 410 になるとは限らない。

汎九方陣で成り立つ式でも、対称の位置に無ければ、必ずしも 410 になるとは限らない。

.....

(初稿 : December 13, 2000; MacOS9 & MWCW による作表・計数 : 摂田 寛二 (Kanji Setsuda))

第2節：対称汎九方陣の具体的構成（改訂版）

1. 不等式は、 $n1 < n81$ と $n1 < n9 < n73$ の2本しか立てない。汎魔方陣でかつ対称方陣だから、厳格にし過ぎてどちらの性質も抑圧してしまわないように配慮する。

$n1 = 1$ の「標準形」も想定しない。

万全の準備をした上で、パソコン用のプログラムを組み、解を探索・構成する。

.....

ところが、ところがである。最初の解がなかなか出て来ない。七方陣の時とは様子がだいぶ違う。やはり $81 \times 80 \times 79 \times \dots$ 通りを調べるのは、最新のスーパー・パソコンでも決して容易ではないようだ。

研究上、「たとえ1個でも良い。そのタイプの解の実例が欲しい」と言う場合が、えてしてあるものだ。何かの存在そのものを確認したい時などである。そんなこんなで急いでいる時に正攻法で行くのは、かえって馬鹿げている。たとえ裏技でも良い。速く解に出逢える方法は無いだろうか。

2. 近道が無いわけではない。余り好ましいやり方ではないかも知れないが、先人の努力を継承するのである。良く言えばそうだが、悪く言えば、途中まで先人の解を横着に拝借するのである。その方法を内緒で教えよう（この個所は、2000年稿のまま）。

(1) 例えば、鈴木教授のデータベースの中から、汎九方陣の解の1例を借りる。

[先人の解]	[派生形を読み取る]
41 30 52 61 31 1 21 81 51	1 21 81 51 41 30 52 61 31 1
11 9 24 77 66 44 16 67 55	44 16 67 55 11 9 24 77 66 44
71 40 29 47 63 14 5 20 80	14 5 20 80 71 40 29 47 63 14
59 10 8 25 76 65 39 18 69	→ 65 39 18 69 59 10 8 25 76 65
75 54 42 33 46 62 34 4 19	62 34 4 19 75 54 42 33 46 62
70 58 28 3 27 78 50 38 17	78 50 38 17 70 58 28 3 27 78
23 74 53 43 13 64 57 36 6	64 57 36 6 23 74 53 43 13 64
12 72 60 32 2 26 79 49 37	26 79 49 37 12 72 60 32 2 26
7 22 73 48 45 15 68 56 35	15 68 56 35 7 22 73 48 45 15
	1 21 81 51 41 30 52 61 31 1

(2) これを汎魔方陣の約束に従って拡張空間の中に置き、移動させて上図右のようにする。そして好きな方角から1つの派生形を読み取って清書し、サンプルとする。

[サンプル]	[拝借する部分]
1 44 14 65 62 78 64 26 15	1 44 14 65 62 78 64 26 15
31 66 63 76 46 27 13 2 45	31 * * * * * * * *
61 77 47 25 33 3 43 32 48	61 * * * * * * * *
52 24 29 8 42 28 53 60 73	→ 52 * * * * * * * *
30 9 40 10 54 58 74 72 22	* * * * * * * *
41 11 71 59 75 70 23 12 7	* * * * * * * *
51 55 80 69 19 * * * *	* * * * * * * *
81 67 20 * * * * * *	* * * * * * * *
21 16 * * * * * *	* * * * * * * *

(3) そして上図右側のように12個の数値を拝借し、初期値として自分のプログラムで利用させていただくのである。

対称方陣の補数配置から、下側の12個の数値が一義的に決まるので、24個の初期値を予め決めておいてスタートすることになる。残りの57個を心配すれば良い。補数ペアを考慮すれば、28組の配置を調べれば良いと言うわけだ。

```

1 44 14 65 62 78 64 26 15
31 * * * * * * * *
61 * * * * * * * *
52 * * * * * * * *
* * * * 41 * * * *
* * * * * * * * 30
* * * * * * * * 21
* * * * * * * * 51
67 56 18 4 20 17 68 38 81

```

これならば、少しは素早く解にたどり着けるはずだ。ただし、数値の幅は2～80 と広い。80×79×78×……通りを調べるのであるから、必ずしもチョー簡単、チョー高速になるわけではない。

このような借用は、必ずしも倫理的とはいえないが、目的が良ければ、決してアンフェアではないと思う（アカデミックな研究の役にさえ立てば、鈴木教授も許して下さると思う。出典を一言断りさえすれば……）。

例の選び方が悪いと、正解にはいつまでもたどり着けない。しかし、汎九方陣を例に採れば、答は大概近傍にあるものだ。その成功例を下に示す。

*** 対称汎九方陣の解の実例 ***

#1

```

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

```

#2

```

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

```

#3

```

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

```

#4

```

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

```

#5

```

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

```

#6

```

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

```

#7

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

#8

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

#9

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

#10

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

#11

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

#12

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

..... (と続く)

3. 正攻法で求めて素早く高次方陣の解を得たければ、最初から珍しい解を探すしかない。
 例えば、**多重三方陣内包型の対称汎九方陣**を作るとかしよう。早速にも設計に入る。

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

<基本図>

* 多重三方陣の各行列の定和を定義する連立方程式: $S=123$ *

Block #0:

$$\begin{aligned} n1+n2+n3=S & \dots rw00; & n10+n11+n12=S & \dots rw01; & n19+n20+n21=S & \dots rw02; \\ n1+n10+n19=S & \dots cl00; & n2+n11+n20=S & \dots cl01; & n3+n12+n21=S & \dots cl02; \end{aligned}$$

Block #1:

$$\begin{aligned} n4+n5+n6=S & \dots rw10; & n13+n14+n15=S & \dots rw11; & n22+n23+n24=S & \dots rw12; \\ n4+n13+n22=S & \dots cl10; & n5+n14+n23=S & \dots cl11; & n6+n15+n24=S & \dots cl12; \end{aligned}$$

Block #2:

$$\begin{aligned} n7+n8+n9=S & \dots rw20; & n16+n17+n18=S & \dots rw21; & n25+n26+n27=S & \dots rw22; \\ n7+n16+n25=S & \dots cl20; & n8+n17+n26=S & \dots cl21; & n9+n18+n27=S & \dots cl22; \end{aligned}$$

Block #3:

$$\begin{aligned} n28+n29+n30=S & \dots rw30; & n37+n38+n39=S & \dots rw31; & n46+n47+n48=S & \dots rw32; \\ n28+n37+n46=S & \dots cl30; & n29+n38+n47=S & \dots cl31; & n30+n39+n48=S & \dots cl32; \end{aligned}$$

Block #4:

$$\begin{aligned} n31+n32+n33=S & \dots rw40; & n40+n41+n42=S & \dots rw41; & n49+n50+n51=S & \dots rw42; \\ n31+n40+n49=S & \dots cl40; & n32+n41+n50=S & \dots cl41; & n33+n42+n51=S & \dots cl42; \end{aligned}$$

Block #5:

$$\begin{aligned} n34+n35+n36=S & \dots rw50; & n43+n44+n45=S & \dots rw51; & n52+n53+n54=S & \dots rw52; \\ n34+n43+n52=S & \dots cl50; & n35+n44+n53=S & \dots cl51; & n36+n45+n54=S & \dots cl52; \end{aligned}$$

Block #6:

$$\begin{aligned} n55+n56+n57=S & \dots rw60; & n64+n65+n66=S & \dots rw61; & n73+n74+n75=S & \dots rw62; \\ n55+n64+n73=S & \dots cl60; & n56+n65+n74=S & \dots cl61; & n57+n66+n75=S & \dots cl62; \end{aligned}$$

Block #7:

$$\begin{aligned} n58+n59+n60=S & \dots rw70; & n67+n68+n69=S & \dots rw71; & n76+n77+n78=S & \dots rw72; \\ n58+n67+n76=S & \dots cl70; & n59+n68+n77=S & \dots cl71; & n60+n69+n78=S & \dots cl72; \end{aligned}$$

Block #8:

$$\begin{aligned} n61+n62+n63=S & \dots rw80; & n70+n71+n72=S & \dots rw81; & n79+n80+n81=S & \dots rw82; \\ n61+n70+n79=S & \dots cl80; & n62+n71+n80=S & \dots cl81; & n63+n72+n81=S & \dots cl82; \end{aligned}$$

* 補数の対称配置と定和を定義する式: $C=82$ *

$$\begin{aligned} n1+n81=C; & n2+n80=C; & n3+n79=C; & n4+n78=C; \\ n5+n77=C; & n6+n76=C; & n7+n75=C; & n8+n74=C; \\ n9+n73=C; & n10+n72=C; & n11+n71=C; & n12+n70=C; \\ n13+n69=C; & n14+n68=C; & n15+n67=C; & n16+n66=C; \\ n17+n65=C; & n18+n64=C; & n19+n63=C; & n20+n62=C; \\ n21+n61=C; & n22+n60=C; & n23+n59=C; & n24+n58=C; \\ n25+n57=C; & n26+n56=C; & n27+n55=C; & n28+n54=C; \\ n29+n53=C; & n30+n52=C; & n31+n51=C; & n32+n50=C; \\ n33+n49=C; & n34+n48=C; & n35+n47=C; & n36+n46=C; \\ n37+n45=C; & n38+n44=C; & n39+n43=C; & n40+n42=C; & n41+n41=C; \dots ssc \end{aligned}$$

* 汎対角線の定和を定義する連立方程式: $CC=369$ *

$$\begin{aligned} n1+n11+n21+n31+n41+n51+n61+n71+n81=CC & \dots pd1; \\ n2+n12+n22+n32+n42+n52+n62+n72+n73=CC & \dots pd2; \\ n3+n13+n23+n33+n43+n53+n63+n64+n74=CC & \dots pd3; \\ n4+n14+n24+n34+n44+n54+n55+n65+n75=CC & \dots pd4; \\ n5+n15+n25+n35+n45+n46+n56+n66+n76=CC & \dots pd5; \\ n6+n16+n26+n36+n37+n47+n57+n67+n77=CC & \dots pd6; \\ n7+n17+n27+n28+n38+n48+n58+n68+n78=CC & \dots pd7; \\ n8+n18+n19+n29+n39+n49+n59+n69+n79=CC & \dots pd8; \\ n9+n10+n20+n30+n40+n50+n60+n70+n80=CC & \dots pd9; \\ n1+n18+n26+n34+n42+n50+n58+n66+n74=CC & \dots pb1; \\ n2+n10+n27+n35+n43+n51+n59+n67+n75=CC & \dots pb2; \\ n3+n11+n19+n36+n44+n52+n60+n68+n76=CC & \dots pb3; \\ n4+n12+n20+n28+n45+n53+n61+n69+n77=CC & \dots pb4; \\ n5+n13+n21+n29+n37+n54+n62+n70+n78=CC & \dots pb5; \\ n6+n14+n22+n30+n38+n46+n63+n71+n79=CC & \dots pb6; \\ n7+n15+n23+n31+n39+n47+n55+n72+n80=CC & \dots pb7; \\ n8+n16+n24+n32+n40+n48+n56+n64+n81=CC & \dots pb8; \\ n9+n17+n25+n33+n41+n49+n57+n65+n73=CC & \dots pb9; \end{aligned}$$

* リスト整形出力用の不等式 *

$$n1 < n9, n1 < n73, n1 < n81, n2 > n10$$

最初の行列定和の定義式が九方陣らしくないが、まず 3×3 のブロックに分けて各個別に 3 行 3 列の定和を定義している。式がたくさん出来たので、求めやすいはずだ。

補数に関する定義式と汎対角線定和の連立方程式、それに不等式は、九方陣本来のものを使う。我々は、あくまで対称汎九方陣を作りたいからである。

その方針のもとに実際にプログラムを組んで、いざ計算させてみると、これがなかなか答が出て来ない。最初は自分のプログラムにミスがあるのかと、チェックにチェックを重ねたが、そういうことではないらしい。やはり、高次方陣は手強い。何時間も待つと、ようやく答えを数個出して来た。全体像は、とても掴めない。実用的なスピードではないのだ。

もう少し厳しい注文を付けて、対称汎九方陣の中の特別の珍品を探ししかないようだ。

例えば、こんなのはどうであろうか。

2本の主対角線上の各9個の数値に特別の条件を付ける。共通の n_{41} は、対称方陣の中心点なので、既に 41 が来ることが決まっているから、残りの各8個の数値に注文を付ける。

$$\{n1, n11, n21, n31, n51, n61, n71, n81\} = \{1, 11, 21, 31, 51, 61, 71, 81\}$$

$$\{n9, n17, n25, n33, n49, n57, n65, n73\} = \{9, 17, 25, 33, 49, 57, 65, 73\}$$

と制限してはどうだろうか。ポジションについては、順不同とし、同じ対角線上でならどこにでも行けるとする。

1	*	*	*	*	*	*	*	9
*	11	*	*	*	*	*	*	17
*	*	21	*	*	*	25	*	*
*	*	*	31	*	33	*	*	*
*	*	*	*	41	*	*	*	*
*	*	*	49	*	51	*	*	*
*	*	57	*	*	*	61	*	*
*	65	*	*	*	*	*	71	*
73	*	*	*	*	*	*	*	81

ただし、 $\{n1, n81\}$, $\{n11, n71\}$, $\{n21, n61\}$, $\{n31, n51\}$, $\{n9, n73\}$, $\{n17, n65\}$, $\{n25, n57\}$, $\{n33, n49\}$ のペアは、相互に点対称の配置にあるから、82の補数を成す。従って、ペアの一方の値が決まれば、他方の値が自動的に決まる。実質自分で決めなければならないのは、8個に過ぎない。

かなり厳しい条件となるが、こうすれば、解の総数を減らして、素早く数え切れるのではないか。相当の珍品を作れそうなので、早速にも実験してみよう。

4. 次ページに実験結果を紹介する。標準解のリスト(抜粋)とそれを求めるために用いたプログラムの全リストを掲載することにしよう。

かなり特殊な解集合ではあるが、それでもこれは対称汎九方陣である。珍品である。何らかの「基本解」になっている可能性がある。

しかも、あっという間に解を全数リスト・アップしてくれた。

自分では、結構満足している成果だ。

ここではまだ三進法も「新オイラー法」も四次元超立体三方陣の展開図も使っていない。初稿の背景を考慮して、そうした。当時は、まだ思いつきもしていなかったのだから。

(初稿 : December 13, 2000; MacOS9 & MWCW による作表・計数;

改訂新稿 : August 25, 2008; MacOSX & Xcode3.0 による計数;

清書稿 : July 29, 2017; macOS Sierra 10.12.6 & Xcode 7.3 による計数;)

撰田 寛二(Kanji Setsuda); E-Mail Address: jag12001@nifty.com

* 多重三方陣型対称汎九方陣を作る実験 by 摂田 寛二(Kanji Setsuda) *

* 標準解のリスト (主対角線の内容に特別の注文を付けた : チェックサム・モニター付き) *

1# Row C1m\Pd1/Pd2										2# Row C1m\Pd1/Pd2									
1	68	54	46	5	72	74	40	9	369 369\369/369	1	68	54	30	13	80	66	32	25	369 369\369/369
44	21	58	62	39	22	16	57	50	369 369\369/369	44	21	58	70	47	6	8	73	42	369 369\369/369
78	34	11	15	79	29	33	26	64	369 369\369/369	78	34	11	23	63	37	49	18	56	369 369\369/369
6	70	47	31	27	65	69	52	2	369 369\369/369	22	62	39	31	27	65	77	36	10	369 369\369/369
37	23	63	75	41	7	19	59	45	369 369\369/369	29	15	79	75	41	7	3	67	53	369 369\369/369
80	30	13	17	55	51	35	12	76	369 369\369/369	72	46	5	17	55	51	43	20	60	369 369\369/369
18	56	49	53	3	67	71	48	4	369 369\369/369	26	64	33	45	19	59	71	48	4	369 369\369/369
32	25	66	60	43	20	24	61	38	369 369\369/369	40	9	74	76	35	12	24	61	38	369 369\369/369
73	42	8	10	77	36	28	14	81	369 369\369/369	57	50	16	2	69	52	28	14	81	369 369\369/369
3# Row C1m\Pd1/Pd2										4# Row C1m\Pd1/Pd2									
1	68	54	22	62	39	26	64	33	369 369\369/369	1	68	54	46	62	15	74	16	33	369 369\369/369
44	21	58	29	15	79	40	9	74	369 369\369/369	44	21	58	5	39	79	40	57	26	369 369\369/369
78	34	11	72	46	5	57	50	16	369 369\369/369	78	34	11	72	22	29	9	50	64	369 369\369/369
30	13	80	31	27	65	45	19	59	369 369\369/369	6	37	80	31	27	65	69	19	35	369 369\369/369
70	47	6	75	41	7	76	35	12	369 369\369/369	70	23	30	75	41	7	52	59	12	369 369\369/369
23	63	37	17	55	51	2	69	52	369 369\369/369	47	63	13	17	55	51	2	45	76	369 369\369/369
66	32	25	77	36	10	71	48	4	369 369\369/369	18	32	73	53	60	10	71	48	4	369 369\369/369
8	73	42	3	67	53	24	61	38	369 369\369/369	56	25	42	3	43	77	24	61	38	369 369\369/369
49	18	56	43	20	60	28	14	81	369 369\369/369	49	66	8	67	20	36	28	14	81	369 369\369/369
5# Row C1m\Pd1/Pd2										6# Row C1m\Pd1/Pd2									
1	68	54	6	70	47	18	56	49	369 369\369/369	1	68	54	30	70	23	66	8	49	369 369\369/369
44	21	58	37	23	63	32	25	66	369 369\369/369	44	21	58	13	47	63	32	73	18	369 369\369/369
78	34	11	80	30	13	73	42	8	369 369\369/369	78	34	11	80	6	37	25	42	56	369 369\369/369
46	5	72	31	27	65	53	3	67	369 369\369/369	22	29	72	31	27	65	77	3	43	369 369\369/369
62	39	22	75	41	7	60	43	20	369 369\369/369	62	15	46	75	41	7	36	67	20	369 369\369/369
15	79	29	17	55	51	10	77	36	369 369\369/369	39	79	5	17	55	51	10	53	60	369 369\369/369
74	40	9	69	52	2	71	48	4	369 369\369/369	26	40	57	45	76	2	71	48	4	369 369\369/369
16	57	50	19	59	45	24	61	38	369 369\369/369	64	9	50	19	35	69	24	61	38	369 369\369/369
33	26	64	35	12	76	28	14	81	369 369\369/369	33	74	16	59	12	52	28	14	81	369 369\369/369
7# Row C1m\Pd1/Pd2										8# Row C1m\Pd1/Pd2									
1	68	54	22	29	72	26	40	57	369 369\369/369	1	68	54	6	37	80	18	32	73	369 369\369/369
44	21	58	62	15	46	64	9	50	369 369\369/369	44	21	58	70	23	30	56	25	42	369 369\369/369
78	34	11	39	79	5	33	74	16	369 369\369/369	78	34	11	47	63	13	49	66	8	369 369\369/369
30	70	23	31	27	65	45	76	2	369 369\369/369	46	62	15	31	27	65	53	60	10	369 369\369/369
13	47	63	75	41	7	19	35	69	369 369\369/369	5	39	79	75	41	7	3	43	77	369 369\369/369
80	6	37	17	55	51	59	12	52	369 369\369/369	72	22	29	17	55	51	67	20	36	369 369\369/369
66	8	49	77	3	43	71	48	4	369 369\369/369	74	16	33	69	19	35	71	48	4	369 369\369/369
32	73	18	36	67	20	24	61	38	369 369\369/369	40	57	26	52	59	12	24	61	38	369 369\369/369
25	42	56	10	53	60	28	14	81	369 369\369/369	9	50	64	2	45	76	28	14	81	369 369\369/369
9# Row C1m\Pd1/Pd2										10# Row C1m\Pd1/Pd2									
1	68	54	46	5	72	64	50	9	369 369\369/369	1	68	54	30	13	80	56	42	25	369 369\369/369
44	21	58	62	39	22	26	57	40	369 369\369/369	44	21	58	70	47	6	18	73	32	369 369\369/369
78	34	11	15	79	29	33	16	74	369 369\369/369	78	34	11	23	63	37	49	8	66	369 369\369/369
6	70	47	51	7	65	69	52	2	369 369\369/369	22	62	39	51	7	65	77	36	10	369 369\369/369
37	23	63	55	41	27	19	59	45	369 369\369/369	29	15	79	55	41	27	3	67	53	369 369\369/369
80	30	13	17	75	31	35	12	76	369 369\369/369	72	46	5	17	75	31	43	20	60	369 369\369/369
8	66	49	53	3	67	71	48	4	369 369\369/369	16	74	33	45	19	59	71	48	4	369 369\369/369
42	25	56	60	43	20	24	61	38	369 369\369/369	50	9	64	76	35	12	24	61	38	369 369\369/369
73	32	18	10	77	36	28	14	81	369 369\369/369	57	40	26	2	69	52	28	14	81	369 369\369/369

11# Row C1m\Pd1/Pd2													12# Row C1m\Pd1/Pd2												
1	68	54	22	62	39	16	74	33	369	369	369	369	1	68	54	46	62	15	64	26	33	369	369	369	369
44	21	58	29	15	79	50	9	64	369	369	369	369	44	21	58	5	39	79	50	57	16	369	369	369	369
78	34	11	72	46	5	57	40	26	369	369	369	369	78	34	11	72	22	29	9	40	74	369	369	369	369
30	13	80	51	7	65	45	19	59	369	369	369	369	6	37	80	51	7	65	69	19	35	369	369	369	369
70	47	6	55	41	27	76	35	12	369	369	369	369	70	23	30	55	41	27	52	59	12	369	369	369	369
23	63	37	17	75	31	2	69	52	369	369	369	369	47	63	13	17	75	31	2	45	76	369	369	369	369
56	42	25	77	36	10	71	48	4	369	369	369	369	8	42	73	53	60	10	71	48	4	369	369	369	369
18	73	32	3	67	53	24	61	38	369	369	369	369	66	25	32	3	43	77	24	61	38	369	369	369	369
49	8	66	43	20	60	28	14	81	369	369	369	369	49	56	18	67	20	36	28	14	81	369	369	369	369
13# Row C1m\Pd1/Pd2													14# Row C1m\Pd1/Pd2												
1	68	54	6	70	47	8	66	49	369	369	369	369	1	68	54	30	70	23	56	18	49	369	369	369	369
44	21	58	37	23	63	42	25	56	369	369	369	369	44	21	58	13	47	63	42	73	8	369	369	369	369
78	34	11	80	30	13	73	32	18	369	369	369	369	78	34	11	80	6	37	25	32	66	369	369	369	369
46	5	72	51	7	65	53	3	67	369	369	369	369	22	29	72	51	7	65	77	3	43	369	369	369	369
62	39	22	55	41	27	60	43	20	369	369	369	369	62	15	46	55	41	27	36	67	20	369	369	369	369
15	79	29	17	75	31	10	77	36	369	369	369	369	39	79	5	17	75	31	10	53	60	369	369	369	369
64	50	9	69	52	2	71	48	4	369	369	369	369	16	50	57	45	76	2	71	48	4	369	369	369	369
26	57	40	19	59	45	24	61	38	369	369	369	369	74	9	40	19	35	69	24	61	38	369	369	369	369
33	16	74	35	12	76	28	14	81	369	369	369	369	33	64	26	59	12	52	28	14	81	369	369	369	369
15# Row C1m\Pd1/Pd2													16# Row C1m\Pd1/Pd2												
1	68	54	22	29	72	16	50	57	369	369	369	369	1	68	54	6	37	80	8	42	73	369	369	369	369
44	21	58	62	15	46	74	9	40	369	369	369	369	44	21	58	70	23	30	66	25	32	369	369	369	369
78	34	11	39	79	5	33	64	26	369	369	369	369	78	34	11	47	63	13	49	56	18	369	369	369	369
30	70	23	51	7	65	45	76	2	369	369	369	369	46	62	15	51	7	65	53	60	10	369	369	369	369
13	47	63	55	41	27	19	35	69	369	369	369	369	5	39	79	55	41	27	3	43	77	369	369	369	369
80	6	37	17	75	31	59	12	52	369	369	369	369	72	22	29	17	75	31	67	20	36	369	369	369	369
56	18	49	77	3	43	71	48	4	369	369	369	369	64	26	3369	19	35	71	48	4	369	369	369	369	
42	73	8	36	67	20	24	61	38	369	369	369	369	50	57	16	52	59	12	24	61	38	369	369	369	369
25	32	66	10	53	60	28	14	81	369	369	369	369	9	40	74	2	45	76	28	14	81	369	369	369	369
17# Row C1m\Pd1/Pd2													18# Row C1m\Pd1/Pd2												
1	50	72	64	5	54	76	38	9	369	369	369	369	1	50	72	56	24	43	60	46	17	369	369	369	369
42	61	20	24	43	56	30	25	68	369	369	369	369	42	61	20	13	35	75	38	9	76	369	369	369	369
80	12	31	35	75	13	17	60	46	369	369	369	369	80	12	31	54	64	5	25	68	30	369	369	369	369
8	48	67	11	63	49	53	66	4	369	369	369	369	16	29	78	11	63	49	45	55	23	369	369	369	369
37	59	27	79	41	3	55	23	45	369	369	369	369	48	67	8	79	41	3	74	15	34	369	369	369	369
78	16	29	33	19	71	15	34	74	369	369	369	369	59	27	37	33	19	71	4	53	66	369	369	369	369
36	22	65	69	7	47	51	70	2	369	369	369	369	52	14	57	77	18	28	51	70	2	369	369	369	369
14	57	52	26	39	58	62	21	40	369	369	369	369	6	73	44	7	47	69	62	21	40	369	369	369	369
73	44	6	28	77	18	10	32	81	369	369	369	369	65	36	22	39	58	26	10	32	81	369	369	369	369
19# Row C1m\Pd1/Pd2													20# Row C1m\Pd1/Pd2												
1	50	72	64	24	35	76	30	17	369	369	369	369	1	50	72	56	13	54	60	38	25	369	369	369	369
42	61	20	5	43	75	38	25	60	369	369	369	369	42	61	20	24	35	64	46	9	68	369	369	369	369
80	12	31	54	56	13	9	68	46	369	369	369	369	80	12	31	43	75	5	17	76	30	369	369	369	369
8	37	78	11	63	49	53	55	15	369	369	369	369	16	48	59	11	63	49	45	74	4	369	369	369	369
48	59	16	79	41	3	66	23	34	369	369	369	369	29	67	27	79	41	3	55	15	53	369	369	369	369
67	27	29	33	19	71	4	45	74	369	369	369	369	78	8	37	33	19	71	23	34	66	369	369	369	369
36	14	73	69	26	28	51	70	2	369	369	369	369	52	6	65	77	7	39	51	70	2	369	369	369	369
22	57	44	7	39	77	62	21	40	369	369	369	369	14	73	36	18	47	58	62	21	40	369	369	369	369
65	52	6	47	58	18	10	32	81	369	369	369	369	57	44	22	28	69	26	10	32	81	369	369	369	369

** Special type of Simultaneous Magic Squares of Order 9:
 Self-complementary, Pan-diagonal and Multiple 3*3*9;
 [Standard Solutions with precise Check-sum Monitors] **

21# Row C1m\Pd1/Pd2													22# Row C1m\Pd1/Pd2												
1	50	72	16	29	78	52	14	57	369	369	369	369	1	50	72	8	48	67	36	22	65	369	369	369	369
42	61	20	48	67	8	6	73	44	369	369	369	369	42	61	20	37	59	27	14	57	52	369	369	369	369
80	12	31	59	27	37	65	36	22	369	369	369	369	80	12	31	78	16	29	73	44	6	369	369	369	369
56	24	43	11	63	49	77	18	28	369	369	369	369	64	5	54	11	63	49	69	7	47	369	369	369	369
13	35	75	79	41	3	7	47	69	369	369	369	369	24	43	56	79	41	3	26	39	58	369	369	369	369
54	64	5	33	19	71	39	58	26	369	369	369	369	35	75	13	33	19	71	28	77	18	369	369	369	369
60	46	17	45	55	23	51	70	2	369	369	369	369	76	38	9	53	66	4	51	70	2	369	369	369	369
38	9	76	74	15	34	62	21	40	369	369	369	369	30	25	68	55	23	45	62	21	40	369	369	369	369
25	68	30	4	53	66	10	32	81	369	369	369	369	17	60	46	15	34	74	10	32	81	369	369	369	369
23# Row C1m\Pd1/Pd2													24# Row C1m\Pd1/Pd2												
1	50	72	16	48	59	52	6	65	369	369	369	369	1	50	72	8	37	78	36	14	73	369	369	369	369
42	61	20	29	67	27	14	73	36	369	369	369	369	42	61	20	48	59	16	22	57	44	369	369	369	369
80	12	31	78	8	37	57	44	22	369	369	369	369	80	12	31	67	27	29	65	52	6	369	369	369	369
56	13	54	11	63	49	77	7	39	369	369	369	369	64	24	35	11	63	49	69	26	28	369	369	369	369
24	35	64	79	41	3	18	47	58	369	369	369	369	5	43	75	79	41	3	7	39	77	369	369	369	369
43	75	5	33	19	71	28	69	26	369	369	369	369	54	56	13	33	19	71	47	58	18	369	369	369	369
60	38	25	45	74	4	51	70	2	369	369	369	369	76	30	17	53	55	15	51	70	2	369	369	369	369
46	9	68	55	15	53	62	21	40	369	369	369	369	38	25	60	66	23	34	62	21	40	369	369	369	369
17	76	30	23	34	66	10	32	81	369	369	369	369	9	68	46	4	45	74	10	32	81	369	369	369	369
25# Row C1m\Pd1/Pd2													26# Row C1m\Pd1/Pd2												
1	50	72	64	5	54	46	68	9	369	369	369	369	1	50	72	56	24	43	30	76	17	369	369	369	369
42	61	20	24	43	56	60	25	38	369	369	369	369	42	61	20	13	35	75	68	9	46	369	369	369	369
80	12	31	35	75	13	17	30	76	369	369	369	369	80	12	31	54	64	5	25	38	60	369	369	369	369
8	48	67	71	3	49	53	66	4	369	369	369	369	16	29	78	71	3	49	45	55	23	369	369	369	369
37	59	27	19	41	63	55	23	45	369	369	369	369	48	67	8	19	41	63	74	15	34	369	369	369	369
78	16	29	33	79	11	15	34	74	369	369	369	369	59	27	37	33	79	11	4	53	66	369	369	369	369
6	52	65	69	7	47	51	70	2	369	369	369	369	22	44	57	77	18	28	51	70	2	369	369	369	369
44	57	22	26	39	58	62	21	40	369	369	369	369	36	73	14	7	47	69	62	21	40	369	369	369	369
73	14	36	28	77	18	10	32	81	369	369	369	369	65	6	52	39	58	26	10	32	81	369	369	369	369
27# Row C1m\Pd1/Pd2													28# Row C1m\Pd1/Pd2												
1	50	72	64	24	35	46	60	17	369	369	369	369	1	50	72	56	13	54	30	68	25	369	369	369	369
42	61	20	5	43	75	68	25	30	369	369	369	369	42	61	20	24	35	64	76	9	38	369	369	369	369
80	12	31	54	56	13	9	38	76	369	369	369	369	80	12	31	43	75	5	17	46	60	369	369	369	369
8	37	78	71	3	49	53	55	15	369	369	369	369	16	48	59	71	3	49	45	74	4	369	369	369	369
48	59	16	19	41	63	66	23	34	369	369	369	369	29	67	27	19	41	63	55	15	53	369	369	369	369
67	27	29	33	79	11	4	45	74	369	369	369	369	78	8	37	33	79	11	23	34	66	369	369	369	369
6	44	73	69	26	28	51	70	2	369	369	369	369	22	36	65	77	7	39	51	70	2	369	369	369	369
52	57	14	7	39	77	62	21	40	369	369	369	369	44	73	6	18	47	58	62	21	40	369	369	369	369
65	22	36	47	58	18	10	32	81	369	369	369	369	57	14	52	28	69	26	10	32	81	369	369	369	369
29# Row C1m\Pd1/Pd2													30# Row C1m\Pd1/Pd2												
1	50	72	16	29	78	22	44	57	369	369	369	369	1	50	72	8	48	67	6	52	65	369	369	369	369
42	61	20	48	67	8	36	73	14	369	369	369	369	42	61	20	37	59	27	44	57	22	369	369	369	369
80	12	31	59	27	37	65	6	52	369	369	369	369	80	12	31	78	16	29	73	14	36	369	369	369	369
56	24	43	71	3	49	77	18	28	369	369	369	369	64	5	54	71	3	49	69	7	47	369	369	369	369
13	35	75	19	41	63	7	47	69	369	369	369	369	24	43	56	19	41	63	26	39	58	369	369	369	369
54	64	5	33	79	11	39	58	26	369	369	369	369	35	75	13	33	79	11	28	77	18	369	369	369	369
30	76	17	45	55	23	51	70	2	369	369	369	369	46	68	9	53	66	4	51	70	2	369	369	369	369
68	9	46	74	15	34	62	21	40	369	369	369	369	60	25	38	55	23	45	62	21	40	369	369	369	369
25	38	60	4	53	66	10	32	81	369	369	369	369	17	30	76	15	34	74	10	32	81	369	369	369	369

** Recompiled and listed by Kanji Setsuda on July 29, 2017
 working with macOS Sierra 10.12.6 and Xcode 7.3 **

```

31#|Row|C1m\Pd1/Pd2|
1 50 72 16 48 59 22 36 65|369|369\369/369|
42 61 20 29 67 27 44 73 6|369|369\369/369|
80 12 31 78 8 37 57 14 52|369|369\369/369|
56 13 54 71 3 49 77 7 39|369|369\369/369|
24 35 64 19 41 63 18 47 58|369|369\369/369|
43 75 5 33 79 11 28 69 26|369|369\369/369|
30 68 25 45 74 4 51 70 2|369|369\369/369|
76 9 38 55 15 53 62 21 40|369|369\369/369|
17 46 60 23 34 66 10 32 81|369|369\369/369|

32#|Row|C1m\Pd1/Pd2|
1 50 72 8 37 78 6 44 73|369|369\369/369|
42 61 20 48 59 16 52 57 14|369|369\369/369|
80 12 31 67 27 29 65 22 36|369|369\369/369|
64 24 35 71 3 49 69 26 28|369|369\369/369|
5 43 75 19 41 63 7 39 77|369|369\369/369|
54 56 13 33 79 11 47 58 18|369|369\369/369|
46 60 17 53 55 15 51 70 2|369|369\369/369|
68 25 30 66 23 34 62 21 40|369|369\369/369|
9 38 76 4 45 74 10 32 81|369|369\369/369|

33#|Row|C1m\Pd1/Pd2|
11 78 34 47 6 70 66 32 25|369|369\369/369|
54 1 68 63 37 23 8 73 42|369|369\369/369|
58 44 21 13 80 30 49 18 56|369|369\369/369|
15 79 29 31 27 65 60 43 20|369|369\369/369|
46 5 72 75 41 7 10 77 36|369|369\369/369|
62 39 22 17 55 51 53 3 67|369|369\369/369|
26 64 33 52 2 69 61 38 24|369|369\369/369|
40 9 74 59 45 19 14 81 28|369|369\369/369|
57 50 16 12 76 35 48 4 71|369|369\369/369|

45#|Row|C1m\Pd1/Pd2|
11 58 54 37 6 80 66 32 25|369|369\369/369|
34 21 68 63 47 13 8 73 42|369|369\369/369|
78 44 1 23 70 30 49 18 56|369|369\369/369|
5 79 39 31 27 65 60 53 10|369|369\369/369|
46 15 62 75 41 7 20 67 36|369|369\369/369|
72 29 22 17 55 51 43 3 77|369|369\369/369|
26 64 33 52 12 59 81 38 4|369|369\369/369|
40 9 74 69 35 19 14 61 48|369|369\369/369|
57 50 16 2 76 45 28 24 71|369|369\369/369|

57#|Row|C1m\Pd1/Pd2|
21 68 34 30 23 70 66 32 25|369|369\369/369|
44 1 78 80 37 6 8 73 42|369|369\369/369|
58 54 11 13 63 47 49 18 56|369|369\369/369|
22 72 29 31 27 65 67 36 20|369|369\369/369|
39 5 79 75 41 7 3 77 43|369|369\369/369|
62 46 15 17 55 51 53 10 60|369|369\369/369|
26 64 33 35 19 69 71 28 24|369|369\369/369|
40 9 74 76 45 2 4 81 38|369|369\369/369|
57 50 16 12 59 52 48 14 61|369|369\369/369|

69#|Row|C1m\Pd1/Pd2|
21 40 62 66 23 34 60 38 25|369|369\369/369|
32 81 10 4 45 74 46 9 68|369|369\369/369|
70 2 51 53 55 15 17 76 30|369|369\369/369|
26 28 69 11 63 49 35 64 24|369|369\369/369|
39 77 7 79 41 3 75 5 43|369|369\369/369|
58 18 47 33 19 71 13 54 56|369|369\369/369|
52 6 65 67 27 29 31 80 12|369|369\369/369|
14 73 36 8 37 78 72 1 50|369|369\369/369|
57 44 22 48 59 16 20 42 61|369|369\369/369|

```

[Count = 72] OK!

[Program List: プログラムのリスト]

```

//
/** Basic Form for our Object **
//
// 6 7 8 9| 1| 2| 3| 4| 5| 6| 7| 8| 9| 1 2 3 4 5
// |-----|
// 15 16 17 18|10|11|12|13|14|15|16|17|18|10 11 12 13 14
// |-----|
// 24 25 26 27|19|20|21|22|23|24|25|26|27|19 20 21 22 23
// |-----|
// 33 34 35 36|28|29|30|31|32|33|34|35|36|28 29 30 31 32
// |-----|
// 42 43 44 45|37|38|39|40|41|42|43|44|45|37 38 39 40 41
// |-----|
// 51 52 53 54|46|47|48|49|50|51|52|53|54|46 47 48 49 50
// |-----|
// 60 61 62 63|55|56|57|58|59|60|61|62|63|55 56 57 58 59
// |-----|
// 69 70 71 72|64|65|66|67|68|69|70|71|72|64 65 66 67 68
// |-----|
// 78 79 80 81|73|74|75|76|77|78|79|80|81|73 74 75 76 77
// |-----|
/** Multiple 3*3*9 Conditions: SS=123; **

```

```

// Block #0:
// n1+n2+n3=SS ...rw00; n10+n11+n12=SS ...rw01; n19+n20+n21=SS ...rw02;
// n1+n10+n19=SS ...cl00; n2+n11+n20=SS ...cl01; n3+n12+n21=SS ...cl02;
// Block #1:
// n4+n5+n6=SS ...rw10; n13+n14+n15=SS ...rw11; n22+n23+n24=SS ...rw12;
// n4+n13+n22=SS ...cl10; n5+n14+n23=SS ...cl11; n6+n15+n24=SS ...cl12;
// Block #2:
// n7+n8+n9=SS ...rw20; n16+n17+n18=SS ...rw21; n25+n26+n27=SS ...rw22;
// n7+n16+n25=SS ...cl20; n8+n17+n26=SS ...cl21; n9+n18+n27=SS ...cl22;
// Block #3:
// n28+n29+n30=SS ...rw30; n37+n38+n39=SS ...rw31; n46+n47+n48=SS ...rw32;
// n28+n37+n46=SS ...cl30; n29+n38+n47=SS ...cl31; n30+n39+n48=SS ...cl32;
// Block #4:
// n31+n32+n33=SS ...rw40; n40+n41+n42=SS ...rw41; n49+n50+n51=SS ...rw42;
// n31+n40+n49=SS ...cl40; n32+n41+n50=SS ...cl41; n33+n42+n51=SS ...cl42;
// Block #5:
// n34+n35+n36=SS ...rw50; n43+n44+n45=SS ...rw51; n52+n53+n54=SS ...rw52;
// n34+n43+n52=SS ...cl50; n35+n44+n53=SS ...cl51; n36+n45+n54=SS ...cl52;
// Block #6:
// n55+n56+n57=SS ...rw60; n64+n65+n66=SS ...rw61; n73+n74+n75=SS ...rw62;
// n55+n64+n73=SS ...cl60; n56+n65+n74=SS ...cl61; n57+n66+n75=SS ...cl62;
// Block #7:
// n58+n59+n60=SS ...rw70; n67+n68+n69=SS ...rw71; n76+n77+n78=SS ...rw72;
// n58+n67+n76=SS ...cl70; n59+n68+n77=SS ...cl71; n60+n69+n78=SS ...cl72;
// Block #8:
// n61+n62+n63=SS ...rw80; n70+n71+n72=SS ...rw81; n79+n80+n81=SS ...rw82;
// n61+n70+n79=SS ...cl80; n62+n71+n80=SS ...cl81; n63+n72+n81=SS ...cl82;
//
//** Additional Conditions: **
// n1+n2+n3+n4+nn10+n11+n12+n19+n20+n28=n81+n80+n79+n78+n72+n71+n70+n63+n62+n54=410;
// n6+n7+n8+n9+nn16+n17+n18+n26+n27+n36=n76+n75+n74+n73+n66+n65+n64+n56+n55+n46=410;
//
//** Self-Complementary Conditions: SC=82; **
// n1+n81=SC; n2+n80=SC; n3+n79=SC; n4+n78=SC;
// n5+n77=SC; n6+n76=SC; n7+n75=SC; n8+n74=SC;
// n9+n73=SC; n10+n72=SC; n11+n71=SC; n12+n70=SC;
// n13+n69=SC; n14+n68=SC; n15+n67=SC; n16+n66=SC;
// n17+n65=SC; n18+n64=SC; n19+n63=SC; n20+n62=SC;
// n21+n61=SC; n22+n60=SC; n23+n59=SC; n24+n58=SC;
// n25+n57=SC; n26+n56=SC; n27+n55=SC; n28+n54=SC;
// n29+n53=SC; n30+n52=SC; n31+n51=SC; n32+n50=SC;
// n33+n49=SC; n34+n48=SC; n35+n47=SC; n36+n46=SC;
// n37+n45=SC; n38+n44=SC; n39+n43=SC; n40+n42=SC;
// n41+n41=SC;
//
//** Pandiagonal Conditions: PC=369; **
// n1+n11+n21+n31+n41+n51+n61+n71+n81=PC ... pd1;
// n2+n12+n22+n32+n42+n52+n62+n72+n82=PC ... pd2;
// n3+n13+n23+n33+n43+n53+n63+n73+n83=PC ... pd3;
// n4+n14+n24+n34+n44+n54+n64+n74+n84=PC ... pd4;
// n5+n15+n25+n35+n45+n55+n65+n75+n85=PC ... pd5;
// n6+n16+n26+n36+n46+n56+n66+n76+n86=PC ... pd6;
// n7+n17+n27+n37+n47+n57+n67+n77+n87=PC ... pd7;
// n8+n18+n28+n38+n48+n58+n68+n78+n88=PC ... pd8;
// n9+n19+n29+n39+n49+n59+n69+n79+n89=PC ... pd9;
//
// n1+n18+n26+n34+n42+n50+n58+n66+n74=PC ... pb1;
// n2+n10+n27+n35+n43+n51+n59+n67+n75=PC ... pb2;
// n3+n11+n19+n36+n44+n52+n60+n68+n76=PC ... pb3;
// n4+n12+n20+n28+n45+n53+n61+n69+n77=PC ... pb4;
// n5+n13+n21+n29+n37+n54+n62+n70+n78=PC ... pb5;
// n6+n14+n22+n30+n38+n46+n63+n71+n79=PC ... pb6;
// n7+n15+n23+n31+n39+n47+n55+n72+n80=PC ... pb7;
// n8+n16+n24+n32+n40+n48+n56+n64+n81=PC ... pb8;
// n9+n17+n25+n33+n41+n49+n57+n65+n73=PC ... pb9;

```

```

//
/** Special type of Simultaneous Magic Squares of Order 9: **
/** Self-complementary, Pan-diagonal and Multiple 3*3*9; **
/** 'SmLMS99Mul.c': dictated by Kanji Setsuda **
/** in 2000, 2006, 2008, 2013 and July 29, 2017 **
/** with macOS Sierra 10.12.6 and Xcode 7.3; **
//
#include <stdio.h>
//
/** Global Variables *
short cnt, cnt2;
short cnt3;
short LSM, SSM, SXN, CC;
short nm[82], uflg[82];
short pm[8], pn[8];
short dn[3][82];
short cs[3][37];
//
short nt11[62], nt17[50], nt23[38], nt25[34], nt29[26], nt33[18];
//
/** Sub-Routines *
void stp01(void), stp02(void), stp03(void), stp04(void);
void stp05(void), stp06(void), stp07(void), stp08(void);
void stp09(void), stp10(void);
void stp11(void), stp12(void), stp13(void), stp14(void);
void stp15(void), stp16(void), stp17(void), stp18(void);
void stp19(void), stp20(void), stp21(void), stp22(void);
void stp23(void), stp24(void), stp25(void), stp26(void);
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);
void ansprint(void), prxans(short x);
void chksms(short x);
//
/** Main Program *
int main(){
short n;
printf("//\n");
printf("/** Special type of Simultaneous Magic Squares of Order 9:\n");
printf("// Self-complementary, Pan-diagonal and Multiple 3*3*9;\n");
printf("// [Standard Solutions with precise Check-sum Monitors] **\n");
for(n=0;n<82;n++){nm[n]=0; uflg[n]=0;}
LSM=369; SSM=123; SXN=41; CC=82; cnt=0; cnt3=0;
pm[0]=1; pm[1]=11; pm[2]=21; pm[3]=31; pm[4]=51; pm[5]=61; pm[6]=71; pm[7]=81;
pn[0]=9; pn[1]=17; pn[2]=25; pn[3]=33; pn[4]=49; pn[5]=57; pn[6]=65; pn[7]=73;
nm[41]=41; uflg[41]=1;
stp01(); /** Begin The Calculations *
if(cnt3>0){prxans(cnt3);}
uflg[41]=0;
printf("//\n");
printf("// [Count = %d] OK!\n",cnt);
printf("//\n");
printf("/** Recompiled by Kanji Setsuda on July 29, 2017\n");
printf("// working with macOS Sierra 10.12.6 and Xcode 7.3 **\n");
printf("//\n");
return 0;
}
//
/** Begin The Calculations *
/** Define Level 1: *
/** Set (N1,n81), (N11,n71), (N21,n61), (N31,n51) *
/** Set N1 & n81 & N1<n81 *

```

```

void stp01(){
short a,b,n;
for(n=0;n<8;n++){a=pm[n]; b=CC-a;
if((a<b)&&(uflg[a]==0)&&(uflg[b]==0)){
nm[1]=a; nm[81]=b;
uflg[a]=1; uflg[b]=1;
stp02();
uflg[b]=0; uflg[a]=0;}}
}
}
/* Set N11 & n71 *
void stp02(){
short a,b,n;
for(n=0;n<8;n++){a=pm[n]; b=CC-a;
if((uflg[a]==0)&&(uflg[b]==0)){cnt2=0;
nm[11]=a; nm[71]=b;
uflg[a]=1; uflg[b]=1;
stp03();
uflg[b]=0; uflg[a]=0;}}
}
}
/* Set N21 & n61 *
void stp03(){
short a,b,n;
for(n=0;n<8;n++){a=pm[n]; b=CC-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[21]=a; nm[61]=b;
uflg[a]=1; uflg[b]=1;
stp04();
uflg[b]=0; uflg[a]=0;}}
}
}
/* Set N31 & n51 *
void stp04(){
short a,b,n;
for(n=0;n<8;n++){a=pm[n]; b=CC-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[31]=a; nm[51]=b;
uflg[a]=1; uflg[b]=1;
stp05();
uflg[b]=0; uflg[a]=0;}}
}
}
/* Set (N9,n73), (N17,n65), (N25,n57), (N33,n49) *
/* Set N9(>N1) & n73(N1) *
void stp05(){
short a,b,n;
for(n=0;n<8;n++){a=pm[n]; b=CC-a;
if((a>nm[1])&&(b>nm[1])){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[9]=a; nm[73]=b;
uflg[a]=1; uflg[b]=1;
stp06();
uflg[b]=0; uflg[a]=0;}}}
}
}
/* Set N17 & n65 *
void stp06(){
short a,b,n;
for(n=0;n<8;n++){a=pm[n]; b=CC-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[17]=a; nm[65]=b;
uflg[a]=1; uflg[b]=1;
stp07();
uflg[b]=0; uflg[a]=0;}}
}
}

```

```

}
}
/* Set N25 & n57 */
void stp07(){
short a,b,n;
for(n=0;n<8;n++){a=pn[n]; b=CC-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[25]=a; nm[57]=b;
uflg[a]=1; uflg[b]=1;
stp08();
uflg[b]=0; uflg[a]=0;}}
}
}
/* Set N33 & n49 */
void stp08(){
short a,b,n;
for(n=0;n<8;n++){a=pn[n]; b=CC-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[33]=a; nm[49]=b;
uflg[a]=1; uflg[b]=1;
stp09();
uflg[b]=0; uflg[a]=0;}}
}
}
/* Set n32=SSM-n31-n33 & n50 */
void stp09(){
short a,b;
a=SSM-nm[31]-nm[33];
b=SSM-nm[51]-nm[49];
if((0<a)&&(a<82)&&(a+b==CC)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[32]=a; nm[50]=b;
uflg[a]=1; uflg[b]=1;
stp10();
uflg[b]=0; uflg[a]=0;}}}
}
}
/* Set n40=SSM-n31-n49 & n42 */
void stp10(){
short a,b,sm;
a=SSM-nm[31]-nm[49];
b=SSM-nm[51]-nm[33];
if((0<a)&&(a<82)&&(a+b==CC)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[40]=a; nm[42]=b;
uflg[a]=1; uflg[b]=1;
sm=nm[40]+nm[41]+nm[42];
if(sm==SSM){stp11();}}
uflg[b]=0; uflg[a]=0;}}}
}
}
/* Define Level 2: */
/* Set N2 & n80 */
void stp11(){
short a,b,n,m;
m=1;
for(n=1;n<82;n++){if(uflg[n]==0){nt11[m]=n; m++;}}
nt11[m]=0;
for(n=m-1;n>0;n--){if(nm[1]==1){cnt2=0;}}
a=nt11[n]; b=CC-a;
nm[2]=a; nm[80]=b;
uflg[a]=1; uflg[b]=1;
stp12();
uflg[b]=0; uflg[a]=0;}}
}
}
/* Set n3=SSM-n1-n2 & n79 */

```

```

void stp12(){
short a,b;
a=SSM-nm[1]-nm[2];
b=SSM-nm[81]-nm[80];
if((0<a)&&(a<82)&&(a+b==CC)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[3]=a; nm[79]=b;
uflg[a]=1; uflg[b]=1;
stp13();
uflg[b]=0; uflg[a]=0;}}
}
/* Set n12=SSM-n3-n21 & n70 *
void stp13(){
short a,b;
a=SSM-nm[3]-nm[21];
b=SSM-nm[79]-nm[61];
if((0<a)&&(a<82)&&(a+b==CC)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[12]=a; nm[70]=b;
uflg[a]=1; uflg[b]=1;
stp14();
uflg[b]=0; uflg[a]=0;}}
}
/* Set n10=SSM-n11-n12 & n72 & n10<N2 *
void stp14(){
short a,b;
a=SSM-nm[11]-nm[12];
b=SSM-nm[71]-nm[70];
if((0<a)&&(a<nm[2])&&(a+b==CC)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[10]=a; nm[72]=b;
uflg[a]=1; uflg[b]=1;
stp15();
uflg[b]=0; uflg[a]=0;}}
}
/* Set n19=SSM-n1-n10 & n63 *
void stp15(){
short a,b;
a=SSM-nm[1]-nm[10];
b=SSM-nm[81]-nm[72];
if((0<a)&&(a<82)&&(a+b==CC)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[19]=a; nm[63]=b;
uflg[a]=1; uflg[b]=1;
stp16();
uflg[b]=0; uflg[a]=0;}}
}
/* Set n20=SSM-n2-n11 & n62 *
void stp16(){
short a,b,c,d;
a=SSM-nm[2]-nm[11];
b=SSM-nm[80]-nm[71];
c=SSM-nm[19]-nm[21];
d=SSM-nm[63]-nm[61];
if((0<a)&&(a<82)&&(a+b==CC)&&(a==c)&&(b==d)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[20]=a; nm[62]=b;
uflg[a]=1; uflg[b]=1;
stp17();
uflg[b]=0; uflg[a]=0;}}
}
/* Set N8 & n74 *
void stp17(){
short a,b,n,m;
m=1;

```

```

for(n=1;n<82;n++){if(uflg[n]==0){nt17[m]=n; m++;}}
nt17[m]=0;
for(n=m-1;n>0;n--){
a=nt17[n]; b=CC-a;
nm[8]=a; nm[74]=b;
uflg[a]=1; uflg[b]=1;
stp18();
uflg[b]=0; uflg[a]=0;
}
}
/* Set n7=SSM-n9-n8 & n75 *
void stp18(){
short a,b;
a=SSM-nm[9]-nm[8];
b=SSM-nm[73]-nm[74];
if((0<a)&&(a<82)&&(a+b==CC)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[7]=a; nm[75]=b;
uflg[a]=1; uflg[b]=1;
stp19();
uflg[b]=0; uflg[a]=0;}}
}
/* Set n16=SSM-n7-n25 & n66 *
void stp19(){
short a,b;
a=SSM-nm[7]-nm[25];
b=SSM-nm[75]-nm[57];
if((0<a)&&(a<82)&&(a+b==CC)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[16]=a; nm[66]=b;
uflg[a]=1; uflg[b]=1;
stp20();
uflg[b]=0; uflg[a]=0;}}
}
/* Set n18=SSM-n16-n17 & n64 *
void stp20(){
short a,b;
a=SSM-nm[16]-nm[17];
b=SSM-nm[66]-nm[65];
if((0<a)&&(a<82)&&(a+b==CC)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[18]=a; nm[64]=b;
uflg[a]=1; uflg[b]=1;
stp21();
uflg[b]=0; uflg[a]=0;}}
}
/* Set n26=SSM-n8-n17 & n56 *
void stp21(){
short a,b;
a=SSM-nm[8]-nm[17];
b=SSM-nm[74]-nm[65];
if((0<a)&&(a<82)&&(a+b==CC)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[26]=a; nm[56]=b;
uflg[a]=1; uflg[b]=1;
stp22();
uflg[b]=0; uflg[a]=0;}}
}
/* Set n27=SSM-n9-n18 & n55 *
void stp22(){
short a,b,c,d;
a=SSM-nm[9]-nm[18];
b=SSM-nm[73]-nm[64];
c=SSM-nm[25]-nm[26];
d=SSM-nm[57]-nm[56];

```

```

if((0<a)&&(a<82)&&(a+b==CC)&&(a==c)&&(b==d)){
    if((uflg[a]==0)&&(uflg[b]==0)){
        nm[27]=a; nm[55]=b;
        uflg[a]=1; uflg[b]=1;
        stp23();
        uflg[b]=0; uflg[a]=0;}}
}
/* Define Level 3: *
/* Set N4 & n78 *
void stp23(){
    short a,b,n,m;
    m=1;
    for(n=1;n<82;n++){if(uflg[n]==0){nt23[m]=n; m++;}}
    nt23[m]=0;
    for(n=1;n<m;n++){
        a=nt23[n]; b=CC-a;
        nm[4]=a; nm[78]=b;
        uflg[a]=1; uflg[b]=1;
        stp24();
        uflg[b]=0; uflg[a]=0;
    }
}
/* Set n28=SXN-n4+n21 & n54=SXN-n78+n61 *
// (n1+n2+n3+n4+nn10+n11+n12+n19+n20+n28=410;
// n81+n80+n79+n78+nn72+n71+n70+n63+n62+n54=410;)
void stp24(){
    short a,b;
    a=SXN-nm[4]+nm[21];
    b=SXN-nm[78]+nm[61];
    if((0<a)&&(a<82)&&(a+b==CC)){
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[28]=a; nm[54]=b;
            uflg[a]=1; uflg[b]=1;
            stp25();
            uflg[b]=0; uflg[a]=0;}}
}
/* Set N6 & n76 */
void stp25(){
    short a,b,n,m;
    m=1;
    for(n=1;n<82;n++){if(uflg[n]==0){nt25[m]=n; m++;}}
    nt25[m]=0;
    for(n=m-1;n>0;n--){
        a=nt25[n]; b=CC-a;
        nm[6]=a; nm[76]=b;
        uflg[a]=1; uflg[b]=1;
        stp26();
        uflg[b]=0; uflg[a]=0;
    }
}
/* Set n5=LSM-n1-n2-n3-n4-n6-n7-n8-n9 & n77 *
void stp26(){
    short a,b;
    a=LSM-nm[1]-nm[2]-nm[3]-nm[4]-nm[6]-nm[7]-nm[8]-nm[9];
    b=SSM-nm[78]-nm[76];
    if((0<a)&&(a<82)&&(a+b==CC)){
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[5]=a; nm[77]=b;
            uflg[a]=1; uflg[b]=1;
            stp27();
            uflg[b]=0; uflg[a]=0;}}
}
/* Set n36=SXN-n6+n25 & n46=SXN-n76+n57 *
// (n6+n7+n8+n9+nn16+n17+n18+n26+n27+n36=410;
// n76+n75+n74+n73+nn66+n65+n64+n56+n55+n46=410;)

```

```

void stp27(){
short a,b;
a=SXN-nm[6]+nm[25];
b=SXN-nm[76]+nm[57];
if((0<a)&&(a<82)&&(a+b==CC)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[36]=a; nm[46]=b;
uflg[a]=1; uflg[b]=1;
stp28();
uflg[b]=0; uflg[a]=0;}}
}
/* Set n37=LSM-n1-n10-n19-n28-n46-n55-n64-n73 & n45 *
void stp28(){
short a,b;
a=LSM-nm[1]-nm[10]-nm[19]-nm[28]-nm[46]-nm[55]-nm[64]-nm[73];
b=SSM-nm[54]-nm[36];
if((0<a)&&(a<82)&&(a+b==CC)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[37]=a; nm[45]=b;
uflg[a]=1; uflg[b]=1;
stp29();
uflg[b]=0; uflg[a]=0;}}
}
/* Define Level 4: *
/* Set N13 & n69 *
void stp29(){
short a,b,n,m;
m=1;
for(n=1;n<82;n++){if(uflg[n]==0){nt29[m]=n; m++;}}
nt29[m]=0;
for(n=1;n<m;n++){
a=nt29[n]; b=CC-a;
nm[13]=a; nm[69]=b;
uflg[a]=1; uflg[b]=1;
stp30();
uflg[b]=0; uflg[a]=0;
}
}
/* Set n29=LSM-n5-n13-n21-n37-n54-n62-n70-n78 & n53 *
void stp30(){
short a,b;
a=LSM-nm[5]-nm[13]-nm[21]-nm[37]-nm[54]-nm[62]-nm[70]-nm[78];
b=LSM-nm[4]-nm[12]-nm[20]-nm[28]-nm[45]-nm[61]-nm[69]-nm[77];
if((0<a)&&(a<82)&&(a+b==CC)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[29]=a; nm[53]=b;
uflg[a]=1; uflg[b]=1;
stp31();
uflg[b]=0; uflg[a]=0;}}
}
/* Set n22=SSM-n4-n13 & n60 *
void stp31(){
short a,b;
a=SSM-nm[4]-nm[13];
b=SSM-nm[78]-nm[69];
if((0<a)&&(a<82)&&(a+b==CC)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[22]=a; nm[60]=b;
uflg[a]=1; uflg[b]=1;
stp32();
uflg[b]=0; uflg[a]=0;}}
}
/* Set n30=SSM-n28-n29 & n52 *
void stp32(){
short a,b,c,d;

```

```

a=LSM-nm[9]-nm[10]-nm[20]-nm[40]-nm[50]-nm[60]-nm[70]-nm[80];
b=LSM-nm[2]-nm[12]-nm[22]-nm[32]-nm[42]-nm[62]-nm[72]-nm[73];
c=SSM-nm[28]-nm[29];
d=SSM-nm[54]-nm[53];
if((0<a)&&(a<82)&&(a+b==CC)&&(a==c)&&(b==d)){
    if((uflg[a]==0)&&(uflg[b]==0)){
        nm[30]=a; nm[52]=b;
        uflg[a]=1; uflg[b]=1;
        stp33();
        uflg[b]=0; uflg[a]=0;}}
}
/* Set N15 & n67 *
void stp33(){
short a,b,n,m;
m=1;
for(n=1;n<82;n++){if(uflg[n]==0){nt33[m]=n; m++;}}
nt33[m]=0;
for(n=1;n<m;n++){
a=nt33[n]; b=CC-a;
nm[15]=a; nm[67]=b;
uflg[a]=1; uflg[b]=1;
stp34();
uflg[b]=0; uflg[a]=0;
}
}
/* Set n14=SSM-n13-n15 & n68 *
void stp34(){
short a,b;
a=SSM-nm[13]-nm[15];
b=SSM-nm[69]-nm[67];
if((0<a)&&(a<82)&&(a+b==CC)){
    if((uflg[a]==0)&&(uflg[b]==0)){
        nm[14]=a; nm[68]=b;
        uflg[a]=1; uflg[b]=1;
        stp35();
        uflg[b]=0; uflg[a]=0;}}
}
/* Set n23=SSM-n5-n14 & n59 *
void stp35(){
short a,b;
a=SSM-nm[5]-nm[14];
b=SSM-nm[77]-nm[68];
if((0<a)&&(a<82)&&(a+b==CC)){
    if((uflg[a]==0)&&(uflg[b]==0)){
        nm[23]=a; nm[59]=b;
        uflg[a]=1; uflg[b]=1;
        stp36();
        uflg[b]=0; uflg[a]=0;}}
}
/* Set n24=SSM-n22-n23 & n58 *
void stp36(){
short a,b,c,d;
a=SSM-nm[22]-nm[23];
b=SSM-nm[60]-nm[59];
c=SSM-nm[6]-nm[15];
d=SSM-nm[76]-nm[67];
if((0<a)&&(a<82)&&(a+b==CC)&&(a==c)&&(b==d)){
    if((uflg[a]==0)&&(uflg[b]==0)){
        nm[24]=a; nm[58]=b;
        uflg[a]=1; uflg[b]=1;
        stp37();
        uflg[b]=0; uflg[a]=0;}}
}
/* Define Level 5: *
/* Set n35=LSM-n5-n15-n25-n45-n46-n56-n66-n76 & n47 *

```

```

void stp37(){
short a,b;
a=LSM-nm[5]-nm[15]-nm[25]-nm[45]-nm[46]-nm[56]-nm[66]-nm[76];
b=LSM-nm[6]-nm[16]-nm[26]-nm[36]-nm[37]-nm[57]-nm[67]-nm[77];
if((0<a)&&(a<82)&&(a+b==CC)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[35]=a; nm[47]=b;
uflg[a]=1; uflg[b]=1;
stp38();
uflg[b]=0; uflg[a]=0;}}
}
/** Set n34=LSM-n1-n18-n26-n42-n50-n58-n66-n74 & n48 *
void stp38(){
short a,b,c,d;
a=LSM-nm[1]-nm[18]-nm[26]-nm[42]-nm[50]-nm[58]-nm[66]-nm[74];
b=LSM-nm[8]-nm[16]-nm[24]-nm[32]-nm[40]-nm[56]-nm[64]-nm[81];
c=SSM-nm[36]-nm[35];
d=SSM-nm[46]-nm[47];
if((0<a)&&(a<82)&&(a+b==CC)&&(a==c)&&(b==d)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[34]=a; nm[48]=b;
uflg[a]=1; uflg[b]=1;
stp39();
uflg[b]=0; uflg[a]=0;}}
}
/** Set n39=LSM-n8-n18-n19-n29-n49-n59-n69-n79 & n43 *
void stp39(){
short a,b,c,d;
a=LSM-nm[8]-nm[18]-nm[19]-nm[29]-nm[49]-nm[59]-nm[69]-nm[79];
b=LSM-nm[3]-nm[13]-nm[23]-nm[33]-nm[53]-nm[63]-nm[64]-nm[74];
c=LSM-nm[7]-nm[15]-nm[23]-nm[31]-nm[47]-nm[55]-nm[72]-nm[80];
d=LSM-nm[2]-nm[10]-nm[27]-nm[35]-nm[51]-nm[59]-nm[67]-nm[75];
if((0<a)&&(a<82)&&(a+b==CC)&&(a==c)&&(b==d)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[39]=a; nm[43]=b;
uflg[a]=1; uflg[b]=1;
stp40();
uflg[b]=0; uflg[a]=0;}}
}
/** Check some Sums *
void stp40(){
short sm1,sm2;
sm1=nm[30]+nm[39]+nm[48];
sm2=nm[52]+nm[43]+nm[34];
if((sm1==SSM)&&(sm2==SSM)){stp41();}
}
/** Set n38=LSM-n4-n14-n34-n44-n54-n55-n65-n75 & n44 *
void stp41(){
short a,b,c,d;
a=LSM-nm[7]-nm[17]-nm[27]-nm[28]-nm[48]-nm[58]-nm[68]-nm[78];
b=LSM-nm[4]-nm[14]-nm[24]-nm[34]-nm[54]-nm[55]-nm[65]-nm[75];
c=LSM-nm[6]-nm[14]-nm[22]-nm[30]-nm[46]-nm[63]-nm[71]-nm[79];
d=LSM-nm[3]-nm[11]-nm[19]-nm[36]-nm[52]-nm[60]-nm[68]-nm[76];
if((0<a)&&(a<82)&&(a+b==CC)&&(a==c)&&(b==d)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[38]=a; nm[44]=b;
uflg[a]=1; uflg[b]=1;
stp42();
uflg[b]=0; uflg[a]=0;}}
}
/** Check some Sums *
void stp42(){
short sm1,sm2,sm3,sm4;
sm1=nm[29]+nm[38]+nm[47];
sm2=nm[53]+nm[44]+nm[35];

```

```

sm3=nm[37]+nm[38]+nm[39];
sm4=nm[45]+nm[44]+nm[43];
if((sm1==SSM)&&(sm2==SSM)&&(sm3==SSM)&&(sm4==SSM)){ansprint();}
}
//
/** Print the Answers *
void ansprint(void){
short n;
cnt++; cnt2++;
if(cnt2==1){
dn[cnt3][0]=cnt;
for(n=1;n<82;n++){dn[cnt3][n]=nm[n];}
chksms(cnt3); cnt3++;
if(cnt3==2){prxans(cnt3); cnt3=0;}
}
}
}
/** Print Answers with Check-sums *
void prxans(short x){
short m,l,l9,n;
printf("//");
for(m=0;m<x;m++){
printf("%26d#|Row|C|m\\Pd1/Pd2|",dn[m][0]);
if(m+1<x){printf(" ");}}
printf("\n");
for(l=0;l<9;l++){l9=l*9;
printf("//");
for(m=0;m<x;m++){
for(n=1;n<10;n++){printf("%3d",dn[m][l9+n]);}
printf("%3d|%3d\\%3d/%3d|",cs[m][l+1],cs[m][l+10],cs[m][l+19],cs[m][l+28]);
if(m+1<x){printf(" ");}
}
printf("\n");
}
}
}
//
/** Check Sum Monitor *
void chksms(short x){
cs[x][ 1]=dn[x][ 1]+dn[x][ 2]+dn[x][ 3]+dn[x][ 4]+dn[x][ 5]+dn[x][ 6]+dn[x][ 7]+dn[x][ 8]+dn[x][ 9];
cs[x][ 2]=dn[x][10]+dn[x][11]+dn[x][12]+dn[x][13]+dn[x][14]+dn[x][15]+dn[x][16]+dn[x][17]+dn[x][18];
cs[x][ 3]=dn[x][19]+dn[x][20]+dn[x][21]+dn[x][22]+dn[x][23]+dn[x][24]+dn[x][25]+dn[x][26]+dn[x][27];
cs[x][ 4]=dn[x][28]+dn[x][29]+dn[x][30]+dn[x][31]+dn[x][32]+dn[x][33]+dn[x][34]+dn[x][35]+dn[x][36];
cs[x][ 5]=dn[x][37]+dn[x][38]+dn[x][39]+dn[x][40]+dn[x][41]+dn[x][42]+dn[x][43]+dn[x][44]+dn[x][45];
cs[x][ 6]=dn[x][46]+dn[x][47]+dn[x][48]+dn[x][49]+dn[x][50]+dn[x][51]+dn[x][52]+dn[x][53]+dn[x][54];
cs[x][ 7]=dn[x][55]+dn[x][56]+dn[x][57]+dn[x][58]+dn[x][59]+dn[x][60]+dn[x][61]+dn[x][62]+dn[x][63];
cs[x][ 8]=dn[x][64]+dn[x][65]+dn[x][66]+dn[x][67]+dn[x][68]+dn[x][69]+dn[x][70]+dn[x][71]+dn[x][72];
cs[x][ 9]=dn[x][73]+dn[x][74]+dn[x][75]+dn[x][76]+dn[x][77]+dn[x][78]+dn[x][79]+dn[x][80]+dn[x][81];
//
cs[x][10]=dn[x][ 1]+dn[x][10]+dn[x][19]+dn[x][28]+dn[x][37]+dn[x][46]+dn[x][55]+dn[x][64]+dn[x][73];
cs[x][11]=dn[x][ 2]+dn[x][11]+dn[x][20]+dn[x][29]+dn[x][38]+dn[x][47]+dn[x][56]+dn[x][65]+dn[x][74];
cs[x][12]=dn[x][ 3]+dn[x][12]+dn[x][21]+dn[x][30]+dn[x][39]+dn[x][48]+dn[x][57]+dn[x][66]+dn[x][75];
cs[x][13]=dn[x][ 4]+dn[x][13]+dn[x][22]+dn[x][31]+dn[x][40]+dn[x][49]+dn[x][58]+dn[x][67]+dn[x][76];
cs[x][14]=dn[x][ 5]+dn[x][14]+dn[x][23]+dn[x][32]+dn[x][41]+dn[x][50]+dn[x][59]+dn[x][68]+dn[x][77];
cs[x][15]=dn[x][ 6]+dn[x][15]+dn[x][24]+dn[x][33]+dn[x][42]+dn[x][51]+dn[x][60]+dn[x][69]+dn[x][78];
cs[x][16]=dn[x][ 7]+dn[x][16]+dn[x][25]+dn[x][34]+dn[x][43]+dn[x][52]+dn[x][61]+dn[x][70]+dn[x][79];
cs[x][17]=dn[x][ 8]+dn[x][17]+dn[x][26]+dn[x][35]+dn[x][44]+dn[x][53]+dn[x][62]+dn[x][71]+dn[x][80];
cs[x][18]=dn[x][ 9]+dn[x][18]+dn[x][27]+dn[x][36]+dn[x][45]+dn[x][54]+dn[x][63]+dn[x][72]+dn[x][81];
//
cs[x][19]=dn[x][ 1]+dn[x][11]+dn[x][21]+dn[x][31]+dn[x][41]+dn[x][51]+dn[x][61]+dn[x][71]+dn[x][81];
cs[x][20]=dn[x][ 2]+dn[x][12]+dn[x][22]+dn[x][32]+dn[x][42]+dn[x][52]+dn[x][62]+dn[x][72]+dn[x][81];
cs[x][21]=dn[x][ 3]+dn[x][13]+dn[x][23]+dn[x][33]+dn[x][43]+dn[x][53]+dn[x][63]+dn[x][73]+dn[x][81];
cs[x][22]=dn[x][ 4]+dn[x][14]+dn[x][24]+dn[x][34]+dn[x][44]+dn[x][54]+dn[x][64]+dn[x][74]+dn[x][81];
cs[x][23]=dn[x][ 5]+dn[x][15]+dn[x][25]+dn[x][35]+dn[x][45]+dn[x][55]+dn[x][65]+dn[x][75]+dn[x][81];
cs[x][24]=dn[x][ 6]+dn[x][16]+dn[x][26]+dn[x][36]+dn[x][46]+dn[x][56]+dn[x][66]+dn[x][76]+dn[x][81];
cs[x][25]=dn[x][ 7]+dn[x][17]+dn[x][27]+dn[x][37]+dn[x][47]+dn[x][57]+dn[x][67]+dn[x][77]+dn[x][81];
cs[x][26]=dn[x][ 8]+dn[x][18]+dn[x][28]+dn[x][38]+dn[x][48]+dn[x][58]+dn[x][68]+dn[x][78]+dn[x][81];
cs[x][27]=dn[x][ 9]+dn[x][19]+dn[x][29]+dn[x][39]+dn[x][49]+dn[x][59]+dn[x][69]+dn[x][79]+dn[x][81];
//
cs[x][28]=dn[x][ 1]+dn[x][18]+dn[x][26]+dn[x][34]+dn[x][42]+dn[x][50]+dn[x][58]+dn[x][66]+dn[x][74];

```

```
cs[x][29]=dn[x][ 2]+dn[x][10]+dn[x][27]+dn[x][35]+dn[x][43]+dn[x][51]+dn[x][59]+dn[x][67]+dn[x][75];
cs[x][30]=dn[x][ 3]+dn[x][11]+dn[x][19]+dn[x][36]+dn[x][44]+dn[x][52]+dn[x][60]+dn[x][68]+dn[x][76];
cs[x][31]=dn[x][ 4]+dn[x][12]+dn[x][20]+dn[x][28]+dn[x][45]+dn[x][53]+dn[x][61]+dn[x][69]+dn[x][77];
cs[x][32]=dn[x][ 5]+dn[x][13]+dn[x][21]+dn[x][29]+dn[x][37]+dn[x][54]+dn[x][62]+dn[x][70]+dn[x][78];
cs[x][33]=dn[x][ 6]+dn[x][14]+dn[x][22]+dn[x][30]+dn[x][38]+dn[x][46]+dn[x][63]+dn[x][71]+dn[x][79];
cs[x][34]=dn[x][ 7]+dn[x][15]+dn[x][23]+dn[x][31]+dn[x][39]+dn[x][47]+dn[x][55]+dn[x][72]+dn[x][80];
cs[x][35]=dn[x][ 8]+dn[x][16]+dn[x][24]+dn[x][32]+dn[x][40]+dn[x][48]+dn[x][56]+dn[x][64]+dn[x][81];
cs[x][36]=dn[x][ 9]+dn[x][17]+dn[x][25]+dn[x][33]+dn[x][41]+dn[x][49]+dn[x][57]+dn[x][65]+dn[x][73];
}
//
/** Recompiled by Kanji Setsuda on July 29, 2017
//   working with macOS Sierra 10.12.6 and Xcode 7.3 **
//
//End_Of_File
```