

第4章：魔方阵の基礎的研究（上級）： 摂田 寛二

第2節： 正方連結汎立体四方陣の代数的研究： Part 1

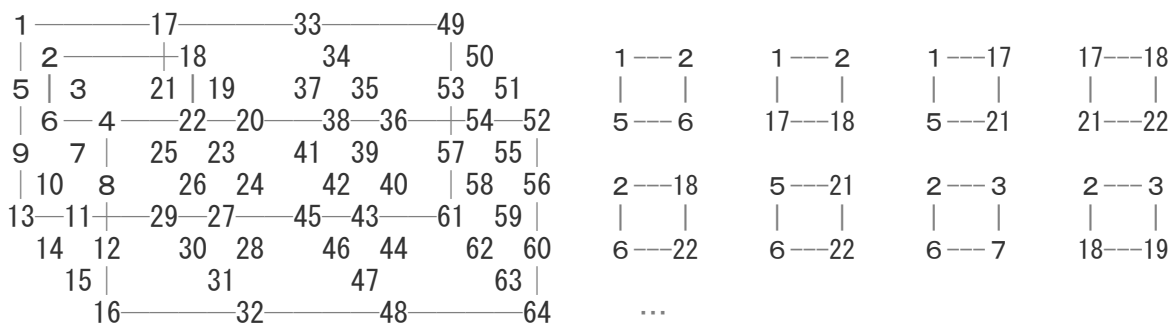
0. 最近昔の記事を読み直していて、幾つかの箇所で大変な遺漏のあることに気が付いた。例えば、今のこの節である。以前の記事では、**正方連結汎立体四方陣**の代数的研究ということで、この方陣の「基本解」を捜して特定するという作業を行っている。

この型に関する最初の説明はともかく、すぐ次に「補数セット」だとか、「調べる場合の数」だとか、変換法の話が出て来て、とてもすぐには理解し難い大変な展開になっている。

第一部なのに、いきなり不親切な記事を書いたものだと、我ながら反省している。

ここでは、目的方陣の「概念」を素直に説明し、先ず解の総数を普通の方法で求めて置くべきだろう。そこで大幅補充のために、節をいくつかに分ける。**Part 1**では、他の**正方連結完全立体四方陣**と**正方連結対称立体四方陣**についても、導入的な説明をしておこう。

1. **[定義]** 立方体四方陣の基本ポジションは、下図の通り（右図は、**正方連結四数**の例）。



2. 「**正方連結型**」(あるいは「**相結陣**」)の立体四方陣では、前後・上下・縦横の各平面上のどの2×2の**正方4数**を取っても、その和が一定になる。そう要請される。

具体的には、左端の第1面で

$$\begin{aligned} n_1 + n_2 + n_5 + n_6 &= K \dots\dots (1) \\ n_2 + n_3 + n_6 + n_7 &= K \dots\dots (2) \\ n_3 + n_4 + n_7 + n_8 &= K \dots\dots (3) \\ n_5 + n_6 + n_9 + n_{10} &= K \dots\dots (4) \\ n_6 + n_7 + n_{10} + n_{11} &= K \dots\dots (5) \\ n_7 + n_8 + n_{11} + n_{12} &= K \dots\dots (6) \\ n_9 + n_{10} + n_{13} + n_{14} &= K \dots\dots (7) \\ n_{10} + n_{11} + n_{14} + n_{15} &= K \dots\dots (8) \\ n_{11} + n_{12} + n_{15} + n_{16} &= K \dots\dots (9) \end{aligned}$$

第2面で、

$$\begin{aligned} n_{17} + n_{18} + n_{21} + n_{22} &= K \dots\dots (10) \\ n_{18} + n_{19} + n_{22} + n_{23} &= K \dots\dots (11) \\ n_{19} + n_{20} + n_{23} + n_{24} &= K \dots\dots (12) \\ n_{21} + n_{22} + n_{25} + n_{26} &= K \dots\dots (13) \\ n_{22} + n_{23} + n_{26} + n_{27} &= K \dots\dots (14) \\ n_{23} + n_{24} + n_{27} + n_{28} &= K \dots\dots (15) \\ n_{25} + n_{26} + n_{29} + n_{30} &= K \dots\dots (16) \\ n_{26} + n_{27} + n_{30} + n_{31} &= K \dots\dots (17) \\ n_{27} + n_{28} + n_{31} + n_{32} &= K \dots\dots (18) \end{aligned}$$

第3面で、

$$\begin{aligned} n_{33} + n_{34} + n_{37} + n_{38} &= K \dots\dots (19) \\ n_{34} + n_{35} + n_{38} + n_{39} &= K \dots\dots (20) \\ n_{35} + n_{36} + n_{39} + n_{40} &= K \dots\dots (21) \\ n_{37} + n_{38} + n_{41} + n_{42} &= K \dots\dots (22) \\ n_{38} + n_{39} + n_{42} + n_{43} &= K \dots\dots (23) \\ n_{39} + n_{40} + n_{43} + n_{44} &= K \dots\dots (24) \\ n_{41} + n_{42} + n_{45} + n_{46} &= K \dots\dots (25) \\ n_{42} + n_{43} + n_{46} + n_{47} &= K \dots\dots (26) \\ n_{43} + n_{44} + n_{47} + n_{48} &= K \dots\dots (27) \end{aligned}$$

第4面で、

$$\begin{aligned} n_{49} + n_{50} + n_{53} + n_{54} &= K \dots\dots (28) \\ n_{50} + n_{51} + n_{54} + n_{55} &= K \dots\dots (29) \\ n_{51} + n_{52} + n_{55} + n_{56} &= K \dots\dots (30) \\ n_{53} + n_{54} + n_{57} + n_{58} &= K \dots\dots (31) \\ n_{54} + n_{55} + n_{58} + n_{59} &= K \dots\dots (32) \\ n_{55} + n_{56} + n_{59} + n_{60} &= K \dots\dots (33) \\ n_{57} + n_{58} + n_{61} + n_{62} &= K \dots\dots (34) \\ n_{58} + n_{59} + n_{62} + n_{63} &= K \dots\dots (35) \\ n_{59} + n_{60} + n_{63} + n_{64} &= K \dots\dots (36) \end{aligned}$$

同様に前後方向の第1面で、

$$\begin{aligned} n_1 + n_{17} + n_5 + n_{21} &= K \dots\dots (37) \\ n_{17} + n_{33} + n_{21} + n_{37} &= K \dots\dots (38) \\ n_{33} + n_{49} + n_{37} + n_{53} &= K \dots\dots (39) \\ n_5 + n_{21} + n_9 + n_{25} &= K \dots\dots (40) \\ n_{21} + n_{37} + n_{25} + n_{41} &= K \dots\dots (41) \\ n_{37} + n_{53} + n_{41} + n_{57} &= K \dots\dots (42) \\ n_9 + n_{25} + n_{13} + n_{29} &= K \dots\dots (43) \end{aligned}$$

第2面で、

$$\begin{aligned} n_2 + n_{18} + n_6 + n_{22} &= K \dots\dots (46) \\ n_{18} + n_{34} + n_{22} + n_{38} &= K \dots\dots (47) \\ n_{34} + n_{50} + n_{38} + n_{54} &= K \dots\dots (48) \\ n_6 + n_{22} + n_{10} + n_{26} &= K \dots\dots (49) \\ n_{22} + n_{38} + n_{26} + n_{42} &= K \dots\dots (50) \\ n_{38} + n_{54} + n_{42} + n_{58} &= K \dots\dots (51) \\ n_{10} + n_{26} + n_{14} + n_{30} &= K \dots\dots (52) \end{aligned}$$

$$n_{25}+n_{41}+n_{29}+n_{45}=K \dots\dots (44) \quad n_{26}+n_{42}+n_{30}+n_{46}=K \dots\dots (53)$$

$$n_{41}+n_{57}+n_{45}+n_{61}=K \dots\dots (45) \quad n_{42}+n_{58}+n_{46}+n_{62}=K \dots\dots (54)$$

第3面で,

$$n_3+n_{19}+n_7+n_{23}=K \dots\dots (55)$$

$$n_{19}+n_{35}+n_{23}+n_{39}=K \dots\dots (56)$$

$$n_{35}+n_{51}+n_{39}+n_{55}=K \dots\dots (57)$$

$$n_7+n_{23}+n_{11}+n_{27}=K \dots\dots (58)$$

$$n_{23}+n_{39}+n_{27}+n_{43}=K \dots\dots (59)$$

$$n_{39}+n_{55}+n_{43}+n_{59}=K \dots\dots (60)$$

$$n_{11}+n_{27}+n_{15}+n_{31}=K \dots\dots (61)$$

$$n_{27}+n_{43}+n_{31}+n_{47}=K \dots\dots (62)$$

$$n_{43}+n_{59}+n_{47}+n_{63}=K \dots\dots (63)$$

第4面で,

$$n_4+n_{20}+n_8+n_{24}=K \dots\dots (64)$$

$$n_{20}+n_{36}+n_{24}+n_{40}=K \dots\dots (65)$$

$$n_{36}+n_{52}+n_{40}+n_{56}=K \dots\dots (66)$$

$$n_8+n_{24}+n_{12}+n_{28}=K \dots\dots (67)$$

$$n_{24}+n_{40}+n_{28}+n_{44}=K \dots\dots (68)$$

$$n_{40}+n_{56}+n_{44}+n_{60}=K \dots\dots (69)$$

$$n_{12}+n_{28}+n_{16}+n_{32}=K \dots\dots (70)$$

$$n_{28}+n_{44}+n_{32}+n_{48}=K \dots\dots (71)$$

$$n_{44}+n_{60}+n_{48}+n_{64}=K \dots\dots (72)$$

同様に上下方向の第1面で,

$$n_1+n_{17}+n_2+n_{18}=K \dots\dots (73)$$

$$n_{17}+n_{33}+n_{18}+n_{34}=K \dots\dots (74)$$

$$n_{33}+n_{49}+n_{34}+n_{50}=K \dots\dots (75)$$

$$n_2+n_{18}+n_3+n_{19}=K \dots\dots (76)$$

$$n_{18}+n_{34}+n_{19}+n_{35}=K \dots\dots (77)$$

$$n_{34}+n_{50}+n_{35}+n_{51}=K \dots\dots (78)$$

$$n_3+n_{19}+n_4+n_{20}=K \dots\dots (79)$$

$$n_{19}+n_{35}+n_{20}+n_{36}=K \dots\dots (80)$$

$$n_{35}+n_{51}+n_{36}+n_{52}=K \dots\dots (81)$$

第2面で,

$$n_5+n_{21}+n_6+n_{22}=K \dots\dots (82)$$

$$n_{21}+n_{37}+n_{22}+n_{38}=K \dots\dots (83)$$

$$n_{37}+n_{53}+n_{38}+n_{54}=K \dots\dots (84)$$

$$n_6+n_{22}+n_7+n_{23}=K \dots\dots (85)$$

$$n_{22}+n_{38}+n_{23}+n_{39}=K \dots\dots (86)$$

$$n_{38}+n_{54}+n_{39}+n_{55}=K \dots\dots (87)$$

$$n_7+n_{23}+n_8+n_{24}=K \dots\dots (88)$$

$$n_{23}+n_{39}+n_{24}+n_{40}=K \dots\dots (89)$$

$$n_{39}+n_{55}+n_{40}+n_{56}=K \dots\dots (90)$$

第3面で,

$$n_9+n_{25}+n_{10}+n_{26}=K \dots\dots (91)$$

$$n_{25}+n_{41}+n_{11}+n_{27}=K \dots\dots (92)$$

$$n_{41}+n_{57}+n_{27}+n_{43}=K \dots\dots (93)$$

$$n_{10}+n_{26}+n_{11}+n_{27}=K \dots\dots (94)$$

$$n_{26}+n_{42}+n_{27}+n_{43}=K \dots\dots (95)$$

$$n_{42}+n_{58}+n_{43}+n_{59}=K \dots\dots (96)$$

$$n_{11}+n_{27}+n_{12}+n_{28}=K \dots\dots (97)$$

$$n_{27}+n_{43}+n_{28}+n_{44}=K \dots\dots (98)$$

$$n_{43}+n_{59}+n_{44}+n_{60}=K \dots\dots (99)$$

第4面で,

$$n_{13}+n_{29}+n_{14}+n_{30}=K \dots\dots (100)$$

$$n_{29}+n_{45}+n_{24}+n_{40}=K \dots\dots (101)$$

$$n_{45}+n_{61}+n_{46}+n_{31}=K \dots\dots (102)$$

$$n_{14}+n_{30}+n_{15}+n_{28}=K \dots\dots (103)$$

$$n_{30}+n_{46}+n_{31}+n_{47}=K \dots\dots (104)$$

$$n_{46}+n_{62}+n_{47}+n_{63}=K \dots\dots (105)$$

$$n_{15}+n_{31}+n_{16}+n_{32}=K \dots\dots (106)$$

$$n_{31}+n_{47}+n_{32}+n_{48}=K \dots\dots (107)$$

$$n_{47}+n_{63}+n_{48}+n_{64}=K \dots\dots (108)$$

この 108個の正方連結四数の定和条件だけで, どのような方陣が作れるだろうか。

3. 上の各式を相互に関わらせよう。例えば, (1)~(36)の中から次の16式を選んで辺々を加えると,

$$n_1+n_2+n_5+n_6=K \dots\dots (1)$$

$$n_3+n_4+n_7+n_8=K \dots\dots (3)$$

$$n_9+n_{10}+n_{13}+n_{14}=K \dots\dots (7)$$

$$n_{11}+n_{12}+n_{15}+n_{16}=K \dots\dots (9)$$

$$n_{17}+n_{18}+n_{21}+n_{22}=K \dots\dots (10)$$

$$n_{19}+n_{20}+n_{23}+n_{24}=K \dots\dots (12)$$

$$n_{25}+n_{26}+n_{29}+n_{30}=K \dots\dots (16)$$

$$n_{27}+n_{28}+n_{31}+n_{32}=K \dots\dots (18)$$

$$n_{33}+n_{34}+n_{37}+n_{38}=K \dots\dots (19)$$

$$n_{35}+n_{36}+n_{39}+n_{40}=K \dots\dots (21)$$

$$n_{41}+n_{42}+n_{45}+n_{46}=K \dots\dots (25)$$

$$n_{43}+n_{44}+n_{47}+n_{48}=K \dots\dots (27)$$

$$n_{49}+n_{50}+n_{53}+n_{54}=K \dots\dots (28)$$

$$n_{51}+n_{52}+n_{55}+n_{56}=K \dots\dots (30)$$

$$n_{57}+n_{58}+n_{61}+n_{62}=K \dots\dots (34)$$

$$n_{59}+n_{60}+n_{63}+n_{64}=K \dots\dots (36)$$

$$\{n_1+n_2+n_3+n_4+\dots+n_{63}+n_{64}\}=16*K$$

この式の左辺には1~64の自然数列が全て入るのであるから, その総和は

$$64*(64+1) \div 2 = 16*K$$

$$2080 = 16*K$$

$$\therefore K = 130 \dots\dots (109)$$

定和は, 一般的な立体四方陣に共通の 130 と決まった。

一方,

$$n_1+n_2+n_5+n_6=K \dots\dots (1)$$

$$n_2+n_3+n_6+n_7=K \dots\dots (2)$$

$$n_3+n_4+n_7+n_8=K \dots\dots (3)$$

$$n_5+n_6+n_9+n_{10}=K \dots\dots (4)$$

$$n_6+n_7+n_{10}+n_{11}=K \dots\dots (5)$$

$$n_7+n_8+n_{11}+n_{12}=K \dots\dots (6)$$

$$n_9+n_{10}+n_{13}+n_{14}=K \dots\dots (7)$$

$$n_{10}+n_{11}+n_{14}+n_{15}=K \dots\dots (8)$$

$$n_{11}+n_{12}+n_{15}+n_{16}=K \dots\dots (9)$$

(1)+(3)の内容を組み替えると,

$$\{n1+n5+n4+n8\} + \{n2+n3+n6+n7\} = 2*K$$

ところが(2)式によって, $\{n1+n5+n4+n8\} + K = 2*K$
 $\therefore \{n1+n5+n4+n8\} = K \dots\dots (110)$

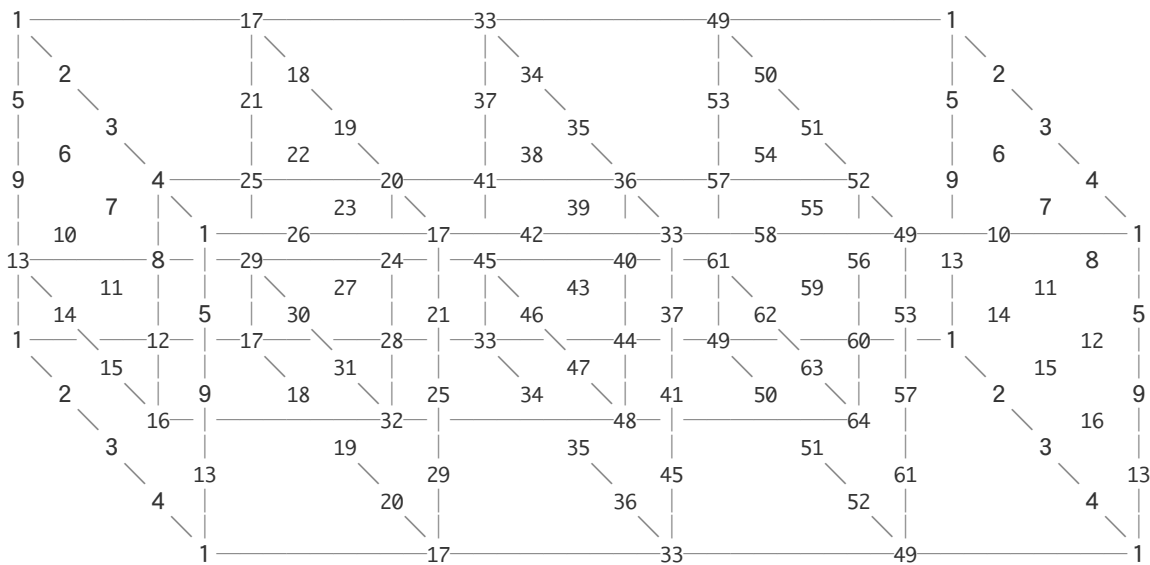
同様に(4)+(6)の内容を組み替えると,

$$\{n5+n9+n8+n12\} + \{n6+n7+n10+n11\} = 2*K$$

ところが(5)式によって, $\{n5+n9+n8+n12\} + K = 2*K$
 $\therefore \{n5+n9+n8+n12\} = K \dots\dots (111)$

同様にして(7)~(9)から $\{n9+n13+n12+n16\} = K \dots\dots (112)$

(110)~(112)式は, 何を意味するだろうか。一番奥の $\{n1+n5+n9+n13\}$ の1列を一番手前に持って来ても, 正方4数の和はKに等しいということだ。同様の手順を何度も繰り返せば, ついには奥の面全体を一番手前に持って来ることができることになる。前後方向ばかりでなく上下・左右にも同じことが行える。つまり, 下図のように立体方陣空間を幾らでも拡張できるわけで, 正方連結型四方陣は汎四方陣の性質を分有している。



$$\begin{aligned} n1+n2+n5+n6 &= K \dots\dots (1) & n17+n18+n21+n22 &= K \dots\dots (10) \\ n3+n4+n7+n8 &= K \dots\dots (3) & n19+n20+n23+n24 &= K \dots\dots (12) \\ n9+n10+n13+n14 &= K \dots\dots (7) & n25+n26+n29+n30 &= K \dots\dots (16) \\ n11+n12+n15+n16 &= K \dots\dots (9) & n27+n28+n31+n32 &= K \dots\dots (18) \end{aligned}$$

(1)+(3)+(7)+(9)

$$\{n1+n4+n13+n16\} + \{(n2+n3+n6+n7) + (n5+n9+n8+n12) + (n10+n11+n14+n15)\} = 4*K$$

ところが(2), (111), (8)式から,

$$\{n1+n4+n13+n16\} + \{K+K+K\} = 4*K$$

$$\therefore \{n1+n4+n13+n16\} = K \dots\dots (113)$$

同様にして $\{n1+n4+n49+n52\} = K \dots\dots (114)$
 $\{n1+n13+n49+n61\} = K \dots\dots (115)$

これは, 立体の頂点上の大正方4数の和も 130 に等しいことを表すだけでなく, 上図の拡張空間で右下手前の小正方4数の和が 130 に等しいことを表している。

また $\{n17+n20+n29+n32\} = K \dots\dots (116)$
 $\{n49+n52+n61+n64\} = K \dots\dots (117)$
 $\{n13+n16+n61+n64\} = K \dots\dots (118)$

なども言える。

だが, 各平面上の行列の和が必ずしも 130 に等しいとは限らない。上の 108個の条件だけでは, 例えば $n1+n2+n3+n4=K$ を導き出せないし, $n1+n5+n9+n13=K$ も言えない。たまたまそうなることはあるが, いつも必ずそうなるとは限らない。

4. ところで, 正方連結型立方体四方陣では,

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

$$n_1 + n_2 + n_5 + n_6 = K \dots\dots (1)$$

$$n_5 + n_6 + n_9 + n_{10} = K \dots\dots (4)$$

$$n_9 + n_{10} + n_{13} + n_{14} = K \dots\dots (7)$$

ゆえに

$$n_1 + n_2 = n_9 + n_{10} (=L \text{ とおく})$$

$$n_5 + n_6 = n_{13} + n_{14} (=M \text{ とおく})$$

と言える。

同様にして

$$n_1 + n_2 = n_{33} + n_{34} = n_9 + n_{10} = L$$

$$n_{17} + n_{18} = n_{49} + n_{50} = n_5 + n_6 = M$$

前後方向に全てを調べ整理すると、

$$n_1 + n_2 = n_9 + n_{10} = n_{21} + n_{22} = n_{29} + n_{30} = n_{33} + n_{34} = n_{41} + n_{42} = n_{53} + n_{54} = n_{61} + n_{62} = L$$

$$n_3 + n_4 = n_{11} + n_{12} = n_{23} + n_{24} = n_{31} + n_{32} = n_{35} + n_{36} = n_{43} + n_{44} = n_{55} + n_{56} = n_{63} + n_{64} = N$$

$$n_5 + n_6 = n_{13} + n_{14} = n_{17} + n_{18} = n_{25} + n_{26} = n_{37} + n_{38} = n_{45} + n_{46} = n_{49} + n_{50} = n_{57} + n_{58} = M$$

$$n_7 + n_8 = n_{15} + n_{16} = n_{19} + n_{20} = n_{27} + n_{28} = n_{39} + n_{40} = n_{47} + n_{48} = n_{51} + n_{52} = n_{59} + n_{60} = P$$

(1), (3)から、 $L + M = K$, $N + P = K$

そして、例えば、 $\{n_1 + n_2 + n_3 + n_4\} = \{n_9 + n_{10} + n_{11} + n_{12}\} = L + N \dots\dots (119)$

は言えるが、 $L + N = K$, $N = M$ とは必ずしも言えない。

同様に $\{n_5 + n_6 + n_7 + n_8\} = \{n_{13} + n_{14} + n_{15} + n_{16}\} = M + P \dots\dots (120)$

ではあるが、 $M + P = K$, $L = P$ とは必ずしも言えない。

(119)および(120)式は、各面の第1行列と第3行列が交換可能。第2行列と第4行列も、交換可能であることを示している。三次元に拡張すれば、どの方向でも、第1面と第3面の交換が可能であり、第2面と第4面の交換も可能である。これは、汎立体方陣と同じ性質である。

だが、この正方連結条件だけでは、肝心の**行列柱の定和**が依然言えない。

5. これは本来選択の問題だが、行列柱の定和が成り立たないのでは、「魔方陣」にならず、たとえ作っても、らしくなくて、満足できない。

そこで、目標を高くして、親和性のある汎立体四方陣の中から、正方連結型のものをさがすことにしよう。つまり、**行列柱の定和**が成り立ち、かつ**汎対角線の定和**も成り立つ。その上で正方連結型の立方体四方陣を求めることにしよう。条件は、いやが上にも多くなる。果たして可能なのかどうかさえわからないが、あえて探究してみよう。

付け加える条件は、先ず次の3個だ。正方連結型では、この3個があればよい。

$$n_1 + n_2 + n_3 + n_4 = K; \quad n_1 + n_5 + n_9 + n_{13} = K; \quad n_1 + n_{17} + n_{33} + n_{49} = K;$$

正方格子を通してこの条件が隅々にまで行き渡り、全ての行列柱の定和が成り立つ。

次の4式も付け加えよう。

$$n_1 + n_{22} + n_{43} + n_{64} = K; \quad n_1 + n_{24} + n_{43} + n_{62} = K;$$

$$n_1 + n_{30} + n_{43} + n_{56} = K; \quad n_1 + n_{32} + n_{43} + n_{54} = K;$$

この性質も正方格子を伝わって方陣の隅々にまで行き渡り、全ての汎対角線が定和になる。

1つだけ証明しておこう。

$$n_1 + n_4 = n_{21} + n_{24} = n_{41} + n_{44} = n_{61} + n_{64} = Q$$

$$n_2 + n_3 = n_{22} + n_{23} = n_{42} + n_{43} = n_{62} + n_{63} = R$$

ところが、 $n_1 + n_2 + n_3 + n_4 = K$ と新たに仮定した。ゆえに $Q + R = K$

だから $n_1 + n_4 + n_{22} + n_{23} = K$

$$n_{42} + n_{43} + n_{61} + n_{64} = K$$

辺々を加えると、 $\{n_1 + n_{22} + n_{43} + n_{64}\} + \{n_4 + n_{23} + n_{43} + n_{61}\} = 2 * K$

ゆえに $n_1 + n_{22} + n_{43} + n_{64} = K$ ならば、 $n_4 + n_{23} + n_{43} + n_{61} = K \dots\dots (121)$

かくしてパーフェクトになる（なぜそう言えるのかは、ひとえに伝統的美意識の問題）。

6. 上の定義のもとで、解を求めよう。普通の方法によって。

早速プログラムを次のように書いた（英語版のままで申し訳ない）。そして実行した。

```

/** 'Composite & Pan-triagonal' Magic Cubes of Order 4 **
/** 'CPMC444.c': Dictated by Kanji Setsuda **
/** in '02,'07,'12 and Revised on Sep. 9, 2014 **
/** WorKing with MacOSX 10.9.4 and Xcode 5.1.1 **
//
#include <stdio.h>
//
// Global Var.
short int cnt, cnt2;
short LSM, cnt3, typ;
short nm[65], uflg[65];
short nt[7][65];
//
// Functions(Sub-Routines)
void stp00(void);
void stp01(void), stp02(void), stp03(void), stp04(void);
void stp05(void), stp06(void), stp07(void), stp08(void);
void stp09(void), stp10(void), stp11(void), stp12(void);
void stp13(void), stp14(void), stp15(void), stp16(void);
void stp17(void), stp18(void), stp19(void), stp20(void);
void stp21(void), stp22(void), stp23(void), stp24(void);
void 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);
void ansprint(void);
void prans(short x), pradj6(short x);
//
/** Main Program *
int main(){
short n;
printf("\n"); typ=0; // typ=0: normal; typ=1: 6 Adjacent Numbers next to n1
printf("** 'Composite & Pan-triagonal' Magic Cubes of Order 4 **\n");
if(typ==0){printf(" ** List of Standard Solutions **\n");}
else{printf(" ** List of 6 Adjacent Numbers next to n1 and [n43] **\n");}
LSM=130; cnt=0; cnt3=0;
for(n=1;n<65;n++){nm[n]=0; uflg[n]=0;};
stp00(); // Begin the Calculations
if(cnt3>0){
if(typ==0){prans(cnt3);}
else{pradj6(cnt3);}}
printf(" [Solution Counts = %d] [OK!]\n",cnt);
printf("\n");
printf("** Calculated and Listed by Kanji Setsuda on\n");
printf(" Sep. 9, 2014 with MacOSX 10.9.4 & Xcode 5.1.1 **\n");
printf("\n");
return 0;
}
//
/** Begin the Calculations *
//Set n1
void stp00(){
short a;
for(a=1;a<33;a++){cnt2=0;
if(uflg[a]==0){
nm[1]=a; uflg[a]=1;
stp01();
uflg[a]=0;}}
}
//Set n2
void stp01(){
short a;
for(a=64;a>0;a--){
if(uflg[a]==0){if(nm[1]<3){cnt2=0;}}
nm[2]=a; uflg[a]=1;
}
}
}

```



```

a=LSM-nm[11]-nm[12]-nm[15];
if((nm[1]<a)&&(a<65)){
    if(uflg[a]==0){nm[16]=a;
        uflg[a]=1; stp16();
        uflg[a]=0;}}
}
//Check n12+n16+n9+n13?=LSM
void stp16(){
    short sm1,sm2,sm3;
    sm1=nm[12]+nm[16]+nm[9]+nm[13];
    sm2=nm[15]+nm[16]+nm[3]+nm[4];
    sm3=nm[16]+nm[13]+nm[4]+nm[1];
    if((sm1==LSM)&&(sm2==LSM)&&(sm3==LSM)){
        stp17();}
}
// Search Level #2:
//Set n17
void stp17(){
    short a;
    for(a=64;a>0;a--){
        if(uflg[a]==0){
            nm[17]=a; uflg[a]=1;
            stp18();
            uflg[a]=0;}}
}
//Set n18=LSM-n1-n2-n17
void stp18(){
    short a;
    a=LSM-nm[1]-nm[2]-nm[17];
    if((0<a)&&(a<65)){
        if(uflg[a]==0){nm[18]=a;
            uflg[a]=1; stp19();
            uflg[a]=0;}}
}
//Set n19=LSM-n2-n3-n18
void stp19(){
    short a;
    a=LSM-nm[2]-nm[3]-nm[18];
    if((0<a)&&(a<65)){
        if(uflg[a]==0){nm[19]=a;
            uflg[a]=1; stp20();
            uflg[a]=0;}}
}
//Set n20=LSM-n3-n4-n19
void stp20(){
    short a,b;
    a=LSM-nm[3]-nm[4]-nm[19];
    b=LSM-nm[1]-nm[4]-nm[17];
    if((0<a)&&(a<65)&&(a==b)){
        if(uflg[a]==0){nm[20]=a;
            uflg[a]=1; stp21();
            uflg[a]=0;}}
}
//Set n21=LSM-n1-n5-n17
void stp21(){
    short a;
    a=LSM-nm[1]-nm[5]-nm[17];
    if((0<a)&&(a<65)){
        if(uflg[a]==0){nm[21]=a;
            uflg[a]=1; stp22();
            uflg[a]=0;}}
}
//Set n22=LSM-n2-n6-n18
void stp22(){

```

```

short a,b,c;
a=LSM-nm[17]-nm[18]-nm[21];
b=LSM-nm[5]-nm[6]-nm[21];
c=LSM-nm[2]-nm[6]-nm[18];
if((0<a)&&(a<65)&&(a==b)&&(a==c)){
    if(uflg[a]==0){nm[22]=a;
        uflg[a]=1; stp23();
        uflg[a]=0;}}
}
//Set n23=LSM-n3-n7-n19
void stp23(){
    short a,b,c;
    a=LSM-nm[18]-nm[19]-nm[22];
    b=LSM-nm[6]-nm[7]-nm[22];
    c=LSM-nm[3]-nm[7]-nm[19];
    if((0<a)&&(a<65)&&(a==b)&&(a==c)){
        if(uflg[a]==0){nm[23]=a;
            uflg[a]=1; stp24();
            uflg[a]=0;}}
}
//Set n24=LSM-n4-n8-n20
void stp24(){
    short a,b,c;
    a=LSM-nm[19]-nm[20]-nm[23];
    b=LSM-nm[7]-nm[8]-nm[23];
    c=LSM-nm[4]-nm[8]-nm[20];
    if((0<a)&&(a<65)&&(a==b)&&(a==c)){
        if(uflg[a]==0){nm[24]=a;
            uflg[a]=1; stp25();
            uflg[a]=0;}}
}
//Set n25=LSM-n5-n9-n21
void stp25(){
    short a;
    a=LSM-nm[5]-nm[9]-nm[21];
    if((0<a)&&(a<65)){
        if(uflg[a]==0){nm[25]=a;
            uflg[a]=1; stp26();
            uflg[a]=0;}}
}
//Set n26=LSM-n6-n10-n22
void stp26(){
    short a,b,c;
    a=LSM-nm[21]-nm[22]-nm[25];
    b=LSM-nm[9]-nm[10]-nm[25];
    c=LSM-nm[6]-nm[10]-nm[22];
    if((0<a)&&(a<65)&&(a==b)&&(a==c)){
        if(uflg[a]==0){nm[26]=a;
            uflg[a]=1; stp27();
            uflg[a]=0;}}
}
//Set n27=LSM-n7-n11-n23
void stp27(){
    short a,b,c;
    a=LSM-nm[22]-nm[23]-nm[26];
    b=LSM-nm[10]-nm[11]-nm[26];
    c=LSM-nm[7]-nm[11]-nm[23];
    if((0<a)&&(a<65)&&(a==b)&&(a==c)){
        if(uflg[a]==0){nm[27]=a;
            uflg[a]=1; stp28();
            uflg[a]=0;}}
}
//Set n28=LSM-n8-n12-n24
void stp28(){
    short a,b,c;

```

```

a=LSM-nm[23]-nm[24]-nm[27];
b=LSM-nm[11]-nm[12]-nm[27];
c=LSM-nm[8]-nm[12]-nm[24];
if((0<a)&&(a<65)&&(a==b)&&(a==c)){
    if(uflg[a]==0){nm[28]=a;
        uflg[a]=1; stp29();
        uflg[a]=0;}}
}
//Set n29=LSM-n9-n13-n25
void stp29(){
    short a,b;
    a=LSM-nm[9]-nm[13]-nm[25];
    b=LSM-nm[13]-nm[1]-nm[17];
    if((0<a)&&(a<65)&&(a==b)){
        if(uflg[a]==0){nm[29]=a;
            uflg[a]=1; stp30();
            uflg[a]=0;}}
}
//Set n30=LSM-n10-n14-n26
void stp30(){
    short a,b,c;
    a=LSM-nm[25]-nm[26]-nm[29];
    b=LSM-nm[13]-nm[14]-nm[29];
    c=LSM-nm[10]-nm[14]-nm[26];
    if((0<a)&&(a<65)&&(a==b)&&(a==c)){
        if(uflg[a]==0){nm[30]=a;
            uflg[a]=1; stp31();
            uflg[a]=0;}}
}
//Set n31=LSM-n11-n15-n27
void stp31(){
    short a,b,c;
    a=LSM-nm[26]-nm[27]-nm[30];
    b=LSM-nm[14]-nm[15]-nm[30];
    c=LSM-nm[11]-nm[15]-nm[27];
    if((0<a)&&(a<65)&&(a==b)&&(a==c)){
        if(uflg[a]==0){nm[31]=a;
            uflg[a]=1; stp32();
            uflg[a]=0;}}
}
//Set n32=LSM-n12-n16-n28
void stp32(){
    short a,b,c;
    a=LSM-nm[27]-nm[28]-nm[31];
    b=LSM-nm[15]-nm[16]-nm[31];
    c=LSM-nm[12]-nm[16]-nm[28];
    if((0<a)&&(a<65)&&(a==b)&&(a==c)){
        if(uflg[a]==0){nm[32]=a;
            uflg[a]=1; stp33();
            uflg[a]=0;}}
}
//Check Others
void stp33(){
    short sm1,sm2,sm3;
    sm1=nm[25]+nm[29]+nm[28]+nm[32];
    sm2=nm[13]+nm[29]+nm[16]+nm[32];
    sm3=nm[20]+nm[24]+nm[28]+nm[32];
    if((sm1==LSM)&&(sm2==LSM)&&(sm3==LSM)){
        stp34();}
}
// Search Level #3:
//Set n49(<n13)
void stp34(){
    short a;
    for(a=nm[13]-1;a>0;a--){

```

```

        if(uflg[a]==0){
            nm[49]=a; uflg[a]=1;
            stp35();
            uflg[a]=0;}}
    }
}
//Set n33=LSM-n1-n17-n49
void stp35(){
    short a;
    a=LSM-nm[1]-nm[17]-nm[49];
    if((0<a)&&(a<65)){
        if(uflg[a]==0){nm[33]=a;
            uflg[a]=1; stp36();
            uflg[a]=0;}}
}
//Set n34=LSM-n17-n18-n33
void stp36(){
    short a;
    a=LSM-nm[17]-nm[18]-nm[33];
    if((0<a)&&(a<65)){
        if(uflg[a]==0){nm[34]=a;
            uflg[a]=1; stp37();
            uflg[a]=0;}}
}
//Set n37=LSM-n17-n21-n33
void stp37(){
    short a;
    a=LSM-nm[17]-nm[21]-nm[33];
    if((0<a)&&(a<65)){
        if(uflg[a]==0){nm[37]=a;
            uflg[a]=1; stp38();
            uflg[a]=0;}}
}
//Set n38=LSM-n18-n22-n34
void stp38(){
    short a,b,c,d;
    a=LSM-nm[33]-nm[34]-nm[37];
    b=LSM-nm[21]-nm[22]-nm[37];
    c=LSM-nm[18]-nm[22]-nm[34];
    d=LSM-nm[16]-nm[27]-nm[49];
    if((0<a)&&(a<65)&&(a==b)&&(a==c)&&(a==d)){
        if(uflg[a]==0){nm[38]=a;
            uflg[a]=1; stp39();
            uflg[a]=0;}}
}
//Set n35=LSM-n18-n19-n34
void stp39(){
    short a;
    a=LSM-nm[18]-nm[19]-nm[34];
    if((0<a)&&(a<65)){
        if(uflg[a]==0){nm[35]=a;
            uflg[a]=1; stp40();
            uflg[a]=0;}}
}
//Set n36=LSM-n19-n20-n35
void stp40(){
    short a;
    a=LSM-nm[19]-nm[20]-nm[35];
    if((0<a)&&(a<65)){
        if(uflg[a]==0){nm[36]=a;
            uflg[a]=1; stp41();
            uflg[a]=0;}}
}
//Set n41=LSM-n21-n25-n37
void stp41(){

```



```

void stp52(){
short a;
a=LSM-nm[34]-nm[35]-nm[50];
if((0<a)&&(a<65)){
if(uflg[a]==0){nm[51]=a;
uflg[a]=1; stp53();
uflg[a]=0;}}
}
//Set n52=LSM-n35-n36-n51 & n1<n52
void stp53(){
short a,b,c,d;
a=LSM-nm[35]-nm[36]-nm[51];
b=LSM-nm[49]-nm[50]-nm[51];
c=LSM-nm[4]-nm[20]-nm[36];
d=LSM-nm[13]-nm[26]-nm[39];
if((nm[1]<a)&&(a<65)&&(a==b)&&(a==c)&&(a==d)){
if(uflg[a]==0){nm[52]=a;
uflg[a]=1; stp54();
uflg[a]=0;}}
}
//Set n53=LSM-n33-n37-n49
void stp54(){
short a;
a=LSM-nm[33]-nm[37]-nm[49];
if((0<a)&&(a<65)){
if(uflg[a]==0){nm[53]=a;
uflg[a]=1; stp55();
uflg[a]=0;}}
}
//Set n54=LSM-n34-n38-n50
void stp55(){
short a,b,c,d;
a=LSM-nm[49]-nm[50]-nm[53];
b=LSM-nm[37]-nm[38]-nm[53];
c=LSM-nm[34]-nm[38]-nm[50];
d=LSM-nm[6]-nm[22]-nm[38];
if((0<a)&&(a<65)&&(a==b)&&(a==c)&&(a==d)){
if(uflg[a]==0){nm[54]=a;
uflg[a]=1; stp56();
uflg[a]=0;}}
}
//Set n55=LSM-n35-n39-n51
void stp56(){
short a,b,c,d;
a=LSM-nm[50]-nm[51]-nm[54];
b=LSM-nm[38]-nm[39]-nm[54];
c=LSM-nm[35]-nm[39]-nm[51];
d=LSM-nm[7]-nm[23]-nm[39];
if((0<a)&&(a<65)&&(a==b)&&(a==c)&&(a==d)){
if(uflg[a]==0){nm[55]=a;
uflg[a]=1; stp57();
uflg[a]=0;}}
}
//Set n56=LSM-n36-n40-n52
void stp57(){
short a,b,c,d;
a=LSM-nm[51]-nm[52]-nm[55];
b=LSM-nm[39]-nm[40]-nm[55];
c=LSM-nm[36]-nm[40]-nm[52];
d=LSM-nm[1]-nm[30]-nm[43];
if((0<a)&&(a<65)&&(a==b)&&(a==c)&&(a==d)){
if(uflg[a]==0){nm[56]=a;
uflg[a]=1; stp58();
uflg[a]=0;}}
}
}

```

```

//Set n57=LSM-n37-n41-n53
void stp58(){
short a,b,c;
a=LSM-nm[37]-nm[41]-nm[53];
b=LSM-nm[16]-nm[19]-nm[38];
c=LSM-nm[14]-nm[19]-nm[40];
if((0<a)&&(a<65)&&(a==b)&&(a==c)){
if(uflg[a]==0){nm[57]=a;
uflg[a]=1; stp59();
uflg[a]=0;}}
}
//Set n58=LSM-n38-n42-n54
void stp59(){
short a,b,c,d;
a=LSM-nm[53]-nm[54]-nm[57];
b=LSM-nm[41]-nm[42]-nm[57];
c=LSM-nm[38]-nm[42]-nm[54];
d=LSM-nm[13]-nm[20]-nm[39];
if((0<a)&&(a<65)&&(a==b)&&(a==c)&&(a==d)){
if(uflg[a]==0){nm[58]=a;
uflg[a]=1; stp60();
uflg[a]=0;}}
}
//Set n59=LSM-n39-n43-n55
void stp60(){
short a,b,c,d;
a=LSM-nm[54]-nm[55]-nm[58];
b=LSM-nm[42]-nm[43]-nm[58];
c=LSM-nm[39]-nm[43]-nm[55];
d=LSM-nm[16]-nm[17]-nm[38];
if((0<a)&&(a<65)&&(a==b)&&(a==c)&&(a==d)){
if(uflg[a]==0){nm[59]=a;
uflg[a]=1; stp61();
uflg[a]=0;}}
}
//Set n60=LSM-n40-n44-n56
void stp61(){
short a,b,c,d;
a=LSM-nm[55]-nm[56]-nm[59];
b=LSM-nm[43]-nm[44]-nm[59];
c=LSM-nm[40]-nm[44]-nm[56];
d=LSM-nm[13]-nm[18]-nm[39];
if((0<a)&&(a<65)&&(a==b)&&(a==c)&&(a==d)){
if(uflg[a]==0){nm[60]=a;
uflg[a]=1; stp62();
uflg[a]=0;}}
}
//Set n61=LSM-n41-n45-n57 & n1<n61
void stp62(){
short a,b,c,d;
a=LSM-nm[41]-nm[45]-nm[57];
b=LSM-nm[4]-nm[23]-nm[42];
c=LSM-nm[2]-nm[23]-nm[44];
d=LSM-nm[12]-nm[23]-nm[34];
if((nm[1]<a)&&(a<65)&&(a==b)&&(a==c)&&(a==d)){
if(uflg[a]==0){nm[61]=a;
uflg[a]=1; stp63();
uflg[a]=0;}}
}
//Set n62=LSM-n42-n46-n58
void stp63(){
short a,b,c,d;
a=LSM-nm[57]-nm[58]-nm[61];
b=LSM-nm[45]-nm[46]-nm[61];
c=LSM-nm[42]-nm[46]-nm[58];

```

```

d=LSM-nm[1]-nm[24]-nm[43];
if((0<a)&&(a<65)&&(a==b)&&(a==c)&&(a==d)){
    if(uflg[a]==0){nm[62]=a;
        uflg[a]=1; stp64();
        uflg[a]=0;}}
}
//Set n63=LSM-n43-n47-n59
void stp64(){
    short a,b,c,d;
    a=LSM-nm[58]-nm[59]-nm[62];
    b=LSM-nm[46]-nm[47]-nm[62];
    c=LSM-nm[43]-nm[47]-nm[59];
    d=LSM-nm[4]-nm[21]-nm[42];
    if((0<a)&&(a<65)&&(a==b)&&(a==c)&&(a==d)){
        if(uflg[a]==0){nm[63]=a;
            uflg[a]=1; stp65();
            uflg[a]=0;}}
}
//Set n64=LSM-n44-n48-n60 & n1<n64
void stp65(){
    short a,b,c,d;
    a=LSM-nm[59]-nm[60]-nm[63];
    b=LSM-nm[47]-nm[48]-nm[63];
    c=LSM-nm[44]-nm[48]-nm[60];
    d=LSM-nm[1]-nm[22]-nm[43];
    if((nm[1]<a)&&(a<65)&&(a==b)&&(a==c)&&(a==d)){
        if(uflg[a]==0){nm[64]=a;
            uflg[a]=1; stp66();
            uflg[a]=0;}}
}
//The Last Check of Line-Sums
void stp66(){
    short sm1,sm2,sm3;
    sm1=nm[20]+nm[24]+nm[28]+nm[32];
    sm2=nm[16]+nm[32]+nm[48]+nm[64];
    sm3=nm[4]+nm[16]+nm[52]+nm[64];
    if((sm1==LSM)&&(sm2==LSM)&&(sm3==LSM)){
        ansprint();}
}
//
/* Print the Answers *
void ansprint(){
    short n;
    cnt++; cnt2++;
    if(cnt2==1){
        nt[cnt3][0]=cnt;
        for(n=1;n<65;n++){nt[cnt3][n]=nm[n];}
        cnt3++;
        if(typ==1){if(cnt3==6){pradj6(cnt3); cnt3=0;}}
        else{if(cnt3==3){prans(cnt3); cnt3=0;}}
    }
}
//
/* Print the Answers: Sub *
void prans(short x){
    short n;
    for(n=0;n<x;n++){printf("%29d#",nt[n][0]);
        if(n+1<x){printf(" ");}}
    printf("\n");
    for(n=0;n<x;n++){
        printf(" %2d-----%2d-----%2d-----%2d",nt[n][1],nt[n][17],nt[n][33],nt[n][49]);
        if(n+1<x){printf(" ");}}
    printf("\n");
    for(n=0;n<x;n++){
        printf(" |%2d      %2d      %2d      |%2d",nt[n][2],nt[n][18],nt[n][34],nt[n][50]);
        if(n+1<x){printf(" ");}}
}

```

```

printf("\n");
for(n=0;n<x;n++){
    printf(" %2d %2d %2d %2d %2d %2d %2d %2d",
           nt[n][ 5],nt[n][ 3],nt[n][21],nt[n][19],nt[n][37],nt[n][35],nt[n][53],nt[n][51]);
    if(n+1<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf(" !%2d %2d---%2d--%2d---%2d--%2d---+%2d--%2d",
           nt[n][ 6],nt[n][ 4],nt[n][22],nt[n][20],nt[n][38],nt[n][36],nt[n][54],nt[n][52]);
    if(n+1<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf(" %2d %2d | %2d %2d %2d %2d %2d %2d |",
           nt[n][ 9],nt[n][ 7],nt[n][25],nt[n][23],nt[n][41],nt[n][39],nt[n][57],nt[n][55]);
    if(n+1<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf(" !%2d %2d %2d %2d %2d %2d !%2d %2d",
           nt[n][10],nt[n][ 8],nt[n][26],nt[n][24],nt[n][42],nt[n][40],nt[n][58],nt[n][56]);
    if(n+1<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf(" %2d--%2d-!-%2d--%2d---%2d--%2d---%2d %2d |",
           nt[n][13],nt[n][11],nt[n][29],nt[n][27],nt[n][45],nt[n][43],nt[n][61],nt[n][59]);
    if(n+1<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf(" %2d %2d %2d %2d %2d %2d %2d %2d",
           nt[n][14],nt[n][12],nt[n][30],nt[n][28],nt[n][46],nt[n][44],nt[n][62],nt[n][60]);
    if(n+1<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf(" %2d | %2d %2d %2d |",nt[n][15],nt[n][31],nt[n][47],nt[n][63]);
    if(n+1<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf(" %2d-----%2d-----%2d-----%2d",nt[n][16],nt[n][32],nt[n][48],nt[n][64]);
    if(n+1<x){printf(" ");}}
printf("\n");
}
//
// 13
// n4 |
// \ |
// 49---n1---17
// | \
// | n2
// n5
//
/** Print the 6 Adjacent Numbers */
void pradj6(short x){
    short n;
    for(n=0;n<x;n++){printf("%4d#",nt[n][0]);
        if(n+1<x){printf(" ");}}
    printf("\n");
    for(n=0;n<x;n++){printf("%9d",nt[n][13]);
        if(n+1<x){printf(" ");}}
    printf("\n");
    for(n=0;n<x;n++){printf("%6d |",nt[n][4]);
        if(n+1<x){printf(" ");}}
    printf("\n");
    for(n=0;n<x;n++){printf(" \ \ |");
        if(n+1<x){printf(" ");}}
    printf("\n");
    for(n=0;n<x;n++){
        printf("%4d---%02d---%2d",nt[n][49],nt[n][1],nt[n][17]);
        if(n+1<x){printf(" ");}}
    printf("\n");
    for(n=0;n<x;n++){printf(" | \ \");
}

```

```

    if(n+1<x){printf("  ");}
printf("\n");
for(n=0;n<x;n++){printf("      | %2d",nt[n][2]);
    if(n+1<x){printf("  ");}
printf("\n");
for(n=0;n<x;n++){printf("      | %2d",nt[n][5]);
    if(n+1<x){printf("  ");}
printf("\n");
for(n=0;n<x;n++){printf("      | [%2d]",nt[n][43]);}
printf("\n\n");
}

```

7. 正方連結汎立体四方陣の標準解のリスト (抜粋)

<p>1#</p> <pre> 1-----56-----25-----48 64 9 40 17 62 2 11 55 38 26 19 47 3 63--54-10--27-39--46-18 7 61 50 12 31 37 42 20 58 4 15 53 34 28 23 45 60--8- 13-49--36--32--21 41 5 57 52 16 29 33 44 24 59 14 35 22 6-----51-----30-----43 </pre>	<p>13#</p> <pre> 1-----56-----25-----48 64 9 40 17 60 2 13 55 36 26 21 47 5 63--52-10--29-39--44-18 7 59 50 14 31 35 42 22 58 6 15 51 34 30 23 43 62--8- 11-49--38--32--19 41 3 57 54 16 27 33 46 24 61 12 37 20 4-----53-----28-----45 </pre>	<p>25#</p> <pre> 1-----60-----21-----48 64 5 44 17 56 2 13 59 36 22 25 47 9 63--52- 6--29-43--40-18 11 55 50 14 31 35 38 26 54 10 15 51 34 30 27 39 62--12- 7-49--42--32--19 37 3 53 58 16 23 33 46 28 61 8 41 20 4-----57-----24-----45 </pre>
<p>37#</p> <pre> 1-----60-----13-----56 64 5 52 9 48 2 21 59 36 14 25 55 17 63--44- 6--29-51--40-10 19 47 42 22 31 35 38 26 46 18 23 43 34 30 27 39 62--20- 7-41--50--32--11 37 3 45 58 24 15 33 54 28 61 8 49 12 4-----57-----16-----53 </pre>	<p>49#</p> <pre> 1-----60-----13-----56 64 5 52 9 32 2 37 59 20 14 41 55 33 63--28- 6--45-51--24-10 35 31 26 38 47 19 22 42 30 34 39 27 18 46 43 23 62--36- 7-25--50--48--11 21 3 29 58 40 15 17 54 44 61 8 49 12 4-----57-----16-----53 </pre>	<p>61#</p> <pre> 1-----56-----25-----48 64 9 40 17 63 3 10 54 39 27 18 46 2 62--55-11--26-38--47-19 6 61 51 12 30 37 43 20 59 4 14 53 35 28 22 45 60--8- 13-49--36--32--21 41 5 57 52 16 29 33 44 24 58 15 34 23 7-----50-----31-----42 </pre>
<p>73#</p> <pre> 1-----63-----18-----48 64 2 47 17 60 3 6 61 43 20 21 46 5 62--59- 4--22-45--44-19 13 58 51 8 30 41 36 23 52 7 14 57 35 24 29 42 56--15- 10-49--39--32--25 34 9 50 55 16 26 33 40 31 54 12 37 27 11-----53-----28-----38 </pre>	<p>79#</p> <pre> 1-----63-----18-----48 64 2 47 17 56 3 10 61 39 20 25 46 9 62--55- 4--26-45--40-19 13 54 51 12 30 37 36 27 52 11 14 53 35 28 29 38 60--15- 6-49--43--32--21 34 5 50 59 16 22 33 44 31 58 8 41 23 7-----57-----24-----42 </pre>	<p>85#</p> <pre> 1-----63-----10-----56 64 2 55 9 48 3 18 61 39 12 25 54 17 62--47- 4--26-53--40-11 21 46 43 20 30 37 36 27 44 19 22 45 35 28 29 38 60--23- 6-41--51--32--13 34 5 42 59 24 14 33 52 31 58 8 49 15 7-----57-----16-----50 </pre>
<p>91#</p> <pre> 1-----63-----10-----56 64 2 55 9 32 3 34 61 23 12 41 54 33 62--31- 4--42-53--24-11 37 30 27 36 46 21 20 43 28 35 38 29 19 44 45 22 60--39- 6-25--51--48--13 18 5 26 59 40 14 17 52 47 58 8 49 15 7-----57-----16-----50 </pre>	<p>97#</p> <pre> 1-----62-----19-----48 64 3 46 17 63 5 4 58 45 23 18 44 2 60--61- 7--20-42--47-21 10 59 53 8 28 41 39 22 55 6 12 57 37 24 26 43 56--14- 11-49--38--32--25 35 9 51 54 16 27 33 40 30 52 15 34 29 13-----50-----31-----36 </pre>	<p>103#</p> <pre> 1-----63-----18-----48 64 2 47 17 62 5 4 59 45 22 19 44 3 60--61- 6--20-43--46-21 11 58 53 8 28 41 38 23 54 7 12 57 37 24 27 42 56--15- 10-49--39--32--25 34 9 50 55 16 26 33 40 31 52 14 35 29 13-----51-----30-----36 </pre>
<p>109#</p> <pre> 1-----63-----34-----32 64 2 31 33 56 5 10 59 23 38 41 28 9 60--55- 6--42-27--24-37 25 52 39 14 58 19 8 45 40 13 26 51 7 46 57 20 48--29- 18-35--15--62--49 4 17 36 47 30 50 3 16 61 44 22 11 53 21-----43-----54-----12 </pre>	<p>111#</p> <pre> 1-----63-----34-----32 64 2 31 33 48 5 18 59 15 38 49 28 17 60--47- 6--50-27--16-37 25 44 39 22 58 11 8 53 40 21 26 43 7 54 57 12 56--29- 10-35--23--62--41 4 9 36 55 30 42 3 24 61 52 14 19 45 13-----51-----46-----20 </pre>	<p>113#</p> <pre> 1-----63-----18-----48 64 2 47 17 32 5 34 59 15 22 49 44 33 60--31- 6--50-43--16-21 41 28 23 38 58 11 8 53 24 37 42 27 7 54 57 12 56--45- 10-19--39--62--25 4 9 20 55 46 26 3 40 61 52 14 35 29 13-----51-----30-----36 </pre>

115#
 1-----62-----35-----32
 |64 3 30 33|
 63 9 4 54 29 43 34 24
 | 2 56 61 11 36 22 31 41
 18 55 | 45 12 52 21 15 42 |
 |47 10 20 53 13 44 |50 23
 48 26 |19 37 14 60 49 7 |
 17 39 46 28 51 5 16 58
 40 | 27 6 57 |
 25-----38-----59-----8

121#
 1-----56-----25-----48
 |63 10 39 18|
 62 2 11 55 38 26 19 47
 | 4 64 53 9 28 40 45 17
 7 61 | 50 12 31 37 42 20 |
 |57 3 16 54 33 27 |24 46
 60 8 |13 49 36 32 21 41 |
 6 58 51 15 30 34 43 23
 59 | 14 35 22 |
 5-----52-----29-----44

181#
 1-----56-----25-----48
 |63 10 39 18|
 64 4 9 53 40 28 17 45
 | 2 62 55 11 26 38 47 19
 5 61 | 52 12 29 37 44 20 |
 |59 3 14 54 35 27 |22 46
 60 8 |13 49 36 32 21 41 |
 6 58 51 15 30 34 43 23
 57 | 16 33 24 |
 7-----50-----31-----42

217#
 1-----62-----19-----48
 |63 4 45 18|
 64 6 3 57 46 24 17 43
 | 2 60 61 7 20 42 47 21
 9 59 | 54 8 27 41 40 22 |
 |55 5 12 58 37 23 |26 44
 56 14 |11 49 38 32 25 35 |
 10 52 53 15 28 34 39 29
 51 | 16 33 30 |
 13-----50-----31-----36

235#
 1-----62-----35-----32
 |63 4 29 34|
 64 10 3 53 30 44 33 23
 | 2 56 61 11 36 22 31 41
 17 55 | 46 12 51 21 16 42 |
 |47 9 20 54 13 43 |50 24
 48 26 |19 37 14 60 49 7 |
 18 40 45 27 52 6 15 57
 39 | 28 5 58 |
 25-----38-----59-----8

241#
 1-----56-----25-----48
 |62 11 38 19|
 63 3 10 54 39 27 18 46
 | 4 64 53 9 28 40 45 17
 6 61 | 51 12 30 37 43 20 |
 |57 2 16 55 33 26 |24 47
 60 8 |13 49 36 32 21 41 |
 7 59 50 14 31 35 42 22
 58 | 15 34 23 |
 5-----52-----29-----44

301#
 1-----56-----25-----48
 |62 11 38 19|
 64 4 9 53 40 28 17 45
 | 3 63 54 10 27 39 46 18
 5 61 | 52 12 29 37 44 20 |
 |58 2 15 55 34 26 |23 47
 60 8 |13 49 36 32 21 41 |
 7 59 50 14 31 35 42 22
 57 | 16 33 24 |
 6-----51-----30-----43

337#
 1-----63-----18-----48
 |62 4 45 19|
 64 7 2 57 47 24 17 42
 | 3 60 61 6 20 43 46 21
 9 58 | 55 8 26 41 40 23 |
 |54 5 12 59 37 22 |27 44
 56 15 |10 49 39 32 25 34 |
 11 52 53 14 28 35 38 29
 50 | 16 33 31 |
 13-----51-----30-----36

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

361#
 1-----56-----25-----48
 |60 13 36 21|
 63 5 10 52 39 29 18 44
 | 6 64 51 9 30 40 43 17
 4 59 | 53 14 28 35 45 22 |
 |57 2 16 55 33 26 |24 47
 62 8 |11 49 38 32 19 41 |
 7 61 50 12 31 37 42 20
 58 | 15 34 23 |
 3-----54-----27-----46

421#
 1-----56-----25-----48
 |60 13 36 21|
 64 6 9 51 40 30 17 43
 | 5 63 52 10 29 39 44 18
 3 59 | 54 14 27 35 46 22 |
 |58 2 15 55 34 26 |23 47
 62 8 |11 49 38 32 19 41 |
 7 61 50 12 31 37 42 20
 57 | 16 33 24 |
 4-----53-----28-----45

457#
 1-----63-----18-----48
 |60 6 43 21|
 64 7 2 57 47 24 17 42
 | 5 62 59 4 22 45 44 19
 9 58 | 55 8 26 41 40 23 |
 |52 3 14 61 35 20 |29 46
 56 15 |10 49 39 32 25 34 |
 13 54 51 12 30 37 36 27
 50 | 16 33 31 |
 11-----53-----28-----38

475#
 1-----63-----34-----32
 |60 6 27 37|
 64 13 2 51 31 46 33 20
 | 5 56 59 10 38 23 28 41
 17 52 | 47 14 50 19 16 45 |
 |44 9 22 55 11 42 |53 24
 48 29 |18 35 15 62 49 4 |
 21 40 43 26 54 7 12 57
 36 | 30 3 61 |
 25-----39-----58-----8

481#
 1-----60-----21-----48
 |56 13 36 25|
 63 9 6 52 43 29 18 40
 | 10 64 51 5 30 44 39 17
 4 55 | 57 14 24 35 45 26 |
 |53 2 16 59 33 22 |28 47
 62 12 |7 49 42 32 19 37 |
 11 61 50 8 31 41 38 20
 54 | 15 34 27 |
 3-----58-----23-----46

541#
 1-----60-----21-----48
 |56 13 36 25|
 64 10 5 51 44 30 17 39
 | 9 63 52 6 29 43 40 18
 3 55 | 58 14 23 35 46 26 |
 |54 2 15 59 34 22 |27 47
 62 12 |7 49 42 32 19 37 |
 11 61 50 8 31 41 38 20
 53 | 16 33 28 |
 4-----57-----24-----45

577#
 1-----63-----18-----48
 |56 10 39 25|
 64 11 2 53 47 28 17 38
 | 9 62 55 4 26 45 40 19
 5 54 | 59 12 22 37 44 27 |
 |52 3 14 61 35 20 |29 46
 60 15 |6 49 43 32 21 34 |
 13 58 51 8 30 41 36 23
 50 | 16 33 31 |
 7-----57-----24-----42

595#
 1-----63-----34-----32
 |56 10 23 41|
 64 13 2 51 31 46 33 20
 | 9 60 55 6 42 27 24 37
 17 52 | 47 14 50 19 16 45 |
 |40 5 26 59 7 38 |57 28
 48 29 |18 35 15 62 49 4 |
 25 44 39 22 58 11 8 53
 36 | 30 3 61 |
 21-----43-----54-----12

601#
 1-----60-----13-----56
 |48 21 36 25|
 63 17 6 44 51 29 10 40
 | 18 64 43 5 30 52 39 9
 4 47 | 57 22 16 35 53 26 |
 |45 2 24 59 33 14 |28 55
 62 20 |7 41 50 32 11 37 |
 19 61 42 8 31 49 38 12
 46 | 23 34 27 |
 3-----58-----15-----54

661#

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

697#

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

715#

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

721#

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

781#

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

817#

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

835#

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

841#

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

961#

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

1081#

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

1177#

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

1273#

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

1369#

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

1465#

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

1561#

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

2185#

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

2713#

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

3241#

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

3673#

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

4033#

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

4321#

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

4729#

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

5041#

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

5281#

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

5449#

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

5617#

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

5713#

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

5761#

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

6001#

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

6169#

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

6289#

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

6361#

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

6457#

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

6505#

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

6529#

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

6625#

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

$$\begin{aligned}n_1 + n_{22} + n_{43} + n_{64} &= (n_1 + n_{43}) + (n_{22} + n_{64}) = CC + CC = K (=130) \\n_1 + n_{30} + n_{43} + n_{56} &= (n_1 + n_{43}) + (n_{30} + n_{56}) = CC + CC = K (=130) \\n_1 + n_{54} + n_{43} + n_{32} &= (n_1 + n_{43}) + (n_{32} + n_{54}) = CC + CC = K (=130) \\n_1 + n_{62} + n_{43} + n_{24} &= (n_1 + n_{43}) + (n_{24} + n_{62}) = CC + CC = K (=130)\end{aligned}$$

主対角線 4 本は,

$$\begin{aligned}n_1 + n_{22} + n_{43} + n_{64} &= (n_1 + n_{43}) + (n_{22} + n_{64}) = CC + CC = K (=130) \\n_4 + n_{23} + n_{42} + n_{61} &= (n_4 + n_{42}) + (n_{23} + n_{61}) = CC + CC = K (=130) \\n_{13} + n_{26} + n_{39} + n_{52} &= (n_{13} + n_{39}) + (n_{26} + n_{52}) = CC + CC = K (=130) \\n_{16} + n_{27} + n_{38} + n_{49} &= (n_{16} + n_{38}) + (n_{27} + n_{49}) = CC + CC = K (=130)\end{aligned}$$

一筋ずれたところにある汎対角線も,

$$\begin{aligned}n_2 + n_{23} + n_{44} + n_{61} &= (n_2 + n_{44}) + (n_{23} + n_{61}) = CC + CC = K (=130) \\n_5 + n_{26} + n_{47} + n_{52} &= (n_5 + n_{47}) + (n_{26} + n_{52}) = CC + CC = K (=130) \\n_{17} + n_{38} + n_{59} + n_{16} &= (n_{17} + n_{59}) + (n_{38} + n_{16}) = CC + CC = K (=130)\end{aligned}$$

というふうに決まって行く。

$n_1 + n_{43} = 65$ と定義したので, $n_1 = 1$ ならば, 必ず $n_{43} = 64$ である。

このタイプしかないのが, 単なる汎立体四方陣と決定的に違う点だ。当然解の総数は, 少なくなる。より稀少な魔方陣として, 珍重されている。

その他の定義, 例えば**正方連結四数定和条件**などは, 前の汎対角線型と全く同じで良い。

9. プログラムは, 次のように組んだ。

補数の定義式によって, 一方の $n_1, n_2, n_3, n_4, \dots$ を決めると他方の $n_{43}, n_{44}, n_{41}, n_{42}, \dots$ が自動的に決まるタイプなので, 1 ステップで 2 つの変数値を決めることができる。

```

/** 'Composite & Complete' Magic Cubes of Order 4 **
/** Compact List of the 960 Standard Solutions **
/** 'CCMC444St.c': Dictated by Kanji Setsuda **
/** in '02,'07,'12; Revised on Dec.28, 2014; **
/** and Recompiled on Aug. 7, 2015 **
/** Working with MacOSX 10.10.4 and Xcode 6.4 **
//
#include <stdio.h>
//
// Global Var.
short int cnt, cnt2;
short cnt3;
short CC, SSM, LSM;
short nm[65], uflg[65];
short tn[8][65];
//
/** Functions(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), 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 ansprint(void);
void prans(short x);
//
/** Main Program *
int main(){
short n;
printf("\n");
printf("*** 'Composite & Complete' Magic Cubes of Order 4: ***\n");
printf("*** Compact List of 960 Standard Solutions Normalized ***\n");

```



```

CC=65; SSM=130; LSM=130; cnt=0; cnt3=0;
for(n=0;n<65;n++){nm[n]=0; uflg[n]=0;}
stp01();    /* Begin the Calculations *
if(cnt3>0){prans(cnt3);}
printf(" [Solution Counts = %d]\n",cnt);
printf("\n");
printf(" [OK!]\n");
printf("*** Calculated and Listed by Kanji Setsuda on\n");
printf("   Aug. 7, 2015 with MacOSX 10.10.4 & Xcode 6.4 **\n");
printf("\n");
return 0;
}
//
/* Begin the Calculations *
/* Define Level 1: *
/* Set N1 & n43 *
void stp01(){
short a,b;
for(a=1;a<33;a++){b=CC-a;
if((uflg[a]==0)&&(uflg[b]==0)){cnt2=0;
nm[1]=a; nm[43]=b;
uflg[a]=1; uflg[b]=1;
stp02();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set N2 & n44 *
void stp02(){
short a,b;
for(a=64;a>0;a--){b=CC-a;
if((uflg[a]==0)&&(uflg[b]==0)){if(nm[1]==1){cnt2=0;}
nm[2]=a; nm[44]=b;
uflg[a]=1; uflg[b]=1;
stp03();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set N5(<n2) & n47 *
void stp03(){
short a,b;
for(a=nm[2]-1;a>0;a--){b=CC-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[5]=a; nm[47]=b;
uflg[a]=1; uflg[b]=1;
stp04();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set n6=130-n1-n2-n5 & n48 *
void stp04(){
short a,b;
a=LSM-nm[1]-nm[2]-nm[5];
if((0<a)&&(a<65)){
b=LSM-nm[43]-nm[44]-nm[47];
if(a+b==CC){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[6]=a; nm[48]=b;
uflg[a]=1; uflg[b]=1;
stp05();
uflg[b]=0; uflg[a]=0;}}}
}
/* Set n17(<n5) & n59 *
void stp05(){
short a,b;
for(a=nm[5]-1;a>0;a--){b=CC-a;

```

```

        if((uflg[a]==0)&&(uflg[b]==0)){if(nm[1]==1){cnt2=0;}
            nm[17]=a; nm[59]=b;
            uflg[a]=1; uflg[b]=1;
            stp06();
            uflg[b]=0; uflg[a]=0;}
    }
}
/** Set n18=130-n1-n2-n17 & n60 *
void stp06(){
    short a,b;
    a=LSM-nm[1]-nm[2]-nm[17];
    if((0<a)&&(a<65)){
        b=LSM-nm[43]-nm[44]-nm[59];
        if(a+b==CC){
            if((uflg[a]==0)&&(uflg[b]==0)){
                nm[18]=a; nm[60]=b;
                uflg[a]=1; uflg[b]=1;
                stp07();
                uflg[b]=0; uflg[a]=0;}}}
}
/** Set n21=130-n1-n5-n17 & n63 *
void stp07(){
    short a,b;
    a=LSM-nm[1]-nm[5]-nm[17];
    if((0<a)&&(a<65)){
        b=LSM-nm[43]-nm[47]-nm[59];
        if(a+b==CC){
            if((uflg[a]==0)&&(uflg[b]==0)){
                nm[21]=a; nm[63]=b;
                uflg[a]=1; uflg[b]=1;
                stp08();
                uflg[b]=0; uflg[a]=0;}}}
}
/** Set n22=130-n2-n6-n18 & n64(>n1) *
void stp08(){
    short a,b,c,d;
    a=LSM-nm[2]-nm[6]-nm[18];
    if((0<a)&&(a<65)){
        b=LSM-nm[44]-nm[48]-nm[60];
        if((a+b==CC)&&(b>nm[1])){
            c=LSM-nm[5]-nm[6]-nm[21];
            d=LSM-nm[17]-nm[18]-nm[21];
            if((a==c)&&(a==d)){
                if((uflg[a]==0)&&(uflg[b]==0)){
                    nm[22]=a; nm[64]=b;
                    uflg[a]=1; uflg[b]=1;
                    stp09();
                    uflg[b]=0; uflg[a]=0;}}}}
}
/** Level 2: *
/** Set N4(>n1) & n42 *
void stp09(){
    short a,b;
    for(a=64;a>nm[1];a--){b=CC-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[4]=a; nm[42]=b;
            uflg[a]=1; uflg[b]=1;
            stp10();
            uflg[b]=0; uflg[a]=0;}
    }
}
/** Set n3=130-n1-n2-n4 & n41 *
void stp10(){
    short a,b;
    a=LSM-nm[1]-nm[2]-nm[4];

```

```

if((0<a)&&(a<65)){
    b=LSM-nm[43]-nm[44]-nm[42];
    if(a+b==CC){
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[3]=a; nm[41]=b;
            uflg[a]=1; uflg[b]=1;
            stp11();
            uflg[b]=0; uflg[a]=0;}}}
}
/* Set n7=130-n2-n3-n6 & n45 *
void stp11(){
    short a,b;
    a=LSM-nm[2]-nm[3]-nm[6];
    if((0<a)&&(a<65)){
        b=LSM-nm[44]-nm[41]-nm[48];
        if(a+b==CC){
            if((uflg[a]==0)&&(uflg[b]==0)){
                nm[7]=a; nm[45]=b;
                uflg[a]=1; uflg[b]=1;
                stp12();
                uflg[b]=0; uflg[a]=0;}}}
}
/* Set n8=130-n3-n4-n7 & n46 *
void stp12(){
    short a,b,c,d;
    a=LSM-nm[3]-nm[4]-nm[7];
    if((0<a)&&(a<65)){
        b=LSM-nm[41]-nm[42]-nm[45];
        if(a+b==CC){
            c=LSM-nm[4]-nm[1]-nm[5];
            d=LSM-nm[42]-nm[43]-nm[47];
            if((a==c)&&(b==d)){
                if((uflg[a]==0)&&(uflg[b]==0)){
                    nm[8]=a; nm[46]=b;
                    uflg[a]=1; uflg[b]=1;
                    stp13();
                    uflg[b]=0; uflg[a]=0;}}}}}
}
/* Set n19=130-n2-n3-n18 & n57 *
void stp13(){
    short a,b;
    a=LSM-nm[2]-nm[3]-nm[18];
    if((0<a)&&(a<65)){
        b=LSM-nm[44]-nm[41]-nm[60];
        if(a+b==CC){
            if((uflg[a]==0)&&(uflg[b]==0)){
                nm[19]=a; nm[57]=b;
                uflg[a]=1; uflg[b]=1;
                stp14();
                uflg[b]=0; uflg[a]=0;}}}
}
/* Set n20=130-n3-n4-n19 & n58 *
void stp14(){
    short a,b,c,d;
    a=LSM-nm[3]-nm[4]-nm[19];
    if((0<a)&&(a<65)){
        b=LSM-nm[41]-nm[42]-nm[57];
        if(a+b==CC){
            c=LSM-nm[4]-nm[1]-nm[17];
            d=LSM-nm[42]-nm[43]-nm[59];
            if((a==c)&&(b==d)){
                if((uflg[a]==0)&&(uflg[b]==0)){
                    nm[20]=a; nm[58]=b;
                    uflg[a]=1; uflg[b]=1;
                    stp15();

```

```

        uflg[b]=0; uflg[a]=0;}}}}
}
/** Set n23=130-n3-n7-n19 & n61(>n1) *
void stp15(){
short a,b,c,d;
a=LSM-nm[3]-nm[7]-nm[19];
if((0<a)&&(a<65)){
b=LSM-nm[41]-nm[45]-nm[57];
if((a+b==CC)&&(b>nm[1])){
c=LSM-nm[6]-nm[7]-nm[22];
d=LSM-nm[48]-nm[45]-nm[64];
if((a==c)&&(b==d)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[23]=a; nm[61]=b;
uflg[a]=1; uflg[b]=1;
stp16();
uflg[b]=0; uflg[a]=0;}}}}
}
/** Set n24=130-n4-n8-n20 & n58 *
void stp16(){
short a,b,c,d,e;
a=LSM-nm[4]-nm[8]-nm[20];
if((0<a)&&(a<65)){
b=LSM-nm[42]-nm[46]-nm[58];
if(a+b==CC){
c=LSM-nm[7]-nm[8]-nm[23];
d=LSM-nm[19]-nm[20]-nm[23];
e=LSM-nm[20]-nm[17]-nm[21];
if((a==c)&&(a==d)&&(a==e)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[24]=a; nm[62]=b;
uflg[a]=1; uflg[b]=1;
stp17();
uflg[b]=0; uflg[a]=0;}}}}
}
/** Level 3: *
/** Set N13(>n1) & n39 *
void stp17(){
short a,b;
for(a=64;a>nm[1];a--){b=CC-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[13]=a; nm[39]=b;
uflg[a]=1; uflg[b]=1;
stp18();
uflg[b]=0; uflg[a]=0;}}
}
}
/** Set n9=130-n1-n5-n13 & n35 *
void stp18(){
short a,b;
a=LSM-nm[1]-nm[5]-nm[13];
if((0<a)&&(a<65)){
b=LSM-nm[43]-nm[47]-nm[39];
if(a+b==CC){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[9]=a; nm[35]=b;
uflg[a]=1; uflg[b]=1;
stp19();
uflg[b]=0; uflg[a]=0;}}}}
}
/** Set n10=130-n5-n6-n9 & n36 *
void stp19(){
short a,b;
a=LSM-nm[5]-nm[6]-nm[9];
if((0<a)&&(a<65)){

```

```

    b=LSM-nm[47]-nm[48]-nm[35];
    if(a+b==CC){
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[10]=a; nm[36]=b;
            uflg[a]=1; uflg[b]=1;
            stp20();
            uflg[b]=0; uflg[a]=0;}}
    }
    /* Set n11=130-n6-n7-n10 & n33 *
void stp20(){
    short a,b;
    a=LSM-nm[6]-nm[7]-nm[10];
    if((0<a)&&(a<65)){
        b=LSM-nm[48]-nm[45]-nm[36];
        if(a+b==CC){
            if((uflg[a]==0)&&(uflg[b]==0)){
                nm[11]=a; nm[33]=b;
                uflg[a]=1; uflg[b]=1;
                stp21();
                uflg[b]=0; uflg[a]=0;}}
        }
    /* Set n12=130-n7-n8-n11 & n34 *
void stp21(){
    short a,b,c,d;
    a=LSM-nm[7]-nm[8]-nm[11];
    if((0<a)&&(a<65)){
        b=LSM-nm[45]-nm[46]-nm[33];
        if(a+b==CC){
            c=LSM-nm[8]-nm[5]-nm[9];
            d=LSM-nm[46]-nm[47]-nm[35];
            if((a==c)&&(b==d)){
                if((uflg[a]==0)&&(uflg[b]==0)){
                    nm[12]=a; nm[34]=b;
                    uflg[a]=1; uflg[b]=1;
                    stp22();
                    uflg[b]=0; uflg[a]=0;}}}}
        }
    /* Set n14=130-n9-n10-n13 & n40 *
void stp22(){
    short a,b,c,d;
    a=LSM-nm[9]-nm[10]-nm[13];
    if((0<a)&&(a<65)){
        b=LSM-nm[35]-nm[36]-nm[39];
        if(a+b==CC){
            c=LSM-nm[13]-nm[1]-nm[2];
            d=LSM-nm[39]-nm[43]-nm[44];
            if((a==c)&&(b==d)){
                if((uflg[a]==0)&&(uflg[b]==0)){
                    nm[14]=a; nm[40]=b;
                    uflg[a]=1; uflg[b]=1;
                    stp23();
                    uflg[b]=0; uflg[a]=0;}}}}
        }
    /* Set n15=130-n10-n11-n14 & n37 *
void stp23(){
    short a,b,c,d;
    a=LSM-nm[10]-nm[11]-nm[14];
    if((0<a)&&(a<65)){
        b=LSM-nm[36]-nm[33]-nm[40];
        if(a+b==CC){
            c=LSM-nm[14]-nm[2]-nm[3];
            d=LSM-nm[40]-nm[44]-nm[41];
            if((a==c)&&(b==d)){
                if((uflg[a]==0)&&(uflg[b]==0)){
                    nm[15]=a; nm[37]=b;

```

```

        uflg[a]=1; uflg[b]=1;
        stp24();
        uflg[b]=0; uflg[a]=0;}}}}
}
/* Set n16=130-n11-n12-n15 & n1<n16 & n38 *
void stp24(){
short a,b,c,d;
a=LSM-nm[11]-nm[12]-nm[15];
if((nm[1]<a)&&(a<65)){
b=LSM-nm[33]-nm[34]-nm[37];
if(a+b==CC){
c=LSM-nm[15]-nm[3]-nm[4];
d=LSM-nm[34]-nm[35]-nm[39];
if((a==c)&&(b==d)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[16]=a; nm[38]=b;
uflg[a]=1; uflg[b]=1;
stp25();
uflg[b]=0; uflg[a]=0;}}}}
}
/* Level 4: *
/* Set n25=130-n5-n9-n21 & n51 *
void stp25(){
short a,b,c,d;
a=LSM-nm[5]-nm[9]-nm[21];
if((0<a)&&(a<65)){
b=LSM-nm[47]-nm[35]-nm[63];
if(a+b==CC){
c=LSM-nm[21]-nm[37]-nm[41];
d=LSM-nm[63]-nm[15]-nm[3];
if((a==c)&&(b==d)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[25]=a; nm[51]=b;
uflg[a]=1; uflg[b]=1;
stp26();
uflg[b]=0; uflg[a]=0;}}}}
}
/* Set n29=130-n9-n13-n25 & n55 *
void stp26(){
short a,b,c,d;
a=LSM-nm[9]-nm[13]-nm[25];
if((0<a)&&(a<65)){
b=LSM-nm[35]-nm[39]-nm[51];
if(a+b==CC){
c=LSM-nm[25]-nm[41]-nm[45];
d=LSM-nm[39]-nm[43]-nm[59];
if((a==c)&&(b==d)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[29]=a; nm[55]=b;
uflg[a]=1; uflg[b]=1;
stp27();
uflg[b]=0; uflg[a]=0;}}}}
}
/* Set n26=130-n6-n10-n22 & n52(>n1) *
void stp27(){
short a,b,c,d,e;
a=LSM-nm[6]-nm[10]-nm[22];
if((0<a)&&(a<65)){
b=LSM-nm[48]-nm[36]-nm[64];
if((a+b==CC)&&(b>nm[1])){
c=LSM-nm[9]-nm[10]-nm[25];
d=LSM-nm[21]-nm[22]-nm[25];
e=LSM-nm[22]-nm[38]-nm[42];
if((a==c)&&(a==d)&&(a==e)){
if((uflg[a]==0)&&(uflg[b]==0)){

```



```

        nm[26]=a; nm[52]=b;
        uflg[a]=1; uflg[b]=1;
        stp28();
        uflg[b]=0; uflg[a]=0;}}}}
}
/* Set n30=130-n10-n14-n26 & n56 *
void stp28(){
short a,b,c,d,e;
a=LSM-nm[10]-nm[14]-nm[26];
if((0<a)&&(a<65)){
b=LSM-nm[36]-nm[40]-nm[52];
if(a+b==CC){
c=LSM-nm[13]-nm[14]-nm[29];
d=LSM-nm[25]-nm[26]-nm[29];
e=LSM-nm[29]-nm[45]-nm[46];
if((a==c)&&(a==d)&&(a==e)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[30]=a; nm[56]=b;
uflg[a]=1; uflg[b]=1;
stp29();
uflg[b]=0; uflg[a]=0;}}}}
}
}
/* Set n27=130-n7-n11-n23 & n49(>n1) *
void stp29(){
short a,b,c,d,e;
a=LSM-nm[7]-nm[11]-nm[23];
if((0<a)&&(a<65)){
b=LSM-nm[45]-nm[33]-nm[61];
if((a+b==CC)&&(b>nm[1])){
c=LSM-nm[10]-nm[11]-nm[26];
d=LSM-nm[22]-nm[23]-nm[26];
e=LSM-nm[23]-nm[39]-nm[43];
if((a==c)&&(a==d)&&(a==e)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[27]=a; nm[49]=b;
uflg[a]=1; uflg[b]=1;
stp30();
uflg[b]=0; uflg[a]=0;}}}}
}
}
/* Set n28=130-n8-n12-n24 & n50 *
void stp30(){
short a,b,c,d,e;
a=LSM-nm[8]-nm[12]-nm[24];
if((0<a)&&(a<65)){
b=LSM-nm[46]-nm[34]-nm[62];
if(a+b==CC){
c=LSM-nm[11]-nm[12]-nm[27];
d=LSM-nm[23]-nm[24]-nm[27];
e=LSM-nm[27]-nm[43]-nm[44];
if((a==c)&&(a==d)&&(a==e)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[28]=a; nm[50]=b;
uflg[a]=1; uflg[b]=1;
stp31();
uflg[b]=0; uflg[a]=0;}}}}
}
}
/* Set n31=130-n11-n15-n27 & n53 *
void stp31(){
short a,b,c,d,e;
a=LSM-nm[11]-nm[15]-nm[27];
if((0<a)&&(a<65)){
b=LSM-nm[33]-nm[37]-nm[49];
if(a+b==CC){
c=LSM-nm[14]-nm[15]-nm[30];
d=LSM-nm[26]-nm[27]-nm[30];

```

```

    e=LSM-nm[27]-nm[43]-nm[47];
    if((a==c)&&(a==d)&&(a==e)){
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[31]=a; nm[53]=b;
            uflg[a]=1; uflg[b]=1;
            stp32();
            uflg[b]=0; uflg[a]=0;}}}}
}
/* Set n32=130-n12-n16-n28 & n54 *
void stp32(){
    short a,b,c,d,e;
    a=LSM-nm[12]-nm[16]-nm[28];
    if((0<a)&&(a<65)){
        b=LSM-nm[34]-nm[38]-nm[50];
        if(a+b==CC){
            c=LSM-nm[15]-nm[16]-nm[31];
            d=LSM-nm[27]-nm[28]-nm[31];
            e=LSM-nm[48]-nm[29]-nm[45];
            if((a==c)&&(a==d)&&(a==e)){
                if((uflg[a]==0)&&(uflg[b]==0)){
                    nm[32]=a; nm[54]=b;
                    uflg[a]=1; uflg[b]=1;
                    ansprint(); /* Print the Answers *
                    uflg[b]=0; uflg[a]=0;}}}}
}
//
/* Print the Answers *
void ansprint(){
    short n;
    cnt++; cnt2++;
    if(cnt2==1){
        tn[cnt3][0]=cnt;
        for(n=1;n<65;n++){tn[cnt3][n]=nm[n];}
        cnt3++;
        if(cnt3==3){prans(cnt3); cnt3=0;}
    }
}
//
/* Print the Answers: Sub *
void prans(short x){
    short n;
    for(n=0;n<x;n++){printf("%29d/",tn[n][0]);
        if(n+1<x){printf(" ");}}
    printf("\n");
    for(n=0;n<x;n++){
        printf(" %2d-----%2d-----%2d-----%2d",tn[n][1],tn[n][2],tn[n][3],tn[n][4]);
        if(n+1<x){printf(" ");}}
    printf("\n");
    for(n=0;n<x;n++){
        printf(" |%2d |%2d |%2d |%2d",tn[n][17],tn[n][18],tn[n][19],tn[n][20]);
        if(n+1<x){printf(" ");}}
    printf("\n");
    for(n=0;n<x;n++){
        printf(" %2d %2d %2d %2d %2d %2d %2d %2d",
            tn[n][5],tn[n][33],tn[n][6],tn[n][34],tn[n][7],tn[n][35],tn[n][8],tn[n][36]);
        if(n+1<x){printf(" ");}}
    printf("\n");
    for(n=0;n<x;n++){
        printf(" |%2d %2d--+%2d--%2d--+%2d--%2d--+%2d--%2d",
            tn[n][21],tn[n][49],tn[n][22],tn[n][50],tn[n][23],tn[n][51],tn[n][24],tn[n][52]);
        if(n+1<x){printf(" ");}}
    printf("\n");
    for(n=0;n<x;n++){
        printf(" %2d %2d | %2d %2d | %2d %2d | %2d %2d |",
            tn[n][9],tn[n][37],tn[n][10],tn[n][38],tn[n][11],tn[n][39],tn[n][12],tn[n][40]);

```

```

    if(n+1<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf(" %2d %2d %2d %2d %2d %2d %2d %2d",
           tn[n][25],tn[n][53],tn[n][26],tn[n][54],tn[n][27],tn[n][55],tn[n][28],tn[n][56]);
    if(n+1<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf(" %2d--%2d-l-%2d--%2d-l-%2d--%2d-l-%2d %2d l",
           tn[n][13],tn[n][41],tn[n][14],tn[n][42],tn[n][15],tn[n][43],tn[n][16],tn[n][44]);
    if(n+1<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf(" %2d %2d %2d %2d %2d %2d %2d %2d",
           tn[n][29],tn[n][57],tn[n][30],tn[n][58],tn[n][31],tn[n][59],tn[n][32],tn[n][60]);
    if(n+1<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf(" %2d l %2d l %2d l %2d l",tn[n][45],tn[n][46],tn[n][47],tn[n][48]);
    if(n+1<x){printf(" ");}}
printf("\n");
for(n=0;n<x;n++){
    printf(" %2d-----%2d-----%2d-----%2d",tn[n][61],tn[n][62],tn[n][63],tn[n][64]);
    if(n+1<x){printf(" ");}}
printf("\n");
}
//

```

10. 次にすぐにも実行結果を示そう。

正方連結完全汎立体四方陣の標準解のリスト (抜粋)

** Compact List of the 960 Standard Solutions Normalized: CCMC444 **

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

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

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

67/
 1 62 4 63
 48 19 45 18
 60 49 7 14 57 52 6 15
 21 32 +42 -35 +24 -29 +43 -34
 13 12 50 55 16 9 51 54 1
 36 37 31 26 33 40 30 27
 56-61-11-2-53-64-10 3 1
 25 20 38 47 28 17 39 46
 8 59 5 58
 41 22 44 23

73/
 1 62 4 63
 32 35 29 34
 60 49 7 14 57 52 6 15
 37 48 +26 -19 +40 -45 -27 -18
 13 12 50 55 16 9 51 54 1
 20 21 47 42 17 24 46 43
 56-61-11-2-53-64-10 3 1
 41 36 22 31 44 33 23 30
 8 59 5 58
 25 38 28 39

79/
 1 62 4 63
 48 19 45 18
 56 49 11 14 53 52 10 15
 25 32 +38 -35 +28 -29 +39 -34
 13 8 50 59 16 5 51 58 1
 36 41 31 22 33 44 30 23
 60-61-7-2-57-64-6 3 1
 21 20 42 47 24 17 43 46
 12 55 9 54
 37 26 40 27

85/
 1 62 4 63
 32 35 29 34
 56 49 11 14 53 52 10 15
 41 48 +22 -19 +44 -45 +23 -18
 13 8 50 59 16 5 51 58 1
 20 25 47 38 17 28 46 39
 60-61-7-2-57-64-6 3 1
 37 36 26 31 40 33 27 30
 12 55 9 54
 21 42 24 43

91/
 1 62 4 63
 32 35 29 34
 48 41 19 22 45 44 18 23
 49 56 +14 -11 +52 -53 -15 -10
 21 8 42 59 24 5 43 58 1
 12 25 55 38 9 28 54 39
 60-61-7-2-57-64-6 3 1
 37 36 26 31 40 33 27 30
 20 47 17 46
 13 50 16 51

97/
 1 60 6 63
 48 21 43 18
 56 49 13 12 51 54 10 15
 25 32 +36 -37 +30 -27 +39 -34
 11 8 50 61 16 3 53 58 1
 38 41 31 20 33 46 28 23
 62-59-7-2-57-64-4 5 1
 19 22 42 47 24 17 45 44
 14 55 9 52
 35 26 40 29

103/
 1 60 6 63
 32 37 27 34
 56 49 13 12 51 54 10 15
 41 48 +20 -21 +46 -43 +23 -18
 11 8 50 61 16 3 53 58 1
 22 25 47 36 17 30 44 39
 62-59-7-2-57-64-4 5 1
 35 38 26 31 40 33 29 28
 14 55 9 52
 19 42 24 45

109/
 1 60 6 63
 32 37 27 34
 48 41 21 20 43 46 18 23
 49 56 +12 -13 +54 -51 -15 -10
 19 8 42 61 24 3 45 58 1
 14 25 55 36 9 30 52 39
 62-59-7-2-57-64-4 5 1
 35 38 26 31 40 33 29 28
 22 47 17 44
 11 50 16 53

115/
 1 56 10 63
 32 41 23 34
 48 37 25 20 39 46 18 27
 49 60 +8 -13 +58 -51 -15 -6
 19 12 38 61 28 3 45 54 1
 14 21 59 36 5 30 52 43
 62-55-11-2-53-64-4 9 1
 35 42 22 31 44 33 29 24
 26 47 17 40
 7 50 16 57

121/
 2 64 9 55
 59 5 52 14
 61 38 3 28 54 45 12 19
 8 31 +58 -33 +15 -24 +49 -42
 20 25 46 39 27 18 37 48 1
 41 36 23 30 34 43 32 21
 47-56-17-10-40-63-26 1 1
 22 13 44 51 29 6 35 60
 11 53 4 62
 50 16 57 7

241/
 3 64 9 54
 58 5 52 15
 61 39 2 28 55 45 12 18
 8 30 +59 -33 +14 -24 +49 -43
 20 25 47 38 26 19 37 48 1
 41 36 22 31 35 42 32 21
 46-56-17-11-40-62-27 1 1
 23 13 44 50 29 7 34 60
 10 53 4 63
 51 16 57 6

337/
 4 63 10 53
 57 6 51 16
 62 40 1 27 56 46 11 17
 7 29 +60 -34 +13 -23 +50 -44
 19 26 48 37 25 20 38 47 1
 42 35 21 32 36 41 31 22
 45-55-18-12-39-61-28 2 1
 24 14 43 49 30 8 33 59
 9 54 3 64
 52 15 58 5

433/
 5 64 3 58
 52 9 54 15
 59 45 2 24 61 43 8 18
 14 28 +55 -33 +12 -30 +49 -39
 22 19 47 42 20 21 41 48 1
 35 38 26 31 37 36 32 25
 44-62-17-7-46-60-23 1 1
 29 11 40 50 27 13 34 56
 4 57 6 63
 53 16 51 10

505/
 6 63 4 57
 51 10 53 16
 60 46 1 23 62 44 7 17
 13 27 +56 -34 +11 -29 +50 -40
 21 20 48 41 19 22 42 47 1
 36 37 25 32 38 35 31 26
 43-61-18-8-45-59-24 2 1
 30 12 39 49 28 14 33 55
 3 58 5 64
 54 15 52 9

577/
 7 62 4 57
 50 11 53 16
 60 47 1 22 63 44 6 17
 13 26 +56 -35 +10 -29 +51 -40
 21 20 48 41 18 23 43 46 1
 36 37 25 32 39 34 30 27
 42-61-19-8-45-58-24 3 1
 31 12 38 49 28 15 33 54
 2 59 5 64
 55 14 52 9

631/
 8 61 3 58
 49 12 54 15
 59 48 2 21 64 43 5 18
 14 25 +55 -36 +9 -30 +52 -39
 22 19 47 42 17 24 44 45 1
 35 38 26 31 40 33 29 28
 41-62-20-7-46-57-23 4 1
 32 11 37 50 27 16 34 53
 1 60 6 63
 56 13 51 10

** Compact List of 960 Standard Solutions Normalized: CCMC444: Calculated and Listed by Kanji Setsuda **

標準解の総数は、前の汎立体四方陣の7分の1の960個しかなかった。 $960 = 2^6 \times 5 \times 3$;
 $n1 = 1$ の時の解も、予想通り7分の1の120個だった。 $120 = 2^3 \times 5 \times 3$;
 前のタイプの解の7分の1しかない理由は、18 ページに示した表に関係がある。
 実は、完全立体四方陣は、汎立体四方陣の一部であり、前者は後者の中に含まれる。汎対角線型の $n1 = 1$ の時の $n43 = 64$ のタイプの解120個が、実は完全型なのだ。
 そう推理されたので、実際に調べて見たところ、事実その通りであった。

正方連結完全型は、三次元立方体四方陣にしろ、二次元正方形八方陣にしろ、特に我が国日本で、古くから珍重され研究されて来た歴史を持つ由緒ある魔方陣だ。私自身も昔、阿部楽方老師からハッパをかけられて、随分と肩入れしつつ研究・発表して来た経緯がある。

11. 正方連結対称立体四方陣の研究

次は、「対称型」である。同じ正方連結型の立方体四方陣に対称型は無いものだろうか？
 「完全型」と「対称型」には、二次元にしろ三次元にしろ、その間に特別の連携・関係があり、昔からそれを愛され研究されて来た経緯がある。

我々古くからのファンは、だから、どうしても対称型を求めてしまう。

定義は、次のようにする。正方連結四数定和の条件などは、今までと全く同じだ。

** 立方体四方陣の基本図 **

1	2	3	4
17	18	19	20
5	33	6	34
21	49	22	50
9	37	10	38
25	53	26	54
13	41	14	42
29	57	30	58
45	46	47	48
61	62	63	64

** 正方連結四数定和の定義式: C=130; **

$n1+n2+n5+n6=C$... c1a;	$n1+n2+n17+n18=C$... c1b;	$n1+n5+n17+n21=C$... c1c;
$n2+n3+n6+n7=C$... c2a;	$n2+n3+n18+n19=C$... c2b;	$n2+n6+n18+n22=C$... c2c;
$n3+n4+n7+n8=C$... c3a;	$n3+n4+n19+n20=C$... c3b;	$n3+n7+n19+n23=C$... c3c;
$n4+n1+n8+n5=C$... c4a;	$n4+n1+n20+n17=C$... c4b;	$n4+n8+n20+n24=C$... c4c;
$n5+n6+n9+n10=C$... c5a;	$n5+n6+n21+n22=C$... c5b;	$n5+n9+n21+n25=C$... c5c;
$n6+n7+n10+n11=C$... c6a;	$n6+n7+n22+n23=C$... c6b;	$n6+n10+n22+n26=C$... c6c;
$n7+n8+n11+n12=C$... c7a;	$n7+n8+n23+n24=C$... c7b;	$n7+n11+n23+n27=C$... c7c;
$n8+n5+n12+n9=C$... c8a;	$n8+n5+n24+n21=C$... c8b;	$n8+n12+n24+n28=C$... c8c;
$n9+n10+n13+n14=C$... c9a;	$n9+n10+n25+n26=C$... c9b;	$n9+n13+n25+n29=C$... c9c;
$n10+n11+n14+n15=C$... c10a;	$n10+n11+n26+n27=C$... c10b;	$n10+n14+n26+n30=C$... c10c;
$n11+n12+n15+n16=C$... c11a;	$n11+n12+n27+n28=C$... c11b;	$n11+n15+n27+n31=C$... c11c;
$n12+n9+n16+n13=C$... c12a;	$n12+n9+n28+n25=C$... c12b;	$n12+n16+n28+n32=C$... c12c;
$n13+n14+n1+n2=C$... c13a;	$n13+n14+n29+n30=C$... c13b;	$n13+n1+n29+n17=C$... c13c;
$n14+n15+n2+n3=C$... c14a;	$n14+n15+n30+n31=C$... c14b;	$n14+n2+n30+n18=C$... c14c;
$n15+n16+n3+n4=C$... c15a;	$n15+n16+n31+n32=C$... c15b;	$n15+n3+n31+n19=C$... c15c;
$n16+n13+n4+n1=C$... c16a;	$n16+n13+n32+n29=C$... c16b;	$n16+n4+n32+n20=C$... c16c;
$n17+n18+n21+n22=C$... c17a;	$n17+n18+n33+n34=C$... c17b;	$n17+n21+n33+n37=C$... c17c;
$n18+n19+n22+n23=C$... c18a;	$n18+n19+n34+n35=C$... c18b;	$n18+n22+n34+n38=C$... c18c;
$n19+n20+n23+n24=C$... c19a;	$n19+n20+n35+n36=C$... c19b;	$n19+n23+n35+n39=C$... c19c;
$n20+n17+n24+n21=C$... c20a;	$n20+n17+n36+n33=C$... c20b;	$n20+n24+n36+n40=C$... c20c;
$n21+n22+n25+n26=C$... c21a;	$n21+n22+n37+n38=C$... c21b;	$n21+n25+n37+n41=C$... c21c;
$n22+n23+n26+n27=C$... c22a;	$n22+n23+n38+n39=C$... c22b;	$n22+n26+n38+n42=C$... c22c;
$n23+n24+n27+n28=C$... c23a;	$n23+n24+n39+n40=C$... c23b;	$n23+n27+n39+n43=C$... c23c;
$n24+n21+n28+n25=C$... c24a;	$n24+n21+n40+n37=C$... c24b;	$n24+n28+n40+n44=C$... c24c;
$n25+n26+n29+n30=C$... c25a;	$n25+n26+n41+n42=C$... c25b;	$n25+n29+n41+n45=C$... c25c;
$n26+n27+n30+n31=C$... c26a;	$n26+n27+n42+n43=C$... c26b;	$n26+n30+n42+n46=C$... c26c;
$n27+n28+n31+n32=C$... c27a;	$n27+n28+n43+n44=C$... c27b;	$n27+n31+n43+n47=C$... c27c;

n28+n25+n32+n29=C ... c28a;	n28+n25+n44+n41=C ... c28b;	n28+n32+n44+n48=C ... c28c;
n29+n30+n17+n18=C ... c29a;	n29+n30+n45+n46=C ... c29b;	n29+n17+n45+n33=C ... c29c;
n30+n31+n18+n19=C ... c30a;	n30+n31+n46+n47=C ... c30b;	n30+n18+n46+n34=C ... c30c;
n31+n32+n19+n20=C ... c31a;	n31+n32+n47+n48=C ... c31b;	n31+n19+n47+n35=C ... c31c;
n32+n29+n20+n17=C ... c32a;	n32+n29+n48+n45=C ... c32b;	n32+n20+n48+n36=C ... c32c;
n33+n34+n37+n38=C ... c33a;	n33+n34+n49+n50=C ... c33b;	n33+n37+n49+n53=C ... c33c;
n34+n35+n38+n39=C ... c34a;	n34+n35+n50+n51=C ... c34b;	n34+n38+n50+n54=C ... c34c;
n35+n36+n39+n40=C ... c35a;	n35+n36+n51+n52=C ... c35b;	n35+n39+n51+n55=C ... c35c;
n36+n33+n40+n37=C ... c36a;	n36+n33+n52+n49=C ... c36b;	n36+n40+n52+n56=C ... c36c;
n37+n38+n41+n42=C ... c37a;	n37+n38+n53+n54=C ... c37b;	n37+n41+n53+n57=C ... c37c;
n38+n39+n42+n43=C ... c38a;	n38+n39+n54+n55=C ... c38b;	n38+n42+n54+n58=C ... c38c;
n39+n40+n43+n44=C ... c39a;	n39+n40+n55+n56=C ... c39b;	n39+n43+n55+n59=C ... c39c;
n40+n37+n44+n41=C ... c40a;	n40+n37+n56+n53=C ... c40b;	n40+n44+n56+n60=C ... c40c;
n41+n42+n45+n46=C ... c41a;	n41+n42+n57+n58=C ... c41b;	n41+n45+n57+n61=C ... c41c;
n42+n43+n46+n47=C ... c42a;	n42+n43+n58+n59=C ... c42b;	n42+n46+n58+n62=C ... c42c;
n43+n44+n47+n48=C ... c43a;	n43+n44+n59+n60=C ... c43b;	n43+n47+n59+n63=C ... c43c;
n44+n41+n48+n45=C ... c44a;	n44+n41+n60+n57=C ... c44b;	n44+n48+n60+n64=C ... c44c;
n45+n46+n33+n34=C ... c45a;	n45+n46+n61+n62=C ... c45b;	n45+n33+n61+n49=C ... c45c;
n46+n47+n34+n35=C ... c46a;	n46+n47+n62+n63=C ... c46b;	n46+n34+n62+n50=C ... c46c;
n47+n48+n35+n36=C ... c47a;	n47+n48+n63+n64=C ... c47b;	n47+n35+n63+n51=C ... c47c;
n48+n45+n36+n33=C ... c48a;	n48+n45+n64+n61=C ... c48b;	n48+n36+n64+n52=C ... c48c;
n49+n50+n53+n54=C ... c49a;	n49+n50+n1+n2=C ... c49b;	n49+n53+n1+n5=C ... c49c;
n50+n51+n54+n55=C ... c50a;	n50+n51+n2+n3=C ... c50b;	n50+n54+n2+n6=C ... c50c;
n51+n52+n55+n56=C ... c51a;	n51+n52+n3+n4=C ... c51b;	n51+n55+n3+n7=C ... c51c;
n52+n49+n56+n53=C ... c52a;	n52+n49+n4+n1=C ... c52b;	n52+n56+n4+n8=C ... c52c;
n53+n54+n57+n58=C ... c53a;	n53+n54+n5+n6=C ... c53b;	n53+n57+n5+n9=C ... c53c;
n54+n55+n58+n59=C ... c54a;	n54+n55+n6+n7=C ... c54b;	n54+n58+n6+n10=C ... c54c;
n55+n56+n59+n60=C ... c55a;	n55+n56+n7+n8=C ... c55b;	n55+n59+n7+n11=C ... c55c;
n56+n53+n60+n57=C ... c56a;	n56+n53+n8+n5=C ... c56b;	n56+n60+n8+n12=C ... c56c;
n57+n58+n61+n62=C ... c57a;	n57+n58+n9+n10=C ... c57b;	n57+n61+n9+n13=C ... c57c;
n58+n59+n62+n63=C ... c58a;	n58+n59+n10+n11=C ... c58b;	n58+n62+n10+n14=C ... c58c;
n59+n60+n63+n64=C ... c59a;	n59+n60+n11+n12=C ... c59b;	n59+n63+n11+n15=C ... c59c;
n60+n57+n64+n61=C ... c60a;	n60+n57+n12+n9=C ... c60b;	n60+n64+n12+n16=C ... c60c;
n61+n62+n49+n50=C ... c61a;	n61+n62+n13+n14=C ... c61b;	n61+n49+n13+n1=C ... c61c;
n62+n63+n50+n51=C ... c62a;	n62+n63+n14+n15=C ... c62b;	n62+n50+n14+n2=C ... c62c;
n63+n64+n51+n52=C ... c63a;	n63+n64+n15+n16=C ... c63b;	n63+n51+n15+n3=C ... c63c;
n64+n61+n52+n49=C ... c64a;	n64+n61+n16+n13=C ... c64b;	n64+n52+n16+n4=C ... c64c;

** 対称型補数の定義式: CC=65; **

n1+n64=n2+n63=n3+n62=n4+n61=n5+n60=n6+n59=n7+n58=n8+n57=
n9+n56=n10+n55=n11+n54=n12+n53=n13+n52=n14+n51=n15+n50=n16+n49=
n17+n48=n18+n47=n19+n46=n20+n45=n21+n44=n22+n43=n23+n42=n24+n41=
n25+n40=n26+n39=n27+n38=n28+n37=n29+n36=n30+n35=n31+n34=n32+n33=CC;
n33+n32=n34+n31=n35+n30=n36+n29=n37+n28=n38+n27=n39+n26=n40+n25=
n41+n24=n42+n23=n43+n22=n44+n21=n45+n20=n46+n19=n47+n18=n48+n17=
n49+n16=n50+n15=n51+n14=n52+n13=n53+n12=n54+n11=n55+n10=n56+n9=
n57+n8=n58+n7=n59+n6=n60+n5=n61+n4=n62+n3=n63+n2=n64+n1=CC;

** 基本定義式: C=130; **

n1+n2+n3+n4=C ... b1; n1+n5+n9+n13=C ... b2; n1+n17+n33+n49=C... b3;
n1+n22+n43+n64=C ... p1;

対称型補数は、立方体の中心点に関して相互に対称の位置にある2数の間に定和条件を要請したものである。対称の位置にある2数(a, b)の間に $a+b=65$ が成り立つ時、(a, b)を「65の補数」と言う。あるいは、「aは、bの補数」とも、「bは、aの補数」とも言う。

果たして、上の定義式で望みの「魔方陣」が構成できるものだろうか？

12. 正方連結対称立体四方陣が構成不可能な理由

ところが、どうプログラムを組んでも、これが構成できずに「解無し」となる。最初は自分のミスかと思ひ、何度も構成・校正を試みたが、結果は常に **No!** であった。

