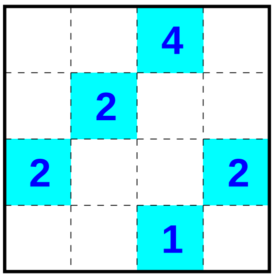
Walls

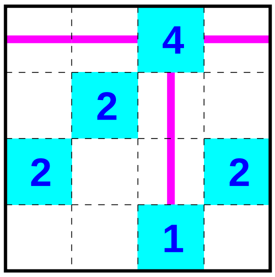
This is one of a number of puzzles which I have presented on the web site invented by Naoki Inaba. The others included here are **L-Panel** and **Marupeke**. Another of his puzzles he called **Block Number** which is now appearing in some newspapers as **Suguru** in a trivially small form which will exercise the mind for a minute or so. Larger forms I find tricky to make, and as they are also rather simple to solve I have left them off the menu. Alex Bellos reckons that Inaba is the most prolific puzzle inventor of our age so there should be plenty more to come.

The object of this puzzle is very simple – fill each empty cell with a wall section, pointing either across or up and down. The number given in a shaded cell is the total number of wall sections which join to that cell. Every empty cell must be filled with a wall section with no ambiguity as to its orientation, but it is not required that all wall sections lead to a shaded cell.

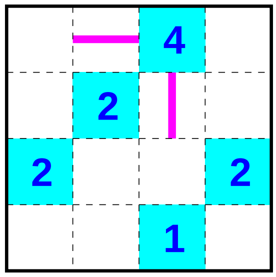
The diagram **A** below is an initial puzzle. In **B** all the possible wall sections which can be joined to the 4 in the top row are shown, and as there are five sections, one must be rejected. What is important here is that there are two wall sections which cannot be chosen for rejection, as in **C**. These wall sections now reduce the possibilites for the 2 on the second row to three positions, and **D** shows the one of these which must be chosen. This one then restricts the 2 in the left column to three possibilities, again with one which must be chosen as in **E**.



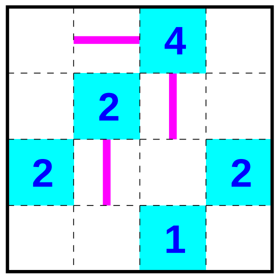
A



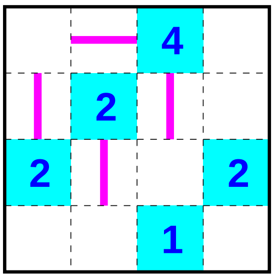
B



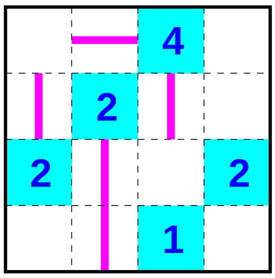
C



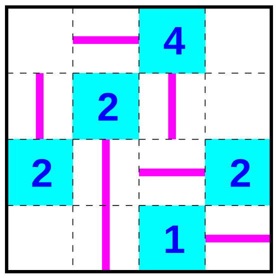
D



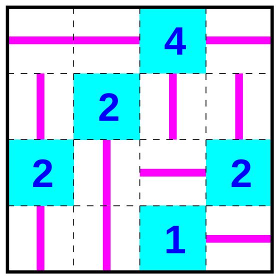
E



F



G



H

The 2 on the second row is now determined, giving **F**. Here we see two cells which adjoin the lone 1, but it is obvious which we have to choose, as setting the vertical option would join to the section already in place and would give too many sections from the 1; this wall section must therefore be set across as in **G**, contributing to the score for the 2 in the fourth column. Only one more section is needed to satisfy this 2, so the top cell in the fourth column must be set across. This and the last available wall section is needed by the 4, and the puzzle is then solved as in **H**.

I have found these puzzles quite infuriating to make at times, as I make a small change to get what I'm aiming for, and suddenly there are a host of other solutions. But then the next one that I try to make smiles at me from the drawing board and I wonder whether I'm being teased! When you've had your fill with these try making some yourself as this is just as much fun – when you have worked out some strategy for avoiding or rectifying the multiple solutions.

Walls 7×7

1

		3				4
			1			
	6				6	
3			3			0
	5				2	
			5			
1				4		

2

		5				
	1				3	
		5		2		
4			4			5
		4		2		
	1				2	
				3		

3

2		3			1	
				2		
	3					3
			5			
4					2	
		0				
	2			2		4

4

	1				5	
			4			
5						2
		3	2	2		
3						5
			3			
	3				3	

5

		3				4
	3		2			
5					1	
			2			
	4					3
			3		2	
4				8		

6

			5			
	4				3	
2						3
		6		3		
3						5
	4				2	
			5			

7

	1		2			2
	2			4		
4				4		
5						2
		3				5
		4			2	
3			1		1	

8

			2			3
	2			0		
2						
	4				3	
						7
		5			2	
4			6			

Walls 9×9

1

	4				5			
				5			2	
3					6			
		6						7
	4			3			4	
4						5		
			3					4
	4			6				
			3				2	

2

5					6			
		5					2	
				6				
	4					3		
			6					7
5					4			
		4					2	
				5				
	3					7		

3

	4				3			
		3	3				4	
6						2		
	3		4			1		
				4				
		4			2		6	
		0						5
	4				2	2		
			4				2	

4

4				5				5
	2						1	
			4		4			
		3				4		
3				2				4
		3				1		
			7		4			
	3						4	
3				4				5

5

			3		3			
	2			3			3	
				1				
2		4				2		4
			5		6			
2		2				3		3
				4				
	4			5			2	
			3		3			

6

			6					
6					5			4
	0							
		3				2	3	
5				5				
							3	
	2		6					2
4			3			3		
					6			2

Walls 10×10

1

3			5					5	
							4		
	5		5		6				
							2		4
		4							
							5		
5		3							
				8		5		2	
		4							
	4					6			5

2

5							4		
			4					4	
0						5			4
	5		1						
				7			6		
		5			6				
						3		6	
5			2						1
	5					7			
		5							3

3

5			3				4		
				8					2
		0			2				
			5					5	
4					2		3		
		3		4					6
	4					6			
				5			2		
4					4				
		4				3			4

4

4				6	4				2
	2							3	
		0					2		
			4			4			
2									3
6									5
			3			3			
		3					3		
	2							3	
4				3	6				3

5

	2			2	2				
	1							6	2
		3					3		
			3	1		3			
5						6			3
3			3						6
			4		2	3			
		2					0		
1	3							4	
				4	3			2	

6

8				5				3	
						3			
		5					5		
			3		6				6
	7					0			
			3					5	
7				4		1			
		5					5		
			6						
	7					3			6

Walls 10×10

7

				4			3		
		3				3			
4				2	2			6	
	3		2			3			
		6					4		4
6		6					5		
			3			2		4	
	3			1	2				1
			1				4		
		5			3				

8

		3				4			
	2			4				2	
		4			3		2		0
2			0	1		2			
		3				3		2	
	5		2				2		
			4		2	5			4
3		2		2			2		
	2				2			1	
			2				2		

9

2			5				3		
		0			6				
	5						5		4
3				5		1			
			0					6	
	2					3			
			2		2				3
6		4						3	
				7			6		
		2				3			3

10

			2	2					3
	6						5		
			5					4	
2		6				3			
3				3					
					5				5
			2				4		3
	5					3			
		8						2	
3					1	0			

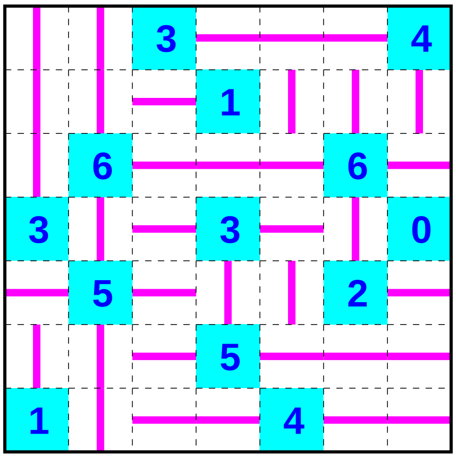
11

	2	3					3		
						1			2
4			1	2					1
	6						5		
				2	4		1		
		3		3	1				
		1						5	
3					2	2			4
2			5						
		3					4	2	

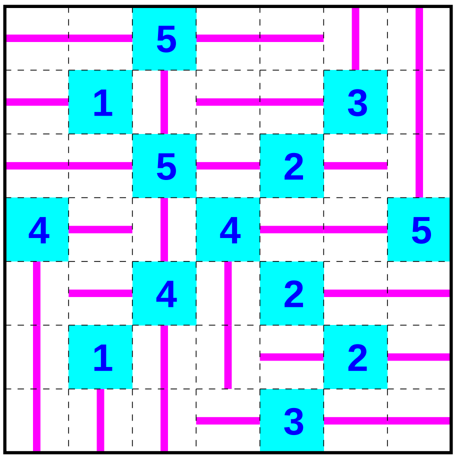
12

	7				0				4
		3							
3		3					5		
		4		2				4	
						6			5
2			3						
	8				4		4		
		4					3		5
							4		
4				2				1	

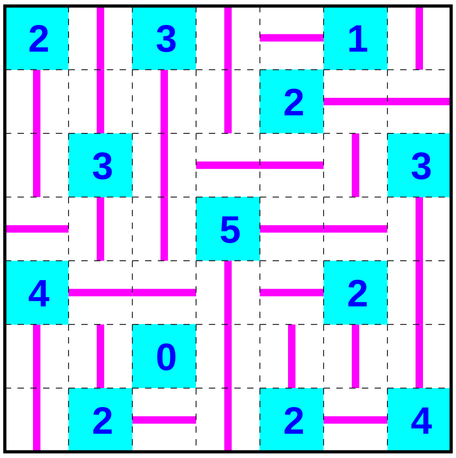
Walls 7×7 **solutions**



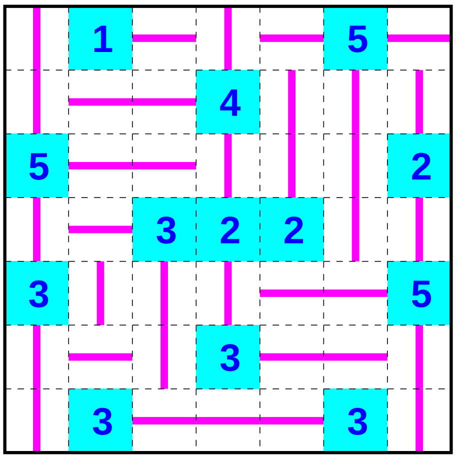
1



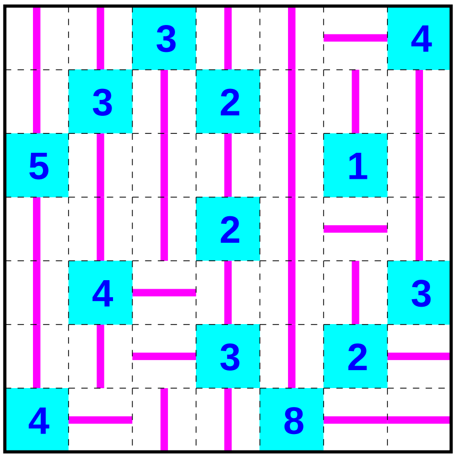
2



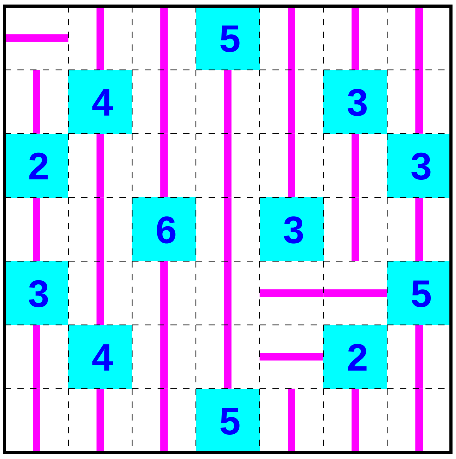
3



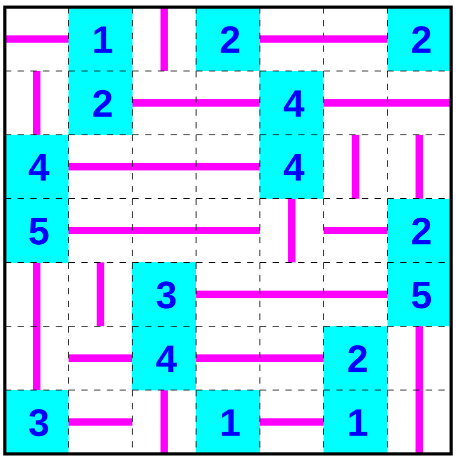
4



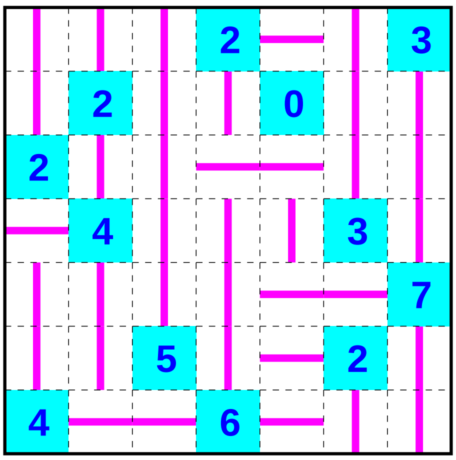
5



6

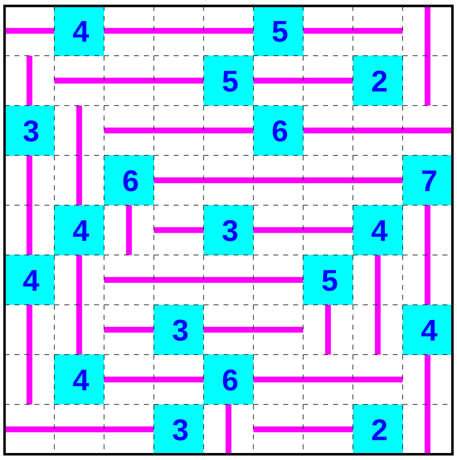


7

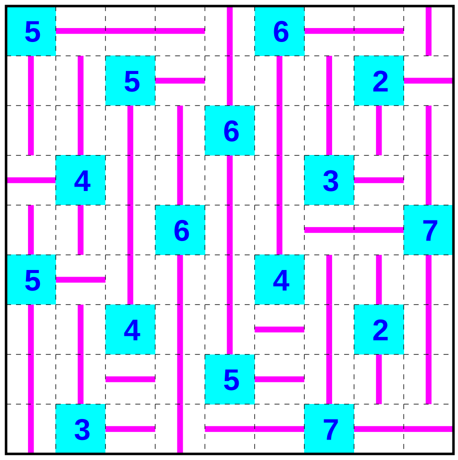


8

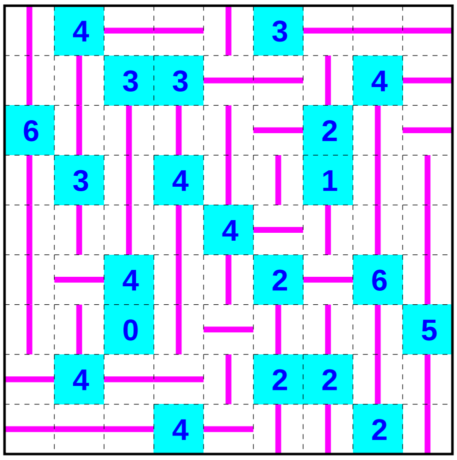
Walls 9×9 **solutions**



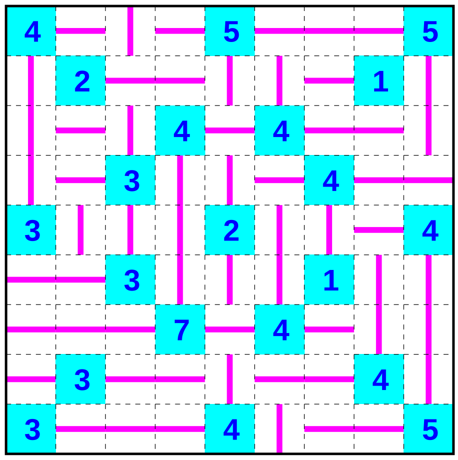
1



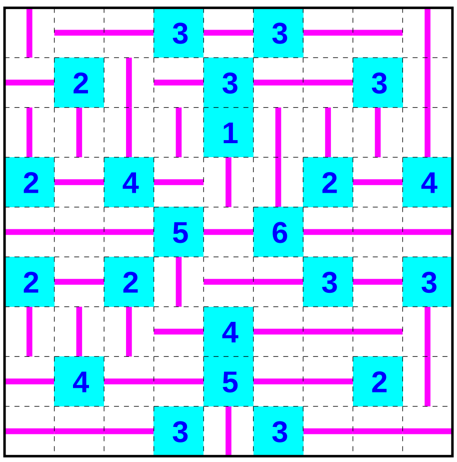
2



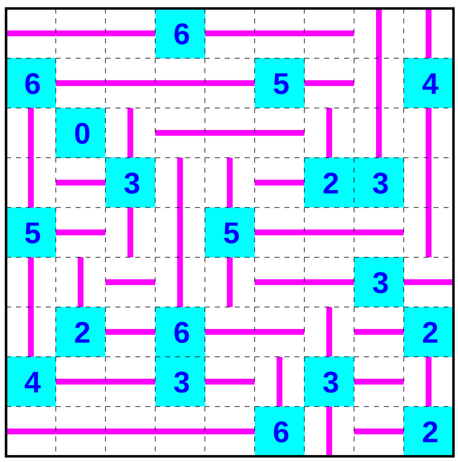
3



4

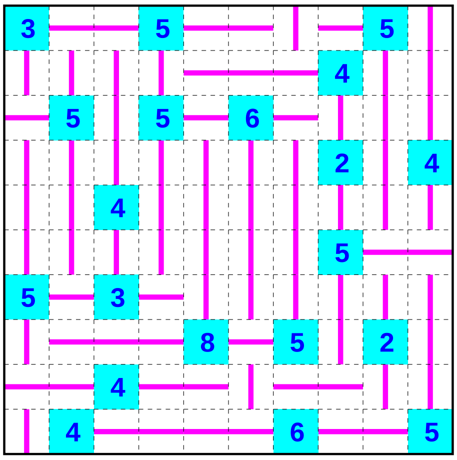


5



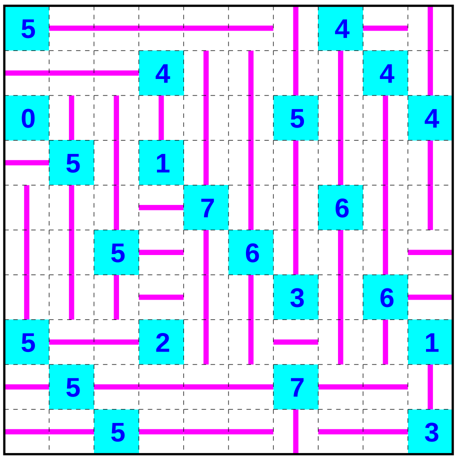
6

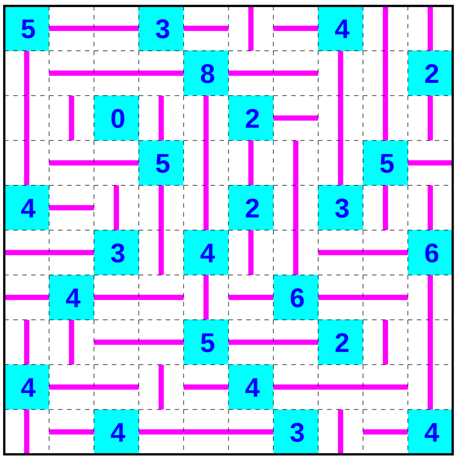
Walls 10×10 **solutions**



1

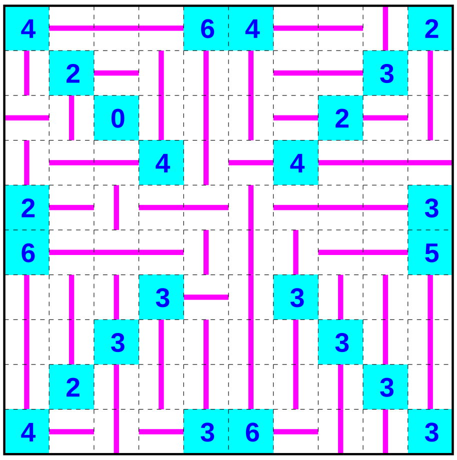
2

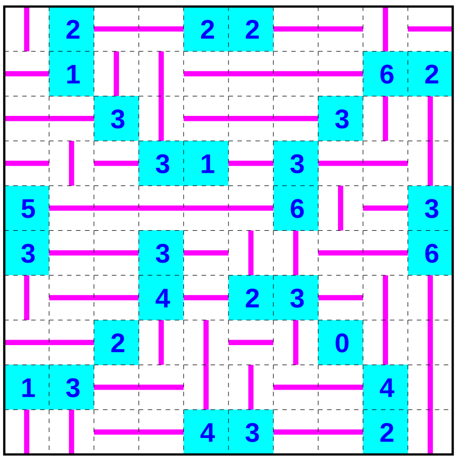




3

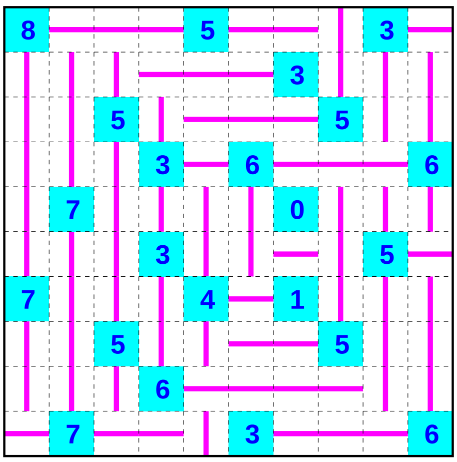
4



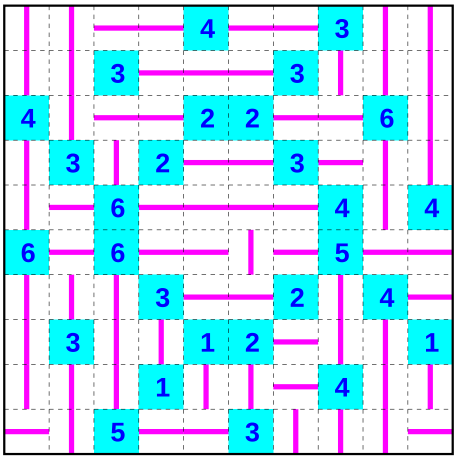


5

6

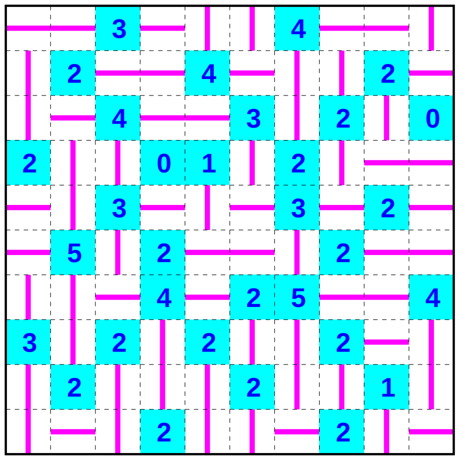


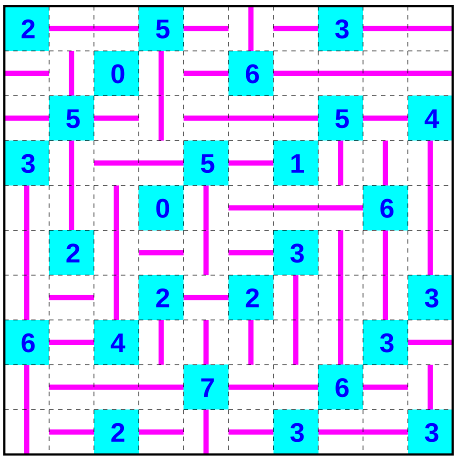
Walls 10×10 **solutions**



7

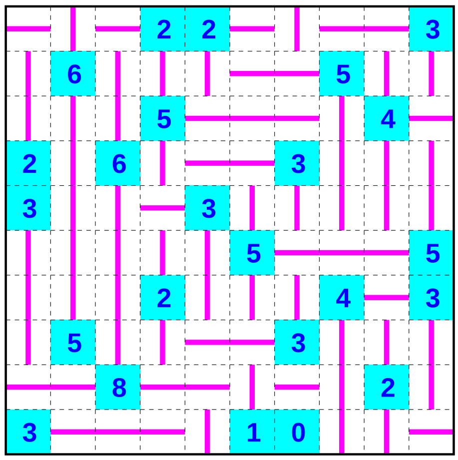
8

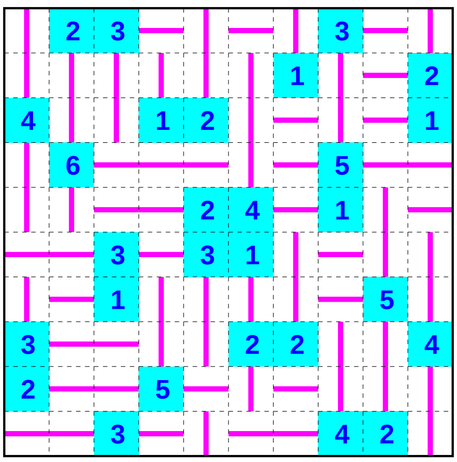




9

10





11

12