Exclusively ORs has hired you to implement their exclusive-OR-based problem solver. The solver is given a problem description in three parts:

- a list of all possible propositions, by default truth unknown
- a list of those propositions known to be true
- a list of lists of propositions – the exclusive ORs (XORs)

Assume propositions are just simple names. The solver tries to assign true and false values to the unknown propositions such that each XOR list ends up with one true, and all the rest false. Two obvious inference rules for doing this are:

- A true + XOR list [A B C] means B and C must be false
- A and B false + [A B C] means C must be true

This can solve simple 3x3 Sudoku-like puzzles, where 1, 2 and 3 have to appear exactly once in every row and column. The puzzle below has exactly one solution. The input to the solver would be:

- the list [A1 A2 A3 B1 B2 … I1 I2 I3] for the propositions "cell A has a 1," "cell A has a 2," "cell A has a 3," "cell B has a 1," etc.
- the list [A2 F3] for the 2 given values in this puzzle
- the list of 27 XOR lists below that are true for all 3x3 puzzles

2	B	C
D	E	3
G	H	I

-- Every cell has a 1, 2 or 3
[A1 A2 A3] [B1 B2 B3] [C1 C2 C3] [D1 D2 D3] [E1 E2 E3] [F1 F2 F3] [G1 G2 G3] [H1 H2 H3] [I1 I2 I3]
-- Every row has 1, 2 and 3
[A1 B1 C1] [D1 E1 F1] [G1 H1 I1] [A2 B2 C2] [D2 E2 F2] [G2 H2 I2] [A3 B3 C3] [D3 E3 F3] [G3 H3 I3]
-- Every column has a 1, 2 and 3)
[A1 D1 G1] [B1 E1 H1] [C1 F1 I1] [A2 D2 G2] [B2 E2 H2] [C2 F2 I2] [A3 D3 G3] [B3 E3 H3] [C3 F3 I3]

By repeatedly applying the 2 inference rules, the solver can solve the entire puzzle, e.g.,

- A2 true + [A2 D2 G2] means D2 is false
- F3 true + [D3 E3 F3] means D3 is false
- D2 and D3 false + [D1 D2 D3] means D1 is true, and so on

Use any programming language you like to implement the XOR solver. Use appropriate data structures for propositions, their true/false values, and the XOR lists. Code should be correct, general for any XOR problem, not just 3x3 Sudoku, and reasonably efficient.