Backtracking

1. **N-Queens**

**Send Feedback**

**You are given N, and for a given N x N chessboard, find a way to place N queens such that no queen can attack any other queen on the chess board. A queen can be killed when it lies in the same row, or same column, or the same diagonal of any of the other queens. You have to print all such configurations.**

**Input Format :**

Line 1 : Integer N

**Output Format :**

One Line for every board configuration.

Every line will have N\*N board elements printed row wise and are separated by space

**Note : Don't print anything if there isn't any valid configuration.**

Constraints :

***1<=N<=10***

Sample Input 1:

4

Sample Output 1 :

0 1 0 0 0 0 0 1 1 0 0 0 0 0 1 0

0 0 1 0 1 0 0 0

1. **Sudoku Solver**

**Send Feedback**

Given a 9\*9 sudoku board, in which some entries are filled and others are 0 (0 indicates that the cell is empty), you need to find out whether the Sudoku puzzle can be solved or not i.e. return true or false.

**Input Format :**

9 Lines where ith line contains ith row elements separated by space

**Output Format :**

true or false

**Sample Input :**

9 0 0 0 2 0 7 5 0

6 0 0 0 5 0 0 4 0

0 2 0 4 0 0 0 1 0

2 0 8 0 0 0 0 0 0

0 7 0 5 0 9 0 6 0

0 0 0 0 0 0 4 0 1

0 1 0 0 0 5 0 8 0

0 9 0 0 7 0 0 0 4

0 8 2 0 4 0 0 0 6

**Sample Output :**

true