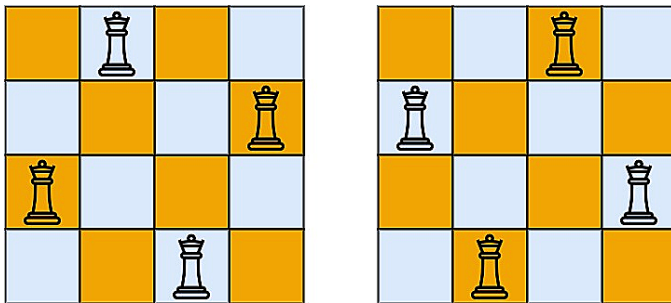


1) The **n-queens** puzzle is the problem of placing n queens on an n x n chessboard such that no two queens attack each other.

Given an integer n, return *all distinct solutions to the n-queens puzzle*. You may return the answer in **any order**.

Each solution contains a distinct board configuration of the n-queens' placement, where 'Q' and '.' both indicate a queen and an empty space, respectively.

Example 1:



Input: n = 4

Output: [[".Q...", "...Q", "Q...", "...Q."], ["..Q.", "Q...", "...Q", ".Q.."]]

Explanation: There exist two distinct solutions to the 4-queens puzzle as shown above

Example 2:

Input: n = 1

Output: [["Q"]]

2) Write a program to convert a number into a mono-digit number.

Conditions:

- a) You are allowed to add and subtract the consecutive digits (starting from left).
- b) You are allowed to do only one operation on a digit.
- c) You cannot perform any operation on a resultant digit of the previous operation.
- d) Your code should also find if a given number cannot be converted to a mono digit number.

Input	Output
72581	7(2+5)81 77(8-1) 777
3962	cannot create a mono digit number

3) Write a program to print all permutations of a given string. Note here you need to take all combinations as well, say for the input ABC the output should be as follows:

Input: ABC

Output:

A

B C

AB AC BA BC CA CB

ABC ACB BCA BAC CBA CAB

4) Write a program to solve a Sudoku puzzle by filling the empty cells.

A sudoku solution must satisfy **all of the following rules**:

1. Each of the digits 1-9 must occur exactly once in each row.
2. Each of the digits 1-9 must occur exactly once in each column.
3. Each of the digits 1-9 must occur exactly once in each of the 9 3x3 sub-boxes of the grid.

The '.' character indicates empty cells

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
			8			7	9	

Input: board =

```
[["5","3",".",".","7",".",".","."],["6",".",".","1","9","5",".","."],  
[".","9","8",".",".",".","6","."],["8",".",".",".","6",".","."],  
[".","3"],["4",".",".","8",".","3",".","."],["7",".",".",".","2","."],  
[".","6"],[".","6",".",".",".","2","8","."],[".",".",".","4","1",  
"9",".","5"],[".",".",".","8",".","","7","9"]]
```

Output:

```
[["5","3","4","6","7","8","9","1","2"],["6","7","2","1","9","5","3","4","8"],  
["1","9","8","3","4","2","5","6","7"],["8","5","9","7","6","1","4","2","3"],  
["4","2","6","8","5","3","7","9","1"],["7","1","3","9","2","4","8","5","6"],  
["9","6","1","5","3","7","2","8","4"],["2","8","7","4","1","9","6","3","5"],  
["3","4","5","2","8","6","1","7","9"]]
```

Explanation: The input board is shown above and the only valid solution is shown below:

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9