In [2]:
```
# Title :- Implement N-Queen problem using Backtracking.
```

In [2]:
```
def backtracking_n_queen(n):

    def is_safe(board, row, col):
        return all(board[i] != row and board[i] != row+col-i and board[i] != row-

    def solve_util(board, col):
        if col >= n:
            return True
        for row in range(n):
            if is_safe(board, row, col):
                board[col] = row
                if solve_util(board, col + 1):
                    return True
        return False

    board = [-1] * n
    if solve_util(board, 0):
        for row in board:
            print(' '.join(['Q' if i == row else '-' for i in range(n)]))
    else:
        print("No solution Exists")

n = 8
backtracking_n_queen(4)
```

```
- Q - -
- - - Q
Q - - -
- - Q -
```