```
def is_valid(board, row, col):
    for existing_col, existing_row in enumerate(board):
        if existing_row == row or abs(existing_col - col) == abs(existing_row - row):
            return False
    return True

def solve_n_queens(n):
    solutions = []
    def dfs(board):
        if len(board) == n:
            solutions.append(board.copy())
            return
        current_col = len(board)
        for row in range(n):
            if is_valid(board, row, current_col):
                board.append(row)
                dfs(board)
                board.pop()
    dfs([])
    return solutions

def display_solution(board):
    n = len(board)
    for row in range(n):
        line = []
        for col in range(n):
            if board[col] == row:
                line.append('Q')
            else:
                line.append('.')
        print(' '.join(line))
    print()

if __name__ == "__main__":
    for n in [4, 5, 6]:
        solutions = solve_n_queens(n)
        print(f"Solutions for {n}-Queens:")
        for solution in solutions:
```

```
        display_solution(solution)
    print(f"Total solutions: {len(solutions)}\n")
```

Output:

```
Solutions for 4-Queens:
. . Q .
Q . . .
. . . Q
. Q . .

. Q . .
. . . Q
Q . . .
. . Q .

Total solutions: 2

Solutions for 5-Queens:
Q . . . .
. . . Q .
. Q . . .
. . . . Q
. . Q . .

Q . . . .
. . Q . .
. . . . Q
. Q . . .
. . . Q .

. . Q . .
Q . . . .
. . . Q .
. Q . . .
. . . . Q

. . . Q .
Q . . . .
. . Q . .
. . . . Q
. Q . . .

. Q . . .
. . . Q .
Q . . . .
. . Q . .
. . . . Q

. . . . Q
. . Q . .
Q . . . .
. . . Q .
. Q . . .
. Q . . .
. . . . Q
. . Q . .
Q . . . .
. . . Q .

. . . . Q
. Q . . .
. . . Q .
Q . . . .
. . Q . .

. . . Q .
. Q . . .
. . . . Q
. . Q . .
Q . . . .

. . Q . .
. . . . Q
. Q . . .
. . . Q .
Q . . . .

Total solutions: 10
```

Solutions for 6-Queens:

```
. . . Q . .
Q . . . . .
. . . . Q .
. Q . . . .
. . . . . Q
. . Q . . .

. . . . Q .
. . Q . . .
Q . . . . .
. . . . . Q
. . . Q . .
. Q . . . .

. Q . . . .
. . . Q . .
. . . . . Q
Q . . . . .
. . Q . . .
. . . . Q .

. . Q . . .
. . . . . Q
. Q . . . .
. . . . Q .

Q . . . . .

. . . Q . .


Total solutions: 4
```