N-Queens Problem with Dynamic board

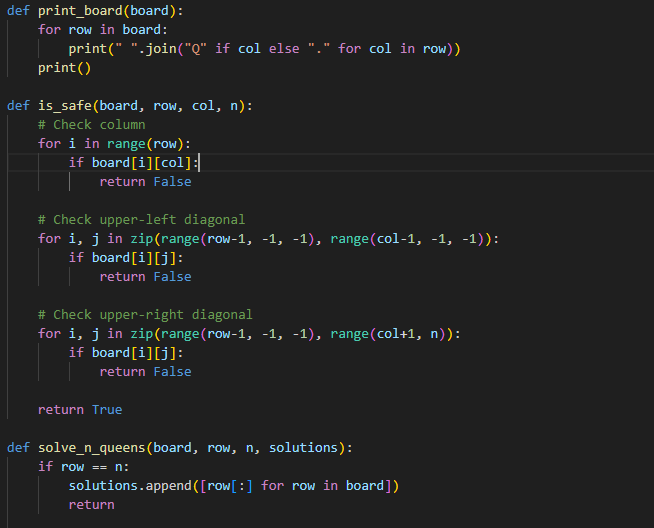
# Objective

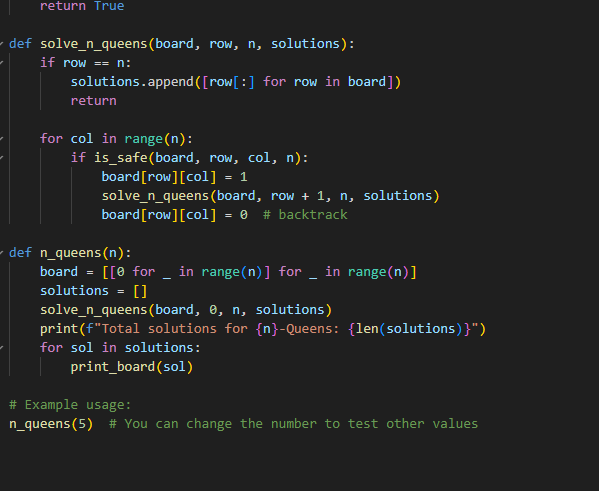
The objective of this task is to solve the N-Queens problem using backtracking in Python. The goal is to place N queens on an N×N chessboard such that no two queens attack each other.

# Algorithm Overview

The backtracking algorithm attempts to place a queen in each row while ensuring that the current placement does not result in any conflicts with previously placed queens. Conflicts are checked along columns and both diagonals. If a valid position is found, the algorithm moves to the next row. If no valid position exists in the current row, it backtracks to the previous row to try a new position.

# Python Code

  
Here the complete code for the n-queens solutions

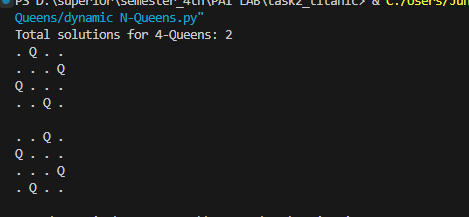


# Step-by-Step Explanation

The code defines helper functions to check if a position is safe for a queen, places queens row by row, and uses recursion with backtracking to explore all valid arrangements. Once a valid board configuration is found, it is stored as a solution.

# Sample Output

The function n\_queens(3) prints all valid 5x5 board arrangements with 5 queens such that no queens threaten each other.  
It also displays the total number of such valid arrangements.



# Conclusion

The N-Queens problem illustrates the power of recursion and backtracking in solving complex combinatorial problems. This Python implementation successfully demonstrates how to explore and validate board configurations using these techniques.