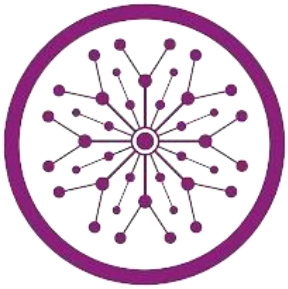
Programming for AI

Lab

Task 4



The Superior University

Faculty of Computer Science & Information Technology

2023-2027

Submitted To:

Sir Rasikh Ali

Submitted By:

Waqar Abbas

Roll No:

SU92-BSDSM-F23-010

Section:

4A

Department:

Software Engineering

Overview:

The N-Queens problem is a puzzle that involves placing N-queens on an N × N chessboard in such a way that no two queens can attack each other. This code dynamically solves the N-Queens problem, ensuring that no two queens are placed in the same row, column, or diagonal.

Code Overview:

This code solves the N-Queen problem using backtracking. This code involves:

- Recursively placing queens row by row.
- Checking weather placing a queen is save or not.
- Backtracking when necessary.
- Printing all the possible solutions.

Functions:

- **def print_board(board):**
  printing the chessboard where "Q" represents the placed queen and "." Represents the empty space.

  ### Parameters:

  - borad: A 2-D array representing the chessboard.

- **def placing_safe(board, row, col, n):**
  Checking weather placing a queen is save or not at the position.
  - Checking upper left diagonal
  - Checking queen in same column
  - Checking upper right diagonal

  ### Parameters:

  - borad: Checking weather placing a queen is save or not.
  - row : row index where the queen is being placed.
  - col: column index where the queen is being placed.
  - n: size of chessboard (N*N).

  returns **True** if the queen is placed safely otherwise **False**.

- **def solution_Nqueen(board, row,n):**
  Recursively trying to place queen row by row while ensuring no queen is attacking.

  Parameters:

  - borad: Checking weather placing a queen is save or not.
  - row : the current row is processing.
  - n: size of chessboard (N*N).

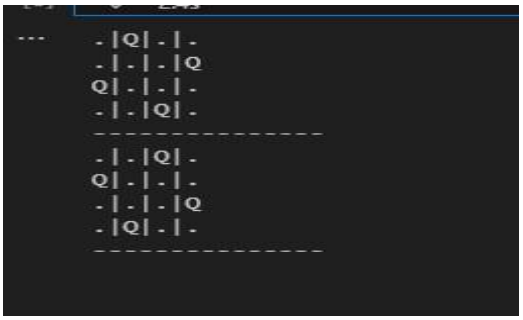
returns **True** if the solution is found otherwise **False**

- **def n_queens():**
  Tacking input from user and initializing the size of board.
  Prints the solution if found, otherwise prints "Solution not found".

Examples of the N-Queens problem solved for different values of N:

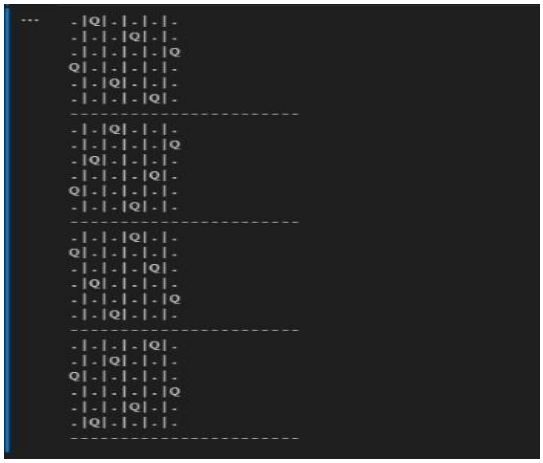
Example 1:

For **4x4** chessboard, every possible solution is:



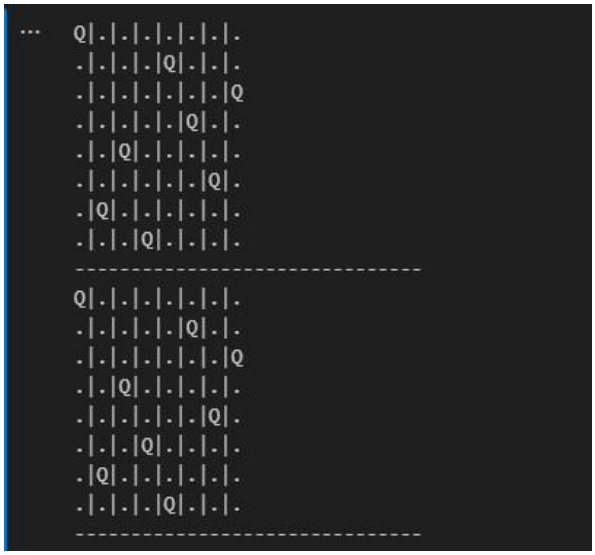
Example 2:

For **6x6** chessboard, every possible solution is:



Example 3:

For **8x8** chessboard, every possible solution is:



Example 4:

For 16x16 chessboard, every possible solution is: