Implement N-Queen and print all places where queen is placed.

CODE:

public class NQueen\_Placed{

    static void solveNQueens(boolean[][] board, int currentRow) {

        if (currentRow == board.length) {

            // If all queens are placed, print the board

            printBoard(board);

            System.out.println();

            return;

        }

        for (int col = 0; col < board[currentRow].length; col++) {

            if (isSafe(board, currentRow, col)) {

                board[currentRow][col] = true;

                solveNQueens(board, currentRow + 1);

                board[currentRow][col] = false; // Backtrack

            }

        }

    }

    static boolean isSafe(boolean[][] board, int row, int col) {

        // Check if there is a queen in the same column and above row

        for (int i = row; i >= 0; i--) {

            if (board[i][col]) {

                return false;

            }

        }

        // Check in the upper left diagonal

        for (int i = row, j = col; i >= 0 && j >= 0; i--, j--) {

            if (board[i][j]) {

                return false;

            }

        }

        // Check in the upper right diagonal

        for (int i = row, j = col; i >= 0 && j < board.length; i--, j++) {

            if (board[i][j]) {

                return false;

            }

        }

        return true;

    }

    static void printBoard(boolean[][] board) {

        int n = board.length;

        for (int i = 0; i < n; i++) {

            for (int j = 0; j < n; j++) {

                if (board[i][j]) {

                    System.out.print("Q ");

                } else {

                    System.out.print(". ");

                }

            }

            System.out.println();

        }

    }

    public static void main(String[] args) {

        int n = 4; // Number of queens and size of the board

        boolean[][] board = new boolean[n][n];

        solveNQueens(board, 0);

    }

}

OUTPUT :

A screenshot of a computer

Description automatically generated with medium confidence