Pract 4

```
import java.util.Scanner;

public class Main{

   private static int[] queens;

   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter the number of queens: ");
      int n = scanner.nextInt();
      scanner.close();

      queens = new int[n];
      solveNQueens(n);
   }

   private static void solveNQueens(int n) {
      if (placeQueen(0, n)) {
         System.out.println("Solution found:");
         printSolution(n);
      } else {
         System.out.println("No solution exists for " + n + " queens.");
      }
   }

   private static boolean placeQueen(int row, int n) {
      if (row == n) {
         return true; // All queens are placed successfully
      }

      for (int col = 0; col < n; col++) {
         if (isSafe(row, col)) {
            queens[row] = col; // Place queen at this position
            if (placeQueen(row + 1, n)) {
               return true; // If placing next queen is successful
            }
            // If placing next queen is not successful, backtrack
         }
      }
      return false; // If no position is found to place the queen
   }

   private static boolean isSafe(int row, int col) {
      // Check if the current queen can be placed in this column without threatening other queens
      for (int prevRow = 0; prevRow < row; prevRow++) {
         int prevCol = queens[prevRow];
         if (prevCol == col || Math.abs(row - prevRow) == Math.abs(col - prevCol)) {
            return false;
         }
      }
      return true;
   }

   private static void printSolution(int n) {
      // Print the chessboard with queens placed
```

```
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                if (queens[i] == j) {
                    System.out.print("1 ");
                } else {
                    System.out.print("0 ");
                }
            }
            System.out.println();
        }
    }
}
```