**package** lp3;

**public** **class** NQueens {

**private** **int** N;

**private** **int**[] queens;

**public** NQueens(**int** N) {

**this**.N = N;

**this**.queens = **new** **int**[N];

}

**public** **void** solveNQueens() {

**if** (placeQueens(0)) {

printQueens();

} **else** {

System.***out***.println("No solution exists.");

}

}

**private** **boolean** placeQueens(**int** row) {

**if** (row == N) {

**return** **true**;

}

**for** (**int** col = 0; col < N; col++) {

**if** (isSafe(row, col)) {

queens[row] = col;

**if** (placeQueens(row + 1)) {

**return** **true**;

}

// Backtrack

queens[row] = -1;

}

}

**return** **false**;

}

**private** **boolean** isSafe(**int** row, **int** col) {

**for** (**int** i = 0; i < row; i++) {

**if** (queens[i] == col || queens[i] - i == col - row || queens[i] + i == col + row) {

**return** **false**;

}

}

**return** **true**;

}

**private** **void** printQueens() {

**for** (**int** i = 0; i < N; i++) {

**for** (**int** j = 0; j < N; j++) {

**if** (queens[i] == j) {

System.***out***.print("Q ");

} **else** {

System.***out***.print(". ");

}

}

System.***out***.println();

}

}

**public** **static** **void** main(String[] args) {

**int** N = 4; // Change N to the desired value

NQueens nQueens = **new** NQueens(N);

nQueens.solveNQueens();

}

}