import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class Solution {

public void addSolution(List<List<String>> board, List<List<String>> ans) {

List<String> temp = new ArrayList<>();

for (List<String> row : board) {

temp.add(String.join("", row));

}

ans.add(temp);

}

public boolean isSafe(int row, int col, List<List<String>> board, int n) {

// Check the column

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

if (board.get(i).get(col).equals("Q")) {

return false;

}

}

// Check upper left diagonal

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

if (board.get(i).get(j).equals("Q")) {

return false;

}

}

// Check upper right diagonal

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

if (board.get(i).get(j).equals("Q")) {

return false;

}

}

return true;

}

public void solve(int row, List<List<String>> ans, List<List<String>> board, int n) {

if (row == n) {

addSolution(board, ans);

return;

}

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

if (isSafe(row, col, board, n)) {

board.get(row).set(col, "Q");

solve(row + 1, ans, board, n);

board.get(row).set(col, ".");

}

}

}

public List<List<String>> solveNQueens(int n) {

List<List<String>> board = new ArrayList<>();

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

List<String> row = new ArrayList<>();

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

row.add(".");

}

board.add(row);

}

List<List<String>> ans = new ArrayList<>();

solve(0, ans, board, n);

return ans;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the size of the board (e.g., for 4-queen problem, enter 4): ");

int size = scanner.nextInt();

Solution s = new Solution();

List<List<String>> ans = s.solveNQueens(size);

for (List<String> row : ans) {

for (String col : row) {

System.out.println(col);

}

System.out.println();

}

}

}