#include <stdio.h>

#define N 10

int attacked(int board[N][N], int row, int col, int n) {

int i, j;

//row

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

if (board[i][col]) {

return 1;

}

}

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

if (board[i][j]) {

return 1;

}

}

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

if (board[i][j]) {

return 1;

}

}

return 0;

}

void solve(int board[N][N], int row, int n, int \*count) {

int col;

if (row == n) {

(\*count)++;

printf("Solution %d:\n", \*count);

int i, j;

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

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

printf("%c ", board[i][j] ? 'Q' : '.');

}

printf("\n");

}

printf("\n");

return;

}

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

if (!attacked(board, row, col, n)) {

board[row][col] = 1;

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

board[row][col] = 0;

}

}

}

int main() {

int n, board[N][N], i, j, count = 0;

printf("Enter number of queens (max %d): ", N);

scanf("%d", &n);

// Initialize the board

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

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

board[i][j] = 0;

}

}

solve(board, 0, n, &count);

if (count == 0) {

printf("No solution found.\n");

}

return 0;

}Text

Description automatically generated