**#include <stdio.h>**

**#include <stdbool.h>**

**#include <stdlib.h>**

**void printSolution(int \*\*board, int N) {**

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

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

**printf("%2d ", board[i][j]);**

**}**

**printf("\n");**

**}**

**printf("\n");**

**}**

**bool isSafe(int \*\*board, int N, int row, int col) {**

**int i, j;**

**// Check this row on left side**

**for (i = 0; i < col; i++)**

**if (board[row][i])**

**return false;**

**// Check upper diagonal on left side**

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

**if (board[i][j])**

**return false;**

**// Check lower diagonal on left side**

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

**if (board[i][j])**

**return false;**

**return true;**

**}**

**bool solveNQUtil(int \*\*board, int N, int col) {**

**if (col >= N) {**

**printSolution(board, N);**

**return true;**

**}**

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

**if (isSafe(board, N, i, col)) {**

**board[i][col] = 1;**

**if (solveNQUtil(board, N, col + 1)) {**

**return true;**

**}**

**board[i][col] = 0; // BACKTRACK**

**}**

**}**

**return false;**

**}**

**void solveNQ(int N) {**

**int \*\*board = (int \*\*)malloc(N \* sizeof(int \*));**

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

**board[i] = (int \*)malloc(N \* sizeof(int));**

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

**board[i][j] = 0;**

**}**

**}**

**if (!solveNQUtil(board, N, 0)) {**

**printf("Solution does not exist\n");**

**}**

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

**free(board[i]);**

**}**

**free(board);**

**}**

**int main() {**

**int N;**

**printf("Enter the value of N: ");**

**scanf("%d", &N);**

**if (N <= 0) {**

**printf("N must be a positive integer\n");**

**return 1;**

**}**

**solveNQ(N);**

**return 0;**

**}**

