8 وزیر :زهرا محمدی

```
#include <stdio.h>
#include <stdlib.h>

#define N 8

int board[N][N]; // صفحه شطرنج

// این تابع برای چاپ جواب استفاده میشه
void printSolution() {
  for (int i = 0; i < N; i++) {
    for (int j = 0; j < N; j++)
      printf(" %d ", board[i][j]);
    printf("\n");
  }
}

// تابعی برای بررسی امکان قرار دادن وزیر در board[row][col]
int isSafe(int row, int col) {
  int i, j;

// بررسی ستون
  for (i = 0; i < row; i++)
    if (board[i][col])
      return 0;

// بررسی قطر راست بالا
  for (i = row, j = col; i >= 0 && j < N; i--, j++)
    if (board[i][j])
      return 0;

// بررسی قطر چپ بالا
  for (i = row, j = col; i >= 0 && j >= 0; i--, j--)
    if (board[i][j])
      return 0;

  return 1;
}

// تابع بازگشتی برای حل مسئله 8 وزیر
int solveNQUtil(int row) {
// اگر تمام وزیرها قرار گرفتند یعنی  جواب پیدا شده
  if (row >= N)
```

```
    return 1;

// بررسی این که آیا وزیر می‌تونه در هر ستونی در ردیف row قرار بگیره
  for (int col = 0; col < N; col++) {
// اگه می‌توان وزیر را قرار داد
    if (isSafe(row, col)) {
      board[row][col] = 1;

// بازگشتی به سایر ردیف‌ها
      if (solveNQUtil(row + 1))
        return 1;

// اگه قرار دادن وزیر در board[i][col] به جواب نرسه اونو خالی می‌کنیم
      board[row][col] = 0;
    }
  }

// اگر وزیر نتونه در هیچ کدام از ستون‌ها در ردیف row قرار بگیره , بازگشت
  return 0;
}

// این تابع مسئله 8 وزیر را حل می‌کنه با استفاده از ()solveNQUtil
int solveNQ() {
  for (int i = 0; i < N; i++)
    for (int j = 0; j < N; j++)
      board[i][j] = 0;

  if (solveNQUtil(0) == 0) {
    printf("Solution does not exist");
    return 0;
  }

  printSolution();
  return 1;
}

int main() {
  solveNQ();
  return 0;
}
```