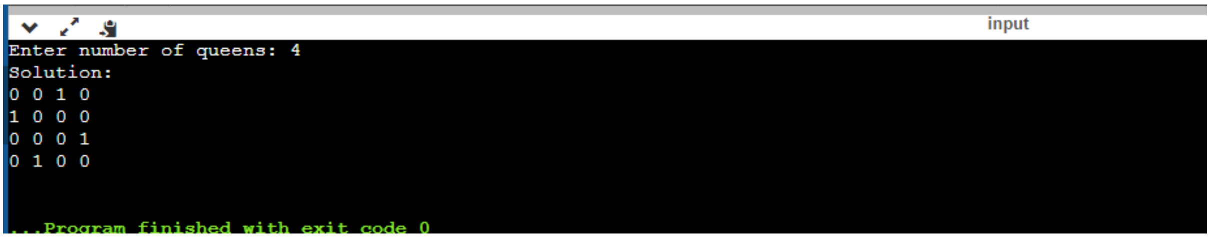
```
#include <stdio.h>
#include <stdbool.h>
#define N 10
int board[N][N];
bool isSafe(int row, int col, int n) {
    for (int i = 0; i < col; i++) {
        if (board[row][i])
            return false;    }
    for (int i = row, j = col; i >= 0 && j >= 0; i--, j--) {
        if (board[i][j])
            return false;    }
    for (int i = row, j = col; j >= 0 && i < n; i++, j--) {
        if (board[i][j])
            return false;    }
    return true; }
bool solveNQueens(int col, int n) {
    if (col >= n)
        return true;
    for (int i = 0; i < n; i++) {
        if (isSafe(i, col, n)) {
            board[i][col] = 1;
            if (solveNQueens(col + 1, n))
                return true;
            board[i][col] = 0;    }    }
 return false; }
void printSolution(int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            printf("%d ", board[i][j]);    }
        printf("\n");   } }
int main() {
    int n;

    printf("Enter number of queens: ");

    scanf("%d", &n);

    if (n <= 0 || n > N) {

        printf("Invalid input. N should be between 1 and %d.\n", N);

        return 1;    }

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

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

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

    if (!solveNQueens(0, n)) {

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

        return 0;  }

    printf("Solution:\n");

    printSolution(n);

  return 0;

}
```



```
Enter number of queens: 4
Solution:
0 0 1 0
1 0 0 0
0 0 0 1
0 1 0 0


...Program finished with exit code 0
```