```
#include <iostream>
#include <vector>

using namespace std;

const int N = 8;

bool isSafe(vector<vector<int> >& board, int row, int col)
{
	for (int x = 0; x < col; x++)
		if (board[row][x] == 1)
			return false;
	for (int x = row, y = col; x >= 0 && y >= 0; x--, y--)
		if (board[x][y] == 1)
			return false;
	for (int x = row, y = col; x < N && y >= 0; x++, y--)
		if (board[x][y] == 1)
			return false;
	return true;
}

bool solveNQueens(vector<vector<int> >& board, int col)
{
	if (col == N) {
		for (int i = 0; i < N; i++) {
			for (int j = 0; j < N; j++)
				cout << board[i][j] << " ";
			cout << endl;
		}
```

```
            cout << endl;
            return true;
        }
        for (int i = 0; i < N; i++) {
            if (isSafe(board, i, col)) {
                board[i][col] = 1;
                if (solveNQueens(board, col + 1))
                    return true;
                board[i][col] = 0;
            }
        }
        return false;
}

int main()
{
        vector<vector<int> > board(N, vector<int>(N, 0));
        if (!solveNQueens(board, 0))
            cout << "No solution found";
        return 0;
}
```

Output:-

1 0 0 0 0 0 0 0

0 0 0 0 0 0 1 0

0 0 0 0 1 0 0 0

0 0 0 0 0 0 0 1

0 1 0 0 0 0 0 0

0 0 0 1 0 0 0 0

0 0 0 0 0 1 0 0

0 0 1 0 0 0 0 0