|  |  |  |  |
| --- | --- | --- | --- |
|  | Bansilal Ramnath Agarwal Charitable Trust's  Vishwakarma Institute of Information Technology  **Department of**  **Artificial Intelligence and Data Science** | | |
| Name: Siddhesh Dilip Khairnar | | | |
| Class: TY | Division: B | | Roll No: 372028 |
| Semester: V | | Academic Year: 2023-24 | |
| Subject Name & Code: Design and Analysis of Algorithm: ADUA31202 | | | |
| Title of Assignment: Implement N- queen Problem using Back tracking method. A suitable message is to be displayed if the given Problem instance doesn't have a solution. | | | |

**ASSIGNMENT NO. 6**

A paper with writing on it

Description automatically generated

A piece of paper with writing on it

Description automatically generated

A piece of paper with writing on it

Description automatically generated

**Program Code:**

#include <bits/stdc++.h>

#define N 4

using namespace std;

void printSolution(int board[N][N])

{

    for (int i = 0; i < N; i++)

    {

        for (int j = 0; j < N; j++)

            cout << " " << board[i][j] << " ";

        printf("\n");

    }

}

bool isSafe(int board[N][N], int row, int col)

{

    int i, j;

    for (i = 0; i < col; i++)

        if (board[row][i])

            return false;

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

        if (board[i][j])

            return false;

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

        if (board[i][j])

            return false;

    return true;

}

bool solveNQUtil(int board[N][N], int col)

{

    if (col >= N)

        return true;

    for (int i = 0; i < N; i++)

    {

        if (isSafe(board, i, col))

        {

            board[i][col] = 1;

            if (solveNQUtil(board, col + 1))

                return true;

            board[i][col] = 0;

        }

    }

    return false;

}

bool solveNQ()

{

    int board[N][N] = {{0, 0, 0, 0},

                       {0, 0, 0, 0},

                       {0, 0, 0, 0},

                       {0, 0, 0, 0}};

    if (solveNQUtil(board, 0) == false)

    {

        cout << "Solution does not exist";

        return false;

    }

    printSolution(board);

    return true;

}

int main()

{

    solveNQ();

    return 0;

}

**Output:**

A computer screen with white text

Description automatically generated