

""

Name: Anuj Mahendra Mutha

Class:BE-4 Computer Engineering

Batch : R4

Lab Assignment No: 05

Title: Design n-Queens matrix having first Queen placed. Use backtracking to place remaining

Queens to generate the final n-queen's matrix.

""

```
#include <iostream>
using namespace std;
int grid[10][10];
void print(int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cout << grid[i][j] << " ";
        }
        cout << endl;
    }
    cout << endl;
    cout << endl;
}
bool isSafe(int col, int row, int n) {
    //check for same column
    for (int i = 0; i < row; i++) {
        if (grid[i][col]) {
            return false;
        }
    }
    //check for upper left diagonal
    for (int i = row, j = col; i >= 0 && j >= 0; i--, j--) {
        if (grid[i][j]) {
            return false;
        }
    }
    //check for upper right diagonal
    for (int i = row, j = col; i >= 0 && j < n; j++, i--) {
        if (grid[i][j]) {
            return false;
        }
    }
    return true;
}

bool solve (int n, int row) {
    if (n == row) {
        print(n);
        return true;
    }
    bool res = false;
    for (int i = 0; i < n; i++) {
        if (isSafe(i, row, n)) {
            grid[row][i] = 1;
            res = solve(n, row+1) || res;
            grid[row][i] = 0;
        }
    }
}
```

```

    }
    return res;
}
int main()
{
    int n;
    cout<<"Enter the number of queen"<<endl;
    cin >> n;
    for (int i = 0;i < n;i++) {
        for (int j = 0;j < n;j++) {
            grid[i][j] = 0;
        }
    }
    bool res = solve(n, 0);
    if(!res) {
        cout << -1 << endl;
    } else {
        cout << endl;
    }
    return 0;
}

```

/*

Output :

Enter the number of queen

4

0 1 0 0

0 0 0 1

1 0 0 0

0 0 1 0

0 0 1 0

1 0 0 0

0 0 0 1

0 1 0 0

*/