

**ARTIFICIAL INTELLIGENCE LAB**  
**ASSIGNMENT – 2**

**NAME : PRATHAPANI SATWIK**

**REG.NO. : 20BCD7160**

**Q) 1. TO FIND MINIMAL MOVES IN 8-QUEENS PROBLEM**

**CODE :**

```
import java. util.Arrays;

class Main
{
    public static final int N = 8; private static Boolean
    isSafe(char mat[][], int r, int c)
    {
        for (int i = 0; i < r;
        i++)
        { if (mat[i][c] == 'Q')
        { return false;
        } } for (int i = r, j = c; i >= 0 && j >= 0; i--
        , j--)
        { if (mat[i][j] ==
        'Q') { return false;
        } } for (int i = r, j = c; i >= 0 && j < N; i--,
        j++)
```

```

    { if (mat[i][j] ==
    'Q')
    { return
    false;
    } } return true; } private static void
    printSolution(char mat[][])
    {
    for (int i = 0; i < N; i++) {
    System.out.println(Arrays.toString(mat[i]).replaceAll(",", ""));
    }
    System.out.println();
    } private static void nQueen(char mat[][], int
    r)
    { if (r == N) {
    printSolution(mat);
    return; } for (int i
    = 0; i < N; i++)
    { if (isSafe(mat, r, i))
    { mat[r][i] = 'Q';
    nQueen(mat, r + 1);
    mat[r][i] = '4';
    }
    } } } public static void main(String[]
    args)

```

```

{
    char[][] mat = new
char[N][N];    for (int i = 0; i <
N; i++) {
    Arrays.fill(mat[i], '4');
}
nQueen(mat,0);
}
}

```

### **OUTPUT :**

```
java -cp /tmp/VRha0U9NAm Main
```

```

[Q 4 4 4 4 4 4 4]
[4 4 4 4 Q 4 4 4]
[4 4 4 4 4 4 4 Q]
[4 4 4 4 4 Q 4 4]
[4 4 Q 4 4 4 4 4]
[4 4 4 4 4 4 Q 4]
[4 Q 4 4 4 4 4 4]
[4 4 4 Q 4 4 4 4]

```

```

[Q 4 4 4 4 4 4 4]
[4 4 4 4 4 Q 4 4]
[4 4 4 4 4 4 4 Q]
[4 4 Q 4 4 4 4 4]
[4 4 4 4 4 4 Q 4]
[4 4 4 Q 4 4 4 4]

```

[4 Q 4 4 4 4 4 4]

[4 4 4 4 Q 4 4 4]

[Q 4 4 4 4 4 4 4]

[4 4 4 4 4 4 Q 4]

[4 4 4 Q 4 4 4 4]

[4 4 4 4 4 Q 4 4]

[4 4 4 4 4 4 4 Q]

[4 Q 4 4 4 4 4 4]

[4 4 4 4 Q 4 4 4]

[4 4 Q 4 4 4 4 4]

[Q 4 4 4 4 4 4 4]

[4 4 4 4 4 4 Q 4]

[4 4 4 4 Q 4 4 4]

[4 4 4 4 4 4 4 Q]

[4 Q 4 4 4 4 4 4]

[4 4 4 Q 4 4 4 4]

[4 4 4 4 4 Q 4 4]

[4 4 Q 4 4 4 4 4]

[4 Q 4 4 4 4 4 4]

[4 4 4 Q 4 4 4 4]

[4 4 4 4 4 Q 4 4]

[4 4 4 4 4 4 4 Q]

[4 4 Q 4 4 4 4 4]

[Q 4 4 4 4 4 4 4]

[4 4 4 4 4 4 Q 4]

[4 4 4 4 Q 4 4 4]

[4 Q 4 4 4 4 4 4]

[4 4 4 4 Q 4 4 4]

[4 4 4 4 4 4 Q 4]

[Q 4 4 4 4 4 4 4]

[4 4 Q 4 4 4 4 4]

[4 4 4 4 4 4 4 Q]

[4 4 4 4 4 Q 4 4]

```

26 }
27 return true;
28 }
29 private static void printSolution(char mat[][]){
30 {
31 for (int i = 0; i < N; i++) {
32 System.out.println(Arrays.toString(mat[i]).replaceAll(", ", ""));
33 }
34 System.out.println();
35 }
36 private static void nQueen(char mat[][], int r)
37 {
38 if (r == N)
39 {
40 printSolution(mat);
41 return;
42 }
43 for (int i = 0; i < N; i++)
44 {
45 if (isSafe(mat, r, i)) {
46 mat[r][i] = 'Q';
47 nQueen(mat, r + 1);
48 mat[r][i] = '4';
49 }
50 }
51 }
52 public static void main(String[] args)
53 {
54 char[][] mat = new char[N][N];
55 for (int i = 0; i < N; i++) {
56 Arrays.fill(mat[i], '4');
57 }
58 nQueen(mat,0);

```

[Q 4 4 4 4 4 4 4]  
[4 4 4 4 4 Q 4 4 4]  
[4 4 4 4 4 4 4 4 Q]  
[4 4 4 4 4 4 Q 4 4]  
[4 4 Q 4 4 4 4 4 4]  
[4 4 4 4 4 4 4 Q 4]  
[4 Q 4 4 4 4 4 4 4]  
[4 4 4 Q 4 4 4 4 4]

[Q 4 4 4 4 4 4 4 4]  
[4 4 4 4 4 4 Q 4 4]  
[4 4 4 4 4 4 4 4 Q]  
[4 4 Q 4 4 4 4 4 4]  
[4 4 4 4 4 4 4 Q 4]  
[4 4 4 Q 4 4 4 4 4]  
[4 Q 4 4 4 4 4 4 4]  
[4 4 4 4 Q 4 4 4 4]

[Q 4 4 4 4 4 4 4]  
[4 4 4 4 4 4 Q 4]  
[4 4 4 Q 4 4 4 4]  
[4 4 4 4 4 Q 4 4]  
[4 4 4 4 4 4 4 Q]

[illegible]