## ARTIFICIAL INTELLIGENCE LAB ASSIGNMENT – 1

**NAME: PRATHAPANI SATWIKA** 

**REG.NO.: 20BCD7160** 

## Q) 1.WRITE A PROGRAM TO SOLVE ANY 2 PLAYER GAME SCENARIOS

**EX: 8 PUZZLE PROBLEM** 

## **CODE:**

```
import java. util.Arrays; class Main  \{ \\ \text{public static final int } N = 8; \\ \text{private static boolean isSafe(char mat[][], int } r, int c) \\ \{ \\ \text{for (int } i = 0; i < r; i++) \\ \{ \\ \text{if (mat[i][c] == 'Q') } \{ \\ \text{return false;} \\ \} \\ \} \\ \text{for (int } i = r, j = c; i >= 0 \&\& j >= 0; i--, j--) \\ \{ \\ \text{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 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 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 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 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 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 Q]
- [4 4 4 4 4 Q 4 4]

```
1 · import java. util.Arrays;
                                                                                           __ java -cp /tmp/VRha0U9NAm Main
 2 class Main
 3 * {
                                                                                             [Q 4 4 4 4 4 4 4]
 4 public static final int N = 8;
                                                                                             [4 4 4 4 Q 4 4 4]
 5 private static boolean isSafe(char mat[][], int r, int c)
                                                                                             [4 4 4 4 4 4 4 0]
 6 * {
                                                                                             [4 4 4 4 4 Q 4 4]
 7 for (int i = 0; i < r; i++)
                                                                                             [4 4 Q 4 4 4 4 4]
 8 - {
                                                                                             [4 4 4 4 4 4 0 4]
9 • if (mat[i][c] == 'Q') {
                                                                                             [4 Q 4 4 4 4 4 4]
                                                                                             [4 4 4 0 4 4 4 4]
10 return false;
11 }
12 }
                                                                                             [Q 4 4 4 4 4 4 4]
13 for (int i = r, j = c; i \ge 0 \&\& j \ge 0; i - -, j - -)
                                                                                             [4 4 4 4 4 Q 4 4]
14 - {
                                                                                             [4 4 4 4 4 4 4 0]
15 if (mat[i][j] == 'Q')
                                                                                             [4 4 Q 4 4 4 4 4]
16 * {
                                                                                             [4 4 4 4 4 4 Q 4]
17 return false;
                                                                                             [4 4 4 Q 4 4 4 4]
18 }
                                                                                             [4 Q 4 4 4 4 4 4]
19 }
                                                                                             [4 4 4 4 Q 4 4 4]
20 for (int i = r, j = c; i \ge 0 \&\& j < N; i--, j++)
                                                                                             [Q 4 4 4 4 4 4 4]
21 - {
22  if (mat[i][j] == 'Q')
                                                                                             [4 4 4 4 4 4 0 4]
23 * {
                                                                                             [4 4 4 Q 4 4 4 4]
24 return false;
                                                                                             [4 4 4 4 4 Q 4 4]
25 }
                                                                                             [4 4 4 4 4 4 4 0]
26 }
                                                                                                      [0 4 4 4 4 4 4 4]
 27 return true;
                                                                                                       [4 4 4 4 4 4 Q 4]
 28 }
                                                                                                       [4 4 4 Q 4 4 4 4]
 29 private static void printSolution(char mat[][])
                                                                                                       [4 4 4 4 4 Q 4 4]
 30 + {
                                                                                                       [4 4 4 4 4 4 4 0]
 31 + for (int i = 0; i < N; i++) {
                                                                                                       [4 Q 4 4 4 4 4 4]
 32 System.out.println(Arrays.toString(mat[i]).replaceAll(",", ""));
                                                                                                       [4 4 4 4 Q 4 4 4]
 33 }
                                                                                                       [4 4 Q 4 4 4 4 4]
 34 System.out.println();
                                                                                                       [Q 4 4 4 4 4 4 4]
 35 }
                                                                                                       [4 4 4 4 4 4 Q 4]
 36 private static void nQueen(char mat[][], int r)
                                                                                                       [4 4 4 4 Q 4 4 4]
 37 + {
                                                                                                       [4 4 4 4 4 4 4 Q]
 38 if (r == N)
                                                                                                       [4 Q 4 4 4 4 4 4]
 39 + {
                                                                                                      [4 4 4 Q 4 4 4 4]
 40 printSolution(mat);
                                                                                                       [4 4 4 4 4 0 4 4]
 41 return;
                                                                                                       [4 4 Q 4 4 4 4 4]
 42 }
                                                                                                       [4 Q 4 4 4 4 4 4]
 43 for (int i = 0; i < N; i++)
                                                                                                       [4 4 4 Q 4 4 4 4]
 44 - {
                                                                                                       [4 4 4 4 4 Q 4 4]
 45 \cdot if (isSafe(mat, r, i)) {
                                                                                                       [4 4 4 4 4 4 4 0]
 46 mat[r][i] = 'Q';
                                                                                                       [4 4 Q 4 4 4 4 4]
 47 nQueen(mat, r + 1);
                                                                                                       [0 4 4 4 4 4 4 4]
 48 mat[r][i] = '4';
                                                                                                       [4 4 4 4 4 4 Q 4]
 49 }
                                                                                                       [4 4 4 4 Q 4 4 4]
 50 }
 51 }
                                                                                                       [4 Q 4 4 4 4 4 4]
 52 public static void main(String[] args)
                                                                                                       [4 4 4 4 Q 4 4 4]
 53 + {
                                                                                                       [4 4 4 4 4 4 Q 4]
 54
        char[][] mat = new char[N][N];
                                                                                                       [Q 4 4 4 4 4 4 4]
 55 +
       for (int i = 0; i < N; i++) {
                                                                                                       [4 4 Q 4 4 4 4 4]
 56 Arrays.fill(mat[i], '4');
                                                                                                       [4 4 4 4 4 4 4 0]
 57 }
```

58 nQueen(mat,0);

[4 4 4 4 4 Q 4 4]