ARTIFICIAL INTELLIGENCE LAB

ASSIGNMENT – 2

NAME : PRATHAPANI SATWIKA

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]

```java
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;
}
```

```
java -cp /tmp/VRhaOU9NAm 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 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 4 4 4 Q 4]
[4 4 4 4 Q 4 4 4]
```

```java
}
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);
```

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