PRATICAL-12

AIM: Write a program to implement N- Queen problem using Branch and Bound Techniques.

SOURCE CODE:-

```
class FourQueensBB {
  private static final int N = 4;
  private static boolean[] rowUsed = new boolean[N];
  private static boolean[] d1 = new boolean[2 * N];
  private static boolean[] d2 = new boolean[2 * N];

  private static void printSolution(int board[][]) {
    for (int i = 0; i < N; i++) {
      for (int j = 0; j < N; j++) {
        System.out.print(board[i][j] + " ");
      }
      System.out.println();
    }
    System.out.println();
  }

  private static boolean solveNQUtil(int board[][], int col) {
    if (col >= N) {
      printSolution(board);
      return true;
    }

    boolean res = false;
    for (int i = 0; i < N; i++) {
      if (!rowUsed[i] && !d1[i - col + N - 1] && !d2[i + col]) {
        board[i][col] = 1;
        rowUsed[i] = d1[i - col + N - 1] = d2[i + col] = true;
        res = solveNQUtil(board, col + 1) || res;
        board[i][col] = 0;
        rowUsed[i] = d1[i - col + N - 1] = d2[i + col] = false;
      }
    }
    return res;
  }

  public static void solveNQ() {
    int board[][] = new int[N][N];
    if (!solveNQUtil(board, 0)) {
      System.out.println("No solution exists");
    }
  }

  public static void main(String[] args) {
```

```
    solveNQ();
  }
}
```

OUTPUT:-

```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Users\Ankit raj\
0 0 1 0
1 0 0 0
0 0 0 1
0 1 0 0

0 1 0 0
0 0 0 1
1 0 0 0
0 0 1 0
```