

Lab 1

Problem Statement: Tic-tac-toe- AI and non-AI approach

NAME: Harshita Bhagat

ROLLNO: 31

CLASS: TY - IT A

BATCH: 2

Tic-tac-toe non-AI

Program:

```
package AI;

import java.util.*;
// Non AI!!
public class ttt {
    public static void main(String[] args) {
        String a[] = new String[9]; // Array for X's and O's
        int in1[] = new int[9]; // Array for 1's and 2's
        int i;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter input data: ");
        for(i = 0; i<9; i++) {
            a[i] = sc.next();
        }

        // Printing the Tic Tac Toe Vector
        System.out.print("Tictactoe vector [");
        for(i = 0; i<9; i++) {
            if (a[i].equals("X")) {
                a[i] = "1";
            }
            else if (a[i].equals("O")) {
                a[i] = "2" ;
            }
            System.out.print(" " + a[i] );
        }
    }
}
```

```

        System.out.println("]");

        // Converting "1" and "2" into 1 and 2 {parse method}
        for(i = 0; i<9; i++) {
            inta[i] = Integer.parseInt(a[i]);
        }

        // Checking for validity

        int count1 = 0, count2 = 0;

        for (int j = 0; j < 9; j++) {
            if (inta[j] == 1) {
                count1++;
            } else if (inta[j] == 2) {
                count2++;
            }
        }

        if (Math.abs(count1 - count2) == 1) {
            System.out.println("Valid!");
        }

        else {
            System.out.println("Invalid!!");
            System.exit(0);
        }

        // Calculating Index in Rule Base
        // {Unique Scoring System For each board position}

        double index = 0;
        for (int j = 0; j < 9; j++) {
            index += (inta[j] * Math.pow(3, 8 - j));
        }
        System.out.println("Index is: " + index);
    }
}

```

```
Enter input data:
X O X
O O O
O O O
Tictactoe vector [ 1 0 1 2 0 0 0 0 0]
Valid!
Index is: 7776.0
```

Tic Tac Toe: AI

Program

```
package AI;
import java.util.*;

public class tttAI {
    public static void main(String[] args) {
        String a[] = new String[9];
        int inta[] = new int[9];
        int i;

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter input data: ");
        for (i = 0; i < 9; i++) {
            a[i] = sc.next();
        }

        System.out.print("Tictactoe vector [");
        for (i = 0; i < 9; i++) {
            if (a[i].equals("X")) {
                a[i] = "1";
            } else if (a[i].equals("O")) {
                a[i] = "2";
            }
            System.out.print(" " + a[i]);
        }
    }
}
```

```

System.out.println("]");

for (i = 0; i < 9; i++) {
    inta[i] = Integer.parseInt(a[i]);
}

// Checking for validity
int count1 = 0, count2 = 0, count0 = 0;

for (int j = 0; j < 9; j++) {
    if (inta[j] == 1) {
        count1++;
    } else if (inta[j] == 2) {
        count2++;
    } else {
        count0++;
    }
}

if (Math.abs(count1 - count2) <= 1) {
    System.out.println("Valid!");
} else {
    System.out.println("Invalid!!");
    System.exit(0);
}

// Calculating Index
double index = 0;
for (int j = 0; j < 9; j++) {
    index += (inta[j] * Math.pow(3, 8 - j));
}
System.out.println("Index is: " + index);

// Computing Score
int[] computingScores = new int[9];

//      System.out.println("Calculating Computing Score: ");
//      System.out.println(" ");

// Making a probability matrix
System.out.println("Possible Moves = ");

```

```

        if (count1 <= count2) {
            // X's Turn {The "X" player goes first always!}
            System.out.println("Predicting for X turn");
            int rowCounter = 0;
            int colCounter = 0;
            int diagCounter = 0; // have to reset zero for each
move
            // For X's Turn
            for (i = 0; i < 9; i++) {
                if (inta[i] == 0) {
                    int[] copyInta = inta.clone();
                //      System.out.println("inta:" + inta[i]);
                    copyInta[i] = 1;
                    printMatrix(copyInta);

                    for (int j = 0; j < 3; j++) {
                        if (checkRow(copyInta, j)) {
                            rowCounter++;
                //      System.out.println("rowCounter" +
rowCounter);
                        }
                    }

                //      System.out.println("Intermediate
rowCounter: " + rowCounter); // Debug

                    for (int j = 0; j < 3; j++) {
                        if (checkColumn(copyInta, j)) {
                            colCounter++;
                //      System.out.println("colCounter" +
colCounter);
                        }
                    }

                //      System.out.println("Intermediate
colCounter: " + colCounter); // Debug

                    if (checkDiagonal(copyInta)) {
                //      System.out.println("diagCounter" +
diagCounter);

```

```

    }

//          System.out.println("Intermediate
diagCounter: " + diagCounter); // Debug

        computingScores[i] = rowCounter + colCounter
+ diagCounter;
    }
}

} else {
    // O's Turn
    System.out.println("Predicting O turn");
    for (i = 0; i < 9; i++) {
        if (inta[i] == 0) {
            int[] copyInta = inta.clone();
            copyInta[i] = 2;
            printMatrix(copyInta);

            int rowCounter = 0;
            int colCounter = 0;
            int diagCounter = 0;

            for (int j = 0; j < 3; j++) {
                if (checkRow(copyInta, j))
                    rowCounter++;
            }

            for (int j = 0; j < 3; j++) {
                if (checkColumn(copyInta, j))
                    colCounter++;
            }

            if (checkDiagonal(copyInta)) {
                diagCounter++;
                System.out.println(diagCounter);
            }

```

```

        computingScores[i] = rowCounter + colCounter
+ diagCounter;
    }
}
}

```

```

//      System.out.println("Computing Scores for each
position: ");
//      for (i = 0; i < 9; i++) {
//          if (inta[i] == 0) {
//              System.out.println("Position " + (i + 1) + ":
" + computingScores[i]);
//          }
//      }
}

```

```

private static void printMatrix(int[] array) {
    for (int i = 0; i < 9; i++) {
        if (array[i] == 1)
            System.out.print("X ");
        else if (array[i] == 2)
            System.out.print("O ");
        else
            System.out.print("0 ");

        if ((i + 1) % 3 == 0) {
            System.out.println();
        }
    }
    System.out.println();
}

```

```

private static boolean checkRow(int[] array, int row) {
    int start = row * 3;

    int value = array[start];
    if (value == 0)
        return false;

    for (int i = start + 1; i < start + 3; i++) {
        if (array[i] != value)

```

```

        return false;
    }

    return true;
}

private static boolean checkColumn(int[] array, int col) {
    int start = col;

    int value = array[start];
    if (value == 0)
        return false;

    for (int i = start + 3; i < start + 7; i += 3) {
        if (array[i] != value)
            return false;
    }
    return true;
}

private static boolean checkDiagonal(int[] array) {
    if ((array[0] != 0 && array[0] == array[4] && array[0] ==
array[8]) ||
        (array[2] != 0 && array[2] == array[4] &&
array[2] == array[6]))
        return true;

    return false;
}
}

```

Output:


```
"C:\Program Files\Java\jdk-17.0.5\bin\java.exe
```

```
Enter input data:
```

```
X 0 X
```

```
0 0 0
```

```
0 0 0
```

```
Tictactoe vector [ 1 0 1 2 0 0 0 0 0]
```

```
Valid!
```

```
Index is: 7776.0
```

```
Possible Moves =
```

```
Predicting 0 turn
```

```
X 0 X
```

```
0 0 0
```

```
0 0 0
```

```
X 0 X
```

```
0 0 0
```

```
0 0 0
```

```
X 0 X
```

```
0 0 0
```

```
0 0 0
```

```
X 0 X
```

```
0 0 0
```

```
0 0 0
```

```
X 0 X
```

```
0 0 0
```

```
0 0 0
```

```
X 0 X
```

```
0 0 0
```

0 0 0