

Code:

```
import java.util.*;

public class DFA {
    public static void printMatrix(ArrayList<ArrayList<Integer>> v, int row, int col) {
        System.out.println("Number of rows: " + v.size());
        for (int i = 0; i < row; ++i) {
            System.out.println("Size of row: " + v.get(i).size());
            for (int j = 0; j < col; ++j) {
                System.out.print(v.get(i).get(j) + " ");
            }
            System.out.println();
        }
    }

    public static boolean accepts(ArrayList<ArrayList<Integer>> v, ArrayList<Character> sym,
int row, int col, String str) {
        int l, i, j, found, currState = 0;
        l = str.length();
        for (i = 0; i < l; ++i) {
            j = 0;
            found = 0;
            while (j < col && found == 0) {
                if (sym.get(j) == str.charAt(i)) {
                    found = 1;
                } else {
                    j++;
                }
            }
            if (j == col) {
                return false;
            } else {
                currState = v.get(currState).get(j);
                if (currState == -1) {
                    return false;
                }
            }
        }
        return currState == row - 1;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int numStates, numSym, i, j, x;
        ArrayList<Character> sym = new ArrayList<>();
        char ch;
```

```

System.out.print("Enter number of symbols: ");
numSym = scanner.nextInt();
System.out.print("Enter number of states: ");
numStates = scanner.nextInt();
ArrayList<ArrayList<Integer>> dfa = new ArrayList<>(numStates);

System.out.println("Enter " + numSym + " symbols for the DFA: ");
for (i = 0; i < numSym; ++i) {
    ch = scanner.next().charAt(0);
    sym.add(ch);
}

System.out.println("Enter DFA matrix: ");
for (i = 0; i < numStates; i++) {
    System.out.println("From q" + i + ": ");
    ArrayList<Integer> row = new ArrayList<>(numSym);
    for (j = 0; j < numSym; ++j) {
        System.out.print("\tOn " + sym.get(j) + ": ");
        x = scanner.nextInt();
        row.add(x);
    }
    dfa.add(row);
}

int choice, stop = 0;
String str;
while (stop == 0) {
    System.out.print("1. Check string \n2. Stop\nEnter choice: ");
    choice = scanner.nextInt();
    if (choice == 1) {
        System.out.print("Enter a string: ");
        str = scanner.next();
        if (accepts(dfa, sym, numStates, numSym, str)) {
            System.out.println("Accepted");
        } else {
            System.out.println("Not accepted");
        }
    } else {
        stop = 1;
    }
}
scanner.close();
}
}

```

Output:

```
Enter number of symbols: 2
Enter number of states: 6
Enter 2 symbols for the DFA:
0 1
Enter DFA matrix:
From q0:
    On 0: -1
    On 1: 1
From q1:
    On 0: 2
    On 1: -1
From q2:
    On 0: 4
    On 1: 3
From q3:
    On 0: -1
    On 1: 2
From q4:
    On 0: -1
    On 1: 5
From q5:
    On 0: -1
    On 1: -1
1.check string
2.stop
Enter choice: 1
Enter a string: 101101
Accepted
1.check string
2.stop
Enter choice: 1
Enter a string: 1011101
Not accepted
1.check string
2.stop
Enter choice: 2
```

Observation:

In this experiment I learnt what is a deterministic automata , about how to construct a deterministic finite automata using the regular expression, representing a deterministic finite automata using 5 tuples and the process of validating a string using the deterministic finite automata.

Conclusion:

Implementation and validation of string using Deterministic Finite Automata was studied successfully.