**The Main Class:**

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

// Major : Computer Science

// Course : Computer Theory - 601322

// Author : Ibrahim Alhamad

// Author ID : 201220475

// Description : It is a Finite Automata that accept the set of all strings that, viewed as natural numbers in unsigned // binary notation, represent numbers divisible by 5.

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

**import** java.util.Scanner;

**public** **class** Main {

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

Character Alphabet[] = {'1', '0'}; // set the Σ

State state0 = **new** State("0");

State state1 = **new** State("1");

State state2 = **new** State("2");

State state3 = **new** State("3");

State state4 = **new** State("4");

State FinalStates[] = {state0}; // set the final states // Transition Table

// | 1 | 0

String transitionTable[][] = {{state0.label, "1", "0"}, // -+ 0 | 1 | 0

{state1.label, "3", "2"}, // 1 | 3 | 2

{state2.label, "0", "4"}, // 2 | 0 | 4

{state3.label, "2", "1"}, // 3 | 2 | 1

{state4.label, "4", "3"}};// 4 | 4 | 3

String token = **new** Scanner(System.***in***).next();// get the token from the user.

DFA dfa = **new** DFA(Alphabet, transitionTable, state0, token, FinalStates);

System.***out***.print(dfa.isItAccepted()); // print either the token is Accepted or Rejected.

}

}

**The DFA Class:**

**import** java.util.Arrays;

**import** java.util.List;

**public** **class** DFA {

**private** String transitionTable[][]; // transition table as a 2D array

**private** List<Character> Alphabet;

**private** State FinalStates[];

**private** String currentState;

**Public** String token;

**Public** DFA(Character Alphabet[], String transitionTable[][], State startState, String token, State FinalStates[]){

**this**.Alphabet = Arrays.*asList*(Alphabet); // convert the Σ into a List

**this**.transitionTable = transitionTable; // set the transition table

**this**.currentState = startState.label; // set the state label

**this**.FinalStates = FinalStates; // set the final states array

**this**.token = token;

}

**Public** String isItAccepted() {

**for**(**int** index = 0;index < token.length();index++) {

**if** (Alphabet.contains(token.charAt(index))) { // check if the 1st Character is in the Alphabet

**int** csAsIndex = Integer.*parseInt*(currentState);// copy the currentState as an Index

**if** (token.charAt(index) == Alphabet.get(0)) {// if the current character is '1'

**if** (!(transitionTable[csAsIndex][1] == currentState)) {// check if the next state is != current state

currentState = transitionTable[csAsIndex][1];// if true set the current state = next state

}

} **else** **if** (token.charAt(index) == Alphabet.get(1)) {// if the current character is '0'

**if** (!(transitionTable[csAsIndex][2] == currentState)) {// check if the next state is != current state

currentState = transitionTable[csAsIndex][2];// if true set the current state = next state

}

}

}

**else** {

**return** "Rejected";

}

}

**for** (**int** i = 0; i < FinalStates.length; i++) {

**if** (currentState.equals(FinalStates[i].label)){//check if the current state is one of the final states

**return** "Accepted";

}

}

**return** "Rejected";

}

}

**The State Class:**

**public** **class** State {

**Public** String label;

**Public** State(String label) {

**this**.label = label;

}

}