

**Problem Statement 5:**

Design and implement an automaton (DFA) in 'C' programming language for the following language L.

$L = \{10^n 1^m 0 \mid n, m \geq 0\}$ , Where, input alphabet  $\Sigma = \{0, 1\}$

**Input from keyboard:**

DFA Parameters

Input String

**Output on the screen:**

Whether the string is accepted or not.

**Description of the problem statement:**

This Program is to implement Deterministic Finite Automata (DFA) which accepts a language

$L = \{10^n 1^m 0 \mid n, m \geq 0\}$ , Where, input alphabet  $\Sigma = \{0, 1\}$

**Algorithm:**

START

Step 1: Read all the DFA parameters from the user input.

Step 2: Read the input string which is to be checked whether it is accepted or not.

Step 3: DFA Simulation:

    Step 3.1: Initialize a variable currentState to the initial state.

    Step 3.2: For each character c in the input string:

        Step 3.2.1: Find the index of c in the array of input symbols.

        Step 3.2.2: Update currentState using the transition function.

Step 4: After simulating the DFA on the entire input string, check if the final currentState is in the set of accepting states. If it is, then the input string is accepted by the DFA. Otherwise, it's rejected.

Step 5: Print whether the input string was accepted or rejected by the DFA.

STOP

OUTPUT:

```
Enter the DFA Parameters
Enter the no. of States
6
Enter the States
1 2 3 4 5 6
Enter the initial state
1
Enter the no. of final States
2
Enter the final States
3 5
Enter the no. of input symbols
2
Enter the symbols
01
Enter the transition table as (current state, current symbol)
for (1 0)= 6
for (1 1)= 2
for (2 0)= 3
for (2 1)= 4
for (3 0)= 3
for (3 1)= 4
for (4 0)= 5
for (4 1)= 4
for (5 0)= 6
for (5 1)= 6
for (6 0)= 6
for (6 1)= 6

Transition Table:
Current State  0      1
1              6      2
2              3      4
3              3      4
4              5      4
5              6      6
6              6      6
Enter the input string
1010
Input string accepted by the DFA
Enter 1 to continue or -1 to exit
```