2.Write a C program to simulate a Deterministic Finite Automata (DFA) for the given language representing strings that start with 0 and end with 1

AIM: To write a C program to simulate a Deterministic Finite Automata (DFA) for the given language representing strings that start with 0 and end with 1

ALGORITHEM:

1. Define the DFA:

• Set of states: Q = {q0, q1, q2}

• Alphabet: Σ = {0, 1}

• Transition function: δ(q, a) where q is the current state and a is the input symbol

• Initial state: q0

• Accepting states: F = {q2}

2. Initialize the current state to the initial state (q0).

3. For each input symbol in the string: a. Use the transition function δ to determine the next state based on the current state and the input symbol. b. If the transition is defined (i.e., δ(q, a) is not undefined), update the current state to the next state. c. If the transition is undefined at any point, reject the string.

4. After processing all input symbols, check if the final state is an accepting state (i.e., q2). If yes, accept the string; otherwise, reject the string

PROGRAM:

#include <stdio.h>

#include <stdbool.h>

int transition(int state, char input) {

if (state == 0 && input == '0') {

return 1;

} else if (state == 1 && input == '1') {

return 2;

} else {

return -1;

}

}

bool simulateDFA(const char \*input) {

int currentState = 0;

for (int i = 0; input[i] != '\0'; ++i) {

currentState = transition(currentState, input[i]);

if (currentState == -1) {

return false;

}

}

return currentState == 2;

}

int main() {

char input[100];

printf("Enter a string: ");

scanf("%s", input);

if (simulateDFA(input)) {

printf("Accepted: The string follows the pattern (starts with 0 and ends with 1).\n");

} else {

printf("Rejected: The string does not follow the pattern.\n");

}

return 0;

}

OUTPUT:



