**Aim:NFA Which Accepts Strings Start With ‘0’ And End With ‘1’.**

**Algorithm**

* Input\_String: The string to be checked by the NFA
* Initialize the current state to the initial state of the NFA.
* For each character (symbol) c in Input\_String
* Get the set of next states from the current state using the transition function for symbol c. b. Update the current state to the union of the next states.
* After processing all characters in Input\_String, check if the current state contains the accepting state (i.e., if it is possible to reach the accepting state from the current state).
* If the current state contains the accepting state, output "Accepted," indicating that the input string satisfies the condition (starts with '0' and ends with '1'). Otherwise, output "Rejected."

**Program:**

#include <stdio.h>

#include <stdbool.h>

bool simulateNFA(const char \*input)

{

int currentState = 0;

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

{

if (currentState == 0 && input[i] == '0')

currentState = 1;

else if (currentState == 1 && input[i] == '1')

currentState = 2;

else

{

currentState = -1;

break;

}

}

return (currentState == 2);

}

int main()

{

char input[100];

printf("Enter a String : ");

scanf("%s", input);

if (simulateNFA(input))

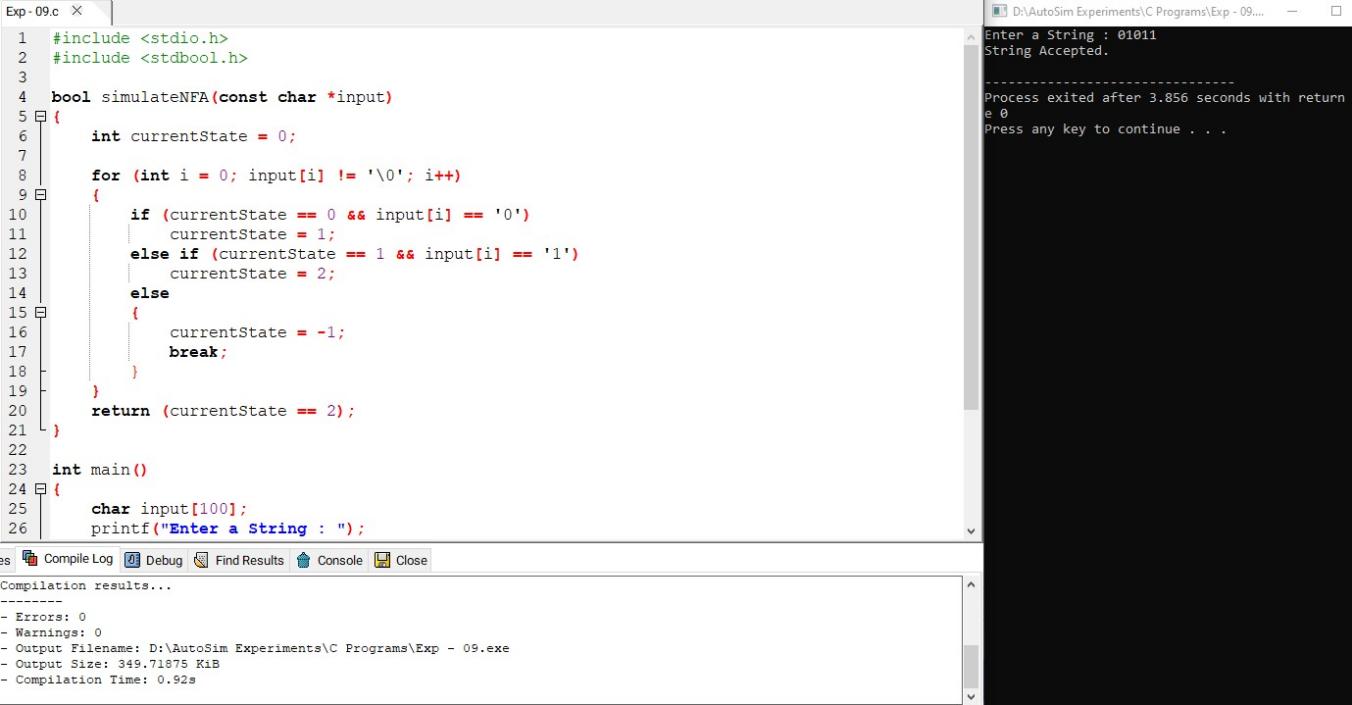
printf("String Not Accepted.\n");

else

printf("String Accepted.\n");

return 0;

}



**Result**

Thus we have successfully implemented and executed the program and the strings given as inputs are verified