Practical 2

AIM :

Practical Definition
String Validation Using Finite Automata

Objective
To implement a program that validates a given string against rules defined in terms of finite automata.

Language Constraint
The program can be implemented in any programming language

Input requirement
- Accept rules in the form of finite automata (e.g., states, transitions, start state, accept states) as input.
- Accept a string to be validated against the provided finite automata rules.

Expected output
- If the string adheres to the rules of the finite automata, the program should output: "Valid String".
- If the string does not adhere to the rules, the program should output: "Invalid String".

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

#define MAX_STATES 100
#define MAX_SYMBOLS 100

void finite_automata() {
   int num_inputs, num_states, initial_state, num_accepting_states;
   int accepting_states[MAX_STATES];
   char input_symbols[MAX_SYMBOLS];
   int transition_table[MAX_STATES][MAX_SYMBOLS];
```

```
  char input_string[100];

  printf("Finite Automata Setup\n");

  // Step 1: Take inputs
  printf("Number of input symbols: ");
  scanf("%d", &num_inputs);
  printf("Enter the input symbols (space-separated): ");
  for (int i = 0; i < num_inputs; i++) {
    scanf(" %c", &input_symbols[i]);
  }

  printf("Number of states: ");
  scanf("%d", &num_states);

  printf("Initial state: ");
  scanf("%d", &initial_state);

  printf("Number of accepting states: ");
  scanf("%d", &num_accepting_states);
  printf("Enter the accepting states (space-separated): ");
  for (int i = 0; i < num_accepting_states; i++) {
    scanf("%d", &accepting_states[i]);
  }

    printf("\nEnter the transition table:\n");
  for (int i = 1; i <= num_states; i++) {
    for (int j = 0; j < num_inputs; j++) {
      printf("State %d on input '%c' transitions to: ", i, input_symbols[j]);
      scanf("%d", &transition_table[i][j]);
    }
```

```
  }

  while (1) {
    printf("\nEnter input string: ");
    scanf("%s", input_string);

    int current_state = initial_state;
    bool valid = true;

    // Step 3: Traverse the automaton
    for (int i = 0; input_string[i] != '\0'; i++) {
      char input_char = input_string[i];
      int symbol_index = -1;

      // Find the symbol index
      for (int j = 0; j < num_inputs; j++) {
        if (input_symbols[j] == input_char) {
          symbol_index = j;
          break;
        }
      }

      if (symbol_index == -1) {
        printf("Invalid symbol '%c' in input string.\n", input_char);
        valid = false;
        break;
      }

      current_state = transition_table[current_state][symbol_index];
    }
```

```
      if (valid) {

        bool accepted = false;
        for (int i = 0; i < num_accepting_states; i++) {
          if (current_state == accepting_states[i]) {
            accepted = true;
            break;
          }
        }

        if (accepted) {
          printf("\nInput string is ACCEPTED by the finite automaton.\n");
        } else {
          printf("\nInput string is REJECTED by the finite automaton.\n");
        }
      }

      printf("\nDo you want to test another string? Press 1 for Yes or 0 for Exit: ");
      int choice;
      scanf("%d", &choice);
      if (choice != 1) {
        printf("Exiting...\n");
        break;
      }
    }
}

int main() {
    finite_automata();
    return 0;
}
```

```
Finite Automata Setup
Number of input symbols: 2
Enter the input symbols (space-separated): a b
Number of states: 3
Initial state: 1
Number of accepting states: 1
Enter the accepting states (space-separated): 3

Enter the transition table:
State 1 on input 'a' transitions to: 2
State 1 on input 'b' transitions to: 1
State 2 on input 'a' transitions to: 2
State 2 on input 'b' transitions to: 3
State 3 on input 'a' transitions to: 3
State 3 on input 'b' transitions to: 3

Enter input string: abb

Input string is ACCEPTED by the finite automaton.

Do you want to test another string? Press 1 for Yes or 0 for Exit: 1

Enter input string: aaa

Input string is REJECTED by the finite automaton.

Do you want to test another string? Press 1 for Yes or 0 for Exit: 0
Exiting...


...Program finished with exit code 0
Press ENTER to exit console.
```