

Problem Statement 3: Write a C Program to implement Deterministic Finite Automata (DFA) which accepts a set of input strings.

Input: DFA Parameters

Input String

Output expected: Whether the string is accepted or not.

Description of problem statement: This program is to implement Deterministic Finite Automata (DFA) which accepts a set of input strings.

Algorithm:

Step 1: Start

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 cs for current state to the initial state.

Step 3.2 : For each character in the input string.

Step 3.2.1: Find the index of each character in the array of input symbols. Step 3.2.2 :

Update cs using the transition functions

Step 4: After simulating the DFA on the entire input string, check if the final current state i.e. cs is in the set of accepting states. If it is ,then the input string is accepted by DFA otherwise it is rejected.

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

DFA. Step 6: Stop

CODE:-

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define max 100
int main() {
    int i, j, n, m, o, q[max], ia[max], is, fs[max], tf[max][max], ch, cs;

    char in[max];

    printf("enter the no of states");

    scanf("%d", &n);

    if (n == 0) {
        printf("error");
        exit(0);
    } else {
        printf("enter the states name");
        for (i = 0; i < n; i++) {
            scanf("%d", &q[i]);
        }
    }

    printf("enter the no of input alphabet");

    scanf("%d", &m);

    if (m == 0) {
        printf("error");

        exit(0);
    } else {
        printf("enter the alphabet name");

        for (i = 0; i < m; i++) {
            scanf("%d", &ia[i]);
        }
    }

    printf("enter the initial state");
    scanf("%d", &is);
    if (is >= n) {
```

```
    printf("error");
    exit(0);
}

printf("enter the no of final states");
scanf("%d", &o);

if (o >= n) {
    printf("error");
    exit(0);
} else {
    for (i = 0; i < o; i++) {
        scanf("%d", &fs[i]);
        if (fs[i] >= n) {
            printf("error");
            exit(0);
        }
    }
}

for (i = 0; i < n; i++) {
    for (j = 0; j < m; j++) {
        printf("enter the state at tf[%d][%d]", i, j);
        scanf("%d", &tf[i][j]);
        if (tf[i][j] >= n) {
            printf("error");
            exit(0);
        }
    }
}

printf("enter the input string");
scanf("%s", in);
cs = is;
puts(in);
for (i = 0; i < strlen(in); i++) {
    ch = in[i];
    ch = ch - 48;
    cs = tf[cs][ch];
}
printf("the current state is=%d", cs);
for (i = 0; i < o; i++) {
    if (cs == fs[i]) {
        printf("\nstring accepted");
        exit(0);
    }
}
```

```
}  
printf("\nstring rejected");  
return 0;  
}
```

OUTPUT:

```
enter the no of states3  
enter the states name0  
1  
2  
enter the no of input alphabet2  
enter the alphabet name0  
1  
enter the initial state0  
enter the no of final states2  
1  
2  
enter the state at tf[0][0]1  
enter the state at tf[0][1]0  
enter the state at tf[1][0]2  
enter the state at tf[1][1]1  
enter the state at tf[2][0]0  
enter the state at tf[2][1]1  
enter the input string01  
01  
the current state is=1  
string accepted  
Process returned 0 (0x0)   execution time : 80.906 s  
Press any key to continue.  
_
```