10.Write a C program to find ε -closure for all the states in a Non-Deterministic Finite Automata (NFA) with ε -moves.

Program:

#include <stdio.h>

#include <stdbool.h>

#define MAX\_STATES 100

#define MAX\_TRANSITIONS 100

typedef struct {

int state;

int transition;

} Transition;

typedef struct {

int numStates;

int numTransitions;

Transition transitions[MAX\_TRANSITIONS];

} NFA;

void epsilonClosure(NFA nfa, int state, bool visited[], bool closure[]) {

visited[state] = true;

closure[state] = true;

for (int i = 0; i < nfa.numTransitions; i++) {

Transition transition = nfa.transitions[i];

if (transition.state == state && transition.transition == -1) {

int nextState = nfa.transitions[i].transition;

if (!visited[nextState]) {

epsilonClosure(nfa, nextState, visited, closure);

}

}

}

}

int main() {

NFA nfa;

printf("Enter the number of states: ");

scanf("%d", &nfa.numStates);

printf("Enter the number of transitions: ");

scanf("%d", &nfa.numTransitions);

printf("Enter transition information:\n");

for (int i = 0; i < nfa.numTransitions; i++) {

printf("Transition %d: (state, transition) = ", i + 1);

scanf("%d%d", &nfa.transitions[i].state, &nfa.transitions[i].transition);

}

printf("\nε-Closure for each state:\n");

for (int i = 0; i < nfa.numStates; i++) {

bool visited[MAX\_STATES] = {false};

bool closure[MAX\_STATES] = {false};

epsilonClosure(nfa, i, visited, closure);

printf("ε-Closure(q%d): { ", i);

for (int j = 0; j < nfa.numStates; j++) {

if (closure[j]) {

printf("q%d ", j);

}

}

printf("}\n");

}

return 0;

}