11.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 <stdlib.h>

#include <stdbool.h>

#define MAX\_STATES 100

typedef struct Node {

int state;

struct Node\* next;

} Node;

typedef struct {

Node\* head;

} List;

typedef struct {

int numStates;

List transitions[MAX\_STATES][2];

} NFA;

void addTransition(NFA\* nfa, int fromState, int toState, int type) {

Node\* newNode = (Node\*)malloc(sizeof(Node));

newNode->state = toState;

newNode->next = nfa->transitions[fromState][type].head;

nfa->transitions[fromState][type].head = newNode;

}

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

visited[state] = true;

closure[state] = true;

Node\* current = nfa->transitions[state][0].head;

while (current != NULL) {

if (!visited[current->state]) {

epsilonClosure(nfa, current->state, visited, closure);

}

current = current->next;

}

}

int main() {

NFA nfa;

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

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

int numTransitions;

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

scanf("%d", &numTransitions);

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

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

int fromState, toState, type;

printf("Transition %d: (from state, to state, type (0 for epsilon, 1 for normal)) = ", i + 1);

scanf("%d %d %d", &fromState, &toState, &type);

addTransition(&nfa, fromState, toState, type);

}

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;

}