EXP NO: 3:

## FINDING ε-CLOSURE FOR NFA WITH ε-MOVES:

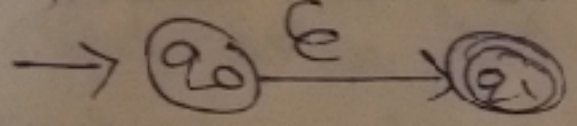
AIM: To write a c program to find ε-closure of a non-deterministic finite Automata with ε-moves.

ALGORITHM:

1. Get the following input from the user
   (i) Number of states in NFA
   (ii) Number of symbol input alphabet include
   (iii) input symbols.
   (iv) Number of ~~Symb~~ final states and their names.

2. Declare a 3-dimensional matrix to store the transistions and initialize.

3. Get the transitions from every state for input symbol from the user and state




4. Initialize of two-dimensional matrix ε-closure with -1 in all entries

5. ε-closure of state q is defined at set state that can be reached.

   ε-closure (0) = 0, 1
   ε-closure (1) = 1,
   ε-closure (2) = 2,

6. For every state. Print e-closure values.

Program:

```
#include <stdio.h>
#include <string.h>
  int trans_table [10][5][3];
  char symbol [5], e;
  int e_closure [10][10], n, state;
  void find_e_closure (int x);
  int main()
{
int i, j, k, n, num_state, num_symbols;
for (i=0; i<10; i++)
 {
  for (j=0; j<5; j++)
  {
   for (k=0; k<3; k++)
   {
    trans_table[i][j][k] = -1;
   }
  }
}

num_states = 3;
 num_symbols = 2;
  symbol[0] = 'e';
  n=1;
 tran_table [0] [0][0] = 1;
 for (i=0; i<10', i++)
{
 for(j=0; j<10; j++)
{
 e_closure [i][j] e = 1;
 }
}
for(i=0; i<num_state; i++)
  e_ closure [i][0] = 1;
for (i=0; i<num_state; i++)
{
```

```
}
for(i=0; i<num_states; i++)
{
printf("e-closure(%d): ",i);
for(j=0; j<num_alphabets; j++)
{if(e_closure[i][j]!=-1)
{
printf("%d ",e_closure[i][j]);
}
}
printf("\n");
}}
void find_e_closure(int x)
{int i,j,y[10],num_trans;
i=0;
while(trans_table[x][0][i]!=-1){
y[i]=trans_table[x][0][i];
i=i+1;
}
num_trans=i;
for(j=0;j<num_trans;j++)
{
e_closure[state][pt][v]=y[i]
pt=pt+1;
find_e_closure(y[i]);
}}
```

and put:

e-closure(a) = {a,c}

e-closure(c) = {c}

e-closure(e) = {e}.