

Question 1

Write program to find ϵ – closure of all states of any given NFA with ϵ transition.

Program

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int st;
struct node *link;
};
void findclosure(int,int);
void insert_trantbl(int ,char, int);
int findalpha(char);
void print_e_closure(int);
static int set[20],nostate,noalpha,s,notransition,c,r,buffer[20];
char alphabet[20];
static int e_closure[20][20]={0};
struct node * transition[20][20]={NULL};
void main()
{
int i,j,k,m,t,n;
struct node *temp;
printf("Enter the number of alphabets?\n");
scanf("%d",&noalpha);
getchar();
printf("NOTE:- [ use letter e as epsilon]\n");
printf("NOTE:- [e must be last character ,if it is present]\n");
printf("\nEnter alphabets?\n");
for(i=0;i<noalpha;i++)
{
alphabet[i]=getchar();
getchar();
}
printf("\nEnter the number of states?\n");
scanf("%d",&nostate);
printf("\nEnter no of transition?\n");
scanf("%d",&notransition);
printf("NOTE:- [Transition is in the form-> qno alphabet
qno]\n",notransition);
```

```

printf("NOTE:- [States number must be greater than zero]\n");
printf("\nEnter transition?\n");
for(i=0;i<notransition;i++)
{
scanf("%d %c%d",&r,&c,&s);
insert_trantbl(r,c,s);
}
printf("\n");
printf("e-closure of states.....\n");
printf("_____\n");
for(i=1;i<=nostate;i++)
{
c=0;
for(j=0;j<20;j++)
{
buffer[j]=0;
e_closure[i][j]=0;
}
findclosure(i,i);
printf("\ne-closure(q%d): ",i);
print_e_closure(i);
}
}
void findclosure(int x,int sta)
{
struct node *temp;
int i;
if(buffer[x])
return;
e_closure[sta][c++]=x;
buffer[x]=1;
if(alphabet[noalpha-1]=='e' && transition[x][noalpha-1]!=NULL)
{
temp=transition[x][noalpha-1];
while(temp!=NULL)
{
findclosure(temp->st,sta);
temp=temp->link;
}
}
}
void insert_trantbl(int r,char c,int s)
{
int j;
struct node *temp;
j=findalpha(c);
if(j==999)

```

```

{
printf("error\n");
exit(0);
}
temp=(struct node *)malloc(sizeof(struct node));
temp->st=s;
temp->link=transition[r][j];
transition[r][j]=temp;
}
int findalpha(char c)
{
int i;
for(i=0;i<noalpha;i++)
if(alphabet[i]==c)
return i;
return(999);
}
void print_e_closure(int i)
{
int j;
printf("{");
for(j=0;e_closure[i][j]!=0;j++)
printf("q%d,",e_closure[i][j]);
printf("}");
}

```

Output :

```

Enter the number of states: 4
Enter the number of alphabets: 2
Enter the alphabets:
(Enter e for Epsilon & add it as the last alphabet)
a
e
Enter the number of transistions: 4

NOTE: State number begins at 1
Enter transition in the format: CurrentStateNumber Alphabet NextStateNumber
1 e 2
2 a 3
1 e 3
3 e 4
Epsilon closure for the states are:
e-closure(q1) = { q1, q3, q4, q2 }
e-closure(q2) = { q2 }
e-closure(q3) = { q3, q4 }
e-closure(q4) = { q4 }

```

