Aim: write program to find ε – closure of all states of any given NFA with ε transition.

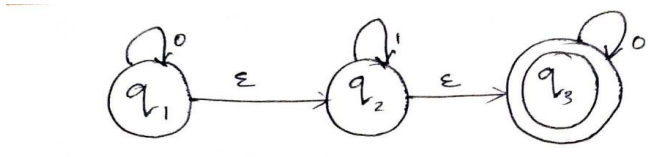
Program:

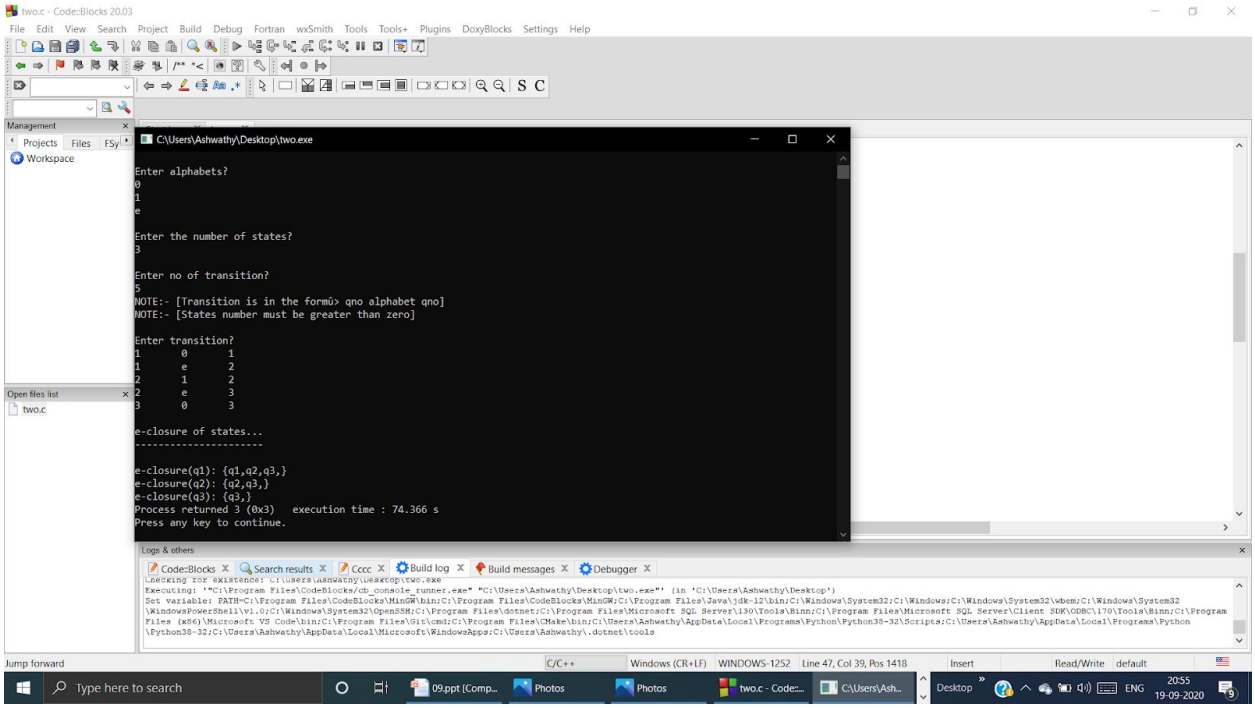
```
#include<stdio.h>
#include<stdlib.h>
struct node
{
	int st;
	struct node *link;
};

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




two.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
C:\Users\Ashwathy\Desktop\two.exe
Enter alphabets?
0
1
e
Enter the number of states?
3
Enter no of transition?
5
NOTE:- [Transition is in the formü> qno alphabet qno]
NOTE:- [States number must be greater than zero]
Enter transition?
1 0 1
1 e 2
2 1 2
2 e 3
3 0 3
e-closure of states...
e-closure(q1): {q1,q2,q3,}
e-closure(q2): {q2,q3,}
e-closure(q3): {q3,}
Process returned 3 (0x3) execution time : 74.366 s
Press any key to continue.
Management
Projects Files FSy
Workspace
Open files list
two.c
Logs & others
Code::Blocks Search results Cccc Build log Build messages Debugger
Jump forward
C/C++ Windows (CR+LF) WINDOWS-1252 Line 47, Col 39, Pos 1418 Insert Read/Write default
Type here to search
09.ppt [Comp... Photos Photos two.c - Code::... C:\Users\Ash...
Desktop ENG 20:55 19-09-2020