Akshat Agarwal  
RA1911003010646

**Compiler Design**

**Lab – 4**

**RE to NFA Conversion**

#include<stdio.h>

#include<string.h>

#include<ctype.h>

int main()

{

char m[20],t[10][10]; int n,i,j,r=0,c=0;

printf("\t\t\t\tSIMULATION OF NFA"); printf("\n\t\t\t\t \*");

for(i=0;i<10;i++)

{

for(j=0;j<10;j++)

{

t[i][j]=' ';

}

}

printf("\nEnter a regular expression:"); scanf("%s",m);

n=strlen(m); for(i=0;i<n;i++)

{

switch(m[i])

{

case '|' : { t[r][r+1]='E';

t[r+1][r+2]=m[i-1];

t[r+2][r+5]='E';

t[r][r+3]='E';

t[r+4][r+5]='E';

t[r+3][r+4]=m[i+1];

r=r+5; break;

}

case '\*':{ t[r-1][r]='E';

t[r][r+1]='E';

t[r][r+3]='E';

t[r+1][r+2]=m[i-1];

t[r+2][r+1]='E';

t[r+2][r+3]='E';

r=r+3; break;

}

case '+': { t[r][r+1]=m[i-1];

t[r+1][r]='E';

r=r+1; break;

}

default:

{

if(c==0)

{

if((isalpha(m[i]))&&(isalpha(m[i+1])))

{

t[r][r+1]=m[i];

t[r+1][r+2]=m[i+1];

r=r+2; c=1;

} c=1;

}

else if(c==1)

{

if(isalpha(m[i+1]))

{ t[r][r+1]=m[i+1];

r=r+1; c=2;

}

}

else

{

if(isalpha(m[i+1]))

{ t[r][r+1]=m[i+1];

r=r+1; c=3;

}

}

}

break;

}

}

printf("\n"); for(j=0;j<=r;j++) printf(" %d",j);

printf("\n \n");

printf("\n"); for(i=0;i<=r;i++)

{

for(j=0;j<=r;j++)

{

printf(" %c",t[i][j]);

}

printf(" | %d",i);

printf("\n");

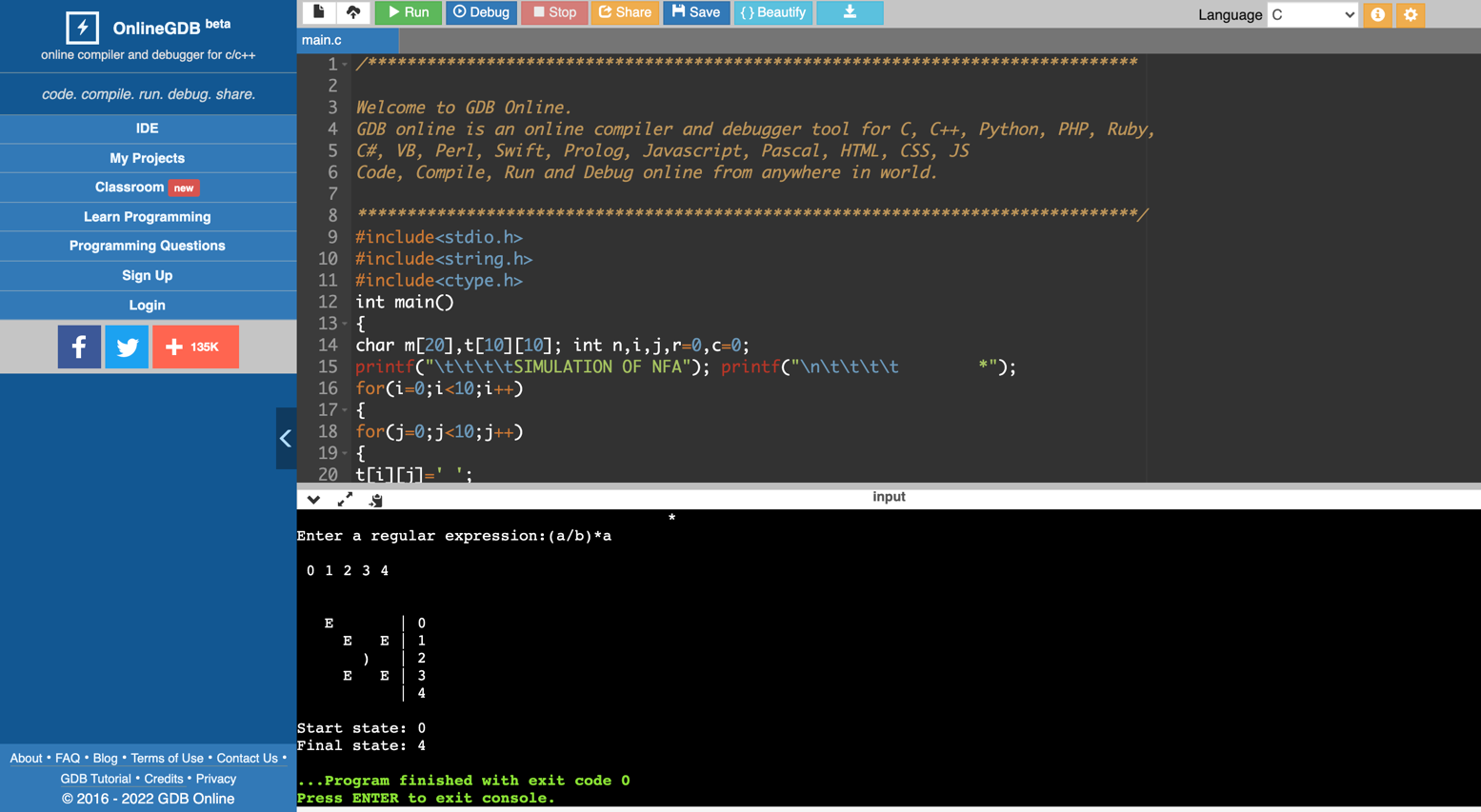
}

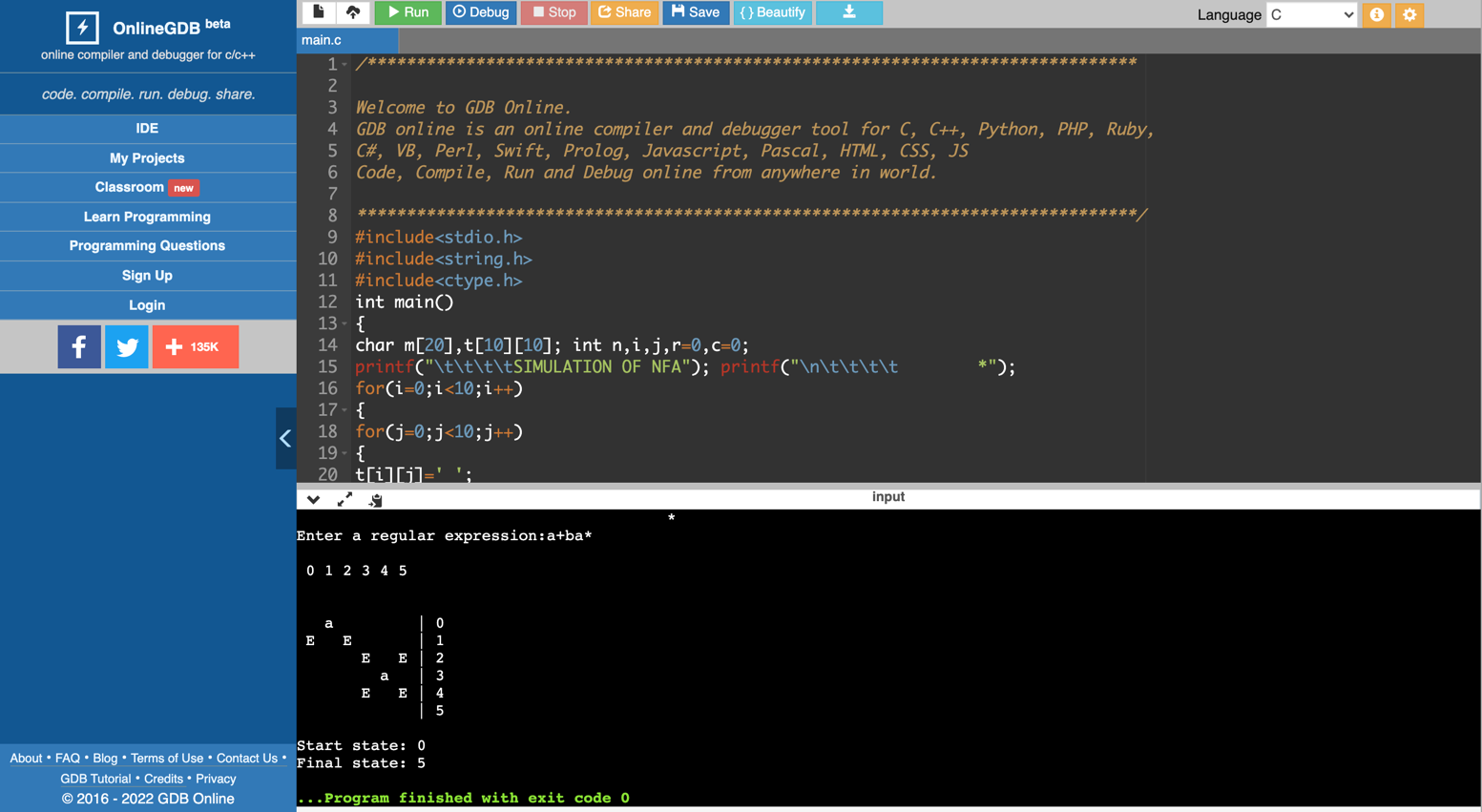
printf("\nStart state: 0\nFinal state: %d",i-1);

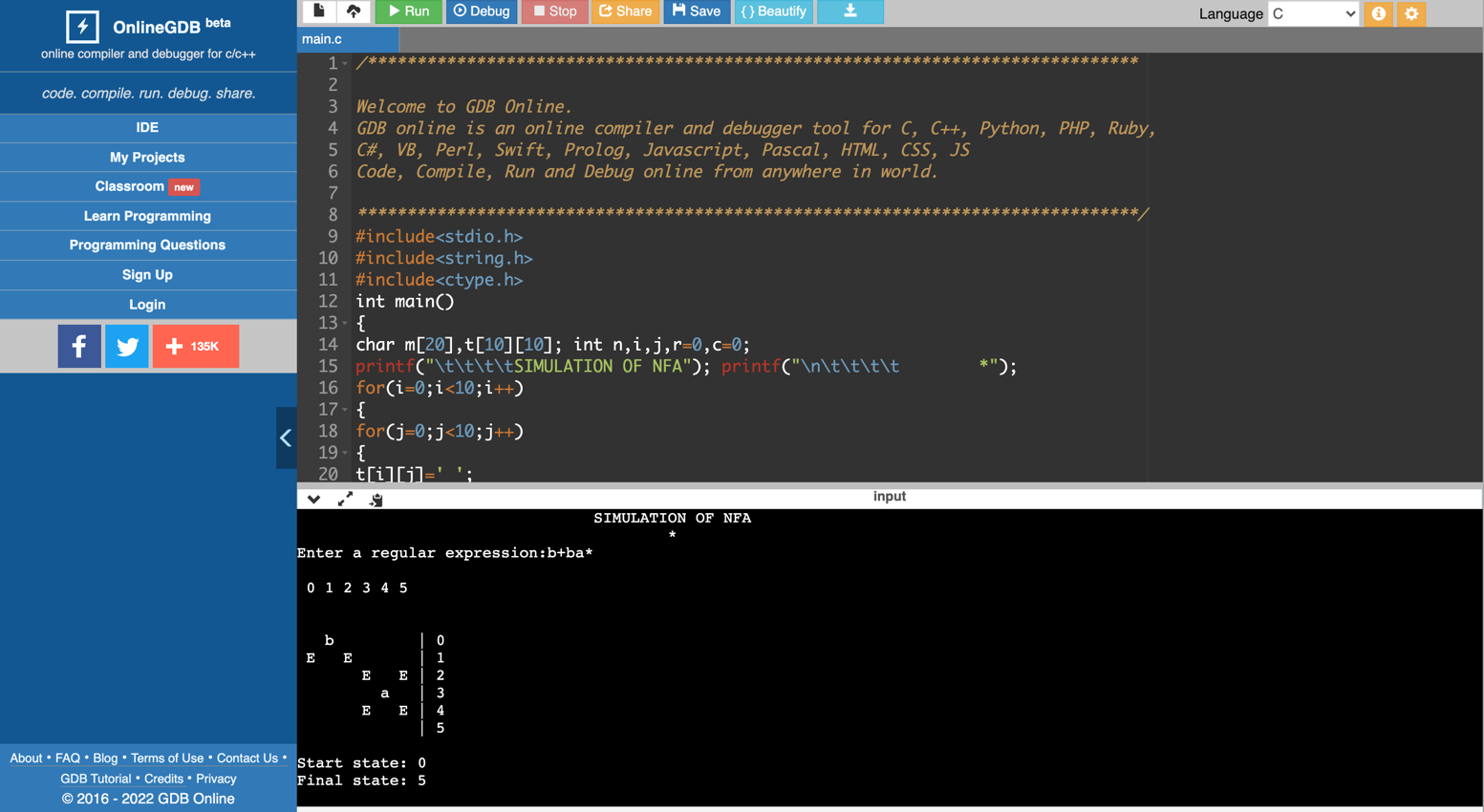
return 0;

}

**OUTPUT :**

****

****

****