**PRACTICAL NO.5**

**AIM:** To compute first() and follow() of a given grammar

**PROGRAM:**

#include<stdio.h>

#include<string.h>

#include<ctype.h>

int n,m=0,p,i=0,j=0;

char a[10][10],f[10];

void follow(char c);

void first(char c);

int main(){

int i,z;

char c,ch;

printf("Enter the no of prooductions:\n");

scanf("%d",&n);

printf("Enter the productions(epsilon=$):\n");

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

scanf("%s%c",&a[i],&ch);

do{

m=0;

printf("Enter the elements whose first & follow is to be found:");

scanf("%c",&c);

first(c);

printf("First(%c)={",c);

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

printf("%c",f[i]);

printf("}\n");

strcpy(f," ");

// flushall();

m=0;

follow(c);

printf("Follow(%c)={",c);

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

printf("%c",f[i]);

printf("}\n");

printf("Continue(0/1)?");

scanf("%d%c",&z,&ch);

}while(z==1);

return(0);

}

void first(char c)

{

int k;

if(!isupper(c))

f[m++]=c;

for(k=0;k<n;k++)

{

if(a[k][0]==c)

{

if(a[k][2]=='$')

follow(a[k][0]);

else if(islower(a[k][2]))

f[m++]=a[k][2];

else first(a[k][2]);

}

}

}

void follow(char c)

{

if(a[0][0]==c)

f[m++]='$';

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

{

for(j=2;j<strlen(a[i]);j++)

{

if(a[i][j]==c)

{

if(a[i][j+1]!='\0')

first(a[i][j+1]);

if(a[i][j+1]=='\0' && c!=a[i][0])

follow(a[i][0]);

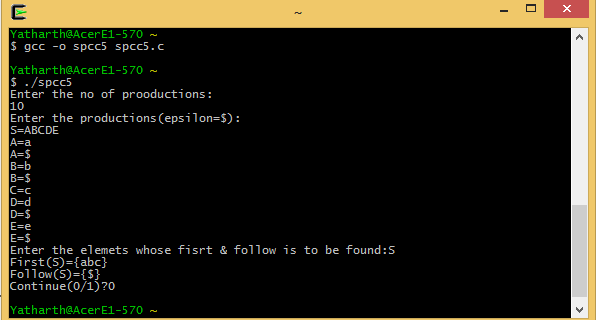
}

}

}

}

**OUTPUT:**



**CONCLUSION:** Thus a C program for computing first() and follow() of a given grammar has been implemented.