**Experiment No. 5**

**A.I.M -:** Write a C program to find first of a given string.

Source Code-:

#include<stdio.h>

#include<conio.h>

void FIRST(char);

int count, n=0;

char prodn[10][10] , first[10];

void main()

{

int i, choice ;

char c,ch;

printf(“how many productions? :\n”);

scanf(“%d”,&count);

printf(“enter %d products with epsilon =$:\n”,count);

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

{

scanf(“%s %c “,prodn[i],&ch);

do

{

n=0;

printf(“element :\n”);

scanf(“%c”,&c);

FIRST(c) ;

Printf(“\n FIRST(%c) = {“,c);

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

{

printf(“%c”,first[i]);

printf(“}\n”);

printf(“press 1 to continue:”);

scanf(“%d %c”,& choice,&ch);

}

while(choice==1);

}

void FIRST(char c)

{

int j;

if( isupper(c) first[n++]==c)

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

{

if(prodn[j][2]==’$’)

first[n++]==’$’;

elseif(islower(prodn[j][2]) )

first[n++]=prodn[j][2];

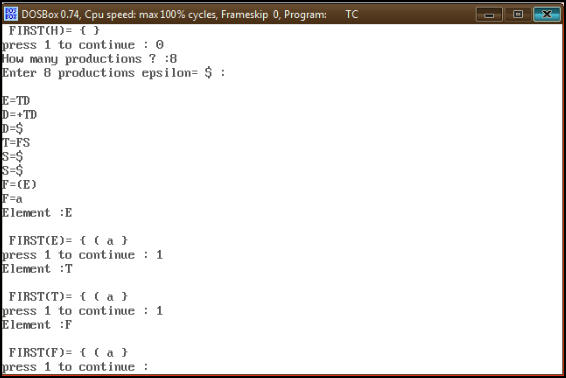
else FIRST(prodn[j][2]);

}

}

}

Output- :



**Experiment No. 6**

**A.I.M-:** Write a program to find follow of a given string

Source Code -:

#include<stdio.h>

#include<conio.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 number of productions:”);

Scanf(“%d”,&n);

Printf(“\n enter the productions with epsilon &”,\n);

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

Scanf(“%s%c,a[i],&ch);

Do

{

m=0;

printf(“enter the element whose follow is to be found:”);

scanf(“%c”,&c);

follow(c);

printf(“follow(%c)={“,c);

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

printf(“%c”,f[i]);

printf(“}\n”);

printf(“do you want to continue? \n press 1 \n”);

scanf(“%d%c”,&z,&ch);

}

While(z==1)

}

Void follw(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]==()

{

If(a[i][j+1]!=’\0’)

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

If(a[i][j+1]==’\0’ &&c !=[i][0])

Follow(a[i][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[i][0]);

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

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

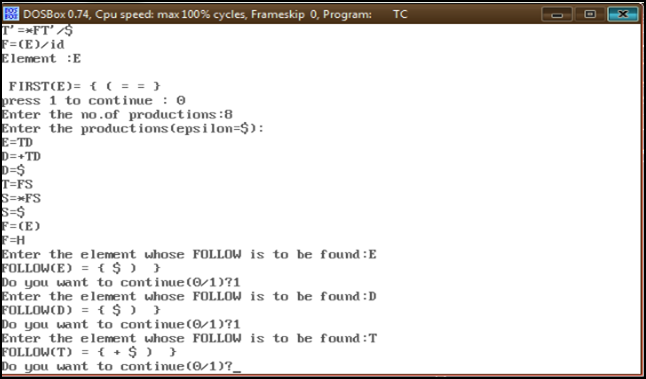
Else first(a[k][2]);

}

}

}

Output-:



**Experiment No.7**

**A.I.M-:** Write a program to implement 3 address code.

**Source Code-:**

#include<stdio.h>

#include<string.h>

void pm();

void plus();

void div();

int i,ch,j,l,addr=100;

char ex[10],exp[10],exp1[10],exp2[10],id1[5],op[5],id2[5];

void main()

{

clrscr();

while(1)

{

printf("\n1.assignment\n2.arithmetic\n3.relational\n4.Exit\nEnter the choice:");

scanf("%d",&ch);

switch(ch)

{

case 1:

printf("\nEnter the expression with assignment operator:");

scanf("%s",exp);

l=strlen(exp);

exp2[0]='\0';

i=0;

while(exp[i]!='=')

{

i++;

}

strncat(exp2,exp,i);

strrev(exp);

exp1[0]='\0';

strncat(exp1,exp,l-(i+1));

strrev(exp1);

printf("Three address code:\ntemp=%s\n%s=temp\n",exp1,exp2);

break;

case 2:

printf("\nEnter the expression with arithmetic operator:");

scanf("%s",ex);

strcpy(exp,ex);

l=strlen(exp);

exp1[0]='\0';

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

{

if(exp[i]=='+'||exp[i]=='-')

{

if(exp[i+2]=='/'||exp[i+2]=='\*')

{

pm();

break;

}

else

{

plus();

break;

}

}

else if(exp[i]=='/'||exp[i]=='\*')

{

div();

break;

}

}

break;

case 3:

printf("Enter the expression with relational operator");

scanf("%s%s%s",&id1,&op,&id2);

if(((strcmp(op,"<")==0)||(strcmp(op,">")==0)||(strcmp(op,"<=")==0)||(strcmp(op,">=")==0)||(strcmp(op,"==")==0)||(strcmp(op,"!=")==0))==0)

printf("Expression is error");

else

{

printf("\n%d\tif %s%s%s goto %d",addr,id1,op,id2,addr+3);

addr++;

printf("\n%d\t T:=0",addr);

addr++;

printf("\n%d\t goto %d",addr,addr+2);

addr++;

printf("\n%d\t T:=1",addr);

}

break;

case 4:

exit(0);

}

}

}

void pm()

{

strrev(exp);

j=l-i-1;

strncat(exp1,exp,j);

strrev(exp1);

printf("Three address code:\ntemp=%s\ntemp1=%c%ctemp\n",exp1,exp[j+1],exp[j]);

}

void div()

{

strncat(exp1,exp,i+2);

printf("Three address code:\ntemp=%s\ntemp1=temp%c%c\n",exp1,exp[i+2],exp[i+3]);

}

void plus()

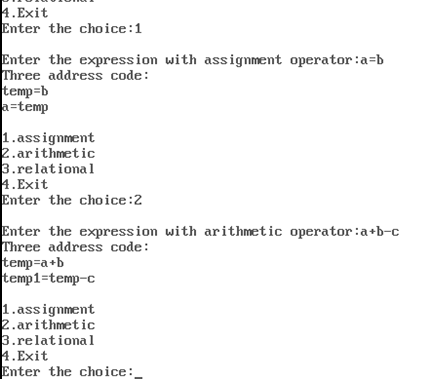
{

strncat(exp1,exp,i+2);

printf("Three address code:\ntemp=%s\ntemp1=temp%c%c\n",exp1,exp[i+2],exp[i+3]);

}

***Output-:***

******