Compiler Design

EXPERIMENT 3

RAHUL KUMAR

RA1911003010699

AIM:

A program for Elimination of Left Recursion.

ALGORITHM:

1. Start the program.
2. Initialize the arrays for taking input from the user.
3. Prompt the user to input the no. of non-terminals having left recursion and no. of productions for these non-terminals.
4. Prompt the user to input the production for non-terminals.
5. Eliminate left recursion using the following rules
    a. A->Aα1| Aα2 | . . . . . |Aαm
    b. A->β1| β2| . . . . .| βn
    c. Then replace it by
    d. A-> βi A’ i=1,2,3,…..m
    e. A’-> αj A’ j=1,2,3,…..n
    f. A’-> Ɛ
6. After eliminating the left recursion by applying these rules, display the productions without left recursion.
7. Stop.

PROGRAM:

```
#include<string.h>
#include<stdio.h>
#include<stdlib.h>
void main()
{
    char a[10],b[50][10]={""},d[50][10]={""},ch;
    int i,n,c[10]={0},j,k,t,n1;

    printf("\nEnter the left production(s) (NON TERMINALS) : ");
    scanf("%s",a);
    n=strlen(a);
    for(i=0;i<n;i++)
    {
        printf("\nEnter the number of productions for %c : ",a[i]);
        scanf("%d",&c[i]);
    }
    t=0;
    for(i=0;i<n;i++)
    {
        printf("\nEnter the right productions for %c",a[i]);
        k=t;
        for(j=0;j<c[i];j++)
        {
            printf("\n%c->",a[i]);
            do
            {
                scanf("%s",b[k]);
                k++;
            }while(k<j);
        }
        t=t+10;
    }
    t=0;
    for(i=0;i<n;i++)
    {
        if(a[i]==b[t][0])
        {
            n1=strlen(b[t]);
            for(k=1;k<n1;k++)
            {
                d[t][k-1]=b[t][k];
            }
        }
        t=t+10;
    }
    t=0;
    printf("\n\nThe resulting productions after eliminating Left
Recursion are : \n");
    for(i=0;i<n;i++)
    {
        if(a[i]==b[t][0])
        {
            for(j=1;j<c[i];j++)
            {
                printf("\n%c -> %s%c'",a[i],b[t+j],a[i]);
            }
        }
        t=t+10;
    }
    t=0;
    for(i=0;i<n;i++)
    {
        if(a[i]==b[t][0])
            printf("\n%c' -> %s%c'|%c",a[i],d[t],a[i],(char)238);
        else
            for(j=0;j<c[i];j++)
                printf("\n%c -> %s",a[i],b[t+j]);
        t=t+10;
    }
}
```

INPUT:

Enter the left production(s) (NON TERMINALS) : ETF

Enter the number of productions for E : 2

Enter the number of productions for T : 2

Enter the number of productions for F : 2

Enter the right productions for E

E->E+T

E->T

Enter the right productions for T

T->T*F

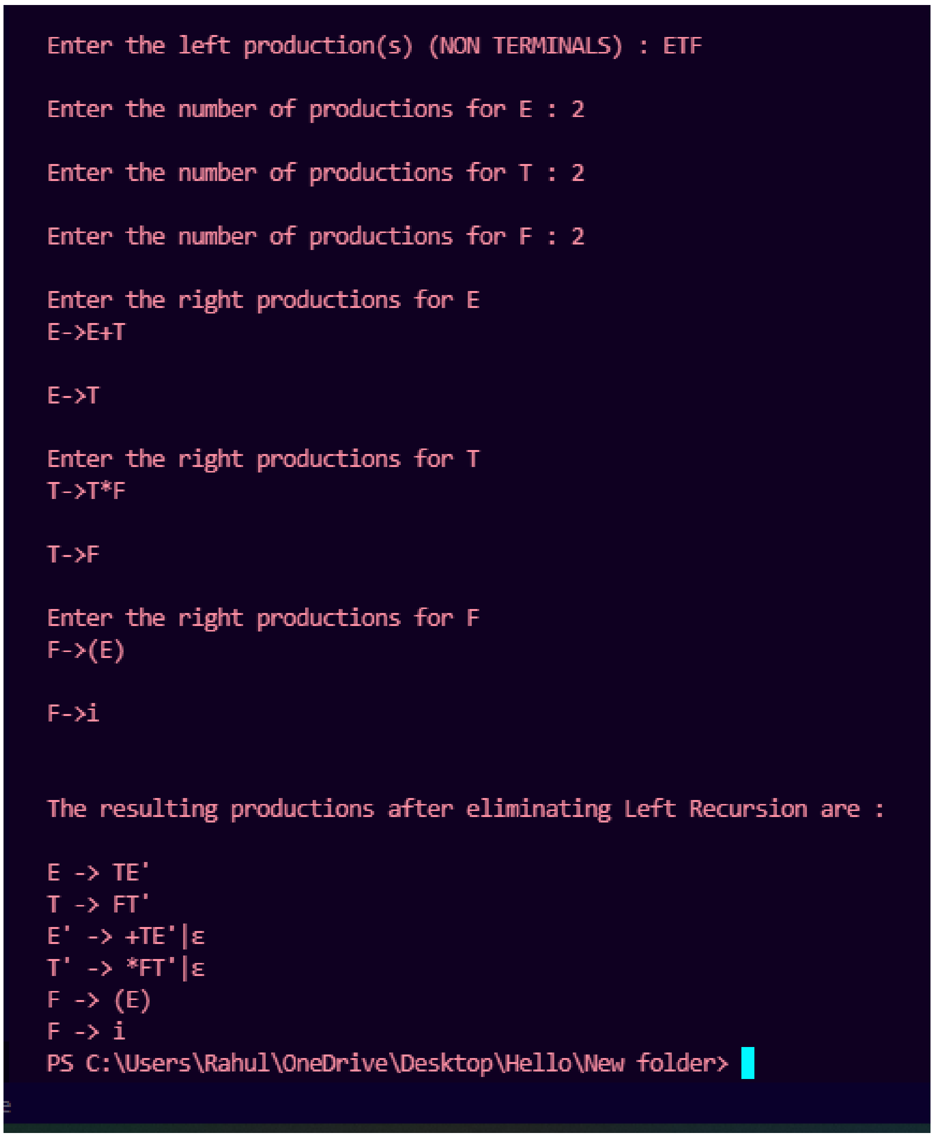
T->F

Enter the right productions for F

F->(E)

F->i

OUTPUT:



```
Enter the left production(s) (NON TERMINALS) : ETF

Enter the number of productions for E : 2

Enter the number of productions for T : 2

Enter the number of productions for F : 2

Enter the right productions for E
E->E+T

E->T

Enter the right productions for T
T->T*F

T->F

Enter the right productions for F
F->(E)

F->i


The resulting productions after eliminating Left Recursion are :

E -> TE'
T -> FT'
E' -> +TE'|ε
T' -> *FT'|ε
F -> (E)
F -> i
PS C:\Users\Rahul\OneDrive\Desktop\Hello\New folder>
```