Lab program 4:

4. Write a program to eliminate Left Recursion from the grammar.

Aim: Program to eliminate left recursion from the given grammar.

Algorithm:

1. Assign an ordering $A1,\ldots,An$ to the nonterminals of the grammar.
2. for i:=1to n do begin
3. for j:=1 to $i-1$ do begin
4. for each production of the form $Ai \rightarrow Aj\alpha$ do begin
5. remove $Ai \rightarrow Aj\alpha$ from the grammar
6. for each production of the form $Aj \rightarrow \beta$ do begin
7. add $Ai \rightarrow \beta\alpha$ to the grammar

end
end
end
8. transform the Ai-productions to eliminate direct left recursion
end

Program:

```
#include<stdio.h>
#include<string.h>
#define SIZE 10
int main ( )
{
    char non_terminal;
    char beta,alpha;
    int num;
    char production[10][SIZE];
    int index=3; /* starting of the string following "->" */
    printf("Enter Number of Production : ");
    scanf("%d",&num);
    printf("Enter the grammar as E->E-A :\n");
    for(int i=0;i<num;i++)
    {
        scanf("%s",production[i]);
    }
    for(int i=0;i<num;i++)
    {
        printf("\nGRAMMAR : : : %s",production[i]);
        non_terminal=production[i][0];
        if(non_terminal==production[i][index])
        {
            alpha=production[i][index+1];
            printf(" is left recursive.\n");
            while(production[i][index]!=0 && production[i][index]!='|')
                index++;
            if(production[i][index]!=0)
            {
                beta=production[i][index+1];
```

```
                printf("Grammar without left recursion:\n");
                printf("%c->%c%c\'",non_terminal,beta,non_terminal);
                printf("\n%c\'->%c%c\'|E\n",non_terminal,alpha,non_terminal);
            }
            else
                printf(" can't be reduced\n");
        }
        else
            printf(" is not left recursive.\n");
        index=3;
    }
}
```

Output:

Enter Number of Production: 3
Enter the grammar as E ->E-A :
E ->EA | A
A ->AT | a
T -> a

GRAMMAR : : : E ->EA| A is left recursive.
Grammar without left recursion:
E ->AE ᴵ
E ᴵ ->AE ᴵ|ϵ

GRAMMAR : : : A ->AT| a is left recursive.
Grammar without left recursion:
A ->aA ᴵ
A ᴵ ->TA ᴵ|ϵ

GRAMMAR : : : T ->a is not left recursive.