## **Experiment - 5 WAP to remove left recursion from a grammar**

```
Name:
                                                   Rollno
Class:
              TE.CO
                                                   Batch
#Source Code
author = 'Shadab Shaikh'
__title__ = 'Finding & resolving left recursion from a grammar'
__date__ = '26-02-2019'
__version__ = '1.0'
                     : ' + __author__)
print('Author
                     : ' + __title__)
print('Title
print('Date
                     : ' + __date__)
                     : ' + __version_ )
print('Version
grammararr=[]
                     #stores the grammar maintaining the index
alpharest=[]
                     #stores the alpha and rest values
                     #stores beta
beta=[]
inputs=""
newprod=""
                     #to display new production
while(inputs!='no'):
       grammar = input("\nEnter the grammar left should be variable following with ->
format eg: S \rightarrow a \ n'')
       grammar=grammar.replace(" ","")
                                                   #replacing whitespaces with none
       if(grammar[0].islower()):
              grammar[0].upper()
                                                   #making left most as variable
       grammararr.append(grammar)
                                                   #storing into list
       inputs = input("\nPress no to stop writing productions or write anything to continue")
                                           #asking for the continuity of grammar
def findalpharest(grammararr,k):
       """function to find alpha and rest values"""
       if(grammararr[k][0]==grammararr[k][3]):
              alpharest.append(grammararr[k][0]+grammararr[k][4:])
       k+=1
       if(k<len(grammararr)):</pre>
              findalpharest(grammararr,k)
findalpharest(grammararr,0)
                                           #calling findalpharest function
def findbeta(grammararr,alpharest,l,m):
```

```
"""function to find beta for corresponding alpha values"""
       if(grammararr[1][0]==alpharest[m][0]):
               if(grammararr[1][0]!=grammararr[1][3]):
                       beta.append(grammararr[l][0]+grammararr[l][3])
       l+=1
       if(l<len(grammararr)):</pre>
               findbeta(grammararr,alpharest,l,m)
       else:
               l=0
               m+=1
               if(m<len(alpharest)):</pre>
                       findbeta(grammararr,alpharest,l,m)
findbeta(grammararr,alpharest,0,0) #calling findbeta function
def formnewprod(grammararr,alpharest,beta,i,j):
       #function to print new production
       print(alpharest[i][0]+"->"+beta[j][1]+alpharest[i][0]+"")
       print(alpharest[i][0]+""+"->"+alpharest[i][1]+alpharest[i][0]+""")
       print(alpharest[i][0]+"""+"#")
       i+=1
       i+=1
       if(i<len(alpharest) and j<len(beta)):</pre>
               formnewprod(grammararr,alpharest,beta,i,j)
formnewprod(grammararr,alpharest,beta,0,0)
                                                     #calling formnewprod function
#print(*alpharest, sep='\n')
#print(*beta, sep='\n')
#Sample input
1.
       E \rightarrow E + T/T
2.
       S \rightarrow (L)/x
```

L-Ls/s

## **#Output**

## 1. Output for the 1<sup>st</sup> sample input

```
alktc@alktc1:/media/alktc/D2CS-103D/Academics/[Study Material]/Degree/CBGS REV 2016 2017-2020 MUMBAI UNIVERSITY/TE/Sem 6/Labs/CSL602 System Software Lab 01/Experiment 5 - + × File Edit View Search Terminal Help alkt@alktc1:/media/alktc/D2CS-103D/Academics/[Study Material]/Degree/CBGS REV 2016 2017-2020 MUMBAI UNIVERSITY/TE/Sem 6/Labs/CSL602 System Software Lab 01/Experiment 5 - Left Recursions python3 Leftrec.py  
Author : Shadab Shaikh  
Title : Finding 6 resolving left recursion from a grammar  
Date : 26-02-2019  
Version : 1.0  
Enter the grammar Left should be variable following with -> format eg: S->a  
E->ET  
Press no to stop writing productions or write anything to continueno  
E->TE  
E->  
Press no to stop writing productions or write anything to continueno  
E->TE  
E-> Left Recursions  
Baltc@alktc1:/media/alktc/D2CS-103D/Academics/[Study Material]/Degree/CBGS REV 2016 2017-2020 MUMBAI UNIVERSITY/TE/Sem 6/Labs/CSL602 System Software Lab 01/Experiment 5  
- Left Recursions  
Baltc@alktc1:/media/alktc/D2CS-103D/Academics/[Study Material]/Degree/CBGS REV 2016 2017-2020 MUMBAI UNIVERSITY/TE/Sem 6/Labs/CSL602 System Software Lab 01/Experiment 5  
- Left Recursions  
Baltc@alktc1:/media/alktc/D2CS-103D/Academics/[Study Material]/Degree/CBGS REV 2016 2017-2020 MUMBAI UNIVERSITY/TE/Sem 6/Labs/CSL602 System Software Lab 01/Experiment 5  
- Left Recursions  
Baltc@alktc1:/media/alktc/D2CS-103D/Academics/[Study Material]/Degree/CBGS REV 2016 2017-2020 MUMBAI UNIVERSITY/TE/Sem 6/Labs/CSL602 System Software Lab 01/Experiment 5  
- Left Recursions  
Baltc@alktc1:/media/alktc/D2CS-103D/Academics/[Study Material]/Degree/CBGS REV 2016 2017-2020 MUMBAI UNIVERSITY/TE/Sem 6/Labs/CSL602 System Software Lab 01/Experiment 5  
- Left Recursions  
Baltc@alktc1:/media/alktc/D2CS-103D/Academics/[Study Material]/Degree/CBGS REV 2016 2017-2020 MUMBAI UNIVERSITY/TE/Sem 6/Labs/CSL602 System Software Lab 01/Experiment 5  
- Left Recursions  
- Left Re
```

## 2. Output for the 2<sup>nd</sup> sample input

```
alktc@alktc1:/media/alktc/D2C5-103D/Academics/[Study Material]/Degree/CBGS REV 2016 2017-2020 MUMBAI UNIVERSITY/TE/Sem 6/Labs/CSL602 System Software Lab 01/Experiment 5 - Left Recursions pythonal leftrec.py
alktc@alktcl.predia/alktc/D2C5-103D/Academics/[Study Material]/Degree/CBGS REV 2016 2017-2020 MUMBAI UNIVERSITY/TE/Sem 6/Labs/CSL602 System Software Lab 01/Experiment 5 - Left Recursions pythonal leftrec.py
alktc@alktcl.predia/alktc/D2C5-103D/Academics/[Study Material]/Degree/CBGS REV 2016 2017-2020 MUMBAI UNIVERSITY/TE/Sem 6/Labs/CSL602 System Software Lab 01/Experiment 5 - Left Recursions pythonal leftrec.py
alktc@alktcl.predia/alktc/D2C5-103D/Academics/[Study Material]/Degree/CBGS REV 2016 2017-2020 MUMBAI UNIVERSITY/TE/Sem 6/Labs/CSL602 System Software Lab 01/Experiment 5 - Left Recursions 1 - December 1 - Left Recursions 1 - December 2 - Left Recursions 3 - Left Recursions 4 - Left Recursions 3 - Left Recursions 3 - Left Recursions 3 - Left Recursions 4 - Left Recursions 5 - Left Rec
```