***Elimination of Left Recursion and Left Factoring.***

***Ayush Jindal***

***RA1911003010308***

**AIM:**

Elimination of Left Recursion and Left Factoring.

**CODE and OUTPUT:**

**(LEFT RECURSION)**

def func(n):

s= input("Enter the grammar:")

if s[0]!=s[3]:

print("No left recursion")

if s[0]==s[3]:

l=len(s)

alpha=''

beta=''

b=s[0]+"'"

for i in range(l):

if s[i]=="|":

alpha=alpha+s[i+1:]

for i in range(l):

if s[i]=="|":

beta=beta+s[4:i]

alpha=alpha+b

beta=beta+b

print("\_\_\_")

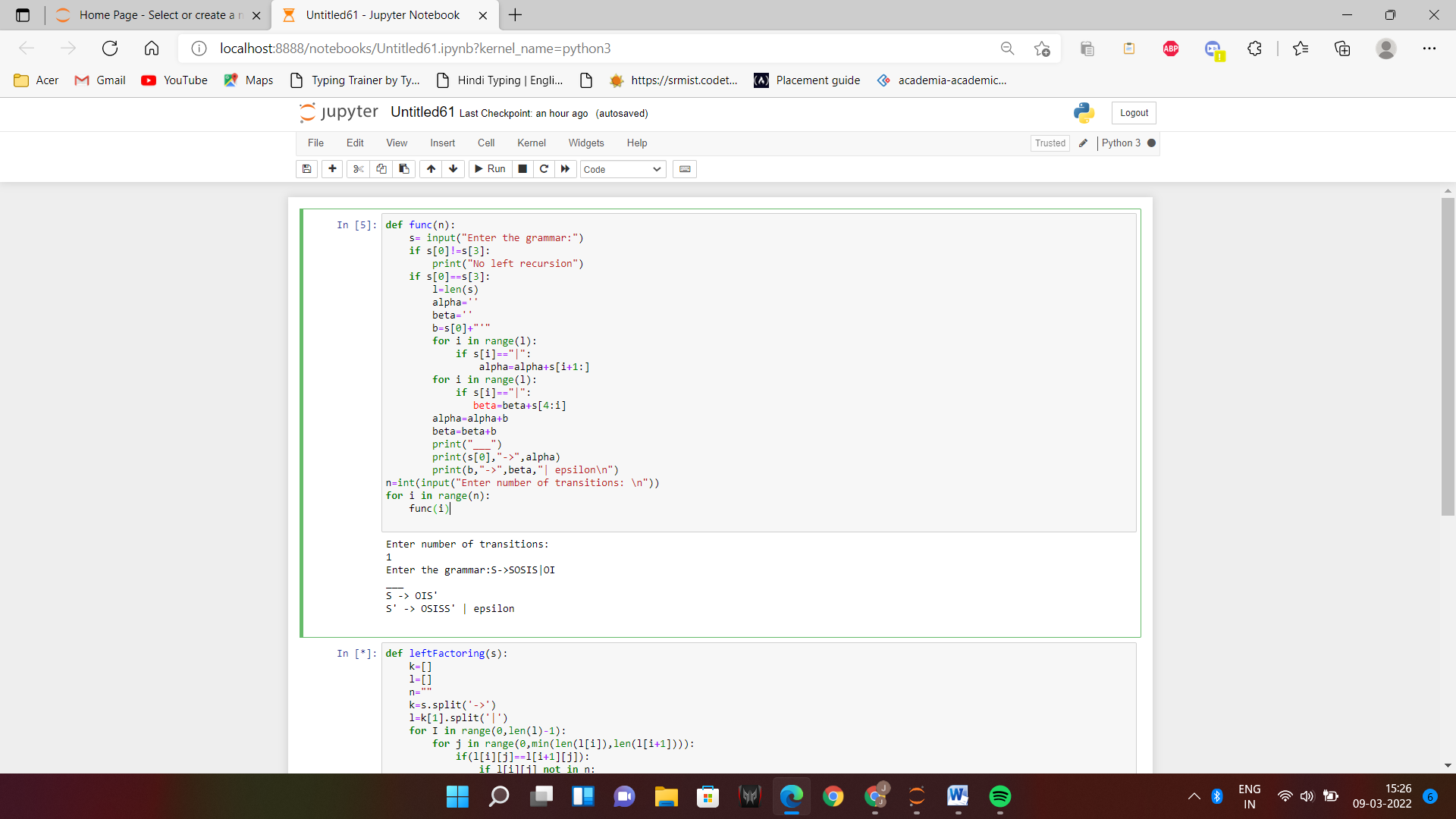
print(s[0],"->",alpha)

print(b,"->",beta,"| epsilon\n")

n=int(input("Enter number of transitions: \n"))

for i in range(n):

func(i)



**(LEFT FACTORING)**

def leftFactoring(s):

k=[]

l=[]

n=""

k=s.split('->')

l=k[1].split('|')

for I in range(0,len(l)-1):

for j in range(0,min(len(l[i]),len(l[i+1]))):

if(l[i][j]==l[i+1][j]):

if l[i][j] not in n:

n=n+l[i][j]

print(k[0]+'->'+n+"R")

m=k[1].split(n)

print("R->",end="")

for I in range(1,len(m)):

print(m[i],end="")

s=input("Enter the production: ") #main function

while(True):

leftFactoring(s)

print("\ndo you have another production?")

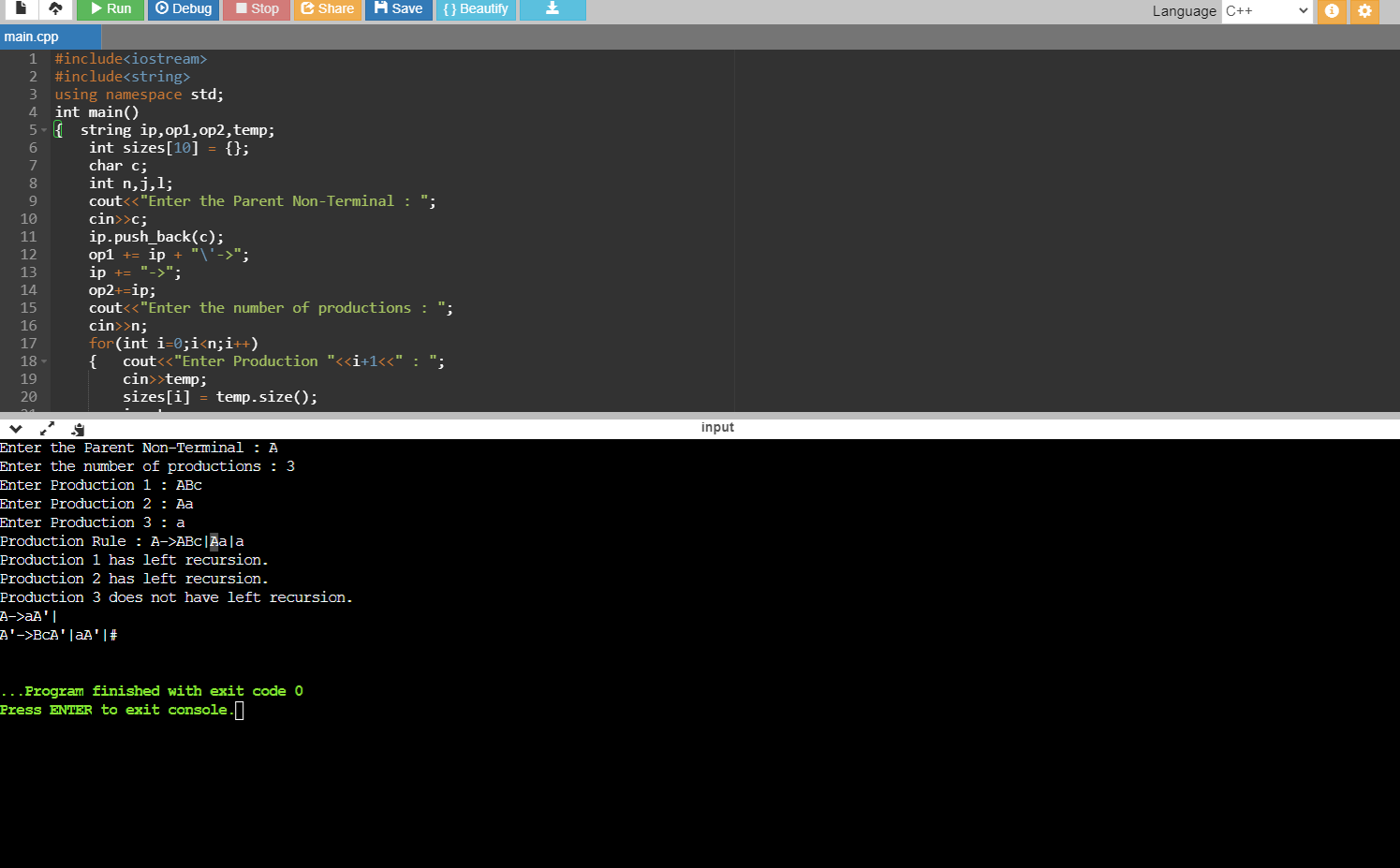
T=input("y/n:")

if T=='y':

s=input("Enter the production: ")

elif T=='n':

break



**RESULT:**

Successfully eliminated Left Recursion and Left Factoring.