REMOVING LEFT RECURSSION

def check\_left\_recursion(A,prod):

for x in prod:

if len(x)>1 and x[0] == A:

return True

return False

n=input("\ninput number of non terminals: ")

n=int(n)

NT =[]

productions = []

NewNT = []

NewProd = []

for i in range(n):

NT.append(input(f"\nenter non terminal {i+1}: "))

print("\n use @ for epsilon\n")

for x in NT:

K = input(f"\nenter productions for {x} seperated by pipe(|): ")

productions.append(K.split("|"))

for i in zip(NT,productions):

if(check\_left\_recursion(i[0],i[1])):

alpha = []

beta =[]

print(f"\nfor {i[0]}--> {i[1]} Left recursion occurs")

for prod in i[1]:

if len(prod) > 1 and i[0] == prod[0]:

alpha.append(prod[1:])

else:

beta.append(prod)

NewNT.append(i[0])

prod1 =[]

for x in beta:

if(x!='@'):# @ is epsilon

prod1.append(f"{x}{i[0]}\'")

else:

prod1.append(f"{i[0]}\'")

NewProd.append(prod1);

NewNT.append(f"{i[0]}\'")

prod2 = []

for x in alpha:

prod2.append(f"{x}{i[0]}\'")

prod2.append("@")

NewProd.append(prod2);

else:

NewNT.append(i[0])

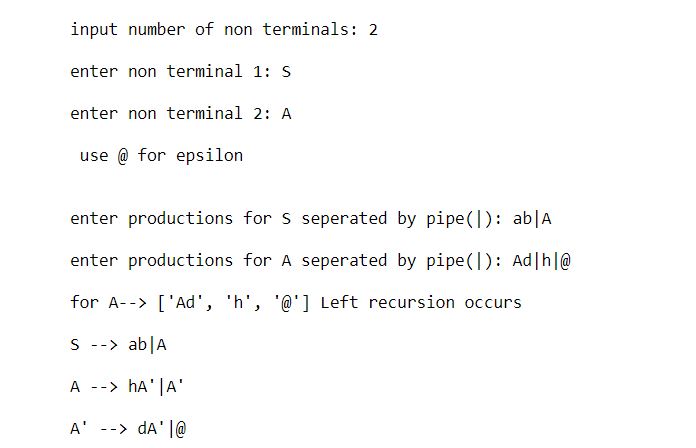
NewProd.append(i[1])

for x,p in zip(NewNT,NewProd):

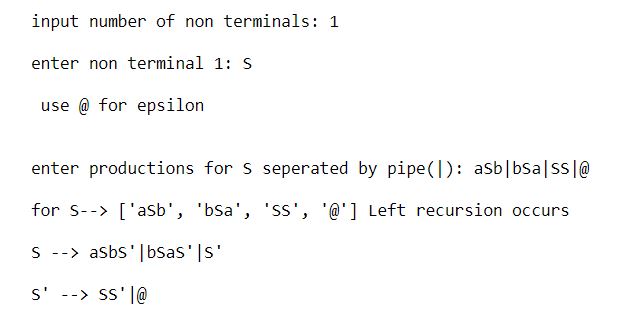
string = ""SS

**OUTPUT**

OUTPUT 1



OUTPUT 2



OUTOUT 3

