**Name: Atharva Paliwal**

**Roll No. 6B-40**

**Practical No. 4**

**Aim:** Write a program to perform string validation for LR grammar.

**Program:**

#Driver Code

Non\_Terminal,Terminal,Rules,parse\_table=list(),list(),dict(),dict()

# Input Non Terminal

print('Enter Non Terminals : ')

Non\_Terminal=list(str(i) for i in input().split())

for x in Non\_Terminal:

    Rules[x]=[]

# Input Terminal

print('Enter Terminals:')

Terminal=list(str(i) for i in input().split())+['$']

rule\_index=[]

# Input Production Rules

print('Enter rules in format A : BC|D and 0 for null : ')

for \_ in range(len(Non\_Terminal)):

    rule=input().split(':')

    for i in rule[1].split('|'):

        Rules[rule[0].strip()].append(i.strip())

        rule\_index.append((rule[0],i))

print(rule\_index)

print('Enter Parsing Table state,NT/T:action/Goto (-1 to exit): ')

while(True):

    s=input()

    if s=='-1':

        break

    s=s.split(':')

    s1=s[0].split(',')

    parse\_table[(s1[0],s1[1])]=s[1]

for i,j in parse\_table.items():

    print(i,' -> ',j)

stack='00'

string=input('Enter string to validate : ')

buffer=string+'$'

result='0'

flag=True

print('Stack\tBuffer\tAction')

while result!='Accept':

    print(stack,end='\t')

    print(buffer,end='\t')

    try:

        result=parse\_table[(stack[-2]+stack[-1],buffer[0])]

    except:

        flag=False

        break

    print(result)

    if result=='Accept':

        print('String Accepted')

    elif result[0]=='S':

        stack=stack+buffer[0]+'0'+result[1]

        buffer=buffer[1:]

    elif result[0]=='R':

        rule=rule\_index[int(result[1:])-1]

        l=3\*len(rule[1])

        stack=stack[:len(stack)-l]

        stack=stack+rule[0]

        stack=stack+parse\_table[(stack[-3]+stack[-2],stack[-1])]

if flag==False:

    print('String is not Accepted.')

**Input & Output:**







