ROLL NO:24UCS088

OPERATORS=set(['+','-','\*','/','(',')','^'])

PRIORITY={'+':1,'-':1,'\*':2,'/':2,'^':3}

def Infix\_to\_postfix(expression):

stack=[]

output=''

for ch in expression:

if ch not in OPERATORS:

output+=ch

elif ch=='(':

stack.append('(')

elif ch==')':

while stack and stack[-1]!='(':

output+=stack.pop()

stack.pop()

else:

while stack and stack[-1]!='(' and PRIORITY[ch]<=PRIORITY[stack[-1]]:

output+=stack.pop()

stack.append(ch)

while stack:

output+=stack.pop()

return output

expression=input("enter a Infix expression:")

print("Infix expression:",expression)

print("postfix expression:",Infix\_to\_postfix(expression))

OUTPUT

enter a Infix expression:a+b\*(c^d-e)^(f+g\*h)-i

Infix expression: a+b\*(c^d-e)^(f+g\*h)-i

postfix expression: abcd^e-fgh\*+^\*+i-