PROGRAM:

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 infix expression:")

print("infix expression:",expression)

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

OUTPUT:

Enter infix expression:a+(b\*c(d/e^f)\*g)\*h

infix expression: a+(b\*c(d/e^f)\*g)\*h

postfix expression: abcdef^/\*g\*h\*+