def infix\_to\_postfix(exp):

stack = [] # to store operators

postfix = "" # result string

# precedence order

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

for ch in exp:

# If operand, add to result

if ch.isalnum():

postfix += ch

# If '(', push to stack

elif ch == '(':

stack.append(ch)

# If ')', pop until '('

elif ch == ')':

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

postfix += stack.pop()

stack.pop() # remove '('

# If operator

else:

while stack and stack[-1] != '(' and prec.get(stack[-1],0) >= prec.get(ch,0):

postfix += stack.pop()

stack.append(ch)

# Pop remaining operators

while stack:

postfix += stack.pop()

return postfix

# ----------- Main ------------

exp = input("Enter infix expression: ")

print("Postfix expression:", infix\_to\_postfix(exp))

Output

Enter an infix expression:A+B\*C

Postfix Expression:ABC