**Code: -**

def evaluate\_postfix(expression):

    stack = []  # Stack to store operands

    for char in expression:

        if char.isdigit():

            stack.append(int(char))  # If the character is a number (operand), push it onto the stack

        else:  # Pop the two topmost elements from the stack for the operation

            operand2 = stack.pop()

            operand1 = stack.pop()

            if char == '+':

                result = operand1 + operand2

            elif char == '-':

                result = operand1 - operand2

            elif char == '\*':

                result = operand1 \* operand2

            elif char == '/':

                result = operand1 / operand2

            stack.append(result)  # Push the result of the operation back onto the stack

    return stack.pop()  # The final result will be the only element in the stack

# Example usage

postfix\_expression = "23\*5+"

result = evaluate\_postfix(postfix\_expression)

print(f"Postfix Expression: {postfix\_expression}")

print(f"Result: {result}")

**Output:**

