Postfix to Prefix Conversion using Stack

Introduction

Postfix notation (Reverse Polish Notation) is a mathematical notation in which the operator appears after its operands. In contrast, Prefix notation (Polish Notation) places the operator before the operands. To convert a Postfix expression to a Prefix expression, we use a stack-based approach. This document provides a detailed explanation of the C++ code, including a step-by-step dry run.

Thought Process

1. **Using a Stack:**
   - A stack is used to store operands and intermediate expressions.
   - When an operator is encountered, the top two elements from the stack are popped, and a new prefix expression is created and pushed back onto the stack.

2. **Traversing the Postfix Expression:**
   - If a digit (operand) is found, push it onto the stack.
   - If an operator is found:
     - Pop the top two elements from the stack.
     - Form a prefix expression using the operator.
     - Push the new expression back onto the stack.

3. **Final Result:**
   - After processing the entire postfix expression, only one element remains in the stack, which is the final prefix expression.

Corrected Dry Run

Example 1

Input: `126+4*8/+3-`

Step	Stack Content	Operation
1	1	Push 1
2	1, 2	Push 2
3	1, 2, 6	Push 6
4	1, +26	Operator + applied to 2 and 6
5	1, +26, 4	Push 4
6	1, +26, 4, 8	Push 8
7	1, +26, *48	Operator * applied to 4 and 8

8	1, /+26*48	Operator / applied to +26 and *48
9	1, /+26*48, 3	Push 3
10	-+1/*+26483	Operator - applied to +1 and / * +2648 3

Output: `- + 1 / * + 2 6 4 8 3`