

Evaluate Reverse Polish Notation

For this problem the task is to evaluate expressions given in postfix notation (operators come after operands).

[Key Idea:]

Use a stack:

1. Iterate through tokens.
2. If token is a number \rightarrow push onto stack.
3. If token is an operator:
 - Pop top two numbers ($\frac{1}{2}$ a).
 - Compute a operator b
 - Push result back to stack.

At the end, the stack will contain one element \rightarrow the answer.

[Important Details:]

- Division truncates toward zero (C++ default integer division handles this correctly).
- Operands are in reverse order when popped:
 - First pop = second operand (b)
 - Second pop = first operand (a)
 - Compute a op b.

[Complexity:]

- Time: $O(n)$ (one pass through tokens)
- Space: $O(n)$ (stack storage)