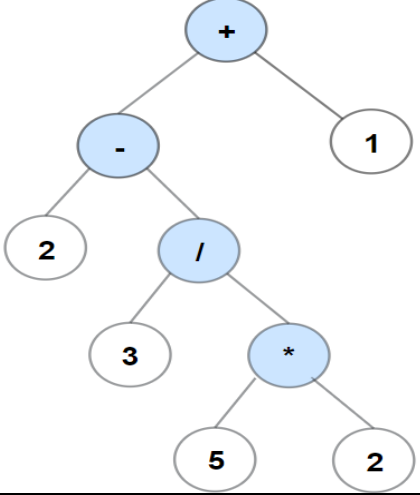
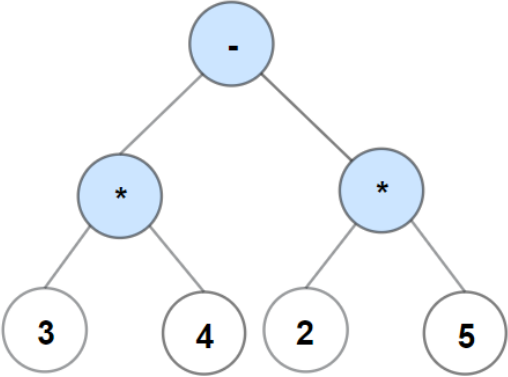
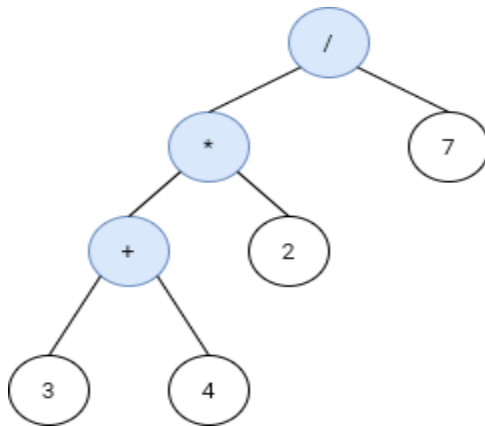


Q	Batch – S2 and S3
1	<p data-bbox="320 331 1161 369">Build Binary Expression Tree from Infix Expression</p> <p data-bbox="320 398 1393 577">A Binary Expression Tree is a kind of binary tree used to represent arithmetic expressions. Each node of a binary expression tree has either zero or two children. Leaf nodes (nodes with 0 children) correspond to operands (numbers), and internal nodes (nodes with 2 children) correspond to the operators '+' (addition), '-' (subtraction), '*' (multiplication), and '/' (division).</p> <p data-bbox="320 577 1393 689">For each internal node with operator o, the infix expression that it represents is $(A \circ B)$, where A is the expression the left subtree represents and B is the expression the right subtree represents.</p> <p data-bbox="320 723 1393 801">You are given a string s, an infix expression containing operands, the operators described above, and parentheses '(' and ')'. Return the <i>binary expression tree</i>, which its <i>in-order traversal</i> reproduce s.</p> <p data-bbox="320 907 1393 1019">Please note that order of operations applies in s. That is, expressions in parentheses are evaluated first, and multiplication and division happen before addition and subtraction.</p> <div data-bbox="320 1108 742 1646"> <p>Example 1:</p>  <pre> graph TD Plus((+)) --- Minus((-)) Plus --- One((1)) Minus --- Two1((2)) Minus --- Div(/) Div --- Three((3)) Div --- Mult(*) Mult --- Five((5)) Mult --- Two2((2)) </pre> </div> <p data-bbox="320 1653 639 1686">Input: $s = "2-3/(5*2)+1"$</p> <p data-bbox="320 1709 1107 1742">Output: $[+, -, 1, 2, /, \text{null}, \text{null}, \text{null}, \text{null}, 3, *, \text{null}, \text{null}, 5, 2]$</p>

	<p>Example 2:</p>  <p>Input: $s = "3 * 4 - 2 * 5"$ Output: $[-, *, *, 3, 4, 2, 5]$</p>
Q	Batch – S4
2	<p>Design an Expression Tree with Evaluate Function</p> <p>Given the postfix tokens of an arithmetic expression, build and return the binary expression tree that represents this expression.</p> <p>Postfix notation is a notation for writing arithmetic expressions in which the operands (numbers) appear before their operators. For example, the postfix tokens of the expression $4*(5-(7+2))$ are represented in the array <code>postfix = ["4", "5", "7", "2", "+", "-", "*"]</code>.</p>

Example 1:

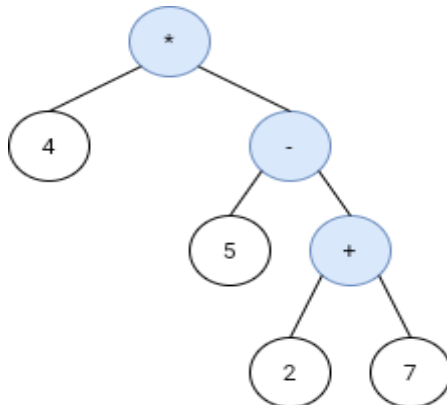


Input: $s = ["3", "4", "+", "2", "*", "7", "/"]$

Output: 2

Explanation: this expression evaluates to the above binary tree with expression $((3+4)*2)/7 = 14/7 = 2$.

Example 2:



Input: $s = ["4", "5", "2", "7", "+", "-", "*"]$

Output: -16

Explanation: this expression evaluates to the above binary tree with expression $4*(5-(2+7)) = 4*(-4) = -16$.

Q

Batch – S5

3

Given 2 binary expression trees tree1 and tree2. The leaves of a binary expression tree are variable names and the other nodes contain operators. Find out if the expressions represented by these trees are equal or not.

	<p>There are only plus signs + and letters in the tree. Input is guaranteed to be valid.</p> <hr/> <p>Example 1:</p> <p>Input:</p> <pre> tree1 + / \ a b tree2 + / \ b a </pre> <p>Output: true</p> <p>Explanation: $a + b = b + a$</p> <hr/> <p>Example 2:</p> <p>Input:</p> <pre> tree1 + / \ a + /\ c de tree2 + / \ + de /\ a c </pre> <p>Output: true</p> <p>Explanation: $a + (c + de) == (a + c) + de$</p>
--	---