Smart Coding & Interview Series
Top-20 Basic Program
(Stack Applications)

First, understand the solution building strategies and coding for the problems in LIVE/VIDEO session and then you apply those strategies discussed in LIVE/VIDEO session to solve the following problems. Use your favourite language(C/C++/Java/C#/Python/Scala) for coding.

1) Simplify Path: Given an absolute path for a file (Unix-style), simplify it. Or in other words, convert it to the canonical path. In a UNIX-style file system, a period `.` refers to the current directory. Furthermore, a double period `..` moves the directory up a level. Note that the returned canonical path must always begin with a slash `/`, and there must be only a single slash `/` between two directory names. The last directory name (if it exists) must not end with a trailing `/`. Also, the canonical path must be the shortest string representing the absolute path.
Example:
Input: `"/a/../../b/../c//.//"`
Output: `"/c"`
Source: *https://leetcode.com/problems/simplify-path/description/*

2) Validate Stack Sequences: Given two sequences `pushed` and `popped` with distinct values, return `true` if and only if this could have been the result of a sequence of push and pop operations on an initially empty stack.
Example:
Input: `pushed = [1,2,3,4,5], popped = [4,5,3,2,1]`
Output: True
Source: https://leetcode.com/problems/validate-stack-sequences/

3) Basic Calculator II: Implement a basic calculator to evaluate a simple expression string. The expression string contains only non-negative integers, `+`, `-`, `*`, `/` operators and empty spaces. The integer division should truncate toward zero.
Example:
*Input: "3+2*2"*
Output: 7
Note:

- *You may assume that the given expression is always valid.*
- ***Do not** use the `eval` built-in library function.*

Source : *https://leetcode.com/problems/basic-calculator-ii/description/*

4) Expression Evaluation: Given an expression string array, return the final result of this expression. The expression contains only integer, +, -, *, /, (,).
Example:

Input: “4-(2-3)*2+5/5”
Output: 7
Source : http://www.lintcode.com/en/problem/expression-evaluation/

5) Next Greater Element (Linear): You are given two arrays (without duplicates) nums1 and nums2 where nums1’s elements are subset of nums2. Find all the next greater numbers for nums1's elements in the corresponding places of nums2. The Next Greater Number of a number x in nums1 is the first greater number to its right in nums2. If it does not exist, output -1 for this number. Assume that all elements in nums1 and nums2 are unique.

Example:
Input: nums1 = [4,1,2], nums2 = [1,3,4,2].
Output: [-1,3,-1]
Explanation:
For number 4 in the first array, you cannot find the next greater number for it in the second array, so output -1.
For number 1 in the first array, the next greater number for it in the second array is 3.
For number 2 in the first array, there is no next greater number for it in the second array, so output -1.
Source: *https://leetcode.com/problems/next-greater-element-i/*