Stack Interview Questions Assignment

1. Evaluate Reverse Polish Notation

Problem Description:

Evaluate the value of an arithmetic expression in Reverse Polish Notation (RPN). Valid operators are +, -, *, and /. Each operand may be an integer or another expression. Note that division between two integers should truncate toward zero.

Input:

- An array of strings tokens where tokens[i] is a valid operand or operator.

Output:

- Return the value of the arithmetic expression as an integer.

Example 1:

Input: ["2", "1", "+", "3", "*"]

Output: 9

Explanation: ((2 + 1) * 3) = 9

Example 2:

Input: ["4", "13", "5", "/", "+"]

Output: 6

Explanation: (4 + (13 / 5)) = 6

Example 3:

Input: ["10", "6", "9", "3", "+", "-11", "*", "/", "*", "17", "+", "5", "+"]

Output: 22

Explanation:

```
  ((10 * (6 / ((9 + 3) * -11))) + 17) + 5

= ((10 * (6 / (12 * -11))) + 17) + 5

= ((10 * (6 / -132)) + 17) + 5

= ((10 * 0) + 17) + 5

= (0 + 17) + 5

= 22
```

Constraints:

- 1 <= tokens.length <= 10^4

- tokens[i] is either an operator: +, -, *, or /, or an integer in the range [-200, 200].

2. Min Stack

Problem Description:

Design a stack that supports push, pop, top, and retrieving the minimum element in constant time.

Implement the MinStack class:

- MinStack(): initializes the stack object.

- void push(int val): pushes the element val onto the stack.

- void pop(): removes the element on the top of the stack.

- int top(): gets the top element of the stack.

- int getMin(): retrieves the minimum element in the stack.

Example 1:

```
MinStack minStack = new MinStack();

minStack.push(-2);

minStack.push(0);

minStack.push(-3);

minStack.getMin(); // return -3

minStack.pop();

minStack.top();    // return 0

minStack.getMin(); // return -2
```

Constraints:

- -2^31 <= val <= 2^31 - 1

- Methods pop(), top(), and getMin() operations will always be called on non-empty stacks.

- At most 3 * 10^4 calls will be made to push, pop, top, and getMin.

3. Daily Temperatures

Problem Description:

Given a list of daily temperatures T, return a list such that, for each day in the input, tells you how many days you would have to wait until a warmer temperature. If there is no future day for which this is possible, put 0 instead.

Input:

- An array of integers T representing the daily temperatures.

Output:

- Return an array of integers, where the ith element is the number of days you have to wait until a warmer temperature.

If there is no future day for which this is possible, put 0 instead.

Example 1:

Input: [73, 74, 75, 71, 69, 72, 76, 73]

Output: [1, 1, 4, 2, 1, 1, 0, 0]

Constraints:

- 1 <= T.length <= 10^5

- 30 <= T[i] <= 100