

**The objective of this lab is to:**

This lab aims to enhance students' understanding of **stacks** and their applications in solving real-world problems efficiently.

**Instructions:**

- 1) Use **stacks** to solve both tasks efficiently.
- 2) Do not use built-in stack libraries
- 3) Ensure correctness by testing with multiple **input cases**.
- 4) Implement each function **efficiently** to minimize time and space complexity.

**Task 01(Remove K Digits to Get the Smallest Number)****15 Marks**

Given a non-negative integer represented as a string num, and an integer k, remove k digits from the number so that the resulting number is the smallest possible.

If the resulting number is empty, return "0".

**Example:**

**Input:** "1432219", k = 3

**Process:**

- Remove 4 → "132219"
- Remove 3 → "12219"
- Remove 2 → "1219"

**Output:** "1219"

**Constraints :** Solve the problem **O(n)** time complexity

**Function Prototype:** string removeKdigits(string num, int k)

**Task 02 (Evaluate Boolean Expression)****15 Marks**

You are given a valid boolean expression as a string. The expression follows these rules:

1. 't' → Represents **true**.
2. 'f' → Represents **false**.
3. '!(subExpr)' → Represents **logical NOT** of subExpr.
4. '&(subExpr1, subExpr2, ..., subExprn)' → Represents **logical AND** of n subexpressions (**n ≥ 1**).
5. '|(subExpr1, subExpr2, ..., subExprn)' → Represents **logical OR** of n subexpressions (**n ≥ 1**).

Return the **boolean result (true or false)** after evaluating the expression.

**Example:**

**Input:** expression = "&(t, !(f), t)"

**Output:** true

**Explanation:**

- $!(f) \rightarrow \text{true}$
- $\&(t, \text{true}, t) \rightarrow \text{AND of } (\text{true}, \text{true}, \text{true}) \rightarrow \text{true}$

**Constraints :** Solve the problem **O(n)** time complexity

**Function Prototype:** bool evaluateBooleanExpression(string expression);

**Good Luck!**

-----

**Note:** You must complete all your tasks individually. Absolutely NO collaboration is allowed. Any case of plagiarism/cheating would result in 0 marks in sessional activities.