

---

# Data Structures

---

BS (DS) Fall 2022

## Lab - 07

### Lab Objectives

Abstract Data Types

Queue implementation with Array

Queue implementation with Linked Memory

## Task 01:

### Password Encryption via Queue

Encryption scrambles your password so it's unreadable and/or unusable by hackers. That simple step protects your password while it's sitting in a server, and it offers more protection as your password zooms across the internet.

Imagine that you've created the strongest password possible. Now, imagine that all of your hard work is stored in plain text on your company's server. If a hacker gets inside, what happens next? All of your efforts go to waste, and your username and password are sold on the open market to the highest bidder.

In cryptography, encryption is the process of transforming information (referred to as plaintext) using an algorithm (called cipher) to make it unreadable to anyone except those possessing special knowledge. The result of the process is encrypted information. In many contexts, the word encryption also implicitly refers to the reverse process, decryption to make the encrypted information readable again

If your password is stored as **a2+b\*3** then it is of no use to the hacker because this is encrypted. The actual password is **ab6+\*** (decrypted)

*How it works?*

*The answer is simple, you can decrypt it via queue operations easily!*

Step 1: You start reading from the left most character (do this character by character)

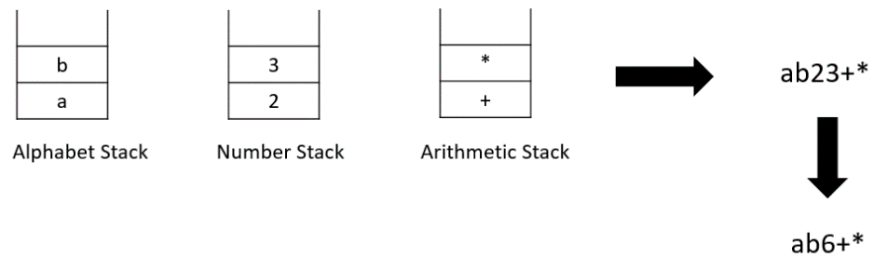
- If it is an **alphabet** put it in **alphabet queue** -> **a,b**
- If it is a **number** put it in **number queue** -> **2,3**
- If it is an **operator** put it in **arithmetic queue** -> **+,\***

Step 2: From **arithmetic queue**, find the operator with highest precedence and then apply that operation to all the numbers present in **number queue** -> **2\*3=6**

Step 3: Final decrypted password is concatenation of **alphabet queue + number queue + arithmetic queue** -> **ab6+\***

Input Password: a2+b\*3            Output Password: ab6+\*

Steps to follow:



Your task is to take an encrypted password from the user in the form of a string. The password must only contains Alphanumeric characters and Arithmetic operations +, -, \*, / (No brackets should be included).

Your decrypted password should be in the following format:

- 1) The alphabets must comes first.
- 2) Then comes the numbers.
- 3) At the end, there are arithmetic characters.

Note: If the input password contains other than alphanumeric characters and arithmetic operators, ignore them while processing.

## Task 02:

Find the smallest subsequence of distint characters using Queue operations.

**Input string = "bcbac"**

**Output string = "bca"**

### Steps

- Read the input string character by character and put each character in a queue.
- Everytime checks for the character
  - If it is already present in queue, discard the character and check for the next one
  - If the character is not present in the queue, place the character in the queue

