National University of Computer and Emerging Sciences

Lahore Campus

# Data Structures Lab

# Sessional- Exam

Date: October 17th 2024

**Course Instructor(s)**

Aqib Zeeshan

**Total Time (Hrs):**
**1 H 30M**
**Total Marks: 65**
**Total Questions:**

_____ _____ _____

Roll No Section Student Signature

**Do not write below this line**

**Attempt all the questions.**

## Q1 [Marks: 40]

You have a **browser** of one tab where you start on the homepage and you can visit another url, get back in the history number of steps or move forward in the history number of steps.
Implement the BrowserHistory class: (**Note: Use doubly linked list and iterator to implement browsing history.**)

Implement **iterator** functionalities for the same functions as well

- BrowserHistory(string homepage) Initializes the object with the homepage of the browser.
- void visit(string url) Visits url from the current page. It clears up all the forward history.
- string back(int steps) Move steps back in history. If you can only return x steps in the history and steps > x, you will return only x steps. Return the current url after moving back in history **at most** steps.
- string forward(int steps) Move steps forward in history. If you can only forward x steps in the history and steps > x, you will forward only x steps. Return the current url after forwarding in history **at most** steps
- void Delete(string url , int stepStart , int stepDelete) it delete history from step start to step delete.

- **Explanation:**
- BrowserHistory browserHistory = new BrowserHistory("mid1.com");
- browserHistory.visit("google.com"); // You are in "mid1.com". Visit "google.com"
- browserHistory.visit("facebook.com"); // You are in "google.com". Visit "facebook.com"
- browserHistory.visit("youtube.com"); // You are in "facebook.com". Visit "youtube.com"
- browserHistory.back(1); // You are in "youtube.com", move back to "facebook.com" return "facebook.com"
- browserHistory.back(1); // You are in "facebook.com", move back to "google.com" return "google.com"
- browserHistory.forward(1); // You are in "google.com", move forward to "facebook.com" return "facebook.com"
- browserHistory.visit("linkedin.com"); // You are in "facebook.com". Visit "linkedin.com"

- browserHistory.forward(2); // You are in "linkedin.com", you cannot move forward any steps.
- browserHistory.back(2); // You are in "linkedin.com", move back two steps to "facebook.com" then to "google.com". return "google.com"
- browserHistory.back(7); // You are in "google.com", you can move back only one step to "mid1.com". return "mid1.com"

**Example for Delete function:**

After visiting pages: History:
homepage.com page1.com page2.com [page3.com]
Going back 2 steps:
Current page: page1.com
History: homepage.com [page1.com] page2.com page3.com
Going forward 1 step:
Current page: page2.com
History: homepage.com page1.com [page2.com] page3.com
After visiting page4.com:
History: homepage.com page1.com [page4.com]
After deleting step 1 to 2:
History: homepage.com [page4.com]

## Q2. [Marks: 25]

You are given two integers nums1 and nums2 of lengths m and n respectively. Store the numbers in 2 array such that arraynums1 and arraynums2 represent the digits of two numbers. You are also given an integer k.

Create the maximum number of length k <= m + n from digits of the two numbers. The order of the digits from the same array must be preserved.
Return an array of the k digits representing the answer.
Example 1:
Input: nums1 = 3465, nums2 = 912583, k = 5
Output: [9,8,6,5,3]

Example 2:
Input: nums1 = 67, nums2 = 604, k = 5
Output: [6,7,6,0,4]

Example 3:
Input: nums1 = 39, nums2 = 89, k = 3
Output: [9,8,9]