Project Code: OmniBrowser Engine (CLI Version)

Duration: 60mins

Domain: System Design & Data Structures

Output: Interactive Console Application (Java/C++/Python)

1. The Scenario

You have been hired by a browser startup called **"OmniBrowser"**. Your team leads have built the core rendering engine, but the user-facing session management is missing.

Your task is to build the **Command Line Interface (CLI) Engine** that manages user input, memory, and session history. The browser must launch a shell prompt where users can interactively open tabs, navigate the web, and manage their history in real-time.

2. Functional Requirements

You must implement a central Browser class and a Tab structure/class. The system must support the following three subsystems:

A. The Tab System (The Container)

- Users can open a **New Tab**.
- Users can **Close** the current Tab.
- Users can **Switch** between tabs by ID.
- **Constraint:** The browser must be able to handle an infinite number of tabs efficiently. If a user closes a tab in the middle (e.g., Tab 2 of 5), the browser should immediately switch to the previous tab (Tab 1) or the next one if Tab 1 doesn't exist.

B. Navigation Logic (The Engine)

Each Tab acts as an independent browsing session. Inside **each** tab:

- **Visit URL:** The user types a command to visit a URL.
- **Back:** Go back to the previously visited page.
- **Forward:** If the user went "Back", they can go "Forward" again to the page they just left.
- **The "New Path" Rule:** If a user goes "Back" to Page A, and then visits a *new* Page C, the "Forward" history is deleted. (You cannot go forward to Page B anymore).

C. Global History (The Log)

- The browser must maintain a list of the **last 10 websites visited** across ALL tabs.

- This list is chronological (most recent first).
- Once the list hits 10 URLs, the oldest one is automatically removed to make space for the new one.

3. Technical Architecture (Hints)

To succeed within the 2-hour limit, you must select the correct Data Structure for each subsystem:

1. **Tabs:** How do you store a list where items are frequently inserted and deleted from the middle?
2. **Navigation:** How do you implement Last-In-First-Out (LIFO) logic for the Back/Forward buttons?
3. **Global History:** How do you store a fixed-size list where new items push out old items (FIFO)?

4. CLI Specification & Commands

Your program must run inside an infinite loop (e.g., while(true)), displaying a prompt (e.g., OmniBrowser >) and waiting for user commands.

Required Commands

Command	Arguments	Description
NEW	[url]	Opens a new tab and visits the URL.
VISIT	[url]	Visits a URL in the *current* tab.
BACK	None	Moves the current tab back one step.
FWD	None	Moves the current tab forward one step.
TAB	[index]	Switches control to the tab at the specific index (1-based).
CLOSE	None	Closes the current tab.
HISTORY	None	Prints the global history (last 10).

EXIT	None	Terminates the program.

Sample Interactive Session

(User input is marked with >>)

Welcome to OmniBrowser v1.0
Type 'EXIT' to quit.

OmniBrowser > NEW www.google.com
[Success] Tab 1 created. Loading www.google.com...

OmniBrowser > VISIT www.facebook.com
[Nav] Tab 1: www.google.com ->
www.facebook.com

OmniBrowser > NEW www.youtube.com
[Success] Tab 2 created. Loading www.youtube.com...

OmniBrowser > TAB 1
[Switch] Now in Tab 1. Current: www.facebook.com

OmniBrowser > BACK
[Back] Tab 1: www.facebook.com ->
www.google.com

OmniBrowser > VISIT www.twitter.com
[Nav] Tab 1: www.google.com ->
www.twitter.com
(Note: Forward history for Tab 1 is now cleared)

OmniBrowser > HISTORY
Global History:
1. www.twitter.com
2. www.google.com (via Back)
3. www.youtube.com
4. www.facebook.com
5. www.google.com

OmniBrowser > EXIT
Closing OmniBrowser... Goodbye.

5. Deliverables

1. **Source Code:** Clean, modular code using Classes.
2. **Interactive Loop:** A main function that implements the CLI loop, parsing strings to identify commands and arguments.
3. **Edge Case Handling:**
    - Pressing BACK when there is no history should print "No history to go back to."
    - Switching to an invalid TAB ID should print "Invalid Tab ID."