**Intern Exercise: Browser History Using ArrayDeque**

**Problem Statement:**

You are tasked with simulating a **browser history system**:

* Users visit different websites.
* The browser maintains **recent history**.
* Users can:
  + Go **back** to the previous website.
  + Go **forward** if they navigated back.
* Use **ArrayDeque** to efficiently handle **back and forward operations**.

**Requirements:**

1. **ArrayDeque for Back History:**
   * Use one ArrayDeque<String> to store visited websites.
   * New visits are added to the **end** of the deque.
   * pop() / removeLast() simulates **back button**.
2. **ArrayDeque for Forward History:**
   * Optional: Use a second deque to store websites for **forward navigation**.
3. **Input from User:**
   * Ask user to **visit website**, **go back**, or **exit**.
4. **Process Operations:**
   * Display **current website** after each operation.
   * Maintain correct **history behavior** like a browser.

**Full Solution Code:**

import java.util.\*;

public class BrowserHistory {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

ArrayDeque<String> backHistory = new ArrayDeque<>();

ArrayDeque<String> forwardHistory = new ArrayDeque<>();

String current = null;

boolean running = true;

while (running) {

System.out.println("\nOptions: 1=Visit 2=Back 3=Forward 4=Exit");

System.out.print("Choose option: ");

int choice = sc.nextInt();

sc.nextLine(); // consume newline

switch (choice) {

case 1: // Visit website

System.out.print("Enter website URL: ");

String site = sc.nextLine();

if (current != null) backHistory.addLast(current); // add current to back

current = site;

forwardHistory.clear(); // clear forward history

System.out.println("Visited: " + current);

break;

case 2: // Back

if (!backHistory.isEmpty()) {

forwardHistory.addFirst(current); // save current for forward

current = backHistory.removeLast();

System.out.println("Back to: " + current);

} else {

System.out.println("No history to go back.");

}

break;

case 3: // Forward

if (!forwardHistory.isEmpty()) {

backHistory.addLast(current); // save current to back

current = forwardHistory.removeFirst();

System.out.println("Forward to: " + current);

} else {

System.out.println("No forward history.");

}

break;

case 4: // Exit

running = false;

System.out.println("Exiting browser simulation.");

break;

default:

System.out.println("Invalid choice.");

}

}

sc.close();

}

}

**Sample Run:**

Options: 1=Visit 2=Back 3=Forward 4=Exit

Choose option: 1

Enter website URL: google.com

Visited: google.com

Choose option: 1

Enter website URL: youtube.com

Visited: youtube.com

Choose option: 2

Back to: google.com

Choose option: 3

Forward to: youtube.com

**✅ Key Learning Points for Interns:**

1. **ArrayDeque** allows adding/removing from **both ends efficiently**.
2. **Simulates real-world browser history** (back and forward).
3. **No sorting**, purely **insertion and removal order**.
4. Helps understand **stack behavior (back)** and **queue behavior (forward)** in one example.