Homework 3: Web Browser Simulator

OVERVIEW

In this assignment, you will simulate the core functionalities of a web browser using **stacks** and a **queue**. Your program will handle page navigation (the back and forward buttons) and manage a simple download queue. This assignment will test your understanding of LIFO (Last-In, First-Out) and FIFO (First-In, First-Out) data structures. You will need to implement two stacks for navigation history and one queue for downloads.

INPUT FORMAT

1. Input File (input.txt)
The input file will contain a single line with the starting URL for the browser session. This will be considered the "homepage".
Example input.txt:

https://www.google.com

2. Command File (command.txt)
The command file contains a sequence of actions the user performs.

- VISIT(url)
  Navigates to a new URL. The page you were on before this command is pushed onto the back stack. Any pages in the forward stack are cleared. The new URL becomes the current page. The forward stack starts empty and is cleared every time a new page is visited.
- BACK
  Navigates to the previous page. The current page is pushed onto the forward stack, and the new page is popped from the top of the back stack. If the back stack is empty, this command does nothing.
- FORWARD
  Navigates to the next page. The current page is pushed onto the back stack, and the new page is popped from the top of the forward stack. If the forward stack is empty, this command does nothing.
- DOWNLOAD(filename)
  Adds a file to the download queue.
- PROCESS_DOWNLOAD
  "Completes" the download at the front of the queue by removing it.
- PRINT(target)
  Prints the contents of a specific part of the browser's state to the output file. The target can be one of the following:
  - current: Prints the current URL.
  - back: Prints the contents of the back stack, from top to bottom.

- forward: Prints the contents of the forward stack, from top to bottom.
- downloads: Prints the contents of the download queue, from front to back.

EXAMPLE WALKTHROUGH

Let's trace a session to see how the data structures change.

input.txt

https://a.com

command.txt

VISIT(https://b.com)
VISIT(https://c.com)
BACK
DOWNLOAD(file1.zip)
FORWARD
PRINT(current)
PRINT(back)
PRINT(forward)
PRINT(downloads)

Step-by-Step Execution:

1. **Initial State:**
   - Current Page: https://a.com
   - Back Stack: [empty]
   - Forward Stack: [empty]
   - Download Queue: [empty]
2. **VISIT(https://b.com):**
   - https://a.com is pushed to the back stack.
   - Current Page becomes https://b.com.
   - Back Stack: [https://a.com]
3. **VISIT(https://c.com):**
   - https://b.com is pushed to the back stack.
   - Current Page becomes https://c.com.
   - Back Stack: [https://b.com, https://a.com]
4. **BACK:**
   - https://c.com is pushed to the forward stack.
   - https://b.com is popped from the back stack and becomes the Current Page.
   - Current Page: https://b.com
   - Back Stack: [https://a.com]

    - Forward Stack: [https://c.com]
5. **DOWNLOAD(file1.zip):**
    - file1.zip is added to the download queue.
    - Download Queue: [file1.zip]
6. **FORWARD:**
    - https://b.com is pushed to the back stack.
    - https://c.com is popped from the forward stack and becomes the Current Page.
    - Current Page: https://c.com
    - Back Stack: [https://b.com, https://a.com]
    - Forward Stack: [empty]
7. **PRINT Commands:** The program now prints the final state.

Final output.txt:

Current Page: https://c.com
Back Stack:
https://b.com
https://a.com
Forward Stack:
[empty]
Download Queue:
file1.zip

OUTPUT FORMAT

The output file should only contain the results from PRINT commands. Each printout should start with a descriptive header, and if a stack or queue is empty, it should print [empty] on the line below the header.

RESOURCES & SUBMISSION

Please refer to the course website for detailed instructions on how to compile, test, and submit your homework on the server.

- **General Homework Information & Submission:** https://uh.edu/nouhadrizk/about/courses/programming-and-data-structures/homework/
- **Testing Your Program:** https://uh.edu/nouhadrizk/about/courses/programming-and-data-structures/homework/homework-introduction/

You must submit your C++ source files (.cpp and .h) to a folder named **hw3** (case-sensitive) in your root directory on the server.

ACADEMIC INTEGRITY

This is an individual assignment. All work submitted must be your own. Your submission will be automatically checked for plagiarism. Any detected instances of copying or cheating will result in a grade of **0** and potential further disciplinary action.