CL2001 – Data Structures Lab

Lab Task # 06

Note:

- Copied task will be awarded **zero** marks.
- Use comments wherever applicable.
- Submit a pdf file containing all your C++ code with all possible screenshots of every task output on Google Classroom. The name of file should be your roll no followed by your name (roll-no-name.pdf) i.e., (23P-1234-Ali.pdf).
- Variables and functions names should be meaningful.

Problem: 1 | Stack for Web Browsing

You're tasked with developing a C++ program to model a browser's back and forward navigation functionality using stacks. The program should allow users to navigate through a series of web pages they've visited and simulate the behavior of the back and forward buttons in a typical web browser. Design the program to implement the following features:

1. **Stacks for Navigation:** Utilize two stacks—one for the back navigation history and one for the forward navigation history—to track the sequence of visited web pages.
2. **Navigation Operations:** Implement functions to:
    - Push a newly visited web page onto the back navigation stack.
    - Pop a web page from the back navigation stack when the user clicks the back button.
    - Push a web page onto the forward navigation stack when the user clicks the back button.
    - Pop a web page from the forward navigation stack when the user clicks the forward button.
    - Push a web page from the back navigation stack when the user clicks the forward button.
3. **Display Current Page:** Display the URL of the current web page being viewed.
4. **Error Handling:** Implement error handling for cases where the user attempts to navigate back or forward when there are no more pages in the respective stack.

Additionally, demonstrate the functionality of your program by simulating a user's navigation through multiple web pages and showing the sequence of back and forward button clicks along with the displayed URLs.

Problem: 2 | Valid Parentheses

Given a string s containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

An input string is valid if:

1. Open brackets must be closed by the same type of brackets.

2. Open brackets must be closed in the correct order.

3. Every close bracket has a corresponding open bracket of the same type.

Example 1:

Input: s = "()"

Output: true

Example 2:

Input: s = "()[]{}"

Output: true

Example 3:

Input: s = "(]"

Output: false

Example 4:

Input: s = "({ }]"

Output: false