Assignment # 02

Data Structures and Algorithm

Huzaifa Rizwan
10-25-2024

Q1. Suppose you are developing a browsing history feature like the "Back" button in browsers. In this scenario, you will employ a Stack data structure, which can be thought of as a stack of items, to keep track of the websites visited during a browsing session. Each item (or stack element) will store essential information about a visited webpage, including its URL and page title.

You can add the following functions: [CLO:4, SO:2,3,4] (10 Marks)

- Add New Visited URL: Implement a function to push a new visited URL onto the stack. Each entry should include the URL and the page title.
- Clear Browsing History: Implement a function to clear all entries from the browsing history stack.
- Display Browsing History: Implement a function to display the visited pages from the most recently visited to the oldest.

CODE :

```
#include <iostream>
#include <stack>
#include <string>

using namespace std;

class BrowserHistory { private:
   stack<string> backHistory;
```

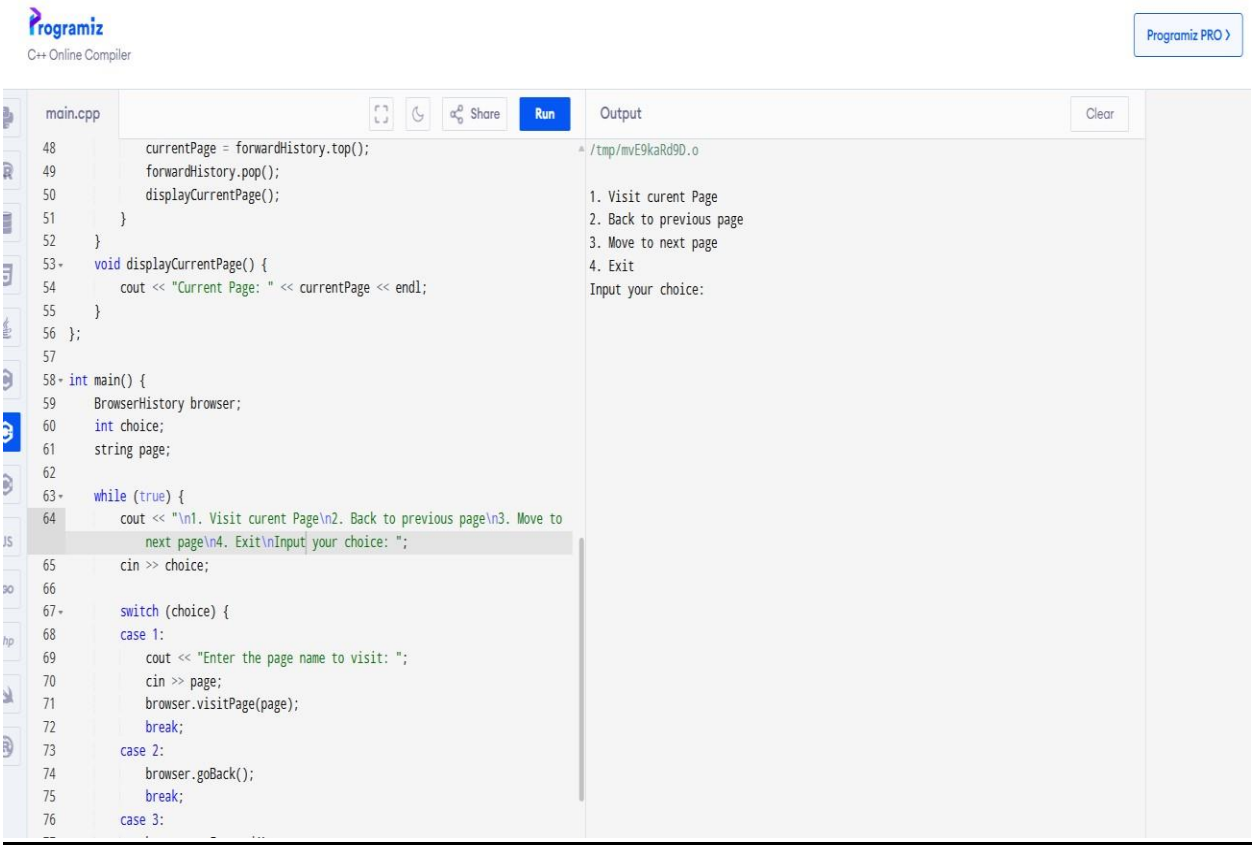
```
    stack<string> forwardHistory;    string
currentPage;

public:    BrowserHistory() {
currentPage = "Home";
    }


    void visitPage(const string &page) {
if (!currentPage.empty()) {
backHistory.push(currentPage);
        }
        currentPage = page;
while (!forwardHistory.empty()) {
forwardHistory.pop();
        }
        displayCurrentPage();
    }


    void goBack() {
        if (backHistory.empty()) {          cout << "No
pages to go back to!" << endl;
```

```
        } else {
            forwardHistory.push(currentPage);
currentPage = backHistory.top();
backHistory.pop();
displayCurrentPage();
        }
    }


    void goForward() {     if (forwardHistory.empty()) {
cout << "No pages to go forward to!" << endl;
        } else {
            backHistory.push(currentPage);
currentPage = forwardHistory.top();
forwardHistory.pop();
displayCurrentPage();
        }
    }
    void displayCurrentPage() {
        cout << "Current Page: " << currentPage << endl;
    }
};
```

```
int main() {
    BrowserHistory browser;
int choice;    string page;

    while (true) {        cout << "\n1. Visit current Page\n2. Back to
previous page
\n3. Next page \n4. Exit\n Input Your Choice  : ";
cin >> choice;

       switch (choice) {
case 1:
           cout << "Enter the page name to visit: ";
cin >> page;           browser.visitPage(page);
           break;
case 2:
browser.goBack();
break;       case 3:
           browser.goForward();
           break;
case 4:
```

```
            cout << "Exiting the browser." << endl;
return 0;       default:
            cout << "Invalid choice! Please try again." << endl;
        }
    }

    return 0;
}
```

Output :




```
            currentPage = forwardHistory.top();
            forwardHistory.pop();
            displayCurrentPage();
        }
    }
    void displayCurrentPage() {
        cout << "Current Page: " << currentPage << endl;
    }
};

int main() {
    BrowserHistory browser;
    int choice;
    string page;

    while (true) {
        cout << "\n1. Visit curent Page\n2. Back to previous page\n3. Move to
            next page\n4. Exit\nInput your choice: ";
        cin >> choice;

        switch (choice) {
        case 1:
            cout << "Enter the page name to visit: ";
            cin >> page;
            browser.visitPage(page);
            break;
        case 2:
            browser.goBack();
            break;
        case 3:
```