PIMPRI CHINCHWAD EDUCATION TRUST's.

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

(An Autonomous Institute)

Class : SY BTech **Acad. Yr. 2025-26** **Semester : I**

Name of the student: Varad Amol Pisale **PRN : 124B1B043**

Department: Computer Engineering **Division : A**

Course Name : Data Structures Laboratory **Code:BCE23PC02**

Completion Date : 29/10/2025

October Challenge

Problem Statement:
Problem Statement: "Browser History Management"
You are tasked with implementing a simplified browser history management system. This system should allow users to navigate between previously visited pages, and also add new pages to their history. The system needs to efficiently handle both "back" and "forward" navigation, as well as adding new pages, while maintaining a limited history size.
Requirements:
When a user visits a new page, it should be added to the current history. If there are any "forward" pages available (i.e., pages visited after the current page in the history), they should be cleared, as visiting a new page effectively creates a new branch in the history.
Users should be able to navigate to the previous page in their history. If there's no previous page, they remain on the current page.
Users should be able to navigate to a page they previously went "back" from. If there's no page to go forward to, they remain on the current page.
The browser history should have a maximum capacity. If adding a new page exceeds this capacity, the oldest page in the history should be automatically removed to make space.
Input:
A sequence of operations:
visit(url): Adds url to the history.
back(steps): Navigates back by steps pages.
forward(steps): Navigates forward by steps pages.
Output:
For each visit, back, or forward operation, output the URL of the page the user is currently on after the operation.

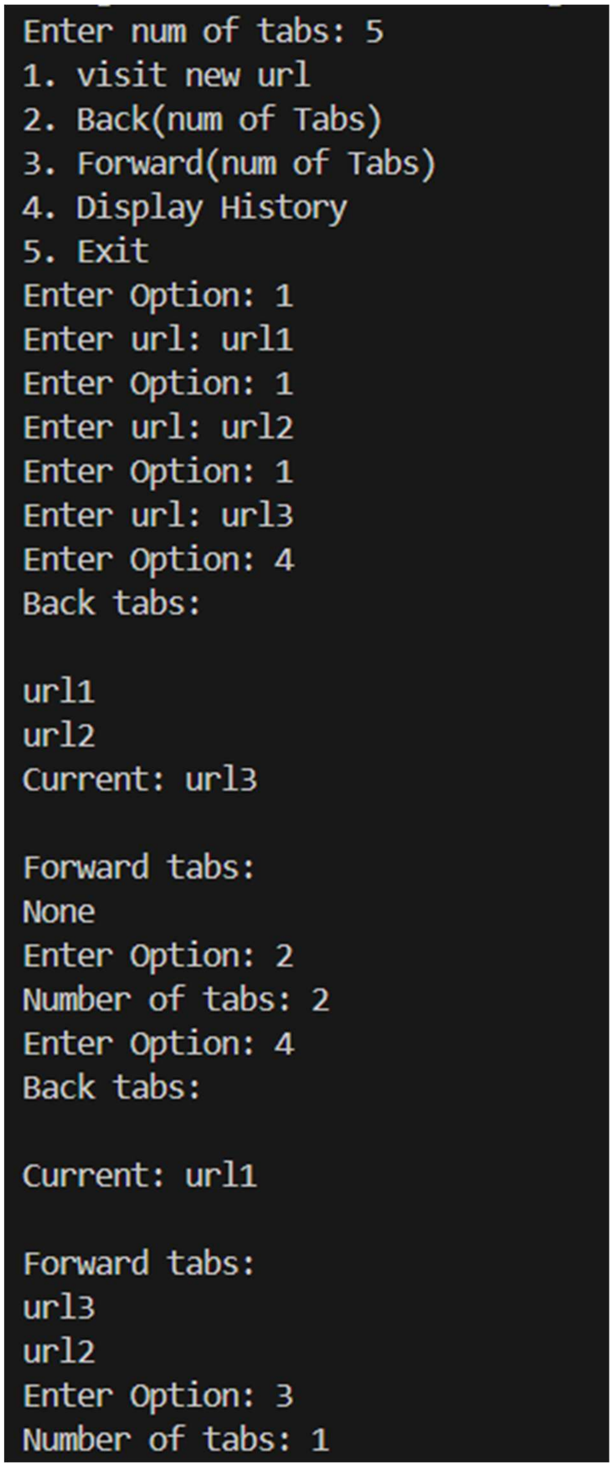
Source Code :

```
#include <bits/stdc++.h>
using namespace std;

class Browser_history
{
   deque<string> back, forward;
   string current;
   int curr_size = 0;

public:
   int size;

   void visit_new(string url)
   {
      curr_size++;
      if (curr_size > size)
      {
         back.pop_front();
         curr_size--;
      }
      back.push_back(current);
      current = url;
      forward.clear();
   }

   void move_back(int steps)
   {
      int s = 1;
      while (!back.empty() && s <= steps)
      {
         forward.push_back(current);
         string temp = back.back();
         back.pop_back();
         current = temp;
         s++;
      }
   }

   void move_forward(int steps)
   {
      int s = 1;
      while (!forward.empty() && s <= steps)
```

```
            {
                back.push_back(current);
                string temp = forward.back();
                forward.pop_back();
                current = temp;
                s++;
            }
        }

        void display_history()
        {
            cout << "Back tabs: " << endl;
            if(back.empty()) cout<<"None"<<endl;
            for (auto it = back.begin(); it != back.end(); ++it)
            {
                cout << *it << endl;
            }
            cout << "Current: " << current << endl<<endl;
            cout << "Forward tabs: " << endl;
            if(forward.empty()) cout<<"None"<<endl;
            for (auto it = forward.begin(); it != forward.end(); ++it)
            {
                cout << *it << endl;
            }
        }
    };

    int main()
    {
        Browser_history sys;
        cout << "Enter num of tabs: ";
        cin >> sys.size;
        getchar();
        cout << "1. visit new url " << endl;
        cout << "2. Back(num of Tabs) " << endl;
        cout << "3. Forward(num of Tabs) " << endl;
        cout << "4. Display History" << endl;
        cout << "5. Exit" << endl;
        while (true)
        {
            int option;
            cout << "Enter Option: ";
            cin >> option;
            getchar();
            switch (option)
            {
```

```
            case 1:
            {
                string url;
                cout << "Enter url: ";
                getline(cin, url);
                sys.visit_new(url);
                break;
            }
            case 2:
            {
                int st;
                cout << "Number of tabs: ";
                cin >> st;
                sys.move_back(st);
                break;
            }
            case 3:
            {
                int st;
                cout << "Number of tabs: ";
                cin >> st;
                sys.move_forward(st);
                break;
            }
            case 4:
            {
                sys.display_history();
                break;
            }
            case 5:
            {
                cout << "Exiting " << endl
                     << "THANK_YOU";
                return 0;
            }

            default:
                cout << "Enter valid option" << endl;
                break;
            }
        }

        return 0;
    }
```

Screen Shot of Output :



```
Enter num of tabs: 5
1. visit new url
2. Back(num of Tabs)
3. Forward(num of Tabs)
4. Display History
5. Exit
Enter Option: 1
Enter url: url1
Enter Option: 1
Enter url: url2
Enter Option: 1
Enter url: url3
Enter Option: 4
Back tabs:

url1
url2
Current: url3

Forward tabs:
None
Enter Option: 2
Number of tabs: 2
Enter Option: 4
Back tabs:

Current: url1

Forward tabs:
url3
url2
Enter Option: 3
Number of tabs: 1
```

Conclusion: Hence we have implemented a Browser Management System