Browser Tab Navigation System - C Programming Project

Name: Mayur Krishna D K

USN: 24UG00521

Problem Statement

Design and simulate a browser tab navigation system using C programming, based on the concepts of data structures. For each page of the browser, store a unique pageID (integer) and URL (string).

Functionalities to implement:

1. Visit a new page

2. Go back

3. Go forward

4. Show current tab

5.Close current tab

6. Show history

7. Exit

Approach

The project is done using Doubly ;Linked List data structure. Each memory location represents a browser tab with fields for pageID, URL, and pointers to the next and previous tabs. This method is used for effective usage of going to next webpage or previous webpage using doubly linked list concept.

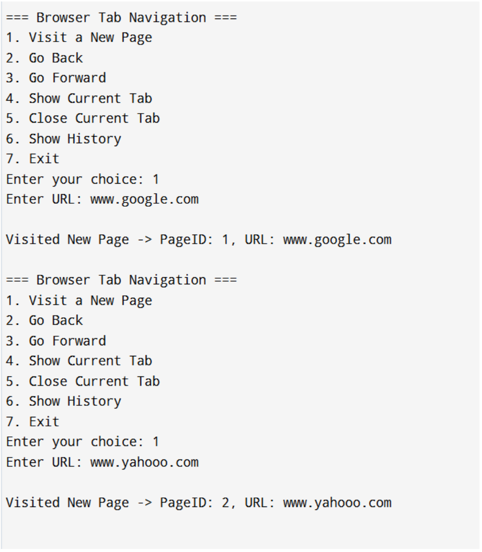
C Program Code

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct Tab {

   int pageID;

   char url[100];
```

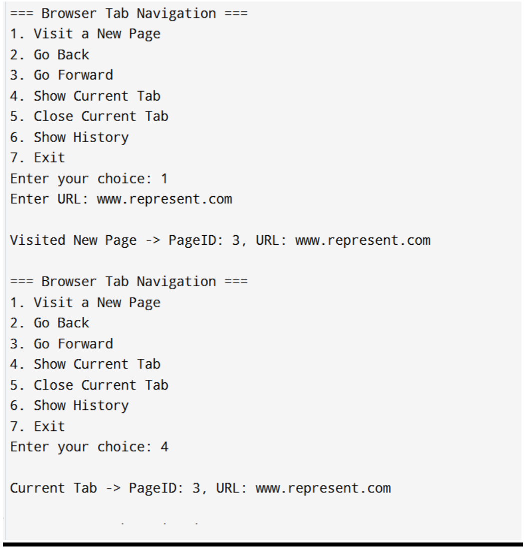
```
    struct Tab *next;
    struct Tab *prev;
} Tab;

Tab *head = NULL, *tail = NULL, *current = NULL;
int pageCounter = 1;
Tab* createTab(char *url) {
    Tab *newTab = (Tab*)malloc(sizeof(Tab));
    newTab->pageID = pageCounter++;
    strcpy(newTab->url, url);
    newTab->next = NULL;
    newTab->prev = NULL;
    return newTab;
}
void visitNewPage(char *url) {
    Tab *newTab = createTab(url);
    if (head == NULL) {
        head = tail = current = newTab;
    } else {
        tail->next = newTab;
        newTab->prev = tail;
        tail = newTab;
        current = newTab;
    }
    printf("\nVisited New Page -> PageID: %d, URL: %s\n", current->pageID, current->url);
}
void goForward() {
    if (current != NULL && current->next != NULL) {
        current = current->next;
        printf("\nMoved Forward -> PageID: %d, URL: %s\n", current->pageID, current->url);
    } else {
        printf("\nNo forward tab exists!\n");
    }
}
void goBack() {
    if (current != NULL && current->prev != NULL) {
        current = current->prev;
        printf("\nMoved Back -> PageID: %d, URL: %s\n", current->pageID, current->url);
    } else {
        printf("\nNo back tab exists!\n");
    }
}
void showCurrent() {
    if (current != NULL) {
        printf("\nCurrent Tab -> PageID: %d, URL: %s\n", current->pageID, current->url);
    } else {
        printf("\nNo tab is currently open!\n");
    }
}
void closeCurrent() {
    if (current == NULL) {
        printf("\nNo tab to close!\n");
        return;
    }
    printf("\nClosing Tab -> PageID: %d, URL: %s\n", current->pageID, current->url);
    if (current->prev != NULL) current->prev->next = current->next;
    else head = current->next;

    if (current->next != NULL) current->next->prev = current->prev;
    else tail = current->prev;

    Tab *temp = current;
    if (current->next != NULL) current = current->next;
    else current = current->prev;

    free(temp);
}
void showHistory() {
    if (head == NULL) {
        printf("\nNo history available!\n");
        return;
    }
    printf("\nBrowser History:\n");
    Tab *temp = head;
    while (temp != NULL) {
        printf("PageID: %d, URL: %s\n", temp->pageID, temp->url);
        temp = temp->next;
    }
}
int main() {
    int choice;
    char url[100];
```

```
while (1) {
    printf("\n=== Browser Tab Navigation ===\n");
    printf("1. Visit a New Page\n");
    printf("2. Go Back\n");
    printf("3. Go Forward\n");
    printf("4. Show Current Tab\n");
    printf("5. Close Current Tab\n");
    printf("6. Show History\n");
    printf("7. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);
    getchar(); // clear input buffer

    switch (choice) {
        case 1:
            printf("Enter URL: ");
            fgets(url, sizeof(url), stdin);
            url[strcspn(url, "\n")] = 0;
            visitNewPage(url);
            break;
        case 2:
            goBack();
            break;
        case 3:
            goForward();
            break;
        case 4:
            showCurrent();
            break;
            case 5:
                closeCurrent();
                break;
            case 6:
                showHistory();
                break;
            case 7:
                printf("\nExiting Browser...\n");
                exit(0);
            default:
                printf("\nInvalid choice! Try again.\n");
        }
    }
    return 0;
}
```

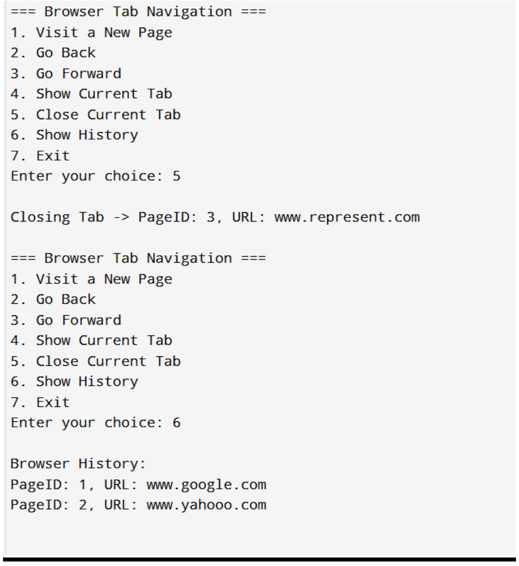
Output Screenshots



```
=== Browser Tab Navigation ===
1. Visit a New Page
2. Go Back
3. Go Forward
4. Show Current Tab
5. Close Current Tab
6. Show History
7. Exit
Enter your choice: 1
Enter URL: www.google.com

Visited New Page -> PageID: 1, URL: www.google.com

=== Browser Tab Navigation ===
1. Visit a New Page
2. Go Back
3. Go Forward
4. Show Current Tab
5. Close Current Tab
6. Show History
7. Exit
Enter your choice: 1
Enter URL: www.yahooo.com

Visited New Page -> PageID: 2, URL: www.yahooo.com
```



```
=== Browser Tab Navigation ===
1. Visit a New Page
2. Go Back
3. Go Forward
4. Show Current Tab
5. Close Current Tab
6. Show History
7. Exit
Enter your choice: 1
Enter URL: www.represent.com

Visited New Page -> PageID: 3, URL: www.represent.com

=== Browser Tab Navigation ===
1. Visit a New Page
2. Go Back
3. Go Forward
4. Show Current Tab
5. Close Current Tab
6. Show History
7. Exit
Enter your choice: 4

Current Tab -> PageID: 3, URL: www.represent.com
```



```
=== Browser Tab Navigation ===
1. Visit a New Page
2. Go Back
3. Go Forward
4. Show Current Tab
5. Close Current Tab
6. Show History
7. Exit
Enter your choice: 5

Closing Tab -> PageID: 3, URL: www.represent.com

=== Browser Tab Navigation ===
1. Visit a New Page
2. Go Back
3. Go Forward
4. Show Current Tab
5. Close Current Tab
6. Show History
7. Exit
Enter your choice: 6

Browser History:
PageID: 1, URL: www.google.com
PageID: 2, URL: www.yahooo.com
```

Conclusion

This project demonstrates how data structures, particularly the Doubly Linked List, can be applied to simulate browser tab navigation. The program successfully implements all required features including visiting new tabs, back/forward navigation, showing the current tab, closing tabs, and displaying browsing history.