Lab Practice Week 3

Name Heng Sovannreach

Exercise 1

```
57      void deleteHead(){
58          if(head == nullptr){
59              cout << "List is empty"<< endl;
60              return;
61          }
62          if(head->next == head){
63              delete head;
64              head = nullptr;
65              return;
66          }
67          node *t = head;
68          while (t->next != head){
69              t = t->next;
70          }
71          node* toDelete = head;
72          head = head->next;
73          t->next = head;
74          delete toDelete;
75
76      }
77  };
78  int main () {
79      CircularLinkedList s1 ;
80      s1.add(23);
81      s1.AddBegin(33);
82      s1.display();
83      s1.deleteHead();
84      s1.display();
85      return 0 ;
86  }
```

Output :

```
PS C:\Users\USER\OneDrive\CADT_YEAR_2\term-1\Structure\Data-Structure-and-Algorithms>  & '
 '--stdin=Microsoft-MIEngine-In-auhga1qn.dfb' '--stdout=Microsoft-MIEngine-Out-nm1d321j.0u
s64\ucrt64\bin\gdb.exe' '--interpreter=mi'
List diplay : 33 --> 23 --> back to head
NULL
List diplay : 23 --> back to head
NULL
PS C:\Users\USER\OneDrive\CADT_YEAR_2\term-1\Structure\Data-Structure-and-Algorithms>
```

Exercise 2 :

```
    bool search(int key) {
        if (head == nullptr){
            cout<<"List is empty"<<endl;
            return false;
        }
        node* t= head;
        do {
            if (t->data == key) {
                cout<<"Node with value "<< key <<" exist in the list" <<endl;
                return true;
            }
            t = t->next;
        } while (t != head);
        cout<<"Node with value "<< key <<" not found in the list" <<endl;
        return false;
    }
};
int main () {
    CircularLinkedList s1 ;
    s1.AddBegin(23);
    s1.AddBegin(67);
    s1.AddBegin(404);
    s1.display();
    s1.search(67);
    s1.search(99);
    return 0 ;
}
```

Output :

```
PS C:\Users\USER\OneDrive\CADT_YEAR_2\term-1\Structure>  & 'c:\Users\USER\.vs
\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-nrurx5jq.k2m' '--st
e-Error-nd5jryb4.jxj' '--pid=Microsoft-MIEngine-Pid-dqtzv4pf.wla' '--dbgExe=C
List diplay : 404 --> 67 --> 23 --> back to head
NULL
Node with value 67 exist in the list
Node with value 99 not found in the list
PS C:\Users\USER\OneDrive\CADT_YEAR_2\term-1\Structure>
```

Exercise 3 :

```
    void insertBegin (string s){
        node *Addnew = new node (s);
        if (head == nullptr) {
            head = Addnew;
            tail = Addnew;
        } else {
            head ->previous = Addnew;
            Addnew->next = head;
            head = Addnew ;

        }
    }

};
int main () {
    DoublyLinkedLis s1 ;
    s1.insertBegin("Facebook.com");
    s1.display();
    s1.insertBegin("Youtube.com");
    s1.display();
    s1.insertBegin("Tiktok.com");
    s1.display();
    s1.insertBegin("Github.com");
    s1.display();
    return 0;

}
```

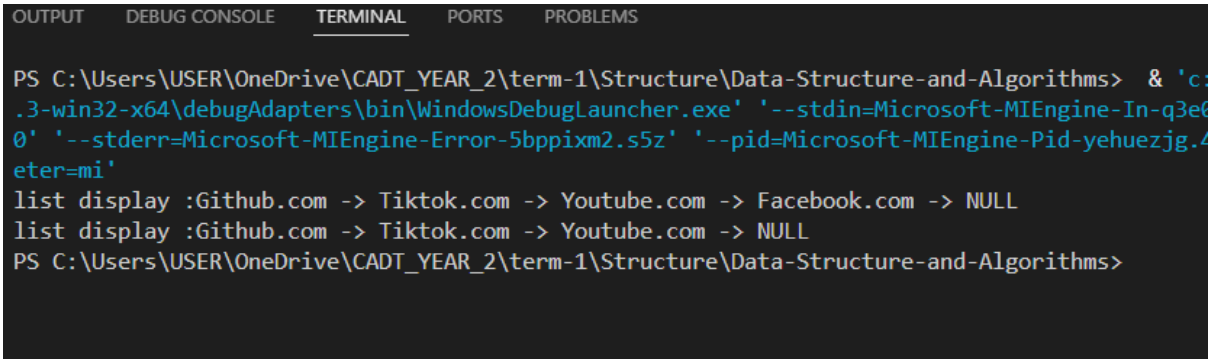
Output :

```
OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   PROBLEMS

PS C:\Users\USER\OneDrive\CADT_YEAR_2\term-1\Structure\Data-Structure-and-Algorithms>  & 'c:\
.3-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-44pwt
o' '--stderr=Microsoft-MIEngine-Error-5lppkx2d.tuw' '--pid=Microsoft-MIEngine-Pid-m5aba2vs.4c
eter=mi'
list display :Facebook.com -> NULL
list display :Youtube.com -> Facebook.com -> NULL
list display :Tiktok.com -> Youtube.com -> Facebook.com -> NULL
list display :Github.com -> Tiktok.com -> Youtube.com -> Facebook.com -> NULL
PS C:\Users\USER\OneDrive\CADT_YEAR_2\term-1\Structure\Data-Structure-and-Algorithms>
```

Exercise4 :

```
    void deletelast(){
        if (head == nullptr) {
            cout << "List is empty"<<endl;
            return;
        }
        if (head->next == nullptr) {
            delete head;
            head = nullptr;
            tail = nullptr;
            return;
        }
        node* temp = head;
        while (temp->next) {
            temp = temp->next;
        }
        temp->previous->next = nullptr;
        delete temp;
    }

};
int main () {
    DoublyLinkedLis s1 ;
    s1.insertBegin("Facebook.com");
    s1.insertBegin("Youtube.com");
    s1.insertBegin("Tiktok.com");
    s1.insertBegin("Github.com");
    s1.display();
    s1.deletelast();
    s1.display();
    return 0;
```

Output:


OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PROBLEMS
PS C:\Users\USER\OneDrive\CADT_YEAR_2\term-1\Structure\Data-Structure-and-Algorithms> & 'c:
.3-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-q3e0
0' '--stderr=Microsoft-MIEngine-Error-5bppixm2.s5z' '--pid=Microsoft-MIEngine-Pid-yehuezjg.4
eter=mi'
list display :Github.com -> Tiktok.com -> Youtube.com -> Facebook.com -> NULL
list display :Github.com -> Tiktok.com -> Youtube.com -> NULL
PS C:\Users\USER\OneDrive\CADT_YEAR_2\term-1\Structure\Data-Structure-and-Algorithms>

Exercise 5 :

```cpp
#include <iostream>
using namespace std;

class Node {
    public:
    string song ;
    Node* next ;
    Node (string s) {
        song = s;
        next = nullptr;
    }
};
class MusicPlayList {
    private:
        Node * tail;
        Node * head ;
        Node* current;
    public:
    MusicPlayList () {
        head = nullptr;
        tail = nullptr;
        current = nullptr;
    }
    void addSong (string s){
        Node* NewSong = new Node(s);
        if ( head == nullptr){
            head = NewSong;
            tail= NewSong;
            current = head;
            tail->next = head;
        } else {
            tail->next = NewSong ;
            NewSong->next = head ;
            tail = NewSong;
        }
    }
    void PlayNext () {
        if (head == nullptr){
            cout << "PlayList is empty" <<endl;
        }
        cout << "Music :" << current->song << " is playing."<< endl;
        current = current->next;
    }
```

```
43      }
44      void display() {
45          if (head == nullptr) {
46              cout << "Playlist is empty" << endl;
47              return;
48          }
49          Node* temp = head;
50          cout << "Playlist: ";
51          do {
52              cout << temp->song << " -> ";
53              temp = temp->next;
54          } while (temp != head);
55          cout << "(back to start)" << endl;
56      }
57  };
58  int main () {
59      MusicPlayList Mix1;
60      Mix1.addSong("Joji - PIXELATED KISSES");
61      Mix1.addSong("Ricky Montgomery - Mr Loverman");
62      Mix1.addSong("innocence – daniel caesar");
63      Mix1.addSong("The We4kend - រូបកាយ ");
64      Mix1.addSong("Joji - SLOW DANCING IN THE DARK");
65      Mix1.display();
66      Mix1.PlayNext();
67      Mix1.PlayNext();
68      Mix1.PlayNext();
69      Mix1.PlayNext();
70      Mix1.PlayNext();
71  }
```

Output :

```
14ea.5ui' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Playlist: Joji - PIXELATED KISSES -> Ricky Montgomery - Mr Loverman -> innocence – daniel caesar -> The We4kend - រូបកាយ  -> Joji - SLOW DANCING IN THE DARK -> (back to start)
Music :Joji - PIXELATED KISSES is playing.
Music :Ricky Montgomery - Mr Loverman is playing.
Music :innocence – daniel caesar is playing.
Music :The We4kend - រូបកាយ  is playing.
Music :Joji - SLOW DANCING IN THE DARK is playing.
PS C:\Users\USER\OneDrive\CADT_YEAR_2\term-1\Structure\Data-Structure-and-Algorithms>
```

Exercise6 :

```cpp
#include <iostream>
using namespace std;

class Node {
public:
    string url;
    Node* next;
    Node* prev;

    Node(string u) {
        url = u;
        next = nullptr;
        prev = nullptr;
    }
};
class BrowserHistory {
    private:
    Node* current;
    public:
    BrowserHistory() {
        current = nullptr;
    }
    ~BrowserHistory() {
        if (current == nullptr) {
            return;
        }
        Node* temp = current;
        while (temp->prev) {
            temp = temp->prev;
        }
        while (temp) {
            Node* toDelete = temp;
            temp = temp->next;
            delete toDelete;
        }
        current = nullptr;
    }
    void visit(string url) {
        Node* newPage = new Node(url);
        if (current != nullptr) {
            Node* temp = current->next;
            current->next = nullptr;
```

```
            while (temp != nullptr) {
                Node* toDelete = temp;
                temp = temp->next;
                delete toDelete;
            }
            current->next = newPage;
            newPage->prev = current;
        }
        current = newPage;
        cout << "Visited: " << url << endl;
    }
    void back() {
        if (current == nullptr || current->prev == nullptr) {
            cout << "No previous history!" << endl;
            return;
        }
        current = current->prev;
        cout << "Go back to: " << current->url << endl;
    }
    void forward() {
        if (current == nullptr || current->next == nullptr) {
            cout << "No forward history!" << endl;
            return;
        }
        current = current->next;
        cout << "Go forward to: " << current->url << endl;
    }
    void showHistory() {
        if (current == nullptr) {
            cout << "No history available!" << endl;
            return;
        }
        Node* temp = current;
        while (temp->prev != nullptr)
            temp = temp->prev;
        cout << "Browser History: ";
```

```
78          cout << "Browser History: ";
79          while (temp != nullptr) {
80              if (temp == current)
81                  cout << "[" << temp->url << "] "; //marked the current page
82              else
83                  cout << temp->url << " ";
84              temp = temp->next;
85          }
86          cout << endl;
87      }
88  };
89  int main() {
90      BrowserHistory browser;
91
92      browser.visit("google.com");
93      browser.visit("youtube.com");
94      browser.visit("wikipedia.org");
95
96      browser.showHistory();
97      browser.back();
98       browser.showHistory();
99      browser.back();
100      browser.showHistory();
101     browser.forward();
102     browser.showHistory();
103     browser.visit("github.com");
104     browser.showHistory();
105
106     return 0;
107 }
```

Output :

```
PS C:\Users\USER\OneDrive\CADT_YEAR_2\term-1\Structure\Data-Structure-and-Algorithms>
PS C:\Users\USER\OneDrive\CADT_YEAR_2\term-1\Structure\Data-Structure-and-Algorithms>  & 'c:\Users
 '--stdin=Microsoft-MIEngine-In-kvpzhmtt.ose' '--stdout=Microsoft-MIEngine-Out-4b10vmvt.zcc' '--st
s64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Visited: google.com
Visited: youtube.com
Visited: wikipedia.org
Browser History: google.com youtube.com [wikipedia.org]
Go back to: youtube.com
Browser History: google.com [youtube.com] wikipedia.org
Go back to: google.com
Browser History: [google.com] youtube.com wikipedia.org
Go forward to: youtube.com
Browser History: google.com [youtube.com] wikipedia.org
Visited: github.com
Browser History: google.com youtube.com [github.com]
PS C:\Users\USER\OneDrive\CADT_YEAR_2\term-1\Structure\Data-Structure-and-Algorithms>
```