



PIMPRI CHINCHWAD EDUCATION TRUST'S.  
**PIMPRI CHINCHWAD COLLEGE OF ENGINEERING**  
(An Autonomous Institute)

**S.Y. B. TECH**

**Year: 2024 – 25**

**Semester: I**

**Name:** Abhishek Joshi

**PRN:** 123B1B150

**Department:** Computer Engineering

**Division:** C (C1)

**Course:** Data Structures Laboratory

**Course Code:** BCE23PC02

**Date:**

## Assignment –4

- **Aim:**

Implement a simple text editor application using a doubly linked list to manage the text buffer. Text editor should support the following functionalities:

- a. Insert text
- b. Delete text
- c. Display text
- d. Search text
- e. Print text in reverse

- **Source Code :**

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

```
int size = 0;
class node{
public:
    string text;
    node* next;
    node* prev;
    node(string txt){
        text = txt;
        next = NULL;
        prev = NULL;
    }
};

class TextEditor{
public:
    node* head = NULL;
    void insert_txt(string txt){
        size++;
        node* newTxt = new node(txt);
```

```

    if(head == NULL){
        head = newTxt;
    }
    else{
        node* temp;
        temp = head;
        while(temp->next != NULL){
            temp = temp->next;
        }
        temp->next = newTxt;
        newTxt->prev = temp;
        newTxt->next = NULL;
    }

}

void dlt_txt(int pos){
    node* temp = head;
    int count = 1;
    if(pos == 1){
        head = temp->next;
        head->prev = NULL;
        delete temp;
    }
    else if(pos == size){
        while(temp->next->next != NULL){
            temp = temp->next;
        }
        node* curr = temp->next;
        temp->next = NULL;
        delete curr;
    }
    else{
        while((count+1) != pos){
            temp = temp->next;
            count++;
        }
        node* curr = temp->next;
        temp->next = curr->next;
        curr->next->prev = temp;
        delete curr;
    }
}

void displayTxt(){
    node* temp = head;

```

```

        while(temp->next != NULL){
            cout<<temp->text<<" "<<endl;
            temp = temp->next;
        }
        cout<<temp->text<<" "<<endl;
    }
    bool searchTxt(string txt){
        node* temp = head;
        while(temp != NULL){
            if(temp->text == txt){
                return true;
            }
            temp = temp->next;
        }
        return false;
    }
    void printReverseTexts(){
        node* temp = head;
        while(temp->next != NULL){
            temp = temp->next;
        }
        while (temp != NULL){
            cout<<temp->text<<" "<<endl;
            temp = temp->prev;
        }
    }
};

int main(){
    TextEditor editor;
    int choice;
    string text;
    int position;
    do {
        cout << "\n--- Text Editor Menu ---\n";
        cout << "1. Insert Text\n";
        cout << "2. Delete Text by Index\n";
        cout << "3. Display All Texts\n";
        cout << "4. Display Texts in Reverse\n";
        cout << "5. Search Text\n";
        cout << "6. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;
        cin.ignore();
    } while (choice != 6);
}

```

```
switch (choice) {
    case 1:
        cout << "Enter text to insert: ";
        getline(cin, text);
        editor.insert_txt(text);
        cout << "Text inserted successfully." << endl;
        break;

    case 2:
        cout << "Enter index to delete: ";
        cin >> position;
        editor.dlt_txt(position);
        break;

    case 3:
        cout << "Displaying all texts:"<<endl;
        editor.displayTxt();
        break;

    case 4:
        cout << "Displaying texts in reverse:"<<endl;
        editor.printReverseTexts();
        break;

    case 5:
        cout << "Enter text to search: ";
        getline(cin, text);
        if (editor.searchTxt(text)) {
            cout << "Text found..." << endl;
        } else {
            cout << "Text not found..." << endl;
        }
        break;

    case 6:
        cout << "Exiting program..." << endl;
        break;

    default:
        cout << "Invalid choice! Please try again." << endl;
}
} while (choice != 6);
return 0;
}
```

- Screen Shot of Output :

Output

Clea

```
--- Text Editor Menu ---
```

1. Insert Text
2. Delete Text by Index
3. Display All Texts
4. Display Texts in Reverse
5. Search Text
6. Exit

```
Enter your choice: 1
```

```
Enter text to insert: hello pccoe
```

```
Text inserted successfully.
```

```
--- Text Editor Menu ---
```

1. Insert Text
2. Delete Text by Index
3. Display All Texts
4. Display Texts in Reverse
5. Search Text
6. Exit

```
Enter your choice: 2
```

```
Enter index to delete: 1
```

## Output

```
--- Text Editor Menu ---  
1. Insert Text  
2. Delete Text by Index  
3. Display All Texts  
4. Display Texts in Reverse  
5. Search Text  
6. Exit  
Enter your choice: 3  
Displaying all texts:  
Text editor is empty.
```

```
--- Text Editor Menu ---  
1. Insert Text  
2. Delete Text by Index  
3. Display All Texts  
4. Display Texts in Reverse  
5. Search Text  
6. Exit  
Enter your choice: 4  
Displaying texts in reverse:  
Text editor is empty.
```

## Output

Clear

```
--- Text Editor Menu ---
1. Insert Text
2. Delete Text by Index
3. Display All Texts
4. Display Texts in Reverse
5. Search Text
6. Exit
Enter your choice: 5
Enter text to search: hello pccoe
Text not found.

--- Text Editor Menu ---
1. Insert Text
2. Delete Text by Index
3. Display All Texts
4. Display Texts in Reverse
5. Search Text
6. Exit
Enter your choice: 6
Exiting program...

=== Code Execution Successful ===
```



- **Conclusion:**

Hence, we studied about application of Doubly Linked List such as Adding Node, Deleting Node, Searching Node, Reversing List and Displaying List with their algorithm and programs.