



## **Experiment – 1.4**

**Student Name:** Aman

**Branch:** CSE

**Semester:** 5<sup>th</sup>

**Subject Name:** DAA

**UID:** 21BCS9545

**Section/Group:** 21BCS\_KRG\_SN1

**Date of Performance:** 21<sup>nd</sup> Aug'23

**Subject Code:** 21CSH-311

### **Aim:**

- i. Apply the concept of Linked list and write code to Insert and Delete an element at the beginning and end of Singly Linked List.
- ii. Apply the concept of Linked list and write code to Insert and Delete an element at the beginning and end of Doubly Linked List.

### **Objective:**

- The objective of this experiment is to gain knowledge regarding Linked List.

### **Algorithm:**

- **Insertion at the beginning of a Singly Linked List.**
  1. *Start*
  2. *Create a new node*
  3. *Store data in the new node*
  4. *Change the new node to point to the head of the list.*
  5. *Change the head to point to recently created head.*

- **Insertion at the end of a Singly Linked List.**

1. *Allocate a new node.*
2. *Store the data in the new node.*
3. *Traverse to the last node.*
4. *Change the last node to recently created node.*

- **Insertion at the Beginning of a Doubly Linked list.**

1. Create a new node with the data you want to insert.
2. Set the next pointer of the new node to point to the current head of the list.
3. Set the previous pointer of the current head (if it exists) to point to the new node.
4. Update the head pointer of the list to point to the new node.

- **Insertion at the end of a Doubly Linked list.**

- **Step 1:** IF PTR = NULL

Write OVERFLOW

Go to Step 11

[END OF IF]

- **Step 2:** SET NEW\_NODE = PTR

- **Step 3:** SET PTR = PTR -> NEXT

- **Step 4:** SET NEW\_NODE -> DATA = VAL

- **Step 5:** SET NEW\_NODE -> NEXT = NULL

- **Step 6:** SET TEMP = START

- **Step 7:** Repeat Step 8 while TEMP -> NEXT != NULL

- **Step 8:** SET TEMP = TEMP -> NEXT

[END OF LOOP]

- **Step 9:** SET TEMP -> NEXT = NEW\_NODE

- **Step 10C:** SET NEW\_NODE -> PREV = TEMP

- **Step 11:** EXIT

## Script:

1.

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

struct Node{
    int data = 41;
    Node* next;
};

void printList(Node* head){
    while(head != NULL){
        cout << head -> data << " -> ";
        head = head -> next;
    } cout << "\n";
}

int main(){
    Node* head = new Node();
    head -> next = NULL;

    Node* Tail = NULL;
    int option;

    do{
        cout << "Do you want to insert a new node at the beginning: (1) or the end: (2) or Exit(3)?\n";
        cin >> option;

        Node* newNode = new Node();

        switch(option){
            case(1):
                cout << "Enter the value for the new Node to insert the beginning: " << "\n";
                int newData1;
                cin >> newData1;

                if(head -> next == NULL){
                    Tail = newNode;
                }

                newNode -> data = newData1;
                newNode -> next = head -> next;
                head -> next = newNode;
                break;

            case(2):
                cout << "Enter the value of the node to insert at the End : " << "\n";
                int newData2;
                cin >> newData2;
                newNode -> data = newData2;

                if(Tail == NULL){
                    head -> next = newNode;
                    Tail = newNode;
                }
                else{
                    Tail -> next = newNode;
                    Tail = Tail -> next;
                }
                break;

            case(3):
                cout << "END\n";
                break;

        }
        printList(head->next);
    }
    while(option != 3);
}
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Screenshot of Output:

1.

```
abeltheog — fakeg — 143x24
23 ->
Do you want to insert a new node at the beginning: (1) or the end: (2) or Exit(3)? 1
Enter the value for the new Node to insert the beginning:
23
23 -> 23 ->
Do you want to insert a new node at the beginning: (1) or the end: (2) or Exit(3)? 1
Enter the value for the new Node to insert the beginning:
45
45 -> 23 -> 23 ->
Do you want to insert a new node at the beginning: (1) or the end: (2) or Exit(3)? 2
Enter the value of the node to insert at the End :
49
45 -> 23 -> 23 -> 49 ->
Do you want to insert a new node at the beginning: (1) or the end: (2) or Exit(3)? 3
45 -> 23 -> 23 -> 49 ->
END

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

(Process completed)_
```

## Script:

2.

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

class Node {
public:
    int data;
    Node* next;
    Node* prev;

    Node(int value) : data(value), next(nullptr), prev(nullptr) {}
};

class DoublyLinkedList {
public:
    Node* head;

    DoublyLinkedList() : head(nullptr) {}

    // Function to insert a node at the beginning
    void insertAtBeginning(int data) {
        Node* newNode = new Node(data);

        // Update next and prev pointers
        newNode->next = head;
        if (head != nullptr) {
            head->prev = newNode;
        }

        // Update head
        head = newNode;
    }

    // Function to display the linked list
    void display() {
        Node* current = head;
        while (current != nullptr) {
            std::cout << current->data << " <-> ";
            current = current->next;
        }
        std::cout << "nullptr" << std::endl;
    }
};

int main() {
    DoublyLinkedList dll;
    dll.insertAtBeginning(3);
    dll.insertAtBeginning(2);
    dll.insertAtBeginning(1);

    dll.display(); // Output: 1 <-> 2 <-> 3 <-> nullptr

    return 0;
}
```

## Output:

```

Last login: Tue Sep 12 03:29:08 on ttys001
/Users/abeltheog/Downloads/xcudes/xcudes/nig/nig/main ; exit;
abeltheog@Mohammads-Air ~ % /Users/abeltheog/Downloads/xcudes/xcudes/nig/nig/main ; exit;
1 <-> 2 <-> 3 <-> nullptr

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Process completed]_

```

## Script:

2.1.

```

#include <iostream>

class Node {
public:
    int data;
    Node* next;
    Node* prev;

    Node(int value) : data(value), next(nullptr), prev(nullptr) {}
};

class DoublyLinkedList {
public:
    Node* head;
    DoublyLinkedList() : head(nullptr) {}
    // Function to insert a node at the end
    void insertAtEnd(int data) {
        Node* newNode = new Node(data);

        if (head == nullptr) {
            // If the list is empty, make the new node the head
            head = newNode;
        } else {
            // Traverse to find the last node
            Node* current = head;
            while (current->next != nullptr) {
                current = current->next;
            }

            // Update pointers to insert at the end
            current->next = newNode;
            newNode->prev = current;
        }
    }

    // Function to display the linked list
    void display() {
        Node* current = head;
        while (current != nullptr) {
            std::cout << current->data << " <-> ";
            current = current->next;
        }
        std::cout << "nullptr" << std::endl;
    }
};

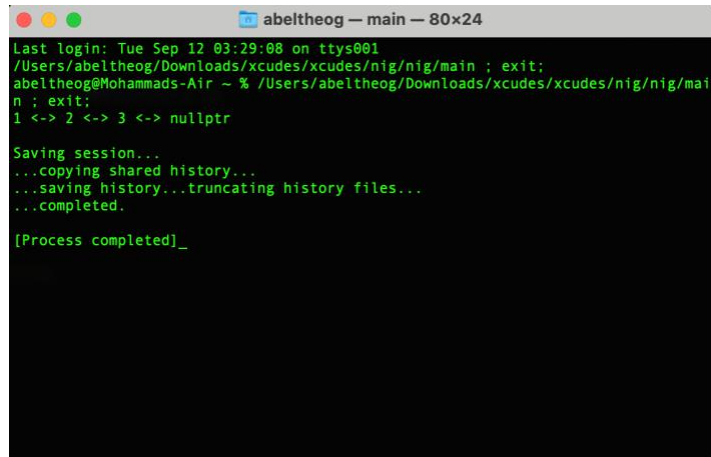
int main() {
    DoublyLinkedList dll;
    dll.insertAtEnd(1);
    dll.insertAtEnd(2);
    dll.insertAtEnd(3);

    dll.display(); // Output: 1 <-> 2 <-> 3 <-> nullptr

    return 0;
}

```

## Screenshots:



```
abeltheog — main — 80x24
Last login: Tue Sep 12 03:29:08 on ttys001
/Users/abeltheog/Downloads/xcudes/xcudes/nig/nig/main : exit;
abeltheog@Mohammads-Air ~ % /Users/abeltheog/Downloads/xcudes/xcudes/nig/nig/main : exit;
1 <-> 2 <-> 3 <-> nullptr

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Process completed]_
```

## Observation/Outcome:

- We got to learn about the Linked list.