



Bharatiya Vidya Bhavan's  
**Sardar Patel Institute of Technology**  
(Autonomous Institute Affiliated to University of Mumbai)

Name: Jhaveri Varun Nimitt

UID: 2023800042

Batch: CSE A Batch C

Experiment No.:4

Aim: Doubly Linked List application

Problem:

- 1- Create a doubly Linked List. All possible test cases to be considered
- 2- Deletion of a given node. All possible test cases to be considered



Bharatiya Vidya Bhavan's

# Sardar Patel Institute of Technology

(Autonomous Institute Affiliated to University of Mumbai)

## Solution:

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
    int data;
    struct Node* next;
    struct Node* prev;
};

struct Node* head = NULL;

struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    newNode->prev = NULL;
    return newNode;
}

int getLength() {
    struct Node* temp = head;
    int length = 0;
    while (temp != NULL) {
        length++;
        temp = temp->next;
    }
    return length;
}

void insertAtHead(int data) {
    struct Node* newNode = createNode(data);
    if (head == NULL) {
        head = newNode;
        return;
    }
    newNode->next = head;
    head->prev = newNode;
    head = newNode;
}

void insertAtTail(int data) {
    struct Node* newNode = createNode(data);
    if (head == NULL) {
        head = newNode;
        return;
    }
```



```
}
struct Node* temp = head;
while (temp->next != NULL) {
    temp = temp->next;
}
temp->next = newNode;
newNode->prev = temp;
}

void insertAtPosition(int data, int position) {
    if (position < 1 || position > getLength() + 1) {
        printf("Invalid position\n");
        return;
    }
    if (position == 1) {
        insertAtHead(data);
        return;
    }
    struct Node* newNode = createNode(data);
    struct Node* temp = head;
    for (int i = 1; i < position - 1; i++) {
        temp = temp->next;
    }
    if (temp == NULL || temp->next == NULL) {
        insertAtTail(data);
    } else {
        newNode->next = temp->next;
        newNode->prev = temp;
        temp->next->prev = newNode;
        temp->next = newNode;
    }
}

void deleteHead() {
    if (head == NULL) {
        printf("List is empty\n");
        return;
    }
    struct Node* temp = head;
    head = head->next;
    if (head != NULL) head->prev = NULL;
    free(temp);
}

void deleteTail() {
    if (head == NULL) {
        printf("empty list\n");
```



```
        return;
    }
    struct Node* temp = head;
    if (head->next == NULL) {
        free(head);
        head = NULL;
        return;
    }
    while (temp->next != NULL) {
        temp = temp->next;
    }
    temp->prev->next = NULL;
    free(temp);
}

void deleteAtPosition(int position) {
    if (head == NULL || position < 1 || position > getLength()) {
        printf("wrong pos or empty list\n");
        return;
    }
    if (position == 1) {
        deleteHead();
        return;
    }
    struct Node* temp = head;
    for (int i = 1; i < position; i++) {
        temp = temp->next;
    }
    if (temp->prev != NULL) temp->prev->next = temp->next;
    if (temp->next != NULL) temp->next->prev = temp->prev;
    free(temp);
}

void displayForward() {
    struct Node* temp = head;
    if (head == NULL) {
        printf("empty list\n");
        return;
    }
    while (temp != NULL) {
        printf("%d ", temp->data);
        temp = temp->next;
    }
    printf("\n");
}

void displayBackward() {
```



Bharatiya Vidya Bhavan's

# Sardar Patel Institute of Technology

(Autonomous Institute Affiliated to University of Mumbai)

```
if (head == NULL) {
    printf("empty list\n");
    return;
}
struct Node* temp = head;
while (temp->next != NULL) {
    temp = temp->next;
}
while (temp != NULL) {
    printf("%d ", temp->data);
    temp = temp->prev;
}
printf("\n");
}

int main() {

    deleteHead();
    deleteTail();
    deleteAtPosition(1);

    insertAtHead(10);
    displayForward();

    insertAtHead(5);
    displayForward();

    insertAtTail(20);
    displayForward();

    insertAtPosition(15, 3);
    displayForward();

    insertAtPosition(25, 6);

    insertAtPosition(25, 5);
    displayForward();

    deleteHead();
    displayForward();

    deleteTail();
    displayForward();

    deleteAtPosition(2);
    displayForward();
}
```



Bharatiya Vidya Bhavan's

# Sardar Patel Institute of Technology

(Autonomous Institute Affiliated to University of Mumbai)

```
deleteAtPosition(5);

displayForward();
displayBackward();

return 0;
}
```

## OUTPUT:

A screenshot of a Linux terminal window. The top status bar shows system information: 1% battery, 4.7568 GB memory, 54.8°C temperature, 51°C temperature, 14:05 time, and 17-09-24 date. The terminal output shows the execution of a program named 'doubly.c'. The user enters 'gcc doubly.c' and './a.out'. The program outputs: 'List is empty', 'empty list', 'wrong pos or empty list', '10', '5 10', '5 10 20', '5 10 15 20', 'Invalid position', '5 10 15 20 25', '10 15 20 25', '10 15 20', '10 20', 'wrong pos or empty list', '10 20', and '20 10'. The terminal prompt is '~ /Desktop/College/Data Structures Sem 3/Experiment 4' and the cursor is on the line 'main ?1'.