**Name :M Ijaz**

**Submitted to**

**Sir Rasikh**

**Roll no . 073**

**Subject . Lab Data structure**

**Lab .7**

**Answer the following questions**

Question no

Implement functions to insert node at first, last, Nth location, and centre of a doubly linked list. And display in order and display in reverse order.

#include <iostream> using namespace std;

class Node { public: int data;

Node\* next;

Node\* prev;

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

};

class DoublyLinkedList { public:

Node\* head;

DoublyLinkedList() : head(nullptr) {}

void insertFirst(int val) {

Node\* newNode = new Node(val);

if (!head) { head = newNode;

} else {

newNode->next = head; head->prev = newNode; head = newNode;

}

}

void insertLast(int val) {

Node\* newNode = new Node(val);

if (!head) { head = newNode;

} else {

Node\* temp = head; while (temp->next) temp = temp->next; temp->next = newNode; newNode->prev = temp;

}

}

void insertNth(int val, int n) { if (n == 1) { insertFirst(val); return;

}

Node\* newNode = new Node(val); Node\* temp = head; for (int i = 1; temp != nullptr && i < n - 1; i++) { temp = temp->next;

} if (!temp) return; newNode->next = temp->next; if (temp->next) temp->next->prev = newNode; temp->next = newNode; newNode->prev = temp;

}

void insertCentre(int val) { if (!head) return;

Node\* slow = head; Node\* fast = head; while (fast && fast->next) { slow = slow->next; fast = fast->next->next;

}

Node\* newNode = new Node(val); newNode->next = slow->next; if (slow->next) slow->next->prev = newNode; slow->next = newNode;

newNode->prev = slow;

}

void displayForward() { Node\* temp = head; while (temp) { cout << temp->data << " "; temp = temp->next;

}

cout << endl;

}

void displayReverse() { if (!head) return; Node\* temp = head; while (temp->next) temp = temp->next; while (temp) { cout << temp->data << " "; temp = temp->prev;

}

cout << endl;

}

};

int main() { DoublyLinkedList dll; dll.insertFirst(10); dll.insertFirst(20); dll.insertLast(5); dll.insertNth(15, 3); dll.insertCentre(25); dll.displayForward(); dll.displayReverse(); return 0;

}

