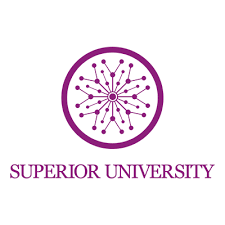
**TASK NO 7**

****

**Name:  
M. Zuhaib Anwar**

**Class :**

**SE – 3A**

**Roll No:**

**SU92-BSSEM-S24-005**

#include <iostream>

using namespace std;

class Node

{

public:

int data;

Node\* next;

Node\* prev;

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

};

class DLL

{

private:

Node\* head;

public:

DLL() : head(nullptr) {}

void insertFirst(int val)

{

Node\* newNode = new Node(val);

if (head)

{

newNode->next = head;

head->prev = newNode;

}

head = newNode;

}

void insertLast(int val)

{

Node\* newNode = new Node(val);

if (!head)

{

head = newNode;

return;

}

Node\* temp = head;

while (temp->next) temp = temp->next;

temp->next = newNode;

newNode->prev = temp;

}

void insertNth(int val, int n)

{

Node\* newNode = new Node(val);

if (n == 0)

{

insertFirst(val);

return;

}

Node\* temp = head;

for (int i = 0; temp && i < n - 1; i++) temp = temp->next;

if (temp)

{

newNode->next = temp->next;

newNode->prev = temp;

if (temp->next) temp->next->prev = newNode;

temp->next = newNode;

}

}

void insertCenter(int val)

{

Node\* newNode = new Node(val);

if (!head)

{

head = newNode;

return;

}

Node\* slow = head;

Node\* fast = head;

while (fast && fast->next)

{

fast = fast->next->next;

if (fast) slow = slow->next;

}

newNode->next = slow;

newNode->prev = slow->prev;

if (slow->prev) slow->prev->next = newNode;

slow->prev = newNode;

if (slow == head) head = newNode;

}

void displayForward()

{

Node\* temp = head;

while (temp)

{

cout << temp->data << " ";

temp = temp->next;

}

cout << endl;

}

void displayReverse()

{

Node\* temp = head;

if (!temp) return;

while (temp->next) temp = temp->next;

while (temp)

{

cout << temp->data << " ";

temp = temp->prev;

}

cout << endl;

}

};

int main() {

DLL dll;

dll.insertFirst(10);

dll.insertLast(20);

dll.insertNth(15, 1);

dll.insertCenter(12);

dll.displayForward();

dll.displayReverse();

return 0;

}

OUTPUT:

