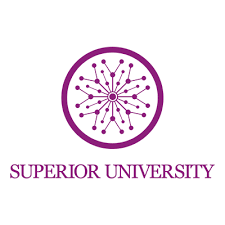
**TASK NO 6**

****

**Name:  
M. Zuhaib Anwar**

**Class :**

**SE – 3A**

**Roll No:**

**SU92-BSSEM-S24-005**

#include <iostream>

using namespace std;

class Node

{

public:

int value;

Node\* next;

Node(int val) : value(val), next(nullptr) {}

};

class LinkedList

{

private:

Node\* head;

public:

LinkedList() : head(nullptr) {}

void addToEnd(int val)

{

Node\* newNode = new Node(val);

if (!head)

{

head = newNode;

return;

}

Node\* current = head;

while (current->next)

{

current = current->next;

}

current->next = newNode;

}

void removeFirst()

{

if (!head) return;

Node\* temp = head;

head = head->next;

delete temp;

}

void removeLast()

{

if (!head) return;

if (!head->next)

{

delete head;

head = nullptr;

return;

}

Node\* current = head;

while (current->next && current->next->next)

{

current = current->next;

}

delete current->next;

current->next = nullptr;

}

void removeAt(int position)

{

if (position <= 0 || !head) return;

if (position == 1)

{

removeFirst();

return;

}

Node\* current = head;

for (int i = 1; i < position - 1 && current; ++i)

{

current = current->next;

}

if (current && current->next)

{

Node\* nodeToRemove = current->next;

current->next = nodeToRemove->next;

delete nodeToRemove;

}

}

void removeMiddle()

{

if (!head) return;

if (!head->next)

{

delete head;

head = nullptr;

return;

}

Node\* slow = head;

Node\* fast = head;

Node\* prev = nullptr;

while (fast && fast->next)

{

prev = slow;

slow = slow->next;

fast = fast->next->next;

}

if (prev)

{

prev->next = slow->next;

delete slow;

}

}

void display()

{

if (!head)

{

cout << "Empty" << endl;

return;

}

Node\* current = head;

while (current)

{

cout << current->value << " ";

current = current->next;

}

cout << endl;

}

};

int main() {

LinkedList list;

list.addToEnd(1);

list.addToEnd(2);

list.addToEnd(3);

list.addToEnd(4);

list.addToEnd(5);

cout << "Original List: ";

list.display();

list.removeFirst();

cout << "After Removing the First Node: ";

list.display();

list.removeLast();

cout << "After Removing the Last Node: ";

list.display();

list.removeAt(2);

cout << "After Removing the 2nd Node: ";

list.display();

list.removeMiddle();

cout << "After Removing the Middle Node: ";

list.display();

}

OUTPUT:

