# Lab Task 6

## Code Explanation:

Singly Linked List implementation with functions to delete first, last, at position, and center nodes.

## Code:

#include <iostream>  
using namespace std;  
  
class Node {  
public:  
 int data;  
 Node\* next;  
 Node(int val) : data(val), next(nullptr) {}  
};  
  
class SinglyLinkedList {  
private:  
 Node\* head;  
 int size;  
  
public:  
 SinglyLinkedList() : head(nullptr), size(0) {}  
  
 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;  
 }  
 size++;  
 }  
  
 void deleteFirst() {  
 if (!head) return;  
 Node\* temp = head;  
 head = head->next;  
 delete temp;  
 size--;  
 }  
  
 void deleteLast() {  
 if (!head) return;  
 if (!head->next) {  
 delete head;  
 head = nullptr;  
 } else {  
 Node\* temp = head;  
 while (temp->next->next) temp = temp->next;  
 delete temp->next;  
 temp->next = nullptr;  
 }  
 size--;  
 }  
  
 void deleteAt(int pos) {  
 if (pos < 1 || pos > size) return;  
 if (pos == 1) return deleteFirst();  
   
 Node\* temp = head;  
 for (int i = 1; i < pos - 1; i++) temp = temp->next;  
   
 Node\* toDelete = temp->next;  
 temp->next = temp->next->next;  
 delete toDelete;  
 size--;  
 }  
  
 void deleteCenter() {  
 deleteAt((size / 2) + (size % 2));  
 }  
  
 void display() {  
 Node\* temp = head;  
 while (temp) {  
 cout << temp->data << " ";  
 temp = temp->next;  
 }  
 cout << endl;  
 }  
};  
  
int main() {  
 SinglyLinkedList sll;  
 sll.insertLast(10);  
 sll.insertLast(20);  
 sll.insertLast(30);  
 sll.insertLast(40);  
 sll.insertLast(50);  
  
 sll.display();  
  
 sll.deleteFirst();  
 sll.display();  
  
 sll.deleteLast();  
 sll.display();  
  
 sll.deleteAt(2);  
 sll.display();  
  
 sll.deleteCenter();  
 sll.display();  
  
 return 0;  
}

## Output:

```  
10 20 30 40 50  
20 30 40 50  
20 30 40  
20 40  
40  
  
```