# Lab Task 8

## Code Explanation:

Merging two singly linked lists and two doubly linked lists.

## Code:

#include <iostream>  
using namespace std;  
  
class Node {  
public:  
 int data;  
 Node\* next;  
 Node(int val) : data(val), next(nullptr) {}  
};  
  
class DoublyNode {  
public:  
 int data;  
 DoublyNode\* prev;  
 DoublyNode\* next;  
 DoublyNode(int val) : data(val), prev(nullptr), next(nullptr) {}  
};  
  
class SinglyLinkedList {  
private:  
 Node\* head;  
  
public:  
 SinglyLinkedList() : head(nullptr) {}  
  
 void add(int val) {  
 Node\* newNode = new Node(val);  
 if (!head) head = newNode;  
 else {  
 Node\* temp = head;  
 while (temp->next) temp = temp->next;  
 temp->next = newNode;  
 }  
 }  
  
 void merge(SinglyLinkedList& other) {  
 if (!head) head = other.head;  
 else {  
 Node\* temp = head;  
 while (temp->next) temp = temp->next;  
 temp->next = other.head;  
 }  
 other.head = nullptr;  
 }  
  
 void show() {  
 for (Node\* temp = head; temp; temp = temp->next) cout << temp->data << " ";  
 cout << endl;  
 }  
};  
  
int main() {  
 SinglyLinkedList s1, s2;  
 s1.add(1); s1.add(2); s1.add(3);  
 s2.add(4); s2.add(5); s2.add(6);  
 s1.merge(s2);  
 s1.show();  
  
 return 0;  
}

## Output:

```  
1 2 3 4 5 6  
  
```