SUPERIOR UNIVERSITY

NAME :ZOHAIB KHAN

ROLL_ NO : 022

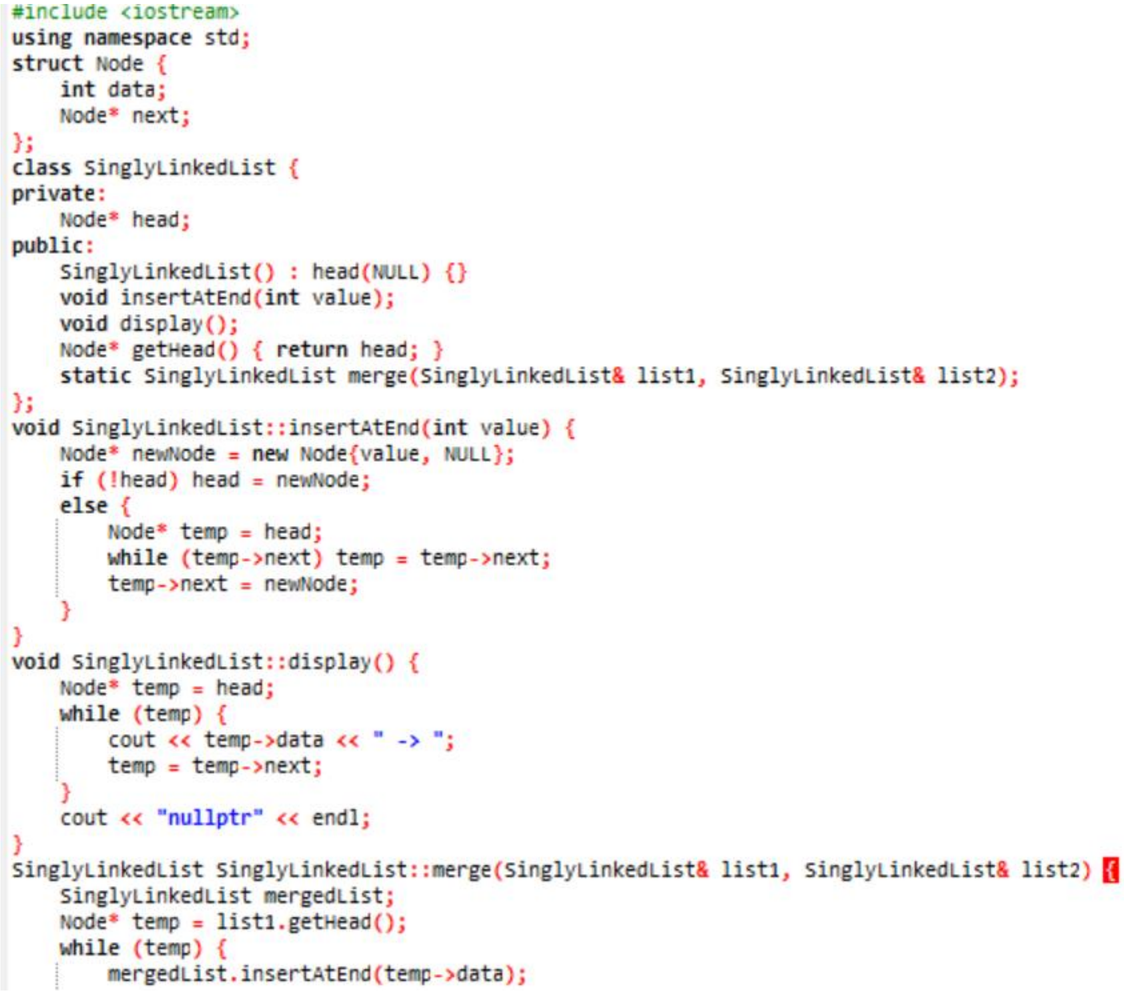
SUBJECT: DSA LAB

LAB NO 8

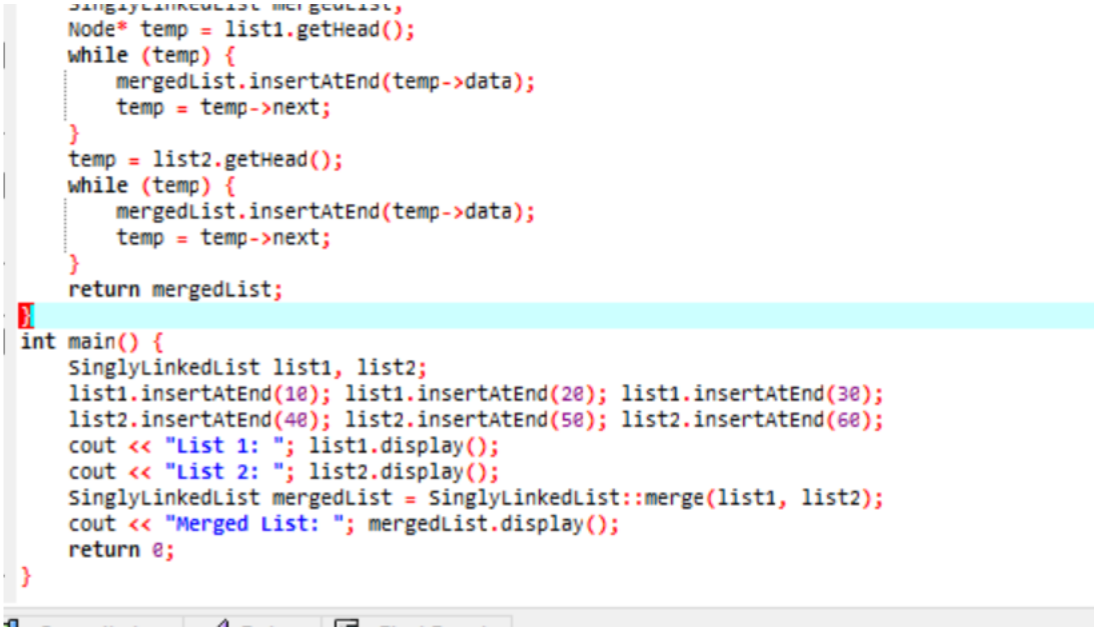
Task:

1. Create 2 Singly LinkedLists and Merge them and display them.

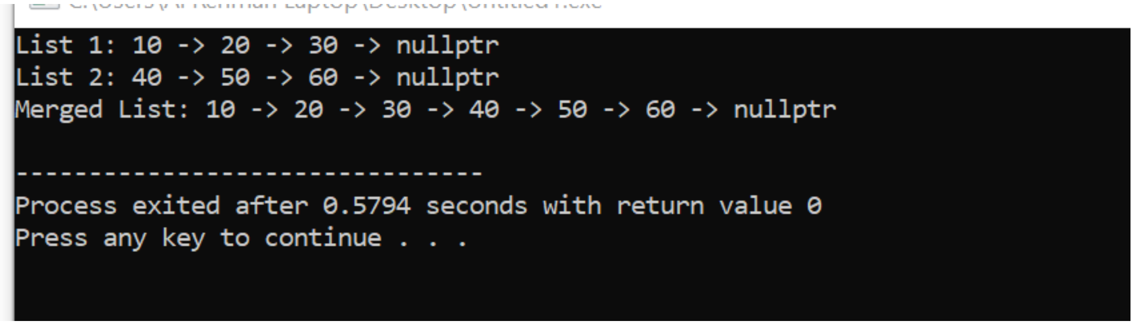
2. Create 2 Double LinkedLists and Merge them and display them.



```
#include <iostream>
using namespace std;
struct Node {
    int data;
    Node* next;
};
class SinglyLinkedList {
private:
    Node* head;
public:
    SinglyLinkedList() : head(NULL) {}
    void insertAtEnd(int value);
    void display();
    Node* getHead() { return head; }
    static SinglyLinkedList merge(SinglyLinkedList& list1, SinglyLinkedList& list2);
};
void SinglyLinkedList::insertAtEnd(int value) {
    Node* newNode = new Node{value, NULL};
    if (!head) head = newNode;
    else {
        Node* temp = head;
        while (temp->next) temp = temp->next;
        temp->next = newNode;
    }
}
void SinglyLinkedList::display() {
    Node* temp = head;
    while (temp) {
        cout << temp->data << " -> ";
        temp = temp->next;
    }
    cout << "nullptr" << endl;
}
SinglyLinkedList SinglyLinkedList::merge(SinglyLinkedList& list1, SinglyLinkedList& list2) {
    SinglyLinkedList mergedList;
    Node* temp = list1.getHead();
    while (temp) {
        mergedList.insertAtEnd(temp->data);
```



```
    SinglyLinkedList mergedList;
    Node* temp = list1.getHead();
    while (temp) {
        mergedList.insertAtEnd(temp->data);
        temp = temp->next;
    }
    temp = list2.getHead();
    while (temp) {
        mergedList.insertAtEnd(temp->data);
        temp = temp->next;
    }
    return mergedList;
}
int main() {
    SinglyLinkedList list1, list2;
    list1.insertAtEnd(10); list1.insertAtEnd(20); list1.insertAtEnd(30);
    list2.insertAtEnd(40); list2.insertAtEnd(50); list2.insertAtEnd(60);
    cout << "List 1: "; list1.display();
    cout << "List 2: "; list2.display();
    SinglyLinkedList mergedList = SinglyLinkedList::merge(list1, list2);
    cout << "Merged List: "; mergedList.display();
    return 0;
}
```



```
List 1: 10 -> 20 -> 30 -> nullptr
List 2: 40 -> 50 -> 60 -> nullptr
Merged List: 10 -> 20 -> 30 -> 40 -> 50 -> 60 -> nullptr

--------------------------------
Process exited after 0.5794 seconds with return value 0
Press any key to continue . . .
```

1. insertAtEnd(int value): Inserts a node at the end of the singly linked list.
2. display(): Displays the singly linked list.
3. getHead(): Returns the head of the list.
4. merge(SinglyLinkedList& list1, SinglyLinkedList& list2): Merges two singly linked lists.