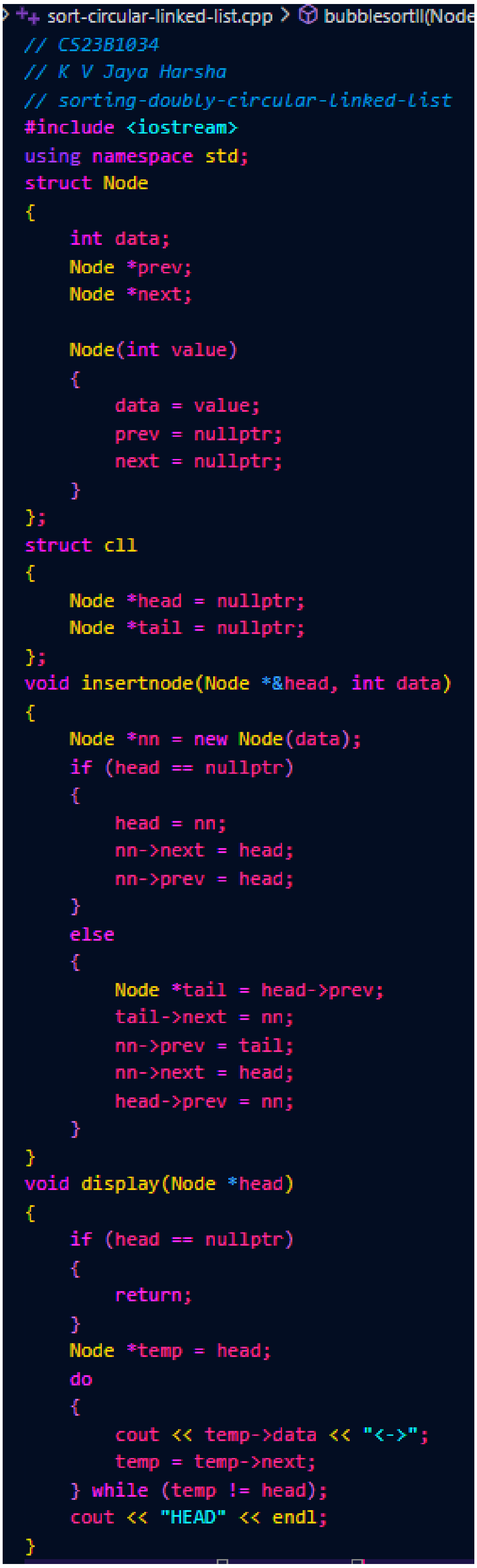
LAB-08

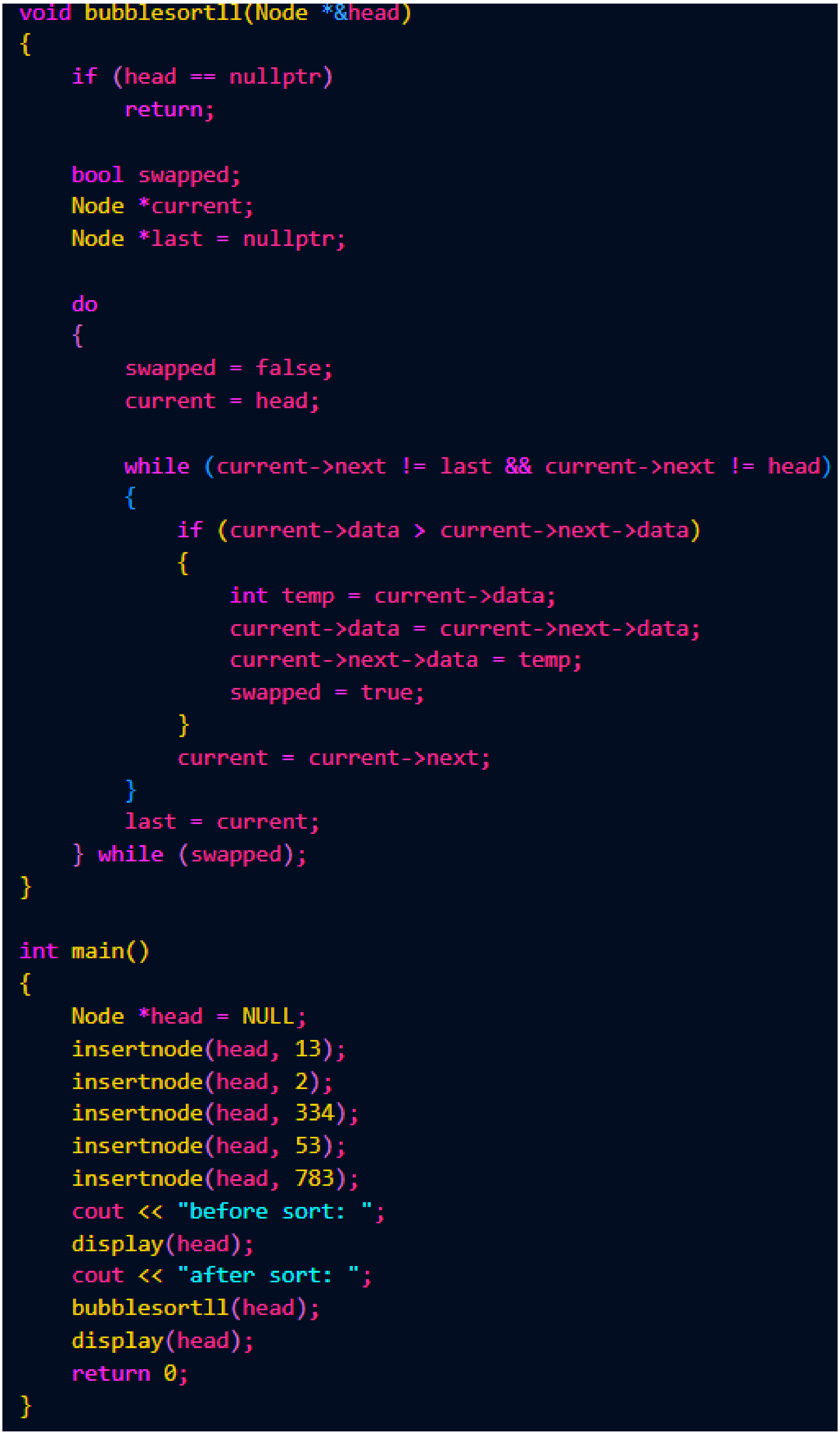
Name: K V Jaya Harsha

Roll no: CS23B1034 Date: 18-09-2024

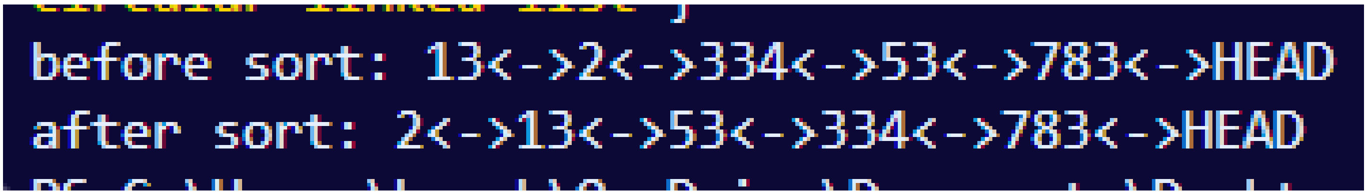
Q1. bubble-sorting-doubly-circular-linked-list. (in cpp)



```cpp
// CS23B1034
// K V Jaya Harsha
// sorting-doubly-circular-linked-list
#include <iostream>
using namespace std;
struct Node
{
    int data;
    Node *prev;
    Node *next;

    Node(int value)
    {
        data = value;
        prev = nullptr;
        next = nullptr;
    }
};
struct cll
{
    Node *head = nullptr;
    Node *tail = nullptr;
};
void insertnode(Node *&head, int data)
{
    Node *nn = new Node(data);
    if (head == nullptr)
    {
        head = nn;
        nn->next = head;
        nn->prev = head;
    }
    else
    {
        Node *tail = head->prev;
        tail->next = nn;
        nn->prev = tail;
        nn->next = head;
        head->prev = nn;
    }
}
void display(Node *head)
{
    if (head == nullptr)
    {
        return;
    }
    Node *temp = head;
    do
    {
        cout << temp->data << "<->";
        temp = temp->next;
    } while (temp != head);
    cout << "HEAD" << endl;
}
```

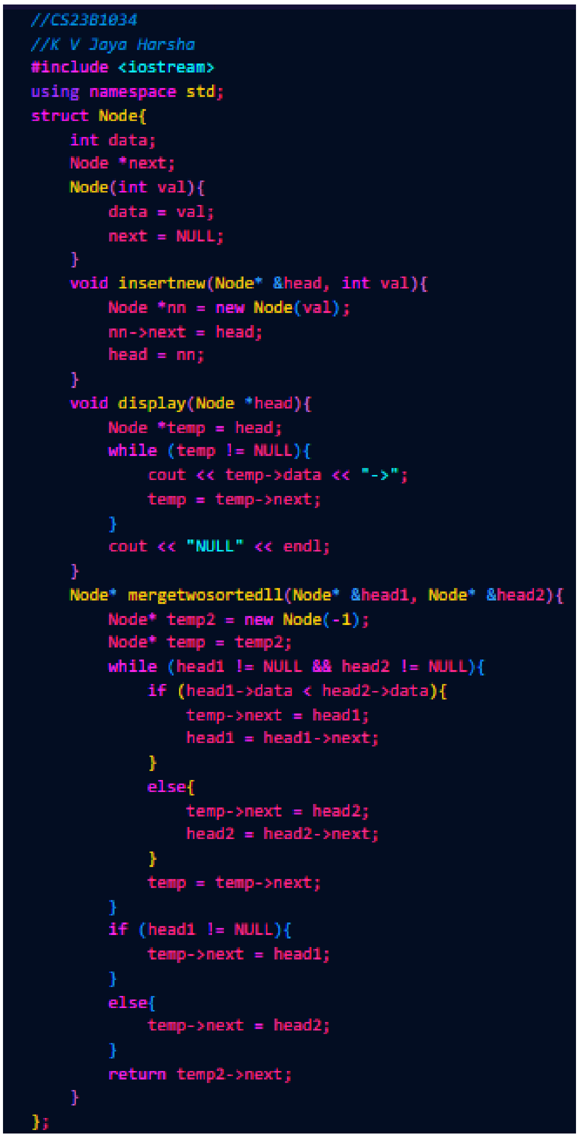


```cpp
void bubblesortll(Node *&head)
{
    if (head == nullptr)
        return;

    bool swapped;
    Node *current;
    Node *last = nullptr;

    do
    {
        swapped = false;
        current = head;

        while (current->next != last && current->next != head)
        {
            if (current->data > current->next->data)
            {
                int temp = current->data;
                current->data = current->next->data;
                current->next->data = temp;
                swapped = true;
            }
            current = current->next;
        }
        last = current;
    } while (swapped);
}

int main()
{
    Node *head = NULL;
    insertnode(head, 13);
    insertnode(head, 2);
    insertnode(head, 334);
    insertnode(head, 53);
    insertnode(head, 783);
    cout << "before sort: ";
    display(head);
    cout << "after sort: ";
    bubblesortll(head);
    display(head);
    return 0;
}
```

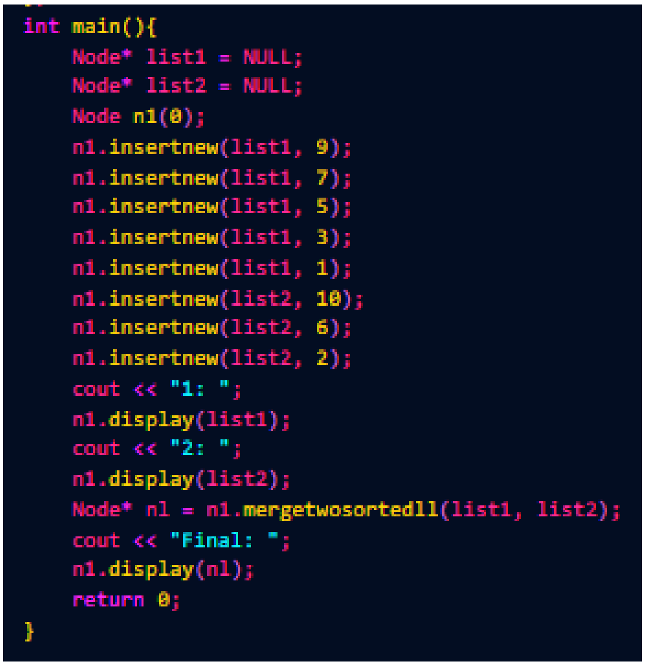


```
before sort: 13<->2<->334<->53<->783<->HEAD
after sort: 2<->13<->53<->334<->783<->HEAD
```

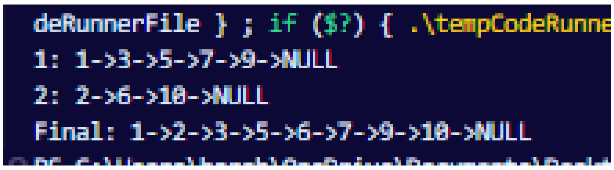
Q2. Merge two sorted circular single linked list. (in cpp)



```cpp
//CS23B1034
//K V Jaya Harsha
#include <iostream>
using namespace std;
struct Node{
    int data;
    Node *next;
    Node(int val){
        data = val;
        next = NULL;
    }
    void insertnew(Node* &head, int val){
        Node *nn = new Node(val);
        nn->next = head;
        head = nn;
    }
    void display(Node *head){
        Node *temp = head;
        while (temp != NULL){
            cout << temp->data << "->";
            temp = temp->next;
        }
        cout << "NULL" << endl;
    }
    Node* mergetwosortedll(Node* &head1, Node* &head2){
        Node* temp2 = new Node(-1);
        Node* temp = temp2;
        while (head1 != NULL && head2 != NULL){
            if (head1->data < head2->data){
                temp->next = head1;
                head1 = head1->next;
            }
            else{
                temp->next = head2;
                head2 = head2->next;
            }
            temp = temp->next;
        }
        if (head1 != NULL){
            temp->next = head1;
        }
        else{
            temp->next = head2;
        }
        return temp2->next;
    }
};
```



```cpp
int main(){
    Node* list1 = NULL;
    Node* list2 = NULL;
    Node n1(0);
    n1.insertnew(list1, 9);
    n1.insertnew(list1, 7);
    n1.insertnew(list1, 5);
    n1.insertnew(list1, 3);
    n1.insertnew(list1, 1);
    n1.insertnew(list2, 10);
    n1.insertnew(list2, 6);
    n1.insertnew(list2, 2);
    cout << "1: ";
    n1.display(list1);
    cout << "2: ";
    n1.display(list2);
    Node* nl = n1.mergetwosortedll(list1, list2);
    cout << "Final: ";
    n1.display(nl);
    return 0;
}
```



```
deRunnerFile } ; if ($?) { .\tempCodeRunne
1: 1->3->5->7->9->NULL
2: 2->6->10->NULL
Final: 1->2->3->5->6->7->9->10->NULL
```