

```
/*Create two doubly linked lists. Sort them after
creation using pointer manipulation. Merge these two
lists into one list so that the merged list is in sorted
order. (No new Node should be created.)*/
```

```
#include <iostream>
using namespace std;
```

```
struct Node {
    int data;
    Node* next;
    Node* prev;
};
```

```
Node* head1 = NULL;
Node* head2 = NULL;
```

```
void insert(int data, Node*& head) {
    Node* temp = new Node();
    temp->data = data;
    temp->next = NULL;
    temp->prev = NULL;
    if (!head) {
        head = temp;
    } else {
        Node* ptr = head;
        while (ptr->next) {
            ptr = ptr->next;
        }
        ptr->next = temp;
        temp->prev = ptr;
    }
}
```

```
void sortList(Node*& head) {
    Node* ptr = head;
    while (ptr->next) {
        Node* temp = ptr->next;
        while (temp) {
            if (ptr->data > temp->data) {
                swap(ptr->data, temp->data);
            }
            temp = temp->next;
        }
        ptr = ptr->next;
    }
}
```

```
void mergeLists() {
    Node* ptr1 = head1;
    Node* ptr2 = head2;
    while (ptr1->next) {
        ptr1 = ptr1->next;
    }
    ptr1->next = ptr2;
    ptr2->prev = ptr1;
}
```

```
void printList(Node* head) {
    Node* ptr = head;
```

```

        while (ptr) {
            cout << ptr->data << " ";
            ptr = ptr->next;
        }
        cout << endl;
    }

int main() {
    insert(5, head1);
    insert(2, head1);
    insert(4, head1);
    insert(7, head1);
    sortList(head1);
    printList(head1);

    insert(3, head2);
    insert(1, head2);
    insert(6, head2);
    insert(9, head2);
    sortList(head2);
    printList(head2);

    mergeLists();
    printList(head1);

    return 0;
}

```