

Grupo 13 - Laboratorio 3

Estudiantes:

- FERNANDO HERMOSO CARA (C40)
- IGNACIO PALLÁS GOZÁLVEZ (C62)

Fichero 2020_02_13/C40/2-C02-1/2339_AC/Prc2.cut.cpp

```
/*@ <answer>
* Comienza poniendo el nonmbre de los/as componentes del grupo:
*
* Estudiante 1:
* Estudiante 2:
* @ </answer> */
* @ <answer>
*/
```

10

```
using namespace std;
// coste  $O(n + m)$  respecto al tamao de las 2 listas
```

$O(n+m)$ donde n es la longitud de $l1$ y m la longitud de $l2$

```
/*
IGNACIO PALLS GOZLVEZ
FERNANDO HERMOSO CARA
*/
void ListLinkedList::merge(ListLinkedList &other) {
    // ...
    Node* aux;
    Node* current = this->head;
    Node* current2 = other.head;
```

```
    if (current && current2) {
        if (current->value <= current2->value) {
            current = current->next;
            aux = this->head;
        }
        else {
```

```
            this->head = current2;
            current2 = current2->next;
            this->head->next = current;
            aux = this->head;
            other.head = current2;
```

No hace falta

```
        }
        while (current2 && current) {
            if (current->value <= current2->value) {
                aux->next = current;
                aux = current;
```

puede traer aquí No depende del if

```

        current = aux->next;
    }
    else {
        aux->next = current2;
        aux = current2;
        current2 = aux->next;
    }
}
if (current2 == nullptr) {
    aux->next = current;
}
else if (current == nullptr) {
    aux->next = current2;
}

}
else if (current == nullptr && current2 != nullptr) {
    this->head = other.head;
}
other.head = nullptr;
}

/*
ListLinkedSingle merge_lists(const ListLinkedSingle &l1, const ListLinkedSingle &l2) {
    // ...
}*/

void tratar_caso() {
    int dato;
    ListLinkedSingle l1;
    ListLinkedSingle l2;
    cin >> dato;
    while (dato != 0) {
        l1.push_back(dato);
        cin >> dato;
    }
    cin >> dato;
    while (dato != 0) {
        l2.push_back(dato);
        cin >> dato;
    }
    l1.merge(l2);
    l1.display();
    cout << endl ;
}

int main() {
    int num_casos;
    /*

```

```

#ifndef DOMJUDGE
    std::ifstream in("sample.in");
    auto cinbuf = std::cin.rdbuf(in.rdbuf());
#endif*/

    cin >> num_casos;

    while (num_casos > 0) {
        tratar_caso();
        num_casos--;
    }

/*
#ifndef DOMJUDGE
    std::cin.rdbuf(cinbuf);
    // Descomentar si se trabaja en Visual Studio
    system("PAUSE");

#endif*/
    return 0;
}

```