



Nome(s): **Kelvin Araújo Ferreira - 2019037653**

Nota: _____

EXERCÍCIO - Aula 08 - Listas Duplamente Encadeadas

→ Tendo como base o código visto na **Aula 08 – Listas Duplamente Encadeadas** implemente:

1) Uma função que **adicione** um elemento ao **final** da lista.

List_add_last(List *L, int val)

```
void list_add_last(List *L, int val) {
    Node *p = create_node(val);
    if (list_is_empty(L)) {
        L->end = p;
        L->begin = p;
    } else {
        p->prev = L->end;
        L->end->next = p;
    }
    L->end = p;
    L->size++;
}
```

2) Implemente uma função que **remova** um elemento da lista. Neste sentido a função deve **procurar** o elemento na lista (pode ser inicio, meio ou fim), **remove-lo** e atualizar os ponteiros para manter as propriedades de uma lista duplamente encadeada.

List_remove(List *L, int val)

```
void list_remove(List *L, int val) {
    Node *p = L->begin;
    while (p != NULL) {
        if (p->val == val) {
            if (L->begin->val == val) {
                L->begin = p->next;
                p->next->prev = NULL;
            } else {
                p->prev->next = p->next;
            }
            free(p);
            L->size--;
        }
        p = p->next;
    }
}
```