



1. Crie a função **iterativa** `ordenada(L)`, que devolve 1 se a lista `L` estiver ordenada de forma crescente e sem itens repetidos (ou 0, caso contrário).
2. Para cada caso a seguir, crie uma versão **recursiva** e uma versão **iterativa**:
 - (a) Crie a função `insere_sem_repeticao(x, &L)`, que insere o item `x` na lista ordenada `L`, somente se `x` ainda não estiver em `L` (caso `x` já esteja armazenado em `L`, nada precisa ser feito). Use a função `no()` para criar os nós.
 - (b) Crie a função `remove_toda_ocorrencia(x, &L)`, que remove toda ocorrência do item `x` na lista ordenada `L`. Use a função a seguir para eliminar/descartar os nós da lista ordenada:

```
Item descarta(Lista *L) {  
    if( *L == NULL ) abort();  
    Lista n = *L;  
    Item x = n->item;  
    *L = n->prox;  
    free(n);  
    return x;  
}
```

- (c) Crie a função `remove_maximo(&L)`, que remove e devolve um item máximo da lista ordenada `L`. Use a função `descarta(&L)`, definida no exercício anterior, para eliminar/descartar o nó da lista ordenada.
-