

Estrutura de dados II

Exercícios – Listas Generalizadas

1:-) Fazer um algoritmo para destruir uma lista generalizada (liberar todos os nodos de uma lista).

Três casos a considerar:

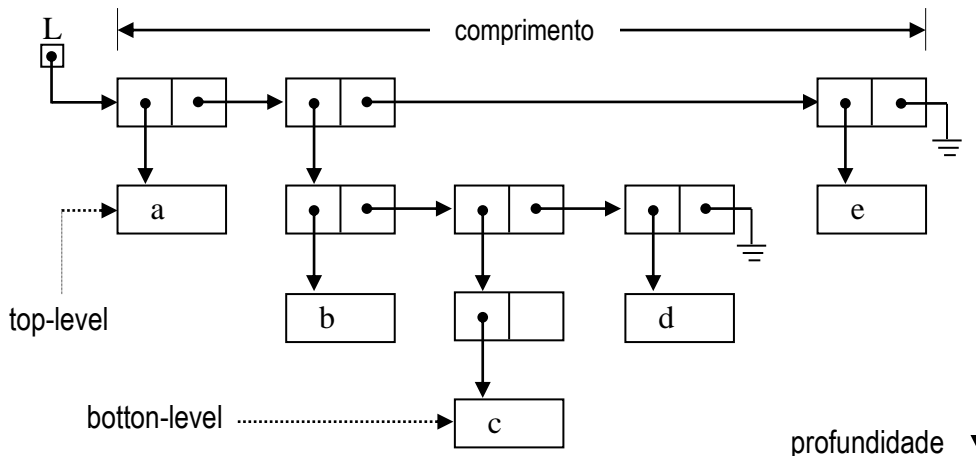
- **L** tem valor nulo, representando uma lista vazia;
- **L** aponta um nodo terminal, representando um átomo;
- **L** aponta um nodo não terminal, representando uma lista generalizada.

2:-) Fazer um algoritmo para duplicar uma lista generalizada fornecida por parâmetro.

3:-) Fazer um algoritmo (função *boolean*) para comparar duas listas generalizadas passadas por parâmetro.

4:-) Numa lista generalizada, cada par de colchetes serve para indicar um nível de aninhamento. Por exemplo, na lista $[a, [b, [c], d], e]$, os elementos **a**, **[b, [c], d]** e **e** pertencem ao primeiro nível de aninhamento, os elementos **b**, **[c]** e **d** pertencem ao segundo nível, e o elemento **c** pertence ao terceiro. Frequentemente, o primeiro nível de aninhamento é denominado *top-level*, enquanto o último (aquele mais interno) é chamado *botton-level*. Com base nisto, podemos definir dois novos conceitos relacionados a uma lista generalizada L:

- **comprimento**: número de elementos existentes no *top-level* da lista L;
- **profundidade**: nível em que se encontra o *botton-level* da lista L.



a) codifique a rotina **Len(L)**, que dá o comprimento de uma lista generalizada L. Admita que a lista nula tem comprimento 0 e que a operação não pode ser aplicada a átomos.

b) codifique a rotina **Deph(L)**, que dá a profundidade de uma lista generalizada L. Admita que átomos têm profundidade 0 e listas vazias têm profundidade 1.

5:-) Codifique a rotina **Append(L, M)**, que concatena as listas generalizadas **L** e **M**. Por exemplo, para **L: [a, [b, [c], d], e, [f]]** e **M: [g, [], [h]]**, a operação **R:=Append(L, M)** resultará na lista **R: [a, [b, [c], d], e, [f], g, [], [h]]**.

6:-) Codifique a rotina **Replace(O,N,L)**, que substitui todas as ocorrências do elemento **O**, em **L**, por um novo elemento **N**.

7:-) Codifique a rotina **TopLevel(L)**, que cria e retorna lista contendo apenas os elementos que se encontram no *top-level* da lista **L**.

8:-) Codifique a rotina **Nth(L,n)**, que retorna o n-ésimo elemento existente no *top-level* de **L**.

9:-) Codifique a rotina **Plain(L)**, que transforma a lista **L** numa lista de profundidade 1. Por exemplo, sendo a lista **L: [a, [b, [c, d]], [], e]**, **Plain(L)** transformá-la-ia na lista **L: [a, b, c, d, e]**.