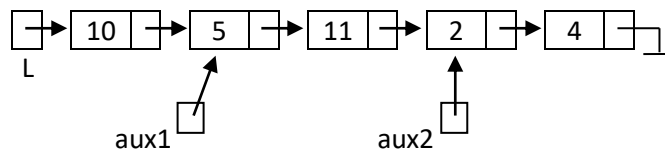




Lista de Exercícios (Listas Encadeadas) – Fundamentos de Programação
Professor Leonardo Vianna [2020/1]

QUESTÃO 01:

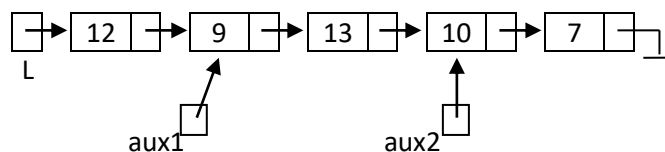
Considere uma lista encadeada com a seguinte configuração na memória principal, onde *L*, *aux1* e *aux2* são do tipo *TLista*:



- a. O que os seguintes elementos armazenam?
- i. L
 - ii. aux1->prox->
 - iii. aux1->valor
 - iv. aux2->prox->prox
- b. Qual o resultado (explique com detalhes) da execução dos comandos a seguir (considerando que os mesmos não são executados em sequência)?
- i. aux2 = aux1->prox;
 - ii. free (aux2->prox);
 - iii. aux1->prox = aux2->prox;
 - iv. L->prox = NULL;

QUESTÃO 02:

Considere uma lista encadeada com a seguinte configuração na memória principal, onde *L*, *aux1* e *aux2* são do tipo *TLista*:



A seguir, são apresentados comandos a serem executados sobre esta lista. Classifique-os como *válidos (V)* ou *inválidos (I)*. Se forem válidos, explicar o resultado de sua execução sobre a lista *L*; caso contrário, justificar o motivo.

Nota: todos os comandos são aplicados sobre a lista na configuração apresentada; isto é, não são executados de maneira sequencial.

- () aux1 = aux2->valor;
- () free (aux2->prox);
- () aux1->prox->prox->valor = L->valor;
- () aux2 = aux1->prox;
- () aux1->prox = aux2;

QUESTÃO 03:

A seguir são apresentadas duas funções que manipulam listas encadeadas do tipo *TLista*, sendo garantido que não há repetição de elementos na mesma estrutura.

Pede-se a descrição do objetivo de cada uma das funções, cabendo ressaltar que *funcao02* chama *funcao01*.

```
int funcao01 (TLista L, int A) {
    TLista aux = L;
    int B = 0;

    while (aux != NULL) {
        if (aux->valor % A == 0) {
            B++;
        }
        aux = aux->prox;
    }
    return B;
}

int funcao02 (TLista L1, TLista L2) {
    TLista aux = L1;
    int C = 0;

    while (aux != NULL) {
        if (funcao01 (L2, aux->valor) > 0) {
            C++;
        }
        aux = aux->prox;
    }
    return C;
}
```

QUESTÃO 04:

Implementar uma função que, dadas duas listas dinâmicas do tipo *TLista*, verifique se estas são iguais; isto é, contêm os mesmos elementos, na mesma ordem.

QUESTÃO 05:

Implementar uma função que, dadas duas listas dinâmicas do tipo *TLista*, verifique se elas possuem os mesmos elementos, independente da ordem na qual apareçam.

QUESTÃO 06:

Desenvolver uma função que, dada uma lista *L1*, crie uma nova lista *L2*, cópia de *L1*.

QUESTÃO 07:

Desenvolver uma função que insira um número inteiro *N* na *i*-ésima posição de uma lista encadeada *L*.

Observação: caso a posição *i* informada seja inválida, a função deverá retornar o valor 0; caso contrário, o retorno será igual a 1.

QUESTÃO 08:

Implementar uma função que crie uma lista encadeada (dinâmica) com os *N* primeiros termos de uma PA (progressão aritmética) de razão *R* e primeiro termo igual a *A1*.

QUESTÃO 09:

Implementar uma função que, dada uma lista dinâmica do tipo *TLista*, verifique se os elementos da estrutura encontram-se ordenados de forma crescente ou não.