

## LISTA DE EXERCÍCIOS 1

Considere as estruturas de Listas, Pilhas e Filas vistas em sala de aula e baseadas em alocação dinâmica. Você pode utilizar qualquer função já vista em sala de aula (ie. não é necessário implementá-las).

1) Considere que `p` e `q` são do tipo "Celula\*". Explique o que acontece nas atribuições abaixo. Você pode utilizar desenhos para facilitar sua explicação.

a) `p->prox = q;`

b) `p->prox = q->Prox;`

c) `p->item= q->item;`

d) `p = q;`

e) `p->prox = NULL;`

f) `p = p->prox;`

g) `p = p->prox->prox;`

2) Imagine e descreva uma situação em que faz sentido utilizar uma Lista Duplamente Encadeada. Explique o porquê.

3) Explique com suas palavras o que é uma Pilha e como é possível implementá-la a partir de uma Lista Simplesmente Encadeada. Você pode utilizar desenhos para facilitar sua explicação.

4) Faça o mesmo para uma Fila.

5) Explique os benefícios de utilizar uma estrutura dinâmica para implementar uma fila (ponteiros), em comparação ao uso de uma estrutura estática (vetores).

**6) Escreva uma função em C para verificar se uma lista simplesmente encadeada L1 está ordenada de maneira crescente. Retornar 0 caso não esteja, e 1 caso esteja.**

```
int ordenada(Lista* L1);
```

**7) Escreva uma função em C para inverter uma lista simplesmente encadeada L1, retornando um resultado L2.**

```
Lista* inverte(Lista* L1);
```

**8) Escreva uma função em C para intercalar duas filas F1 e F2, gerando uma fila F3.**

```
Fila* intercala(Fila* F1, Fila* F2);
```