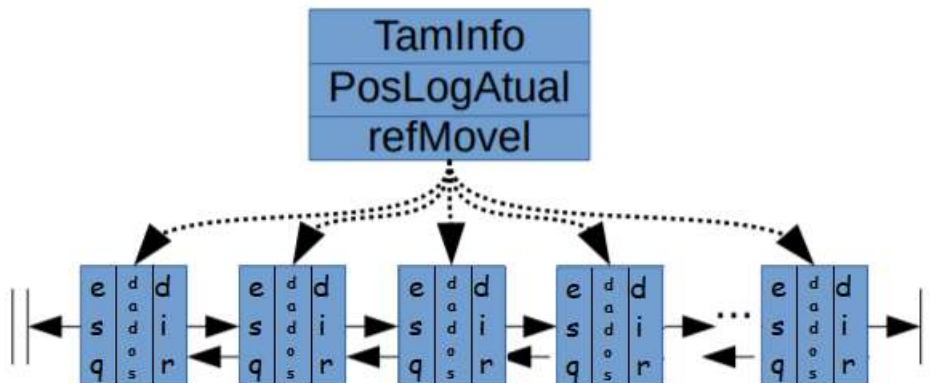


## TAREFA INDIVIDUAL A SER ANEXADA **MANUSCRITA**



### Descrição:

Considere uma LDDE (figura acima) com descritor que possui referência móvel para a lista.

A lista não vazia se alonga da esquerda para a direita: o primeiro elemento (se houver) ocorrerá sempre na extremidade esquerda e o último (se houver) ocorrerá sempre na extremidade direita da lista.

O descritor possui:

- a) Campo *tamInfo*: tamanho em bytes da informação;
- b) Campo *posLogAtual*:
  - Corresponde a posição atual do nó atualmente apontado por *refMovel*;
  - É igual a -1 para a lista vazia.
- d) Campo ponteiro *refMovel* não aponta para nenhuma posição fixa predeterminada, o comportamento do ponteiro é o seguinte:
  - É anulado (Null) para a lista vazia
  - Aponta para o nó recém inserido;
  - Aponta para o nó recém buscado;
  - Aponta para o vizinho à esquerda ou à direita do nó recém removido.

## Pede-se:

1. As seguintes implementações tirando o máximo proveito das características da lista, de maneira que o ponteiro *refMoveI* se desloque o **MÍNIMO** possível:

a) *int tamanhoDaLista(Lista \*p)*

*p*: endereço do descritor da lista;

A função retorna o tamanho da lista.

b) *int buscaNaPosLog(Lista \*p, int posAlvo, info \*reg)*

*p*: endereço do descritor da lista;

*posAlvo*: posição do item de interesse a ser buscado na sequência da lista;

*reg*: endereço de uma variável do tipo *info*, a qual receberá o conteúdo na *posAlvo* se esta existir.

Lembrando que *posAlvo* é válida se  $1 \leq posAlvo \leq tamLista$

A função retorna 1 no caso de sucesso e zero no caso de fracasso na busca.

c) *int removeDaPosLog(Lista \*p, int posAlvo, info \*reg)*

*p*: endereço do descritor da lista;

*posAlvo*: posição do item de interesse a ser removido na sequência da lista;

*reg*: endereço de uma variável do tipo *info*, a qual receberá o conteúdo de informação contida no nó a ser removido da *posAlvo*, se esta existir.

Lembrando que *posAlvo* é válida se  $1 \leq posAlvo \leq tamLista$

A função retorna 1 no caso de sucesso e zero no caso de fracasso na remoção (inexistência do item alvo da remoção).

2. A **simulação** (teste de mesa) de exemplos demonstrando as funções implementadas.

## Entrega:

Upload via Moodle;

O documento de entrega terá que estar digitalizado, manuscrito e de autoria individual.

DICA:

- Se  $\text{posAlvo} == \text{posLogAtual}$ : `refMovel` está sobre o nó procurado;
- Se  $\text{posAlvo} < \text{posLogAtual}$ : `refMovel` é deslocado para a esquerda com deslocamento igual a  $\text{posLogAtual} - \text{posAlvo}$ ;
- Se  $\text{posAlvo} > \text{posLogAtual}$ : `refMovel` é deslocado para a direita com deslocamento igual a  $\text{posAlvo} - \text{posLogAtual}$ .