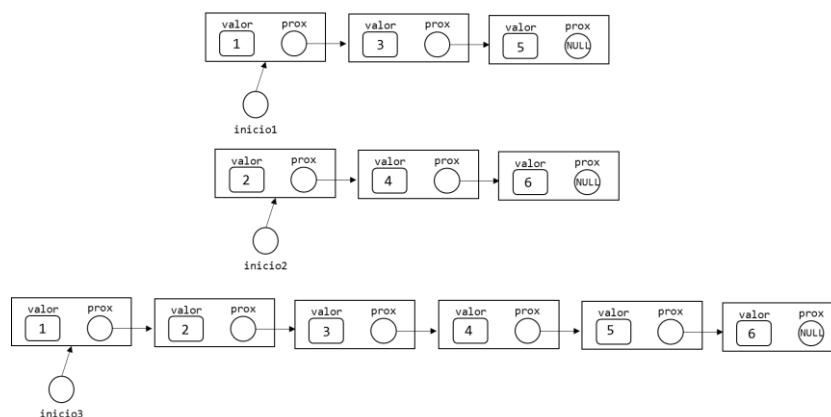


Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas
Disciplina: Estrutura de Dados 2 – 1s2021 – Prof. Vitor
EXERCÍCIOS RESOLVIDOS COM L.L.S.E.

1. Construa uma função em linguagem C que receba dois parâmetros: um ponteiro `inicio1` (por valor) para uma LLSE de valores inteiros e um ponteiro `inicio2` (por valor) para uma LLSE de valores inteiros. A função deverá retornar 1 (verdadeiro) caso ambas as LLSE sejam exatamente iguais ou 0 (falso), caso contrário. Por exatamente iguais entenda-se que as duas LLSE armazenam a mesma quantidade de nós e todos os nós armazenados em posições equivalentes de ambas as listas têm o mesmo conteúdo.
2. Construa uma função em linguagem C que receba três parâmetros: um ponteiro `inicio1` (por referência) para uma LLSE de valores inteiros, um ponteiro `inicio2` (por referência) para uma LLSE de valores inteiros e um ponteiro `inicio3` (por referência) que aponta para uma LLSE vazia. A função deverá mover todos os nós das LLSE apontadas por `inicio1` e `inicio2` (nessa ordem) para a LLSE apontada por `inicio3`. Ao término da execução da função, tanto `inicio1` quanto `inicio2` deverão apontar para NULL.
3. Construa uma função em linguagem C que receba dois parâmetros: um ponteiro `inicio` (por valor) para uma LLSE de valores inteiros e um ponteiro `novo` (por valor), que aponta para um nó que será inserido na LLSE apontada por `inicio`. A função deverá inserir o novo nó exatamente na posição equivalente à metade da quantidade de nós armazenada na LLSE. Assumir que a LLSE não está vazia e é par a quantidade de nós que ela armazena.
4. Construa uma função em linguagem C que receba três parâmetros: um ponteiro `inicio1` (por referência) para uma LLSE de valores inteiros, um ponteiro `inicio2` (por referência) para uma LLSE de valores inteiros e um ponteiro `inicio3` (por referência) que aponta para uma LLSE vazia. A função deverá mover todos os nós armazenados nas listas apontadas por `inicio1` e `inicio2` para a lista apontada por `inicio3` de maneira intercalada, ou seja, o primeiro nó da lista de `inicio1` será o primeiro nó da lista de `inicio3`, o primeiro nó da lista de `inicio2` será o segundo nó da lista de `inicio3`, o segundo nó da lista de `inicio1` será o terceiro nó da lista de `inicio3`, o segundo nó da lista de `inicio2` será o quarto nó da lista de `inicio3` e assim por diante. Assumir que as listas apontadas por `inicio1` e `inicio2` têm exatamente a mesma quantidade de nós armazenados. No encerramento da função as listas apontadas por `inicio1` e `inicio2` deverão estar vazias. O esquema a seguir auxilia na compreensão da lógica de execução da função.



5. Implemente novamente as funções solicitadas nas questões 2, 3 e 4 acima, com a diferença que elas deverão ser executadas em uma Lista Linear **Duplamente** Encadeada (LLDE), ou seja, uma lista onde cada nó contém dois ponteiros (`No *prox` e `No *anterior`): `prox` aponta para o nó sucessor (a frente em relação ao nó atual) e `anterior` aponta para o nó antecessor (atrás em relação ao nó atual).
6. Construa uma função que receba um ponteiro `inicio` (por referência) para uma LLDE (Lista Linear **Duplamente** Encadeada) de valores inteiros. Ela deverá encontrar o nó que armazena o maior valor dentro da lista e ***copiá-lo*** para o início da lista. Assume-se que existe somente um nó que contém o maior valor.
7. Construa uma função que receba um ponteiro `inicio` (por referência) para uma LLDE (Lista Linear **Duplamente** Encadeada) de valores inteiros. Ela deverá encontrar o nó que armazena o menor valor dentro da lista e ***movê-lo*** para o final da lista. Assume-se que existe somente um nó que contém o menor valor.
8. Construa e apresente uma função em linguagem C que execute as seguintes funcionalidades:
 - a) Receber dois parâmetros: um ponteiro `inicio` (por referência) para uma LLDE e um ponteiro `novo` (por valor) apontando para um nó que contém um número inteiro.
 - b) Fazer a inserção do nó apontado por `novo` na LLDE usando o seguinte raciocínio: caso o número seja menor que zero ele deverá ser inserido no início da lista; caso seja maior que zero deverá ser inserido no final da lista.
 - c) Caso o número seja 0 (zero), ele deverá ser inserido no meio da lista, entre os números menores e maiores do que ele.

Observações:

 - O número 0 (zero) poderá ser inserido somente uma vez na LLDE.
 - A função pode se utilizar de outras funções prontas para manipulação de LLDE que executem operações como, por exemplo, inserção de nós no início e final da lista.
9. Construa e apresente uma função em linguagem C que execute as seguintes funcionalidades:
 - a) Receber três parâmetros: um ponteiro `inicio` (por valor) para uma LLDE que armazena inteiros, um ponteiro `inicioNegativo` (por referência) que aponta para uma LLDE vazia que pode armazenar inteiros e um último ponteiro `inicioPositivo` (por referência) que também aponta para uma LLDE vazia que pode armazenar inteiros.
 - b) Copiar todos os nós apontados por `inicio` que armazenam números negativos para a lista apontada por `inicioNegativo`.
 - c) Copiar todos os nós apontados por `inicio` que armazenam números positivos para a lista apontada por `inicioPositivo`.
10. Implemente uma função que receba três parâmetros: um ponteiro `inicio` (por referência) para uma LLDE, um ponteiro `novo` (por valor) para um novo nó que vai ser inserido na LLDE e um inteiro `n` (por valor) que indica em qual posição da lista o novo nó será inserido (o novo nó deverá ser inserido na *n-ésima* posição da LLDE). A função deverá retornar 1 (verdadeiro), caso a inserção seja bem sucedida e 0 (falso), caso contrário. A inserção não será bem sucedida caso o valor de `n` seja negativo ou ainda, caso seu valor seja superior ao tamanho da LLDE (quantidade de elementos já armazenados).