

ORIENTAÇÕES PARA A ATIVIDADE ESTRUTURADA 5

TÍTULO DA ATIVIDADE ESTRUTURADA: Lista de Exercícios – Listas Encadeadas

OBJETIVO: Pesquisar no material indicado e no conteúdo de aula e responder o questionário proposto.

COMPETÊNCIAS/HABILIDADES: Conceituar listas lineares encadeadas Conceituar alocação dinâmica de memória Conceituar listas lineares simplesmente encadeadas Representar listas lineares simplesmente encadeadas Implementar operações com listas lineares simplesmente encadeadas, realizando aplicações.

DESENVOLVIMENTO DO TRABALHO:

Listas Lineares Simplesmente Encadeadas (não circulares)

Faça um programa em C++ com listas simplesmente encadeadas de inteiros, em que seja solicitada uma quantidade n de nós para a lista. Note que n deve ser um número maior ou igual a zero. Após esta entrada, faça o que se pede:

1) Escreva uma função em C++ para inserir um valor inteiro no início da lista. Esta função deverá ser chamada na main, de dentro de um loop, para que seja construída uma lista com n nós. Não se esqueça de inicializar a lista.

- Protótipo da função: `no* insereFrente(no * inicio, int valor);`
- Parâmetros: Ponteiro para o início da lista e o valor inteiro a ser inserido ou adicionado.
- Retorno: Ponteiro para o início da lista

2) Escreva uma função em C++ para percorrer uma lista simplesmente encadeada e imprimir na tela, todos os seus dados.

- Protótipo da função : `void imprimir(no * inicio);`
- Parâmetro: Ponteiro para o início da lista
- Retorno: Não há

3) Escreva uma função em C++ para substituir o valor de um nó passado, por um novo valor também informado. Considerando que o 1º. nó é o nó 1, o 2º. nó é o nó 2 e assim, sucessivamente, deverão ser passados como parâmetros : o ponteiro para o início da lista, o no. do nó que sofrerá a substituição e o novo valor. Note que, esta função não retornará valor algum. Considere que ao ser lido, na main, o no. do nó que sofrerá a substituição, este no. deverá ser criticado/testado para que a função somente seja chamada quando o número do nó for maior que zero e menor ou igual que a quantidade de nós da lista.

- Protótipo da função : `void substituir(no * inicio, int valor, int novoValor);`

4) Escreva uma função em C++ para remover o 1º. nó de uma lista simplesmente encadeada não vazia, eliminando-o. Teste, na main, se a lista está vazia e emita mensagem adequada para este caso.

- Protótipo da função : `no * removerInicio(no * inicio);`
- Parâmetro: ponteiro para o início da lista
- Retorno: ponteiro para a lista

Note que, neste exercício, não se está interessado no valor armazenado no nó removido, que deve ser eliminado.

5) Escreva uma função em C++ para realizar uma busca ou pesquisa sequencial na lista, retornando NULL (fracasso na busca) ou o endereço do nó com o valor encontrado (sucesso na busca).

- Protótipo da função : `no * busca(no * inicio, int valor);`
- Parâmetros: ponteiro para o início da lista e valor a ser procurado
- Retorno: NULL (fracasso na busca) ou o endereço do nó com o valor encontrado (sucesso na busca).

Note que, neste exercício, não se está interessado no valor armazenado no nó removido, que deve ser eliminado.

6) Faça uma função em C++ para contar o número de nós de uma lista simplesmente encadeada qualquer, ou seja, a lista pode estar vazia ou não.

- Protótipo: `int contaNos(no * inicio);`
- Retorno: A quantidade de nós da lista, que pode ser zero.

7) Escreva uma função em C++ para inserir um valor inteiro no fim da lista. Esta função deverá ser chamada na main de dentro de um loop, para que seja construída uma lista com m nós. Não se esqueça de ler a quantidade m de nós e de inicializar a lista. Note que, m deve ser maior ou igual a zero.

- Protótipo da função: `no* insereFim(no * inicio, int valor);`
- Parâmetros: Ponteiro para o início da lista e o valor inteiro a ser inserido (ou adicionado).
- Retorno: Ponteiro para o início da lista

8) Escreva uma função em C++ para remover o último nó de uma lista.

- Protótipo da função: `no* removeFim(no * inicio);`
- Parâmetros: Ponteiro para o início da lista e o valor inteiro a ser inserido (ou adicionado).
- Retorno: Ponteiro para a lista

9) Escreva uma função em C++ para concatenar as duas listas construídas até agora, colocando a lista com m nós no final da lista com n nós.

- Protótipo: `no *concatena(no * inicio1, no * inicio2);`
- Parâmetros: Ponteiro para o início da 1ª. lista (que pode ser NULL) e ponteiro para o início da 2ª. lista (que pode ser NULL).
- Retorno: Ponteiro para o início da lista resultante da concatenação

Após a chamada desta função na main, imprima os dados da lista resultante da concatenação.

PRODUTO/RESULTADO: Lista de exercícios respondida em um arquivo no formato Word (DOCX), entregue pelo sistema SIA (WebAula) de acordo com o PDF de Instruções e Informações Gerais disponível no site da disciplina (<http://www.caetano.eng.br>, na seção Ensino > Estrutura de Dados > Ativ. Estruturada).