Aula Prática: Lista Linear Estática/Sequencial

Objetivo: *Capacitar ao aluno desenvolver as operações básicas de listas lineares (ordenadas e não ordenadas), utilizando a implementação estática/sequencial.*

Exercícios:

1. Implementar, utilizando a alocação estática e o acesso seqüencial, o TAD lista linear não-ordenada de números inteiros definido na aula teórica. Nessa implementação a lista deve ter no máximo 10 elementos e deve contemplar as operações básicas: *inicializar_lista, lista_vazia, lista_cheia, insere_elem* e *remove_elem*. Além disso, desenvolva um programa aplicativo que permita ao usuário inicializar a lista, inserir e remover elementos da lista e imprimir a lista. Teste este programa com a seguinte seqüencia de operações:

    - Inicialize a lista;
    - Imprima a lista;
    - Insira os elementos {0,1,2,3,4,5,6,7,8,9,10};
    - Imprima a lista;
    - Remova o elemento 8;
    - Imprima a lista;
    - Inicialize a lista (a mesma lista);
    - Imprima a lista.

2. Complementar a TAD lista linear não-ordenada implementando a função remove_todos( ). Essa função recebe um valor inteiro (*Elem*) e deve remover todas as ocorrências de Elem em uma lista lista linear passada por referência. A função deve percorrer a lista uma única vez. Teste a nova função, repetindo a sequência de operações anterior, mas trocando a operação “Remova o elemento 8” por “Remova todas ocorrências do elemento 8” e a lista de entrada por {0,1,8,3,4,8,6,7,8,8}.

3. Refazer o exercício 1 para o TAD lista ordenada. Obs: Isto poderia ser realizado apenas com a modificação do arquivo com extensão c pertinente à implementação do TAD. Entretanto, por uma questão de padronização, crie novas operações: *insere_ord* e *remove_ord* e crie um novo projeto.Isto acarretará a necessidade de alteração também dos demais arquivos para adequação dos nomes das operações utilizadas (aplicativo) e inclusão dos protótipos para as novas funções (arquivo cabeçalho).

4. Complementar a TAD lista linear não-ordenada implementando a função intercala_ord( ). Essa função recebe duas listas ordenadas de inteiros (que podem ter tamanhos diferentes)e deve elaborar uma nova lista ordenada que seja a junção dos elementos das duas listas (deve-se percorrer as duas listas à medida que a lista 3 é elaborada). Teste este programa com a seguinte seqüência de operações:

- Inicialize a lista1 ;
- Insira os elementos na seguinte ordem: 4,0,7,1,9,5;
- Imprima a lista1;
- Inicialize a lista2 ;
- Insira os elementos na seguinte ordem: 8,2,6,3;
- Imprima a lista2;
- Gere a lista3 intercalando lista 1 e lista 2 de forma ordenada.
- Imprima a lista3;
- Inicialize a lista1;
- Gere a lista3 intercalando lista 1 e lista 2 de forma ordenada.
- Imprima a lista3;

5. Repita a implementação dos exercícios 1 e 3 para uma lista ordenada de strings (criar novos projetos).