Exercícios ordenação

1) Implementar os algoritmos de ordenação:
   a) Seleção
   b) Inserção
   c) Bolha

2) Implementar um algoritmo para ordenar um vetor em ordem decrescente (adaptar um dos algoritmos acima).

3) O que acontece no algoritmo de ordenação por inserção se o laço mais externo for iniciado em 0 em vez de 1? Faça o teste.

4) Implementar uma função que recebe um vetor de palavras e as ordena em ordem alfabética. Dica: usar a função *strcmp(str1, str2),* verificando o retorno:

0	indica que str1 e str2 são iguais.
negativo	indica que o primeiro caractere que não corresponde tem um valor ASCII menor em str1 do que em str2.
positivo	indica que o primeiro caractere que não corresponde tem um valor ASCII maior em str1 do que em str2.

5) A estrutura abaixo guarda o nome e o coeficiente de um aluno:

```
struct aluno {
        int ra;
        char nome[50];
        float coeficiente;
};
```

Implementar:
   a) Uma função que recebe um vetor de alunos e ordena o vetor em ordem alfabética.
   b) Uma função que recebe um vetor de alunos e ordena o vetor por coeficiente, em ordem crescente.

6) Faz sentido ordenar uma estrutura do tipo:
   a) Lista;
   b) Pilha;
   c) FIla.

7) Implementar uma lista encadeada para armazenar dados de alunos da UTFPR. Cada aluno deve ser representado conforme a estrutura apresentada no exercício 5. Criar as funções básicas da lista: criar lista vazia, inserir item, buscar item, remover item, imprimir a lista e liberar a lista. A função de inserir deve inserir um item de modo que a lista fique ordenada por ordem crescente de coeficiente.