

ORDENAÇÃO DE VETORES		ORDENAÇÃO DE VETORES
Qual a importância da Ordenação de Vetores ?		Não se pode considerar uma base de dados apenas como um repositório de dados, pois seria de pouca utilidade armazenar dados que não podem ser facilmente encontrados. Para trazer ordem ao caos foram criados os algoritmos de ordenação e de pesquisa.
Como são os Algoritmos de Ordenação ?		São Algoritmos que percorrem uma lista a fim de trazer algum resultado ordenado para o usuário. A eficiência de um método ou algoritmo de ordenação pode ser medida por duas variáveis: o número de comparações e o número de trocas e quanto maior forem esses números, mais ineficiente é o algoritmo.
Quais são os tipos de Algoritmos de Ordenação mais utilizados?		Existem 3 métodos gerais de ordenação: - Por Troca: Quando os valores são trocados um pelo outro até seguirem uma ordenação desejada, esse é o famoso "BubbleSort"; - Por Seleção: Ordenados por uma ordem de seleção específica, com ordem Alfabética, Numérica, Maior para o Menor e etc; - Por Inserção: Ordenados por ordem de chegada ao array;
Como se dá a Ordenação por Troca (Bolha - BubbleSort) ?		A ordenação por troca é o método mais básico de ordenação, e consiste em dois laços aninhados. Para facilitar considere os laços como laço i e laço j. O laço j é o interno e ele passa por cada item do vetor (do fim para o início) comparando o elemento da posição (j) com o próximo (j+1). Caso seja maior que o próximo, estes dois elementos são trocados. Após j ter passado por toda o vetor, o valor de i é incrementado e essa operação é repetida tantas vezes quantas sejam o número de elementos do vetor. A seguir é apresentada a função com o método de ordenação por troca: void Bubble_Sort (int vet[], int tam){
- reservada para a questão acima -		<pre>int i, j; int aux; for (i=1; i<tam;i++){ for (j=0;j<tam-i;j++){ if (vet[j] > vet[j+1]){ aux = vet[j]; vet[j] = vet[j+1]; vet[j+1] = aux; } } }</pre>
- reservada para a questão acima -		