

Lista 4
---------

1. Crie uma **função** que recebe um vetor e seu tamanho por parâmetro além de dois números inteiros  $i$  e  $j$  que são índices do vetor (ou seja  $0 \leq i, j \leq n$ ). A função deve então trocar os elementos das posições  $i$  e  $j$  entre si.
2. Faça um programa que lê um vetor de 30 inteiros e guarda o vetor na ordem inversa que foi lido em um outro vetor de saída.
3. Faça uma **função** que recebe um vetor de inteiros e seu tamanho como parâmetros, e ao final da execução da função o vetor esteja invertido. Utilize a seguinte idéia: troque os elementos da posição 0 e 29 entre si, depois da posição 1 e 28 etc. Pense bem no critério de parada.
4. Escreva uma **função** que recebe um vetor de inteiros e seu tamanho como parâmetros, e devolve a soma dos números pares deste vetor.
5. Faça uma **função** que recebe um vetor de números reais e o seu tamanho por parâmetro e devolve o desvio padrão dos números do vetor usando a seguinte fórmula:

$$\sqrt{\frac{1}{n-1} \left( \sum_{i=1}^n x_i^2 - \frac{1}{n} \left( \sum_{i=1}^n x_i \right)^2 \right)}$$

onde  $n$  é o número de elementos.

6. Modifique cada um dos seguintes algoritmos vistos em sala, para que estes ordenem um vetor em ordem **decrecente** de valor:
  - selectionSort
  - BubbleSort
7. Use cada um dos algoritmos abaixo e mostre passo-a-passo, como visto em aula, os passos para ordenar o vector (31, 41, 59, 26, 41, 58, 15, 19):
  - selectionSort
  - BubbleSort
8. Crie uma função para determinar o número total de inversões em um vetor **vet**. Uma inversão existe quando um elemento em uma posição  $i < j$  é tal que  $vet[i] > vet[j]$ . Por exemplo, no vetor (10, 4, 6, 1, 2) existem 4 inversões para o número 10, 2 inversões para o número 4, 2 inversões para o número 6, nenhuma inversão para 1, e nenhuma para o 2. Portanto o total de inversões é 8. Modifique o algoritmo *bubbleSort* para computar o número de inversões em um vetor.