



### Instruções para entrega do roteiro:

- Entregue o roteiro apenas no formato *.pdf* com o nome ***Y\_roteiroX.pdf***, onde **X** é o número do roteiro e **Y** é o número da sua matrícula. Não serão aceitos outros formatos.
- Inclua nome e matrícula, e mantenha a resolução dos exercícios **ordenada** e **legível**.
- Códigos completos (com `int main`), em texto, sem executáveis.  
**Para cada um, apresente uma imagem da tela de saída do seu programa.**
- Após a data de entrega, a nota da entrega é 0.
- Em caso de dúvidas, procurem os monitores. Haverá um monitor após as aulas de laboratório para tirar dúvidas sobre a lista.

## Roteiro 10

### Algoritmos de Ordenação (Parte 1)

**Data máxima de entrega: 16/06/2025 - 11:59h**  
**(Entrega: pelo SIGAA, na sua turma de laboratório.)**

## 1 Algoritmos de Ordenação

- 1.1 Faça um programa para ler um valor  $N$  e em seguida  $N$  inteiros, armazenando esses inteiros em um vetor. Em seguida, ordenar esses valores utilizando os três métodos de ordenação vistos: *SelectionSort*, *InsertionSort* e *BubbleSort*.
- 1.2 Modifique os algoritmos de ordenação anteriores, para que ordenem de forma decrescente os números (do maior para o menor) e teste os novos métodos.
- 1.3 Considerando os 3 métodos vistos, utilize um programa para ordenar grande quantidade de valores e verifique o aumento do número de comparações, movimentações e tempo de execução conforme se aumenta o volume de dados processados (veja os códigos da aula teórica).
  - Para tal, vamos utilizar os dados que estão no arquivo *ordenacao.zip*, disponível no sistema e drive. Ele contém vários arquivos com dados numéricos.
  - Cada arquivo começa com um número  $N$  que indica a quantidade de valores no arquivo, e em seguida, vem os  $N$  valores inteiros, que são os valores a serem ordenados.
  - O nome dos arquivos indica a quantidade de elementos e a disposição dos elementos:
    - ordenado: os valores já estão em ordem
    - contrario: os valores estão na ordem inversa
    - misturado: os valores estão distribuídos aleatoriamente
    - quaseordenado: a maioria dos valores está em ordem, apenas alguns estão fora do lugar

- 1.4 Modifique as funções que implementam o método **Seleção** e **Inserção** para fazer a ordenação de um vetor de struct do tipo Pessoa, que contém nome e idade. Conforme visto em sala de aula, implemente uma função de comparação de dois níveis, de forma que, ordene os valores pelo nome, e em caso de nomes iguais, ordene pela idade. Além disso, implemente cada método duas vezes, para cada um exercer uma ordenação crescente e também decrescente.

Link do roteiro (read-only) L<sup>A</sup>T<sub>E</sub>X: <https://pt.overleaf.com/read/frkvqpzzsxbc#1b0d2b>