

Trabalho 3-1

No primeiro trabalho da M3 o objetivo é explorar um pouco o uso de algoritmos de ordenação, utilizando alguns conceitos que já vimos anteriormente.

1. Criem uma struct Jogador, ela deve conter uma string nome, e um inteiro pontuação
 - a. Criem um vetor de struct do tipo Jogador, com até 1000 jogadores.
 - b. Eles devem ser preenchidos com valores aleatórios (os nomes não precisam fazer sentido)
 - c. Não deve existir um vetor de nomes e um vetor de pontos separados.
2. Devem existir três funções, `exibirJogadores()` e `melhoresPontuacoes10()` e `melhoresPontuacoes100()` :
 - a. `exibirJogadores` deve mostrar no console o nome de todos os jogadores ordenados em ordem alfabética.
 - b. `melhoresPontuacoes10()` deve mostrar no console as 10 melhores pontuações, ordenadas da melhor para a pior.
 - c. `melhoresPontuacoes100()` deve mostrar no console as 100 melhores pontuações, ordenadas da melhor para a pior e **jogadores com pontuações iguais devem ser aparecer ordenados em ordem alfabética.**
3. Para fazer a ordenação, vocês devem implementar 3 algoritmos de ordenação diferentes:
 - a. Cada função deve utilizar um algoritmo de ordenação diferente.
 - b. Um dos algoritmos deve ser escolhido entre: Insertion Sort ou Bubble Sort.
 - c. Outro algoritmo deve ser escolhido entre: Quick Sort ou Merge Sort.
 - d. O último algoritmo deve ser o Shell Sort (Não visto em aula).