INSTITUTO FED. DE EDUCAÇÃO, CIÊNC. E TEC. DE PERNAMBUCO
CURSO: TEC. EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
DISCIPLINA: ALGORITMOS E ESTRUTURAS DE DADOS
PROFESSOR: RAMIDE DANTAS
ASSUNTO: ORDENAÇÃO DE ARRAYS.

Prática 05

OBS.: Veja o código que acompanha a prática.

Parte 1: Implementando Algoritmos de Ordenação

Problema 1: No arquivo **ordenacao.h**, implemente **um** dos algoritmos com complexidade $O(N^2)$ estudados: Bubble Sort, Selection Sort, ou Insertion Sort.

Rode o **main.cpp** com arrays de tamanho pequeno e grande (10000 ou mais).

Desafio (opcional): Implemente os demais algoritmos.

Problema 2: No arquivo **ordenacao.h**, implemente **um** dos algoritmos com complexidade $O(NlogN)$ estudados: Merge Sort ou Quick Sort. Para o Merge Sort, implemente a função `merge()`; para o Quick Sort, a função `partition()`.

Rode o **main.cpp** e compare o desempenho com o algoritmo do **Problema 1**.

Desafio (opcional): Implemente o outro algoritmo.

Parte 2: Usando Algoritmos de Ordenação

Problema 1: Resolva o problema *Sort Colors* (LC0075) com um algoritmo de complexidade $O(N)$, usando o arquivo **colors.cpp** para testes.

O problema consiste em, dado um array (vector) de interios com números 0, 1, e 2 representando cores, retornar o array ordenado em ordem crescente.

ATENÇÃO: Não use a função `std::sort()` ou os algoritmos implementados antes. A solução para esse problema já foi vista num outro problema de uma prática passada.

Problema 2: Implemente a função `groupStrings()` em **strings.cpp**. Essa função deve retornar as strings agrupadas por tamanho, dado um conjunto inicial de strings.

Ordene as strings usando `std::sort()`, porém use um comparador customizado como parâmetro. Esse comparador é uma função que compara as strings apenas olhando seus tamanhos. Depois de ordenar, varra o array, colocando as strings de tamanho igual em grupos (vectors), e esses vectors serão colocados num vector maior contendo os grupos. Lembrando que strings de mesmo tamanhos serão consecutivas se o array estiver ordenado corretamente.

Desafio: Resolva o problema *Group Anagrams* (LC0049) em **anagrams.cpp**.

O problema consiste em, dado um conjunto de strings, agrupar as strings que são anagramas umas das outras. Anagramas tem em comum possuírem a mesma frequência de letras. A abordagem é a mesma do **Problema 2**: se as strings forem ordenadas baseadas na frequência de letras, aquelas que forem anagramas ficarão juntas. Depois basta percorrer o array ordenado, agrupando strings com as mesmas frequências de letras.