

**Universidade de São Paulo**  
**Instituto de Ciências Matemáticas e Computação**  
**Departamento de Ciências de Computação**  
**SCC0201 - Introdução à Ciência de Computação II**

**2022**

**Professor:** Fernando Pereira dos Santos

**Monitores:**

Adrio Oliveira Alves

Gabriel Natal Coutinho

João Guilherme Jarochinski Marinho

**1º Trabalho prático**

**Motivação (historinha):**

Ricardo sempre foi um grande amante de vinhos, então logo após terminar a escola, iniciou sua graduação em química no IQSC. Durante todo esse período, participou de uma pesquisa para catalogar as características químicas de mais de 500 vinhos e organizou-as em um único arquivo csv.

Com o fim da catalogação, Ricardo percebeu que não tinha estabelecido nenhum critério de organização dos vinhos catalogados. Dessa forma, se tornava muito lento o trabalho de verificar a existência de determinado valor para uma característica ou quantos vinhos possuíam o mesmo valor da característica buscada.

Com medo de perder todo seu intenso trabalho, Ricardo foi até o ICMC buscar ajuda de seus estimados estudantes para possibilitar esta tarefa tão crucial.

**Objetivo do trabalho:**

Desenvolva um programa em C que leia todos os dados de um arquivo csv e possa realizar busca e contagem dos vinhos com o valor da característica escolhida quantas vezes forem desejadas.

### Entrada:

A entrada que seu programa irá ler será organizada da seguinte forma:

- Nome do arquivo csv;
- Número de buscas sobre o arquivo;
- Nome (*string*) e valor (*double*) da característica selecionada (n vezes, de acordo com o número de buscas).
  - o As características que podem ser selecionadas são:  
**citric\_acid, residual\_sugar, density, pH, alcohol.**

### Saída:

A saída esperada para seu programa mostrará, para cada busca, todas as características (com **5 casas decimais** para os valores tipo double) do primeiro vinho (no vetor ordenado) que possui os parâmetros buscados e a quantidade de vinhos que também possuem o mesmo valor buscado.

Caso nenhum vinho tenha sido encontrado na busca, seu programa deverá imprimir “Nenhum vinho encontrado”.

**Exemplo (não representa um caso real, apenas para mostrar a formatação):**

#### Input:

```
vinho.csv
3
citric_acid 0.55
residual_sugar 4.1
pH 12.0
```

#### Output:

```
ID: 145, Citric Acid: 0.55000, Residual Sugar 1.80000, Density
0.99680, pH 3.17000, Alcohol 9.40000
```

```
Total de vinhos encontrados: 10
```

```
ID: 484, Citric Acid: 0.68000, Residual Sugar 4.10000, Density
0.99700, pH 3.06000, Alcohol 13.40000
```

```
Total de vinhos encontrados: 1
```

```
Nenhum vinho encontrado
```

## Dicas / Orientações importantes:

Ao longo da elaboração de seu código, leve em conta os seguintes passos para não acabar tendo problemas durante a escrita do programa e em seu resultado:

**1) Leitura dos dados:** Para familiarizá-lo, o arquivo csv que será lido pelo seu programa (da maneira que você já aprendeu em ICC1 para arquivos de texto) possui a primeira linha como cabeçalho (nome das características dos vinhos pesquisados) e as demais como os valores destas características para cada vinho em específico (todos como double, exceto o campo "id"). Além disso, os valores de cada linha estão separados por vírgula sempre.

Após separar a linha em strings dos valores, use a função **atof** da biblioteca *stdlib.h* para convertê-los a double. Veja mais informações sobre essa funcionalidade em [C library function - atof\(\)](http://tutorialspoint.com/C-library-function-atof/) ([tutorialspoint.com](http://tutorialspoint.com)).

**2) Estruturação dos dados:** Para guardar as características de cada vinho na memória, utilize de matrizes ou vetor de structs (para revisar melhor sobre como utilizar structs, visite [Vetor de Struct - Criando vetor de Registros em C - Embarcados](#) ou o pdf auxiliar sobre structs disponível junto com este trabalho).

**3) Busca dos vinhos:** Com os dados organizados em seu programa, a busca do vinho com o valor da característica indicada será pela **BUSCA BINÁRIA**, portanto a estrutura deverá ser ordenada antes de realizar a busca.

Após encontrar o valor desejado, verifique se ele é o primeiro dentre todos os outros iguais no vetor, caso contrário, busque o primeiro deles.

**4) Ordenação da estrutura:** Note que seu algoritmo de ordenação trabalhará com estruturas maiores que apenas um simples número como elemento para ordenar, portanto tome cuidado principalmente na troca entre os elementos no vetor. Além disso, para o critério de desempate de chaves de mesmo valor na ordenação, use a característica "id", mantendo uma ordem crescente deles.

**5) Algoritmo de ordenação:** Neste trabalho, exigimos um algoritmo de ordenação próprio e diferente dos demais aprendidos em sala. Este algoritmo seguirá apenas 3 passos:

- Percorra todo o vetor de tamanho **n** buscando a localização da maior dentre todas as chaves (considerando o critério de desempate falado acima caso tenha chaves iguais).
- Após localizada a maior chave, troque-a com o último elemento do vetor.
- Repita esses 2 passos agora com os **n-1** primeiros elementos, trocando a maior chave com o penúltimo elemento. Faça isso até restar apenas 1 elemento na interação.

### **Exigências para o trabalho:**

É de extrema importância cumprir TODOS estes itens para seu trabalho:

- Escreva um cabeçalho no topo de seu código, indicando nome, número USP e nome da disciplina pelo menos;
- Siga todas as 5 orientações propostas neste trabalho. Algumas são apenas dicas para te ajudar, mas outras serão levadas em conta na correção de seu código;
- Comente devidamente seu código;
- Divida seu programa em funções, deixando-o mais organizado;
- Não plagie seu trabalho. Caso for identificado cópia entre 2 ou mais trabalhos, todos envolvidos receberão nota 0 neste.