

Problema: Análise de Histórico Escolar

Você deverá escrever um programa que leia o histórico escolar de um aluno a partir de um arquivo CSV. Cada linha do arquivo representa uma disciplina, contendo o nome da disciplina, a nota final (0 a 10) e a frequência percentual (0 a 100). O programa deve calcular:

1. O **coeficiente de rendimento (CR)** do aluno: média aritmética simples das notas em que a frequência for **maior ou igual a 75%**;
2. A **disciplina com a melhor nota**;
3. A **disciplina com a pior nota**;
4. Uma **lista de todas as disciplinas com frequência inferior a 75%**, exibidas na ordem em que aparecem no arquivo.

Entrada

O programa recebe como entrada **apenas** o nome de um arquivo CSV.

Cada linha do arquivo segue o formato:

```
nome_da_disciplina,nota,frequencia
```

Restrições:

- `nome_da_disciplina` : string sem vírgulas, com no máximo 50 caracteres;
- `nota` : número real entre 0.00 e 10.00 (inclusive);
- `frequência` : número real entre 0.00 e 100.00 (inclusive);
- Máximo de 100 disciplinas no arquivo.

O arquivo **não contém acentos**, apenas letras sem acentuação, vírgulas e quebras de linha.

Exemplo de entrada:

```
historico.csv
```

Conteúdo de `historico.csv` :

Matematica,8.5,90
Historia,7.0,80
Fisica,5.5,60
Quimica,9.2,85
Filosofia,6.8,74

Saída

Coeficiente de Rendimento: 7.88
Melhor Disciplina: Quimica (nota 9.2)
Pior Disciplina: Fisica (nota 5.5)
Disciplinas com frequencia inferior a 75%:
- Fisica (frequencia 60.0)
- Filosofia (frequencia 74.0)

Cr terios

- O **CR** considera **apenas** disciplinas com frequ ncia $\geq 75\%$.
- Em **empates** (melhor ou pior nota), imprimir a **primeira ocorr ncia** no arquivo.
- Todos os valores num ricos devem ter **duas casas decimais**.

Dicas de Implementa  o

- Voc  pode criar uma `struct Disciplina` com os campos: `nome`, `nota` e `frequ ncia`.
- Voc  pode usar um vetor para armazenar os dados lidos do arquivo (ou talvez n o?).
- Para leitura direta de cada linha do CSV, voc  pode utilizar `fscanf()` :

```
char nome[??];  
float nota, freq;  
  
while (fscanf(fp, "%50[^,],%f,%f\n", nome, &nota, &freq) == 3) {  
    // Armazene os dados em um vetor de struct  
}
```

- Voc  pode (ou n o) usar um vetor para:
 - Calcular o CR apenas para disciplinas com frequ ncia $\geq 75\%$.

- Determinar a melhor e a pior nota.
- Listar disciplinas com frequência $< 75\%$ mantendo a ordem do arquivo.

A escolha de utilizar ou não vetor depende de sua decisão.