

Recursão e Ordenação

O professor Hilario, de idade já avançada, leciona a disciplina “Programação II” há algumas décadas e já teve centenas de milhares de alunos. Ele sempre aplica 2 provas valendo 30 pontos e 1 trabalho valendo 40. Curioso, Hilario decidiu fazer algumas estatísticas a respeito das notas ao longo das várias décadas em que ele dá aula. Ele armazena uma planilha com todas as notas dos alunos desde os primórdios. A planilha contém:

- A matrícula do aluno.
- O nome do aluno.
- O semestre letivo.
- A nota do aluno em cada uma das 3 avaliações.
- O número de faltas do aluno.

A nota final do aluno é a soma das notas das 3 avaliações. Caso não tenha faltado nenhuma aula, ele ganha até 2 pontos de bônus (desde que o total não passe de 100).

Com preguiça calcular as tais estatísticas, ele resolveu passar isso como o trabalho prático desse semestre.

Sua missão será a seguinte:

1. Leia o arquivo binário “entrada.bin” fornecido pelo professor, contendo a planilha de notas no formato de um dicionário em Python. A chave do dicionário é a matrícula do aluno, e o conteúdo é uma tupla contendo o nome, o semestre letivo (tupla com dois inteiros representando o ano e o semestre), as notas do aluno (tupla com 3 inteiros representando, respectivamente, a nota da primeira prova, a nota do trabalho e a nota de segunda prova) e um inteiro com o número de faltas.
2. Crie uma lista contendo **apenas** os números de matrícula dos alunos. Ordene essa lista de acordo com os seguintes critérios, nesta ordem:
 - (a) Nota total (da maior para a menor).
 - (b) Em caso de empate, pela nota sem o bônus.
 - (c) Em caso de novo empate, pelo semestre letivo (do mais recente para o mais antigo).
 - (d) Em caso de novo empate, em ordem alfabética do nome.
 - (e) Caso ainda haja empate, em ordem crescente de matrícula (não existem duas matrículas iguais).

A ordenação deve utilizar uma única função de comparação que recebe duas matrículas m_1 e m_2 , além do dicionário de alunos, e retorna:

- Verdadeiro, caso o aluno da matrícula m_1 deva aparecer na lista ordenada **antes** do aluno da matrícula m_2 .
 - Falso, caso contrário.
3. Crie um arquivo texto chamado “saida.txt” contendo o resultado da ordenação. O arquivo deve conter, em cada linha, o nome do aluno (na ordem desejada pelo professor), sua nota (sem o bônus) e o bônus (caso haja). Exemplo:
- ```
Maria da Silva - 100
João Pedro Moraes - 98 +2
José Ferreira - 99
Bianca de Souza - 78 +2
```
4. Com o auxílio da busca binária (você pode adaptá-la se necessário), imprima **apenas** um inteiro na tela indicando quantos alunos já foram aprovados pelo professor (ou seja, tiraram ao menos 60 pontos, mesmo que com auxílio do bônus). **Atenção:** não imprima mais nada na tela. Absolutamente nada! E não salve nada no arquivo texto além do que foi pedido.

A ordenação deve ser feita com algum dos algoritmos recursivos de ordenação ensinados nesta disciplina que executem em  $O(n \cdot \lg n)$  no caso médio, e deve ser executada uma única vez (ou seja, uma única ordenação já deve ser capaz de classificar os alunos de acordo com todos os critérios citados acima).

## Observações

- O trabalho vale 40 pontos, pode ser feito em grupo de até 2 integrantes e deve ser entregue até **16 de novembro**.
- Trabalhos considerados **plágio** terão nota 0 para quem copiou e para quem forneceu o trabalho. Além disso, serão enviados para o Conselho de Ética.
- O código deve ser feito em Python3.
- Trabalhos entregues após o prazo, com erro de execução, com formato de saída incorreto, ou que não compilarem terão nota 0.
- O trabalho deve ser enviado na sala da disciplina do AVA.
- Em caso de dúvidas na especificação do trabalho ou no próprio trabalho, contate-me em [hsjunioe@gmail.com](mailto:hsjunioe@gmail.com)