

### 3ª Lista de Exercícios

1. Para criar um registro de tamanho fixo em uma linguagem como C, podemos usar uma *struct* com campos de tamanho fixo. Como podemos criar registros de tamanho fixo em Python?
2. Registros de tamanho variável podem ser implementados de várias maneiras, entre elas, utilizando-se (a) registros com um número fixo de campos, (b) colocando uma indicação de tamanho no início do registro (c) utilizando metadados. Qual a vantagem de cada opção?
3. Por que a leitura de registros em blocos pode fazer com que o tempo de transferência aumente na busca por um registro?
4. (a) Quantas leituras seriam feitas em média para se encontrar um registro usando busca sequencial em um arquivo contendo 10.000 registros? (b) Se o registro procurado não existe no arquivo, quantas leituras serão feitas? (c) Se o arquivo for organizado em blocos de 20 registros, quantos acessos serão necessários em média?
5. O que caracteriza o acesso sequencial e o que caracteriza o acesso direto? Ambos os tipos acesso são possíveis em qualquer arquivo?
6. A figura abaixo representa a saída de um editor hexadecimal (*dump* de arquivo), que descreve os primeiros bytes de um arquivo do mesmo tipo que é produzido pelo programa *escreve\_registros.py* (Atividade Prática 1). A última coluna (mais à direita, onde aparecem os pontinhos) foi propositalmente ocultada. Utilize um editor hexadecimal e responda: (a) Qual é o tamanho do 1º registro do arquivo? (b) Qual é o seu conteúdo?

00244475	6D707C46	7265647C	38323120	.....
4B6C7567	657C4861	636B6572	7C50417C	.....
36353533	357C0000	00000000	00000000	.....

7. *Suponha* que você tenha um conjunto de campos de tamanho fixo cuja soma dos tamanhos resulta em 30 bytes, ou seja, registros de 30 bytes seriam suficientes para armazená-los. Se esses registros forem ser armazenados em um disco com setores de 512 bytes, poderia ser interessante utilizar registros de 32 bytes de tamanho em vez de 30. Qual seria a vantagem nesse caso?
8. Reescreva os programas *escreve\_registros.py* e *le\_registros.py* (Atividade Prática 1) de modo que eles usem registros de tamanho fixo de 128 bytes.
9. Escreva um programa similar ao programa *busca\_seq.py* (Atividade Prática 2) que implemente as seguintes alterações: (a) em vez ler chaves de busca a partir do teclado, o programa deve lê-las a partir de um arquivo de transações que contém somente as chaves dos registros que devem ser buscados; (b) em vez de imprimir os registros buscados na tela, os escreva em um arquivo de saída. Considere que tanto o arquivo de dados quanto o arquivo de transação estão desordenados.