

Computação I - Python

Laboratório 9

Seguindo com nossas boas práticas, para cada um dos exercícios a seguir:

- antes de começar a escrever código, faça o estudo do problema e o planejamento de sua solução.
- lembre de botar a **documentação**:

```
'''Calcula e retorna a divisão de a por b;  
int, int -> float'''
```

- escolha **nomes elucidativos** para suas funções e parâmetros;
- pense em **valores de teste** relevantes para testar sua função.
- recorra ao **teste de mesa** quando estiver com dificuldade para entender algum erro de funcionamento ou resultado inadequado de sua função.
- **para fazer a entrega desta atividade prática, escreva suas funções na ferramenta Machine Teaching.**

Vamos lá!

1. Faça uma função booleana chamada **eh_quadrada** para identificar se uma matriz é quadrada. Observação: uma matriz vazia (sem nenhuma linha nem coluna) é considerada quadrada.
2. Faça uma função chamada **conta_numero** que dado um número inteiro e uma matriz de inteiros de tamanho qualquer, conta e retorna quantas vezes aquele número aparece na matriz.
3. Faça uma função chamada **media_matriz** que dada uma matriz de inteiros não vazia, retorna a média de todos os números da matriz (com exatamente duas casas decimais de precisão).
4. Uma pista de Kart permite 10 voltas para cada um dos 6 corredores. Faça uma função chamada **melhor_volta** que receba como entrada uma matriz 6×10 com os tempos em segundos dos corredores em cada volta. A função deve retornar uma tupla informando: De quem foi a melhor volta da prova, com qual tempo e em que volta. Assuma que os corredores tem tempos diferentes. (dica: usar a função *min*)
5. Suponha que os dados de funcionários de uma empresa sejam armazenados em uma matriz como a do exemplo a seguir:

| | | | |
|----------------------|-----------|-------------------|--------------------|
| 'AdalbertoFerreira' | '1091982' | 'Contabilidade' | '(21)99281 – 2983' |
| 'JulianaVasconcelos' | '1111722' | 'RecursosHumanos' | '(21)99848 – 1902' |
| 'FlaviaAmorim' | '1128938' | 'Contabilidade' | '(22)99273 – 9404' |

Cada linha da matriz tem quatro entradas, representando as informações referentes a nome, registro, setor e telefone de um funcionário, nesta ordem. O número de linhas depende da quantidade de funcionários. Todas as entradas da matriz estão em formato string.

Escreva uma função chamada **busca** que receba uma string e uma matriz como a do exemplo e faça uma busca por setor, ou seja, dado o nome de um setor da empresa, a função retorna uma lista com os dados de todos os funcionários daquele setor.

Por exemplo: `buscaSetor('Contabilidade',P) = [['Adalberto Ferreira', '1091982', '(21)99281-2983'], ['Flavia Amorim', '1128938', '(22)99273-9404']]`

onde P é a matriz do exemplo acima. Se nenhum registro for encontrado, a função deverá retornar uma lista vazia.

6. **Insertion sort**, ou **ordenação por inserção**, é um simples algoritmo de ordenação, eficiente quando aplicado a um pequeno número de elementos. Em termos gerais, ele percorre um vetor de elementos da esquerda para a direita e à medida que avança vai deixando os elementos mais à esquerda ordenados. O algoritmo de inserção funciona da mesma maneira com que muitas pessoas ordenam cartas em um jogo de baralho (*fonte: wikipedia*)

Não é um método muito eficiente porém sua implementação é simples: supondo que queremos ordenar uma lista de números em ordem crescente. Inicialmente verificamos se o segundo elemento da lista é menor que o primeiro. Caso seja, trocamos eles de posição. Como à esquerda do segundo elemento da lista, só temos o primeiro elemento, nada mais precisa ser feito.

Passamos então para o terceiro elemento da lista. Temos que compará-lo com o segundo elemento e trocá-los de posição caso o terceiro seja menor. Caso tal troca seja feita, devemos comparar o novo valor que está na segunda posição com o valor da primeira, trocando eles de lugar quando for necessário. Se o terceiro elemento não for trocado com o segundo, isso indica que toda a parte inicial da lista (do primeiro ao terceiro elementos) já está ordenada, e devemos portanto passar para o quarto elemento. Este processo vai se repetindo até atingirmos o último elemento da lista.

Para entender melhor a idéia, assista aos vídeos:

<https://www.youtube.com/watch?v=R0a1U379l3U>

<https://youtu.be/-Z00it6Nkz8>

Implemente uma função **ordena_por_insercao** que recebe uma lista e a ordena em ordem crescente, utilizando o método de ordenação por inserção.