

Computação I - Python

Laboratório 9

1. Faça uma função que dada uma matriz de inteiros de tamanho qualquer e um número inteiro, conta e retorna quantas vezes aquele número aparece na matriz.
2. Faça uma função que, dada uma matriz qualquer, gera e retorna a matriz transposta.
3. Faça uma função que dada uma matriz de inteiros, retorna a média de todos os números da matriz.
4. Uma pista de Kart permite 10 voltas para cada um dos 6 corredores. Faça uma função que receba como entrada uma matriz 6 x 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. (use a função *min*)
5. Suponha que os dados de funcionários de uma empresa sejam armazenados em uma matriz como a do exemplo a seguir:

$$\begin{pmatrix} 'AdalbertoFerreira' & '1091982' & 'Contabilidade' & '(21)99281 - 2983' \\ 'JulianaVasconcelos' & '1111722' & 'RecursosHumanos' & '(21)99848 - 1902' \\ 'FlaviaAmorim' & '1128938' & 'Contabilidade' & '(22)99273 - 9404' \end{pmatrix}$$

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

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

em que P é a matriz do exemplo acima. Se nenhum registro for encontrado, a função deverá retornar a mensagem 'Nenhum registro encontrado'.

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=R0a1U37913U>

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

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