

Ifes Campus Serra
BSI – Bacharelado de Sistemas de Informação
Programação II

Prof. Ernani L. R. Filho

Importante: ao construir as respostas, sempre modularize o seu código com funções mais programa principal.

Enunciados

1. Tabelas Veiculos x Proprietarios

Crie 2 tabelas nos moldes do que foi discutido em aula (listas internas são as linhas da tabela). A primeira tabela, Veiculos, possui como colunas placa, modelo, marca, cpf do proprietario. A segunda tabela, Proprietarios, possui como colunas cpf, nome, e-mail, celular do proprietario.

Alimente ambas as tabelas por meio de redirecionamento de entrada a partir dos arquivos de dados fornecidos pelo Professor. Ao final, construa um relatório de Proprietários x Veiculos contendo para cada Nome, e-mail de proprietário a lista de placas, modelo, marca de veículos que ele possui.

2. Tabelas: Criação de Nomes

Crie 2 listas nos moldes do que foi discutido em aula. A primeira lista, Nomes, contém 20 dos nomes de pessoas mais frequentes no Brasil. A segunda lista, Sobrenomes, possui uma lista dos 15 sobrenomes mais frequentes no Brasil.

Construa uma função que recebe como parâmetros as duas listas e um número inteiro representando o tamanho do nome da pessoa em partes. Por exemplo, 4: 1 nome + 3 sobrenomes. A função deve retornar um nome de pessoa contendo uma quantidade de partes definida pelo terceiro parâmetro.

Observação: O nome de pessoa não deve conter sobrenomes repetidos.

3. Matrizes

Crie um arquivo em Python chamado **libmatriz.py**. Neste arquivo, crie 2 funções: loadMatriz(nome_arquivo) e salvaMatriz(mat, nome_arquivo). A função loadMatriz deve processar o arquivo de nome nome_arquivo carregando o seu conteúdo e retornando-o como uma matriz. A função salvaMatriz deve salvar a matriz mat em um arquivo de nome nome_arquivo.

Observação: Todos os arquivos de matrizes, para leitura ou para escrita, devem estar organizados no formato de matrizes, cada linha do arquivo é uma linha da matriz com os respectivos elementos separados por espaço.

Utilizando as funções loadMatriz(..) e salvaMatriz(..), crie as funções mostradas nas figuras 1 e 2 a seguir. Armazene essas funções no arquivo libmatriz.py. Teste cada uma das funções criando uma aplicação de testes.

Bons estudos!

Algumas funcionalidades da Interface do TAD Matriz (parte I)

a) `extraí(Matriz,1,2,3,3)`

Matriz						Extraída		
0	1	2	3	4	5	0	1	2
0	5	4	1	8	3	6	5	6
1	3	0	5	6	7	4	2	8
2	9	1	2	8	4	3	9	0
3	4	5	1	9	0	2		
4	9	0	7	4	1	3		

b) `insere(Matriz,1,2,Inserida)`

Matriz						Inserida		
0	1	2	3	4	5	0	1	2
0	5	4	1	8	3	6	1	1
1	3	0	1	1	1	4	1	1
2	9	1	1	1	1	3	1	1
3	4	5	1	1	1	2		
4	9	0	7	4	1	3		

c) `deslocaEsq(Matriz)`

Matriz						Matriz deslocada à esquerda					
0	1	2	3	4	5	0	1	2	3	4	5
0	5	4	1	8	3	6	4	1	8	3	6
1	3	0	5	6	7	4	0	5	6	7	4
2	9	1	2	8	4	3	1	2	8	4	3
3	4	5	1	9	0	2	5	1	9	0	2
4	9	0	7	4	1	3	0	7	4	1	3

d) `deslocaDir(Matriz)`

Matriz						Matriz deslocada à direita					
0	1	2	3	4	5	0	1	2	3	4	5
0	5	4	1	8	3	6	0	5	4	1	8
1	3	0	5	6	7	4	0	3	0	5	6
2	9	1	2	8	4	3	0	9	1	2	8
3	4	5	1	9	0	2	0	4	5	1	9
4	9	0	7	4	1	3	0	9	0	7	4

Figura 1

Algumas funcionalidades da Interface do TAD Matriz (parte II)

e) `rotDir(Matriz)`

Matriz						Matriz rotacionada à direita					
0	1	2	3	4	5	0	1	2	3	4	5
0	5	4	1	8	3	6	6	5	4	1	8
1	3	0	5	6	7	4	4	3	0	5	6
2	9	1	2	8	4	3	3	9	1	2	8
3	4	5	1	9	0	2	2	4	5	1	9
4	9	0	7	4	1	3	3	9	0	7	4

f) `rotEsq(Matriz)`

Matriz						Matriz rotacionada à esquerda					
0	1	2	3	4	5	0	1	2	3	4	5
0	5	4	1	8	3	6	4	1	8	3	6
1	3	0	5	6	7	4	0	5	6	7	4
2	9	1	2	8	4	3	1	2	8	4	3
3	4	5	1	9	0	2	5	1	9	0	2
4	9	0	7	4	1	3	0	7	4	1	3

g) `vizinhos(Matriz,1,2)`

Matriz						Vizinhos de (1,2)							
0	1	2	3	4	5	0	1	2	3	4	5	6	7
0	5	4	1	8	3	6	4	1	8	6	8	2	1
1	3	0	5	6	7	4							
2	9	1	2	8	4	3							
3	4	5	1	9	0	2							
4	9	0	7	4	1	3							

Figura 2

Figura 1