

Lista 9

Matrizes

Questão 1

Assinale as alternativas **corretas**?

- (a) Matrizes em **Python** são listas de listas.
- (b) O primeiro elemento da matriz **A** se encontra na linha 1 e coluna 1 e, para acessar esse elemento, utiliza-se o seguinte comando: **A[1][1]**.
- (c) Uma matriz denominada **A** de **l** linhas e **c** colunas terá o último elemento na linha **l-1** e coluna **c-1** e, para acessar esse elemento, utiliza-se o seguinte comando: **A[l-1][c-1]**.
- (d) O número de linhas de uma matriz **A** pode ser recuperado com o comando **len(A)**.
- (e) O número de colunas de uma matriz **A** pode ser recuperado com o comando **len(A[0])**, assumindo que **A** tem pelo menos uma linha.
- (f) Em **Python** todas as linhas de uma matriz precisam ter o mesmo número de colunas.

Questão 2

Escreva uma função **cria_matriz(l, c)** que receba como parâmetro o número de linhas **l** e o número de colunas **c** de uma matriz **M** e leia do teclado os valores informados pelo usuário na matriz $M_{l \times c}$.

Questão 3

Escreva uma função **imprime_matriz(M, l, c)** que receba uma matriz $M_{l \times c}$ e imprima a matriz linha por linha com os valores separados por **vírgula** e tab.

Exemplo:

```
1 M = [[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]]
2 imprime_matriz(M, 4, 3)
3 >>>
4 1, 2, 3
5 4, 5, 6
6 7, 8, 9
7 10, 11, 12
```

Questão 4

O quê a função tarefa irá imprimir com a execução do código abaixo?

```
1 def tarefa(mat):
2     dim = len(mat)
3     for i in range(dim):
4         print(mat[i][dim-1-i], end=" ")
5
6 mat = [[1,2,3],[4,5,6],[7,8,9]]
7 tarefa(mat)
```

Questão 5

Escreva uma função `dimensoes(matriz)` que recebe uma matriz como parâmetro e imprime as dimensões da matriz recebida, no formato `iXj`.

Exemplos:

```
1 minha_matriz = [[1], [2], [3]]
2 dimensoes(minha_matriz)
3 3X1
```

```
1 minha_matriz = [[1, 2, 3], [4, 5, 6]]
2 dimensoes(minha_matriz)
3 2X3
```

Questão 6

Escreva uma função em **Python** que receba 2 matrizes, mostre-as na tela e calcule a soma entre as duas matrizes caso as matrizes tenham dimensões iguais. Caso contrário, a função deve devolver `False`.

Relembrando...

$$\begin{matrix} & A & & B & & C \\ \begin{bmatrix} 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \end{bmatrix} & + & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \end{bmatrix} & = & \begin{bmatrix} 0 & 1 & 0 & 2 & 3 \\ 1 & 2 & 1 & 3 & 4 \\ 0 & 1 & 0 & 2 & 3 \\ 2 & 3 & 2 & 4 & 5 \\ 3 & 4 & 3 & 5 & 6 \end{bmatrix} \end{matrix}$$

Exemplos:

```
1 m1 = [[1, 2, 3], [4, 5, 6]]
2 m2 = [[2, 3, 4], [5, 6, 7]]
3 soma_matrizes(m1, m2)
4 >>> [[3, 5, 7], [9, 11, 13]]
```

Questão 7

Duas matrizes são multiplicáveis se o número de **colunas da primeira** é igual ao número de **linhas da segunda**. Escreva a função `sao_multiplicaveis(m1, m2)` que recebe duas matrizes como parâmetro e devolve `True` se as matrizes forem multiplicáveis (na ordem dada) e `False` caso contrário.

Exemplos:

```
1 m1 = [[1, 2, 3], [4, 5, 6]]
2 m2 = [[2, 3, 4], [5, 6, 7]]
3 sao_multiplicaveis(m1, m2)
4 >>> False
```

```
1 m1 = [[1], [2], [3]]
2 m2 = [[1, 2, 3]]
3 sao_multiplicaveis(m1, m2)
4 >>> True
```

Questão 8

Escreva uma função que receba uma matriz `A` de números inteiros de dimensão $l \times c$ e retorne o número de linhas e colunas que tem apenas zeros.

Exemplo:

```
1 M = [[0, 0, 0, 0, 1],
2      [0, 0, 0, 0, 0],
3      [0, 1, 0, 0, 0],
4      [0, 0, 0, 0, 0]]
5
6 contagem_zeros(M, 4, 5)
7
8 >> Linhas nulas = 2
9 >> Colunas nulas = 3
```

Questão 9

Escreva uma função em **Python** que receba todos os elementos de uma matriz `A` com dimensões $l \times c$ e retorne a sua transposta A^T .

Relembrando...

Matriz	Transposta
$\begin{bmatrix} 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 \end{bmatrix}$

Questão 10

Escreva uma função em **Python** que receba todos os elementos de uma matriz **A** com dimensões $l \times c$ e retorna **True** se **A** for simétrica, e **False**, caso contrário.

Pre-condicao: você pode assumir que a matriz **A** é quadrada.

Exemplo:

```
1 A = [[1, 2, 3],
2       [2, 3, 4],
3       [3, 4, 7]]
4
5 >>> simetrica(A, 3, 3)
6 >>> True
```