

Lista de Exercícios 2: *entrega → 26/11 (entregar soluções em PAPEL, pode fazer em grupo)*

(1) – Escreva o trecho de código Python que consiga descobrir se uma matriz *mat* de números reais, já armazenada em uma lista 2d em memória, possui algum número acima de 1 milhão.

(2) – Faça um programa que **leia do teclado para uma lista 2d** uma matriz quadrada real *A*, de ordem *m* ($m \leq 50$, pergunte ao usuário a dimensão antes do início da digitação e rejeite se ele digitar *m* acima de 50). A seguir, verifique se a matriz é simétrica, ou seja, se $A[i][j] = A[j][i]$ para todo $i, j < m$. Imprima na tela a palavra “SIMÉTRICA” se a matriz *A* for simétrica e “NÃO-SIMÉTRICA” caso contrário.

(3) – Construa um programa Python que, utilizando uma lista 2d, **crie e preencha automaticamente uma matriz 8x8**, onde a primeira célula contenha o valor 1 e as células seguintes o dobro da anterior. Ou seja, a matriz deve possuir o valor 1 na primeira célula (célula [0][0]), valor 2 na segunda célula (célula [0][1]), valor 4 na terceira célula (célula [0][2]), valor 8 na quarta célula (célula [0][3]) e assim sucessivamente até a sexagésima-quarta célula (célula [7][7]).

(4) – Crie uma função “f_maior_menor” que deverá funcionar da seguinte maneira:

- A função **deverá** possuir 1 **parâmetro**: uma matriz, estruturada em uma lista 2d.
- Ela deverá executar, em sequência, as seguintes ações:
 - **Imprimir a matriz** passada como entrada, utilizando o “layout bonito”, como mostrado nos slides 23 a 25 da aula 11.
 - **Encontrar e imprimir o maior elemento da linha onde se encontra o menor elemento da matriz.**
 - **Imprimir também a posição** (ou seja a linha e coluna) deste elemento.
- Abaixo demonstra-se um exemplo de funcionamento de “f_maior_menor” para uma matriz 3x2 (porém sua função deve funcionar para qualquer matriz numérica de qualquer tamanho).

```
u = [ [1000, 33], [90, 100], [7, 8] ]
```

```
f_menor_maior(u)
```

resultado que deve ser impresso por f_menor_maior nesse caso:

```
| 1000    33 |
|   90   100 |
|    7     8 |
```

Maior elemento da linha onde se encontra o menor elemento: 8

Posição deste elemento: [2][1]

(5) – Considere uma matriz A com m linhas e n colunas estruturada em uma lista 2d.

- (a) Escreva o trecho de código capaz de determinar o número de **linhas nulas** da matriz
- (b) Escreva o trecho de código capaz de determinar o número de **colunas nulas** da matriz

Por exemplo, se:

```
A = [[2, 0, 6, 2, 7],
      [1, 0, 5, 9, 8],
      [0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0]]
```

Temos 2 linhas nulas (resposta do item a) e 1 coluna nula (resposta do item b)

(6) – Escreva uma função chamada “transforma_em_2d” que receba como entrada uma lista contendo n elementos.

- Se $n < 2$, a função deverá retornar None
- Caso contrário, deverá retornar uma lista 2d, contendo 2 colunas e $n/2$ linhas, da forma mostrada nos exemplos a seguir:
- se: `lst = [2, 0, 6, 2, 7, 10]`
- a função deve retornar: `[[2, 0], [6, 2], [7, 10]]`
- se: `lst = [2, 0, 6, 2, 7]`
- a função deve retornar: `[[2, 0], [6, 2]]`
 - Veja que neste caso, como a lista de entrada possui um número ímpar de elementos, o último foi descartado.

(7) - Crie uma **função booleana** chamada “testa_lista” capaz de indicar se uma lista de inteiros w com n elementos é da forma $w = [x, 2x, 4x, 8x, \dots]$, ou seja, uma lista onde o valor do elemento de uma posição $i+1$ é o dobro do valor do elemento da posição i , para qualquer i .

- A função deve receber como entrada w .
- Como saída, deve retornar True, se w for da forma $w = [x, 2x, 4x, 8x, \dots]$ ou False caso contrário.

(8) - Escreva uma função chamada “converte” que converta uma lista de inteiros passada como entrada para um único inteiro. Exemplo:

```
w = [10, 222, 7]
n = converte(w)
```

```
print(n)
> 102227
```

Se houver algum número negativo na lista, ele deverá ser considerado como positivo para possibilitar o processo de conversão. Exemplo:

```
w = [3, -7, -61]
n = converte(w)
```

```
print(n)
> 3761
```

(9) - Escreva uma função chamada “f_troca” que receba como entrada uma lista *w* e que retorne uma cópia modificada dessa lista, em que a posição de cada par de valores consecutivos esteja trocada. Veja os exemplos:

```
w = ["CO", "CA", "CO", "LA"]
v = f_troca(w)
```

```
print(v)
> ["CA", "CO", "LA", "CO"]
```

```
w = [10, 20, 30, 40, 50]
v = f_troca(w)
```

```
print(v)
> [20, 10, 40, 30, 50]
```

(10) - Escreva uma função que transforme duas listas de tamanho idêntico em um dicionário, conforme mostrado no exemplo a seguir:

```
disc = ["PROBAB", "BD", "CALC I", "CALC II"]
salas = [105, 203, 107, 305]
```

```
d = f_transforma(disc, salas)
print(d)
```

```
> {'PROBAB': 105, 'BD': 203, 'CALC I': 107, 'CALC II': 305}
```

(11) - Escreva uma função chamada “f_muda_texto” que receba como entrada uma string e que retorne uma cópia modificada da string da seguinte forma: ela deve estar convertida para minúsculo e ter as vogais removidas. Importante: vogais acentuadas devem ser levadas em consideração. Veja os exemplos:

```
s = "Quintino"
s2 = f_muda_texto(s)
print(s2)
> 'qntn'
```

```
s = "BÃOOO!!!"
s2 = f_muda_texto(s)
print(s2)
> 'b!!!!'
```

```
s = "aí"
s2 = f_muda_texto(s)
print(s2)
> ''
```

(12) - Considere um arquivo CSV chamado “países.csv”, em que cada registro armazena os dados de um país. Cada país é descrito por 5 variáveis:

- sigla
- nome
- continente
- população
- área em km²

Abaixo um exemplo que apresenta a linha de cabeçalho e os dados dos 5 primeiros países do arquivo (neste arquivo, os países não estão ordenados por nenhum critério específico).

```
sigla,nome,continente,população,area  
BRA,Brasil,América,212559409,8510345  
AUS,Austrália,Oceania,25499881,7741220  
POR,Portugal,Europa,10196707,92090  
CRI,Costa Rica,América,5094114,51100  
KOR,Coréia do Sul,Ásia,51269183,99720  
...
```

Faça um programa que seja capaz de processar o arquivo para **criar um arquivo de saída** contendo os seguintes resultados:

- Em que linha do arquivo (informe o número), o país com a sigla “NZL” aparece. Por exemplo, no arquivo acima BRA aparece na linha 1, AUS na linha 2, etc. (veja que a linha de cabeçalho é desconsiderada)
- O total de países do continente ‘Oceania’
- A densidade populacional do continente ‘Oceania’. A densidade populacional da Oceania é calculada através da soma da população de todos os países que pertencem a este continente, dividida pela soma da área ocupada por estes mesmos países.