

Frequência — 2ª Época

Duração: 2h

Cotação: 14 valores

Observações e Regras

- As respostas são para escrever no próprio enunciado.
- É permitido entregar a prova escrita a lápis.
- Não serão respondidas questões após 30 minutos do início do teste.
- Não é permitido sair da sala durante a realização da frequência.
- Só é permitido entregar ou desistir após a primeira hora.
- É estritamente proibido manipular quaisquer aparelhos eletrónicos (telemóveis, tablets, etc).
A violação desta regra implica a anulação da prova.
- Quando indicado, devem ser utilizadas funções, procedimentos ou classes desenvolvidas em questões anteriores.
- Não é necessário escrever quaisquer instruções de import.

Número:

--	--	--	--	--

Nome:

--

Questão 1 (3 valores)

1.1 Existem diferentes formas de calcular a distância entre vetores. Uma delas é a distância de Chebyshev. Desenvolva uma função que calcula a distância entre dois vetores representados por objetos do tipo `numpy.ndarray`, sem recorrer a qualquer função da biblioteca `numpy`. A distância entre dois vetores $\mathbf{u} = u_0, u_1, \dots, u_{n-1}$ e $\mathbf{v} = v_0, v_1, \dots, v_{n-1}$ é dada pela seguinte fórmula:

$$D_{\text{Chebyshev}}(\mathbf{u}, \mathbf{v}) = \max_{0 \leq i < n} (|u_i - v_i|)$$

Se os vetores não tiverem a mesma dimensão, a função deve devolver `None`.

Por exemplo, dados os vetor `[1.0 7.0 3.0]` e `[8.0 2.0 6.0]` a distância de Chebyshev entre os dois é 7.

```
def distance(u, v):
```

1.2 Desenvolva uma função que dada uma matriz representada por um objeto do tipo `numpy.ndarray` em que cada coluna representa um vetor, devolve a matriz de distâncias entre cada um dos vetores (representada por um objeto do tipo `numpy.ndarray`). Caso a matriz dada tenha n colunas, a matriz a devolver deve ser uma matriz D quadrada $n \times n$, em que o valor em cada posição $d_{i,j}$ corresponde à distância de Chebyshev entre o vetor na coluna i e o vetor na coluna j da matriz dada.

*Esta função **deve** usar a função desenvolvida na questão anterior. Se não a conseguiu completar, assuma que está disponível.*

Questão 2 (1 valor)

Desenvolva um procedimento que dada uma matriz representada por um objeto do tipo `numpy.ndarray`, modifica a matriz de modo a que esta passe a ser uma matriz triangular inferior (os valores acima da diagonal principal passam a ser zero). *Só deve modificar a matriz se esta for quadrada.*

Por exemplo, após a execução deste procedimento a matriz

```
[[ 1.  2.  3.]  
 [ 2.  8.  7.]  
 [ 3.  7. 21.]]
```

 passa a

```
[[ 1.  0.  0.]  
 [ 2.  8.  0.]  
 [ 3.  7. 21.]]
```

```
def lower_triangular(m):
```

Questão 3 (10 valores)

Considere a organização de uma conferência internacional. Pretende-se para esta conferência construir um *painel* com as *bandeiras* de todos os países intervenientes. Cada bandeira é identificada pelo nome do país e pelas suas dimensões reais (largura × altura).

3.1 Defina uma classe para representar o conceito de *bandeira*. Uma bandeira deve ter as seguintes propriedades *não alteráveis*:

- O nome do país (uma cadeia de caracteres);
- A altura (número decimal);
- A largura (número decimal).

Esta classe deve dispor de um construtor com três parâmetros, sendo estes o nome do país, a altura e a largura. Além disso deve ser possível consultar cada uma das propriedades, bem como obter a área da bandeira.

```
class Flag:
```

3.2 Defina uma classe que representa o conceito de *painel*. Um objeto deste tipo deve guardar a coleção de bandeiras que o compõem. Deve definir os métodos necessários para:

- Adicionar uma nova bandeira ao painel. Não é permitida a inserção de bandeiras de países que já estejam no painel;
- Remover uma bandeira do painel, dado o nome do país;
- Dado o nome de um país, devolver um valor booleano que indica se a bandeira desse país está no painel ou não;
- Devolver a área (mínima) do painel (que corresponde à soma das áreas das bandeiras de todos os países);
- Devolver média da área das bandeiras;
- Dados um valor para a largura e um valor para a altura, devolver os nomes de todos os países cujas bandeiras têm a largura e a altura dadas;
- Devolver a bandeira com a área mais próxima da área média (mesmo que existam mais que uma bandeira, devolve apenas uma);
- Mostrar o painel no ecrã. Deve ser fornecido o *número mínimo* e o *número máximo* de países por linha. Cada linha da exibição do painel deve ter, se possível, o número máximo de países por linha dado. Se o número de países da última linha (ou única linha) for menor que o mínimo definido, o método deve escrever no ecrã uma mensagem a dizer que não é possível mostrar o painel. Os nomes dos países aparecem todos com a mesma dimensão e alinhados à esquerda. Para isso pode usar a função `ljust(length)` da classe `str`.

Por exemplo, num painel com 5 países e dados 2 e 3 como números mínimo e máximo de países por linha, o resultado deve ser similar ao exemplo abaixo.

```
|Portugal|Espanha |Itália |  
|Grécia  |Irlanda  |
```

Caso os números mínimo e o máximo de países por linha fossem 2 e 4, respectivamente, na última linha ficaria apenas uma bandeira, o que faria com que fosse apresentada a mensagem de erro e não fosse mostrado o painel.

Deverá ser utilizada a classe `Flag` definida na questão anterior.

