

Laboratório 7 – Subprogramação

INSTRUÇÕES:

- Vocês devem fazer o laboratório em dupla. Os quatro arquivos com os programas desenvolvidos devem ser colocados na atividade Laboratório 7 no Google Classroom. Coloquem o nome de vocês como comentário no programa. Lembrem-se que a média dos laboratórios vale 20% da nota final.
 - Exercícios traduzidos de <https://edabit.com/challenges/python3>
1. Escreva uma função que recebe um número inteiro e que retorna a função de Kempner deste número que é o menor inteiro maior que zero cujo fatorial é dividido exatamente pelo número.

Exemplos:

kempner(6) → 3

Porque:

1! = 1 % 6 > 0

2! = 2 % 6 > 0

3! = 6 % 6 == 0

kempner(10) → 5

Porque:

1! = 1 % 10 > 0

2! = 2 % 10 > 0

3! = 6 % 10 > 0

4! = 24 % 10 > 0

5! = 120 % 10 == 0

2. Escreva uma função que gera um tabuleiro de xadrez. Ela recebe como entrada a variável n e dois elementos, e gera um tabuleiro $n \times n$ com os elementos alterados em cada casa. Ambos elementos podem ser strings ou inteiros. O primeiro elemento deve estar no canto superior esquerdo do tabuleiro. Caso os dois elementos sejam iguais, a função deve retornar o string “Inválido”.

Exemplos:

tabuleiro_xadrez(2, 7, 6) → [
 [7, 6],
 [6, 7]
]

tabuleiro_xadrez(3, "A", "B") → [
 ["A", "B", "A"],
 ["B", "A", "B"],
 ["A", "B", "A"]
]

```
tabuleiro_xadrez(4, "c", "d") → [  
  ["c", "d", "c", "d"],  
  ["d", "c", "d", "c"],  
  ["c", "d", "c", "d"],  
  ["d", "c", "d", "c"]  
]
```

```
tabuleiro_xadrez(4, "c", "c") → "invalido"
```

3. Uma tripla pitagórica é um conjunto de três números inteiros que formam um triângulo retângulo. A soma dos quadrados dos dois números menores deve ser igual ao quadrado do maior número. Dados três números a , b e c (sendo c o maior):

$$a^2 + b^2 = c^2$$

Além disso, uma tripla pitagórica é considerada primitiva se os três números são coprimos por pares - isto é, o maior fator primo comum entre quaisquer dois números é 1.

Escreva uma função que recebe uma lista de três números (não ordenados) e retorna True se os números constituírem uma tripla pitagórica primitiva, caso contrário False.

Exemplos

```
eh_tripla_pitag_prim ([4, 5, 3]) → True
```

```
eh_tripla_pitag_prim ([7, 12, 13]) → False
```

```
eh_tripla_pitag_prim ([39, 15, 36]) → False  
# Tripla pitagórica, mas não primitiva.
```

```
eh_tripla_pitag_prim ([77, 36, 85]) → True
```

4. Um grupo de n amigos vai ver um filme. Eles gostariam de encontrar um local onde pudessem sentar-se lado a lado na mesma fileira. O layout dos assentos de um cinema pode ser representado como uma matriz 2-D, onde 0s representam assentos vazios e 1s representam assentos ocupados.

```
[[1, 0, 0, 0, 1, 1, 1],  
 [1, 1, 1, 0, 1, 1, 1],  
 [1, 0, 1, 0, 1, 0, 1],  
 [1, 1, 0, 1, 1, 0, 1],  
 [1, 0, 1, 1, 1, 1, 1],  
 [1, 0, 1, 1, 0, 0, 0]]
```

Escreva uma função que, recebe um layout de ocupação de assentos e o número de amigos n , e retorna o número de configurações disponíveis para todos os n amigos sentarem juntos. No exemplo acima, se $n = 3$, haveria 2 configurações possíveis (na primeira e na última fila).

Exemplos

```
conf_assentos_juntos ([  
  [1, 0, 1, 0, 1, 0, 1],  
  [0, 1, 0, 1, 0, 1, 0],  
  [0, 0, 1, 1, 1, 1, 1],  
  [1, 0, 1, 1, 0, 0, 1],  
  [1, 1, 1, 0, 1, 0, 1],  
  [0, 1, 1, 1, 1, 0, 0]  
, 2) → 3
```

```
conf_assentos_juntos ([  
  [1, 0, 1, 0, 1, 0, 1],  
  [0, 1, 0, 0, 0, 0, 0],  
, 4) → 2
```