

FUNÇÕES AUXILIARES DO TRABALHO FINAL

Leia atentamente as recomendações a seguir, nela estão alguns dos critérios de correção para as funções auxiliares:

- **Importante!!!** Escolha somente um dos trabalhos para fazer as funções, seja 2048 ou batalha naval, ao enviar as funções, envie num arquivo separado no seguinte formato: `funcoes_auxiliares_nome_aluno.py`
- **MUITO IMPORTANTE!!!** Ao definir o trabalho final, colocar comentários no início do código o trabalho que vai fazer, seu nome e dre. **Exemplo**

```
# Trabalho final - Batalha Naval
# Arquivo de funcoes auxiliares
# Nome do Aluno
# DRE
```

- Documentação da função é essencial
- Não esqueça de colocar os tipos de entrada e saída
- Nomes elucidativos para as variáveis, funções e parâmetros são **MUITO IMPORTANTES**, nos ajudam a entender o seu código e te ajudarão a entendê-lo quando for revê-lo daqui a um mês.
- Não use acentos/cedilha, em nomes de funções variáveis ou parâmetros, o mais simples é sempre melhor. Ao separar nomes compostos recomenda-se o uso de underline (_) ou letras maiúsculas, ex: `area_retangulo`, `areaRetangulo`
- Tente escrever o código o mais simples possível, se houver alguma parte que seja de difícil compreensão, recomenda-se o uso de comentários ex: (`# essa linha de código faz tal coisa`), mas use-o com moderação, muitos comentários dificultam a compreensão do código.
- Sua solução não deve usar de conceitos e bibliotecas que não foram vistos ao longo das últimas semanas.
- **para fazer a entrega desta atividade prática, escreva suas funções no editor do IDLE, salvando todas em um único arquivo.**
- **as questões que tiverem matéria que ainda não foi apresentada, serão desconsideradas**
- **Trabalhos plagiados/copiados/iguais serão zerados**

1. 2048

2. FUNÇÕES AUXILIARES 1

- (1) Escreva uma função que recebe uma lista numérica e some os números adjacentes que sejam iguais, a soma deve ser feita da esquerda para a direita. A função deve

modificar a própria lista, retornando True se houve a modificação.

Dica: A questão Repetidos do MT07 pode ajudar nesse problema **Cuidado com o exemplo abaixo:**

```
>> l = [1,2,2,4,8]
>> funcao_auxiliar1(l) #nome generico nao use esse nome
>> l
[1,0,4,4,8] # l nao pode ser [1,0,0,0,16]
```

- (2) Modifique ligeiramente a função 1, para receber dois parâmetros, a lista e a direção (esquerda/direita) para onde deseja-se realizar a soma

Exemplos:

```
>> l = [1,2,2,4,8]
>> funcao_auxiliar1(l,'dir') #nome generico nao use esse nome
>> l
[1,0,4,4,8]
>> l2 = [1,2,2,4,8]
>> funcao_auxiliar1(l2,'esq')
>> l2
[1,4,0,4,8]
```

- (3) Crie uma função que dado uma lista numérica de tamanho qualquer, junte todos os elementos diferentes de zero no extremo direito da lista.

Exemplo:

```
>> l = [1,2,2,4,0]
>> funcao_auxiliar2(l) #nome generico nao use esse nome
>> l
[0,1,2,2,4]
```

- (4) Modifique a função acima de maneira que ela receba mais um parâmetro, a extremidade que se deseja juntar.

Exemplo:

```
>> l = [0,1,2,2,4,0]
>> funcao_auxiliar2(l, 'dir') #nome generico nao use esse nome
[0,0,1,2,2,4]
>> funcao_auxiliar2(l, 'esq')
[1,2,2,4,0,0]
```

Essa função não precisa modificar a lista original, pode devolver uma outra modificada, mas fiquem livres decidir

3. FUNÇÕES AUXILIARES 2

- (5) Faça uma função que receba duas matrizes e confere se elas são iguais, retornando True caso seja verdade False caso não seja
- (6) Quando o jogo 2048 é inicializado, ele cria um tabuleiro onde todos os elementos são vazios exceto 2. Seguindo esse raciocínio, crie uma função que crie uma matriz de zeros n por n, exceto em duas posições aleatórias que devem conter o número 2. O valor default de n deve ser 4.
- (7) Crie uma função que receba uma matriz numérica e substitua aleatoriamente um dos números zeros da matriz pelos número 2 ou 4, onde o número 2 tem uma chance maior de acontecer que o número 4. Se a inserção não puder ser feita, deve retorna False, caso contrário True.
Dica: olhe a função random.choices
- (8) Crie uma função que retorne a transposta da matriz de entrada
<https://brasilescola.uol.com.br/matematica/matriz-transposta.htm>

4. FUNÇÕES AUXILIARES 3

- (9) Crie uma função para mostrar o tabuleiro de 2048.
textbfCuidado!!! Ao começarem a aparecer números maiores o tabuleiro mostrado não pode deixar o formato padrão
Para o jogo não ficar poluído substitua zeros por '*' ou '-' quando for

mostrar para o jogador

32	16	4	-
16	-	4	-
4	-	-	-
-	-	-	-

- (10) Crie uma função que pega um número do usuário, confira se é uma potência de 2, e então retorne o número inteiro. Por exemplo, se o usuário digitar 64, é um número válido, já se digitar 78, não é um número plausível. **Atenção: Lembre-se que o usuário pode digitar qualquer coisa no shell não somente números**
Atenção 2: A função recebe a informação a partir no shell, não confunda com parâmetros
- (11) Crie uma função que irá pegar alguma jogada válida do usuário. No 2048, as jogadas válidas serão 'a', 's', 'd', 'w'. Não aceite qualquer outro tipo de informação
Atenção 2: A função recebe a informação a partir no shell, não confunda com parâmetros

5. BATALHA NAVAL

- (1) Escreva uma função que recebe uma lista numérica de cujos itens são únicos, ou seja, não pode existir uma lista dessa maneira : [1,1,2,2,3,4,5], pois tem dois 1 e dois 2, e uma lista cujos itens desejamos retirar da lista original. A função deve excluir esses elementos da lista original. Retorna True caso a operação seja possível
Exemplo:

```
>> l = [0,1,2,4,7,8]
>> ret = [1,2,4]
>> funcao_auxiliar1(l, ret) #nome generico nao use esse nome
>> l
[0,7,8]
```

- (2) Escreva uma função que recebe um número e devolve uma lista de listas de um único elemento que vai do 0 até o número-1.

Exemplo:

```
>> funcao_auxiliar2(10) #nome generico nao use esse nome
[[0], [1], [2], [3], [4], [5], [6], [7], [8], [9]]
>> funcao_auxiliar2(5)
[[0], [1], [2], [3], [4]]
```

- (3) Modifique a função 2(não crie outra função, só modifique a função já feita) de maneira que ela retorne uma lista de tuplas, onde as tuplas, são todas as combinações possíveis de de 0 a 9. Ex: (0,0),(0,1),(1,0),(9,7)..
- (4) No jogo batalha naval, precisamos sortear as posições iniciais dos navios no tabuleiro. Para isso escreva uma função que receba uma lista de posições válidas e o tamanho do navio desejado e encontre uma posição de elementos suscetivos para ele.

- (pos, 4) \rightarrow [(0,0),(0,1),(0,2),(0,3)]

- $(pos, 4) \rightarrow [(1,2),(2,2),(3,2),(4,2)]$

Atenção: Não esqueça que os navios podem ser horizontais ou verticais e essa escolha tem que ser aleatória

Dica1: As posições fornecidas pela função 2 te ajudarão com o entrada de teste

Dica2: A função `random.choices` pode te ajudar, caso ache que não será de grande ajuda, dê uma olhadinha no módulo `random` :)

- (5) Assim como no jogo da forca, no batalha naval, temos um tabuleiro que contém uma espécie de gabarito, que não é visível ao jogador e um tabuleiro que fica visível. Crie uma função que cria uma máscara para o jogo.

Importante: Aqui estamos trabalhando com matrizes, uma lista única de máscaras não é viável

Dica: Para facilitar sua vida, ao invés de usar o tracinho como foi utilizado na forca, agora utilize o til, ele parece água

6. FUNÇÕES AUXILIARES 3

- (6) Crie uma função que receba os tabuleiros do jogador 1 e do 2 e os mostre da seguinte

```

Activities  Toplevel  set 20 12:06
*Python 3.8.5 Shell*
File Edit Shell Debug Options Window Help
a
  A B C D E F G H I J      A B C D E F G H I
0 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  0 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
1 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  1 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  2 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  3 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  4 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
5 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  5 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
6 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  6 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  7 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
8 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  8 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
9 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  9 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
jogador 1                  jogador 2
Jogador 1 digite uma posicao para atacar: a0
Xiiii acertou na água
  A B C D E F G H I J      A B C D E F G H I
0 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  0 * ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
1 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  1 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  2 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  3 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  4 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
5 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  5 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
6 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  6 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  7 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
8 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  8 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
9 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  9 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
jogador 1                  jogador 2
Jogador 2 digite uma posicao para atacar:

```

forma:

- (7) Crie uma função que recebe uma indicação do jogador, se é o 1 ou o 2 e as jogadas anteriores que o jogador fez e pega a posição do usuário que ele deseja acertar no tabuleiro, com a mensagem respectiva ao Jogador 1 ou 2, como no exemplo da imagem anterior.

A função deve retornar a posição correspondente na matriz. Ex: 'A0' → [0,0]

Atenção: O usuário pode colocar qualquer tipo de informação no shell, só retorne a posição, quando for uma posição válida

Dica: Ao digitar uma jogada possível, já atualize a lista de jogadas passadas