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

(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. Funções Auxiliares 4

(12) Crie a função que movimentará o tabuleiro para direita, esquerda, cima e baixo. Pode ser uma função única que recebe como parâmetro o tabuleiro e a movimentação ou várias funções que recebem só o tabuleiro. Essas funções devem movimentar o tabuleiro na mesma posição, ou seja, elas não retornarão nada.
Dica: Para a movimentação em cima e baixo, utilize a função transposta já feita

(13) Crie uma função que retorna o status do tabuleiro. A função receberá o tabuleiro do 2048 e o número máximo(o número máximo será uma potência de 2 a ser almejada, ou seja, 64,2048,4096, etc...) e retorna o status do tabuleiro. Alguns status necessários são: se ainda existe zero presente no tabuleiro, se o número máximo está presente no tabuleiro e se ainda existem jogadas a serem feitas caso o tabuleiro esteja completamente cheio e se não existem mais jogadas possíveis
Sugestão: Usar como retorno de os status os inteiros, onda cada inteiro identifica um tipo de caso. Ex: 1 caso tenha a potencia escolhida, 2 caso tenha zeros, etc...

(14) Crie um menu para o jogo, que informa como o jogo é jogado, quais são as teclas válidas. Não precisa estar perfeito, pois ao juntar tudo no programa principal, talvez precisem mudar.

6. Função Principal

Agora que a maioria das funções necessária para criar o jogo já estão prontas, só falta criar o jogo :).
O jogo rodando deve estar em outro arquivo que as funções enviadas anteriormente.
Nesse jogo, o jogador deve ter a possibilidade de escolher qual com qual valor atingido ganhará o jogo, o jogador pode escolher 64, 128, etc.., sempre uma potência de 2. Ao ganhar o jogo, o jogo deve mostrar uma mensagem parabenizando o jogador por ter conseguido chegar na potência e o jogador deve ter a possibilidade de continuar jogando o jogo (CUIDADO!!! após conseguir atingir a potência o jogo

não deve continuar mostrando a mensagem de parabéns)
O jogador utilizará as teclas 'a','w','s','d' para movimentar o tabuleiro para a esquerda, cima, baixo e direita respectivamente
O jogador a todo momento do jogo pode escolher em sair do jogo, reiniciar o jogo e mostrar o menu
Lembre-se de adicionar o 2 ou 4 toda vez que o tabuleiro houver modificado alguma posição
Dica: caso ache necessário fazer a cópia do tabuleiro, utilize o deepcopy do módulo copy ***Bônus: o 2048 também salva um score para o usuário, adicione esse score ao jogo***

IMPORTANTE!!!

Se for necessário modifique as funções auxiliares
O jogo deve ser entregue rodando, então ponderação na hora de modificar o jogo caso adicione o score.

BOA SORTE :)

7. 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 inicias 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.

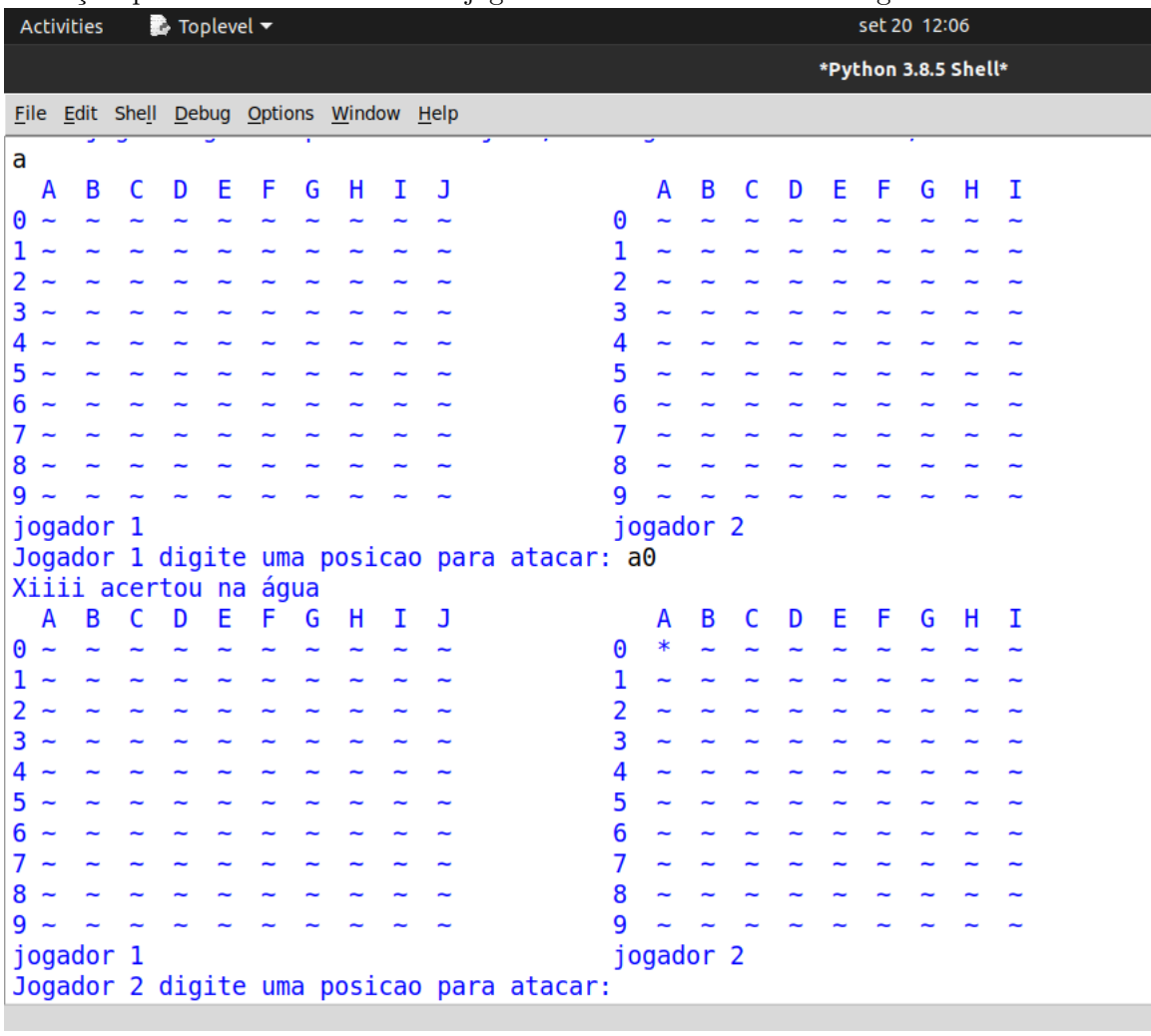
- (pos, 4) → [(0,0),(0,1),(0,2),(0,3)]
- (pos, 4) → [(1,2),(2,2),(3,2),(4,2)]

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

8. Funções Auxiliares 3

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

9. Funções Auxiliares 4

(8) Faça uma função que crie o tabuleiro do jogo batalha naval. O tabuleiro precisa conter, 1 porta-avião, 2 navios-tanque, 3 contratorpedeiros e 4 submarinos, onde cada um dele ocupa 5,4,3 e 2 posições respectivamente. Sugestão: crie um tabuleiro 10x10, onde as posições que contém água sejam 'a' e as posições que contenham o navio tenham uma indicação do mesmo. Ex: [['a','sub0','sub0','a'....][...]...[...,'a']]. Se for usar essa configuração, não esqueça de diferenciar os navios de mesmo tipo.

(9) Crie um menu para o jogo. Não precisa estar perfeito, provavelmente haverão alterações futuras

10. Função Principal

Agora que a maioria das funções necessárias para criar o jogo já estão prontas, só falta criar o jogo :).
O jogo rodando deve estar em outro arquivo que as funções enviadas anteriormente.
Nesse jogo, os jogadores a qualquer momento podem escolher sair do jogo, mostrar o menu, reiniciar o jogo e mostrar quantas e quais embarcações já foram eliminadas do total (Sugestão: usar a forma 0/n_posições embarcação para mostrar quantas já foram. Ex: (1/3) contratorpedeiros)
Se o jogador acertar um navio ele deve continuar jogando, a vez só passa do jogador, se ele acertar a água
Quando o jogador acertar água, o jogo deve mostrar uma mensagem informando que o mesmo errou e colocar '*' na posição jogada e passar a vez para o outro jogador
Caso o jogador acerte uma posição do barco, o jogo deve mostrar 'X' na posição acerta e deve continuar pedindo posições ao jogador, até que o mesmo erre
Assim que uma embarcação for totalmente afundada o jogo deve informar ao jogador qual foi o tipo de embarcação eliminada
O jogo só termina assim que todas as embarcações de um dos jogadores forem eliminadas ou eles pedirem para encerrar o jogo

Bônus: Adicione a possibilidade do jogo ser jogado contra o computador. Não precisa ter nenhuma inteligência, as posições podem ser jogadas aleatórias, mas uma sugestão é sempre que uma embarcação é atingida é sempre interessante chutar nos arredores dela

IMPORTANTE!!!

Se for necessário modifique as funções auxiliares
O jogo deve ser entregue rodando, então ponderação na hora de modificar o jogo caso adicione a possibilidade de jogar contra o computador.

BOA SORTE :)