



Certified Tech Developer
The Ultimate Degree

Descrição do problema

É necessário determinar o vencedor de um concurso de stand up composto por 3 apresentações por participante.

O público, no máximo 100 pessoas, votou quem eles acham que era melhor no final de cada etapa. Por exemplo, Alice sobe, se apresenta e desce; Bob sobe, aparece e desce.

Após a apresentação, o público vota indicando quem eles acham que ganhou aquela rodada.

Após isso, continua o próximo lote assim como o primeiro. E finalmente um terceiro.

Se você quisesse seguir esta final, como você representaria esse problema em uma tabela semelhante a uma planilha?

participante	1º votos	1º vencedor	2º votos	2º vencedor	3º votos	3º vencedor	resultado

Como esses concursos são realizados online, milhares acontecem todos os dias. Chegamos ao nosso servidor as informações para cada concurso em formato de array, uma para cada participante, portanto, as recebemos da seguinte forma:



A matriz de Alice é: **alice** = [23, 82, 46]

A matriz de Bob é: **bob** = [45, 8, 32]

Então vá para o código enquanto pensa em como comparar cada etapa.

```
const alicia = [23, 69, 32];  
const bob = [12, 67, 43];  
  
function encontrarGanhador(a, b) {  
  //solução  
}
```

A tarefa é enfrentar esses votos comparando **a[0]** com **b[0]**, **a[1]** com **b[1]** e **a[2]** com **b[2]**.

Se $a[i] > b[i]$, então Alice recebe 1 ponto.

Se $a[i] < b[i]$, então Bob recebe 1 ponto.

Se $a[i] = b[i]$, nenhuma pessoa recebe um ponto.

Os pontos de comparação são os pontos totais que uma pessoa ganhou. Não os votos, mas os pontos ganhos em cada etapa.

exemplo:

const alicia = [23, 67, 32];

const bob = [12, 67, 43];

puntosAlicia = 1

puntosBob = 1

Para os itens do índice 0, Alice recebe um ponto porque $a[0] > b[0]$.

Para a próxima fase, $a[1]$ e $b[1]$ são iguais, não são obtidos pontos.

Finalmente, para elementos do índice 2 (terceira etapa), $a[2] < b[2]$ para que Bob receba um ponto.

```
const alicia = [23, 69, 32];
const bob = [12, 67, 43];

function encontrarGanhador (a, b) {
  let pontosPrimeiroParticipante = 0;
  let pontosSegundoParticipante = 0;
  //continua con esto.
}

console.log("O ganador é: " + encontrarGanhador (a, b))
```

dado os slogans e como funciona o sistema de pontos, devemos calcular e encontrar o vencedor utilizando uma estrutura for que evite ter que calcular cada lote separadamente, lembre-se de usar as estruturas if/else, se necessário, para comparar os votos de cada participante!.

Bônus extra

Se você chegou aqui você é mais do que bom! Parabéns!

Para que você não fique com o desejo ou que você possa continuar praticando se quiser, propomos mais alguns exercícios, tenha em mente que daqui os exercícios podem subir em dificuldade tanto estrutural quanto lógico; como sempre dizemos, paciência, ignore a complexidade e tente resolvê-la com as ferramentas que você tem à sua disposição, você também pode procurar informações extras no google ou documentações que você conhece!!

digitalHouse()

Crie a função *digitalHouse()* que receberá 2 números como parâmetros. A função deve imprimir na tela os números de 1 a 100, mas levando em conta os seguintes critérios:

- Se o número a ser impresso for um múltiplo do primeiro parâmetro inserido, você deve exibir a string "Digital" em vez do número.
- Se o número a ser impresso for um múltiplo do segundo parâmetro inserido, você deve exibir a string "House" em vez disso.
- Se o número a ser imprimir for um múltiplo de ambos os parâmetros, você deve exibir a string "Digital House" em vez do número.

somaArray()... Reloaded

Vamos pegar o exercício *sumArray()* que fizemos na classe arrays, mas desta vez vamos modificá-lo para que ele possa receber um array de números de qualquer quantidade de elementos.

Se você não se lembrar da instrução original:

Você deve criar uma função *somaArray* que aceita uma matriz de números e retorna a soma de todos eles.

exemplo:

- `somaArray([1,2,3]) // 6`
- `somaArray([10, 3, 10, 4]) // 27`
- `somaArray([-5,100]) // 95`

e como estamos ajustando as funções, vamos fazer o mesmo com a função do exercício de simulação *join()*.

you must modify the `join()` function so that it can receive an array of strings of any quantity of elements.

Importante: You cannot use the original `Array.join()` method.

for example:

`join(["o", "l", "a"])` must return the string `"ola"`.

`join(["t", "c", "h", "a", "u"])` must return the string `"tchau"`.