



Atividade Array String (Lista 6)

Uau. Se você realmente chegou até aqui, Parabéns! Você já possui uma gama de recursos relevantes para ser um desenvolvedor!

Tendo isto em mente, vamos começar a mexer cada vez mais com coisas abstratas, como dados, entidades, etc. As atividades a partir de agora podem ser repetitivas e algumas podem ter soluções de uma linha, mas lembre-se que o foco delas aqui é fazer você aprender sempre um conceito novo. A idéia é que você aplique algumas técnicas antes de usar elas prontas, por isso é importante seguir as recomendações de leitura, links e vídeos.

- (OBRIGATÓRIO)** Faça um algoritmo que receba o nome de dois jogadores e após isso peça 5 números de 1 a 10 para cada jogador. Primeiro para o jogador A e depois para o jogador B.
Após a escolha de cada jogador o algoritmo deve apresentar quais foram os números iguais que o jogador A e B colocaram.
EX: Jogador A : 1, 2, 3, 4, 5
Jogador B : 1,2,3,8,7
Algoritmo imprime :
1,2,3
Esse é um problema classicamente conhecido como Inner Join
- (OBRIGATÓRIO)** Faça um algoritmo que receba o nome de dois jogadores e após isso peça 5 números de 1 a 10 para cada jogador. Primeiro para o jogador A e depois para o jogador B.
Após a escolha de cada jogador o algoritmo deve apresentar quais foram os números que apenas o jogador A colocou diferente do B.
EX: Jogador A : 1, 2, 3, 4, 5
Jogador B : 1,2,3,8,7
Algoritmo imprime :
Esse é um problema classicamente conhecido como Left Join
- (OBRIGATÓRIO)** Faça um algoritmo que receba o nome de dois jogadores e após isso peça 5 números de 1 a 10 para cada jogador. Primeiro para o jogador A e depois para o jogador B.
Após a escolha de cada jogador o algoritmo deve apresentar quais foram os números que apenas o jogador A e B colocou sem repetir nenhum número
EX: Jogador A : 1, 2, 3, 4, 5
Jogador B : 1,2,3,8,7
Algoritmo imprime :

1,2,3,4,5,7,8

4. **(OBRIGATÓRIO)** Crie um programa que receba uma palavra e exiba quantas consoantes e vogais essa palavra tem.
EX : saúde
2 consoantes
3 vogais
5. **(OBRIGATÓRIO)** Faça um programa que receba um nome e imprima a terceira letra do nome.
Utilize o método [charAt\(\)](#) para esta atividade.
6. **(OBRIGATÓRIO)** Faça um programa que receba uma palavra e verifique se ela termina com "al" se termina com "al" então imprime a palavra se não imprime "palavra não identificada".
Utilize o método [endsWith\(\)](#)
7. **(OBRIGATÓRIO)** Faça um programa que receba uma palavra e verifique se a palavra está cometendo o seguinte erro de português
Antes de "p" ou "b" devemos usar M
Caso a palavra esteja errada informe o erro
Se não disser que a palavra está certa.
Utilize o método [includes\(\)](#)
8. **(OBRIGATÓRIO)** Faça um programa que receba uma palavra e diga se há vogais ou não.
Utilize o método [indexOf\(\)](#)
9. **(OBRIGATÓRIO)** [O censor] - Faça um programa que receba uma frase e troque a palavra 'bosta' por 'estrume'. (*eu sei, nosso trabalho como programador de vez em quando é uma estrume*).
Utilize o método [replace\(\)](#)
10. **(OBRIGATÓRIO)** Faça um algoritmo que receba um texto e imprima ele em letras maiúsculas.
Utilize o método [toUpperCase\(\)](#)
11. Faça um programa que receba uma string, o programa vai estar esperando uma data no formato dd/MM/yyyy onde :
 - a. dd -> dia
 - b. mm -> mês
 - c. yyyy -> ano

Utilize o método [match\(\)](#) para fazer essa validação, você deverá utilizar um expressão regular para resolver este problema.

Caso a data esteja valida o algoritmo deve imprimir :

Entrada -> 22/05/2020

O algoritmo imprime :

Dia : 22

Mes : 05

Ano : 2020

12. Faça um algoritmo que receba uma palavra, após isso ele remove todas as vogais dessa palavra e imprime a palavra novamente porém sem as vogais.

EX: Entrada banana

O algoritmo imprime :

Bnn

Utilize o método [split\(\)](#) para quebrar a string e depois montar ela sem vogais.

13. Faça um algoritmo que verifica se uma string comece com a palavra google
Caso comece imprima que a palavra é valida se não diga que não conhece o serviço.

EX: entrada Google drive

O algoritmo imprime

Palavra valida

Utilize o método [startsWith\(\)](#)

14. Faça um algoritmo que receba um texto e imprima ele em letra minuscula novamente.

Utilize o método [toLowerCase\(\)](#)

15. Faça um algoritmo que receba um texto e remova todos os espaços em brancos no começo e no final do texto

Utilize o método [trim\(\)](#)

16. Faça um algoritmo que pergunta para o usuário se ele quer criptografar ou descriptografar.

Utilize como criptografia a [cifra de César](#)

Exemplo ->

Entrada : ARBYTE

SAIDA: DUEBWH

Caso o usuario opte por criptografar o algoritmo deve aplicar a cifra de César e mostrar o valor criptografado

Caso o usuário opte por descriptografar o algoritmo deve fazer a engenharia reversa e

retorna ao valor antes de criptografia.

Utilize todo o conhecimento até aqui pra desenvolver este algoritmo

17. Faça um programa que vai receber duas entradas

A primeira entrada é o cabeçalho (HEADER) ela vai possuir o seguinte formato

campo1;campo2;campo3;campo4

Onde se quebrarmos a string por ';' teremos o seguinte array

[0] campo1

[1] campo2

[2] campo3

[3] campo4

A quantidade de campos depende da quantidade de ';' na entrada do cabeçalho então se o usuário inserir mais ; terá mais campos no array

Utilize o método split(';') para converter a string para array

A segunda entrada é o dado (DATA) ele vai possuir o mesmo formato acima porém com valores entre ;

valor1;valor2;valor3;valor4

[0] valor1

[1] valor2

[2] valor3

[3] valor4

A quantidade de campos depende da quantidade de ';' na entrada do dado então se o usuário inserir mais ; terá mais campos no array

Utilize o método split(';') para converter a string para array

As duas entradas devem possuir a mesma quantidade de ";" isso vai permitir que elas tenham tamanhos iguais no array e então teremos uma associação parecida com essa abaixo :

Suponha que o usuario inseriu :

Cabeçalho : Nome;Sobrenome;Idade;Peso

Dado : Fulano;Ciclano;22;80

Campo1 :	valor1
Campo2 :	valor2
Campo3 :	valor3
Campo4 :	valor4

Nome :	Fulano
Sobrenome :	Ciclano
Idade :	22
Peso :	80

Caso a entrada possua mais campos (';') que os valores ou vice versa o algoritmo deve lançar um erro e dizer que o formato não está compatível.

Sendo assim o algoritmo imprime :

Nome : Fulano
Sobrenome : Ciclano
Idade : 22
Peso : 80

18. Faça um algoritmo em que o usuário possa cadastrar preços de produtos e no final exiba o preço de cada produto junto com o valor total. após inserir todos os produtos o programa deve pedir o valor a ser pago, caso o valor passe do da compra exibir o troco, se o valor for menor informar que a compra vai ser cancelada.

EX entradas :

0.75

0.25

2.50

Valor total pago: 5.00

O programa imprime :

Produto 1 - 0.75

Produto 2 - 0.25

Produto 3 - 2.50

Total da compra : 3.50 \$

Troco : 1.50 \$

19. Peça ao usuário para digitar dez valores numéricos e ordene por ordem crescente esses valores, guardando-os num vetor e depois exibindo para o usuário. Para esta

atividade tente usar o algoritmo Bubble Sort (Método da bolha), pesquise sobre ele na internet.

Vídeo bem detalhado do assunto: <https://www.youtube.com/watch?v=ZYO43bi0IXY>

Video ilustrativo : <https://www.youtube.com/watch?v=lyZQPjUT5B4>