Avaliação de Algoritmos - 2024 - 3º Bimestre - A

1) (vale 2.0 pontos) - Considerando que a matriz representa um pedacinho do mapa no Minecraft, ou seja, cada elemento da matriz está preenchido com um material conforme a legenda.

1	Terra
2	Pedra
3	Água
4	Grama

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	3	1	1	1	1
2	1	1	2	2	2	1	1	1	1	1	3	3	1	1	1
3	1	1	2	2	2	1	1	1	3	3	3	3	3	1	1
4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
6	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
7	1	4	4	4	4	2	2	4	4	4	4	1	4	4	4
8	1	1	4	4	4	4	2	2	2	2	1	1	4	4	4
9	1	1	1	1	4	4	4	4	4	1	1	4	4	4	4

Faça uma função (apenas a função) que possibilite a troca de um material por outro. Por exemplo, trocar todas as pedras por terra. Lembre-se que a função tem que ter entrada de dados, processamento e saída.

2) (vale 2.0 pontos) - Considere o agregado homogêneo multidimensional e o agregado homogêneo unidimensional abaixo para fazer e mostrar o teste de mesa. Mostre também a resposta resultante (o que o console.log(resp) vai imprimir).

let mat = [[5, 6, 7, 8, 5], [4, 6, 6, 4, 4], [6, 5, 4, 3, 3]];
let vet = ["Homer", "Clancy", "Eddie"]

```
function doIt(matriz) {
    let ss = [];
    for (let i = 0; i < matriz.length; i++) {
        let a = 0;
        for (let j = 0; j < matriz[i].length; j++) {
            a += matriz[i][j];
        }
        ss.push(a);
    }
    return ss;
}
```

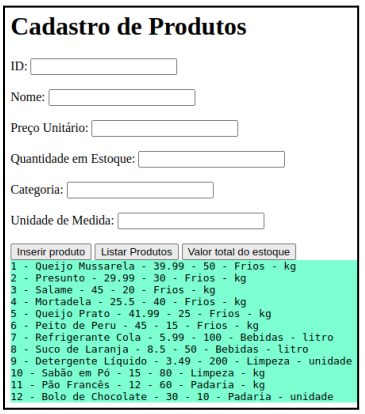
```
function execute() {
    let sss = doIt(mat);
    let resp = "";
    for (let i = 0; i < vet.length; i++) {
        resp += vet[i] + " - " + sss[i] + "\n"
    }
    console.log(resp);

execute()
```

3) (vale 6.0 pontos) - Considerando um cadastro de produtos, temos:

```
// Produto.js (Model)
class Produto {
    constructor(id, nome, precoUnitario, quantidadeEstoque, categoria, unidadeDeMedida) {
        this.id = id;
        this.nome = nome;
        this.precoUnitario = precoUnitario;
        this.quantidadeEstoque = quantidadeEstoque;
        this.categoria = categoria;
        this.unidadeDeMedida = unidadeDeMedida;
    }
}
```

Produto.html (View)




```
<body>
    <h1>Cadastro de Produtos</h1>
    <label for="id">ID:</label>
    <input type="number" id="id" name="id"><br><br>
    <label for="nome">Nome:</label>
    <input type="text" id="nome" name="nome"><br><br>
    <label for="precoUnitario">Preço Unitário:</label>
    <input type="number" step="0.01" id="precoUnitario" name="precoUnitario"><br><br>
    <label for="quantidadeEstoque">Quantidade em Estoque:</label>
    <input type="number" id="quantidadeEstoque" name="quantidadeEstoque"><br><br>
    <label for="categoria">Categoria:</label>
    <input type="text" id="categoria" name="categoria"><br><br>
    <label for="unidadeDeMedida">Unidade de Medida:</label>
    <input type="text" id="unidadeDeMedida" name="unidadeDeMedida"><br><br>
    <input type="button" value="Inserir produto" onclick="inserirProduto()">
    <input type="button" value="Listar Produtos" onclick="listarProdutos()">
    <input type="button" value="Valor total do estoque" onclick="valorTotalDoEstoque()">
    <div id="outputLista" style="background-color: aquamarine;"> </div>
    <script src="./Produto.js"></script>
    <script src="./ProdutoControl.js"></script>
</body>
```

ProdutoControl.js, complete nas funções o que estiver faltando….

```
let listaDeProdutos = [];
// ao iniciar/atualizar chama a função para inserir os dados iniciais
window.onload = inserirDadosIniciais();
function inserirDadosIniciais() { //esta função será excluída quando terminarem os testes
    listaDeProdutos = [];
    listaDeProdutos.push(new Produto(1, "Queijo Mussarela", 39.99, 50, "Frios", "kg"));
    listaDeProdutos.push(new Produto(3, "Salame", 45.00, 20, "Frios", "kg"));
    listaDeProdutos.push(new Produto(4, "Mortadela", 25.50, 40, "Frios", "kg"));
    listaDeProdutos.push(new Produto(6, "Peito de Peru", 45.00, 15, "Frios", "kg"));
    listaDeProdutos.push(new Produto(7, "Refrigerante Cola", 5.99, 100, "Bebidas", "litro"));
    listaDeProdutos.push(new Produto(8, "Suco de Laranja", 8.50, 50, "Bebidas", "litro"));
    listaDeProdutos.push(new Produto(9, "Detergente Líquido", 3.49, 200, "Limpeza", "unidade"));
    listaDeProdutos.push(new Produto(10, "Sabão em Pó", 15.00, 80, "Limpeza", "kg"));
    listaDeProdutos.push(new Produto(11, "Pão Francês", 12.00, 60, "Padaria", "kg"));
    listaDeProdutos.push(new Produto(12, "Bolo de Chocolate", 30.00, 10, "Padaria", "unidade"));
}
function gerarPrint(vetor){
    let resp = "";
    for (let i = 0; i < vetor.length; i++) {
        const prod = vetor[i];
        resp += prod.id + " - " + prod.nome + " - " +  prod.precoUnitario + " - " +
            prod.quantidadeEstoque + " - " +  prod.categoria + " - " +  prod.unidadeDeMedida + "<br>"
    }
    return resp;
}
function inserirProduto() { } // vale 1.5 pontos - adicione um produto na lista.
function listarProdutos() { } // vale 1.5 pontos - listar todos os produtos cadastrados
function valorTotalDoEstoque() { } // vale 3.0 pontos - calcular e mostrar o valor total do estoque. Deve-se considerar as quantidades e os preços dos produtos na lista.
```