**2ª Avaliação**

1. (2,0 Pontos) Desenvolva uma função que dado um array de números inteiros como parâmetro, retorne a média aritmética desses números.

function media(numeros) {

var soma = 0;

for (var i = 0; i < numeros.length; i++) {

soma = soma + numeros[i];

}

return soma / numeros.length;

}

1. (2,0 Pontos) Escreva uma função que receba um array de números inteiros como parâmetro e retorne a média aritmética simples entre o maior e o menor número contido neste array. Observação: não pode usar a função array.sort() nessa questão!

function media(numeros) {

var maior, menor;

maior = numeros[0];

menor = numeros[0];

for (var i = 0; i < numeros.length; i++) {

if (numeros[i] > maior) {

maior = numeros[i];

}

if (numeros[i] < menor) {

menor = numeros[i];

}

}

return (maior + menor) / 2;

}

1. (2,0 Pontos) Elabore uma função que receba como parâmetros dois números inteiros a e b e conte quantos números primos há entre a e b (inclusive). Exemplo: se os números forem a = 2 e b = 13, a função terá que retornar o valor 6 (pois entre 2 e 13 temos 6 números primos: 2, 3, 5, 7, 11 e 13).

function primos(a, b) {

var contador = 0;

for (var i = a; i <= b; i++) {

ehPrimo = true;

for (var j = 2; j < i; j++) {

if (i % j == 0) {

ehPrimo = false;

break;

}

}

if (ehPrimo) {

contador++;

}

}

return contador;

}

1. (2,0 Pontos) Desenvolva uma função recursiva e uma não recursiva para calcular o fatorial de um número inteiro n (n!). Observação: lembre-se que, por convenção, 0! = 1

function fatorial(n) {

var fat = 1;

for (var i = 1; i <= n; i++) {

fat = fat \* i;

}

return fat;

}

function fatorialR(n) {

if (n == 0) {

return 1;

}

return n \* fatorialR(n - 1);

}

1. (2,0 Pontos) Elabore uma função que dada uma matriz qualquer, retorne a transposta dessa matriz.

function transposta(m) {

var linhas = m.length;

var colunas = m[0].length;

var t = [];

for (var l = 0; l < linhas; l++) {

t[l] = [];

for (var c = 0; c < colunas; c++) {

t[l][c] = 0;

}

}

for (var l = 0; l < linhas; l++) {

for (var c = 0; c < colunas; c++) {

t[c][l] = m[l][c];

}

}

return t;

}