

## Programação para Design – Lista de Exercícios

1 - ) Escreva uma função (com function) que receba três variáveis do tipo number e retorne a maior e a menor variável em uma tupla [number,number] onde o primeiro elemento da tupla é o maior e o segundo elemento da tupla é o menor.

2 - ) Escreva uma função (com function) que receba um vetor de números inteiros não repetidos e retorne o maior e o menor número em uma tupla [number,number] onde o primeiro elemento da tupla é o maior elemento do vetor e o segundo elemento da tupla é o menor elemento do vetor.

3 - ) Escreva uma função (com function) que receba duas variáveis do tipo **enum JoKenPo**, descrito abaixo:

```
enum JoKenPo {  
  
    PAPEL = "PAPEL",  
    PEDRA = "PEDRA",  
    TESOURA = "TESOURA"  
}
```

Cada variável é relativa a um jogador (jogador 1 e jogador 2). Sua função deve retornar o vencedor do jogo JoKenPo.

4 - ) Escreva uma função (com function) que receba dois números inteiros como parâmetro e calcule o resultado do primeiro número elevado ao segundo. Em suma, implemente a função de potência sem usar atalhos da linguagem.

5 - ) Escreva uma função (com function) que receba os números a, b e c e calcule a os “x”s da fórmula de Bháskara. Retorne as raízes em uma tupla.

$$\Delta = b^2 - 4.a.c$$
$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$

6 - ) Escreva uma função (com function) que receba um número como parâmetro e calcule se o número é par ou ímpar. A função deve retornar um boolean.

7 - ) Seja a tupla:

```
let aluno: [string, number] // o primeiro elemento é o nome e o segundo o ira
```

Escreva uma função arrow que receba um vetor de aluno ( alunos: aluno[] ) e imprima apenas os nomes dos alunos que estão abaixo da média da turma em relação ao ira.

8 - ) Seja a union:

```
let indeciso: (boolean | number | string)
```

Escreva uma função arrow que receba um vetor de indeciso (indecisos:indeciso[]) e, fazendo uso do typeof, indique qual o tipo do elemento na posição i do vetor (veja slide 34)

9 - ) Escreva uma função (com function) que receba como parâmetro dois vetores de inteiros. Na função, imprima apenas os números do segundo vetor que forem maiores que a média dos números do primeiro vetor.

10 - ) Usando a função potência implementada em 4 - ), escreva uma função arrow que receba um vetor de inteiros e imprima cada inteiro elevado ao quadrado fazendo uso da função implementada em 4 - ).

11 - ) Faça uma função arrow que receba três vetores: os dois primeiros conterão números e o terceiro será um vetor de caracteres preenchidos com as quatro operações aritméticas básicas. O programa efetuará as operações contidas no terceiro vetor com o primeiro vetor como operando 1 e o segundo vetor como operando 2. O resultado deve ser armazenado em um quarto vetor. O quarta vetor é então retornado pela função. Veja o Exemplo:

V1	10	5	7	11	3	45
V2	1	5	4	15	1	5
V3	+	*	-	-	+	/
Resultado	11	25	3	-4	4	9

12 - ) Escreva uma função (com function) que receba um vetor como parâmetro e retorne um outro vetor que é a inversão do primeiro (elemento i recebe o valor do elemento i + 1, no entanto o último elemento recebe o primeiro). Veja o exemplo:

Original	10	5	7	11	3	45
Modificado	5	7	11	3	45	10

13 - ) Faça uma função arrow que receba o valor apurado por um vendedor de sapatos durante um dia de trabalho e retorne o valor do seu bonus diário calculado da seguinte forma: se o valor apurado for mais que R\$ 2000,00, o vendedor recebe um bônus de 10% em cima do valor e 7.5% casos contrários.

14 - ) Faça uma função que leia um inteiro N e mostre todos os N primeiros números primos.

15 - ) Faça uma função arrow que receba um number e retorne o seu fatorial.