

**Atividade de Fixação - Teste Automatizados – Jasmine**

**Parte I**

1. Crie um arquivo (conversorTmp.js) para compor um objeto ConversorTmp, esse objeto terá as funções: Celsius To Fahrenheit, Fahrenheit To Celsius, Celsius To Kelvin e Kelvin To Celsius.

Para se ter certeza das funções implementadas corretamente, crie um arquivo (conversorTmp.spec.js) para compor todos os testes das funções.

2. Crie um arquivo (dia\_semana.js) para compor uma função que receba como parâmetro um número e retorne o dia correspondente da semana (1-Domingo, 2- Segunda, etc.), qualquer outro valor digitado deverá retornar, inválido.

Para se ter certeza da função implementada corretamente, crie um arquivo (dia\_semana.spec.js) para compor todos os testes da função.

3. Crie um arquivo (funcao\_conceito.js) para compor uma função que receberá como parâmetros duas notas de um aluno, essa função deverá retornar um conceito, a atribuição de conceitos obedece a lista abaixo:
  - Entre 9.0 e 10.0-->A
  - Entre 7.5 e 8.9-->B
  - Entre 6.0 e 7.4-->C

Para se ter certeza da função implementada corretamente, crie um arquivo (funcao\_conceito.spec.js) para compor todos os testes da função.

4. Crie um arquivo (verifica\_triangulo.js) para compor uma função que receba como parâmetros os lados de um triangulo, a função deverá verificar se os lados (a, b, c) podem ser um triângulo seguindo a regra de existência do triângulo.

O programa deverá indicar, caso os lados obedeçam a regra de existência do triângulo, se o mesmo é: equilátero, isósceles ou escaleno.

Três lados formam um triângulo quando a soma de quaisquer dois lados for maior que o terceiro, veja a seguir:

- $a+b > c$
- $a+c > b$
- $b+c > a$

Tipos de Triângulos:

- Triângulo Equilátero: três lados iguais;
- Triângulo Isósceles: quaisquer dois lados iguais;
- Triângulo Escaleno: três lados diferentes.

Para se ter certeza da função implementada corretamente, deverá ser criando um arquivo (`verifica_triangulo.spec.js`) para compor todos os testes da função.

5. As Organizações Tabajara resolveram dar um aumento de salário aos seus colaboradores.

Crie um arquivo (`reajuste.js`) para compor uma função que recebe como parâmetro o salário de um colaborador e retorne o reajuste segundo o critério:

- salários até R\$ 280,00 (incluindo) : aumento de 20%
- salários acima de R\$ 280,00 até R\$ 700,00 : aumento de 15%
- salários acima de R\$ 700,00 até R\$ 1500,00 : aumento de 10%
- salários acima de R\$ 1500,00 : aumento de 5%

Para se ter certeza da função implementada corretamente, deverá ser criando um arquivo (`reajuste.spec.js`) para compor todos os testes da função.

6. Um posto está vendendo combustível com a seguinte tabela de descontos:

- Álcool:
  - até 20 litros, desconto de 3% por litro
  - acima de 20 litros, desconto de 5% por litro
- Gasolina:
  - até 20 litros, desconto de 4% por litro
  - acima de 20 litros, desconto de 6% por litro

Crie um arquivo (`posto_combustivel.js`) para compor uma função que receba como parâmetros o número de litros vendidos e o tipo de combustível (codificado da seguinte forma: A->álcool, G->gasolina), a função deve calcular e retornar o valor a ser pago pelo cliente, sabe-se que o preço do litro da gasolina é R\$ 3,70 e o preço do litro do álcool é R\$ 3,20.

Para se ter certeza da função implementada corretamente, deverá ser criando um arquivo (`posto_combustivel.spec.js`) para compor todos os testes da função.

Boa sorte !