Disciplina de Programação Funcional
1ª Lista de Exercícios
Curso de Engenharia de Computação
UEMG Ituiutaba
https://bit.ly/2A0eUl8
https://github.com/mauro-hemerly/UEMG-2018-2

1. Teste as seguintes expressões no sistema WinGHCi, descreva a operação realizada e informe o resultado obtido:
   - `6 ‘div‘ 3`
   - `div 6 3`
   - `10 ‘mod‘ 7`
   - `mod 10 4 == 2`
   - `5^3`
   - `5**3`
   - `5^(-3)`
   - `5**(-3)`
   - `2^3^4`
   - `2**3**4`
   - `4*2^3`
   - `sqrt ((5**2) + (9**2))`
   - `sqrt 25 + 73`
   - `sin(pi/6)`
   - `cos 0.5`
   - `(+) 1 ((+) 2 3)`
   - `36*14 == 450-23/2`
   - `length [’a’..’z’]`
   - `"codigo" ++ "-fonte"`
   - `if 12>5 then 100 else 200`
   - `sum[1..115]`
   - `log 2.718`
   - `log 10`
   - `exp 2`
   - `floor (exp 2)`
   - `log (exp 2)`
   - `(sin x)^2 + (cos x)^2 where x = 2`
   - `pi * r * r where r = 3`
   - `add 2 3 where add a b = a + b`
   - `add 4 5`
2. Qual o tipo de cada um dos valores abaixo?
   (a) `[’a’,’b’,’c’]`
   (b) `(’a’,’b’,’c’)`
   (c) `[(False ,’0’), (True ,’1’)]`
   (d) `([False,True],[’0’,’1’])`
   (e) `[tail,init,reverse]`
3. Qual o tipo de cada uma das funções abaixo ? (**determinar a assinatura**)
   (a) `second xs = head (tail xs)`
   (b) `swap(x,y) = (y,x)`
   (c) `pair x y = (x,y)`
   (d) `double x = x*2`
   (e) `palindrome xs = reverse xs == xs`
   (f) `twice f x = f(fx)`
4. Analise a função seguinte e explique sua finalidade.

```
fun m n p = (m==n) && (n==p)
```

5. Sejam as duas funções abaixo que verificam se um dado número é par. Teste cada função e explique a estratégia utilizada na implementação de cada uma.

```
par x = (mod x 2) == 0

par1 x = if (x == 0) then True
                     else not (par1 (x-1))
```

6. Considere a seguinte função:

```
test n = if (n `mod` 2 == 0) then n
                     else test(2 * n + 1)
```

   Para quais valores de entrada (n) a função não se encerra? Por quê? Use exemplos simples para explicar sua resposta.

1. Fornecidos três valores, a, b e c, escreva uma função que retorne quantos dos três são iguais. A resposta pode ser 3 (todos iguais), 2 (dois iguais e o terceiro diferente) ou 0 (todos diferentes).
2. Fornecidos três valores, a, b e c, elaborar uma função que retorne quantos desses três números são maiores que o valor médio entre eles.
3. Escrever uma função `potencia_2` que retorne o quadrado de um número (x^2).
4. Reutilizando a função `potencia_2`, construir uma função `potencia_4` que retorne o seu argumento elevado à quarta potência.

5. Implemente em Haskell a função do or-exclusivo, a qual é dada por:

$$a \otimes b = (a \vee b) \wedge \sim (a \wedge b)$$

■ OBSERVAÇÃO ■ Os conectivos lógicos encontram-se na seção 6.1.1.

6. Escrever duas funções, `x_maior` que retorne o maior e `x_menor` que retorne menor valor real, das raízes de uma equação do 2º grau. A expressão genérica é dada por:

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$