Universidade Federal de Uberlândia - UFU
Faculdade de Computação - FACOM
Lista de exercícios de Programação Funcional

1. Mostre os resultados das seguintes expressões:

> [1,2,3,4] ++ [9,10,11,12]
```
[1,2,3,4,9,10,11,12]
```

> "hello" ++ " " ++ "world"
```
"hello world"
```

> 5:[1,2,3,4,5]
```
[5,1,2,3,4,5]
```

> [[1,2,3,4],[5,3,3,3],[1,2,2,3,4],[1,2,3]] ++ [[1,1,1,1]]
```
[[1,2,3,4],[5,3,3,3],[1,2,2,3,4],[1,2,3],[1,1,1,1]]
```

> [9.4,33.2,96.2,11.2,23.25] !! 1
```
33.2
```

> head [5,4,3,2,1]
```
5
```

> tail [5,4,3,2,1]
```
[4,3,2,1]
```

> last [5,4,3,2,1]
```
1
```

> init [5,4,3,2,1]
```
[5,4,3,2]
```

> head []
```
*** Exception: Prelude.head: empty list
```

> length [5,4,3,2,1]
```
5
```

> null [1,2,3]
```
False
```

> null []
```
True
```

> reverse [5,4,3,2,1]
```
[1,2,3,4,5]
```

> take 3 [5,4,3,2,1]
```
[5,4,3]
```

> take 1 [3,9,3]
```
[3]
```

> drop 3 [8,4,2,1,5,6]
```
[1,5,6]
```

> minimum [8,4,2,1,5,6]
```
1
```

> maximum [1,9,2,3,4]
```
9
```

> sum [5,2,1,6,3,2,5,7]
```
31
```

> product [6,2,1,2]
```
24
```

> elem 4 [3,4,5,6]
```
True
```

> [1..9]
```
[1,2,3,4,5,6,7,8,9]
```

> [2,4..20]
```
[2,4,6,8,10,12,14,16,18,20]
```

> [3,6..20]
```
[3,6,9,12,15,18]
```

> zip [1,2,3,4,5] [5,5,5,5,5]
```
[(1,5),(2,5),(3,5),(4,5),(5,5)]
```

> zip [1 .. 5] ["one", "two", "three", "four", "five"]
```
[(1,"one"),(2,"two"),(3,"three"),(4,"four"),(5,"five")]
```

2. Qual o resultado das expressões a seguir? Justifique.

> snd (fst ((True, 4), "Bom"))
```
4
```

> fst ((True, 'a'), 7)
```
(True,'a')
```

> snd (snd ((True, 4), (False, 7)))
```
7
```

3. Implemente uma função que receba o primeiro e o último nome de alguém e retorne suas iniciais em uma tupla. Por exemplo:
> iniciais "Sara" "Luzia"
('S','L')

```
iniciais :: String -> String -> (Char,Char)
iniciais nome sobrenome = (head nome, head sobrenome)

iniciais "Sara" "Luzia"
('S','L')
```

4. Faça uma função que receba o nome de um aluno e uma lista com suas três notas. A função deve então devolver uma tupla com o nome do aluno e a média das notas.
Exemplo: aprovado "Joao" [80, 60, 100]
("Joao", 80)

```
exe4::String->[Float]->(String, Float)
exe4 aluno [x,y,z] = (aluno, (x+y+z)/3)
exe4 "Sara" [10.2,20.3,33.4]
("Sara",21.300001)
```

5. Diga quais são os tipos das seguintes expressões:

a) False
   Bool
b) (["foo", "bar"], 'a') (["foo", "bar"], 'a')
   ([String,String], Char)
c) [(True, []), (False, [['a']])]
   Lista de tuplas, sendo duas tuplas do tipo Boll e Char, respectivamente.
d) 2.5
   Float
e) (1, 1.5, 2.5, 10)
   Tupla do tipos Int e Float.

6. Crie as seguintes funções em Haskell definindo corretamente os tipos de dados.

a) Função para calcular a média de 4 números em ponto flutuante.

```
--a) Função para calcular a média de 4 números em ponto flutuante.
media :: Float -> Float -> Float -> Float -> Float
media a b c d = (a+b+c+d)/3
```

b) Função soma de 3 números inteiros.

```
-- b)   Função soma de 3 números inteiros.
soma :: Int -> Int -> Int-> Int
soma a b c  = (a+b+c)
```

c) Função que retorne a raiz quadrada de um número real. Utilize a função sqrt para o cálculo da raiz.

```
--c)Função que retorne a raiz quadrada de um número real. Utilize a função
sqrt para o cálculo da raiz.
raiz :: Float -> Float
raiz x = sqrt x
```

7. Defina uma lista por compreensão onde cada posição é uma tupla (x,y), com x < y, assumindo x como valores pertencentes à lista [1,3,5] e y pertencente a [2,4,6].

```
[(x,y) | x <- [1,3,5], y <- [2,4,6], x < y]
[(1,2),(1,4),(1,6),(3,4),(3,6),(5,6)]
```

8. Defina a lista abaixo por compreensão:

[0,3,6,9,12,15]

```
[3*x | x <- [0..5]]
```