

2º Trabalho de Programação Funcional

Curso de Engenharia de Computação - UEMG Ituiutaba

2º Semestre de 2019

Observação: Todas as questões abaixo referem-se à linguagem de programação funcional Haskell

Aluno(a): _____

Aluno(a): _____

1. Seja a função abaixo:

```
1: seg :: (Bool, Int) -> Int
2: seg (x, y) = y
```

Se não indicarmos o tipo **seg::(Bool,Int)->Int** qual seria o tipo de **seg** ?

2. Definir uma função que determina se uma lista é vazia.

```
vazia :: [a] -> Bool
```

3. A função que soma os elementos de uma lista até à 3ª posição pode ser definida da seguinte forma:

```
1: soma3 :: [Integer] -> Integer
2: soma3 [] = 0
3: soma3 (x:y:z:t) = x+y+z
4: soma3 (x:y:t) = x+y
5: soma3 (x:t) = x
```

Em **soma3** a ordem das definições é importante? Por quê? Será que obtemos a mesma função se alterarmos a ordem das definições? Defina uma **função equivalente** a esta usando apenas as funções pré-definidas **take** e **sum**.

```
take :: Int -> [a] -> [a]
sum :: Num a => [a] -> a
```

4. Considere a função abaixo:

```
1: opp :: (Int,(Int,Int)) -> Int
2: opp z = if ((fst z) == 1)
              then (fst (snd z)) + (snd (snd z))
              else if ((fst z) == 2)
                      then (fst (snd z)) - (snd (snd z))
                      else 0
```

Defina uma outra versão da função **opp** que tire proveito do mecanismo de *pattern matching* (casamento de padrões). Qual das versões lhe parece mais legível ?

5. Defina a função

```
transf :: [a] -> [a]
```

que faz a seguinte transformação: recebe uma lista e, caso essa lista tenha pelo menos 4 elementos, troca o 1º com o 2º elemento, e o último com o penúltimo elemento da lista. Caso contrário, devolve a mesma lista. Por exemplo: **transf** [1,2,3,4,5,6,7] [2,1,3,4,5,7,6].

6. Escrever a definição da função e sua assinatura, **primeiros**, que obtem os **n primeiros elementos** de uma lista.

7. Escrever qual seria o resultado da avaliação da função abaixo.

```
Prelude> map  (++) " PM" ["5:00","10:30"]
```