**DESCRIÇÃO**

A atividade consiste em resolver os exercícios abaixo utilizando o paradigma funcional, e pode ser feita **em dupla**.

1. Qual é o resultado da expressão abaixo? Por quê?

(1 == 4 && True, mod (4\*8) 31^2-5)

1. Cite um valor matemático para o qual os operadores \*\* e ^ não apresentam o mesmo comportamento.
2. Crie a função dobro (dobro :: Double -> Double), que deve retornar o dobro de um número.
3. Crie as funções incremento e decremento, que devem adicionar e remover 1 a um valor, respectivamente.
4. Interprete o comando abaixo:

\*Main> :t decremento (incremento 9)

decremento (incremento 9) :: Num a => a

1. Implemente a função sobeDesce (sobeDesce :: (Num t, Num t) => (t, t) -> (t, t)). Esta função recebe um par ordenado e devolve um par ordenado com o primeiro valor somado a 1 e o segundo valor subtraído a 1.
2. Implemente a função sobeDesce2, semelhante à questão anterior, mas utilizando as funções incremento e decremento.
3. Implemente a função trocaValor (a, b), que deve inverter os valores de a e b no par ordenado.
4. A função negate (negate :: Num a => a -> a) serve para mudar o sinal de um número. Crie um exemplo válido utilizando esta função.
5. Um programador iniciante está aprendendo Haskell e decidiu tentar resolver a lista do professor Ricardo. Ele tentou resolver o exercício 9, mas não obteve sucesso e disse que a lista do professor é impossível. O código dele encontra-se abaixo. Sua tarefa é explicar ao programador por que ele obteve erro.

| Prelude> negate -8  <interactive>:3:1:  Non type-variable argument in the constraint: Num (a -> a)  (Use FlexibleContexts to permit this)  When checking that ‘it’ has the inferred type  it :: forall a. (Num a, Num (a -> a)) => a -> a |
| --- |

**PESO DA AVALIAÇÃO**

Notas de aula.

**OBSERVAÇÕES**

* Plágio = **ZERO (inclui cópia ou simples alteração de trabalho de colegas)**