

Programação Funcional – BSI – 1º Período

Segunda lista de exercícios

Implementar as tarefas de programação descritas no presente documento, usando a linguagem Haskell .

1. Crie uma função que intercala os elementos de duas listas, de qualquer tamanho, contendo números inteiros, numa nova lista . Ex:

```
>intercala [1,2,3,4] [100,200]
[1, 100, 2, 200, 3, 4]
```

```
intercala::[Int]->[Int]->[Int]
intercala x [] = x
intercala [] x = x
intercala (a:xs) (b:ys) = a: b: intercala xs ys
```

2. Defina novos tipos para representar os dados contidos numa agenda telefônica pessoal. Em seguida, crie uma função para recuperar o nome de um contato, a partir de um número de telefone. Caso o número não seja encontrado, retornar a mensagem “Telefone desconhecido”.

```
-- contato = (nome, endereço, telefone, email)
type Contato = ([Char],[Char],[Char],[Char])
type Agenda = [Contato]
```

```
listaContato::Agenda
listaContato = [("Ana","SP","1234-1234","ss@ss.ss"),
                ("Clara","MA","2341-2341","cc@ss.ss"),
                ("José","DF","3423-3423","gg@ss.ss"),
                ("Silvio","MG","2134-2134","pp@ss.ss"),
                ("Marta","SC","4213-4213","kk@ss.ss")]
```

[illegible]

3. Faça uma função para converter uma data, escrita no formato dd/mm/aaaa (calendário Gregoriano) na correspondente data (ddd/aaaa) do calendário Juliano.

```
type MesesAno = [Int]

mesesBissexto::MesesAno
mesesBissexto = [31,29,31,30,31,30,31,31,30,31,30,31]

meses::MesesAno
meses = [31,28,31,30,31,30,31,31,30,31,30,31]

-- Função que verifica se um ano é bissexto
bissexto::Int->Bool
bissexto a = if mod a 4 == 0 then True else False

-- Função de conversão de data Gregoriana em Juliana
converteGregJul::Int->Int->Int->(Int,Int)
converteGregJul d m a =
    if bissexto a then (somaDias d m mesesBissexto,a)
    else (somaDias d m meses,a)

-- Função que soma os dias corridos até uma data
somaDias::Int->Int->[Int]->Int
somaDias d 1 lista = d
somaDias d m (a:xs) = a + somaDias d (m-1) xs
```