Programação Funcional
Roteiro de atividades práticas 4
Listas por compreensão e recursão em listas

Esse roteiro deve ser desenvolvido de forma assíncrona pelo aluno. Para que essas atividades sejam avaliadas e contabilizadas (nota e presença) o arquivo .hs referente às atividades abaixo deve ser enviado para a monitora Domitila Crispim, conforme explicado em sala.

Data máxima de envio: até 10/09/2020 (23H59)

1) Avalie os seguintes exemplos.

```
lst1 = [x*2 | x <- [1..10], x*2 >= 12]

lst2 = [ x | x <- [50..100], mod x 7 == 3]

lst3 = [ x | x <- [10..20], x /= 13, x /= 15, x /= 19]

lst4=[(x,y)| x <- [1..4], y <- [x..5]]
```

2) Escreva a função quadrados que recebe dois inteiros e retorna os quadrados dos números entre eles. E.g.:

```
> quadrados 4 9
[16,25,36,49,64,81]
```

3) Usando lista por compreensão, escreva a função seleciona_ímpares que recebe um lista de inteiros e retorna uma nova lista com todos os números ímpares presentes na lista de entrada.

```
> seleciona_ímpares [2,5,1,4,7]
[5,1,7]
```

4) Escreva a função tabuada que recebe um valor inteiro e retorna a lista de seus dez primeiros múltiplos. E.g.:

```
> tabuada 7
[7,14,21,28,35,42,49,56,63,70]
```

5) Escreva a função bissextos a seguir que recebe uma lista de inteiros e retorna uma lista com os valores que representam anos bissextos. Dica: use a função bissexto do roteiro anterior.

```
> bissextos [100,400,2020,2021,2022,2024]
[400,2020,2024]
```

6) Usando lista por compreensão, escreva a função sublistas que recebe uma lista formada por sublistas de um mesmo tipo e retorna uma lista com todos os elementos da lista de entrada na mesma ordem, mas no nível da lista principal, sem sublistas.

```
> sublistas [[2,5],[1],[4,7,2]]
[2,5,1,4,7,2]
```

7) Sejam os tipos Data, Emprestimo, Emprestimos e a variável bdEmprestimo do exemplo da Biblioteca. Escreva a função atrasados que recebe um parâmetro do tipo Emprestimos e a Data atual, e retorna uma lista com todos os empréstimos atrasados.

```
type Data = (Int, Int, Int)
type Emprestimo = (String, String, Data, Data, String)
type Emprestimos = [Emprestimo]
bdEmprestimo::Emprestimos
bdEmprestimo =
  [("H123C9","BSI945",(12,9,2009),(20,09,2009),"aberto"),
   ("L433C5","BCC021",(01,9,2009),(10,09,2009),"encerrado"),
   ("M654C3","BCC008",(04,9,2009),(15,09,2009),"aberto")]
> atrasados bdEmprestimo (18,9,2009)
[("L433C5","BCC021",(01,9,2009),(10,09,2009),"encerrado"),
 ("M654C3","BCC008",(4,9,2009),(15,9,2009),"aberto")]
```

8) Escreva a função recursiva npares que recebe uma lista de inteiros e retorna a quantidade de números pares pertencentes à lista.

9) Escreva a função recursiva produtorio que recebe uma lista de números e retorna o produto de todos os seus elementos.

10) Escreva a função recursiva comprime a seguir que recebe uma lista de listas e retorna uma lista contendo todos os elementos das sublistas.

> comprime [[1,2],[3,4,5],[],[6]]

[1,2,3,4,5,6]

11) Escreva a função tamanho a seguir que recebe uma lista polimórfica (de qualquer tipo) e retorna a quantidade de elementos que ela possui.

> tamanho [1,3,5,7,9]

5

12) Usando compreensão de listas, escreva a função uniaoNRec a seguir que faz a união de duas listas de modo que ela mantenha todos os elementos da 1a lista na mesma ordem e no final acrescenta apenas os elementos da 2a lista que não estejam presentes na 1a lista.

> uniaoNRec [1,2,3,4,5,6,7] [2,9,7,10,4]

[1,2,3,4,5,6,7,9,10]

13) Escreva a função recursiva uniaoRec2 a seguir que faz a união de duas listas de modo que ela mantenha todos os elementos da 1a lista na mesma ordem e no final acrescenta apenas os elementos da 2a lista que não estejam presentes na 1a lista.

> uniaoRec2 [1,2,3,4,5,6,7] [2,9,7,10,4]

[1,2,3,4,5,6,7,9,10]