

Programação Funcional
Roteiro de atividades práticas 4
Listas por compreensão e enumeradas

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 o e-mail claudineyrt@gmail.com.

Data de envio: até 01/04/2021 (Quinta) até 23H59

1) Abra o ambiente interativo GHCi e avalie as seguintes expressões.

```
1:[2,3,4]
'a':['b','c','d']
head [1,2,3]
tail [1,2,3]
[1,5,2,3]!!1
[1,5,2,3]!!3
elem 2 [1,5,2,3]
take 2 [1,5,2,3,7]
drop 2 [1,5,2,3,7]
[1,2] ++ [3,4]
[1..10]
[7,6..3]
['b'..'g']
take 5 [1,3..]
sum [1..10]
maximum [1,5,2,3,7]
minimum [1,5,2,3,7]
```

2) Gere por enumeração as seguintes listas:

- a) [5,4,3,2,1]
- b) [a,c,e]
- c) [1,4,7,10,13,16]
- d) [(1,1),(-2,5),(-5,9),(-8,13),(-11,17)] *obs: também é necessário usar a função zip para criar as tuplas

3) Funções que utilizam listas enumeradas

a) Utilizando enumeração, construir uma função que dados dois inteiros a e b construa a lista dos inteiros contidos no intervalo fechado [a,b]. Quando a for igual a b, a função devolve a lista unitária [a]. Quando a > b a função deverá devolver a lista vazia.

b) Utilizando enumeração, construir uma função que dados dois inteiros a e b construa a lista dos inteiros pares contidos no intervalo aberto (a,b). Quando a for igual a b ou a > b a função devolve a lista vazia. (*Dica: verificar se a é par ou ímpar)

4) Abra o ambiente interativo GHCi e avalie as seguintes expressões:

```
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]]
```

5) Usando lista por compreensão, 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]
```

6) 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]
```

7) 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]
```

8) 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]
```

9) 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]
```

10) 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")]
```

11) 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]
```