

Programação Funcional

Laboratório 3: Listas por Compreensão

Envie seu arquivo .hs para bruno.welldm@gmail.com.

Assunto do email: PF Lab 3

Corpo do email: nome do aluno

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
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 * n | n <- [1..10] ]
```

- Gere a lista: [5,4,3,2,1]

2) 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]]
```

```
npares xs = length [x | x<-xs, even x]
> npares [2,5,1,4,7]
```

```
boomBangs xs = [ if x < 10 then "BOOM!" else "BANG!" | x <- xs, odd x]
> boomBangs [12,3,45,6,10,18,9]
```

3) Usando lista por compreensão, gere a lista `listaQuad` contendo os quadrados dos números de 1 a 100.

4) Usando lista por compreensão, gere a lista `listaQuadPares` contendo os quadrados dos números pares de 1 a 100.

5) 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` (Lab.2).

```
> 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. Dica: use a função definida no Lab 2, exercício 8.

```
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")]
```