Correção da Prova 1 Haskell

1)Mostre todos os passos (aplicações da função) na execução da função:

foo 80,15,sendo:

foo x y = if x >= y then 1 + foo (x-y) y else 0

Qual é a finalidade dessa função?

Ela serve para calcular o quociente inteiro da divisão de x por y

Chamada	**Resultado Parcial**	**Explicação**
foo 80 15	1 + foo 65 15	80 ≥ 15
foo 65 15	1 + foo 50 15	65 ≥ 15
foo 50 15	1 + foo 35 15	50 ≥ 15
foo 35 15	1 + foo 20 15	35 ≥ 15
foo 20 15	1 + foo 5 15	20 ≥ 15
foo 5 15	0	5 < 15 (parada)
Resultado	**5**	80 ÷ 15 = 5

2)Mostre todos os passos(aplicações da função) na execução da função:

bar[(1,2),(5,6),(10,20),(2,1)] ,sendo:
bar [] = []
bar ((x,y):xs)= if x>y then (y,x):bar xs else (x,y):bar xs

Qual o tipo inferido para a função?

bar :: Ord a => [(a, a)] -> [(a, a)]

3)Declare uma função que retorne o último elemento de uma lista
Exemplo : ultimo[10,20,30] = 30

resolução:
ultimo [x] = x

```
ultimo (x:xs) = ultimo xs
```

4)Declare uma função que replique todos os elementos de uma lista n vezes, sendo o valor de n um dos parâmetros para a função.

Exemplo: replicar 3 “abc” => “aaabbbccc”
replicar 2 [1,2,3,4] => [1,1,2,2,3,3,4,4]

resolução:

```
replicar _ [] = []
replicar n (x:xs) = replica n x ++ replicar n xs
  where
    replica 0 _ = []
    replica n x = x : replica (n - 1) x
```

5)Considerando que o primeiro elemento de uma Lista está na posição zero, declare uma função que receba como parâmetros, além da lista, a posição inicial e final de um segmento e retorne esse segmento.

Exemplo: fatiar 2 5 [1,2,3,4,5,6,7,8,9,10] => [3,4,5,6]
fatiar 3 4 “abcdef” => “de”
fatiar 2 10 “abcdecfg” => “cdefg”

```
nprimeiros 0 _ = []
nprimeiros _ [] = []
nprimeiros n (x:xs) = x:nprimeiros(n-1) xs

fatiar 0 f (xs) = n primeiros (f+1) xs
fatiar i f (_:xs) = fatiar (i-1) (f-1) xs
```