

Exercícios Práticos

- 1) Escreva uma função *kSmallest* que recebe uma lista de inteiros e um inteiro *k* e retorna uma lista com os *k* menores inteiros, preservando a ordem original em que aparecem.

```
kSmallest :: [Int] -> Int -> [Int]
kSmallest [1, 5, 3, 2, 0] 3 = [1, 2, 0]
```

- 2) Implemente a função *composites* que, dado uma lista *ps* de números primos e uma lista *ns* de números inteiros, retorna uma lista com os inteiros de *ns* que são formados apenas pelo produto de primos de *ps*.

```
composites :: [Int] -> [Int] -> [Int]
composites [2, 3, 7] [4, 5, 6, 7, 10, 14, 15] = [4, 6, 7, 14]
```

P.S.: O número 1 pode ser considerado o produto de 0 primos.

- 3) Combinação é um subconjunto com *p* elementos de um conjunto maior com *n* elementos. Crie uma função *combinations* :: [Int] -> [[Int]] para gerar todas as possíveis combinações sem repetição dos inteiros de uma lista de entrada.

```
combinations [1, 2, 3] = [[], [3], [2], [1], [2, 1], [1, 3], [2, 3], [2, 1, 3]]
combinations [10, -80, 14, 16] =
  [[], [10], [-80], [10, -80], [14], [10, 14], [-80, 14], [10, -80, 14], [16], [10,
  16], [-80, 16], [10, -80, 16], [14, 16], [10, 14, 16], [-80, 14, 16], [10, -80,
  14, 16]]
```

Os exercícios práticos devem ser realizados individualmente e enviados por e-mail com o assunto [IF686EC] EXERCÍCIOS PRÁTICOS 01 para monitoria-if686-ec-l@cin.ufpe.br até as 23:59 de sexta (30.03.2018).

As resoluções dos exercícios devem estar em arquivos diferentes, um arquivo por exercício com os nomes no formato Q[número do exercício].hs