
Número:Nome:

Questão 1: Considere o seguinte excerto de um módulo:

```
module Questao1 where
{-|
...
-}
f :: ...
f l x = procura x li
      where li = zip l [0..]
            procura x [] = Nothing
            procura x ((y,n):ys) | x==y = Just n
                                   | otherwise = procura x ys
```

1. Indique um tipo possível para a função `f`.
2. Documente `f` em Haddock com uma breve descrição do que a função faz e dois exemplos de teste.
3. Defina em HUnit dois testes unitários à função `f`.

```
module Questao1_Spec where

import Questao1
import Test.HUnit

teste1 = ... ~: ... ~=? ...
...
testes = test [teste1, teste2, teste3]
```

Questão 2: Temos várias filas de espera de pessoas e pretendemos agregar todas as filas numa só, atribuindo um número de ordem sequencial a cada pessoa na fila final.

1. Defina uma função **g** que recebe uma lista de listas de nomes de pessoas e calcula uma lista de pares com o nome de cada pessoa e respectiva posição na fila final.
2. Indique o tipo de **g**, complete o comentário da função em Haddock e defina três testes unitários em HUnit.

```
module Questao2 where
{-|
...
== Exemplos:
>>> g [ ["AA", "BB", "CC"], ["DD", "EE", "FF", "GG"] ]
[ ("AA",1), ("BB",2), ("CC",3), ("DD",4), ("EE",5), ("FF",6), ("GG",7) ]
>>> g [ ["AA", "BB", "CC"], [] ]
[ ("AA",1), ("BB",2), ("CC",3) ]
...
-}
g :: ...
g ...
```