

Nome: _____

Nº mecanográfico: _____

- **Duração: 2h + 30m tolerância.**
- **Este exame contém 6 questões e 4 páginas.**
- **Responda às questões no espaço marcado no enunciado.**
- **Pode usar funções auxiliares e/ou do prelúdio-padrão de Haskell**

1. (30%) Responda a cada uma das seguintes questões, indicando **apenas** o resultado de cada expressão.

(a) `length (filter (>5) [2,4..10])` = _____

(b) `tail (1:(5:(4:(3:[]))))` = _____

(c) `[[1,2,3,4],[5,6,7,8],[2,3,4]] !! 1 !! 2` = _____

(d) `takeWhile (>5) [2,4..10]` = _____

(e) `[x+y | x <- [1..3], y <- [x..4]]` = _____

(f) `foldr (*) 2 [1,3..7]` = _____

(g) Sem usar explicitamente a lista dada, defina a seguinte lista em compreensão:

`[(0,1),(1,3),(2,5),(3,7),(4,9),(5,11)]` = _____

(h) Dada a seguinte definição em Haskell:

```
h (x:xs) = (2*x) : h xs
h [x] = []
h [] = []
```

A avaliação de `h [1,2,1,5]` tem como resultado: _____

(i) Indique um tipo admissível para `foldr (++) []`: _____

(j) Indique um tipo admissível para `[(*3), id]`: _____

(k) `data Exp a = V a | Plus (Exp a) (Exp a)`

```
avalua (V x) f = f x
avalua (Plus x y) f = (avalua x f) + (avalua y f)
```

O tipo inferido pelo Haskell para a função `avalua` é: _____

(l) Indique o tipo mais geral de `\(x,y) -> x!!y`:

2. (15%) Acerca de listas - I.

- (a) Defina uma função `uns`, que dada uma lista com `n` elementos, retorne uma lista com `n` 1s.
- (b) Defina uma função `maiores` que, dada uma lista de valores, retorna `True` se todos os elementos da lista inicial são maiores do que os seus vizinhos (à direita) e `False` caso contrário.

Nota: Pode utilizar funções do prelúdio-padrão e/ou listas em compreensão mas não deve usar **recursão**.

3. (20%) Acerca de listas - II. A função `scan` funciona como uma junção entre as funções `map` e `fold`: acumula um valor como a função `fold`, mas retorna a lista de todos os valores intermédios. Os elementos podem ser acumulados da esquerda para a direita (`scanl`), ou da direita para a esquerda (`scanr`) Por exemplo:

`scanl (+) 0 [1,2,3] = [0,1,3,6]`

e

`scanr (+) 0 [1,2,3] = [6,5,3,0]`

- (a) Escreva uma definição recursiva da função `scanr`.
- (b) Escreva uma definição da função `scanr` utilizando a função `foldr`.

4. (10%) Listas infinitas. Defina a função `somasS` que, dada uma função `f` e um valor inicial `i`, determina a série de somas infinita:

`[f(i), f(i)+f(i+1), f(i)+f(i+1)+f(i+2), f(i)+f(i+1)+f(i+2)+f(i+3), ...]`

5. (15%) Considere a seguinte declaração de tipo:

`data Arv a = Vazia | No a (Arv a) (Arv a)`

Nota: As seguintes alíneas deverão ser definidas para árvores de valores numéricos.

- (a) Defina uma função `soma`, que calcule a soma dos valores da árvore dada.
- (b) Defina uma função `valorArv`, que devolva o valor na raiz da árvore dada, ou zero no caso da árvore ser vazia.
- (c) Defina uma função `somasTree`, que devolva uma árvore em que cada nó contém as somas parciais dos valores em cada sub-árvore da árvore dada.

Nota: A soma dos elementos em cada sub-árvore deve ser efectuada uma única vez e **sem recorrer** à função `soma` definida anteriormente.

6. (10%) Responda (**apenas**) a uma das seguintes alíneas, usando indução matemática.

- (a) Considerando as funções definidas na questão anterior, mostre que para qualquer árvore numérica `t`, `soma t = valor (somaTree t)`.
- (b) Considerando as definições das funções `++` e `foldr` dadas nas aulas, mostre que para quaisquer `f`, `v` e `xs`: `foldr f v (xs++ys) = foldr f (foldr f v ys) xs`.