

Ficha 1

Programação Funcional

2015/16

1. Usando as seguintes funções pré-definidas do Haskell:

- `length l`: o número de elementos da lista `l`
- `head l`: a cabeça da lista (não vazia) `l`
- `tail l`: a cauda lista (não vazia) `l`
- `last l`: o último elemento da lista (não vazia) `l`
- `sqrt x`: a raiz quadrada de `x`
- `div x y`: a divisão inteira de `x` por `y`
- `mod x y`: o resto da divisão inteira de `x` por `y`

defina as seguintes funções:

- (a) `perimetro` – que calcula o perímetro de uma circunferência, dado o comprimento do seu raio.
- (b) `dist` – que calcula a distância entre dois pontos no plano Cartesiano. Cada ponto é um par de valores do tipo `Float`.
- (c) `primUlt` – que recebe uma lista e devolve um par com o primeiro e o último elemento dessa lista.
- (d) `multiplo` – tal que `multiplo m n` testa se o número inteiro `m` é múltiplo de `n`.
- (e) `truncaImpar` – que recebe uma lista `e`, se o comprimento da lista for ímpar retira-lhe o primeiro elemento, caso contrário devolve a própria lista.
- (f) `max2` – que calcula o maior de dois números inteiros.
- (g) `max3` – que calcula o maior de três números inteiros, usando a função `max2`.

2. Defina as seguintes funções sobre polinómios de 2^a grau:

- (a) A função `nRaizes` que recebe os (3) coeficientes de um polinómio de 2^o grau e que calcula o número de raízes (reais) desse polinómio.
- (b) A função `raizes` que, usando a função anterior, recebe os coeficientes do polinómio e calcula a lista das suas raízes reais.

3. Vamos representar um ponto por um par de números que representam as suas coordenadas no plano Cartesiano.

```
type Ponto = (Float,Float)
```

- (a) Defina uma função que recebe 3 pontos que são os vértices de um triângulo e devolve um tuplo com o comprimento dos seus lados.
- (b) Defina uma função que recebe 3 pontos que são os vértices de um triângulo e calcula o perímetro desse triângulo.
- (c) Defina uma função que recebe 2 pontos que são os vértices da diagonal de um rectângulo paralelo aos eixos e constroi uma lista com os 4 pontos desse rectângulo.

4. Vamos representar horas por um par de números inteiros:

```
type Hora = (Int,Int)
```

Assim o par (0,15) significa *meia noite e um quarto* e (13,45) *duas menos um quarto*. Defina funções para:

- (a) testar se um par de inteiros representa uma hora do dia válida;
- (b) testar se uma hora é ou não depois de outra (comparação);
- (c) converter um valor em horas (par de inteiros) para minutos (inteiro);
- (d) converter um valor em minutos para horas;
- (e) calcular a diferença entre duas horas (cujo resultado deve ser o número de minutos)
- (f) adicionar um determinado número de minutos a uma dada hora.