

4
16
4
112
12
224
124
344

Instruções para a prova

- A prova é sem consulta;
- A prova dura 1 hora e 30 minutos;
- A prova deve ser implementada usando programação funcional.

Questão 1 : (20 pontos)

Explique o que quer dizer transparência referencial em programas implementados usando programação funcional.

Questão 2 : (40 pontos)

Considere a função *exists*, que recebe como entrada dois parâmetros: uma lista L e uma função F:(Any=>Boolean). A avaliação desta função *exists* retorna verdadeiro se, para pelo menos 1 elemento pertencente à lista, a função F será avaliada para verdadeiro.

- Implemente uma função *exists* usando recursão.
- Qual o tipo de recursão usada? Explique o porquê desta escolha.
- Dê um exemplo de chamada da função criada.

Observações:

- não é permitido usar as funções de percurso de lista nativas de linguagens de programação (ex, *reduceLeft*, *reduceRight*, *filter*, *forAll*, *exists*, *forEach*)
- as seguintes funções de acesso a elementos de uma lista podem ser usadas: *tail(l : List)* – retorna a lista, menos o primeiro elemento; *head(l : List)* – retorna o primeiro elemento da lista; *isEmpty(l : List)* – retorna “true “ se a lista está vazia; *size(l : List)* – retorna o tamanho da lista; *last(l : List)* – retorna o último elemento

Questão 3 : (25 pontos)

- Dada a lista List ("a",4,"6",4,7,12,"a"), implemente uma chamada de função que multiplique os números da esquerda para a direita. A função *def isNumber(param : Any): Boolean* retorna verdadeiro se um parâmetro de entrada é um número.
- Mostre a avaliação da chamada criada.

Observações:

- Exemplo de funções pré-definidas em linguagens de programação funcional que poderão ser usadas: *reduceLeft*, *reduceRight*, *filter*, *forAll*, *exists*, *forEach*.

Questão 4 : (15 pontos)

Dada a função com três parâmetros abaixo, use a técnica de **currying** para reescrevê-la em uma (ou mais) funções que possuam apenas um parâmetro. Implemente também uma chamada a esta função usando 5, 9 e 15 como parâmetros, respectivamente.

```
def concatenaCoordenadas(graus, minutos, segundos : Int) : String =
    graus + 'o ' + minutos + 'm' + segundos + 's'
```