

1. [2.0] Descreva a sequência de passos referente a execução das funções abaixo:

a. $\text{replicate}' :: \text{Int} \rightarrow a \rightarrow [a]$
 $\text{replicate}' 0 _ = []$
 $\text{replicate}' n\ x = x : \text{replicate}' (n-1)\ x$

para $\text{replicate}' 4\ 'a'$

b. $\text{length}' :: [a] \rightarrow \text{Int}$
 $\text{length}' [] = 0$
 $\text{length}' (x:xs) = 1 + \text{length}' xs$

para $\text{length}' [4,7,10,28]$

2. [2.0] Reduza a expressão lambda à forma normal:

$$(\lambda x. \lambda y. (x(\lambda u. \lambda v. u))y) (\lambda a. \lambda b. a)(\lambda c. \lambda d. d)$$

3. [1.0] **multiplos3ou5**: Se listarmos todos os números naturais menores que 10 que são múltiplos de 3 ou 5, teremos 3, 5, 6 e 9. A soma destes números é 23. Desenvolva uma função que retorne a soma de todos os múltiplos de 3 ou 5 menores que 1000.

$\text{multiplos3ou5} :: \text{Int}$

Entrada: não há.

Saída: a soma dos múltiplos de 3 ou 5 menores de 1000.

Importante:

- Essa função não tem um parâmetro de entrada, já que o valor a ser calculado é fixo.
- Não devem ser utilizadas funções padrões de bibliotecas Haskell, exceto funções de ordem superior.
- Sugere-se utilizar compreensão de listas na resolução deste problema.

4. [2.5] **velasBoloAniversario**: Você é responsável pelo bolo de aniversário de uma criança. Você decidiu que o bolo terá uma vela para cada ano de sua idade total. As crianças só serão capazes de apagar as velas mais altas. Dada uma lista de velas de tipo $[Int]$, conte quantas são as velas mais altas.

Exemplo: $\text{velas} = [4,4,1,3]$

A altura máxima das velas é de 4 unidades de altura. Existem 2 delas, então deve ser retornado 2.

$\text{velasBoloAniversario} :: [Int] \rightarrow \text{Int}$

Entrada: lista com as alturas das velas.

Saída: o número de velas que tem maior tamanho.

Importante:

- É possível utilizar funções auxiliares no desenvolvimento deste problema.
- Não devem ser utilizadas funções padrões de bibliotecas Haskell, exceto funções de ordem superior.

5. [2.5] **converterHorario**: Dado um horário no formato 12 horas AM/PM, converta-o para o horário no formato de 24 horas.

Exemplos:

Entrada: "12:01:00PM" Saída: "12:01:00"	Entrada: "12:01:00AM" Saída: "00:01:00"	Entrada: "07:05:45PM" Saída: "19:05:45"
--	--	--

`converterHorario :: [Char] → [Char]`

Entrada: um horário no formato de 12 horas- hh:mm:ssAM ou hh:mm:ssPM

Saída: horário no formato de 24 horas- hh:mm:ss

Importante:

- 12:00:00AM no formato de 12 horas é 00:00:00 no formato de 24 horas;
- 12:00:00PM no formato de 12 horas é 12:00:00 no formato de 24 horas;
- É possível utilizar funções auxiliares no desenvolvimento deste problema;
- Não devem ser utilizadas funções padrões de bibliotecas Haskell, exceto funções de ordem superior, funções da biblioteca Data.Char e funções read e show.
- Considere que a entrada está exatamente no formato indicado, sem espaços em branco ou letras minúsculas.

Regras:

1. Essa prova é individual e sem consulta.
2. A prova deve ser feita até às 23:59 do dia 13/04, e este é o limite de horário para submissão no Moodle.
3. Devem ser entregues 2 arquivos:
 - a. Prova_SeuNome.hs com a resolução das questões de implementação 3, 4 e 5.
 - b. Prova_SeuNome.docx, com as demais questões.
4. As questões 1 e 2 podem ser feitas em papel e depois tirar foto e colar no documento.
5. Questões de implementação devem respeitar o nome da função principal descrito neste documento.