

UNIVERSIDADE FEDERAL DE SERGIPE

CENTRO DE EXATAS E TECNOLOGIA

DEPARTAMENTO DE COMPUTAÇÃO

PROFESSORES: GIOVANNY F. L. PALMA E LEILA M. A. SILVA

TERCEIRA PROVA DE PROGRAMAÇÃO FUNCIONAL

INSTRUÇÕES: Esta prova tem **2:30h de duração, incluindo o tempo de envio** pelo **Google Classroom**. Gere um único arquivo contendo as respostas textuais de todas as questões. Insira seu **nome completo e matrícula** no cabeçalho da sua resposta. O arquivo com as soluções deve ser em formato **PDF**. O nome de seu arquivo deve possuir o formato **SeuNomeUltimoSobrenome-P2.pdf**. Por exemplo, para o nome da professora Leila seria LeilaSilva-P2.pdf. As questões podem ser feitas no editor de texto de sua preferência. **Para cada 5 minutos de atraso que exceder o tempo de prova estipulado o aluno será descontado de -2,0 pontos.**

IMPORTANTE: Nesta prova você **deve usar funções de alta ordem** e funções pré-definidas em qualquer biblioteca Haskell. Você **não poderá usar compreensões, nem recursão apenas, quando funções de alta ordem puderem ser utilizadas para realizar a mesma tarefa**, pois o objetivo desta prova é verificar o conhecimento adquirido com o conteúdo ministrado na terceira unidade do curso.

Nome do Aluno:

Matrícula:

Escolha apenas 4 questões para resolver. Cada questão vale 2,5 pontos.

1. Considere uma lista de triplas do tipo $[(Int, Int, Int)]$. Elabore uma função que dado uma lista de triplas, devolve outra lista de triplas, cujo produto dos elementos da tripla seja um número par. Além disso, os elementos das triplas na lista de saída estão ordenados em ordem crescente. Por exemplo, se a entrada for $[(2, 1, 5), (5, 1, 5)]$ a função devolverá $[(1, 2, 5)]$, pois somente a primeira tripla possui um produto par, no caso 10. **Nesta questão é proibido o uso de compreensão e recursão.**
2. Considere o tipo `Pessoa` definido como `(Nome, DataNascimento)`. Complete a definição deste tipo, utilizando registro para o tipo `DataNascimento`. Em seguida, defina uma árvore binária em que a informação do nó é do tipo `Pessoa`. Considerando esta árvore, elabore uma função que dada uma árvore binária cujos nós são do tipo `Pessoa`, a data corrente e uma idade x , devolve a quantidade de pessoas que possuem idade superior ou igual a x .
3. Qual é o tipo mais geral da função f definida a seguir. Justifique a resposta. Em particular,
 - a. Se o tipo de f tiver alguma restrição de classe, diga o motivo.

- b. Justifique o tipo de retorno de f .
- c. Para cada variável na definição (y , g e gs), explicita seu tipo com relação ao tipo de f e explique por que tem este tipo.
- d. Se o tipo de f tiver ocorrências repetidas de uma mesma variável de tipo, explique por que são as mesmas. Tente explicar em termos de unificação.

$$f\ y\ [] = \max\ y$$

$$f\ y\ (g:gs) = g\ .\ f\ y\ gs$$

4. Elabore um programa que leia uma frase do teclado com pontuação. Em seguida, extraia somente as palavras (sem a pontuação) e imprima as palavras da frase, sendo uma em cada linha, em ordem alfabética. Por exemplo:

Entrada:

Boa prova, boas festas, descensem bastante!

Saída:

bastante

Boa

boas

descensem

festas

prova

5. Prove que

$$\text{sum}\ (\text{triplica}\ (xs++ys)) = 3*(\text{sum}\ xs) + 3*(\text{sum}\ ys)$$

considerando as seguintes definições de funções:

$$[]\ ++\ zs = zs \quad (++) .1)$$

$$(w:ws)\ ++\ zs = w:(ws++zs) \quad (++) .2)$$

$$\text{triplica}\ [] = [] \quad (t.1)$$

$$\text{triplica}\ (w:ws) = 3*w:\text{triplica}\ ws \quad (t.2)$$

$$\text{sum}\ [] = 0 \quad (s.1)$$

$$\text{sum}\ (y:ys) = y + \text{sum}\ ys \quad (s.2)$$