

UNIVERSIDADE FEDERAL DE SERGIPE

CENTRO DE EXATAS E TECNOLOGIA

DEPARTAMENTO DE COMPUTAÇÃO

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

SEGUNDA PROVA DE PROGRAMAÇÃO FUNCIONAL

INSTRUÇÕES: Esta prova tem **2:30h de duração, incluindo o tempo de envio pelo Google Classroom. Cada questão vale 2,0 pontos.** 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 **só pode utilizar recursão** e funções pré-definidas do Prelude e da biblioteca Data.Char **que foram dadas nas aulas disponibilizadas nos slides até a semana 9.** Você **não poderá usar compreensões, outras funções das bibliotecas do Haskell que não estejam nos slides até a semana 9, e/ou funções de alta ordem** na solução das questões, pois o objetivo desta prova é verificar o conhecimento adquirido com o conteúdo ministrado até a segunda unidade do curso.

Nome do Aluno:

Matrícula:

1. Elabore uma função que receba uma lista de números inteiros e retorne uma lista contendo somente os números negativos da lista de entrada.
2. Defina uma função que receba como argumentos uma lista de inteiros e mais três inteiros m , n e p , e que retorne uma outra lista formada por todos os inteiros da lista de entrada que estão nas posições m , n .. p . As posições são contadas a partir de zero. Assim, por exemplo, se a lista for $[7, 1, 3, 2, 5, 1, 6, 8, 10, 11]$ e m , n e p forem 1 , 3 e 7 , a função deve retornar uma lista formada pelos elementos das posições $1, 3, 5$ e 7 , ou seja, a lista $[1, 2, 1, 8]$. Assuma que m , n , e p são maiores ou iguais a zero e que n é estritamente maior que m . Sua definição não pode usar ranges nem a operação $(!!)$ e nem também definir funções auxiliares que façam o mesmo que um range ou $(!!)$.
3. Seja L uma lista de listas de inteiros. Chama-se de *sublistas* as listas que são os elementos de L . Por exemplo, a lista L dada por $[[1, 2], [], [2, 4]]$, possui as sublistas $[1, 2]$, $[]$ e $[2, 4]$. Dada uma lista de listas de inteiros, elabore uma função para determinar se todas as sublistas possuem o mesmo tamanho.

4. Usando a técnica de **pedir mais informação ao amigo**, defina uma função que, dada uma lista de inteiros `xs`, calcule quantas vezes ocorre o maior elemento de `xs`. **Sua solução não pode utilizar nenhuma função pré-definida no Prelude nem definir função auxiliar que faça o mesmo que alguma destas.**
5. Defina uma função que construa, como uma `Picture`, um polígono regular. A função deve aceitar como argumento o número de lados do polígono regular e o tamanho do lado.