

UNIVERSIDADE FEDERAL DE SERGIPE

CENTRO DE EXATAS E TECNOLOGIA

DEPARTAMENTO DE COMPUTAÇÃO

PROFESSORES: GIOVANNY F. L. PALMA

PRIMEIRA PROVA DE PROGRAMAÇÃO FUNCIONAL

INSTRUÇÕES: Esta prova tem **2h de duração e 30 min de tolerância** para o 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-P1.pdf**. Por exemplo, no caso do professor da disciplina seria **GiovannyLuceroPalma-P1.pdf**. As questões podem ser feitas no editor de texto de sua preferência.

IMPORTANTE: Nesta prova você só pode utilizar funções pré-definidas do Prelude e da biblioteca `Data.Char` e compreensões. Caso você seja um aluno que tenha um conhecimento de Haskell anterior ao curso, não poderá usar recursão e/ou funções de alta ordem na solução das questões, nem funções pré-definidas de outras bibliotecas de Haskell, pois o objetivo desta prova é verificar o conhecimento adquirido com o conteúdo ministrado até o momento da avaliação.

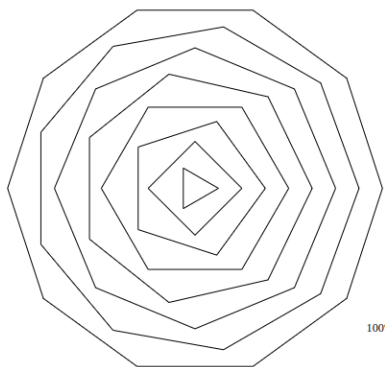
1. Escreva uma função que aceita como parâmetro um inteiro positivo e verifica se o número formado por seus dois últimos dígitos o divide. Caso o parâmetro seja negativo, a função é indefinida com mensagem de erro “número inválido”. Se o parâmetro tiver somente um dígito, a função será indefinida (error) com mensagem “entrada invalida”.
2. Considere uma lista de tuplas em que o primeiro elemento da tupla é o nome de uma pessoa, o segundo é o gênero, o terceiro o ano de nascimento e o quarto o estado civil. O gênero admite os valores ‘F’, ‘M’ e ‘X’, os quais denotam, respectivamente, os gêneros feminino, masculino e demais gêneros. O estado civil admite os valores ‘C’, ‘S’, ‘V’ e ‘O’, denotando, respectivamente, os estados civis casado, solteiro, viúvo e outros estados civis. Declare tipos para todos os dados e elabore uma função que recebe como entrada esta lista de tuplas, um gênero g e o ano corrente, e retorna a média da idade de todas as pessoas casadas, do gênero g e que tenham nascido depois do ano 1985.

3. Elabore uma função para calcular a seguinte soma, em que os valores de n e x são fornecidos como parâmetros da função:

$$\sum_{i=1}^n ix^i = 1x^1 + 2x^2 + 3x^3 + \dots + nx^n$$

4. Supponha uma lista `xs` de inteiros. Defina a função `organize`, tal que dado `xs`, devolve uma lista com todos os números negativos, seguidos de todos os iguais a zero e seguidos dos números positivos. Ex: Para
`organize [1,-2,0,-1,0]` a função devolverá `[-2,-1,0,0,1]`
`organize [1,-2,9,-1,5]` a função devolverá `[-2,-1,1,9,5]`.

5. Defina uma função que permita criar figuras do seguinte tipo



A função deve receber como parâmetro o número de polígonos regulares que serão colocados na `Picture`. Assim, para a construção da figura acima, o parâmetro passado à função deverá ser 8, pois a figura está formada por oito polígonos. Para a construção de um único polígono regular você pode usar a seguinte definição vista em sala de aula.

```
poligonoRegular :: Int -> Double -> Picture
poligonoRegular n r =
    polygon [ (r * cos (i * theta), r * sin (i*theta))
              | i <- [0 .. fromIntegral n - 1] ]
  where
    theta = 2*pi / fromIntegral n
```