UNIVERSIDADE FEDERAL DE SERGIPE

CENTRO DE EXATAS E TECNOLOGIA

DEPARTAMENTO DE COMPUTAÇÃO

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

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 da professora da disciplina seria LeilaSilva-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. Elabore uma função para receber uma nota de aluno e retornar o conceito (A, B, C, D ou E) em que esta nota se enquadra. As notas serão fornecidas apenas com uma casa decimal. Os conceitos seguem a seguinte tabela:

Nota	**Conceito**
9< nota <=10,0	A
8< nota <= 9,0	B
7 < nota <= 8,0	C
6 < nota <= 7,0	D
nota <= 6,0	E

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 de casado, solteiro, viúvo e outros estados civis. Declare tipos para todos os dados e elabore uma função para receber essa lista de tuplas, o ano corrente e uma idade *x,*

e retornar `True` se a maioria das pessoas da lista possui idade superior a *x*; caso contrário, a função deve retornar False.

3. A série infinita a seguir converge ao valor de $\pi/4$.

$$\sum_{i=0}^{\infty} \frac{(-1)^i}{(2i+1)} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \ldots = \frac{\pi}{4}$$

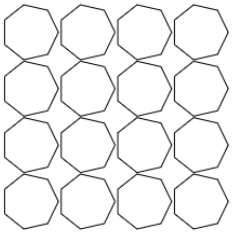
Defina uma função que utilize apenas a soma dos n primeiros termos dessa série para calcular uma aproximação de $\pi/4$. O número n deve ser parâmetro da função.

4. Elabore uma função `nenhumaOcorrencia` tal que dados um caractere `p` e uma lista de palavras `ps`, retorna `True` se toda palavra de `ps` não possui o caractere `p`. Por exemplo,

`nenhumaOcorrencia 'a' ["arara", "gato", "lebre", "asno"]` devolverá `False` e

`nenhumaOcorrencia 'c' ["arara", "gato", "lebre", "asno"]` devolverá `True`

5. Defina uma função que construa uma `Picture` formada por uma grade (matriz) com $m$filas e m colunas onde cada célula contém um mesmo polígono regular. Por exemplo, a grade



tem 4 filas e 4 colunas e cada célula contém um heptágono. Sua função deve aceitar como argumentos o número m, o número de lados do polígono em cada célula, assim como também o raio do polígono.
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
```