

**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. Considere que você trabalhe em uma empresa que realiza mudanças pequenas. A empresa cobra de seus clientes um valor fixo de acordo com o valor da carga, acrescido de um percentual sobre o valor da carga a depender da região do Brasil. Defina uma função que receba uma string representando uma região do Brasil e o valor da carga em reais e retorne o valor em reais do que será cobrado ao cliente, de acordo com as tabelas a seguir:

| <b>Valor da Carga</b>     | <b>Preço Fixo de Partida</b> |
|---------------------------|------------------------------|
| < 10.000,00               | 1.000,00                     |
| >=10.000,00 e < 30.000,00 | 1.500,00                     |
| >=30.000,00 e < 50.000,00 | 3.000,00                     |
| >=50.000,00               | 4.500,00                     |

| <b>Região</b> | <b>Percentual de acréscimo</b> |
|---------------|--------------------------------|
| Sul           | 10,00%                         |
| Norte         | 15,00%                         |
| Nordeste      | 2,0%                           |
| Centro-Oeste  | 7,00%                          |
| Sudeste       | 5,00%                          |

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 e um gênero, e retornar `True` se a maioria das pessoas da lista for do gênero informado e `False`, caso contrário.
3. Elabore uma função para calcular a seguinte soma, em que os valores de  $n$  e  $x$  são fornecidos como parâmetros da função:

$$\sum_{i=0}^n x^i = x^0 + x^1 + x^2 + \dots + x^n$$

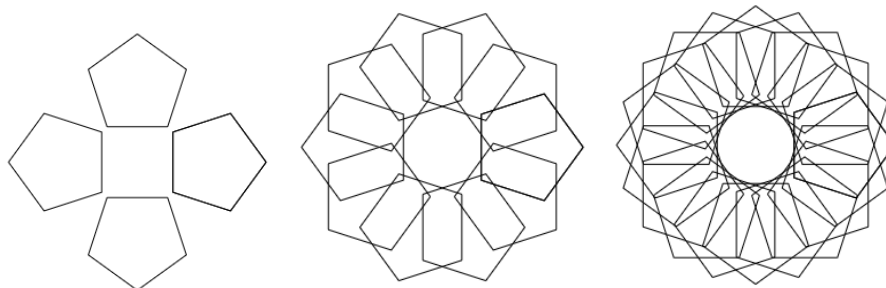
4. Elabore uma função `ultimaOcorrencia` tal que dada uma palavra  $p$  e uma lista de palavras  $ps$ , determina a posição da última ocorrência de  $p$  em  $ps$ . Caso não haja nenhuma ocorrência, a função deve retornar -1. Por exemplo,

`ultimaOcorrencia "arara" ["arara", "gato", "lebre", "arara"]`  
devolverá 4,

`ultimaOcorrencia "arara" ["arara", "gato", "lebre"]` devolverá 1 e

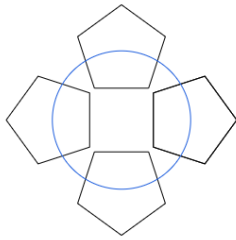
`ultimaOcorrencia "cabra" ["arara", "gato", "lebre", "arara"]`  
devolverá -1.

5. As três figuras a seguir são construídas distribuindo uniformemente vários pentágonos ao longo de uma circunferência imaginária.



Os centros de todos os pentágonos estão dentro desta circunferência imaginária. Assim, para a primeira figura, a circunferência imaginária é aquela desenhada em

azul aqui



Observe que no caso das duas outras figuras, há sobreposição de pentágonos. Isto acontece porque o raio dos pentágonos é fixo e a distribuição deles na circunferência imaginária é uniforme, assim, quantos mais pentágonos são distribuídos, maior é a sobreposição.

Defina uma função que crie este tipo de figuras. A função deve receber como entradas: o número total de pentágonos, o raio, comum a cada pentágono, e o raio do círculo imaginário.

Para construir um pentágono você pode utilizar a seguinte função 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
```