Disciplina: Paradigmas de Programação
Professor: Maicon Rafael Zatelli
Entrega: *Moodle*

Atividade IX - Haskell

Atenção: Faça um ZIP com todos os arquivos de solução. Use o nome do arquivo de maneira a entender qual problema você está resolvendo. Por exemplo, problema1.hs, problema2.hs e assim por diante.

Utilize expressões Lambda em Haskell para resolver os seguintes problemas:

1. Crie uma expressão Lambda que receba dois valores booleanos (x, y) retorne o resultado do "ou exclusivo" (XOR) sobre eles. Leia os valores x e y do teclado.

2. Crie uma expressão Lambda que receba três notas de um aluno (a, b, c), calcule a média e retorne se o aluno foi aprovado ou reprovado. Para um aluno ser aprovado, ele deve possuir nota igual ou superior a 6. Leia as notas dos alunos do teclado.

3. Crie uma expressão Lambda que compute o n-ésimo número de Fibonacci. Leia n do teclado.

4. Crie uma expressão Lambda que resolva uma equação de segundo grau da forma $ax^2 + bx + c$ utilizando a fórmula de Bhaskara. Leia os coeficientes a, b e c do teclado.

5. Crie uma expressão Lambda que dados dois pontos no espaço 3D, (x1, y1, z1) e (x2, y2, z2), compute a distância entre eles. Leia as posições dos pontos do teclado.

6. Crie uma expressão Lambda que receba 3 valores numéricos (a, b, c) e retorne o maior deles. Não utilize nenhuma forma de ordenação. Leia os valores a, b, c do teclado.

7. Utilize a função **map** que recebe como parâmetros uma lista numérica e uma função lambda. A função lambda deve retornar par ou ímpar para cada número, ou seja, uma lista de booleans.

Trabalhando com matrizes em Haskell

- Saber trabalhar com matrizes em Haskell será de grande utilidade para o Trabalho I de Haskell.

8. Pesquise como trabalhar com matrizes em Haskell e resolva os problemas abaixo.
   (a) Crie uma função que receba uma matriz e retorne a soma de seus elementos.

   (b) Crie uma função que receba duas matrizes e retorne a soma delas.

   (c) Dizemos que uma matriz quadrada inteira A_{nxn} é um quadrado mágico se é formada pelos números (não repetidos) de 1 à n^2 e a soma dos elementos de cada linha, a soma dos elementos de cada coluna e a soma dos elementos das diagonais principal e secundária são todas iguais. O exemplo abaixo mostra um quadrado mágico válido. Faça um função que retorne se uma matriz é um quadrado mágico ou não.

$$\begin{bmatrix} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{bmatrix}$$

(d) Dizemos que uma matriz inteira A_{nxn} é uma matriz de permutação se em cada linha e em cada coluna houver $n-1$ elementos nulos e um único elemento igual a 1. O exemplo abaixo mostra uma matriz de permutação. Faça uma função que retorne se uma matriz é uma matriz de permutação ou não.

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

(e) Dizemos que uma matriz quadrada inteira A_{nxn} é um quadrado mágico se a soma dos elementos de uma determinada linha, coluna ou diagonal é sempre igual. Faça uma função que receba como parâmetro uma matriz com alguns números do quadrado mágico já preenchidos e retorne uma matriz com o quadrado mágico completo. Considere que números vão de 1 até 1000 (inclusive) e podem se repetir. As posições da matriz com 0 indicam que aquela posição não está preenchida. Abaixo, são ilustrados dois exemplos de matrizes dadas como entrada e o resultado esperado da sua função. Note que podem existir vários resultados válidos, mas também pode ocorrer de não existir uma solução para a matriz dada. Neste caso, retorne uma matriz toda zerada. PS: note que a definição de quadrado mágico neste exercício é diferente do exercício anterior. <u>DICA</u>: utilize o método da tentativa e erro (*backtracking*).

Entrada:

$$\begin{bmatrix} 0 & 12 & 12 \\ 16 & 10 & 0 \\ 8 & 0 & 0 \end{bmatrix}$$

Resultado:

$$\begin{bmatrix} 6 & 12 & 12 \\ 16 & 10 & 4 \\ 8 & 8 & 14 \end{bmatrix}$$

Entrada:

$$\begin{bmatrix} 0 & 468 & 0 \\ 0 & 522 & 414 \\ 441 & 0 & 549 \end{bmatrix}$$

Resultado:

$$\begin{bmatrix} 495 & 468 & 603 \\ 630 & 522 & 414 \\ 441 & 576 & 549 \end{bmatrix}$$

(f) Crie uma função que receba um tabuleiro de Sudoku (em forma de uma matriz 9x9) já preenchido e diga se a solução dele é válida ou não. Para ser uma solução do problema, cada linha e coluna deve conter todos os números de 1 a 9. Além disso, se dividirmos a matriz em 9 regiões 3 x 3, cada uma destas regiões também deve conter os números de 1 a 9. O exemplo abaixo mostra uma matriz que é uma solução do problema.

$$\begin{bmatrix} 1 & 3 & 2 & 5 & 7 & 9 & 4 & 6 & 8 \\ 4 & 9 & 8 & 2 & 6 & 1 & 3 & 7 & 5 \\ 7 & 5 & 6 & 3 & 8 & 4 & 2 & 1 & 9 \\ 6 & 4 & 3 & 1 & 5 & 8 & 7 & 9 & 2 \\ 5 & 2 & 1 & 7 & 9 & 3 & 8 & 4 & 6 \\ 9 & 8 & 7 & 4 & 2 & 6 & 5 & 3 & 1 \\ 2 & 1 & 4 & 9 & 3 & 5 & 6 & 8 & 7 \\ 3 & 6 & 5 & 8 & 1 & 7 & 9 & 2 & 4 \\ 8 & 7 & 9 & 6 & 4 & 2 & 1 & 5 & 3 \end{bmatrix}$$

(g) <u>DESAFIO</u>: Crie uma função que receba um tabuleiro de Sudoku em seu estado inicial (com alguns valores já preenchidos), em forma de uma matriz 9x9, e resolva o Sudoku. As posições ainda não preenchidas devem conter o valor 0. Ao final, imprima a solução para o Sudoku. <u>DICA</u>: utilize o método da tentativa e erro (*backtracking*).