**Programação ImperaDva e Funcional** 

**Avaliação o2**

**Atenção:**

• **O arquivo correspondente de cada questão deverá iniciar com a letra** q **em minúsculo, seguida pelo número da questão, *underline*, as iniciais do e-mail em minúsculas e com extensão** .hs **Exemplo: considerando que o e-mail é est@cesar.school, o arquivo da primeira questão será** q1\_est.hs **. Qualquer arquivo enviado fora desse padrão será automaDcamente descartado;**

• **Os arquivos deverão ser salvos dentro de um diretório cujo nome será formado pelas iniciais do e-mail em minúsculas. Exemplo: considerando que o e-mail é est@cesar.school, o diretório será** est **Qualquer diretório enviado fora desse padrão será automaDcamente descartado.;**

• **O diretório será compactado em um arquivo**.zip**, e este deve ser disponibilizado para o professor via Classroom. O nome do arquivo será formado pelas iniciais do e-mail em minúsculas. Exemplo: considerando que o e-mail é est@cesar.school, o arquivo a ser entregue será** est.zip . **Qualquer arquivo enviado fora desse padrão será automaDcamente descartado.**

• **Não serão válidas questões que se uDlizarem da importação de módulos.** • **Todas as funções devem apresentar o cabeçalho correspondente**

• **Todas as questões devem ser respondidas com a linguagem de programação Haskell.** • **IdenDficada a cópia de qualquer questão, seja com relação a outre alune ou de alguma outra fonte, TODA a avaliação será descartada**

1. (1.0) Defina a função joinAndMap que recebe como parâmetros duas funções e duas listas, nessa ordem, e retorna como resultado o mapeamento da segunda função sobre a lista resultante da junção das duas listas após aplicação da primeira função. A junção pode ser vista como a união de duas listas de Dpos a e b (quaisquer) aplicando a cada par de elementos a primeira função dos parâmetros, resultando numa lista Dpo c qualquer. Observações: a junção de listas de tamanhos diferentes , o tamanho da lista resultante é igual ao da menor lista. Não use funções predefinidas de Haskell

joinAndMap (+) (abs) [1..10] [-10..(-1)]

[9,7,5,3,1,1,3,5,7,9]

Conteúdo relacionado: funções de alta ordem, polimorfismo, lista

2. (1.0) crie a função remove que dada uma lista de listas, retorne uma lista que contém todos os elementos das sub-listas da lista de entrada, removendo duplicação, mantendo sempre o úlDmo da duplicação que aparecer, e os não duplicados devem aparecer na saída na mesma

ordem em que aparecem nas sub-listas de entrada. Não use funções predefinidas de Haskell, tão pouco compreensão de listas.

\*Main> remove [[1..10], [5..20], [6, 8.. 15]]

[1,2,3,4,5,7,9,11,13,15,16,17,18,19,20,6,8,10,12,14]

\*Main>

Conteúdo relacionado: lista, polimorfismo

3. (2.0) Considere a seguinte assinatura de função:

decom :: [(a -> b -> b)] -> a -> [(b -> b)]

Implemente a função de forma que ela receba uma lista de funções do Dpo indicado e um parâmetro válido para aquelas funções e o retorno dela deve ser uma lista de funções parcialmente aplicadas. Após implementar a função acima, implemente a seguinte função :

aplicador :: [(a -> b -> b)] -> a -> b -> [b]

Essa função deve receber como entrada uma lista funções como indicado e deve retornar uma lista com as respostas para os parâmetros fornecidos. Importante: a resposta para a função aplicador DEVE uDlizar a função decom para aDngir a resposta. Exemplos:

\*Main> aplicador [(+),(-),(\*),(/)] 8 2

[10.0,6.0,16.0,4.0]

Conteúdo relacionado: lista e funções de alta ordem

4. (3.0) Defina uma função polimórfica rmPattern :: Eq w => [[w]] -> [w] -> [[w]], que recebe uma lista de listas, e uma lista, ambas contendo elementos do mesmo Dpo. Essa função deverá procurar o padrão da segunda lista recebida nas primeiras listas, inclusive algum que comece numa lista e termine na próxima. E o retorno será a nova lista de listas com os padrões reDrados. Para esclarecer, um exemplo:

\*Main> rmPattern ["oitudobom","blzcomovaitu","dootimocara"] "tudo"

O padrão "tudo" está conDdo duas vezes nesta lista de listas, uma dentro da primeira das 3 listas, e o outro começa na segunda lista e acaba na terceira. O algoritmo deve reDrar esses padrões e retornar uma nova lista de listas, que ficará assim:

["oibom","blzcomovai","otimocara"]

Mais exemplos:

\*Main> rmPattern ["abcde","abcdefgabc","defghij"] "abcde"

["","fg","fghij"]

**1SPHSBNNJOH 1SPCMFNT**

**1**

multithreaded program that tests your solution to Exercise 3.20. You will create a number of threads— for example, 100—and each thread will request a pid, sleep for a random period of time, and then release the

\*Main> rmPattern [[1,2,3],[4,5,6],[3,4]] [3,4]

pid. (Sleeping for a random period of time approximates the typical pid [[1,2],[5,6],[]]

\*Main>

usage in which a pid is assigned to a new process, the process executes and then terminates, and the pid is released on the process’s termina tion.) On UNIX and Linux systems, sleeping is accomplished through

Não use funções predefinidas de Haskell, tão pouco compreensão de listas. the sleep() function, which is passed an integer value representing the

number of seconds to sleep. This problem will be modi!ed in Chapter 7.

Conteúdo relacionado: lista

Exercise 3.25 in Chapter 3 involves designing an echo server using the Java threading API. This server is single-threaded, meaning that the

5. (3.0) Sudoku é um quebra-cabeça montado sobre um tabuleiro de números 9x9, que server cannot respond to concurrent echo clients until the current client

consequentemente se divide em 9 quadrados 3x3. Nesse jogo, 4 regras devem ser seguidas.

exits. Modify the solution to Exercise 3.25 so that the echo server services

(existem variantes de tamanho, mas devem ser desconsideradas):

each client in a separate request.

1. Apenas números de 1 a 9 são aceitos

Programming Projects

2. Nenhum número pode se repeDr em uma linha

**1SPKFDU 4VEPLV 4PMVUJPO 7BMJEBUPS**

3. Nenhum número pode se repeDr em uma coluna

4. Nenhum número pode se repeDr em um quadrado 3x3

A ***Sudoku*** puzzle uses a 9 × 9 grid in which each column and row, as well as each of the nine 3 × 3 subgrids, must contain all of the digits 1 ⋅⋅⋅ 9. Figure Para melhor visualização: h^p://pt.wikipedia.org/wiki/Sudoku

4.26 presents an ~~example of a valid Sudoku puzzle.~~ This project consists of designing a multithreaded application that determines whether the solution to a Sudoku puzzle is valid.

Represente um tabuleiro Sudoku por uma lista de listas, como indicado a seguir, e codifique a There are several different ways of multithreading this application. One

função checkSudoku [[Int]] -> Bool, que recebe o tabuleiro totalmente preenchido e suggested strategy is to create threads that check the following criteria:

retorna True se está correto, ou seja, segue as 4 regras descritas anteriormente, e False caso

contrário .

• A thread to check that each column contains the digits 1 through 9 • A thread to check that each row contains the digits 1 through 9

Para o tabuleiro abaixo

A lista de lista de inteiros seria

624539187 519728634 837614295 143865729 958247361 762391458 371956842 496182573 285473916

Figure 4.26 Solution to a 9 × 9 Sudoku puzzle.

[[6,2,4,5,3,9,1,8,7],[5,1,9,7,2,8,6,3,4],[8,3,7,6,1,4,2,9,5], [1,3,8,6,5,7,2,9],[9,5,8,2,4,7,3,6,1],[7,6,2,3,9,1,4,5,8], [3,7,1,9,5,6,8,4,2],[4,9,6,1,8,2,5,7,3], [2,8,5,4,7,3,9,1,6]]

Segue um exemplo

\* M a i n > c h e c k S u d o k u [ [ 6 , 2 , 4 , 5 , 3 , 9 , 1 , 8 , 7 ] , [ 5 , 1 , 9 , 7 , 2 , 8 , 6 , 3 , 4 ] , [8,3,7,6,1,4,2,9,5],[1,4,3,8,6,5,7,2,9],[9,5,8,2,4,7,3,6,1], [7,6,2,3,9,1,4,5,8],[3,7,1,9,5,6,8,4,2],[4,9,6,1,8,2,5,7,3], [2,8,5,4,7,3,9,1,6]]

True

\*Main>

Não use funções predefinidas de Haskell, tão pouco compreensão de listas. Conteúdo relacionado: lista