



Laboratório de Linguagens de Programação
Prof. Fabrício Rodrigues Inácio

Trabalho Prático III

1. Objetivo

O objetivo desse trabalho prático é permitir que os alunos pratiquem os conceitos de programação funcional. Os alunos deverão desenvolver algumas aplicações usando a linguagem de programação Haskell (<https://haskell.org>).

2. Instruções

Para cada um dos problemas a seguir, desenvolva um programa em Haskell que os resolva. Não se pode usar nenhuma função pré-definidas da biblioteca de Haskell ou qualquer outra de terceiros a não ser aquelas vistas em aula (nos slides), mas é permitido reimplementar alguma função caso julgue necessário.

- 1) Defina uma função recursiva para o cálculo de potência de dois números inteiros, onde o primeiro número é elevado ao segundo. Não se pode usar o operador de potência (^).

```
Ex.: > potencia 2 3 -- 8=2*2*2  
8
```

- 2) Calcular o somatório dos elementos ímpares de uma lista de inteiros.

```
Ex.: > somaImpares [1,3,2,7,4,6,5] -- 16=1+3+7+5  
16
```

- 3) Substituir todos elementos de um determinado valor de uma lista de inteiros por um outro valor.

```
Ex.: > substituir 1 0 [1,2,1,3,1]  
[0,2,0,3,0]
```

- 4) Verificar se um número é primo.

```
Ex.: > primo 17  
True  
> primo 0  
False
```

- 5) Verifique se um número é perfeito, isto é, é igual a soma de seus divisores (exceto o próprio número).

```
Ex.: > perfeito 28 -- 28=1+2+4+7+14  
True
```

- 6) Função que retorna uma lista com a representação em binário de um número inteiro.

```
Ex.: > binario 20  
[1,0,1,0,0]
```



Laboratório de Linguagens de Programação
Prof. Fabrício Rodrigues Inácio

- 7) Verificar se todos os elementos de uma lista são distintos.
Ex.: `> distintos [1,2,4,2,5]`
False
`> distintos [3,2,1]`
True
- 8) Verificar se duas listas são disjuntas.
Ex.: `> disjuntas [1,2,3] [5,4,6,0]`
True
- 9) Verificar se uma lista de inteiros é palíndromo.
Ex.: `> palindromo [1,2,3,4,3,2,1]`
True
- 10) Calcular todas as somas parciais de uma lista de inteiros.
Ex.: `> somaParciais [1,2,3,4] -- [1,1+2,1+2+3,1+2+3+4]`
`[1, 3, 6, 10]`
- 11) Linearizar uma lista de listas de inteiros.
Ex.: `> linearizar [[1,2], [5], [0,4,2]]`
`[1,2,5,0,4,2]`
- 12) Deslocar todos elementos de uma lista de inteiros k posições para a esquerda.
Ex.: `> shift 3 [1,5,6,7,3,4,1] -- k=3`
`[7,3,4,1,1,5,6]`
- 13) Remover os n últimos elementos de uma lista de inteiros.
Ex.: `> removerFim 2 [1,2,3,4,5,6] -- n=2`
`[1,2,3,4]`
- 14) Dadas duas listas ordenadas de forma crescente, obter a lista ordenada resultante da intercalação delas.
Ex.: `> intercalar [1,5,10] [2,7,9,20,25]`
`[1,2,5,7,9,10,20,25]`
- 15) Desenvolver uma solução para um quiosque de saque eletrônico que, para um determinado valor, deve entregar o menor número de cédulas de R\$1, R\$5, R\$10, R\$50 e R\$100, da menor para a maior.
Ex.: `> trocar 162`
`[1, 1, 10, 50, 100]`

Cada programa deve estar em um arquivo com o nome da função do item seguido de *.hs*. Por exemplo, para o primeiro item o arquivo deve ser nomeado *potencia.hs*, enquanto para o segundo *somaImpares.hs* e assim por diante. **Se não estiver no padrão, o programa será desconsiderado.**

Laboratório de Linguagens de Programação
Prof. Fabrício Rodrigues Inácio

3. Avaliação

O trabalho deve ser feito em grupo de até dois alunos, sendo esse limite superior estrito. O trabalho será avaliado em 15 pontos, onde essa nota será multiplicada por um fator entre 0.0 e 1.0 para compor a nota de cada aluno individualmente. Esse fator poderá estar condicionado a apresentações presenciais a critério do professor.

Trabalhos copiados, parcialmente ou integralmente, serão avaliados com nota **NEGATIVA** (nesse caso -15 pontos), sem direito a contestação. Você é responsável pela segurança de seu código, não podendo alegar que outro grupo o utilizou sem o seu consentimento.

4. Submissão

O trabalho deverá ser submetido até as 23:55 do dia 17/06/2019 (segunda-feira) via sistema acadêmico (Moodle) em pasta específica. Não serão aceitos, em hipótese alguma, trabalhos enviados por e-mail ou por qualquer outra fonte.