

"A compreensão das questões faz parte da avaliação"

Paradigmas de Linguagens de Programação

Bacharelado em Ciência da Computação

1ª Avaliação Individual

29 de setembro de 2016

Nome: _____ Período: _____

Leia atentamente as instruções:

- (i) Durante a avaliação não é permitido consultar a internet ou qualquer outro tipo de material;
- (ii) Enquanto você resolve as questões seu celular/smartphone deve ser deixado com o professor;
- (iii) A resolução da questão 1 deverá constar no verso deste enunciado, para posterior correção;
- (iv) Nas questões 2, 3 e 4 você pode/deve utilizar o **GHC**, e nas questões 5 e 6 o **SWI-Prolog**;
- (v) Para cada uma das questões você deve criar uma pasta chamada **Questão i** ($i=2,3,4,5,6$);
- (vi) Por fim, submeta no Blackboard um arquivo .zip (ou .rar) com o código das suas resoluções.

Questão 1: (2.0 pontos)

Defina (em **Lisp**) uma função recursiva chamada **recursive-reverse** que retorne o reverso de uma lista de elementos.

Exemplo:

```
(write(recursive-reverse(list 2 3 5 7 11 13)))  
(13 11 7 5 3 2)  
(write(recursive-reverse(list(list 1 1 2 3 5 8))))  
((1 1 2 3 5 8))
```

Questão 2: (1.5 pontos)

Implemente (em **Haskell**) uma função **dropEvery** que receba como argumentos uma lista e um valor inteiro N não negativo e exclua repetidamente cada N -ésimo elemento desta lista.

Exemplo:

```
Main> dropEvery [1,3,5,7,9,11] 3  
[1,3,7,9]  
Main> dropEvery "abcdefghijk" 2  
"acegik"
```

Questão 3: (1.5 pontos)

Implemente (em **Haskell**) uma função **ehDecrescente** que recebe uma lista (contendo apenas números inteiros) e retorna **True** caso ela esteja em ordem decrescente, ou **False** caso contrário. Os casos especiais de lista vazia e lista unitária podem ser considerados decrescente.

Exemplo:

```
Main> ehDecrescente [0,-3,-5,-10,-77]  
True  
Main> ehDecrescente [0,-3,258,-5,-10]  
False
```

Questão 4: (1.5 pontos)

Um grupo de n soldados está cercado por uma tropa inimiga e não existe esperança de vitória para eles caso não chegue reforço. Então, decidem que alguém entre eles terá que utilizar o único cavalo que possuem para escapar e buscar ajuda. Para escolher quem será este soldado todos eles formam um círculo, aí escrevem seus nomes em pedaços de papéis que são colocados dentro de um chapéu. Em seguida um dos nomes dentro do chapéu é sorteado e inicia-se uma contagem no sentido horário a partir do soldado cujo nome estava no papel. Quando a contagem atinge o valor k o soldado correspondente é retirado do círculo e a contagem recomeça a partir do soldado seguinte. Novamente é considerado k passos até que mais um soldado seja retirado do círculo. Este processo se repete até que sobre apenas um único soldado, que será o responsável por buscar ajuda para o grupo. Tal situação pode ser representada por uma recorrência matemática famosa que é atribuída a **Flavius Josephus**. Desta forma, se os soldados estão numerados de 1 até n e k é o número de passos, o último soldado restante pode ser determinado por meio da fórmula:

$$f(n, k) = ((f(n - 1, k) + k - 1) \bmod n) + 1, \text{ com } f(1, k) = 1$$

Implemente (em **Haskell**) uma função `josephus` que resolve o problema de escolher o soldado para buscar ajuda para o grupo.

Exemplo:

```
Main> josephus 10 2
5
Main> josephus 100 7
50
```

Questão 5: (1.5 pontos)

Implemente (em **Prolog**) um predicado `mdc(A, B, M)` que é satisfeito quando M é o máximo divisor comum de A e B . (**Dica:** use o algoritmo de Euclides)

Exemplo:

```
?- mdc(15, 5, M).
M = 5.
?- mdc(19, 3, M).
M = 1.
```

Questão 6: (2.0 pontos)

A sequência de Padovan é uma sequência de naturais $P(n)$ definida pelos valores iniciais $P(0) = P(1) = P(2) = 1$ e a seguinte relação recursiva

$$P(n) = P(n - 2) + P(n - 3) \text{ se } n > 2$$

Alguns valores da sequência são 1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16, 21, 28, ... Implemente (em **Prolog**) um predicado `padovan(N, P)` que é verdadeiro sempre que P é o N -ésimo número da sequência de Padovan.

Exemplo:

```
?- padovan(5, 3).
true.
?- padovan(7, 4).
false.
```

“Esta avaliação terá duração máxima de 3 horas”