

“A compreensão das questões faz parte da avaliação”

## Paradigmas de Linguagens de Programação

Bacharelado em Ciência da Computação

1ª Avaliação Individual

30 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 **compress** que elimina duplicação consecutiva de elementos em uma lista.

Exemplo:

```
(write(compress '(a a a a b c c a a d e e e e)))  
(A B C A D E)  
(write(compress '(1 2 3 3 2 4 5 5 7 4 4 4 9 8)))  
(1 2 3 2 4 5 7 4 9 8)
```

---

#### Questão 2: (1.5 pontos)

Implemente (em **Haskell**) uma função **rotate** que receba como argumentos uma lista e um valor inteiro  $N$  e rotacione (circularmente) os elementos desta lista  $N$  posições para a esquerda.

Exemplo:

```
Main> rotate [1,3,5,7,9,11] 3  
[7,9,11,1,3,5]  
Main> rotate "abcdefghijk" 2  
"cdefghijkab"
```

---

#### Questão 3: (1.5 pontos)

Implemente (em **Haskell**) uma função **num2digits** que transforma um inteiro em uma lista de dígitos correspondente ao número. (Dica: use as funções **mod** e **div**)

Exemplos:

```
Main> num2digits 3701  
[3,7,0,1]  
Main> num2digits 0042  
[4,2]
```

---

**Questão 4: (1.5 pontos)**

Um número é dito *chic* se o dígito resultante da soma de seus dígitos ocorre no número. Se o resultado da soma dos dígitos for um número com mais de um dígito, então o processo deve ser repetido até que se obtenha um único dígito. Por exemplo, 1276 é *chic*, pois  $1 + 2 + 7 + 6 = 16$ ,  $1 + 6 = 7$ . Por outro lado, 123 não é *chic* uma vez que  $1 + 2 + 3 = 6$ . Implemente (em **Haskell**) uma função *chic* que retorna **True** se o número for *chic* e **False** caso contrário.

Exemplo:

```
Main> chic 436
True
Main> chic 571
False
```

---

**Questão 5: (1.5 pontos)**

Implemente (em **Prolog**) um predicado *primo(P)* que é satisfeito quando P é um número primo. Lembre-se que um número é primo quando ele é divisível apenas por 1 e ele próprio.

Exemplo:

```
?- primo(7).
true.
?- primo(10).
false.
```

---

**Questão 6: (2.0 pontos)**

Implemente (em **Prolog**) um predicado *remove(LA, E, LB)* que remove apenas a primeira ocorrência do elemento E da lista LA, produzindo a lista LB. O predicado não deve falhar se o elemento não estiver na lista, neste caso ele deve retornar a lista de entrada LA.

Exemplo:

```
?- remove([1,3,5,1,3,11,13], 3, L).
L = [1, 5, 1, 3, 11, 13].
?- remove([2,4,6,8,10], 5, L).
L = [2, 4, 6, 8, 10].
```

---

**“Esta avaliação terá duração máxima de 3 horas”**