

Lista de Exercícios I

Universidade Federal do Ceará
Campus de Quixadá
Ciência da Computação
Programação Funcional
Prof.^o Ricardo Reis
20 de Fevereiro de 2014

Utilizando Haskell, construir as funções seguintes.

1. **menorDeDois**

INPUT: Dois números, x e y

OUTPUT: menor valor entre x e y

2. **menorDeTres**

INPUT: Três números, x , y e z

OUTPUT: menor valor entre x , y e z

3. **fatorial**

INPUT: Um natural n

OUTPUT: O fatorial de n

4. **fibonacci**

INPUT: Inteiro positivo n

OUTPUT: n -ésimo termo da sequência de Fibonacci (iniciando em com 0 e 1)

5. **elemento**

INPUT: Lista u e um natural n

OUTPUT: n -ésimo termo de u

Ex(s):

`elemento 2 [2,7,3,9] ==> 3`

6. **pertence**

INPUT: Lista u e valor x

OUTPUT: Verdadeiro se $x \in u$ e falso do contrário

Ex(s):

`pertence 1 [3,7,4,2] ==> False`

7. **total**

INPUT: Lista u

OUTPUT: total de elementos de u . Não use função `length`.

8. **maior**

INPUT: Lista u

OUTPUT: A chave de valor máximo em u . Não usar função `max`.

9. **frequencia**

INPUT: Lista u e valor x

OUTPUT: Retorna o total de ocorrências de x em u .

Ex(s):

`frequencia 5 [4,5,2,1,5,5,9] ==> 3`

10. **unico**

INPUT: Lista u e valor x

OUTPUT: Verdadeiro se x ocorre exatamente uma vez em u e falso do contrário

Ex(s):

`unico 2 [1,2,3,2] ==> False`

`unico 2 [3,1] ==> False`

`unico 2 [2] ==> True`

11. **maioresQue**

INPUT: Número x e uma lista u de números

OUTPUT: Sublista de u cujos números sejam maiores que x

Ex(s):

`maioresQue 10 [4 6 30 3 15 3 10 7] ==> [30, 15]`

12. **concat**

INPUT: Duas listas a e b

OUTPUT: Concatenação entre a e b

Ex(s):

`concat [] [] ==> []`

`(concat [1,2] [3,4] ==> [1,2,3,4])`

13. **calda**

INPUT: Uma lista u

OUTPUT: Calda de u (u sem a primeira chave)

Ex(s):

`calda [1,2,3,4] ==> [2,3,4]`

14. **corpo**

INPUT: Uma lista u

OUTPUT: Corpo de u (u sem a última chave)

Ex(s):

`corpo [1,2,3,4] ==> [1,2,3]`

15. **unique**

INPUT: Lista u com possíveis chaves repetidas

OUTPUT: Lista com as chaves de u sem repetições

Ex(s):

`[1,2,5,2,5,7,2,5] ==> [1,2,5,7]`

16. **menores**

INPUT: Natural n e lista u

OUTPUT: Lista com os n menores elementos de u na ordem que aparecem em u

Ex(s):

- menores 3 [5,3,1,9,7,2] ==> [3,1,2]
 menores 5 [6,1,2,3,4] ==> [6,1,2,3,4]
 menores 4 [3,1,2] ==> [3,1,2]
17. **alter**
 INPUT: Inteiro n
 OUTPUT: Lista [1, -1, 2, -2, 3, -3, ..., n , - n]
18. **reverso**
 INPUT: Lista u
 OUTPUT: Lista das chaves de u na ordem inversa.
 Ex(s):
 reverso [1,2,3,4] ==> [4,3,2,1]
19. **divide**
 INPUT: Lista u e um natural n
 OUTPUT: Tupla de duas listas, (A, B) , onde A é formada pelas n primeiras chaves de u e B pelos elementos restantes
 Ex(s):
 divide [1,2,3,4] 2 ==> ([1,2], [3,4])
 divide [1,2,3,4] 0 ==> ([], [1,2,3,4])
20. **intercal**
 INPUT: Duas listas a e b
 OUTPUT: Lista com os elementos de a e b intercalados
 Ex(s):
 intercal [1,2,3] [7,8,9] ==> [1,7,2,8,3,9]
 intercal [1,2,3] [8,9] ==> [1,8,2,9,3]
 intercal [] [1,2,6] ==> [1,2,6]
21. **uniao**
 INPUT: Duas listas a e b sem repetição de chaves
 OUTPUT: Lista das chaves de a e b sem repetição
 Ex(s):
 uniao [1,2,3] [2,4,6] ==> [1,2,3,4,6]
 uniao [4,5] [1] ==> [4,5,1]
22. **intersec**
 INPUT: Duas listas a e b sem repetição de chaves
 OUTPUT: Lista das chaves que a e b possuem em comum
 Ex(s):
 intersec [3,6,5,7] [9,7,5,1,3] ==> [3,5,7]
23. **sequencia**
 INPUT: Dois números naturais n e m
 OUTPUT: Lista [$m, m+1, m+2, \dots, m+n-1$]
 Ex(s):
 sequencia 0 2 ==> []
 sequencia 3 4 ==> [4,5,6]
24. **inserir**
 INPUT: Número x e lista, u , de números ordenados ascendentemente
 OUTPUT: Lista de números ordenados ascendentemente, oriunda da inserção apropriada de x em u
 inserir 3 [2,7,12] ==> [2,3,7,12]
25. **isSorted**
 INPUT: Lista de números, u
 OUTPUT: Verdadeiro se u é ordenada e falso do contrário
26. **qsort**
 INPUT: Lista u de objetos ordenáveis
 OUTPUT: Lista ordenada das chaves de u pelo método de ordenação rápida
 Ex(s):
 qsort [7,3,5,7,8,4,4] ==> [3,4,4,5,7,7,8]
27. **rotEsq**
 INPUT: Um natural n e uma lista ou string S
 OUTPUT: Lista S rotacionada n vezes à esquerda
 Ex(s):
 rotEsq 0 "asdfg" ==> "asdfg"
 rotEsq 1 "asdfg" ==> "sdfga"
 rotEsq 3 "asdfg" ==> "fgasd"
 rotEsq 4 "asdfg" ==> "gasdf"
28. **rotDir**
 INPUT: Um natural n e uma lista ou string S
 OUTPUT: Lista S rotacionada n vezes à direita
 Ex(s):
 rotDir 0 "asdfg" ==> "asdfg"
 rotDir 1 "asdfg" ==> "gasdf"
 rotDir 3 "asdfg" ==> "fgasd"
 rotDir 4 "asdfg" ==> "dfgas"
29. **upper**
 INPUT: Uma string S
 OUTPUT: Versão em caixa alta da string S
 Ex(s):
 upper "abc" ==> "ABC"
 upper "a Casa Caiu" ==> "A CASA CAIU"
 upper "tenho 45 ABCs" ==> "TENHO 45 ABCS"
30. **titulo**
 INPUT: String S
 OUTPUT: Versão de S contendo todos os caracteres em caixa baixa exceto aqueles que, por serem iniciais de palavras, devem aparecer em caixa alta
 Ex(s):
 titulo "FuLaNo bElTrAnO silva" ==> "Fulano Beltrano Silva"
31. **selec**
 INPUT: Uma lista qualquer u e uma lista de posições P
 OUTPUT: Lista das chaves de u cujas posições estão em P
 Ex(s):
 selec "abcdef"[0,3,2,3] ==> "adcd"
32. **isPalind**
 INPUT: Uma string qualquer S
 OUTPUT: Verdadeiro se S é um palíndromo e falso do contrário
 Ex(s):

```
isPalind "ana" ==> True
isPalind "123aa321" ==> True
isPalind "cachorro" ==> False
```

os subconjuntos distintos e possíveis de C) partes $[2,3,2,31] \Rightarrow [[],[2],[3],[31],[2,2],[2,3],[2,31],[3,31],[2,2,3],[2,2,31],[2,3,31],[2,2,3,31]]$

33. **primo**

INPUT: Um natural n

OUTPUT: Verdadeiro se n é primo e falso do contrário

34. **sdig**

INPUT: Natural n

OUTPUT: Soma dos dígitos de n

Ex(s):

sdig 328464584658 ==> 63

35. **bubblesort**

INPUT: Lista ordenável u

OUTPUT: Versão ordenada de u pelo método de ordenação em bolhas

36. **compac**

INPUT: Lista de números u

OUTPUT: Lista de listas. Cada lista-componente possui um ou dois elementos. Quando possui dois, representa uma sequência de chaves repetidas de u , sendo o primeiro valor o total de repetições e o segundo a chave que se repete. Quando possui um elemento contém uma chave de u que não se repete.

Ex(s):

compac [2,2,2,3,4,4,2,9,5,2,4,5,5,5] ==> [[3,2],[3],[2,4],[2],[9],[5],[2],[4],[3,5]]

37. **splitints**

INPUT: Lista de inteiros positivos, u

OUTPUT: Tupla de duas listas, (A, B) , onde A e B são respectivamente compostos pelos inteiros ímpares e pares de u

Ex(s):

splitints [1,2,3,4,5,6,7] => ([1,3,5,7],[2,4,6])

38. **perfeito**

INPUT: Número inteiro positivo, n

OUTPUT: Verdadeiro se n for um quadrado perfeito e falso do contrário (Um *quadrado perfeito* é um número inteiro cuja raiz quadrada é também um número inteiro). Não utilizar operadores ou funções que retornem números reais.

39. **base**

INPUT: Dois inteiros positivos, n e b ($1 < b < 37$)

OUTPUT: Representação na base b do inteiro n

Ex(s):

base 17 2 ==> "10001"

base 26 16 ==> "1A"

40. **partes**

INPUT: Lista u

OUTPUT: Conjunto das partes de u (O conjunto das partes de um conjunto C é o conjunto de todos