

Escreva uma implementação de uma lista encadeada simples. A lista encadeada é estrutura de dados fundamental na ciência da computação. Dado a seguinte definição de lista encadeada:

data LinkedList a = Vazia — No a (LinkedList a) deriving (Eq, Show)

Implemente as seguintes funções:

#### **fromList**

Escreva a função

fromList :: [a] → LinkedList a

tal que (fromList xs) devolve uma lista encadeada com os elementos da lista xs seguindo a ordem dos elementos da lista.

fromList [1,2,3] == No 1 (No 2 (No 3 Vazia))

fromList [1,2,3,4] == No 1 (No 2 (No 3 (No 4 Vazia)))

fromList [1,2,3,4,2] == No 1 (No 2 (No 3 (No 4 (No 2 Vazia))))

#### **toList**

Escreva a função

toList :: LinkedList a → [a]

tal que (toList l) devolve uma lista com os elementos da lista encadeada seguindo a ordem dos elementos da lista encadeada.

toList ( fromList [1,2,3,4,2] ) == [1,2,3,4,2]

toList ( fromList [1,6,5,4,2] ) == [1,6,5,4,2]

toList ( fromList [1,6,5,3,2] ) == [1,6,5,3,2]

#### **append**

Escreva a função

append :: a → LinkedList a → LinkedList a

tal que (append x l) devolve uma lista encadeada com o elemento x no final da lista encadeada l.

append 3 (fromList [4,3,1,2]) == No 4 (No 3 (No 1 (No 2 (No 3 Vazia))))

append 8 (fromList [4,3,1,2]) == No 4 (No 3 (No 1 (No 2 (No 8 Vazia))))

append 7 (fromList [4,3,1,2]) == No 4 (No 3 (No 1 (No 2 (No 7 Vazia))))

#### **reverseLinkedList**

Escreva a função

reverseLinkedList :: LinkedList a → LinkedList a

tal que (reverseLinkedList l) devolve uma a lista encadeada l invertida.

reverseLinkedList (No 1 (No 2 (No 3 Vazia))) == No 3 (No 2 (No 1 Vazia))

reverseLinkedList (No 5 (No 4 (No 2 Vazia))) == No 2 (No 4 (No 5 Vazia))