

## 1. Nombres romans (amb recursivitat)

Definiu una funció *roman2int* :: **String** → **Int** que converteixi un número romà al seu enter equivalent *tot usant recursivitat*.

Recordeu que els números romans s'escriuen amb els símbols I, V, X, L, C, D i M, amb valors 1, 5, 10, 50, 100, 500 i 1000 respectivament. En aquest sistema, per obtenir el nombre representat, se sumen els valors dels símbols, excepte els símbols situats a l'esquerra d'un símbol de valor més gran, que es resten.

## 2. Nombres romans (sense recursivitat)

Definiu una funció *roman2int'* :: **String** → **Int** que faci el mateix que la funció anterior però *sense usar recursivitat*: useu una o més funcions d'ordre superior.

## 3. Arrels

La sèrie de Taylor per calcular  $\sqrt{x}$  és:

$$f_1(x) = x$$
$$f_n(x) = \frac{1}{2} \left( f_{n-1}(x) + \frac{x}{f_{n-1}(x)} \right)$$

Definiu una funció *arrels* :: **Float** → [**Float**] que, donat un real  $x$ , retorni la llista infinita dels termes del desenvolupament de Taylor de  $\sqrt{x}$ .

## 4. Més arrels

Escriviu una funció *arrel* :: **Float** → **Float** → **Float** que a partir d'una  $x$  i un  $\epsilon$ , approximi l'arrel de  $x$  amb un error inferior o igual a  $\epsilon$  utilitzant la llista infinita anterior. L'error en el terme  $t_i$  de la sèrie (amb  $i > 1$ ) és la diferència en valor absolut entre  $t_i$  i  $t_{i-1}$ .

## 5. Escriptura d'arbres

Considereu el següent tipus genèric *LTree*  $a$  d'arbres binaris amb valors  $a$  les fulles:

```
data LTree a = Leaf a | Node (LTree a) (LTree a)
```

Feu que els arbres siguin instàncies de la classe **Show** visualitzant-se segons els exemples.

## 6. Creació d'arbres equilibrats

Feu una funció *build* ::  $[a] \rightarrow LTree\ a$  que, donada una llista no buida, construeix el *LTree* equilibrat (a l'esquerra) que conté els elements de la llista en el mateix ordre d'esquerra a dreta. Diem que un arbre es equilibrat a l'esquerra si tots els subarbres tenen el fill esquerre amb la mateixa profunditat que el fill dret o la mateixa més 1.

## 7. Mònades i arbres

Definiu una funció *zipLTrees* ::  $LTree\ a \rightarrow LTree\ b \rightarrow \mathbf{Maybe}\ (LTree\ (a,b))$  que combini els valors de les fulles de dos arbres amb la mateixa estructura.

Si les estructures dels dos arbres no encaixen, retorna **Nothing** i, si encaixen, retorna **Just** de l'arbre que té a cada fulla el parell amb el primer element del primer arbre i el segon del segon arbre en la mateixa posició.

Utilitzeu la notació **do**.

### Exemple d'entrada 1

```
roman2int "I"
roman2int "IV"
roman2int "MCCCXIX"
roman2int "MMXVIII"
```

### Exemple de sortida 1

```
1
4
1319
2018
```

### Exemple d'entrada 2

```
roman2int' "I"
roman2int' "IV"
roman2int' "MCCCXIX"
roman2int' "MMXVIII"
```

### Exemple de sortida 2

```
1
4
1319
2018
```

### Exemple d'entrada 3

```
take 10 $ arrels 4.0
take 10 $ arrels 100.0
```

### Exemple de sortida 3

```
[4.0,2.5,2.05,2.0006099,2.0,2.0,2.0,2.0,2.0,2.0]
[100.0,50.5,26.240099,15.02553,10.840435,10.032578,10.000053,10.0,10.0,10.0]
```

### Exemple d'entrada 4

```
arrel 4.0 0.00001
arrel 100.0 0.1
```

### Exemple de sortida 4

```
2.0
10.000053
```

### Exemple d'entrada 5

```
Node (Leaf 3) (Node (Leaf 8) (Leaf 7))
Node (Leaf 1) (Node (Node (Leaf 3) (Leaf 4)) (Node (Leaf 8) (Leaf 7)))
Node (Leaf "Albert") (Node (Leaf "Gerard") (Leaf "Jordi"))
Leaf 'x'
```

### Exemple de sortida 5

```
<{3},<{8},{7}>>
<{1},<<{3},{4}>,<{8},{7}>>>
<{"Albert"},<{"Gerard"},{"Jordi"}>>
{'x'}
```

### Exemple d'entrada 6

```
build [3, 2, 5]
build [3, 2, 8, 5, 1]
build ['a', 'b', 'c', 'd']
build [[1, 2, 3]]
```

### Exemple de sortida 6

```
<<{3},{2}>,{5}>
<<<{3},{2}>,{8}>,<{5},{1}>>
<<{'a'},{'b'}>,<{'c'},{'d'}>>
{[1,2,3]}
```

### Exemple d'entrada 7

```
let t1 = Node (Leaf "a") (Node (Leaf "b") (Leaf "c"))
let t2 = Node (Leaf 0) (Node (Leaf 1) (Leaf 2))
let t3 = Node (Node (Leaf 1) (Leaf 2)) (Leaf 0)
zipLTrees t1 t2
zipLTrees t1 t3
```

### Exemple de sortida 7

```
Just <{("a",0)},<{("b",1)},{("c",2)}>>
Nothing
```

## Informació del problema

Autoria: Jordi Petit, Albert Rubio, Gerard Escudero

Generació: 2026-02-03T17:06:31.211Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>