

---

## Haskell — Definició de funcions d'ordre superior (1) P90677\_ca

---

Aquest problema explora la definició de funcions d'ordre superior sobre llistes. Implementeu les funcions següents mimetitzant les funcions originals de Haskell però sense usar la funció original (és a dir, no podeu usar *foldl* per implementar *myFoldl* però sí per implementar *myAll*). A més, només podeu utilitzar recursivitat per definir *myFoldl*, *myFoldr*, *myIterate*, *myUntil* i *myZip*.

1.  $myFoldl :: (a \rightarrow b \rightarrow a) \rightarrow a \rightarrow [b] \rightarrow a$
2.  $myFoldr :: (a \rightarrow b \rightarrow b) \rightarrow b \rightarrow [a] \rightarrow b$
3.  $myIterate :: (a \rightarrow a) \rightarrow a \rightarrow [a]$
4.  $myUntil :: (a \rightarrow \mathbf{Bool}) \rightarrow (a \rightarrow a) \rightarrow a \rightarrow a$
5.  $myMap :: (a \rightarrow b) \rightarrow [a] \rightarrow [b]$
6.  $myFilter :: (a \rightarrow \mathbf{Bool}) \rightarrow [a] \rightarrow [a]$
7.  $myAll :: (a \rightarrow \mathbf{Bool}) \rightarrow [a] \rightarrow \mathbf{Bool}$
8.  $myAny :: (a \rightarrow \mathbf{Bool}) \rightarrow [a] \rightarrow \mathbf{Bool}$
9.  $myZip :: [a] \rightarrow [b] \rightarrow [(a, b)]$
10.  $myZipWith :: (a \rightarrow b \rightarrow c) \rightarrow [a] \rightarrow [b] \rightarrow [c]$

### Puntuació

Cada funció puntua 10 punts.

### Exemple d'entrada

```
myFoldl (+) 1 [1..5]
myFoldr (+) 1 [1..5]
take 10 $ myIterate (*2) 1
myUntil (>100) (*2) 1
myMap ("la "++) ["joana", "mireia"]
myFilter odd [1..10]
myAll odd [1,3,5,3,1]
myAny odd [2,4,6,8,10]
myZip [1..4] [1..3]
myZipWith (+) [1..4] [1..3]
```

### Exemple de sortida

```
16
16
[1,2,4,8,16,32,64,128,256,512]
128
["la joana","la mireia"]
[1,3,5,7,9]
```

True

False

[ (1, 1), (2, 2), (3, 3) ]

[2, 4, 6]

## Informació del problema

Autor : Albert Rubio / Jordi Petit

Generació : 2024-05-03 08:39:47

© *Jutge.org*, 2006–2024.

<https://jutge.org>