

## Haskell - Màxim i mínim 3

W84320\_ca

Feu un programa que llegeixi tots els enters de l'entrada estàndard i escrigui el més gran i el més petit. En cas de no haver enters o algun element que no és un enter hauria d'escriure `Nothing`.

Necesitareu la funció `readMaybe` per a la conversió a enters:

```
import Text.Read (readMaybe)
readMaybe :: Read a => String -> Maybe a
```

i les funcions `foldM` i `join` per a que el recorregut pari en trobar un resultat invàlid:

```
import Control.Monad (join,foldM)
==
join :: Monad m => m (m a) -> m a
join $ Just $ Just 2 = Just 2
==
foldM :: (Foldable t, Monad m) => (b -> a -> m b) -> b -> t a -> m b
foldM f a1 [x1, x2, ..., xm]
==
do
  a2 <- f a1 x1
  a3 <- f a2 x2
  ...
  f am xm
```

### Restriccions:

- No podeu utilitzar la recursivitat per resoldre aquest exercici.
- Només podeu utilitzar un `foldM` per recorre la llista d'enters.
- Heu d'aprofitar les instàncies de *functors*, *applicatius* i *mònades* sempre que pugueu.

**Nota:** podeu provar si la vostra funció para utilitzant `undefined`. Quelcom com `f [Just 3, Nothing, undefined]` hauria de donar `Nothing`. Si dona un error és que no està parant.

### Exemple d'entrada 1

```
2 1
3
```

### Exemple de sortida 1

```
Just (3,1)
```

### Exemple d'entrada 2

```
1 a
```

### Exemple de sortida 2

```
Nothing
```

### Exemple d'entrada 3

### Exemple de sortida 3

```
Nothing
```

## **Informació del problema**

Autoria: Gerard Escudero

Generació: 2026-04-23T12:34:33.742Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>