
Haskell — Expressions**P70540_ca**

Es vol tenir un mòdul per a manipular i avaluar expressions d'enters amb operacions de suma, resta, multiplicació i divisió. Per això, es defineix el tipus següent:

```
data Expr = Val Int | Add Expr Expr | Sub Expr Expr | Mul Expr Expr | Div Expr Expr
```

Per exemple, `Add (Val 3) (Div (Val 4) (Val 2))` representa $3 + 4/2$, que s'avalua a 5.

1. Avaluació sense errors (20 punts)

Utilitzant el tipus `Expr`, definiu una operació `eval1 :: Expr → Int` que, donada una expressió, en retorni la seva avalució. Podeu suposar que mai hi haurà divisions per zero.

2. Avaluació amb indicació d'error (30 punts)

Utilitzant el tipus `Expr`, definiu una operació `eval2 :: Expr → Maybe Int` que, donada una expressió, en retorni la seva avalució com un valor **Just**. En el cas que es produeixi una divisió per zero, el resultat ha de ser **Nothing**. Segurament voleu usar la notació **do** sobre la mònada **Maybe** *a*.

3. Avaluació amb text d'error (30 punts)

Utilitzant el tipus `Expr`, definiu una operació `eval3 :: Expr → Either String Int` que, donada una expressió, en retorni la seva avalució com un valor **Right**. En el cas que es produeixi una divisió per zero, el resultat ha de ser **Left** `"div0"` per indicar l'error en qüestió. Segurament voleu usar la notació **do** sobre la mònada **Either** *a b*.

Exemple d'entrada 1

```
eval1 (Val 2)
eval1 (Add (Val 2) (Val 3))
eval1 (Sub (Val 2) (Val 3))
eval1 (Div (Val 4) (Val 2))
eval1 (Mul (Add (Val 2) (Val 3)) (Sub (Val 2) (Val 3)))
```

Exemple de sortida 1

```
2
5
-1
2
-5
```

Exemple d'entrada 2

```
eval2 (Val 2)
eval2 (Add (Val 2) (Val 3))
eval2 (Sub (Val 2) (Val 3))
eval2 (Div (Val 4) (Val 2))
eval2 (Mul (Add (Val 2) (Val 3)) (Sub (Val 2) (Val 3)))
eval2 (Div (Val 4) (Val 0))
```

```
eval2 (Add (Div (Val 4) (Val 0)) (Val 3))
eval2 (Add (Val 3) (Div (Val 4) (Val 0)))
```

Exemple de sortida 2

```
Just 2
Just 5
Just (-1)
Just 2
Just (-5)
Nothing
Nothing
Nothing
```

Exemple d'entrada 3

```
eval3 (Val 2)
eval3 (Add (Val 2) (Val 3))
eval3 (Sub (Val 2) (Val 3))
eval3 (Div (Val 4) (Val 2))
eval3 (Mul (Add (Val 2) (Val 3)) (Sub (Val 2) (Val 3)))
eval3 (Div (Val 4) (Val 0))
eval3 (Add (Div (Val 4) (Val 0)) (Val 3))
eval3 (Add (Val 3) (Div (Val 4) (Val 0)))
```

Exemple de sortida 3

```
Right 2
Right 5
Right (-1)
Right 2
Right (-5)
Left "div0"
Left "div0"
Left "div0"
```

Informació del problema

Autoria: Jordi Petit

Generació: 2026-02-03T17:02:07.912Z

© *Jutge.org*, 2006–2026.
<https://jutge.org>