

---

## Suma de temps

P38424\_ca

---

Tenim definit un tipus per guardar duracions temporals de la forma següent:

**data** Temps = Temps Int Int

on el primer enter correspon a les hores i el segon als minuts. Se suposa que les hores es troben entre 0 i 99 i els minuts entre 0 i 59.

També tenim arbres genèrics definits així:

**data** Arbre a = Arbre a [Arbre a]

1. Feu que *Temps* sigui instància de la classe **Show** tot fent que els dos nombres es mostrin sempre amb dues xifres amb un ':' al mig.

```
ghci> Temps 1 5
```

```
01:05
```

```
ghci> Temps 2 10
```

```
02:10
```

2. Implementeu una funció *suma :: Temps → Temps → Temps* que, donats dos temps, en torni la seva suma.

```
ghci> suma (Temps 1 5) (Temps 2 10)
```

```
03:15
```

```
ghci> suma (Temps 1 35) (Temps 2 40)
```

```
04:15
```

3. Implementeu una funció *sumes :: [Temps] → Temps* sense recursivitat i en notació "point free" (sense paràmetres) que, donada una llista de temps, retorni la suma dels seus elements.

```
ghci> sums [Temps 1 35, Temps 2 40]
```

```
04:15
```

```
ghci> sums [Temps 1 35, Temps 2 40, Temps 9 10]
```

```
13:25
```

4. Implementeu una funció *sumesArbre :: Arbre Temps → Temps* que, donat un arbre general de duracions, calculi la suma dels seus temps. Per exemple:

```
ghci> let f = Arbre (Temps 2 40) [Arbre (Temps 1 1) []]
```

```
ghci> let t = (Arbre (Temps 1 35) [f, Arbre (Temps 9 10) []])
```

```
ghci> sumsArbre f
```

```
03:41
```

```
ghci> sumsArbre t
```

```
14:26
```

5. Feu un programa implementant una acció *main :: IO ()* que:

- (a) obtingui una llista de temps separats per espais de l'entrada estàndar (en una única línia),

- (b) parsegi l'entrada obtenint una llista de temps,
- (c) en calculi la seva suma i,
- (d) mostri el resultat per la sortida estàndar.

### **Informació del problema**

Autoria: Gerard Escudero

Generació: 2026-02-03T17:09:30.721Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>