

---

**Haskell — Funcions amb nombres****P77907\_ca**

---

En aquest problema heu d'implementar una sèrie de funcions en Haskell. No cal que pregunteu si podeu fer servir funcions auxiliars, és evident que sí.

1. Feu una funció *absValue* :: **Int** → **Int** que, donat un enter, retorni el seu valor absolut.
2. Feu una funció *power* :: **Int** → **Int** → **Int** que, donats un enter  $x$  i un natural  $p$ , retorni  $x$  elevat a  $p$ , és a dir,  $x^p$ .
3. Feu una funció *isPrime* :: **Int** → **Bool** que, donat un natural, indiqui si aquest és primer o no.
4. Feu una funció *slowFib* :: **Int** → **Int** que retorni l' $n$ -èsim element de la sèrie de Fibonacci tot utilitzant l'algorisme recursiu que la defineix ( $f(0) = 0, f(1) = 1, f(n) = f(n-1) + f(n-2)$  per  $n \geq 2$ ).
5. Feu una funció *quickFib* :: **Int** → **Int** que retorni l' $n$ -èsim element de la sèrie de Fibonacci tot utilitzant un algorisme més eficient.

**Puntuació**

Cada funció puntua 20 punts.

**Exemple d'entrada 1**

```
absValue (-666)
power 2 3
isPrime 17
slowFib 5
quickFib 40
```

**Exemple de sortida 1**

```
666
8
True
5
102334155
```

**Informació del problema**

Autoria: Albert Rubio / Jordi Petit

Generació: 2026-02-03T17:04:16.020Z

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