

---

## Haskell — Functions with numbers

P77907\_en

---

In this problem you have to implement several functions in Haskell. You do not need to ask permission to write auxiliary functions, of course you can!

1. Write a function `absValue :: Int → Int` that, given an integer, returns its absolute value.
2. Write a function `power :: Int → Int → Int` that, given an integer  $x$  and a natural  $p$ , returns the  $p$ -th power of  $x$ , that is,  $x^p$ .
3. Write a function `isPrime :: Int → Bool` that, given a natural, tells whether it is a prime number or not.
4. Write a function `slowFib :: Int → Int` that returns the  $n$ -th element of the Fibonacci sequence using the recursive algorithm that defines it ( $f(0) = 0$ ,  $f(1) = 1$ ,  $f(n) = f(n-1) + f(n-2)$  for  $n \geq 2$ ).
5. Write a function `quickFib :: Int → Int` that returns the  $n$ -th element of the Fibonacci sequence using a more efficient algorithm.

### Scoring

Each function scores 20 points.

### Sample input

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

### Sample output

```
666
8
True
5
102334155
```

### Problem information

Author : Albert Rubio / Jordi Petit

Translator : Jordi Petit

Generation : 2016-02-16 12:36:43

© Jutge.org, 2006–2016.

<http://www.jutge.org>