

**Exercise 1** *Recursion on Numbers*

In this exercise, you will define two functions to calculate the  $n$ th element of the *Fibonacci sequence*  $1, 1, 2, 3, 5, 8, \dots$ . Each element in this sequence is the sum of the previous two elements.

**Examples:**

```
fib 0 == 1, fib 1 == 1, fib 2 == 2,  
fib 3 == 3, fib 4 == 5, fib 5 == 8, ...
```

1. Write a function `fib1 n` which returns the  $n$ th Fibonacci number. Use pattern matching on numbers for the two base cases. What is the most general type of `fib1`?
2. Write a function `fib2 n` which returns the  $n$ th Fibonacci number, but this time using a guard to identify the two base cases. What is the most general type of `fib2`?
3. What is the difference between pattern matching and guards in Haskell?