

Tail recursion examples

The aim of this class is to familiarise you with writing Erlang programs using tail recursion.

Fibonacci numbers

The Fibonacci sequence is given by 0, 1, 1, 2, 3, 5, ... where subsequent values are given by adding the two previous values in the sequence.

The function `fib/1` that we defined earlier is exponentially complex ... ouch! Define an efficient Fibonacci function `fib/3` using a tail recursion with **two** accumulating parameters that hold the last two Fibonacci numbers, computing the Fibonacci numbers, and give a step-by-step evaluation of `fib(4)`.

Perfect numbers

A positive integer is **perfect** when it is the sum of its divisors, e.g. $6=1+2+3$, $28=1+2+4+7+14$.

Define a function `perfect/1` that takes a positive number `N` and returns a boolean which indicates whether or not the number is perfect. You may well want to use an accumulating parameter to hold the sum of the divisors “so far”.
