Pattern matching and recursion examples

The aim of this class is to familiarise you with writing Erlang programs using more complex pattern matching, as well as contrasting direct and tail recursion.

Shapes

Define a function `perimeter/1` which takes a shape and returns the perimeter of the shape.

Choose a suitable representation of triangles, and augment `area/1` and `perimeter/1` to handle this case too.

Define a function `enclose/1` that takes a shape an returns the smallest enclosing rectangle of the shape.

Summing the bits

Define a function `bits/1` that takes a positive integer `N` and returns the sum of the bits in the binary representation. For example `bits(7)` is `3` and `bits(8)` is `1`.

See whether you can make both a direct recursive and a tail recursive definition.

Which do you think is better? Why?
