

Defining functions over lists – part 2

The aim of this activity is to familiarise you with other ways of defining functions over lists in Erlang, in particular the different way that recursive functions can *construct* lists.

Transforming list elements

Define an Erlang function `double/1` to double the elements of a list of numbers.

Filtering lists

Define a function `evens/1` that extracts the even numbers from a list of integers.

Direct and tail recursion

In each case you could give either a direct recursion or a tail recursive definition. Now give the other kind of definition for `double` and `evens`.

Going further

If you want to try some other recursions on lists try to define functions to give

- the `nub` of a list (that is the list with all repetitions removed);
 - the `median` of a list of numbers: this is the middle element when the list is ordered (if the list is of even length you should average the middle two);
 - the `modes` of a list of numbers: this is a list consisting of the numbers that occur most frequently in the list; if there is just one, this will be a list with one element only.
-