Exercises

Exercise 1

- Write a function `lreduce` that takes a 2-argument function F, a list $[a_1, \dots, a_n]$ and produces $F(\dots F(F(a_1, a_2), a_3) \dots, a_n)$

If the list is empty the function raises an exception

Exercise 2

- Write a curried function substring that takes two parameters and checks whether the first is a substring of the second
- Hint: test for sublists, then use explode

Exercise 3

- Write, in curried form, a function `makeFnList` that takes a function `F` from `D` to `R`. The result should be a function `G` that takes a list of elements of type `D` and that applies `F` to each element in the list

Exercise 4

- In the following exercise, use `map`, `foldr` and `foldl`
  - Define a function that turns an integer list L (e.g., `L=[~1,~1,0,3])` into a list of reals, each of which is the absolute value of the elements in L, (e.g., `[1.0,1.0,0.0,3.0])`)
    - `abs: int -> int` can be used for computing the absolute value

Exercise 5

- In the following exercise, use `map`, `foldr` and `foldl`
  - Define the function `concat`, e.g., `concat`[“`hello`”,” “,”`world`”]=“`hello world`”