Introduction to Functional Programming with Haskell
Exercise Sheet 1

1 Starting Haskell and running your first program

Start Linux and open a terminal. Within the terminal, `cd` to the directory in which you want to store your work. Start the haskell interpreter by typing `ghci`.

Use your favourite text editor to create a file called `Exercises.hs` in the same directory with this content:

```
module Exercises where
 test :: String

test = "Hello World!"
```

Going back to ghci, load your new file and test it like so:

```
> :load Exercises
> test
```

Add your solutions to the following problems to `Exercises.hs`. When you make changes you need to reload the file in ghci:

```
> :reload Exercises
```

2 Exercises

1. Write a function `square :: Int -> Int` which returns the square of a number. Use `square` to write a function `sumsquare :: Int -> Int -> Int` which returns the sum of the squares of its two arguments.

2. Write a (recursive) function `length' :: [a] -> Int` which calculates the length of a list. Think about the base case of the empty list and the case when the list contains elements.

3. Write a function `drop' :: Int -> [a] -> [a]`, where `drop' n xs` returns `xs` with its first n elements removed.

4. Write a function `take' :: Int -> [a] -> [a]`, where `take' n xs` returns the first n elements of `xs` as a list (if `xs` contains less than `n` elements, your function should return all of `xs`).