

Development 8 - Exercises

Unit 4

Exercise 1:

Implement a function

```
let filter = <a>(predicate: (x: a) => boolean) => (l: List<a>): List<a>
```

that inserts in the output list only the elements for which `predicate` returns true

Exercise 2:

Implement a function

```
let map = <a, b>(f: (x: a) => b) => (l: List<a>): List<b>
```

that applies the function `f` to all the elements of `l` and returns a list containing the results.

Exercise 3:

Implement a function

```
let fold = <s, a>(f: (state: s) => (x: a) => s) => (init: s) => (l: List<a>):  
s
```

that applies the function `f` that updates an accumulator of type `s` with each element of `l`, the accumulator itself, and a list `l`, and returns the final value of the accumulator.

Exercise 4:

Implement a function

```
let mapFold = <a, b>(f: (x: a) => b) => (l: List<a>): List<b>
```

that implements `map` only using `fold`

Exercise 5:

Implement a function

```
let filterFold = <a>(predicate: (x: a) => boolean) => (l: List<a>): List<a>
```

that implements **filter** only using **fold**