TD - Programmation Fonctionnelle 2

Louis Thevenet

1. TD5

1.1. Exercice 1

```
module type FL2C = sig
  type zero
  type _ succ
  type 'a fichier

  val open_ : string -> zero fichier
  val read : 'n fichier -> char * 'n succ fichier
  val close : zero succ succ fichier -> unit
end
```

```
module type FLPair = sig
type even
type odd
  type  fichier

  val open_ : string -> (even, odd) fichier
  val read : ('a*'b) fichier -> char * ('b*'a) succ fichier
  val close : (even*odd) fichier -> unit
end
```

1.2. Exercice 2

```
type 'a perfect_tree = Empty | Node of 'a * ('a * 'a) perfect_tree

let rec split : 'a. ('a * 'a) perfect_tree -> 'a perfect_tree * 'a perfect_tree
    =
 fun tree ->
  match tree with
  | Empty -> (Empty, Empty)
  | Node ((l1, l2), subtree) ->
      let t1, t2 = split subtree in
      (Node (l1, t1), Node (l2, t2))
```