

Correction du DS 1

Option informatique, première année

Julien REICHERT

Question 1 : Pour un tableau, `t.(0)` ; pour une chaîne, `s.[0]` ; pour une liste, `List.hd l` ou par un filtrage.

Question 2 : On utilise `&&`.

Question 3 : Pour un tableau, `t.(i) <- x` si `i` est entre 0 (inclus) et la taille de `t` (exclue) et si `x` est du type des éléments de `t` ; pour une chaîne de caractère, c'est désormais impossible, mais on peut admettre l'ancienne version `s.[i] <- x` aux mêmes conditions (donc `x` doit être un caractère) ; pour une liste c'est impossible, et pas seulement parce que l'accès direct à un élément est interdit ; pour une référence, on écrit `r := x` sous réserve là aussi que le type soit compatible.

Question 4 : `print_string "Bonjour"; print_int n;;` (sans les deux points-virgules finaux quand le code est à insérer dans une expression)

Question 5 : Le mot-clé `let` introduit une ou plusieurs (re)création(s) de valeur, qui est(sont) fermée(s) par deux points-virgules si la création est sur toute l'expression, sinon par `in`, ce qui est le cas dans les boucles.

Question 6 : Le balisage des boucles est `do ... done`.

```
(* exo1 : unit -> float array *)
let exo1 () =
  let tab = Array.make 42 0. in
  for i = 1 to 42 do
    tab.(i-1) <- 1. /. float_of_int i
  done; tab;;

(* produit_ref : int ref -> int ref -> int ref *)
let produit_ref a b = ref (!a * !b);;

(* somme_liste : int list -> int *)
let rec somme_liste l = match l with
| [] -> 0
| a::q -> a + somme_liste q;;
```