

# Types, listes et premiers

OPTION INFORMATIQUE - Devoir n° 1 - Olivier Reynet

Toutes les réponses à cet examen sont à fournir en langage OCaml.

## A Types

A1. Indiquez, en justifiant votre réponse, le type des expressions  $e$  suivantes :

- (a) `let e = 42`
- (b) `let e = 42.`
- (c) `let e = true`
- (d) `let e = [3 ; 5]`
- (e) `let e = let a = 3 in let b = 4 in a + b`
- (f) `let e r = let pi = 3.14 in pi *. r *. r`
- (g) `let e = let pi2 = 3.14 in let pi5 = 3.14159 in pi5 - pi2`
- (h) `let e a b = let s = ref 0 in for i = 1 to b do s := !s + a done; !s`
- (i) `let e a b = if a < b then a else b`
- (j) `let k g b = if b = 0 then g else k (g*g)(b-1)`
- (k) `let e = let n = ref 3 in n := !n + 1`
- (l) `let e = let n = ref 0 in while !n < 10 do n := !n +1 done`

## B Listes et filtrage de motif

L'usage des fonctions du module List d'OCaml n'est pas autorisé.

- B2. Écrire une fonction **récursive** de signature `length : 'a list -> int` qui renvoie la longueur d'une liste.
- B3. Quelle est la complexité temporelle de la fonction `length`?
- B4. Écrire une fonction de signature `second : 'a list -> 'a` qui renvoie le deuxième élément d'une liste s'il existe et lève une exception de message `"Action impossible"` dans le cas contraire.  
On suppose maintenant qu'on manipule une liste d'entiers.
- B5. Écrire une fonction de signature `sum : int list -> int` qui renvoie la somme des entiers d'une liste d'entiers.
- B6. Écrire une fonction de signature `parite : int list -> bool list` qui renvoie la liste des parités de chaque élément d'une liste d'entiers. Par exemple, `parite [2 ; 3 ; 5 ; 42]` renvoie `[true; false; false; true]`.
- B7. Écrire une fonction de signature `minimum : int list -> int` qui renvoie le minimum d'une liste d'entiers.

## C Premiers

- C8.** Écrire une fonction de signature `nbdivinf : int -> int -> int` qui renvoie le nombre de diviseurs d'un nombre entier naturel inférieurs ou égaux à un autre. La fonction s'utilise ainsi : `nbdivinf n d` où  $n$  et  $d$  sont des entiers naturels. Par exemple, `nbdivinf 40 9` renvoie 5 car 40 est divisible par les nombres 1, 2, 4, 5 et 8 qui sont inférieurs à 9. On fera attention de bien gérer le cas où le diviseur est zéro en levant une exception. On pourra garantir par une assertion que le diviseur est inférieur ou égal au nombre.
- C9.** Démontrer la correction de la fonction `nbdivinf n d`.
- C10.** En utilisant la fonction précédente, écrire une fonction de signature `nbdiviseurs : int -> int` qui renvoie le nombre de diviseurs d'un nombre.
- C11.** En déduire une fonction de signature `estpremier : int -> bool` qui statue sur le fait qu'un nombre est premier. On rappelle que 1 n'est pas un nombre premier.
- C12.** Quelle est la complexité temporelle de `estpremier` dans le pire des cas ?
- C13.** Écrire une fonction de signature `listediviseurs : int -> int list` qui renvoie la liste des diviseurs d'un nombre. Par exemple, `listediviseurs 40` renvoie `[40 ; 20 ; 10 ; 8 ; 5 ; 4 ; 2 ; 1]`.