

Exercice 1 (4pts) Que retournent les expressions suivantes ? Donner la **valeur et le type**.

1. $3 + 4$
2. $3. +. 4.$
3. `fun x -> x * x`
4. `fun x -> (x, x * x)`
5. `fun x y -> 2 * x + 3 * y`
6. `let f x y = 2 * x + 3 * y`
7. `f 2`
8. `f 2 3`

Exercice 2 (3pts) Le plus grand commun diviseur (pgcd) de deux nombres entiers peut se calculer en utilisant la méthode des *soustractions successives* résumée par la formule suivante.

$$\text{pgcd}(x, y) = \text{si } x = y \text{ alors } x \text{ sinon si } x > y \text{ alors } \text{pgcd}(x - y, y) \text{ sinon } \text{pgcd}(x, y - x)$$

9. Implémenter la fonction `pgcd x y` en utilisant la méthode des soustractions successives.
10. Quel est le type de la fonction `pgcd` ?

Exercice 3 (5pts) Pour cet exercice, on utilisera le type `'a mylist` défini en cours :

```
type 'a mylist = Nil | C of 'a * 'a mylist
```

11. Écrire une fonction `sequence_from_i pred i n` qui retourne sous forme de liste, la séquence ordonnée des n premiers entiers naturels à partir de i qui vérifient le prédicat `pred`.
12. Quel est le type de la fonction `sequence_from_i` ?
13. En déduire une fonction `sequence pred n` qui retourne les n premiers entiers vérifiant le prédicat `pred`.
14. Quel est le type de la fonction `sequence` ?

Exemples :

```
# sequence_from_i (fun x -> true) 5 10;;
- : int mylist =
C (5, C (6, C (7, C (8, C (9, C (10, C (11, C (12, C (13, C (14, Nil))))))))))
# sequence_from_i (fun x -> x mod 2 = 0) 5 10;;
- : int mylist =
C (6, C (8, C (10, C (12, C (14, C (16, C (18, C (20, C (22, C (24, Nil))))))))))
# sequence (fun x -> x mod 2 = 0) 10;;
- : int mylist =
C (0, C(2, C (4, C (6, C (8, C (10, C (12, C (14, C (16, C (18, Nil))))))))))
```

Exercice 4 (8pts)

```
let i_member i set = set i
let i_empty = fun i -> false
let i_evens = fun i -> i mod 2 = 0
let i_odds = fun i -> not (i_member i i_evens)
let i_union s1 s2 = fun i -> i_member i s1 || i_member i s2
```

FIG. 1: Implémentation pour les ensembles d'entiers

On représente un ensemble $E \subseteq \mathbb{Z}^1$, fini ou infini, par sa fonction caractéristique f_E :

$$f_E: \mathbb{Z} \longrightarrow \mathbb{B} \\ i \mapsto f_E(i) \quad \text{avec } f_E(i) \iff i \in E$$

Soit le début d'implémentation donné par la figure 1.

15. Quel est le type de la valeur de la variable `i_evens` ?
16. Que retournent les expressions suivantes (valeur et type) ?
 - a. `i_member 3 i_empty`
 - b. `i_member 2 i_evens`
17. Définir une variable `i_integers` représentant \mathbb{Z} .
18. Écrire une fonction `multiple_of n` qui retourne l'ensemble des entiers naturels multiples de `n`.
19. Écrire une fonction `i_complement set` qui retourne le complémentaire de `set` dans \mathbb{Z} .
20. Écrire une fonction `i_intersection set1 set2` qui retourne l'intersection des ensembles `set1` et `set2`.
21. Écrire une fonction `i_sequence set n` qui retourne la liste des `n` premiers entiers qui appartiennent à l'ensemble `set`².

Exemples :

```
# i_member 3 i_integers;;
- : bool = true
# let s = i_intersection (multiple_of 2) (multiple_of 3);;
val s : int -> bool = <fun>
# i_member 12 s;;
- : bool = true
# i_member 9 s;;
- : bool = false
# i_member 12 (i_complement s);;
- : bool = false
# i_member
77
(i_intersection
  i_odds (i_intersection (i_complement (multiple_of 3)) (i_complement (multiple_of 5))))
- : bool = true
# i_sequence i_evens 10;;
- : int mylist =
C (0, C (2, C (4, C (6, C (8, C (10, C (12, C (14, C (16, C (18, Nil))))))))))
```

FIN

1. \mathbb{Z} est l'ensemble de tous les entiers relatifs, c'est-à-dire les entiers positifs ou négatifs

2. On pourra utiliser la fonction `sequence` de l'exercice précédent.