

# TD n° 1 : Expressions et types

## EXERCICE 1 *Liaisons et fonctions*

Prévoir le résultat des expressions suivantes :

1. `let x = 2 in let x = 1 in x;;`
2. `let x = 2 in let y = x in let x = 1 in 10 * x + y;;`
3. `let y = 2 in let x = y + 1 in let y = x + 2 in x + y;;`
4. `let x = 2 in let x = 2 * x in x = 2 * x;;`
5. `let x = 2 in let x = 2 * x + let x = 2 * x;;`
6. `let x = 2 in let x = 2 * x in let x = 2 * x in x;;`
7. `let (x, y) = (1, 2) in y;;`
8. `let x = 1 in let y = 2 in let (y, x) = (x, y) in y;;`
9. `let f x = 3 * x in f 3 - 1;;`
10. `let f x = 3 * x in (f 3) - 1;;`

## EXERCICE 2 *Portée statique en CAML*

Prévoir la réponse de CAML pour chacune des phrases dans la séquence suivante :

```
OCAML
let a = 2;;
let f x = a * x;;
f 1;;
let a = 3 in f 1;;
let a = 3;;
f 1;;
```

## EXERCICE 3

Que fait ce programme ? Quel est son type ?

OCAML

```
let maximum (a, b, c) =
  let m = a in
  if b > a then
    if c > b then m = c else m = b
  else
    if c > a then m = c else m = a
;;
```

## EXERCICE 4

Écrire une fonction `meme_parite` qui prend en argument un couple d'entiers et qui vérifie qu'ils ont même parité.

## EXERCICE 5

On suppose que `test1`, `test2` et `test3` sont des booléens. Simplifiez l'expression suivante, en utilisant `&&` et `||` et `not` :

OCAML

```
if test1 = false then
  if test2 = false then test3 else false
else
  if test3 = true then true else true = false
;;
```

## EXERCICE 6

La fonction prédéfinie `int_of_float` renvoie la troncature d'un flottant. Écrire une fonction `round` : `float` -> `int` arrondissant un flottant à l'entier le plus proche.