TD 1 - Les bases

Questions de types

Exercice 1 Évaluer la suite d'expressions suivante :

```
let x=0;;
x=0;;
x;;
0;;
2+2;;
x=1;;
if x=1 then 0 else 1;;
let x=1 and y=2;;
let x=y;;
x=y;;
let z=1 and y=z;;
let x=y+1;;
let x=4 in y=4;;
x;;y;;
```

```
let y=let x=0 in 4;;
let y=let x=0 and y=4;;
let y=if x=0 then 4
  else 5;;
let y=let x=1 in
  if x=0 then 4 else 5;;
x;;
y;;
let x=2 and
   y=x+2;;
let x=1 in y=x+2;;
x;;
let y = let x=1 in x+2;;
(x,y);;
```

Exercice 2 Donner un exemple d'expression OCaml du type demandé :

1. `bool * int`
2. `unit`
3. `int -> float`
4. `bool -> int`
5. `(int -> int) -> int`

Quelques fonctions

Exercice 3 La fonction ou exclusif n'existe pas en OCaml, écrire cette fonction de type `bool * bool -> bool`

Exercice 4 Écrire une fonction `multiple` :

- En entrée : un entier a.
- En sortie : 1 si a est multiple de 3 et 0 sinon.

1. Écrire une version utilisant l'instruction `if`
2. ★ Écrire une autre version n'utilisant pas de test, mais juste une formule.

Correction 1 Tapez et vérifez :)

Correction 2 Quelques propositions :

1.
```
# true, 2;;
- : bool * int = (true, 2)
```

2.
```
# ();;
- : unit = ()
```

ou encore

```
# print_string "coucou";;
coucou- : unit = ()
```

3.
```
# float_of_int;;
- : int -> float = <fun>
```

4.
```
# let f(x) = if x = true then 17 else 32;;
val f : bool -> int = <fun>
```

5.
```
# let g(f) = f(1)+4;;
val g : (int -> int) -> int = <fun>
```

Correction 3 Il y a de multiples solutions avec des tests, par exemple :

```
let xou(a, b) =
        if (a, b) = (true, true) || (a, b) = (false, false)
        then false
        else true
;;
```

ou plus court, en donnant un nom au couple paramètre :

```
let xou c =
        if c = (true, true) || c = (false, false)
        then false
        else true
;;
```

la plus courte est surement celle-ci :

```
let xou(a, b) = (a != b);;
```

Cependant cette fonction est de type `'a * 'a -> bool`. En effet, rien n'oblige avec cette déclaration à ce que `a` et `b` soient des booléens. On peut forcer le type si on le souhaite.

```
let xou((a:bool), b) = (a != b);;
```

Correction 4 1. Avec un test :

```
let multiple a = if a mod 3 = 0 then 1 else 0;;
```

2. Sans test, si on note a la variable de départ, et b le reste dans la division de a par 3, alors $b \in \{0, 1, 2\}$. Il s'agit donc de trouver une fonction f qui vérifie $f(0) = 1$, $f(1) = 0$ et $f(2) = 0$. On peut chercher une solution sous forme d'un polynôme de degré 2. Les deux dernières conditions nous informent que 1 et 2 sont des racines de f. Donc $f(x) = a(x-1)(x-2)$ enfin $f(0) = 1 \iff a \times (-1) \times (-2) = 1 \iff 2a = 1 \iff a = \dfrac{1}{2}$. Ainsi $f(x) = \dfrac{(x-1)(x-2)}{2}$. Finalement, voici une fonction répondant à la question :

```
let multiple a = let b = a mod 3 in (b-1)*(b-2)/2;;
```