

TP n° 3 : Récursivité



La primitive `trace` permet de « tracer » le déroulement d'une fonction et est très utile pour visualiser les appels récursifs.

OCAML

```
let rec factorielle n =
  if n = 0 then
    1
  else
    n * factorielle (n - 1)
;;

#trace factorielle;;

factorielle 10;;

#untrace factorielle;;

factorielle 50;;
```

On pourra utiliser la fonction `trace` dans les exercices suivants afin de bien comprendre le comportement des fonctions récursives.

EXERCICE 1

Écrire une fonction `mult : int -> int -> int` telle que `mult x y` calcule le produit de x par y sans effectuer de multiplications (*i.e.* sans utiliser le symbole « `*` »).

EXERCICE 2

Écrire une fonction `somme_termes` qui, à une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ et à un entier n , associe $\sum_{k=0}^n f(k)$. Calculer la somme des cubes des entiers naturels inférieurs ou égaux à 100.

EXERCICE 3

Pour certains arguments, une fonction peut ne pas être définie. On le signale en CAML à l'aide de `failwith "message"`, ce qui va déclencher une exception (ici `Failure "message"`) et interrompre le programme¹.

1. Écrire deux fonctions récursives `quotient` et `reste` qui, à partir de deux entiers positifs a et b , renvoient respectivement le quotient et le reste de la division de a par b , sans utiliser bien sûr l'opérateur de division ou l'opérateur `mod`.
2. Écrire une fonction `divmod` qui calcule les deux à la fois et renvoie le couple (*quotient*, *reste*).

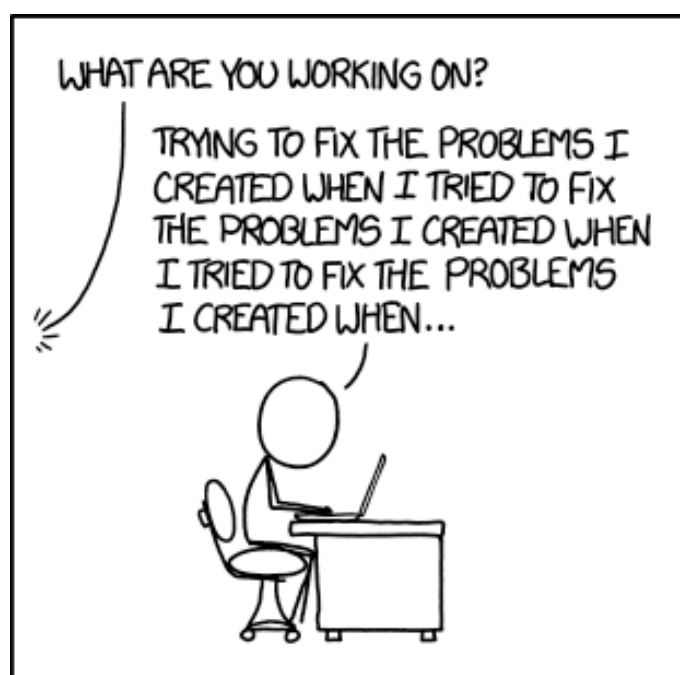
Bonus : Écrire une version de `divmod` dans laquelle a et b peuvent être négatifs.

EXERCICE 4 *Moyenne arithmético-géométrique (à faire avec CAML)*

Soit $a, b \in \mathbb{R}_+$, on définit deux suites positives $(u_n)_{n \in \mathbb{N}}$ et $(v_n)_{n \in \mathbb{N}}$, de premiers termes $u_0 = a$ et $v_0 = b$ et satisfaisant les relations de récurrence :

$$u_{n+1} = \sqrt{u_n v_n} \quad v_{n+1} = \frac{u_n + v_n}{2}$$

1. Montrer que les suites $(u_n)_{n \in \mathbb{N}^*}$ et $(v_n)_{n \in \mathbb{N}^*}$ sont adjacentes.
2. Pour $a = 1$ et $b = 2$, calculer u_{10} et v_{10} . A-t-on $(u \ 10) = (v \ 10)$? Expliquer.
3. Écrire une fonction `suites` telle que `suites a b` renvoie le couple des suites $((u_n)_{n \in \mathbb{N}}, (v_n)_{n \in \mathbb{N}})$. Quel est son type?
4. En utilisant la fonction précédente, écrire une fonction `termes` telle que `termes a b n` renvoie le couple des n -ième termes (u_n, v_n) .
5. Calculer u_{10} et v_{10} pour $(a, b) \in \{(1, 1000), (2, 487), (500, 501)\}$.



<https://xkcd.com/1739/>

1. L'utilisation des exceptions en dehors de ce cas simple est hors programme, mais très utile.