

Programmation : TP 4

Objectifs : [Ocaml] Fonctions récursives, récursivité terminale.

Interdiction...

... de programmer dans le style impératif dans ce TP. En conséquence,

ne pas utiliser de boucle `for` ou `while` ni de références.

[O'Caml] PGCD de deux nombres

Définir une fonction qui calcule un PGCD (Plus Grand Diviseur Commun) en utilisant la propriété suivante (EUCLIDE) : $PGCD(m, n) = PGCD(n, r)$ où r est le reste dans la division euclidienne de m par n (l'opérateur infix `mod` permet de calculer ce reste).

```
#pgcd 35 28;;
- : int = 7
```

Montrer que cette fonction termine (c'est-à-dire ne boucle pas indéfiniment).

[O'Caml] Conjecture de Syracuse (2ème épisode)

Rappelons l'énoncé de cette conjecture (déjà vue en TP3). Soit la suite d'entiers naturels u_n définie par son premier terme u_0 , et par :

$$u_{n+1} = \begin{cases} \frac{u_n}{2} & \text{si } u_n \text{ pair} \\ 3u_n + 1 & \text{sinon.} \end{cases}$$

On conjecture que $\forall u_0, \exists n_0$ tel que $u_{n_0} = 1$

Cette conjecture énoncée par L. Collatz (c.f. TP 3) a provoqué l'intérêt de nombreux mathématiciens (ce qui fait qu'on la connaît aussi sous le nom de problème d'Ulam ou de Kakutani). Tout un vocabulaire s'est développé autour de ce problème. On appelle *vol* la séquence de nombres entre le premier terme u_0 et le terme $u_{n_0} = 1$. L'entier n_0 est la *durée* du vol. L'*altitude* d'un vol est le plus grand terme de la suite. Un vol de terme initial u_0 a une altitude record si aucun vol de terme initial inférieur n'atteint une altitude aussi haute, et une durée de vol record si aucun vol de terme initial inférieur n'est aussi long.

1. Écrire une fonction récursive qui renvoie la durée de vol. Votre fonction est-elle récursive terminale ?

```
#duree 76;;
- : int = 22
```

2. Modifier la fonction précédente pour qu'elle renvoie la durée du vol et son altitude.

Pour renvoyer deux valeurs, une fonction doit renvoyer un couple : un couple est construit en plaçant les deux expressions entre parenthèses, séparées par une virgule : `(n, altimax)`.

```
#syracuse 76;;
- : int * int = (22, 88)
```

3. En considérant les termes initiaux compris entre 1 et une borne m donnée, cherchez le terme initial pour lequel la durée de vol est maximale. S'il y a ex-aequo, retenir le plus petit. La fonction doit renvoyer un couple (`terme initial`, `duree maximale`).

```
#duree_max 50;;
- : int * int = (27, 111)
```

4. De même, chercher le terme initial pour lequel l'altitude est maximale. La fonction doit renvoyer un couple (terme initial, altitude maximale).

```
#alti_max 50;;  
- : int * int = (27, 9232)
```

5. Construire une fonction qui calcule et affiche tous les records d'altitude pour les entiers compris entre 1 et une borne m donnée.

```
#records_alti 1000;;  
1 : 1  
2 : 2  
3 : 16  
7 : 52  
15 : 160  
27 : 9232  
255 : 13120  
447 : 39364  
639 : 41524  
703 : 250504  
- : unit = ()
```