

## TD10 : Fonctions OCaml

**Question 1 :** Écrire une fonction `nb_de_chiffres : int -> int` qui donne le nombre de chiffres de son argument (par convention 0 possède un chiffre).

**Question 2 :** Écrire une fonction `croissante : ('a -> 'b) -> 'a -> 'a -> bool` qui teste si l'évaluation de la fonction fournie en premier argument est supérieure sur le troisième argument que sur le deuxième.

**Question 3 :** En s'inspirant de la formule

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h},$$

écrire une fonction `derivee : (float -> float) -> float -> float` qui calcule une valeur approchée de la dérivée d'une fonction en un point.

**Question 4 :** Écrire une fonction `multiple : int -> int list -> int list` telle que `multiple n lst` s'évalue en une liste de la même longueur que `lst` et dont les maillons contiennent les valeurs des maillons de `lst` multipliées par `n`. En déduire une fonction `double : int list -> int list` telle que `double lst` s'évalue en une liste de la même longueur que `lst` et dont les maillons contiennent les doubles des valeurs des maillons de `lst`.

**Question 5 :**

1. Écrire une fonction `insertion : 'a -> 'a list -> 'a list` qui prend en argument une valeur et une liste supposée triée de valeurs du même type, et crée une nouvelle liste où la valeur est insérée à la bonne place.  
Exemple :

```
utop # insertion 0 [-1; 1; 2; 3; 3; 4; 5; 9];;  
- : int list = [-1; 0; 1; 2; 3; 3; 4; 5; 9]
```

```
utop # insertion 6 [-1; 1; 2; 3; 3; 4; 5; 9];;  
- : int list = [-1; 1; 2; 3; 3; 4; 5; 6; 9]
```

2. Écrire une fonction qui fait le tri par insertion.
3. Prouver que votre fonction possède une correction totale.