

TD machine n°1

Exercice 1. Écrire la fonction `est_multiple` qui prend comme argument deux entiers et renvoie le booléen `true` si le premier argument est un multiple du second.

Exercice 2. Écrire une fonction `somme_f_i` de deux arguments dont le premier est du type `float` et le second du type `int` et dont la valeur est un `float` qui vaut la somme des deux nombres.

Exercice 3. Écrire une fonction `f` dont le seul argument est de la forme $(n, (x, y))$ et dont la valeur est indéfinie si `n` est négatif ou égale à la somme $\sum_{i=0}^n (x - i)(y + i)$ si `n` est positif ou nul.

Exercice 4. Écrire une fonction `neg_bf` qui prend en argument une fonction `f : bool -> bool` et qui rend un booléen. La valeur de `neg_bf f` est `true` si `f` vaut `true` sur au moins une valeur booléenne et `false` sinon.

Exercice 5. Écrire une fonction `double_argument` qui prend en argument une fonction `f` (de type `'a -> 'a -> 'b`) et un argument `x` (de type `'a`) et qui renvoie le résultat de l'application de `f` sur l'argument `x` répété deux fois (soit `f x x`).

En utilisant cette fonction, définir une fonction `seize_fois : string -> string` (qui renvoie la concaténation de 16 copies de la chaîne de caractères en entrée) en n'utilisant (`^`) que quatre fois.

Exemple :

```
1 seizeFois "ab";;  
2 - : string = "ababababababababababababababab"
```

Exercice 6. Pour toutes les questions de cet exercice, il est interdit d'utiliser la fonction puissance de Ocaml `** : float -> float -> float`. Il est également interdit de résoudre l'exercice en utilisant la conversion d'entier en chaîne de caractères.

1. Définir la fonction `de_b_en_10 : int -> int -> int` qui convertit un entier d'une base `b` en base 10. Exemple :

```
1 de_b_en_10 1101 2 ;;  
2 - : int = 13
```

2. Définir la fonction `de_10_en_b : int -> int -> int` qui convertit un entier de base 10 en base `b`.

```
1 de_10_en_b 10 2  
2 1010
```

3. Pour finir, combiner ces deux fonctions pour définir la fonction

`de_b_en_b2 : int -> int -> int -> int` qui convertit un entier d'une base `b` à une base `b'`. La fonction `de_b_en_b2` prend trois paramètres : la valeur à convertir, sa base et la base d'arrivée.

```
1 de_b_en_b2 110 2 8  
2 6
```