

# Évaluation 1

*Lundi 8 décembre de 17h à 18h*

## Consignes

- Tous les programmes seront documentés.
- Pour chacune des fonctions construites, on effectuera une batterie de tests de manière à couvrir les différents cas possibles, en particulier les cas « limite ».

## Exercice 1 (Affichage des carrés des entiers de 2 à 20 à la console)

Écrire un programme Python qui affiche à la console les carrés des entiers de 2 à 20.

## Exercice 2 (Produit des termes d'une liste d'entiers)

Écrire une fonction Python nommée `produit_liste`

- d'argument une liste `L` d'entiers ;
- qui renvoie 1 si la liste `L` est vide, et le produit de tous les termes de la liste `L` sinon.

## Exercice 3 (Nombres parfaits)

1. Si `n` est une expression de type `int`, si `d` est expression type `int` supérieure ou égale à 1, alors

$$n\%d$$

est l'expression de type `int` qui a pour valeur le reste de la division euclidienne de `n` par `d`. En utilisant l'opérateur `%`, écrire une fonction Python nommée `divise`

- d'argument un couple `(a, b)` d'expressions de type `int`, où `a` est supérieure ou égale à 1 ;
- qui retourne `True` si `a` divise `b`, et `False` sinon.

2. Écrire une fonction Python nommée `somme_diviseur`

- d'argument une expression `n` de type `int`, supposée supérieure ou égale à 1 ;
- qui retourne la somme des diviseurs de `n` qui sont supérieurs ou égaux à 1.

On pourra appeler la fonction `divise` au sein de la fonction `somme_diviseur`.

3. Un nombre entier naturel non nul `n` est dit parfait si la somme des diviseurs de `n` qui sont supérieurs ou égaux à 1 est égale au double de `n`. Par exemple, 6 est un nombre parfait. Écrire un programme Python qui affiche à la console les nombres parfaits compris entre 1 et 10000 (au sens large). On pourra appeler la fonction `somme_diviseur` dans le corps de ce programme.