ISM MPSI/PCSI, Informatique
Année 2026/2027

TP Informatique 08

⚠ Télécharger le fichier `TP08_eleves.py` sur *Cahier de prépa*, compléter les fonctions définies au fur et à mesure en enlevant la mot clé `pass`.
Après avoir vérifié que votre script s'exécute correctement, vous déposerez votre fichier `.py` sur cahier de prépa (pas de consignes concernant le nom du fichier).

Exercice 1

1. Écrire une fonction `somme(n)` prenant en argument un entier naturel n et renvoyant la somme : $\sum_{1\leqslant 3k+1\leqslant n} (3k+1)^2$.
   Le choix du type de boucle utilisée est libre.
   Par exemple, `somme(10)` renvoie 166.
2. La suite de Fibonacci est définie par la donnée de ses premiers termes $u_0 = 0$ et $u_1 = 1$, et la relation de récurrence :
   $$\forall n \in \mathbb{N},\ u_{n+2} = u_{n+1} + u_n$$
   Écrire une fonction `fibo(n)` prenant en argument un entier naturel n et revoyant la valeur de u_n.
   On impose l'usage d'une boucle inconditionnelle.
   Par exemple, `fibo(10)` renvoie 55.
3. La suite $\left(\frac{1}{n!}\right)_{n\in\mathbb{N}}$ tend vers 0.
   Écrire une fonction `seuil(eps)` prenant en argument un flottant *eps* et renvoyant le plus petit entier n tel que $\frac{1}{n!} \leqslant eps$.
   On impose l'usage d'une seule boucle conditionnelle, et on interdit l'usage d'une fonction auxiliaire, locale ou globale, pour calculer la factorielle.
   Par exemple, `seuil(1e-10)` renvoie 14.

Exercice 2

Autour des nombres premiers.

Diviseurs et primalité.

1. Écrire une fonction `liste_div(n)` prenant en argument un entier naturel non nul n et renvoyant la liste des diviseurs positifs de n.
   Par exemple `liste_div(12)` renvoie `[1,2,3,4,6,12]` et `liste_div(5)` renvoie `[1,5]`.
2. On rappelle qu'un entier naturel n est *premier* s'il admet exactement deux diviseurs positifs, en l'occurrence 1 et n.
   Écrire une fonction `est_premier(n)` prenant en argument un entier naturel non nul n et renvoyant `True` si n est premier, et `False` sinon.
   Cette fonction doit impérativement utiliser la fonction `liste_div`.

3. Écrire une fonction `liste_premiers(n)` prenant en argument un entier naturel non nul n et renvoyant la liste des nombres premiers inférieurs ou égaux à n.

   Cette fonction doit impérativement utiliser la fonction `est_premier`.

4. Écrire une fonction `nieme_premier(n)` prenant en argument un entier naturel non nul n et renvoyant le n-ième nombre premier (au sens de l'ordre usuel des entiers).

   Par exemple `nieme_premier(1)` renvoie 2 et `nieme_premier(6)` renvoie 13.

   Cette fonction doit impérativement utiliser la fonction `est_premier` et une boucle conditionnelle. L'utilisation de `liste_premiers` est interdite.

Compléments

5. Pour tout entier naturel non nul m, on définit $f(m)$ comme étant le plus petit entier naturel n ayant exactement m diviseurs positifs.

   Écire une fonction `f(n)` prenant en argument un entier naturel m et renvoyant l'entier $f(m)$ défini auparavant.

   Par exemple, `f(1)` renvoie 1, `f(2)` renvoie 2 et `f(3)` renvoie 4.

   La fonction pourra utiliser la fonction `liste_div`.

6. Deux nombres premiers p et q, où $p < q$, sont dits *jumeaux* si $q = p + 2$. On conjecture qu'il y a une infinité de couples de nombres premiers jumeaux.

   Écrire une fonction `liste_premiers_jumeaux(n)` qui prend en argument un entier n et renvoie la liste des n premiers couples de nombres premiers jumeaux.

   On représentera les couples d'entiers par des listes à deux éléments.

   Par exemple, `liste_premier_jumeaux(5)` renvoie la liste `[[3, 5], [5, 7], [11, 13], [17, 19], [29, 31]]`.

   La fonction pourra utiliser la fonction `est_premier`.

7. Tout entier naturel n supérieur ou égal à 2 admet une décomposition première, c'est-à-dire une écriture de la forme

$$n = \prod_{i=1}^{r} p_i^{\alpha_i}$$

   où $r \in \mathbb{N}^*$, $p_1, \ldots, p_r$ sont des nombres premiers tels que $p_1 < p_2 < \cdots < p_r$, et $\alpha_1, \ldots, \alpha_r \in \mathbb{N}^*$.

   Écrire une fonction `facto(n)` prenant en argument un entier naturel n supérieur ou égal à 2 et renvoyant la liste de ses diviseurs premiers répétés avec leurs multiplicités.

   Par exemple `facto(12)` renvoie `[2,2,3]` puisque $12 = 2^2 \times 3$.

   La fonction n'utilisera aucune des fonctions des questions précédentes.