

1I001 – MIPI 14.2 TME Solitaire

le 28 Octobre 2014

Le seul document permis est la carte de référence.
--

Exercice 1 (4 points)

Question 1 : Ecrivez une fonction **somme** qui prend en entrée un entier naturel n et renvoie la valeur de

$$\sum_{0 \leq i < j \leq n} 2i + 3j.$$

Question 2 : Ecrivez un jeu de tests pour la fonction précédente.

Exercice 2 : Anagrammes (8 points) Le but de cet exercice est de faire appel à des notions d'itérations et de fonctions partielles sur des chaînes de caractères. Au sens mathématique, deux mots de même longueur a_1, a_2, \dots, a_n et b_1, b_2, \dots, b_n sont anagrammes s'il existe une permutation σ de l'ensemble des entiers $[1, n]$ tel que pour tout $i : b_i = a_{\sigma(i)}$. Cela signifie que deux mots sont anagrammes quand l'un peut être obtenu depuis l'autre par permutation des lettres. Par exemple, "parisien" est un anagramme de "aspirine" et "amal" est un anagramme de "lama".

Dans la suite, on considère qu'un mot est une chaîne de caractère qui ne contient pas d'espace.

Question 1 : Donnez la définition d'une fonction partielle **moins_lettre**(c, a) qui renvoie :

- la chaîne obtenue à partir de la chaîne c en supprimant la première occurrence de la lettre a dans c , si c contient au moins une fois a .
- None si c ne contient pas a .

Question 2 : Ecrivez la fonction **anagramme** qui étant donnés deux mots m_1 et m_2 renvoie True si les mots sont anagrammes et False sinon.

Par exemple :

```
>>> anagramme("cecile","frederic")
False
>>> anagramme("amal","lama")
True
>>> anagramme("alexandra","axel")
False
```

Exercice 3 : Le problème du logarithme discret (8 points) Le problème du logarithme discret est un problème cousin de la factorisation qui est, lui aussi, utilisé en cryptologie. Le problème est le suivant. Etant donné un entier naturel h et un entier naturel g il s'agit de retrouver l'entier x (en supposant ici qu'il existe) tel que $h = g^x$. L'entier x est appelé le logarithme discret de h en base g . Par exemple, le logarithme discret de 8 en base 2 est $x = 3$.

Question 1 : Ecrivez une fonction **puissance_naive** qui étant donnés deux entiers naturels g et n retourne la liste des n premières puissances successives de $g : g, g^2, g^3, \dots, g^n$.

Par exemple :

```
>>> puissance_naive(2,6)
[2,4,8,16,32,64]
```

Question 2 : Complétez votre fonction précédente pour que soit utilisé l'élément précédent g^{i-1} déjà calculé pour le calcul de l'élément suivant g^i .

Question 3 : Ecrivez une fonction **logarithme_discret** qui, étant donnés trois entiers naturels g, h et n retourne le logarithme discret de h en base g , s'il existe, False sinon. Vous ferez l'hypothèse que $h \leq g^n$.

Par exemple :

```
>>> logarithme_discret(2,32,6)
5
>>> logarithme_discret(3,10460353203,25)
21
```

Indication : utilisez la fonction `puissance_naive` !

Question 4 : Quel est le logarithme discret de 96889010407 en base 7 ?