

## TP n° 6 : examen pratique

Vous pouvez traiter les exercices dans l'ordre que vous voulez, ils ne sont pas ordonnés par ordre de difficulté.

En fin d'épreuve, vous devrez avoir créé un fichier de code Python qui s'appellera `votrenomdefamille.py` où votre nom de famille sera écrit en lettres minuscules, sans accents, sans cédille, sans espaces et dans le cas des deux homonymes, précédé par l'initiale du prénom.

Puis vous vous identifierez sur [prepabellevue.org](http://prepabellevue.org), vous afficherez l'article 428 intitulé « Epreuve pratique d'informatique » (<http://prepabellevue.org/index.php?article=428>) de la rubrique MPSI / Informatique et vous enverrez votre fichier en utilisant le formulaire en ligne.

En cas de problème d'envoi, signalez-le tout de suite! Et n'attendez pas la dernière minute pour le faire!

Les symboles  $n, k, p$  désignent toujours des entiers,  $L, M$  des listes d'entiers naturels.

### I Exercice 1

On rappelle les fonctions Python suivantes :

- `str(x)` transforme un objet  $x$  en une chaîne de caractères
- `int(x)` transforme un objet  $x$  en un entier si cela a un sens
- `list(x)` transforme un objet  $x$  parcourable en une liste si cela a un sens

Soit  $k$  un entier naturel non nul.

Soit  $n$  un entier naturel, on considère ses chiffres dans l'écriture en base 10 :  $c_{p-1}, \dots, c_0$ . On dit que  $n$  est un nombre d'Armstrong d'ordre  $k$  (ou plus simplement un  $k$ -Armstrong) quand  $n$  est égal à la somme des puissances  $k$ -èmes de ses chiffres :  $n = \sum_{i=0}^{p-1} c_i^k$ .

Par exemple, 153 est un 3-Armstrong, car  $153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27$ .

- Écrivez une fonction `somme(L,k)` qui calcule la somme de puissances  $k$ -èmes des éléments d'une liste  $L$ .
- Écrivez une fonction `chiffres(n)` qui calcule la liste des chiffres d'un entier  $n$ .
- Écrivez une fonction `arm(n,k)` qui calcule le booléen «  $n$  est un  $k$ -Armstrong ».
- Écrivez une fonction `arms(n,k)` qui calcule la liste des  $k$ -Armstrong qui sont inférieurs ou égaux à  $n$ .

### II Exercice 2

- Écrivez une fonction `ordonnee(L)` qui calcule le booléen « la liste  $L$  est rangée dans l'ordre croissant ».
- Étant données deux listes  $L, M$  qui sont rangées dans l'ordre croissant, on peut en fabriquer une troisième en fusionnant les deux listes.

Par exemple, avec  $L = [1, 3, 8, 9, 15]$  et  $M = [2, 4, 7, 12, 17]$ , on peut construire la liste  $[1, 2, 3, 4, 7, 8, 9, 12, 15, 17]$ .

Écrivez une fonction `fusion(L,M)` qui vérifie d'abord si les deux listes sont bien rangées (si ce n'est pas le cas, elle retourne `False`) puis qui retourne la liste fusionnée si elles sont bien rangées. Attention : **il est interdit d'utiliser une quelconque fonction de tri** (comme `sort`), puisque cet algorithme est utilisé justement pour trier des listes (voir cours de seconde année).

### III Exercice 3

- Écrivez une fonction `melange(L)` qui calcule à partir d'une liste  $L$  la liste contenant d'abord les éléments de rangs pairs suivis des éléments de rangs impairs en respectant l'ordre d'apparition dans la liste  $L$ .

Par exemple, `melange([5,1,2,8,7,3]) = [5,2,7,1,8,3]`

- On prend une liste  $L$  quelconque et on lui applique successivement la fonction `melange` : au bout d'un certain nombre d'itérations, on retrouve la liste  $L$ , ce nombre est appelé la période de  $L$ .

Par exemple, avec  $L = [1, 2, 3, 4, 5]$ , on a `melange(L) = [1, 3, 5, 2, 4]`,

puis `melange([1, 3, 5, 2, 4]) = [1, 5, 4, 3, 2]`,

puis `melange([1, 5, 4, 3, 2]) = [1, 4, 2, 5, 3]`,

puis `melange([1, 4, 2, 5, 3]) = [1, 2, 3, 4, 5]`

Il a fallu 4 applications successives de la fonction `melange` pour retrouver la liste  $L$ , qui est donc de période 4.

Écrivez une fonction `periode(L)` qui calcule la période d'une liste  $L$ .

- Parmi toutes les listes du type  $[1, 2, 3, \dots, k]$  (où  $k \geq 2$ ), on cherche celles dont le rapport  $\frac{\text{periode}}{\text{longueur}}$  soit maximal.

Écrivez une fonction `rapportmax(n)` qui calcule l'entier  $k \in \{2, \dots, n\}$  tel que le rapport précédent soit maximal.

Par exemple, on a `rapportmax(10) = 5`.

### IV Exercice 4

Cette question est directement inspirée du site [prepabellevue.org](http://prepabellevue.org).

Quand les colleurs enregistrent les notes de colles, ils voient la liste des étudiants en trois colonnes, rangés par ordre alphabétique, comme dans les exemples ci-dessous.

On représente les étudiants par un numéro de 0 à  $n - 1$  (où  $n$  est l'effectif de la classe). L'objectif est d'arriver aux types d'affichages suivants :

si  $n = 9$ , alors les colleurs voient le tableau

0	3	6
1	4	7
2	5	8

si  $n = 10$ , alors les colleurs voient le tableau

0	4	7
1	5	8
2	6	9
3		

si  $n = 11$ , alors les colleurs voient le tableau

0	4	8
1	5	9
2	6	10
3	7	

Mais la difficulté est qu'un navigateur internet a besoin qu'on les lui donne en lignes, comme ci-dessous.

Dans le premier exemple, on envoie au navigateur internet la liste de listes

`[ [0,3,6], [1,4,7], [2,5,8] ]`.

Dans le deuxième exemple, on envoie au navigateur internet la liste de listes

`[ [0,4,7], [1,5,8], [2,6,9], [3] ]`.

Dans le troisième exemple, on envoie au navigateur internet la liste de listes

`[ [0,4,8], [1,5,9], [2,6,10], [3,7] ]`.

Écrivez une fonction `enligne(n)` qui calcule en fonction de l'entier  $n$  la liste de listes nécessaire pour l'affichage correct des  $n$  étudiants d'une classe (les trois exemples précédents sont donc `enligne(9)`, `enligne(10)` et `enligne(11)`).

Généralisez : écrivez une fonction `Enligne(n,p)` qui répartit de même les étudiants mais sur  $p$  colonnes (en particulier l'appel `Enligne(n,3)` doit donner le même résultat que `enligne(n)`).