Interrogation (à blanc) de *logique et approche mathématique de la programmation* – 15 novembre 2019

La clarté du code, l'efficacité de vos programmes et les spécifications de fonctions interviendront dans la cotation finale. Consignes :

- Produire un code lisible et efficace
- Commenter votre code de façon intelligible
- Les questions doivent être clairement séparées
- Utiliser des noms de variables explicites
- Vos fonctions doivent être spécifiées
- Tester vos programmes (y compris sur les cas pathologiques)
- Sauver régulièrement ; vous devez fournir un fichier `.py` ou `.ipynb` (sous Jupyter) contenant en commentaire vos nom et prénom

1. Ecrire une fonction qui prend comme argument une liste d'entiers $(a_1, \ldots, a_n)$ et renvoie une chaîne de caractères de la forme

   $$\text{“}a_1 \texttt{ -> } a_2 \texttt{ -> } \cdots \texttt{ -> } a_n \texttt{ -> } a_1\text{ ”.}$$

   Si la liste est vide, le programme renvoie la chaîne “liste vide”. Par exemple, la donnée [1,2,3,14] fournit la chaîne de caractères `1->2->3->14->1`.

2. Une permutation μ de $\{0, \ldots, n-1\}$ est codée par une liste ℓ de longueur n telle que $\ell[i] = \mu(i)$. Par exemple, la permutation

   $$\mu = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 3 & 2 & 6 & 5 & 0 \end{pmatrix}$$

   est codée par la liste [1,4,3,2,6,5,0]. Un cycle de longueur k est codé par une liste de longueur k codant les images successives d'un élément. Ainsi, le cycle

   $$\begin{pmatrix} 0 & 1 & 4 & 6 \\ 1 & 4 & 6 & 0 \end{pmatrix} = (0\ 1\ 4\ 6)$$

   est codé par la liste [0,1,4,6].

   2.a) Ecrire une fonction qui prend comme paramètres une liste codant une permutation μ de $\{0, \ldots, n-1\}$, deux naturels i, t (avec $0 \leq i < n$) et qui renvoie $\mu^t(i)$. Ainsi, on a

   ```
   puissance_permutation([1,4,3,2,6,5,0],0,3)
   > 6
   ```

   car $\mu^3(0) = \mu^2(1) = \mu(4) = 6$.

   2.b) Ecrire une fonction qui prend comme paramètres une liste codant une permutation μ de $\{0, \ldots, n-1\}$ et un naturel $i < n$ et qui renvoie le cycle contenant i dans la décomposition de μ en produit de cycles. Ainsi,

```
cycle([1,4,3,2,6,5,0],0)
> [0, 1, 4, 6]
cycle([1,4,3,2,6,5,0],1)
> [1, 4, 6, 0]
cycle([1,4,3,2,6,5,0],5)
> [5]
```

2.c) Question **bonus**. Décomposer une permutation en produits de cycles : Ecrire une fonction qui prend comme paramètres une liste codant une permutation μ et renvoie la liste des cycles qui la composent (donc, une liste de listes). Ainsi,

```
decompose([1,4,3,2,6,5,0])
> [[0, 1, 4, 6], [2, 3], [5]]
```

3. 3.a) Ecrire une fonction qui prend en entrée une chaîne de caractères. Elle renvoie `True` si et seulement si la chaîne est formée uniquement de `a` et de `b` et que le double du nombre de `b` moins 1 vaut le nombre de `a`. Utiliser cette fonction pour lister (afficher) tous les mots de longueur 11 ayant cette propriété. Par exemple, `aaaaaaabbbb` renvoie `True`. (Question **bonus** : combien de mots la liste ainsi produite contient-elle ? Sauvegarder cette liste dans un fichier).

   3.b) Même question mais avec la condition supplémentaire que pour renvoyer `True`, on ne peut jamais trouver strictement plus de 3 `a` consécutifs au sein d'un mot (on peut en avoir 3 mais pas 4). Ainsi, `aaaaaaabbbb` donne `False` mais, `aaabbaabaab` est `True`. Pour vérifier votre résultat, il y a 155 mots de longueur 11 répondant à la question.

4. Ecrire une fonction `f(n,q)` qui renvoie une liste contenant toutes les partitions du naturel n en q naturels non nuls, i.e., toutes les listes de q éléments non nuls dont la somme donne n. On supposera que l'ordre a de l'importance, i.e., la liste [1,2] et la liste [2,1] sont deux sorties distinctes pour `f(3,2)`. Par exemple `f(5,3)` donne

```
[[1,2,2], [2,1,2], [2,2,1], [1,1,3], [1,3,1], [3,1,1]]
```

Hors interrogation. Lors de l'élaboration de celle-ci, nous avons aussi pensé à la question suivante.

5. On considère les suites $(u_n)_{n\geq 0}$ et $(w_n)_{n\geq 0}$ définies par

$$u_0 = 0, u_1 = 1, w_0 = 2, w_1 = 3$$

et pour tout $n \geq 0$,

$$\begin{cases} u_{n+2} = u_{n+1} + w_n \\ w_{n+2} = w_{n+1} - u_n \end{cases}$$

Ecrire deux fonctions permettant de calculer u_n et v_n. Vérifier que

$u_{100} = 11157408183601449$ et $w_{100} = 13016848061392159$.