Examen de *logique et approche mathématique de la programmation*
26 janvier 2022

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
- 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` contenant en commentaire vos nom et prénom
- Fin de l'examen : 12h30

Répondre à 3 questions, **au choix**, parmi les 4 questions suivantes. Chaque question est cotée sur le même nombre de points. Préciser la question non retenue. Vos solutions doivent être envoyées à `Carole.Baum@uliege.be` et `M.Rigo@uliege.be`.

1. Soit $m \geq 1$. Une liste `x` d'entiers de longueur $m+1$ est dite *symétrique*, s'il existe un entier `k` $\in \mathbb{Z}$ tel que pour tout indice valide `i`,

   `x[i]-k = k-x[m-i]`

   L'entier `k` est appelé *hauteur* de la suite symétrique. Par exemple, les listes [6,3,-1,5,1,-2] et [6,3,-1,2,5,1,-2] sont symétriques (toutes deux, avec une hauteur $k = 2$). On supposera que la liste vide et les listes de longueur 1 sont symétriques. On supposera de plus que la liste vide est de hauteur nulle.

   a) Ecrire une fonction qui teste si une liste fournie en entrée est symétrique et renvoie un booléen (`True` ou `False`).

   b) Ecrire une fonction à qui on fournit une liste en entrée et renvoie sa hauteur lorsqu'elle est symétrique et un message d'erreur sinon. Par exemple, si on fournit la liste $[6, 3, -1, 5, 1, -2]$, la fonction retourne l'entier 2.

   c) Ecrire une fonction `gen` qui étant donné une liste de longueur m et un entier k, produit une liste de longueur $2m$ symétrique et de hauteur k. Ainsi, si on fournit $[6, 3, -1]$ et 2, la fonction retourne $[6, 3, -1, 5, 1, -2]$. Que produit `gen(gen(gen([1],2),3),4)` ?

   d) Ecrire une fonction qui étant donné une liste de longueur m et un entier k, produit une liste de longueur $2m+1$ symétrique et de hauteur k.

2. Soient un nombre réel x compris strictement entre 0 et 1, un nombre réel $\beta > 1$ et un naturel $k > 0$.

   a) Ecrire une fonction `dev(b,x,k)` qui renvoie une liste contenant le développement

   $$x = \sum_{i=1}^{k} \frac{d_i}{\beta^i} + R$$

   où les d_i sont des entiers compris entre 0 et $\lceil \beta \rceil - 1$ et vérifiant, pour tout $j \leq k$,

   $$x - \sum_{i=1}^{j} \frac{d_i}{\beta^i} < \frac{1}{\beta^j}.$$

En particulier, $0 \leq R < \frac{1}{\beta^k}$. Ainsi avec $x = \pi - 3$, $\beta = 10$ (donc $\lceil \beta \rceil - 1 = 10 - 1 = 9$) et $k = 6$, la fonction renvoie $[1, 4, 1, 5, 9, 2]$ qui sont les 6 premières décimales de π.

b) Avec $\beta = \frac{1+\sqrt{5}}{2}$ et uniquement à l'aide de votre programme, que pouvez-vous conjecturer sur le développement de $1/2$?

3. On considère des matrices `k`$\times$`k` encodées par une liste de listes et `k`≥ 1 est une variable globale du programme que l'on pourra modifier.
   a) Ecrire une fonction qui étant donné un naturel non nul `n`, renvoie une matrice aléatoire formée d'entiers compris entre 0 et `n`.
   b) Ecrire une fonction qui affiche de façon lisible une matrice.
   c) Ecrire une fonction qui pour une matrice et deux indices valides `i` et `j`, renvoie une matrice où les lignes `i` et `j` ont été permutées. Faire de même pour permuter deux colonnes.
   d) Ecrire une fonction qui pour une matrice M et deux indices valides `i` et `j`, renvoie une matrice N où la ligne `i` et la colonne `j` ont été permutées, i.e. $N_{i,n} = M_{n,j}$ et $N_{n,j} = M_{i,n}$ pour tout n. Pour lever toute ambiguïté, on impose $N_{i,j} = M_{i,i}$. Ainsi,
$$M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix},\ i = 1, j = 2 \text{ donnent } N = \begin{pmatrix} 2 & 1 & 8 \\ 4 & 2 & 6 \\ 7 & 3 & 9 \end{pmatrix}$$
   e) Ecrire une fonction qui renvoie une liste de longueur `k` contenant les sommes des éléments de chaque ligne. Faire de même pour les colonnes.

   Pour répondre à cette question, on ne peut **pas** utiliser de modules spécialisés dans la gestion de matrices comme `numpy`.

4. Le *problème de Josephus*: Soient $n, k \geq 1$ avec $k < n$. Des personnes numérotées de 1 à n sont placées en cercle et attendent d'être exécutées. Le comptage commence en 1 et se poursuit en parcourant le cercle toujours dans la même direction. Après avoir épargné k personnes vivantes, le suivant est exécuté. La procédure est répétée avec les personnes survivantes en partant de la position de la dernière victime et jusqu'à ce qu'il ne reste qu'un unique survivant.

   Par exemple, avec $n = 5$ et $k = 1$, on va dans l'ordre, exécuter $2, 4, 1$ puis 5 et l'unique survivant est 3. En effet, on commence par épargner 1, on supprime 2, on épargne 3, on supprime 4, on épargne 5. Dans le cercle, on revient en 1 que l'on supprime. A ce stade, sont encore vivants 3 et 5. Puisque l'on vient de supprimer 1, on épargne 3, puis on supprime enfin 5.
   a) Ecrire une fonction `survivant(n,k)` qui pour un entier n et un paramètre k, renvoie le numéro du survivant. Ainsi, avec l'exemple précédent, l'entrée $n = 5$ et $k = 1$ renvoie 3. Avec le paramètre $k = 1$, les premières valeurs pour $n = 2, 3, \ldots$ sont $1, 3, 1, 3, 5, 7, 1, 3, 5, \ldots$.
   b) Ecrire dans un fichier les valeurs obtenues pour $k = 1, 2, 3$ et $k < n < 100$.
   c) De la même façon, écrire une fonction `executes(n,k)` qui renvoie la liste ordonnée des exécutés. Toujours avec le même exemple ($n = 5$, $k = 1$), cette fonction renvoie $[2, 4, 1, 5]$.