Informatique tronc commun TP 09

25 janvier 2018

1. **Lisez attentivement tout l'énoncé avant de commencer.**
2. Commencez la séance en créant un dossier au nom du TP dans le répertoire dédié à l'informatique de votre compte.
3. Après la séance, vous devez rédiger un compte-rendu de TP et l'envoyer au format électronique à votre enseignant.
4. Vous rendrez un compte-rendu sous forme d'un fichier d'extension `.py`, en respectant exactement les spécifications données plus bas.
5. Ce TP est à faire en binôme, vous ne rendrez donc qu'un compte-rendu pour deux.
6. Ayez toujours un crayon et un papier sous la main. Quand vous réfléchissez à une question, utilisez les !
7. Vous devez être autonome. Ainsi, avant de poser une question à l'enseignant, merci de commencer par :
   - relire l'énoncé du TP (beaucoup de réponses se trouvent dedans) ;
   - relire les passages du cours [1] relatifs à votre problème ;
   - effectuer une recherche dans l'aide disponible sur votre ordinateur (ou sur internet) concernant votre question.

   Il est alors raisonnable d'appeler votre enseignant pour lui demander des explications ou une confirmation !

Cette séance sera consacrée à un TP de révision ainsi qu'à l'étude des complexités des fonctions écrites.

Instructions de rendu

Attention : suivez précisément ces instructions. Vous enverrez à votre enseignant un fichier d'extension `.py` (script `Python`) nommé

`tp09_berne_durif.py`,

où les noms de vos enseignants sont à remplacer par ceux des membres du binôme. Le nom de ce fichier ne devra comporter ni espace, ni accent, ni apostrophe, ni majuscule. Dans ce fichier, vous respecterez les consignes suivantes.

1. Dans le cas fort improbable où vous ne vous en souviendriez pas.

— Écrivez d'abord en commentaires (ligne débutant par #), le titre du TP, les noms et prénoms des étudiants du groupe.

— Commencez chaque question par son numéro écrit en commentaires.

— Les questions demandant une réponse écrite seront rédigées en commentaires.

— Les questions demandant une réponse sous forme de fonction ou de script respecteront pointilleusement les noms de variables et de fonctions demandés.

Calculs de complexité.

Pour chaque fonction écrite dans les questions 3, 4, 5 et 6, on précisera sa complexité, en fonction de la taille des données passées en argument. On justifiera aussi brièvement tous cela.

La complexité des fonctions sera prise en compte dans l'évaluation : le fait d'avoir une fonction dont la complexité est clairement sous-optimale sera sanctionné.

1 Calcul des termes successifs d'une suite récurrente.

On considère la suite u définie par $u_0 \in [-2;2]$ et $\forall n \in \mathbb{N}, u_{n+1} = \sqrt{2-u_n}$. On rappelle que u converge vers 1.

Q1 Écrire une fonction `valeur_u(n,u0)` qui, à un entier naturel `n` et un flottant `u0`, renvoie u_n.

Q2 Écrire une fonction `approche_u(eps,u0)` qui, à deux flottants `eps` et `u0`, renvoie le plus petit rang $n \in \mathbb{N}$ tel que $|u_n - 1| \leq$ `eps`.

2 Records d'un tableau.

Dans un tableau $t = [t_0, \ldots, t_{n-1}]$, on dit que l'on a un record à une position $0 \leq k < n$ si $\forall i < k,\ t_i < t_k$. Par définition, on a toujours un record en position 0.

Q3 Écrire une fonction `record(t,k)` qui, à un tableau de nombres `t` et un entier `k`, renvoie le booléen `True` si `t` admet un record en position `k`, `False` sinon.

Q4 Écrire une fonction `nb_records(t)` qui, à un tableau de nombres `t`, renvoie le nombre de records dans `t`.

3 Palindromes.

Une chaîne de caractères $s_0 s_1 \ldots s_{n-1}$ est un *palindrome* si elle est « symétrique » : $\forall k \in \{0, \ldots, n-1\}, s_k = s_{n-1-k}$.

Q5 Écrire une fonction `est_pal(s)` qui, à une chaîne de caractères `s`, renvoie le booléen `True` si `s` est un palindrome et `False` sinon.

Q6 Écrire une fonction `max_pal(s)` qui, à une chaîne de caractères `s`, renvoie le couple `(p,k)` où :

— `p` est la plus grande sous chaîne palindromique centrée (c'est la plus grande sous-chaîne palindromique de la forme $s_k s_{k+1} \ldots s_{n-1-k}$) ;

— `k` est l'indice du début de cette sous-chaîne dans `s` (dans le cas ou `p` est vide, on renverra n'importe quel nombre).

4 Résumé d'un fichier.

Q7 Écrire une fonction `resume(nom_de_fichier)` qui prend en argument une chaîne de caractères `nom_de_fichier` et qui renvoie le triplet `(L,m,c)` où, pour le fichier dont le chemin est `nom_de_fichier` ; :

— `L` est le nombre de lignes du fichier ;

— `m` est le nombre de mots (sous chaîne maximale non vide de caractères consécutifs, sans blanc ni tabulation) du fichier ;

— `c` est le nombre de caractères du fichier (retours chariot `'\n'`, tabulations `'\t'` et espaces compris).

5 Crible d'Ératosthène.

On rappelle que, si un entier naturel $p \geq 2$ n'a aucun diviseur inférieur à $\sqrt{p}$ (sauf 1), alors p est premier.

Q8 Écrire une fonction `crible(p)` qui, à un entier naturel `p`, renvoie le tableau des nombres premiers inférieurs ou égaux à $\texttt{p}^2$, triés par ordre croissant.

6 Une petite énigme.

Q9 Trouver la somme de tous les entiers qui sont la somme des factorielles de leurs chiffres, en écriture décimale. On écrira une fonction `enigme()` donnant le résultat.