Informatique en CPGE (2014-2015)
Devoir maison 2

Ce devoir sera fait en groupe (un devoir par groupe de colles).

Chacun des noms des fichiers programmes doit commencer par le nom d'un élève (les noms des autres élèves ayant participé seront écrits en commentaire au début de chaque programme). La présentation est importante ; pour écrire un commentaire sur une ligne, on commence par le caractère #, pour des commentaires sur plusieurs lignes, on les entoure par des triples guillemets (""""commentaires ..."""").

Les programmes seront envoyés en une seule fois par email.

Exercice 1

Il s'agit d'écrire dans un premier fichier, nommé "nom_dm2ex1.py", un programme qui calcule les termes de la suite (u_n) définie par $u_0 = 2$, $u_1 = -4$, et $u_{n+1} = 111 - \frac{1130}{u_n} + \frac{3000}{u_n \times u_{n-1}}$.

1. Ecrire un programme qui utilise le type **float** et calcule les 50 premiers termes. Attention : pour calculer un terme le programme doit connaitre les deux termes précédents.
2. Ecrire un deuxième programme en recopiant le précédent et en le modifiant afin d'utiliser des fractions.

   Rappel : les commandes suivantes permettent de définir par exemple la fraction $\frac{5}{3}$:

   ```
   from fractions import Fraction as f
   a=f(5,3)
   ```

   Convertir les valeurs en flottant avant de les afficher et comparer les résultats obtenus par les deux programmes.

Exercice 2

Pour représenter un caractère dans la machine, on attribue un nombre à chaque caractère par exemple à l'aide du code ASCII binaire (American Standard Code for Information Interchange). Ce code utilise un octet par caractère, avec le premier bit toujours à 0, et permet donc de représenter $2^7 = 128$ caractères. Ce sont les caractères que l'on trouve sur les touches d'un clavier d'ordinateur : par exemple, on attribue les nombres de "65" à "90" aux 26 lettres majuscules de l'alphabet A, B, C, ..., Z ; les nombres de "97" à "122" aux 26 lettres minuscules a, b, c, ..., z ; les nombres de "48" à "57" aux neuf chiffres 0, 1, 2, ..., 9 ; l'espace est codé par le nombre "32", le point par le nombre "46", etc. (voir une table complète sur le site www.table-ascii.com)

L'instruction `chr(65)` permet d'afficher 'A' et l'instruction `ord('A')` permet d'afficher 65.

1. Codage d'un message : dans un fichier, nommé "nom_dm2ex2.py", écrire un programme qui permet de convertir un texte saisi en entrée en un suite de nombres. Le texte sera saisi par l'utilisateur uniquement en majuscules, avec des espaces et de la ponctuation.

   L'entrée est donc de type **str** ; il faut récupérer chaque caractère et le convertir en un nombre à deux chiffres avec la fonction **ord**. Puis constituer le "message codé" en accolant les nombres représentant chaque caractère, y compris les espaces ou la ponctuation. Par exemple "HELLO !" sera codé par "72697676793233". Le message codé est donc de type **str** et il est affiché en sortie.
2. Décodage d'un message : écrire un deuxième programme qui prend en entrée une suite de nombres codant un texte et affiche en sortie le texte décodé.
3. Complément : dans un troisième programme, on complète le premier programme en ajoutant une transformation des nombres représentant les caractères. On utilise la transformation $x \to 100 - x$. Le message "HELLO !" est alors codé par "28312424216867". Attention, à chaque caractère doit correspondre un unique nombre à deux chiffres.

   Ecrire alors un quatrième programme modifiant en conséquence le programme décodant les messages et décoder le message suivant :

   34183514215668142115176835143110681618211514316867