Feuille de TP n° 3 - Effet de bord, dictionnaires

Exercice 1. La fonction suivante est-elle à effet de bord ?

```
def Reduction(L):
    n=len(L)
    while (n>2):
        L.pop()
        n=len(L)
    return L
```

Ecrire une fonction `Reduction2` qui produit le même résultat, mais sans effet de bord.

Exercice 2. La fonction suivante est-elle à effet de bord ?

```
def f(L):
    for k in range(len(L)):
        L.append(L[k])
    return L
```

Ecrire une fonction `f2` qui produit le même résultat, mais sans effet de bord.

Exercice 3.

1. Ecrire une fonction `pgcd` qui prend en entrée deux entiers a, b et qui renvoie le pgcd de a et de b.
   On utilisera une boucle `while`.
2. Déterminer une fonction f de terminaison pour `pgcd`.
3. A l'aide du chapitre 6, déterminer un invariant de boucle H_r pour `pgcd` (où r représente le nombre de passages dans la boucle durant l'exécution de la fonction).
4. Ecrire une fonction `pgcC` qui prend en entrée $a, b \in \mathbb{N}$ et renvoie $pgcd(a, b)$ ainsi que le nombre de divisions euclidiennes effectuées.
5. Ecrire une fonction **CompteurDiv** qui prend en entrée un entier n et qui renvoie le plus grand nombre de divisions euclidiennes à effectuer pour calculer $pgcd(a, n)$, pour toutes les valeurs de $a \in \{1, n-1\}$.
   On utilisera du pseudo-code pour mettre en place les différents éléments de la fonction.

Exercice 4 (Dictionnaires)**.**

1. Utiliser les instructions suivantes, et indiquer leur effet.

```
dico_en_fr={}
dico_en_fr["yes"]="oui"
dico_en_fr["no"]="non"
dico_en_fr["why"]="pourquoi"
"yes" in dico_en_fr
"pourquoi" in dico_en_fr
del(dico_en_fr["no"])
dico_en_fr
dico_en_fr.clear()
dico_en_fr
```

2. Définir le dictionnaire `dict_es_fr` de plusieurs façons :

```
# définition directe
dico_es_fr={"si":"oui","hoy":"aujourd'hui","porque":"pourquoi"}
# en utilisant la fonction dict
 dico_en_nb=dict(one=1,two=2,three=3)
# en transformant une liste de tuple à 2 éléments
 liste_es_nb=[("uno",1),("dos",2),("tres",3)]
 dico_es_nb=dict(liste_es_nb)
```

3. Essayer les commandes pour parcourir le dictionnaire précédent. Indiquer leurs effets

```
len(dico_es_fr)
for i in dico_es_fr.keys(): print(i)
dico_es_fr.values()
dico_es_fr.items()
```

Exercice 5 (Comptage avec un dictionnaire)**.**

1. Compléter le code suivant, qui crée un dictionnaire comptant le nombre d'apparitions de chaque lettre dans un mot :

```
mot = "abracadabra"
dico = dict()
for lettre in mot:
    if lettre in dico:
        dico[lettre] = ...
    else:
        dico[lettre] = ...
print(dico)
```

2. Écrire une fonction `occurrences(a,s)` qui renvoie le nombre d'occurrences d'un caractère `a` dans une chaîne de caractères `s`.
3. Ecrire une fonction `OccurrAlpha` qui prend en entrée une chaîne de caractères `s` t qui renvoie un dictionnaire qui associe à chaque lettre de l'alphabet le nombre de fois qu'elle apparaît dans `s`.
   On utilisera la fonction `c.isalpha()` qui teste si le caractère `c` est une lettre, et la fonction `lower` qui convertit une lettre majuscule en sa version minuscule.

4. Tester la fonction `OccurrAlpha` sur le fichier `Quadrature.txt`. On utilisera pour cela les commandes

```
montxt=open('chemin\\fichier.txt','r') # Ouvrir un fichier avec Python, en lecture (read)
s=montxt.read() #Créée une chaîne de caractères qui correspond au contenu du fichier texte
montxt.close() #Fermer le fichier dans Python
```

5. Écrire une fonction `pourcentages` qui prend en argument un dictionnaire associant à chaque clé un nombre d'occurrences et renvoie un nouveau dictionnaire associant à chaque clé le pourcentage que représente les occurrences de cette clé sur le total des occurrences du dictionnaire.
   On utilisera la fonction `round(x,n)`, qui arrondit un nombre flottant `x` à la nème décimale.

Exercice 6. Une *base de données* est un ensemble de dictionnaires qui ont tous les mêmes clés. On peut interpréter cela comme un dictionnaire dont les éléments sont des listes, avec une structure commune pour chaque liste. (Ex : Les élèves de PTSI et la liste de leurs notes aux DS de mathématiques/physique/SI).
Tout comme un long texte est stocké hors de Python dans un fichier texte, une grande base de données est stockée hors de Python dans un tableur (habituellement au format csv).

1. Avec la librairie `csv`, on peut ouvrir le fichier `countries.csv` et le charger dans Python comme une base de données (un tableur).

```
import csv
fichier=open('chemin\\countries.csv','r')
lecture = csv.reader(fichier)
```

2. A partir de cette base de données, on peut créer de petits dictionnares. Que représente le dictionnaire suivant ?

```
capitale = {}
for ligne in lecture:
    capitale[ligne[1]]=ligne[2]
```

3. Créer, de la même façon, un dictionnaire `continent` associant les continents aux pays et un dictionnaire `monnaie` associant les monnaies aux pays.
4. Afficher les noms des pays d'Océanie, puis tous ceux dont la monnaie est le franc pacifique.
   Pour afficher une information, on utilise la commande `print`.
5. Créer un dictionnaire `capitale2` qui associe à chaque capitale son pays.
6. Peut-on créer un dictionnaire `monnaie2` qui associe à chaque monnaie son pays ?
7. A partir de la base de données, créer un dictionnaire `OcurrMonn` qui associe à chaque monnaie le nombre de pays qui possèdent cette monnaie.
   On utilisera à nouveau la commande `for ligne in lecture:`.
8. Déterminer la monnaie qui est présente dans le plus grand nombre de pays.