TP 4 informatique

BCPST 1 2017-2018

Fonctions

Points abordés

— Fonctions.

---

1 Introduction

Jusqu'à présent, les programmes utilisés ne sont pas exploitables : ils affichent un résultat mais on ne peut pas les utiliser directement. De plus, on demande toujours à l'utilisateur de saisir les données, ce qui empêche une automatisation des tâches. L'avantage des fonctions est de plusieurs ordres :

— d'une part, si la fonction renvoie une donnée, il est possible de la récupérer pour effectuer un nouveau traitement ;

— d'autre part, il est possible d'appliquer une fonction à un nombre conséquent de données ;

— de plus, une tierce personne (qui peut être soi-même trois mois plus tard...) peut utiliser une fonction déjà implantée, pour peu qu'on lui donne le code source de la fonction, le point étant qu'il n'y a **pas besoin de savoir comment** la fonction a été implantée, mais seulement de savoir ce qu'elle prend en entrée, ce qu'elle est censée faire et ce qu'elle renvoie.

Exemple 1. La fonction ci-dessous

```
def SommePuissance(n,p) :
    S=0
    for i in range(1,n+1) :
        S=S+i**p
    return S
```

prend en argument deux entiers : n et p, et renvoie la somme des n premières puissances p^{e} entières.

Expliquons les différents mots-clés de cette fonction.

1. le mot `def` permet de définir une fonction. Ici, on a donné comme nom à la fonction `SommePuissance`.
2. Le couple `(n,p)` correspond aux arguments de la fonction.
3. Le mot `return` qui correspond à la valeur de retour de la fonction.

Il est possible d'utiliser la fonction directement dans l'interpréteur de la manière suivante :

Exemple 2. `SommePuissance(10,2)`

2 Exercices

Exercice 1. Finir les exercices des autres feuilles de TP non abordés à l'aide de fonctions.

Exercice 2. Écrire une fonction `Occurrence(chaine,lettre)` qui prend en argument une chaîne de caractères `chaine` et une lettre `lettre` et qui retourne le nombre de fois où `lettre` apparaît dans `chaine`.

Exercice 3. Écrire une fonction `maxLettre(chaine)` qui prend en argument une `chaine` et qui retourne un couple $(lettre, nombre)$, où *lettre* est la lettre la plus fréquente de chaîne et *nombre* le nombre d'occurrence de *nombre*.

Exercice 4. Écrire une fonction `SontAnagrammes(chaine1,chaine2)` qui renvoie `True` si `chaine1` et `chaine2` sont anagrammes et `False` sinon.

Exercice 5. Écrire une fonction qui renvoie le nombre de lettres alphanumériques d'une chaîne de caractères.

Exercice 6. Écrire une fonction qui prend en argument une chaîne de caractères et renvoie le retourné de cette chaîne. En déduire une fonction qui prend en argument une chaîne et qui retourne `True` si la chaîne est un palindrome et `False` sinon.

Exercice 7. $\star$

Un habitué du jardin du Luxembourg nourrit régulièrement une horde de pigeons. On sait que les pigeons arrivent toujours de la manière suivante :

— la première minute, un pigeon se présente,
— la deuxième minute, deux pigeons le rejoignent,
— la troisième minute, trois pigeons intègrent le groupe,
— et ainsi de suite.

Sachant qu'un pigeon picore une portion de miettes de pain en une minute, qu'il n'est jamais rassasié, et que les premiers pigeons sont les premiers servis, écrire une fonction `nourrirPigeon(NbrePortions)` qui prend en argument un entier qui correspond aux nombres de portions de miettes de pain, et qui renvoie le nombre de pigeons ayant été nourri au moins une fois.

Exercice 8. $\star$

Écrire une fonction `Chifoumi(n)` qui prend en argument un entier n qui correspond au nombre de parties, et qui affiche si vous avez gagné, perdu, ou s'il y a match nul.

Exercice 9. $\star$

César codait les messages de la manière suivante : étant donné un mot, il décalait chaque lettre de 3. Ainsi chaque lettre était substituée par une autre de la manière suivante :

$$\begin{array}{cccccccccccccc} a & b & c & d & e & f & g & h & i & j & \cdots & x & y & z \\ d & e & f & g & h & i & j & k & l & m & \cdots & a & b & c \end{array}$$

pour le mot `bizarre`, on obtient le mot `elccuuh`.

1. Écrire une fonction `Cesar(mot,decalage)` qui prend en argument une chaîne de caractères `mot` et une lettre `decalage` et qui retourne un mot où on décale chaque lettre de mots par `decalage`, où a correspond à un décalage de 1, b à un décalage de 2 etc.
2. Vigenère, pour plus de sécurité, utilisait un mot pour effectuer son décalage. Par exemple, si son mot permettant le décalage est "boum" et que le message à coder est "c'est la fete au village", on obtient comme message codé : `"e'tng np arvt vh xxgycvz"`. Écrire une fonction `Vigenere(mot,cle)` qui prend en argument une chaîne de caractères `mot` et qui retourne le message codé grâce à `cle`.