

Python : Récursivité

Kais Klai

kais.klai@lipn.univ-paris13.fr

Dans ce qui suit, et pour chaque exercice, vous devez d'abord écrire l'algorithme avant de le traduire en Python.

1. **A moins que ça soit demandé explicitement, ne jamais utiliser les instructions `print()` et `input()` dans une fonction.**
2. **Prenez le temps de vérifier que le prototype de chaque fonction est bien correct avant de définir la fonction.**
3. **Dans ce qui suit, et pour chaque question, créez deux fichiers : le premier contient la définition de la fonction demandée alors que le deuxième "importe" le premier fichier, et contient quelques instructions permettant de tester la fonction.**
4. **Avant d'écrire une fonction récursive, il faut réaliser les deux étapes suivantes :**
 - (a) Formulez le problème de manière récursive,
 - (b) Trouvez la condition d'arrêt (quand arrêter de faire des appels récursifs).Une fois ces deux tâches réalisées, la définition de la fonction récursive devient triviale.

Fonctions d'affichage

1. Ecrire une fonction récursive qui affiche les **n** premiers nombres dans l'ordre croissant.
2. Ecrire une fonction récursive qui affiche les **n** premiers nombres dans l'ordre décroissant.
3. Ecrire une fonction récursive qui affiche un triangle avec le caractère `*` sur un nombre de lignes donné. Par exemple, le triangle ci-dessous est affiché sur 5 lignes:

```
*
**
***
****
*****
```
4. Maintenant, sous cette forme ...

```
*****
****
***
**
*
```

5. Maintenant, sous cette forme ...

```

      *
    * - *
  * - * - *
* - * - * - *
* - * - * - * - *

```

6. Et maintenant, sous cette forme

```

* - * - * - * - *
* - * - * - *
* - * - *
* - *
*

```

Chaines de caractères

1. Ecrire une fonction récursive qui inverse une chaîne de caractères.
 Entrée : `reverseString`
 Sortie :
 La chaîne inversée est : `gnirtSesrever`
2. Ecrire une fonction récursive qui copie une chaîne de caractères dans une autre.
 Entrée : `StringToBeCopied`
 Sortie :
 Chaîne initiale : `StringToBeCopied`
 Chaîne copiée : `StringToBeCopied`
3. Ecrire une fonction récursive qui compte le nombre de voyelles dans une chaîne de caractères.
4. Ecrire une fonction récursive qui vérifie si deux chaînes de caractères possèdent le même nombre de voyelles.
5. Ecrire une fonction récursive qui vérifie si une chaîne de caractères est un *palindrome*.

Nombres

1. Ecrire une fonction récursive qui calcule x^n .
2. Ecrire une fonction récursive qui calcule le *plus grand commun diviseur* (pgcd) de deux nombres par soustractions successives.

$$gcd(a, b) = \begin{cases} gcd(a - b, b) & \text{if } a > b \\ gcd(a, b - a) & \text{if } b > a \\ a & \text{if } a = b \end{cases}$$

Note: Nous supposons que a et b sont deux nombres positifs.

3. Ecrire une fonction récursive qui convertit un nombre décimal en binaire.

Entrée : 67

Sortie :

La représentation binaire de 67 est : 1000011

4. Ecrire une fonction récursive qui calcule les valeur de la suite de *Fibonacci* jusqu'à un rang donné.

Entrée : 10

Sortie :

Les valeur de la série sont :

1 1 2 3 5 8 13 21 34 55

5. Ecrire une fonction récursive qui calcule le nombre de chiffres composant un nombre entier positif.
6. Ecrire une fonction récursive qui calcule la somme des chiffres composant un nombre entier positif.

Entrée : 1453

Sortie : 13

Listes

1. Ecrire une fonction récursive qui retourne la position de la première occurrence d'une valeur dans une liste (ou -1) si la valeur n'existe pas.
2. Ecrire une fonction récursive qui calcule le nombre d'occurrence d'une valeur dans une liste.
3. Ecrire une fonction récursive qui calcule le minimum d'une liste.
4. Ecrire une fonction récursive qui cherche une valeur dans une liste ordonnée (dans l'ordre croissant) en utilisant la méthode de la *recherche dichotomique*.
5. Ecrire une fonction récursive qui trie une liste dans l'ordre croissant.