Numérique et Science Informatique

Spécialité de terminale

TD : Récursion

Jean-Damien Leroy et Arnaud Saint-Martin

Thème 4 : Langages et programmation

Exercice 1			*

Commençons par un appel récursif simple en python :

```
def fonct(n):
    if n>0:
        fonct(n-1)
    print(n)

fonct(3)
```

Vous remarquez que cette fonction s'appelle elle-même lorsque n>0. Une fonction récursive est une fonction qui s'appelle elle-même avec un paramètre (très souvent décroissant) et une condition d'arrêt de ces appels (ici, il n'y a plus d'appels récursifs si n est inférieur ou égal à 0).

1. Exécutez l'appel fonct(3) manuellement. Préciser les appels successifs à fonct. Quelle est la première valeur affichée par lors de l'appel fonct(3) ?
2. Utilisez Python pour vérifier que les affichages se font bien dans le même ordre que lors de votre exécution à la question précédente.

Les appels à fonct se font dans cet ordre : fonct(3) (qui appelle fonct(2) (qui appelle fonct(1) (qui appelle fonct(0) et affiche 0) et affiche 1) et affiche 2) et affiche 3.
L'interpréteur Python utilise une Pile dans laquelle il empile fonct(3) puis fonct(2) puis fonct(1) puis fonct(0) puis il dépile fonct(0) et affiche 0, ensuite il dépile fonct(1) et affiche 1...

Exercice 2			*

Un cas où le paramètre d'appel n'est pas décroissant : Syracuse.

Définition admettant la conjecture de Syracuse (qui dit que toute suite de Syracuse aboutit au terme 1): Une suite de Syracuse sera donc pour nous une suite de nombres entiers (voir programme de spécialité maths de première) finie définie par :

Prendre un nombre entier:
- Si ce nombre est pair, le diviser par 2.
- Si ce nombre est impair, prendre le triple et ajouter 1.

On obtient un nouveau nombre entier et on recommence :
- Si ce nombre est pair, le diviser par 2.
- Si ce nombre est impair, prendre le triple et ajouter 1.

Et on recommence ainsi de suite avec les entiers successifs obtenus...

1. Déterminer manuellement les termes de cette suite avec 12 comme premier terme.
2. Le site web : https://www.dcode.fr/conjecture-syracuse permet d'automatiser ces calculs, vérifiez que vos calculs avec 12 sont les mêmes.
3. Vous avez remarqué qu'à chaque étape, on fait appel aux mêmes calculs sur un nombre différent, proposez une fonction récursive en langage naturel qui affiche les termes successifs d'une suite de Syracuse et faites-là tester à votre enseignant.

Exercice 3			* *

Un grand classique mathématique : la fonction factorielle.

Voici 2 définitions de la factorielle d'un entier :

```
def factoiter(n):
    """factoiter(n): calcul itératif de la factorielle de n (entier >=
0)"""
    x=1
    if n==0 :
        return 1
    for i in range(1,n+1):
        x=x*i
    return x

def factorecur(n):
    """factorecur(n): calcul récursif de la factorielle de n (entier >=
0)"""
    if n<2:
        return 1
    return n*factorecur(n-1)
```

1. Calculer manuellement factorecur(3) et factoiter(3).
2. Laquelle de ces 2 fonctions vous paraît permettre le mieux exprimer la définition de la factorielle d'un nombre ?

3. Essayez, en utilisant votre interface python, de faire afficher factoiter(1000) et factorecur(1000). Quel est le problème ?

Vous allez maintenant comparer les 2 fonctions en temps, pour cela vous utiliserez la bibliothèque time dont la fonction time.time() renvoie le temps horloge en secondes de type float.

4. Créez la fonction tempsiter(n) qui renvoie le tuple (factoiter(n), temps de calcul)
5. Déterminez le temps d'exécution pour n=5, n=10, n=50, n=100, n=1000, n=10000.

Il aurait été bon de pouvoir comparer les temps de calcul de factoiter et factorecur mais les limitations de la pile rendent les temps d'exécution trop petits pour pouvoir être mesurés. Sachez que les 2 méthodes se valent en termes de temps de calcul.

Exercice 4			* *

!lindrome ou !(!lindrome) ?

Un palindrome est un nombre ou une chaîne de caractères qui se lit de la même manière de la gauche vers la droite que de la droite vers la gauche. 12321, 6886, 'SEES', 'LAVAL' ou encore 'ABBA' et 'ZAZ' sont des palindromes (on ne tient pas compte des accents, des espaces, de la ponctuation et de la casse). Il existe en version plus compliquée des phrases ou des textes palindromes : 'ROSESVERTESETREVESOR' pour les poètes ou le grand palindrome de Georges Perec pour les fanatiques de l'OULIPO (https://www.ecriture-art.com/palindrome.pdf).

1. En remarquant que pour qu'une chaîne de caractère soit un palindrome, il faut que la première et la dernière lettre soient identiques et que le reste de cette chaîne soit aussi un palindrome, créez une fonction python testPalindrome(chaine) (récursive bien sur) qui renvoie True si chaine est un palindrome et False sinon.
2. Testez votre fonction avec 'Elu par cette crapule.', 'Engage le jeu que je le gagne' et 'Eric, notre valet, alla te laver ton ciré.'. Vous penserez à retirer les accents, les espaces, la ponctuation et à n'utiliser que des majuscules.

Exercice 5			* * *

Tours de Hanoi.

Pour cet exercice, vous utiliserez le simulateur : https://fr.goobix.com/jeux-en-ligne/tours-de-hanoi/#

1. Résoudre le problème avec 3 anneaux.
2. Résoudre le problème avec 4 anneaux puis essayez d'expliciter une méthode pour 5 anneaux.
3. Imaginez que vous disposez d'une fonction hanoi7(a,b) qui résout le problème avec 7 anneaux de la tour a à la tour b, créez une fonction en langage naturel qui résout le problème avec 8 anneaux de la tour 1 à la tour 3.
4. Réfléchir à la fonction récursive hanoi(a,b,n) qui envoie les n anneaux supérieurs de la tour a sur la tour b