NSI · TERMINALE / EXERCICES CORRIGÉS

La récursivité

Astrolab Cours · astrolabcours.fr

Ex. 1 Somme des entiers ●○○ Application

Écrire une fonction récursive `somme(n)` qui renvoie $1 + 2 + \cdots + n$, avec `somme(0) = 0`.

Indice

$somme(n) = n + somme(n-1)$.

Le cas de base

Si `n == 0`, on renvoie 0.

La fonction

```
def somme(n):
    if n == 0:
        return 0
    return n + somme(n - 1)
```

Ex. 2 La pile d'appels ●●○ Méthode

Écrire la suite des appels et des retours lors de `puissance(2, 5)` avec la fonction de puissance rapide du cours.

Indice

5 est impair, 4 est pair, 2 est pair, 1 est impair.

La descente

`puissance(2, 5)` appelle `puissance(2, 4)`, qui appelle `puissance(2, 2)`, qui appelle `puissance(2, 1)`, qui appelle `puissance(2, 0)`.

La remontée

`puissance(2, 0)` = 1, puis `puissance(2, 1)` = 2 × 1 = 2, `puissance(2, 2)` = 2 × 2 = 4, `puissance(2, 4)` = 4 × 4 = 16, `puissance(2, 5)` = 2 × 16 = 32.

© Astrolab Cours · astrolabcours.fr/nsi/terminale/recursivite/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

Le résultat

On obtient 32 en 5 appels.

Ex. 3 La fonction qui ne s'arrête pas ●●○ Piège

Un élève écrit :

```
def compte(n):
    print(n)
    compte(n - 1)
```

Que se passe-t-il pour `compte(3)` ? Corriger pour afficher 3, 2, 1.

Indice

Où est le cas de base ?

L'erreur

Il n'y a pas de cas de base : la fonction affiche 3, 2, 1, 0, −1… jusqu'à la RecursionError.

La correction

On ajoute un cas de base :

```
def compte(n):
    if n == 0:
        return
    print(n)
    compte(n - 1)
```

Ex. 4 Nombre de chiffres ●●● Type bac

Écrire une fonction récursive `nb_chiffres(n)` qui renvoie le nombre de chiffres d'un entier `n` positif (par exemple 3 pour 507). Justifier qu'elle termine.

Indice

`n // 10` retire le dernier chiffre.

L'idée

Un nombre inférieur à 10 a 1 chiffre. Sinon, il a un chiffre de plus que `n // 10`.

© Astrolab Cours · astrolabcours.fr/nsi/terminale/recursivite/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

La fonction

```
def nb_chiffres(n):
    if n < 10:
        return 1
    return 1 + nb_chiffres(n // 10)
```

La terminaison

`n // 10` est un entier positif strictement plus petit que `n` quand `n ≥ 10` : on atteint forcément le cas `n < 10`.

© Astrolab Cours · astrolabcours.fr/nsi/terminale/recursivite/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits