

EXERCICE 2 (4 points)

Cet exercice est consacré à l'analyse et à l'écriture de programmes récur­sifs.

1.

- a. Expliquer en quelques mots ce qu'est une fonction récur­sive.
- b. On considère la fonction Python suivante :

Numéro de lignes	Fonction <code>compte_rebours</code>
1	<code>def compte_rebours(n):</code>
2	<code> """ n est un entier positif ou nul """</code>
3	<code> if n >= 0:</code>
4	<code> print(n)</code>
5	<code> compte_rebours(n - 1)</code>

L'appel `compte_rebours(3)` affiche successivement les nombres 3, 2, 1 et 0. Expliquer pourquoi le programme s'arrête après l'affichage du nombre 0.

2. En mathématiques, la factorielle d'un entier naturel n est le produit des nombres entiers strictement positifs inférieurs ou égaux à n . Par convention, la factorielle de 0 est 1. Par exemple :

- la factorielle de 1 est 1
- la factorielle de 2 est $2 \times 1 = 2$
- la factorielle de 3 est $3 \times 2 \times 1 = 6$
- la factorielle de 4 est $4 \times 3 \times 2 \times 1 = 24 \dots$

Recopier et compléter sur votre copie le programme donné ci-dessous afin que la fonction récur­sive `fact` renvoie la factorielle de l'entier passé en paramètre de cette fonction.

Exemple : `fact(4)` renvoie 24.

Numéro de lignes	Fonction <code>fact</code>
1	<code>def fact(n):</code>
2	<code> """ Renvoie le produit des nombres entiers</code>
3	<code> strictement positifs inférieurs à n """</code>
4	<code> if n == 0:</code>
5	<code> return à compléter</code>
6	<code> else:</code>
7	<code> return à compléter</code>

3. La fonction `somme_entiers_rec` ci-dessous permet de calculer la somme des entiers, de 0 à l'entier naturel `n` passé en paramètre.

Par exemple :

- Pour `n = 0`, la fonction renvoie la valeur 0.
- Pour `n = 1`, la fonction renvoie la valeur $0 + 1 = 1$.
- ...
- Pour `n = 4`, la fonction renvoie la valeur $0 + 1 + 2 + 3 + 4 = 10$.

Numéro de lignes	Fonction <code>somme_entiers_rec</code>
1 2 3 4 5 6 7 8	<pre>def somme_entiers_rec(n): """ Permet de calculer la somme des entiers, de 0 à l'entier naturel n """ if n == 0: return 0 else: print(n) <i>#pour vérification</i> return n + somme_entiers_rec(n - 1)</pre>

L'instruction `print(n)` de la ligne 7 dans le code précédent a été insérée afin de mettre en évidence le mécanisme en œuvre au niveau des appels récursifs.

- a. Écrire ce qui sera affiché dans la console après l'exécution de la ligne suivante :

```
res = somme_entiers_rec(3)
```

- b. Quelle valeur sera alors affectée à la variable `res` ?

4. Écrire en Python une fonction `somme_entiers` non récursive : cette fonction devra prendre en argument un entier naturel `n` et renvoyer la somme des entiers de 0 à `n` compris. Elle devra donc renvoyer le même résultat que la fonction `somme_entiers_rec` définie à la question 3.

Exemple : `somme_entiers(4)` renvoie 10.