

# 1 TP 1

Nom et prénom : .....

Ouvrir le logiciel *EduPython*. La fenêtre est partagée en trois parties : en haut à droite, c'est l'éditeur, là où il y aura le programme, et en bas c'est la console, où seront affichés les résultats.

## Exercice 1.1

Ouvrir le fichier `ts4-tp1-ex1.py` et l'enregistrer sous la forme `nom-prenom-ts4-tp1-ex1.py`

On voit alors dans l'éditeur un programme composé de deux fonctions.

Ces deux fonctions sont dans les tableaux 1.1 et 1.2 ci-dessous. Dans chaque tableau, la fonction est traduite à gauche sous forme d'algorithme.

Pour les variables  $x$ ,  $y$ ,  $a$ ,  $k$ ,  $n$ , on précise que  $x$ ,  $y$  et  $a$  sont des nombres réels et que  $k$  et  $n$  sont des entiers naturels.

	Algorithme 1	Fonction en python3
1		<code>def f(x):</code>
2	$y \leftarrow 2x + 5$	<code>    y=2*x+5</code>
3		<code>    return y</code>

Tab. 1.1

	Algorithme 2	Fonction en python3
1		<code>def calcul(n):</code>
2	$a \leftarrow 1$	<code>    a=1</code>
3	Pour $k = \dots$ jusqu'à $k = \dots$	<code>    for k in range(n):</code>
4	$a \leftarrow f(a)$	<code>        a=f(a)</code>
5	Fin du Pour	
6		<code>    return a</code>

Tab. 1.2

1. Lancer ce programme en cliquant sur le bouton avec un triangle vert. Dans la console, on voit simplement le message : `*** Console de processus distant réinitialisée ***`.

2. Exécuter maintenant la fonction `f` en saisissant `f(4)` dans la console, et en validant (appuyer sur la touche Entrée). On voit :

```
>>> f(4)
```

```
13
```

Rien d'étonnant puisque :  $f(4) = 2 \times 4 + 5 = 13$

3. Intéressons nous maintenant à la fonction `calcul`. Contrairement à la fonction `f`, cette fonction ne ressemble pas à une fonction mathématique.

Exécuter maintenant la fonction `calcul` en saisissant `calcul(3)` dans la console, puis en validant. On voit :

```
>>> calcul(3)
```

```
43
```

Suite au verso

a) Détailler ci-dessous les calculs qui aboutissent à 43.

.....

.....

.....

.....

.....

.....

Si on ne comprend pas les calculs qui sont exécutés, on peut ajouter des commandes `print` dans la fonction `calcul` par exemple, en complétant comme ci-dessous. Quand c'est fait, ne pas oublier de relancer ce programme en cliquant à nouveau sur le bouton avec un triangle vert.

```
def calcul(n):
    a=1
    print("a = ",a)
    print("-----")
    for k in range(n):
        print("k = ",k)
        a=f(a)
        print("a = ",a)
        print("-----")
    return a
```

b) Lorsque  $n = 3$ , dans l'instruction `for k in range(n):`, les valeurs de  $k$  vont de quel nombre à quel nombre ?

.....

.....

c) Ces calculs permettent d'obtenir un terme d'une suite.

- Comment est définie cette suite ?

.....

.....

- Dans cette suite, quelle est la valeur de  $u_0$  ?

.....

.....

- Le résultat 43 correspond à quel terme ?  $u_1$  ?  $u_2$  ?  $u_3$  ?  $u_4$  ?  $u_5$  ?  $u_6$  ?

.....

.....

**Exercice 1.2**

Ouvrir le fichier `ts4-tp1-ex2.py` qui contient la fonction `calcul2` qui est dans le tableau 1.3.

	Algorithme 3	Fonction en python3
1		<code>def calcul2(A):</code>
2	$k \leftarrow 0$	<code>    k=0</code>
3	Tant que $u < A$	<code>    while(u&lt;A):</code>
4	$k \leftarrow k + 1$	<code>        k=k+1</code>
5	$u \leftarrow 7k^3 + 91k^2$	<code>        u=7*k**3+91*k**2</code>
6	Fin du Tant que	
7		<code>    return k</code>

Tab. 1.3

1. Lancer ce programme en cliquant sur le bouton avec un triangle vert.
2. Exécuter maintenant la fonction `calcul2` en saisissant `calcul2(1500)` dans la console, et en validant. Le résultat est 4.
3. Détailler ci-dessous les calculs qui aboutissent à 4. Indiquer la valeur de  $k$  et de  $u$  à chaque étape.

.....  
 .....  
 .....  
 .....  
 .....

4. Ce programme est lié à une suite  $(u_n)$ . Comment est définie cette suite?

.....  
 .....

5. Pour la suite  $(u_n)$ , que représente la valeur finale de la variable  $k$  dans cet algorithme, quand on entre un nombre  $A$  strictement positif?

.....  
 .....

6. Exécuter la fonction `calcul2` en saisissant `calcul2(100000)` dans la console, puis même chose avec `calcul2(1000000)`. Noter les résultats ci-dessous :

$A = 100\,000$ ,  $k = \dots\dots\dots$  ;  $A = 1\,000\,000$ ,  $k = \dots\dots\dots$

7. Pour un nombre  $A$  positif aussi grand que l'on veut, on est certain que la boucle Tant que de cet algorithme se termine. Pourquoi?

.....  
 .....  
 .....