

TP 1 : Quelques bases en Python

I Opérations élémentaires

Exercice 1 :

1. Taper dans la console les lignes de commandes suivantes :

6/2 6//2 7/2 7//2 2**3 3**2 16%3 17%3

2. Expliquer par une courte phrase ce que fait chacune des opérations `/` , `//` , `**` et `%` .

II La boucle for

Exercice 2 :

Que font les six instructions Python suivantes :

1. `for i in range(1, 12, 1):`
 `print(i)`

2. `for i in range(1, 12):`
 `print(i)`

3. `for i in range(0, 13, 3):`
 `print(i)`

4. `for i in range(6):`
 `print(i)`

5. `for i in range(16,6,-1):`
 `print(i)`

6. `for i in range(19,-7,-4):`
 `print(i)`

Exercice 3 :

Quel programme écrire pour obtenir l'affichage suivant :

3
7
11
15
19

III La boucle while

Exercice 4 :

Expliquer en une phrase que fait ce programme :

```
u=1
while u<500:
    print(u)
    u=u*2
```

Exercice 5 :

Écrire un programme similaire pour obtenir la liste des puissances de 3 inférieures ou égales à 2000.

IV Fonction en python

Exercice 6 :

Voici le code d'une fonction python :

```
def opp(a):
    return -a
```

1. Que renvoie `opp(-3)` ?

2. Que fait la fonction `opp` en général ?

3. Écrire une fonction `inv` qui renvoie l'inverse d'un réel non nul.

4. Tester la fonction `inv` avec le paramètre 5, puis le paramètre $\frac{1}{3}$. Donner les valeurs obtenues.

Exercice 7 :

Écrire une fonction `perim` qui prend en paramètre la longueur et la largeur d'un rectangle et renvoie son périmètre.

Exercice 8 :

1. Soit f la fonction définie sur $\mathbb{R}$ par $f(x) = x^3 - 5x^2 + 3$.

La fonction python suivante **f** de paramètre **x** permet de renvoyer la valeur de $f(x)$.

```
def f(x):  
    return x**3-5*x**2+3
```

Utiliser cette fonction pour donner $f(-1)$, $f(1)$, $f(1,5)$ et $f\left(\frac{5}{2}\right)$.

2. Soit g la fonction définie sur $\mathbb{R}$ par $g(x) = 2x^3 + 3x - 5$.

- (a) Écrire une fonction python **g** de paramètre **x** qui renvoie la valeur de $g(x)$.

- (b) Utiliser cette fonction pour compléter le tableau de valeur suivant :

x	-2	-1	0	1	2	3
$g(x)$						

V Structure conditionnelle

Exercice 9 :

Voici le code d'une fonction python :

```
def f(a):  
    if a<0:  
        return -a  
    else:  
        return a
```

1. Que renvoie $f(2)$ et $f(-3)$?

2. Que fait la fonction **f** en général ?

VI Pour aller plus loin : Syracuse

Exercice 10 : Cet exercice porte sur la suite de Syracuse à partir d'un entier positif x , on construit à la main une suite (u_n) de la manière suivante :

- $u_0 = x$
- $u_{n+1} = \begin{cases} \frac{u_n}{2} & \text{si } u_n \text{ est pair} \\ 3u_n + 1 & \text{si } u_n \text{ est impair} \end{cases}$

1. Prenons $x = 5$, calculer les valeurs de $u_1, u_2, u_3, u_4, \dots, u_9$

2. On considère l'algorithme ci-dessous :

Algo

Variables : x,n nombres entiers naturels

Traitement :

Entrée x

n prend la valeur 0

Tant que x différent de 1 faire

 Si x est un nombre pair alors

 x prend la valeur x/2

 Sinon:

 x prend la valeur 3*x+1

 n prend la valeur n+1

Fin Tant que

Renvoyer n

Fin

Exécuter l'algorithme à la main en complétant le tableau suivant pour les valeurs de x suivantes :

Pour $x = 3$:

	x	n	$x \neq 1$
Initialisation	3	0	Vrai

Pour $x = 9$:

	x	n	$x \neq 1$
Initialisation	9	0	Vrai

Pour $x = 6$:

	x	n	$x \neq 1$
Initialisation	6	0	Vrai

3. Écrire la fonction correspondante en Python de paramètre x et la tester avec les valeurs de x ci-dessus ainsi que pour $x = 127$.