

Exercice 1 :

On se donne la fonction suivante :

```
def echanger(a, b):
    z=a
    a=b
    b=z
a=5
b=3
echanger(a, b)
print ("a vaut : " + str(a))
print ("b vaut : " + str(b))
```

- (1) À quoi sert cette fonction ?
- (2) Tester la fonction, que remarquez-vous ?
- (3) Corriger ce programme pour le faire fonctionner.

Exercice 2 :

On se donne la fonction suivante :

```
def echanger2(l):
    z=l[0]
    l[0]=l[1]
    l[1]=z
l = [1, 2]
echanger2(l)
print ("l vaut : " + str(l))
```

- (1) À quoi sert cette fonction ?
- (2) Tester ce programme.
- (3) Quelle est la différence avec le programme précédent ?

Exercice 3 :

Tester la fonction :

```
def echanger3():
    global a, b
    z=a
    a=b
    b=z
print ("a vaut : " + str(a))
print ("b vaut : " + str(b))
```

Expliquer pourquoi ce programme doit fonctionner.

Exercice 4 :

Sans ordinateur

Qu'affiche le programme suivant ? Pourquoi ?

```
def g(x):
    global a
    a=10
    return 2*x
def f(x):
    v=1
```

```

    return g(x)+v
a=3
print (f(a)+a)

```

Exercice 5 :

Écrire une fonction qui prend en argument un entier n et renvoie l'entier 2^n

Exercice 6 :

Écrire en Python une fonction qui prend comme argument un flottant x et renvoie la valeur de $\sqrt{x} + \cos x$. On utilisera une fonction dans une bibliothèque bien choisie (l'aide est ici <https://docs.python.org/3/>)

Exercice 7 :

On se donne la suite u définie par :

$$\begin{cases} u_{n+1} &= 2u_n + 1 \\ u_0 &= 1 \end{cases}$$

- (1) Proposer une fonction qui permet d'afficher les 100 premiers termes.
- (2) Proposer une fonction qui permet d'afficher le 100ème terme.
- (3) Proposer une fonction qui permet d'afficher le plus petit indice n tel que $u_n > 1000000$.

Exercice 8 :

$n!$ est le nombre définie par :

$$n! = 1 \times 2 \times 3 \times \dots \times n.$$

Ce nombre est le terme général de la suite définie par :

$$u_{n+1} = (n+1)u_n \text{ et } u_0 = 1$$

On se donne la fonction suivante :

```

def factorielle(n):
    if n == 0:
        return 1
    else:
        return n * factorielle(n-1)

```

- (1) Tester cette fonction pour différente valeur de n .
- (2) Pourquoi ce programme fonctionne t'il ?
- (3) Écrire la fonction **factorielle_iterative** qui calcule la fonction factorielle à l'aide d'une boucle.

On appelle ce type de fonction des fonctions récursives

Exercice 9 :

Écrire une fonction **chercher** qui prend en argument une liste et un nombre et renvoie l'indice de la première occurrence de ce nombre dans la liste si il existe et -1 sinon. Par exemple :

```

L = [1,2,5,1]
print (chercher(L,2))
--> 1
print (chercher(L,3))
--> -1
print (chercher(L,1))
--> 0

```