

Exercice 1 Combien d'opérations élémentaires effectuent ces programmes en python ?

```
def A():
    a=5
    while a > 0:
        a = a - 1
def B():
    for i in range(7):
        A()
def C(n):
    a = 1
    for i in range(1,n+1):
        a = i * a
    return a
```

Exercice 2 Soit un entier n . Comparez n , $2^{\lfloor \log_2 n \rfloor}$ et $2^{\lfloor \log_2 n \rfloor + 1}$.

Exercice 3 Montrez que $\log_b n = \log_b a \times \log_a n$.

Exercice 4 Démontrez que $\log_b n = O(\log_2 n)$ pour toute base $b > 1$.

Exercice 5 Codez 1023 en base 2.

Exercice 6 On considère le programme python suivant:

```
def D(n,b):
    tab=[]
    while n>0:
        tab=[n%b]+tab
        n=n//b
    return tab
```

1. Donnez l'équation reliant les composantes, notées a_0, a_1, \dots, a_{k-1} , du tableau renvoyé par $D(n, b)$ et les entiers n, b .
2. Donnez l'équation reliant les entiers n, b, k .

Exercice 7 Montrez que, pour $c > 0$ et $f(n) = 1 + c + c^2 + \dots + c^n$ alors

1. $f(n) = \Theta(1)$ si $c < 1$
2. $f(n) = \Theta(n)$ si $c = 1$
3. $f(n) = \Theta(c^n)$ si $c > 1$

Exercice 8 Montrez que $\log(n!) = \Theta(n \log n)$.

Exercice 9 Montrez que $\sum_{i=1}^{i=n} \frac{1}{i} = \Theta(\log n)$.

Exercice 10 Montrez que $E(n, p)$ est exponentiel.

```
def E(n,p):
    if p==n or p==0:
        return 1
    return E(n-1,p-1)+E(n-1,p)
```

Exercice 11 Donnez les 5 lignes de code python constituant un algorithme récursif `Euclide(a,b)` renvoyant le pgcd de deux entiers $a \geq b$, et montrer qu'il est $O(\log b)$.

Exercice 12 Remplissez les lignes manquantes:

```
def tri_den(A):
    n=len(A)
    m=max(A)
    B=[]
    for j in range(n):
        B.append(0)
    C=[]
    for i in range(m+1):
        C.append(0)
    for j in range(n):

        for i in range(1,m+1):

            for j in range(n):
                B[C[A[j]]-1]=A[j]
    A=B
```

Exercice 13 Remplissez les lignes manquantes:

```
def tri_insertion(tab):
    for j in range(1,len(tab)):

        i=j-1
        while i>=0 and tab[i]>x:

            i=i-1
            tab[i+1]=x
```

Exercice 14 Donnez la sortie écran à l'appel de la fonction `tri(tab)` avec l'entrée `tab=[1,8,7,54,78,53]`.

```
def tri(tab):
    print(*tab)
    if len(tab) >1:
        mi = len(tab)//2
        L = tab[:mi] # le sous-tableau tab[0]..tab[mi-1]
        R = tab[mi:] # le sous-tableau tab[mi]..tab[len-1]
        tri(L)
        tri(R)
        i = j = k = 0
        while i < len(L) and j < len(R):
            if L[i] < R[j]:
                tab[k] = L[i]
                i+= 1
            else:
                tab[k] = R[j]
                j+= 1
            k+= 1
        while i < len(L):
            tab[k] = L[i]
            i+= 1
            k+= 1
        while j < len(R):
            tab[k] = R[j]
            j+= 1
            k+= 1
```