

Fonctions

Exercices

Exercice 1

Ecrire une fonction **MIN** et une fonction **MAX** qui déterminent le minimum et le maximum de deux nombres réels.
Ecrire un programme se servant des fonctions **MIN** et **MAX** pour déterminer le minimum et le maximum de quatre nombres réels entrés au clavier.

Exercice 2

Ecrire un programme se servant d'une fonction **f** pour afficher la table de valeurs de la fonction définie par

$$f(x) = \sin(x) + \ln(x) - \sqrt{x}$$

où x est un entier compris entre 1 et 10.

Exercice 3

En utilisant une boucle **while**, écrire la fonction **logarithme_binaire**, qui à un réel strictement positif **a**

associe : $\min \{n \in \mathbb{N}, 2^n > a\}$

Exercice 4

En utilisant la boucle **for** écrire la fonction **premier** qui détermine si un entier naturel **n** est premier ou non.

Exercice 5

En utilisant une boucle **while**, écrire la fonction **plus_petit** qui, à un entier naturel **n ≥ 2**, associe son plus petit diviseur premier.

Exercice 6

Définir la fonction **facteursPremiers** qui affiche la décomposition d'un entier $n \geq 2$ en produit de facteurs premiers. (On pourra utiliser la fonction **plus_petit** ci-dessus).

- a- Version itérative
- b- Version récursive

Exemple :

```
>>> facteursPremiers(132)
2 * 2 * 3 * 11

>>> facteursPremiers_rec(132)
2 * 2 * 3 * 11
```

Exercice 7 : (Calcul d'une suite de Syracuse)

Ecrire la fonction **Syracuse(n)** définie par la suite:

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

qui affiche les n termes de cette suite.

- a- Version itérative
- b- Version récursive

Exemple :

```
>>> Syracuse(13)
40 20 10 5 16 8 4 2 1

>>> Syracuse_rec(13)
40 20 10 5 16 8 4 2 1
```

Correction

<pre>##Exercice 1 def MIN(A,B) : if A<B: return A else: return B def MAX(A,B) : if A>B: return A else: return B ##Programme principal a=float(input("Donne a:")) b=float(input("Donne b:")) c=float(input("Donne c:")) d=float(input("Donne d:")) print("Maximum=",MAX(MAX(a,b),MAX(c,d))) print("Miniimum=",MIN(MIN(a,b),MIN(c,d)))</pre>	<pre>##Exercice 5 def plus_petit(n): i=2 while i<=n: if n%i==0 and premier(i): return i i=i+1</pre>
<pre>##Exercice 2 from math import sin, log, sqrt def f(x) : return sin(x) + log(x) - sqrt(x) ###Programme for x in range (1,11): print (x,'\t',f(x))</pre>	<pre>##Exercice 6 def facteurs (n): i=2 print (n,"=", end=' ') while(i<=n): p= plus_petit(n) if(i<=n/p): print (p,"*", end=' ') else: print (p) n=n/p i=i+1 def facteurs_rec(n): if premier(n): print(n) else: p=plus_petit(n) print (p,"*", end=' ') facteurs_rec(n//p)</pre>
<pre>##Exercice 3 def logarithme_binaire(a): n=0 while 2**n<=a: n=n+1 return n;</pre>	<pre>##Exercice 7 def Syracuse(n): while(n>1): if n%2==0: n=n//2 else: n=3*n+1 print (n, end=' ') def Syracuse_rec(n): if n>1: if n%2==0: n=n//2 else: n=3*n+1 print (n, end=' ') Syracuse_rec (n)</pre>
<pre>##Exercice 4 from math import sqrt def premier(x): for i in range (2, int(sqrt(x)+1)): if(x%i==0): return False return True</pre>	