Fonctions en Python

Exercice 1 (QCM – sans utiliser Python)

Voici ci-contre une fonction définie en Python.
que renvoie la fonction f si le paramètre x vaut 15 ?

```
def f(x):
    for d in range(2,x):
        if x%d == 0:
            return d
```

1. 3
2. 5
3. 3 et 5
4. 3, 5 et 15

Exercice 2

Écrire une fonction qui prend en paramètre deux nombres et renvoie le plus grand des deux.

Exercice 3

Écrire une fonction qui prend en paramètre trois nombres et renvoie le plus grand des trois nombres.

Exercice 4

Écrire une fonction `SommeCarres` qui prend en paramètre un entier strictement positif `k` et qui renvoie la somme des `k` premier carrés non nuls.

Exercice 5

Écrire une fonction `SommeDiviseurs` qui prend en paramètre un entier naturel non nul et renvoie la somme de ses diviseurs.

Exercice 6

Écrire une fonction `Premier` qui prend en paramètre un entier naturel non nul et renvoie `True` si ce nombre est premier et `False` sinon.

Exercice 7

En utilisant la fonction `randint` du module `random`, écrire une fonction `Jeu` qui prend en paramètre un nombre entier `n` stritement positif, simule `n` fois le tirage d'un dé cubique équilibré (donc une valeur au hasard parmi les nombres 1,2,3,4,5 et 6), puis renvoie le pourcentage de 6 obtenus.

Exercice 8 (Facultatif)

définir une fonction qui prend pour paramètres trois entiers n, inf et sup, et qui affiche la table de multiplication de n entre inf et sup. Par exemple, l'exécution de la fonction avec les paramètres valant respectivement 2, 3 et 5 affichera :

```
2*3=6
2*4=8
2*5=10
```

Penser à vérifier que inf est bien inférieur à sup et, si ce n'est pas le cas, échanger les valeurs.

Exercice 9 (Facultatif)

Soit f la fonction mathématique définie par $f(x) = x^3 - 3x + 2$.
Définir une fonction `tabulation` qui prend en paramètre trois nombres inf, sup et pas de type `float` et qui affiche les images des valeurs de inf à sup avec un pas de pas, sous une forme compréhensible (comme `f(1)=0`). Même chose que pour l'exercice précédent pour ce qui est de inf et sup.

Encore d'autres exercices

Exercice 10
Définir une fonction qui prend en argument un nombre entier n et qui :
- s'il est pair, le divise par 2 ;
- s'il est impair, le multiplie par 3 et lui ajoute 1.

puis qui retourne la nouvelle valeur.

Exercice 11
On considère une fonction f dont l'expression est $f(x) = -2x$ si $x < 0$ et $f(x) = x^2$ sinon.
Définir une fonction qui prend en argument un nombre flottant x et qui retourne l'image $f(x)$.

Exercice 12
Même chose que l'exercice précédent, mais avec une fonction g définie en trois morceaux :

$$g(x) = \begin{cases} x^3 + 2 & \text{si } x < 0 \\ x^2 + 2 & \text{si } 0 \leqslant x < 3 \\ 14 - x & \text{si } x \geqslant 3 \end{cases}$$

Exercice 13
Définir une fonction factorielle qui prend comme argument un entier n et qui retourne le produit des entiers de 1 à n.

Exercice 14
Définir une fonction qui prend comme un argument un entier n et qui retourne le nombre d'entiers entre $-n$ et n (compris) pour lesquels $f(x) \leqslant g(x)$ (fonctions définies dans les exercices précédents).

Exercice 15
Définir une fonction qui prend comme argument un entier n et qui retourne la somme des inverses des entiers entre 1 et n.

Exercice 16
Définir une fonction qui prend comme argument un entier n, qui demande n nombres (flottants) et retourne la moyenne de ces nombres.

Exercice 17
Définir une fonction qui prend pour argument un nombre flottant $A > 0$ et qui retourne le plus petit entier n tel que $n^3 + 2n + 1 \geqslant A$.

Exercice 18
Définir une fonction SommeDuree qui prend pour argument quatre nombres entiers $m1$,$s1$,$m2$,$s2$ représentant deux durées en minutes et secondes, et qui retourne la somme des deux durées en minutes et secondes sous forme de chaîne de caractères.

Exercice 19
Définir une fonction sans argument qui demande deux nombres (flottants) et un symbole d'opération ($+$, $-$, / ou $*$) puis qui effectue et retourne le résultat du calcul à effectuer.

Exercice 20
Un nombre parfait est un nombre n dont la somme des diviseurs propres (c'est à dire strictement inférieurs à n) est égal à n. Définir une fonction qui affiche les entiers n parfaits situés entre 2 et 1000.
Pour cela on pourra réutiliser la fonction SommeDiviseurs vue dans un exercice précédent..

Exercice 21
Deux nombres entiers M et N sont amicaux si la somme des diviseurs propres de M est égal à N et si la somme des diviseurs propres de N est égale à M.
Définir une fonction qui prend en argument un nombre entier max et qui affiche les couples de nombres amicaux inférieurs à max.
On pourra tester la fonction avec max=3000.

Si vous avez terminé les exercices de cette fiche, allez sur le site france-IOI ou sur le site checkiO, lesquels nécessitent d'être inscrit, pour poursuivre l'entraînement.