

# DEVOIR N° 4 - CORRIGÉ

à rendre pour le jeudi 19 avril **DERNIER DÉLAI**

## PARTIE « ALGORITHMIQUE »

### Exercice n° 1 (.../5 points)

- ❖ Écris un algorithme permettant de construire un triangle équilatéral connaissant deux sommets de ce triangle.
  - Demander** le premier sommet A
  - Demander** le second sommet B
  - Faire** la construction de C, image de B par la rotation de centre A et d'angle  $60^\circ$
  - Afficher** le triangle ABC
- ❖ Écris un algorithme permettant le calcul de la somme des carrés de deux nombres.
  - Demander** le premier nombre x
  - Demander** le second nombre y
  - Faire** le calcul de  $x^2 + y^2$ , que l'on affecte dans r
  - Afficher** r
- ❖ Écris un algorithme permettant de construire un carré connaissant deux sommets consécutifs de ce carré.
  - Demander** le premier sommet A
  - Demander** le second sommet B
  - Faire** la construction de D, image de B par la rotation de centre A et d'angle  $60^\circ$
  - Faire** la construction de C, image de B par la translation de vecteur AD.
  - Afficher** le carré ABCD
- ❖ Écris un algorithme permettant d'afficher le plus grand des nombres parmi deux nombres donnés.
  - Demander** le premier nombre x
  - Demander** le second nombre y
  - Si**  $x > y$  **alors**  $r \leftarrow x$ , **sinon**  $r \leftarrow y$
  - Afficher** r
- ❖ Écris un algorithme qui permette d'envoyer un SMS à quelqu'un (qui n'est pas nécessairement dans les contacts de la personne qui envoie le SMS).
  - Demander** le texte t
  - Si** destinataire e **répertoire** **alors**
  - Choisir** nom
  - $n \leftarrow$  numéro associé
  - Sinon demander** numéro n
  - Envoyer** t à n

### Exercice n° 2 (.../3 points)

Voici un algorithme :

<b>Demander</b> un nombre a	#1
<b>Demander</b> un nombre b	#2
<b>Faire</b> le calcul de $a + 2b$	#3
<b>Affecter</b> le résultat dans b	#4
<b>Faire</b> le calcul de $b - a$	#5
<b>Affecter</b> le résultat dans a	#6
<b>Afficher</b> le nombre a	#7

1. Fais fonctionner cet algorithme pour  $a = 9$  et  $b = 5$ .

- ligne 4* : on affecte  $a + 2b = 9 + 2 \times 5 = 9 + 10 = 19$  dans b (qui change donc de valeur !)
- ligne 6* : on affecte  $b - a = 19 - 9 = 10$  dans a (et  $b = 19$ )
- ligne 7* : le programme affichera « a = 10 ».

2. Recommence avec deux autres valeurs que tu choisiras.

- Je choisis  $a = 2$  et  $b = 20$  :* *ligne 4* : on affecte  $a + 2b = 2 + 2 \times 20 = 2 + 40 = 42$  dans b
- ligne 6* : on affecte  $b - a = 42 - 2 = 40$  dans a (et  $b = 42$ )
- ligne 7* : le programme affichera « a = 40 ».

3. Que fait finalement cet algorithme ?

Le programme renvoie la valeur de  $a$  égale à  $2b$  (en effet : «  $a = b - a$  (ligne 5) =  $a + 2b - a$  (ligne 3) =  $2b$  ».

### Exercice n° 3 (.../5 points)

Voici un algorithme, dit « de Syracuse » (on suppose qu'il s'arrête) :

```
Demander un nombre entier naturel  $x$ 
Tant que  $x \neq 1$  faire
  Si  $x$  est impair alors
    Faire le calcul de  $3x + 1$ 
    Affecter le résultat dans  $x$ 
    Afficher  $x$ 
  Sinon
    Faire le calcul de  $x/2$ 
    Affecter le résultat dans  $x$ 
    Afficher  $x$ 
  Fin du « si »
Fin de la boucle
```

Fais fonctionner cet algorithme successivement pour  $x = 1$ ,  $x = 2$ ,  $x = 3$ ,  $x = 39$  et  $x = 40$ .

$x = 1$ : rien ne s'affichera car l'algorithme ne rentre pas dans la boucle ! $x = 2$ : 1 $x = 40$ : 20 - 10 - 5 - 16 - 8 - 4 - 2 - 1	$x = 39$ : 118 - 59 - 178 - 89 - 268 - 134 - 67 - 202 - 101 - 304 - 152 - 76 - 38 - 19 - 58 - 29 - 88 - 44 - 22 - 11 - 34 - 17 - 52 - 26 - 13 - 40 - 20 - 10 - 5 - 16 - 8 - 4 - 2 - 1
---	---

## PARTIE « PROGRAMMATION »

### Exercice n° 4 (.../5 points)

- ❖ Écris un programme qui demande un nombre entier et qui affiche les 10 nombres entiers suivants.
- ❖ Écris un programme qui réponde à la question suivante : « Pour quel entier  $n$ , la somme des  $n$  premiers entiers dépasse-t-elle 1000 ? »
- ❖ Écris un programme qui permette de permuter les valeurs de trois nombres  $a$ ,  $b$ ,  $c$  donnés (c'est-à-dire qui met  $a$  dans  $b$ ,  $b$  dans  $c$  et  $c$  dans  $a$ ).
- ❖ Écris un programme permettant d'afficher le plus grand des nombres parmi deux nombres donnés (adaptation en programme d'un algorithme de l'exercice 1).
- ❖ Écris un programme qui, en fonction d'un nombre entier naturel  $n$  donné, affiche  $n$  lignes de la forme suivante :

```
*
* *
* * *
* * * *
* * * * *
* * * * *
etc.
```

Il doit y avoir autant de lignes que le nombre  $n$  !

Voici le programme pour tout l'exercice (début à gauche, suite du programme à droite) :

```
# -*- coding:Latin-1 -*-
# DÉFINITION DES PROGRAMMES POUR CHAQUE EXERCICE
# -----
def prog1(n):
    for i in range(9):
        print n+i,"-"
```

```
# CORPS PRINCIPAL DU CODE
# -----
print("EXERCICE 5 - PROGRAMMES 1 et 5")
print("-----")
n = input("Entrez un entier n : ")
print("Programme 1 =>"),
prog1(n)
```





## Exercice n° 5 [.../2 points]

1. Convertis 12680 secondes en heures, minutes et secondes (12680 s = **3 h 31 min 20 s**) en détaillant les opérations effectuées.

$$12680 = 3 \times 3600 + 1880, \text{ donc } 12680 \text{ s} = 3 \text{ h } 1880 \text{ s}$$

$$\text{Or } 1880 = 31 \times 60 + 20, \text{ donc } 1880 \text{ s} = 31 \text{ min } 20 \text{ s, d'où } 12680 \text{ s} = 3 \text{ h } 31 \text{ min } 20 \text{ s.}$$

2.  
3. Écris un programme qui, pour un temps donné en secondes, le convertit en heures, minutes et secondes.

Voici le programme :

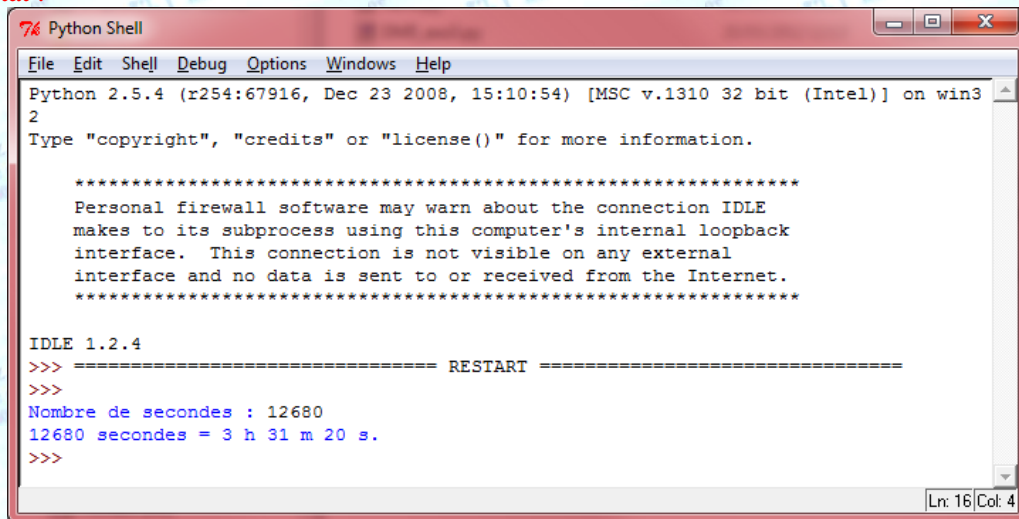
```
# -*- coding:Latin-1 -*-
# CORPS PRINCIPAL DU CODE
# -----
temps = input("Nombre de secondes : ")

# Détermination des heures :
h = temps/3600      # division entière !
reste = temps%3600 # le reste des secondes (qui ne font pas une heure)

# Détermination des minutes :
m = reste/60       # division entière !
s = reste%60       # le reste des secondes (qui ne font pas une minute)

# Affichage du résultat :
print temps,"secondes = ",h,"h",m,"m",s,"s."
```

Voici le résultat :



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 2.5.4 (r254:67916, Dec 23 2008, 15:10:54) [MSC v.1310 32 bit (Intel)] on win32
2
Type "copyright", "credits" or "license()" for more information.

*****
Personal firewall software may warn about the connection IDLE
makes to its subprocess using this computer's internal loopback
interface. This connection is not visible on any external
interface and no data is sent to or received from the Internet.
*****

IDLE 1.2.4
>>> ===== RESTART =====
>>>
Nombre de secondes : 12680
12680 secondes = 3 h 31 m 20 s.
>>>
```



LE CORRIGÉ SERA DISPONIBLE SUR

[WWW.CAPES-DE-MATHS.COM](http://WWW.CAPES-DE-MATHS.COM)

DÈS LE 19 AVRIL 2012.

PENSEZ À CONSULTER !!!