

## TD : boucles conditionnelles while

### Exercice 1

On donne la fonction mystere suivante qui n'a pas de paramètre.

```
def mystere():  
    n = 0  
    u = 1 # ICI  
    while u < 50:  
        u = u * 2  
        n = n+1 #LA  
    return n
```

On effectue un appel de cette fonction.

1. Compléter le tableau.

	u<50	u	n
#ICI	X	1	0
#LA	True	2	1
#LA			
#LA			
#LA			
#LA			
#LA			
#LA			

2. Que retourne cet appel?
3. Que fait cette fonction?

## Exercice 2

Voici une fonction qui effectue la division euclidienne entre deux nombres entiers positifs  $a$  et  $b$ , on suppose que  $a > b$  et retourne le quotient  $q$  et le reste  $r$  de cette division.

L'algorithme est le suivant:

Tant que l'on peut soustraire  $b$  au nombre de départ  $r$  on effectue cette soustraction et incrémente le quotient de 1. Puis on retourne le quotient  $q$  et le reste  $r$

```
def division(a: int,b: int):
    r = a
    q = 0 #ICI
    while r >= b:
        r = r-b
        q = q + 1 #LA
    return q,r
```

On effectue l'appel `division(16,3)`, dans la fonction  $a$  est affecté de la valeur 16 et  $b$  de la valeur 3.

1. Compléter la trace de cet appel:

	$r \geq b$	$r$	$q$
#ICI	X	16	0
#LA	True	13	1
#LA			
#LA			
#LA			
#LA			
#LA			

2. Que retourne cette fonction?

**Exercice 3**

```
def seuil() -> int:
    s = 0
    n = 0 #ICI
    while s < 20:
        n = n + 1
        s = s + n #LA
    return n
```

Dans cette fonction  $s$  est un accumulateur qui contient la somme des entiers de 1 à  $n$ . On effectue l'appel de `seuil`.

1. Compléter la trace de cet appel.

	$s < 20$	$n$	$s$
#ICI	X	0	0
#LA	True	1	1
#LA			
#LA			
#LA			
#LA			
#LA			
#LA			

2. Que retourne cet appel?
3. Traduire cela par une inégalité.

**Exercice 4**

Voici une fonction `est_premier` qui prend en argument un nombre entier positif `n` et retourne `True` si le nombre est premier et `False` sinon.

L'algorithme est le suivant: on effectue la division de `n` par tous les nombres compris entre 2 et `n//2` si aucun `n`'a divisé `n` alors `n` est premier.

```
def est_premier(n: int) -> bool:
    premier = True
    i = 1
    fin = n // 2
    while i <= fin:
        i = i + 1
        if n % i == 0:
            premier = False
            break
        #ICI
    return premier
```

1. On effectue l'appel `est_premier(329)`. Dans ce cas `fin` est associé à la valeur 164 et `i` est initialisée à 1. Compléter la trace de cet appel. Dans cet exercice, on ne considère que ce qu'il se passe à la fin de chaque tour de boucle, c'est-à-dire au moment du commentaire `#ICI`.

<code>i&lt;=fin</code>	<code>i</code>	<code>n % i == 0</code>	<code>premier</code>
True	2	False	True


2. Que retourne cet appel? Conclure: Le nombre 329 est-il premier?
3. En prenant modèle sur les questions 1 et 2, faire la trace de l'appel `est_premier(7)`. Que retourne cet appel? Que peut-on dire du nombre 7?

**Exercice 5**

1. Quelle est la valeur de  $n$  à la fin du programme suivant? Justifier en donnant l'état des variables  $a$  et  $n$  ainsi que, si possible, la valeur du test  $a \geq b$  au moment des commentaires *#ICI* et *#LA*.

```
n = 0
a = 27
b = 5 #ICI
while a >= b:
    n = n + 1
    a = a - b #LA
```

2. Quelle est la valeur à la fin de l'exécution du programme ci-dessous lorsque  $a = 10$  et  $b = 3$ . Justifier en donnant l'état des variables  $a$ ,  $b$  et  $r$  ainsi que la valeur du test  $n \neq 0$  au moment du commentaire *LA*

```
r = 0
while b != 0:
    r = a % b
    a = b
    b = r #LA
```

3. Même consigne que pour la question 2, en prenant  $a = 10$  et  $b = 15$ .

## Exercice 6

En 2006 Microsoft commercialise le Zune un lecteur MP3. Le 31 décembre 2008 à minuit tous ces appareils se sont arrêtés de fonctionner, ne répondant plus à aucune commande. En fait, ils n'étaient pas arrêtés, mais simplement bloqués et ne pouvaient que vider leur batterie. Recharger la batterie le 1ier janvier aboutissait au même problème. Ce n'est qu'à partir du 2 janvier que les Zune purent être à nouveau utilisés normalement.

Que c'est-il passé? Une erreur dans le programme écrit en C. Voici l'extrait du code fautif :

```
while( days > 365) {
    if( IsLeapYear( year ) ) {
        if ( days > 366) {
            days = days - 366;
            year = year + 1 ;
        }
    } e l s e {
        days = days - 365;
        year = year +1;
    }
}
```

days et year contiennent respectivement le numéro du jour et le numéro de l'année. La fonction IsLeapYear retourne true si l'année est bissextile et false sinon.

Sachant que l'année 2008 était bissextile, pouvez-vous expliquer ce qu'il s'est passé?