A.3.4 Découverte des boucles finies for

`range(fin)` retourne la progression croissante d'entiers consécutifs de `0` inclus à `fin` exclu.

`range(debut, fin)` retourne la progression croissante d'entiers consécutifs de `debut` inclus jusqu'à `fin` exclu.

`range(debut, fin, pas)` retourne la progression croissante d'entiers : `debut`; `debut+pas`; `debut+2*pas`; `debut+3*pas` ... strictement inférieurs à `fin`.

■ **Exemple A.3** On peut parcourir les éléments d'un objet `range` à l'aide d'une boucle `for`.

```
>>> for i in range(5):
...     print(i)
...
0
1
2
3
4
```

```
>>> for x in range(3, 8):
...     print(x)
...
3
4
5
6
7
```

```
>>> for entier in range(-3,2) :
...     nbr = entier**2
...     print(nbr)
...
9
4
1
0
1
```

Exercice 17 Choisir la bonne réponse :

1. En langage Python lors de instruction `for k in range(2,7)`, `k` prend les valeurs :
   (A) 2 et 7 (B) 2; 3; 4; 5 et 6 (C) 2; 3; 4; 5; 6 et 7 (D) 2; 3; 4; 5; 6; 7 et 8
2. En langage Python lors de instruction `for k in range(5)`, `k` prend les valeurs :
   (A) 0; 1; 2; 3 et 4 (B) 0; 1; 2; 3; 4 et 5 (C) 1; 2; 3 et 4 (D) 1; 2; 3; 4 et 5
3. La variable `k` prenne successivement toutes les valeurs entières de 0 à 33 si on utilise
   (A) `for k in range(0,33)` (B) `for k in range(0,34)` (C) `for k in range(33)` (D) `for k in range(34)`
4. En langage Python, lors de l'instruction `for k in range(2,29)`, `k` prend :
   (A) 29 valeurs (B) 28 valeurs (C) 27 valeurs (D) 26 valeurs

Exercice 18 Compléter :

Les valeurs prises par `i` sont .....

.....

À la fin du script, `a` vaut .....

À la fin du script, `b` vaut .....

```
1 a = 0
2 b = 0
3 for i  in range(3,17) :
4   a = a +1
5   b = b +1
```

Exercice 19 Compléter :

À la fin du script la variable a contient .....

.....

.....

```
a = "Je dis"
for i in range(1,5) :
    a = a+" bravo"
a = a+" !"
```

Exercice 20

Script A

```
n = 10
compteur = 0
for i in range(1,nbr) :
    if  n % i == 0 :
        print(i)
        compteur = compteur + 1
```

Nombre de boucles exécutées =

Script A affiche :

En fin de script A, compteur=

Script B

```
n = 11
compteur = 0
for i in range(1,n) :
    if  n % i == 0 :
        print(i)
    compteur = compteur + 1
```

Nombre de boucles exécutées =

Script B affiche :

En fin de script B, compteur=

Script C

```
n = 12
compteur = 0
for i in range(1,n) :
    if  n % i == 0 :
        print(i)
compteur = compteur + 1
```

Nombre de boucles exécutées =

Script C affiche :

En fin de script C, compteur=

Script D

```
p= 10
compteur = 0
for i in range(5) :
    p = 2 * p
    compteur = compteur + 1
```

Nombre de boucles exécutées =

En fin de script D :

compteur=

p=

Script E

```
nombre = 0
for lettre in "mathematiques" :
    if lettre == "e"
        nombre=nombre+1
```

Nombre de boucles exécutées =

En fin de script E :

nombre=

■ **Exemple A.4 — Je fais : principe accumulateur pour calculer une somme ou un produit.**

```
def calculA() :
    nbr = 0
    compteur = 0
    for i in range(100) :
        nbr = nbr + i
        compteur = compteur + 1
    return nbr, compteur
```

```
def calculB() :
    nbr = 0
    compteur = 0
    for i in range(5,15) :
        nbr = nbr + i**2
        compteur = compteur + 1
    return nbr, compteur
```

1. L'appel `calculA()` exécute .....boucles, et retourne `nbr` correspond à la somme :

   \+ + + + +...+ =

2. L'appel `calculB()` exécute .....boucles et retourne `nbr` correspond à la somme :

   \+ + + + +...+ =

Exercice 21 — à vous.

```
def calculC() :
    nbr = 0
    for i in range(5,10) :
        nbr = nbr + i**3
    return nbr
```

```
def calculD() :
    nbr = 1
    for i in range(2,6) :
        nbr = nbr * i
    return nbr
```

1. L'appel `calculC()` retourne :

2. L'appel `calculD()` retourne :

```
def calculE(n) :
    nbr = 0
    for i in range(1, n) :
        nbr = nbr + 2**(-i)
    return nbr
```

```
def calculF(n) :
    nbr = 0
    for i in range(n+1) :
        nbr = nbr + 4**(-i)
    return nbr
```

3. Les appels `calculE(10)` et `calculF(10)` retournent le résultat des calculs :