

Question [0pE101] On souhaite écrire un programme calculant le triple d'un nombre décimal et affichant le résultat. On a saisi le code suivant :

```
nombre = "5"
triple = nombre * 3
print(triple)
```

Quel va être le résultat affiché ?

☒ 555

☐ nombrenombrenombre

☐ 15

☐ 15.0

Question [0pE102] On a saisi le code suivant :

```
a = '1_+_1'
b = 1 + 1
c = '2'
d = 2
```

Quelle instruction permet d'afficher le message $1 + 1 = 2$?

☒ `print(a + '_=' + c)`
☐ `print(a + '_=' + d)`
☐ `print(b + '_=' + c)`
☐ `print(b + '_=' + d)`

Question [0pE103] On a saisi le code suivant :

```
a = 8
b = 5
a = a + b
b = a - b
a = a - b
```

Quelles sont les valeurs de a et b à la fin du programme ?

☒ a = 5 et b = 8

☐ a = 8 et b = 5

☐ a = 8 et b = 13

☐ a = 13 et b = 5

Question [0pE1]

☒
☐
☐
☐

Question [0pAr01] On exécute l'instruction ci-après. Quel est l'affichage attendu ?

```
>>> 4%2
```

☒ 0

☐ 1

☐ 2

☐ 4

Question [entiers 01] Quel est l'entier positif codé en base 2 sur 8 bits par le code 0011 1010 ?

☒ 58

☐ 45

☐ 25

☐ -12

Question [entiers 02] Le résultat de l'addition des deux nombres binaires 1101 et 0101 est:

☒ 10010

☐ 10110

☐ 10011

☐ 11010

Question [entiers 03] Convertir la valeur décimale 155 en binaire (sur un octet).

☒ 10011011

☐ 11011011

☐ 01111111

☐ 10010111

Question [relatifs 01] Quel est l'entier relatif codé en complément à 2 sur un octet par le code 1111 1111 ?

☒ -1

☐ 255

☐ 127

☐ 45

CATALOGUE

Question [hexa 01] Convertir la valeur décimale 195 en hexadécimal.

- ☒ C3
 ☐ A5
 ☐ B9
 ☐ C9

Question [hexa 01]

- ☒
☐
☐
☐

Question [bool01] En logique (algèbre de Boole), l'expression: `not (A or B)` est équivalente à :

- ☒ `(not A)and (not B)`
☐ `(not A)or (not B)`
☐ `A or B`
☐ `A and B`

Question [bool02] Laquelle de ces propriétés est toujours vraie ?

- ☒ `a and (not a)== False`
☐ `a and (not a)== True`
☐ `a and (not a)== not a`
☐ `a and (not a)== a`

Question [bool]

- ☒
☐
☐
☐

Question [str01] On a saisi le code suivant :`mot = 'première'`. Quelle affirmation est vraie ?

- ☒ `mot[7]` vaut 'e'
 ☐ `mot[1]` vaut 'p'
 ☐ `len(mot)` vaut 7
 ☐ `len(mot)` vaut 6

Question [str] Quelle est le résultat de : `'orange'[-3]` ?

- ☒ 'n'
 ☐ 'e'
 ☐ 'g'
 ☐ Error : Negative index

Question [str]

- ☒
☐
☐
☐

Question []

- ☒
☐
☐
☐

Question [TtQue01] On a saisi le code suivant :

```
n = 8.0
while n > 1.0 :
    n = n / 2
```

Quelle est la valeur de `n` après l'exécution du code ?

- ☒ 1.0
 ☐ 4.0
 ☐ 2.0
 ☐ 0.5

Question []

- ☒
☐
☐
☐

Question [for01] On souhaite écrire un programme affichant tous les entiers multiples de 3 entre 6 et 288 inclus. Quel code est correct ?

- ☒ `for n in range(6, 290, 3) :`
 `print(n)`
☐ `for n in range(6, 289) :`
 `print(n / 3)`
- ☐ `for n in range(6, 288, 3) :`
 `print(n)`
☐ `for n in range(6, 289) :`
 `print(3 * n)`

Question [for02] On a saisi le code suivant :

```
a = 12
for i in range(3) :
    a = a * 2
    a = a - 10
```

Quelle est la valeur de `a` après l'exécution du code ?

☒ 26

☐ 18

☐ 18.0

☐ 26.0

Question [for]

☒
☐
☐
☐

Question [fonclist01] Voici une fonction Python de recherche d'un maximum :

```
def maxi(t):
    m = -1
    for k in range(len(t)):
        if t[k] > m:
            m = t[k]
    return m
```

Avec quelle précondition sur la liste `t`, la postcondition “`m` est un élément maximum de la liste `s`” n'est-elle pas assurée ?

☒ Tout élément de `t` est un entier supérieur ou égal à -2.

☐ Tout élément de `t` est un entier positif ou nul.

☐ Tout élément de `t` est un entier supérieur ou égal à 11.

☐ Tout élément de `t` est un entier strictement supérieur à -2.

Question [fonclist02] On dispose d'un tableau d'entiers, ordonné en ordre croissant. On désire connaître le nombre de valeurs distinctes contenues dans ce tableau. Quelle est la fonction qui ne convient pas ?

☒

```
def compte(t):
    cpt = 0
    for i in range(0, len(t)-1):
        cpt = cpt + int(t[i] != t[i+1])
    return cpt
```

☐

```
def compte(t):
    cpt = 1
    for i in range(1, len(t)):
        if t[i] != t[i-1]:
            cpt = cpt + 1
    return cpt
```

☐

```
def compte(t):
    cpt = 1
    for i in range(0, len(t)-1):
        cpt = cpt + int(t[i] != t[i+1])
    return cpt
```

☐

```
def compte(t):
    cpt = 0
    for i in range(0, len(t)-1):
        if t[i] != t[i+1]:
            cpt = cpt + 1
    return cpt+1
```