

## CATALOGUE

**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(b + '_=' + c)`

☐ `print(a + '_=' + d)`

☐ `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 [0pE104]** Que contient la variable a si on exécute ce script ?

```
def diff(val1, val2):
    return val2 - val1
a = diff(3.0, -2.0)
```

☒ -5.0

☐ 5.0

☐ 1.0

☐ -1.0

**Question [0pE105]** Que contient la variable a si on exécute ce script ?

```
def carre(val):
    return val*val
def inc(val):
    return val + 1
a = carre(inc(3.0))
```

☒ 16.0

☐ 9.0

☐ 10.0

☐ 12.0

**Question [0pE106]** Que contiennent les variables a et b si on exécute ce script ?

```
def func(a):
    a += 2.0
    return a
a = 5.0
b = func(a)
```

☒ 5.0 et 7.0

☐ 5.0 et 5.0

☐ 7.0 et 5.0

☐ 7.0 et 7.0

# CATALOGUE

**Question [OpE107]** Que taper en Python pour obtenir  $3^8$  ?

☒ 3\*\*8

☐ 3^8

☐ 3\*8

☐ 3&8

**Question [OpE1]**

☒

☐

☐

☐

**Question [OpAr01]** On exécute l'instruction ci-après. Quel est l'affichage attendu ?

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

☒ 0

☐ 1

☐ 2

☐ 4

**Question [OpAr]**

☒

☐

☐

☐

**Question [fon01]** Avec la fonction donnée ci-dessous l'instruction `mystere(0,1)` retourne :

```
def mystere(a,b):
    reponse=1
    if a==0:
        if b==0:
            reponse=0
    return reponse
```

☒ 1

☐ 0

☐ True

☐ False

**Question [fon]**

☒

☐

☐

☐

**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 [entiers 04]** Quelle est la valeur décimale de l'entier binaire 00011010 ?

☒ 26

☐ 22

☐ 51

☐ 24

**Question [entiers 05]** Donner le résultat de l'addition binaire : 1101 + 1001.

☒ 10110

☐ 01001

☐ 00110

☐ 11010

**Question [entiers 06]** Donner le résultat de l'addition binaire 101101 + 1011.

☒ 111000

☐ 110110

☐ 101000

☐ 111100

**Question [entiers 07]** Donner l'écriture décimale du nombre binaire 10011.

☒ 19

☐ 17

☐ 23

☐ 21

# CATALOGUE

**Question** [entiers 08] Donner l'écriture décimale du nombre binaire 110101.

- ☒ 53
 ☐ 13
 ☐ 47
 ☐ 51

**Question** [entiers 09] Donner l'écriture binaire du nombre 137.

- ☒ 10001001
 ☐ 10111001
 ☐ 10001010
 ☐ 10010001

**Question** [entiers 10] Donner l'écriture binaire du nombre 34.

- ☒ 100010
 ☐ 010010
 ☐ 100001
 ☐ 100110

**Question** [entiers 11] Combien de chiffres binaires sont nécessaires pour coder le nombre 287 ?

- ☒ 9
 ☐ 7
 ☐ 8
 ☐ 10

**Question** [entiers 12] Combien de chiffres possède l'écriture binaire du nombre 75 ?

- ☒ 7
 ☐ 5
 ☐ 6
 ☐ 8

**Question** [entiers 13] 1 octet représente combien de bit(s) ?

- ☒ 8
 ☐ 2
 ☐ 3
 ☐ 6

**Question** [entiers 14] Combien faut-il de bits minimum pour représenter le nombre décimal 16 ?

- ☒ 5
 ☐ 3
 ☐ 4
 ☐ 6

**Question** [entiers 15] Quelle est la valeur décimale de l'entier binaire 00011010 ?

- ☒ 26
 ☐ 32
 ☐ 41
 ☐ 24

**Question** [entiers 16] Avec 5 bits, on peut compter de .... à .... ?

- ☒ 0 à 31
 ☐ 1 à 32
 ☐ 0 à 32
 ☐ 1 à 31

**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

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

- ☒ C3
 ☐ A5
 ☐ B9
 ☐ C9

**Question** [hexa 02] Donner l'écriture hexadécimale du nombre binaire 1001011.

- ☒ 4B
 ☐ 3D
 ☐ 49
 ☐ 5B

**Question** [hexa 03] Donner l'écriture hexadécimale du nombre binaire 110101.

- ☒ 35
 ☐ 6B
 ☐ 65
 ☐ 56

**Question** [hexa 04] Donner l'écriture binaire du nombre hexadécimal 6E.

- ☒ 01101110
 ☐ 01110110
 ☐ 01101101
 ☐ 01110010

**Question** [hexa 05] Donner l'écriture binaire du nombre hexadécimal B5.

- ☒ 10110101
 ☐ 10110111
 ☐ 00110101
 ☐ 10111101

# CATALOGUE

**Question [hexa 06]** Quelle est la représentation binaire du nombre  $5D_{16}$  ?

- ☒ 01011101      ☐ 01101101      ☐ 10101101      ☐ 01011110

**Question [hexa 07]** Quelle est la valeur hexadécimale de l'entier binaire 10110110 ?

- ☒ B6      ☐ C4      ☐ B8      ☐ C6

**Question [hexa 08]**

- ☒      ☐      ☐      ☐

**Question [hexa ]**

- ☒      ☐      ☐      ☐

**Question [hexa ]**

- ☒      ☐      ☐      ☐

**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 [bool03]**

- ☒      ☐      ☐      ☐

**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 [str02]** Quelle est le résultat de : `'orange'[-3]` ?

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

**Question [str03]** Soit le texte suivant : `texte = "Un_chasseur_sachant_chasser_doit_savoir_chasser_sans_son_chien."`. Qu'affiche la ligne suivante : `print(texte[5])` ?

- ☒ "a"      ☐ "h"      ☐ "s"      ☐ "Un\_ch"

**Question [str04]** Soit le texte suivant : `texte = "Un_chasseur_sachant_chasser_doit_savoir_chasser_sans_son_chien."`. Qu'affiche la ligne suivante : `print(texte[:5])` ?

- ☒ "Un\_ch"  
☐ "Un\_cha"  
☐ "Un\_chasseur\_sachant\_chasser\_doit"  
☐ "Un\_chasseur\_sachant\_chasser\_doit\_savoir\_chasser\_sans\_son\_chien."

**Question [str05]** Soit le texte suivant : `texte = "Un_chasseur_sachant_chasser_doit_savoir_chasser_sans_son_chien."`. Qu'affiche la ligne suivante : `print(texte[5:10])` ?

- ☒ "asseu"      ☐ "hasseu"      ☐ "asseur"      ☐ "ar"

## CATALOGUE

**Question [str06]** Soit le texte suivant : `texte = "Un_chasseur_sachant_chasser_doit_savoir_chasser_sans_son_chien."`. Qu'affiche la ligne suivante : `print(texte[:2]+texte[6:8])` ?

- ☒ "Unss"
 ☐ "Un\_ss"
 ☐ "Unsse"
 ☐ "Un\_sse"

**Question [str07]** Soit le texte suivant : `texte = "Un_chasseur_sachant_chasser_doit_savoir_chasser_sans_son_chien."`.

Que faut-il mettre à la place des ... pour afficher "sachant" ?

- ☒ `print(texte[12:19])`
☐ `print(texte[12:18])`
☐ `print(texte[11:18])`
☐ `print(texte[12]+texte[18])`

**Question [str]**

- ☒
☐
☐
☐

**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 [for03] Pour  $i$  allant de 0 à 9 s'écrit :

- ☒ `for i in range(10)`
☐ `for i in range(8)`
☐ `for i in range(9)`
☐ `for i in range(11)`

Question [for04] pour  $i$  allant de 1 à 10 s'écrit :

- ☒ `for i in range(1,11)`
☐ `for i in range(10)`
☐ `for i in range(1,10)`
☐ `for i in range(0,10)`

Question [for]

- ☒
☐
☐
☐

Question [list01] Quelle est le résultat de : `[(a,b) for a in range(3) for b in range(a)]` ?

- ☒ `[(1,0),(2,0),(2,1)]`  
☐ `[(1,0),(2,1),(2,1)]`  
☐ `[(1,0),(2,1),(3,2)]`  
☐ `[(0,0),(1,1),(2,2)]`

Question [list01]

- ☒
☐
☐
☐

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
```

**Question** []



**Question** [LiCo01] Quelle est le résultat de : `[(a,b) for a in range(3) for b in range(a)]` ?



`[(1,0), (2,0), (2,1)]`



`[(1,0), (2,1), (2,1)]`



`[(1,0), (2,1), (3,2)]`



`[(0,0), (1,1), (2,2)]`

**Question** [lico]

