

DEVOIR SURVEILLÉ D'INFORMATIQUE N°1 (CORRIGÉ!)

<b>NOM</b>	
------------	--

Les candidats ne doivent faire usage d'aucun document, l'utilisation de toute calculatrice et de tout matériel électronique est **interdite**

Pour chaque question, une et une seule réponse proposée est correcte. Il n'y a pas de point négatif dans ce QCM. Chaque bonne réponse vous rapportera 0,35 point.

**Vous rendrez cette feuille en entourant clairement au bic noir les réponses choisies.**

**❶ Les nombres et les variables**

1. On tape  $2 + 5 * 3$  dans la console, on obtient :

- A) 17. ✓                      B) 21.                      C) 252525.                      D) Un message d'erreur.

2. On veut utiliser la constante  $\pi$  :

A) On doit d'abord importer la bibliothèque numpy.

B) Il suffit d'écrire `pi` dans la console.

C) On doit d'abord importer la bibliothèque math. ✓

D) Aucune des réponses précédentes.

3. Taper `m.pi` donne une erreur (on a oublié d'importer une bibliothèque), que faut-il taper ?

- A) `from math import`. B) `math import`.                      C) `math import as m`. D) `import math as m`. ✓

4. Une fois la bonne bibliothèque importée, qu'obtient-on en tapant `m.sin(m.pi)` ?

- A) 0.                      C) Une quantité proche de 0. ✓  
B) -1.                      D) Rien.

5. Qu'obtient-on en tapant `type(4/2)` ?

- A) `str`.                      B) `float`. ✓                      C) 2.                      D) `int`.

6. Qu'obtient-on en tapant `11%2` ?

- A) 5.                      B) 1. ✓                      C) 5.5.                      D) 2.

On tape, dans la console, cette suite d'instructions :

```
x, y = 2, 6
w(t)=13
z=x**y
x=w
y+=1
```

7. Combien cette suite d'instructions comporte-t-elle d'erreurs?  
A) 0.                      B) 1.                      C) 2. ✓                      D) 3.
8. À la fin de cette suite d'instructions, que vaut la variable  $y$ ?  
A) 7. ✓                      B) 6.                      C) 2.                      D) 2,6.
9. À la fin de cette suite d'instructions, que vaut la variable  $x$ ?  
A) 6.                      B) 13.                      C) 2,6.                      D) 2. ✓
10. À la fin de cette suite d'instructions, que vaut la variable  $z$ ?  
A) 32.                      B) 64. ✓                      C) 36.                      D) 8.

## ② Les séquences

On tape, dans la console, cette suite d'instructions :

```
ch=4 * "cot! " + "codec!"
L=[1,2,3, [4,5], ch]
ch2=ch
L2=L
L3=L[: ]
ch2="Bonjour"
L[0]=7
L[3][1]=9
```

11. Combien cette suite d'instructions comporte-t-elle d'erreurs?  
A) 0. ✓                      B) 1.                      C) 2.                      D) 3.
12. À la fin de cette suite d'instructions, que vaut  $\text{len}(ch2)$ ?  
A) 22.                      B) 7. ✓                      C) 6.                      D) 10.
13. À la fin de cette suite d'instructions, que vaut  $\text{len}(L)$ ?  
A) 5. ✓                      B) 6.                      C) 27.                      D) 23.
14. À la fin de cette suite d'instructions, que vaut  $ch$ ?

- A) "cot!cot!cot!cot!codec!". ✓  
 B) "Bonjour".  
 C) "4cot!codec!".  
 D) "4cot!+codec!".

15. À la fin de cette suite d'instructions, que vaut L[-1] ?

- A) "!".  
 B) "ch".  
 C) "cot!cot!cot!cot!codec!". ✓  
 D) [4,5].

16. À la fin de cette suite d'instructions, que vaut L[-1][-1] ?

- A) "!". ✓  
 B) "h".  
 C) "cot!cot!cot!cot!codec!".  
 D) Cela donne un message d'erreur.

17. Deux éléments de cette suite qui ont la même adresse à la fin sont :

- A) ch et ch2.  
 B) L et L2. ✓  
 C) L et L3.  
 D) L3 et L2.

18. À la fin de cette suite d'instructions, que vaut L3[0] ?

- A) 1. ✓  
 B) 7.  
 C) 9.  
 D) 2.

19. À la fin de cette suite d'instructions, que vaut L2[0] ?

- A) 1.  
 B) 7. ✓  
 C) 9.  
 D) 2.

20. À la fin de cette suite d'instructions, que vaut L3[3] ?

- A) [4,5].  
 B) [4,9]. ✓  
 C) [9,5].  
 D) 9.

21. À la fin de cette suite d'instructions, que vaut L2[3] ?

- A) [4,5].  
 B) [4,9]. ✓  
 C) [9,5].  
 D) 9.

### ③ Les dictionnaires et les tableaux

On tape, dans la console, cette suite d'instructions :

```
T = np.ones([2,3])
T2=T
T[1,2]=0
T[1,2]==8
T.append(range(2))
T[2,2]=0
Loisirs = {'Capucine' : ['Tennis', 'AS'], 'Mado' : ['Cirque', 'AS']}
D2=Loisirs
D2['Capucine'] = ['Tennis']
D2['Isaure']=['Gym']
D2['Gym']
```

22. Combien cette suite d'instructions comporte-t-elle d'erreurs ?

- A) 0.                      B) 1.                      C) 2.                      D) 3. ✓
23. À la fin de cette suite d'instructions, que vaut `len(Loisirs)` ?
- A) 6.                      B) 5.                      C) 3. ✓                      D) 2.
24. À la fin de cette suite d'instructions, que vaut `T[1,2]` ?
- A) 0. ✓                      B) 1.                      C) 8.                      D) -1.
25. À la fin de cette suite d'instructions, que vaut `T2[1,2]` ?
- A) 0. ✓                      B) 1.                      C) 8.                      D) -1.
26. À la fin de cette suite d'instructions, que vaut `Loisirs['Isaure']` ?
- A) ['Gym']. ✓                      C) 'Mado'.  
B) Rien.                      D) Cela donne un message d'erreur.
27. À la fin de cette suite d'instructions, que vaut `Loisirs('Isaure')` ?
- A) ['Gym'].                      C) 'Mado'.  
B) Rien.                      D) Cela donne un message d'erreur. ✓
28. À la fin de cette suite d'instructions, que vaut `Loisirs['Isaure']][0]` ?
- A) ['Gym'].                      C) '.'.  
B) G.                      D) 'Gym'. ✓
29. À la fin de cette suite d'instructions, que vaut `Loisirs['Isaure']][0][0]` ?
- A) ['Gym'].                      C) '.'.  
B) G. ✓                      D) Cela donne un message d'erreur.
30. Combien d'adresses en tout sont utilisées par `Loisirs`, `D2`, `T` et `T2` ?
- A) 1.                      B) 2. ✓                      C) 3.                      D) 4.
31. De quel type est `Loisirs` ?
- A) C'est un dictionnaire. ✓                      C) C'est un tableau.  
B) C'est une liste.                      D) C'est un tuple.
32. On veut utiliser `T2` dans `D2` :
- A) `T2` peut être une clé et une valeur de `D2`.  
B) `T2` ne peut pas être une clé mais peut être une valeur de `D2`. ✓  
C) `T2` ne peut pas être une valeur mais peut être une clé de `D2`.  
D) `T2` ne peut pas être ni une clé, ni une valeur de `D2`.

#### ④ Fonctions et test

On tape les programmes suivants dans l'éditeur :

```
def g(a, b=2):  
    if a>b:  
        m=a  
    elif a<0:  
        m=b  
    else:  
        m=c  
    return m
```

```
def h():  
    m=input("Donne moi un entier.")  
    n=eval(m)  
    if n>=0:  
        print(n>5)
```

33. On tape `g(3, 1)` dans la console. Ce programme renvoie :

- A) Une phrase.
- B) L'entier 3. ✓
- C) Un message d'erreur.
- D) Rien.

34. On tape `g(-3, -5)` dans la console. Ce programme renvoie :

- A) L'entier -5.
- B) L'entier -5 et l'entier -3.
- C) L'entier -3. ✓
- D) La lettre c.

35. On tape `g(3)` dans la console. Ce programme renvoie :

- A) L'entier 3. ✓
- B) Une phrase.
- C) Un message d'erreur.
- D) Rien.

36. On tape `g(0, 4)` dans la console. Ce programme renvoie :

- A) Le flottant 4.
- B) Un message d'erreur. ✓
- C) Rien.
- D) L'entier 4.

37. La fonction `g` a :

- A) deux valeurs d'entrée, trois valeurs de sortie.
- B) une valeur d'entrée, trois valeurs de sortie.
- C) deux valeurs d'entrée, une valeur de sortie. ✓
- D) une valeur d'entrée, aucune valeur de sortie.

38. La fonction `h` a :

- A) aucune valeur d'entrée, une valeur de sortie.
- B) aucune valeur d'entrée, deux valeurs de sortie.
- C) une valeur d'entrée, une valeur de sortie.
- D) aucune valeur d'entrée, aucune valeur de sortie. ✓

39. De quel type est le `m` qui apparaît dans le programme `h` ?

- A) un entier.
- B) une chaîne de caractère. ✓
- C) une question.
- D) un booléen.

40. De quel type est le  $n$  qui apparaît dans le programme  $h$  ?

- A) une chaîne de caractère.
- B) un flottant.
- C) un booléen.
- D) un entier. ✓

41. De quel type est la sortie du programme  $h$  ?

- A) none, il n'y a pas de sortie. ✓
- B) un entier.
- C) une chaîne de caractère.
- D) un booléen.

42. On tape  $h()$  dans la console puis on répond à la question qu'on nous pose par 6.  $h$  affiche alors :

- A) Le booléen `True`. ✓
- B) Le booléen `False`.
- C) Un message d'erreur.
- D) La phrase "`n>5`".

43. Attention, ce n'est pas la même question que la précédente ! On tape  $h()$  dans la console puis on répond à la question qu'on nous pose par 6.  $h$  retourne alors :

- A) Le booléen `True`.
- B) Rien,  $h$  n'a pas de sortie. ✓
- C) Un message d'erreur.
- D) La phrase "`n>5`".

44. On tape  $h()$  dans la console puis on répond à la question qu'on nous pose par 1.  $h$  affiche alors :

- A) Un message d'erreur.
- B) Le booléen `False`. ✓
- C) Le booléen `True`.
- D) La phrase "`n>5`".

45. On tape  $h()$  dans la console puis on répond à la question qu'on nous pose par  $-1$ .  $h$  affiche alors :

- A) Le booléen `False`.
- B) rien. ✓
- C) Un message d'erreur.
- D) La phrase "`n>5`".

## ⑤ Boucles

On souhaite écrire un programme pour expliciter les termes de la suite  $(u_n)_{n \in \mathbb{N}}$  définie par :

$$u_0 = 2 \text{ et , pour tout entier naturel } n, \text{ on pose : } u_{n+1} = nu_n - u_n^2.$$

Pour cela, on tape les programmes suivants dans l'éditeur :

```
def suite1(n):
    u = 2
    for k in range(n):
        u = k*u - u**2
    return u
```

```
def suite2(n):
    u = 2
    for k in range(n):
        u = n*u - u**2
    return u
```

```
def suite3(n):
    while k < n+1:
        k = k+1
        u = n*u - u**2
    return u
```

```
def suite4(n):
    u = 2
    for k in range(n+1):
        u = (k+1)*u-u**2
    return u
```

```
def suite5(n):
    u = 0
    for k in range(n+1):
        u = n*u-u**2
    return u
```

```
def suite6(n):
    u, k=2, 0
    while k<n+1:
        k, u=k+1, k*u-u**2
    return u
```

46.  $u_3$  vaut :

- A) -440. ✓      B) -120.      C) 2.      D) -194920.

47. suite1(3) renvoie :

- A) 0.      B) -440. ✓      C) 2.      D) -20.

48. suite2(3) renvoie :

- A) 2. ✓      B) -440.      C) -88.      D) -20.

49. suite3(3) renvoie :

- A) Erreur. ✓      B) -20.      C) -194920.      D) 2.

50. suite4(3) renvoie :

- A) -20.      B) -194920.      C) 2.      D) Erreur. ✓

51. suite5(3) renvoie :

- A) -440.      B) 0. ✓      C) 2.      D) -20.

52. suite6(3) renvoie :

- A) -20.      B) Erreur.      C) -440.      D) -194920. ✓

53. suite1(0) renvoie :

- A) -440.      B) Erreur.      C) -20.      D) 2. ✓

54. suite1( $n$ ) avec  $n$  entier naturel renvoie :

- A) Erreur.      B)  $u_{n-1}$ .      C)  $u_n$ . ✓      D)  $u_{n+1}$ .

55. Combien de fonctions vérifient le cahier des charges?

- A) Aucune.      B) Trois.      C) Une. ✓      D) Toutes.

On appelle désormais suite une fonction vérifiant le cahier des charges, c'est-à-dire un programme qui prend en entrée un entier naturel  $n$  et qui renvoie  $u_n$ . On écrit après ce programme :

```

def test(min):
    u = 2
    A = 0
    L = []
    while u > min:
        u = k*u - u**2
        A = A + 1
        L.append(u)
    return A, L

```

56. Ce programme a pour but :

- A) de conjecturer que la limite de  $(u_n)_{n \in \mathbb{N}}$  est  $-\infty$ . ✓
- B) de conjecturer que la limite de  $(u_n)_{n \in \mathbb{N}}$  est  $+\infty$ .
- C) de conjecturer le maximum de  $(u_n)_{n \in \mathbb{N}}$ .
- D) de conjecturer que la limite de  $(u_n)_{n \in \mathbb{N}}$  est 0.

57. Si on pose  $a = \text{test}(-1000)[1]$  alors :

- A)  $a$  est une liste. ✓
- B) on obtient un message d'erreur.
- C)  $a$  est un flottant.
- D)  $a$  est un entier.

58. Si on pose  $a = \text{test}(-4)[0]$  alors :

- A)  $a$  est une liste.
- B)  $a$  est 0.
- C)  $a$  est 1. ✓
- D)  $a$  est 2.

59. Si on pose  $a = \text{test}(-5)[0]$  alors :

- A)  $a$  est une liste.
- B)  $a$  est 0.
- C)  $a$  est 1.
- D)  $a$  est 2. ✓

60. Si on pose  $a = \text{test}(-5)[1]$  alors  $\text{len}(A)$  est :

- A)  $a$  est une liste.
- B)  $a$  est 0.
- C)  $a$  est 1.
- D)  $a$  est 2. ✓