

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

NOM	
------------	--

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 0,4 point.

❶ **Les nombres**

- On tape $a = 131.5$ puis, après avoir totalement importé la bibliothèque `math`, on tape `ceil(a)`. On obtient :
A) une liste. B) 132. ✓ C) 131. D) une chaîne de caractère.
- On tape `n%2` avec n un entier. Cela nous donne :
A) la moitié de n . B) 0 si n est pair. ✓ C) 0 si n est impair. D) un message d'erreur.
- On tape `0.7**2` dans la console, on obtient :
A) une quantité numérique proche de 0,49. ✓ C) (0,49).
B) 0.49 précisément. D) 1.4.
- En tapant `int('9' + '2')`, on obtient :
A) '92' B) Erreur C) 92 ✓ D) 11

❷ **Les variables**

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

```
2=largeur
largeur+=2
largeur, longueur = 2, 4
aire = longueur * largeur
largeur = 12
longueur==5
aire=aire+1
```

5. Cette suite d'instruction donne un certain nombre de messages d'erreurs. Combien ?

- A) 0. B) 1. C) 2. ✓ D) 3.

6. À la fin de cette suite d'instruction, que vaut la variable *longueur* ?

- A) 5. B) 4. ✓ C) 2. D) 12.

7. À la fin de cette suite d'instruction, que vaut la variable *largeur* ?

- A) 5. B) 4. C) 2. D) 12. ✓

8. À la fin de cette suite d'instruction, que vaut la variable *aire* ?

- A) 9. ✓ B) 1. C) 61. D) 8.

③ Chaînes de caractères

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

```
ch = "Bonjour, j'aime les devoirs_d'info"  
CH=4*ch  
CH[0]=ch[-1]
```

9. Cette suite d'instruction donne un certain nombre de messages d'erreurs. Combien ? :

- A) 0. B) 1. ✓ C) 2. D) 3.

10. L'instruction `CH[1] == ch[-1]` donne :

- A) *True*. ✓ C) Un message d'erreur.
B) *False*. D) une modification de CH.

11. L'instruction `len(CH)` donne :

- A) 136. ✓ B) 132. C) 39. D) 4.

12. L'instruction `CH[14:]` donne :

- A) "e les devoirs d'infoBonjour, j'aime les devoirs d'infoBonjour, j'aime les devoirs d'infoBon-
jour, j'aime les devoirs d'info". ✓
B) "me les devoirs d'infoBonjour, j'aime les devoirs d'infoBonjour, j'aime les devoirs d'info-
Bonjour, j'aime les devoirs d'info".
C) "e".
D) "Bonjour, j'aim".

13. L'instruction `CH[: 14]` donne :

- A) "Bonjour, j'aim". ✓
B) "Bonjour, j'aime".
C) "me les devoirs d'infoBonjour, j'aime les devoirs d'infoBonjour, j'aime les devoirs d'info-
Bonjour, j'aime les devoirs d'info".
D) "e".

14. L'instruction `CH[5 : 24 : 2]` donne :

A) une chaîne de caractère vide.

B) 'u, jam e eo'. ✓

C) "r 'ielsdv".

D) "r, j'aime les devo".

15. L'instruction `CH[25 : 4 : -2]` donne :

A) une chaîne de caractère vide.

B) 'roe e maj,u'. ✓

C) "or 'ielsdv".

D) 'u, jam e eo'.

16. L'instruction `CH[5 : 24 : -2]` donne :

A) une chaîne de caractère vide. ✓

B) 'roe e maj,u'.

C) "or 'ielsdv".

D) 'u, jam e eo'.

17. L'instruction "Capucine a 2023-2007 ans" donne :

A) "Capucine a 2023-2007 ans". ✓

B) "Capucine a 16 ans".

C) Une erreur.

D) "Capucinea2023-2007ans".

18. L'instruction "Capucine a " + "3" + "2" + " ans" donne :

A) "Capucine a 5 ans".

B) "Capucine a 32 ans". ✓

C) Une erreur.

D) "Capucinea3+2ans".

19. L'instruction "Capucine a " + `str(2023-2007)` + " ans" donne :

A) "Capucine a 2023-2007 ans".

B) "Capucine a 16 ans". ✓

C) Une erreur.

D) "Capucinea2023-2007ans".

④ Listes

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

```
| Li = 5*['Python', 0, 2]+[10]
```

20. Que donne `len(Li)` ?

A) 16. ✓

B) 41.

C) 5.

D) 20.

21. Que donne `Li(2)` ?

A) 'Python'.

B) 0.

C) 2.

D) Un message d'erreur. ✓

22. Que donne `Li[2]` ?

A) 'Python'.

B) 0.

C) 2. ✓

D) Un message d'erreur.

23. Que donne `Li[-1]=5` ?

- A) 10.
- B) False.
- C) Rien, cela modifie un terme. ✓
- D) Un message d'erreur.

24. Que donne `Li[0][0]`?

- A) 'Python'.
- B) 0.
- C) 'P'. ✓
- D) Un message d'erreur.

25. Que donne `Li[1][1]`?

- A) 'Python'.
- B) 0.
- C) 'P'.
- D) Un message d'erreur. ✓

⑤ Dictionnaires

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

```
| dico = { 'maths' : 'poullaouec', 'info': 'bacquelin' }
```

26. Que donne `len(dico)`?

- A) 0.
- B) 4.
- C) 2. ✓
- D) 28.

27. Quelles sont les clés de `dico`?

- A) 'poullaouec' et 'bacquelin'.
- B) 'maths', 'poullaouec', 'info' et 'bacquelin'.
- C) 'maths' et 'info'. ✓
- D) 'info' et 'bacquelin'.

28. Que donne `dico['maths']`?

- A) 'poullaouec'. ✓
- B) 0.
- C) 'info'.
- D) Un message d'erreur.

29. Que donne `dico['Maths']`?

- A) 'Poullaouec'.
- B) 'poullaouec'.
- C) 'Info'.
- D) Un message d'erreur. ✓

30. Que donne `dico['poullaouec']`?

- A) 'maths'.
- B) 'poullaouec'.
- C) 0.
- D) Un message d'erreur. ✓

31. Que donne `dico.get('Maths', 0)`?

- A) 'Poullaouec'.
- B) 'poullaouec'.
- C) 0. ✓
- D) Un message d'erreur.

32. Que donne `dico['maths'] = 'Poullaouec'`?

- A) 'True'.
- B) Cela ne donne rien, cela modifie un élément. ✓
- C) Cela agrandit le dictionnaire.
- D) Un message d'erreur.

33. Que donne `dico['chimie'] = 'beauvais'`?

- A) 'True'
- B) 'beavais'.
- C) Cela agrandit le dictionnaire. ✓
- D) Un message d'erreur

6 Tableaux

On dit que R est `np.floor(5*np.random.rand(2,3))` (après avoir importé une bonne bibliothèque) et on définit T comme étant le tableau $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix}$.

34. Que donne `np.size(R,0)` ?
- A) 2. ✓
 - B) 3.
 - C) (10,15).
 - D) 10.
35. Que veut dire le `np` dans `np.floor(5*np.random.rand(2,3))` ?
- A) numpy. ✓
 - B) new python.
 - C) no python.
 - D) nupi.
36. Que peut donner `R[0]` ?
- A) 2.44452.
 - B) [2,4].
 - C) [2,4,2]. ✓
 - D) 5.
37. Que donne `T[1,1]` ?
- A) 1.
 - B) [1,2,3,4].
 - C) 6. ✓
 - D) 5.
38. Que donne `T[1,:]` ?
- A) 1.
 - B) [1,2,3,4].
 - C) [5,6,7,8]. ✓
 - D) [[5,6,7,8],[9,10,11,12]].
39. Que donne `T[1:]` ?
- A) 1.
 - B) [1,2,3,4].
 - C) [5,6,7,8].
 - D) [[5,6,7,8],[9,10,11,12]]. ✓

7 Les fonctions

On tape les programmes suivants dans l'éditeur :

```
def abso(x):
    if x>0:
        return(x)
    elif x<0:
        return(-x)
j,i=1,1
```

```
def plus(f, a):
    n= int(input("On ajoute quoi?"))
    global i
    i+=5
    j,b=2,4
    return(f(a)+n>1)
```

40. Combien de variables de sorties a la fonction `abso` ? :

A) Aucune. B) 1. ✓ C) 2 D) 3.

41. On tape `abso(-5)` dans la console. Ce programme renvoie :

A) Une phrase. C) L'entier -5.
B) Un message d'erreur. D) L'entier 5. ✓

42. On tape `abso(0)` dans la console. Ce programme renvoie :

A) une phrase. C) rien. ✓
B) un message d'erreur. D) l'entier 0.

43. On tape `plus(3,2)` dans la console. Ce programme renvoie :

A) une phrase. C) rien.
B) un message d'erreur. ✓ D) un entier.

44. Taper `plus(m.cos,2)` 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`. ✓

45. De quel type est le `n` qui apparaît dans le programme `plus` ?

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

46. La différence entre le nombre de valeurs d'entrée et de sortie du programme `plus` est :

A) 0. B) 1. ✓ C) 2. D) 3.

47. Le programme `plus` renvoie :

A) un entier. B) un texte. C) un flottant. D) un booléen. ✓

48. Si on tape `plus(abso, 9)` et qu'on demande après la valeur de `j`, l'ordinateur répondra :

A) 1. ✓ B) Erreur. C) 2. D) 4.

49. Si on tape `plus(abso, 9)` et qu'on demande après la valeur de `i`, l'ordinateur répondra :

A) 1. B) 6. ✓ C) 2. D) 4.

50. Si on tape `plus(abso, 9)` et qu'on demande après la valeur de `b`, l'ordinateur répondra :

A) 1. B) Erreur. ✓ C) 2. D) 4.

⑧ Mutabilité

On tape ces suites d'instruction dans la console :

```

x=list(range(11))
y=x
z=x[: ]
w=x.copy()
x[1]='Coucou'

```

51. Quelle(s) variable(s) a (ont) la même adresse que x ? :

- A) Aucune B) Toutes C) y ✓ D) z et w

52. À la fin de ces lignes d'instruction, on peut dire que :

- A) y et z ont les mêmes valeurs. C) w et z ont les mêmes valeurs. ✓
B) y et w ont les mêmes valeurs. D) w et x ont les mêmes valeurs.

On tape ces suites d'instruction dans la console :

```

x=[1, [2, 3]]
z=x[: ]
y=x
import copy
w=copy.deepcopy(x)
x[1][1]=5
y[0]=2

```

53. À la fin de ces lignes d'instruction, on peut dire que :

- A) z[0]=2 B) w[0]=2 C) z[0]=1 ✓ D) w[1]=2

54. À la fin de ces lignes d'instruction, on peut dire que :

- A) z est [1, [2, 3]] B) z est [2, [2, 3]] C) z est [1, [2, 5]] ✓ D) z est [1, 5]

55. À la fin de ces lignes d'instruction, on peut dire que :

- A) w est [1, [2, 3]] ✓ B) y est [2, [2, 3]] C) w est [1, [2, 5]] D) y est [5, [2, 3]]

On tape le programmes suivant dans l'éditeur :

```

def fonction(a):
    a[0]='B'
    return a

```

56. Si on écrit, dans la console, ch='bonjour' puis fonction(ch) alors :

A) On obtient un message d'erreur. ✓

B) On obtient False.

C) On obtient 'bonjour'.

D) On obtient 'Bonjour'.

57. Si on écrit, dans la console, Li=['bonjour', 1,2,3] puis fonction(Li) alors :

A) On obtient un message d'erreur.

B) On obtient False.

C) Li est devenu ['B', 1,2,3]. ✓

D) Li reste ['bonjour', 1,2,3].