

DEVOIR SURVEILLÉ D'INFORMATIQUE N°3 (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 0,42 point.

1. Parcours de séquences

On tape le programme suivant dans l'éditeur :

```
def palin(ch):  
    resultat1 = ""  
    resultat2 = ""  
    for c in ch:  
        resultat1 = resultat1 + c  
        resultat2 = c + resultat2  
    print (resultat1)  
    print (resultat2)
```

- On appelle n la somme du nombre de sortie et d'entrée de ce programme. n vaut :
A) 0. B) 1. ✓ C) 2. D) 3.
- Si ch , la donnée en entrée, est une chaîne de caractères alors `resultat1` est :
A) Erreur. B) Une liste. C) Un entier. D) Une chaîne de caractères. ✓
- En tapant `palin(1)` dans la console, on obtient :
A) Erreur. ✓ B) [1]. C) 1. D) 11111.
- En tapant `palin("Bon")` dans la console, on obtient :
A) Une erreur. C) "n" puis "no" puis "noB" puis "Bon"
B) "B" puis "Bo" puis "Bon" puis "noB" ✓ D) "Bon" puis "noB"

5. En tapant `palin([1])` dans la console, on obtient :

- A) Erreur. ✓ B) [1]. C) 1. D) 11111.

6. En tapant `palin(Bon)` dans la console, on obtient :

- A) Une erreur. ✓ C) "n" puis "no" puis "noB" puis "Bon"
B) "B" puis "Bo" puis "Bon" puis "noB" D) "Bon" puis "noB"

On tape le programme suivant dans l'éditeur :

```
def compar(Li1, Li2):  
    n = len(Li1)  
    s=0  
    for x in range(n):  
        if not (Li1[x]==Li2[x]):  
            s+=1  
    return s==0
```

7. Combien d'erreur(s) ce programme comporte-t-il?

- A) 0. ✓ B) 1. C) 2. D) 3.

8. Ce programme renvoie :

- A) Rien. B) Un booléen. ✓ C) Une liste. D) Un entier.

9. Supposons que les données en entrée soient [1, [2, 3], 4] et [5, 6, 7, 8] alors *n* sera :

- A) 3. ✓ B) 4. C) 2. D) une erreur.

10. Si on tape `compar([1, 2, 3, 4], [1, 2, 3, 4, 9])`, on obtient :

- A) True. ✓ B) False. C) 0. D) Erreur.

11. Si on tape `compar([1, 2, 3, 4], [1, 2, 3, 5, 9])`, on obtient :

- A) True. B) False. ✓ C) 0. D) Erreur.

2. Liste et suite

On importe totalement la bibliothèque `numpy` et on tape ce programme :

```
def limite(e):  
    u, n = 1-pi/4, 0  
    while (abs(u)>e):  
        u= 1/(2*n+3)-u  
        n=n+1  
    return n
```

12. Importer totalement la bibliothèque numpy, cela signifie que l'on a écrit :

- A) `from numpy import *` ✓ C) `import numpy as np`
B) `import numpy` D) `import numpy as *`

13. Le programme `limite` a un rapport avec la suite $(u_n)_{n \in \mathbb{N}}$ définie par :

- A) $u_0 = 0$ et $\forall n \in \mathbb{N}, u_{n+1} = \frac{1}{2 * n + 3} - u_n$. C) $u_0 = 1 - \frac{\pi}{4}$ et $\forall n \in \mathbb{N}, u_{n+1} = \frac{1}{2 * n + 3} - u_n$. ✓
B) $u_0 = 1 - \frac{\pi}{4}$ et $\forall n \in \mathbb{N}, u_{n+1} = \frac{1}{2 * k + 3} - u_n$. D) $u_0 = 0$ et $\forall n \in \mathbb{N}, u_n = \frac{1}{2 * n + 3} - u_n$.

14. Dans le programme `limite`, la variable d'entrée `e` est a priori :

- A) une liste. C) un flottant proche de 0. ✓
B) un entier naturel grand. D) un entier pair.

15. En tapant `limite(0.1)` et en notant $(u_n)_{n \in \mathbb{N}}$ la suite ayant un rapport avec ce programme, on obtient :

- A) u_{11} .
B) Le plus petit entier naturel n tel que $-0.1 \leq u_n \leq 0,1$. ✓
C) Le plus grand entier naturel n tel que $-0.1 \leq u_n \leq 0,1$.
D) Le plus petit entier naturel n tel que $u_n > 0,1$.

16. En sachant que `limite(0.01)` renvoie 24, que peut renvoyer `limite(0.0001)` ?

- A) Une erreur. B) 258. ✓ C) 4. D) 23.

On tape le programme suivant dans l'éditeur :

```
def Liste(n):  
    u,m,L=1,1,[]  
    for k in range(n):  
        u= 2*u**2-k  
        L.append(u)  
        if u>m:  
            m=u  
    print(m)  
    return L
```

17. En tapant `Liste(0)`, on obtient :

- A) Rien. B) Une liste vide. ✓ C) [1]. D) 1.

18. Le programme `Liste` a un rapport avec la suite $(u_n)_{n \in \mathbb{N}}$ définie par $u_0 = 1$ et :

- A) Pour tout entier naturel $n, u_{n+1} = 2u_n^2 - k$.
B) Pour tout entier naturel $n, u_{n+2} = 2u_{n+1}^2 - u_n$.
C) Pour tout entier naturel $n, u_{n+1} = 2u_n^2 - n$. ✓
D) Pour tout entier naturel $n, u_n = 2u_{n+1}^2 - (n - 1)$.

$(u_n)_{n \in \mathbb{N}}$ désignera, jusqu'à la fin de cette partie, cette suite.

19. Liste(10) renvoie :

- A) $[u_0, u_1, \dots, u_{10}]$. B) $[u_1, u_2, \dots, u_{10}]$. C) u_{10} . D) u_{11} .

20. Liste(10) affiche :

- A) $[u_0, u_1, \dots, u_{10}]$. C) Le maximum de $\{u_1, u_2, \dots, u_{10}\}$.
B) u_1 puis u_2 puis u_3 jusqu'à u_{10} . D) Le maximum de $\{u_0, u_1, \dots, u_{10}\}$.

On tape enfin ces programmes dans l'éditeur :

```
def suiteredouble(n):  
    u,v,L=1,1,[u,v]  
    for k in range(n-1):  
        u,v=v,v+2*u/(k+2)  
        L.append(v)  
    return L  
  
def suiteredouble2(n):  
    L=[1,1]  
    for k in range(n-1):  
        L.append(L[-1]+2*L[-2]/(k+2))  
    return L  
  
def verif(n):  
    for k in range(n):  
        a=suiteredouble(k)==suiteredouble2(k)  
        if not(a):  
            return k  
    return True
```

21. Si n , la donnée en entrée de `verif`, est un entier naturel alors `verif(n)` peut renvoyer :

- A) deux entiers, B) une liste, C) Erreur, D) un booléen.

22. Dans le programme `verif`, `a` est :

- A) un entier, B) une liste, C) un flottant, D) un booléen.

On note désormais $(u_n)_{n \in \mathbb{N}}$ la suite ayant un rapport avec le programme `suiteredouble` et $(v_n)_{n \in \mathbb{N}}$ la suite ayant un rapport avec le programme `suiteredouble2`.

23. Si `verif(13)` renvoie 5, que peut-on en déduire?

- A) $(u_n)_{n \in \mathbb{N}}$ et $(v_n)_{n \in \mathbb{N}}$ ont 5 termes communs sur l'ensemble $\llbracket 0, 13 \rrbracket$.
B) On peut juste dire que $u_5 = v_5$.
C) On peut juste dire que $u_5 \neq v_5$.
D) On peut dire que $u_5 \neq v_5$ et en déduire d'autre chose.

24. Si `verif(13)` renvoie 13, que peut-on en déduire?

- A) $(u_n)_{n \in \mathbb{N}}$ et $(v_n)_{n \in \mathbb{N}}$ sont confondues sur l'ensemble $\llbracket 0, 12 \rrbracket$.
B) $(u_n)_{n \in \mathbb{N}}$ et $(v_n)_{n \in \mathbb{N}}$ sont confondues sur l'ensemble $\llbracket 0, 13 \rrbracket$.

- C) On peut dire que $u_{13} \neq v_{13}$.
 D) `verif(13)` ne peut pas renvoyer 13. ✓
25. Si `verif(13)` renvoie `True`, que peut-on en déduire?
 A) $(u_n)_{n \in \mathbb{N}}$ et $(v_n)_{n \in \mathbb{N}}$ sont confondues sur l'ensemble $[[0, 12]]$. ✓
 B) $(u_n)_{n \in \mathbb{N}}$ et $(v_n)_{n \in \mathbb{N}}$ sont confondues sur l'ensemble $[[0, 13]]$.
 C) On peut dire que $u_{13} \neq v_{13}$.
 D) `verif(13)` ne peut pas renvoyer `True`.
26. Si `verif(13)` renvoie 5, que renvoie `verif(12)`?
 A) 5. ✓ B) 4. C) `False`. D) `True`.
27. Si `verif(13)` renvoie 5, que renvoie `verif(14)`?
 A) 5. ✓ B) 4. C) `False`. D) `True`.
28. Si `verif(13)` renvoie 5, que renvoie `verif(4)`?
 A) 5. B) 4. C) `False`. D) `True`. ✓
29. $(u_n)_{n \in \mathbb{N}}$ est définie par $u_0 = 1$, $u_1 = 1$ et, pour tout entier naturel n , on a :
 A) $u_{n+1} = u_n + \frac{2u_n}{n+2}$. C) $u_{n+2} = u_{n+1} + \frac{2u_n}{n+2}$. ✓
 B) $u_{n+1} = u_n + \frac{2u_n}{(n-1)+2}$. D) $u_{n+2} = u_n + \frac{2u_{n+1}}{n+2}$.
30. $(v_n)_{n \in \mathbb{N}}$ est définie par $v_0 = 1$, $v_1 = 1$ et, pour tout entier naturel n , on a :
 A) $v_{n+1} = v_n + \frac{2v_n}{n+2}$. C) $v_{n+2} = v_{n+1} + \frac{2v_n}{n+2}$. ✓
 B) $v_{n+1} = v_n + \frac{2v_n}{(n-1)+2}$. D) $v_{n+2} = v_n + \frac{2v_{n+1}}{n+2}$.

3. Les graphiques

On tape le script suivant dans la console :

```
import matplotlib.pyplot as plt
t = np.linspace(0, 10, 100)
y = [a*m.sin(a) ...]
b=np.shape(y)[0]==np.shape(t)[0]
plt.plot(t,y,'--',color='red')
plt.ylim([0,5])
```

31. Combien faut-il encore importer de bibliothèque(s) pour que ce script tourne? :
 A) 1. B) 2. ✓ C) 3. D) 4.

40. Que dire de la quantité esp de ce programme ?

- A) A priori, c'est un entier naturel grand.
- B) A priori, c'est un flottant proche de 0. ✓
- C) $f(esp)$ doit être proche de 0.
- D) A la fin du programme, $b - a$ est strictement supérieur à esp .

41. Si on souhaite savoir le nombre d'itérations effectuées, on va :

- A) mettre $n = n + 1$ dans la boucle while.
- B) mettre $n = 0$ avant la boucle while et $n = 2 * n$ dans la boucle while.
- C) mettre $n = 1$ avant la boucle while et $n = n + 1$ dans la boucle while.
- D) mettre $n = 0$ avant la boucle while et $n = n + 1$ dans la boucle while. ✓

42. Si $di cho(f, 0, 3, 0.1)$ renvoie 1, que peut-on en déduire ?

- A) $f(1)$ doit être proche de 0. ✓
- B) $f(3)$ doit être proche de 0.
- C) $f(0.1)$ doit être proche de 0.
- D) $f(1)$ est compris entre -0.1 et 0.1 .

5. Intégration numérique

On vous propose le programme suivant dans l'objectif de calculer $\int_0^1 \cos(t) dt$:

```
def Rectangles(f, n):  
    R=0  
    for k in range(n):  
        R=R+ xxxxxxxxxxxxxxxxxxxxxxxxxxxx  
    return R/n
```

43. Que faut-il mettre à la place des xxxxxxxxxxxxxxxxxxxxxxxxxxxx ?

- A) $f(k/n)$. ✓
- B) $f(k)$.
- C) k/n .
- D) $1/nf(k/n)$.

On imagine avoir correctement remplacé les xxxxxxxxxxxxxxxxxxxxxxxxxxxx.

44. Quelle exécution est crédible pour obtenir ce qu'on veut ?

- A) `Rectangles()`
- B) `Rectangles(m.cos, 100)`. ✓
- C) `Rectangles(10, m.cos)`.
- D) `Rectangles(m.cos, 0, 1)`.

45. Si on souhaite calculer $\int_1^2 \cos(t) dt$ avec ce programme, on va définir d'abord une fonction g qui prend un réel t et qui renvoie :

- A) $\cos(t + 1)$. ✓
- B) $\cos(t - 1)$.
- C) $\cos(t/2)$.
- D) $\cos(t^2)$.

46. On utilise Python pour transformer une photo en tableau. Si ce tableau est constitué de flottants de $[0,1]$, on peut dire que notre photo était :

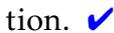
- A) en niveau de gris. B) en couleur. C) en noir et blanc. D) était carrée.



6. Trions!

47. Dans la liste suivante, quel est le tri qui est à votre programme ?

- A) Le tri par sélection. B) Le tri à peigne. C) Le tri rapide. D) Le tri par tas.



On vous propose le programme suivant :

```
def tri(T):  
    for i in range(len(T)-1,0,-1):  
        for j in range(i):  
            if T[j+1]<T[j]:  
                T[j], T[j+1]=T[j+1], T[j]  
    return T
```

48. Ce programme :

- A) trie une liste dans l'ordre croissant. ✓
B) trie une liste dans l'ordre décroissant.
C) ne fonctionne pas, il comporte deux erreurs.
D) ne trie pas totalement la liste, il se contente d'amener le plus petit élément de cette liste à la fin.

49. Que fait-on à l'intérieur des boucles for ?

- A) On échange deux termes consécutifs s'ils ne sont pas dans l'ordre croissant, rien sinon. ✓
B) On échange deux termes consécutifs s'ils ne sont pas dans l'ordre décroissant, rien sinon.
C) On cherche le maximum de la liste d'entrée.
D) On cherche le minimum de la liste d'entrée.

50. La dernière opération effectuée par ce programme consistera à :

- A) ranger dans le bon ordre les deux derniers termes de la liste d'entrée.
B) ranger dans le bon ordre les deux premiers termes de la liste d'entrée. ✓
C) trouver le maximum entre les deux derniers termes de la liste d'entrée.
D) trouver le minimum entre les deux premiers termes de la liste d'entrée.