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 0,5 point.

1. Les boucles

On considère la fonction suivante.

```
def validation(chaine, caractere_valide):
    resultat = True
    for caractere in chaine:
        if caractere != caractere_valide:
            resultat = False
    return resultat
```

1. Que renvoie l'instruction `validation("aaaaaa", "a")` ?

   A) True ✔ B) False C) 6 D) 1

2. Que renvoie l'instruction `validation("abaaaa", "a")` ?

   A) True B) False ✔ C) 5 D) 1

3. Que renvoie l'instruction `validation("abaaaa", "b")` ?

   A) True B) False ✔ C) 1 D) 2

4. Que renvoie l'instruction `validation("ababab", "ab")` ?

   A) True B) False ✔ C) 3 D) 1

On souhaite maintenant écrire une fonction qui prend en entrée une chaîne de caractères et un caractère et qui renvoie l'indice de la première occurrence du caractère dans la chaîne s'il y est présent, -1 sinon. On propose les six fonctions suivantes :

```python
def ipo1(chaine, caractere):
    i = -1
    k = 0
    while k < len(chaine):
        if chaine[k] == caractere:
            i = k
        k = k + 1
    return i

def ipo2(chaine, caractere):
    i = -1
    k = 0
    while i == -1 or k < len(chaine):
        if chaine[k] == caractere:
            i = k
        k = k + 1
    return i

def ipo3(chaine, caractere):
    i = -1
    k = 0
    while i == -1 or k >= len(chaine):
        if chaine[k] == caractere:
            i = k
        k = k + 1
    return i

def ipo4(chaine, caractere):
    i = -1
    k = 0
    while i == -1 and k < len(chaine):
        if chaine[k] == caractere:
            i = k
            k = k + 1
    return i

def ipo5(chaine, caractere):
    i = -1
    k = 0
    while i == -1 and k < len(chaine):
        if chaine[k] == caractere:
            i = k
        k = k + 1
    return i

def ipo6(chaine, caractere):
    i = -1
    k = 0
    while i == -1 and k < len(chaine):
        if chaine[k] == caractere:
            i = k
        i = i + 1
    return i
```

5. Que renvoie `ipo1("gfdabaaagaz", "g")` ?

   A) 0. B) 1. C) 7. D) 8. ✔

6. Que renvoie `ipo2("gfdabaaagaz", "g")` ?

   A) 0. B) 1. C) Erreur. D) 8. ✔

7. Que renvoie `ipo2("gfdabaaagaz", "h")` ?

   A) 0. B) 1. C) Erreur. ✔ D) 8.

8. Que renvoie `ipo3("gfdabaaagaz", "h")` ?

   A) 0. B) 1. C) Erreur. ✔ D) 8.

9. Que renvoie `ipo3("gfdabaaagaz", "g")` ?

   A) 0. ✔ B) 1. C) 7. D) 8.

10. Que renvoie `ipo6("gfdabaaaaz", "t")`?

A) 0. ✔ B) 1. C) 7. D) −1.

11. Combien de fonctions vérifient le cahier des charges ?

A) Aucune. B) Une. ✔ C) Trois. D) Toutes.

12. Combien de fonctions peuvent boucler à l'infini ?

A) Aucune. B) Une. ✔ C) Trois. D) Toutes.

2. Liste et suite

On tape le programme suivant dans l'éditeur :

```
def Liste(n):
    v,u,m,L=1,1,1,[]
    for k in range(n):
        v=3*v-2
        u= 2*u**2-k
        L.append(u)
        if u>m:
            m=u
    print(m)
    return L
```

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

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

14. En tapant A=`Liste(3)`, A est :

A) [2,7]. B) [1,1,1]. C) [2,7,96]. ✔ D) [−1,−1,−1].

15. `Range(5)` est la liste :

A) [0,1,2,3,4]. ✔ B) [1,2,3,4,5]. C) [0,1,2,3,4,5]. D) [5,4,3,2,1,0].

16. Le programme `Liste` a un rapport avec la suite $(v_n)_{n\in\mathbb{N}}$ définie par $v_0 = 1$ et :
   A) Pour tout entier naturel n, $v_n = 3v_n - 2$.
   B) Pour tout entier naturel n, $v_{n+1} = 3(v_n - 2)$.
   C) Pour tout entier naturel n, $v_{n+1} = 3v_n - 2$. ✔
   D) Pour tout entier naturel n, $v_{n+1} = 3v_n - 2n$.

17. 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 pour les trois prochaines questions cette suite.**

18. `Liste(10)` **renvoie** :

A) $[u_0, u_1, \cdots, u_{10}]$.
B) $[u_1, u_2, \cdots, u_{10}]$. ✔
C) u_{10}.
D) Le maximum de $\{u_0, u_1, \cdots, u_{10}\}$.

19. `Liste(10)` **affiche** :

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

20. On voudrait obtenir que les rangs impairs de cette suite. Pour cela, il suffit de :

A) taper `Liste(n)` avec n un entier naturel impair.
B) remplacer $u = 2*u**2-k$ par $u = 2*u**2-(2*k+1)$
C) remplacer `Range(n)` par `Range(1, n, 2)`.
D) rajouter une condition avant le L.append(u). ✔

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=len(y)==len(t)
plt.plot(t,y,'--',color='red')
plt.ylim([0,5])
```

21. Pour que ce script tourne, il faut encore importer :

A) Une bibliothèque.
B) Deux bibliothèques. ✔
C) Trois bibliothèques.
D) Quatre bibliothèques.

22. Il faut terminer la définition de y. Vous proposez de remplacer !!!!!!!!!!!!!! par :

A) `for a in range(t)`
B) `for t in t`
C) `while a < t`
D) `for a in t` ✔

On imagine désormais, pour la suite des questions, que le programme tourne bien.

23. Ce script trace le graphe de f avec :

A) $f : x \mapsto x\sin(x)$. ✔
B) $f = \sin$.
C) $f : x \mapsto a*m\sin(a)$.
D) $f : x \mapsto \sin(x^2)$.

24. Dans ce script, b est :

A) `True`. ✔
B) `False`.
C) 100.
D) 99.

25. On va voir le tracé de cette fonction sur :

A) $\mathbb{R}$. B) $[0,10]$. ✔ C) $[0,5]$. D) $[0,100]$.

26. Le tracé sera :

A) rouge en continue.
B) rouge en pointillé. ✔
C) noir en continue.
D) noir en pointillé.

27. Pour ce tracé, le nombre de points utilisés est :

A) 10. B) 100. ✔ C) 99. D) $\frac{1}{10}$.

4. Dichotomie

On vous propose ce programme (que l'on va compléter !) :

```
def f(x):
    return x**2-3

def Ite(f, N):
    a=0
    b=5
    u=(a+b)/2
    n=0
    eps=10**(-N)
    while ???????:
        u=(a+b)/2
        if f(u)*f(a) <=0:
            ++++++++++++++
        else:
            !!!!!!!!!!!!!!!
        n=n+1
    return n
```

28. Ce programme a un rapport avec :

A) l'algorithme de Lagrange.
B) Le tri par sélection.
C) L'algorithme de Dichotomie. ✔
D) La méthode des rectangles.

29. Avec ce programme, on souhaite trouver une valeur approchée de :

A) π. B) $\sqrt{3}$. ✔ C) $-\sqrt{2}$. D) $\ln(2)$.

30. Par quoi, afin que le programme soit intéressant, peut-on remplacer ???????

A) $u > eps$ B) $b-a < eps$ C) $b-a > eps$ ✔ D) $a+b < eps$

31. Le programme renvoie :
   A) le nombre d'itération dont on va avoir besoin pour approximer une bonne quantité. ✔
   B) une solution d'une équation.
   C) le nombre de solution d'une équation.
   D) une valeur approchée d'une solution d'une équation.
32. Par quoi, afin que le programme soit intéressant, peut-on remplacer ++++++++++++++ ?
   A) b=u. ✔ B) a=u. C) u=b. D) u=(a+b)/2.
33. Par quoi, afin que le programme soit intéressant, peut-on remplacer !!!!!!!!!!!!!!!?
   A) a=u. ✔ B) u=a. C) a=b. D) a=f(u).

5. Tri

On vous propose le programme suivant qui prend en entrée une liste de nombres :

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

34. 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.
35. Quel tri de votre programme utilise une liste intermédiaires des occurences de la liste à trier ?
   A) Le tri par sélection.
   B) Le tri par comptage. ✔
   C) Le tri par insertion.
   D) Le tri à bulles.
36. `range(8,0,-1)` est :
   A) une erreur. B) [0, 1, 2, 3, 4, 5, 6, 7]. C) [8, 7, 6, 5, 4, 3, 2, 1, 0]. D) [8, 7, 6, 5, 4, 3, 2, 1]. ✔
37. 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.

38. Si on tape `tri([9,3,2,7,14,1,3,2])` sur la console, quelle est la première liste que l'on va voir apparaître ?

    A) [3,2,7,9,1,3,2,14] ✔B) [9,3,7,14,2,3,2,1] C) [2,3,7,1,3,2,9,14] D) [3,2,2,7,1,1,2,2]

39. Si on tape `tri([9,3,2,7,14,1,3,2])` sur la console, quelle est la deuxième liste que l'on va voir apparaître ?

    A) [9,7,14,3,3,2,2,1] B) [2,2,2,1,1,1,2,2] C) [2,3,7,1,3,2,9,14] ✔D) [3,2,2,7,1,1,2,2]

40. Si on tape `tri([9,3,2,7,14,1,3,2])` sur la console, quelle est la troisième liste que l'on va voir apparaître ?

    A) [2,3,1,3,2,7,9,14] ✔B) [1,2,2,3,3,7,9,14] C) [2,1,3,2,3,7,9,14] D) [2,1,1,1,1,1,2,2]

41. 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.
42. Le 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.

6. Intégration numérique

On supose les bibliothèques importées. On vous propose ce programme (que l'on va compléter !) :

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

def f(t):
    return m.pi/2*  m.cos (m.pi*t/2)

def Err(n,f):
    return np.abs(rectangles(n,f)- 1)

def trace(f,n):
    abs=np.linspace(1, n, n)
    y=[ Err(a,f) for a in abs]
    plt.plot(abs, y)
    plt.show()
```

43. On voudrait que la fonction `Rectangles` ait un rapport avec la méthode des rectangles. Par quoi remplacer!!! ?

A) $f(k)$.

B) $f(n)$.

C) $f\left(\frac{k}{n}\right)$. ✔

D) $f\left(\frac{1}{n}\right)$.

44. Avec cette fonction, on souhaite trouver une valeur approchée de :

A) π.

B) $\sin\left(\frac{\pi}{4}\right)$.

C) l'intégrale de f sur $[0,1]$. ✔

D) l'intégrale de f sur $[0,\pi]$.

45. `trace` renvoie (et pas affiche !) :

A) un flottant.

B) une liste.

C) un graphique.

D) rien ! ✔

46. Que représente le 1 qui se trouve dans le `return` de la fonction `Err` ?

A) un nombre de rectangle.

B) la qualité de l'approximation.

C) l'aire du premier rectangle.

D) $\sin\left(\frac{\pi}{2}\right)-\sin(0)$. ✔

47. Le tracé que l'on obtient grâce à `trace` est a priori le tracé d'une fonction :

A) décroissante. ✔

B) croissante.

C) intégrable.

D) On ne s'attend à rien de précis.

48. Le n qui se trouve à l'entrée de la fonction `Rectangles` est a priori :

A) un entier naturel grand. ✔

B) un réel positif proche de 0.

C) une fonction croissante.

D) une fonction continue sur $[0,1]$.

49. Quel est votre professeur de sciences préféré ?

A) Mme Ghanthous. ✔

B) M Poullaouec. ✔

C) Mme Beauvais. ✔

D) M Bacquelin. ✔