

Spécialité NSI Terminale

Contrôle n° 3

Sujet A

Mardi 5 décembre 2023

Le barème est indicatif. Tous les programmes sont à écrire en Python.

Rendre le sujet avec la copie.

A. Variables locales et globales, effets de bord (6 points)

1. On exécute le programme suivant :

```
def maFct1(x, y) :  
    x = x + y  
    y = x*y  
    print("x :", x, "y :", y)  
    return y - x  
  
R = 10  
S = 15  
T = maFct1(S, R)  
print("R :", R, "S :", S, "T :", T)
```

Que s'affiche-t-il dans la console ?

2. On exécute le programme suivant :

```
def maFct2(A) :  
    A = 2*A  
    print("(1) A :", A)  
  
A = 35  
maFct2(A)  
print("(2) A :", A)
```

Que s'affiche-t-il dans la console ?

3. On exécute le programme suivant :

```
def maFct3(x, y) :  
    global T, V  
    T = x + y  
    V = x*y  
    print("x :", x, "y :", y)  
    return V - T  
  
T = 40  
V = 10  
W = maFct3(T, V)  
print("T :", T, "V :", V, "W :", W)
```

Que s'affiche-t-il dans la console ?

4. a. On considère le programme suivant :

```
tab1 = [11, 13, 17, 19]  
tab2 = tab1  
tab2[3] = 23
```

Quelles sont les valeurs des variables `tab1` et `tab2` après exécution de ce programme ?

b. On considère le programme suivant :

```
tab3 = [9, 16, 25, 36]
tab4 = tab3.copy()
tab4.append(49)
```

Quelles sont les valeurs des variables `tab3` et `tab4` après exécution de ce programme ?

5. On exécute le programme suivant :

```
def decupler(tab) :
    lg = len(tab)
    for i in range(lg) :
        tab[i] = 10*tab[i]
    return tab

T1 = [3, 5, 7, 9]
T2 = decupler(T1)
print("T1 :", T1)
print("T2 :", T2)
```

a. Que s'affiche-t-il dans la console ?

b. Modifier le code de la fonction `decupler` sans changer le programme principal de telle façon que l'exécution du programme donne l'affichage suivant :

```
T1 : [3, 5, 7, 9]
T2 : [30, 50, 70, 90]
```

B. Récursivité (7 points)

6. On considère la fonction `somme` suivante :

```
def somme(n) :
    if n == 0 :
        return 0
    return n + somme(n-1)
```

a. Pourquoi peut-on dire qu'il s'agit d'une fonction récursive ?

b. Comment appelle-t-on le cas où l'argument `n` est égal à 0 ?

c. Que renvoient les appels de fonctions suivants :

`somme(1)` `somme(2)` `somme(5)`

d. Que se passe-t-il si on exécute l'instruction suivante : `res = somme(-2)` ? Justifiez votre réponse.

7. a. Compléter le code de la fonction récursive `palindrome(texte)` qui renvoie `True` si la chaîne de caractères `texte` est un palindrome et `False` sinon. On rappelle qu'un palindrome est un texte qui reste identique qu'on le lise de gauche à droite ou de droite à gauche.

```
def palindrome(texte) :
    lg = len(texte)
    if ... :
        return True
    texte_central = texte[1 : lg-1]
    return ... == texte[lg-1] and palindrome(...)
```

b. Écrire un programme principal qui teste cette fonction sur les chaînes de caractères `"kayak"` et `"1589751"`.

8. a. La suite de Syracuse (s_n) est une suite d'entiers naturels définie de la façon suivante :

- * on choisit un entier naturel quelconque pour la valeur du premier terme s_0 ;
- * on définit par récurrence ensuite la valeur des termes suivants de la façon suivante :
 - Si s_n est pair alors $s_{n+1} = \frac{s_n}{2}$
 - Si s_n est impair alors $s_{n+1} = 3s_n + 1$

Recopier et compléter le code de la fonction récursive `syracuse(n, s0)` qui renvoie la valeur du terme de rang n si le premier terme prend la valeur s_0 .

```
def syracuse(n, s0) :  
    if n == ... :  
        return ...  
    s = syracuse(..., ...)  
    if s%2 == 0 :  
        return ...//2  
    else :  
        return 3*... + 1
```

b. Écrire le programme principal qui affiche les valeurs des 10 premiers termes de la suite de Syracuse lorsque le terme initial est égal à 7.

C. Algorithmes de tri (6 points) (

9. On a écrit le code la fonctions `echanger` de la façon suivante :

```
def echanger(tab, a, b) :  
    tab[a], tab[b] = tab[b], tab[a]
```

On dispose aussi de la fonction `minTab(tab)` qui renvoie l'indice où se trouve la plus petite valeur du tableau de nombres `tab`.

a. A l'aide de ces deux fonctions, compléter le code de la fonction `tri_selection(tab)` qui effectue le tri par ordre croissant et en place d'un tableau de nombres par l'algorithme de tri pas sélection.

```
def tri_selection(tab) :  
    lg =len(tab)  
    for i in range(...) :  
        k = ...  
        ...
```

b. En s'inspirant du code de la fonction `tri_selection` et en utilisant la méthode `copy()` de la classe Python `list`, écrire une fonction `tri_selection2(tab)` qui renvoie un tableau trié par ordre croissant sans modifier le tableau `tab` passé en argument.

c. Écrire un programme principal qui testent les fonctions `tri_selection` et `tri_selection2` sur le tableau `[8, 6, 2, -7, 21]`.

10. On a effectué en 3 millisecondes le tri d'un tableau de 1000 nombres avec une fonction mettant en œuvre un algorithme de tri par sélection. Combien de temps environ devrait prendre le tri d'un tableau de 5000 nombres sur la même machine et avec la même fonction ?

11. a. On suppose que l'on dispose des fonctions `decouper` et `fusion`. Recopier et compléter le code de la fonction récursive `tri_fusion(tab)` qui *renvoie* un tableau trié par ordre croissant (sans modifier le tableau de nombres `tab`) en suivant une programmation récursive et l'algorithme de tri par fusion.

```
def tri_fusion(tab) :  
    lg = len(tab)  
    if ... :  
        return ...  
    tab1, tab2 = decouper(tab)  
    tab1 = ...  
    tab2 = ...  
    return fusion(tab1, tab2)
```

b. Expliquer pourquoi ce programme met en œuvre une stratégie de "diviser pour régner".