

DEVOIR SURVEILLÉ D'INFORMATIQUE N°2 (CORRIGÉ!)

Les candidats ne doivent faire usage d'aucun document, l'utilisation de toute calculatrice et de tout matériel électronique est **interdite**. Dans l'écriture de vos programmes en Python, **respectez la ponctuation et l'indentation**.

Exercice 1 — Un petit pot pourri... (Aucune cohérence dans cet exercice, ce sont juste des extraits de vos TP!)



1. Créer un programme prenant en entrée un entier naturel n et renvoyant n^2 si n est pair, $2n$ sinon.

Réponse : On propose ce programme suivant :

```
def question1(n):  
    if n%2==0:  
        return n**2  
    else:  
        return 2*n
```

2. Écrire une fonction prenant en entrée trois réels a , b et c qui renvoie True si au moins l'un des trois nombres a , b et c est nul, False sinon.

Réponse : On propose ce programme suivant :

```
def aumoinsnuls(a,b,c):  
    if a==0 or b==0 or c==0:  
        return True  
    else:  
        return False
```

3. On dit qu'une année est bissextile si elle est soit divisible par 4 mais non divisible par 100, soit divisible par 400. Écrire un programme prenant en entrée N un entier naturel et permettant de savoir si l'année N est bissextile ou non.

Réponse : On propose ce programme suivant :

```
def bissextile(N):  
    if (N%4==0 and not N%100==0) or N%400==0:  
        return "Ann e bissextile"  
    return "Ann e pas bissextile"
```

4. Écrire une fonction prenant en paramètre n un entier naturel non nul et calculant le produit des n premiers entiers naturels non nuls, c'est-à-dire $1 \times 2 \times 3 \times \dots \times n$.

Réponse : On propose ce programme suivant :

```
def produit(n):
    a=1
    for k in range(1,n+1):
        a=a*k
    return a
```

5. Vous voulez courir un marathon, c'est-à-dire une course de 52 km. Sans entraînement, vous êtes capable de courir 10 kilomètres de suite. Vous vous entraînez quotidiennement et votre distance maximale parcourue augmente de 3% chaque jour. De combien de jours d'entraînement avez-vous besoin?

Réponse : On propose ce programme suivant :

```
def nbjours():
    n=0
    j=0
    u=10
    while u<52:
        u=u+u*0.03
        n=n+1
    return "Le nombre de jour d'entrainement est ",n
```

6. Écrire une fonction prenant en paramètre n un entier naturel non nul et calculant $1 + 3 + 5 + 7 + \dots + (2n - 1)$.

Réponse : On propose ce programme suivant :

```
def exo1c(n):
    s=0
    for k in range(1, 2*n,2):
        s=s+k
    return s
```

7. L'algorithme de Syracuse consiste à prendre un nombre entier n et lui associer $\frac{n}{2}$ si n est pair, $3n + 1$ sinon. Écrire une fonction Leun qui prend en entrée un entier naturel non nul n et qui effectue l'algorithme de Syracuse à partir de n jusqu'à obtenir 1 et affichant tous les nombres obtenus jusqu'au premier 1.

Réponse : On propose ce programme suivant :

```
def Leun(n):
    while n!=1:
        if n%2==0:
            n=n/2
        else:
            n=3*n+1
    print(n)
```

Exercice 2 — On remplit les bulletins

Remplir les bulletins a toujours fatigué Mme B. Apprenant par le plus grand des hasard le langage Python, elle se rend compte qu'elle peut automatiser sa saisie. Nous allons l'aider.

1. Cette fonction prend en entrée une liste de nombres. Dire, sans justifier, le nom de cette fonction, le nombre d'entrée, de sortie et ce qu'elle fait concrètement :

```
def Mystere(li):  
    m = li[0],  
    for el in li:  
        if el > m:  
            m = el  
    return m
```

Réponse : Cette fonction s'appelle Mystere, elle a une entrée, une sortie et elle calcule le maximum de la liste de nombres donnée en entrée.

2. Écrire une fonction moyenne qui prend comme argument une liste de notes et qui retourne la moyenne des éléments de la liste.

Réponse : On propose le programme suivant :

```
def moyenne(L):  
    m=0  
    for k in L:  
        m+=k  
    return m/len(L)
```

3. Écrire une fonction extrema qui prend comme argument une liste de notes et qui renvoie une liste de deux nombres : le premier est l'élément minimal de la liste de notes et le deuxième est son élément maximal. Si la liste d'entrée est [12, 5, 3, 4, 19, 2, 3], cette fonction renverra [2, 19].

Réponse : On propose le programme suivant :

```
def extrema(L):  
    m=M=L[0]  
    for k in L:  
        if k>M:  
            M=k  
        elif k<m:  
            m=k  
    return [m,M]
```

4. Écrire une fonction croissante qui prend pour argument une liste de nombres et qui retourne True si les éléments de la liste sont dans l'ordre strictement croissant et False dans le cas contraire. On imagine écrire de la même façon une fonction décroissante qui prend pour argument une liste de nombres et qui retourne True si les éléments de la liste sont dans l'ordre décroissant et False dans le cas contraire.

Réponse : On propose les programme suivants :

```
def croissante(Li) :  
    b = True  
    k = 0  
    while k<len(Li) - 1 and b == True :  
        if Li[k]>= Li[k+1] :  
            b = False  
        k = k + 1  
    return b  
  
def decroissante(Li) :  
    b = True
```

```

k = 0
while k < len(Li) - 1 and b == True :
    if Li[k] <= Li[k+1] :
        b = False
    k = k + 1
return b

```

5. Cette fonction `cumuls` prend en entrée une liste de nombres. Dire, sans justifier, ce que cette fonction renverra si la liste d'entrée est `[12,5,3,4,19,2,3]` :

```

def cumuls(L):
    Lnouveau=L[:]
    for k in range(1, len(L)):
        Lnouveau[k]=Lnouveau[k-1]+L[k]
    return Lnouveau

```

Réponse : Si la liste d'entrée est `[12,5,3,4,19,2,3]`, `cumuls` renverra la liste `[12,17,20,24,43,45,48]`.

6. Écrire une fonction `evolution` qui prend comme argument une liste de notes et qui renvoie une liste dont le k -ème élément est la k -ème moyenne partielle, c'est-à-dire la moyenne des k premières notes. Si la liste d'entrée est `[12,5,4,3,19,5,8]`, cette fonction renverra `[12,8.5,7,6,8.6,8,8]`.

Réponse : On propose le programme suivant :

```

def evolution(Li):
    Lnouveau= cumuls(L)
    for k in range(1, len(L)):
        Lnouveau[k]=Lnouveau[k]/(k+1)
    return Lnouveau

```

7. Écrire enfin une fonction `appreciationMmeB` qui prend pour argument une liste de notes et qui, en utilisant les fonctions précédentes, retourne une chaîne de caractères constituée ainsi :

- On débute la chaîne par "Excellent trimestre. " si la moyenne est entre 15 et 20 ; par "Bon trimestre. " si la moyenne est entre 10 et 15 ; par "Trimestre décevant. " si la moyenne est entre 5 et 10 et par "Trimestre insuffisant. " si la moyenne est entre 0 et 5.
- On poursuit la chaîne par "Résultats réguliers. " si l'écart entre la note maximale et minimale est inférieur ou égal à 2 ; par "Résultats irréguliers. " si l'écart entre la note maximale et minimale est supérieur ou égal à 7 et par la chaîne vide " " dans les autres cas.
- On termine la chaîne par "En progrès." si la liste des moyennes partielles est strictement croissante ; "En baisse." si la liste des moyennes partielles est strictement décroissante et par la chaîne vide " " dans les autres cas.

Réponse : On propose le programme suivant :

```

def appreciation(Li) :
    m = moyenne(Li)
    if m <= 5 :
        app = "Trimestre insuffisant. "
    elif 5 < m <= 10 :
        app = "Trimestre d cevant. "
    elif 10 < m <= 15 :
        app = "Bon trimestre. "
    else :
        app = "Excellent trimestre. "

```

```

minmax = extrema(Li)
etendue = minmax[1] - minmax[0]
if etendue <= 2 :
    app = app + "R sultats r guliers. "
elif etendue >= 7 :
    app = app + "R sultats irr guliers. "
evol = evolution(Li)
if croissante(evol) == True :
    app = app + "En progr s."
elif decroissante(evol) == True :
    app = app + "En baisse."
return app

```

