

Contrôle de bases de la programmation :

CORRIGÉ

Vendredi 29 octobre 2021

Responsable : S. Letellier

Durée : 1h

Documents autorisés : 1 demi-feuille A4 bleu ciel (distribuée en TD), recto-verso,
manuscrite ou imprimée

Calculette : non autorisée

A remplir par l'étudiant :

NOM :

Prénom :

Groupe :

Consignes et remarques :

Les programmes doivent être écrits en **langage Python**.

Seule la syntaxe étudiée en cours est autorisée.

Attention à **bien** indenter les programmes (c'est-à-dire à marquer **nettement** les niveaux d'imbrication). Vous commencerez à écrire vos programmes en suivant la ligne tracée dans l'espace réservé (exercices 6, 7 et 8).

Ne pas utiliser de liste (ou de tableau) dans les programmes à écrire.

Le barème est indicatif. Il pourra être légèrement modifié.

Le sujet comprend 8 exercices (le dernier exercice est le plus difficile).

Les exercices 1 à 5 devraient être traités rapidement.

Exercice n°1 (1 point) :

Soient les deux programmes suivants et soit la variable entière *borne* dont la valeur est entrée par l'utilisateur.

Programme 1	Programme 2
<pre>x = 0 i = 1 borne = int(input("borne ?")) while i <= borne : x = x + i i = i + 1 print(x)</pre>	<pre>x = 0 i = 1 borne = int(input("borne ?")) while i <= borne and x < 10 : x = x + i i = i + 1 print(x)</pre>

Quelle est la valeur numérique qui est affichée par chacun des programmes quand borne vaut 5 ? Et quelle est la valeur de *i* à la fin de l'exécution de chaque programme quand borne vaut 5 ?

	Programme 1	Programme 2
Valeur numérique affichée	15	10
Valeur de <i>i</i> à la fin de l'exécution du programme	6	5

Exercice n°2 (1 point) :

Soit le programme suivant :

```
n = input("n ? ")
resu = n ** n
print("Résultat = " + resu)
```

Ce programme ne fonctionne pas. Apporter la ou les modifications nécessaires pour qu'il n'y ait plus d'erreur à l'exécution.

```
n = int(input("n ? "))      # ou bien float(...)
resu = n ** n
print("Résultat = " + str(resu))
```

Exercice n°3 (2 points) :

Soit le programme suivant :

```
from random import randint
n = randint(0,20)

while n % 4 != 0 and (n <= 10 or n % 3 != 0) :
    n = randint(0,20)
print (n)
```

Ecrire la condition du *while* : on doit arrêter les tirages quand on a obtenu soit un multiple de 4 soit un nombre strictement supérieur à 10 qui est un multiple de 3. Ne pas utiliser l'opérateur *not*. Utiliser l'opérateur *%* pour tester si un nombre est un multiple d'un autre nombre.

Exercice n°4 (2 points) :

Soit la liste liVal = [8.8, 1995, [1,2,3], "vacances", 2005] et la chaîne de caractères ch = "ctl".

Précisez ce qui est affiché par chacun des *print* suivants :

print (liVal [1])	1995
print (liVal [-1])	2005
print(liVal [2:4])	[[1, 2, 3], 'vacances']
print(liVal [4:])	[2005]
for i in liVal [2]: print(i*i)	1 4 9
val = liVal [3] print(val[1:3])	ac
print(liVal [3][1])	a
print(liVal [3][10])	erreur
print(ch[1] * 3)	ttt
print(ch.append("s"))	erreur

Exercice n°5 (2 points) :

Quelles sont les valeurs affichées par le programme suivant (attention à bien respecter les retours à la ligne et les espaces).

```
for i in range (1,4):
    print(i, " : ", end = "")
    for j in range(i, 4*i+1, i):
        print(j, " ", end = "")
    print()
print(i, j)
```

**1 : 1 2 3 4
2 : 2 4 6 8
3 : 3 6 9 12
3 12**

Exercice n°6 (3,5 points) :

- Ecrire un programme (voir plus bas) qui demande n réels à l'utilisateur puis affiche la moyenne des réels > 0 et la moyenne des réels < 0 . La valeur de n est saisie par l'utilisateur et on la suppose > 0 sans avoir à le vérifier. Il est demandé d'utiliser la boucle while (pas la boucle *for*). L'utilisation de *if* est bien sûr autorisée. Chacune des deux moyennes doit être accompagnée d'un libellé parlant à l'affichage. Ne pas utiliser de liste (pour stocker les réels par exemple). **Attention**, l'utilisateur peut n'entrer que des réels > 0 ou que des réels < 0 !!

Exemple :

Si l'utilisateur entre **6** pour n puis entre les nombres **3.1, -4.2, -5.3, 6.4, 0.0** et **7.8** alors le programme doit afficher :

Moyenne des réels $> 0 : 5.76$ (qui est la valeur de $((3.1+6.4+7.8)/3)$)

Moyenne des réels $< 0 : -4.75$ (qui est la valeur de $((-4.2+-5.3)/2)$)

- Ajouter des instructions pour déterminer et afficher, en plus, la valeur du plus grand nombre > 0 saisi.

Exemple : Si l'utilisateur entre **3** pour n puis entre **-9.1, 3.8** et **8.8** alors il faut afficher : **Plus grand nombre $> 0 : 8.8$**

Programme :

```
n = int(input("n ? "))
somPos = 0
cptPos = 0
somNeg = 0
cptNeg = 0
i = 1
max = 0
while i <= n:
    nb = float(input("Entrez un réel ? "))
    if nb > 0:
        somPos = somPos + nb
        cptPos = cptPos + 1
        if max < nb:
            max = nb
    elif nb < 0:
        somNeg = somNeg + nb
        cptNeg = cptNeg + 1
    i = i + 1
if cptPos != 0:
    print("Moyenne des réels > 0 : ", somPos/cptPos)
else:
    print("Moyenne des réels > 0 : aucun réel > 0 entré ")
if cptNeg != 0:
    print("Moyenne des réels < 0 : ", somNeg/cptNeg)
else:
    print("Moyenne des réels < 0 : aucun réel < 0 entré ")
if cptPos != 0: # ou alors : if max != 0
    print("Plus grand nombre > 0 : ", max)
else:
    print("Plus grand nombre > 0 : aucun réel > 0 entré ")
```

Exercice n°7 (3 points) :

Soient deux entiers $b1$ et $b2$ saisis par l'utilisateur (on suppose que $b1$ et $b2$ sont > 0 sans avoir à le vérifier). On veut afficher tous les entiers entre $b1$ et $b2$ compris. L'affichage ne se fait que si l'écart entre $b1$ et $b2$ est inférieur ou égal à 20. Si $b1$ est strictement supérieur à $b2$, on affiche les entiers dans l'ordre décroissant sinon on les affiche dans l'ordre croissant. Compléter le programme suivant en utilisant uniquement la boucle `for` (pas la boucle `while`). L'utilisation de `if` est bien sûr autorisée. Les nombres peuvent être tous affichés sur une même ligne ou bien chacun sur une ligne différente (au choix).

Exemples :

- Si $b1 = 2$ et $b2 = 6$ alors on affiche les nombres : 2 3 4 5 6
- Si $b1 = 2$ et $b2 = 30$ alors on affiche « écart trop grand »
- Si $b1 = 2$ et $b2 = 2$ alors on affiche le nombre : 2
- Si $b1 = 8$ et $b2 = 3$ alors on affiche les nombres : 8 7 6 5 4 3
- Si $b1 = 30$ et $b2 = 1$ alors affiche « écart trop grand »

Programme à compléter :

```
b1 = int(input("saisir la borne b1 :"))
b2 = int(input("saisir la borne b2 :"))
if b2 > b1 :
    if b2 - b1 > 20 :
        print("Ecart trop important")
    else:
        for i in range(b1,b2+1):
            print(i," ", end="")
else :
    if b1 - b2 > 20 :
        print("Ecart trop important")
    else:
        for i in range(b1,b2-1,-1):
            print(i , " ", end="")
```

ou bien

```
b1 = int(input("saisir la borne b1 :"))
b2 = int(input("saisir la borne b2 :"))
if b2 - b1 > 20 or b1 - b2 > 20 :
    print("Ecart trop important")
else:
    if b1 <= b2 :
        for i in range(b1,b2+1):
            print(i," ", end="")
    else :
        for i in range(b1,b2-1,-1):
            print(i , " ", end="")
```

Exercice n°8 (4,5 points + 1 point) :

- a) Soient n , $n1$ et $n2$, trois entiers entrés par l'utilisateur (on suppose que les trois valeurs entrées sont strictement positives et que $n1$ est inférieur ou égal à $n2$, sans avoir à le vérifier). Compléter le programme suivant : il doit afficher les n premiers multiples > 0 de $n1$ sur une première ligne puis les n premiers multiples > 0 de $(n1+1)$ sur une deuxième ligne puis les n premiers multiples > 0 de $(n1+2)$ sur une troisième ligne puis ... puis les n premiers multiples > 0 de $n2$ sur une dernière ligne. Les entiers affichés doivent être séparés les uns des autres par un espace. N'utiliser que la boucle while (pas la boucle *for*). L'utilisation de *if* est bien sûr autorisée.

Exemple : avec $n = 6$, $n1 = 2$ et $n2 = 6$, on affiche :

```
2 4 6 8 10 12  
3 6 9 12 15 18  
4 8 12 16 20 24  
5 10 15 20 25 30  
6 12 18 24 30 36
```

Programme à compléter :

```
n = int(input("n ? "))  
n1 = int(input("n1 ? "))  
n2 = int(input("n2 ? "))  
i = n1  
while i <= n2:  
    j = 1  
    while j <= n:  
        print(i*j, end = " ")  
        j = j + 1  
    print()  
    i = i + 1
```

ou bien

```
n = int(input("n ? "))  
n1 = int(input("n1 ? "))  
n2 = int(input("n2 ? "))  
i = n1  
while i <= n2:  
    j=i  
    while j <= i * n:  
        print( j, end = " ")  
        j = j + i  
    print()  
    i = i + 1
```

- b) Modifiez le programme précédent de manière à afficher le message « Erreur : $n_1 > n_2$ » quand n_1 est strictement supérieur à n_2 et le message « Erreur : $n \leq 0$ » quand n est inférieur ou égal à 0 (on suppose toujours, sans avoir à le vérifier, que $n_1 > 0$ et $n_2 > 0$). Ecrivez les modifications à l'aide d'un stylo d'une autre couleur (ou à défaut de stylo, soulignez ces modifications). Soyez très précis quant aux emplacements des modifications apportées.

Exemples :

- Avec $n = -6$, $n_1 = 2$, $n_2 = 5$, on affiche :

Erreur : $n \leq 0$

- Avec $n = -4$, $n_1 = 12$, $n_2 = 3$, on affiche :

Erreur : $n \leq 0$

Erreur : $n_1 > n_2$

```
n = int(input("n ? "))
n1 = int(input("n1 ? "))
n2 = int(input("n2 ? "))

if n > 0 and n1 <= n2 :
    i = n1
    while i <= n2:
        j = 1
        while j <= n:
            print(i*j, end = " ")
            j = j + 1
        print()
        i = i + 1

else :
    if n <= 0 :
        print("Erreur : n <= 0")
    if n1 > n2 :
        print("Erreur : n1 > n2")
```