

---

Ce contrôle comporte 8 questions; il sera noté sur 20 points. Les réponses sont à porter sur une copie comportant votre nom. Il n'est pas nécessaire de répondre aux questions dans l'ordre — commencez par celles où vous vous sentez le plus à l'aise (*mais ne tentez les questions bonus qu'après avoir fini le reste!!*). Une attention particulière sera portée à la qualité et à la clarté du code que vous aurez écrit.

---

1. Questions à choix multiples (aucune justification de la réponse n'est nécessaire):

(a) (1 point) Quelle est la syntaxe correcte pour un 'if' en Python?

- ☐ if condition ( ... )
- ☒ if condition: ...
- ☐ if: condition { ... }
- ☐ if condition { ... }

(b) (1 point) Si on souhaite répéter un bloc de code tant qu'une condition est vraie, quelle structure utilise-t-on en Python?

- ☐ do { ... } while condition
- ☐ repeat ... until condition
- ☒ while condition: ...
- ☐ for condition: ...

(c) (1 point) Comment ajoute-t-on un élément à la fin d'une liste en Python?

- ☐ liste.add(element)
- ☐ liste.insert(element)
- ☒ liste.append(element)
- ☐ liste.push(element)

(d) (1 point) Quelle est la sortie du code suivant?

```
1 txt = "ha"
2 for i in range(3):
3     txt = txt + "ha"
4 print(txt)
```

- ☐ ha
- ☐ haha
- ☐ hahaha
- ☒ hahahaha

2. Fonction d'échange d'éléments dans une liste

(a) (2 points) Ecriture du code: Complétez la fonction 'Echange(L, i, j)' ci-dessous qui prend en entrée une liste L, deux indices i et j, et retourne une liste où les éléments L[i] et L[j] auront été échangés et le reste de la liste sera identique à L.

```
1 def Echange(L, i, j):
2     ## A COMPLETER: code effectuant l'echange des elements d'indice i et j
3     return ...
```

**Solution:**

```
def Echange(L, i, j):  
    temp = L[i]  
    L[i] = L[j]  
    L[j] = temp  
    return L
```

(b) Exécution du code:

- i. ( $\frac{1}{2}$  point) Que retournera l'appel `echange([1,2,3], 1, 2)`?

**Solution:** *Note: si on voulait être strictement exact, cet appel aurait renvoyé une erreur sur le fait que la fonction "echange" n'était pas définie — en effet j'ai fait une faute de frappe en rédigeant l'énoncé: la fonction qu'on a définie à la question précédente était "Echange" (avec un "E" majuscule)...*

Mais sinon la réponse attendue ici était:

[1, 3, 2]

- ii. ( $\frac{1}{2}$  point) Que retournera l'appel `echange([1,2,3], 2, 3)`?

**Solution:** On aurait obtenu une erreur du type "index out of range"; en effet, L contient 3 éléments, qui sont donc indexés de 0 à 2 — et donc L[3] n'existe pas.

3. (2 points) *Comptage jusqu'à 10:* Complétez le code ci-dessous pour afficher à l'écran les nombres de 1 à 10 en utilisant une boucle "while".

```
1 def compte10():  
2     i = 0  
3     while ...  
4         ## A COMPLETER: code affichant, au moyen d'une boucle while, les nombres de 1 à 10  
5     print(i)
```

**Solution:**

```
def compte10():  
    i = 0  
    while i < 10:  
        i += 1  
        print(i)
```

4. (1 point) *Recherche dans une liste:* Ecrivez une fonction 'recherche\_element(liste, element)' qui prend une liste et un élément en entrée, et renvoie 'OUI' si l'élément est présent dans la liste, sinon renvoie 'NON'.

**Solution:**

```
def recherche_element(liste, element):  
    resultat = "NON"  
    if element in liste:  
        resultat = "OUI"  
    return resultat
```

5. (3 points) *Evolution du nombre d'abonnés d'une chaîne YouTube*: Un vlogueur commence avec 1000 abonnés sur sa chaîne YouTube. Chaque mois, il gagne 3% de nouveaux abonnés en plus et il perd 50 abonnés qui se désinscrivent. Écrivez une fonction 'NbAbos(nbMois)' qui retourne une liste contenant l'évolution du nombre d'abonnés sur un nombre nbMois de mois. *Indice: pour conserver la partie entière d'un nombre N plus 3%, on utilise la formule `int(1.03 * N)`.*

**Solution:**

```
def NbAbos(nbMois): # Evidemment presque'identique a celle des abones de triathlon faite en cours
    listeNbAbos = [1000]
    for i in range(nbMois):
        listeNbAbos.append(int(listeNbAbos[i] * 1.03 - 50))
    return listeNbAbos
```

6. (2 points) *Table de multiplication*: Complétez la fonction 'Multiples(num)' ci-dessous: elle prend en entrée un nombre entier 'num' et retourne la liste de ses dix premiers multiples (donc [num\*1, num\*2, num\*3 .... num\*10]).

```
1 def Multiples(num):
2     Lst ....
3     for i ...
4         ## A COMPLETER: code constituant la liste des multiples de num
5     return Lst
```

**Solution:**

```
def Multiples(num):
    Lst = []
    for i in range(10):
        i += 1
        Lst.append(i * num)
    return Lst
```

7. (2 points) *Vérification de parité*: Ecrivez une fonction 'est\_pair(n)' qui prend un entier 'n' en entrée et renvoie 'OUI' si 'n' est pair, sinon renvoie 'NON'.

**Solution:**

```
def est_pair(n):
    if n % 2 == 0:
        resultat = "OUI"
    else:
        resultat = "NON"
    return resultat
```

8. (3 points) Correction de code: même principe que dans l'interrogation précédente — dans les quatre extraits ci-dessous, il y a trois erreurs à identifier et à corriger.

```
1 # A: Affichage des carrés d'entiers 0 à 5
2 for i in range(6):
3     print(i ** 2)
4
5 # B: Compte à rebours partant de 10
6 x = 10
7 while x > 0:
8     print(x)
9     x = x - 1
```

```
1 # C: Affichage des 10 premiers entiers
2 i = 0
3 while i < 10:
4     print(i)
5
6 # D: Doublerment dernier élément d'une liste
7 def DoubleDernier(Lst):
8     a = len(Lst)
9     resultat = Lst[a] * 2
10    return resultat
```

**Solution:**

Bloc A: pas d'erreur, le code fait ce qu'on lui demande.

Bloc B: sortie en erreur pour cause d'indentation erronée après le "while x > 0:"

Bloc C: boucle infinie — on n'incrmente jamais i, il reste donc à 0, et le programme affichera à l'écran une série infinie de 0.

Bloc D: erreur d'indexation: une liste est indexée de 0 à (len(liste) - 1): donc liste[len(liste)] fait référence un index qui n'existe pas.

(Question bonus 1): Écrivez un programme qui demande à l'utilisateur deux nombres entiers 'a' et 'b', puis utilise une boucle while pour calculer le produit 'a x b' sans utiliser l'opérateur de multiplication (\*).

**Solution:**

```
a = int(input("Entrez le premier entier: "))
b = int(input("Entrez le second entier: "))
# Ce n'était pas spécifié mais il fallait évidemment que b soit positif pour une solution
# raisonnablement simple au problème.
i = 0
resultat = 0
while i < b:
    resultat += a
    i += 1
print(resultat)
```

(Question bonus 2): Écrivez un programme qui demande à l'utilisateur d'entrer une liste de nombres (la fin de la liste est marquée par l'entrée d'un nombre négatif), puis utilise une boucle **for** pour calculer et afficher la moyenne de ces nombres.

**Solution:**

```
# Constitution de la liste de nombres pour commencer
Lst = []
while a >= 0:
    a = int(input("Nombre suivant SVP: "))
    if a > 0:
        Lst.append(a)
# Calcul de la somme de tous les nombres de la liste ensuite
moy = 0
for i in range(len(Lst)):
    moy += Lst[i]
# Calcul de la moyenne enfin
moy /= len(Lst)
print(moy)
```