

Test Diagnostique — Python

Instructeur :	Youssef Ben Hsan	Section :	Section Informatique
Nom et prénom :	_____	Durée :	45 min
Date :	_____		

Bloc 1 — Conditions

Q1 · QCM · Conditions — opérateurs logiques

/1 pt

Quelle condition vérifie que n est dans l'intervalle [2, 200] ?

- ☐ A) `n > 2 and n < 200`
- ☐ B) `2 <= n <= 200`
- ☐ C) `n >= 2 or n <= 200`
- ☐ D) `n == 2 or n == 200`

Q2 · Complétion · Conditions — validation de saisie

/2 pts

Complétez la condition pour vérifier que la chaîne s représente un entier décimal valide et que cet entier est compris entre 1 et 100.

```
s = input('Entrez un entier : ')
if _____ :
    n = int(s)
    print('Valide :', n)
else :
    print('Saisie invalide')
```

Bloc 2 — Boucles

Q3 · Lecture · Boucle for — range

/1 pt

Que retourne ce code ?

```
s = 0
for i in range(1, 5) :
    s = s + i
print(s)
```

Q4 · Correction · Boucle while — erreur logique

/2 pts

Ce code est censé afficher les entiers de 3 à 1 en ordre décroissant. Corrigez la ligne fautive.

```
n = 3
while n > 0 :
    print(n)
    n = n + 1  # erreur
```

Bloc 3 — Tableaux numpy

Q5 · Complétion · Tableaux numpy — déclaration et accès

/2 pts

Complétez le programme qui saisit n notes réelles, puis affiche uniquement celles qui sont supérieures à la moyenne du tableau.

```
from numpy import array
n = int(input('Taille : '))
T = _____
for i in range(n) :
    T[i] = float(input('Note ' + str(i) + ' : '))
somme = 0
for i in range(n) :
    somme = somme + T[i]
moy = somme / n
for i in range(n) :
    if _____ :
        print(T[i])
```

Q6 · Lecture · Tableaux numpy — lecture de code

/2 pts

Que fait ce programme ? Donnez la sortie pour le tableau [4, 1, 7, 2, 9, 3].

```
from numpy import array
T = array([4, 1, 7, 2, 9, 3])
n = len(T)
R = array([0] * n)
for i in range(n) :
    R[i] = T[n-1-i]
for i in range(n) :
    print(R[i])
```

Bloc 4 — Chaînes de caractères

Q7 · QCM · Chaînes — fonctions autorisées

/1 pt

Quelle instruction affiche la chaîne ch en majuscules ?

- **A)** ch.maj()
- **B)** ch.upper()
- **C)** upper(ch)
- **D)** ch.toUpperCase()

Q8 · Complétion · Chaînes — recherche et sous-chaîne

/2 pts

Complétez le programme qui vérifie la présence du caractère '@' dans une adresse email saisie, puis affiche la partie avant le '@'.

```
email = input('Email : ')
pos = email.find('@')
if pos ____ -1 :
    nom = email[____:____]
    print('Nom :', nom)
else :
    print('Email invalide')
```

Bloc 5 — Enregistrements (dict)

Q9 · Complétion · Enregistrements — création et accès

/2 pts

Complétez la fonction qui affiche les noms des élèves ayant une moyenne ≥ 10 . Chaque enregistrement a deux champs : nom (str) et moyenne (float).

```
def afficher_reussites(enregs) :
    for e in enregs :
        if _____ :
            print(_____)
```

Q10 · QCM · Enregistrements — syntaxe dict

/1 pt

On a l'enregistrement : `eleve = dict(nom=str(), note=float())`. Quelle instruction affecte correctement la note 14.5 ?

- **A)** `eleve.note = 14.5`
- **B)** `eleve['note'] = 14.5`
- **C)** `eleve[note] = 14.5`
- **D)** `dict(eleve, note=14.5)`

Bloc 6 — Récursivité

Q11 · Complétion · Récursivité — factorielle

/2 pts

Complétez la fonction récursive qui calcule la factorielle de n ($n \geq 0$). Rappel : $0! = 1$, $n! = n \times (n-1)!$

```
def fact(n) :  
    if n == 0 :  
        return 1  
    else :  
        return _____
```

Q12 · Lecture · Récursivité — trace d'exécution

/2 pts

Que retourne l'appel $f(4)$ pour la fonction suivante ? Expliquez brièvement.

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

Bloc 7 — Fichiers

Q13 · Complétion · Fichiers — texte et binaire (pickle)

/3 pts

13a — Complétez la lecture ligne par ligne du fichier notes.txt pour afficher uniquement les notes ≥ 10 .

13b — Complétez l'ajout d'un enregistrement dans le fichier binaire eleves.dat.

```
f = open('notes.txt', 'r')  
ch = f.readline()  
while ch _____ :  
    note = int(ch.strip())  
    if note >= 10 :  
        print(note)  
    ch = _____  
f.close()  
  
from pickle import dump  
def ajouter(enr) :  
    f = open('eleves.dat', _____)  
    _____  
    f.close()
```

Grille de Correction

N°	Concept	Réponse attendue	Score
1	Conditions — opérateurs logiques	B	__ / __
2	Conditions — validation de saisie	<code>s.isdecimal()</code> and <code>1<=int(s)<=100</code>	__ / __
3	Boucle for — range	10	__ / __
4	Boucle while — erreur logique	<code>n = n - 1</code>	__ / __
5	Tableaux numpy — déclaration / filtre	<code>array([0.0]*n) / T[i] > moy</code>	__ / __
6	Tableaux numpy — lecture / inversion	3 2 9 7 1 4 (tableau inversé)	__ / __
7	Chaînes — fonctions autorisées	B	__ / __
8	Chaînes — sous-chaîne / recherche	<code>pos != -1 / email[0:pos]</code>	__ / __
9	Enregistrements — filtre	<code>e['moyenne']>=10 / e['nom']</code>	__ / __
10	Enregistrements — syntaxe dict	B	__ / __
11	Récurtivité — factorielle	<code>n * fact(n - 1)</code>	__ / __
12	Récurtivité — trace	10 (0+1+2+3+4)	__ / __
13	Fichiers texte et binaire	<code>!= '' / f.readline() / 'ab' / dump(enr,f)</code>	__ / __

TOTAL __ / __