

Terminale MOREA  
Épreuve de NSI niveau première générale  
QCM d'entraînement

Durée indicative de l'épreuve : 1 heure  
Séance de préparation : 2 heures

Les calculatrices ne sont pas autorisées pour l'épreuve.

*Consignes.* Le devoir qui suit est un questionnaire à choix multiple qui comporte 20 questions. Pour chaque question, quatre réponses sont proposées. Une seule des quatre réponses proposées est exacte. Une réponse juste rapporte 1 point. Une réponse incorrecte ou une réponse multiple ne rapporte ni n'enlève aucun point. **Pour chaque question, entourer la lettre de l'unique réponse juste. Le porte mine ou crayon à papier est donc recommandé pour éviter les ratures.**

**Question 1**

On a exécuté le programme Python suivant.

```
x = 2
y = 3
x = 6
y = x - 1
x = x + 2
print(x)
print(y)
```

L'affichage obtenu dans le terminal est :

**Réponses :**

A. 8  
5

B. 2  
3

C. 5  
8

D. 3  
2

**Question 2**

On a exécuté le programme Python suivant.

```
tab1 = [4, 5, 0, 1, 2, 0]
print(tab1[0])
```

L'affichage obtenu dans le terminal est :

**Réponses :**

A. un message d'erreur

B. 4

C. 2

D. 3

**Question 3**

On a exécuté le programme Python suivant.

```
tab1 = [4, 5, 1, 0, 2, 0]
print(tab1[2])
```

L'affichage obtenu dans le terminal est :

**Réponses :**

- A. un message d'erreur
- B. 4
- C. 1
- D. 5

**Question 4**

On a exécuté le programme Python suivant.

```
tab1 = [4, 5, 1, 0, 2, 7]
print(tab1[5])
```

L'affichage obtenu dans le terminal est :

**Réponses :**

- A. un message d'erreur
- B. 1
- C. 2
- D. 7

**Question 5**

On a exécuté le programme Python suivant.

```
tab1 = [4, 5, 1, 0, 2, 6]
print(tab1[6])
```

L'affichage obtenu dans le terminal est :

**Réponses :**

- A. un message d'erreur
- B. 6
- C. 5
- D. 4

**Question 6**

On a exécuté le programme Python suivant.

```
tab1 = [13, 5, 0, 2, 6]
tab1[3] = 10
print(tab1)
```

L'affichage obtenu dans le terminal est :

**Réponses :**

- A. un message d'erreur
- B. [13, 5, 0, 10, 6]
- C. [13, 5, 10, 2, 6]
- D. [13, 10, 0, 2, 6]

**Question 7**

On a exécuté le programme Python suivant.

```
tab1 = [13, 5, 0, 2, 4, 6]
tab1[6] = 10
print(tab1)
```

L'affichage obtenu dans le terminal est :

**Réponses :**

- A. un message d'erreur
- B. [13, 5, 0, 2, 10]
- C. [13, 10, 0, 2, 6]
- D. [13, 5, 10, 2, 6]

**Question 8**

On a exécuté le programme Python suivant.

```
tab1 = [13, 5, 0, 2, 6, 8, 12, 23, 5]
print(len(tab1))
```

L'affichage obtenu dans le terminal est :

**Réponses :**

- A. un message d'erreur
- B. 6
- C. 7
- D. 8

**Question 9**

On a exécuté le programme Python suivant.

```
tab1 = [24]
print(len(tab1))
```

L'affichage obtenu dans le terminal est :

**Réponses :**

- A. un message d'erreur
- B. 24
- C. 1
- D. 0

**Question 10**

On a exécuté le programme Python suivant.

```
tab1 = [13, 25, 0, 12]
for elt in tab1:
    print(elt)
```

L'affichage obtenu dans le terminal est :

**Réponses :**

- A. un message d'erreur
- B. 13  
25  
0  
12
- C. 0  
1  
2  
3
- D. 12

## Question 11

On a exécuté le programme Python suivant.

```
for i in range(6):  
    print(i)
```

L'affichage obtenu dans le terminal est :

**Réponses :**

A. 0  
1  
2  
3  
4  
5  
6

B. 1  
2  
3  
4  
5  
6  
7

C. 0  
1  
2  
3  
4  
5

D. 1  
2  
3  
4  
5  
6

## Question 12

On souhaite obtenir l'affichage suivant dans le terminal :

```
0
1
2
3
4
```

Quel est le seul programme ci-dessous qui permet d'obtenir cet affichage exact ?

Réponses :

A. 

```
for i in range(3):
    print(i)
```

B. 

```
for i in range(5):
    print(i)
```

C. 

```
for i in range(6):
    print(i)
```

D. 

```
for i in range(4):
    print(i)
```

## Question 13

On souhaite obtenir l'affichage suivant dans le terminal :

```
23
5
0
2
6
9
```

Quel est le seul programme ci-dessous qui permet d'obtenir cet affichage exact ?

**Réponses :**

A. 

```
tab1 = [23, 5, 0, 2, 6, 9]
for i in range(5):
    print(i)
```

B. 

```
tab1 = [23, 5, 0, 2, 6, 9]
for i in range(5):
    print(tab1[i])
```

C. 

```
tab1 = [23, 5, 0, 2, 6, 9]
for i in range(6):
    print(i)
```

D. 

```
tab1 = [23, 5, 0, 2, 9]
for i in range(6):
    print(tab1[i])
```

## Question 14

On exécute le programme Python donné ci-dessous.

```
def mystere(x):  
    return 2 * x + 3  
  
print(mystere(0))  
print(mystere(2))
```

Quel est l'affichage obtenu dans le terminal ?

**Réponses :**

A. Un message d'erreur.

B. 0  
10

C. 4  
3

D. 3  
7

## Question 15

On exécute le programme Python donné ci-dessous.

```
def mystere(x, y):  
    return 3 * y - 2 * x + 5  
  
print(mystere(0, 0))  
print(mystere(1, 3))
```

Quel est l'affichage obtenu dans le terminal ?

**Réponses :**

A. Un message d'erreur.

B. 0  
-6

C. 5  
12

D. 5  
4

## Question 16

La fonction ci-dessous doit calculer la moyenne des éléments du tableau `tab` passé en paramètre.

```
def moyenne(tab):  
    res = 0  
    for elt in tab:  
        res = ...  
    return res / ...
```

Par exemple :

- L'appel `moyenne([6, 4, 2])` renvoie `4.0`.  
En effet,  $(6 + 4 + 2)/3 = 12/3 = 4$ .
- L'appel `moyenne([6, 4, 2, 2, 1])` renvoie `3.0`.  
En effet,  $((6 + 4 + 2 + 2 + 1)/5 = 15/5 = 3$ .

Par quoi doit-on remplacer les `...` ?

Réponses :

- A. `elt` et `len(tab) - 1`
- B. `res + elt` et `len(tab) - 1`
- C. `elt` et `len(tab)`
- D. `res + elt` et `len(tab)`

## Question 17

On donne ci-dessous le code d'une fonction `mystere` qui prend en paramètre un tableau `tab`.

```
def mystere(tab):  
    res = 0  
    for elt in tab:  
        if elt < 10:  
            res = res + 1  
    return res
```

Quelle valeur renvoie l'appel `mystere([12, 10, 5, 14, 20, 8, 2])` ?

Réponses :

- A. `1`
- B. `2`
- C. `3`
- D. `4`

## Question 18

On donne ci-dessous le code incomplet d'une fonction `compare`.

```
def compare(tab1, tab2):  
    for i in range(len(tab1)):  
        if not tab1[i] == ...:  
            return ...  
    return ...
```

Cette fonction prend en paramètre deux tableaux `tab1` et `tab2` de même taille et doit renvoyer :

- `True` si les deux tableaux possèdent les mêmes éléments dans le même ordre ;
- `False` sinon.

Par exemple :

- L'appel `compare([7, 5, 8], [7, 5, 8])` renvoie `True`.
- L'appel `compare([7, 5, 6, 4], [7, 5, 8, 4])` renvoie `False`.

Par quoi doit-on remplacer les `...` ?

Réponses :

- A. `tab2[i + 1]`, `True` et `False`
- B. `tab2[i]`, `False` et `True`
- C. `tab2[i]`, `True` et `False`
- D. `tab1[i + 1]`, `False` et `True`

## Question 19

On donne ci-dessous le code incomplet d'une fonction `est_trie`.

```
def est_trie(tab):  
    for i in range(...):  
        if tab[i] > tab[...]:  
            return False  
    return True
```

Cette fonction prend en paramètre un tableau `tab` et possédant au moins deux éléments. La fonction `est_trie` doit renvoyer :

- `True` si `tab` est trié dans l'ordre croissant ;
- `False` sinon.

Par exemple :

- L'appel `est_trie([5, 7, 8, 9, 12, 15, 21, 21, 27])` renvoie `True`.
- L'appel `est_trie([9, 7, 8, 9, 12, 15, 21, 21, 27])` renvoie `False`.

Par quoi doit-on remplacer les `...` ?

**Réponses :**

- A. `len(tab)`, et `i + 1`
- B. `len(tab)`, et `i - 1`
- C. `len(tab) - 1` et `i + 1`
- D. `len(tab) - 1` et `i - 1`

## Question 20

On donne ci-dessous le code incomplet d'une fonction `recherche` prenant en paramètres un tableau `tab` et une valeur `val` (nombre) à rechercher dans `tab`.

```
def recherche(val, tab):  
    res = ...  
    for i in range(len(tab)):  
        if ... == val:  
            res = ...  
    return res
```

La fonction `recherche` doit renvoyer :

- l'indice de la dernière case contenant `val` (cas où `val` est présente au moins une fois dans `tab`);
- `False` sinon (cas où `val` est absente de `tab`).

Par exemple :

- L'appel `recherche(10, [15, 7, 9, 10, 12, 10])` renvoie `5`.
- L'appel `recherche(11, [15, 7, 9, 10, 12, 10])` renvoie `False`.

Par quoi doit-on remplacer les `...` ?

Réponses :

- A. `True`, `i` et `tab[i]`
- B. `False`, `tab[i]` et `i`
- C. `False`, `i` et `tab[i]`
- D. `True`, `tab[i]` et `i`