



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

- ☐ `for` et `range`
- ☐ `while` et `for`
- ☐ `if` et `else`
- ☐ `input()` et `print()`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ provoque une erreur
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':' :

- ☐ commence par **begin** et se termine par **end**
- ☐ contient obligatoirement un test simple
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ est indenté par rapport à la ligne qui le précède

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0
- ☐ tester si la variable `x%3` vaut 0
- ☐ affecter la valeur 0 à la variable x
- ☐ rien, elle est invalide et produira une erreur

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ SyntaxError ☐ 13 ☐ 12 ☐ 30 ☐ x

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ est une boucle conditionnelle
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ interdite car la valeur de i n'est jamais utilisée

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 8 ☐ 16 ☐ 32 ☐ 2



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

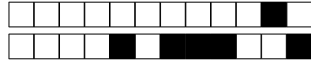
- ☐ `if` et `else`
- ☐ `input()` et `print()`
- ☐ `for` et `range`
- ☐ `while` et `for`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ provoque une erreur
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ contient obligatoirement un test simple
- ☐ commence par **begin** et se termine par **end**
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ est indenté par rapport à la ligne qui le précède

Question 5 `x` étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si le reste de la division euclidienne de `x` par 3 vaut 0
- ☐ rien, elle est invalide et produira une erreur
- ☐ tester si la variable `x%3` vaut 0
- ☐ affecter la valeur 0 à la variable `x`

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ 30 ☐ 13 ☐ 12 ☐ SyntaxError ☐ x

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ est une boucle conditionnelle
- ☐ interdite car la valeur de `i` n'est jamais utilisée
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 32 ☐ 2 ☐ 16 ☐ 8



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....
.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

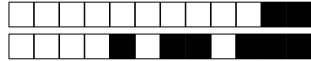
- ☐ `input()` et `print()`
- ☐ `while` et `for`
- ☐ `if` et `else`
- ☐ `for` et `range`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ provoque une erreur
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ est indenté par rapport à la ligne qui le précède
- ☐ contient obligatoirement un test simple
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ commence par **begin** et se termine par **end**

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si la variable `x%3` vaut 0
- ☐ affecter la valeur 0 à la variable `x`
- ☐ rien, elle est invalide et produira une erreur
- ☐ tester si le reste de la division euclidienne de `x` par 3 vaut 0

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ x ☐ SyntaxError ☐ 12 ☐ 30 ☐ 13

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ est une boucle conditionnelle
- ☐ interdite car la valeur de `i` n'est jamais utilisée
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 8 ☐ 32 ☐ 2 ☐ 16



- ☐ 0 ☐ 0
- ☐ 1 ☐ 1
- ☐ 2 ☐ 2
- ☐ 3 ☐ 3
- ☐ 4 ☐ 4
- ☐ 5 ☐ 5
- ☐ 6 ☐ 6
- ☐ 7 ☐ 7
- ☐ 8 ☐ 8
- ☐ 9 ☐ 9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

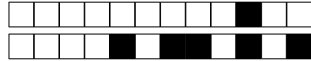
- ☐ `if` et `else`
- ☐ `while` et `for`
- ☐ `for` et `range`
- ☐ `input()` et `print()`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ provoque une erreur
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐ F ☐ AB ☐ TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ commence par **begin** et se termine par **end**
- ☐ contient obligatoirement un test simple
- ☐ est indenté par rapport à la ligne qui le précède

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si la variable `x%3` vaut 0
- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0
- ☐ rien, elle est invalide et produira une erreur
- ☐ affecter la valeur 0 à la variable x

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ 30 ☐ SyntaxError ☐ x ☐ 13 ☐ 12

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ est une boucle conditionnelle
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ interdite car la valeur de i n'est jamais utilisée

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 8 ☐ 16 ☐ 2 ☐ 32



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

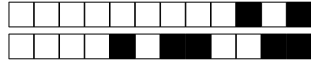
- ☐ `for` et `range`
- ☐ `input()` et `print()`
- ☐ `while` et `for`
- ☐ `if` et `else`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ affecte la chaîne de caractères `'0.75'` à la variable `3quarts`
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ provoque une erreur

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ contient obligatoirement un test simple
- ☐ est indenté par rapport à la ligne qui le précède
- ☐ commence par **begin** et se termine par **end**
- ☐ est délimité par des simples quotes ou des doubles quotes

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0
- ☐ affecter la valeur 0 à la variable x
- ☐ rien, elle est invalide et produira une erreur
- ☐ tester si la variable $x\%3$ vaut 0

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ 12 ☐ x ☐ SyntaxError ☐ 30 ☐ 13

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ est une boucle conditionnelle
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ interdite car la valeur de i n'est jamais utilisée

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 32 ☐ 2 ☐ 8 ☐ 16



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

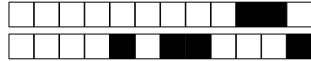
- ☐ `for` et `range`
- ☐ `input()` et `print()`
- ☐ `if` et `else`
- ☐ `while` et `for`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ provoque une erreur

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ contient obligatoirement un test simple
- ☐ est indenté par rapport à la ligne qui le précède
- ☐ commence par **begin** et se termine par **end**

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si la variable `x%3` vaut 0
- ☐ rien, elle est invalide et produira une erreur
- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0
- ☐ affecter la valeur 0 à la variable x

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ 30 ☐ 12 ☐ SyntaxError ☐ 13 ☐ x

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ interdite car la valeur de i n'est jamais utilisée
- ☐ est une boucle conditionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 8 ☐ 16 ☐ 32 ☐ 2



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

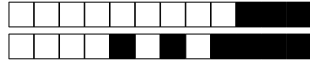
- ☐ while et for
- ☐ if et else
- ☐ input() et print()
- ☐ for et range

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ affecte la chaine de caractères '0.75' à la variable `3quarts`
- ☐ provoque une erreur

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ contient obligatoirement un test simple
- ☐ commence par **begin** et se termine par **end**
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ est indenté par rapport à la ligne qui le précède

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0
- ☐ rien, elle est invalide et produira une erreur
- ☐ affecter la valeur 0 à la variable x
- ☐ tester si la variable $x\%3$ vaut 0

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ x ☐ 12 ☐ 13 ☐ 30 ☐ SyntaxError

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ interdite car la valeur de i n'est jamais utilisée
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ est une boucle conditionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 8 ☐ 16 ☐ 32 ☐ 2



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

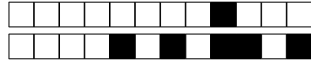
- ☐ while et for
- ☐ if et else
- ☐ for et range
- ☐ input() et print()

Question 2 On considère l'instruction `3quarts = '0.75'` . L'exécution de cette instruction:

- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ provoque une erreur
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ affecte la chaine de caractères '0.75' à la variable `3quarts`

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ contient obligatoirement un test simple
- ☐ est indenté par rapport à la ligne qui le précède
- ☐ commence par **begin** et se termine par **end**
- ☐ est délimité par des simples quotes ou des doubles quotes

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ affecter la valeur 0 à la variable x
- ☐ tester si la variable `x%3` vaut 0
- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0
- ☐ rien, elle est invalide et produira une erreur

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ SyntaxError ☐ 12 ☐ 13 ☐ 30 ☐ x

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

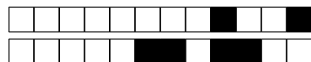
- ☐ est une boucle conditionnelle
- ☐ termine toujours car c'est une boucle bornée ou incondionnelle
- ☐ interdite car la valeur de i n'est jamais utilisée
- ☐ dangereuse, car elle peut conduire à une boucle infinie

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 16 ☐ 32 ☐ 8 ☐ 2



- ☐ 0 ☐ 0
- ☐ 1 ☐ 1
- ☐ 2 ☐ 2
- ☐ 3 ☐ 3
- ☐ 4 ☐ 4
- ☐ 5 ☐ 5
- ☐ 6 ☐ 6
- ☐ 7 ☐ 7
- ☐ 8 ☐ 8
- ☐ 9 ☐ 9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....
.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

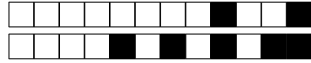
- ☐ `if` et `else`
- ☐ `while` et `for`
- ☐ `input()` et `print()`
- ☐ `for` et `range`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ provoque une erreur
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐ F ☐ AB ☐ TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ commence par `begin` et se termine par `end`
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ contient obligatoirement un test simple
- ☐ est indenté par rapport à la ligne qui le précède

Question 5 `x` étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si la variable `x%3` vaut 0
- ☐ rien, elle est invalide et produira une erreur
- ☐ affecter la valeur 0 à la variable `x`
- ☐ tester si le reste de la division euclidienne de `x` par 3 vaut 0

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ `SyntaxError` ☐ 13 ☐ `x` ☐ 12 ☐ 30

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ interdite car la valeur de `i` n'est jamais utilisée
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ est une boucle conditionnelle
- ☐ termine toujours car c'est une boucle bornée ou incondionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 2 ☐ 16 ☐ 32 ☐ 8



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

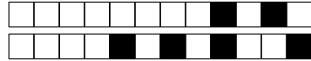
- ☐ `input()` et `print()`
- ☐ `for` et `range`
- ☐ `while` et `for`
- ☐ `if` et `else`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ provoque une erreur

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ contient obligatoirement un test simple
- ☐ est indenté par rapport à la ligne qui le précède
- ☐ commence par **begin** et se termine par **end**

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ rien, elle est invalide et produira une erreur
- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0
- ☐ tester si la variable `x%3` vaut 0
- ☐ affecter la valeur 0 à la variable x

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ 12 ☐ 13 ☐ 30 ☐ x ☐ SyntaxError

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

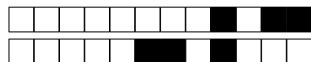
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ interdite car la valeur de i n'est jamais utilisée
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ est une boucle conditionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 2 ☐ 16 ☐ 8 ☐ 32



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....
.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

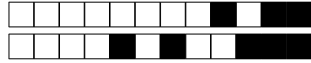
- ☐ while et for
- ☐ input() et print()
- ☐ for et range
- ☐ if et else

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ provoque une erreur
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':' :

- ☐ est indenté par rapport à la ligne qui le précède
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ contient obligatoirement un test simple
- ☐ commence par `begin` et se termine par `end`

Question 5 `x` étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si la variable `x%3` vaut 0
- ☐ tester si le reste de la division euclidienne de `x` par 3 vaut 0
- ☐ affecter la valeur 0 à la variable `x`
- ☐ rien, elle est invalide et produira une erreur

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ `SyntaxError` ☐ 13 ☐ 30 ☐ 12 ☐ `x`

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

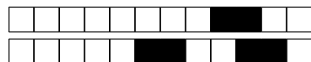
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ est une boucle conditionnelle
- ☐ interdite car la valeur de `i` n'est jamais utilisée

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 2 ☐ 32 ☐ 16 ☐ 8



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

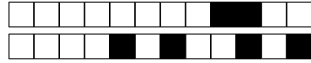
- ☐ `if` et `else`
- ☐ `input()` et `print()`
- ☐ `while` et `for`
- ☐ `for` et `range`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ provoque une erreur

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ contient obligatoirement un test simple
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ commence par **begin** et se termine par **end**
- ☐ est indenté par rapport à la ligne qui le précède

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0
- ☐ rien, elle est invalide et produira une erreur
- ☐ tester si la variable `x%3` vaut 0
- ☐ affecter la valeur 0 à la variable x

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ SyntaxError ☐ 13 ☐ 12 ☐ 30 ☐ x

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

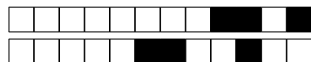
- ☐ interdite car la valeur de i n'est jamais utilisée
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ est une boucle conditionnelle
- ☐ dangereuse, car elle peut conduire à une boucle infinie

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 2 ☐ 32 ☐ 16 ☐ 8



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

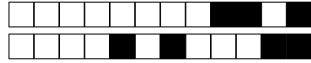
- ☐ `input()` et `print()`
- ☐ `while` et `for`
- ☐ `for` et `range`
- ☐ `if` et `else`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ provoque une erreur
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ est indenté par rapport à la ligne qui le précède
- ☐ contient obligatoirement un test simple
- ☐ commence par **begin** et se termine par **end**
- ☐ est délimité par des simples quotes ou des doubles quotes

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si la variable `x%3` vaut 0
- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0
- ☐ rien, elle est invalide et produira une erreur
- ☐ affecter la valeur 0 à la variable x

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ 30 ☐ x ☐ 12 ☐ 13 ☐ SyntaxError

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ est une boucle conditionnelle
- ☐ interdite car la valeur de i n'est jamais utilisée
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 8 ☐ 32 ☐ 2 ☐ 16



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

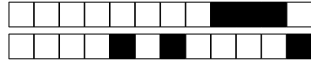
- ☐ `for` et `range`
- ☐ `while` et `for`
- ☐ `input()` et `print()`
- ☐ `if` et `else`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ provoque une erreur

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ contient obligatoirement un test simple
- ☐ commence par **begin** et se termine par **end**
- ☐ est indenté par rapport à la ligne qui le précède

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ affecter la valeur 0 à la variable x
- ☐ rien, elle est invalide et produira une erreur
- ☐ tester si la variable `x%3` vaut 0
- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ SyntaxError ☐ x ☐ 12 ☐ 30 ☐ 13

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ interdite car la valeur de i n'est jamais utilisée
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ est une boucle conditionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 16 ☐ 2 ☐ 8 ☐ 32



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

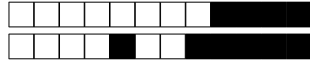
- ☐ `if` et `else`
- ☐ `while` et `for`
- ☐ `input()` et `print()`
- ☐ `for` et `range`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ provoque une erreur
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ affecte la chaine de caractères '0.75' à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ contient obligatoirement un test simple
- ☐ est indenté par rapport à la ligne qui le précède
- ☐ commence par **begin** et se termine par **end**

Question 5 `x` étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si le reste de la division euclidienne de `x` par 3 vaut 0
- ☐ rien, elle est invalide et produira une erreur
- ☐ affecter la valeur 0 à la variable `x`
- ☐ tester si la variable `x%3` vaut 0

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ `x` ☐ 12 ☐ `SyntaxError` ☐ 13 ☐ 30

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ est une boucle conditionnelle
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ interdite car la valeur de `i` n'est jamais utilisée
- ☐ dangereuse, car elle peut conduire à une boucle infinie

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 16 ☐ 32 ☐ 2 ☐ 8



- ☐ 0 ☐ 0
- ☐ 1 ☐ 1
- ☐ 2 ☐ 2
- ☐ 3 ☐ 3
- ☐ 4 ☐ 4
- ☐ 5 ☐ 5
- ☐ 6 ☐ 6
- ☐ 7 ☐ 7
- ☐ 8 ☐ 8
- ☐ 9 ☐ 9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

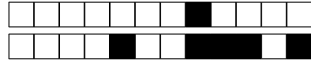
- ☐ `input()` et `print()`
- ☐ `while` et `for`
- ☐ `if` et `else`
- ☐ `for` et `range`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ provoque une erreur
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐ F ☐ AB ☐ TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ contient obligatoirement un test simple
- ☐ est indenté par rapport à la ligne qui le précède
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ commence par `begin` et se termine par `end`

Question 5 `x` étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ rien, elle est invalide et produira une erreur
- ☐ tester si le reste de la division euclidienne de `x` par 3 vaut 0
- ☐ tester si la variable `x%3` vaut 0
- ☐ affecter la valeur 0 à la variable `x`

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ `x` ☐ 12 ☐ 30 ☐ 13 ☐ `SyntaxError`

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

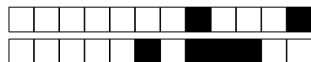
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ est une boucle conditionnelle
- ☐ interdite car la valeur de `i` n'est jamais utilisée
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 2 ☐ 8 ☐ 32 ☐ 16



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

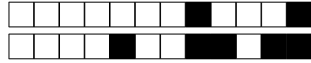
- ☐ `while` et `for`
- ☐ `for` et `range`
- ☐ `input()` et `print()`
- ☐ `if` et `else`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ provoque une erreur
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ contient obligatoirement un test simple
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ commence par **begin** et se termine par **end**
- ☐ est indenté par rapport à la ligne qui le précède

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si la variable `x%3` vaut 0
- ☐ affecter la valeur 0 à la variable `x`
- ☐ tester si le reste de la division euclidienne de `x` par 3 vaut 0
- ☐ rien, elle est invalide et produira une erreur

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ 12 ☐ x ☐ 13 ☐ 30 ☐ SyntaxError

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

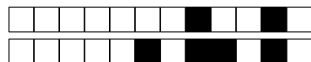
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ interdite car la valeur de `i` n'est jamais utilisée
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ est une boucle conditionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 16 ☐ 32 ☐ 8 ☐ 2



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

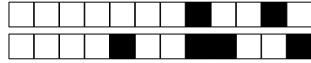
- ☐ `for` et `range`
- ☐ `while` et `for`
- ☐ `if` et `else`
- ☐ `input()` et `print()`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ provoque une erreur
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ contient obligatoirement un test simple
- ☐ commence par **begin** et se termine par **end**
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ est indenté par rapport à la ligne qui le précède

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si la variable `x%3` vaut 0
- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0
- ☐ rien, elle est invalide et produira une erreur
- ☐ affecter la valeur 0 à la variable x

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ SyntaxError ☐ 30 ☐ x ☐ 12 ☐ 13

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

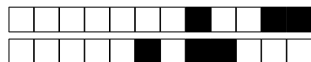
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ est une boucle conditionnelle
- ☐ interdite car la valeur de i n'est jamais utilisée
- ☐ termine toujours car c'est une boucle bornée ou incondionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 16 ☐ 32 ☐ 2 ☐ 8



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

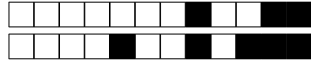
- ☐ `input()` et `print()`
- ☐ `for` et `range`
- ☐ `while` et `for`
- ☐ `if` et `else`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ provoque une erreur
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ est indenté par rapport à la ligne qui le précède
- ☐ commence par **begin** et se termine par **end**
- ☐ contient obligatoirement un test simple
- ☐ est délimité par des simples quotes ou des doubles quotes

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0
- ☐ affecter la valeur 0 à la variable x
- ☐ rien, elle est invalide et produira une erreur
- ☐ tester si la variable $x\%3$ vaut 0

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ 12 ☐ 13 ☐ 30 ☐ x ☐ SyntaxError

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ est une boucle conditionnelle
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ interdite car la valeur de i n'est jamais utilisée

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 16 ☐ 2 ☐ 8 ☐ 32



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

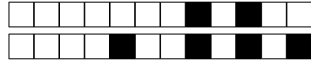
- ☐ `input()` et `print()`
- ☐ `if` et `else`
- ☐ `while` et `for`
- ☐ `for` et `range`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ provoque une erreur
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ commence par `begin` et se termine par `end`
- ☐ contient obligatoirement un test simple
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ est indenté par rapport à la ligne qui le précède

Question 5 `x` étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si le reste de la division euclidienne de `x` par 3 vaut 0
- ☐ affecter la valeur 0 à la variable `x`
- ☐ rien, elle est invalide et produira une erreur
- ☐ tester si la variable `x%3` vaut 0

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ 30 ☐ 12 ☐ 13 ☐ SyntaxError ☐ x

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

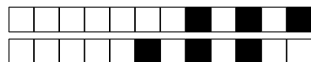
- ☐ est une boucle conditionnelle
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ interdite car la valeur de `i` n'est jamais utilisée

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 8 ☐ 32 ☐ 16 ☐ 2



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

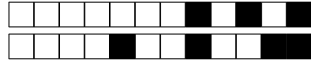
- ☐ while et for
- ☐ if et else
- ☐ input() et print()
- ☐ for et range

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ provoque une erreur
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ est indenté par rapport à la ligne qui le précède
- ☐ commence par **begin** et se termine par **end**
- ☐ contient obligatoirement un test simple

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ affecter la valeur 0 à la variable x
- ☐ rien, elle est invalide et produira une erreur
- ☐ tester si la variable `x%3` vaut 0
- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ 13 ☐ 12 ☐ x ☐ 30 ☐ SyntaxError

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ interdite car la valeur de i n'est jamais utilisée
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ est une boucle conditionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 16 ☐ 32 ☐ 8 ☐ 2



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

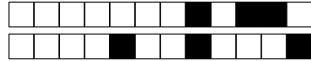
- ☐ `if` et `else`
- ☐ `for` et `range`
- ☐ `input()` et `print()`
- ☐ `while` et `for`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ provoque une erreur
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ contient obligatoirement un test simple
- ☐ commence par **begin** et se termine par **end**
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ est indenté par rapport à la ligne qui le précède

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0
- ☐ tester si la variable `x%3` vaut 0
- ☐ rien, elle est invalide et produira une erreur
- ☐ affecter la valeur 0 à la variable x

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ x ☐ SyntaxError ☐ 30 ☐ 13 ☐ 12

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ est une boucle conditionnelle
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ interdite car la valeur de i n'est jamais utilisée

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 16 ☐ 2 ☐ 32 ☐ 8



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

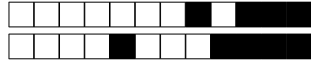
- ☐ `if` et `else`
- ☐ `while` et `for`
- ☐ `input()` et `print()`
- ☐ `for` et `range`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ provoque une erreur

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ commence par `begin` et se termine par `end`
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ est indenté par rapport à la ligne qui le précède
- ☐ contient obligatoirement un test simple

Question 5 `x` étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si le reste de la division euclidienne de `x` par 3 vaut 0
- ☐ rien, elle est invalide et produira une erreur
- ☐ tester si la variable `x%3` vaut 0
- ☐ affecter la valeur 0 à la variable `x`

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ 30 ☐ `SyntaxError` ☐ 13 ☐ `x` ☐ 12

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ interdite car la valeur de `i` n'est jamais utilisée
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ est une boucle conditionnelle
- ☐ termine toujours car c'est une boucle bornée ou incondionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 2 ☐ 16 ☐ 8 ☐ 32



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

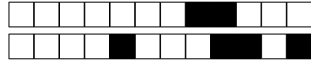
- ☐ `for` et `range`
- ☐ `while` et `for`
- ☐ `input()` et `print()`
- ☐ `if` et `else`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ provoque une erreur

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ commence par **begin** et se termine par **end**
- ☐ est indenté par rapport à la ligne qui le précède
- ☐ contient obligatoirement un test simple

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ affecter la valeur 0 à la variable x
- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0
- ☐ tester si la variable `x%3` vaut 0
- ☐ rien, elle est invalide et produira une erreur

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ SyntaxError ☐ 12 ☐ 30 ☐ x ☐ 13

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ est une boucle conditionnelle
- ☐ interdite car la valeur de i n'est jamais utilisée
- ☐ termine toujours car c'est une boucle bornée ou incondionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 2 ☐ 32 ☐ 16 ☐ 8



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

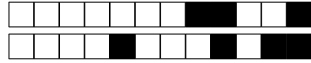
- ☐ `while` et `for`
- ☐ `input()` et `print()`
- ☐ `if` et `else`
- ☐ `for` et `range`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ provoque une erreur
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ est indenté par rapport à la ligne qui le précède
- ☐ commence par **begin** et se termine par **end**
- ☐ contient obligatoirement un test simple

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ rien, elle est invalide et produira une erreur
- ☐ affecter la valeur 0 à la variable x
- ☐ tester si la variable $x\%3$ vaut 0
- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ SyntaxError ☐ 30 ☐ 12 ☐ x ☐ 13

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

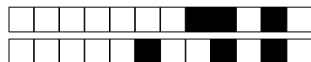
- ☐ interdite car la valeur de i n'est jamais utilisée
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ est une boucle conditionnelle
- ☐ termine toujours car c'est une boucle bornée ou incondionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 8 ☐ 2 ☐ 16 ☐ 32



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

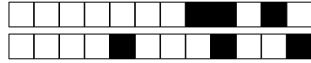
- ☐ `input()` et `print()`
- ☐ `while` et `for`
- ☐ `for` et `range`
- ☐ `if` et `else`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ provoque une erreur

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':' :

- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ commence par **begin** et se termine par **end**
- ☐ contient obligatoirement un test simple
- ☐ est indenté par rapport à la ligne qui le précède

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ affecter la valeur 0 à la variable x
- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0
- ☐ rien, elle est invalide et produira une erreur
- ☐ tester si la variable $x\%3$ vaut 0

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ SyntaxError ☐ 30 ☐ x ☐ 12 ☐ 13

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

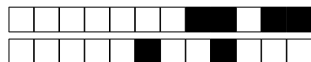
- ☐ interdite car la valeur de i n'est jamais utilisée
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ est une boucle conditionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 32 ☐ 2 ☐ 8 ☐ 16



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

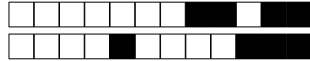
- ☐ `if` et `else`
- ☐ `while` et `for`
- ☐ `for` et `range`
- ☐ `input()` et `print()`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ provoque une erreur
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ est indenté par rapport à la ligne qui le précède
- ☐ commence par **begin** et se termine par **end**
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ contient obligatoirement un test simple

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ rien, elle est invalide et produira une erreur
- ☐ affecter la valeur 0 à la variable x
- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0
- ☐ tester si la variable `x%3` vaut 0

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ SyntaxError ☐ 12 ☐ 30 ☐ x ☐ 13

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

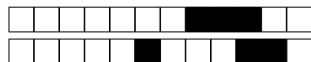
- ☐ interdite car la valeur de i n'est jamais utilisée
- ☐ est une boucle conditionnelle
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ termine toujours car c'est une boucle bornée ou incondionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 2 ☐ 32 ☐ 16 ☐ 8



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

- ☐ `for` et `range`
- ☐ `if` et `else`
- ☐ `while` et `for`
- ☐ `input()` et `print()`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ provoque une erreur

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ est indenté par rapport à la ligne qui le précède
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ commence par **begin** et se termine par **end**
- ☐ contient obligatoirement un test simple

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ rien, elle est invalide et produira une erreur
- ☐ affecter la valeur 0 à la variable x
- ☐ tester si la variable $x\%3$ vaut 0
- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ 12 ☐ SyntaxError ☐ x ☐ 30 ☐ 13

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

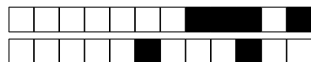
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ interdite car la valeur de i n'est jamais utilisée
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ est une boucle conditionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 16 ☐ 2 ☐ 8 ☐ 32



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

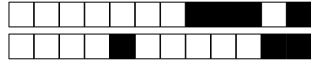
- ☐ if et else
- ☐ while et for
- ☐ input() et print()
- ☐ for et range

Question 2 On considère l'instruction `3quarts = '0.75'` . L'exécution de cette instruction:

- ☐ provoque une erreur
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ affecte la chaine de caractères '0.75' à la variable `3quarts`
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ est indenté par rapport à la ligne qui le précède
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ commence par **begin** et se termine par **end**
- ☐ contient obligatoirement un test simple

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si la variable `x%3` vaut 0
- ☐ affecter la valeur 0 à la variable `x`
- ☐ tester si le reste de la division euclidienne de `x` par 3 vaut 0
- ☐ rien, elle est invalide et produira une erreur

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ 12 ☐ 13 ☐ 30 ☐ x ☐ SyntaxError

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

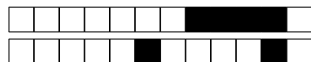
- ☐ est une boucle conditionnelle
- ☐ interdite car la valeur de `i` n'est jamais utilisée
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ termine toujours car c'est une boucle bornée ou incondionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 16 ☐ 2 ☐ 32 ☐ 8



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

- ☐ `for` et `range`
- ☐ `while` et `for`
- ☐ `if` et `else`
- ☐ `input()` et `print()`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ provoque une erreur
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ est indenté par rapport à la ligne qui le précède
- ☐ contient obligatoirement un test simple
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ commence par **begin** et se termine par **end**

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ tester si la variable `x%3` vaut 0
- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0
- ☐ affecter la valeur 0 à la variable x
- ☐ rien, elle est invalide et produira une erreur

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ SyntaxError ☐ 13 ☐ 30 ☐ 12 ☐ x

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ est une boucle conditionnelle
- ☐ interdite car la valeur de i n'est jamais utilisée
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 32 ☐ 2 ☐ 16 ☐ 8



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

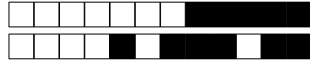
- ☐ `if` et `else`
- ☐ `while` et `for`
- ☐ `input()` et `print()`
- ☐ `for` et `range`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ provoque une erreur

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) / 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ est indenté par rapport à la ligne qui le précède
- ☐ contient obligatoirement un test simple
- ☐ commence par **begin** et se termine par **end**

Question 5 `x` étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ affecter la valeur 0 à la variable `x`
- ☐ tester si le reste de la division euclidienne de `x` par 3 vaut 0
- ☐ tester si la variable `x%3` vaut 0
- ☐ rien, elle est invalide et produira une erreur

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ 30 ☐ 12 ☐ SyntaxError ☐ 13 ☐ x

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ interdite car la valeur de `i` n'est jamais utilisée
- ☐ est une boucle conditionnelle
- ☐ termine toujours car c'est une boucle bornée ou incondionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 2 ☐ 8 ☐ 16 ☐ 32



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

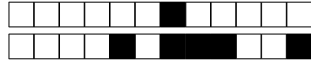
- ☐ `for` et `range`
- ☐ `while` et `for`
- ☐ `if` et `else`
- ☐ `input()` et `print()`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ provoque une erreur

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ commence par `begin` et se termine par `end`
- ☐ contient obligatoirement un test simple
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ est indenté par rapport à la ligne qui le précède

Question 5 `x` étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ affecter la valeur 0 à la variable `x`
- ☐ tester si le reste de la division euclidienne de `x` par 3 vaut 0
- ☐ tester si la variable `x%3` vaut 0
- ☐ rien, elle est invalide et produira une erreur

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ `x` ☐ 13 ☐ 12 ☐ 30 ☐ `SyntaxError`

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ est une boucle conditionnelle
- ☐ interdite car la valeur de `i` n'est jamais utilisée
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 2 ☐ 32 ☐ 16 ☐ 8



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

- ☐ while et for
- ☐ for et range
- ☐ input() et print()
- ☐ if et else

Question 2 On considère l'instruction `3quarts = '0.75'` . L'exécution de cette instruction:

- ☐ affecte la chaine de caractères '0.75' à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ provoque une erreur

Question 3 Écrire un programme qui demande à l'utilisateur une entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':' :

- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ contient obligatoirement un test simple
- ☐ commence par **begin** et se termine par **end**
- ☐ est indenté par rapport à la ligne qui le précède

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ affecter la valeur 0 à la variable x
- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0
- ☐ tester si la variable `x%3` vaut 0
- ☐ rien, elle est invalide et produira une erreur

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ SyntaxError ☐ 12 ☐ 13 ☐ 30 ☐ x

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ est une boucle conditionnelle
- ☐ interdite car la valeur de i n'est jamais utilisée
- ☐ dangereuse, car elle peut conduire à une boucle infinie

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 2 ☐ 8 ☐ 16 ☐ 32



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

- ☐ `while` et `for`
- ☐ `for` et `range`
- ☐ `if` et `else`
- ☐ `input()` et `print()`

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ provoque une erreur
- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':' :

- ☐ est indenté par rapport à la ligne qui le précède
- ☐ contient obligatoirement un test simple
- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ commence par **begin** et se termine par **end**

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ affecter la valeur 0 à la variable x
- ☐ rien, elle est invalide et produira une erreur
- ☐ tester si la variable `x%3` vaut 0
- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ x ☐ 13 ☐ SyntaxError ☐ 12 ☐ 30

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ est une boucle conditionnelle
- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ interdite car la valeur de i n'est jamais utilisée
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 16 ☐ 8 ☐ 32 ☐ 2



- ☐0 ☐0
- ☐1 ☐1
- ☐2 ☐2
- ☐3 ☐3
- ☐4 ☐4
- ☐5 ☐5
- ☐6 ☐6
- ☐7 ☐7
- ☐8 ☐8
- ☐9 ☐9

← codez votre numéro d'identification
ci-contre, et inscrivez votre nom et prénom
ci-dessous.

Nom et prénom :

.....

.....

Durée : 10 minutes.

L'usage de la calculatrice n'est pas autorisé.

Les questions ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Question 1 Les instructions d'entrées/sorties en python sont

- ☐ if et else
- ☐ for et range
- ☐ input() et print()
- ☐ while et for

Question 2 On considère l'instruction `3quarts = '0.75'`. L'exécution de cette instruction:

- ☐ affecte le nombre décimal 0.75 à la variable `3quarts`
- ☐ affecte la chaîne de caractères '0.75' à la variable `3quarts`
- ☐ provoque l'affichage `3quarts = '0.75'` à l'écran
- ☐ provoque une erreur

Question 3 Écrire un programme qui demande à l'utilisateur un entier positif n , puis calcule et affiche la somme $1 + 2 + 3 + \dots + n$ ainsi que l'entier $n * (n + 1) // 2$.

☐F ☐AB ☐TB



Question 4 En python, un bloc d'instructions suivant le caractère ':'

- ☐ est délimité par des simples quotes ou des doubles quotes
- ☐ commence par **begin** et se termine par **end**
- ☐ contient obligatoirement un test simple
- ☐ est indenté par rapport à la ligne qui le précède

Question 5 x étant une variable ayant été affectée, la construction `if x%3 == 0:` signifie:

- ☐ affecter la valeur 0 à la variable x
- ☐ tester si le reste de la division euclidienne de x par 3 vaut 0
- ☐ rien, elle est invalide et produira une erreur
- ☐ tester si la variable $x\%3$ vaut 0

Question 6 L'exécution du code:

```
x = 10
for i in range(3):
    x = x + i
print(x)
```

produira l'affichage:

- ☐ 13 ☐ SyntaxError ☐ x ☐ 30 ☐ 12

Question 7 La séquence d'instruction

```
for i in range(3):
    x = x + 1
```

est:

- ☐ dangereuse, car elle peut conduire à une boucle infinie
- ☐ termine toujours car c'est une boucle bornée ou inconditionnelle
- ☐ interdite car la valeur de i n'est jamais utilisée
- ☐ est une boucle conditionnelle

Question 8 L'exécution du code:

```
n = 16
x = 32
while x < n:
    x = x - 8
print(x)
```

produira l'affichage:

- ☐ 2 ☐ 8 ☐ 16 ☐ 32