

Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut x après l'appel `x = 2 ; y = h(x) ?`

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 32  
(b) 2  
(c) 16  
(d) 64

2. Que vaut x après l'appel `x = 2 ; x = h(x) ?`

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 2  
(b) 16  
(c) 32  
(d) 64

3. Que vaut x après l'appel `x = f(15,21) ?`

```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```

- (a) 35  
(b) 7  
(c) 5  
(d) 3

4. Un appel récursif est un appel \_\_\_\_\_

- (a) dont le résultat est retourné par la fonction

- (b) dont l'exécution est un processus récursif
- (c) d'une fonction par elle-même
- (d) dont l'exécution est un processus itératif

5. Que vaut  $x$  après l'appel  $t = [2,4,6]$  ;  $f(t)$  ?

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```

- (a)  $[0,4,6]$
- (b)  $[2,4,6]$
- (c)  $[2,4,0]$
- (d)  $[2,0,6]$

6. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la référence du paramètre formel dans le paramètre effectif correspondant
- (b) la valeur du paramètre formel dans le paramètre effectif correspondant
- (c) la référence du paramètre effectif dans le paramètre formel correspondant
- (d) la valeur du paramètre effectif dans le paramètre formel correspondant

7. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre effectif dans le paramètre formel correspondant
- (b) la référence du paramètre effectif dans le paramètre formel correspondant
- (c) la valeur du paramètre formel dans le paramètre effectif correspondant
- (d) la référence du paramètre formel dans le paramètre effectif correspondant

8. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) la fonction doit vérifier impérativement ses préconditions
- (b) la fonction doit être correctement paramétrée
- (c) l'utilisation de la fonction doit être conviviale
- (d) l'implémentation de la fonction doit être conforme aux jeux de tests

Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut x après l'appel `x = 2 ; y = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 32
- (b) 16
- (c) 64
- (d) 2

☐  
☐  
☐  
☐

2. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre formel dans le paramètre effectif correspondant
- (b) la référence du paramètre formel dans le paramètre effectif correspondant
- (c) la valeur du paramètre effectif dans le paramètre formel correspondant
- (d) la référence du paramètre effectif dans le paramètre formel correspondant

☐  
☐  
☐  
☐

3. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) l'implémentation de la fonction doit être conforme aux jeux de tests
- (b) l'utilisation de la fonction doit être conviviale
- (c) la fonction doit être correctement paramétrée
- (d) la fonction doit vérifier impérativement ses préconditions

☐  
☐  
☐  
☐

4. Que vaut x après l'appel `x = f(15,21)` ?

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

- (a) 7
- (b) 35
- (c) 3

☐  
☐  
☐

(d) 5

5. Que vaut  $x$  après l'appel  $x = 2 ; x = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

(a) 32

(b) 16

(c) 64

(d) 2

6. Que vaut  $x$  après l'appel  $t = [2,4,6] ; f(t)$  ?

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```

(a) [2,0,6]

(b) [0,4,6]

(c) [2,4,0]

(d) [2,4,6]

7. Un appel récursif est un appel \_\_\_\_\_

(a) dont l'exécution est un processus itératif

(b) dont l'exécution est un processus récursif

(c) d'une fonction par elle-même

(d) dont le résultat est retourné par la fonction

8. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

(a) la valeur du paramètre formel dans le paramètre effectif correspondant

(b) la valeur du paramètre effectif dans le paramètre formel correspondant

(c) la référence du paramètre formel dans le paramètre effectif correspondant

(d) la référence du paramètre effectif dans le paramètre formel correspondant

Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut x après l'appel  $t = [2,4,6]$  ;  $f(t)$  ?

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a) [2,4,6]
- (b) [2,0,6]
- (c) [2,4,0]
- (d) [0,4,6]

2. Que vaut x après l'appel  $x = f(15,21)$  ?

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

- (a) 35
- (b) 3
- (c) 5
- (d) 7

3. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la référence du paramètre formel dans le paramètre effectif correspondant
- (b) la valeur du paramètre effectif dans le paramètre formel correspondant
- (c) la référence du paramètre effectif dans le paramètre formel correspondant
- (d) la valeur du paramètre formel dans le paramètre effectif correspondant

4. Un appel récursif est un appel \_\_\_\_\_

- (a) dont l'exécution est un processus récursif
- (b) d'une fonction par elle-même
- (c) dont le résultat est retourné par la fonction
- (d) dont l'exécution est un processus itératif

5. Que vaut  $x$  après l'appel  $x = 2 ; x = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 32
- (b) 2
- (c) 64
- (d) 16

6. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) la fonction doit vérifier impérativement ses préconditions
- (b) la fonction doit être correctement paramétrée
- (c) l'utilisation de la fonction doit être conviviale
- (d) l'implémentation de la fonction doit être conforme aux jeux de tests

7. Que vaut  $x$  après l'appel  $x = 2 ; y = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 32
- (b) 16
- (c) 64
- (d) 2

8. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la référence du paramètre effectif dans le paramètre formel correspondant
- (b) la valeur du paramètre formel dans le paramètre effectif correspondant
- (c) la valeur du paramètre effectif dans le paramètre formel correspondant
- (d) la référence du paramètre formel dans le paramètre effectif correspondant

Nom :	Prénom :	Groupe :	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
-------	----------	----------	----------	----------	----------	----------

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Un appel récursif est un appel \_\_\_\_\_
  - (a) dont l'exécution est un processus itératif ☐
  - (b) dont le résultat est retourné par la fonction ☐
  - (c) d'une fonction par elle-même ☐
  - (d) dont l'exécution est un processus récursif ☐
2. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_
  - (a) la fonction doit être correctement paramétrée ☐
  - (b) l'utilisation de la fonction doit être conviviale ☐
  - (c) la fonction doit vérifier impérativement ses préconditions ☐
  - (d) l'implémentation de la fonction doit être conforme aux jeux de tests ☐
3. Que vaut x après l'appel  $t = [2,4,6]$  ;  $f(t)$  ?
 

```
def f(t) :
    t[len(t)-1] = 0
    return
```

 \_\_\_\_\_
  - (a)  $[2,4,6]$  ☐
  - (b)  $[2,0,6]$  ☐
  - (c)  $[0,4,6]$  ☐
  - (d)  $[2,4,0]$  ☐
4. Le passage des paramètres par référence consiste à copier \_\_\_\_\_
  - (a) la référence du paramètre formel dans le paramètre effectif correspondant ☐
  - (b) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
  - (c) la référence du paramètre effectif dans le paramètre formel correspondant ☐
  - (d) la valeur du paramètre effectif dans le paramètre formel correspondant ☐

5. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la référence du paramètre formel dans le paramètre effectif correspondant
- (b) la valeur du paramètre formel dans le paramètre effectif correspondant
- (c) la référence du paramètre effectif dans le paramètre formel correspondant
- (d) la valeur du paramètre effectif dans le paramètre formel correspondant

6. Que vaut x après l'appel `x = 2 ; y = h(x)` ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 2
- (b) 64
- (c) 32
- (d) 16

7. Que vaut x après l'appel `x = f(15,21)` ?

```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```

- (a) 35
- (b) 3
- (c) 7
- (d) 5

8. Que vaut x après l'appel `x = 2 ; x = h(x)` ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 32
- (b) 16
- (c) 64
- (d) 2



Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) l'utilisation de la fonction doit être conviviale ☐
- (b) la fonction doit être correctement paramétrée ☐
- (c) l'implémentation de la fonction doit être conforme aux jeux de tests ☐
- (d) la fonction doit vérifier impérativement ses préconditions ☐

2. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
- (b) la référence du paramètre effectif dans le paramètre formel correspondant ☐
- (c) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
- (d) la référence du paramètre formel dans le paramètre effectif correspondant ☐

3. Que vaut x après l'appel  $x = 2 ; y = h(x)$  ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 32 ☐
- (b) 2 ☐
- (c) 64 ☐
- (d) 16 ☐

4. Que vaut x après l'appel  $t = [2,4,6] ; f(t)$  ?

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a) [2,4,0] ☐
- (b) [0,4,6] ☐
- (c) [2,4,6] ☐

(d) [2,0,6]

5. Un appel récursif est un appel \_\_\_\_\_

- (a) dont l'exécution est un processus récursif
- (b) dont le résultat est retourné par la fonction
- (c) dont l'exécution est un processus itératif
- (d) d'une fonction par elle-même

6. Que vaut  $x$  après l'appel  $x = f(15,21)$  ?

```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```

- (a) 35
- (b) 5
- (c) 3
- (d) 7

7. Que vaut  $x$  après l'appel  $x = 2 ; x = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 16
- (b) 32
- (c) 2
- (d) 64

8. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la référence du paramètre effectif dans le paramètre formel correspondant
- (b) la valeur du paramètre effectif dans le paramètre formel correspondant
- (c) la valeur du paramètre formel dans le paramètre effectif correspondant
- (d) la référence du paramètre formel dans le paramètre effectif correspondant

Nom :	Prénom :	Groupe :	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
-------	----------	----------	----------	----------	----------	----------

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut  $x$  après l'appel  $x = 2 ; x = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 16
- (b) 32
- (c) 64
- (d) 2

2. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la référence du paramètre effectif dans le paramètre formel correspondant
- (b) la référence du paramètre formel dans le paramètre effectif correspondant
- (c) la valeur du paramètre effectif dans le paramètre formel correspondant
- (d) la valeur du paramètre formel dans le paramètre effectif correspondant

3. Un appel récursif est un appel \_\_\_\_\_

- (a) dont l'exécution est un processus récursif
- (b) d'une fonction par elle-même
- (c) dont l'exécution est un processus itératif
- (d) dont le résultat est retourné par la fonction

4. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) la fonction doit être correctement paramétrée
- (b) la fonction doit vérifier impérativement ses préconditions
- (c) l'utilisation de la fonction doit être conviviale
- (d) l'implémentation de la fonction doit être conforme aux jeux de tests

5. Que vaut  $x$  après l'appel  $t = [2,4,6]$  ;  $f(t)$  ?

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```

- (a)  $[2,4,6]$
- (b)  $[0,4,6]$
- (c)  $[2,0,6]$
- (d)  $[2,4,0]$

6. Que vaut  $x$  après l'appel  $x = f(15,21)$  ?

```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```

- (a) 7
- (b) 3
- (c) 5
- (d) 35

7. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la référence du paramètre formel dans le paramètre effectif correspondant
- (b) la référence du paramètre effectif dans le paramètre formel correspondant
- (c) la valeur du paramètre effectif dans le paramètre formel correspondant
- (d) la valeur du paramètre formel dans le paramètre effectif correspondant

8. Que vaut  $x$  après l'appel  $x = 2$  ;  $y = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 32
- (b) 2
- (c) 16
- (d) 64

Nom :	Prénom :	Groupe :	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
-------	----------	----------	----------	----------	----------	----------

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Le passage des paramètres par référence consiste à copier \_\_\_\_\_
  - (a) la référence du paramètre formel dans le paramètre effectif correspondant ☐
  - (b) la référence du paramètre effectif dans le paramètre formel correspondant ☐
  - (c) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
  - (d) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
  
2. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_
  - (a) l'implémentation de la fonction doit être conforme aux jeux de tests ☐
  - (b) la fonction doit être correctement paramétrée ☐
  - (c) l'utilisation de la fonction doit être conviviale ☐
  - (d) la fonction doit vérifier impérativement ses préconditions ☐
  
3. Un appel récursif est un appel \_\_\_\_\_
  - (a) d'une fonction par elle-même ☐
  - (b) dont l'exécution est un processus itératif ☐
  - (c) dont l'exécution est un processus récursif ☐
  - (d) dont le résultat est retourné par la fonction ☐
  
4. Que vaut x après l'appel  $t = [2,4,6]$  ;  $f(t)$  ?
 

```
def f(t) :
    t[len(t)-1] = 0
    return
```

 \_\_\_\_\_
  - (a)  $[2,0,6]$  ☐
  - (b)  $[0,4,6]$  ☐
  - (c)  $[2,4,0]$  ☐
  - (d)  $[2,4,6]$  ☐

5. Que vaut  $x$  après l'appel  $x = 2 ; x = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 16
- (b) 64
- (c) 2
- (d) 32

6. Que vaut  $x$  après l'appel  $x = f(15,21)$  ?

```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```

- (a) 5
- (b) 3
- (c) 7
- (d) 35

7. Que vaut  $x$  après l'appel  $x = 2 ; y = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 16
- (b) 64
- (c) 32
- (d) 2

8. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre formel dans le paramètre effectif correspondant
- (b) la référence du paramètre formel dans le paramètre effectif correspondant
- (c) la référence du paramètre effectif dans le paramètre formel correspondant
- (d) la valeur du paramètre effectif dans le paramètre formel correspondant

Nom :	Prénom :	Groupe :	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
-------	----------	----------	----------	----------	----------	----------

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la référence du paramètre effectif dans le paramètre formel correspondant ☐
- (b) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
- (c) la référence du paramètre formel dans le paramètre effectif correspondant ☐
- (d) la valeur du paramètre effectif dans le paramètre formel correspondant ☐

2. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) la fonction doit vérifier impérativement ses préconditions ☐
- (b) la fonction doit être correctement paramétrée ☐
- (c) l'implémentation de la fonction doit être conforme aux jeux de tests ☐
- (d) l'utilisation de la fonction doit être conviviale ☐

3. Que vaut x après l'appel `x = 2 ; y = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 32 ☐
- (b) 64 ☐
- (c) 2 ☐
- (d) 16 ☐

4. Un appel récursif est un appel \_\_\_\_\_

- (a) d'une fonction par elle-même ☐
- (b) dont l'exécution est un processus itératif ☐
- (c) dont l'exécution est un processus récursif ☐
- (d) dont le résultat est retourné par la fonction ☐

5. Que vaut  $x$  après l'appel  $x = f(15, 21)$  ?

```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```

- (a) 3
- (b) 7
- (c) 35
- (d) 5

6. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la référence du paramètre effectif dans le paramètre formel correspondant
- (b) la valeur du paramètre formel dans le paramètre effectif correspondant
- (c) la référence du paramètre formel dans le paramètre effectif correspondant
- (d) la valeur du paramètre effectif dans le paramètre formel correspondant

7. Que vaut  $x$  après l'appel  $t = [2, 4, 6]$  ;  $f(t)$  ?

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```

- (a) [0, 4, 6]
- (b) [2, 4, 0]
- (c) [2, 4, 6]
- (d) [2, 0, 6]

8. Que vaut  $x$  après l'appel  $x = 2$  ;  $x = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 32
- (b) 64
- (c) 16
- (d) 2



Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut x après l'appel `x = 2 ; y = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 32
- (b) 16
- (c) 64
- (d) 2

2. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre formel dans le paramètre effectif correspondant
- (b) la référence du paramètre formel dans le paramètre effectif correspondant
- (c) la valeur du paramètre effectif dans le paramètre formel correspondant
- (d) la référence du paramètre effectif dans le paramètre formel correspondant

3. Que vaut x après l'appel `t = [2,4,6] ; f(t)` ?

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a) [2,4,0]
- (b) [2,0,6]
- (c) [2,4,6]
- (d) [0,4,6]

4. Un appel récursif est un appel \_\_\_\_\_

- (a) d'une fonction par elle-même
- (b) dont l'exécution est un processus itératif
- (c) dont l'exécution est un processus récursif

(d) dont le résultat est retourné par la fonction

5. Que vaut  $x$  après l'appel  $x = 2 ; x = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 16
- (b) 32
- (c) 2
- (d) 64

6. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la référence du paramètre formel dans le paramètre effectif correspondant
- (b) la valeur du paramètre formel dans le paramètre effectif correspondant
- (c) la valeur du paramètre effectif dans le paramètre formel correspondant
- (d) la référence du paramètre effectif dans le paramètre formel correspondant

7. Que vaut  $x$  après l'appel  $x = f(15,21)$  ?

```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```

- (a) 3
- (b) 7
- (c) 35
- (d) 5

8. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) l'utilisation de la fonction doit être conviviale
- (b) la fonction doit être correctement paramétrée
- (c) l'implémentation de la fonction doit être conforme aux jeux de tests
- (d) la fonction doit vérifier impérativement ses préconditions

Nom :	Prénom :	Groupe :	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
-------	----------	----------	----------	----------	----------	----------

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
- (b) la référence du paramètre effectif dans le paramètre formel correspondant ☐
- (c) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
- (d) la référence du paramètre formel dans le paramètre effectif correspondant ☐

2. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) la fonction doit vérifier impérativement ses préconditions ☐
- (b) la fonction doit être correctement paramétrée ☐
- (c) l'utilisation de la fonction doit être conviviale ☐
- (d) l'implémentation de la fonction doit être conforme aux jeux de tests ☐

3. Que vaut x après l'appel `x = 2 ; x = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 16 ☐
- (b) 32 ☐
- (c) 64 ☐
- (d) 2 ☐

4. Que vaut x après l'appel `t = [2,4,6] ; f(t)` ?

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a) [2,0,6] ☐
- (b) [0,4,6] ☐
- (c) [2,4,0] ☐

(d) [2,4,6]

5. Que vaut x après l'appel  $x = f(15,21)$  ?

```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```

- (a) 5
- (b) 3
- (c) 35
- (d) 7

6. Un appel récursif est un appel \_\_\_\_\_

- (a) dont l'exécution est un processus récursif
- (b) dont l'exécution est un processus itératif
- (c) dont le résultat est retourné par la fonction
- (d) d'une fonction par elle-même

7. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre effectif dans le paramètre formel correspondant
- (b) la valeur du paramètre formel dans le paramètre effectif correspondant
- (c) la référence du paramètre formel dans le paramètre effectif correspondant
- (d) la référence du paramètre effectif dans le paramètre formel correspondant

8. Que vaut x après l'appel  $x = 2 ; y = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 32
- (b) 2
- (c) 64
- (d) 16

Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
- (b) la référence du paramètre effectif dans le paramètre formel correspondant ☐
- (c) la référence du paramètre formel dans le paramètre effectif correspondant ☐
- (d) la valeur du paramètre formel dans le paramètre effectif correspondant ☐

2. Que vaut x après l'appel `x = 2 ; x = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 64 ☐
- (b) 32 ☐
- (c) 16 ☐
- (d) 2 ☐

3. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) l'implémentation de la fonction doit être conforme aux jeux de tests ☐
- (b) la fonction doit être correctement paramétrée ☐
- (c) l'utilisation de la fonction doit être conviviale ☐
- (d) la fonction doit vérifier impérativement ses préconditions ☐

4. Que vaut x après l'appel `x = 2 ; y = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 64 ☐
- (b) 16 ☐

(c) 32

(d) 2

5. Un appel récursif est un appel \_\_\_\_\_

(a) d'une fonction par elle-même

(b) dont l'exécution est un processus récursif

(c) dont le résultat est retourné par la fonction

(d) dont l'exécution est un processus itératif

6. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

(a) la référence du paramètre effectif dans le paramètre formel correspondant

(b) la référence du paramètre formel dans le paramètre effectif correspondant

(c) la valeur du paramètre formel dans le paramètre effectif correspondant

(d) la valeur du paramètre effectif dans le paramètre formel correspondant

7. Que vaut x après l'appel  $t = [2, 4, 6]$  ;  $f(t)$  ?

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```

(a) [0, 4, 6]

(b) [2, 4, 0]

(c) [2, 0, 6]

(d) [2, 4, 6]

8. Que vaut x après l'appel  $x = f(15, 21)$  ?

```
def f(x, y) :  
    if y == 0 : return x  
    else : return f(y, x%y)
```

(a) 7

(b) 35

(c) 3

(d) 5

Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) la fonction doit être correctement paramétrée ☐
- (b) l'implémentation de la fonction doit être conforme aux jeux de tests ☐
- (c) l'utilisation de la fonction doit être conviviale ☐
- (d) la fonction doit vérifier impérativement ses préconditions ☐

2. Un appel récursif est un appel \_\_\_\_\_

- (a) dont l'exécution est un processus récursif ☐
- (b) d'une fonction par elle-même ☐
- (c) dont le résultat est retourné par la fonction ☐
- (d) dont l'exécution est un processus itératif ☐

3. Que vaut x après l'appel `x = 2 ; x = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 64 ☐
- (b) 16 ☐
- (c) 2 ☐
- (d) 32 ☐

4. Que vaut x après l'appel `x = 2 ; y = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 16 ☐
- (b) 64 ☐

(c) 32

(d) 2

5. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

(a) la valeur du paramètre effectif dans le paramètre formel correspondant

(b) la valeur du paramètre formel dans le paramètre effectif correspondant

(c) la référence du paramètre effectif dans le paramètre formel correspondant

(d) la référence du paramètre formel dans le paramètre effectif correspondant

6. Que vaut x après l'appel  $t = [2,4,6]$  ;  $f(t)$  ?

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```

(a)  $[2,4,6]$

(b)  $[2,4,0]$

(c)  $[2,0,6]$

(d)  $[0,4,6]$

7. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

(a) la valeur du paramètre effectif dans le paramètre formel correspondant

(b) la valeur du paramètre formel dans le paramètre effectif correspondant

(c) la référence du paramètre effectif dans le paramètre formel correspondant

(d) la référence du paramètre formel dans le paramètre effectif correspondant

8. Que vaut x après l'appel  $x = f(15,21)$  ?

```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```

(a) 7

(b) 35

(c) 5

(d) 3



Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut x après l'appel `x = f(15,21)` ?

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

- (a) 3
- (b) 5
- (c) 7
- (d) 35

2. Que vaut x après l'appel `x = 2 ; y = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 32
- (b) 16
- (c) 2
- (d) 64

3. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre effectif dans le paramètre formel correspondant
- (b) la référence du paramètre formel dans le paramètre effectif correspondant
- (c) la référence du paramètre effectif dans le paramètre formel correspondant
- (d) la valeur du paramètre formel dans le paramètre effectif correspondant

4. Que vaut x après l'appel `x = 2 ; x = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 2

(b) 64

(c) 32

(d) 16

5. Que vaut  $x$  après l'appel  $t = [2,4,6]$  ;  $f(t)$  ?

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```

(a)  $[2,0,6]$

(b)  $[0,4,6]$

(c)  $[2,4,0]$

(d)  $[2,4,6]$

6. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

(a) la fonction doit être correctement paramétrée

(b) la fonction doit vérifier impérativement ses préconditions

(c) l'implémentation de la fonction doit être conforme aux jeux de tests

(d) l'utilisation de la fonction doit être conviviale

7. Un appel récursif est un appel \_\_\_\_\_

(a) d'une fonction par elle-même

(b) dont l'exécution est un processus itératif

(c) dont le résultat est retourné par la fonction

(d) dont l'exécution est un processus récursif

8. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

(a) la valeur du paramètre effectif dans le paramètre formel correspondant

(b) la référence du paramètre formel dans le paramètre effectif correspondant

(c) la valeur du paramètre formel dans le paramètre effectif correspondant

(d) la référence du paramètre effectif dans le paramètre formel correspondant

Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_
  - (a) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
  - (b) la référence du paramètre formel dans le paramètre effectif correspondant ☐
  - (c) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
  - (d) la référence du paramètre effectif dans le paramètre formel correspondant ☐
2. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_
  - (a) l'utilisation de la fonction doit être conviviale ☐
  - (b) la fonction doit être correctement paramétrée ☐
  - (c) la fonction doit vérifier impérativement ses préconditions ☐
  - (d) l'implémentation de la fonction doit être conforme aux jeux de tests ☐
3. Le passage des paramètres par référence consiste à copier \_\_\_\_\_
  - (a) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
  - (b) la référence du paramètre effectif dans le paramètre formel correspondant ☐
  - (c) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
  - (d) la référence du paramètre formel dans le paramètre effectif correspondant ☐
4. Un appel récursif est un appel \_\_\_\_\_
  - (a) dont l'exécution est un processus itératif ☐
  - (b) dont le résultat est retourné par la fonction ☐
  - (c) dont l'exécution est un processus récursif ☐
  - (d) d'une fonction par elle-même ☐

5. Que vaut x après l'appel  $t = [2, 4, 6]$  ;  $f(t)$  ? \_\_\_\_\_

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a) [2,4,0]
- (b) [2,4,6]
- (c) [2,0,6]
- (d) [0,4,6]

6. Que vaut x après l'appel `x = 2 ; x = h(x)` ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 2
- (b) 32
- (c) 64
- (d) 16

7. Que vaut x après l'appel `x = 2 ; y = h(x)` ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 32
- (b) 2
- (c) 64
- (d) 16

8. Que vaut x après l'appel `x = f(15,21)` ?

```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```

- (a) 3
- (b) 5
- (c) 35
- (d) 7

Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Un appel récursif est un appel \_\_\_\_\_
  - (a) dont l'exécution est un processus récursif ☐
  - (b) d'une fonction par elle-même ☐
  - (c) dont l'exécution est un processus itératif ☐
  - (d) dont le résultat est retourné par la fonction ☐
2. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_
  - (a) la référence du paramètre effectif dans le paramètre formel correspondant ☐
  - (b) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
  - (c) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
  - (d) la référence du paramètre formel dans le paramètre effectif correspondant ☐
3. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_
  - (a) l'implémentation de la fonction doit être conforme aux jeux de tests ☐
  - (b) la fonction doit être correctement paramétrée ☐
  - (c) l'utilisation de la fonction doit être conviviale ☐
  - (d) la fonction doit vérifier impérativement ses préconditions ☐
4. Le passage des paramètres par référence consiste à copier \_\_\_\_\_
  - (a) la référence du paramètre effectif dans le paramètre formel correspondant ☐
  - (b) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
  - (c) la référence du paramètre formel dans le paramètre effectif correspondant ☐
  - (d) la valeur du paramètre effectif dans le paramètre formel correspondant ☐

5. Que vaut x après l'appel `t = [2,4,6] ; f(t) ?` \_\_\_\_\_

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a) [2,4,0]
- (b) [0,4,6]
- (c) [2,4,6]
- (d) [2,0,6]

6. Que vaut  $x$  après l'appel  $x = f(15,21)$  ?

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

- (a) 7
- (b) 3
- (c) 35
- (d) 5

7. Que vaut  $x$  après l'appel  $x = 2 ; x = h(x)$  ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 64
- (b) 2
- (c) 32
- (d) 16

8. Que vaut  $x$  après l'appel  $x = 2 ; y = h(x)$  ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 64
- (b) 32
- (c) 2
- (d) 16

Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Un appel récursif est un appel \_\_\_\_\_

- (a) dont l'exécution est un processus récursif ☐
- (b) d'une fonction par elle-même ☐
- (c) dont l'exécution est un processus itératif ☐
- (d) dont le résultat est retourné par la fonction ☐

2. Que vaut x après l'appel  $t = [2,4,6]$  ;  $f(t)$  ?

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a)  $[2,4,0]$  ☐
- (b)  $[2,0,6]$  ☐
- (c)  $[2,4,6]$  ☐
- (d)  $[0,4,6]$  ☐

3. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
- (b) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
- (c) la référence du paramètre effectif dans le paramètre formel correspondant ☐
- (d) la référence du paramètre formel dans le paramètre effectif correspondant ☐

4. Que vaut x après l'appel  $x = 2$  ;  $x = h(x)$  ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 64 ☐
- (b) 16 ☐
- (c) 2 ☐

(d) 32

5. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la référence du paramètre formel dans le paramètre effectif correspondant
- (b) la valeur du paramètre effectif dans le paramètre formel correspondant
- (c) la référence du paramètre effectif dans le paramètre formel correspondant
- (d) la valeur du paramètre formel dans le paramètre effectif correspondant

6. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) l'implémentation de la fonction doit être conforme aux jeux de tests
- (b) la fonction doit vérifier impérativement ses préconditions
- (c) l'utilisation de la fonction doit être conviviale
- (d) la fonction doit être correctement paramétrée

7. Que vaut x après l'appel  $x = f(15, 21)$  ?

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

- (a) 7
- (b) 3
- (c) 35
- (d) 5

8. Que vaut x après l'appel  $x = 2 ; y = h(x)$  ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 64
- (b) 2
- (c) 32
- (d) 16



Nom :	Prénom :	Groupe :	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
-------	----------	----------	----------	----------	----------	----------

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la référence du paramètre effectif dans le paramètre formel correspondant ☐
- (b) la référence du paramètre formel dans le paramètre effectif correspondant ☐
- (c) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
- (d) la valeur du paramètre formel dans le paramètre effectif correspondant ☐

2. Que vaut x après l'appel `x = 2 ; x = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 32 ☐
- (b) 16 ☐
- (c) 2 ☐
- (d) 64 ☐

3. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) la fonction doit être correctement paramétrée ☐
- (b) l'implémentation de la fonction doit être conforme aux jeux de tests ☐
- (c) la fonction doit vérifier impérativement ses préconditions ☐
- (d) l'utilisation de la fonction doit être conviviale ☐

4. Un appel récursif est un appel \_\_\_\_\_

- (a) dont l'exécution est un processus récursif ☐
- (b) dont l'exécution est un processus itératif ☐
- (c) dont le résultat est retourné par la fonction ☐
- (d) d'une fonction par elle-même ☐

5. Que vaut  $x$  après l'appel  $x = 2 ; y = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 32
- (b) 64
- (c) 2
- (d) 16

6. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre formel dans le paramètre effectif correspondant
- (b) la référence du paramètre effectif dans le paramètre formel correspondant
- (c) la référence du paramètre formel dans le paramètre effectif correspondant
- (d) la valeur du paramètre effectif dans le paramètre formel correspondant

7. Que vaut  $x$  après l'appel  $t = [2,4,6] ; f(t)$  ?

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```

- (a) [2,4,6]
- (b) [2,4,0]
- (c) [0,4,6]
- (d) [2,0,6]

8. Que vaut  $x$  après l'appel  $x = f(15,21)$  ?

```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```

- (a) 5
- (b) 7
- (c) 35
- (d) 3

Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
- (b) la référence du paramètre effectif dans le paramètre formel correspondant ☐
- (c) la référence du paramètre formel dans le paramètre effectif correspondant ☐
- (d) la valeur du paramètre formel dans le paramètre effectif correspondant ☐

2. Que vaut x après l'appel  $x = 2 ; x = h(x)$  ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 32 ☐
- (b) 64 ☐
- (c) 2 ☐
- (d) 16 ☐

3. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la référence du paramètre effectif dans le paramètre formel correspondant ☐
- (b) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
- (c) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
- (d) la référence du paramètre formel dans le paramètre effectif correspondant ☐

4. Un appel récursif est un appel \_\_\_\_\_

- (a) dont le résultat est retourné par la fonction ☐
- (b) d'une fonction par elle-même ☐
- (c) dont l'exécution est un processus itératif ☐
- (d) dont l'exécution est un processus récursif ☐

5. Que vaut  $x$  après l'appel  $t = [2,4,6]$  ;  $f(t)$  ?

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```

- (a)  $[0,4,6]$
- (b)  $[2,4,6]$
- (c)  $[2,0,6]$
- (d)  $[2,4,0]$

6. Que vaut  $x$  après l'appel  $x = 2$  ;  $y = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 32
- (b) 64
- (c) 2
- (d) 16

7. Que vaut  $x$  après l'appel  $x = f(15,21)$  ?

```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```

- (a) 3
- (b) 7
- (c) 5
- (d) 35

8. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) la fonction doit être correctement paramétrée
- (b) l'utilisation de la fonction doit être conviviale
- (c) l'implémentation de la fonction doit être conforme aux jeux de tests
- (d) la fonction doit vérifier impérativement ses préconditions

Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut x après l'appel `x = f(15,21)` ?

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

- (a) 5
- (b) 7
- (c) 35
- (d) 3

2. Que vaut x après l'appel `x = 2 ; y = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 64
- (b) 2
- (c) 32
- (d) 16

3. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la référence du paramètre effectif dans le paramètre formel correspondant
- (b) la référence du paramètre formel dans le paramètre effectif correspondant
- (c) la valeur du paramètre formel dans le paramètre effectif correspondant
- (d) la valeur du paramètre effectif dans le paramètre formel correspondant

4. Que vaut x après l'appel `x = 2 ; x = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 64

- (b) 16 ☐
- (c) 2 ☐
- (d) 32 ☐
5. Que vaut  $x$  après l'appel  $t = [2,4,6]$  ;  $f(t)$  ? 

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```
- (a)  $[2,4,6]$  ☐
- (b)  $[2,4,0]$  ☐
- (c)  $[0,4,6]$  ☐
- (d)  $[2,0,6]$  ☐
6. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement,
- (a) l'utilisation de la fonction doit être conviviale ☐
- (b) la fonction doit être correctement paramétrée ☐
- (c) l'implémentation de la fonction doit être conforme aux jeux de tests ☐
- (d) la fonction doit vérifier impérativement ses préconditions ☐
7. Un appel récursif est un appel
- (a) dont l'exécution est un processus itératif ☐
- (b) dont l'exécution est un processus récursif ☐
- (c) d'une fonction par elle-même ☐
- (d) dont le résultat est retourné par la fonction ☐
8. Le passage des paramètres par référence consiste à copier
- (a) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
- (b) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
- (c) la référence du paramètre effectif dans le paramètre formel correspondant ☐
- (d) la référence du paramètre formel dans le paramètre effectif correspondant ☐

Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) la fonction doit vérifier impérativement ses préconditions ☐
- (b) l'utilisation de la fonction doit être conviviale ☐
- (c) la fonction doit être correctement paramétrée ☐
- (d) l'implémentation de la fonction doit être conforme aux jeux de tests ☐

2. Que vaut x après l'appel `x = 2 ; x = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 2 ☐
- (b) 32 ☐
- (c) 16 ☐
- (d) 64 ☐

3. Que vaut x après l'appel `x = f(15,21)` ?

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

- (a) 7 ☐
- (b) 5 ☐
- (c) 3 ☐
- (d) 35 ☐

4. Un appel récursif est un appel \_\_\_\_\_

- (a) d'une fonction par elle-même ☐
- (b) dont le résultat est retourné par la fonction ☐
- (c) dont l'exécution est un processus récursif ☐

(d) dont l'exécution est un processus itératif

5. Que vaut  $x$  après l'appel  $t = [2,4,6]$  ;  $f(t)$  ?

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```

(a)  $[2,4,6]$

(b)  $[2,0,6]$

(c)  $[2,4,0]$

(d)  $[0,4,6]$

6. Que vaut  $x$  après l'appel  $x = 2$  ;  $y = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

(a) 2

(b) 16

(c) 32

(d) 64

7. Le passage des paramètres par valeur consiste à copier

(a) la valeur du paramètre formel dans le paramètre effectif correspondant

(b) la valeur du paramètre effectif dans le paramètre formel correspondant

(c) la référence du paramètre effectif dans le paramètre formel correspondant

(d) la référence du paramètre formel dans le paramètre effectif correspondant

8. Le passage des paramètres par référence consiste à copier

(a) la référence du paramètre effectif dans le paramètre formel correspondant

(b) la référence du paramètre formel dans le paramètre effectif correspondant

(c) la valeur du paramètre formel dans le paramètre effectif correspondant

(d) la valeur du paramètre effectif dans le paramètre formel correspondant



Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut x après l'appel `x = 2 ; y = h(x)` ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 16
- (b) 64
- (c) 2
- (d) 32

2. Que vaut x après l'appel `t = [2,4,6] ; f(t)` ?

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```

- (a) [0,4,6]
- (b) [2,4,0]
- (c) [2,0,6]
- (d) [2,4,6]

3. Que vaut x après l'appel `x = 2 ; x = h(x)` ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 2
- (b) 64
- (c) 32
- (d) 16

4. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la référence du paramètre formel dans le paramètre effectif correspondant

- (b) la référence du paramètre effectif dans le paramètre formel correspondant ☐
- (c) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
- (d) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
5. Que vaut  $x$  après l'appel  $x = f(15, 21)$  ?
- ```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```
- (a) 5 ☐
- (b) 7 ☐
- (c) 3 ☐
- (d) 35 ☐
6. Le passage des paramètres par valeur consiste à copier
- (a) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
- (b) la référence du paramètre formel dans le paramètre effectif correspondant ☐
- (c) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
- (d) la référence du paramètre effectif dans le paramètre formel correspondant ☐
7. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement,
- (a) la fonction doit vérifier impérativement ses préconditions ☐
- (b) la fonction doit être correctement paramétrée ☐
- (c) l'utilisation de la fonction doit être conviviale ☐
- (d) l'implémentation de la fonction doit être conforme aux jeux de tests ☐
8. Un appel récursif est un appel
- (a) d'une fonction par elle-même ☐
- (b) dont l'exécution est un processus itératif ☐
- (c) dont l'exécution est un processus récursif ☐
- (d) dont le résultat est retourné par la fonction ☐

|       |          |          |   |   |   |   |
|-------|----------|----------|---|---|---|---|
| Nom : | Prénom : | Groupe : | 3 | 2 | 1 | 0 |
|-------|----------|----------|---|---|---|---|

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Un appel récursif est un appel \_\_\_\_\_

- (a) dont le résultat est retourné par la fonction ☐
- (b) d'une fonction par elle-même ☐
- (c) dont l'exécution est un processus récursif ☐
- (d) dont l'exécution est un processus itératif ☐

2. Que vaut x après l'appel  $x = 2 ; y = h(x)$  ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 32 ☐
- (b) 2 ☐
- (c) 64 ☐
- (d) 16 ☐

3. Que vaut x après l'appel  $t = [2,4,6] ; f(t)$  ?

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a) [2,4,0] ☐
- (b) [0,4,6] ☐
- (c) [2,4,6] ☐
- (d) [2,0,6] ☐

4. Que vaut x après l'appel  $x = f(15,21)$  ?

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

- (a) 3 ☐
- (b) 35 ☐

- (c) 7 ■
- (d) 5 ■
5. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_
- (a) la référence du paramètre effectif dans le paramètre formel correspondant ■
  - (b) la référence du paramètre formel dans le paramètre effectif correspondant ■
  - (c) la valeur du paramètre formel dans le paramètre effectif correspondant ■
  - (d) la valeur du paramètre effectif dans le paramètre formel correspondant ■
6. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_
- (a) la fonction doit être correctement paramétrée ■
  - (b) l'utilisation de la fonction doit être conviviale ■
  - (c) l'implémentation de la fonction doit être conforme aux jeux de tests ■
  - (d) la fonction doit vérifier impérativement ses préconditions ■
7. Le passage des paramètres par référence consiste à copier \_\_\_\_\_
- (a) la valeur du paramètre formel dans le paramètre effectif correspondant ■
  - (b) la valeur du paramètre effectif dans le paramètre formel correspondant ■
  - (c) la référence du paramètre formel dans le paramètre effectif correspondant ■
  - (d) la référence du paramètre effectif dans le paramètre formel correspondant ■
8. Que vaut x après l'appel `x = 2 ; x = h(x)` ?
- ```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```
- (a) 64 ■
- (b) 16 ■
- (c) 2 ■
- (d) 32 ■

Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut x après l'appel `x = 2 ; y = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 32
- (b) 64
- (c) 2
- (d) 16

2. Que vaut x après l'appel `x = f(15,21)` ?

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

- (a) 3
- (b) 5
- (c) 35
- (d) 7

3. Que vaut x après l'appel `t = [2,4,6] ; f(t)` ?

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a) [2,0,6]
- (b) [0,4,6]
- (c) [2,4,0]
- (d) [2,4,6]

4. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la référence du paramètre effectif dans le paramètre formel correspondant
- (b) la référence du paramètre formel dans le paramètre effectif correspondant

- (c) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
- (d) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
5. Un appel récursif est un appel \_\_\_\_\_ ☐
- (a) d'une fonction par elle-même ☐
- (b) dont l'exécution est un processus récursif ☐
- (c) dont le résultat est retourné par la fonction ☐
- (d) dont l'exécution est un processus itératif ☐
6. Que vaut  $x$  après l'appel  $x = 2 ; x = h(x)$  ? ☐
- ```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```
- (a) 32 ☐
- (b) 64 ☐
- (c) 2 ☐
- (d) 16 ☐
7. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_ ☐
- (a) la fonction doit vérifier impérativement ses préconditions ☐
- (b) l'implémentation de la fonction doit être conforme aux jeux de tests ☐
- (c) la fonction doit être correctement paramétrée ☐
- (d) l'utilisation de la fonction doit être conviviale ☐
8. Le passage des paramètres par référence consiste à copier \_\_\_\_\_ ☐
- (a) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
- (b) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
- (c) la référence du paramètre effectif dans le paramètre formel correspondant ☐
- (d) la référence du paramètre formel dans le paramètre effectif correspondant ☐

|       |          |          |   |   |   |   |
|-------|----------|----------|---|---|---|---|
| Nom : | Prénom : | Groupe : | 3 | 2 | 1 | 0 |
|-------|----------|----------|---|---|---|---|

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Un appel récursif est un appel \_\_\_\_\_

- (a) dont le résultat est retourné par la fonction ☐
- (b) dont l'exécution est un processus récursif ☐
- (c) d'une fonction par elle-même ☐
- (d) dont l'exécution est un processus itératif ☐

2. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) la fonction doit être correctement paramétrée ☐
- (b) la fonction doit vérifier impérativement ses préconditions ☐
- (c) l'implémentation de la fonction doit être conforme aux jeux de tests ☐
- (d) l'utilisation de la fonction doit être conviviale ☐

3. Que vaut x après l'appel  $x = f(15, 21)$  ? \_\_\_\_\_

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

- (a) 5 ☐
- (b) 35 ☐
- (c) 7 ☐
- (d) 3 ☐

4. Que vaut x après l'appel  $x = 2$  ;  $x = h(x)$  ? \_\_\_\_\_

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 32 ☐
- (b) 16 ☐
- (c) 64 ☐

(d) 2

5. Que vaut  $x$  après l'appel  $x = 2 ; y = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

(a) 64

(b) 2

(c) 16

(d) 32

6. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

(a) la valeur du paramètre effectif dans le paramètre formel correspondant

(b) la référence du paramètre formel dans le paramètre effectif correspondant

(c) la valeur du paramètre formel dans le paramètre effectif correspondant

(d) la référence du paramètre effectif dans le paramètre formel correspondant

7. Que vaut  $x$  après l'appel  $t = [2,4,6] ; f(t)$  ?

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```

(a) [0,4,6]

(b) [2,0,6]

(c) [2,4,0]

(d) [2,4,6]

8. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

(a) la référence du paramètre formel dans le paramètre effectif correspondant

(b) la valeur du paramètre effectif dans le paramètre formel correspondant

(c) la référence du paramètre effectif dans le paramètre formel correspondant

(d) la valeur du paramètre formel dans le paramètre effectif correspondant



|       |          |          |          |          |          |          |
|-------|----------|----------|----------|----------|----------|----------|
| Nom : | Prénom : | Groupe : | <b>3</b> | <b>2</b> | <b>1</b> | <b>0</b> |
|-------|----------|----------|----------|----------|----------|----------|

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_
  - (a) la référence du paramètre effectif dans le paramètre formel correspondant ☐
  - (b) la référence du paramètre formel dans le paramètre effectif correspondant ☐
  - (c) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
  - (d) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
2. Un appel récursif est un appel \_\_\_\_\_
  - (a) d'une fonction par elle-même ☐
  - (b) dont l'exécution est un processus itératif ☐
  - (c) dont l'exécution est un processus récursif ☐
  - (d) dont le résultat est retourné par la fonction ☐
3. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_
  - (a) l'utilisation de la fonction doit être conviviale ☐
  - (b) la fonction doit vérifier impérativement ses préconditions ☐
  - (c) la fonction doit être correctement paramétrée ☐
  - (d) l'implémentation de la fonction doit être conforme aux jeux de tests ☐
4. Le passage des paramètres par référence consiste à copier \_\_\_\_\_
  - (a) la référence du paramètre formel dans le paramètre effectif correspondant ☐
  - (b) la référence du paramètre effectif dans le paramètre formel correspondant ☐
  - (c) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
  - (d) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
5. Que vaut x après l'appel  $x = 2 ; y = h(x)$  ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

\_\_\_\_\_

- (a) 2
- (b) 16
- (c) 64
- (d) 32

6. Que vaut x après l'appel `x = 2 ; x = h(x)` ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 64
- (b) 32
- (c) 2
- (d) 16

7. Que vaut x après l'appel `x = f(15,21)` ?

```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```

- (a) 7
- (b) 35
- (c) 3
- (d) 5

8. Que vaut x après l'appel `t = [2,4,6] ; f(t)` ?

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```

- (a) [2,4,6]
- (b) [2,0,6]
- (c) [0,4,6]
- (d) [2,4,0]

|       |          |          |          |          |          |          |
|-------|----------|----------|----------|----------|----------|----------|
| Nom : | Prénom : | Groupe : | <b>3</b> | <b>2</b> | <b>1</b> | <b>0</b> |
|-------|----------|----------|----------|----------|----------|----------|

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut  $x$  après l'appel  $x = 2 ; y = h(x)$  ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 16
- (b) 32
- (c) 64
- (d) 2

☐  
☐  
☐  
☐

2. Un appel récursif est un appel \_\_\_\_\_

- (a) dont l'exécution est un processus récursif
- (b) d'une fonction par elle-même
- (c) dont l'exécution est un processus itératif
- (d) dont le résultat est retourné par la fonction

☐  
☐  
☐  
☐

3. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) la fonction doit vérifier impérativement ses préconditions
- (b) l'implémentation de la fonction doit être conforme aux jeux de tests
- (c) l'utilisation de la fonction doit être conviviale
- (d) la fonction doit être correctement paramétrée

☐  
☐  
☐  
☐

4. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre effectif dans le paramètre formel correspondant
- (b) la valeur du paramètre formel dans le paramètre effectif correspondant
- (c) la référence du paramètre effectif dans le paramètre formel correspondant
- (d) la référence du paramètre formel dans le paramètre effectif correspondant

☐  
☐  
☐  
☐

5. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre effectif dans le paramètre formel correspondant
- (b) la valeur du paramètre formel dans le paramètre effectif correspondant
- (c) la référence du paramètre formel dans le paramètre effectif correspondant
- (d) la référence du paramètre effectif dans le paramètre formel correspondant

6. Que vaut x après l'appel `x = 2 ; x = h(x)` ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 64
- (b) 32
- (c) 2
- (d) 16

7. Que vaut x après l'appel `t = [2,4,6] ; f(t)` ?

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```

- (a) [2,4,0]
- (b) [2,0,6]
- (c) [2,4,6]
- (d) [0,4,6]

8. Que vaut x après l'appel `x = f(15,21)` ?

```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```

- (a) 35
- (b) 5
- (c) 3
- (d) 7

|       |          |          |          |          |          |          |
|-------|----------|----------|----------|----------|----------|----------|
| Nom : | Prénom : | Groupe : | <b>3</b> | <b>2</b> | <b>1</b> | <b>0</b> |
|-------|----------|----------|----------|----------|----------|----------|

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut  $x$  après l'appel  $x = 2 ; x = h(x)$  ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 2
- (b) 32
- (c) 16
- (d) 64

☐  
☐  
☐  
☐

2. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre effectif dans le paramètre formel correspondant
- (b) la référence du paramètre formel dans le paramètre effectif correspondant
- (c) la valeur du paramètre formel dans le paramètre effectif correspondant
- (d) la référence du paramètre effectif dans le paramètre formel correspondant

☐  
☐  
☐  
☐

3. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) l'implémentation de la fonction doit être conforme aux jeux de tests
- (b) la fonction doit être correctement paramétrée
- (c) la fonction doit vérifier impérativement ses préconditions
- (d) l'utilisation de la fonction doit être conviviale

☐  
☐  
☐  
☐

4. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la référence du paramètre effectif dans le paramètre formel correspondant
- (b) la référence du paramètre formel dans le paramètre effectif correspondant
- (c) la valeur du paramètre effectif dans le paramètre formel correspondant
- (d) la valeur du paramètre formel dans le paramètre effectif correspondant

☐  
☐  
☐  
☐

5. Un appel récursif est un appel \_\_\_\_\_

- (a) dont l'exécution est un processus récursif
- (b) dont le résultat est retourné par la fonction
- (c) dont l'exécution est un processus itératif
- (d) d'une fonction par elle-même

6. Que vaut x après l'appel `t = [2,4,6] ; f(t) ?`

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a) `[2,4,6]`
- (b) `[2,4,0]`
- (c) `[0,4,6]`
- (d) `[2,0,6]`

7. Que vaut x après l'appel `x = f(15,21) ?`

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

- (a) 3
- (b) 5
- (c) 35
- (d) 7

8. Que vaut x après l'appel `x = 2 ; y = h(x) ?`

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 32
- (b) 64
- (c) 2
- (d) 16

|       |          |          |   |   |   |   |
|-------|----------|----------|---|---|---|---|
| Nom : | Prénom : | Groupe : | 3 | 2 | 1 | 0 |
|-------|----------|----------|---|---|---|---|

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut x après l'appel  $t = [2,4,6]$  ;  $f(t)$  ?

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a) [0,4,6]
- (b) [2,0,6]
- (c) [2,4,6]
- (d) [2,4,0]

☐  
☐  
☐  
☐

2. Que vaut x après l'appel  $x = f(15,21)$  ?

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

- (a) 3
- (b) 7
- (c) 35
- (d) 5

☐  
☐  
☐  
☐

3. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre formel dans le paramètre effectif correspondant
- (b) la référence du paramètre formel dans le paramètre effectif correspondant
- (c) la valeur du paramètre effectif dans le paramètre formel correspondant
- (d) la référence du paramètre effectif dans le paramètre formel correspondant

☐  
☐  
☐  
☐

4. Que vaut x après l'appel  $x = 2$  ;  $y = h(x)$  ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 16
- (b) 2

☐  
☐

- (c) 64 ■  
 (d) 32 ■
5. Un appel récursif est un appel \_\_\_\_\_  
 (a) dont l'exécution est un processus itératif ■  
 (b) dont l'exécution est un processus récursif ■  
 (c) d'une fonction par elle-même ■  
 (d) dont le résultat est retourné par la fonction ■
6. Que vaut  $x$  après l'appel  $x = 2 ; x = h(x)$  ?
- ```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```
- (a) 2 ■  
 (b) 64 ■  
 (c) 32 ■  
 (d) 16 ■
7. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_  
 (a) la fonction doit vérifier impérativement ses préconditions ■  
 (b) la fonction doit être correctement paramétrée ■  
 (c) l'utilisation de la fonction doit être conviviale ■  
 (d) l'implémentation de la fonction doit être conforme aux jeux de tests ■
8. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_  
 (a) la référence du paramètre effectif dans le paramètre formel correspondant ■  
 (b) la référence du paramètre formel dans le paramètre effectif correspondant ■  
 (c) la valeur du paramètre formel dans le paramètre effectif correspondant ■  
 (d) la valeur du paramètre effectif dans le paramètre formel correspondant ■



Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
- (b) la référence du paramètre effectif dans le paramètre formel correspondant ☐
- (c) la référence du paramètre formel dans le paramètre effectif correspondant ☐
- (d) la valeur du paramètre effectif dans le paramètre formel correspondant ☐

2. Que vaut  $x$  après l'appel  $x = f(15, 21)$  ? \_\_\_\_\_

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

- (a) 35 ☐
- (b) 3 ☐
- (c) 5 ☐
- (d) 7 ☐

3. Un appel récursif est un appel \_\_\_\_\_

- (a) d'une fonction par elle-même ☐
- (b) dont le résultat est retourné par la fonction ☐
- (c) dont l'exécution est un processus récursif ☐
- (d) dont l'exécution est un processus itératif ☐

4. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
- (b) la référence du paramètre effectif dans le paramètre formel correspondant ☐
- (c) la référence du paramètre formel dans le paramètre effectif correspondant ☐
- (d) la valeur du paramètre formel dans le paramètre effectif correspondant ☐

5. Que vaut  $x$  après l'appel  $x = 2 ; x = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 2
- (b) 16
- (c) 64
- (d) 32

6. Que vaut  $x$  après l'appel  $x = 2 ; y = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 16
- (b) 64
- (c) 2
- (d) 32

7. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) l'implémentation de la fonction doit être conforme aux jeux de tests
- (b) la fonction doit vérifier impérativement ses préconditions
- (c) l'utilisation de la fonction doit être conviviale
- (d) la fonction doit être correctement paramétrée

8. Que vaut  $x$  après l'appel  $t = [2,4,6] ; f(t)$  ?

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```

- (a) [2,0,6]
- (b) [2,4,0]
- (c) [2,4,6]
- (d) [0,4,6]

Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut x après l'appel `t = [2,4,6]` ; `f(t)` ?

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a) `[2,4,6]`
- (b) `[2,0,6]`
- (c) `[2,4,0]`
- (d) `[0,4,6]`

☐  
☐  
☐  
☐

2. Que vaut x après l'appel `x = 2` ; `y = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 16
- (b) 64
- (c) 2
- (d) 32

☐  
☐  
☐  
☐

3. Que vaut x après l'appel `x = 2` ; `x = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 2
- (b) 64
- (c) 16
- (d) 32

☐  
☐  
☐  
☐

4. Que vaut x après l'appel `x = f(15,21)` ?

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

☐

- (a) 7 ☐
  - (b) 3 ☐
  - (c) 5 ☐
  - (d) 35 ☐
5. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_
- (a) la fonction doit vérifier impérativement ses préconditions ☐
  - (b) la fonction doit être correctement paramétrée ☐
  - (c) l'utilisation de la fonction doit être conviviale ☐
  - (d) l'implémentation de la fonction doit être conforme aux jeux de tests ☐
6. Un appel récursif est un appel \_\_\_\_\_
- (a) d'une fonction par elle-même ☐
  - (b) dont l'exécution est un processus itératif ☐
  - (c) dont le résultat est retourné par la fonction ☐
  - (d) dont l'exécution est un processus récursif ☐
7. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_
- (a) la référence du paramètre effectif dans le paramètre formel correspondant ☐
  - (b) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
  - (c) la référence du paramètre formel dans le paramètre effectif correspondant ☐
  - (d) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
8. Le passage des paramètres par référence consiste à copier \_\_\_\_\_
- (a) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
  - (b) la référence du paramètre formel dans le paramètre effectif correspondant ☐
  - (c) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
  - (d) la référence du paramètre effectif dans le paramètre formel correspondant ☐

Nom :	Prénom :	Groupe :	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
-------	----------	----------	----------	----------	----------	----------

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut x après l'appel  $t = [2,4,6]$  ;  $f(t)$  ?

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a) [2,4,6]
- (b) [2,0,6]
- (c) [2,4,0]
- (d) [0,4,6]

☐  
☐  
☐  
☐

2. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la référence du paramètre formel dans le paramètre effectif correspondant
- (b) la valeur du paramètre effectif dans le paramètre formel correspondant
- (c) la valeur du paramètre formel dans le paramètre effectif correspondant
- (d) la référence du paramètre effectif dans le paramètre formel correspondant

☐  
☐  
☐  
☐

3. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre effectif dans le paramètre formel correspondant
- (b) la référence du paramètre effectif dans le paramètre formel correspondant
- (c) la valeur du paramètre formel dans le paramètre effectif correspondant
- (d) la référence du paramètre formel dans le paramètre effectif correspondant

☐  
☐  
☐  
☐

4. Que vaut x après l'appel  $x = f(15,21)$  ?

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

- (a) 3
- (b) 5
- (c) 7
- (d) 35

☐  
☐  
☐  
☐

5. Que vaut  $x$  après l'appel  $x = 2 ; y = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 2
- (b) 64
- (c) 16
- (d) 32

6. Que vaut  $x$  après l'appel  $x = 2 ; x = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 32
- (b) 16
- (c) 64
- (d) 2

7. Un appel récursif est un appel \_\_\_\_\_

- (a) dont l'exécution est un processus itératif
- (b) dont le résultat est retourné par la fonction
- (c) d'une fonction par elle-même
- (d) dont l'exécution est un processus récursif

8. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) l'utilisation de la fonction doit être conviviale
- (b) l'implémentation de la fonction doit être conforme aux jeux de tests
- (c) la fonction doit être correctement paramétrée
- (d) la fonction doit vérifier impérativement ses préconditions

Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut x après l'appel `t = [2,4,6]` ; `f(t)` ?

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a) `[2,4,6]`
- (b) `[2,4,0]`
- (c) `[0,4,6]`
- (d) `[2,0,6]`

☐  
☐  
☐  
☐

2. Un appel récursif est un appel \_\_\_\_\_

- (a) dont l'exécution est un processus récursif
- (b) dont l'exécution est un processus itératif
- (c) dont le résultat est retourné par la fonction
- (d) d'une fonction par elle-même

☐  
☐  
☐  
☐

3. Que vaut x après l'appel `x = f(15,21)` ?

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

- (a) 5
- (b) 35
- (c) 7
- (d) 3

☐  
☐  
☐  
☐

4. Que vaut x après l'appel `x = 2` ; `y = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 64
- (b) 16

☐  
☐

- (c) 2
- (d) 32

5. Que vaut  $x$  après l'appel  $x = 2 ; x = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 2
- (b) 32
- (c) 64
- (d) 16

6. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre formel dans le paramètre effectif correspondant
- (b) la référence du paramètre effectif dans le paramètre formel correspondant
- (c) la valeur du paramètre effectif dans le paramètre formel correspondant
- (d) la référence du paramètre formel dans le paramètre effectif correspondant

7. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) l'implémentation de la fonction doit être conforme aux jeux de tests
- (b) la fonction doit être correctement paramétrée
- (c) la fonction doit vérifier impérativement ses préconditions
- (d) l'utilisation de la fonction doit être conviviale

8. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la référence du paramètre formel dans le paramètre effectif correspondant
- (b) la valeur du paramètre effectif dans le paramètre formel correspondant
- (c) la référence du paramètre effectif dans le paramètre formel correspondant
- (d) la valeur du paramètre formel dans le paramètre effectif correspondant



Nom :	Prénom :	Groupe :	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
-------	----------	----------	----------	----------	----------	----------

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut  $x$  après l'appel  $x = 2 ; x = h(x)$  ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 16
- (b) 2
- (c) 64
- (d) 32

2. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la référence du paramètre formel dans le paramètre effectif correspondant
- (b) la valeur du paramètre effectif dans le paramètre formel correspondant
- (c) la référence du paramètre effectif dans le paramètre formel correspondant
- (d) la valeur du paramètre formel dans le paramètre effectif correspondant

3. Un appel récursif est un appel \_\_\_\_\_

- (a) dont l'exécution est un processus itératif
- (b) d'une fonction par elle-même
- (c) dont l'exécution est un processus récursif
- (d) dont le résultat est retourné par la fonction

4. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la référence du paramètre formel dans le paramètre effectif correspondant
- (b) la valeur du paramètre formel dans le paramètre effectif correspondant
- (c) la valeur du paramètre effectif dans le paramètre formel correspondant
- (d) la référence du paramètre effectif dans le paramètre formel correspondant

5. Que vaut  $x$  après l'appel  $x = f(15, 21)$  ?

```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```

- (a) 3
- (b) 5
- (c) 7
- (d) 35

6. Que vaut  $x$  après l'appel  $t = [2, 4, 6]$  ;  $f(t)$  ?

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```

- (a) [2, 4, 6]
- (b) [2, 4, 0]
- (c) [0, 4, 6]
- (d) [2, 0, 6]

7. Que vaut  $x$  après l'appel  $x = 2$  ;  $y = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 64
- (b) 32
- (c) 2
- (d) 16

8. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) la fonction doit être correctement paramétrée
- (b) l'implémentation de la fonction doit être conforme aux jeux de tests
- (c) l'utilisation de la fonction doit être conviviale
- (d) la fonction doit vérifier impérativement ses préconditions

Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut x après l'appel `x = 2 ; x = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 32
- (b) 2
- (c) 64
- (d) 16

2. Que vaut x après l'appel `x = 2 ; y = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 32
- (b) 64
- (c) 2
- (d) 16

3. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre formel dans le paramètre effectif correspondant
- (b) la valeur du paramètre effectif dans le paramètre formel correspondant
- (c) la référence du paramètre effectif dans le paramètre formel correspondant
- (d) la référence du paramètre formel dans le paramètre effectif correspondant

4. Que vaut x après l'appel `t = [2,4,6] ; f(t)` ?

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a) [2,0,6]

- (b) [2,4,0] ☐
- (c) [2,4,6] ☐
- (d) [0,4,6] ☐
5. Le passage des paramètres par référence consiste à copier \_\_\_\_\_
- (a) la référence du paramètre effectif dans le paramètre formel correspondant ☐
- (b) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
- (c) la référence du paramètre formel dans le paramètre effectif correspondant ☐
- (d) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
6. Que vaut x après l'appel  $x = f(15,21)$  ? ☐
- ```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```
- (a) 35 ☐
- (b) 7 ☐
- (c) 3 ☐
- (d) 5 ☐
7. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_
- (a) l'utilisation de la fonction doit être conviviale ☐
- (b) la fonction doit être correctement paramétrée ☐
- (c) la fonction doit vérifier impérativement ses préconditions ☐
- (d) l'implémentation de la fonction doit être conforme aux jeux de tests ☐
8. Un appel récursif est un appel \_\_\_\_\_
- (a) d'une fonction par elle-même ☐
- (b) dont le résultat est retourné par la fonction ☐
- (c) dont l'exécution est un processus récursif ☐
- (d) dont l'exécution est un processus itératif ☐

|       |          |          |   |   |   |   |
|-------|----------|----------|---|---|---|---|
| Nom : | Prénom : | Groupe : | 3 | 2 | 1 | 0 |
|-------|----------|----------|---|---|---|---|

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut x après l'appel  $t = [2, 4, 6]$  ;  $f(t)$  ?

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a)  $[2, 0, 6]$
- (b)  $[0, 4, 6]$
- (c)  $[2, 4, 6]$
- (d)  $[2, 4, 0]$

2. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) l'utilisation de la fonction doit être conviviale
- (b) l'implémentation de la fonction doit être conforme aux jeux de tests
- (c) la fonction doit être correctement paramétrée
- (d) la fonction doit vérifier impérativement ses préconditions

3. Que vaut x après l'appel  $x = 2$  ;  $x = h(x)$  ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 64
- (b) 16
- (c) 2
- (d) 32

4. Que vaut x après l'appel  $x = 2$  ;  $y = h(x)$  ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 64 ☐
- (b) 16 ☐
- (c) 2 ☐
- (d) 32 ☐
5. Le passage des paramètres par référence consiste à copier \_\_\_\_\_
- (a) la référence du paramètre formel dans le paramètre effectif correspondant ☐
- (b) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
- (c) la référence du paramètre effectif dans le paramètre formel correspondant ☐
- (d) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
6. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_
- (a) la référence du paramètre formel dans le paramètre effectif correspondant ☐
- (b) la référence du paramètre effectif dans le paramètre formel correspondant ☐
- (c) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
- (d) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
7. Que vaut x après l'appel  $x = f(15, 21)$  ? ☐
- ```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```
- (a) 5 ☐
- (b) 7 ☐
- (c) 35 ☐
- (d) 3 ☐
8. Un appel récursif est un appel \_\_\_\_\_
- (a) dont l'exécution est un processus récursif ☐
- (b) dont l'exécution est un processus itératif ☐
- (c) dont le résultat est retourné par la fonction ☐
- (d) d'une fonction par elle-même ☐

Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut x après l'appel `x = f(15,21)` ?

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

- (a) 35
- (b) 7
- (c) 5
- (d) 3

☐  
☐  
☐  
☐

2. Que vaut x après l'appel `x = 2 ; y = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 64
- (b) 2
- (c) 16
- (d) 32

☐  
☐  
☐  
☐

3. Que vaut x après l'appel `t = [2,4,6] ; f(t)` ?

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a) [2,4,0]
- (b) [0,4,6]
- (c) [2,0,6]
- (d) [2,4,6]

☐  
☐  
☐  
☐

4. Un appel récursif est un appel \_\_\_\_\_

- (a) d'une fonction par elle-même
- (b) dont le résultat est retourné par la fonction

☐  
☐

- (c) dont l'exécution est un processus itératif ■
- (d) dont l'exécution est un processus récursif ■
5. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_
- (a) la référence du paramètre formel dans le paramètre effectif correspondant ■
- (b) la référence du paramètre effectif dans le paramètre formel correspondant ■
- (c) la valeur du paramètre formel dans le paramètre effectif correspondant ■
- (d) la valeur du paramètre effectif dans le paramètre formel correspondant ■
6. Que vaut  $x$  après l'appel  $x = 2 ; x = h(x)$  ?
- ```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```
- (a) 64 ■
- (b) 2 ■
- (c) 16 ■
- (d) 32 ■
7. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_
- (a) la fonction doit vérifier impérativement ses préconditions ■
- (b) la fonction doit être correctement paramétrée ■
- (c) l'implémentation de la fonction doit être conforme aux jeux de tests ■
- (d) l'utilisation de la fonction doit être conviviale ■
8. Le passage des paramètres par référence consiste à copier \_\_\_\_\_
- (a) la référence du paramètre formel dans le paramètre effectif correspondant ■
- (b) la valeur du paramètre effectif dans le paramètre formel correspondant ■
- (c) la référence du paramètre effectif dans le paramètre formel correspondant ■
- (d) la valeur du paramètre formel dans le paramètre effectif correspondant ■



|       |          |          |   |   |   |   |
|-------|----------|----------|---|---|---|---|
| Nom : | Prénom : | Groupe : | 3 | 2 | 1 | 0 |
|-------|----------|----------|---|---|---|---|

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut x après l'appel  $x = 2$  ;  $y = h(x)$  ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 32
- (b) 16
- (c) 64
- (d) 2

2. Que vaut x après l'appel  $t = [2,4,6]$  ;  $f(t)$  ?

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a) [2,4,0]
- (b) [2,0,6]
- (c) [0,4,6]
- (d) [2,4,6]

3. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la référence du paramètre effectif dans le paramètre formel correspondant
- (b) la référence du paramètre formel dans le paramètre effectif correspondant
- (c) la valeur du paramètre effectif dans le paramètre formel correspondant
- (d) la valeur du paramètre formel dans le paramètre effectif correspondant

4. Que vaut x après l'appel  $x = f(15,21)$  ?

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

- (a) 3
- (b) 35

- (c) 5 ■
- (d) 7 ■
5. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_
- (a) la fonction doit être correctement paramétrée ■
- (b) la fonction doit vérifier impérativement ses préconditions ■
- (c) l'utilisation de la fonction doit être conviviale ■
- (d) l'implémentation de la fonction doit être conforme aux jeux de tests ■
6. Que vaut  $x$  après l'appel  $x = 2 ; x = h(x)$  ?
- ```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```
- (a) 16 ■
- (b) 32 ■
- (c) 64 ■
- (d) 2 ■
7. Un appel récursif est un appel \_\_\_\_\_
- (a) dont le résultat est retourné par la fonction ■
- (b) d'une fonction par elle-même ■
- (c) dont l'exécution est un processus itératif ■
- (d) dont l'exécution est un processus récursif ■
8. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_
- (a) la référence du paramètre effectif dans le paramètre formel correspondant ■
- (b) la valeur du paramètre effectif dans le paramètre formel correspondant ■
- (c) la valeur du paramètre formel dans le paramètre effectif correspondant ■
- (d) la référence du paramètre formel dans le paramètre effectif correspondant ■

Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Un appel récursif est un appel \_\_\_\_\_

- (a) d'une fonction par elle-même ☐
- (b) dont le résultat est retourné par la fonction ☐
- (c) dont l'exécution est un processus récursif ☐
- (d) dont l'exécution est un processus itératif ☐

2. Que vaut x après l'appel  $x = f(15, 21)$  ? \_\_\_\_\_

```
def f(x,y) :
    if y == 0 : return x
    else : return f(y,x%y)
```

- (a) 35 ☐
- (b) 5 ☐
- (c) 7 ☐
- (d) 3 ☐

3. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la référence du paramètre formel dans le paramètre effectif correspondant ☐
- (b) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
- (c) la valeur du paramètre effectif dans le paramètre formel correspondant ☐
- (d) la référence du paramètre effectif dans le paramètre formel correspondant ☐

4. Que vaut x après l'appel  $x = 2$  ;  $x = h(x)$  ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 2 ☐
- (b) 64 ☐
- (c) 16 ☐

(d) 32

5. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la référence du paramètre effectif dans le paramètre formel correspondant
- (b) la valeur du paramètre formel dans le paramètre effectif correspondant
- (c) la valeur du paramètre effectif dans le paramètre formel correspondant
- (d) la référence du paramètre formel dans le paramètre effectif correspondant

6. Que vaut x après l'appel  $x = 2 ; y = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 16
- (b) 32
- (c) 2
- (d) 64

7. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) l'utilisation de la fonction doit être conviviale
- (b) la fonction doit être correctement paramétrée
- (c) l'implémentation de la fonction doit être conforme aux jeux de tests
- (d) la fonction doit vérifier impérativement ses préconditions

8. Que vaut x après l'appel  $t = [2,4,6] ; f(t)$  ?

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```

- (a) [2,0,6]
- (b) [0,4,6]
- (c) [2,4,6]
- (d) [2,4,0]

Nom :	Prénom :	Groupe :	3	2	1	0
-------	----------	----------	---	---	---	---

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut x après l'appel `x = 2 ; x = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 32
- (b) 2
- (c) 16
- (d) 64

2. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la référence du paramètre effectif dans le paramètre formel correspondant
- (b) la référence du paramètre formel dans le paramètre effectif correspondant
- (c) la valeur du paramètre formel dans le paramètre effectif correspondant
- (d) la valeur du paramètre effectif dans le paramètre formel correspondant

3. Que vaut x après l'appel `x = 2 ; y = h(x)` ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 32
- (b) 64
- (c) 2
- (d) 16

4. Que vaut x après l'appel `t = [2,4,6] ; f(t)` ?

```
def f(t) :
    t[len(t)-1] = 0
    return
```

- (a) [2,4,0]

- (b) [2,4,6] ☐
- (c) [0,4,6] ☐
- (d) [2,0,6] ☐
5. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_
- (a) la fonction doit vérifier impérativement ses préconditions ☐
- (b) l'utilisation de la fonction doit être conviviale ☐
- (c) l'implémentation de la fonction doit être conforme aux jeux de tests ☐
- (d) la fonction doit être correctement paramétrée ☐
6. Un appel récursif est un appel \_\_\_\_\_
- (a) dont l'exécution est un processus récursif ☐
- (b) dont le résultat est retourné par la fonction ☐
- (c) d'une fonction par elle-même ☐
- (d) dont l'exécution est un processus itératif ☐
7. Que vaut x après l'appel  $x = f(15, 21)$  ? ☐
- ```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```
- (a) 5 ☐
- (b) 7 ☐
- (c) 3 ☐
- (d) 35 ☐
8. Le passage des paramètres par référence consiste à copier \_\_\_\_\_
- (a) la valeur du paramètre formel dans le paramètre effectif correspondant ☐
- (b) la référence du paramètre effectif dans le paramètre formel correspondant ☐
- (c) la référence du paramètre formel dans le paramètre effectif correspondant ☐
- (d) la valeur du paramètre effectif dans le paramètre formel correspondant ☐

|       |          |          |          |          |          |          |
|-------|----------|----------|----------|----------|----------|----------|
| Nom : | Prénom : | Groupe : | <b>3</b> | <b>2</b> | <b>1</b> | <b>0</b> |
|-------|----------|----------|----------|----------|----------|----------|

DURÉE: 5'

DOCUMENTS, TÉLÉPHONES, CALCULETTES ET ORDINATEURS INTERDITS.

Pour chaque question, noircir la case ( ☐ → ☐ ) correspondant à la bonne réponse (il n'y a qu'une seule bonne réponse par question).

Pour une question donnée, la réponse est considérée comme fausse si

- aucune case n'est noircie ou
- plusieurs cases sont noircies (même si la bonne réponse est noircie) ou
- une seule case est noircie mais ne correspond pas à la bonne réponse.

### QCM : Appels de fonctions

1. Que vaut  $x$  après l'appel  $x = 2 ; x = h(x)$  ?

```
def f(x) :
    x = 2*x
    return x
```

```
def g(x) :
    x = 2*f(x)
    return x
```

```
def h(x) :
    x = 2*g(f(x))
    return x
```

- (a) 2
- (b) 16
- (c) 64
- (d) 32

2. La réutilisabilité d'un algorithme est son aptitude à résoudre des tâches équivalentes à celle pour laquelle il a été conçu. Plus concrètement, \_\_\_\_\_

- (a) la fonction doit vérifier impérativement ses préconditions
- (b) la fonction doit être correctement paramétrée
- (c) l'implémentation de la fonction doit être conforme aux jeux de tests
- (d) l'utilisation de la fonction doit être conviviale

3. Le passage des paramètres par valeur consiste à copier \_\_\_\_\_

- (a) la référence du paramètre effectif dans le paramètre formel correspondant
- (b) la valeur du paramètre effectif dans le paramètre formel correspondant
- (c) la valeur du paramètre formel dans le paramètre effectif correspondant
- (d) la référence du paramètre formel dans le paramètre effectif correspondant

4. Un appel récursif est un appel \_\_\_\_\_

- (a) dont l'exécution est un processus itératif
- (b) d'une fonction par elle-même
- (c) dont le résultat est retourné par la fonction
- (d) dont l'exécution est un processus récursif

5. Que vaut  $x$  après l'appel  $t = [2,4,6]$  ;  $f(t)$  ?

```
def f(t) :  
    t[len(t)-1] = 0  
    return
```

- (a)  $[2,4,6]$
- (b)  $[0,4,6]$
- (c)  $[2,4,0]$
- (d)  $[2,0,6]$

6. Que vaut  $x$  après l'appel  $x = 2$  ;  $y = h(x)$  ?

```
def f(x) :  
    x = 2*x  
    return x
```

```
def g(x) :  
    x = 2*f(x)  
    return x
```

```
def h(x) :  
    x = 2*g(f(x))  
    return x
```

- (a) 32
- (b) 64
- (c) 2
- (d) 16

7. Le passage des paramètres par référence consiste à copier \_\_\_\_\_

- (a) la valeur du paramètre formel dans le paramètre effectif correspondant
- (b) la valeur du paramètre effectif dans le paramètre formel correspondant
- (c) la référence du paramètre formel dans le paramètre effectif correspondant
- (d) la référence du paramètre effectif dans le paramètre formel correspondant

8. Que vaut  $x$  après l'appel  $x = f(15,21)$  ?

```
def f(x,y) :  
    if y == 0 : return x  
    else : return f(y,x%y)
```

- (a) 7
- (b) 5
- (c) 3
- (d) 35