

Exemplaire numéro 0

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) L'expression suivante ...

`[10, 20, 0] * 3`

- | | |
|-------------------------------------|-------------------------------|
| a) échoue | f) a comme résultat une liste |
| b) a comme résultat 90 | g) n'a aucun résultat |
| c) a comme résultat un dictionnaire | h) a comme résultat 0 |
| d) a comme résultat True | i) a comme résultat 30 |
| e) a comme résultat False | j) a comme résultat 3 |
| | k) a comme résultat 9 |

- | | |
|---|---|
| a) affiche « a » une infinité de fois | f) affiche « bar » une fois |
| b) affiche « bar » une infinité de fois | g) affiche « foo » une fois et « bar » une fois |
| c) affiche « a » une fois | h) échoue |
| d) affiche « foo » une fois | i) affiche « foo » une infinité de fois |
| e) n'affiche rien | |

Question 2 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- | | |
|-------------------------------|---|
| a) affiche « foo » 27 fois | i) affiche « foo » 30 fois |
| b) n'affiche rien | j) affiche « foo » 20 fois |
| c) affiche « foo » 10000 fois | k) affiche « foo » 3 fois |
| d) affiche « foo » 1000 fois | l) affiche « foo » 100 fois |
| e) échoue | m) affiche « foo » une infinité de fois |
| f) affiche « foo » 6 fois | n) affiche « foo » 9 fois |
| g) affiche « foo » 10 fois | |
| h) affiche « foo » une fois | |

Question 6 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- | | |
|----------------------|-------------------|
| a) affiche « 14 » | g) affiche « 15 » |
| b) affiche « 13 » | h) échoue |
| c) affiche « 11 » | i) affiche « 10 » |
| d) boucle infiniment | j) n'affiche rien |
| e) affiche « 16 » | k) affiche « a » |
| f) affiche « b » | l) affiche « 12 » |

Question 3 (2 points) Le programme suivant « ...

```
while False:
    print ("foo")
print ("bar")
```

- | | |
|---|---|
| a) affiche juste « bar », une fois | e) échoue |
| b) affiche « foo » une infinité de fois | f) affiche « foo » une fois et « bar » une fois |
| c) affiche « bar » une infinité de fois | g) n'affiche rien |
| d) affiche juste « foo », une fois | h) affiche « foo » une fois et « bar » une infinité de fois |

Question 7 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- | | |
|---|---|
| a) affiche « bar » une infinité de fois | e) affiche « foo » une fois et « bar » une infinité de fois |
| b) affiche « foo » une infinité de fois | f) affiche juste « bar », une fois |
| c) affiche juste « foo », une fois | g) échoue |
| d) n'affiche rien | h) affiche « foo » une fois et « bar » une fois |

Question 4 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- | | |
|--------------------------|--------------------------|
| a) affiche « foo » | f) affiche « x » |
| b) affiche « (foo (7)) » | g) échoue |
| c) affiche « 7 » | h) boucle infiniment |
| d) affiche « foo (7) » | i) affiche « (foo (x)) » |
| e) n'affiche rien | |

Question 8 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- | | |
|---|----------------------------|
| a) affiche « foo » une infinité de fois | d) affiche « foo » 9 fois |
| b) échoue | e) affiche « foo » 11 fois |
| c) n'affiche rien | f) affiche « foo » 10 fois |

Question 5 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

Question 9 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- | | |
|--|--------------------------------|
| a) affiche tout nombre entier (y compris les négatifs) | f) affiche tout nombre naturel |
| b) n'affiche rien | g) échoue |
| c) affiche « i » une fois | |
| d) affiche « range (x) » une fois | |

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Exemplaire numéro 1

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- a) affiche « foo » 6 fois
- b) affiche « foo » 10 fois
- c) échoue
- d) affiche « foo » 9 fois
- e) affiche « foo » 1000 fois
- f) affiche « foo » 3 fois
- g) affiche « foo » 20 fois
- h) affiche « foo » 27 fois
- i) affiche « foo » une infinité de fois
- j) affiche « foo » 30 fois
- k) affiche « foo » 100 fois
- l) affiche « foo » 10000 fois
- m) affiche « foo » une fois
- n) n'affiche rien

Question 2 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- a) affiche juste « foo », une fois
- b) affiche « foo » une fois et « bar » une infinité de fois
- c) affiche juste « bar », une fois
- d) affiche « foo » une infinité de fois
- e) n'affiche rien
- f) échoue
- g) affiche « bar » une infinité de fois
- h) affiche « foo » une fois et « bar » une fois

Question 3 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 12 »
- b) boucle infiniment
- c) affiche « 15 »
- d) affiche « 13 »
- e) échoue
- f) affiche « b »
- g) affiche « a »
- h) n'affiche rien
- i) affiche « 10 »
- j) affiche « 16 »
- k) affiche « 11 »
- l) affiche « 14 »

Question 4 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) a comme résultat 0
- b) a comme résultat 30
- c) a comme résultat un dictionnaire
- d) a comme résultat 9
- e) a comme résultat une liste
- f) échoue
- g) n'a aucun résultat
- h) a comme résultat True
- i) a comme résultat 3
- j) a comme résultat False
- k) a comme résultat 90

Question 5 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- a) affiche « i » une fois
- b) affiche « range (x) » une fois
- c) affiche tout nombre entier (y compris les négatifs)
- d) échoue
- e) affiche tout nombre naturel
- f) n'affiche rien

Question 6 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- a) affiche « foo » une fois
- b) affiche « bar » une fois
- c) affiche « a » une infinité de fois
- d) n'affiche rien
- e) affiche « bar » une infinité de fois
- f) échoue
- g) affiche « a » une fois
- h) affiche « foo » une infinité de fois
- i) affiche « foo » une fois et « bar » une fois

Question 7 (2 points) Le programme suivant ...

```
while False:
    print ("foo")
print ("bar")
```

- a) n'affiche rien
- b) affiche « foo » une fois et « bar » une fois
- c) affiche juste « foo », une fois
- d) affiche « foo » une fois et « bar » une infinité de fois
- e) échoue
- f) affiche « bar » une infinité de fois
- g) affiche « foo » une infinité de fois
- h) affiche juste « bar », une fois

Question 8 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- a) affiche « x »
- b) n'affiche rien
- c) affiche « (foo (x)) »
- d) affiche « foo (7) »
- e) échoue
- f) affiche « (foo (7)) »
- g) boucle infiniment
- h) affiche « foo »
- i) affiche « 7 »

Question 9 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- a) échoue
- b) affiche « foo » une infinité de fois
- c) n'affiche rien
- d) affiche « foo » 9 fois
- e) affiche « foo » 11 fois
- f) affiche « foo » 10 fois

Exemplaire numéro 2

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) L'expression suivante ...

`[10, 20, 0] * 3`

- | | |
|-------------------------------|-------------------------------------|
| a) a comme résultat 90 | g) a comme résultat True |
| b) a comme résultat une liste | h) a comme résultat un dictionnaire |
| c) a comme résultat 0 | |
| d) a comme résultat 3 | i) a comme résultat False |
| e) a comme résultat 9 | j) n'a aucun résultat |
| f) échoue | k) a comme résultat 30 |

- | | |
|----------------------|--------------------------|
| a) échoue | f) affiche « (foo (x)) » |
| b) affiche « foo » | g) affiche « (foo (7)) » |
| c) boucle infiniment | h) affiche « foo (7) » |
| d) n'affiche rien | i) affiche « x » |
| e) affiche « 7 » | |

Question 2 (2 points) Le programme suivant ...

```
def foo (x):  
    for i in range (x):  
        print (i)
```

- | | | |
|--|------|--------------------------------|
| a) affiche tout nombre entier (y compris les négatifs) | fois | e) échoue |
| b) affiche « i » une fois | | f) affiche tout nombre naturel |
| c) n'affiche rien | | |
| d) affiche « range (x) » une fois | | |

Question 6 (2 points) Le programme suivant « ... »

```
while False:  
    print ("foo")  
print ("bar")
```

- | | |
|---|---|
| a) affiche juste « foo », une fois | e) affiche « bar » une infinité de fois |
| b) échoue | f) affiche « foo » une infinité de fois |
| c) affiche juste « bar », une fois | g) n'affiche rien |
| d) affiche « foo » une fois et « bar » une fois | h) affiche « foo » une fois et « bar » une infinité de fois |

Question 3 (3 points) Le programme suivant ...

```
a = 10  
for b in range (4):  
    a = a + b  
print (a)
```

- | | |
|----------------------|-------------------|
| a) boucle infiniment | g) affiche « 15 » |
| b) affiche « 14 » | h) échoue |
| c) affiche « 11 » | i) affiche « a » |
| d) affiche « 12 » | j) n'affiche rien |
| e) affiche « 13 » | k) affiche « b » |
| f) affiche « 16 » | l) affiche « 10 » |

Question 7 (2 points) Le programme suivant ...

```
a = 8  
if a > 20:  
    print ("foo")  
else a <= 20:  
    print ("bar")
```

- | | |
|---|---|
| a) affiche « bar » une infinité de fois | f) affiche « a » une fois |
| b) affiche « foo » une fois et « bar » une fois | g) n'affiche rien |
| c) échoue | h) affiche « bar » une fois |
| d) affiche « foo » une fois | i) affiche « foo » une infinité de fois |
| e) affiche « a » une infinité de fois | |

Question 4 (2 points) Le programme suivant ...

```
for i in range (10):  
    for j in range (10):  
        for k in range (10):  
            print ("foo")
```

- | | |
|-------------------------------|---|
| a) affiche « foo » 3 fois | i) n'affiche rien |
| b) affiche « foo » 100 fois | j) affiche « foo » 6 fois |
| c) affiche « foo » 9 fois | k) affiche « foo » 30 fois |
| d) affiche « foo » une fois | l) affiche « foo » 10 fois |
| e) échoue | m) affiche « foo » une infinité de fois |
| f) affiche « foo » 27 fois | |
| g) affiche « foo » 10000 fois | n) affiche « foo » 1000 fois |
| h) affiche « foo » 20 fois | |

Question 8 (2 points) Le programme suivant ...

```
a = 10  
while a > 0:  
    print ("foo")
```

- | | |
|---|----------------------------|
| a) affiche « foo » une infinité de fois | d) échoue |
| b) n'affiche rien | e) affiche « foo » 10 fois |
| c) affiche « foo » 11 fois | f) affiche « foo » 9 fois |

Question 9 (3 points) Le programme suivant ...

```
for foo in [True]:  
    print ("foo")  
print ("bar")
```

- | | |
|---|---|
| a) affiche « foo » une infinité de fois | « bar » une fois |
| b) affiche « bar » une infinité de fois | e) affiche juste « foo », une fois |
| c) affiche juste « bar », une fois | f) affiche « foo » une fois et « bar » une infinité de fois |
| d) affiche « foo » une fois et | g) n'affiche rien |
| | h) échoue |

Question 5 (2 points) Le programme suivant ...

```
def foo (x):  
    return x  
print (foo (7))
```

Exemplaire numéro 3

Programmation Python
Contrôle des connaissances : contrôle surprise numéro 1
IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant « ... »

```
while False:
    print ("foo")
print ("bar")
```

- a) affiche « foo » une fois et « bar » une infinité de fois
- b) affiche « foo » une fois et « bar » une fois
- c) n'affiche rien
- d) affiche juste « foo », une fois
- e) affiche « bar » une infinité de fois
- f) affiche juste « bar », une fois
- g) affiche « foo » une infinité de fois
- h) échoue

- a) affiche « bar » une fois
- b) affiche « a » une infinité de fois
- c) affiche « foo » une fois et « bar » une fois
- d) affiche « foo » une fois
- e) affiche « foo » une infinité de fois
- f) affiche « a » une fois
- g) n'affiche rien
- h) affiche « bar » une infinité de fois
- i) échoue

Question 2 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- a) échoue
- b) affiche « foo » 10 fois
- c) affiche « foo » 11 fois
- d) n'affiche rien
- e) affiche « foo » une infinité de fois
- f) affiche « foo » 9 fois

Question 6 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- a) affiche « foo » 10 fois
- b) affiche « foo » 3 fois
- c) affiche « foo » 100 fois
- d) échoue
- e) affiche « foo » 10000 fois
- f) affiche « foo » une infinité de fois
- g) n'affiche rien
- h) affiche « foo » 9 fois
- i) affiche « foo » 27 fois
- j) affiche « foo » une fois
- k) affiche « foo » 30 fois
- l) affiche « foo » 20 fois
- m) affiche « foo » 1000 fois
- n) affiche « foo » 6 fois

Question 3 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- a) échoue
- b) affiche « foo » une fois et « bar » une infinité de fois
- c) affiche juste « foo », une fois
- d) affiche juste « bar », une fois
- e) n'affiche rien
- f) affiche « foo » une fois et « bar » une fois
- g) affiche « bar » une infinité de fois
- h) affiche « foo » une infinité de fois

Question 7 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) échoue
- b) a comme résultat 30
- c) n'a aucun résultat
- d) a comme résultat 3
- e) a comme résultat un dictionnaire
- f) a comme résultat True
- g) a comme résultat une liste
- h) a comme résultat 9
- i) a comme résultat False
- j) a comme résultat 0
- k) a comme résultat 90

Question 4 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- a) affiche « i » une fois
- b) échoue
- c) affiche tout nombre entier (y compris les négatifs)
- d) affiche « range (x) » une fois
- e) n'affiche rien
- f) affiche tout nombre naturel

Question 8 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 15 »
- b) affiche « b »
- c) n'affiche rien
- d) affiche « 11 »
- e) boucle infiniment
- f) affiche « 14 »
- g) affiche « 16 »
- h) affiche « 12 »
- i) affiche « a »
- j) affiche « 13 »
- k) affiche « 10 »
- l) échoue

Question 5 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

Question 9 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- a) affiche « x »
- b) affiche « (foo (x)) »
- c) affiche « foo (7) »
- d) boucle infiniment
- e) échoue
- f) affiche « (foo (7)) »
- g) affiche « 7 »
- h) n'affiche rien
- i) affiche « foo »

Programmation Python
Contrôle des connaissances : contrôle surprise numéro 1
IUT de Villetaneuse, Université Paris 13
Luca Saiu

Exemplaire numéro 4

Documents (livres, slides, polycopiés, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- a) affiche « bar » une infinité de fois
- b) affiche « foo » une infinité de fois
- c) affiche « a » une infinité de fois
- d) affiche « foo » une fois
- e) affiche « bar » une fois
- f) affiche « foo » une fois et « bar » une fois
- g) échoue
- h) affiche « a » une fois
- i) n'affiche rien

Question 2 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- a) affiche tout nombre entier (y compris les négatifs)
- b) n'affiche rien
- c) affiche tout nombre naturel
- d) affiche « i » une fois
- e) échoue
- f) affiche « range (x) » une fois

Question 3 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- a) affiche « foo »
- b) échoue
- c) n'affiche rien
- d) affiche « (foo (x)) »
- e) affiche « x »
- f) affiche « (foo (7)) »
- g) affiche « 7 »
- h) affiche « foo (7) »
- i) boucle infiniment

Question 4 (2 points) L'expression suivante ...

[10, 20, 0] * 3

- a) a comme résultat False
- b) a comme résultat 30
- c) a comme résultat 90
- d) a comme résultat 9
- e) a comme résultat True
- f) échoue
- g) a comme résultat 0
- h) n'a aucun résultat
- i) a comme résultat 3
- j) a comme résultat un dictionnaire
- k) a comme résultat une liste

Question 5 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- a) affiche « foo » 9 fois
- b) échoue
- c) affiche « foo » 11 fois
- d) n'affiche rien
- e) affiche « foo » 10 fois
- f) affiche « foo » une infinité de fois

Question 6 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- a) affiche juste « bar », une fois
- b) affiche « bar » une infinité de fois
- c) n'affiche rien
- d) affiche « foo » une fois et « bar » une fois
- e) affiche « foo » une fois et « bar » une infinité de fois
- f) affiche juste « foo », une fois
- g) échoue
- h) affiche « foo » une infinité de fois

Question 7 (2 points) Le programme suivant « ...

```
while False:
    print ("foo")
print ("bar")
```

- a) affiche juste « foo », une fois
- b) échoue
- c) affiche « foo » une infinité de fois
- d) affiche juste « bar », une fois
- e) affiche « bar » une infinité de fois
- f) affiche « foo » une fois et « bar » une fois
- g) n'affiche rien
- h) affiche « foo » une fois et « bar » une infinité de fois

Question 8 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 16 »
- b) affiche « b »
- c) affiche « 15 »
- d) affiche « 11 »
- e) affiche « 14 »
- f) affiche « 10 »
- g) n'affiche rien
- h) échoue
- i) affiche « 12 »
- j) boucle infiniment
- k) affiche « a »
- l) affiche « 13 »

Question 9 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- a) affiche « foo » 10 fois
- b) affiche « foo » 100 fois
- c) affiche « foo » 27 fois
- d) affiche « foo » une infinité de fois
- e) affiche « foo » 1000 fois
- f) affiche « foo » une fois
- g) affiche « foo » 20 fois
- h) affiche « foo » 3 fois
- i) affiche « foo » 6 fois
- j) échoue
- k) n'affiche rien
- l) affiche « foo » 9 fois
- m) affiche « foo » 10000 fois
- n) affiche « foo » 30 fois

Programmation Python
Contrôle des connaissances : contrôle surprise numéro 1
IUT de Villetaneuse, Université Paris 13
Luca Saiu

Exemplaire numéro 5

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- | | |
|---|-----------------------------|
| a) affiche « foo » 3 fois | h) affiche « foo » 30 fois |
| b) affiche « foo » une infinité de fois | i) affiche « foo » une fois |
| c) affiche « foo » 10 fois | j) affiche « foo » 100 fois |
| d) échoue | k) n'affiche rien |
| e) affiche « foo » 10000 fois | l) affiche « foo » 27 fois |
| f) affiche « foo » 9 fois | m) affiche « foo » 20 fois |
| g) affiche « foo » 1000 fois | n) affiche « foo » 6 fois |

Question 2 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- | | |
|--------------------------|--------------------------|
| a) affiche « 7 » | f) affiche « foo » |
| b) n'affiche rien | g) affiche « (foo (x)) » |
| c) affiche « x » | h) échoue |
| d) boucle infiniment | i) affiche « foo (7) » |
| e) affiche « (foo (7)) » | |

Question 3 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- | | |
|---------------------------------|----------------------------|
| a) affiche « foo » 10 fois | de fois |
| b) affiche « foo » 9 fois | e) échoue |
| c) n'affiche rien | f) affiche « foo » 11 fois |
| d) affiche « foo » une infinité | |

Question 4 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- | | |
|--|-----------------------------------|
| a) n'affiche rien | d) affiche « range (x) » une fois |
| b) affiche tout nombre entier (y compris les négatifs) | e) affiche « i » une fois |
| c) affiche tout nombre naturel | f) échoue |

Question 5 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- | | |
|---|---------------------------------------|
| a) affiche « foo » une infinité de fois | e) affiche « a » une fois |
| b) affiche « foo » une fois et « bar » une fois | f) n'affiche rien |
| c) affiche « foo » une fois | g) affiche « a » une infinité de fois |
| d) affiche « bar » une infinité de fois | h) affiche « bar » une fois |
| | i) échoue |

Question 6 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- | | |
|---|---|
| a) affiche juste « bar », une fois | e) affiche « foo » une fois et « bar » une fois |
| b) affiche « foo » une infinité de fois | f) affiche « bar » une infinité de fois |
| c) échoue | g) affiche « foo » une fois et « bar » une infinité de fois |
| d) affiche juste « foo », une fois | h) n'affiche rien |

Question 7 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- | | |
|-------------------|----------------------|
| a) affiche « 14 » | g) affiche « 11 » |
| b) affiche « 13 » | h) boucle infiniment |
| c) affiche « 12 » | i) affiche « a » |
| d) affiche « 10 » | j) n'affiche rien |
| e) échoue | k) affiche « 15 » |
| f) affiche « 16 » | l) affiche « b » |

Question 8 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- | | |
|-------------------------------|-------------------------------------|
| a) a comme résultat 0 | g) a comme résultat False |
| b) a comme résultat 9 | h) a comme résultat 30 |
| c) échoue | i) n'a aucun résultat |
| d) a comme résultat une liste | j) a comme résultat un dictionnaire |
| e) a comme résultat 90 | k) a comme résultat True |
| f) a comme résultat 3 | |

Question 9 (2 points) Le programme suivant « ...

```
while False:
    print ("foo")
print ("bar")
```

- | | |
|---|---|
| a) n'affiche rien | e) échoue |
| b) affiche juste « foo », une fois | f) affiche « foo » une fois et « bar » une fois |
| c) affiche juste « bar », une fois | g) affiche « bar » une infinité de fois |
| d) affiche « foo » une infinité de fois | h) affiche « foo » une fois et « bar » une infinité de fois |

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 6

Luca Saiu

Documents (livres, slides, polycopiés, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
def foo (x):  
    for i in range (x):  
        print (i)
```

- a) affiche « range (x) » une fois
- b) affiche tout nombre entier (y compris les négatifs)
- c) n'affiche rien
- d) affiche « i » une fois
- e) affiche tout nombre naturel
- f) échoue

Question 2 (2 points) Le programme suivant ...

```
def foo (x):  
    return x  
print (foo (7))
```

- a) affiche « x »
- b) affiche « (foo (x)) »
- c) échoue
- d) affiche « foo (7) »
- e) affiche « foo »
- f) boucle infiniment
- g) affiche « 7 »
- h) affiche « (foo (7)) »
- i) n'affiche rien

Question 3 (2 points) Le programme suivant ...

```
a = 8  
if a > 20:  
    print ("foo")  
else a <= 20:  
    print ("bar")
```

- a) échoue
- b) affiche « foo » une fois
- c) affiche « a » une fois
- d) n'affiche rien
- e) affiche « bar » une fois
- f) affiche « foo » une fois et « bar » une fois
- g) affiche « bar » une infinité de fois
- h) affiche « foo » une infinité de fois
- i) affiche « a » une infinité de fois

Question 4 (2 points) Le programme suivant ...

```
a = 10  
while a > 0:  
    print ("foo")
```

- a) affiche « foo » 11 fois
- b) échoue
- c) n'affiche rien
- d) affiche « foo » 9 fois
- e) affiche « foo » 10 fois
- f) affiche « foo » une infinité de fois

Question 5 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) a comme résultat 30
- b) a comme résultat 3
- c) échoue
- d) a comme résultat False
- e) a comme résultat un dictionnaire
- f) a comme résultat une liste
- g) a comme résultat 90
- h) a comme résultat 0
- i) a comme résultat True
- j) n'a aucun résultat
- k) a comme résultat 9

Question 6 (2 points) Le programme suivant « ...

```
while False:  
    print ("foo")  
print ("bar")
```

- a) affiche « foo » une fois et « bar » une infinité de fois
- b) échoue
- c) affiche « foo » une fois et « bar » une fois
- d) affiche juste « foo », une fois
- e) affiche « bar » une infinité de fois
- f) n'affiche rien
- g) affiche « foo » une infinité de fois
- h) affiche juste « bar », une fois

Question 7 (2 points) Le programme suivant ...

```
for i in range (10):  
    for j in range (10):  
        for k in range (10):  
            print ("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » 100 fois
- c) affiche « foo » 3 fois
- d) affiche « foo » 10000 fois
- e) affiche « foo » 6 fois
- f) affiche « foo » 1000 fois
- g) affiche « foo » 9 fois
- h) échoue
- i) affiche « foo » une infinité de fois
- j) affiche « foo » 10 fois
- k) affiche « foo » 20 fois
- l) affiche « foo » 30 fois
- m) n'affiche rien
- n) affiche « foo » 27 fois

Question 8 (3 points) Le programme suivant ...

```
a = 10  
for b in range (4):  
    a = a + b  
print (a)
```

- a) échoue
- b) affiche « 11 »
- c) n'affiche rien
- d) affiche « 15 »
- e) affiche « 16 »
- f) boucle infiniment
- g) affiche « a »
- h) affiche « 14 »
- i) affiche « 13 »
- j) affiche « b »
- k) affiche « 12 »
- l) affiche « 10 »

Question 9 (3 points) Le programme suivant ...

```
for foo in [True]:  
    print ("foo")  
print ("bar")
```

- a) échoue
- b) affiche « foo » une fois et « bar » une fois
- c) n'affiche rien
- d) affiche « foo » une infinité de fois
- e) affiche « foo » une fois et « bar » une infinité de fois
- f) affiche juste « foo », une fois
- g) affiche juste « bar », une fois
- h) affiche « bar » une infinité de fois

Exemplaire numéro 7

Programmation Python
Contrôle des connaissances : contrôle surprise numéro 1
IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- | | |
|-------------------------------|-------------------------------------|
| a) échoue | g) a comme résultat True |
| b) n'a aucun résultat | h) a comme résultat False |
| c) a comme résultat 0 | i) a comme résultat un dictionnaire |
| d) a comme résultat 9 | j) a comme résultat 90 |
| e) a comme résultat une liste | k) a comme résultat 3 |
| f) a comme résultat 30 | |

- | | |
|--|---------------------------|
| a) affiche « range (x) » une fois | rel |
| b) affiche tout nombre entier (y compris les négatifs) | d) affiche « i » une fois |
| c) affiche tout nombre naturel | e) échoue |
| | f) n'affiche rien |

Question 2 (3 points) Le programme suivant ...

```
for foo in [True]:  
    print ("foo")  
print ("bar")
```

- | | |
|---|---|
| a) n'affiche rien | e) affiche juste « foo », une fois |
| b) affiche « foo » une fois et « bar » une infinité de fois | f) échoue |
| c) affiche juste « bar », une fois | g) affiche « foo » une infinité de fois |
| d) affiche « foo » une fois et « bar » une fois | h) affiche « bar » une infinité de fois |

Question 6 (2 points) Le programme suivant « ...

```
while False:  
    print ("foo")  
print ("bar")
```

- | | |
|---|------------------------------------|
| a) affiche « foo » une infinité de fois | « bar » une fois |
| b) affiche « bar » une infinité de fois | e) affiche juste « bar », une fois |
| c) affiche « foo » une fois et « bar » une infinité de fois | f) échoue |
| d) affiche « foo » une fois et « bar » une fois | g) affiche juste « foo », une fois |
| | h) n'affiche rien |

Question 3 (2 points) Le programme suivant ...

```
for i in range (10):  
    for j in range (10):  
        for k in range (10):  
            print ("foo")
```

- | | |
|-------------------------------|---|
| a) affiche « foo » 1000 fois | i) affiche « foo » une infinité de fois |
| b) affiche « foo » 100 fois | j) affiche « foo » 6 fois |
| c) affiche « foo » 3 fois | k) affiche « foo » 30 fois |
| d) affiche « foo » 20 fois | l) affiche « foo » 9 fois |
| e) affiche « foo » 10000 fois | m) affiche « foo » 27 fois |
| f) n'affiche rien | n) affiche « foo » une fois |
| g) affiche « foo » 10 fois | |
| h) échoue | |

Question 7 (2 points) Le programme suivant ...

```
a = 8  
if a > 20:  
    print ("foo")  
else a <= 20:  
    print ("bar")
```

- | | |
|---|---|
| a) affiche « foo » une infinité de fois | e) affiche « foo » une fois |
| b) affiche « foo » une fois et « bar » une fois | f) affiche « bar » une fois |
| c) n'affiche rien | g) affiche « bar » une infinité de fois |
| d) affiche « a » une infinité de fois | h) échoue |
| | i) affiche « a » une fois |

Question 4 (3 points) Le programme suivant ...

```
a = 10  
for b in range (4):  
    a = a + b  
print (a)
```

- | | |
|-------------------|----------------------|
| a) affiche « a » | g) affiche « 12 » |
| b) affiche « 10 » | h) affiche « 16 » |
| c) affiche « 13 » | i) affiche « b » |
| d) échoue | j) affiche « 14 » |
| e) n'affiche rien | k) boucle infiniment |
| f) affiche « 11 » | l) affiche « 15 » |

Question 8 (2 points) Le programme suivant ...

```
def foo (x):  
    return x  
print (foo (7))
```

- | | |
|--------------------------|------------------------|
| a) échoue | f) affiche « foo » |
| b) affiche « x » | g) boucle infiniment |
| c) affiche « (foo (x)) » | h) n'affiche rien |
| d) affiche « (foo (7)) » | i) affiche « foo (7) » |
| e) affiche « 7 » | |

Question 5 (2 points) Le programme suivant ...

```
def foo (x):  
    for i in range (x):  
        print (i)
```

Question 9 (2 points) Le programme suivant ...

```
a = 10  
while a > 0:  
    print ("foo")
```

- | | |
|---|----------------------------|
| a) affiche « foo » 9 fois | d) affiche « foo » 11 fois |
| b) affiche « foo » une infinité de fois | e) n'affiche rien |
| c) affiche « foo » 10 fois | f) échoue |

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Exemplaire numéro 8

Documents (livres, slides, polycopiés, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
def foo (x):  
    for i in range (x):  
        print (i)
```

- a) échoue
- b) affiche « range (x) » une fois
- c) affiche tout nombre entier (y compris les négatifs)
- d) n'affiche rien
- e) affiche tout nombre naturel
- f) affiche « i » une fois

Question 2 (2 points) Le programme suivant ...

```
for i in range (10):  
    for j in range (10):  
        for k in range (10):  
            print ("foo")
```

- a) affiche « foo » une infinité de fois
- b) affiche « foo » 30 fois
- c) affiche « foo » 1000 fois
- d) affiche « foo » 100 fois
- e) affiche « foo » 10000 fois
- f) échoue
- g) n'affiche rien
- h) affiche « foo » 6 fois
- i) affiche « foo » 9 fois
- j) affiche « foo » 27 fois
- k) affiche « foo » 3 fois
- l) affiche « foo » 20 fois
- m) affiche « foo » 10 fois
- n) affiche « foo » une fois

Question 3 (2 points) Le programme suivant ...

```
def foo (x):  
    return x  
print (foo (7))
```

- a) affiche « 7 »
- b) affiche « foo »
- c) n'affiche rien
- d) boucle infiniment
- e) échoue
- f) affiche « (foo (7)) »
- g) affiche « foo (7) »
- h) affiche « (foo (x)) »
- i) affiche « x »

Question 4 (3 points) Le programme suivant ...

```
a = 10  
for b in range (4):  
    a = a + b  
print (a)
```

- a) affiche « 16 »
- b) affiche « 14 »
- c) affiche « a »
- d) échoue
- e) affiche « 15 »
- f) affiche « 13 »
- g) n'affiche rien
- h) affiche « 11 »
- i) affiche « 10 »
- j) boucle infiniment
- k) affiche « b »
- l) affiche « 12 »

Question 5 (2 points) Le programme suivant ...

```
a = 8  
if a > 20:  
    print ("foo")  
else a <= 20:  
    print ("bar")
```

- a) a comme résultat True
- b) a comme résultat 30
- c) échoue
- d) a comme résultat 90
- e) a comme résultat False
- f) a comme résultat 9
- g) a comme résultat un dictionnaire
- h) a comme résultat 0
- i) n'a aucun résultat
- j) a comme résultat une liste
- k) a comme résultat 3

- a) affiche « bar » une infinité de fois
- b) échoue
- c) affiche « a » une infinité de fois
- d) affiche « foo » une fois et « bar » une fois
- e) affiche « foo » une fois
- f) affiche « foo » une infinité de fois
- g) affiche « a » une fois
- h) n'affiche rien
- i) affiche « bar » une fois

Question 6 (2 points) Le programme suivant ...

```
a = 10  
while a > 0:  
    print ("foo")
```

- a) affiche « foo » 10 fois
- b) n'affiche rien
- c) affiche « foo » 9 fois
- d) affiche « foo » 11 fois
- e) échoue
- f) affiche « foo » une infinité de fois

Question 7 (2 points) Le programme suivant « ...

```
while False:  
    print ("foo")  
print ("bar")
```

- a) affiche « foo » une infinité de fois
- b) affiche « foo » une fois et « bar » une infinité de fois
- c) affiche « foo » une fois et « bar » une fois
- d) affiche « bar » une infinité de fois
- e) affiche juste « bar », une fois
- f) n'affiche rien
- g) échoue
- h) affiche juste « foo », une fois

Question 8 (3 points) Le programme suivant ...

```
for foo in [True]:  
    print ("foo")  
print ("bar")
```

- a) n'affiche rien
- b) affiche « bar » une infinité de fois
- c) affiche juste « bar », une fois
- d) affiche « foo » une fois et « bar » une fois
- e) affiche « foo » une infinité de fois
- f) affiche juste « foo », une fois
- g) affiche « foo » une fois et « bar » une infinité de fois
- h) échoue

Question 9 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 9

Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront soustraits en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) a comme résultat **False**
- b) a comme résultat **3**
- c) a comme résultat une liste
- d) a comme résultat **90**
- e) n'a aucun résultat
- f) a comme résultat **0**
- g) a comme résultat **9**
- h) a comme résultat un dictionnaire
- i) a comme résultat **30**
- j) a comme résultat **True**
- k) échoue

Question 2 (2 points) Le programme suivant « ... »

```
while False:  
    print ("foo")  
print ("bar")
```

- a) n'affiche rien
- b) affiche juste « bar », une fois
- c) affiche « bar » une infinité de fois
- d) affiche « foo » une fois et « bar » une fois
- e) échoue
- f) affiche « foo » une infinité de fois
- g) affiche « foo » une fois et « bar » une infinité de fois
- h) affiche juste « foo », une fois

Question 3 (3 points) Le programme suivant ...

```
for foo in [True]:  
    print ("foo")  
print ("bar")
```

- a) affiche juste « foo », une fois
- b) échoue
- c) affiche « bar » une infinité de fois
- d) affiche juste « bar », une fois
- e) affiche « foo » une infinité de fois
- f) affiche « foo » une fois et « bar » une fois
- g) n'affiche rien
- h) affiche « foo » une fois et « bar » une infinité de fois

Question 4 (2 points) Le programme suivant ...

```
a = 8  
if a > 20:  
    print ("foo")  
else a <= 20:  
    print ("bar")
```

- a) n'affiche rien
- b) affiche « bar » une infinité de fois
- c) affiche « foo » une fois
- d) affiche « foo » une fois et « bar » une fois
- e) affiche « a » une fois
- f) affiche « a » une infinité de fois
- g) affiche « foo » une infinité de fois
- h) affiche « bar » une fois
- i) échoue

Question 5 (2 points) Le programme suivant ...

```
def foo (x):  
    for i in range (x):  
        print (i)
```

- a) n'affiche rien
- b) affiche « i » une fois
- c) échoue
- d) affiche tout nombre naturel
- e) affiche tout nombre entier (y compris les négatifs)
- f) affiche « range (x) » une fois

Question 6 (2 points) Le programme suivant ...

```
def foo (x):  
    return x  
print (foo (7))
```

- a) affiche « 7 »
- b) boucle infiniment
- c) échoue
- d) affiche « x »
- e) affiche « foo »
- f) affiche « (foo (x)) »
- g) affiche « foo (7) »
- h) n'affiche rien
- i) affiche « (foo (7)) »

Question 7 (3 points) Le programme suivant ...

```
a = 10  
for b in range (4):  
    a = a + b  
print (a)
```

- a) boucle infiniment
- b) affiche « 13 »
- c) n'affiche rien
- d) affiche « b »
- e) affiche « 15 »
- f) affiche « 16 »
- g) affiche « a »
- h) affiche « 10 »
- i) affiche « 14 »
- j) affiche « 12 »
- k) affiche « 11 »
- l) échoue

Question 8 (2 points) Le programme suivant ...

```
for i in range (10):  
    for j in range (10):  
        for k in range (10):  
            print ("foo")
```

- a) affiche « foo » 27 fois
- b) affiche « foo » 1000 fois
- c) affiche « foo » 100 fois
- d) affiche « foo » 20 fois
- e) affiche « foo » 3 fois
- f) affiche « foo » 10 fois
- g) échoue
- h) affiche « foo » une infinité de fois
- i) affiche « foo » 10000 fois
- j) affiche « foo » 6 fois
- k) affiche « foo » 30 fois
- l) n'affiche rien
- m) affiche « foo » une fois
- n) affiche « foo » 9 fois

Question 9 (2 points) Le programme suivant ...

```
a = 10  
while a > 0:  
    print ("foo")
```

- a) affiche « foo » 11 fois
- b) affiche « foo » 9 fois
- c) affiche « foo » 10 fois
- d) n'affiche rien
- e) affiche « foo » une infinité de fois
- f) échoue

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 10

Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) **non** autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n , n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant « ... »

```
while False:
    print ("foo")
print ("bar")
```

- a) affiche « bar » une infinité de fois
- b) n'affiche rien
- c) échoue
- d) affiche « foo » une fois et « bar » une infinité de fois
- e) affiche « foo » une fois et « bar » une fois
- f) affiche juste « bar », une fois
- g) affiche juste « foo », une fois
- h) affiche « foo » une infinité de fois

Question 2 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 13 »
- b) affiche « 16 »
- c) affiche « 11 »
- d) affiche « 15 »
- e) affiche « 14 »
- f) affiche « 12 »
- g) affiche « a »
- h) affiche « b »
- i) affiche « 10 »
- j) échoue
- k) boucle infiniment
- l) n'affiche rien

Question 3 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- a) affiche « foo » 10 fois
- b) affiche « foo » 11 fois
- c) n'affiche rien
- d) affiche « foo » une infinité de fois
- e) affiche « foo » 9 fois
- f) échoue

Question 4 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- a) affiche « foo » une infinité de fois
- b) n'affiche rien
- c) affiche « a » une fois
- d) affiche « foo » une fois
- e) affiche « bar » une infinité de fois
- f) affiche « bar » une fois
- g) échoue
- h) affiche « a » une infinité de fois
- i) affiche « foo » une fois et « bar » une fois

Question 5 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- a) affiche « i » une fois
- b) affiche « range (x) » une fois
- c) affiche tout nombre entier (y compris les négatifs)
- d) échoue
- e) affiche tout nombre naturel
- f) n'affiche rien

Question 6 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- a) affiche « (foo (7)) »
- b) affiche « (foo (x)) »
- c) boucle infiniment
- d) n'affiche rien
- e) affiche « foo »
- f) affiche « 7 »
- g) affiche « foo (7) »
- h) échoue
- i) affiche « x »

Question 7 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- a) affiche « foo » 27 fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 1000 fois
- d) affiche « foo » 10 fois
- e) affiche « foo » une fois
- f) affiche « foo » 20 fois
- g) affiche « foo » 100 fois
- h) affiche « foo » 6 fois
- i) affiche « foo » 3 fois
- j) échoue
- k) n'affiche rien
- l) affiche « foo » 10000 fois
- m) affiche « foo » 9 fois
- n) affiche « foo » 30 fois

Question 8 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) a comme résultat 30
- b) n'a aucun résultat
- c) a comme résultat 0
- d) a comme résultat 90
- e) a comme résultat 3
- f) a comme résultat un dictionnaire
- g) a comme résultat une liste
- h) a comme résultat False
- i) a comme résultat 9
- j) a comme résultat True
- k) échoue

Question 9 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- a) affiche juste « foo », une fois
- b) affiche « bar » une infinité de fois
- c) affiche « foo » une fois et « bar » une fois
- d) affiche « foo » une infinité de fois
- e) affiche juste « bar », une fois
- f) affiche « foo » une fois et « bar » une infinité de fois
- g) échoue
- h) n'affiche rien

Exemplaire numéro 11

Programmation Python
Contrôle des connaissances : contrôle surprise numéro 1
IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, polycopiés, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- | | |
|-------------------------------|---|
| a) échoue | i) affiche « foo » 1000 fois |
| b) affiche « foo » 10000 fois | j) n'affiche rien |
| c) affiche « foo » 10 fois | k) affiche « foo » 9 fois |
| d) affiche « foo » 6 fois | l) affiche « foo » une infinité de fois |
| e) affiche « foo » 27 fois | m) affiche « foo » 30 fois |
| f) affiche « foo » 3 fois | n) affiche « foo » 20 fois |
| g) affiche « foo » une fois | |
| h) affiche « foo » 100 fois | |

Question 2 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- | | |
|---|---|
| a) affiche « foo » une fois et « bar » une infinité de fois | e) n'affiche rien |
| b) affiche « foo » une fois et « bar » une fois | f) affiche « foo » une infinité de fois |
| c) affiche juste « foo », une fois | g) échoue |
| d) affiche juste « bar », une fois | h) affiche « bar » une infinité de fois |

Question 3 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- | | |
|---|----------------------------|
| a) affiche « foo » 10 fois | d) échoue |
| b) affiche « foo » une infinité de fois | e) n'affiche rien |
| c) affiche « foo » 9 fois | f) affiche « foo » 11 fois |

Question 4 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- | | |
|-------------------------------------|---------------------------|
| a) a comme résultat une liste | f) a comme résultat False |
| b) a comme résultat un dictionnaire | g) a comme résultat 0 |
| c) a comme résultat 90 | h) a comme résultat 9 |
| d) a comme résultat True | i) n'a aucun résultat |
| e) a comme résultat 3 | j) échoue |
| | k) a comme résultat 30 |

Question 5 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- | | |
|-------------------|----------------------|
| a) n'affiche rien | g) affiche « 16 » |
| b) affiche « 15 » | h) affiche « 11 » |
| c) échoue | i) affiche « a » |
| d) affiche « 14 » | j) boucle infiniment |
| e) affiche « 12 » | k) affiche « 13 » |
| f) affiche « 10 » | l) affiche « b » |

Question 6 (2 points) Le programme suivant « ...

```
while False:
    print ("foo")
print ("bar")
```

- | | |
|---|---|
| a) affiche « foo » une infinité de fois | e) affiche « bar » une infinité de fois |
| b) n'affiche rien | f) échoue |
| c) affiche « foo » une fois et « bar » une infinité de fois | g) affiche juste « bar », une fois |
| d) affiche juste « foo », une fois | h) affiche « foo » une fois et « bar » une fois |

Question 7 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- | | |
|---|---|
| a) affiche « bar » une fois | f) affiche « a » une fois |
| b) affiche « a » une infinité de fois | g) affiche « foo » une fois et « bar » une fois |
| c) échoue | h) affiche « foo » une fois |
| d) affiche « bar » une infinité de fois | i) affiche « foo » une infinité de fois |
| e) n'affiche rien | |

Question 8 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- | | |
|--------------------------|--------------------------|
| a) affiche « 7 » | f) n'affiche rien |
| b) affiche « (foo (x)) » | g) affiche « foo (7) » |
| c) affiche « x » | h) échoue |
| d) boucle infiniment | i) affiche « (foo (7)) » |
| e) affiche « foo » | |

Question 9 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- | | |
|-----------------------------------|--|
| a) affiche tout nombre naturel | fois |
| b) affiche « i » une fois | e) affiche tout nombre entier (y compris les négatifs) |
| c) n'affiche rien | f) échoue |
| d) affiche « range (x) » une fois | |

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 12

Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) a comme résultat True
- b) a comme résultat une liste
- c) a comme résultat 3
- d) échoue
- e) a comme résultat 9
- f) a comme résultat False
- g) a comme résultat un dictionnaire
- h) a comme résultat 90
- i) a comme résultat 30
- j) a comme résultat 0
- k) n'a aucun résultat

- a) affiche « bar » une infinité de fois
- b) affiche « foo » une fois
- c) affiche « bar » une fois
- d) échoue
- e) affiche « a » une fois
- f) n'affiche rien
- g) affiche « foo » une fois et « bar » une fois
- h) affiche « a » une infinité de fois
- i) affiche « foo » une infinité de fois

Question 2 (3 points) Le programme suivant ...

```
for foo in [True]:  
    print ("foo")  
print ("bar")
```

- a) affiche juste « bar », une fois
- b) affiche « bar » une infinité de fois
- c) affiche « foo » une fois et « bar » une infinité de fois
- d) affiche « foo » une infinité de fois
- e) n'affiche rien
- f) échoue
- g) affiche « foo » une fois et « bar » une fois
- h) affiche juste « foo », une fois

Question 6 (2 points) Le programme suivant ...

```
def foo (x):  
    return x  
print (foo (7))
```

- a) affiche « foo (7) »
- b) affiche « x »
- c) boucle infiniment
- d) affiche « foo »
- e) n'affiche rien
- f) échoue
- g) affiche « (foo (x)) »
- h) affiche « (foo (7)) »
- i) affiche « 7 »

Question 3 (2 points) Le programme suivant ...

```
for i in range (10):  
    for j in range (10):  
        for k in range (10):  
            print ("foo")
```

- a) affiche « foo » 100 fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 20 fois
- d) affiche « foo » 10000 fois
- e) affiche « foo » 30 fois
- f) affiche « foo » 9 fois
- g) affiche « foo » 3 fois
- h) échoue
- i) affiche « foo » une fois
- j) affiche « foo » 27 fois
- k) affiche « foo » 1000 fois
- l) affiche « foo » 10 fois
- m) n'affiche rien
- n) affiche « foo » 6 fois

Question 7 (2 points) Le programme suivant ...

```
def foo (x):  
    for i in range (x):  
        print (i)
```

- a) affiche tout nombre entier
- b) n'affiche rien
- c) affiche tout nombre naturel
- d) affiche « range (x) » une fois (y compris les négatifs)
- e) échoue
- f) affiche « i » une fois

Question 4 (3 points) Le programme suivant ...

```
a = 10  
for b in range (4):  
    a = a + b  
print (a)
```

- a) n'affiche rien
- b) affiche « 12 »
- c) affiche « 15 »
- d) affiche « 13 »
- e) affiche « a »
- f) affiche « b »
- g) affiche « 10 »
- h) affiche « 16 »
- i) affiche « 14 »
- j) affiche « 11 »
- k) boucle infiniment
- l) échoue

- a) affiche « bar » une infinité de fois
- b) affiche juste « foo », une fois
- c) affiche « foo » une infinité de fois
- d) n'affiche rien
- e) affiche « foo » une fois et « bar » une infinité de fois
- f) échoue
- g) affiche juste « bar », une fois
- h) affiche « foo » une fois et « bar » une fois

Question 5 (2 points) Le programme suivant ...

```
a = 8  
if a > 20:  
    print ("foo")  
else a <= 20:  
    print ("bar")
```

Question 9 (2 points) Le programme suivant ...

```
a = 10  
while a > 0:  
    print ("foo")
```

- a) affiche « foo » 9 fois
- b) n'affiche rien
- c) affiche « foo » 10 fois
- d) affiche « foo » 11 fois
- e) affiche « foo » une infinité de fois
- f) échoue

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Exemplaire numéro 13

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) **non** autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n , n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- a) n'affiche rien
- b) affiche « a » une fois
- c) affiche « foo » une fois
- d) affiche « bar » une fois
- e) affiche « foo » une infinité de fois
- f) échoue
- g) affiche « a » une infinité de fois
- h) affiche « bar » une infinité de fois
- i) affiche « foo » une fois et « bar » une fois

- a) affiche « foo » 10 fois
- b) affiche « foo » 3 fois
- c) affiche « foo » 100 fois
- d) affiche « foo » 30 fois
- e) n'affiche rien
- f) affiche « foo » une infinité de fois
- g) affiche « foo » 9 fois
- h) affiche « foo » 1000 fois
- i) affiche « foo » une fois
- j) affiche « foo » 6 fois
- k) échoue
- l) affiche « foo » 20 fois
- m) affiche « foo » 27 fois
- n) affiche « foo » 10000 fois

Question 2 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- a) échoue
- b) affiche « foo » une fois et « bar » une infinité de fois
- c) affiche « foo » une infinité de fois
- d) n'affiche rien
- e) affiche « foo » une fois et « bar » une fois
- f) affiche « bar » une infinité de fois
- g) affiche juste « foo », une fois
- h) affiche juste « bar », une fois

Question 6 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- a) affiche « (foo (x)) »
- b) affiche « x »
- c) affiche « (foo (7)) »
- d) affiche « 7 »
- e) affiche « foo (7) »
- f) échoue
- g) n'affiche rien
- h) affiche « foo »
- i) boucle infiniment

Question 3 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- a) affiche « range (x) » une fois
- b) échoue
- c) n'affiche rien
- d) affiche tout nombre naturel
- e) affiche tout nombre entier (y compris les négatifs)
- f) affiche « i » une fois

Question 7 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- a) affiche « foo » une infinité de fois
- b) affiche « foo » 11 fois
- c) n'affiche rien
- d) échoue
- e) affiche « foo » 9 fois
- f) affiche « foo » 10 fois

Question 4 (2 points) Le programme suivant ...

```
while False:
    print ("foo")
print ("bar")
```

- a) affiche « foo » une infinité de fois
- b) affiche « foo » une fois et « bar » une infinité de fois
- c) affiche « foo » une fois et « bar » une fois
- d) n'affiche rien
- e) échoue
- f) affiche juste « bar », une fois
- g) affiche « bar » une infinité de fois
- h) affiche juste « foo », une fois

Question 8 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) a comme résultat 30
- b) échoue
- c) n'a aucun résultat
- d) a comme résultat True
- e) a comme résultat 9
- f) a comme résultat 0
- g) a comme résultat 90
- h) a comme résultat un dictionnaire
- i) a comme résultat False
- j) a comme résultat une liste
- k) a comme résultat 3

Question 9 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 15 »
- b) n'affiche rien
- c) échoue
- d) affiche « 11 »
- e) affiche « 16 »
- f) affiche « b »
- g) affiche « 13 »
- h) affiche « 14 »
- i) affiche « 10 »
- j) boucle infiniment
- k) affiche « 12 »
- l) affiche « a »

Question 5 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

Exemplaire numéro 14

Programmation Python
Contrôle des connaissances : contrôle surprise numéro 1
IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- | | |
|-------------------------------|---|
| a) n'affiche rien | i) affiche « foo » 100 fois |
| b) affiche « foo » 6 fois | j) affiche « foo » une infinité de fois |
| c) affiche « foo » une fois | k) affiche « foo » 3 fois |
| d) affiche « foo » 30 fois | l) affiche « foo » 10 fois |
| e) affiche « foo » 27 fois | m) affiche « foo » 20 fois |
| f) affiche « foo » 10000 fois | n) affiche « foo » 1000 fois |
| g) affiche « foo » 9 fois | |
| h) échoue | |

Question 2 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- | | |
|---------------------------|-------------------------------------|
| a) a comme résultat False | g) a comme résultat 0 |
| b) a comme résultat 90 | h) a comme résultat 3 |
| c) a comme résultat True | i) a comme résultat un dictionnaire |
| d) a comme résultat 9 | j) n'a aucun résultat |
| e) échoue | k) a comme résultat une liste |
| f) a comme résultat 30 | |

Question 3 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- | | |
|--------------------------|--------------------------|
| a) affiche « foo » | f) échoue |
| b) affiche « (foo (7)) » | g) n'affiche rien |
| c) affiche « 7 » | h) affiche « (foo (x)) » |
| d) affiche « x » | i) boucle infiniment |
| e) affiche « foo (7) » | |

Question 4 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- | | |
|----------------------|-------------------|
| a) affiche « 15 » | g) affiche « 13 » |
| b) affiche « 10 » | h) échoue |
| c) affiche « 14 » | i) n'affiche rien |
| d) boucle infiniment | j) affiche « 11 » |
| e) affiche « b » | k) affiche « a » |
| f) affiche « 12 » | l) affiche « 16 » |

Question 5 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- | | |
|---|----------------------------|
| a) affiche « foo » une infinité de fois | d) affiche « foo » 9 fois |
| b) n'affiche rien | e) affiche « foo » 10 fois |
| c) affiche « foo » 11 fois | f) échoue |

Question 6 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- | | |
|---|---|
| a) affiche « a » une fois | de fois |
| b) échoue | g) affiche « foo » une infinité de fois |
| c) affiche « bar » une fois | de fois |
| d) affiche « a » une infinité de fois | h) affiche « foo » une fois et « bar » une fois |
| e) affiche « foo » une fois | i) n'affiche rien |
| f) affiche « bar » une infinité de fois | |

Question 7 (2 points) Le programme suivant « ...

```
while False:
    print ("foo")
print ("bar")
```

- | | |
|---|---|
| a) affiche juste « foo », une fois | de fois |
| b) affiche « foo » une fois et « bar » une infinité de fois | e) n'affiche rien |
| c) affiche juste « bar », une fois | f) affiche « foo » une infinité de fois |
| d) affiche « bar » une infinité de fois | g) échoue |
| | h) affiche « foo » une fois et « bar » une fois |

Question 8 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- | | |
|-----------------------------------|--|
| a) affiche « range (x) » une fois | rel |
| b) n'affiche rien | e) affiche tout nombre entier (y compris les négatifs) |
| c) affiche « i » une fois | f) échoue |
| d) affiche tout nombre naturel | |

Question 9 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- | | |
|---|---|
| a) n'affiche rien | « bar » une infinité de fois |
| b) échoue | f) affiche « foo » une fois et « bar » une fois |
| c) affiche « bar » une infinité de fois | g) affiche juste « foo », une fois |
| d) affiche « foo » une infinité de fois | h) affiche juste « bar », une fois |
| e) affiche « foo » une fois et | |

Exemplaire numéro 15

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- | | |
|-------------------------------|---------------------------|
| a) a comme résultat 9 | tionnaire |
| b) a comme résultat 0 | g) a comme résultat 3 |
| c) échoue | h) a comme résultat 90 |
| d) a comme résultat une liste | i) a comme résultat False |
| e) a comme résultat True | j) a comme résultat 30 |
| f) a comme résultat un dic- | k) n'a aucun résultat |

- | | |
|---|----------------------------|
| a) affiche « foo » 10 fois | d) affiche « foo » 11 fois |
| b) affiche « foo » 9 fois | e) échoue |
| c) affiche « foo » une infinité de fois | f) n'affiche rien |

Question 2 (2 points) Le programme suivant ...

```
def foo (x):  
    for i in range (x):  
        print (i)
```

- | | |
|--|--------------------------------|
| a) échoue | fois |
| b) affiche tout nombre entier (y compris les négatifs) | e) n'affiche rien |
| c) affiche « i » une fois | f) affiche tout nombre naturel |
| d) affiche « range (x) » une | |

Question 6 (3 points) Le programme suivant ...

```
for foo in [True]:  
    print ("foo")  
print ("bar")
```

- | | |
|---|---|
| a) affiche juste « bar », une fois | e) affiche « foo » une infinité de fois |
| b) affiche « bar » une infinité de fois | f) échoue |
| c) affiche juste « foo », une fois | g) affiche « foo » une fois et « bar » une infinité de fois |
| d) n'affiche rien | h) affiche « foo » une fois et « bar » une fois |

Question 3 (2 points) Le programme suivant « ...

```
while False:  
    print ("foo")  
print ("bar")
```

- | | |
|---|---|
| a) affiche juste « foo », une fois | « bar » une fois |
| b) affiche « bar » une infinité de fois | e) n'affiche rien |
| c) affiche juste « bar », une fois | f) affiche « foo » une fois et « bar » une infinité de fois |
| d) affiche « foo » une fois et | g) affiche « foo » une infinité de fois |
| | h) échoue |

Question 7 (2 points) Le programme suivant ...

```
for i in range (10):  
    for j in range (10):  
        for k in range (10):  
            print ("foo")
```

- | | |
|---|-------------------------------|
| a) affiche « foo » 1000 fois | h) affiche « foo » 30 fois |
| b) affiche « foo » 10 fois | i) affiche « foo » 6 fois |
| c) affiche « foo » 9 fois | j) échoue |
| d) affiche « foo » 3 fois | k) affiche « foo » 20 fois |
| e) n'affiche rien | l) affiche « foo » 27 fois |
| f) affiche « foo » une fois | m) affiche « foo » 10000 fois |
| g) affiche « foo » une infinité de fois | n) affiche « foo » 100 fois |

Question 4 (2 points) Le programme suivant ...

```
a = 8  
if a > 20:  
    print ("foo")  
else a <= 20:  
    print ("bar")
```

- | | |
|---|---|
| a) affiche « bar » une infinité de fois | f) affiche « a » une infinité de fois |
| b) affiche « a » une fois | g) affiche « foo » une fois et « bar » une fois |
| c) affiche « bar » une fois | h) affiche « foo » une fois de fois |
| d) affiche « foo » une infinité de fois | i) échoue |
| e) n'affiche rien | |

Question 8 (2 points) Le programme suivant ...

```
def foo (x):  
    return x  
print (foo (7))
```

- | | |
|------------------------|--------------------------|
| a) affiche « foo (7) » | f) affiche « foo » |
| b) n'affiche rien | g) affiche « (foo (x)) » |
| c) boucle infiniment | h) affiche « (foo (7)) » |
| d) affiche « x » | i) échoue |
| e) affiche « 7 » | |

Question 5 (2 points) Le programme suivant ...

```
a = 10  
while a > 0:  
    print ("foo")
```

Question 9 (3 points) Le programme suivant ...

```
a = 10  
for b in range (4):  
    a = a + b  
print (a)
```

- | | |
|----------------------|-------------------|
| a) affiche « b » | g) affiche « 13 » |
| b) boucle infiniment | h) échoue |
| c) affiche « 16 » | i) affiche « 10 » |
| d) affiche « 11 » | j) affiche « a » |
| e) affiche « 14 » | k) affiche « 12 » |
| f) affiche « 15 » | l) n'affiche rien |

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 16

Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) **non** autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n , n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- a) affiche « foo » une fois et « bar » une fois
- b) affiche « foo » une infinité de fois e) affiche « bar » une infinité de fois
- c) affiche « foo » une infinité de fois f) échoue
- d) affiche « foo » une fois et fois g) n'affiche rien
- h) affiche juste « bar », une fois

Question 2 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- a) n'affiche rien e) affiche « foo » 11 fois
- b) affiche « foo » 10 fois f) affiche « foo » une infinité de fois
- c) échoue
- d) affiche « foo » 9 fois

Question 3 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- a) affiche « range (x) » une fois d) n'affiche rien
- b) échoue e) affiche tout nombre naturel
- c) affiche tout nombre entier (y compris les négatifs) f) affiche « i » une fois

Question 4 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « b » g) affiche « 13 »
- b) affiche « a » h) affiche « 15 »
- c) affiche « 10 » i) affiche « 16 »
- d) affiche « 12 » j) échoue
- e) affiche « 14 » k) boucle infiniment
- f) affiche « 11 » l) n'affiche rien

Question 5 (2 points) Le programme suivant « ...

```
while False:
    print ("foo")
print ("bar")
```

- a) affiche « foo » une fois et « bar » une fois e) affiche « bar » une infinité de fois
- b) affiche « foo » une fois et « bar » une infinité de fois f) échoue
- c) affiche « foo » une infinité de fois g) affiche juste « bar », une fois
- d) n'affiche rien h) affiche juste « foo », une fois

Question 6 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- a) affiche « foo » une fois f) affiche « bar » une infinité de fois
- b) n'affiche rien
- c) affiche « bar » une fois g) échoue
- d) affiche « foo » une fois et « bar » une fois h) affiche « foo » une infinité de fois
- e) affiche « a » une infinité de fois i) affiche « a » une fois

Question 7 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) échoue
- b) a comme résultat 3 g) a comme résultat False
- c) a comme résultat 9 h) a comme résultat True
- d) a comme résultat une liste i) a comme résultat 0
- e) a comme résultat 90 j) n'a aucun résultat
- f) a comme résultat un dictionnaire k) a comme résultat 30

Question 8 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- a) affiche « foo » 30 fois h) affiche « foo » 27 fois
- b) affiche « foo » 100 fois i) affiche « foo » 3 fois
- c) affiche « foo » une fois j) affiche « foo » 20 fois
- d) affiche « foo » 10000 fois k) n'affiche rien
- e) échoue l) affiche « foo » 9 fois
- f) affiche « foo » 6 fois m) affiche « foo » 1000 fois
- g) affiche « foo » une infinité de fois n) affiche « foo » 10 fois

Question 9 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- a) échoue f) affiche « (foo (x)) »
- b) affiche « foo » g) affiche « (foo (7)) »
- c) affiche « x » h) affiche « 7 »
- d) n'affiche rien i) affiche « foo (7) »
- e) boucle infiniment

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 17

Luca Saiu

Documents (livres, slides, polycopiés, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) L'expression suivante ...

[10, 20, 0] * 3

- a) a comme résultat 30
- b) a comme résultat un dictionnaire
- c) a comme résultat une liste
- d) a comme résultat 0
- e) a comme résultat False
- f) a comme résultat 9
- g) a comme résultat True
- h) n'a aucun résultat
- i) échoue
- j) a comme résultat 3
- k) a comme résultat 90

- a) échoue
- b) affiche « bar » une infinité de fois
- c) affiche « bar » une fois
- d) affiche « a » une infinité de fois
- e) affiche « foo » une fois
- f) affiche « a » une fois
- g) affiche « foo » une fois et « bar » une fois
- h) n'affiche rien
- i) affiche « foo » une infinité de fois

Question 2 (3 points) Le programme suivant ...

```
for foo in [True]:  
    print ("foo")  
print ("bar")
```

- a) affiche juste « foo », une fois
- b) affiche « foo » une infinité de fois
- c) échoue
- d) n'affiche rien
- e) affiche « foo » une fois et « bar » une fois
- f) affiche « bar » une infinité de fois
- g) affiche « foo » une fois et « bar » une infinité de fois
- h) affiche juste « bar », une fois

Question 6 (3 points) Le programme suivant ...

```
a = 10  
for b in range (4):  
    a = a + b  
print (a)
```

- a) affiche « 12 »
- b) affiche « 10 »
- c) affiche « b »
- d) affiche « 14 »
- e) affiche « 15 »
- f) affiche « 11 »
- g) affiche « 13 »
- h) échoue
- i) affiche « 16 »
- j) n'affiche rien
- k) affiche « a »
- l) boucle infiniment

Question 3 (2 points) Le programme suivant « ...

```
while False:  
    print ("foo")  
print ("bar")
```

- a) n'affiche rien
- b) affiche juste « foo », une fois
- c) affiche juste « bar », une fois
- d) affiche « bar » une infinité de fois
- e) échoue
- f) affiche « foo » une fois et « bar » une infinité de fois
- g) affiche « foo » une fois et « bar » une fois
- h) affiche « foo » une infinité de fois

Question 7 (2 points) Le programme suivant ...

```
def foo (x):  
    return x  
print (foo (7))
```

- a) affiche « foo (7) »
- b) n'affiche rien
- c) affiche « foo »
- d) affiche « (foo (7)) »
- e) échoue
- f) affiche « x »
- g) boucle infiniment
- h) affiche « (foo (x)) »
- i) affiche « 7 »

Question 4 (2 points) Le programme suivant ...

```
for i in range (10):  
    for j in range (10):  
        for k in range (10):  
            print ("foo")
```

- a) affiche « foo » 1000 fois
- b) affiche « foo » 100 fois
- c) affiche « foo » une infinité de fois
- d) affiche « foo » 20 fois
- e) affiche « foo » 6 fois
- f) affiche « foo » 9 fois
- g) affiche « foo » 3 fois
- h) affiche « foo » une fois
- i) affiche « foo » 30 fois
- j) affiche « foo » 27 fois
- k) échoue
- l) affiche « foo » 10 fois
- m) affiche « foo » 10000 fois
- n) n'affiche rien

Question 8 (2 points) Le programme suivant ...

```
a = 10  
while a > 0:  
    print ("foo")
```

- a) n'affiche rien
- b) affiche « foo » 10 fois
- c) échoue
- d) affiche « foo » 9 fois
- e) affiche « foo » 11 fois
- f) affiche « foo » une infinité de fois

Question 9 (2 points) Le programme suivant ...

```
def foo (x):  
    for i in range (x):  
        print (i)
```

- a) affiche tout nombre naturel
- b) affiche « i » une fois
- c) affiche tout nombre entier (y compris les négatifs)
- d) n'affiche rien
- e) échoue
- f) affiche « range (x) » une fois

Question 5 (2 points) Le programme suivant ...

```
a = 8  
if a > 20:  
    print ("foo")  
else a <= 20:  
    print ("bar")
```

- a) affiche tout nombre naturel
- b) affiche « i » une fois
- c) affiche tout nombre entier (y compris les négatifs)
- d) n'affiche rien
- e) échoue
- f) affiche « range (x) » une fois

Exemplaire numéro 18

Programmation Python
Contrôle des connaissances : contrôle surprise numéro 1
IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- | | |
|-----------------------------|---|
| a) affiche « foo » 10 fois | i) affiche « foo » 6 fois |
| b) affiche « foo » 20 fois | j) affiche « foo » 10000 fois |
| c) affiche « foo » 3 fois | k) affiche « foo » une infinité de fois |
| d) n'affiche rien | l) affiche « foo » une fois |
| e) affiche « foo » 9 fois | m) échoue |
| f) affiche « foo » 27 fois | n) affiche « foo » 1000 fois |
| g) affiche « foo » 30 fois | |
| h) affiche « foo » 100 fois | |

Question 2 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- | | |
|---------------------------------|---------------------------|
| a) n'affiche rien | de fois |
| b) affiche « foo » 10 fois | e) échoue |
| c) affiche « foo » 11 fois | f) affiche « foo » 9 fois |
| d) affiche « foo » une infinité | |

Question 3 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- | | |
|-------------------------------------|-------------------------------|
| a) a comme résultat True | f) a comme résultat 3 |
| b) a comme résultat un dictionnaire | g) a comme résultat une liste |
| c) a comme résultat 30 | h) a comme résultat False |
| d) n'a aucun résultat | i) échoue |
| e) a comme résultat 9 | j) a comme résultat 0 |
| | k) a comme résultat 90 |

Question 4 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- | | |
|---|---|
| a) affiche « bar » une infinité de fois | f) affiche « a » une infinité de fois |
| b) affiche « a » une fois | g) affiche « foo » une infinité de fois |
| c) affiche « foo » une fois et « bar » une fois | h) affiche « bar » une fois |
| d) n'affiche rien | i) échoue |
| e) affiche « foo » une fois | |

Question 5 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- | | |
|--|---------------------------|
| a) affiche tout nombre naturel | d) échoue |
| b) affiche tout nombre entier (y compris les négatifs) | e) n'affiche rien |
| c) affiche « range (x) » une fois | f) affiche « i » une fois |

Question 6 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
    print (a)
```

- | | |
|----------------------|-------------------|
| a) affiche « 13 » | g) n'affiche rien |
| b) affiche « 11 » | h) affiche « a » |
| c) boucle infiniment | i) affiche « 16 » |
| d) affiche « b » | j) affiche « 10 » |
| e) échoue | k) affiche « 12 » |
| f) affiche « 14 » | l) affiche « 15 » |

Question 7 (2 points) Le programme suivant ...

```
while False:
    print ("foo")
print ("bar")
```

- | | |
|---|---|
| a) échoue | e) affiche juste « bar », une fois |
| b) affiche « bar » une infinité de fois | f) affiche « foo » une fois et « bar » une fois |
| c) affiche « foo » une fois et « bar » une infinité de fois | g) affiche juste « foo », une fois |
| d) affiche « foo » une infinité de fois | h) n'affiche rien |

Question 8 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- | | |
|----------------------|--------------------------|
| a) n'affiche rien | f) affiche « (foo (x)) » |
| b) affiche « 7 » | g) affiche « (foo (7)) » |
| c) affiche « x » | h) affiche « foo » |
| d) boucle infiniment | i) affiche « foo (7) » |
| e) échoue | |

Question 9 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- | | |
|---|---|
| a) affiche « foo » une infinité de fois | e) affiche juste « bar », une fois |
| b) affiche « bar » une infinité de fois | f) affiche « foo » une fois et « bar » une infinité de fois |
| c) échoue | g) affiche juste « foo », une fois |
| d) affiche « foo » une fois et « bar » une fois | h) n'affiche rien |

Exemplaire numéro 19

Programmation Python
Contrôle des connaissances : contrôle surprise numéro 1
IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, polycopiés, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (3 points) Le programme suivant ...

[10, 20, 0] * 3

```
for foo in [True]:  
    print ("foo")  
print ("bar")
```

- a) affiche « foo » une infinité de fois
- b) échoue
- c) affiche juste « foo », une fois
- d) affiche « foo » une fois et « bar » une infinité de fois
- e) affiche « bar » une infinité de fois
- f) affiche juste « bar », une fois
- g) affiche « foo » une fois et « bar » une fois
- h) n'affiche rien

- a) a comme résultat 90
- b) a comme résultat une liste
- c) échoue
- d) a comme résultat 0
- e) a comme résultat 3
- f) a comme résultat False
- g) a comme résultat True
- h) a comme résultat 30
- i) n'a aucun résultat
- j) a comme résultat un dictionnaire
- k) a comme résultat 9

Question 2 (3 points) Le programme suivant ...

```
a = 10  
for b in range (4):  
    a = a + b  
print (a)
```

- a) échoue
- b) affiche « b »
- c) affiche « 16 »
- d) affiche « 10 »
- e) affiche « 12 »
- f) affiche « a »
- g) affiche « 15 »
- h) boucle infiniment
- i) affiche « 11 »
- j) affiche « 14 »
- k) n'affiche rien
- l) affiche « 13 »

Question 3 (2 points) Le programme suivant ...

```
a = 8  
if a > 20:  
    print ("foo")  
else a <= 20:  
    print ("bar")
```

- a) affiche « bar » une infinité de fois
- b) affiche « a » une fois
- c) affiche « a » une infinité de fois
- d) affiche « foo » une fois
- e) affiche « foo » une fois et « bar » une fois
- f) affiche « bar » une fois
- g) affiche « foo » une infinité de fois
- h) échoue
- i) n'affiche rien

Question 4 (2 points) Le programme suivant « ...

```
while False:  
    print ("foo")  
print ("bar")
```

- a) affiche « foo » une infinité de fois
- b) échoue
- c) affiche « bar » une infinité de fois
- d) n'affiche rien
- e) affiche « foo » une fois et « bar » une fois
- f) affiche juste « bar », une fois
- g) affiche juste « foo », une fois
- h) affiche « foo » une fois et « bar » une fois

Question 5 (2 points) L'expression suivante ...

Question 6 (2 points) Le programme suivant ...

```
def foo (x):  
    for i in range (x):  
        print (i)
```

- a) n'affiche rien
- b) affiche « range (x) » une fois
- c) affiche tout nombre naturel
- d) affiche tout nombre entier (y compris les négatifs)
- e) échoue
- f) affiche « i » une fois

Question 7 (2 points) Le programme suivant ...

```
def foo (x):  
    return x  
print (foo (7))
```

- a) affiche « (foo (7)) »
- b) échoue
- c) n'affiche rien
- d) affiche « foo (7) »
- e) affiche « 7 »
- f) affiche « foo »
- g) affiche « x »
- h) affiche « (foo (x)) »
- i) boucle infiniment

Question 8 (2 points) Le programme suivant ...

```
a = 10  
while a > 0:  
    print ("foo")
```

- a) affiche « foo » 11 fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 9 fois
- d) affiche « foo » 10 fois
- e) n'affiche rien
- f) échoue

Question 9 (2 points) Le programme suivant ...

```
for i in range (10):  
    for j in range (10):  
        for k in range (10):  
            print ("foo")
```

- a) affiche « foo » 20 fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 3 fois
- d) n'affiche rien
- e) échoue
- f) affiche « foo » 9 fois
- g) affiche « foo » une fois
- h) affiche « foo » 10000 fois
- i) affiche « foo » 27 fois
- j) affiche « foo » 30 fois
- k) affiche « foo » 10 fois
- l) affiche « foo » 1000 fois
- m) affiche « foo » 100 fois
- n) affiche « foo » 6 fois

Exemplaire numéro 20

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, polycopiés, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- | | |
|-------------------------------------|-------------------------------|
| a) a comme résultat un dictionnaire | f) a comme résultat 0 |
| b) a comme résultat 30 | g) n'a aucun résultat |
| c) a comme résultat 3 | h) a comme résultat une liste |
| d) a comme résultat True | i) a comme résultat 90 |
| e) a comme résultat False | j) échoue |
| | k) a comme résultat 9 |

Question 2 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- | | |
|----------------------|-------------------|
| a) boucle infiniment | g) échoue |
| b) n'affiche rien | h) affiche « b » |
| c) affiche « 13 » | i) affiche « 12 » |
| d) affiche « a » | j) affiche « 14 » |
| e) affiche « 15 » | k) affiche « 11 » |
| f) affiche « 16 » | l) affiche « 10 » |

Question 3 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- | | |
|----------------------------|---|
| a) affiche « foo » 9 fois | e) affiche « foo » une infinité de fois |
| b) échoue | |
| c) affiche « foo » 10 fois | f) affiche « foo » 11 fois |
| d) n'affiche rien | |

Question 4 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- | | |
|---|---|
| a) échoue | e) affiche « foo » une fois et « bar » une fois |
| b) affiche juste « foo », une fois | f) affiche « foo » une fois et « bar » une infinité de fois |
| c) affiche juste « bar », une fois | g) n'affiche rien |
| d) affiche « bar » une infinité de fois | h) affiche « foo » une infinité de fois |

Question 5 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- | | |
|--------------------------|------------------------|
| a) affiche « (foo (7)) » | f) boucle infiniment |
| b) affiche « x » | g) affiche « 7 » |
| c) affiche « foo » | h) affiche « foo (7) » |
| d) échoue | i) n'affiche rien |
| e) affiche « (foo (x)) » | |

Question 6 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- | | |
|---|---|
| a) affiche « a » une infinité de fois | f) affiche « a » une fois |
| b) affiche « bar » une fois | g) n'affiche rien |
| c) échoue | h) affiche « bar » une infinité de fois |
| d) affiche « foo » une infinité de fois | i) affiche « foo » une fois et « bar » une fois |
| e) affiche « foo » une fois | |

Question 7 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- | | |
|--|-----------------------------------|
| a) échoue | d) affiche « range (x) » une fois |
| b) affiche tout nombre entier (y compris les négatifs) | e) n'affiche rien |
| c) affiche tout nombre naturel | f) affiche « i » une fois |

Question 8 (2 points) Le programme suivant ...

```
while False:
    print ("foo")
print ("bar")
```

- | | |
|---|---|
| a) n'affiche rien | e) affiche « bar » une infinité de fois |
| b) affiche « foo » une infinité de fois | f) échoue |
| c) affiche « foo » une fois et « bar » une infinité de fois | g) affiche « foo » une fois et « bar » une fois |
| d) affiche juste « foo », une fois | h) affiche juste « bar », une fois |

Question 9 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- | | |
|---|-----------------------------|
| a) affiche « foo » 10000 fois | h) affiche « foo » 10 fois |
| b) affiche « foo » 20 fois | i) affiche « foo » 6 fois |
| c) affiche « foo » 1000 fois | j) n'affiche rien |
| d) affiche « foo » 100 fois | k) affiche « foo » une fois |
| e) affiche « foo » 3 fois | l) affiche « foo » 27 fois |
| f) affiche « foo » une infinité de fois | m) échoue |
| g) affiche « foo » 9 fois | n) affiche « foo » 30 fois |

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Exemplaire numéro 21

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) **non** autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n , n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant « ... »

```
while False:
    print ("foo")
print ("bar")
```

- a) affiche « bar » une infinité de fois
- b) affiche juste « bar », une fois
- c) échoue
- d) n'affiche rien
- e) affiche « foo » une fois et
- f) affiche « foo » une fois et « bar » une fois
- g) affiche « foo » une infinité de fois
- h) affiche juste « foo », une fois

Question 2 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- a) affiche « foo » 10 fois
- b) affiche « foo » 9 fois
- c) affiche « foo » 11 fois
- d) n'affiche rien
- e) affiche « foo » une infinité de fois
- f) échoue

Question 3 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- a) affiche « foo » 6 fois
- b) affiche « foo » 100 fois
- c) affiche « foo » 30 fois
- d) affiche « foo » 10000 fois
- e) échoue
- f) n'affiche rien
- g) affiche « foo » 27 fois
- h) affiche « foo » une fois
- i) affiche « foo » 1000 fois
- j) affiche « foo » 9 fois
- k) affiche « foo » une infinité de fois
- l) affiche « foo » 20 fois
- m) affiche « foo » 10 fois
- n) affiche « foo » 3 fois

Question 4 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- a) affiche « range (x) » une fois
- b) affiche « i » une fois
- c) affiche tout nombre naturel
- d) affiche tout nombre entier (y compris les négatifs)
- e) n'affiche rien
- f) échoue

Question 5 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- a) affiche « bar » une fois
- b) n'affiche rien
- c) échoue
- d) affiche « a » une infinité de fois
- e) affiche « foo » une infinité de fois
- f) affiche « a » une fois
- g) affiche « foo » une fois et « bar » une fois
- h) affiche « foo » une fois
- i) affiche « bar » une infinité de fois

Question 6 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- a) affiche « 7 »
- b) n'affiche rien
- c) affiche « x »
- d) échoue
- e) affiche « foo (7) »
- f) affiche « foo »
- g) affiche « (foo (7)) »
- h) affiche « (foo (x)) »
- i) boucle infiniment

Question 7 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- a) affiche juste « bar », une fois
- b) affiche « foo » une fois et « bar » une fois
- c) affiche « foo » une fois et « bar » une infinité de fois
- d) affiche « foo » une infinité de fois
- e) échoue
- f) affiche juste « foo », une fois
- g) affiche « bar » une infinité de fois
- h) n'affiche rien

Question 8 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- a) échoue
- b) affiche « b »
- c) affiche « 11 »
- d) affiche « 15 »
- e) boucle infiniment
- f) n'affiche rien
- g) affiche « 13 »
- h) affiche « 10 »
- i) affiche « 14 »
- j) affiche « 16 »
- k) affiche « 12 »
- l) affiche « a »

Question 9 (2 points) L'expression suivante ...

[10, 20, 0] * 3

- a) a comme résultat False
- b) a comme résultat une liste
- c) a comme résultat 3
- d) n'a aucun résultat
- e) a comme résultat 9
- f) a comme résultat 0
- g) a comme résultat un dictionnaire
- h) a comme résultat True
- i) a comme résultat 30
- j) échoue
- k) a comme résultat 90

Exemplaire numéro 22

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant « ... »

```
while False:
    print ("foo")
print ("bar")
```

- a) affiche « foo » une fois et « bar » une infinité de fois
- b) affiche « foo » une fois et « bar » une fois
- c) échoue
- d) affiche « foo » une infinité de fois
- e) affiche juste « bar », une fois
- f) n'affiche rien
- g) affiche « bar » une infinité de fois
- h) affiche juste « foo », une fois

- a) affiche « foo » 9 fois
- b) affiche « foo » 10 fois
- c) affiche « foo » une infinité de fois
- d) affiche « foo » 3 fois
- e) affiche « foo » 27 fois
- f) affiche « foo » 1000 fois
- g) affiche « foo » une fois
- h) affiche « foo » 10000 fois
- i) affiche « foo » 30 fois
- j) échoue
- k) affiche « foo » 100 fois
- l) affiche « foo » 6 fois
- m) n'affiche rien
- n) affiche « foo » 20 fois

Question 2 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) a comme résultat une liste
- b) a comme résultat 3
- c) a comme résultat True
- d) a comme résultat False
- e) n'a aucun résultat
- f) a comme résultat 90
- g) a comme résultat un dictionnaire
- h) a comme résultat 30
- i) a comme résultat 9
- j) échoue
- k) a comme résultat 0

Question 6 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- a) affiche tout nombre naturel
- b) affiche « range (x) » une fois
- c) n'affiche rien
- d) affiche « i » une fois
- e) affiche tout nombre entier (y compris les négatifs)
- f) échoue

Question 3 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- a) affiche « foo » une fois et « bar » une fois
- b) affiche juste « bar », une fois
- c) affiche « bar » une infinité de fois
- d) affiche « foo » une infinité de fois
- e) échoue
- f) affiche « foo » une fois et « bar » une infinité de fois
- g) affiche juste « foo », une fois
- h) n'affiche rien

Question 7 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « a »
- b) affiche « 13 »
- c) affiche « 15 »
- d) affiche « 10 »
- e) affiche « 14 »
- f) affiche « 12 »
- g) échoue
- h) affiche « 11 »
- i) n'affiche rien
- j) affiche « b »
- k) affiche « 16 »
- l) boucle infiniment

Question 4 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- a) affiche « bar » une infinité de fois
- b) affiche « foo » une fois et « bar » une fois
- c) échoue
- d) affiche « foo » une fois
- e) affiche « bar » une fois
- f) affiche « a » une fois
- g) affiche « a » une infinité de fois
- h) affiche « foo » une infinité de fois
- i) n'affiche rien

Question 8 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- a) affiche « foo » 10 fois
- b) affiche « foo » une infinité de fois
- c) n'affiche rien
- d) échoue
- e) affiche « foo » 9 fois
- f) affiche « foo » 11 fois

Question 9 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- a) boucle infiniment
- b) échoue
- c) affiche « (foo (7)) »
- d) affiche « x »
- e) n'affiche rien
- f) affiche « 7 »
- g) affiche « foo »
- h) affiche « foo (7) »
- i) affiche « (foo (x)) »

Question 5 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- e) n'affiche rien

Exemplaire numéro 23

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) **non** autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n , n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
def foo (x):  
    for i in range (x):  
        print (i)
```

- a) affiche « range (x) » une fois
- b) échoue
- c) affiche « i » une fois
- d) affiche tout nombre naturel
- e) affiche tout nombre entier (y compris les négatifs)
- f) n'affiche rien

Question 2 (2 points) Le programme suivant ...

```
for i in range (10):  
    for j in range (10):  
        for k in range (10):  
            print ("foo")
```

- a) affiche « foo » 10 fois
- b) affiche « foo » 10000 fois
- c) affiche « foo » 100 fois
- d) n'affiche rien
- e) affiche « foo » 27 fois
- f) affiche « foo » 1000 fois
- g) affiche « foo » une fois
- h) affiche « foo » 3 fois
- i) échoue
- j) affiche « foo » 6 fois
- k) affiche « foo » 20 fois
- l) affiche « foo » une infinité de fois
- m) affiche « foo » 30 fois
- n) affiche « foo » 9 fois

Question 3 (3 points) Le programme suivant ...

```
a = 10  
for b in range (4):  
    a = a + b  
print (a)
```

- a) affiche « 16 »
- b) affiche « 10 »
- c) n'affiche rien
- d) affiche « 12 »
- e) affiche « 11 »
- f) boucle infiniment
- g) affiche « b »
- h) affiche « a »
- i) affiche « 14 »
- j) affiche « 13 »
- k) échoue
- l) affiche « 15 »

Question 4 (3 points) Le programme suivant ...

```
for foo in [True]:  
    print ("foo")  
print ("bar")
```

- a) affiche juste « bar », une fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » une fois et « bar » une fois
- d) affiche « foo » une fois et « bar » une fois
- e) affiche « bar » une infinité de fois
- f) affiche juste « foo », une fois
- g) échoue
- h) n'affiche rien
- i) affiche « bar » une infinité de fois

Question 5 (2 points) L'expression suivante ...

[10, 20, 0] * 3

- a) a comme résultat False
- b) a comme résultat True
- c) a comme résultat un dictionnaire
- d) a comme résultat 9
- e) a comme résultat 0
- f) a comme résultat 30
- g) n'a aucun résultat
- h) a comme résultat 90
- i) a comme résultat 3
- j) échoue
- k) a comme résultat une liste

Question 6 (2 points) Le programme suivant ...

```
def foo (x):  
    return x  
print (foo (7))
```

- a) affiche « (foo (7)) »
- b) n'affiche rien
- c) affiche « foo (7) »
- d) affiche « foo »
- e) échoue
- f) boucle infiniment
- g) affiche « 7 »
- h) affiche « (foo (x)) »
- i) affiche « x »

Question 7 (2 points) Le programme suivant ...

```
a = 10  
while a > 0:  
    print ("foo")
```

- a) échoue
- b) affiche « foo » 11 fois
- c) n'affiche rien
- d) affiche « foo » 9 fois
- e) affiche « foo » 10 fois
- f) affiche « foo » une infinité de fois

Question 8 (2 points) Le programme suivant ...

```
while False:  
    print ("foo")  
print ("bar")
```

- a) échoue
- b) n'affiche rien
- c) affiche « foo » une infinité de fois
- d) affiche juste « foo », une fois
- e) affiche juste « bar », une fois
- f) affiche « foo » une fois et « bar » une infinité de fois
- g) affiche « bar » une infinité de fois
- h) affiche « foo » une fois et « bar » une fois

Question 9 (2 points) Le programme suivant ...

```
a = 8  
if a > 20:  
    print ("foo")  
else a <= 20:  
    print ("bar")
```

- a) échoue
- b) affiche « a » une infinité de fois
- c) affiche « foo » une fois et « bar » une fois
- d) affiche « foo » une fois
- e) affiche « bar » une fois
- f) affiche « foo » une infinité de fois
- g) n'affiche rien
- h) affiche « a » une fois
- i) affiche « bar » une infinité de fois

Exemplaire numéro 24

Programmation Python
Contrôle des connaissances : contrôle surprise numéro 1
IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (3 points) Le programme suivant ...

```
for foo in [True]:  
    print ("foo")  
print ("bar")
```

- a) affiche juste « bar », une fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » une fois et « bar » une fois
- d) affiche « bar » une infinité fois
- e) n'affiche rien
- f) affiche « foo » une fois et « bar » une infinité de fois
- g) échoue
- h) affiche juste « foo », une fois

Question 2 (2 points) Le programme suivant « ... »

```
while False:  
    print ("foo")  
print ("bar")
```

- a) affiche « foo » une fois et « bar » une infinité de fois
- b) affiche « bar » une infinité de fois
- c) n'affiche rien
- d) échoue
- e) affiche « foo » une fois et « bar » une fois
- f) affiche juste « bar », une fois
- g) affiche « foo » une infinité de fois
- h) affiche juste « foo », une fois

Question 3 (2 points) Le programme suivant ...

```
for i in range (10):  
    for j in range (10):  
        for k in range (10):  
            print ("foo")
```

- a) affiche « foo » 10 fois
- b) affiche « foo » 3 fois
- c) affiche « foo » 9 fois
- d) affiche « foo » 6 fois
- e) affiche « foo » 100 fois
- f) affiche « foo » 10000 fois
- g) n'affiche rien
- h) affiche « foo » une fois
- i) affiche « foo » 1000 fois
- j) affiche « foo » 30 fois
- k) affiche « foo » 20 fois
- l) affiche « foo » 27 fois
- m) affiche « foo » une infinité de fois
- n) échoue

Question 4 (2 points) Le programme suivant ...

```
def foo (x):  
    return x  
print (foo (7))
```

- a) affiche « 7 »
- b) échoue
- c) affiche « foo (7) »
- d) affiche « x »
- e) affiche « (foo (x)) »
- f) n'affiche rien
- g) boucle infiniment
- h) affiche « foo »
- i) affiche « (foo (7)) »

Question 5 (2 points) Le programme suivant ...

```
a = 10  
while a > 0:  
    print ("foo")
```

- a) affiche « foo » 11 fois
- b) n'affiche rien
- c) échoue
- d) affiche « foo » 10 fois
- e) affiche « foo » une infinité de fois
- f) affiche « foo » 9 fois

Question 6 (2 points) Le programme suivant ...

```
def foo (x):  
    for i in range (x):  
        print (i)
```

- a) affiche « range (x) » une fois
- b) affiche tout nombre entier (y compris les négatifs)
- c) n'affiche rien
- d) affiche tout nombre naturel
- e) échoue
- f) affiche « i » une fois

Question 7 (2 points) Le programme suivant ...

```
a = 8  
if a > 20:  
    print ("foo")  
else a <= 20:  
    print ("bar")
```

- a) affiche « foo » une infinité de fois
- b) affiche « bar » une infinité de fois
- c) affiche « foo » une fois
- d) affiche « a » une fois
- e) n'affiche rien
- f) affiche « a » une infinité de fois
- g) échoue
- h) affiche « bar » une fois
- i) affiche « foo » une fois et « bar » une fois

Question 8 (3 points) Le programme suivant ...

```
a = 10  
for b in range (4):  
    a = a + b  
print (a)
```

- a) affiche « 11 »
- b) affiche « 10 »
- c) affiche « 12 »
- d) affiche « 14 »
- e) affiche « a »
- f) échoue
- g) affiche « 16 »
- h) n'affiche rien
- i) affiche « 13 »
- j) boucle infiniment
- k) affiche « b »
- l) affiche « 15 »

Question 9 (2 points) L'expression suivante ...

[10, 20, 0] * 3

- a) a comme résultat 9
- b) a comme résultat une liste
- c) a comme résultat 3
- d) a comme résultat 0
- e) a comme résultat True
- f) échoue
- g) n'a aucun résultat
- h) a comme résultat False
- i) a comme résultat 90
- j) a comme résultat un dictionnaire
- k) a comme résultat 30

Exemplaire numéro 25

Programmation Python
Contrôle des connaissances : contrôle surprise numéro 1
IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- a) affiche « foo » 30 fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 100 fois
- d) affiche « foo » 9 fois
- e) affiche « foo » 27 fois
- f) affiche « foo » 10000 fois
- g) affiche « foo » 10 fois
- h) affiche « foo » 20 fois
- i) affiche « foo » 1000 fois
- j) affiche « foo » 6 fois
- k) n'affiche rien
- l) échoue
- m) affiche « foo » une fois
- n) affiche « foo » 3 fois

Question 2 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- a) affiche « foo » une infinité de fois
- b) échoue
- c) affiche « foo » une fois et « bar » une infinité de fois
- d) affiche juste « foo », une fois
- e) affiche « foo » une fois et « bar » une fois
- f) n'affiche rien
- g) affiche juste « bar », une fois
- h) affiche « bar » une infinité de fois

Question 3 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- a) affiche tout nombre naturel
- b) échoue
- c) affiche « range (x) » une fois
- d) affiche « i » une fois
- e) n'affiche rien
- f) affiche tout nombre entier (y compris les négatifs)

Question 4 (2 points) L'expression suivante ...

[10, 20, 0] * 3

- a) a comme résultat True
- b) échoue
- c) a comme résultat 9
- d) a comme résultat 90
- e) a comme résultat un dictionnaire
- f) a comme résultat 3
- g) a comme résultat une liste
- h) n'a aucun résultat
- i) a comme résultat 0
- j) a comme résultat False
- k) a comme résultat 30

Question 5 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- a) échoue
- b) affiche « foo »
- c) affiche « x »
- d) affiche « 7 »
- e) affiche « (foo (7)) »
- f) boucle infiniment
- g) affiche « foo (7) »
- h) affiche « (foo (x)) »
- i) n'affiche rien

Question 6 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 14 »
- b) affiche « 12 »
- c) boucle infiniment
- d) affiche « 11 »
- e) affiche « b »
- f) échoue
- g) affiche « 15 »
- h) affiche « 16 »
- i) affiche « 10 »
- j) n'affiche rien
- k) affiche « 13 »
- l) affiche « a »

Question 7 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- a) affiche « bar » une fois
- b) affiche « bar » une infinité de fois
- c) n'affiche rien
- d) affiche « foo » une infinité de fois
- e) affiche « a » une infinité de fois
- f) affiche « foo » une fois
- g) échoue
- h) affiche « a » une fois
- i) affiche « foo » une fois et « bar » une fois

Question 8 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- a) échoue
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 10 fois
- d) n'affiche rien
- e) affiche « foo » 11 fois
- f) affiche « foo » 9 fois

Question 9 (2 points) Le programme suivant ...

```
while False:
    print ("foo")
print ("bar")
```

- a) affiche juste « foo », une fois
- b) échoue
- c) n'affiche rien
- d) affiche « bar » une infinité de fois
- e) affiche juste « bar », une fois
- f) affiche « foo » une infinité de fois
- g) affiche « foo » une fois et « bar » une fois
- h) affiche « foo » une fois et « bar » une infinité de fois

Exemplaire numéro 26

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
def foo (x):  
    return x  
print (foo (7))
```

- | | |
|--------------------------|--------------------|
| a) affiche « 7 » | f) affiche « foo » |
| b) boucle infiniment | g) échoue |
| c) affiche « foo (7) » | h) n'affiche rien |
| d) affiche « (foo (7)) » | i) affiche « x » |
| e) affiche « (foo (x)) » | |

Question 2 (2 points) Le programme suivant « ...

```
while False:  
    print ("foo")  
print ("bar")
```

- | | |
|---|---|
| a) affiche juste « bar », une fois | e) affiche « bar » une infinité de fois |
| b) n'affiche rien | f) affiche « foo » une fois et « bar » une infinité de fois |
| c) affiche « foo » une infinité de fois | g) affiche « foo » une fois et « bar » une fois |
| d) affiche juste « foo », une fois | h) échoue |

Question 3 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- | | |
|-----------------------------|-------------------------------|
| a) a comme résultat False | tionnaire |
| b) a comme résultat 30 | g) n'a aucun résultat |
| c) a comme résultat 90 | h) a comme résultat une liste |
| d) a comme résultat 0 | i) a comme résultat True |
| e) a comme résultat 9 | j) échoue |
| f) a comme résultat un dic- | k) a comme résultat 3 |

Question 4 (2 points) Le programme suivant ...

```
a = 10  
while a > 0:  
    print ("foo")
```

- | | |
|---|----------------------------|
| a) affiche « foo » 9 fois | d) échoue |
| b) affiche « foo » 11 fois | e) affiche « foo » 10 fois |
| c) affiche « foo » une infinité de fois | f) n'affiche rien |

Question 5 (3 points) Le programme suivant ...

```
for foo in [True]:  
    print ("foo")  
print ("bar")
```

- | | |
|---|---|
| a) affiche « foo » une fois et « bar » une fois | e) échoue |
| b) affiche juste « bar », une fois | f) affiche « foo » une fois et « bar » une infinité de fois |
| c) affiche « foo » une infinité de fois | g) affiche juste « foo », une fois |
| d) n'affiche rien | h) affiche « bar » une infinité de fois |

Question 6 (2 points) Le programme suivant ...

```
def foo (x):  
    for i in range (x):  
        print (i)
```

- | | |
|--|-----------------------------------|
| a) affiche tout nombre entier (y compris les négatifs) | d) n'affiche rien |
| b) affiche tout nombre naturel | e) échoue |
| c) affiche « i » une fois | f) affiche « range (x) » une fois |

Question 7 (2 points) Le programme suivant ...

```
a = 8  
if a > 20:  
    print ("foo")  
else a <= 20:  
    print ("bar")
```

- | | |
|---|---|
| a) échoue | g) affiche « bar » une infinité de fois |
| b) n'affiche rien | h) affiche « foo » une infinité de fois |
| c) affiche « bar » une fois | i) affiche « a » une infinité de fois |
| d) affiche « foo » une fois et « bar » une fois | |
| e) affiche « a » une fois | |
| f) affiche « foo » une fois | |

Question 8 (2 points) Le programme suivant ...

```
for i in range (10):  
    for j in range (10):  
        for k in range (10):  
            print ("foo")
```

- | | |
|---|-------------------------------|
| a) affiche « foo » 20 fois | h) affiche « foo » 10000 fois |
| b) affiche « foo » une infinité de fois | i) n'affiche rien |
| c) affiche « foo » 100 fois | j) affiche « foo » une fois |
| d) affiche « foo » 9 fois | k) affiche « foo » 6 fois |
| e) affiche « foo » 10 fois | l) affiche « foo » 3 fois |
| f) échoue | m) affiche « foo » 27 fois |
| g) affiche « foo » 1000 fois | n) affiche « foo » 30 fois |

Question 9 (3 points) Le programme suivant ...

```
a = 10  
for b in range (4):  
    a = a + b  
print (a)
```

- | | |
|----------------------|-------------------|
| a) n'affiche rien | g) affiche « 15 » |
| b) affiche « 11 » | h) échoue |
| c) affiche « 12 » | i) affiche « 16 » |
| d) affiche « 13 » | j) affiche « a » |
| e) boucle infiniment | k) affiche « 14 » |
| f) affiche « 10 » | l) affiche « b » |

Exemplaire numéro 27

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, polycopiés, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) L'expression suivante ...

[10, 20, 0] * 3

- | | |
|-------------------------------------|----------------------------------|
| a) a comme résultat 9 | f) a comme résultat False |
| b) a comme résultat un dictionnaire | g) n'a aucun résultat |
| c) a comme résultat 3 | h) échoue |
| d) a comme résultat True | i) a comme résultat une liste |
| e) a comme résultat 30 | j) a comme résultat 90 |
| | k) a comme résultat 0 |

- | | |
|---|---------------------------------------|
| a) échoue | de fois |
| b) affiche « foo » une infinité de fois | f) affiche « foo » une fois |
| c) affiche « foo » une fois et « bar » une fois | g) affiche « bar » une fois |
| d) n'affiche rien | h) affiche « a » une infinité de fois |
| e) affiche « bar » une infinité | i) affiche « a » une fois |

Question 2 (3 points) Le programme suivant ...

```
for foo in [True]:  
    print ("foo")  
print ("bar")
```

- | | |
|---|---|
| a) échoue | e) affiche « bar » une infinité de fois |
| b) affiche juste « bar », une fois | f) affiche « foo » une fois et « bar » une infinité de fois |
| c) affiche juste « foo », une fois | g) n'affiche rien |
| d) affiche « foo » une fois et « bar » une fois | h) affiche « foo » une infinité de fois |

Question 6 (2 points) Le programme suivant ...

```
def foo (x):  
    return x  
print (foo (7))
```

- | | |
|--------------------------|--------------------------|
| a) affiche « foo » | f) affiche « x » |
| b) affiche « (foo (x)) » | g) n'affiche rien |
| c) affiche « 7 » | h) affiche « (foo (7)) » |
| d) échoue | i) boucle infiniment |
| e) affiche « foo (7) » | |

Question 3 (2 points) Le programme suivant ...

```
def foo (x):  
    for i in range (x):  
        print (i)
```

- | | |
|--|-----------------------------------|
| a) affiche tout nombre naturel | d) n'affiche rien |
| b) affiche tout nombre entier (y compris les négatifs) | e) échoue |
| c) affiche « i » une fois | f) affiche « range (x) » une fois |

Question 7 (2 points) Le programme suivant ...

```
a = 10  
while a > 0:  
    print ("foo")
```

- | | |
|----------------------------|---|
| a) n'affiche rien | e) affiche « foo » 10 fois |
| b) affiche « foo » 11 fois | f) affiche « foo » une infinité de fois |
| c) échoue | |
| d) affiche « foo » 9 fois | |

Question 4 (2 points) Le programme suivant « ... »

```
while False:  
    print ("foo")  
print ("bar")
```

- | | |
|---|---|
| a) affiche « bar » une infinité de fois | e) affiche juste « foo », une fois |
| b) échoue | f) affiche juste « bar », une fois |
| c) affiche « foo » une fois et « bar » une infinité de fois | g) n'affiche rien |
| d) affiche « foo » une infinité de fois | h) affiche « foo » une fois et « bar » une fois |

Question 8 (2 points) Le programme suivant ...

```
for i in range (10):  
    for j in range (10):  
        for k in range (10):  
            print ("foo")
```

- | | |
|------------------------------|---|
| a) affiche « foo » 1000 fois | i) affiche « foo » 10 fois |
| b) affiche « foo » 6 fois | j) affiche « foo » 10000 fois |
| c) affiche « foo » 20 fois | k) affiche « foo » une infinité de fois |
| d) affiche « foo » 30 fois | |
| e) affiche « foo » 3 fois | l) affiche « foo » 9 fois |
| f) affiche « foo » 27 fois | m) affiche « foo » 100 fois |
| g) affiche « foo » une fois | n) n'affiche rien |
| h) échoue | |

Question 5 (2 points) Le programme suivant ...

```
a = 8  
if a > 20:  
    print ("foo")  
else a <= 20:  
    print ("bar")
```

Question 9 (3 points) Le programme suivant ...

```
a = 10  
for b in range (4):  
    a = a + b  
print (a)
```

- | | |
|----------------------|-------------------|
| a) affiche « 15 » | g) affiche « 16 » |
| b) n'affiche rien | h) affiche « 11 » |
| c) affiche « 14 » | i) affiche « a » |
| d) affiche « 12 » | j) échoue |
| e) affiche « 13 » | k) affiche « 10 » |
| f) boucle infiniment | l) affiche « b » |

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 28

Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- a) affiche « foo » 20 fois
- b) affiche « foo » 100 fois
- c) affiche « foo » une fois
- d) affiche « foo » 9 fois
- e) affiche « foo » 3 fois
- f) affiche « foo » 27 fois
- g) affiche « foo » 10000 fois
- h) affiche « foo » 6 fois
- i) échoue
- j) affiche « foo » 1000 fois
- k) affiche « foo » 10 fois
- l) affiche « foo » 30 fois
- m) affiche « foo » une infinité de fois
- n) n'affiche rien

Question 2 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- a) affiche « foo » une fois et « bar » une fois
- b) affiche « bar » une infinité de fois
- c) affiche juste « foo », une fois
- d) affiche « foo » une infinité de fois
- e) affiche juste « bar », une fois
- f) n'affiche rien
- g) affiche « foo » une fois et « bar » une infinité de fois
- h) échoue

Question 3 (2 points) Le programme suivant « ... »

```
while False:
    print ("foo")
print ("bar")
```

- a) affiche « foo » une infinité de fois
- b) affiche juste « foo », une fois
- c) n'affiche rien
- d) affiche « bar » une infinité de fois
- e) affiche juste « bar », une fois
- f) affiche « foo » une fois et « bar » une infinité de fois
- g) affiche « foo » une fois et « bar » une fois
- h) échoue

Question 4 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 11 »
- b) boucle infiniment
- c) affiche « 16 »
- d) échoue
- e) affiche « a »
- f) affiche « 10 »
- g) affiche « 13 »
- h) affiche « 12 »
- i) affiche « 14 »
- j) affiche « b »
- k) n'affiche rien
- l) affiche « 15 »

Question 5 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- a) boucle infiniment
- b) affiche « (foo (x)) »
- c) échoue
- d) affiche « x »
- e) n'affiche rien
- f) affiche « (foo (7)) »
- g) affiche « foo (7) »
- h) affiche « foo »
- i) affiche « 7 »

Question 6 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- a) échoue
- b) affiche « foo » 11 fois
- c) affiche « foo » 9 fois
- d) n'affiche rien
- e) affiche « foo » une infinité de fois
- f) affiche « foo » 10 fois

Question 7 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- a) n'affiche rien
- b) affiche tout nombre naturel
- c) affiche « i » une fois
- d) échoue
- e) affiche « range (x) » une fois
- f) affiche tout nombre entier (y compris les négatifs)

Question 8 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- a) affiche « a » une fois
- b) affiche « a » une infinité de fois
- c) n'affiche rien
- d) affiche « bar » une infinité de fois
- e) affiche « foo » une fois
- f) affiche « foo » une infinité de fois
- g) affiche « bar » une fois
- h) affiche « foo » une fois et « bar » une fois
- i) échoue

Question 9 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) a comme résultat un dictionnaire
- b) échoue
- c) a comme résultat 30
- d) a comme résultat 90
- e) n'a aucun résultat
- f) a comme résultat une liste
- g) a comme résultat 9
- h) a comme résultat 3
- i) a comme résultat False
- j) a comme résultat 0
- k) a comme résultat True

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Exemplaire numéro 29

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) **non** autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n , n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) L'expression suivante ...

`[10, 20, 0] * 3`

- a) a comme résultat True
- b) a comme résultat 30
- c) a comme résultat 9
- d) a comme résultat 90
- e) n'a aucun résultat
- f) a comme résultat False
- g) a comme résultat 0
- h) échoue
- i) a comme résultat une liste
- j) a comme résultat 3
- k) a comme résultat un dictionnaire

- a) affiche « bar » une fois
- b) affiche « a » une infinité de fois
- c) n'affiche rien
- d) affiche « foo » une infinité de fois
- e) affiche « foo » une fois
- f) échoue
- g) affiche « foo » une fois et « bar » une fois
- h) affiche « bar » une infinité de fois
- i) affiche « a » une fois

Question 2 (3 points) Le programme suivant ...

```
a = 10
for b in range(4):
    a = a + b
print(a)
```

- a) affiche « 13 »
- b) affiche « 11 »
- c) affiche « 16 »
- d) affiche « a »
- e) affiche « 12 »
- f) affiche « 15 »
- g) affiche « b »
- h) affiche « 10 »
- i) échoue
- j) affiche « 14 »
- k) boucle infiniment
- l) n'affiche rien

Question 3 (2 points) Le programme suivant ...

```
def foo(x):
    for i in range(x):
        print(i)
```

- a) affiche tout nombre naturel
- b) échoue
- c) affiche « range(x) » une fois
- d) affiche « i » une fois
- e) n'affiche rien
- f) affiche tout nombre entier (y compris les négatifs)

Question 4 (2 points) Le programme suivant ...

```
for i in range(10):
    for j in range(10):
        for k in range(10):
            print("foo")
```

- a) affiche « foo » 27 fois
- b) affiche « foo » 20 fois
- c) affiche « foo » 3 fois
- d) affiche « foo » 100 fois
- e) affiche « foo » 30 fois
- f) affiche « foo » 6 fois
- g) affiche « foo » 10000 fois
- h) affiche « foo » 9 fois
- i) affiche « foo » une fois
- j) affiche « foo » une infinité de fois
- k) affiche « foo » 10 fois
- l) affiche « foo » 1000 fois
- m) n'affiche rien
- n) échoue

Question 5 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print("foo")
else a <= 20:
    print("bar")
```

Question 6 (2 points) Le programme suivant « ...

```
while False:
    print("foo")
print("bar")
```

- a) échoue
- b) affiche « foo » une infinité de fois
- c) affiche « bar » une infinité de fois
- d) n'affiche rien
- e) affiche juste « bar », une fois
- f) affiche « foo » une fois et « bar » une infinité de fois
- g) affiche « foo » une fois et « bar » une fois
- h) affiche juste « foo », une fois

Question 7 (3 points) Le programme suivant ...

```
for foo in [True]:
    print("foo")
print("bar")
```

- a) affiche juste « foo », une fois
- b) affiche « bar » une infinité de fois
- c) n'affiche rien
- d) affiche « foo » une fois et « bar » une infinité de fois
- e) affiche « foo » une fois et « bar » une fois
- f) échoue
- g) affiche juste « bar », une fois
- h) affiche « foo » une infinité de fois

Question 8 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print("foo")
```

- a) affiche « foo » 10 fois
- b) n'affiche rien
- c) affiche « foo » 11 fois
- d) affiche « foo » 9 fois
- e) affiche « foo » une infinité de fois
- f) échoue

Question 9 (2 points) Le programme suivant ...

```
def foo(x):
    return x
print(foo(7))
```

- a) affiche « x »
- b) affiche « foo »
- c) affiche « foo(7) »
- d) boucle infiniment
- e) n'affiche rien
- f) affiche « (foo(7)) »
- g) échoue
- h) affiche « (foo(x)) »
- i) affiche « 7 »

Exemplaire numéro 30

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- | | |
|-------------------------------------|-------------------------------|
| a) a comme résultat 0 | f) n'a aucun résultat |
| b) a comme résultat False | g) a comme résultat 3 |
| c) a comme résultat 9 | h) a comme résultat 90 |
| d) échoue | i) a comme résultat une liste |
| e) a comme résultat un dictionnaire | j) a comme résultat True |
| | k) a comme résultat 30 |

Question 2 (2 points) Le programme suivant ...

```
def foo (x):
```

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

- | | | |
|-----------------------------------|---------------------------|-------------------|
| a) affiche tout nombre naturel | (y compris les négatifs) | d) n'affiche rien |
| b) affiche « range (x) » une fois | e) affiche « i » une fois | f) échoue |
| c) affiche tout nombre entier | | |

Question 3 (2 points) Le programme suivant « ...

```
while False:
```

```
    print ("foo")
```

```
print ("bar")
```

- | | |
|---|---|
| a) affiche « foo » une fois et « bar » une infinité de fois | e) affiche juste « foo », une fois |
| b) affiche « bar » une infinité de fois | f) n'affiche rien |
| c) échoue | g) affiche « foo » une fois et « bar » une fois |
| d) affiche juste « bar », une fois | h) affiche « foo » une infinité de fois |

Question 4 (2 points) Le programme suivant ...

```
a = 10
```

```
while a > 0:
```

```
    print ("foo")
```

- | | |
|----------------------------|---|
| a) affiche « foo » 10 fois | e) affiche « foo » 9 fois |
| b) échoue | f) affiche « foo » une infinité de fois |
| c) affiche « foo » 11 fois | |
| d) n'affiche rien | |

Question 5 (2 points) Le programme suivant ...

```
a = 8
```

```
if a > 20:
```

```
    print ("foo")
```

```
else a <= 20:
```

```
    print ("bar")
```

- | | |
|---|---|
| a) affiche « a » une infinité de fois | e) affiche « a » une fois |
| b) affiche « bar » une infinité de fois | f) échoue |
| c) affiche « foo » une fois et « bar » une fois | g) affiche « foo » une infinité de fois |
| d) affiche « bar » une fois | h) affiche « foo » une fois |
| | i) n'affiche rien |

Question 6 (2 points) Le programme suivant ...

```
for i in range (10):
```

```
    for j in range (10):
```

```
        for k in range (10):
```

```
            print ("foo")
```

- | | |
|---|-------------------------------|
| a) affiche « foo » une fois | h) affiche « foo » 100 fois |
| b) affiche « foo » 10 fois | i) affiche « foo » 20 fois |
| c) échoue | j) affiche « foo » 6 fois |
| d) affiche « foo » 27 fois | k) affiche « foo » 3 fois |
| e) affiche « foo » une infinité de fois | l) affiche « foo » 1000 fois |
| f) affiche « foo » 9 fois | m) affiche « foo » 10000 fois |
| g) n'affiche rien | n) affiche « foo » 30 fois |

Question 7 (3 points) Le programme suivant ...

```
a = 10
```

```
for b in range (4):
```

```
    a = a + b
```

```
print (a)
```

- | | |
|-------------------|----------------------|
| a) affiche « 15 » | g) affiche « 11 » |
| b) affiche « 13 » | h) affiche « b » |
| c) affiche « a » | i) affiche « 14 » |
| d) affiche « 16 » | j) boucle infiniment |
| e) affiche « 10 » | k) échoue |
| f) n'affiche rien | l) affiche « 12 » |

Question 8 (2 points) Le programme suivant ...

```
def foo (x):
```

```
    return x
```

```
print (foo (7))
```

- | | |
|--------------------------|--------------------------|
| a) affiche « x » | f) boucle infiniment |
| b) affiche « 7 » | g) affiche « foo (7) » |
| c) affiche « (foo (7)) » | h) affiche « foo » |
| d) échoue | i) affiche « (foo (x)) » |
| e) n'affiche rien | |

Question 9 (3 points) Le programme suivant ...

```
for foo in [True]:
```

```
    print ("foo")
```

```
print ("bar")
```

- | | |
|---|---|
| a) n'affiche rien | e) affiche juste « foo », une fois |
| b) affiche « foo » une infinité de fois | f) échoue |
| c) affiche « foo » une fois et « bar » une infinité de fois | g) affiche « foo » une fois et « bar » une fois |
| d) affiche « bar » une infinité de fois | h) affiche juste « bar », une fois |

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 31

Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
for i in range (10):
  for j in range (10):
    for k in range (10):
      print ("foo")
```

- a) échoue
- b) affiche « foo » 30 fois
- c) affiche « foo » une fois
- d) n'affiche rien
- e) affiche « foo » 1000 fois
- f) affiche « foo » 9 fois
- g) affiche « foo » 10000 fois
- h) affiche « foo » une infinité
- i) affiche « foo » 3 fois
- j) affiche « foo » 27 fois
- k) affiche « foo » 20 fois
- l) affiche « foo » 6 fois
- m) affiche « foo » 100 fois
- n) affiche « foo » 10 fois

Question 2 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
  print ("foo")
else a <= 20:
  print ("bar")
```

- a) affiche « a » une fois
- b) affiche « foo » une fois et « bar » une fois
- c) échoue
- d) n'affiche rien
- e) affiche « bar » une infinité de fois
- f) affiche « a » une infinité de fois
- g) affiche « foo » une fois
- h) affiche « bar » une fois
- i) affiche « foo » une infinité de fois

Question 3 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) n'a aucun résultat
- b) a comme résultat un dictionnaire
- c) a comme résultat 0
- d) a comme résultat 9
- e) a comme résultat False
- f) a comme résultat 3
- g) a comme résultat True
- h) échoue
- i) a comme résultat une liste
- j) a comme résultat 30
- k) a comme résultat 90

Question 4 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
  a = a + b
print (a)
```

- a) affiche « 10 »
- b) affiche « 14 »
- c) boucle infiniment
- d) échoue
- e) n'affiche rien
- f) affiche « b »
- g) affiche « a »
- h) affiche « 16 »
- i) affiche « 11 »
- j) affiche « 12 »
- k) affiche « 15 »
- l) affiche « 13 »

Question 5 (3 points) Le programme suivant ...

```
for foo in [True]:
  print ("foo")
print ("bar")
```

- a) échoue
- b) affiche juste « bar », une fois
- c) affiche « foo » une fois et « bar » une fois
- d) affiche « foo » une infinité de fois
- e) affiche juste « foo », une fois
- f) n'affiche rien
- g) affiche « foo » une fois et « bar » une infinité de fois
- h) affiche « bar » une infinité de fois

Question 6 (2 points) Le programme suivant ...

```
def foo (x):
  return x
print (foo (7))
```

- a) n'affiche rien
- b) affiche « 7 »
- c) affiche « (foo (x)) »
- d) affiche « foo »
- e) échoue
- f) affiche « x »
- g) affiche « (foo (7)) »
- h) affiche « foo (7) »
- i) boucle infiniment

Question 7 (2 points) Le programme suivant ...

```
while False:
  print ("foo")
print ("bar")
```

- a) affiche juste « foo », une fois
- b) n'affiche rien
- c) affiche juste « bar », une fois
- d) affiche « foo » une fois et « bar » une fois
- e) affiche « bar » une infinité de fois
- f) échoue
- g) affiche « foo » une infinité de fois
- h) affiche « foo » une fois et « bar » une infinité de fois

Question 8 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
  print ("foo")
```

- a) échoue
- b) affiche « foo » 9 fois
- c) affiche « foo » 11 fois
- d) affiche « foo » une infinité de fois
- e) affiche « foo » 10 fois
- f) n'affiche rien

Question 9 (2 points) Le programme suivant ...

```
def foo (x):
  for i in range (x):
    print (i)
```

- a) affiche « i » une fois
- b) affiche tout nombre entier (y compris les négatifs)
- c) affiche tout nombre naturel
- d) affiche « range (x) » une fois
- e) échoue
- f) n'affiche rien

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 32

Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) **non** autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n , n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- a) n'affiche rien
- b) échoue
- c) affiche « 13 »
- d) affiche « 12 »
- e) affiche « b »
- f) affiche « 15 »
- g) boucle infiniment
- h) affiche « 10 »
- i) affiche « 14 »
- j) affiche « a »
- k) affiche « 16 »
- l) affiche « 11 »

Question 2 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) a comme résultat False
- b) a comme résultat 3
- c) a comme résultat 90
- d) a comme résultat 0
- e) n'a aucun résultat
- f) a comme résultat True
- g) a comme résultat 30
- h) a comme résultat 9
- i) échoue
- j) a comme résultat une liste
- k) a comme résultat un dictionnaire

Question 3 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- a) affiche « foo » 9 fois
- b) affiche « foo » 10 fois
- c) affiche « foo » 11 fois
- d) affiche « foo » une infinité
- e) échoue
- f) n'affiche rien

Question 4 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- a) affiche « (foo (x)) »
- b) affiche « (foo (7)) »
- c) affiche « x »
- d) n'affiche rien
- e) affiche « foo »
- f) boucle infiniment
- g) affiche « 7 »
- h) affiche « foo (7) »
- i) échoue

Question 5 (2 points) Le programme suivant « ... »

```
while False:
    print ("foo")
print ("bar")
```

- a) affiche juste « foo », une fois
- b) affiche « bar » une infinité de fois
- c) échoue
- d) affiche juste « bar », une fois
- e) affiche « foo » une fois et « bar » une infinité de fois
- f) n'affiche rien
- g) affiche « foo » une infinité de fois
- h) affiche « foo » une fois et « bar » une fois

Question 6 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » 1000 fois
- c) affiche « foo » 6 fois
- d) affiche « foo » 20 fois
- e) affiche « foo » 10000 fois
- f) échoue
- g) affiche « foo » 100 fois
- h) affiche « foo » 3 fois
- i) affiche « foo » 27 fois
- j) affiche « foo » une infinité de fois
- k) affiche « foo » 9 fois
- l) affiche « foo » 30 fois
- m) affiche « foo » 10 fois
- n) n'affiche rien

Question 7 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- a) affiche juste « foo », une fois
- b) affiche « bar » une infinité de fois
- c) affiche « foo » une infinité de fois
- d) échoue
- e) affiche « foo » une fois et « bar » une fois
- f) affiche juste « bar », une fois
- g) affiche « foo » une fois et « bar » une infinité de fois
- h) n'affiche rien

Question 8 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- a) affiche tout nombre entier (y compris les négatifs)
- b) n'affiche rien
- c) affiche « i » une fois
- d) affiche « range (x) » une fois
- e) échoue
- f) affiche tout nombre naturel

Question 9 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- a) échoue
- b) affiche « a » une infinité de fois
- c) affiche « a » une fois
- d) affiche « bar » une fois
- e) affiche « bar » une infinité de fois
- f) affiche « foo » une fois et « bar » une fois
- g) affiche « foo » une fois
- h) n'affiche rien
- i) affiche « foo » une infinité de fois

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 33

Luca Saiu

Documents (livres, slides, polycopiés, code source, photocopies, notes manuscrites, etc) **non** autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n , n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- a) affiche « a » une fois
- b) affiche « foo » une fois
- c) affiche « bar » une fois
- d) affiche « a » une infinité de fois
- e) échoue
- f) affiche « foo » une fois et
- « bar » une fois
- g) affiche « bar » une infinité de fois
- h) n'affiche rien
- i) affiche « foo » une infinité de fois

Question 2 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- a) affiche tout nombre naturel
- b) affiche « i » une fois
- c) n'affiche rien
- d) affiche tout nombre entier
- (y compris les négatifs)
- e) affiche « range (x) » une fois
- f) échoue

Question 3 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- a) n'affiche rien
- b) affiche « foo » 10000 fois
- c) affiche « foo » 3 fois
- d) affiche « foo » 10 fois
- e) affiche « foo » 9 fois
- f) affiche « foo » 30 fois
- g) affiche « foo » 6 fois
- h) affiche « foo » 1000 fois
- i) affiche « foo » 27 fois
- j) affiche « foo » une infinité de fois
- k) échoue
- l) affiche « foo » une fois
- m) affiche « foo » 20 fois
- n) affiche « foo » 100 fois

Question 4 (2 points) Le programme suivant « ...

```
while False:
    print ("foo")
print ("bar")
```

- a) échoue
- b) n'affiche rien
- c) affiche « foo » une fois et « bar » une infinité de fois
- d) affiche « foo » une fois et « bar » une fois
- e) affiche juste « bar », une fois
- f) affiche juste « foo », une fois
- g) affiche « bar » une infinité de fois
- h) affiche « foo » une infinité de fois

Question 5 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « b »
- b) affiche « 14 »
- c) affiche « 13 »
- d) affiche « 12 »
- e) échoue
- f) affiche « 10 »
- g) affiche « a »
- h) affiche « 11 »
- i) boucle infiniment
- j) affiche « 15 »
- k) n'affiche rien
- l) affiche « 16 »

Question 6 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- a) affiche « foo » 9 fois
- b) n'affiche rien
- c) affiche « foo » une infinité de fois
- d) affiche « foo » 11 fois
- e) échoue
- f) affiche « foo » 10 fois

Question 7 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) échoue
- b) n'a aucun résultat
- c) a comme résultat 9
- d) a comme résultat 30
- e) a comme résultat 3
- f) a comme résultat 90
- g) a comme résultat False
- h) a comme résultat 0
- i) a comme résultat un dictionnaire
- j) a comme résultat True
- k) a comme résultat une liste

Question 8 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- a) affiche « foo » une fois et « bar » une fois
- b) affiche « bar » une infinité de fois
- c) affiche « foo » une fois et « bar » une infinité de fois
- d) affiche juste « foo », une fois
- e) affiche juste « bar », une fois
- f) échoue
- g) affiche « foo » une infinité de fois
- h) n'affiche rien

Question 9 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- a) affiche « foo »
- b) affiche « (foo (x)) »
- c) affiche « foo (7) »
- d) affiche « x »
- e) affiche « 7 »
- f) échoue
- g) affiche « (foo (7)) »
- h) n'affiche rien
- i) boucle infiniment

Exemplaire numéro 34

Programmation Python
Contrôle des connaissances : contrôle surprise numéro 1
IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (3 points) Le programme suivant ...

```
for foo in [True]:  
    print ("foo")  
print ("bar")
```

- a) n'affiche rien
- b) affiche « bar » une infinité de fois
- c) affiche juste « foo », une fois
- d) échoue
- e) affiche « foo » une fois et
- f) « bar » une infinité de fois
- g) affiche « foo » une infinité de fois
- h) affiche juste « bar », une fois
- i) affiche « foo » une fois et
- j) « bar » une fois

Question 2 (2 points) Le programme suivant ...

```
def foo (x):  
    return x  
print (foo (7))
```

- a) boucle infiniment
- b) affiche « x »
- c) affiche « foo »
- d) affiche « (foo (x)) »
- e) affiche « 7 »
- f) n'affiche rien
- g) affiche « (foo (7)) »
- h) échoue
- i) affiche « foo (7) »

Question 3 (2 points) Le programme suivant ...

```
for i in range (10):  
    for j in range (10):  
        for k in range (10):  
            print ("foo")
```

- a) affiche « foo » 1000 fois
- b) affiche « foo » 10 fois
- c) affiche « foo » 6 fois
- d) affiche « foo » 10000 fois
- e) affiche « foo » une fois
- f) affiche « foo » 27 fois
- g) échoue
- h) affiche « foo » 20 fois
- i) n'affiche rien
- j) affiche « foo » 9 fois
- k) affiche « foo » une infinité de fois
- l) affiche « foo » 100 fois
- m) affiche « foo » 3 fois
- n) affiche « foo » 30 fois

Question 4 (2 points) Le programme suivant ...

```
def foo (x):  
    for i in range (x):  
        print (i)
```

- a) affiche « range (x) » une fois
- b) affiche tout nombre naturel
- c) affiche « i » une fois
- d) échoue
- e) n'affiche rien
- f) affiche tout nombre entier (y compris les négatifs)

Question 5 (3 points) Le programme suivant ...

```
a = 10  
for b in range (4):  
    a = a + b  
print (a)
```

- a) affiche « 12 »
- b) affiche « b »
- c) affiche « 13 »
- d) affiche « a »
- e) affiche « 15 »
- f) affiche « 10 »
- g) affiche « 16 »
- h) n'affiche rien
- i) affiche « 11 »
- j) boucle infiniment
- k) échoue
- l) affiche « 14 »

Question 6 (2 points) Le programme suivant ...

```
a = 10  
while a > 0:  
    print ("foo")
```

- a) affiche « foo » 10 fois
- b) échoue
- c) n'affiche rien
- d) affiche « foo » 11 fois
- e) affiche « foo » une infinité de fois
- f) affiche « foo » 9 fois

Question 7 (2 points) Le programme suivant « ...

```
while False:  
    print ("foo")  
print ("bar")
```

- a) affiche juste « bar », une fois
- b) échoue
- c) affiche juste « foo », une fois
- d) n'affiche rien
- e) affiche « foo » une fois et
- f) « bar » une fois
- g) affiche « bar » une infinité de fois
- h) affiche « foo » une infinité de fois
- i) affiche « foo » une fois et
- j) « bar » une infinité de fois

Question 8 (2 points) Le programme suivant ...

```
a = 8  
if a > 20:  
    print ("foo")  
else a <= 20:  
    print ("bar")
```

- a) échoue
- b) n'affiche rien
- c) affiche « foo » une infinité de fois
- d) affiche « bar » une infinité de fois
- e) affiche « a » une fois
- f) affiche « foo » une fois
- g) affiche « foo » une fois et
- h) « bar » une fois
- i) affiche « bar » une fois
- j) affiche « a » une infinité de fois

Question 9 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) a comme résultat False
- b) a comme résultat 30
- c) n'a aucun résultat
- d) a comme résultat une liste
- e) a comme résultat True
- f) a comme résultat 3
- g) a comme résultat 0
- h) échoue
- i) a comme résultat un dictionnaire
- j) a comme résultat 9
- k) a comme résultat 90

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Exemplaire numéro 35

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant « ... »

```
while False:
    print ("foo")
print ("bar")
```

- a) affiche juste « foo », une fois
- b) affiche « foo » une fois et « bar » une fois
- c) n'affiche rien
- d) échoue
- e) affiche « foo » une fois et « bar » une infinité de fois
- f) affiche « bar » une infinité de fois
- g) affiche « foo » une infinité de fois
- h) affiche juste « bar », une fois

Question 2 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- a) affiche « (foo (x)) »
- b) boucle infiniment
- c) échoue
- d) affiche « x »
- e) affiche « foo »
- f) affiche « (foo (7)) »
- g) affiche « 7 »
- h) n'affiche rien
- i) affiche « foo (7) »

Question 3 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) a comme résultat un dictionnaire
- b) échoue
- c) n'a aucun résultat
- d) a comme résultat 3
- e) a comme résultat 9
- f) a comme résultat 30
- g) a comme résultat False
- h) a comme résultat 0
- i) a comme résultat une liste
- j) a comme résultat True
- k) a comme résultat 90

Question 4 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- a) affiche « foo » 11 fois
- b) échoue
- c) affiche « foo » 9 fois
- d) n'affiche rien
- e) affiche « foo » une infinité de fois
- f) affiche « foo » 10 fois

Question 5 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- a) n'affiche rien
- b) affiche « a »
- c) affiche « 12 »
- d) boucle infiniment
- e) affiche « 10 »
- f) affiche « b »
- g) affiche « 14 »
- h) affiche « 11 »
- i) affiche « 16 »
- j) affiche « 15 »
- k) échoue
- l) affiche « 13 »

Question 6 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- a) affiche « bar » une fois
- b) échoue
- c) affiche « bar » une infinité de fois
- d) affiche « foo » une infinité de fois
- e) n'affiche rien
- f) affiche « a » une infinité de fois
- g) affiche « a » une fois
- h) affiche « foo » une fois et « bar » une fois
- i) affiche « foo » une fois

Question 7 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- a) n'affiche rien
- b) échoue
- c) affiche tout nombre naturel
- d) affiche tout nombre entier
- (y compris les négatifs)
- e) affiche « range (x) » une fois
- f) affiche « i » une fois

Question 8 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- a) affiche « bar » une infinité de fois
- b) affiche « foo » une fois et « bar » une infinité de fois
- c) n'affiche rien
- d) échoue
- e) affiche juste « foo », une fois
- f) affiche « foo » une fois et « bar » une fois
- g) affiche « foo » une infinité de fois
- h) affiche juste « bar », une fois

Question 9 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- a) affiche « foo » 3 fois
- b) échoue
- c) affiche « foo » une infinité de fois
- d) affiche « foo » 100 fois
- e) affiche « foo » 9 fois
- f) affiche « foo » 6 fois
- g) affiche « foo » 20 fois
- h) affiche « foo » une fois
- i) affiche « foo » 1000 fois
- j) affiche « foo » 30 fois
- k) affiche « foo » 10 fois
- l) n'affiche rien
- m) affiche « foo » 27 fois
- n) affiche « foo » 10000 fois

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Exemplaire numéro 36

Documents (livres, slides, polycopiés, code source, photocopies, notes manuscrites, etc) **non** autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n , n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
def foo (x):  
    return x  
print (foo (7))
```

- a) affiche « (foo (x)) »
- b) échoue
- c) affiche « foo »
- d) affiche « 7 »
- e) boucle infiniment
- f) affiche « x »
- g) affiche « (foo (7)) »
- h) affiche « foo (7) »
- i) n'affiche rien

Question 2 (3 points) Le programme suivant ...

```
for foo in [True]:  
    print ("foo")  
print ("bar")
```

- a) affiche juste « foo », une fois
- b) n'affiche rien
- c) affiche « foo » une infinité de fois
- d) affiche « foo » une fois et « bar » une infinité de fois
- e) affiche juste « bar », une fois
- f) affiche « bar » une infinité de fois
- g) échoue
- h) affiche « foo » une fois et « bar » une fois

Question 3 (2 points) Le programme suivant ...

```
a = 10  
while a > 0:  
    print ("foo")
```

- a) affiche « foo » 11 fois
- b) n'affiche rien
- c) affiche « foo » 10 fois
- d) affiche « foo » 9 fois
- e) échoue
- f) affiche « foo » une infinité de fois

Question 4 (2 points) Le programme suivant ...

```
def foo (x):  
    for i in range (x):  
        print (i)
```

- a) affiche tout nombre entier (y compris les négatifs)
- b) affiche tout nombre naturel
- c) échoue
- d) n'affiche rien
- e) affiche « range (x) » une fois
- f) affiche « i » une fois

Question 5 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) a comme résultat 3
- b) a comme résultat False
- c) a comme résultat une liste
- d) échoue
- e) a comme résultat un dictionnaire
- f) a comme résultat 30
- g) a comme résultat 0
- h) n'a aucun résultat
- i) a comme résultat True
- j) a comme résultat 9
- k) a comme résultat 90

Question 6 (2 points) Le programme suivant ...

```
a = 8  
if a > 20:  
    print ("foo")  
else a <= 20:  
    print ("bar")
```

- a) affiche « bar » une fois
- b) affiche « a » une fois
- c) échoue
- d) affiche « foo » une infinité de fois
- e) affiche « bar » une infinité de fois
- f) n'affiche rien
- g) affiche « foo » une fois et « bar » une fois
- h) affiche « foo » une fois
- i) affiche « a » une infinité de fois

Question 7 (2 points) Le programme suivant ...

```
for i in range (10):  
    for j in range (10):  
        for k in range (10):  
            print ("foo")
```

- a) affiche « foo » 10 fois
- b) affiche « foo » 10000 fois
- c) affiche « foo » une fois
- d) n'affiche rien
- e) affiche « foo » 27 fois
- f) affiche « foo » 100 fois
- g) affiche « foo » 9 fois
- h) affiche « foo » 3 fois
- i) affiche « foo » 1000 fois
- j) affiche « foo » 30 fois
- k) affiche « foo » une infinité de fois
- l) affiche « foo » 6 fois
- m) affiche « foo » 20 fois
- n) échoue

Question 8 (2 points) Le programme suivant ...

```
while False:  
    print ("foo")  
print ("bar")
```

- a) n'affiche rien
- b) affiche « foo » une fois et « bar » une fois
- c) affiche « foo » une infinité de fois
- d) affiche juste « foo », une fois
- e) affiche « foo » une fois et « bar » une infinité de fois
- f) affiche « bar » une infinité de fois
- g) affiche juste « bar », une fois
- h) échoue

Question 9 (3 points) Le programme suivant ...

```
a = 10  
for b in range (4):  
    a = a + b  
print (a)
```

- a) affiche « a »
- b) affiche « 10 »
- c) affiche « 11 »
- d) affiche « 12 »
- e) affiche « 14 »
- f) affiche « b »
- g) affiche « 15 »
- h) affiche « 16 »
- i) échoue
- j) boucle infiniment
- k) affiche « 13 »
- l) n'affiche rien

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13

Luca Saiu

Exemplaire numéro 37

Documents (livres, slides, polycopiés, code source, photocopies, notes manuscrites, etc) **non** autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n , n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
a = 10
```

```
while a > 0:
```

```
    print ("foo")
```

- a) n'affiche rien
- b) affiche « foo » 10 fois
- c) échoue
- d) affiche « foo » 9 fois
- e) affiche « foo » une infinité de fois
- f) affiche « foo » 11 fois

Question 2 (3 points) Le programme suivant ...

```
a = 10
```

```
for b in range (4):
```

```
    a = a + b
```

```
print (a)
```

- a) affiche « 16 »
- b) affiche « 15 »
- c) affiche « 10 »
- d) échoue
- e) n'affiche rien
- f) affiche « 14 »
- g) affiche « 11 »
- h) affiche « a »
- i) affiche « 13 »
- j) boucle infiniment
- k) affiche « b »
- l) affiche « 12 »

Question 3 (2 points) Le programme suivant « ... »

```
while False:
```

```
    print ("foo")
```

```
print ("bar")
```

- a) affiche juste « foo », une fois
- b) échoue
- c) affiche « foo » une fois et « bar » une fois
- d) affiche juste « bar », une fois
- e) affiche « foo » une fois et « bar » une infinité de fois
- f) affiche « foo » une infinité de fois
- g) affiche « bar » une infinité de fois
- h) n'affiche rien

Question 4 (2 points) Le programme suivant ...

```
for i in range (10):
```

```
    for j in range (10):
```

```
        for k in range (10):
```

```
            print ("foo")
```

- a) affiche « foo » 100 fois
- b) affiche « foo » 20 fois
- c) affiche « foo » 10000 fois
- d) affiche « foo » 1000 fois
- e) affiche « foo » 9 fois
- f) échoue
- g) affiche « foo » une fois
- h) affiche « foo » 27 fois
- i) affiche « foo » 6 fois
- j) affiche « foo » 30 fois
- k) affiche « foo » 10 fois
- l) n'affiche rien
- m) affiche « foo » 3 fois
- n) affiche « foo » une infinité de fois

Question 5 (2 points) Le programme suivant ...

```
a = 8
```

```
if a > 20:
```

```
    print ("foo")
```

```
else a <= 20:
```

```
    print ("bar")
```

- a) affiche « a » une infinité de fois
- b) affiche « a » une fois
- c) affiche « bar » une infinité de fois
- d) affiche « foo » une infinité de fois
- e) affiche « bar » une fois
- f) échoue
- g) affiche « foo » une fois
- h) affiche « foo » une fois et « bar » une fois
- i) n'affiche rien

Question 6 (3 points) Le programme suivant ...

```
for foo in [True]:
```

```
    print ("foo")
```

```
print ("bar")
```

- a) affiche « foo » une fois et « bar » une infinité de fois
- b) affiche « foo » une infinité de fois
- c) échoue
- d) n'affiche rien
- e) affiche juste « foo », une fois
- f) affiche « foo » une fois et « bar » une fois
- g) affiche « bar » une infinité de fois
- h) affiche juste « bar », une fois

Question 7 (2 points) Le programme suivant ...

```
def foo (x):
```

```
    for i in range (x):
```

```
        print (i)
```

- a) affiche tout nombre entier (y compris les négatifs)
- b) n'affiche rien
- c) affiche « range (x) » une fois
- d) affiche « i » une fois
- e) affiche tout nombre naturel
- f) échoue

Question 8 (2 points) Le programme suivant ...

```
def foo (x):
```

```
    return x
```

```
print (foo (7))
```

- a) affiche « (foo (7)) »
- b) affiche « foo »
- c) affiche « 7 »
- d) affiche « foo (7) »
- e) n'affiche rien
- f) affiche « x »
- g) échoue
- h) boucle infiniment
- i) affiche « (foo (x)) »

Question 9 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) a comme résultat un dictionnaire
- b) a comme résultat 90
- c) échoue
- d) n'a aucun résultat
- e) a comme résultat False
- f) a comme résultat une liste
- g) a comme résultat 9
- h) a comme résultat 0
- i) a comme résultat True
- j) a comme résultat 30
- k) a comme résultat 3

Exemplaire numéro 38

Programmation Python
Contrôle des connaissances : contrôle surprise numéro 1
IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
a = 10
```

```
while a > 0:
```

```
    print ("foo")
```

- a) affiche « foo » 9 fois
- b) n'affiche rien
- c) affiche « foo » 11 fois
- d) échoue
- e) affiche « foo » 10 fois
- f) affiche « foo » une infinité de fois

Question 2 (2 points) Le programme suivant ...

```
for i in range (10):
```

```
    for j in range (10):
```

```
        for k in range (10):
```

```
            print ("foo")
```

- a) affiche « foo » 1000 fois
- b) n'affiche rien
- c) affiche « foo » 6 fois
- d) affiche « foo » une fois
- e) affiche « foo » 20 fois
- f) affiche « foo » 30 fois
- g) échoue
- h) affiche « foo » 10000 fois
- i) affiche « foo » 3 fois
- j) affiche « foo » 27 fois
- k) affiche « foo » 9 fois
- l) affiche « foo » une infinité de fois
- m) affiche « foo » 100 fois
- n) affiche « foo » 10 fois

Question 3 (2 points) Le programme suivant ...

```
def foo (x):
```

```
    for i in range (x):
```

```
        print (i)
```

- a) affiche tout nombre naturel
- b) échoue
- c) affiche tout nombre entier (y compris les négatifs)
- d) affiche « i » une fois
- e) affiche « range (x) » une fois
- f) n'affiche rien

Question 4 (2 points) Le programme suivant ...

```
def foo (x):
```

```
    return x
```

```
print (foo (7))
```

- a) affiche « (foo (x)) »
- b) affiche « foo (7) »
- c) affiche « 7 »
- d) n'affiche rien
- e) affiche « (foo (7)) »
- f) affiche « x »
- g) échoue
- h) boucle infiniment
- i) affiche « foo »

Question 5 (3 points) Le programme suivant ...

```
a = 10
```

```
for b in range (4):
```

```
    a = a + b
```

```
print (a)
```

- a) affiche « 12 »
- b) affiche « 14 »
- c) affiche « b »
- d) affiche « 13 »
- e) échoue
- f) affiche « 11 »
- g) affiche « 16 »
- h) boucle infiniment
- i) affiche « a »
- j) n'affiche rien
- k) affiche « 15 »
- l) affiche « 10 »

Question 6 (2 points) Le programme suivant « ...

```
while False:
```

```
    print ("foo")
```

```
print ("bar")
```

- a) affiche « foo » une infinité de fois
- b) affiche « foo » une fois et « bar » une infinité de fois
- c) affiche « foo » une fois et « bar » une fois
- d) affiche juste « foo », une fois
- e) n'affiche rien
- f) échoue
- g) affiche « bar » une infinité de fois
- h) affiche juste « bar », une fois

Question 7 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) a comme résultat True
- b) a comme résultat un dictionnaire
- c) a comme résultat 30
- d) a comme résultat False
- e) a comme résultat 0
- f) a comme résultat 90
- g) a comme résultat une liste
- h) a comme résultat 9
- i) échoue
- j) a comme résultat 3
- k) n'a aucun résultat

Question 8 (2 points) Le programme suivant ...

```
a = 8
```

```
if a > 20:
```

```
    print ("foo")
```

```
else a <= 20:
```

```
    print ("bar")
```

- a) affiche « a » une fois
- b) affiche « bar » une infinité de fois
- c) affiche « bar » une fois
- d) échoue
- e) n'affiche rien
- f) affiche « foo » une fois et « bar » une fois
- g) affiche « foo » une fois
- h) affiche « a » une infinité de fois
- i) affiche « foo » une infinité de fois

Question 9 (3 points) Le programme suivant ...

```
for foo in [True]:
```

```
    print ("foo")
```

```
print ("bar")
```

- a) affiche « foo » une fois et « bar » une fois
- b) échoue
- c) affiche juste « bar », une fois
- d) affiche juste « foo », une fois
- e) affiche « foo » une infinité de fois
- f) affiche « foo » une fois et « bar » une infinité de fois
- g) affiche « bar » une infinité de fois
- h) n'affiche rien

Exemplaire numéro 39

Programmation Python
Contrôle des connaissances : contrôle surprise numéro 1
IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, polycopiés, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
def foo (x):  
    return x  
print (foo (7))
```

- a) affiche « x »
- b) affiche « (foo (x)) »
- c) n'affiche rien
- d) échoue
- e) boucle infiniment
- f) affiche « 7 »
- g) affiche « foo (7) »
- h) affiche « (foo (7)) »
- i) affiche « foo »

Question 2 (2 points) Le programme suivant ...

```
for i in range (10):  
    for j in range (10):  
        for k in range (10):  
            print ("foo")
```

- a) affiche « foo » 100 fois
- b) affiche « foo » 10 fois
- c) affiche « foo » 6 fois
- d) échoue
- e) affiche « foo » une infinité de fois
- f) affiche « foo » 27 fois
- g) affiche « foo » 20 fois
- h) affiche « foo » 1000 fois
- i) n'affiche rien
- j) affiche « foo » 30 fois
- k) affiche « foo » une fois
- l) affiche « foo » 3 fois
- m) affiche « foo » 10000 fois
- n) affiche « foo » 9 fois

Question 3 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) a comme résultat 9
- b) a comme résultat un dictionnaire
- c) a comme résultat une liste
- d) a comme résultat True
- e) a comme résultat False
- f) a comme résultat 3
- g) a comme résultat 30
- h) échoue
- i) n'a aucun résultat
- j) a comme résultat 0
- k) a comme résultat 90

Question 4 (2 points) Le programme suivant ...

```
def foo (x):  
    for i in range (x):  
        print (i)
```

- a) affiche « range (x) » une fois
- b) affiche tout nombre naturel
- c) n'affiche rien
- d) affiche tout nombre entier (y compris les négatifs)
- e) échoue
- f) affiche « i » une fois

Question 5 (3 points) Le programme suivant ...

```
a = 10  
for b in range (4):  
    a = a + b  
print (a)
```

- a) affiche « a »
- b) affiche « 10 »
- c) affiche « 12 »
- d) boucle infiniment
- e) affiche « b »
- f) n'affiche rien
- g) affiche « 11 »
- h) affiche « 15 »
- i) affiche « 16 »
- j) affiche « 14 »
- k) affiche « 13 »
- l) échoue

Question 6 (3 points) Le programme suivant ...

```
for foo in [True]:  
    print ("foo")  
print ("bar")
```

- a) affiche « bar » une infinité de fois
- b) affiche « foo » une fois et « bar » une fois
- c) affiche « foo » une infinité de fois
- d) n'affiche rien
- e) affiche juste « bar », une fois
- f) échoue
- g) affiche « foo » une fois et « bar » une infinité de fois
- h) affiche juste « foo », une fois

Question 7 (2 points) Le programme suivant ...

```
while False:  
    print ("foo")  
print ("bar")
```

- a) affiche « bar » une infinité de fois
- b) affiche juste « bar », une fois
- c) échoue
- d) n'affiche rien
- e) affiche juste « foo », une fois
- f) affiche « foo » une fois et « bar » une infinité de fois
- g) affiche « foo » une infinité de fois
- h) affiche « foo » une fois et « bar » une fois

Question 8 (2 points) Le programme suivant ...

```
a = 8  
if a > 20:  
    print ("foo")  
else a <= 20:  
    print ("bar")
```

- a) échoue
- b) affiche « a » une infinité de fois
- c) affiche « bar » une infinité de fois
- d) affiche « foo » une fois
- e) affiche « foo » une infinité de fois
- f) affiche « bar » une fois
- g) affiche « a » une fois
- h) affiche « foo » une fois et « bar » une fois
- i) n'affiche rien

Question 9 (2 points) Le programme suivant ...

```
a = 10  
while a > 0:  
    print ("foo")
```

- a) affiche « foo » une infinité de fois
- b) affiche « foo » 9 fois
- c) échoue
- d) affiche « foo » 11 fois
- e) n'affiche rien
- f) affiche « foo » 10 fois

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 40

Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant « ... »

```
while False:
    print ("foo")
print ("bar")
```

- a) affiche « foo » une infinité de fois
- b) échoue
- c) affiche « foo » une fois et « bar » une fois
- d) affiche juste « foo », une fois
- e) n'affiche rien
- f) affiche « bar » une infinité de fois
- g) affiche « foo » une fois et « bar » une infinité de fois
- h) affiche juste « bar », une fois

- a) affiche « foo » 100 fois
- b) affiche « foo » 3 fois
- c) n'affiche rien
- d) affiche « foo » 9 fois
- e) affiche « foo » 10000 fois
- f) affiche « foo » 6 fois
- g) affiche « foo » 27 fois
- h) affiche « foo » une infinité de fois
- i) affiche « foo » une fois
- j) affiche « foo » 20 fois
- k) affiche « foo » 30 fois
- l) échoue
- m) affiche « foo » 10 fois
- n) affiche « foo » 1000 fois

Question 2 (2 points) Le programme suivant « ... »

```
a = 10
while a > 0:
    print ("foo")
```

- a) n'affiche rien
- b) affiche « foo » 10 fois
- c) échoue
- d) affiche « foo » 9 fois
- e) affiche « foo » 11 fois
- f) affiche « foo » une infinité de fois

Question 6 (3 points) Le programme suivant « ... »

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 10 »
- b) affiche « 14 »
- c) affiche « b »
- d) boucle infiniment
- e) affiche « 13 »
- f) affiche « 12 »
- g) affiche « 15 »
- h) affiche « a »
- i) affiche « 11 »
- j) n'affiche rien
- k) échoue
- l) affiche « 16 »

Question 3 (2 points) L'expression suivante « ... »

```
[10, 20, 0] * 3
```

- a) a comme résultat 9
- b) a comme résultat une liste
- c) a comme résultat un dictionnaire
- d) a comme résultat 30
- e) a comme résultat 0
- f) a comme résultat 3
- g) a comme résultat True
- h) échoue
- i) a comme résultat 90
- j) n'a aucun résultat
- k) a comme résultat False

Question 7 (2 points) Le programme suivant « ... »

```
def foo (x):
    return x
print (foo (7))
```

- a) affiche « foo »
- b) échoue
- c) affiche « (foo (7)) »
- d) affiche « foo (7) »
- e) affiche « 7 »
- f) boucle infiniment
- g) n'affiche rien
- h) affiche « x »
- i) affiche « (foo (x)) »

Question 4 (2 points) Le programme suivant « ... »

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- a) affiche « a » une infinité de fois
- b) échoue
- c) affiche « foo » une infinité de fois
- d) affiche « bar » une fois
- e) n'affiche rien
- f) affiche « foo » une fois et « bar » une fois
- g) affiche « bar » une infinité de fois
- h) affiche « a » une fois
- i) affiche « foo » une fois

Question 8 (2 points) Le programme suivant « ... »

```
def foo (x):
    for i in range (x):
        print (i)
```

- a) échoue
- b) affiche « range (x) » une fois
- c) affiche tout nombre entier (y compris les négatifs)
- d) n'affiche rien
- e) affiche tout nombre naturel
- f) affiche « i » une fois

Question 5 (2 points) Le programme suivant « ... »

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

Question 9 (3 points) Le programme suivant « ... »

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- a) affiche juste « foo », une fois
- b) échoue
- c) affiche « foo » une fois et « bar » une infinité de fois
- d) n'affiche rien
- e) affiche « bar » une infinité de fois
- f) affiche juste « bar », une fois
- g) affiche « foo » une fois et « bar » une fois
- h) affiche « foo » une infinité de fois

Exemplaire numéro 41

Programmation Python
Contrôle des connaissances : contrôle surprise numéro 1
IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, polycopiés, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (3 points) Le programme suivant ...

```
for foo in [True]:  
    print ("foo")  
print ("bar")
```

- | | |
|---|---|
| a) affiche juste « bar », une fois | e) échoue |
| b) affiche « foo » une fois et « bar » une infinité de fois | f) n'affiche rien |
| c) affiche juste « foo », une fois | g) affiche « foo » une infinité de fois |
| d) affiche « foo » une fois et de fois | h) affiche « bar » une infinité de fois |

Question 2 (2 points) Le programme suivant « ...

```
while False:  
    print ("foo")  
print ("bar")
```

- | | |
|---|---|
| a) affiche juste « foo », une fois | e) affiche juste « bar », une fois |
| b) affiche « foo » une infinité de fois | f) échoue |
| c) n'affiche rien | g) affiche « foo » une fois et « bar » une infinité de fois |
| d) affiche « bar » une infinité de fois | h) affiche « foo » une fois et « bar » une fois |

Question 3 (2 points) Le programme suivant ...

```
a = 10  
while a > 0:  
    print ("foo")
```

- | | |
|----------------------------|---|
| a) affiche « foo » 10 fois | e) affiche « foo » une infinité de fois |
| b) n'affiche rien | f) échoue |
| c) affiche « foo » 9 fois | |
| d) affiche « foo » 11 fois | |

Question 4 (2 points) Le programme suivant ...

```
def foo (x):  
    for i in range (x):  
        print (i)
```

- | | |
|-----------------------------------|--|
| a) n'affiche rien | d) échoue |
| b) affiche « range (x) » une fois | e) affiche « i » une fois |
| c) affiche tout nombre naturel | f) affiche tout nombre entier (y compris les négatifs) |

Question 5 (2 points) Le programme suivant ...

```
def foo (x):  
    return x  
print (foo (7))
```

- | | |
|--------------------------|--------------------------|
| a) affiche « foo » | f) boucle infiniment |
| b) affiche « 7 » | g) n'affiche rien |
| c) affiche « foo (7) » | h) affiche « x » |
| d) échoue | i) affiche « (foo (7)) » |
| e) affiche « (foo (x)) » | |

Question 6 (3 points) Le programme suivant ...

```
a = 10  
for b in range (4):  
    a = a + b  
print (a)
```

- | | |
|----------------------|-------------------|
| a) échoue | g) affiche « b » |
| b) affiche « 13 » | h) affiche « a » |
| c) affiche « 10 » | i) n'affiche rien |
| d) affiche « 15 » | j) affiche « 12 » |
| e) boucle infiniment | k) affiche « 11 » |
| f) affiche « 14 » | l) affiche « 16 » |

Question 7 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- | | |
|-------------------------------------|-------------------------------|
| a) échoue | f) a comme résultat une liste |
| b) a comme résultat 3 | g) a comme résultat True |
| c) a comme résultat un dictionnaire | h) a comme résultat 9 |
| d) a comme résultat False | i) a comme résultat 0 |
| e) a comme résultat 90 | j) n'a aucun résultat |
| | k) a comme résultat 30 |

Question 8 (2 points) Le programme suivant ...

```
for i in range (10):  
    for j in range (10):  
        for k in range (10):  
            print ("foo")
```

- | | |
|---|-----------------------------|
| a) affiche « foo » 9 fois | h) affiche « foo » 6 fois |
| b) affiche « foo » 10000 fois | i) affiche « foo » 3 fois |
| c) affiche « foo » 20 fois | j) affiche « foo » 100 fois |
| d) affiche « foo » une infinité de fois | k) affiche « foo » une fois |
| e) affiche « foo » 10 fois | l) affiche « foo » 27 fois |
| f) échoue | m) affiche « foo » 30 fois |
| g) affiche « foo » 1000 fois | n) n'affiche rien |

Question 9 (2 points) Le programme suivant ...

```
a = 8  
if a > 20:  
    print ("foo")  
else a <= 20:  
    print ("bar")
```

- | | |
|---|---|
| a) affiche « foo » une fois et « bar » une fois | f) affiche « a » une fois |
| b) affiche « foo » une fois | g) affiche « bar » une infinité de fois |
| c) n'affiche rien | h) affiche « foo » une infinité de fois |
| d) échoue | i) affiche « bar » une fois |
| e) affiche « a » une fois | |
| f) affiche « a » une infinité de fois | |

Exemplaire numéro 42

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- | | |
|-------------------------------|---|
| a) affiche « foo » 1000 fois | i) échoue |
| b) affiche « foo » 20 fois | j) affiche « foo » une fois |
| c) affiche « foo » 10000 fois | k) affiche « foo » 10 fois |
| d) n'affiche rien | l) affiche « foo » une infinité de fois |
| e) affiche « foo » 6 fois | m) affiche « foo » 27 fois |
| f) affiche « foo » 3 fois | n) affiche « foo » 100 fois |
| g) affiche « foo » 9 fois | |
| h) affiche « foo » 30 fois | |

Question 2 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- | | |
|--------------------------|--------------------|
| a) affiche « foo (7) » | f) affiche « foo » |
| b) affiche « (foo (7)) » | g) affiche « 7 » |
| c) boucle infiniment | h) affiche « x » |
| d) affiche « (foo (x)) » | i) échoue |
| e) n'affiche rien | |

Question 3 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- | | |
|-------------------------------------|-------------------------------|
| a) a comme résultat un dictionnaire | f) a comme résultat 0 |
| b) a comme résultat 30 | g) a comme résultat une liste |
| c) échoue | h) n'a aucun résultat |
| d) a comme résultat True | i) a comme résultat 3 |
| e) a comme résultat 90 | j) a comme résultat 9 |
| | k) a comme résultat False |

Question 4 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- | | |
|----------------------|-------------------|
| a) affiche « b » | g) affiche « 10 » |
| b) boucle infiniment | h) affiche « 14 » |
| c) affiche « a » | i) échoue |
| d) affiche « 16 » | j) affiche « 12 » |
| e) affiche « 15 » | k) n'affiche rien |
| f) affiche « 11 » | l) affiche « 13 » |

Question 5 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- | | |
|-----------------------------------|--|
| a) affiche « i » une fois | rel |
| b) échoue | e) n'affiche rien |
| c) affiche « range (x) » une fois | f) affiche tout nombre entier (y compris les négatifs) |
| d) affiche tout nombre naturel | |

Question 6 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- | | |
|---|---|
| a) affiche « a » une fois | de fois |
| b) affiche « a » une infinité de fois | f) affiche « bar » une infinité de fois |
| c) affiche « foo » une fois et « bar » une fois | g) n'affiche rien |
| d) échoue | h) affiche « foo » une fois |
| e) affiche « foo » une infinité | i) affiche « bar » une fois |

Question 7 (2 points) Le programme suivant « ...

```
while False:
    print ("foo")
print ("bar")
```

- | | |
|---|---|
| a) affiche « foo » une infinité de fois | fois |
| b) affiche « foo » une fois et « bar » une fois | e) affiche « foo » une fois et « bar » une infinité de fois |
| c) affiche juste « foo », une fois | f) n'affiche rien |
| d) affiche juste « bar », une fois | g) échoue |
| | h) affiche « bar » une infinité de fois |

Question 8 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- | | |
|----------------------------|---|
| a) n'affiche rien | e) affiche « foo » une infinité de fois |
| b) affiche « foo » 11 fois | f) échoue |
| c) affiche « foo » 9 fois | |
| d) affiche « foo » 10 fois | |

Question 9 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- | | |
|---|---|
| a) affiche « bar » une infinité de fois | e) échoue |
| b) n'affiche rien | f) affiche « foo » une fois et « bar » une fois |
| c) affiche juste « bar », une fois | g) affiche juste « foo », une fois |
| d) affiche « foo » une infinité de fois | h) affiche « foo » une fois et « bar » une infinité de fois |

Exemplaire numéro 43

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) échoue
- b) a comme résultat une liste
- c) a comme résultat 90
- d) a comme résultat False
- e) a comme résultat 30
- f) a comme résultat un dic-
- g) tionnaire
- h) n'a aucun résultat
- i) a comme résultat True
- j) a comme résultat 3
- k) a comme résultat 0
- l) a comme résultat 9

Question 2 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- a) échoue
- b) affiche « 15 »
- c) affiche « b »
- d) affiche « 16 »
- e) affiche « 14 »
- f) n'affiche rien
- g) affiche « 10 »
- h) affiche « 12 »
- i) affiche « 11 »
- j) boucle infiniment
- k) affiche « a »
- l) affiche « 13 »

Question 3 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- a) affiche « foo » une fois
- b) affiche « foo » une infinité de fois
- c) affiche « a » une fois
- d) affiche « a » une infinité de fois
- e) affiche « foo » une fois et
- f) « bar » une fois
- g) n'affiche rien
- h) échoue
- i) affiche « bar » une fois
- j) affiche « bar » une infinité de fois

Question 4 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- a) affiche « foo » 10000 fois
- b) affiche « foo » 9 fois
- c) affiche « foo » 20 fois
- d) affiche « foo » une fois
- e) affiche « foo » 3 fois
- f) affiche « foo » 100 fois
- g) affiche « foo » 30 fois
- h) affiche « foo » 27 fois
- i) affiche « foo » une infinité de fois
- j) n'affiche rien
- k) échoue
- l) affiche « foo » 6 fois
- m) affiche « foo » 1000 fois
- n) affiche « foo » 10 fois

Question 5 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- a) échoue
- b) affiche « foo » 10 fois
- c) affiche « foo » une infinité de fois
- d) n'affiche rien
- e) affiche « foo » 9 fois
- f) affiche « foo » 11 fois

Question 6 (2 points) Le programme suivant « ...

```
while False:
    print ("foo")
print ("bar")
```

- a) affiche « bar » une infinité de fois
- b) affiche juste « bar », une fois
- c) affiche « foo » une infinité de fois
- d) affiche « foo » une fois et
- e) « bar » une infinité de fois
- f) échoue
- g) affiche « foo » une fois et « bar » une fois
- h) n'affiche rien
- i) affiche juste « foo », une fois

Question 7 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- a) affiche tout nombre naturel
- b) n'affiche rien
- c) affiche « range (x) » une fois
- d) échoue
- e) affiche tout nombre entier (y compris les négatifs)
- f) affiche « i » une fois

Question 8 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- a) affiche juste « bar », une fois
- b) affiche « bar » une infinité de fois
- c) affiche « foo » une infinité de fois
- d) n'affiche rien
- e) affiche « foo » une fois et « bar » une infinité de fois
- f) échoue
- g) affiche « foo » une fois et « bar » une fois
- h) affiche juste « foo », une fois

Question 9 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- a) affiche « (foo (x)) »
- b) affiche « foo »
- c) affiche « 7 »
- d) affiche « x »
- e) affiche « (foo (7)) »
- f) boucle infiniment
- g) affiche « foo (7) »
- h) n'affiche rien
- i) échoue

Exemplaire numéro 44

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant « ... »

```
while False:
    print ("foo")
print ("bar")
```

- a) affiche « foo » une fois et « bar » une infinité de fois
- b) affiche juste « bar », une fois
- c) affiche « bar » une infinité de fois
- d) affiche juste « foo », une fois
- e) échoue
- f) affiche « foo » une fois et « bar » une fois
- g) n'affiche rien
- h) affiche « foo » une infinité de fois

- a) affiche « 13 »
- b) affiche « 16 »
- c) affiche « 12 »
- d) affiche « b »
- e) affiche « 11 »
- f) affiche « 15 »
- g) affiche « 10 »
- h) affiche « 14 »
- i) échoue
- j) n'affiche rien
- k) boucle infiniment
- l) affiche « a »

Question 2 (2 points) Le programme suivant « ... »

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- a) affiche « foo » 6 fois
- b) échoue
- c) affiche « foo » 27 fois
- d) affiche « foo » 10 fois
- e) affiche « foo » 3 fois
- f) affiche « foo » 10000 fois
- g) affiche « foo » 30 fois
- h) affiche « foo » une infinité de fois
- i) affiche « foo » 100 fois
- j) affiche « foo » 9 fois
- k) affiche « foo » 1000 fois
- l) n'affiche rien
- m) affiche « foo » une fois
- n) affiche « foo » 20 fois

Question 6 (2 points) Le programme suivant « ... »

```
def foo (x):
    return x
print (foo (7))
```

- a) affiche « 7 »
- b) affiche « foo »
- c) échoue
- d) affiche « x »
- e) boucle infiniment
- f) n'affiche rien
- g) affiche « foo (7) »
- h) affiche « (foo (x)) »
- i) affiche « (foo (7)) »

Question 3 (3 points) Le programme suivant « ... »

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- a) échoue
- b) affiche « foo » une fois et « bar » une infinité de fois
- c) affiche « foo » une infinité de fois
- d) affiche « bar » une infinité de fois
- e) affiche juste « foo », une fois
- f) n'affiche rien
- g) affiche « foo » une fois et « bar » une fois
- h) affiche juste « bar », une fois

Question 7 (2 points) Le programme suivant « ... »

```
a = 10
while a > 0:
    print ("foo")
```

- a) affiche « foo » 10 fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 11 fois
- d) échoue
- e) affiche « foo » 9 fois
- f) n'affiche rien

Question 4 (2 points) Le programme suivant « ... »

```
def foo (x):
    for i in range (x):
        print (i)
```

- a) affiche tout nombre entier (y compris les négatifs)
- b) affiche « range (x) » une fois
- c) affiche « i » une fois
- d) n'affiche rien
- e) échoue
- f) affiche tout nombre naturel

Question 8 (2 points) L'expression suivante « ... »

```
[10, 20, 0] * 3
```

- a) a comme résultat 3
- b) a comme résultat False
- c) a comme résultat 9
- d) a comme résultat 90
- e) a comme résultat True
- f) a comme résultat 0
- g) a comme résultat une liste
- h) a comme résultat 30
- i) échoue
- j) n'a aucun résultat
- k) a comme résultat un dictionnaire

Question 5 (3 points) Le programme suivant « ... »

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

Question 9 (2 points) Le programme suivant « ... »

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- a) échoue
- b) affiche « a » une infinité de fois
- c) affiche « foo » une infinité de fois
- d) affiche « bar » une fois
- e) affiche « foo » une fois et « bar » une fois
- f) affiche « foo » une fois
- g) affiche « a » une fois
- h) n'affiche rien
- i) affiche « bar » une infinité de fois

Exemplaire numéro 45

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, polycopiés, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- | | |
|------------------------------|---|
| a) affiche « foo » 9 fois | i) affiche « foo » 10000 fois |
| b) affiche « foo » 20 fois | j) affiche « foo » 10 fois |
| c) affiche « foo » 100 fois | k) affiche « foo » 30 fois |
| d) affiche « foo » une fois | l) affiche « foo » 3 fois |
| e) affiche « foo » 6 fois | m) affiche « foo » une infinité de fois |
| f) échoue | n) n'affiche rien |
| g) affiche « foo » 1000 fois | |
| h) affiche « foo » 27 fois | |

Question 2 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- | | |
|---|---|
| a) affiche juste « foo », une fois | f) affiche « foo » une fois et « bar » une fois |
| b) n'affiche rien | g) affiche « foo » une infinité de fois |
| c) affiche juste « bar », une fois | h) affiche « foo » une fois et « bar » une infinité de fois |
| d) échoue | |
| e) affiche « bar » une infinité de fois | |

Question 3 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- | | |
|-----------------------------------|--|
| a) affiche « range (x) » une fois | d) affiche tout nombre entier (y compris les négatifs) |
| b) échoue | e) n'affiche rien |
| c) affiche tout nombre naturel | f) affiche « i » une fois |

Question 4 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- | | |
|-------------------------------------|---------------------------|
| a) a comme résultat True | f) n'a aucun résultat |
| b) a comme résultat un dictionnaire | g) a comme résultat 0 |
| c) a comme résultat 3 | h) a comme résultat 9 |
| d) échoue | i) a comme résultat False |
| e) a comme résultat une liste | j) a comme résultat 30 |
| | k) a comme résultat 90 |

Question 5 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- | | |
|--------------------------|------------------------|
| a) affiche « (foo (7)) » | f) n'affiche rien |
| b) boucle infiniment | g) échoue |
| c) affiche « foo » | h) affiche « 7 » |
| d) affiche « (foo (x)) » | i) affiche « foo (7) » |
| e) affiche « x » | |

Question 6 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- | | |
|---|---|
| a) affiche « foo » une fois et « bar » une fois | f) échoue |
| b) n'affiche rien | g) affiche « foo » une infinité de fois |
| c) affiche « bar » une fois | h) affiche « a » une fois |
| d) affiche « foo » une fois | i) affiche « a » une infinité de fois |
| e) affiche « bar » une infinité de fois | |

Question 7 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- | | |
|----------------------------|---|
| a) affiche « foo » 10 fois | e) affiche « foo » 11 fois |
| b) n'affiche rien | f) affiche « foo » une infinité de fois |
| c) affiche « foo » 9 fois | |
| d) échoue | |

Question 8 (2 points) Le programme suivant ...

```
while False:
    print ("foo")
print ("bar")
```

- | | |
|---|---|
| a) affiche « foo » une infinité de fois | « bar » une infinité de fois |
| b) affiche « bar » une infinité de fois | e) n'affiche rien |
| c) affiche juste « bar », une fois | f) échoue |
| d) affiche « foo » une fois et « bar » une fois | g) affiche « foo » une fois et « bar » une fois |
| | h) affiche juste « foo », une fois |

Question 9 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- | | |
|-------------------|----------------------|
| a) n'affiche rien | g) affiche « 16 » |
| b) échoue | h) affiche « a » |
| c) affiche « 10 » | i) boucle infiniment |
| d) affiche « 12 » | j) affiche « 13 » |
| e) affiche « 14 » | k) affiche « 11 » |
| f) affiche « b » | l) affiche « 15 » |

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Exemplaire numéro 46

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
a = 10
```

```
while a > 0:
```

```
    print ("foo")
```

- a) affiche « foo » une infinité de fois
- b) affiche « foo » 10 fois
- c) n'affiche rien
- d) échoue
- e) affiche « foo » 11 fois
- f) affiche « foo » 9 fois

Question 2 (3 points) Le programme suivant ...

```
for foo in [True]:
```

```
    print ("foo")
```

```
print ("bar")
```

- a) affiche juste « bar », une fois
- b) n'affiche rien
- c) affiche juste « foo », une fois
- d) affiche « foo » une fois et « bar » une fois
- e) affiche « foo » une infinité de fois
- f) échoue
- g) affiche « foo » une fois et « bar » une infinité de fois
- h) affiche « bar » une infinité de fois

Question 3 (2 points) Le programme suivant ...

```
def foo (x):
```

```
    for i in range (x):
```

```
        print (i)
```

- a) affiche tout nombre naturel
- b) affiche « i » une fois
- c) échoue
- d) affiche « range (x) » une fois
- e) n'affiche rien
- f) affiche tout nombre entier (y compris les négatifs)

Question 4 (2 points) Le programme suivant « ... »

```
while False:
```

```
    print ("foo")
```

```
print ("bar")
```

- a) affiche « foo » une fois et « bar » une fois
- b) échoue
- c) affiche « foo » une infinité de fois
- d) affiche « foo » une fois et « bar » une infinité de fois
- e) affiche « bar » une infinité de fois
- f) affiche juste « bar », une fois
- g) affiche juste « foo », une fois
- h) n'affiche rien

Question 5 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) a comme résultat 3
- b) a comme résultat 90
- c) a comme résultat 0
- d) échoue
- e) a comme résultat un dictionnaire
- f) a comme résultat False
- g) a comme résultat 9
- h) a comme résultat True
- i) a comme résultat une liste
- j) a comme résultat 30
- k) n'a aucun résultat

Question 6 (2 points) Le programme suivant ...

```
a = 8
```

```
if a > 20:
```

```
    print ("foo")
```

```
else a <= 20:
```

```
    print ("bar")
```

- a) n'affiche rien
- b) affiche « a » une infinité de fois
- c) affiche « bar » une fois
- d) affiche « foo » une fois
- e) échoue
- f) affiche « foo » une fois et « bar » une fois
- g) affiche « bar » une infinité de fois
- h) affiche « foo » une infinité de fois
- i) affiche « a » une fois

Question 7 (3 points) Le programme suivant ...

```
a = 10
```

```
for b in range (4):
```

```
    a = a + b
```

```
print (a)
```

- a) affiche « 11 »
- b) boucle infiniment
- c) affiche « a »
- d) affiche « b »
- e) affiche « 16 »
- f) affiche « 10 »
- g) affiche « 13 »
- h) affiche « 15 »
- i) affiche « 14 »
- j) échoue
- k) n'affiche rien
- l) affiche « 12 »

Question 8 (2 points) Le programme suivant ...

```
def foo (x):
```

```
    return x
```

```
print (foo (7))
```

- a) affiche « foo »
- b) boucle infiniment
- c) affiche « 7 »
- d) n'affiche rien
- e) affiche « (foo (x)) »
- f) affiche « x »
- g) affiche « (foo (7)) »
- h) échoue
- i) affiche « foo (7) »

Question 9 (2 points) Le programme suivant ...

```
for i in range (10):
```

```
    for j in range (10):
```

```
        for k in range (10):
```

```
            print ("foo")
```

- a) affiche « foo » 3 fois
- b) affiche « foo » 1000 fois
- c) affiche « foo » 6 fois
- d) affiche « foo » 10 fois
- e) n'affiche rien
- f) échoue
- g) affiche « foo » une infinité de fois
- h) affiche « foo » 27 fois
- i) affiche « foo » 30 fois
- j) affiche « foo » 100 fois
- k) affiche « foo » 10000 fois
- l) affiche « foo » 20 fois
- m) affiche « foo » une fois
- n) affiche « foo » 9 fois

Exemplaire numéro 47

Programmation Python
Contrôle des connaissances : contrôle surprise numéro 1
IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant « ... »

```
while False:
    print ("foo")
print ("bar")
```

- a) affiche « foo » une fois et « bar » une infinité de fois
- b) affiche « bar » une infinité de fois
- c) affiche « foo » une fois et « bar » une fois
- d) affiche juste « foo », une fois
- e) affiche « foo » une infinité de fois
- f) n'affiche rien
- g) échoue
- h) affiche juste « bar », une fois

- f) affiche « foo » une infinité de fois
- g) affiche « bar » une infinité de fois
- h) échoue
- i) affiche « foo » une fois

Question 2 (2 points) Le programme suivant « ... »

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- a) affiche « foo » 10 fois
- b) affiche « foo » 30 fois
- c) n'affiche rien
- d) affiche « foo » 3 fois
- e) affiche « foo » 20 fois
- f) affiche « foo » 1000 fois
- g) affiche « foo » 27 fois
- h) échoue
- i) affiche « foo » une infinité de fois
- j) affiche « foo » 9 fois
- k) affiche « foo » 10000 fois
- l) affiche « foo » 100 fois
- m) affiche « foo » 6 fois
- n) affiche « foo » une fois

Question 6 (2 points) Le programme suivant « ... »

```
def foo (x):
    for i in range (x):
        print (i)
```

- a) affiche « range (x) » une fois
- b) affiche « i » une fois
- c) échoue
- d) affiche tout nombre naturel
- e) n'affiche rien
- f) affiche tout nombre entier (y compris les négatifs)

Question 3 (2 points) Le programme suivant « ... »

```
def foo (x):
    return x
print (foo (7))
```

- a) affiche « foo (7) »
- b) affiche « 7 »
- c) boucle infiniment
- d) affiche « (foo (7)) »
- e) affiche « foo »
- f) affiche « (foo (x)) »
- g) échoue
- h) affiche « x »
- i) n'affiche rien

Question 7 (2 points) Le programme suivant « ... »

```
a = 10
while a > 0:
    print ("foo")
```

- a) affiche « foo » 11 fois
- b) affiche « foo » 9 fois
- c) affiche « foo » une infinité de fois
- d) affiche « foo » 10 fois
- e) échoue
- f) n'affiche rien

Question 4 (2 points) L'expression suivante « ... »

```
[10, 20, 0] * 3
```

- a) a comme résultat un dictionnaire
- b) a comme résultat 30
- c) a comme résultat 90
- d) échoue
- e) a comme résultat une liste
- f) a comme résultat 0
- g) a comme résultat True
- h) a comme résultat False
- i) a comme résultat 9
- j) a comme résultat 3
- k) n'a aucun résultat

Question 8 (3 points) Le programme suivant « ... »

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- a) affiche « foo » une fois et « bar » une infinité de fois
- b) n'affiche rien
- c) affiche juste « bar », une fois
- d) échoue
- e) affiche juste « foo », une fois
- f) affiche « foo » une infinité de fois
- g) affiche « bar » une infinité de fois
- h) affiche « foo » une fois et « bar » une fois

Question 5 (2 points) Le programme suivant « ... »

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

Question 9 (3 points) Le programme suivant « ... »

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 11 »
- b) affiche « a »
- c) affiche « 13 »
- d) affiche « b »
- e) affiche « 10 »
- f) affiche « 12 »
- g) affiche « 15 »
- h) affiche « 16 »
- i) échoue
- j) boucle infiniment
- k) affiche « 14 »
- l) n'affiche rien

Exemplaire numéro 48

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, polycopiés, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (3 points) Le programme suivant ...

```
a = 10
for b in range (4):
    a = a + b
print (a)
```

- | | |
|----------------------|-------------------|
| a) affiche « 13 » | g) affiche « 12 » |
| b) affiche « b » | h) affiche « a » |
| c) affiche « 15 » | i) affiche « 11 » |
| d) boucle infiniment | j) n'affiche rien |
| e) affiche « 10 » | k) échoue |
| f) affiche « 16 » | l) affiche « 14 » |

Question 2 (2 points) Le programme suivant « ... »

```
while False:
    print ("foo")
print ("bar")
```

- | | |
|---|---|
| a) affiche « foo » une fois et « bar » une infinité de fois | e) échoue |
| b) affiche « bar » une infinité de fois | f) affiche « foo » une infinité de fois |
| c) affiche juste « foo », une fois | g) affiche juste « bar », une fois |
| d) n'affiche rien | h) affiche « foo » une fois et « bar » une fois |

Question 3 (3 points) Le programme suivant ...

```
for foo in [True]:
    print ("foo")
print ("bar")
```

- | | |
|---|---|
| a) affiche « bar » une infinité de fois | e) affiche « foo » une infinité de fois |
| b) affiche « foo » une fois et « bar » une fois | f) affiche juste « bar », une fois |
| c) affiche « foo » une fois et « bar » une infinité de fois | g) affiche juste « foo », une fois |
| d) échoue | h) n'affiche rien |

Question 4 (2 points) Le programme suivant ...

```
a = 8
if a > 20:
    print ("foo")
else a <= 20:
    print ("bar")
```

- | | |
|---|---|
| a) affiche « foo » une fois | g) affiche « foo » une infinité de fois |
| b) affiche « a » une fois | h) affiche « a » une infinité de fois |
| c) échoue | i) affiche « bar » une infinité de fois |
| d) n'affiche rien | j) affiche « bar » une infinité de fois |
| e) affiche « bar » une fois | k) affiche « foo » une fois et « bar » une fois |
| f) affiche « foo » une fois et « bar » une fois | |

Question 5 (2 points) Le programme suivant ...

```
def foo (x):
    for i in range (x):
        print (i)
```

- | | |
|--|--------------------------------|
| a) affiche « i » une fois (y compris les négatifs) | e) affiche tout nombre naturel |
| b) affiche « range (x) » une fois | f) n'affiche rien |
| c) échoue | |
| d) affiche tout nombre entier | |

Question 6 (2 points) Le programme suivant ...

```
a = 10
while a > 0:
    print ("foo")
```

- | | |
|----------------------------|---|
| a) affiche « foo » 9 fois | e) affiche « foo » 11 fois |
| b) n'affiche rien | f) affiche « foo » une infinité de fois |
| c) affiche « foo » 10 fois | |
| d) échoue | |

Question 7 (2 points) Le programme suivant ...

```
def foo (x):
    return x
print (foo (7))
```

- | | |
|--------------------------|--------------------------|
| a) affiche « 7 » | f) boucle infiniment |
| b) affiche « foo » | g) échoue |
| c) affiche « (foo (x)) » | h) affiche « (foo (7)) » |
| d) affiche « foo (7) » | i) affiche « x » |
| e) n'affiche rien | |

Question 8 (2 points) Le programme suivant ...

```
for i in range (10):
    for j in range (10):
        for k in range (10):
            print ("foo")
```

- | | |
|-----------------------------|---|
| a) affiche « foo » 6 fois | i) échoue |
| b) affiche « foo » 100 fois | j) affiche « foo » une fois |
| c) affiche « foo » 27 fois | k) affiche « foo » 10000 fois |
| d) affiche « foo » 20 fois | l) affiche « foo » une infinité de fois |
| e) affiche « foo » 9 fois | m) affiche « foo » 30 fois |
| f) affiche « foo » 10 fois | n) affiche « foo » 1000 fois |
| g) n'affiche rien | |
| h) affiche « foo » 3 fois | |

Question 9 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- | | |
|-------------------------------------|-------------------------------|
| a) a comme résultat 90 | f) échoue |
| b) a comme résultat un dictionnaire | g) a comme résultat False |
| c) a comme résultat True | h) n'a aucun résultat |
| d) a comme résultat 30 | i) a comme résultat 3 |
| e) a comme résultat 9 | j) a comme résultat une liste |
| | k) a comme résultat 0 |

Exemplaire numéro 49

Programmation Python

Contrôle des connaissances : contrôle surprise numéro 1

IUT de Villetaneuse, Université Paris 13
Luca Saiu

Documents (livres, slides, photocopies, code source, photocopies, notes manuscrites, etc) non autorisés. Toute communication entre étudiants, par n'importe quel moyen, est interdite pendant le déroulement de l'épreuve. Téléphones éteints et rangés.

*Toute question a une et une seule réponse correcte. La lettre associée à la réponse choisie est à écrire, lisiblement, à côté de la question ou sous la question. Pour chaque question de valeur n, n points seront ajoutés à la note en cas de réponse correcte et n points seront **soustraits** en cas de réponse erronée. Les réponses non indiquées valent 0 points. Les réponses ou corrections ambiguës seront considérées comme erronées.*

Le barème en bas est donné à titre indicatif.

Durée : 10 minutes

Ici « programme » signifie un programme **complet** mais peut-être écrit de façon incorrecte, stocké dans un fichier exécutable ne contenant **rien d'autre**.

Nom (lisible) : _____

Prénom (lisible) : _____

Question 1 (2 points) Le programme suivant ...

```
def foo (x):  
    for i in range (x):  
        print (i)
```

- a) affiche « i » une fois (y compris les négatifs)
- b) affiche tout nombre naturel
- c) échoue
- d) affiche tout nombre entier
- e) n'affiche rien
- f) affiche « range (x) » une fois

- a) affiche « foo » 30 fois
- b) affiche « foo » 100 fois
- c) affiche « foo » 3 fois
- d) affiche « foo » 10000 fois
- e) affiche « foo » 27 fois
- f) affiche « foo » 9 fois
- g) n'affiche rien
- h) affiche « foo » 1000 fois
- i) affiche « foo » une infinité de fois
- j) affiche « foo » une fois
- k) affiche « foo » 10 fois
- l) échoue
- m) affiche « foo » 6 fois
- n) affiche « foo » 20 fois

Question 2 (3 points) Le programme suivant ...

```
for foo in [True]:  
    print ("foo")  
print ("bar")
```

- a) échoue
- b) affiche « foo » une fois et « bar » une infinité de fois
- c) affiche « foo » une fois et « bar » une fois
- d) affiche « bar » une infinité de fois
- e) affiche juste « bar », une fois
- f) affiche juste « foo », une fois
- g) n'affiche rien
- h) affiche « foo » une infinité de fois

Question 3 (2 points) Le programme suivant ...

```
def foo (x):  
    return x  
print (foo (7))
```

- a) affiche « x »
- b) affiche « 7 »
- c) affiche « (foo (x)) »
- d) boucle infiniment
- e) affiche « foo »
- f) n'affiche rien
- g) affiche « foo (7) »
- h) échoue
- i) affiche « (foo (7)) »

Question 4 (3 points) Le programme suivant ...

```
a = 10  
for b in range (4):  
    a = a + b  
print (a)
```

- a) affiche « b »
- b) n'affiche rien
- c) affiche « 11 »
- d) affiche « 13 »
- e) affiche « 15 »
- f) affiche « 12 »
- g) boucle infiniment
- h) affiche « a »
- i) échoue
- j) affiche « 14 »
- k) affiche « 16 »
- l) affiche « 10 »

Question 5 (2 points) Le programme suivant ...

```
for i in range (10):  
    for j in range (10):  
        for k in range (10):  
            print ("foo")
```

Question 6 (2 points) Le programme suivant « ...

```
while False:  
    print ("foo")  
print ("bar")
```

- a) affiche « foo » une fois et « bar » une infinité de fois
- b) échoue
- c) affiche juste « foo », une fois
- d) n'affiche rien
- e) affiche « foo » une fois et « bar » une fois
- f) affiche « bar » une infinité de fois
- g) affiche « foo » une infinité de fois
- h) affiche juste « bar », une fois

Question 7 (2 points) L'expression suivante ...

```
[10, 20, 0] * 3
```

- a) a comme résultat 3
- b) a comme résultat 30
- c) échoue
- d) a comme résultat 90
- e) a comme résultat 0
- f) a comme résultat 9
- g) a comme résultat True
- h) a comme résultat False
- i) a comme résultat une liste
- j) a comme résultat un dictionnaire
- k) n'a aucun résultat

Question 8 (2 points) Le programme suivant ...

```
a = 8  
if a > 20:  
    print ("foo")  
else a <= 20:  
    print ("bar")
```

- a) affiche « bar » une fois
- b) échoue
- c) affiche « foo » une fois
- d) affiche « foo » une infinité de fois
- e) affiche « foo » une fois et « bar » une fois
- f) n'affiche rien
- g) affiche « a » une infinité de fois
- h) affiche « a » une fois
- i) affiche « bar » une infinité de fois

Question 9 (2 points) Le programme suivant ...

```
a = 10  
while a > 0:  
    print ("foo")
```

- a) échoue
- b) affiche « foo » 10 fois
- c) n'affiche rien
- d) affiche « foo » 11 fois
- e) affiche « foo » 9 fois
- f) affiche « foo » une infinité de fois

Solution (generated on : 10 octobre 2018)

Each solution item has the form a_p , where a is the correct answer label and p is the value in points. Questions are ordered like in each paper, but not numbered here.

Paper 0 : $f_2 d_2 a_2 c_2 h_2 e_3 h_3 a_2 b_2$

Paper 1 : $e_2 h_3 j_3 e_2 f_2 f_2 h_2 i_2 b_2$

Paper 2 : $b_2 c_2 f_3 n_2 e_2 c_2 c_2 a_2 d_3$

Paper 3 : $f_2 e_2 f_3 e_2 i_2 m_2 g_2 g_3 g_2$

Paper 4 : $g_2 b_2 g_2 k_2 f_2 d_3 d_2 a_3 e_2$

Paper 5 : $g_2 a_2 d_2 a_2 i_2 e_3 f_3 d_2 c_2$

Paper 6 : $c_2 g_2 a_2 f_2 f_2 h_2 f_2 e_3 b_3$

Paper 7 : $e_2 d_3 a_2 h_3 f_2 e_2 h_2 e_2 b_2$

Paper 8 : $d_2 c_2 a_2 a_3 b_2 f_2 e_2 d_3 j_2$

Paper 9 : $c_2 b_2 f_3 i_2 a_2 a_2 f_3 b_2 e_2$

Paper 10 : $f_2 b_3 d_2 g_2 f_2 f_2 c_2 g_2 c_3$

Paper 11 : $i_2 b_3 b_2 a_2 g_3 g_2 c_2 a_2 c_2$

Paper 12 : $b_2 g_3 k_2 h_3 d_2 i_2 b_2 g_2 e_2$

Paper 13 : $f_2 e_3 c_2 f_2 h_2 d_2 a_2 j_2 e_3$

Paper 14 : $n_2 k_2 c_2 l_3 a_2 b_2 c_2 b_2 f_3$

Paper 15 : $d_2 e_2 c_2 i_2 c_2 h_3 a_2 e_2 c_3$

Paper 16 : $d_3 f_2 d_2 i_3 g_2 g_2 d_2 m_2 h_2$

Paper 17 : $c_2 e_3 c_2 a_2 a_2 i_3 i_2 f_2 d_2$

Paper 18 : $n_2 d_2 g_2 i_2 e_2 i_3 e_2 b_2 d_3$

Paper 19 : $g_3 c_3 h_2 f_2 b_2 a_2 e_2 b_2 l_2$

Paper 20 : $h_2 f_3 e_2 e_3 g_2 c_2 e_2 h_2 c_2$

Paper 21 : $b_2 e_2 i_2 e_2 c_2 a_2 b_3 j_3 b_2$

Paper 22 : $e_2 a_2 a_3 c_2 f_2 c_2 k_3 b_2 f_2$

Paper 23 : $f_2 f_2 a_3 c_3 k_2 g_2 f_2 e_2 a_2$

Paper 24 : $c_3 f_2 i_2 a_2 e_2 c_2 g_2 g_3 b_2$

Paper 25 : $i_2 e_3 e_2 g_2 d_2 h_3 g_2 b_2 e_2$

Paper 26 : $a_2 a_2 h_2 c_2 a_3 d_2 a_2 g_2 i_3$

Paper 27 : $i_2 d_3 d_2 f_2 a_2 c_2 f_2 a_2 g_3$

Paper 28 : $j_2 a_3 e_2 c_3 i_2 e_2 a_2 i_2 f_2$

Paper 29 : $i_2 c_3 e_2 l_2 f_2 e_2 e_3 e_2 i_2$

Paper 30 : $i_2 d_2 d_2 f_2 f_2 l_2 d_3 b_2 g_3$

Paper 31 : $e_2 c_2 i_2 h_3 c_3 b_2 c_2 d_2 f_2$

Paper 32 : $k_3 j_2 d_2 g_2 d_2 b_2 e_3 b_2 a_2$

Paper 33 : $e_2 c_2 h_2 e_2 l_3 c_2 k_2 a_3 e_2$

Paper 34 : $h_3 e_2 a_2 e_2 g_3 e_2 a_2 a_2 d_2$

Paper 35 : $h_2 g_2 i_2 e_2 i_3 b_2 a_2 f_3 i_2$

Paper 36 : $d_2 h_3 f_2 d_2 c_2 c_2 i_2 g_2 h_3$

Paper 37 : $e_2 a_3 d_2 d_2 f_2 f_3 b_2 c_2 f_2$

Paper 38 : $f_2 a_2 f_2 c_2 g_3 h_2 g_2 d_2 a_3$

Paper 39 : $f_2 h_2 c_2 c_2 i_3 b_3 b_2 a_2 a_2$

Paper 40 : $h_2 f_2 b_2 b_2 n_2 l_3 e_2 d_2 g_3$

Paper 41 : $d_3 e_2 e_2 a_2 b_2 l_3 f_2 g_2 d_2$

Paper 42 : $a_2 g_2 g_2 d_3 e_2 d_2 d_2 e_2 f_3$

Paper 43 : $b_2 d_3 g_2 m_2 c_2 b_2 b_2 g_3 c_2$

Paper 44 : $b_2 k_2 g_3 d_2 b_3 a_2 b_2 g_2 a_2$

Paper 45 : $g_2 f_3 e_2 e_2 h_2 f_2 f_2 c_2 g_3$

Paper 46 : $a_2 d_3 e_2 f_2 i_2 e_2 e_3 c_2 b_2$

Paper 47 : $h_2 f_2 b_2 e_2 h_2 e_2 c_2 h_3 h_3$

Paper 48 : $f_3 g_2 b_3 c_2 f_2 f_2 a_2 n_2 j_2$

Paper 49 : $e_2 c_3 b_2 k_3 h_2 h_2 i_2 b_2 f_2$