

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 0

Luca Saiu

Documents (livres, slides, photocopies, code source, photocopiers, 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 ...

```
"foo" * 3
```

- a) a comme résultat un entier
- b) échoue
- c) a comme résultat un booléen
- d) a comme résultat une chaîne de caractères
- e) a comme résultat un float-tant
- f) a comme résultat la valeur de la variable `foo` multipliée par 3
- g) n'a aucun résultat

- a) affiche « `foo` » une fois et « `bar` » une fois
- b) affiche « `x` » une fois
- c) n'affiche rien
- d) affiche « `bar` » une fois
- e) affiche « `foo` » une fois
- f) échoue
- g) affiche « `x` » une infinité de fois
- h) affiche « `bar` » une infinité de fois
- i) 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` » 10000 fois
- b) n'affiche rien
- c) affiche « `foo` » 1000 fois
- d) affiche « `foo` » 9 fois
- e) affiche « `foo` » 30 fois
- f) affiche « `foo` » une fois
- g) affiche « `foo` » 3 fois
- h) affiche « `foo` » 10 fois
- i) affiche « `foo` » 20 fois
- j) affiche « `foo` » 27 fois
- k) affiche « `foo` » 100 fois
- l) échoue
- m) affiche « `foo` » 33 fois
- n) affiche « `foo` » une infinité de fois

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

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

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

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

```
if True:
    print("foo")
print("foo")
else:
    print("foo")
```

- a) affiche « `foo` » trois fois
- b) affiche « `foo` » quatre fois
- c) affiche « `foo` » une fois
- d) n'affiche rien
- e) échoue
- f) affiche « `foo` » une infinité de fois
- g) affiche « `foo` » deux fois

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

```
x = 8
if x > 7:
    print("foo")
else x <= 7:
    print("bar")
```

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

```
a = 8
for b in range(4):
    a = a + b
print(a)
```

- a) affiche « `b` »
- b) boucle infiniment
- c) n'affiche rien
- d) affiche « `14` »
- e) affiche « `12` »
- f) affiche « `13` »
- g) affiche « `11` »
- h) affiche « `8` »
- i) affiche « `9` »
- j) échoue
- k) affiche « `a` »
- l) affiche « `10` »

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

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

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

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

```
a = 7
while a < 8:
    print("foo")
```

- a) n'affiche rien
- b) affiche « `foo` » une infinité de fois
- c) affiche « `foo` » une fois
- d) affiche « `foo` » 8 fois
- e) affiche « `foo` » 7 fois
- f) échoue
- g) affiche « `foo` » 6 fois

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

```
for i in range(3):
    print("foo")
for j in range(2):
    print("foo")
```

- a) affiche « `foo` » 2 fois
- b) affiche « `foo` » 3 fois
- c) affiche « `foo` » une fois
- d) affiche « `foo` » 6 fois
- e) n'affiche rien
- f) échoue
- g) affiche « `foo` » une infinité de fois
- h) affiche « `foo` » 5 fois
- i) affiche « `foo` » 4 fois

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 1

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

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

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

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

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) n'affiche rien
- b) affiche « foo » 10 fois
- c) échoue
- d) affiche « foo » 3 fois
- e) affiche « foo » 9 fois
- f) affiche « foo » 10000 fois
- g) affiche « foo » 33 fois
- h) affiche « foo » une infinité de fois
- i) affiche « foo » 1000 fois
- j) affiche « foo » 100 fois
- k) affiche « foo » 20 fois
- l) affiche « foo » 30 fois
- m) affiche « foo » une fois
- n) affiche « foo » 27 fois

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) échoue
- b) n'affiche rien
- c) affiche « foo » une fois
- d) affiche « foo » une infinité de fois
- e) affiche « foo » quatre fois
- f) affiche « foo » trois fois
- g) affiche « foo » deux fois

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » 8 fois
- c) affiche « foo » 7 fois
- d) affiche « foo » une infinité de fois
- e) échoue
- f) affiche « foo » 6 fois
- g) n'affiche rien

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 12 »
- b) affiche « b »
- c) affiche « 9 »
- d) affiche « 14 »
- e) affiche « a »
- f) affiche « 11 »
- g) affiche « 10 »
- h) affiche « 8 »
- i) affiche « 13 »
- j) échoue
- k) n'affiche rien
- l) boucle infiniment

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

```
"foo" * 3
```

- a) n'a aucun résultat
- b) a comme résultat un entier
- c) échoue
- d) a comme résultat un float-tant
- e) a comme résultat un boolean
- f) léen
- g) a comme résultat une chaîne de caractères
- h) a comme résultat la valeur de la variable foo multipliée par 3

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 5 fois
- b) n'affiche rien
- c) affiche « foo » une fois
- d) affiche « foo » 4 fois
- e) affiche « foo » 6 fois
- f) affiche « foo » 3 fois
- g) affiche « foo » une infinité de fois
- h) affiche « foo » 2 fois
- i) échoue

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

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

- a) affiche « foo » 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 « foo » une fois et « bar » une infinité de fois
- g) affiche « bar » une infinité de fois
- h) affiche juste « bar », une fois

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 2

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) n'affiche rien
- b) échoue
- c) affiche « foo » une fois
- d) affiche « foo » deux fois
- e) affiche « foo » une infinité de fois
- f) affiche « foo » trois fois
- g) affiche « foo » quatre fois

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 2 fois
- b) n'affiche rien
- c) affiche « foo » 4 fois
- d) affiche « foo » une infinité de fois
- e) échoue
- f) affiche « foo » une fois
- g) affiche « foo » 5 fois
- h) affiche « foo » 6 fois
- i) affiche « foo » 3 fois

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

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

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

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 14 »
- b) affiche « 10 »
- c) affiche « 11 »
- d) affiche « 13 »
- e) affiche « 9 »
- f) affiche « a »
- g) affiche « b »
- h) boucle infiniment
- i) affiche « 8 »
- j) affiche « 12 »
- k) échoue
- l) n'affiche rien

Question 5 (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 « bar » une infinité de fois
- d) échoue
- e) n'affiche rien
- f) affiche « foo » une fois et « bar » une fois
- g) affiche juste « bar », une fois
- h) affiche juste « foo », 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 » 20 fois
- b) affiche « foo » 27 fois
- c) affiche « foo » 10 fois
- d) affiche « foo » une fois
- e) affiche « foo » 33 fois
- f) affiche « foo » 9 fois
- g) affiche « foo » 10000 fois
- h) affiche « foo » 100 fois
- i) affiche « foo » une infinité de fois
- j) échoue
- k) affiche « foo » 1000 fois
- l) n'affiche rien
- m) affiche « foo » 30 fois
- n) affiche « foo » 3 fois

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » 6 fois
- b) affiche « foo » 7 fois
- c) affiche « foo » 8 fois
- d) échoue
- e) affiche « foo » une infinité de fois
- f) affiche « foo » une fois
- g) n'affiche rien

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

```
"foo" * 3
```

- a) a comme résultat un float-tant pliée par 3
- b) n'a aucun résultat
- c) a comme résultat un entier
- d) a comme résultat la valeur de la variable foo multi-
- e) a comme résultat un booléen
- f) échoue
- g) a comme résultat une chaîne de caractères

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 3

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

```
"foo" * 3
```

- a) a comme résultat la valeur de la variable `foo` multipliée par 3
- b) a comme résultat un booléen
- c) a comme résultat un float
- d) échoue
- e) n'a aucun résultat
- f) a comme résultat un entier
- g) a comme résultat une chaîne de caractères

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

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

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

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

```
a = 7  
while a < 8:  
    print ("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » 6 fois
- c) n'affiche rien
- d) échoue
- e) affiche « foo » 8 fois
- f) affiche « foo » 7 fois
- g) affiche « foo » une infinité de fois

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

```
a = 8  
for b in range (4):  
    a = a + b  
print (a)
```

- a) affiche « b »
- b) affiche « 8 »
- c) affiche « 13 »
- d) affiche « 14 »
- e) affiche « a »
- f) affiche « 11 »
- g) échoue
- h) boucle infiniment
- i) affiche « 9 »
- j) affiche « 10 »
- k) n'affiche rien
- l) affiche « 12 »

Question 5 (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 » 9 fois
- c) affiche « foo » 100 fois
- d) affiche « foo » 10000 fois
- e) affiche « foo » 3 fois
- f) affiche « foo » 10 fois
- g) affiche « foo » 33 fois
- h) affiche « foo » 30 fois
- i) affiche « foo » 1000 fois
- j) affiche « foo » 20 fois
- k) affiche « foo » une fois
- l) n'affiche rien
- m) échoue
- n) affiche « foo » une infinité de fois

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

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

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

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

```
if True:  
    print ("foo")  
print ("foo")  
else:  
    print ("foo")
```

- a) affiche « foo » deux fois
- b) affiche « foo » une fois
- c) affiche « foo » quatre fois
- d) échoue
- e) affiche « foo » une infinité de fois
- f) affiche « foo » trois fois
- g) n'affiche rien

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

```
x = 8  
if x > 7:  
    print ("foo")  
else x <= 7:  
    print ("bar")
```

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

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

```
for i in range (3):  
    print ("foo")  
for j in range (2):  
    print ("foo")
```

- a) affiche « foo » 3 fois
- b) affiche « foo » 5 fois
- c) affiche « foo » 2 fois
- d) affiche « foo » une infinité de fois
- e) échoue
- f) affiche « foo » 4 fois
- g) affiche « foo » 6 fois
- h) n'affiche rien
- i) affiche « foo » une fois

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villeteuse, Université Paris 13

Exemplaire numéro 4

Luca Saiu

Documents (livres, slides, photocopiés, 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 (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 6 fois
- b) échoue
- c) n'affiche rien
- d) affiche « foo » une fois
- e) affiche « foo » 4 fois
- f) affiche « foo » 5 fois
- g) affiche « foo » 3 fois
- h) affiche « foo » 2 fois
- i) affiche « foo » une infinité de fois

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 13 »
- b) boucle infiniment
- c) affiche « 14 »
- d) affiche « b »
- e) n'affiche rien
- f) affiche « 12 »
- g) affiche « a »
- h) affiche « 8 »
- i) échoue
- j) affiche « 10 »
- k) affiche « 9 »
- l) affiche « 11 »

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

```
"foo" * 3
```

- a) a comme résultat un entier
- b) échoue
- c) n'a aucun résultat
- d) a comme résultat une chaîne de caractères
- e) a comme résultat un booléen
- f) léen
- g) a comme résultat un float-tant
- h) a comme résultat la valeur de la variable foo multipliée par 3

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » 8 fois
- b) affiche « foo » une infinité de fois
- c) échoue
- d) affiche « foo » une fois
- e) affiche « foo » 6 fois
- f) n'affiche rien
- g) affiche « foo » 7 fois

Question 5 (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 « foo » une fois et « bar » une infinité de fois
- e) affiche juste « bar », une fois
- f) affiche « foo » une fois et « bar » une fois
- g) affiche juste « foo », une fois
- h) affiche « bar » une infinité de 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 » 33 fois
- c) affiche « foo » 10000 fois
- d) affiche « foo » une fois
- e) échoue
- f) affiche « foo » 9 fois
- g) affiche « foo » une infinité de fois
- h) affiche « foo » 20 fois
- i) n'affiche rien
- j) affiche « foo » 27 fois
- k) affiche « foo » 30 fois
- l) affiche « foo » 100 fois
- m) affiche « foo » 3 fois
- n) affiche « foo » 1000 fois

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

```
if 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 (2 points) Le programme suivant ...

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » quatre fois
- b) échoue
- c) affiche « foo » trois fois
- d) affiche « foo » une fois
- e) affiche « foo » deux fois
- f) n'affiche rien
- g) affiche « foo » une infinité de fois

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 5

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

```
"foo" * 3
```

- a) a comme résultat un entier
- b) a comme résultat une chaîne de caractères
- c) échoue
- d) a comme résultat un booléen
- e) a comme résultat un float-tant
- f) n'a aucun résultat
- g) a comme résultat la valeur de la variable `foo` multipliée par 3

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

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

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

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

```
a = 7  
while a < 8:  
    print ("foo")
```

- a) affiche « foo » une infinité de fois
- b) affiche « foo » une fois
- c) affiche « foo » 6 fois
- d) affiche « foo » 7 fois
- e) n'affiche rien
- f) affiche « foo » 8 fois
- g) échoue

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

```
for i in range (3):  
    print ("foo")  
for j in range (2):  
    print ("foo")
```

- a) n'affiche rien
- b) affiche « foo » une fois
- c) échoue
- d) affiche « foo » 6 fois
- e) affiche « foo » une infinité de fois
- f) affiche « foo » 4 fois
- g) affiche « foo » 5 fois
- h) affiche « foo » 3 fois
- i) affiche « foo » 2 fois

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

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

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

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

```
x = 8  
if x > 7:  
    print ("foo")  
else x <= 7:  
    print ("bar")
```

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

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

```
a = 8  
for b in range (4):  
    a = a + b  
print (a)
```

- a) affiche « 11 »
- b) affiche « 13 »
- c) affiche « a »
- d) affiche « 10 »
- e) n'affiche rien
- f) affiche « 12 »
- g) échoue
- h) affiche « 9 »
- i) affiche « b »
- j) affiche « 14 »
- k) affiche « 8 »
- l) boucle infiniment

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 » 10 fois
- b) affiche « foo » 30 fois
- c) affiche « foo » 20 fois
- d) affiche « foo » 1000 fois
- e) affiche « foo » 10000 fois
- f) affiche « foo » 27 fois
- g) affiche « foo » une infinité de fois
- h) échoue
- i) affiche « foo » 3 fois
- j) affiche « foo » 33 fois
- k) affiche « foo » 100 fois
- l) affiche « foo » une fois
- m) affiche « foo » 9 fois
- n) n'affiche rien

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

```
if True:  
    print ("foo")  
print ("foo")  
else:  
    print ("foo")
```

- a) affiche « foo » trois fois
- b) affiche « foo » quatre fois
- c) n'affiche rien
- d) affiche « foo » une infinité de fois
- e) échoue
- f) affiche « foo » deux fois
- g) affiche « foo » une fois

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villeteuse, Université Paris 13

Exemplaire numéro 6

Luca Saiu

Documents (livres, slides, photocopiés, 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 = 7
```

```
while a < 8:
```

```
    print ("foo")
```

- a) affiche « foo » une infinité de fois
b) affiche « foo » 8 fois
c) affiche « foo » une fois
d) affiche « foo » 6 fois
e) échoue
f) n'affiche rien
g) affiche « foo » 7 fois

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

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

```
    print ("foo")
```

```
for j in range (2):
```

```
    print ("foo")
```

- a) affiche « foo » une infinité de fois
b) affiche « foo » 6 fois
c) affiche « foo » 4 fois
d) affiche « foo » 3 fois
e) affiche « foo » une fois
f) affiche « foo » 5 fois
g) affiche « foo » 2 fois
h) échoue
i) 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 » 10 fois
b) affiche « foo » 20 fois
c) échoue
d) affiche « foo » 3 fois
e) affiche « foo » 30 fois
f) affiche « foo » 33 fois
g) affiche « foo » 100 fois
h) affiche « foo » 9 fois
i) n'affiche rien
j) affiche « foo » une fois
k) affiche « foo » une infinité de fois
l) affiche « foo » 1000 fois
m) affiche « foo » 27 fois
n) affiche « foo » 10000 fois

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

```
"foo" * 3
```

- a) a comme résultat un booléen
b) échoue
c) a comme résultat une chaîne de caractères
d) a comme résultat la valeur
e) de la variable **foo** multipliée par 3
f) a comme résultat un entier
g) a comme résultat un float-tant
h) n'a aucun résultat

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

```
if 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 juste « bar », une fois
d) échoue
e) affiche « foo » une fois et « bar » une fois
f) n'affiche rien
g) affiche « foo » une infinité de fois
h) affiche juste « foo », une fois

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

```
if True:
```

```
    print ("foo")
```

```
print ("foo")
```

```
else:
```

```
    print ("foo")
```

- a) n'affiche rien
b) affiche « foo » une infinité de fois
c) affiche « foo » trois fois
d) affiche « foo » une fois
e) affiche « foo » deux fois
f) affiche « foo » quatre fois
g) échoue

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

```
a = 8
```

```
for b in range (4):
```

```
    a = a + b
```

```
print (a)
```

- a) affiche « 8 »
b) affiche « a »
c) affiche « 11 »
d) affiche « 10 »
e) affiche « 14 »
f) affiche « b »
g) n'affiche rien
h) échoue
i) affiche « 12 »
j) boucle infiniment
k) affiche « 13 »
l) affiche « 9 »

Question 8 (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) affiche « bar » une infinité de fois
d) affiche « foo » une fois et « bar » une infinité de fois
e) n'affiche rien
f) affiche juste « bar », une fois
g) affiche « foo » une infinité de fois
h) échoue

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

```
x = 8
```

```
if x > 7:
```

```
    print ("foo")
```

```
else x <= 7:
```

```
    print ("bar")
```

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

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 7

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

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

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

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » trois fois
- b) affiche « foo » quatre fois
- c) affiche « foo » deux fois
- d) n'affiche rien
- e) affiche « foo » une fois
- f) échoue
- g) affiche « foo » une infinité de fois

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) n'affiche rien
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 6 fois
- d) affiche « foo » 8 fois
- e) échoue
- f) affiche « foo » 7 fois
- g) affiche « foo » une fois

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » 5 fois
- c) affiche « foo » 3 fois
- d) affiche « foo » une infinité de fois
- e) échoue
- f) affiche « foo » 4 fois
- g) n'affiche rien
- h) affiche « foo » 2 fois
- i) affiche « foo » 6 fois

Question 6 (2 points) L'expression suivante ...

```
"foo" * 3
```

- a) a comme résultat une chaîne de caractères
- b) a comme résultat un float-tant
- c) échoue
- d) a comme résultat un boo-
- e) a comme résultat la valeur de la variable foo multipliée par 3
- f) n'a aucun résultat
- g) a comme résultat un entier

Question 7 (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) affiche juste « bar », une fois
- d) affiche « foo » une infinité de fois
- e) n'affiche rien
- f) affiche « foo » une fois et « bar » une infinité de fois
- g) affiche « foo » une fois et « bar » une fois
- h) échoue

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 8 »
- b) affiche « a »
- c) affiche « b »
- d) affiche « 13 »
- e) affiche « 9 »
- f) n'affiche rien
- g) boucle infiniment
- h) échoue
- i) affiche « 11 »
- j) affiche « 10 »
- k) affiche « 12 »
- l) affiche « 14 »

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 » 3 fois
- c) affiche « foo » 33 fois
- d) échoue
- e) n'affiche rien
- f) affiche « foo » 30 fois
- g) affiche « foo » 10000 fois
- h) affiche « foo » une infinité de fois
- i) affiche « foo » 1000 fois
- j) affiche « foo » 20 fois
- k) affiche « foo » une fois
- l) affiche « foo » 9 fois
- m) affiche « foo » 27 fois
- n) affiche « foo » 100 fois

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 8

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

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

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

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

- a) affiche « foo » une fois
- b) n'affiche rien
- c) affiche « foo » une fois et « bar » une fois
- d) affiche « foo » une infinité de fois
- e) affiche « x » une fois
- f) affiche « bar » une infinité de fois
- g) échoue
- h) affiche « bar » une fois
- i) affiche « x » une infinité de 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 » 20 fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » une fois
- d) affiche « foo » 30 fois
- e) affiche « foo » 3 fois
- f) affiche « foo » 33 fois
- g) affiche « foo » 10 fois
- h) affiche « foo » 1000 fois
- i) affiche « foo » 9 fois
- j) affiche « foo » 27 fois
- k) échoue
- l) n'affiche rien
- m) affiche « foo » 10000 fois
- n) affiche « foo » 100 fois

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

```
"foo" * 3
```

- a) a comme résultat un entier
- b) échoue
- c) a comme résultat un booléen
- d) a comme résultat un float-tant
- e) a comme résultat la valeur de la variable foo multipliée par 3
- f) n'a aucun résultat
- g) a comme résultat une chaîne de caractères

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » 6 fois
- b) affiche « foo » une fois
- c) n'affiche rien
- d) affiche « foo » 8 fois
- e) échoue
- f) affiche « foo » une infinité de fois
- g) affiche « foo » 7 fois

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 6 fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 2 fois
- d) affiche « foo » 4 fois
- e) échoue
- f) affiche « foo » une fois
- g) affiche « foo » 5 fois
- h) n'affiche rien
- i) affiche « foo » 3 fois

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) échoue
- b) n'affiche rien
- c) affiche « foo » trois fois
- d) affiche « foo » une infinité de fois
- e) affiche « foo » deux fois
- f) affiche « foo » une fois
- g) affiche « foo » quatre fois

Question 8 (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) affiche juste « bar », une fois
- d) échoue
- e) affiche « foo » une infinité de fois
- f) n'affiche rien
- g) affiche « foo » une fois et « bar » une infinité de fois
- h) affiche « foo » une fois et « bar » une fois

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « b »
- b) affiche « 9 »
- c) affiche « a »
- d) affiche « 10 »
- e) affiche « 14 »
- f) affiche « 8 »
- g) échoue
- h) affiche « 11 »
- i) affiche « 12 »
- j) affiche « 13 »
- k) n'affiche rien
- l) boucle infiniment

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villeteuse, Université Paris 13

Exemplaire numéro 9

Luca Saiu

Documents (livres, slides, photocopiés, 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 = 7
while a < 8:
    print ("foo")
```

- a) échoue
- b) affiche « foo » 7 fois
- c) affiche « foo » 8 fois
- d) affiche « foo » une infinité
- e) n'affiche rien
- f) affiche « foo » 6 fois
- g) affiche « foo » une fois

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » quatre fois
- b) affiche « foo » deux fois
- c) affiche « foo » une fois
- d) n'affiche rien
- e) affiche « foo » trois fois
- f) échoue
- g) affiche « foo » une infinité de fois

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « a »
- b) affiche « 9 »
- c) affiche « b »
- d) n'affiche rien
- e) affiche « 12 »
- f) affiche « 11 »
- g) boucle infiniment
- h) échoue
- i) affiche « 13 »
- j) affiche « 8 »
- k) affiche « 14 »
- l) affiche « 10 »

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) échoue
- b) affiche « foo » 4 fois
- c) n'affiche rien
- d) affiche « foo » une infinité de fois
- e) affiche « foo » une fois
- f) affiche « foo » 2 fois
- g) affiche « foo » 3 fois
- h) affiche « foo » 6 fois
- i) affiche « foo » 5 fois

Question 5 (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) affiche « foo » une infinité de fois
- d) affiche juste « bar », une fois
- e) n'affiche rien
- f) échoue
- g) affiche juste « foo », une fois
- h) affiche « bar » une infinité de fois

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

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

- a) affiche « bar » une infinité de fois
- b) échoue
- c) affiche juste « bar », une fois
- d) affiche juste « foo », une fois
- e) n'affiche rien
- 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 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) n'affiche rien
- c) affiche « foo » 30 fois
- d) affiche « foo » 10000 fois
- e) affiche « foo » 33 fois
- f) affiche « foo » 100 fois
- g) affiche « foo » 20 fois
- h) affiche « foo » une fois
- i) échoue
- j) affiche « foo » 27 fois
- k) affiche « foo » 1000 fois
- l) affiche « foo » une infinité de fois
- m) affiche « foo » 9 fois
- n) affiche « foo » 3 fois

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

```
"foo" * 3
```

- a) n'a aucun résultat
- b) a comme résultat un entier
- c) échoue
- d) a comme résultat un float-tant
- e) a comme résultat un boo-
- f) a comme résultat la valeur de la variable foo multipliée par 3
- g) a comme résultat une chaîne de caractères

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 10

Luca Saiu

Documents (livres, slides, photocopiés, 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 = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » 6 fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 7 fois
- d) affiche « foo » 8 fois
- e) échoue
- f) affiche « foo » une fois
- g) n'affiche rien

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » quatre fois
- c) affiche « foo » trois fois
- d) affiche « foo » une infinité
- e) n'affiche rien
- f) affiche « foo » deux fois
- g) échoue

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 6 fois
- b) affiche « foo » une fois
- c) affiche « foo » une infinité de fois
- d) n'affiche rien
- e) échoue
- f) affiche « foo » 4 fois
- g) affiche « foo » 5 fois
- h) affiche « foo » 2 fois
- i) affiche « foo » 3 fois

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

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

- a) échoue
- b) affiche « foo » une fois et « bar » une fois
- c) affiche « foo » une infinité de 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

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

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

- a) affiche « foo » une fois et « bar » une infinité de 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) affiche « bar » une infinité de fois
- g) échoue
- h) affiche juste « bar », une fois

Question 6 (2 points) L'expression suivante ...

```
"foo" * 3
```

- a) a comme résultat un entier
- b) a comme résultat un booléen
- c) échoue
- d) a comme résultat un float
- e) n'a aucun résultat
- f) a comme résultat une chaîne de caractères
- g) a comme résultat la valeur de la variable foo multipliée par 3

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

- a) affiche « foo » une infinité de fois
- b) n'affiche rien
- c) affiche « x » une fois
- d) affiche « foo » une fois
- e) affiche « bar » une infinité de fois
- f) affiche « bar » une fois
- g) échoue
- h) affiche « x » une infinité de fois
- i) 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 » une infinité de fois
- b) affiche « foo » 10 fois
- c) affiche « foo » 3 fois
- d) affiche « foo » 30 fois
- e) affiche « foo » une fois
- f) affiche « foo » 27 fois
- g) affiche « foo » 1000 fois
- h) affiche « foo » 10000 fois
- i) échoue
- j) affiche « foo » 20 fois
- k) affiche « foo » 9 fois
- l) affiche « foo » 100 fois
- m) affiche « foo » 33 fois
- n) n'affiche rien

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) boucle infiniment
- b) affiche « 14 »
- c) affiche « 8 »
- d) affiche « 12 »
- e) affiche « 9 »
- f) affiche « b »
- g) affiche « 13 »
- h) échoue
- i) affiche « 10 »
- j) n'affiche rien
- k) affiche « a »
- l) affiche « 11 »

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 11

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) n'affiche rien
- b) affiche « foo » quatre fois
- c) affiche « foo » une infinité de fois
- d) affiche « foo » une fois
- e) affiche « foo » trois fois
- f) affiche « foo » deux fois
- g) échoue

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) boucle infiniment
- b) affiche « 11 »
- c) affiche « 10 »
- d) affiche « 8 »
- e) échoue
- f) affiche « 9 »
- g) affiche « a »
- h) affiche « 14 »
- i) n'affiche rien
- j) affiche « b »
- k) affiche « 12 »
- l) affiche « 13 »

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

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

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

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) échoue
- b) affiche « foo » 100 fois
- c) affiche « foo » 1000 fois
- d) affiche « foo » 9 fois
- e) affiche « foo » 33 fois
- f) affiche « foo » 20 fois
- g) affiche « foo » une infinité de fois
- h) n'affiche rien
- i) affiche « foo » 10 fois
- j) affiche « foo » 3 fois
- k) affiche « foo » 27 fois
- l) affiche « foo » 10000 fois
- m) affiche « foo » 30 fois
- n) affiche « foo » une fois

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

```
"foo" * 3
```

- a) a comme résultat un float-tant
- b) a comme résultat un entier
- c) échoue
- d) n'a aucun résultat
- e) a comme résultat un booléen
- f) a comme résultat une chaîne de caractères
- g) a comme résultat la valeur de la variable foo multipliée par 3

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

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

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

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 2 fois
- b) échoue
- c) affiche « foo » une infinité de fois
- d) affiche « foo » une fois
- e) affiche « foo » 4 fois
- f) n'affiche rien
- g) affiche « foo » 3 fois
- h) affiche « foo » 6 fois
- i) affiche « foo » 5 fois

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » 8 fois
- b) affiche « foo » une fois
- c) n'affiche rien
- d) échoue
- e) affiche « foo » 7 fois
- f) affiche « foo » 6 fois
- g) affiche « foo » une infinité de fois

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villeteuse, Université Paris 13

Exemplaire numéro 12

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

```
"foo" * 3
```

- a) a comme résultat une chaîne de caractères
- b) échoue
- c) a comme résultat un booléen
- d) a comme résultat un float
- e) a comme résultat la valeur de la variable `foo` multipliée par 3
- f) a comme résultat un entier
- g) n'a aucun résultat

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

```
a = 7
while a < 8:
    print("foo")
```

- a) affiche « foo » 6 fois
- b) affiche « foo » 8 fois
- c) affiche « foo » 7 fois
- d) échoue
- e) n'affiche rien
- f) affiche « foo » une infinité de fois
- g) affiche « foo » une fois

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

```
x = 8
if x > 7:
    print("foo")
else x <= 7:
    print("bar")
```

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

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

```
for i in range(3):
    print("foo")
for j in range(2):
    print("foo")
```

- a) affiche « foo » 3 fois
- b) affiche « foo » 4 fois
- c) n'affiche rien
- d) affiche « foo » 2 fois
- e) affiche « foo » 5 fois
- f) affiche « foo » une infinité de fois
- g) affiche « foo » une fois
- h) échoue
- i) affiche « foo » 6 fois

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

```
if True:
    print("foo")
print("foo")
else:
    print("foo")
```

- a) n'affiche rien
- b) affiche « foo » trois fois
- c) affiche « foo » quatre fois
- d) affiche « foo » une infinité
- e) affiche « foo » une fois
- f) affiche « foo » deux fois
- g) affiche « foo » une fois

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

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

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

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

```
a = 8
for b in range(4):
    a = a + b
print(a)
```

- a) affiche « 8 »
- b) boucle infiniment
- c) affiche « b »
- d) affiche « 11 »
- e) échoue
- f) affiche « 12 »
- g) affiche « 13 »
- h) affiche « 14 »
- i) affiche « a »
- j) n'affiche rien
- k) affiche « 9 »
- l) affiche « 10 »

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 » 10 fois
- b) affiche « foo » 9 fois
- c) affiche « foo » 10000 fois
- d) affiche « foo » 27 fois
- e) affiche « foo » 33 fois
- f) échoue
- g) affiche « foo » 1000 fois
- h) affiche « foo » 20 fois
- i) affiche « foo » 30 fois
- j) n'affiche rien
- k) affiche « foo » 100 fois
- l) affiche « foo » 3 fois
- m) affiche « foo » une infinité de fois
- n) affiche « foo » une fois

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

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

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

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 13

Luca Saiu

Documents (livres, slides, photocopiés, 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 « bar », une fois
- b) affiche juste « foo », une 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 « foo » une infinité de fois
- h) affiche « bar » une infinité de fois

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) échoue
- b) affiche « 8 »
- c) affiche « a »
- d) affiche « 12 »
- e) affiche « 9 »
- f) affiche « 11 »
- g) n'affiche rien
- h) affiche « 10 »
- i) affiche « b »
- j) boucle infiniment
- k) affiche « 14 »
- l) affiche « 13 »

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) n'affiche rien
- b) affiche « foo » trois fois
- c) affiche « foo » deux fois
- d) affiche « foo » une fois
- e) échoue
- f) affiche « foo » une infinité de fois
- g) affiche « foo » quatre 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
- b) affiche « foo » 10000 fois
- c) affiche « foo » 20 fois
- d) affiche « foo » 30 fois
- e) n'affiche rien
- f) affiche « foo » 27 fois
- g) affiche « foo » une infinité de fois
- h) affiche « foo » 10 fois
- i) affiche « foo » 9 fois
- j) échoue
- k) affiche « foo » une fois
- l) affiche « foo » 100 fois
- m) affiche « foo » 33 fois
- n) affiche « foo » 1000 fois

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 2 fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 5 fois
- d) affiche « foo » 4 fois
- e) n'affiche rien
- f) échoue
- g) affiche « foo » une fois
- h) affiche « foo » 6 fois
- i) affiche « foo » 3 fois

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » 6 fois
- b) n'affiche rien
- c) affiche « foo » 7 fois
- d) affiche « foo » une infinité de fois
- e) affiche « foo » 8 fois
- f) échoue
- g) affiche « foo » une fois

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

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

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

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

"foo" * 3

- a) a comme résultat un booléen
- b) échoue
- c) a comme résultat un float tant
- d) n'a aucun résultat
- e) a comme résultat un entier
- f) a comme résultat la valeur de la variable foo multipliée par 3
- g) a comme résultat une chaîne de caractères

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 14

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

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

- a) affiche juste « foo », une fois
- b) n'affiche rien
- 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) affiche « bar » une infinité de fois
- h) échoue

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

```
"foo" * 3
```

- a) a comme résultat une chaîne de caractères
- b) a comme résultat un booléen
- c) a comme résultat un float-tant
- d) n'a aucun résultat
- e) a comme résultat un entier
- f) échoue
- g) a comme résultat la valeur de la variable foo multipliée par 3

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » trois fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » une fois
- d) n'affiche rien
- e) affiche « foo » deux fois
- f) échoue
- g) affiche « foo » quatre 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 » 30 fois
- b) affiche « foo » 100 fois
- c) affiche « foo » une fois
- d) affiche « foo » 27 fois
- e) affiche « foo » 3 fois
- f) n'affiche rien
- g) affiche « foo » une infinité de fois
- h) affiche « foo » 20 fois
- i) affiche « foo » 10000 fois
- j) affiche « foo » 33 fois
- k) affiche « foo » 9 fois
- l) échoue
- m) affiche « foo » 10 fois
- n) affiche « foo » 1000 fois

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » une fois
- b) n'affiche rien
- c) échoue
- d) affiche « foo » une infinité de fois
- e) affiche « foo » 5 fois
- f) affiche « foo » 6 fois
- g) affiche « foo » 2 fois
- h) affiche « foo » 4 fois
- i) affiche « foo » 3 fois

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 12 »
- b) échoue
- c) affiche « 11 »
- d) affiche « b »
- e) affiche « 10 »
- f) n'affiche rien
- g) affiche « 14 »
- h) affiche « 8 »
- i) affiche « 9 »
- j) affiche « a »
- k) boucle infiniment
- l) affiche « 13 »

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

Question 8 (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) affiche « bar » une infinité de fois
- d) échoue
- e) affiche « foo » une fois et « bar » une infinité de fois
- f) n'affiche rien
- g) affiche « foo » une fois et « bar » une fois
- h) affiche juste « bar », une fois

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » une infinité de fois
- b) affiche « foo » 8 fois
- c) n'affiche rien
- d) affiche « foo » 7 fois
- e) affiche « foo » une fois
- f) affiche « foo » 6 fois
- g) échoue

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 15

Luca Saiu

Documents (livres, slides, photocopiés, 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 = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » une fois
- b) échoue
- c) affiche « foo » 8 fois
- d) affiche « foo » 7 fois
- e) affiche « foo » une infinité de fois
- f) affiche « foo » 6 fois
- g) n'affiche rien

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

```
"foo" * 3
```

- a) a comme résultat un booléen
- b) échoue
- c) a comme résultat un entier
- d) a comme résultat un float-tant
- e) a comme résultat la valeur de la variable `foo` multipliée par 3
- f) a comme résultat une chaîne de caractères
- g) n'a aucun résultat

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

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

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

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

Question 5 (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
- b) affiche « foo » 20 fois
- c) affiche « foo » une infinité de fois
- d) affiche « foo » 100 fois
- e) affiche « foo » 3 fois
- f) affiche « foo » 33 fois
- g) affiche « foo » une fois
- h) affiche « foo » 10 fois
- i) affiche « foo » 1000 fois
- j) affiche « foo » 27 fois
- k) n'affiche rien
- l) affiche « foo » 10000 fois
- m) échoue
- n) affiche « foo » 30 fois

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

```
a = 8
for b in range (4):
    a = a + b
    print (a)
```

- a) boucle infiniment
- b) affiche « b »
- c) affiche « 9 »
- d) affiche « a »
- e) affiche « 13 »
- f) affiche « 14 »
- g) affiche « 11 »
- h) affiche « 12 »
- i) affiche « 8 »
- j) échoue
- k) n'affiche rien
- l) affiche « 10 »

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 5 fois
- b) affiche « foo » 2 fois
- c) affiche « foo » une fois
- d) affiche « foo » 3 fois
- e) affiche « foo » 4 fois
- f) affiche « foo » 6 fois
- g) échoue
- h) affiche « foo » une infinité de fois
- i) n'affiche rien

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » trois fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » quatre fois
- d) n'affiche rien
- e) échoue
- f) affiche « foo » une fois
- g) affiche « foo » deux fois

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 16

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

- a) affiche « bar » une fois
- b) affiche « x » une fois
- c) affiche « foo » une fois et « bar » une fois
- d) affiche « bar » une infinité de fois
- e) échoue
- f) affiche « x » une infinité de fois
- g) affiche « foo » une fois
- h) n'affiche rien
- i) 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) affiche « foo » 10000 fois
- c) affiche « foo » une infinité de fois
- d) n'affiche rien
- e) affiche « foo » 100 fois
- f) affiche « foo » 3 fois
- g) affiche « foo » 9 fois
- h) échoue
- i) affiche « foo » une fois
- j) affiche « foo » 33 fois
- k) affiche « foo » 27 fois
- l) affiche « foo » 20 fois
- m) affiche « foo » 30 fois
- n) affiche « foo » 10 fois

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 9 »
- b) affiche « 12 »
- c) affiche « 8 »
- d) affiche « a »
- e) affiche « b »
- f) n'affiche rien
- g) affiche « 11 »
- h) affiche « 10 »
- i) affiche « 14 »
- j) échoue
- k) affiche « 13 »
- l) boucle infiniment

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

```
"foo" * 3
```

- a) a comme résultat la valeur de la variable foo multipliée par 3
- b) a comme résultat un entier
- c) a comme résultat une chaîne de caractères
- d) a comme résultat un float-tant
- e) a comme résultat un booléen
- f) échoue
- g) n'a aucun résultat

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 6 fois
- d) affiche « foo » 2 fois
- e) affiche « foo » 5 fois
- f) échoue
- g) n'affiche rien
- h) affiche « foo » 3 fois
- i) affiche « foo » 4 fois

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

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

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

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » trois fois
- c) affiche « foo » quatre fois
- d) affiche « foo » une infinité de fois
- e) affiche « foo » deux fois
- f) n'affiche rien
- g) échoue

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) n'affiche rien
- b) affiche « foo » 6 fois
- c) affiche « foo » une fois
- d) affiche « foo » 7 fois
- e) échoue
- f) affiche « foo » 8 fois
- g) affiche « foo » une infinité de fois

Question 9 (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) échoue
- e) n'affiche rien
- f) affiche « foo » une infinité de fois
- g) affiche juste « foo », une fois
- h) affiche juste « bar », une fois

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 17

Luca Saiu

Documents (livres, slides, photocopiés, 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 = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » 8 fois
- c) affiche « foo » 6 fois
- d) n'affiche rien
- e) affiche « foo » une infinité de fois
- f) affiche « foo » 7 fois
- g) échoue

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » deux fois
- b) affiche « foo » trois fois
- c) n'affiche rien
- d) affiche « foo » une infinité
- e) affiche « foo » quatre fois
- f) échoue
- g) affiche « foo » une fois

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
if False:
    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) affiche « foo » une fois et « bar » une infinité de fois
- g) échoue
- h) affiche juste « foo », une fois

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

```
"foo" * 3
```

- a) a comme résultat un entier
- b) a comme résultat une chaîne de caractères
- c) échoue
- d) a comme résultat un booléen
- e) a comme résultat la valeur de la variable foo multipliée par 3
- f) a comme résultat un float
- g) n'a aucun résultat

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

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 14 »
- b) affiche « a »
- c) affiche « 9 »
- d) affiche « 10 »
- e) affiche « 12 »
- f) boucle infiniment
- g) affiche « 13 »
- h) n'affiche rien
- i) affiche « b »
- j) affiche « 11 »
- k) affiche « 8 »
- 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 » 20 fois
- b) affiche « foo » 9 fois
- c) affiche « foo » 100 fois
- d) affiche « foo » 3 fois
- e) affiche « foo » 33 fois
- f) affiche « foo » 27 fois
- g) affiche « foo » une fois
- h) affiche « foo » 1000 fois
- i) affiche « foo » 30 fois
- j) affiche « foo » 10000 fois
- k) échoue
- l) n'affiche rien
- m) affiche « foo » une infinité de fois
- n) affiche « foo » 10 fois

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 5 fois
- b) affiche « foo » une fois
- c) n'affiche rien
- d) affiche « foo » 3 fois
- e) échoue
- f) affiche « foo » 4 fois
- g) affiche « foo » 2 fois
- h) affiche « foo » 6 fois
- i) affiche « foo » une infinité de fois

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 18

Luca Saiu

Documents (livres, slides, photocopiés, 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 « bar » une infinité de fois
- c) affiche « foo » une infinité de fois
- d) affiche « foo » une fois et
- e) affiche juste « bar », une fois
- f) n'affiche rien
- g) affiche « foo » une fois et « bar » une infinité de fois
- h) échoue

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

```
"foo" * 3
```

- a) échoue
- b) a comme résultat la valeur de la variable `foo` multipliée par 3
- c) a comme résultat un entier
- d) a comme résultat un float-
- e) tant
- f) n'a aucun résultat
- g) a comme résultat un booleen
- h) a comme résultat une chaîne de caractères

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » une infinité de fois
- b) échoue
- c) affiche « foo » quatre fois
- d) affiche « foo » trois fois
- e) affiche « foo » une fois
- f) affiche « foo » deux fois
- g) 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 » 27 fois
- b) affiche « foo » 9 fois
- c) affiche « foo » une fois
- d) affiche « foo » 30 fois
- e) affiche « foo » 3 fois
- f) échoue
- g) affiche « foo » 33 fois
- h) affiche « foo » 1000 fois
- i) affiche « foo » 20 fois
- j) affiche « foo » 10000 fois
- k) n'affiche rien
- l) affiche « foo » 100 fois
- m) affiche « foo » 10 fois
- n) affiche « foo » une infinité de fois

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

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

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

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « b »
- b) affiche « a »
- c) affiche « 14 »
- d) affiche « 8 »
- e) affiche « 10 »
- f) affiche « 12 »
- g) n'affiche rien
- h) affiche « 9 »
- i) échoue
- j) affiche « 11 »
- k) boucle infiniment
- l) affiche « 13 »

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) n'affiche rien
- b) affiche « foo » 7 fois
- c) affiche « foo » 8 fois
- d) affiche « foo » 6 fois
- e) échoue
- f) affiche « foo » une infinité de fois
- g) affiche « foo » une fois

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 4 fois
- b) affiche « foo » 6 fois
- c) affiche « foo » 5 fois
- d) affiche « foo » 3 fois
- e) affiche « foo » une fois
- f) affiche « foo » 2 fois
- g) affiche « foo » une infinité de fois
- h) échoue
- i) n'affiche rien

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 19

Luca Saiu

Documents (livres, slides, photocopiés, 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 = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » 6 fois
- b) affiche « foo » une fois
- c) affiche « foo » une infinité de fois
- d) affiche « foo » 8 fois
- e) affiche « foo » 7 fois
- f) n'affiche rien
- g) échoue

Question 2 (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 fois et « bar » une infinité de fois
- d) affiche juste « foo », une fois
- e) affiche « foo » une infinité de fois
- f) échoue
- g) n'affiche rien
- h) affiche « foo » une fois et « bar » une fois

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) n'affiche rien
- b) affiche « foo » 2 fois
- c) affiche « foo » 3 fois
- d) affiche « foo » 6 fois
- e) affiche « foo » 5 fois
- f) affiche « foo » une infinité de fois
- g) affiche « foo » 4 fois
- h) affiche « foo » une fois
- i) échoue

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

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

- a) n'affiche rien
- b) affiche « bar » une infinité de fois
- 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 juste « bar », une fois
- h) affiche « foo » une fois et « bar » une infinité de fois

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 9 »
- b) échoue
- c) affiche « 11 »
- d) affiche « b »
- e) affiche « 12 »
- f) n'affiche rien
- g) affiche « 10 »
- h) boucle infiniment
- i) affiche « 13 »
- j) affiche « 14 »
- k) affiche « 8 »
- l) affiche « a »

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » quatre fois
- c) affiche « foo » trois fois
- d) affiche « foo » deux fois
- e) n'affiche rien
- f) échoue
- g) affiche « foo » une infinité de fois

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

```
"foo" * 3
```

- a) échoue
- b) a comme résultat la valeur de la variable foo multipliée par 3
- c) n'a aucun résultat
- d) a comme résultat un entier
- e) a comme résultat un boolean
- f) a comme résultat une chaîne de caractères
- g) a comme résultat un float-tant

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) échoue
- b) affiche « foo » 27 fois
- c) affiche « foo » 100 fois
- d) affiche « foo » 3 fois
- e) affiche « foo » 33 fois
- f) n'affiche rien
- g) affiche « foo » 10000 fois
- h) affiche « foo » 10 fois
- i) affiche « foo » 1000 fois
- j) affiche « foo » 20 fois
- k) affiche « foo » une infinité de fois
- l) affiche « foo » 30 fois
- m) affiche « foo » 9 fois
- n) affiche « foo » une fois

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 20

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

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

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

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 4 fois
- b) affiche « foo » 6 fois
- c) échoue
- d) affiche « foo » 3 fois
- e) n'affiche rien
- f) affiche « foo » une fois
- g) affiche « foo » 5 fois
- h) affiche « foo » 2 fois
- i) affiche « foo » une infinité de fois

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » trois fois
- b) affiche « foo » une infinité de fois
- c) n'affiche rien
- d) échoue
- e) affiche « foo » deux fois
- f) affiche « foo » quatre fois
- g) affiche « foo » une fois

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

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

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

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

```
"foo" * 3
```

- a) a comme résultat un entier
- b) n'a aucun résultat
- c) a comme résultat un booléen
- d) a comme résultat la valeur de la variable foo multipliée par 3
- e) échoue
- f) a comme résultat un float
- g) a comme résultat une chaîne de caractères

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 11 »
- b) affiche « 8 »
- c) affiche « 12 »
- d) boucle infiniment
- e) affiche « b »
- f) affiche « a »
- g) affiche « 9 »
- h) affiche « 13 »
- i) affiche « 14 »
- j) échoue
- k) affiche « 10 »
- l) n'affiche rien

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » une fois
- b) échoue
- c) affiche « foo » 7 fois
- d) n'affiche rien
- e) affiche « foo » une infinité de fois
- f) affiche « foo » 8 fois
- g) affiche « foo » 6 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
- b) échoue
- c) affiche « foo » 27 fois
- d) affiche « foo » 20 fois
- e) affiche « foo » 10 fois
- f) affiche « foo » 30 fois
- g) affiche « foo » une infinité de fois
- h) affiche « foo » 3 fois
- i) affiche « foo » 100 fois
- j) affiche « foo » 9 fois
- k) affiche « foo » 1000 fois
- l) affiche « foo » 33 fois
- m) affiche « foo » une fois
- n) n'affiche rien

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 21

Luca Saiu

Documents (livres, slides, photocopiés, 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 = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » 7 fois
- b) affiche « foo » 8 fois
- c) échoue
- d) affiche « foo » une infinité
- e) de fois
- f) affiche « foo » 6 fois
- g) affiche « foo » une fois
- h) n'affiche rien

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

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

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

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

```
"foo" * 3
```

- a) a comme résultat un booleen
- b) a comme résultat la valeur de la variable foo multipliée par 3
- c) n'a aucun résultat
- d) a comme résultat une chaîne de caractères
- e) a comme résultat un entier
- f) a comme résultat un float
- g) échoue

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) n'affiche rien
- c) affiche « foo » 20 fois
- d) affiche « foo » 27 fois
- e) affiche « foo » 30 fois
- f) affiche « foo » 9 fois
- g) affiche « foo » une infinité de fois
- h) affiche « foo » une fois
- i) affiche « foo » 33 fois
- j) échoue
- k) affiche « foo » 10 fois
- l) affiche « foo » 100 fois
- m) affiche « foo » 1000 fois
- n) affiche « foo » 3 fois

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « a »
- b) affiche « 13 »
- c) affiche « b »
- d) boucle infiniment
- e) affiche « 10 »
- f) affiche « 9 »

- g) n'affiche rien
- h) affiche « 14 »
- i) échoue
- j) affiche « 8 »
- k) affiche « 12 »
- l) affiche « 11 »

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » une infinité de fois
- b) affiche « foo » 2 fois
- c) affiche « foo » 3 fois
- d) affiche « foo » une fois
- e) affiche « foo » 5 fois
- f) affiche « foo » 4 fois
- g) n'affiche rien
- h) affiche « foo » 6 fois
- i) 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 infinité de fois
- c) affiche « bar » une infinité de fois
- d) affiche juste « bar », une fois
- e) affiche juste « foo », une fois
- f) affiche « foo » une infinité de fois
- g) échoue
- h) affiche « foo » une fois et « bar » une fois

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « a »
- b) affiche « 13 »
- c) affiche « b »
- d) boucle infiniment
- e) affiche « 10 »
- f) affiche « 9 »
- g) n'affiche rien
- h) affiche « 14 »
- i) échoue
- j) affiche « 8 »
- k) affiche « 12 »
- l) affiche « 11 »

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 22

Luca Saiu

Documents (livres, slides, photocopiés, 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 (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) échoue
- b) affiche « foo » 4 fois
- c) affiche « foo » une fois
- d) affiche « foo » 6 fois
- e) affiche « foo » 2 fois
- f) affiche « foo » 3 fois
- g) n'affiche rien
- h) affiche « foo » 5 fois
- i) affiche « foo » une infinité de fois

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) échoue
- b) affiche « 10 »
- c) affiche « 9 »
- d) affiche « 8 »
- e) n'affiche rien
- f) affiche « 12 »
- g) affiche « 13 »
- h) boucle infiniment
- i) affiche « 14 »
- j) affiche « b »
- k) affiche « a »
- l) affiche « 11 »

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » quatre fois
- b) échoue
- c) affiche « foo » une fois
- d) n'affiche rien
- e) affiche « foo » trois fois
- f) affiche « foo » une infinité de fois
- g) affiche « foo » deux fois

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » une fois
- b) n'affiche rien
- c) affiche « foo » une infinité de fois
- d) affiche « foo » 7 fois
- e) affiche « foo » 6 fois
- f) affiche « foo » 8 fois
- g) échoue

Question 5 (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 » 33 fois
- c) affiche « foo » 10 fois
- d) affiche « foo » 9 fois
- e) affiche « foo » 1000 fois
- f) affiche « foo » 27 fois
- g) échoue
- h) affiche « foo » une fois
- i) affiche « foo » 10000 fois
- j) affiche « foo » 100 fois
- k) n'affiche rien
- l) affiche « foo » 30 fois
- m) affiche « foo » une infinité de fois
- n) affiche « foo » 3 fois

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
"foo" * 3
```

- a) a comme résultat un booléen
- b) a comme résultat la valeur de la variable foo multipliée par 3
- c) échoue
- d) n'a aucun résultat
- e) a comme résultat un entier
- f) a comme résultat un float tant
- g) a comme résultat une chaîne de caractères

Question 8 (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 « bar » une fois
- f) affiche « foo » une infinité de fois
- g) affiche « foo » une fois et « bar » une infinité de fois
- h) affiche « bar » une infinité de fois

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

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

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

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 23

Luca Saiu

Documents (livres, slides, photocopiés, 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 = 7
while a < 8:
    print ("foo")
```

- a) échoue
- b) affiche « foo » 6 fois
- c) affiche « foo » 7 fois
- d) affiche « foo » 8 fois
- e) affiche « foo » une fois
- f) n'affiche rien
- g) affiche « foo » une infinité de fois

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

```
"foo" * 3
```

- a) n'a aucun résultat
- b) a comme résultat la valeur de la variable `foo` multipliée par 3
- c) a comme résultat une chaîne de caractères
- d) a comme résultat un float-tant
- e) échoue
- f) a comme résultat un booléen
- g) a comme résultat un entier

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » une fois
- b) échoue
- c) affiche « foo » une infinité de fois
- d) affiche « foo » trois fois
- e) n'affiche rien
- f) affiche « foo » deux fois
- g) affiche « foo » quatre 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 » 33 fois
- b) affiche « foo » 27 fois
- c) affiche « foo » 3 fois
- d) affiche « foo » 9 fois
- e) affiche « foo » 100 fois
- f) affiche « foo » une fois
- g) affiche « foo » 1000 fois
- h) affiche « foo » 20 fois
- i) affiche « foo » une infinité de fois
- j) échoue
- k) affiche « foo » 10000 fois
- l) affiche « foo » 10 fois
- m) affiche « foo » 30 fois
- n) n'affiche rien

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

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

- a) affiche « foo » une fois et « bar » une fois
- b) affiche « foo » une fois et « bar » une infinité de fois
- c) affiche juste « foo », une fois
- d) échoue
- 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 (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) n'affiche rien
- b) affiche « foo » 3 fois
- c) affiche « foo » 6 fois
- d) affiche « foo » 5 fois
- e) affiche « foo » 4 fois
- f) affiche « foo » une infinité de fois
- g) échoue
- h) affiche « foo » une fois
- i) affiche « foo » 2 fois

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

```
while False:
    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) affiche « foo » une fois et « bar » une fois
- e) affiche juste « foo », une fois
- f) affiche « foo » une infinité de fois
- g) échoue
- h) affiche juste « bar », une fois

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 10 »
- b) affiche « b »
- c) boucle infiniment
- d) affiche « a »
- e) affiche « 11 »
- f) affiche « 14 »
- g) affiche « 8 »
- h) affiche « 12 »
- i) affiche « 13 »
- j) affiche « 9 »
- k) échoue
- l) n'affiche rien

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 24

Luca Saiu

Documents (livres, slides, photocopies, code source, 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 = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » 7 fois
- c) n'affiche rien
- d) échoue
- e) affiche « foo » 6 fois
- f) affiche « foo » 8 fois
- g) affiche « foo » une infinité de fois

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) n'affiche rien
- b) affiche « 11 »
- c) échoue
- d) affiche « a »
- e) affiche « 8 »
- f) boucle infiniment
- g) affiche « 10 »
- h) affiche « 12 »
- i) affiche « 14 »
- j) affiche « 13 »
- k) affiche « b »
- l) affiche « 9 »

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 » 30 fois
- b) affiche « foo » 27 fois
- c) affiche « foo » 10 fois
- d) affiche « foo » une infinité de fois
- e) affiche « foo » 100 fois
- f) affiche « foo » 33 fois
- g) affiche « foo » 9 fois
- h) n'affiche rien
- i) affiche « foo » 10000 fois
- j) affiche « foo » 20 fois
- k) affiche « foo » 3 fois
- l) échoue
- m) affiche « foo » une fois
- n) affiche « foo » 1000 fois

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

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

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

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

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 5 fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 3 fois
- d) affiche « foo » 4 fois
- e) échoue
- f) affiche « foo » 6 fois
- g) n'affiche rien
- h) affiche « foo » une fois
- i) affiche « foo » 2 fois

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

```
"foo" * 3
```

- a) a comme résultat un entier de la variable foo multiplié par 3
- b) a comme résultat une chaîne de caractères
- c) a comme résultat un float-tant
- d) a comme résultat la valeur
- e) échoue
- f) n'a aucun résultat
- g) a comme résultat un booléen

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » trois fois de fois
- b) affiche « foo » quatre fois
- c) n'affiche rien
- d) affiche « foo » une infinité de fois
- e) affiche « foo » deux fois
- f) affiche « foo » une fois
- g) échoue

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

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

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

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 25

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

```
if 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 infinité de fois | f) échoue |
| c) affiche juste « foo », une fois | g) affiche juste « bar », une fois |
| d) n'affiche rien | h) affiche « foo » une fois et « bar » une infinité de fois |

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

```
while False:
```

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

```
a = 8
```

```
for b in range (4):
```

```
    a = a + b
```

```
print (a)
```

- | | |
|-------------------|----------------------|
| a) affiche « 10 » | g) affiche « 8 » |
| b) affiche « 13 » | h) affiche « a » |
| c) n'affiche rien | i) échoue |
| d) affiche « b » | j) affiche « 9 » |
| e) affiche « 11 » | k) affiche « 14 » |
| f) affiche « 12 » | l) boucle infiniment |

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

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

```
    print ("foo")
```

```
for j in range (2):
```

```
    print ("foo")
```

- | | |
|---------------------------|---|
| a) affiche « foo » 2 fois | f) affiche « foo » une infinité de fois |
| b) échoue | g) n'affiche rien |
| c) affiche « foo » 5 fois | h) affiche « foo » une fois |
| d) affiche « foo » 3 fois | i) affiche « foo » 6 fois |
| e) affiche « foo » 4 fois | |

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

```
"foo" * 3
```

- | | |
|--|---|
| a) a comme résultat une chaîne de caractères | e) a comme résultat un entier |
| b) n'a aucun résultat | f) a comme résultat la valeur de la variable foo multipliée par 3 |
| c) échoue | g) a comme résultat un float-léen |
| d) a comme résultat un booléen | |

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 » 10000 fois | de fois | i) affiche « foo » 10 fois |
| b) affiche « foo » 3 fois | | j) affiche « foo » 100 fois |
| c) affiche « foo » 33 fois | | k) échoue |
| d) affiche « foo » 27 fois | | l) n'affiche rien |
| e) affiche « foo » 30 fois | | m) affiche « foo » 1000 fois |
| f) affiche « foo » 9 fois | | n) affiche « foo » une fois |
| g) affiche « foo » 20 fois | | |
| h) affiche « foo » une infinité | | |

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

```
x = 8
```

```
if x > 7:
```

```
    print ("foo")
```

```
else x <= 7:
```

```
    print ("bar")
```

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

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

```
a = 7
```

```
while a < 8:
```

```
    print ("foo")
```

- | | |
|-----------------------------|---|
| a) affiche « foo » une fois | e) affiche « foo » une infinité de fois |
| b) affiche « foo » 7 fois | f) n'affiche rien |
| c) affiche « foo » 8 fois | g) affiche « foo » 6 fois |
| d) échoue | |

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

```
if True:
```

```
    print ("foo")
```

```
print ("foo")
```

```
else:
```

```
    print ("foo")
```

- | | |
|-------------------------------|---|
| a) n'affiche rien | e) affiche « foo » quatre fois |
| b) affiche « foo » une fois | f) affiche « foo » deux fois |
| c) affiche « foo » trois fois | g) affiche « foo » une infinité de fois |
| d) échoue | |

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 26

Luca Saiu

Documents (livres, slides, photocopiés, 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 (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » une infinité de fois
- b) n'affiche rien
- c) affiche « foo » 2 fois
- d) affiche « foo » 4 fois
- e) affiche « foo » une fois
- f) affiche « foo » 5 fois
- g) affiche « foo » 3 fois
- h) affiche « foo » 6 fois
- i) échoue

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » 6 fois
- b) affiche « foo » une infinité de fois
- c) n'affiche rien
- d) affiche « foo » 7 fois
- e) échoue
- f) affiche « foo » 8 fois
- g) affiche « foo » une fois

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

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

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

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) n'affiche rien
- b) affiche « b »
- c) affiche « 11 »
- d) échoue
- e) boucle infiniment
- f) affiche « 12 »
- g) affiche « 13 »
- h) affiche « 10 »
- i) affiche « 14 »
- j) affiche « 9 »
- k) affiche « 8 »
- l) affiche « a »

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

- a) affiche « x » une fois
- b) affiche « foo » une fois et « bar » une fois
- c) affiche « x » une infinité de fois
- d) échoue
- e) affiche « foo » une infinité de fois
- f) affiche « bar » une infinité de fois
- g) affiche « bar » une fois
- h) n'affiche rien
- i) affiche « foo » 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 » 20 fois
- b) affiche « foo » 27 fois
- c) affiche « foo » une infinité de fois
- d) affiche « foo » 10000 fois
- e) affiche « foo » 3 fois
- f) affiche « foo » 1000 fois
- g) affiche « foo » 30 fois
- h) affiche « foo » 100 fois
- i) affiche « foo » une fois
- j) affiche « foo » 10 fois
- k) affiche « foo » 33 fois
- l) affiche « foo » 9 fois
- m) échoue
- n) n'affiche rien

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

```
"foo" * 3
```

- a) échoue
- b) a comme résultat un booléen
- c) a comme résultat une chaîne de caractères
- d) n'a aucun résultat
- e) a comme résultat un float
- f) a comme résultat un entier
- g) a comme résultat la valeur de la variable foo multipliée par 3

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

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

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

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) échoue
- b) affiche « foo » une infinité de fois
- c) affiche « foo » une fois
- d) affiche « foo » trois fois
- e) affiche « foo » quatre fois
- f) affiche « foo » deux fois
- g) n'affiche rien

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 27

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

```
"foo" * 3
```

- a) n'a aucun résultat
- b) a comme résultat la valeur de la variable `foo` multipliée par 3
- c) échoue
- d) a comme résultat une chaîne de caractères
- e) a comme résultat un booléen
- f) a comme résultat un entier
- g) a comme résultat un float
- tant

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

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

- a) affiche juste « foo », une fois
- b) affiche « foo » une fois et « bar » 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 « bar » une infinité de fois
- h) affiche juste « bar », une fois

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

```
if True:
    print("foo")
print("foo")
else:
    print("foo")
```

- a) n'affiche rien
- b) affiche « foo » deux fois
- c) échoue
- d) affiche « foo » trois fois
- e) affiche « foo » quatre fois
- f) affiche « foo » une fois
- g) affiche « foo » une infinité de fois

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

```
for i in range(3):
    print("foo")
for j in range(2):
    print("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » 2 fois
- c) affiche « foo » 4 fois
- d) affiche « foo » 6 fois
- e) n'affiche rien
- f) affiche « foo » 5 fois
- g) affiche « foo » une infinité de fois
- h) échoue
- i) affiche « foo » 3 fois

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

```
a = 7
while a < 8:
    print("foo")
```

- a) affiche « foo » 8 fois
- b) n'affiche rien
- c) affiche « foo » 6 fois
- d) affiche « foo » 7 fois
- e) échoue
- f) affiche « foo » une fois
- g) affiche « foo » une infinité de fois

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

```
x = 8
if x > 7:
    print("foo")
else x <= 7:
    print("bar")
```

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

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

```
a = 8
for b in range(4):
    a = a + b
print(a)
```

- a) boucle infiniment
- b) affiche « 13 »
- c) affiche « a »
- d) affiche « 14 »
- e) affiche « 12 »
- f) affiche « 11 »
- g) affiche « 9 »
- h) affiche « b »
- i) affiche « 10 »
- j) échoue
- k) n'affiche rien
- l) affiche « 8 »

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 » 100 fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » une fois
- d) affiche « foo » 10000 fois
- e) affiche « foo » 30 fois
- f) affiche « foo » 33 fois
- g) échoue
- h) affiche « foo » 10 fois
- i) affiche « foo » 3 fois
- j) affiche « foo » 9 fois
- k) n'affiche rien
- l) affiche « foo » 1000 fois
- m) affiche « foo » 27 fois
- n) affiche « foo » 20 fois

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

```
while False:
    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) affiche juste « foo », une fois
- e) affiche « foo » une fois et « bar » une fois
- f) affiche juste « bar », une fois
- g) échoue
- h) affiche « foo » une infinité de fois

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 28

Luca Saiu

Documents (livres, slides, photocopiés, 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 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 fois
- g) échoue
- h) 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 » une infinité de fois
- b) affiche « foo » 20 fois
- c) affiche « foo » 9 fois
- d) affiche « foo » 10 fois
- e) affiche « foo » 30 fois
- f) n'affiche rien
- g) affiche « foo » 3 fois
- h) échoue
- i) affiche « foo » 27 fois
- j) affiche « foo » 10000 fois
- k) affiche « foo » 33 fois
- l) affiche « foo » 100 fois
- m) affiche « foo » 1000 fois
- n) affiche « foo » une fois

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » 8 fois
- c) échoue
- d) affiche « foo » 7 fois
- e) affiche « foo » 6 fois
- f) n'affiche rien
- g) affiche « foo » une infinité de fois

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 13 »
- b) affiche « 12 »
- c) affiche « 10 »
- d) affiche « 14 »
- e) affiche « 8 »
- f) affiche « b »
- g) affiche « 11 »
- h) n'affiche rien
- i) échoue
- j) boucle infiniment
- k) affiche « a »
- l) affiche « 9 »

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » 3 fois
- c) n'affiche rien
- d) échoue
- e) affiche « foo » 4 fois
- f) affiche « foo » 6 fois
- g) affiche « foo » une infinité de fois
- h) affiche « foo » 2 fois
- i) affiche « foo » 5 fois

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » deux fois
- b) affiche « foo » trois fois
- c) échoue
- d) n'affiche rien
- e) affiche « foo » une fois
- f) affiche « foo » quatre fois
- g) affiche « foo » une infinité de fois

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

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

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

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

```
"foo" * 3
```

- a) n'a aucun résultat
- b) a comme résultat un float-tant
- c) a comme résultat un entier
- d) a comme résultat une chaîne de caractères
- e) a comme résultat un booléen
- f) a comme résultat la valeur de la variable foo multipliée par 3
- g) échoue

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 29

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » quatre fois
- b) échoue
- c) affiche « foo » une infinité de fois
- d) affiche « foo » trois fois
- e) affiche « foo » une fois
- f) affiche « foo » deux fois
- g) n'affiche rien

Question 2 (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
- « bar » une infinité de fois
- e) échoue
- f) n'affiche rien
- g) affiche « foo » une fois et « bar » une fois
- h) affiche juste « foo », une fois

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » une fois
- b) n'affiche rien
- c) affiche « foo » 6 fois
- d) affiche « foo » 2 fois
- e) échoue
- f) affiche « foo » 4 fois
- g) affiche « foo » 3 fois
- h) affiche « foo » 5 fois
- i) affiche « foo » 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 » 33 fois
- b) affiche « foo » une fois
- c) affiche « foo » 10 fois
- d) affiche « foo » 9 fois
- e) n'affiche rien
- f) échoue
- g) affiche « foo » une infinité de fois
- h) affiche « foo » 30 fois
- i) affiche « foo » 1000 fois
- j) affiche « foo » 27 fois
- k) affiche « foo » 20 fois
- l) affiche « foo » 10000 fois
- m) affiche « foo » 3 fois
- n) affiche « foo » 100 fois

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

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

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

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « a »
- b) affiche « 12 »
- c) affiche « 8 »
- d) affiche « 9 »
- e) échoue
- f) affiche « 14 »
- g) affiche « b »
- h) affiche « 10 »
- i) affiche « 13 »
- j) affiche « 11 »
- k) n'affiche rien
- l) boucle infiniment

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) n'affiche rien
- b) échoue
- c) affiche « foo » une fois
- d) affiche « foo » 8 fois
- e) affiche « foo » 7 fois
- f) affiche « foo » 6 fois
- g) affiche « foo » une infinité de fois

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

```
"foo" * 3
```

- a) a comme résultat un float-tant
- b) n'a aucun résultat
- c) a comme résultat un entier
- d) a comme résultat un booléen
- e) a comme résultat une chaîne de caractères
- f) a comme résultat la valeur de la variable foo multipliée par 3
- g) échoue

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villeteuse, Université Paris 13

Exemplaire numéro 30

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » une fois
- b) échoue
- c) affiche « foo » 7 fois
- d) affiche « foo » 8 fois
- e) affiche « foo » 6 fois
- f) affiche « foo » une infinité de fois
- g) n'affiche rien

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

```
"foo" * 3
```

- a) a comme résultat un float-tant
- b) a comme résultat un entier
- c) a comme résultat la valeur de la variable foo multipliée par 3
- d) échoue
- e) a comme résultat un booléen
- f) n'a aucun résultat
- g) a comme résultat une chaîne de caractères

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 10 »
- b) affiche « 13 »
- c) boucle infiniment
- d) affiche « a »
- e) n'affiche rien
- f) affiche « 12 »
- g) affiche « 11 »
- h) affiche « 8 »
- i) affiche « b »
- j) affiche « 9 »
- k) échoue
- l) affiche « 14 »

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » 3 fois
- c) affiche « foo » une infinité de fois
- d) affiche « foo » 2 fois
- e) affiche « foo » 5 fois
- f) affiche « foo » 4 fois
- g) échoue
- h) affiche « foo » 6 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 » 30 fois
- b) affiche « foo » 9 fois
- c) affiche « foo » 33 fois
- d) affiche « foo » 1000 fois
- e) affiche « foo » 27 fois
- f) affiche « foo » 3 fois
- g) affiche « foo » une infinité de fois
- h) affiche « foo » 20 fois
- i) affiche « foo » 100 fois
- j) affiche « foo » 10 fois
- k) affiche « foo » 10000 fois
- l) affiche « foo » une fois
- m) échoue
- n) n'affiche rien

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

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

- a) affiche « bar » une infinité de fois
- b) n'affiche rien
- 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 juste « bar », une fois
- h) affiche « foo » une fois et « bar » une fois

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » deux fois
- b) affiche « foo » trois fois
- c) affiche « foo » une infinité de fois
- d) échoue
- e) affiche « foo » une fois
- f) affiche « foo » quatre fois
- g) n'affiche rien

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

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

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

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 31

Luca Saiu

Documents (livres, slides, photocopiés, 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
- b) affiche « foo » 30 fois
- c) affiche « foo » 20 fois
- d) affiche « foo » 3 fois
- e) affiche « foo » 100 fois
- f) affiche « foo » 1000 fois
- g) affiche « foo » une fois
- h) affiche « foo » 27 fois
- i) n'affiche rien
- j) échoue
- k) affiche « foo » une infinité de fois
- l) affiche « foo » 10 fois
- m) affiche « foo » 10000 fois
- n) affiche « foo » 33 fois

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

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

- a) affiche « bar » une infinité de fois
- b) affiche juste « foo », une fois
- c) n'affiche rien
- d) affiche juste « bar », une fois
- e) échoue
- 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 3 (2 points) Le programme suivant ...

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 3 fois
- b) affiche « foo » 4 fois
- c) affiche « foo » une fois
- d) affiche « foo » 6 fois
- e) affiche « foo » 5 fois
- f) affiche « foo » une infinité de fois
- g) n'affiche rien
- h) échoue
- i) affiche « foo » 2 fois

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

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

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

Question 6 (2 points) L'expression suivante ...

```
"foo" * 3
```

- a) a comme résultat une chaîne de caractères
- b) a comme résultat un entier
- c) échoue
- d) a comme résultat un booléen
- e) n'a aucun résultat
- f) a comme résultat la valeur de la variable foo multipliée par 3
- g) a comme résultat un floatant

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 12 »
- b) affiche « 11 »
- c) affiche « 14 »
- d) affiche « b »
- e) affiche « 9 »
- f) affiche « 13 »
- g) échoue
- h) affiche « a »
- i) boucle infiniment
- j) affiche « 10 »
- k) n'affiche rien
- l) affiche « 8 »

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » quatre fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » trois fois
- d) affiche « foo » deux fois
- e) affiche « foo » une fois
- f) n'affiche rien
- g) échoue

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) n'affiche rien
- b) affiche « foo » 6 fois
- c) affiche « foo » une fois
- d) affiche « foo » 7 fois
- e) affiche « foo » 8 fois
- f) affiche « foo » une infinité de fois
- g) échoue

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 32

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

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

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

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

```
"foo" * 3
```

- a) n'a aucun résultat
- b) a comme résultat un float-tant
- c) échoue
- d) a comme résultat un booléen
- e) a comme résultat un entier
- f) a comme résultat une chaîne de caractères
- g) a comme résultat la valeur de la variable `foo` multipliée par 3

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » quatre fois
- b) échoue
- c) affiche « foo » deux fois
- d) affiche « foo » une fois
- e) affiche « foo » trois fois
- f) n'affiche rien
- g) affiche « foo » 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 » 100 fois
- b) affiche « foo » 33 fois
- c) affiche « foo » 10000 fois
- d) affiche « foo » une infinité de fois
- e) affiche « foo » 1000 fois
- f) affiche « foo » 20 fois
- g) affiche « foo » 27 fois
- h) affiche « foo » 10 fois
- i) affiche « foo » une fois
- j) échoue
- k) n'affiche rien
- l) affiche « foo » 9 fois
- m) affiche « foo » 30 fois
- n) affiche « foo » 3 fois

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » 6 fois
- c) affiche « foo » une infinité de fois
- d) affiche « foo » 7 fois
- e) affiche « foo » 8 fois
- f) échoue
- g) n'affiche rien

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

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

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

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « b »
- b) affiche « 14 »
- c) affiche « 10 »
- d) affiche « 8 »
- e) échoue
- f) affiche « 12 »
- g) n'affiche rien
- h) boucle infiniment
- i) affiche « 13 »
- j) affiche « 11 »
- k) affiche « 9 »
- l) affiche « a »

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 2 fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 5 fois
- d) échoue
- e) n'affiche rien
- f) affiche « foo » 4 fois
- g) affiche « foo » 3 fois
- h) affiche « foo » une fois
- i) affiche « foo » 6 fois

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 33

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

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

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

- a) affiche « foo » 100 fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 20 fois
- d) affiche « foo » 30 fois
- e) affiche « foo » 3 fois
- f) affiche « foo » 27 fois
- g) affiche « foo » une fois
- h) affiche « foo » 9 fois
- i) affiche « foo » 10000 fois
- j) n'affiche rien
- k) échoue
- l) affiche « foo » 10 fois
- m) affiche « foo » 33 fois
- n) affiche « foo » 1000 fois

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 11 »
- b) affiche « 14 »
- c) affiche « 10 »
- d) affiche « 12 »
- e) n'affiche rien
- f) affiche « 13 »
- g) affiche « 8 »
- h) boucle infiniment
- i) affiche « a »
- j) affiche « 9 »
- k) affiche « b »
- l) échoue

Question 6 (2 points) L'expression suivante ...

```
"foo" * 3
```

- a) a comme résultat un entier
- b) a comme résultat la valeur de la variable foo multipliée par 3
- c) a comme résultat un float-tant
- d) a comme résultat une chaîne de caractères
- e) échoue
- f) a comme résultat un booléen
- g) n'a aucun résultat

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 5 fois
- b) n'affiche rien
- c) affiche « foo » 6 fois
- d) échoue
- e) affiche « foo » 2 fois
- f) affiche « foo » 4 fois
- g) affiche « foo » 3 fois
- h) affiche « foo » une fois
- i) affiche « foo » une infinité de fois

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » une infinité de fois
- c) n'affiche rien
- d) affiche « foo » trois fois
- e) échoue
- f) affiche « foo » deux fois
- g) affiche « foo » quatre fois

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

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

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

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 (2 points) Le programme suivant ...

```
a = 7
while a < 8:
    print ("foo")
```

- a) échoue
- b) affiche « foo » 8 fois
- c) affiche « foo » 6 fois
- d) affiche « foo » 7 fois
- e) n'affiche rien
- f) affiche « foo » une fois
- g) affiche « foo » une infinité de fois

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 34

Luca Saiu

Documents (livres, slides, photocopies, code source, photocopiers, 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) échoue
- c) affiche « foo » une infinité de fois
- d) n'affiche rien
- e) affiche juste « bar », une fois
- f) affiche « bar » une infinité de fois
- g) affiche « foo » une fois et « bar » une infinité de fois
- h) affiche « foo » une fois et « bar » une fois

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » une fois
- b) n'affiche rien
- c) affiche « foo » 5 fois
- d) affiche « foo » une infinité de fois
- e) échoue
- f) affiche « foo » 4 fois
- g) affiche « foo » 2 fois
- h) affiche « foo » 3 fois
- i) affiche « foo » 6 fois

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » une infinité de fois
- b) affiche « foo » deux fois
- c) n'affiche rien
- d) affiche « foo » trois fois
- e) affiche « foo » quatre fois
- f) affiche « foo » une fois
- g) échoue

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » 7 fois
- b) affiche « foo » une fois
- c) affiche « foo » une infinité de fois
- d) affiche « foo » 8 fois
- e) échoue
- f) n'affiche rien
- g) affiche « foo » 6 fois

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 8 »
- b) affiche « 9 »
- c) affiche « 13 »
- d) affiche « 14 »
- e) affiche « 10 »
- f) n'affiche rien
- g) boucle infiniment
- h) affiche « 12 »
- i) affiche « a »
- j) affiche « 11 »
- k) échoue
- l) affiche « b »

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

```
"foo" * 3
```

- a) a comme résultat la valeur de la variable foo multipliée par 3
- b) n'a aucun résultat
- c) a comme résultat une chaîne de caractères
- d) a comme résultat un booléen
- e) a comme résultat un float
- f) échoue
- g) a comme résultat un entier

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

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

- a) affiche juste « foo », une fois
- b) affiche « foo » une infinité de fois
- c) affiche juste « bar », une fois
- d) n'affiche rien
- e) affiche « bar » une infinité de fois
- f) affiche « foo » une fois et « bar » une infinité de fois
- g) affiche « foo » une fois et « bar » une fois
- h) 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 » 3 fois
- b) affiche « foo » 100 fois
- c) affiche « foo » 27 fois
- d) affiche « foo » 20 fois
- e) affiche « foo » une infinité de fois
- f) affiche « foo » 33 fois
- g) affiche « foo » 30 fois
- h) affiche « foo » une fois
- i) affiche « foo » 9 fois
- j) n'affiche rien
- k) échoue
- l) affiche « foo » 10000 fois
- m) affiche « foo » 1000 fois
- n) affiche « foo » 10 fois

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 35

Luca Saiu

Documents (livres, slides, photocopiés, 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 = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » 6 fois de fois
- b) affiche « foo » une fois e) affiche « foo » 7 fois
- c) affiche « foo » 8 fois f) n'affiche rien
- d) affiche « foo » une infinité g) échoue

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

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

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

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

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

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

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

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 5 fois f) affiche « foo » une infinité de fois
- b) affiche « foo » une fois
- c) affiche « foo » 4 fois g) affiche « foo » 3 fois
- d) affiche « foo » 6 fois h) échoue
- e) affiche « foo » 2 fois i) n'affiche rien

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

```
a = 8
for b in range (4):
    a = a + b
    print (a)
```

- a) affiche « 14 » g) affiche « b »
- b) boucle infiniment h) n'affiche rien
- c) affiche « 8 » i) échoue
- d) affiche « 10 » j) affiche « 12 »
- e) affiche « 13 » k) affiche « a »
- f) affiche « 9 » l) affiche « 11 »

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » quatre fois e) échoue
- b) n'affiche rien f) affiche « foo » une fois
- c) affiche « foo » trois fois g) affiche « foo » une infinité de fois
- d) affiche « foo » deux fois

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
"foo" * 3
```

- a) a comme résultat un booléen tant
- b) a comme résultat la valeur de la variable foo multipliée par 3 d) n'a aucun résultat
- c) a comme résultat un float- g) échoue
- e) a comme résultat un entier
- f) a comme résultat une chaîne de caractères

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villeteuse, Université Paris 13

Exemplaire numéro 36

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

```
"foo" * 3
```

- a) a comme résultat un booléen
- b) a comme résultat une chaîne de caractères
- c) échoue
- d) a comme résultat un float
- e) tant
- f) n'a aucun résultat
- g) a comme résultat un entier
- h) a comme résultat la valeur de la variable `foo` multipliée par 3

- a) échoue
- b) affiche « foo » quatre fois
- c) affiche « foo » une fois
- d) n'affiche rien
- e) affiche « foo » trois fois
- f) affiche « foo » deux fois
- g) affiche « foo » une infinité de fois

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

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

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

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

```
a = 7  
while a < 8:  
    print ("foo")
```

- a) n'affiche rien
- b) affiche « foo » une fois
- c) échoue
- d) affiche « foo » 8 fois
- e) affiche « foo » une infinité de fois
- f) affiche « foo » 6 fois
- g) affiche « foo » 7 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 » 100 fois
- b) affiche « foo » 1000 fois
- c) affiche « foo » une infinité de fois
- d) affiche « foo » 10 fois
- e) affiche « foo » 3 fois
- f) affiche « foo » 30 fois
- g) affiche « foo » une fois
- h) affiche « foo » 27 fois
- i) affiche « foo » 9 fois
- j) affiche « foo » 10000 fois
- k) affiche « foo » 20 fois
- l) n'affiche rien
- m) échoue
- n) affiche « foo » 33 fois

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

```
x = 8  
if x > 7:  
    print ("foo")  
else x <= 7:  
    print ("bar")
```

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

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

```
a = 8  
for b in range (4):  
    a = a + b  
print (a)
```

- a) affiche « a »
- b) affiche « 9 »
- c) échoue
- d) affiche « 10 »
- e) affiche « 11 »
- f) affiche « 14 »
- g) affiche « 13 »
- h) boucle infiniment
- i) affiche « 12 »
- j) affiche « 8 »
- k) affiche « b »
- l) n'affiche rien

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

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

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

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

```
if True:  
    print ("foo")  
print ("foo")  
else:  
    print ("foo")
```

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

```
for i in range (3):  
    print ("foo")  
for j in range (2):  
    print ("foo")
```

- a) affiche « foo » 4 fois
- b) n'affiche rien
- c) affiche « foo » une fois
- d) affiche « foo » 6 fois
- e) échoue
- f) affiche « foo » 5 fois
- g) affiche « foo » 3 fois
- h) affiche « foo » 2 fois
- i) affiche « foo » une infinité de fois

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 37

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

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

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

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 4 fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 2 fois
- d) échoue
- e) affiche « foo » 6 fois
- f) affiche « foo » 3 fois
- g) affiche « foo » 5 fois
- h) affiche « foo » une fois
- i) n'affiche rien

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) échoue
- b) affiche « foo » trois fois
- c) n'affiche rien
- d) affiche « foo » deux fois
- e) affiche « foo » quatre fois
- f) affiche « foo » une infinité de fois
- g) affiche « foo » une 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 » 9 fois
- b) affiche « foo » une fois
- c) affiche « foo » 33 fois
- d) affiche « foo » 1000 fois
- e) affiche « foo » 20 fois
- f) affiche « foo » 30 fois
- g) échoue
- h) affiche « foo » 100 fois
- i) affiche « foo » une infinité de fois
- j) affiche « foo » 3 fois
- k) n'affiche rien
- l) affiche « foo » 27 fois
- m) affiche « foo » 10 fois
- n) affiche « foo » 10000 fois

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « foo » 7 fois
- b) affiche « foo » une fois
- c) n'affiche rien
- d) échoue
- e) affiche « foo » 8 fois
- f) affiche « foo » 6 fois
- g) affiche « foo » une infinité de fois

- g) affiche « 13 »
- h) affiche « 9 »
- i) affiche « b »
- j) affiche « 8 »
- k) échoue
- l) affiche « a »

Question 6 (2 points) L'expression suivante ...

```
"foo" * 3
```

- a) échoue
- b) a comme résultat un entier
- c) n'a aucun résultat
- d) a comme résultat un booléen
- e) a comme résultat la valeur de la variable foo multipliée par 3
- f) a comme résultat un floatant
- g) a comme résultat une chaîne de caractères

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

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

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

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » 7 fois
- b) affiche « foo » une fois
- c) n'affiche rien
- d) échoue
- e) affiche « foo » 8 fois
- f) affiche « foo » 6 fois
- g) affiche « foo » une infinité de fois

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 38

Luca Saiu

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

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) échoue d) n'affiche rien
- b) affiche « foo » une infinité e) affiche « foo » une fois de fois fois
- c) affiche « foo » trois fois f) affiche « foo » deux fois fois
- g) affiche « foo » quatre fois

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « b » g) n'affiche rien
- b) affiche « 8 » h) échoue
- c) boucle infiniment i) affiche « 10 »
- d) affiche « 13 » j) affiche « 11 »
- e) affiche « 9 » k) affiche « a »
- f) affiche « 14 » l) affiche « 12 »

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 6 fois f) affiche « foo » une infinité de fois fois
- b) affiche « foo » 5 fois
- c) affiche « foo » 3 fois g) échoue
- d) affiche « foo » une fois h) n'affiche rien
- e) affiche « foo » 4 fois i) affiche « foo » 2 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")
```

- a) affiche « foo » 20 fois h) affiche « foo » 3 fois
- b) affiche « foo » 100 fois i) affiche « foo » 33 fois
- c) affiche « foo » 1000 fois j) affiche « foo » 9 fois
- d) affiche « foo » 10 fois k) affiche « foo » une fois
- e) affiche « foo » une infinité l) affiche « foo » 30 fois de fois fois
- m) affiche « foo » 27 fois
- f) n'affiche rien n) échoue
- g) affiche « foo » 10000 fois

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

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

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

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » 8 fois d) n'affiche rien
- b) affiche « foo » une infinité e) affiche « foo » une fois de fois fois
- f) affiche « foo » 7 fois
- c) échoue g) affiche « foo » 6 fois

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

```
"foo" * 3
```

- a) échoue pliée par 3
- b) a comme résultat un entier e) n'a aucun résultat
- c) a comme résultat un booléen f) a comme résultat un float-tant
- d) a comme résultat la valeur de la variable foo multi-g) a comme résultat une chaîne de caractères

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 39

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) n'affiche rien
- b) affiche « foo » deux fois
- c) affiche « foo » une fois
- d) échoue
- e) affiche « foo » quatre fois
- f) affiche « foo » trois fois
- g) affiche « foo » une infinité de fois

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

```
"foo" * 3
```

- a) a comme résultat un entier
- b) a comme résultat la valeur de la variable `foo` multipliée par 3
- c) a comme résultat une chaîne de caractères
- d) a comme résultat un booléen
- e) échoue
- f) a comme résultat un float
- g) n'a aucun résultat

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 3 fois
- b) échoue
- c) affiche « foo » 4 fois
- d) affiche « foo » 5 fois
- e) affiche « foo » une fois
- f) affiche « foo » une infinité de fois
- g) affiche « foo » 6 fois
- h) n'affiche rien
- i) affiche « foo » 2 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
- b) affiche « foo » 33 fois
- c) affiche « foo » 27 fois
- d) affiche « foo » 100 fois
- e) affiche « foo » une infinité de fois
- f) affiche « foo » 1000 fois
- g) affiche « foo » 9 fois
- h) affiche « foo » 30 fois
- i) n'affiche rien
- j) affiche « foo » une fois
- k) affiche « foo » 10 fois
- l) échoue
- m) affiche « foo » 20 fois
- n) affiche « foo » 10000 fois

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » une infinité de fois
- b) affiche « foo » 7 fois
- c) affiche « foo » une fois
- d) affiche « foo » 6 fois
- e) affiche « foo » 8 fois
- f) n'affiche rien
- g) échoue

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 10 »
- b) affiche « 9 »
- c) affiche « 8 »
- d) n'affiche rien
- e) échoue
- f) boucle infiniment
- g) affiche « 12 »
- h) affiche « 13 »
- i) affiche « 11 »
- j) affiche « a »
- k) affiche « b »
- l) affiche « 14 »

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

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

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

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

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

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

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 40

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 6 fois
- b) n'affiche rien
- c) affiche « foo » 4 fois
- d) affiche « foo » 5 fois
- e) affiche « foo » une infinité de fois
- f) affiche « foo » 2 fois
- g) échoue
- h) affiche « foo » 3 fois
- i) affiche « foo » une fois

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » 6 fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 7 fois
- d) affiche « foo » 8 fois
- e) échoue
- f) affiche « foo » une fois
- g) n'affiche rien

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

```
"foo" * 3
```

- a) n'a aucun résultat
- b) a comme résultat une chaîne de caractères
- c) a comme résultat un float-tant
- d) a comme résultat un entier
- e) a comme résultat un booleen
- f) échoue
- g) a comme résultat la valeur de la variable foo multipliée par 3

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

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

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

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » trois fois
- b) n'affiche rien
- c) échoue
- d) affiche « foo » une infinité de fois
- e) affiche « foo » une fois
- f) affiche « foo » quatre fois
- g) affiche « foo » deux 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) n'affiche rien
- b) affiche « foo » 30 fois
- c) affiche « foo » 33 fois
- d) affiche « foo » 10000 fois
- e) affiche « foo » 3 fois
- f) affiche « foo » une fois
- g) échoue
- h) affiche « foo » 100 fois
- i) affiche « foo » 1000 fois
- j) affiche « foo » une infinité de fois
- k) affiche « foo » 9 fois
- l) affiche « foo » 27 fois
- m) affiche « foo » 20 fois
- n) affiche « foo » 10 fois

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 13 »
- b) affiche « 12 »
- c) affiche « 9 »
- d) boucle infiniment
- e) affiche « 8 »
- f) affiche « 11 »
- g) affiche « 14 »
- h) échoue
- i) affiche « 10 »
- j) n'affiche rien
- k) affiche « b »
- l) affiche « a »

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

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

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

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 41

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » trois fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » une fois
- d) n'affiche rien
- e) affiche « foo » deux fois
- f) échoue
- g) affiche « foo » quatre fois

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) n'affiche rien
- b) affiche « 13 »
- c) affiche « 10 »
- d) affiche « a »
- e) affiche « b »
- f) affiche « 8 »
- g) affiche « 11 »
- h) boucle infiniment
- i) affiche « 14 »
- j) affiche « 9 »
- k) affiche « 12 »
- l) échoue

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

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

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

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

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

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

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 2 fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 6 fois
- d) n'affiche rien
- e) affiche « foo » 4 fois
- f) affiche « foo » 3 fois
- g) échoue
- h) affiche « foo » une fois
- i) affiche « foo » 5 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 infinité de fois
- b) affiche « foo » 27 fois
- c) affiche « foo » une fois
- d) affiche « foo » 3 fois
- e) affiche « foo » 9 fois
- f) affiche « foo » 10000 fois
- g) affiche « foo » 33 fois
- h) affiche « foo » 1000 fois
- i) affiche « foo » 10 fois
- j) affiche « foo » 20 fois
- k) échoue
- l) affiche « foo » 100 fois
- m) n'affiche rien
- n) affiche « foo » 30 fois

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

```
"foo" * 3
```

- a) a comme résultat un booléen
- b) a comme résultat un entier
- c) a comme résultat un float
- d) a comme résultat une chaîne de caractères
- e) n'a aucun résultat
- f) a comme résultat la valeur de la variable foo multipliée par 3
- g) échoue

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » 6 fois
- b) affiche « foo » 8 fois
- c) échoue
- d) affiche « foo » une fois
- e) affiche « foo » 7 fois
- f) affiche « foo » une infinité de fois
- g) n'affiche rien

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 42

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

```
"foo" * 3
```

- a) a comme résultat une chaîne de caractères
- b) a comme résultat un float-tant
- c) a comme résultat un entier
- d) a comme résultat la valeur
- e) de la variable `foo` multipliée par 3
- f) a comme résultat un booléen
- g) échoue
- h) n'a aucun résultat

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) échoue
- b) affiche « foo » une fois
- c) n'affiche rien
- d) affiche « foo » 7 fois
- e) affiche « foo » 8 fois
- f) affiche « foo » 6 fois
- g) affiche « foo » une infinité de fois

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

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

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

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « b »
- b) affiche « 11 »
- c) affiche « 13 »
- d) échoue
- e) affiche « 12 »
- f) affiche « 14 »
- g) affiche « 8 »
- h) n'affiche rien
- i) affiche « a »
- j) affiche « 9 »
- k) affiche « 10 »
- l) boucle infiniment

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » quatre fois
- b) affiche « foo » trois fois
- c) échoue
- d) affiche « foo » une fois
- e) n'affiche rien
- f) affiche « foo » deux fois
- g) affiche « foo » une infinité de 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) échoue
- b) affiche « foo » 27 fois
- c) affiche « foo » 20 fois
- d) affiche « foo » 30 fois
- e) affiche « foo » 3 fois
- f) affiche « foo » une infinité de fois
- g) affiche « foo » 10 fois
- h) affiche « foo » 33 fois
- i) affiche « foo » 9 fois
- j) affiche « foo » 100 fois
- k) affiche « foo » une fois
- l) n'affiche rien
- m) affiche « foo » 10000 fois
- n) affiche « foo » 1000 fois

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

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

- a) échoue
- b) affiche « bar » une infinité de fois
- c) affiche « foo » une infinité de fois
- d) affiche juste « bar », une fois
- e) n'affiche rien
- 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 9 (2 points) Le programme suivant ...

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 2 fois
- b) affiche « foo » 6 fois
- c) échoue
- d) affiche « foo » 4 fois
- e) affiche « foo » 3 fois
- f) n'affiche rien
- g) affiche « foo » une infinité de fois
- h) affiche « foo » 5 fois
- i) affiche « foo » une fois

Programmation Python
Contrôle des connaissances : contrôle surprise B1
IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 43

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

- | | |
|---|-----------------------------|
| a) affiche « foo » 7 fois | d) affiche « foo » 6 fois |
| b) affiche « foo » 8 fois | e) affiche « foo » une fois |
| c) affiche « foo » une infinité de fois | f) n'affiche rien |
| | g) échoue |

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

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

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

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

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

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

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

```
"foo" * 3
```

- | | |
|--|---|
| a) a comme résultat une chaîne de caractères | e) a comme résultat la valeur de la variable foo multipliée par 3 |
| b) a comme résultat un booléen | f) a comme résultat un entier |
| c) échoue | g) a comme résultat un float-tant |
| d) n'a aucun résultat | |

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- | | |
|-------------------|----------------------|
| a) échoue | g) affiche « 14 » |
| b) affiche « 10 » | h) affiche « 11 » |
| c) affiche « 12 » | i) affiche « 13 » |
| d) affiche « 9 » | j) affiche « 8 » |
| e) affiche « a » | k) boucle infiniment |
| f) affiche « b » | l) 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 » une infinité de fois | h) affiche « foo » 10 fois |
| b) affiche « foo » 3 fois | i) affiche « foo » une fois |
| c) affiche « foo » 30 fois | j) affiche « foo » 10000 fois |
| d) n'affiche rien | k) affiche « foo » 9 fois |
| e) affiche « foo » 100 fois | l) affiche « foo » 20 fois |
| f) affiche « foo » 33 fois | m) échoue |
| g) affiche « foo » 1000 fois | n) affiche « foo » 27 fois |

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- | | |
|---------------------------|---|
| a) échoue | f) affiche « foo » une fois |
| b) affiche « foo » 3 fois | g) n'affiche rien |
| c) affiche « foo » 2 fois | h) affiche « foo » 4 fois |
| d) affiche « foo » 6 fois | i) affiche « foo » une infinité de fois |
| e) affiche « foo » 5 fois | |

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- | | |
|---------------------------------|--------------------------------|
| a) affiche « foo » deux fois | de fois |
| b) affiche « foo » trois fois | e) n'affiche rien |
| c) affiche « foo » une fois | f) affiche « foo » quatre fois |
| d) affiche « foo » une infinité | g) échoue |

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

```
a = 7
while a < 8:
    print ("foo")
```

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 44

Luca Saiu

Documents (livres, slides, photocopiés, 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 (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 6 fois
- b) affiche « foo » 3 fois
- c) affiche « foo » une infinité de fois
- d) affiche « foo » 5 fois
- e) affiche « foo » 4 fois
- f) n'affiche rien
- g) échoue
- h) affiche « foo » une fois
- i) affiche « foo » 2 fois

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

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

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

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 6 fois
- d) affiche « foo » 8 fois
- e) affiche « foo » 7 fois
- f) échoue
- g) n'affiche rien

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 10 »
- b) affiche « 13 »
- c) affiche « 14 »
- d) échoue
- e) affiche « a »
- f) affiche « 9 »
- g) affiche « 8 »
- h) boucle infiniment
- i) affiche « 11 »
- j) affiche « b »
- k) affiche « 12 »
- l) 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 » 3 fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 10 fois
- d) affiche « foo » 27 fois
- e) échoue
- f) affiche « foo » 33 fois
- g) affiche « foo » 1000 fois
- h) affiche « foo » 20 fois
- i) n'affiche rien
- j) affiche « foo » 100 fois
- k) affiche « foo » 10000 fois
- l) affiche « foo » une fois
- m) affiche « foo » 9 fois
- n) affiche « foo » 30 fois

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » une infinité de fois
- b) affiche « foo » trois fois
- c) affiche « foo » une fois
- d) affiche « foo » quatre fois
- e) n'affiche rien
- f) échoue
- g) affiche « foo » deux fois

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

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

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

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

```
"foo" * 3
```

- a) a comme résultat un float-tant
- b) n'a aucun résultat
- c) a comme résultat un entier
- d) échoue
- e) a comme résultat la valeur de la variable foo multipliée par 3
- f) a comme résultat une chaîne de caractères
- g) a comme résultat un booléen

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 45

Luca Saiu

Documents (livres, slides, photocopiés, 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
- b) affiche « foo » 30 fois
- c) affiche « foo » 3 fois
- d) affiche « foo » une infinité de fois
- e) affiche « foo » 10000 fois
- f) affiche « foo » une fois
- g) affiche « foo » 100 fois
- h) affiche « foo » 20 fois
- i) n'affiche rien
- j) affiche « foo » 9 fois
- k) échoue
- l) affiche « foo » 33 fois
- m) affiche « foo » 1000 fois
- n) affiche « foo » 27 fois

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) n'affiche rien
- b) affiche « foo » 4 fois
- c) affiche « foo » 3 fois
- d) affiche « foo » une infinité de fois
- e) affiche « foo » une fois
- f) affiche « foo » 5 fois
- g) échoue
- h) affiche « foo » 6 fois
- i) affiche « foo » 2 fois

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

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

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

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » 7 fois
- b) affiche « foo » 8 fois
- c) affiche « foo » une infinité de fois
- d) affiche « foo » 6 fois
- e) affiche « foo » une fois
- f) n'affiche rien
- g) échoue

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

```
"foo" * 3
```

- a) a comme résultat un float-tant
- b) échoue
- c) a comme résultat la valeur de la variable foo multipliée par 3
- d) n'a aucun résultat
- e) a comme résultat un booléen
- f) a comme résultat une chaîne de caractères
- g) a comme résultat un entier

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » quatre fois
- b) affiche « foo » une infinité de fois
- c) échoue
- d) n'affiche rien
- e) affiche « foo » deux fois
- f) affiche « foo » trois fois
- g) affiche « foo » une fois

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) échoue
- b) affiche « 11 »
- c) affiche « 14 »
- d) boucle infiniment
- e) affiche « 9 »
- f) affiche « b »
- g) affiche « 13 »
- h) affiche « 8 »
- i) affiche « 10 »
- j) affiche « a »
- k) affiche « 12 »
- l) n'affiche rien

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

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

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

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 46

Luca Saiu

Documents (livres, slides, photocopiés, 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 » 27 fois
- b) affiche « foo » 20 fois
- c) affiche « foo » 100 fois
- d) affiche « foo » une fois
- e) affiche « foo » 9 fois
- f) échoue
- g) affiche « foo » 1000 fois
- h) affiche « foo » 30 fois
- i) affiche « foo » 10000 fois
- j) affiche « foo » 10 fois
- k) affiche « foo » 33 fois
- l) affiche « foo » 3 fois
- m) affiche « foo » une infinité de fois
- n) n'affiche rien

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

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

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

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 2 fois
- b) échoue
- c) affiche « foo » une infinité de fois
- d) affiche « foo » 6 fois
- e) affiche « foo » 3 fois
- f) affiche « foo » une fois
- g) affiche « foo » 5 fois
- h) affiche « foo » 4 fois
- i) n'affiche rien

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

```
"foo" * 3
```

- a) a comme résultat un entier
- b) n'a aucun résultat
- c) a comme résultat la valeur de la variable foo multipliée par 3
- d) échoue
- e) a comme résultat un float
- f) a comme résultat un booléen
- g) a comme résultat une chaîne de caractères

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » une infinité de fois
- b) affiche « foo » deux fois
- c) affiche « foo » trois fois
- d) affiche « foo » une fois
- e) affiche « foo » quatre fois
- f) échoue
- g) n'affiche rien

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » 7 fois
- b) affiche « foo » 6 fois
- c) échoue
- d) affiche « foo » une infinité de fois
- e) affiche « foo » 8 fois
- f) n'affiche rien
- g) affiche « foo » une fois

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

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

- a) affiche « foo » une infinité de fois
- b) affiche « bar » une infinité de fois
- c) affiche « foo » une fois et « bar » une infinité de fois
- d) n'affiche rien
- e) échoue
- f) affiche juste « 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 = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 14 »
- b) boucle infiniment
- c) affiche « 8 »
- d) affiche « b »
- e) affiche « 11 »
- f) affiche « a »
- g) affiche « 12 »
- h) affiche « 9 »
- i) n'affiche rien
- j) affiche « 10 »
- k) échoue
- l) affiche « 13 »

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 47

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

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

- a) affiche « foo » une infinité de fois
- b) affiche « foo » une fois et « bar » une fois
- c) échoue
- d) affiche juste « bar », 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 « foo », une fois

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » deux fois
- b) affiche « foo » quatre fois
- c) échoue
- d) affiche « foo » une fois
- e) affiche « foo » une infinité de fois
- f) n'affiche rien
- g) affiche « foo » trois fois

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » une infinité de fois
- b) n'affiche rien
- c) échoue
- d) affiche « foo » une fois
- e) affiche « foo » 7 fois
- f) affiche « foo » 6 fois
- g) affiche « foo » 8 fois

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 2 fois
- b) échoue
- c) affiche « foo » une infinité de fois
- d) affiche « foo » 3 fois
- e) affiche « foo » une fois
- f) n'affiche rien
- g) affiche « foo » 6 fois
- h) affiche « foo » 5 fois
- i) affiche « foo » 4 fois

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) boucle infiniment
- b) affiche « 11 »
- c) affiche « 12 »
- d) n'affiche rien
- e) affiche « 10 »
- f) affiche « 8 »
- g) affiche « a »
- h) affiche « 14 »
- i) affiche « 9 »
- j) échoue
- k) affiche « 13 »
- l) affiche « b »

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 » 10000 fois
- b) affiche « foo » 20 fois
- c) affiche « foo » 10 fois
- d) affiche « foo » 1000 fois
- e) affiche « foo » 30 fois
- f) échoue
- g) affiche « foo » 9 fois
- h) n'affiche rien
- i) affiche « foo » une infinité de fois
- j) affiche « foo » 27 fois
- k) affiche « foo » 100 fois
- l) affiche « foo » une fois
- m) affiche « foo » 33 fois
- n) affiche « foo » 3 fois

Question 8 (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) affiche « foo » une fois et « bar » une infinité de fois
- d) affiche juste « bar », une fois
- e) affiche « foo » une infinité de fois
- f) échoue
- g) affiche « bar » une infinité de fois
- h) n'affiche rien

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

```
"foo" * 3
```

- a) a comme résultat une chaîne de caractères
- b) échoue
- c) a comme résultat un entier
- d) a comme résultat un float-tant
- e) n'a aucun résultat
- f) a comme résultat un booléen
- g) a comme résultat la valeur de la variable foo multipliée par 3

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 48

Luca Saiu

Documents (livres, slides, photocopiés, 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 = 8
for b in range(4):
    a = a + b
print(a)
```

- a) échoue
- b) affiche « 14 »
- c) affiche « 11 »
- d) affiche « b »
- e) affiche « 13 »
- f) n'affiche rien
- g) affiche « 10 »
- h) affiche « 9 »
- i) affiche « 12 »
- j) affiche « a »
- k) affiche « 8 »
- l) boucle infiniment

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

```
for i in range(3):
    print("foo")
for j in range(2):
    print("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » 2 fois
- d) affiche « foo » 6 fois
- e) affiche « foo » 3 fois
- f) échoue
- g) affiche « foo » 4 fois
- h) affiche « foo » 5 fois
- i) n'affiche rien

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

```
x = 8
if x > 7:
    print("foo")
else x <= 7:
    print("bar")
```

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

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

```
a = 7
while a < 8:
    print("foo")
```

- a) affiche « foo » une infinité de fois
- b) affiche « foo » 8 fois
- c) affiche « foo » une fois
- d) n'affiche rien
- e) affiche « foo » 7 fois
- f) échoue
- g) affiche « foo » 6 fois

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

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

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

Question 6 (2 points) L'expression suivante ...

```
"foo" * 3
```

- a) a comme résultat la valeur de la variable `foo` multipliée par 3
- b) a comme résultat une chaîne de caractères
- c) échoue
- d) a comme résultat un entier
- e) n'a aucun résultat
- f) a comme résultat un booléen
- g) a comme résultat un float

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

```
if True:
    print("foo")
print("foo")
else:
    print("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » trois fois
- c) n'affiche rien
- d) échoue
- e) affiche « foo » deux fois
- f) affiche « foo » quatre fois
- g) affiche « foo » une infinité de 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) échoue
- b) affiche « foo » 9 fois
- c) affiche « foo » 27 fois
- d) affiche « foo » 10 fois
- e) affiche « foo » 33 fois
- f) affiche « foo » 100 fois
- g) affiche « foo » 10000 fois
- h) n'affiche rien
- i) affiche « foo » une infinité de fois
- j) affiche « foo » 30 fois
- k) affiche « foo » 3 fois
- l) affiche « foo » 20 fois
- m) affiche « foo » une fois
- n) affiche « foo » 1000 fois

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

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

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

Programmation Python

Contrôle des connaissances : contrôle surprise B1

IUT de Villetaneuse, Université Paris 13

Exemplaire numéro 49

Luca Saiu

Documents (livres, slides, photocopiés, 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 ...

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

- a) affiche « foo » une infinité de fois
- b) échoue
- c) affiche juste « bar », une 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 infinité de fois
- h) affiche « foo » une fois et « bar » une fois

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

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

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

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

```
a = 7
while a < 8:
    print ("foo")
```

- a) affiche « foo » une infinité de fois
- b) n'affiche rien
- c) affiche « foo » 6 fois
- d) affiche « foo » 8 fois
- e) affiche « foo » 7 fois
- f) échoue
- g) affiche « foo » une fois

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

```
x = 8
if x > 7:
    print ("foo")
else x <= 7:
    print ("bar")
```

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

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

```
if True:
    print ("foo")
print ("foo")
else:
    print ("foo")
```

- a) affiche « foo » une fois
- b) affiche « foo » une infinité de fois
- c) affiche « foo » trois fois
- d) échoue
- e) affiche « foo » deux fois
- f) n'affiche rien
- g) affiche « foo » quatre fois

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

```
for i in range (3):
    print ("foo")
for j in range (2):
    print ("foo")
```

- a) affiche « foo » 3 fois
- b) affiche « foo » 5 fois
- c) affiche « foo » 4 fois
- d) affiche « foo » une infinité de fois
- e) affiche « foo » 2 fois
- f) affiche « foo » une fois
- g) échoue
- h) n'affiche rien
- i) affiche « foo » 6 fois

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

```
a = 8
for b in range (4):
    a = a + b
print (a)
```

- a) affiche « 8 »
- b) affiche « a »
- c) échoue
- d) affiche « 11 »
- e) affiche « 10 »
- f) affiche « 13 »
- g) affiche « b »
- h) affiche « 14 »
- i) n'affiche rien
- j) affiche « 9 »
- k) boucle infiniment
- l) affiche « 12 »

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

```
"foo" * 3
```

- a) a comme résultat un entier
- b) échoue
- c) a comme résultat la valeur de la variable `foo` multipliée par 3
- d) a comme résultat un float-
- e) a comme résultat une chaîne de caractères
- f) n'a aucun résultat
- g) a comme résultat un booléen

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) n'affiche rien
- b) affiche « foo » 9 fois
- c) affiche « foo » 1000 fois
- d) affiche « foo » 10000 fois
- e) affiche « foo » 20 fois
- f) échoue
- g) affiche « foo » une infinité de fois
- h) affiche « foo » 3 fois
- i) affiche « foo » 30 fois
- j) affiche « foo » une fois
- k) affiche « foo » 10 fois
- l) affiche « foo » 27 fois
- m) affiche « foo » 100 fois
- n) affiche « foo » 33 fois

Solution (generated on : 17 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 : $d_2 c_2 b_2 e_2 f_2 d_3 d_3 b_2 h_2$

Paper 1 : $d_3 i_2 a_2 d_2 i_2 d_3 f_2 a_2 h_2$

Paper 2 : $b_2 g_2 e_3 a_3 g_2 k_2 e_2 g_2 g_2$

Paper 3 : $g_2 e_2 g_2 d_3 i_2 c_3 d_2 c_2 b_2$

Paper 4 : $f_2 c_3 d_2 b_2 e_2 n_2 d_3 b_2 i_2$

Paper 5 : $b_2 e_3 a_2 g_2 a_2 e_2 j_3 d_2 e_2$

Paper 6 : $a_2 f_2 l_2 c_2 c_3 g_2 e_3 f_2 i_2$

Paper 7 : $e_3 f_2 b_2 d_2 b_2 a_2 c_2 l_3 i_2$

Paper 8 : $e_3 g_2 h_2 g_2 f_2 g_2 a_2 c_2 e_3$

Paper 9 : $d_2 f_2 k_3 i_2 d_2 c_3 k_2 g_2 e_2$

Paper 10 : $b_2 g_2 g_2 h_2 h_3 f_2 g_2 g_2 b_3$

Paper 11 : $g_2 h_3 f_3 c_2 f_2 h_2 i_2 h_2 g_2$

Paper 12 : $a_2 f_2 g_2 e_2 f_2 c_3 h_3 g_2 g_2$

Paper 13 : $a_2 k_3 e_2 n_2 b_2 c_2 d_2 g_3 g_2$

Paper 14 : $e_3 a_2 f_2 n_2 e_2 g_3 a_2 h_2 a_2$

Paper 15 : $e_2 f_2 d_3 g_2 i_2 f_3 a_2 g_2 e_2$

Paper 16 : $e_2 a_2 i_3 c_2 e_2 d_3 g_2 g_2 h_2$

Paper 17 : $e_2 f_2 c_2 e_3 b_2 d_2 a_3 h_2 a_2$

Paper 18 : $e_2 e_2 b_2 h_2 h_3 c_3 f_2 c_2 f_2$

Paper 19 : $c_2 b_2 e_2 g_3 a_2 j_3 f_2 f_2 i_2$

Paper 20 : $f_3 g_2 d_2 g_2 g_2 b_2 i_3 e_2 k_2$

Paper 21 : $d_2 h_3 d_2 m_2 d_2 b_2 e_2 d_2 h_3$

Paper 22 : $h_2 i_3 b_2 c_2 e_2 f_2 g_2 a_2 h_3$

Paper 23 : $g_2 c_2 b_2 g_2 b_2 h_3 d_2 h_2 f_3$

Paper 24 : $g_2 i_3 n_2 b_2 b_3 a_2 b_2 g_2 g_2$

Paper 25 : $g_3 c_2 k_3 c_2 a_2 m_2 e_2 e_2 d_2$

Paper 26 : $f_2 b_2 c_2 i_3 d_2 f_2 c_2 e_3 a_2$

Paper 27 : $d_2 h_3 c_2 f_2 g_2 i_2 d_3 l_2 f_2$

Paper 28 : $c_2 m_2 g_2 d_3 i_2 e_2 c_2 h_3 d_2$

Paper 29 : $b_2 b_2 h_2 i_2 d_3 h_2 f_3 g_2 e_2$

Paper 30 : $g_2 f_2 g_2 l_3 e_2 d_2 g_3 d_2 d_2$

Paper 31 : $f_2 d_2 g_2 e_2 d_3 a_2 c_3 g_2 f_2$

Paper 32 : $e_3 f_2 b_2 e_2 c_2 c_2 b_3 i_2 c_2$

Paper 33 : $h_3 b_3 a_2 d_2 n_2 d_2 e_2 d_2 g_2$

Paper 34 : $e_2 c_2 g_2 g_2 c_2 d_3 c_2 c_3 m_2$

Paper 35 : $d_2 m_2 a_3 g_2 a_2 a_3 e_2 h_2 f_2$

Paper 36 : $b_2 d_2 b_2 f_3 a_2 e_2 g_2 h_3 f_2$

Paper 37 : $e_3 g_2 a_2 d_2 e_3 g_2 f_2 e_2 g_2$

Paper 38 : $f_2 a_2 f_3 b_2 c_2 g_3 f_2 b_2 g_2$

Paper 39 : $d_2 c_2 d_2 f_2 a_2 l_3 a_3 f_2 c_2$

Paper 40 : $f_2 d_2 b_2 b_2 b_3 c_2 i_2 g_3 f_2$

Paper 41 : $f_2 d_2 i_3 d_3 c_2 i_2 h_2 d_2 f_2$

Paper 42 : $a_2 g_2 g_2 f_3 c_2 n_2 a_2 d_3 h_2$

Paper 43 : $g_2 c_2 g_3 e_2 c_2 e_3 a_2 g_2 g_2$

Paper 44 : $d_2 d_3 b_2 e_2 c_3 g_2 f_2 d_2 f_2$

Paper 45 : $m_2 f_2 b_3 c_2 f_2 c_2 c_3 d_2 h_2$

Paper 46 : $g_2 c_3 g_2 g_2 f_2 f_2 d_2 f_2 a_3$

Paper 47 : $d_3 c_2 a_2 h_2 c_2 h_3 d_2 d_2 a_2$

Paper 48 : $b_3 h_2 h_2 a_2 b_3 b_2 d_2 n_2 g_2$

Paper 49 : $c_3 c_2 a_2 f_2 d_2 b_2 h_3 e_2 c_2$