

EPREUVE COMMUNE DE CONTRÔLE CONTINU

Série générale

EPREUVE BLANCHE

LYCEE PRIVE D'AVESNIERES

Enseignement de spécialité de NSI

Classe de première

Durée : 2 heures

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

Les documents, sous forme papier ou électronique, sont interdits.

Consignes :

L'épreuve consiste en un questionnaire à choix multiples divisé en 6 parties, une pour chaque thématique du programme vu en classe à ce jour. Chaque partie comporte 7 questions.

Pour chaque question, 4 réponses sont proposées dont une seule est correcte.

Notation : Pour chacune des 42 questions, le candidat gagne 3 points pour la réponse correcte, perd 1 point pour une réponse fausse, et obtient un résultat nul pour absence de réponse ou une réponse multiple. Sur chacune des six parties, si le total des points obtenu par le candidat est négatif, son résultat est évalué à 0. Le résultat obtenu est transformé en note sur 20 selon la formule : *nombre de points obtenus* $\times 20 / (3 \times 42)$.

Le nom et la classe du candidat ne devront pas apparaître sur sa copie.

Dès que le sujet vous est remis, assurez-vous qu'il est complet.

Le sujet comporte 8 pages, numérotées de « Page 1 sur 8 » à « Page 8 sur 8 ».

Écrivez ici votre numéro d'anonymat :

Partie 1 : Constructions élémentaires en Python.

1. Voici un programme Python :

```
a = 0
for i in range(4):
    a = a + i
```

Quelle est la valeur de la variable a après l'exécution de ce programme ?

- 0123
- 6
- 3
- 4

2. Quelle est la valeur affichée après l'exécution du programme Python suivant ?

```
x = 1
for i in range(5):
    x = x + i**2
print(x)
```

- 21
- 57
- 31
- 16

3. On donne le programme Python suivant :

```
def calcul(var):
    var = var * 2
    return(var)
```

```
n = 0
for i in range(5):
    n = n + i
    t = calcul(n)
```

Après exécution, la valeur de t est :

- 10
- 15
- 20
- 30

4. Le programme Python suivant ne calcule pas toujours correctement le résultat $x ** y$. Parmi les tests suivants, lequel va permettre de détecter l'erreur ?

```
def puissance(x, y):
    p = x
    for i in range(y-1):
        p = p * x
    return p
```

- puissance(2, 0)
- puissance(2, 1)
- puissance(2, 2)
- puissance(2, 10)

5. Quel est le résultat de l'évaluation de l'expression Python suivante :

```
[n**3 + 3*n for n in range(6)]
```

- [0, 6, 12, 18, 24, 30]
- [0, 4, 14, 36, 76, 140]
- [0, 1, 8, 27, 64, 125]
- [0, 3, 6, 9, 12, 15]

6. On construit la matrice mat ainsi :

```
mat = [3 * [0] for i in range(3)]
for i in range(3):
    mat[i][i] = i + 1
    mat[0][i] = mat[i][0] + i
    mat[i][-i] = mat[-i][i] + i
```

Que contient mat après l'exécution de ce programme ?

- [[1, 1, 0], [0, 2, 1], [0, 0, 0]]
- [[0, 0, 0], [1, 2, 0], [0, 1, 1]]
- [[1, 1, 2], [0, 2, 1], [0, 3, 3]]
- [[1, 0, 0], [0, 0, 0], [0, 0, 0]]

7. L'expression Python

`[n * n for n in range(10)]`

permet d'obtenir une liste par :

- Implémentation
- Compression
- Compréhension
- Assignment

Partie 2 : Type entier et booléen.

8. Si on tape l'instruction suivante :

`a, b = (5, 3)`

Quelle est la valeur de b ?

- 3,5
- 5
- 3
- 2

9. Si A et B sont des variables booléennes, quelle expression booléenne équivaut à :

`not A or B` ?

- `(A and B) or (not A or B) or (not A or not B)`
- `(A and B) or (not A or B)`
- `(not A or B) or (not A or not B)`
- `not(A and not B)`

10. Si on rencontre le nombre 3F5, dans quel système de numération est-on ?

- binaire
- octal
- décimal
- hexadécimal

11. Combien faut-il de bits pour représenter le nombre qui s'écrit 130 en décimal ?

- 4
- 5
- 8
- 16

12. Dans l'algorithme ci-dessous, qui prend en entrée un entier naturel non nul écrit en décimal et renvoie son écriture binaire, remplacer les pointillés par l'opérateur qui convient.

```
def cascade(n):  
    chiffres = "  
    while n != 0:  
        chiffres = str(n ... 2) + chiffres  
        n = n // 2  
    return chiffres
```

- `//`
- `+`
- `*`
- `%`

13. Quelle est la valeur binaire de l'entier décimal 95 ?

- 1111101
- 1011100
- 1001101
- 1011111

14. Combien d'entiers positifs ou nuls différents peut-on représenter en machine sur 32 bits ?

- $2^{32} - 1$
- 2^{32}
- 2×32
- 2^{32}

Partie 3 : Variables de type construit.

15. m et p sont des listes. Voici programme Python :

```
m = [2, 3, 8, 9]  
p = m  
p[2] = 5
```

Que contient m après l'exécution du programme ?

- [2, 5, 8, 9]
- [2, 3, 5, 8, 9]
- [2, 3, 5, 9]
- [2, 3, 8, 5]

16. La chaîne de caractères `ch` a pour valeur `'123456789'`.

Quelle est la valeur de l'expression :

`int(ch[7] + ch[8])` ?

- 17
- Il y a une erreur
- '78'
- 89

17. On affecte une valeur à la variable `Lieux` de la manière suivante :

`Lieux = ('ville', 'campagne', 'Forêt')`

`Lieux` est :

- une liste
- un tuple
- une chaîne de caractères
- mal écrit

18. Soit `t` et `u` deux `n`-uplets tels que :

`t = (1, 2, 3)`

`u = (8, 5)`

`v = t + 2*u`

Que contient `v` ?

- (1, 2, 3, 16, 10)
- ((1, 2, 3), (8, 5), (8, 5))
- 1, 2, 3, 2, 8, 5
- (1, 2, 3, 8, 5, 8, 5)

19. Soit `p` la liste telle que :

`p = ['ab', 'adc', 'bdc', 'ba', 'abc', 'bac']`

Je tape l'instruction : `p.sort()`

Quelle est la valeur de `p` ?

- ['ab', 'adc', 'abc', 'ba', 'bdc', 'bac']
- ['abc', 'ab', 'adc', 'bac', 'ba', 'bdc']
- ['ab', 'abc', 'adc', 'ba', 'bac', 'bdc']
- ['adc', 'abc', 'ab', 'bdc', 'bac', 'ba']

20. On définit un tableau `t` en langage Python.

Ce tableau est une liste de listes, toutes les sous-listes ayant le même nombre d'éléments tous égaux 0.

`t = [[0, 0, ..., 0], [0, 0, ..., 0], ..., [0, 0, ..., 0]]`

On appelle `h` le nombre de listes contenues dans `t` et `l` le nombre d'éléments appartenant à chacune de ces listes.

Parmi les propositions suivantes, laquelle permet de calculer `h` et `l` ?

- `h, l = len(t[0]), len(t)`
- `h, l = len(t[0]), len(t[1])`
- `h, l = len(t), len(t[0])`
- `h, l = len(t[1]), len(t[0])`

21. Avec la variable `voyelle` définie par l'affectation suivante :

`voyelle = ['A', 'E', 'I', 'O', 'U', 'Y']`

Quelle expression permet d'afficher 'OUI' ?

- `int(voyelle[3] + voyelle[4] + voyelle[2])`
- `voyelle[3] + voyelle[4] + voyelle[2]`
- `chr(voyelle[3] + voyelle[4] + voyelle[2])`
- `bin(voyelle[3] + voyelle[4] + voyelle[2])`

Partie 4 : Machines et systèmes d'exploitation.

22. Qui est considéré comme le premier inventeur de la machine à calculer ?

- John Von Neumann
- Edgar Casio
- Blaise Pascal
- Léonard de Vinci

23. Qui a imaginé une machine à calculer utilisant des cartes perforées ?

- Blaise Pascal
- Charles Babbage
- John Von Neumann
- Joseph Marie Jacquard

24. Quel est le nom, en 1946, d'un des premiers ordinateurs ?

- L'APPLE I
- l'Enigma
- l'IBM PC
- L'Eniac

25. Quel élément ne fait pas partie d'une machine décrite par Von Neumann ?

- Une souris
- Une unité de contrôle
- Une horloge
- Un port HDMI

26. Voici un programme écrit dans un langage compréhensible par la machine

```

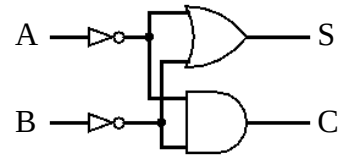
0   LOAD_CONST      0 (1)
3   STORE_NAME     0 (x)
6   LOAD_CONST      1 (2)
9   STORE_NAME     1 (y)
12  LOAD_NAME      0 (x)
15  LOAD_NAME      1 (y)
18  BINARY_ADD
21  STORE_NAME     2 (z)
24  LOAD_CONST      2 (None)
27  RETURN_VALUE

```

Que contiennent respectivement x, y et z ?

- 1 ; 2 ; 2
- 1 ; 3 ; 2
- 1 ; 2 ; 1
- 1 ; 2 ; 3

27. Quelle est la table de vérité de ce circuit ?



A	B	S	C
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

A	B	S	C
0	0	1	1
0	1	1	0
1	0	1	0
1	1	0	0

A	B	S	C
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

28. Parmi les affirmations suivantes, laquelle est vraie ?

- La mémoire RAM est une mémoire accessible en lecture seulement
- La mémoire RAM est une mémoire accessible en écriture seulement
- La mémoire RAM est une mémoire accessible en lecture et en écriture
- La mémoire RAM permet de stocker des données après extinction de la machine

Partie 5 : Entiers relatifs, réels et caractères.

29. Voici deux boucles :

Boucle 1 :

```
for i in range(50000) :
```

```
    a = 1,001 ** i
```

Boucle 2 :

```
for i in range(50000) :
```

```
    a = 2 ** i
```

- La boucle 1 s'exécute plus rapidement que la boucle 2.
- La boucle 2 s'exécute plus rapidement que la boucle 1.
- Les deux ont à peu près la même durée.
- On ne peut pas savoir à l'avance.

30. Si on travaille avec des entiers signés codés sur 8 bits, alors l'addition $53 + 85$ affiche

- 138
- 129
- 126
- 118

31. Soit le nombre x qui a pour écriture binaire $110,11011$ Quelle est sa mantisse ?

- 110
- 1,1011011
- 1011011
- 10

32. Pour représenter la chaîne de caractères 'Payé 2 €' on peut utiliser la table de caractères

- Latin-1
- ASCII
- ISO8859-1
- UTF-8

33. Voici un programme écrit en Python :

```
# Ecriture de données dans un fichier.
f_temporaire = open('fichier.txt', 'w')
f_temporaire.write(str(1.1) + "\t" + str(2.2) + "\n")
f_temporaire.write(str(3.3) + "\t" + str(4.4) + "\n")
f_temporaire.close()
```

```
# Extraction des données dans une liste.
f_temporaire = open('fichier.txt', 'r')
Liste = [ligne.rstrip().split("\t") for ligne in
f_temporaire]
f_temporaire.close()
```

Après exécution, la variable Liste vaut :

- ['3.3', '4.4']
- [['1.1', '2.2'], ['3.3', '4.4']]
- [['1.1', '3.3'], ['2.2', '4.4']]
- [3.3, 4.4]

34. Si on travaille avec des entiers signés sur 1 octet alors le codage de -12 est :

- 1000 1100
- 1111 0011
- 1111 0100
- 1111 1100

35. Quelle opération donne comme résultat un nombre de type int ?

- $4 / 2$
- $2.5 * 2$
- $13 / 8$
- $16 // 8$

36. Voici un programme écrit en Python :

```
f_temporaire = open('fichier.txt', 'w')
f_temporaire.write('Payé ')
f_temporaire.close()
f_temporaire = open('fichier.txt', 'a')
f_temporaire.write('2 €')
f_temporaire.close()
```

Que contient fichier.txt après l'exécution de ce programme ?

- Payé
- Payé 2 €
- 2 €
- Payé
2 €

Partie 6 : Algorithmes fondamentaux.

37. On considère le programme suivant :

```
def maxi(tab):
    """
    tab est une liste de couples (nom, note)
    où nom est de type str
    et où note est un entier entre 0 et 20.
    """

    m = tab[0]
    for x in tab:
        if x[1] >= m[1]:
            m = x
    return m
```

L = [('Adrien', 17), ('Barnabé', 17), ('Casimir', 17), ('Dorian', 17), ('Emilien', 16), ('Fabien', 16)]

Quelle est la valeur de maxi(L) ?

- ('Adrien',17)
- ('Dorian',17)
- ('Fabien',16)
- ('Emilien',16)

38. a et m étant des entiers strictement positifs la fonction suivante calcule $a ** m$:

```
def puissance(a, m):
    p = 1
    n = 0
    while n < m:
        p = p * a
        #...
        n = n + 1
    return p
```

On a ajouté un commentaire pour indiquer ce que contient p après l'exécution de l'instruction

```
p = p * a
```

À la ligne marquée d'un #, on peut, à la place des pointillés, écrire le commentaire :

- p = a ** n
- p = a ** (n-1)
- p = a ** (n+1)
- p = a ** m

39. Voici une fonction Python de recherche d'un maximum dans une liste t :

```
def maxi(t):
    m = -1
    for k in range(len(t)):
        if t[k] > m:
            m = t[k]
    return m
```

Pour quelle situation la valeur du maximum renvoyée n'est-elle pas bonne ?

- Tout élément de t est un entier positif ou nul
- Tout élément de t est un entier inférieur ou égal à -2
- Tout élément de t est un entier supérieur ou égal à -1
- Tout élément de t est un entier supérieur à ou égal à 1

40. La fonction suivante calcule la racine carrée du double d'un nombre flottant.

```
from math import sqrt
```

```
def racine_du_double(x):  
    return sqrt(2 * x)
```

Quelle est la précondition sur l'argument de cette fonction ?

- $x \geq 0.0$
- $x > 0.0$
- $x < 0.0$
- $\text{sqrt}(x) \geq 0.0$

41. La fonction suivante doit calculer la moyenne d'un tableau de nombres, passé en paramètre.

Avec quelles expressions faut-il compléter l'écriture pour que la fonction soit correcte ?

```
def moyenne(tableau):  
    total = ...  
    for valeur in tableau:  
        total = total + valeur  
    return total / ...
```

- 0 et `len(tableau)`
- 0 et `len(tableau) + 1`
- 1 et `len(tableau)`
- 1 et `len(tableau) + 1`

42. Combien d'échanges effectue la fonction Python suivante pour trier une liste de 10 éléments au pire des cas ?

```
def tri(liste):
```

```
    """
```

Cette fonction trie la liste par ordre croissant.

```
    """
```

```
    for i in range(1, len(liste)):  
        for j in range(len(liste) - i):  
            if liste[j] > liste[j+1]:  
                # Echange si l'élément est supérieur au  
                # suivant  
                liste[j], liste[j+1] = liste[j+1], liste[j]
```

- 45
- 100
- 10
- 55