



Enseignants: G. Maatouk, L. Testa
Informatique et Calcul Scientifique (ICS) - CMS
8 novembre 2023
Durée : 105 minutes

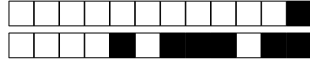
Dalton Joe

SCIPER: 987654

Attendez le début de l'épreuve avant de tourner la page. Ce document est imprimé recto-verso, il contient 13 questions sur 16 pages, les dernières pouvant être vides. Ne pas dégrafer. L'examen comporte un total de 47 points.

- Posez votre **carte d'étudiant.e** sur la table.
- L'utilisation d'une **calculatrice** et de tout **outil électronique** est **interdite** pendant l'épreuve.
- Pour les questions à **choix unique**, on comptera :
 - les points indiqués si la réponse est correcte,
 - 0 point si il n'y a aucune ou plus d'une réponse inscrite,
 - 0 point si la réponse est incorrecte.
- Les algorithmes demandés sont à écrire sous forme de fonctions Python. Mettez votre code en forme en respectant les indentations: 1 carreau = 1 espace (donc 4 carreaux = 1 tab).
- Vous n'avez pas besoin de commenter votre code mais vous pouvez le faire si vous pensez que cela aide à sa compréhension.
- Utilisez un **stylo** à encre **noire ou bleu foncé** et effacez proprement avec du **correcteur blanc** si nécessaire.
- Répondez dans l'espace prévu (**aucune** feuille supplémentaire ne sera fournie).
- Les brouillons ne sont pas à rendre: ils ne seront pas corrigés.

Respectez les consignes suivantes Observe this guidelines Beachten Sie bitte die unten stehenden Richtlinien		
choisir une réponse select an answer Antwort auswählen	ne PAS choisir une réponse NOT select an answer NICHT Antwort auswählen	Corriger une réponse Correct an answer Antwort korrigieren
  		 
ce qu'il ne faut PAS faire what should NOT be done was man NICHT tun sollte		
     		



Première partie, questions à choix unique

Pour chaque énoncé proposé, une ou plusieurs questions sont posées. Pour chaque question, marquer la case correspondante à la réponse correcte sans faire de ratures. Il n'y a qu'une seule réponse correcte par question.

Qu'affiche chacun des codes ci-dessous?

Question 1 (2 points)

```
def mystere(*args):
    d = {}
    j = 0
    for i in range(len(args)-1, -1, -1):
        d[j] = args[i]
        j += 1
    return d

print(mystere(3, 5, 7, 9, 11))
```

☐ {0:11, 1:9, 2:7, 3:5, 4:3}☐ {11:0, 9:1, 7:2, 5:3, 3:4}☐ {11:3, 9:5, 7:7, 5:9, 3:11}☐ {3:3, 5:5, 7:7, 9:9, 11:11}

Question 2 (2 points)

```
try:
    new = []
    new.append(1+"2")
    a = new[1]/0
    a = 1
    print(a)
except ZeroDivisionError :
    print("2")
except IndexError:
    print("3")
except TypeError:
    print("4")
```

☐ 1☐ 2☐ 4☐ 3

Question 3 (2 points)

```
L1 = [0, 1, 2]
L2 = [5, 5, 5]

L3 = [2*x+y for x in L1 for y in L2]

print(L3)
```

☐ [5, 7, 9]☐ [5, 5, 5, 7, 7, 7, 9, 9, 9]☐ [5, 7, 9, 5, 7, 9, 5, 7, 9]☐ [9, 11, 13]☐ [9, 11, 13, 9, 11, 13, 9, 11, 13]☐ [9, 9, 9, 11, 11, 11, 13, 13, 13]

**Question 4** (2 points)

```
def affichages(d):
    for x in d:
        print(d[x], end = " ")
    print()
    for x in d.items():
        print(x[1], end = " ")
    print()
    for x in d.values():
        for y in d:
            if d[y] == x:
                print(y, end = " ")

d1 = {"a":10, "b":20, "c":30}
affichages(d1)
```

☐ a b c
a b c
10 20 30

☐ a b c
10 20 30
a b c

☐ 10 20 30
10 20 30
10 20 30

☐ 10 20 30
a b c
10 20 30

☐ a b c
a b c
a b c

☐ 10 20 30
10 20 30
a b c

Question 5 (2 points)

```
def surprise(t):
    s = 0
    for x in t:
        s += x[1]
    return s

t1 = ((1, 2), (3, 4), (5, 6), (7, 8))
print(surprise(t1))
```

☐ 36
☐ 4

☐ 16
☐ 20

Question 6 (2 points)

```
i = 2
while i < 10:
    if i%2 == 0:
        print(i,end=' ')
    if i == 6:
        break
```

☐ 2 3 4 5 6 7 8 9
☐ 2 4 6

☐ 2 4 6 8
☐ Aucune de ces réponses

**Question 7** (2 points)

```
def oui_non(L, L1 = [1, 3]):  
    for x in L:  
        for y in L1:  
            if x < y:  
                print("oui", end = " ")  
            else:  
                print("non", end = " ")  
  
oui_non([2, 4])
```

☐ non oui non non☐ non non☐ oui oui☐ oui non oui oui**Question 8** (2 point)

```
L = ["a", "b", "a", "a", "b"]  
for i in range(1, 4):  
    L[i] += L[i+1]  
for s in ("ab", "ba"):  
    if s in L:  
        print(s, L.count(s))
```

☐ ab 1
ba 1☐ L'affichage produit une exception IndexError☐ ab 2
ba 1☐ ab 2
ba 2**Question 9** (2 points)

```
L = [[0], [0]]  
L_prime = L.copy()  
L.append([1])  
L[1].append(1)  
print(L, L_prime)
```

☐ [[0], [0, 1], [1]] [[0], [0, 1], [1]]☐ [[0], [0, 1], [1]] [[0], [0, 1]]☐ [[0], [0, 1], [1]] [[0], [0]]☐ [[0], [0, 1], [1]] [[0], [0], [1]]**Question 10** (2 points)

```
for i in range(4):  
    for j in range(i):  
        print('*', sep='|', end = '.')
```

☐

```
*|*|*|.
*|*|.
*|.
```

☐

```
*.
*.*.
*.*.*.
```

☐

```
*.*.*.
*.*.
*.
```

☐

```
*|.
*|*|.
*|*|*|.
```



Deuxième partie, questions de type ouvert

Répondre dans l'espace dédié. Laisser libres les cases à cocher : elles sont réservées au correcteur.

Question 11: *Cette question est notée sur 8 points.*

☐	.5	☐	.5	☐	.5	☐	.5	☐	.5	☐	.5	☐	.5	☐	.5	☐	.5	☐	.5
☐ 0	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5	☐ 6	☐ 7	☐ 8											

Dans cet exercice, on va écrire un entier strictement positif quelconque N sous la forme $N = 2^p + r$, avec p le plus grand entier possible et $0 \leq r < 2^p$. Pour cela, on procède en plusieurs étapes.

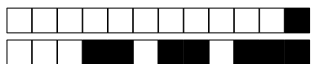
Note : Cet exercice comprend deux sous-questions, chacune peut être résolue indépendamment.

- (a) Ecrivez une fonction `decomposition` qui prend comme paramètre un nombre entier strictement positif N (pas besoin de tester cette condition dans le corps de la fonction), et qui divise ce nombre par deux tant que le résultat de cette opération n'est pas strictement inférieur à 1. Cette fonction doit retourner

- (i) le nombre p de divisions effectuées
- (ii) le reste r de cette opération tel que $N = 2^p + r$.

Par exemple, l'instruction `decomposition(4)` doit retourner $(2,0)$ car $4 = 2*2+0$, `decomposition(7)` doit retourner $(2,3)$ car $7 = 2*2+3$ et `decomposition(9)` doit retourner $(3,1)$ car $9 = 2*2*2+1$.





+1/6/55+





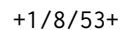
(b) Ecrivez un programme qui

- (i) génère un nombre aléatoire compris entre 1 et 100
- (ii) appelle la fonction définie au point (a)
- (iii) affiche le résultat à l'écran sous la forme : $N = 2**p+r$.

Par exemple, $16 = 2**4+0$ et $15 = 2**3+7$.

Indication: On rappelle que la fonction `randint(a,b)` du module `random` de la bibliothèque standard de Python prend comme paramètres deux nombres `a` et `b` et retourne un nombre aléatoire appartenant à l'intervalle (a,b) .





A horizontal number line is shown with tick marks at each integer from 0 to 6. Below each tick mark is a box containing the integer. Above each integer box is a smaller box containing a decimal point followed by a blank space (e.g., ".5").

Integer	Tenths
0	.5
1	.5
2	.5
3	.5
4	.5
5	.5
6	.5

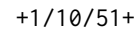
- Les clés de `d` sont les éléments de `L`
- La valeur associée à une clé `x` est **la liste des indices** où `x` apparaît dans la liste `L`.

- Pour la liste [10, 20, 10, 30] en entrée, `list_to_dict` doit retourner {10:[0, 2], 20:[1], 30:[3]}
- Pour une liste vide en entrée, `list_to_dict` doit retourner un dictionnaire vide.



+1/9/52+





A number line from 0 to 13. Above the line, boxes are filled with 0.5 for intervals 0-1, 1-2, 2-3, 3-4, 4-5, 5-6, 6-7, 7-8, 8-9, 9-10, 10-11, 11-12, and 12-13.

```

graph TD
    Feuille -- bat --> Caillou
    Caillou -- bat --> Ciseaux
    Ciseaux -- bat --> Feuille
  
```

	0	1	2
0	Egalité	0	2
1	0	Egalité	1
2	2	1	Egalité

0
1
1
0
2



- (b) Créez une fonction `testinput()` qui demande à l'utilisatrice (Alice) d'entrer un nombre entier compris entre 0 et 2 inclus. Tant que la valeur entrée ne satisfait pas ces conditions, votre fonction doit continuer à demander un nombre à Alice. Elle doit finalement retourner cette valeur.
- Si Alice rentre 4, 2.1 ou "deux", le programme doit afficher : `On avait dit 0, 1 ou 2 !` et lui redemander d'entrer un nombre.





- (c) Ecrivez un programme qui, pour chaque coup joué par Bob, demande à Alice d'entrer un nombre compris entre 0 et 2, compare ce nombre avec celui joué par Bob, et stocke le nom du vainqueur (Alice, Bob ou Egalité) dans une liste `resultat`. Si Alice a joué `[0, 0, 0, 0, 0]`, alors `resultat` doit contenir `["Egalite", "Alice", "Alice", "Egalite", "Bob"]`.

Vous pouvez utiliser directement la variable `ami` ainsi que la fonction `testinput()`.





-

●

-

●

