

Ens. : O. Lévêque, M. Stojilović

Information, Calcul, Communication - GC

Vendredi 7 novembre 2025

Durée: 180 minutes

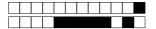
1

Salle: **CM 1 2**

Attendez le début de l'épreuve avant de tourner la page. Ce document est imprimé recto-verso, il contient 16 pages, les dernières pouvant être vides. Ne pas dégrafer.

- Posez votre carte d'étudiant sur la table.
- Document autorisé pour cet examen : un formulaire constitué d'une page A4 rectoverso, manuscrite, préparée avec stylet+tablette ou à l'ordinateur.
- Seule l'utilisation d'une calculatrice simple (de type TI-30 eco RS) est autorisée pendant l'examen. Tout autre appareil électronique (ordinateur, smartphone/watch, tablette) est interdit.
- L'examen est composé uniquement de questions de type ouvert, valant en tout 40 points.
- Concernant l'écriture d'un algorithme, vous avez la possibilité d'utiliser des algorithmes déjà vus en cours sans devoir réécrire ceux-ci, mais il importe de bien spécifier les données d'entrée et de sortie.
- Merci d'avance de soigner la présentation de vos réponses !
- Laissez libres les cases à cocher : elles sont réservées aux correcteurs.
- Si une question est erronée, les enseignants se réservent le droit de l'annuler.



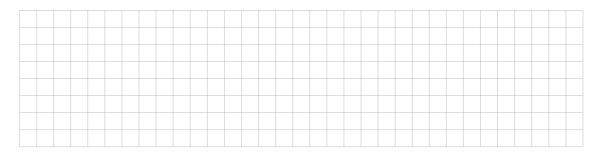


Question 1: Cette question est notée sur 3 points.

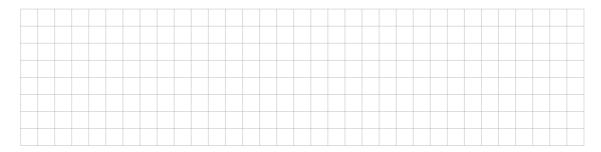
	0	1	2	3

On considère l'algorithme suivant:

a) (1 point) Quelle est la sortie de cet algorithme si n=8 en entrée ?



b) (1 point) Quelle est la sortie de cet algorithme si le nombre n en entrée est une puissance de 2 ?



c) (1 point) Quelle est la complexité temporelle de cet algorithme ? (utiliser la notation $\Theta(\cdot)$)





Question 2: Cette question est notée sur 3 points.

```
0 1 2 3
```

a) (1 point) On considère l'algorithme suivant:

```
algo1 entrée : a,b nombres entiers positifs sortie : ???

Si a \geq b Sortir: a Sortir: a
```

En général, quelle est la sortie de cet algorithme pour deux nombres entiers positifs a,b en entrée ?



b) (1 point) On considère l'algorithme suivant:

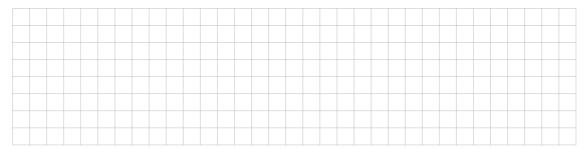
```
algo2
entrée : a, b deux nombres entiers positifs
sortie : ???

Si a = b
Sortir: a

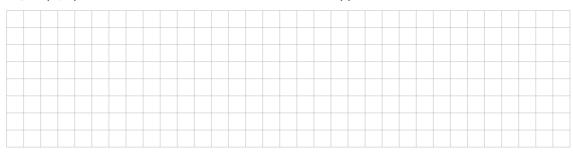
Si a > b
Sortir: algo2(a - b, b)

Sinon
Sortir: algo2(a, b - a)
```

En général, quelle est la sortie de cet algorithme pour deux nombres entiers positifs a,b en entrée ?



c) (1 point) Si $n\geq 1$, $a=2^n$ et $b=2^{n-1}$, combien d'opérations seront effectuées par algo2(a,b) avant de terminer ? (utiliser la notation $\Theta(\cdot)$)





Question 3: Cette question est notée sur 3 points.

On considère le problème suivant :

Etant donné une liste L de n nombres entiers relatifs, trouver le plus grand sous-ensemble $S\subset\{1,\dots,n\}$ tel que

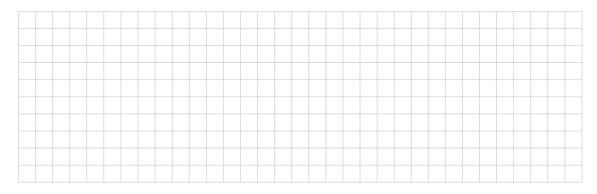
$$\prod_{j \in S} L(j) > 0$$

(Note : Par "le plus grand sous-ensemble S", on entend ici le sous-ensemble S avec le plus d'éléments.)

a) (1 point) Si L=(-4,+12,-3,-5,+7,+8) et n=6, que vaut S ? (notez qu'il existe plusieurs bonnes réponses à cette question)

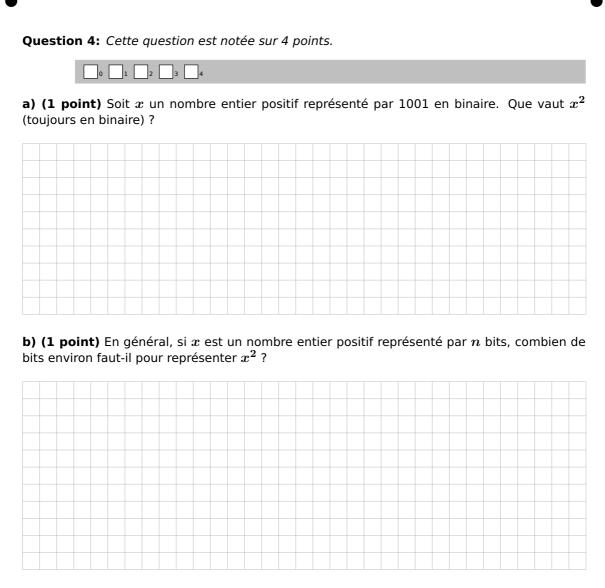


b) (1 point) Sait-on si le problème ci-dessus fait partie de la classe P? Justifiez votre réponse.



c) (1 point) Sait-on si le problème ci-dessus fait partie de la classe NP? Justifiez votre réponse.





c) (2 points) Deux joueurs jouent de façon répétée à un jeu de pile ou face. Chaque lancer de pièce gagné par un joueur lui rapporte un point. Les deux joueurs jouent jusqu'à ce que l'un des deux atteigne en premier un score de 7 points. Combien de bits *au minimum* faut-il pour représenter tous les scores finaux possibles ? (par exemple, 7-3 est un score final) Justifiez votre réponse.



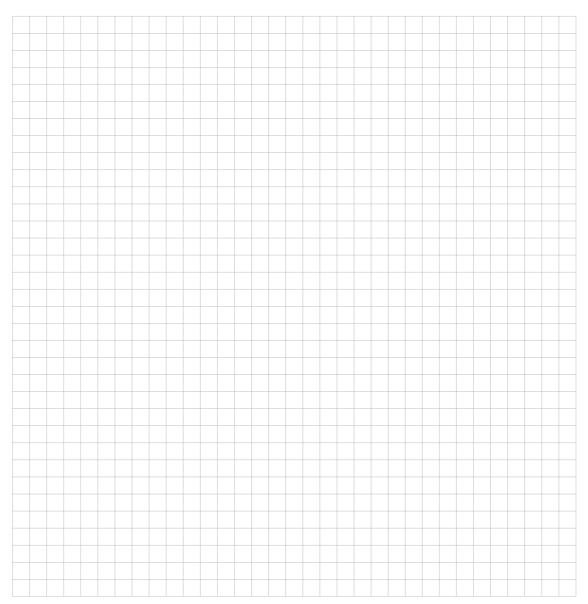
Question 5: Cette question est notée sur 6 points.

0 1 2 3 4 5 6

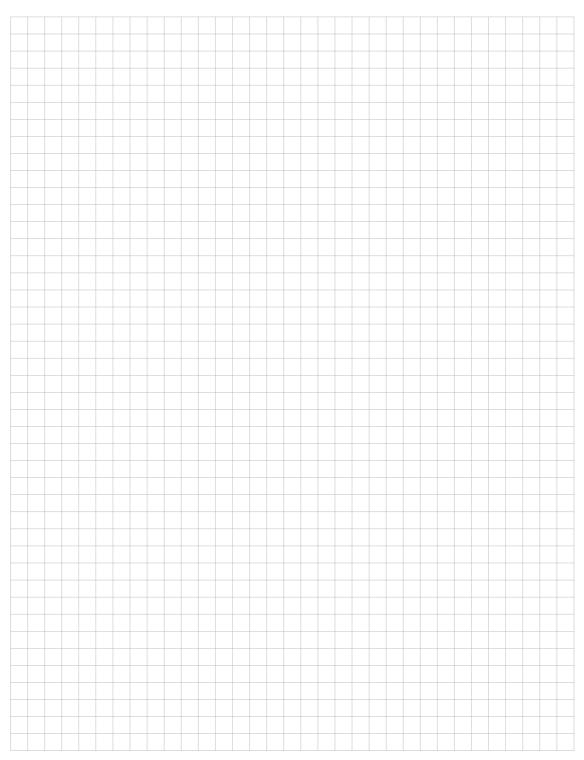
Soit L une liste de n nombres entiers positifs, tous différents les uns des autres. On définit le rang d'un nombre appartenant à cette liste comme étant le nombre entier k compris entre 1 et n tel qu'exactement k-1 nombres de la liste sont plus grands que le nombre en question.

Ainsi, par exemple, dans la liste L=(13,22,12,37,19,8,56,17), le rang du nombre 56 est égal à 1, tandis que celui du nombre 19 est égal à 4.

a) (3 points) Ecrire un algorithme de complexité temporelle $\Theta(n)$ dont les données d'entrée soient la liste L, sa taille n, ainsi qu'un autre nombre entier positif x, et dont la sortie soit le rang de x dans la liste L si x appartient à L, ou 0 si x n'appartient pas à L.

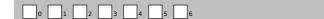


b) (3 points) $De\ plus$, supposons maintenant que la liste L soit triée dans l'ordre $d\'{e}croissant$. Existe-t-il un algorithme dont la sortie soit identique à celle de l'algorithme précédent et dont la complexité temporelle soit moindre (en notation $\Theta(\cdot)$) ? Si oui, écrire un tel algorithme et donner sa complexité temporelle (toujours en notation $\Theta(\cdot)$) ; si non, expliquer pourquoi un tel algorithme n'existe pas.



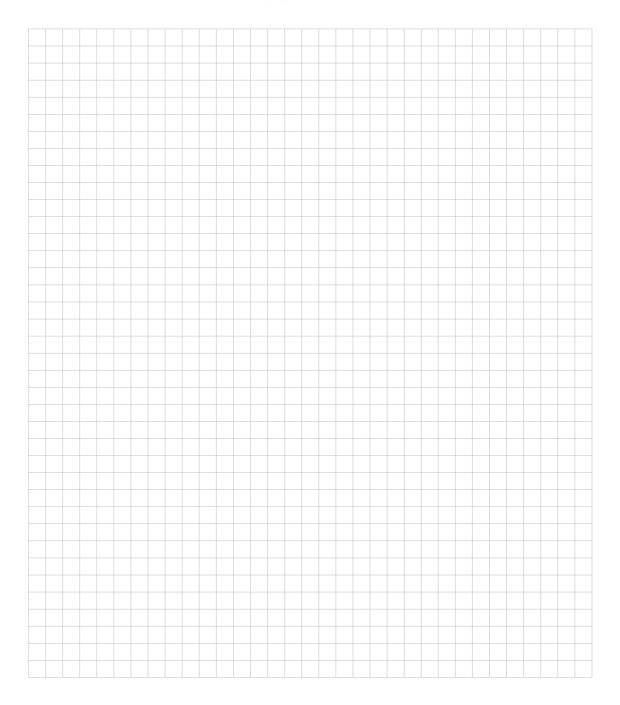


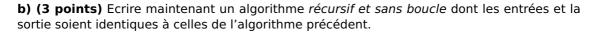
Question 6: Cette question est notée sur 6 points.



a) (3 points) Soit L une liste de n nombres entiers relatifs. Ecrire un algorithme non-récursif de complexité temporelle $\Theta(n)$ qui prenne en entrée la liste L et sa taille n, et dont la sortie soit oui si et seulement s'il existe un nombre $k \in \{1,\ldots,n\}$ tel que

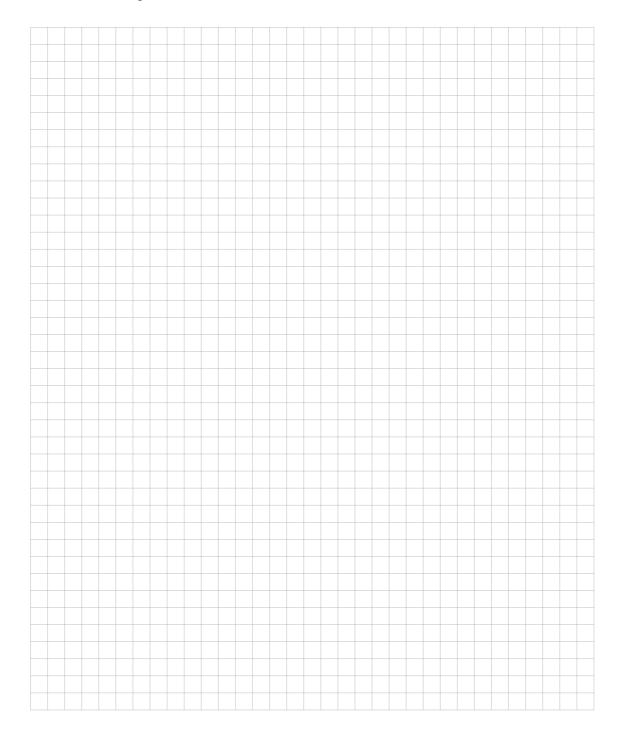
$$\sum_{i=k}^n L(i) \geq 0$$





Note : Merci d'utiliser les notations suivantes pour la manipulation de listes :

- + $L(1:m) = (L(1),L(2),\dots,L(m))$ désigne la sous-liste composée des m premiers éléments de la liste L
- $L \oplus M$ désigne la concaténation des deux listes L et M





Question 7: Cette question est notée sur 2 points.



Qu'affiche ce programme ? L'ordre de priorité des opérations (du plus fort au plus faible) est : exposant, multiplication/division/modulo, addition/soustraction, opérateurs relationnels, puis opérateurs logiques (où or a la priorité la plus faible).

```
x = 12

y = 15

z = x - y

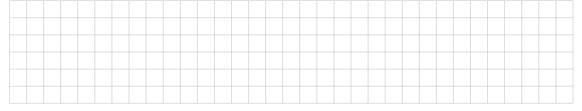
a = y < x < z or z + x == y

b = 3 + z or x

c = b == (x and 3 + z) or False

print(a, b, c)
```

Fournissez uniquement la réponse demandée ; aucune étape intermédiaire n'est nécessaire.



Question 8: Cette question est notée sur 2 points.

```
0 1 2
```

Qu'affiche ce programme?

```
x = [-11, -6, 5, 10, -3]

s = 0

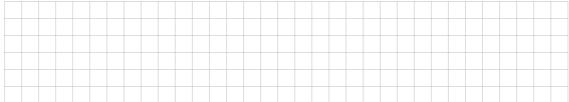
for n in x:

m = n % 4

d = n // 4
```

s += 1 if (m == d or m == -d) else -1

print(s)





Question 9: Cette question est notée sur 2 points.

```
0 1 2
```

Qu'affiche ce programme ? (sum(x) renvoi la somme des éléments de la liste x.)

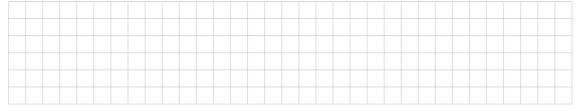
```
my_list1 = [5, True, False, 7, True]
my_list2 = [0, True, False, 0, True]
s = sum(my_list2)

for i. x in enumerate(my_list1):
```

```
for i, x in enumerate(my_list1):
    if i == len(my_list1) - 1:
        break
s = s - x
```

print(s)

Fournissez uniquement la réponse demandée ; aucune étape intermédiaire n'est nécessaire.



Question 10: Cette question est notée sur 2 points.



Qu'affiche ce programme?

```
N = 4

k = 2

cnt = 0

while k < N:

x = 2**k

while x > 0:

cnt += x

x = x - 1

k += 1
```

print(cnt)





Qu'affiche ce programme ?

words = ["morning", "is", "EPFL", "cool", "ICC", "Friday", "boring"]

w1 = words[:5]
w2 = words[-1::-1]

print(w2[0] + "," + w1[len(w1)-1] + "," + words[0] + "," + w1[3])

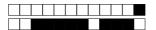
has_boring_in_w2 = "boring" in w2
has_cool_in_w1 = "cool" in w1
has_cool_in_w2 = "cool" in w2
w1_longer = len(w1) > len(w2)

if has_boring_in_w2 == False:
 print("Wait_a_isec?")
elif has_cool_in_w1 and not has_cool_in_w2 and w1_longer:
 print("Words1_iare_icool")
elif has_cool_in_w2 and not has_cool_in_w1:
 print("cool_iriday")
elif has_cool in w1 and has_cool in w2:

print("The_End")

print("cool_everywhere")





Question 12: Cette question est notée sur 2 points.

```
Qu'affiche ce programme ?

n = 10

my_list = []

for a, b in enumerate(range(n, 2*n, 2)):

my_list.append([x for x in range(a, b, 3)])

s = [t[len(t)//2] for t in my_list]

print(s)

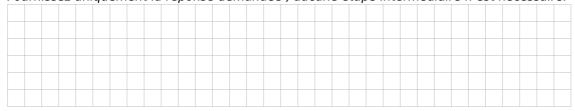
Fournissez uniquement la réponse demandée ; aucune étape intermédiaire n'est nécessaire.
```

Question 13: Cette question est notée sur 3 points.



Quels éléments doivent figurer dans la liste x pour que le programme ci-dessous affiche la liste s = [0, 1, 1, 3, 1, 0, 2]? Plusieurs solutions sont possibles, n'écrivez qu'une seule solution valide. (max(x) renvoie le plus grand élément de la liste x.)

```
s = [0 for r in range(max(x)+1)]
for r in range(len(s)):
    s[r] = x.count(r)
print(s)
```





Vous pouvez utiliser cette page au cas où vous n'auriez pas assez de place pour répondre à une des questions précédentes.

Pour faciliter la correction, merci de bien indiquer à la question donnée qu'il faut venir lire la suite ici!

