



EPFL

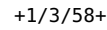
Ens. : O. Lévêque, M. Stojilović
Information, Calcul, Communication - GC
Vendredi 7 novembre 2025
Durée : 180 minutes

1

Salle : **CM 1 2**

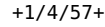
Attendez le début de l'épreuve avant de tourner la page. Ce document est imprimé recto-verso, il contient 16 pages, les dernières pouvant être vides. Ne pas dégrafer.

- Posez votre carte d'étudiant sur la table.
- Document autorisé pour cet examen : un formulaire constitué d'une page A4 recto-verso, 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.



0 1 2 3

algo
entrée : <i>nombre entier positif n</i>
sortie : ???
<pre> s ← n t ← 0 Pour i allant de 1 à n Si s > 1 s ← s/2 t ← t + 1 Sortir: t </pre>

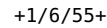


0 1 2 3

algo1
entrée : a, b nombres entiers positifs
sortie : ???
Si $a \geq b$ Sortir: a Sortir: $\text{algo1}(b, a)$

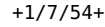
algo2
entrée : a, b deux nombres entiers positifs sortie : ???
Si $a = b$ Sortir: a Si $a > b$ Sortir: $\text{algo2}(a - b, b)$ Sinon Sortir: $\text{algo2}(a, b - a)$





0 1 2 3

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



0 1 2 3 4



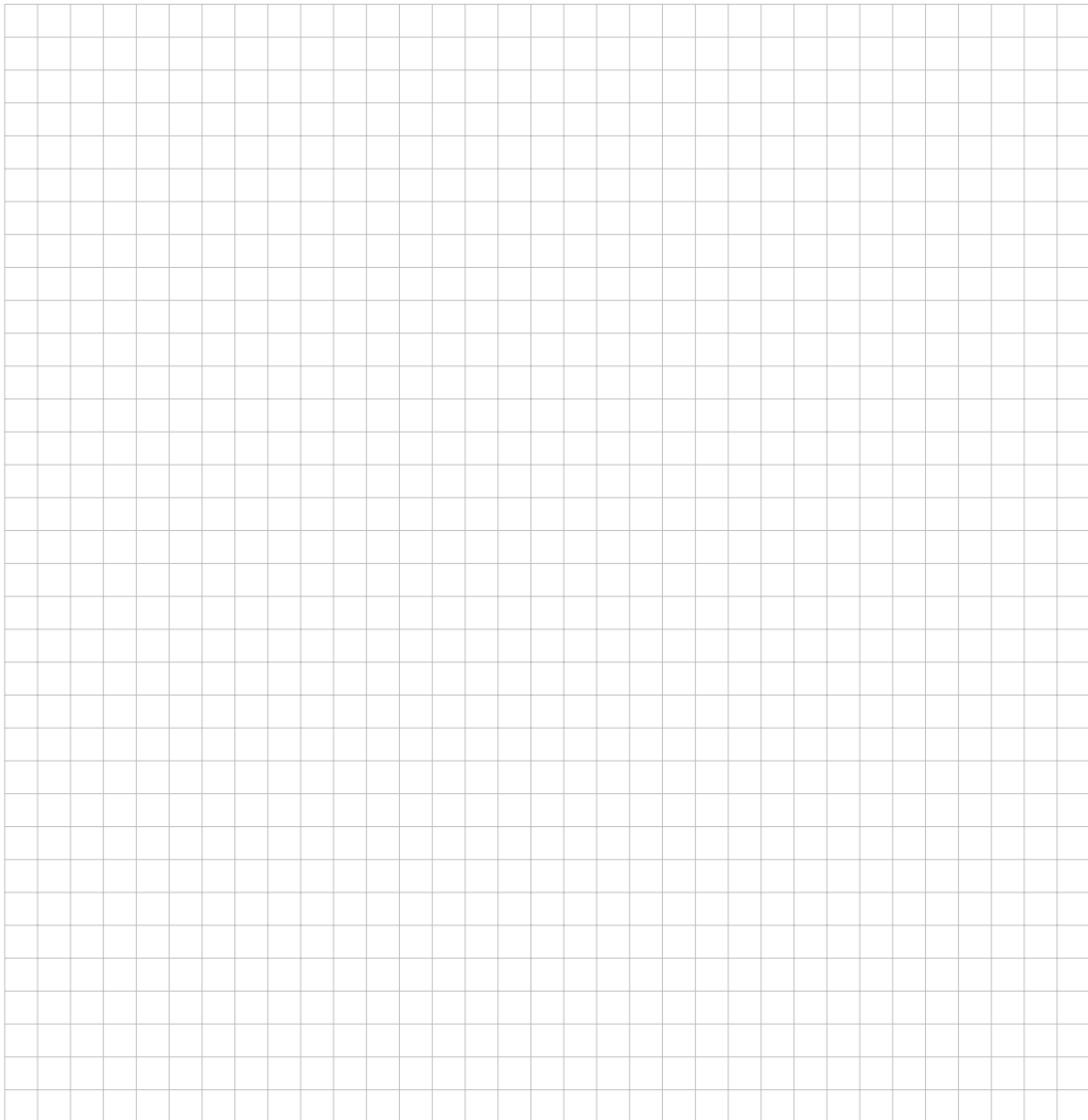
Question 5: Cette question est notée sur 6 points.

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'exactlyement $k - 1$ nombres de la liste sont plus grands que le nombre en question.

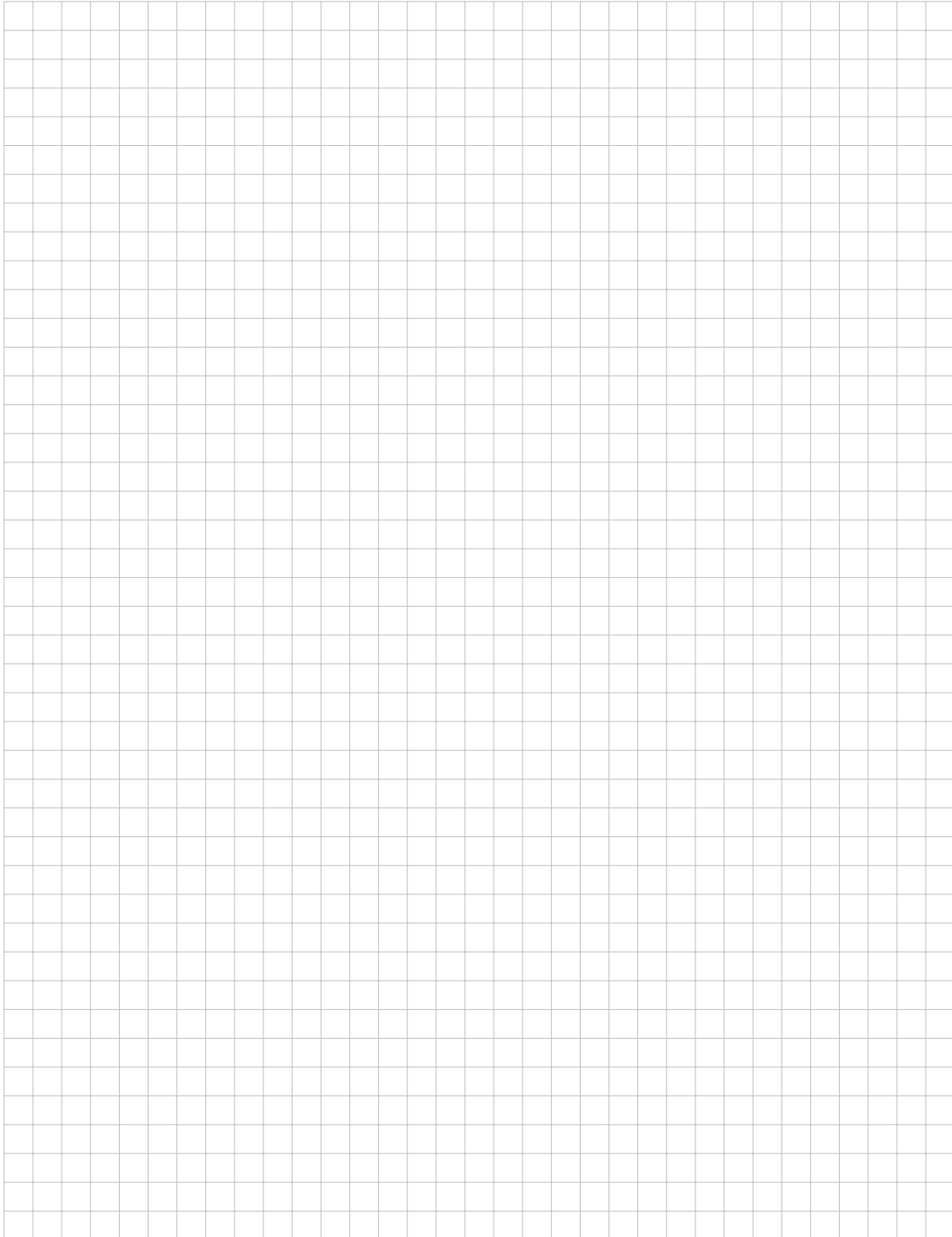
Ainsi, par exemple, dans la liste $L = (13, 22, 12, 37, 19, 8, 56, 17)$, le rang du nombre 56 est égal à 1, tandis que celui du nombre 19 est égal à 4.

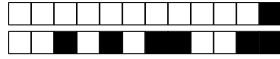
a) (3 points) Ecrire un algorithme de complexité temporelle $\Theta(n)$ dont les données d'entrée soient la liste L , sa taille n , ainsi qu'un autre nombre entier positif x , et dont la sortie soit le rang de x dans la liste L si x appartient à L , ou 0 si x n'appartient pas à L .





b) (3 points) *De plus*, supposons maintenant que la liste L soit triée dans l'ordre *décroissant*. 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.

☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6

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 $k \in \{1, \dots, n\}$ tel que

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

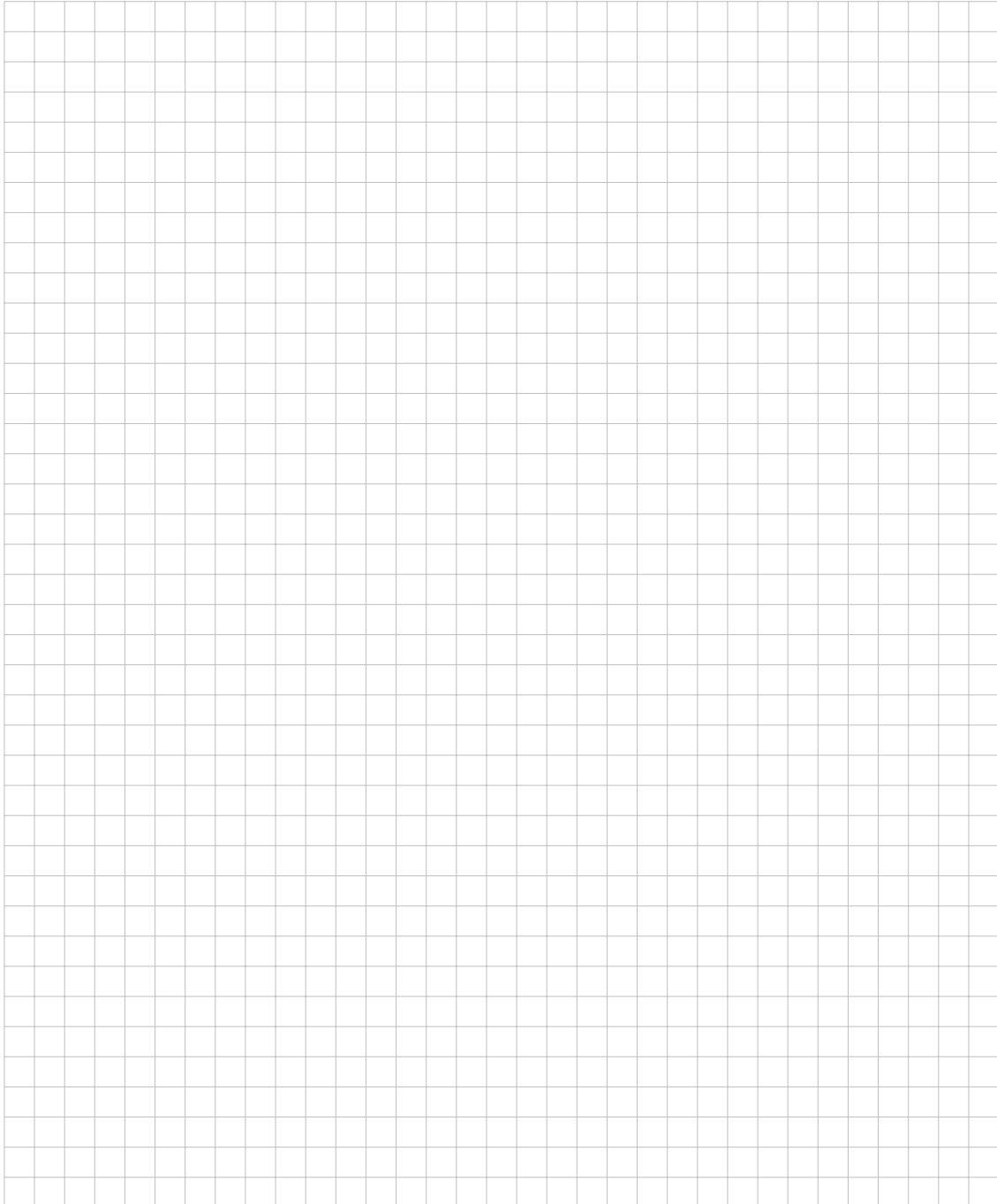


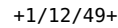


b) (3 points) Ecrire maintenant un algorithme *récuratif et sans boucle* dont les entrées et la sortie soient identiques à celles de l'algorithme précédent.

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



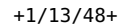


$$\begin{bmatrix} \square \\ \square \end{bmatrix}_0 \begin{bmatrix} \square \\ \square \end{bmatrix}_1 \begin{bmatrix} \square \\ \square \end{bmatrix}_2$$

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

☐ ₀ ☐ ₁ ☐ ₂

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

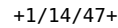


$$\begin{bmatrix} \square \\ \square \end{bmatrix}_0 \begin{bmatrix} \square \\ \square \end{bmatrix}_1 \begin{bmatrix} \square \\ \square \end{bmatrix}_2$$

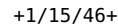
Fournissez uniquement la réponse demandée ; aucune étape intermédiaire n'est nécessaire.

☐ ₀ ☐ ₁ ☐ ₂

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



0 1 2



Question 12: Cette question est notée sur 2 points.

0 1 2

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.

0 1 2 3

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)
```

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



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 !*

