



# Notes d'implémentation

Si vous ne parvenez pas à soumettre une solution sur CMS, vous êtes priés de procéder comme suit :

- Mettez vos fichiers à soumettre dans un dossier nommé `submit_<nom_du_probleme>` sur le bureau (ce dossier devrait déjà être présent) **avant la fin de l'épreuve**.
  - Pour les problèmes "output-only", vos fichiers doivent être nommés `01.out`, `02.out`, ..., `10.out`.
  - Pour les autres problèmes, il doit y avoir **exactement un** fichier dans ce dossier, contenant votre code source.
- Demandez à votre team leader de soumettre un appel.

Si vous pensez que vous devriez avoir du temps additionnel, vous être priés de procéder comme suit :

- Envoyez une requête de clarification (par CMS ou sur papier) le plus rapidement possible.
- Ne quittez pas votre place et ne parlez pas aux autres candidats à la fin de l'épreuve.

Chaque problème a sa propre archive qui est disponible à la fois sur le CMS et sur votre bureau.

Pour les problèmes "output-only" :

- L'archive contient les entrées de chaque test et des données d'exemple. Chaque test est une sous-tâche distincte.
- Vous pouvez soumettre plusieurs fichiers de sortie sous la forme d'un fichier zip. Pour cela, vos fichiers de sortie doivent être nommés `??.out`, où `??` est le numéro du test (e.g., `03.out`). Vous pouvez zipper plusieurs fichiers en utilisant la commande suivante: `zip output.zip *.out`
- Vous pouvez faire jusqu'à 100 soumissions pour les tâches "output-only". Dans chaque soumission, vous pouvez inclure les fichiers de sortie des sous-ensembles de tests que vous désirez.

Pour les autres problèmes :

- L'archive contient des évaluateurs d'exemple, des implémentations d'exemple, des exemples de données de test et des scripts de compilation.

- Vous devez soumettre un seul fichier et vous pouvez faire jusqu'à 50 soumissions par problème.
- Le nom du fichier que vous devez soumettre est donné dans l'en-tête de l'énoncé du problème. Il doit implémenter les fonctions décrites dans l'énoncé du problème et utiliser les signatures données dans l'implémentation d'exemple.
- Vous êtes libre d'implémenter d'autres fonctions.
- Vos soumissions ne doivent pas lire depuis l'entrée standard, écrire sur la sortie standard ou interagir avec des fichiers. Cependant elles peuvent écrire sur la sortie d'erreur standard.
- Vos soumissions **ne doivent pas appeler `exit()` ou `System.exit()`**.
- Lorsque vous testez vos programmes avec l'évaluateur d'exemple, votre entrée doit correspondre au format et aux contraintes de l'énoncé, sans quoi le comportement est indéfini.
- Dans les entrées de l'évaluateur d'exemple, deux valeurs consécutives sur une ligne sont séparées par une espace unique, sauf si un autre format d'entrée est spécifié explicitement.
- Quand vous testez votre code en local, nous vous recommandons d'utiliser les scripts fournis dans l'archive. Sinon, en particulier en C++, assurez-vous d'utiliser l'option `-std=gnu++14` du compilateur.

## Conventions

Les énoncés spécifient les signatures en utilisant les noms génériques `void`, `int`, `int64` et `int[]` (tableau).

Pour chacun des langages supportés, les évaluateurs utilisent les types de données appropriés suivants :

Langage	<code>void</code>	<code>int</code>	<code>int64</code>	<code>int[]</code>	taille du tableau <code>a</code>
C++	<code>void</code>	<code>int</code>	<code>long long</code>	<code>std::vector&lt;int&gt;</code>	<code>a.size()</code>
Java	<code>void</code>	<code>int</code>	<code>long</code>	<code>int[]</code>	<code>a.length</code>

## Limites

Problème	Limite de temps	Limite de mémoire
line	output-only	output-only
vision	1 sec	1024 MB
walk	4 sec	1024 MB