

# PRÉ CQI

2024

PROGRAMMATION



CAHIER DE CHARGE



## Table des matières

1. Mise en contexte .....	3
2. Objectifs.....	3
3. Déroulement du défi .....	3
3.1 Labyrinthe.....	3
3.1.1 Format de sortie .....	5
3.2 Monopoly .....	5
3.3 Hamming .....	7
3.4 Shakespare.....	8
4. Matrice d'évaluation globale .....	9



## 1. Mise en contexte

Bienvenue au programme de recrutement “pee-wee” de l’Agence Spatiale Canadienne (ASC). En tant que potentielle recrue, vous devez vous démarquer de vos collègues. Pour ce faire, vous êtes tâchés de résoudre une suite de défis en équipe. Ces défis vous permettront de démontrer vos capacités à résoudre des défis nouveaux, mais aussi de faire valoir vos qualités en tant que coéquipier.ère. Comme l’individualisme est l’ennemi numéro 1 du programme spatial, vous serez évalués comme un tout; l’important est que votre équipe complète soit performante.

## 2. Objectifs

En équipe de 4, vous devrez résoudre les quatre défis donnés. Chaque défi vaut exactement 25% de la note finale, donc ils sont tous aussi importants les uns que les autres.

À la fin de la période de conception, vous devrez présenter vos solutions devant des juges. Il s’agira d’expliquer votre algorithme de façon claire et concise.

## 3. Déroulement du défi

### 3.1 Labyrinthe

Pour s’assurer de vos capacités à vous orienter dans le système de coordonnées choisi par l’ASC, vous aurez à produire un algorithme de résolution de labyrinthe projeté sur une carte de cases hexagonales.

Les cases hexagonales sont représentées selon le système de coordonnées efficace hexagonal. Dans ce système, la position de chaque case est spécifiée à l’aide de 3 variables; a, r et c. Pour plus d’informations, vous référer à la page Wikipédia (anglais seulement) :

[https://en.wikipedia.org/wiki/Hexagonal\\_Efficient\\_Coordinate\\_System](https://en.wikipedia.org/wiki/Hexagonal_Efficient_Coordinate_System)

Chaque labyrinthe possède une case de début, une de fin, des murs et des surfaces déplaçables. Le but est simple, partir du début pour se rendre à la fin.

Les labyrinthes sont stockés sous format json selon la convention suivante:

- Chaque objet possède un attribut “\_jsontype\_” qui spécifie le type de l’objet sérialisé. Les 3 types d’objets sont “Labyrinth”, “HexTile” et “TileType”.

**- Labyrinth:**

- “\_json\_type\_” : “Labyrinth”
- “tiles” : Liste de paires de (“HexTile”, “TileType”)

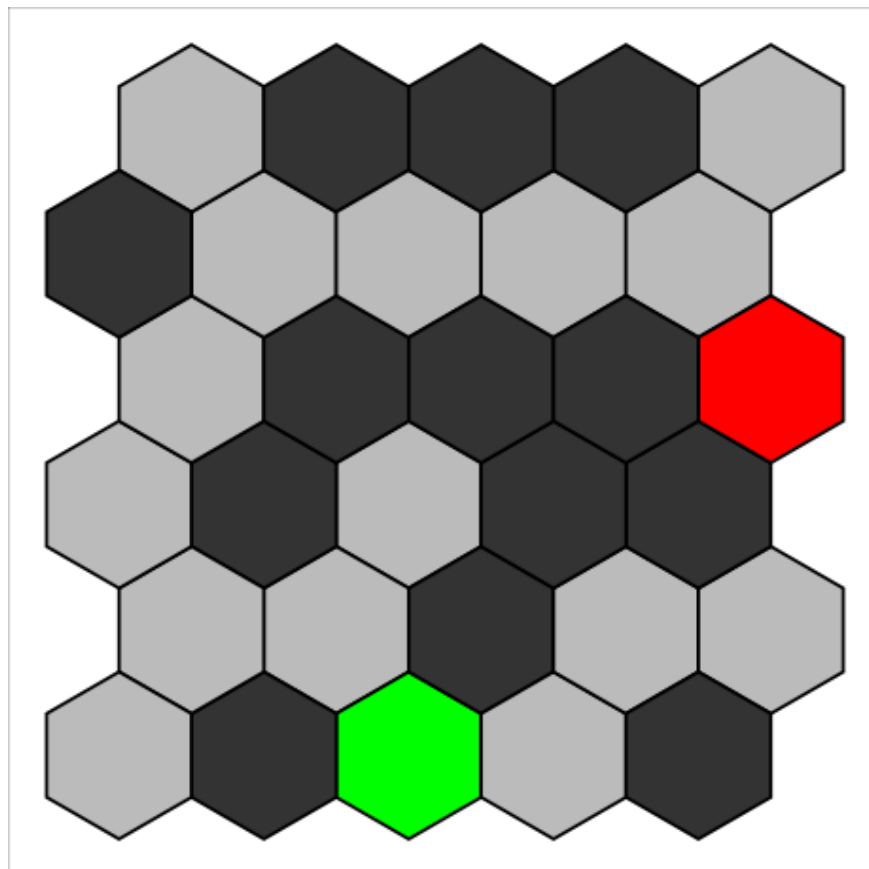
**- HexTile:**

- “\_json\_type\_” : “HexTile”
- “a” : Entier entre 0 et 1
- “r” : Entier non-borné
- “c” : Entier non-borné

**- TileType:**

- “\_json\_type\_” : “TileType”
- “type” : { TileType.NORMAL, TileType.WALL, TileType.START, TileType.END }

Pour vous assurer d’avoir une bonne représentation du labyrinthe, voici un exemple visuel du labyrinthe “easy/labyrinth\_00.json”. Le vert est la case départ et le rouge est la case de fin.





### 3.1.1 Format de sortie

Écrire les coordonnées du trajet dans un dossier “solutions” suivant la même structure que le dossier “labyrinths”. La solution doit être la liste complète des cases marchés (incluant le début et la fin) en spécifiant seulement leurs coordonnées “(a, r, c)”. Il doit y avoir seulement une coordonnée par ligne. Le nom du fichier de solution doit être le même que celui du labyrinthe, par exemple, la solution au défi “./labyrinths/easy/labyrinth\_00.json” se trouvera dans le fichier “./solutions/easy/labyrinth\_00.txt”.

Par exemple:

(0, 5, 6)

(0, 5, 7)

...

Le score pour cette épreuve sera calculé de la façon suivante:

40% des points pour le résultat: il s’agit du pourcentage de labyrinthes complétés. Chaque labyrinthe vaut exactement 1 point, il n’y a pas de pondération en fonction du niveau de difficulté.

40% pour l’essai: vous serez notés autant sur la lisibilité de votre code que sur la qualité et l’ingéniosité de votre algorithme.

20% pour la présentation: Vous serez amenés à présenter votre solution devant des juges. Une visualisation de quelques solutions est fortement conseillée.

## 3.2 Monopoly

Dans l’espace, on n’est jamais trop prudent. Les évaluateurs de l’ASC souhaitent tester votre capacité à estimer des données complexes, tout en réduisant à zéro le risque qu’une estimation erronée cause un accident catastrophique. Bienvenue dans le monde des heuristiques admissibles! On vous donne l’état d’une partie du jeu Monopoly où 4 joueur.euses ont déjà effectué 5 tours chacun.

L’information vous est donnée dans un fichier texte (in.txt) sous le format suivant:

28 lignes de la forme:

Numéro de case	Nom de la case	Nombre de maisons (5 pour un hôtel)	Est hypothéqué (0 si non, 1 si oui)	Propriétaire (Absent si pas encore achetée)



Suivies de 4 lignes de la forme:

Nom du ou de la joueur.euse (A, B, C ou D)	Budget restant	Numéro de case occupée par le ou la joueur.euse
--	----------------	---

Le fichier contient 32 000 lignes, soit l'information relative à 1000 parties différentes.

Les règles standards du Monopoly original<sup>1</sup> s'appliquent, et les cartes Chance<sup>2</sup> et Caisse Commune<sup>3</sup> sont également les mêmes que dans le jeu original.

Les seules différences sont les suivantes:

1. Les joueur.euses ne peuvent ni faire d'échange ni d'enchères (Un terrain sera toujours acheté si le ou la joueur.euse a le budget nécessaire en liquide. Sinon, le terrain restera neutre).
2. Les joueur.euses démarrent le jeu avec le même montant d'argent, différent à chaque partie et choisi aléatoirement entre 500\$ et 2500\$.

Pour chaque partie, vous devrez estimer le montant d'argent avec lequel les joueur.euses ont débuté la partie.

**ATTENTION.** Vous ne devez en **AUCUN** cas surestimer le montant. Vous devez donc avoir la valeur exacte (presque impossible) ou sous-estimer la valeur.

Déposez vos estimations dans un fichier monopoly/out.txt, en écrivant un entier par ligne pour chacune des parties.

Le score pour cette épreuve sera calculé de la façon suivante:

- 40% des points pour le résultat: La somme des différences entre les vraies valeurs et vos estimations pour chaque partie, normalisée en la divisant par le score obtenu par le participant avec le meilleur résultat.
- 40% pour l'essai: vous serez notés autant sur la lisibilité de votre code que sur la qualité et l'ingéniosité de votre algorithme.
- 20% pour la présentation: Vous serez amenés à présenter votre solution devant des juges.

<sup>1</sup> <https://www.hasbro.com/common/instruct/00009.pdf>

<sup>2</sup> <https://monopolyguide.com/traditional/monopoly-list-of-chance-cards-main-version/>

<sup>3</sup> <https://monopolyguide.com/traditional/monopoly-list-of-community-chest-cards-main-version/>



### 3.3 Hamming

Le défi consiste à décoder une suite de messages codés de la façon suivante:

- Conversion du message en binaire à l'aide du standard ASCII (8 bits par caractères)
- Le message binaire complet est permuté à l'aide d'une matrice de permutation d'une taille aléatoire entre 2 et 10 (inclus). Cette matrice est fournie au début du message dans sa forme "plate"; chacune des rangées est mise les unes derrière les autres pour former un vecteur de taille  $n^2$ . IMPORTANT: La matrice donnée est celle utilisée pour encoder le message.
- Finalement, encodage du binaire en Hex-CQI, qui est un système en base 17. Les caractères composant la base sont les 16 caractères de l'hexadécimal (0-9A-F) et CQI, qui doit être considéré comme le 17ème symbole de la base.

Chaque message à décoder est sur une ligne dans un fichier hamming/in.txt, séparée par un "\n". Chaque ligne est composée d'une matrice de permutation et du message encodé. Ces deux éléments sont séparés par un ":".

Par exemple, si vous voyez la ligne suivante:

"0100000100101000:11d48ed9dCQIc6ab6c6147d845e586da03b9"

La matrice de permutation obtenue devrait être:

0	1	0	0
0	0	0	1
0	0	1	0
1	0	0	0

Le message en sortie devrait être "Bonjour le monde!"

Le format de sortie est un fichier composé de tous les messages décodés, chacun sur sa propre ligne et dans l'ordre dans lequel ils se trouvent dans le fichier encodé. Pour sauter un message que votre algorithme n'est pas en mesure de décoder, simplement laisser la ligne vide. Veuillez-vous assurer de bien laisser une ligne vide, sinon le script de correction va se perdre et donner une mauvaise réponse pour chacun des messages qui suit. Sauvegarder ce fichier sous hamming/out.txt .



**IMPORTANT:** Assurez-vous que la lecture du fichier de messages encodés et que l'écriture des réponses dans le fichier de sortie soit toute faite en utf-8.

Le score pour cette épreuve sera calculé de la façon suivante:

- 40% des points pour le résultat: votre fichier de messages décodés sera noté sur 808 messages puis converti en pourcentage. Par exemple, un fichier de 404 messages valides donnerait une note partielle de 50%, donc une note pondérée de 20%.
- 40% pour l'essai: vous serez notés autant sur la lisibilité de votre code que sur la qualité et l'ingéniosité de votre algorithme.
- 20% pour la présentation: Vous serez amenés à présenter votre solution devant des juges.

### 3.4 Shakespeare

Voici une suite logique amusante: -7; 1; -2; 3; -1; 2; 3; 11; 38; 423...

Les évaluateurs de l'ASC veulent valider votre capacité à rédiger des rapports avec éloquence et style. Vous devrez donc écrire un script en SPL (Shakespeare Programming Language) qui produit les 5 éléments suivants de la suite ci-haut lorsqu'on lui donne les deux premiers nombres en entrée (deux nombres sont suffisants pour calculer le reste de la suite). Par exemple, avec les entrées -7 et 1, le script doit produire sur des lignes différentes -7; 1; -2; 3; -1; 2; 3. Votre script devra être remis dans un fichier shakespeare/script.spl.

Note: Cette suite n'est qu'un exemple. En réalité, dû aux limitations de l'interpréteur.trice, aucun nombre négatif ne sera donné en entrée.

Le score pour cette épreuve sera calculé de la façon suivante:

- 40% des points pour le résultat: votre script sera testé pour s'assurer qu'il accomplit bien la tâche demandée.
- 40% pour l'essai: il s'agit de la qualité de la prose dans votre script. Il faut avoir l'impression de lire une pièce de théâtre et non pas un algorithme.
- 20% pour la présentation: Vous serez amenés à présenter votre solution devant des juges.

Ressources pertinentes:

[https://en.wikipedia.org/wiki/Shakespeare\\_Programming\\_Language](https://en.wikipedia.org/wiki/Shakespeare_Programming_Language)

<https://esolangs.org/wiki/Shakespeare>

<https://esolangpark.vercel.app/ide/shakespeare> (interpréteur en ligne)

**Pour toutes questions, vous pouvez communiquer avec :**

Olivier Langlois et Léo Guérin-Morneau ([programmation@cqi-qec.qc.ca](mailto:programmation@cqi-qec.qc.ca)) – Directeurs de la compétition

Frédéric Fortier-Chouinard ([ffortierchouinard@cqi-qec.qc.ca](mailto:ffortierchouinard@cqi-qec.qc.ca)) – Vice-président aux compétitions de la CQI 2024





## 4. Matrice d'évaluation globale

Critère d'évaluation	Pointage
<b>Prototype (qualité du code, ingéniosité de la solution)</b>	<b>40</b>
Labyrinthe	10
Monopoly	10
Hamming	10
Shakespeare	10
<b>Essai</b>	<b>40</b>
Labyrinthe	10
Monopoly	10
Hamming	10
Shakespeare	10
<b>Présentation</b>	<b>20</b>
Éloquence et clarté du discours	10
Clarté et esthétique du support visuel	10
<b>Pénalités</b>	
Plagiat	Élimination
Citation insuffisante	-50
Documents non reçus dans les temps	-50
Membre de l'équipe absent	-25
Non-respect des spécificités des documents soumis	-10
Entrée dans la salle de présentation avant le moment permis (après la première infraction)	-10
<b>Total</b>	<b>100</b>