Epreuve Python

Informations Générales

- ✓ L'épreuve comporte 3 missions à résoudre en **3 heures** consécutives :
  - ***Mission1 :*** *Un QCM en ligne comporte 25 questions*
  - ***Mission2 :*** *Problème1*
  - ***Mission3 :*** *Problème2*
- ✓ Chaque équipe doit créer un dossier de travail portant le nom **« SalleN°_TableN° »** qui sera utilisé pour enregistrer tous les fichiers solutions de l'épreuve.

Exemple : Salle1_Table3

Evaluation & calcul du score

- ✓ Les solutions proposées par les candidats sont évaluées par les membres de jury et notées en fonction de leur exactitude (les bugs ne sont pas tolérés) et de leur vitesse et utilisation mémoire (généralement liées à la complexité en temps et en mémoire de l'algorithme implémenté).
- ✓ Grille de répartition des scores :

MISSION1	MISSION2	MISSION3	BONUS	(*)
30 points	30 points	30 points	10 points	-

compétences de vie : 10 points attribués **pendant l'épreuve** par les membres de jury pour évaluer les capacités des candidats à collaborer et à travailler en équipe.

Mission1 [QCM en ligne]

QCM sur le langage de programmation python comportant 25 questions.

Les règles d'une épreuve en ligne :

- *Durée de l'épreuve : 30 mn*
- *C'est interdit d'utiliser les smartphones,*
- *C'est interdit de naviguer entre les fenêtres ou d'utiliser des IDE (vs-code, Thonny, ect)*

Hachage des transactions sur la blockchain

Vous êtes membre d'une équipe de développement travaillant sur une blockchain décentralisée pour une entreprise de e-commerce. Votre entreprise souhaite assurer la sécurité et l'intégrité des transactions sur sa plateforme en utilisant des fonctions de hachage sécurisées.

Les fonctions de hachage sont utilisées dans de nombreux domaines de l'informatique, notamment pour la sécurisation des mots de passe, la vérification de l'intégrité des données, la création de signatures numériques, la génération de clés de chiffrement, et bien sûr, pour la création et la vérification de transactions dans les blockchains.

Votre mission est de développer en python une fonction de hachage efficace pour la blockchain nommée STEM_HASH(transaction,n), capable de produire une empreinte numérique unique de taille n pour une transaction donnée. Cette empreinte numérique peut ensuite être utilisée pour vérifier l'intégrité et l'authenticité des transactions dans une blockchain.

PRINCIPE DE L'ALGORITHME DE HACHAGE

Etape1 :

Convertir une chaîne de caractères en tableau d'entiers entre 0 et 99 et si besoin rajouter des zéros à la fin pour que le tableau ait une longueur multiple de n.

> *La formule à utiliser pour convertir un caractère en un entier strictement inférieur à 100 est :* ***Code ASCII d'un caractère modulo 100.***

Etape2 :

Partant d'un tableau de longueur multiple de n, on le découpe en séquences de longueur n et on calcule l'empreinte selon l'algorithme suivant :

- On extrait la première séquence du tableau, on effectue 10 tours de mélange STEM **(Voir le principe «Tour de mélange STEM »).**
- On ajoute terme à terme (et modulo 100), le résultat de ce mélange à la seconde séquence.
- On recommence en partant de la séquence suivante.
- Lorsqu'il ne reste plus qu'une séquence de longueur n, on effectue 10 tours de mélange STEM, le résultat est l'empreinte finale.

Exemple d'illustration :

Supposons que nous avons une transaction qui envoie 35 Bitcoins (BTC) au portefeuille de l'association STEM. Les détails de la transaction, tels que les adresses de portefeuille de l'expéditeur et du destinataire, la quantité de BTC envoyée, la date et l'heure de la transaction, etc., sont combinés dans une chaîne de données, qui est ensuite passée à une fonction de hachage cryptographique, dans notre cas ***STEM_HASH(transaction,n)***

Voici un exemple simple d'appel de la fonction de hachage pour valider la solution proposée :

STEM_HASH*("STEM Association +35", 8)*

Etape1:

- *Convertir une chaîne de caractères en tableau d'entiers :*

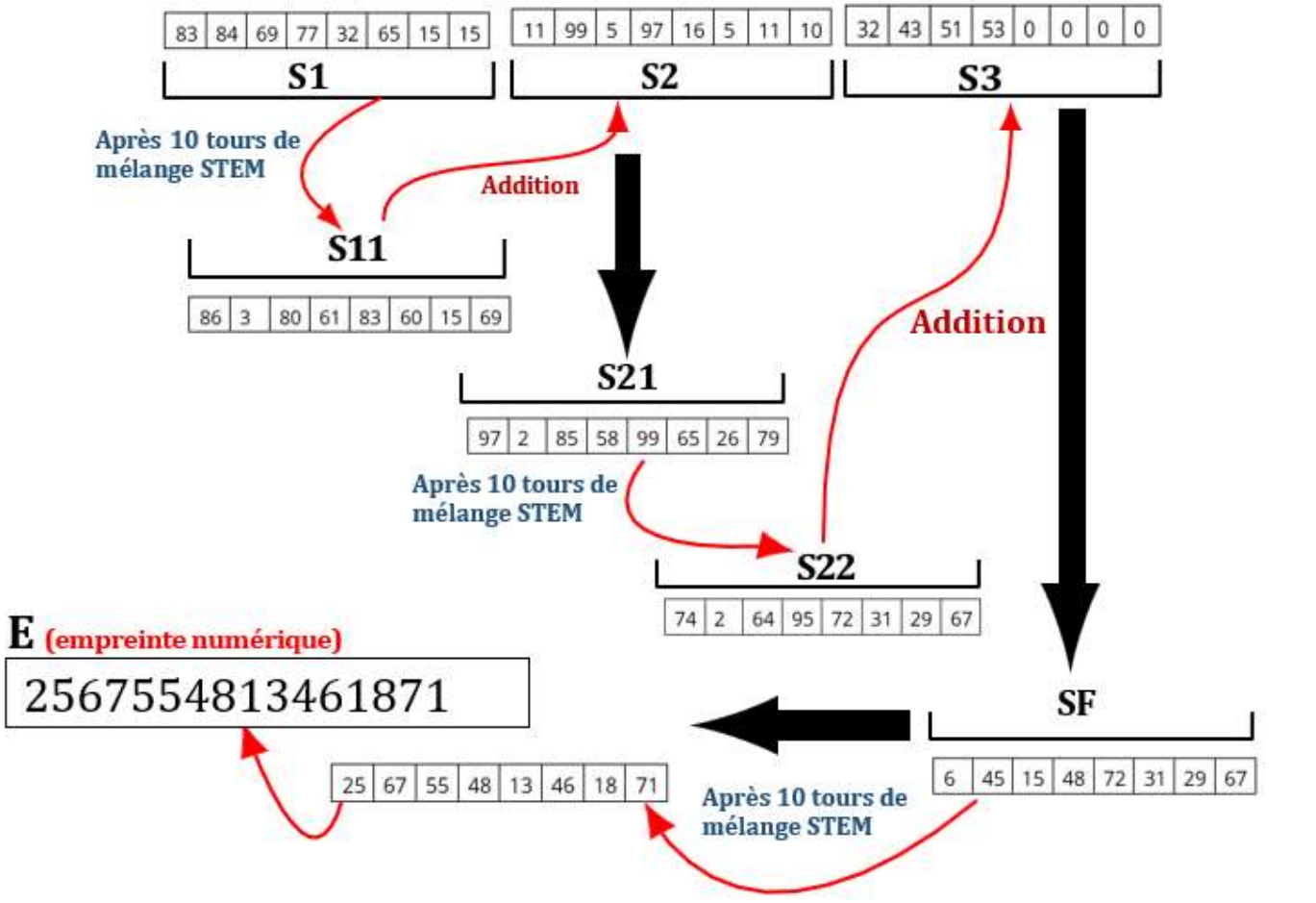
83	84	69	77	32	65	15	15	11	99	5	97	16	5	11	10	32	43	51	53

- *Ajouter des zéros à la fin :*

83	84	69	77	32	65	15	15	11	99	5	97	16	5	11	10	32	43	51	53	0	0	0	0

Etape2:

Voici le schéma du tableau généré dans ***l'étape1*** *: Initialement, il y a trois séquences (S1, S2 et S3) ; dans un second temps, il ne reste plus que deux séquences (S21 et S3) ; finalement, il ne reste qu'une séquence (SF) qui sera convertie en empreinte numérique (E).*



Un Tour de mélange STEM

Un tour de mélange STEM consiste à faire les opérations suivantes :

1) On remplace chaque élément d'indice impair par la somme de l'élément avec son prédécesseur.
2) On multiplie ces entiers par des nombres premiers dans l'ordre commençant par 7, 11, 13,.... et on rajoute 1 :
   N.B : Un nombre premier n'est divisible que par 1 et par lui-même.
3) On effectue une permutation circulaire (le dernier passe devant)
4) On réduit chaque entier modulo 100 afin d'obtenir des entiers entre 0 et 99.

Exemple : Soit la séquence :

83	84	69	77	32	65	15	15

1) Additions :

83	84+83	69	77+69	32	65+32	15	15+15

83	167	69	146	32	97	15	30

2) Multiplications : les 8 nombres premiers : 7, 11, 13, 17, 19, 23, 29, 31

7×83+1	167× 11+1	69× 13 + 1	146× 17 + 1	32× 19 + 1	97× 23 + 1	15× 29 + 1	30× 31 + 1

882	1838	898	2483	609	2232	436	931

3) Permutation circulaire vers la droite :

931	882	1838	898	2483	609	2232	436

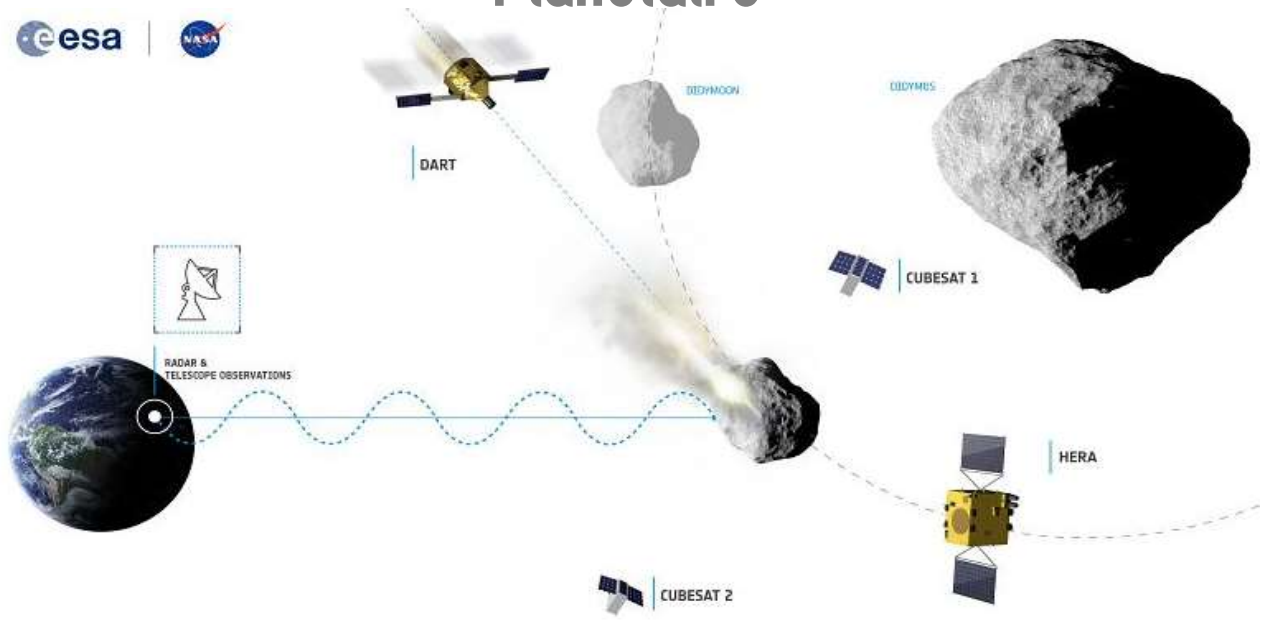
4) Réduction modulo 100 :

31	82	38	98	83	9	32	36

Exemple2 :

```
>>> STEM_HASH("STEM Association -15", 20)
>>>18909542895542849423443461576423723930 24
```

Déviation d'un astéroïde pour la Défense Planétaire



Il existe un grand nombre d'astéroïdes dans le Système Solaire, dont la majeure partie (plus de 98%) est concentrée dans la ceinture principale située entre les orbites de Mars et de Jupiter.

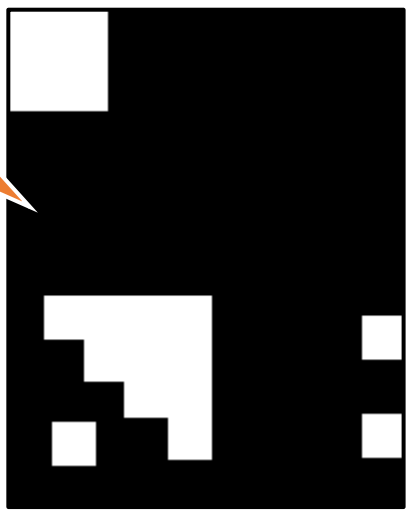
Parfois, sous l'influence de la gravité d'objets plus gros, ils quittent leurs orbites habituelles, volant vers la Terre.

Dans le cadre de la Défense Planétaire, la NASA et l'ESA travaillent sur plusieurs domaines dont la mission DART. C'est une mission de déviation d'un astéroïde.

À partir des images envoyées par les télescopes, la NASA a détecté qu'un astéroïde a quitté son orbite habituelle. Pour cela, elle va chercher à dévier cette cible.

Pour détecter exactement sa position, la NASA a augmenté la sensibilité des télescopes pour le repérer, mais cela a également permis de détecter des satellites en orbite autour de la Terre.

Image1 : Avant déplacement



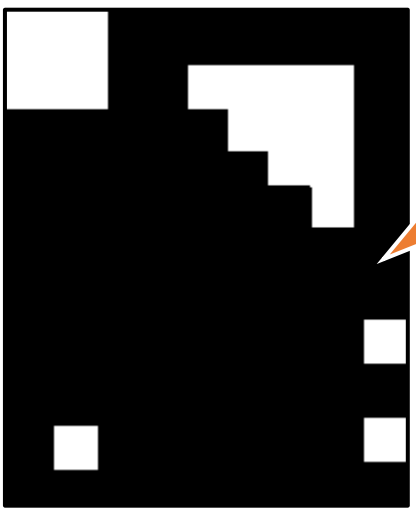


Image2 : Après déplacement

Travail demandé :

- **Vous cherchez à identifier la position exacte de l'astéroïde sur la deuxième image en tenant compte de son déplacement par rapport à la première image supposons que les satellites sont fixes.**
- Votre mission consiste donc à trouver les points qui correspond à l'emplacement de l'astéroïde sur la deuxième image.

Contraintes :

- *1 <= L, C <= 50.*
- *À une position donnée dans les deux matrices :*
  - *1 représente la présence d'un astéroïde ou d'un satellite.*
  - *0 représente une absence d'objet.*

Sortie :

Les coordonnées de l'astéroïde détecté dans la deuxième image, en écrivant : le numéro de la ligne puis le numéro de la colonne. Ces numéros de ligne et de colonne doivent être comptés à partir de 1.

Exemple d'illustration :

Soient M1 et M2 deux images simplifiées envoyées par le télescope avec L=10 et C=8

M1

	1	2	3	4	5	6	7	8
1	1	1	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	1	1	1	1	0	0	0
7	0	0	1	1	1	0	0	0
8	0	0	0	1	1	0	1	0
9	0	1	0	0	1	0	0	0
10	0	0	0	0	0	0	0	0

M2

	1	2	3	4	5	6	7	8
1	1	1	0	0	0	0	0	0
2	1	1	0	1	1	1	1	0
3	0	0	0	0	1	1	1	0
4	0	0	0	0	0	1	1	0
5	0	0	0	0	0	0	1	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	1	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0

Les cordonnées sont : (1, 8), (2, 4), (2, 5), (2, 6), (2, 7), (3, 5), (3, 6), (3, 7), (4, 6), (4, 7), (5, 7)

Mission Bonus

*Etant donné un fichier binaire intitulé **« bd_satellite.bin »** contenant toutes les coordonnées hachées des astéroïdes détectés par la **NASA** durant les 10 dernières années.*
On vous demande d'appliquer la démarche suivante pour vérifier l'existence d'un nouveau astéroïde détecté dans le fichier des cordonnées hachées :

1. *Transformer le script solution du **problème3** en fonction nommée **« Find_Asteroids() »***
2. *Appeler la fonction **STEM_HASH()** pour hacher la chaîne des coordonnées de l'astéroïde détecté renvoyée par la fonction « **Find_Asteroids() ».***
3. *Rechercher dans le fichier **« bd_satellite.bin »** la chaîne hachée, afficher un message indiquant que les coordonnées de l'astéroïde ont été trouvées. Sinon, le programme affiche un message indiquant qu'aucun astéroïde ne correspond aux coordonnées de recherche.*

N.B. :

*Le candidat n'est pas appelé à remplir le fichier **« bd_satellite.bin »***