

SECTION : SCIENCES DE L'INFORMATIQUE

EPREUVE : ALGORITHMIQUE ET PROGRAMMATION

DURÉE : 1 H 30

NOMBRE DE PAGES : 2

COEF : 1

Nom : Prénom : Num :

20

Exercice 1 : (3 pts)

A. Compléter la grille ci-dessous par le nom, en Python, de la fonction ou la méthode en se référant aux rôles donnés.

- 1- Ecrire dans un fichier texte.
- 2- Lier un nom physique à un nom logique.
- 3- Lire à partir d'un fichier binaire.
- 4- Ecrire dans un fichier binaire.
- 5- Permet l'enregistrement dans le fichier.
- 6- Bibliothèque de codage/décodage.
- 7- Opérateur d'appartenance.
- 8- Lire la totalité d'un fichier texte.

1					
2					
3					
	4				
5					
6					
7					
8					

B. Définir le rôle du terme trouvé dans la colonne bordurée en gras.

Exercice 2 : (9 pts)

Le chiffre affine est un chiffre de substitution simple.

Principe de chiffrement :

L'idée est d'utiliser comme fonction de chiffrage une fonction affine du type $y = a*x + b$, où a et b sont des constantes, x est le numéro d'ordre de la lettre du message à crypter et y est le numéro d'ordre de la lettre du message chiffrée.

Evidemment, pour que la lettre chiffrée (y) soit aussi un nombre entre 0 et 25, on travaillera modulo 26. La vraie formule sera donc $y = (a*x + b) \bmod 26$.

N.B. : les numéros d'ordres (x et y) sont selon le tableau ci-dessous :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Exemple :

Soient le message « BRAVO », $a = 3$ et $b = 7$

Message	B	R	A	V	O
x	1	17	0	21	14
y	10	6	7	18	23
Message chiffré	K	G	H	S	X

Principe de Déchiffrement :

Pour déchiffrer une lettre d'ordre y du message chiffré, on calcul l'ordre x de la lettre du message déchiffré par la formule suivante :

$$\begin{cases} x = k * (y - b) \bmod 26 & \text{Si } y \geq b \\ x = 26 + k * (y - b) & \text{Si } y < b \end{cases} \quad \text{Avec } k \text{ est un entier tel que } (a*k) \bmod 26 = 1$$

Exemple :

Soient le message « KGHSX », $a = 3$ et $b = 7$
Selon le calcul k vaut 9

Message chiffré	K	G	H	S	X
y	10	6	7	18	23
x	1	17	0	21	14
Message déchiffré	B	R	A	V	O

Travail demandé :

- 1- Ecrire l'algorithme d'un module, avec les paramètres formels a, b (clés de cryptage) et le fichier à crypter, qui permet de créer et de remplir un fichier texte nommé "**mess_Crypt.txt**" par les chaînes cryptées de chaque ligne d'un fichier texte existant nommé "**messBrut.txt**".
N.B. : Chaque ligne du fichier "**messBrut.txt**" est formée uniquement par des lettres majuscules.
- 2- Ecrire l'algorithme d'un module, avec les paramètres formels a, b (clés de cryptage) et le fichier à décrypter, qui permet de créer et de remplir un fichier texte nommé "**mess_Brut.txt**" par les chaînes décryptées de chaque ligne d'un fichier texte existant nommé "**messCrypt.txt**".
N.B. : Chaque ligne du fichier "**messCrypt.txt**" est formée uniquement par des lettres majuscules.

Exercice 3 : (8 pts)

Une image numérique en couleur est un ensemble de pixels structurés sous la forme d'une matrice, **M1**, de **L** lignes et **C** colonnes où chaque case contient un code de **9 chiffres**, qui correspond à une couleur.

Notre objectif est de transformer cette image en niveaux de gris, cette transformation consiste à remplacer les codes de **9 chiffres** par un entier **décimal** composé de **3 chiffres au maximum**.

Pour déterminer ce code décimal, on suit les étapes suivantes pour chaque case :

- ✓ Les 9 chiffres sont divisés en **3 blocs de 3 chiffres** dans les variables **R**, **V** et **B**.
Exemple : "156107255" → R = "156", V = "107" et B = "255"
- ✓ On détermine la partie entière de la valeur suivante :
 $0.299 * \text{valeur (R)} + 0.587 * \text{valeur (V)} + 0.114 * \text{valeur (B)}$.
Exemple : La partie entière de $(0.299 * 156 + 0.587 * 107 + 0.114 * 255)$ vaut 138
- ✓ Une matrice **M2**, sera rempli par les codes décimaux trouvés à partir des codes de 9 chiffres.

Exemple :

Matrice **M1** pour l'image en couleurs

"156107255"	"103126082"
"230226175"	"001023200"
"000000000"

Matrice **M2** pour l'image en niveaux de gris

138	114
221	36
0

On suppose que le contenu de la matrice correspondante à l'image à traiter est déjà stocké dans le **fichier texte** existant nommé "**image.txt**" de la façon suivante :

- ✓ La 1^{ère} ligne du fichier contient le nombre de lignes de la matrice.
- ✓ La 2^{ème} ligne du fichier contient le nombre de colonnes de la matrice.
- ✓ Les autres lignes du fichier, chacune contient le code de 9 chiffres d'un pixel dans le même ordre de parcourt de la matrice ligne par ligne.

On se propose d'écrire un algorithme, qui permet de créer et de remplir un fichier d'enregistrements nommé "**CoulGris.dat**" par les codes de **9 chiffres** et les codes de **3 chiffres**, sachant que le fichier "**CoulGris.dat**" est rempli de la façon suivante :

- Le premier enregistrement contient le nombre de lignes et le nombre de colonnes.
- Les autres enregistrements contiennent le code de **9 chiffres** et le code décimal qui lui correspond.

Travail demandé :

- 1- Donner les déclarations d'un type pour le fichier "**CoulGris.dat**" ainsi que celles des types nécessaires à sa déclaration et des types pour les matrices **M1** et **M2**.
- 2- Donner, en algorithme, les instructions d'ouverture de deux fichiers.
- 3- Ecrire un algorithme d'un module, nommé **Remplir**, qui permet de remplir une matrice **M1** par les codes de **9 chiffres** à partir du fichier « **image.txt** ».
- 4- Ecrire un algorithme d'un module, nommé **Transformer**, qui permet de remplir une matrice **M2** par les codes décimaux à partir de **M1** en utilisant le principe expliquée ci-dessus.
- 5- Ecrire un algorithme d'un module, nommé **NIV_GRIS**, qui permet de remplir le fichier "**CoulGris.dat**" à partir des matrices **M1** et **M2** en respectant la façon de remplissage annoncée ci-dessus.

Bon Travail