Dernière mise à jour	Informatique	Denis DEFAUCHY
13/02/2023	7 - Matrices de pixels et images	TD 7-13 – Agent secret

Informatique

7 Matrices de pixels et images

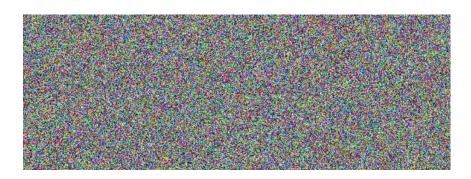
TD 7-13
Agent secret



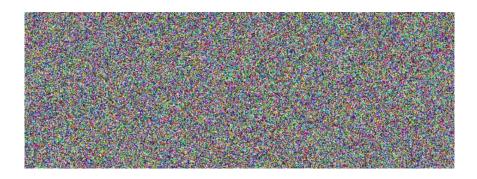
Dernière mise à jour	Informatique	Denis DEFAUCHY
13/02/2023	7 - Matrices de pixels et images	TD 7-13 – Agent secret

Contexte

Vous avez trouvé une clé USB dans votre boite aux lettres contenant l'image « Image_C1.bmp » :



Vous avez par ailler reçu par mail une image semblable « Image_C2.bmp » :

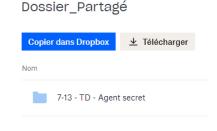


Un message se cache à l'intérieur.

Affichage de l'image

Afin d'assurer un fonctionnement rapide sur tous les ordinateurs, je vous mets à disposition un dossier à télécharger COMPLETEMENT, soit le dossier contenant tous les fichiers, et non les fichiers pris séparément

Sans ouvrir le dossier, faite juste « Télécharger – Téléchargement direct » puis mettez ce dossier dans votre répertoire personnel.



LIEN

Si le téléchargement est sous forme de Rar, Zip... Pensez à dézipper l'archive afin d'avoir le dossier voulu !

Vous y trouverez un code élève et 3 les images « Image_C1.bmp », « Image_C2.bmp » et « Image.bmp » ainsi que leur version numpy « .npy ».

Question 1: Télécharger et exécuter le code fourni qui créera et affichera les images fournies sous Python



Dernière mise à jour	Informatique	Denis DEFAUCHY
13/02/2023	7 - Matrices de pixels et images	TD 7-13 – Agent secret

Cryptage

On souhaite créer deux images im1 et im2 à partir d'une im telles que, en appelant P=[R,G,B] les valeurs des pixels de l'image im :

- Chaque valeur R1, G1 et B1 de chaque pixel P1 de l'image im1 est un entier aléatoire entre 0 et 255 inclus (R1, G1 et B1 sont 3 nombres différents)
- Chaque valeur R2, G2 et B2 de chaque pixel P2 de l'image im2 est la différence P-P1=[R-R1,G-G1,B-B1].

Remarque : En remplaçant dans une copie de im les pixels par ces valeurs [R2,G2,B2] pouvant être hors de l'intervalle [0,255], l'overflow les remettra automatiquement dans l'intervalle [0,255] par complément à 256. En effet :

```
>>> Image[0,0]
array([ 78, 159, 225], dtype=uint8)
>>> Image[0,0] = [-10,-10,-10]
>>> Image[0,0]
array([246, 246, 246], dtype=uint8)
```

Question 2: Proposer une fonction Cryptage(im) renvoyant les deux images attendues Question 3: Tester cette fonction en créant « Image_1 » et « Image_2 » et vérifier que le résultat ressemble aux images « Image_C1 » et « Image_C2 » fournies

Décryptage

Il est possible de sommer deux array, mais nous l'interdisons ici.

Question 4: Proposer une fonction Decryptage(im1,im2) décryptant les deux images im1 et im2

Remarque : Comme vu précédemment, l'overflow ne sara pas non plus un problème lors de la réalisation de cette somme.

```
>>> Image[0,0] = [300,300,300]
>>> Image[0,0]
array([44, 44, 44], dtype=uint8)
```

Question 5: Utiliser cette fonction pour créer l'image « Image_3 » par décryptage de vos images cryptées à la partie précédente « Image_2 » et « Image_3 »

Application

Question 6: Décrypter les images « Image_C1 » et « Image_C2 »

