



BI A

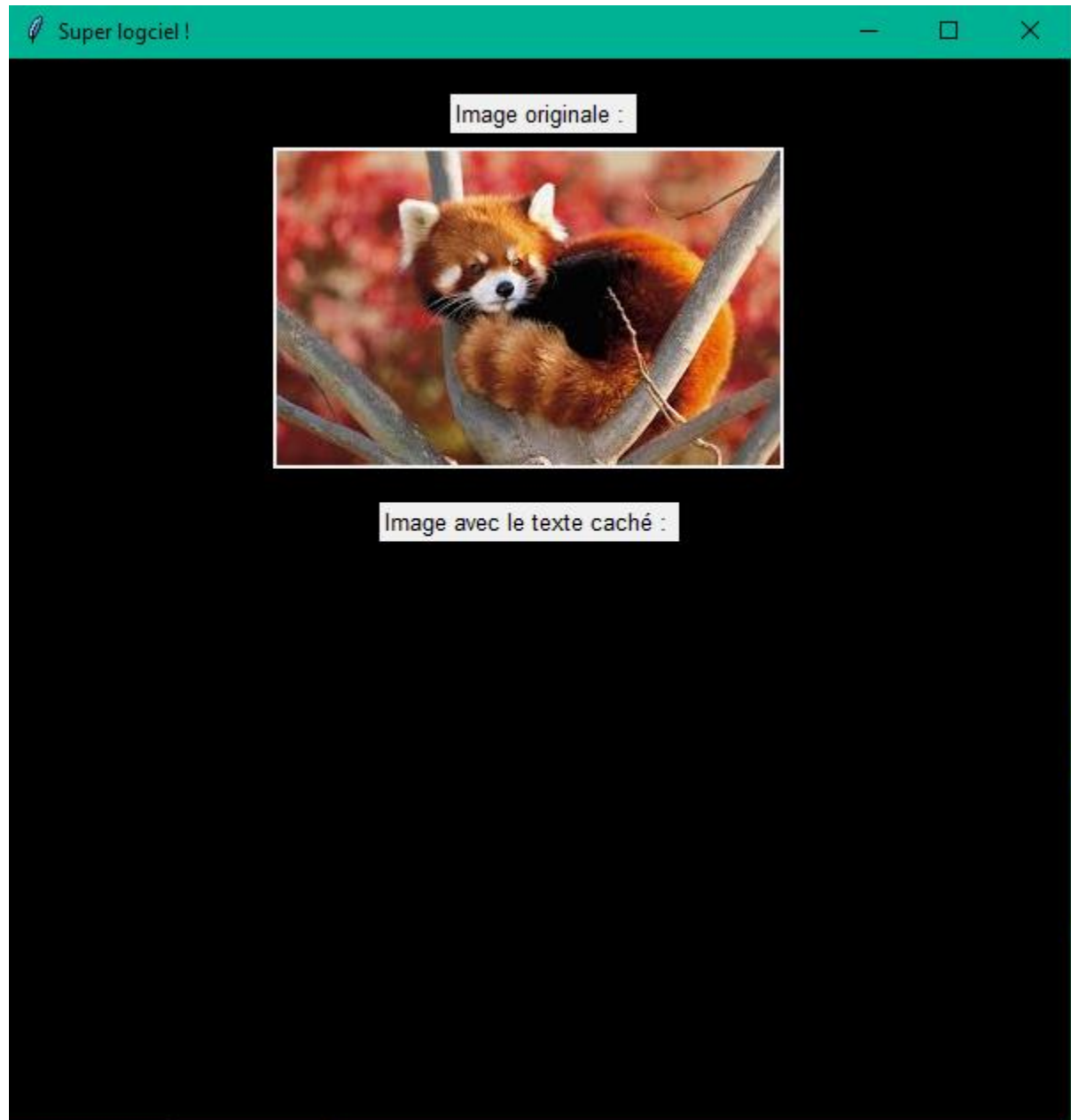
Devoir Final et Devoir N°1 Mathématiques

Devoir Final

Utilisation du logiciel (devoir final)

I) Devoir final

Concernant le devoir final, nous avons réalisé le devoir ainsi que le challenge numéro deux, étant deux sur le projet. L'objectif était de cacher une phrase dans une image. Notre idée était de cacher « Hello World ! » dans chacune des trois couleurs primaires de l'image, le rouge le vert et le bleu. Nous avons donc dans un premier temps installé puis importé un module Python nommé : PIL.



Ce module nous permet d'effectuer des actions sur des images assez facilement, grâce à lui nous avons réussi à récupérer la couleur de chaque pixel de l'image et de créer trois listes, une pour chaque couleur primaire.

Nous avons ensuite extrait les premiers éléments de la liste rouge, de la liste bleue ainsi que de la liste verte, puis les avons converties en valeurs binaires. De même pour notre chaîne de caractère (« Hello World ! ») que nous avons également convertie en valeurs binaires.

Par défaut après la conversion en valeur binaire, Python ajoute « 0b » devant celle-ci, nous avons donc dû supprimer ces deux valeurs systématiquement.

Pour terminer ce tp il nous suffisait de remplacer les chaines de caractère RGB par notre phrase à cacher et ensuite afficher l'image.

Pour le chiffrement de la phrase à cacher, le challenge N°2 donc. Nous avons décidé de créer notre propre système, il est plutôt basique et doit être grandement modifié pour être réellement efficace. De plus notre clé de déchiffrement n'en n'est pas vraiment une. Elle fonctionne sur cette phrase type mais si nous changeons la phrase, le chiffrement changera l'emplacement de chaque grain de sel ajouté dans la phrase, la clé de déchiffrement retirera donc les mauvaises données. Nous pensons faire des boucles avec les valeurs que nous savons possiblement générées par notre système de chiffrement, et ainsi tester toutes les possibilités et quitter la boucle de chaque grain de sel une fois la chaîne de caractère débarrassée de tous les grains.

2) Utilisation du logiciel (devoir final)

Possibilité d'utiliser son IDE ou le PowerShell Windows. Nous avons utilisé beaucoup de 'print' dans notre script python afin que vous puissiez suivre les actions du programme sur l'image et les chaînes de caractères.

```
114
valeurs rouges remplacées -> 1101100
nombre decimal equivalent est -> 108
108
valeurs rouges remplacées -> 1100100
nombre decimal equivalent est -> 100
100
valeurs rouges remplacées -> 1000000
nombre decimal equivalent est -> 32
32
valeurs rouges remplacées -> 100001
nombre decimal equivalent est -> 33
33
valeurs vertes binaires -> 10100101
valeurs vertes binaires -> 10100010
valeurs vertes binaires -> 10011101
valeurs vertes binaires -> 10011100
valeurs vertes binaires -> 10011011
valeurs vertes binaires -> 10010111
valeurs vertes binaires -> 10001101
valeurs vertes binaires -> 10000101
valeurs vertes binaires -> 11111110
valeurs vertes binaires -> 11111111
valeurs vertes binaires -> 11111110
valeurs vertes binaires -> 11111101
valeurs vertes binaires -> 1110101
valeurs vertes binaires -> 1101011
valeurs vertes remplacées -> 10010000
nombre decimal equivalent est -> 72
valeurs vertes remplacées -> 1100101
nombre decimal equivalent est -> 101
valeurs vertes remplacées -> 1101100
nombre decimal equivalent est -> 108
valeurs vertes remplacées -> 1101100
nombre decimal equivalent est -> 108
valeurs vertes remplacées -> 1101111
nombre decimal equivalent est -> 111
valeurs vertes remplacées -> 1000000
nombre decimal equivalent est -> 32
valeurs vertes remplacées -> 1010111
nombre decimal equivalent est -> 87
valeurs vertes remplacées -> 1101111
nombre decimal equivalent est -> 111
valeurs vertes remplacées -> 1110010
nombre decimal equivalent est -> 114
valeurs vertes remplacées -> 1101100
nombre decimal equivalent est -> 108
valeurs vertes remplacées -> 1100100
nombre decimal equivalent est -> 100
valeurs vertes remplacées -> 1000000
nombre decimal equivalent est -> 32
valeurs vertes remplacées -> 100001
nombre decimal equivalent est -> 33
valeurs bleues binaires -> 11111111
valeurs bleues binaires -> 11111100
valeurs bleues binaires -> 11110000
valeurs bleues binaires -> 11101111
valeurs bleues binaires -> 11101111
valeurs bleues binaires -> 11100111
valeurs bleues binaires -> 11011101
valeurs bleues binaires -> 11001110
valeurs bleues binaires -> 11000111
valeurs bleues binaires -> 11001110
valeurs bleues binaires -> 11010000
valeurs bleues binaires -> 11001000
valeurs bleues binaires -> 1100101
valeurs bleues binaires -> 1011100
```