Trouver le code du cadenas de Mario



Mario a perdu la combinaison à 4 lettres du cadenas qui sécurise son coffre.

Le code est caché dans une image numérique.

**Votre objectif : utiliser le protocole de codage pour retrouver le code à 4 lettres et ouvrir le coffre.**

|  |  |
| --- | --- |
| Image originale  Mario101x100.png | Image contenant le code  Mario101x100 - CODE.png |
|  | |

**Protocole utilisé pour cacher le code dans l’image :**

* Chaque lettre du code à cacher sera représentée par son code ASCII, écrit en base 2. Par exemple, le code ASCII de "A" en binaire, sur un octet (8 bits) est 01000001.
* Le codage de la première lettre débute sur le pixel de coordonnées (colonne=19 ; ligne=29). Une seule lettre est cachée par ligne et le codage de la deuxième lettre débute sur le pixel de coordonnées (colonne=19 ; ligne=28)…
* Les caractères « 0 » ou « 1 » sont « cachés » dans les composantes RVB de certains pixels:  
  Si le nombre est pair il correspond à un « 0 »  
  Si le nombre est impair il correspond à un « 1 »
* Si, dans l’image initiale, la parité du nombre ne correspond pas au caractère à cacher, ajouter 1, sinon le laisser inchangé.

Exemple pour coder la lettre « V » (tableau caractères ASCII sur fiche plastifiée) :

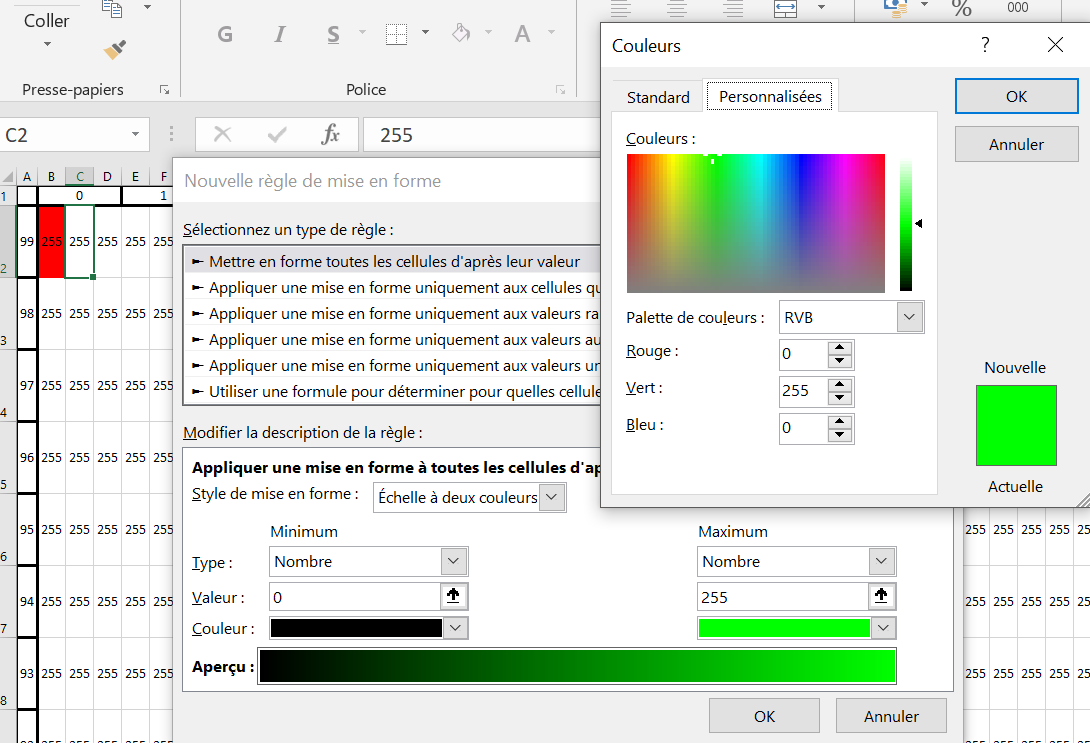
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Pixel 1 | | | Pixel 2 | | | Pixel 3 | | |
| Image initiale | 138 | 65 | 23 | 234 | 87 | 34 | 126 | 90 | 40 |
| Caractère  à cacher | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |  |
| Image modifiée | 138 | 65 | 24 | 235 | 88 | 35 | 127 | 90 | 40 |

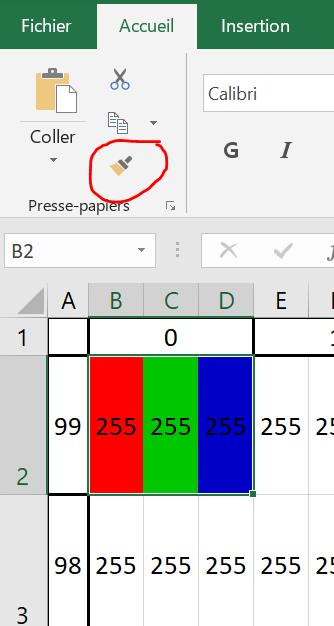
De l’aide pour relever le défi

# Activité de coloriage numérique : (Activité « branchée » et nécessite Excel, par l’enseignant au bureau ? facultatif)

Ouvrez dans Excel le tableau de nombres de l’image Mario100x101.png

Sur la cellule B2 : Accueil\Style\Mise en forme conditionnelle\nouvelle règle :



Renouvelez pour le vert en C2 et le bleu en D2

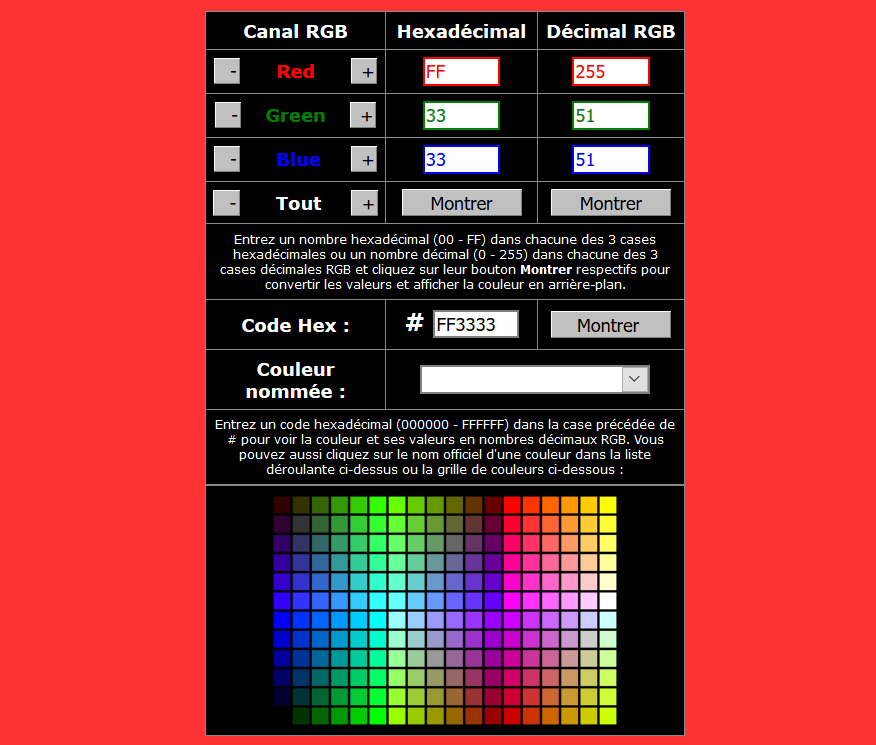
Sélectionnez la plage de cellules (A2 :C2), copiez le format (outil pinceau) à toutes les cellules du tableau.

*Utiliser le zoom pour réduire la taille de l’image au maximum. Voyez-vous l’image ? Percevez-vous les pixels ?*

# Activité Branchée (PC ou smartphone)

Utiliser le simulateur ci-dessous pour déterminer les composantes R, V, B associées à des pixels de différentes couleurs.

<http://www.proftnj.com/RGB3.htm>



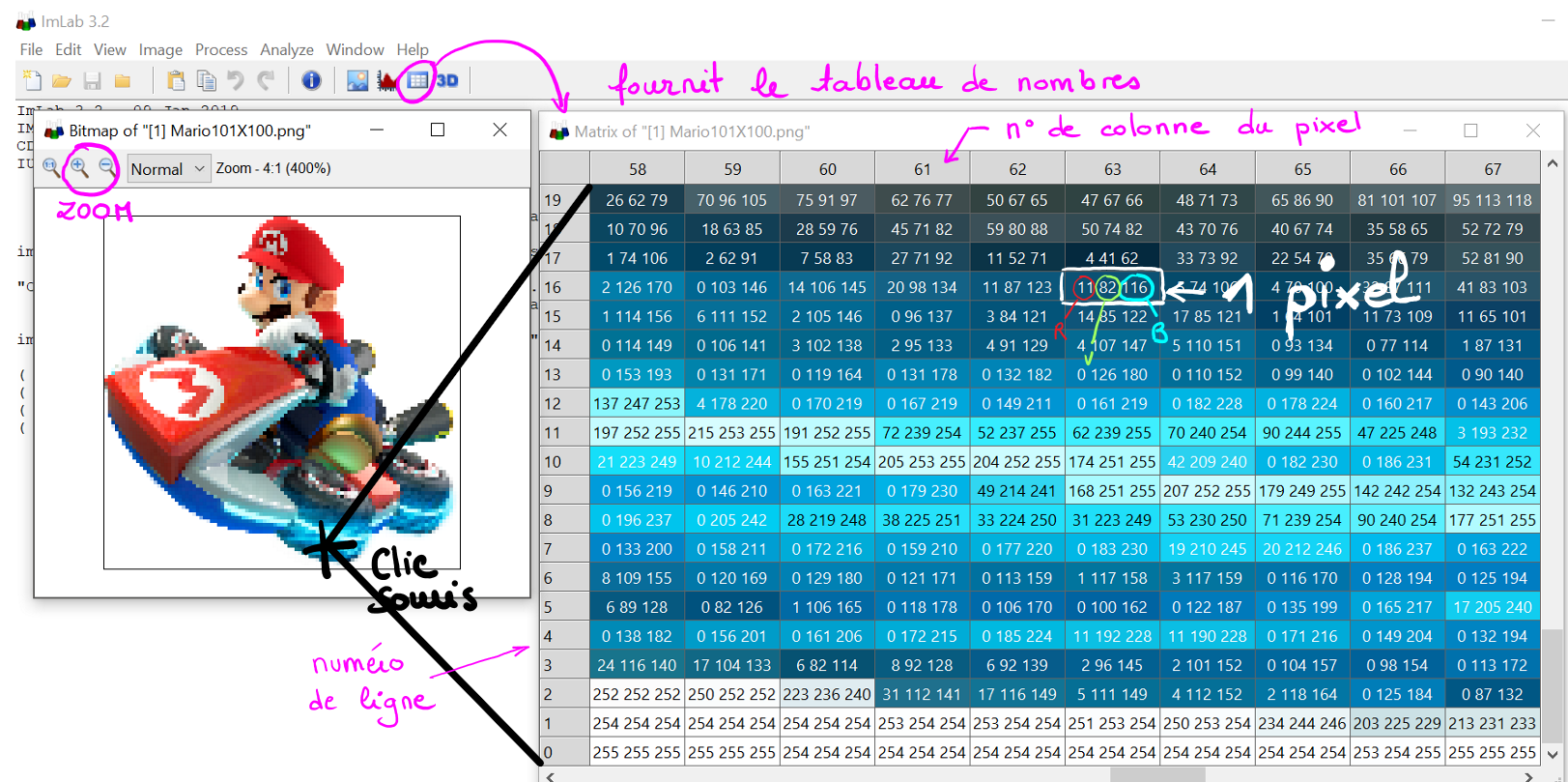
|  |  |  |  |
| --- | --- | --- | --- |
| CYAN | R = | V= | B= |
| JAUNE | R = | V= | B= |
| VIOLET | R = | V= | B= |
| ORANGE | R = | V= | B= |
| GRIS CLAIR |  |  |  |

*Percevez-vous une variation de couleur si vous augmentez chaque composante   
de 1 ?*

# Activité déchiffrage du code (possible en débranché avec fiches plastifiées)

Ouvrez l’image Mario101x100-CODE.png dans le logiciel ImLab.

Affichez le tableau de nombres de l’image, cliquez sur des pixels de différentes couleurs pour vous déplacer dans le tableau de nombres.



* Tracez deux axes fléchés, de même origine 0, orientés dans le sens croissant des numéros de colonne et de ligne du pixel. Ecrire « ligne » et « colonne » à l’extrémité de l’axe correspondant.
* Repérez, par une croix sur l’image ci-dessous, le pixel de coordonnées (colonne=19 ; ligne=29). Faites figurer les coordonnées du pixel sur les axes correspondants.
* Noter sur les axes le nombre total de pixels en hauteur, en largeur.
* Calculer la **définition** de l’image : elle est égale au nombre total de pixels de l’image



Déchiffrez le code. Utilisez ce tableau pour présenter vos résultats.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

*Qu’y a-t-il dans ce coffre de si précieux pour Mario ?*

**À la fin de cette activité je retiens que :**

* Une image est constituée de pixels
* La définition d’une image est égale au nombre total de pixels   
  (3 Mégapixels = 3 millions de pixels)
* Une image peut être représentée par un tableau de nombres où une case représente un pixel
* La position du pixel est repérée par son numéro de colonne (noté x) et son numéro de ligne (noté y) par rapport à une origine.
* L’origine (x=0 ; y=O) est située dans un coin de l’image qu’il faudra identifier. (En bas à gauche dans Imlab, en haut à droite dans le langage Python)

Attention : une image de 100 pixels en hauteur est représentée par un tableau de 100 lignes numérotées de 0 à 99.

* La couleur du pixel s’obtient par synthèse additive des couleurs des   
  3 sous-pixels Rouge, Vert et Bleu
* La couleur d’un sous-pixel est associée à un nombre allant de 0 à 255
* Pour obtenir un pixel gris les trois sous-pixels doivent avoir la même valeur.

|  |  |  |  |
| --- | --- | --- | --- |
| **Couleur du pixel** | **R** | **V** | **B** |
| Noir | 0 | 0 | 0 |
| Blanc | 255 | 255 | 255 |
| Gris clair | 178 | 178 | 178 |
| Rouge | 255 | 0 | 0 |
| Vert | 0 | 255 | 0 |
| Bleu | 0 | 0 | 255 |
| Jaune | 255 | 255 | 0 |
| Cyan | 0 | 255 | 255 |
| Magenta | 255 | 0 | 255 |

Pour le professeur

**Matériel au bureau :**

|  |  |
| --- | --- |
|  | Très visuel pour la synthèse additive. Allumée dès l’entrée en classe |
| Loupe binoculaire avec webcam sur oculaire vidéoprojeté | Permet d’observer et comparer les pixels des écrans de téléphones de différentes marques. Montrer en introduction de l’activité |
| Un coffre fermé par un cadenas 4 lettres (code « CLAN ») | A cette heure, n’ayant pas le cadenas je ne sais pas s’il est possible d’écrire « CLAN ». Au besoin modifier l’activité avec tout autre mot de 4 lettres  Groupes de 4 élèves, 1 mot par élève |
| Peau De Banane, Fruit, Santé, Jaune, Nature, Banane  Une peau de banane pour mettre dans le coffre | ;0)) |
| Prévoir tableau ASCII plastifié (1 pour 2) |  |

**Objectifs visés :**

Travail préparatoire au traitement de l’image en Python :

Lors de la programmation d’un algorithme de traitement d’image, l’élève sera amené à lire des programmes où il devra se « déplacer mentalement » dans le tableau de nombres par des « doubles boucles for » et à modifier par calcul les composantes R,V,B.

Le logiciel ImLab permet de **visualiser la matrice, en couleur,** des nombres de l’image et de **donner du sens aux 5 nombres (2 coordonnées et 3 composantes couleurs) associés à un pixel** **coloré**.

# Activité de coloriage numérique :

1°) Passage d’un tableau de nombres à une image colorée avec mise en forme conditionnelle.

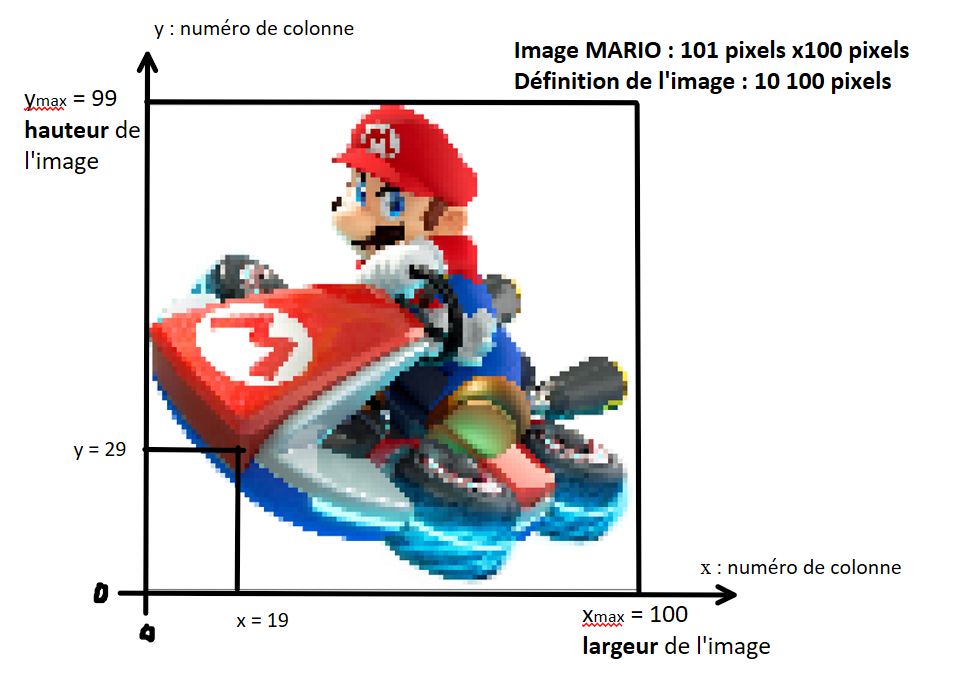
Limite du zoom arrière dans excel, les pixels sont toujours visibles mais l’image apparait. On peut s’éloigner de l’écran.

|  |  |
| --- | --- |
| Tableau de nombres AVANT coloriage | Tableau de nombres APRES coloriage |
|  |  |

2°)Le simulateur de couleur fonctionne sur téléphones portables. Les élèves peuvent scanner le QR-Code. Tester différentes couleurs. Insister sur le GRIS, les couleurs secondaires : Jaunes Cyan, Magenta.

# Activité déchiffrage du code

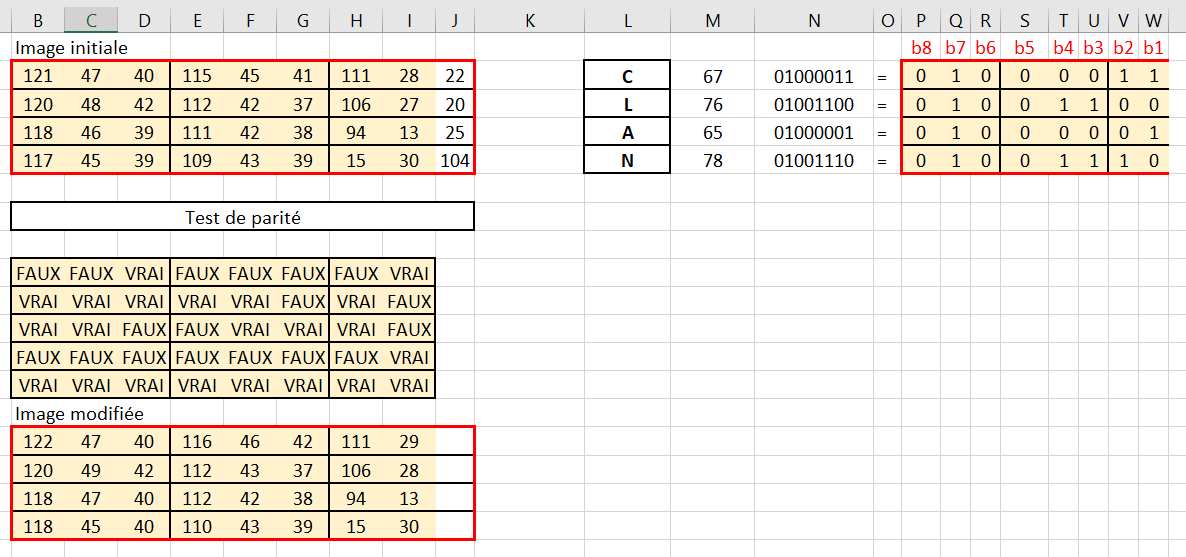
Utiliser le logiciel Imlab (en branché) ou feuille suivante plastifiée (en débranché) pour légender l’image fournie.



Utiliser Imlab (branché) ou feuille plastifiées (en débranché) pour repérer et lire les valeurs RVB des pixels où le code est caché.

Utiliser un convertisseur en ligne (connecté) ou tableau plastifié (débranché) pour convertir le nombre binaire en caractère ASCII correspondant

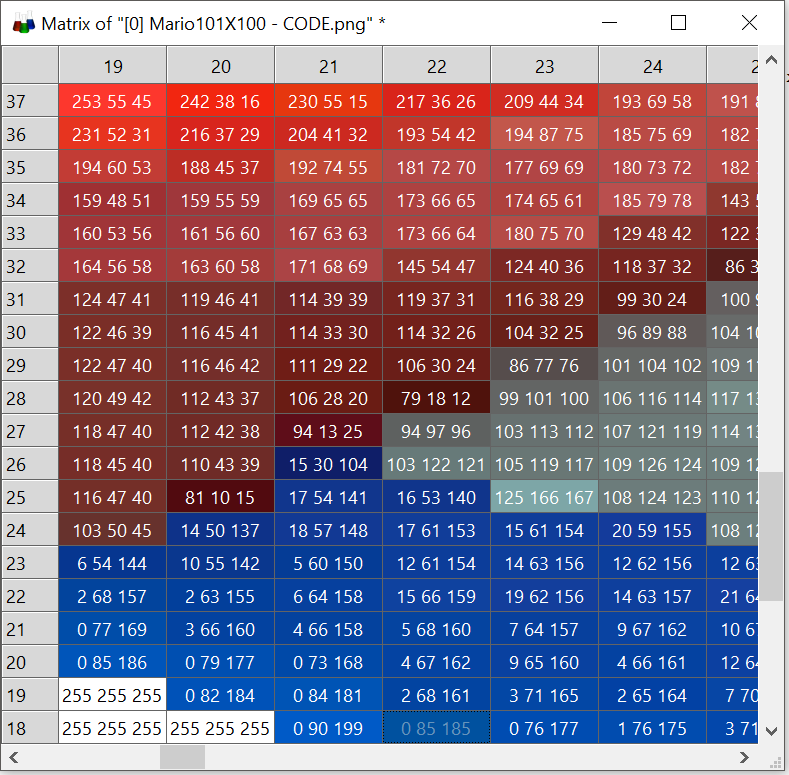
Réponse au défi



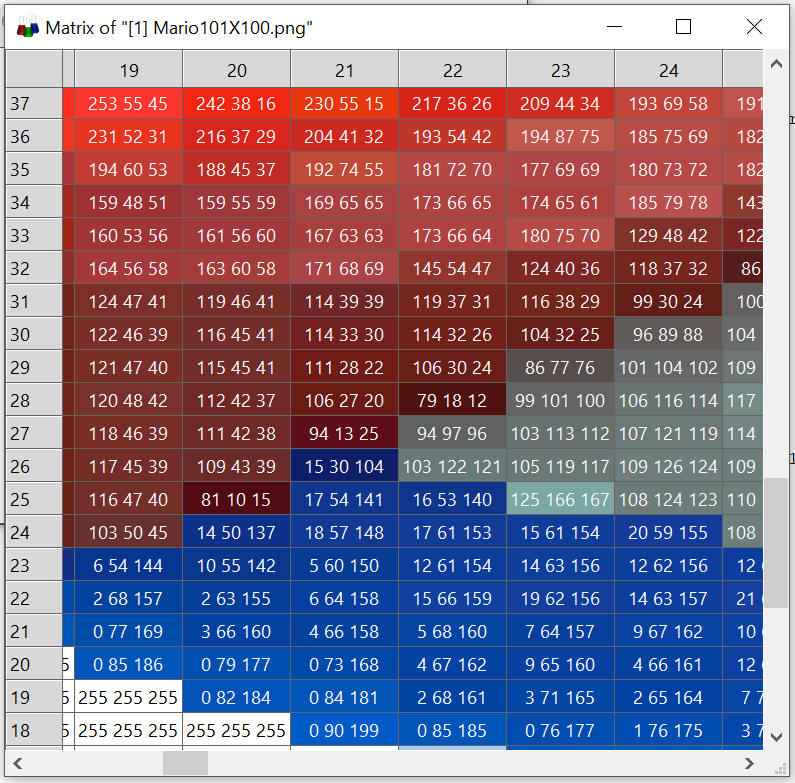
**Activité débranchée : fournir les feuilles suivantes plastifiées**



**Extrait du tableau de nombre de l’image modifiée**



**Extrait du tableau de nombre de l’image modifiée**



**Remarques diverses :**

* Droits image MARIO : <http://pngimg.com/download/30576>, CC 4.O BY-NC
* Logiciel Imlab téléchargeable sur <http://imlab.sourceforge.net/>  
  La version « no setup » ne nécessite pas d’installation : copier-coller dans mes documents et lancer le « imlab.exe »
* Le fichier Excel fourni possède 5 onglets utilisables par l’enseignant
* Les images au format png ont parfois 4 nombres dans la « case » du tableau de nombres : « R,V, B, A ». « A » gère la transparence du pixel. Ce problème n’apparait pas en jpg.
* Pour avoir une « cellule de forme carrée » dans excel, il faut que la hauteur de ligne soit 5,5 fois plus grande que la largeur de colonne
* Cette méthode de codage pose un problème avec le nombre 255. Si le bit à coder est un « 0 » il faudrait modifier et saisir 256 (impossible à coder sur un octet). Donc l’image nécessiterait un prétraitement où tous les nombres 255 serait ramenés à 254. Impossible à la main, automatiser la tâche en python par exemple. Donc veiller à choisir des pixels où aucun nombre n’est égal à 255
* Pourquoi un cadenas avec des lettres plutôt que des chiffres. Parce que le codage en binaire d’un nombre peut prêter à confusion.

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre décimal  Base 10 | En binaire  Base 2 | Caractère | En binaire |
| 3 | 11 | « 3 »  51ème caractère de la table ASCII | 00110011 |