

Préparation aux oraux CCINP n°12

Convolutions

Dans ce TP, nous manipulerons des images en C. Nous utiliserons les deux types suivants :

```
typedef struct {
    int r, g, b; // 0--255
} color;
```

```
typedef struct {
    int width;
    int height;
    color** data;
} image;
```

```
P3      # C'est un fichier en couleurs
100 50  # L'image fait 100x50 pixels
255     # Les couleurs vont jusqu'à 255

# Les pixels :
# R G B
0 255 0  # Un pixel vert
255 255 0 # Un pixel jaune (rouge + vert)
```

Une couleur sera représentée par trois composantes : rouge (*red*), vert (*green*), bleu (*blue*) ; et ces composantes sont des entiers compris entre 0 et 255. Une image sera vue comme une matrice de couleurs.

Afin d'afficher les images, nous allons utiliser le format de fichier PPM comme montré ci-dessus.

- Q1.** Écrire une fonction `image read_image(char*)` prenant en argument un nom de fichier. Cette fonction utilisera le format PPM, comme décrit dans l'encadré, pour lire l'image. Un fichier « `mario.ppm` » permet de tester cette fonction.
- Q2.** Écrire une fonction `void write_image(image, char*)` prenant en argument une image et un nom de fichier qui écrit une image.

On pourra regarder l'Annexe A pour utiliser le lecteur de fichiers PPM.

- Q3.** On souhaite réaliser un *flou* d'une image. Pour cela, pour chaque pixel, on calcule la couleur moyenne avec les 8 pixels voisins (9 valeurs au total en comptant le pixel actuel), et on le définit comme nouvelle couleur. On stocke le résultat dans une *nouvelle* image. Écrire une fonction de signature `image blur_image_seq(image, int, int)` qui réalise ce flou. (On fera attention à ne pas sortir des « bords » de l'image.)
- Q4.** À l'aide de la librairie `pthread`, on réalise cette opération en *parallèle* : chaque pixel réalise ses calculs indépendamment. On écrira une fonction `image blur_image_par(image, int, int)` qui réalise cette opération.
- Q5.** Quels dangers potentiels y a-t-il avec la concurrence ? Quelles solutions peut-on utiliser ? Ces solutions peuvent-elles engendrer des problèmes ?

Annexe A. Lecteur de fichiers PPM

On utilise la commande ci-dessous pour ouvrir une fenêtre affichant l'image PPM demandée (ici, c'est le fichier `image.ppm`).

```
python display.py image.ppm
```