

# TP1

## Formats d'images

Dans ce TD, nous allons étudier les formats de fichiers d'images PBM, qui sont largement répandus en raison de leur simplicité. Nous effectuerons quelques opérations simples sur ces formats : lecture, écriture et conversion.

### 1 Les formats "portable map"

Les formats de fichiers d'images PBM, PGM et PPM, respectivement : portable bitmap, portable grayscalemap et portable pixmap, offrent une solution simple à tout programmeur confronté au problème de la manipulation de fichiers d'images. Dans ces formats, une image est considérée comme une matrice dont les valeurs représentent l'illumination en chaque pixel de l'image : noir ou blanc (PBM), un niveau de gris (PGM) ou trois niveaux de couleurs RGB : rouge, vert, bleu (PPM).

#### Définition

Les fichiers correspondants sont constitués des éléments suivants :

1. Un "nombre magique" pour identifier les type du fichier : P1 ou P4 pour PBM, P2 ou P5 pour PGM et P3 ou p6.
2. Un caractère d'espace (blanc, TABs, CRs, LFs).
3. La largeur de l'image (valeur décimale, codée en ASCII) suivie d'un caractère d'espace, la longueur de l'image (valeur décimale, ASCII) suivie d'un caractère d'espace.
4. Uniquement pour PGM et PPM : l'intensité maximum (valeur décimale comprise entre 0 et 255, codée en ASCII) suivie d'un caractère d'espace.
5. Largeur  $\times$  hauteur nombres. Ces nombres sont soit des valeurs décimales codées en ASCII et séparées par des espaces dans le cas des formats P1, P2, P3, soit directement les valeurs binaires sur 1 ou 2 octets dans le cas des formats P4, P5, P6. Dans ce dernier cas, il n'y pas de caractères d'espace entre les valeurs.

Remarques :

Antoine.Letouzey@inrialpes.fr, Edmond.Boyer@inrialpes.fr,

- Les lignes commençant par le caractère "#" sont ignorées.
- Les lignes contiennent moins de 70 caractères.

## Exemples

```
P1
# feep.pbm
24 7
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0
0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0
0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 1 0
0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

*Fichier PBM d'une image 24×7 dont les valeurs sont codées en ASCII*

```
P2
# feep.pgm
24 7
15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 0 11 0 0 0 0 0 15 0 0 15 0
0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11 0 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0
0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

*Fichier PGM d'une image 24×7. Les valeurs d'intensité codées en ASCII sont au maximum de 15*

```
P3
# feep.ppm
4 4
15
0 0 0 0 0 0 0 0 15 0 15
0 0 0 0 15 7 0 0 0 0 0 0
0 0 0 0 0 0 0 15 7 0 0 0
15 0 15 0 0 0 0 0 0 0 0 0
```

*Fichier PPM d'une image 4×4. Les valeurs d'intensité codées en ASCII sont au maximum de 15*

## 2 Exercice 1

L'archive contient les squelettes de programmes C de conversion entre les formats présentés ainsi qu'un exemple de conversion : le programme pxm-topxm .

1. Testez le programme sur le fichier test.pbm. Quel type de conversion ce programme effectue ?
2. De quelle manière est stockée l'image dans le programme ?
3. À quoi servent les fonctions pm\_getc et pm\_getint du fichier Util.c ?
4. Quel sont les types impliqués pour manipuler les intensités ? dans le cas de valeurs décimales codées en ASCII (P1, P2, P3) ? dans le cas de valeurs binaires (P4, P5, P6) ?
5. Quelle couleur est associée à la valeur d'intensité maximum ?

## 3 Exercice 2

Il s'agit ici de compléter le programme de conversion entre les formats PGM (c'est à dire de P2 vers P5 ou de P5 vers P2) : pgmtopgm.c.

**Notes :** on utilisera le Makefile du répertoire pour la compilation (avec gmake) ainsi que les utilitaires fournies dans Util.c pour compléter ce programme.