Lab Exercises 29/04/2021: Graylevel Images

PGM format

The PGM format (defined on http://netpbm.sourceforge.net/doc/pgm.html) allows storing gray level images. The format has a slightly complex definition which requires a lot of time to be wasted checking out useless details. For this reason, we will use a simplified but compatible definition of the format, a subset of the possibilities.

The files produced with this simplified definition will be correctly interpreted by programs that support PGM (such as XnView) and we will be able to open most of the PGM files, but obviously not those that use the "strange" characteristics of the format that we ignored.

The PGM file will consist of:

1. A "magic number" to identify the type of file. The magic number of a PGM image is the sequence of two characters "P5".
2. The character `\n`, that is the Line Feed (LF), that is the character 10 (`0x0A`), that is a C new line. Beware that this cannot be the pair `\r\n`.
3. An **optional** comment identified by a "#" character followed by any sequence of characters ending with the `\n` character. During reading, it is possible to verify if the # character is present, otherwise this field is not present.
4. The width of the image (L), formatted as a sequence of ASCII characters in decimal.
5. The character `' '`, that is, a space.
6. The height of the image (A), formatted as a sequence of ASCII characters in decimal.
7. The character `\n`.
8. The string `"255"`. This indicates the maximum value that a pixel can take and, in our case, it will always be 255.
9. The character `\n`.
10. A sequence of A rows, in order from top to bottom. Each row consists of L gray levels, in order from left to right. Each gray level is a number from 0 to 255, with 0 indicating black and 255 indicating white. Each gray level is represented as a single byte.

Below there is an example of an image stored in PGM format, seen with a hexadecimal editor:

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000  50 35 0A 23 20 43 72 65 61 74 6F 20 64 61 20 43  P5.# Creato da C
00000010  6F 73 74 61 6E 74 69 6E 6F 20 47 72 61 6E 61 0A  ostantino Grana.
00000020  31 30 20 31 30 0A 32 35 35 0A E0 7F 19 1D 1C 1E  10 10.255.à.....
00000030  1F 1A 16 0D 1D 20 25 25 29 31 30 2E 30 2F 2F 30  ..... %%)10.0//0
00000040  31 35 39 36 32 2F 2B 29 31 34 32 28 22 24 20 1D  15962/+)142("$ .
00000050  1C 1C 13 10 12 14 18 16 10 13 1A 1B 1A 14 84 29  ..............„)
00000060  24 23 21 23 20 20 22 21 20 1F 1F 1E 1C 1E 20 1E  $#!# "! ..... .
00000070  2F 3F 40 3B A3 A6 A2 3D 3A 39 36 32 32 34 32 F7  /?@;£¦¢=:962242÷
00000080  F7 FC 2A 2A 1A 81 E7 15 D9 DE E0 DE 1D 1F        ÷ü**..ç.ÙÞàÞ..
```

There is also a version of the PGM useful for examining its content easily with a text editor, known as "plain" PGM. This is an entirely readable format, with all the values of point 10) (i.e. the pixels of the image) represented as a sequence of ASCII characters in decimal, separated by whitespace. The magic number of this format is "P2".

Here is the image of the previous example in plain PGM format (at the end of each two lines of the image a `\n` character was used, but a space would have also been fine):

```
P2
# Creato da Costantino Grana
10 10
255
224 127 25 29 28 30 31 26 22 13 29 32 37 37 41 49 48 46 48 47
47 48 49 53 57 54 50 47 43 41 49 52 50 40 34 36 32 29 28 28
19 16 18 20 24 22 16 19 26 27 26 20 132 41 36 35 33 35 32 32
34 33 32 31 31 30 28 30 32 30 47 63 64 59 163 166 162 61 58 57
54 50 50 52 50 247 247 252 42 42 26 129 231 21 217 222 224 222 29 31
```

Exercise 1

Write a program that generates a gray level image of 256×256 pixels, in which the first row is made of 256 zeros, the second one of 256 ones, the third one of 256 values 2 and so on. Save the image in the binary PGM and plain PGM formats, described above. Verify that the image is viewable in XnView. The images should appear as a gradient from black to white from top to bottom.

Exercise 2

Write a program that opens the `"frog_asc.pgm"` file and creates an "upside down" version, that is, the first line at the top becomes the last at the bottom of the new image, the second becomes the penultimate and so on. Save the image in the simplified PGM and plain PGM formats, described above. Verify that the image is viewable in XnView.

Exercise 3

Repeat the previous exercise with the `"frog_bin.pgm"` file.