Digital images — Part 3: Handling the ppm image format

This week, we will work on colored images: the portable pixmap format (PPM). In this format, for each color layer (red, green, blue), 0 means "maximum" and 255 "minimum". Figure 1 shows the ppm encoding of the Mona Lisa that you can download later.

⚠️ When using this file format, no line can exceed 70 characters (if you somehow need more, just open a new line). As in Python, lines that begin with a # are ignored (they are just comments).

```
P3
# "P3" means that this is a PPM file, stored in ASCII form
# "181 279" means that this is an image of size 181 x 279 (width x height)
# "255" means that the maximum value for a color layer is 255 (for white)
# Then you have the pixel information
181 279
255
66
88
34
73
95
41
...
```

Figure 1: A sample image in color, stored in the PPM file format.

Start by downloading the following files:

http://www.barsamian.am/2022-2023/S6ICTA/TP17_Joconde_Original.ppm
http://www.barsamian.am/2022-2023/S6ICTA/TP17_Images_colors.py
http://www.barsamian.am/2022-2023/S6ICTA/TP18_Amazon_Original.ppm

1 Finish work 17

This is a copy/paste of the work from last week.

1. Write the function `invert` that inverts the red, blue and green values of the image. You can safely assert that the maximum value of the values is 255. The result should be the one on the middle.

2. Write the function `grayscale` that converts each pixel (r, g, b) to a gray pixel. A gray pixel can be defined by using all three values equal. We will here choose, for each pixel, the average of the three values r, g, b. You can safely assert that the maximum value of the values is 255. The result should be the one on the right.
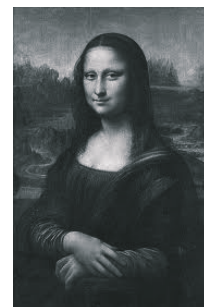
2 Thresholds

3. Write the function `blue_threshold` that converts any pixel whose blue value is less than the threshold to black, and each other pixel to white. The result should be the one on the right.