Task 2 - Blur

Suggested Task Duration: 1:30h-1:45h

Add a *new functionality* to the program developed in Task 1 to apply a blur filter to a given image. The functionality should be executable via command line, as follow:

```
$ python esss_test.py -blur <image-file>
```

The functionality from Task 1 should continue working, but the program should be changed so that it is invoked like this:

```
$ python esss_test.py -rgb <image-file>
```

The blur algorithm

The blur filter creates a new image where the color of each pixel is the average color value of its surrounding pixels in the original image (including itself, i.e., pixels inside the image will take the average of 9 values).

The example below calculates the blurred pixel for a pixel in the interior of the image.

3	16	14	14	5	10
2	7	13	0	13	3
4	14	**1**	**4**	**1**	5
13	7	**2**	**1**	**9**	3
13	13	**4**	**4**	**3**	7
14	8	6	9	10	6

Input example

		?	?	?	
		?	**3**	?	
		?	?	?	

Output example for one pixel

The output for the example above is calculated from (1 + 4 + 1 + 2 + 1 + 9 + 4 + 4 + 3) / 9 and rounded.

Border pixels should be handled by averaging only valid surrounding pixels. So, for example, the output for the upper-left pixel would be calculated by doing (3 + 16 + 2 + 7) / 4.

Notes:

- The input is a color image and the output is also a color image (the blur should be done for each image channel).

- Perform the algorithm on a copy of the image, because computing the blur by reading and writing on the same image will yield incorrect results.
- As in Task 1, the blur algorithm must be done manually without using any external library helper (with the exception of getting and setting pixels).
- If Python is chosen as the language, it *will* be slow, but this is ok: performance is not being evaluated in this test.
- Making it work correctly in the borders is essential for the evaluation process.