Question 1

\[ g = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix} \]

\[ f_1 = \begin{bmatrix} 1 & 0 & 2 \end{bmatrix} \]

\[ f_2 = \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix} \]

Q1

Compute \( g_1, h_1, g_2 \), where:

\[ g_1 = f_1 \otimes g, \quad h_1 = f_1 * g, \quad g_2 = f_2 \otimes g_1 \]

Q2

Compute a mask \( f \) such that:

\[ g_2 = f \otimes g \]

Question 2

Write a program that creates the negative image for a given image. Your program should do the following:

- Read an image.
- Display the input image.
- If the image is color, convert it into a gray level image.
- Display the gray level image.
- Compute the “negative” of the gray level image.
- Display the negative image.
- Write the negative image.

You may want to write your program by modifying the example program “GrayImages.py”.

Question 3

Write a program that reads as input a color image and produces as output an approximate gray-level image. Try the following methods for converting the three values \( r, g, b \) into the gray-level \( y \):

\[ y = g \]

\[ y = \max\{r, g, b\} \]

\[ y = \text{round}((r + g + b)/3.0) \]

\[ y = \text{round}(0.3r + 0.6g + 0.1b) \]

Try your program on several images. Which method appears to be the best among the 4 alternatives?

You may want to use the following two example programs to create your program: “ColorImages.py” and “GrayImages.py”. The first one would show you how to read a color image. The second one would show you how to write a gray level image.