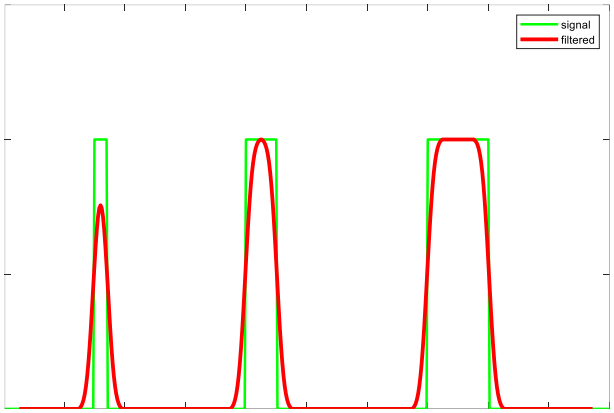
COM322 Lab 5

P1. Write a program to create a 1-D signal (similar to the class example shown below) and apply an averaging (rectangular) filter to it. The class example uses a 500-element vector for the signal (green) and a 15-element Gaussian kernel – you will use a simple rectangular filter instead of the Gaussian kernel. The output is shown in red. You do not need to implement animation. Implement the convolution yourself and do not use Matlab's conv function for this part.



P2. Write a program to blur an image by implementing the filtering algorithm given in class (do not use a MATLAB function for blurring). You can try it on the two given images: '**switzerland.jpg**' and '**Zebra_Botswana.jpg**'. Use a kernel with variable size and filter the input image with different kernel sizes. Display the original and filtered images in separate figures.

P3. Using the solution to problem 2, change the kernel to the one given below. This will sharpen the input image. Try it on '**Fruit_blurred_5x5.jpg**' and '**Zebra_Botswana_blurred_11x11.jpg**'.

-1	-1	-1
-1	9	-1
-1	-1	-1

P4. Use the solution to problem 3 to detect and display horizontal and vertical edges in separate windows and then in the same image. You can use the image called '**switzerland.jpg**'. First convert the image to grayscale using the following formula

Gray Scale Intensity = 0.299R + 0.587G + 0.114B (or use the **rgb2gray** function)

- Use the following kernel for horizontal edge detection

1	2	1
0	0	0
-1	-2	-1

- Use the transpose of the above kernel for vertical edge detection

P5. Using the same approach to image filtering in the previous three problems, use a blurring kernel to filter the image called '**noisy_man.png**' and display the result in a new figure. Now implement a median filter, apply it to the same input image and display the result in a new figure. Compare the two approaches for the given image in terms of its denoising abilities.