

## Delivery for Next Week isA:

Task Name	Post-Processing	User Input	Function Signature
<b>LinearFilter</b> function with any mask size, any origin and any postprocessing	-	<b>I, Filter, Postproc</b> <b>I</b> : input image <b>Filter</b> : 3x3 filter matrix <b>Postproc</b> : string contains 'none', 'cutoff', 'absolute'	<b>Out = LinearFilter(I, Filter, Postproc)</b>
<b>Mean Filter</b> : mean mask creation	-	<b>Rows</b> : Mask #Rows <b>Column</b> : Mask #Column	<b>M = MeanMask(Rows, Column)</b>
<b>Gaussian Filter [Option1]</b> : mask creation	-	Size, Sig <b>Size</b> : MaskSize <b>Sig</b> : Sigma	<b>M = Gauss1(Size,Sig)</b>
<b>Gaussian Filter [Option2]</b> : You have to calculate the appropriate <b>size</b> for the given sigma, check the equation in lab document for calculating the N.	-	<b>Sig</b> : Sigma	<b>M = Gauss2(Sig)</b>
<b>Laplacian Sharpening</b> : Generates 3x3 constant filter as in lab document, selects whatever you want the one with 5 or 9 at center.	Cutoff	-	<b>M = LaplacianSharp()</b>
<b>Sobel Horiz Edge, Sobel Vert Edge</b> : It generates 3x3 constant sobel masks as in lab document.	Absolute Absolute Cutoff	<b>Mask</b> : a char 'H' for horizontal, 'V' for vertical	<b>M = Sobel(Mask)</b>
This function shall do the following: 1. X = Call LinearFilter with sobel H 2. Y = Call LinearFilter with sobel V 3. Calculate the magnitude for the both above $ X + Y $			<b>G = EdgeMagnit(I)</b>