# COC202 Computer Vision
# Lab 2 – Spatial domain image processing – Solutions

1.

```
function nim = negative(im)
% return image negative

nim = 255 - im; % subtract from maximum value (255)
```

```
imshow(negative(imread('rose.tif')));

imshow(imcomplement(imread('rose.tif')));
```

2.

```
function nim = gammacor(im, g)
% perform gamma correction/enhancement

nim = im2uint8(im2double(im).^g);
```

```
imshow(gammacor(imread('temple.tif'),2));
imshow(gammacor(imread('desert.tif'),0.5));

imshow(imadjust(imread('temple.tif'),[],[],2));
imshow(imadjust(imread('desert.tif'),[],[],0.5));
```

3.

```
function nim = equalise(im)
% histogram equalisation

% obtain histogram from image
h = imhist(im);

% calculate cumulative histogram
ch = cumsum(h);

% calculate cumulative distribution function
cdf = ch/ch(end);

% generate the mapping from current to equalised intensities
mapping = floor(255*cdf);

% now, apply the mapping to generate the equalised image
nim = uint8(mapping(im+1));
```

```
imshow(equalise(imread('landscape.tif')));
imshow(histeq(imread('landscape.tif'), 256));
```

4.

```
mf = fspecial('average', 3)
imshow(imfilter(imread('rose_noisy.tif'), mf));

gf = fspecial('gaussian', 3)  % use default for sigma
imshow(imfilter(imread('rose_noisy.tif'), gf));
```

5. Mean/Gaussian filters are not suitable for removing impulse noise.

```
 % use median filter of size 3x3
imshow(medfilt2(imread('rose_noisy.tif'), [3 3]));
```

6.

```
lf = fspecial('laplacian', 0)  % create Laplacian
im = im2double(imread('rose.tif'));  % read & convert image
fim = imfilter(im, lf);  % filter image
sim = im - fim;  % subtract Laplacian image from original
imshow(sim);  % show sharpened image
```

7.

```
function nim = histmatch(ims, imt)
% histogram matching
% perform histogram matching of ims using the histogram of imt

hs = imhist(ims); % histogram of first image
ht = imhist(imt); % histogram of second image

% calculate cumulative histograms
chs = cumsum(hs);
cht = cumsum(ht);

% calculate cumulative distribution functions
cdfs = chs/chs(end);
cdft = cht/cht(end);

% generate the mapping from current to equalised intensities
for i=1:256
    diff = abs(cdfs(i) - cdft);
    [~, m] = min(diff); % get index with smallest difference ->
matching CDFt(i)
    mapping(i) = m - 1; % generate mapping entry
end

% now, apply the mapping to generate the equalised image
nim = uint8(mapping(ims+1));
```

```
% perform histogram matching on retina image

imt = imread('retim1.png');
ims = imread('retim2.png');
figure
subplot(1,3,1);
imshow(imt);
subplot(1,3,2);
imshow(ims);

% normalise each channel using histogram matching
for i=1:size(ims,3)
    % using Matlab
    % ht = imhist(imt(:,:,i));
    % nim(:,:,i) = histeq(ims(:,:,i), ht);

    % using our code
    nim(:,:,i) = histmatch(ims(:,:,i), imt(:,:,i));
end
subplot(1,3,3);
imshow(nim);
```