1

a) $y = V_0 t - 5t^2$

$y' = V_0 - 10t = 0$

$V_0 = 10t$

$t = \frac{V_0}{10}$

b) $f(x,y) = e^{\frac{-(x-x_0)^2 - (y-y_0)^2}{2\sigma^2}}$

$f'_x = e^{\frac{-(x-x_0)^2 - (y-y_0)^2}{2\sigma^2}} \left(-\frac{x-x_0}{\sigma^2}\right) = 0$

$x = x_0$

$f'_y = e^{-\frac{0-(y-y_0)^2}{2\sigma^2}} \quad \frac{y-y_0}{\sigma^2} = 0$

$y = y_0$

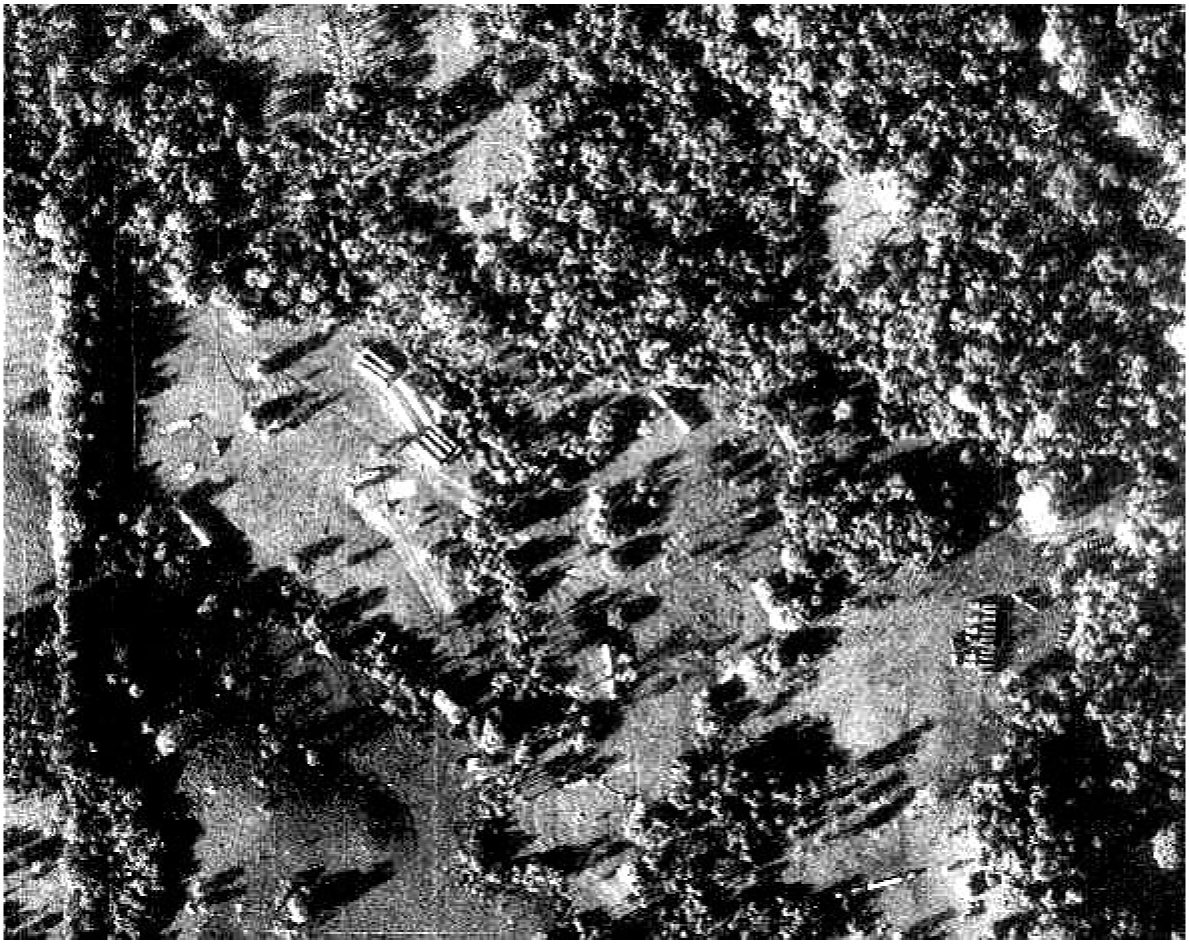
2.

```
% code:
dark = double(rgb2gray(imread('u2dark.png')));
% a
avg=mean2(dark);
min=min(min(dark));
max=max(max(dark));

%b
fixedimg = (dark-min)*256/(max-min+1);
imshow(uint8(fixedimg));

%c
contrasted = 2*(fixedimg-128) + 128;
contrasted = uint8(contrasted);
imshow(contrasted);
```

Images: fixed and contrasted (in this order).



3.code:

```
function edgedetector()
   img = double(rgb2gray(imread('buoys.jpg')));

   edges = DetectVerticalEdges(img);
   blurred_edges = BoxBlur(edges);

   figure('Name','Original Image')
   imshow(img, []);

   figure('Name','Edges')
   imshow(edges, []);

   figure('Name','Blurred Edges')
   imshow(blurred_edges, []);

end
% a. Tiny edges are waves.
function edges = DetectVerticalEdges(img)

   left=circshift(img,[0,1]);
   edges =uint8(img-left); %zeros(height, width-1);

end
% b
function blurred = BoxBlur(img)
   img = double(img);

   height = size(img, 1);
   width = size(img, 2);
   n=5; % width of the blur
   blurred = zeros(height-(n-1),width-(n-1));
     for y=1:height-(n-1)
      for x=1:width-(n-1)
         bla=img(y:y+n-1,x:x+n-1);
         blurred(y,x) = sum(sum(bla));
      end
   end

   blurred = blurred / n^2;
end
```

Images: edges and blurred (in this order).

4.

a) Rotation by 45° and then rotation by 45° is the same as rotation by 90°.

by 90°

b)

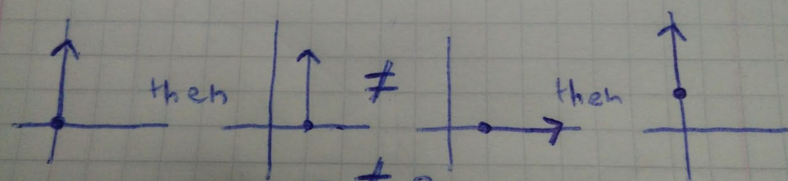
$$X_2 = A \begin{bmatrix} 1 & 0 \\ 0 & 0,5 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

c)

$$T = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/4 \\ \sqrt{2}/2 & \sqrt{2}/4 \end{bmatrix}$$

d)

$$T = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \qquad R = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$P_1 \neq P_2$$

e)

e) $RYA = B$

$YA = R^T B$

$Y = R^T B A^{-1}$

5.

```
flower = double(rgb2gray(imread('flower.bmp')));
[u,s,v] = svd(flower);
diagonal = diag(s);
ten = diagonal(1:10);
x = 1:10;
graph = plot(x,ten);
% first values are the largest

k = 10;
a = u(:, 1:k);
b = s(1:k, 1:k);
vt = transpose(v);
c = vt(1:k, :);
res = a*b*c;
figure;
imshow(uint8(res));

k = 50;
a = u(:, 1:k);
b = s(1:k, 1:k);
vt = transpose(v);
c = vt(1:k, :);
res = a*b*c;
figure;
imshow(uint8(res));

k = 100;
a = u(:, 1:k);
b = s(1:k, 1:k);
vt = transpose(v);
c = vt(1:k, :);
res = a*b*c;
figure
imshow(uint8(res));
```

300p x 300p is 90 000p. Now we have 300 x k + k x k + k x 300 so 300 x 200 + 200 x 200 + 200 x 300 = 160000

We used 1:k range so k+1:r is error

6.

6)

a) $\|A\| = \max \sqrt{(Ax)^T Ax} =$

$= \max \sqrt{(U\Sigma V^T x)^T (U\Sigma V^T x)} =$

$= \max \sqrt{x^T V \Sigma^T U^T U \Sigma V^T x} =$

$= \max \sqrt{w \Sigma^T \Sigma w^T}$

$\Sigma^T \Sigma = \begin{bmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & \vdots \\ \vdots & & \ddots \end{bmatrix}$ და $w = I$ [illegible]

[illegible]

[illegible] σ-ს [illegible].

b) $\|A\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} |a_{ij}}$

$A^T A = (U\Sigma V^T)^T (U\Sigma V^T) = V\Sigma^T U^T U \Sigma V^T =$

$= V \Sigma^T \Sigma V^T$ ანუ $\mathrm{trace}(A^T A) = \mathrm{trace}(\Sigma^T \Sigma) =$

$= \sigma_1^2 + \sigma_2^2 \cdots \sigma_p^2$