Schwartz Inequality.

Schwartz inequality states that, given two vectors f and g, in a vector space V and a scalar product we have the inequality:

$$|\langle f,g\rangle| <= \|f\| \cdot \|g\|$$

In this exercise, we apply Schwartz inequality to **Pattern recognition**.

By calculating

$$Q= \langle f,g\rangle/(\|f\| \cdot \|g\|)$$

as a **similarity measure** you can locate a given pattern (a small image) in a large image.
The pattern is represented as the matrix g, (here of odd size) whereas the matrix f represents a subpart of the image, equal in size with the pattern g.

1. Study the code for the MATLAB function exsim.
The MATLAB function **exsim** is included in this exercise. Exsim takes as **input parameters** an image I, and a pattern g of odd size. The **output** of the function is another image, the "**similarity map**" S. The value of each pixel in S is a measure of the similarity of the corresponding area of I to the pattern g. Similarity values are in the interval [0, 1]. You calculate S by the MATLAB command S=exsim(A,g), where A is the input image, g is the matching pattern, and S is the output "similarity map".

Questions:
Explain how the scalar product <f,g> is calculated (fg in the code).
Explain how the norms ||f|| and ||g|| are calculated (norm_f and norm_g in the code). Notice that ||f|| is an image too, and that ||f||^2 is a scalar product between <f^2, ones> where ones contains only 1s.
Explain why the similarity image is in the interval [0, 1].

2. Pattern matching, synthetic image.
Create a 128 x 128 image. The image should have a 21 x 21 light square (intensity value 200) on a dark background (intensity value 40). Also create a matching pattern g as outlined below. Use the function exsim to calculate the similarity map.

At the MATLAB prompt, write:

```
A=ones(128,128).*40; %dark background, size 128 x 128
A(50:70,50:70)=ones(21,21).*200; %light square, size 21 x 21
%show the image in figure 1, and in gray
figure(1); subplot(2,2,1); imagesc(A); colormap(gray); axis image;

g=ones(21,21).*40; %pattern g
g(1:10,1:21)=ones(10,21).*200;

S=exsim(A,g); %calculate the similarity map
```

Questions:
How does the pattern g look like?
Explain the position of the white dot (=most similar position) and that of the dark dot (=least similar position) in the original image A.

Produce a new pattern which is a rotation of g by 90 degrees.
Run exsim once again. Explain the result.

3. Pattern matching, real image.
Load the image pout.tif. As the matching pattern g take a neighbourhood of the left eye.

```
B=imread('pout.tif'); %load the image
B=double(B); %datatype float
%show the image in figure1, and in gray
figure(1); colormap(gray);
subplot(2,3,1); imagesc(B); axis image;

g=B(80:100,108:128); %left eye as the matching pattern 21 x 21

D=B;
D(80:100,108:128)=ones(21,21).*150;
subplot(2,3,3); imagesc(D); axis image;%show where the pattern is located in the image

S=exsim(B,g);%calculate the similarity map
```

Questions:
Can you find the white dot (=most similar position)?
Find out the coordinates of the white dot by the MATLAB command: [r,c]=find(S==max(max(S))).
Are there some parts of the image where similarity measures not calculated reliably or uniquely ? (Hint: attempt to calculate similarity close to a border)

4. Tolerance to noise.
Add gaussian noise (zero-mean, variance=0.001) to the image pout.tif.
Do the pattern matching by running the MATLAB function exsim.

```
N=(imnoise(B/255,'gaussian',0,0.001)).*255; %add gaussian noise, mean=0, var=0.001
S=exsim(N,g); %calculate the similarity map
```

Questions:
Find out if you still get a correct matching.
Increase the noise in step of 0.001 until the matching fails.
Which variance of the gaussian noise is tolerated for a correct matching?