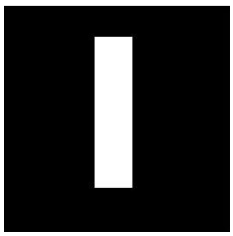
- The fast Fourier transform (FFT) is a fast algorithm for computing the discrete Fourier transform.
- MATLAB has three functions to compute the DFT:
  - `fft` -for one dimension
  - `fft2` -for two dimensions
  - `fftn` -for n dimensions
- MATLAB has three functions that compute the inverse DFT:
  - `ifft`
  - `ifft2`
  - `ifftn`

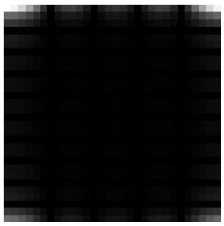
1. Create an image with a white rectangle and black background.

```
f=zeros(30,30);
f(5:24,13:17)=1;
imshow(f,'InitialMagnification','fit')
```



2. Calculate the DFT. Notice how there are real and imaginary parts to `F`. You must use `abs` to compute the magnitude (square root of the sum of the squares of the real and imaginary parts).

```
F=fft2(f);
F2=abs(F);
figure, imshow(F2,[],'InitialMagnification','fit')
```

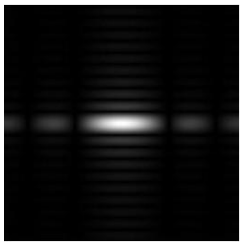


3. To create a finer sampling of the Fourier transform, you can add zero padding to `f` when computing its DFT

```
F=fft2(f, 256,256);
F2=abs(F);
figure, imshow(F2, [])
```

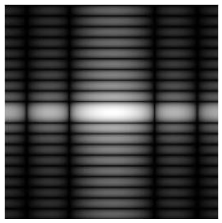
4. The zero-frequency coefficient is displayed in the upper left hand corner. To display it in the center, you can use the function `fftshift`.

```
F2=fftshift(F);
F2=abs(F2);
figure,imshow(F2,[])
```



5. To brighten the display, you can use a `log` function

```
F2=log(1+F2);
figure,imshow(F2,[])
```



To get the results shown in the last image of the table, you can also combine MATLAB calls as in:

```
f=zeros(30,30);
f(5:24,13:17)=1;
F=fft2(f, 256,256);
F2=fftshift(F);
figure,imshow(log(1+abs(F2)),[])
```

Notice in these calls to `imshow`, the second argument is empty square brackets []. This maps the minimum value in the image to black and the maximum value in the image to white.