4.4

```
%Loading in signals and plotting them in fig6.
load('fourier_oefensignalen');
figure(6)
hold on
for i = 1:6
    istr = int2str(i);
    t = eval("t" + istr);
    y = eval("y" + istr);
    plot(t,y);
end
title('fig6')
xlabel('time (s)')
ylabel('amplitude')
hold off

%{
A See fig6.
B For some signals you can allready say if they
have a very high promonent frequency or a low one.
However its very hard to see (especially in high
frequency signals). Looking at (fig6) we can
estimate about 3 periods for y2 and y6 per 10 sec so a dominant
frequency of 3/10 0.3 Hz. The other singals have a to high
frequency to be able to count peaks.

C
y1 Fs = 1/0.01 = 100Hz
y2 Fs = 1/0.2 = 5Hz
y3 Fs = 1/1 = 1Hz
y4 Fs = 1/0.05 = 20Hz
y5 Fs = 1/0.05 = 20Hz
y6 Fs = 1/0.2 = 5Hz
%}

%See advFourTrans for getting fs, n and f and the awnsers to d and e.

%F Plotting the frequency:
figure(7)
title('Fig7')
for i = 1:6
    subplot(2,3,i)
    istr = int2str(i);
    t = eval("t" + istr);
    y = eval("y" + istr);

    [Yamp, Y, f] = advFourTrans(t,y);

    plot(f,Yamp);
    title("y" + istr)
    xlabel('frequency (Hz)')
    ylabel('Amplitude')
end
%{
F Looking at fig7 we can see that for y6 the dominating frequency was
indeed around 0.3Hz, however for y2 it seems that the wave was a
combinations of low frequencies between 0 and 1 with
peaks at 0.2, 0.5 and 1

G For y1, y3, y4, y5 and y6 this is the case.
This mean sthat the a0 is not zero (e.g. a0 * a1*cos(~~~)...).

H For y4 looking at fig7 we can see that there is pretty much random
 noice
in the frequency domain aswel. For y5 you can cleary see a peak at
 1.5Hz,
zooming in on fig6 this is not really vissible in the time domain.

I A longer signal results in a higher resolution over the frequency
 axes,
in the case of y2 and y6 this means that features more difiend by
 curves in
y6 are represented as peaks in y2.
%}
```

```
Error using evalin
Undefined function or variable 'As44'.
```

4.5

```
%{
A To capture the a wave of 2.5Hz in the frequency domain precicely we
 need
a delta-f that is a divider of 2.5Hz.
So delta-f = fs/n, resulting in the following delta-f values:
20/100 = 0.2Hz
20/160 = 0.125Hz
20/250 = 0.08Hz
20/(20*1) = 1Hz
20/(20*2.4) = 0.05Hz
20/(20*5) = 0.025Hz
1, 0.2 and 0.08 will result in leakage because they cannot divide
 2.5hz in
equal whole steps.
0.125, 0.05 and 0.025 will result in an acurate representation in the
frequency domain sinse they can devide 2.5Hz in equal steps.

B At 50Hz,
50/(2*50) = 0.5Hz
50/(5*50) = 0.2Hz
They will both result in leakage.
At 20Hz,
20/(2*20) = 0.5Hz
20/(5*20) = 0.2Hz
```

```
Again they will both result in leakage.
%}
```

Published with MATLAB® R2017b