Matlab 4 project hints

Part 1: filter design by frequency sampling

Setting up the frequency vector correctly can be a little tricky. Matlab defines DFT frequencies on the range [0, 2pi], while typically textbooks work on the range [-pi, pi] (actually most FFT algorithms follow the same convention as Matlab).

Let's say we want to define L points, where L is an odd number. The first frequency bin is DC (0 Hz). If we define a normalized version of the continuous frequency, F = f/fs (where f and fs are in Hz), then we expect F to range between 0 and 1. As an example, consider L = 9. In terms of our usual conventions, the "counting" for the frequency bins is as follows:

0	ΔF	$2\Delta F$	$3\Delta F$	$4\Delta F$	$-4\Delta F$	$-3\Delta F$	$-2\Delta F$	$-\Delta F$

Suggestion: to make it clearer how Matlab handles this, generate a 9 point signal and look at its FFT, using the commands:

```
>>  x = rand(1,9) % get real-valued random signal

>> fft(x) % now FFT it
```

If you look at the FFT output, you should see a) the outputs follow the pattern shown above, and b) the positive and negative frequencies are conjugate symmetric, as expected.

Thus if we want to define a 9 -point lowpass filter that goes up to, say $2\Delta F$, the vector of magnitudes would be:

```
Hmag = [1 1 1 0 0 0 0 1 1];
```

Note there is an additional '1' at the start due to the DC value.

To make the filter causal, we want to apply a time shift of M = (L-1)/2. This gives a phase shift of

phase(n) = exp(j ω(n) M)

To find the frequency ω(n), there are several options. We could just use the tabulated frequencies from the table above, so the radian frequencies would be:

0	$2\pi\ \Delta F$	$2\pi\ 2\Delta F$	$2\pi\ 3\Delta F$	$2\pi\ 4\Delta F$	$-2\pi\ 4\Delta F$	$-2\pi\ 3\Delta F$	$-2\pi\ 2\Delta F$	$-2\pi\ \Delta F$

We could then calculate a vector of phases, and use this to set up the response vector:

```
H = Hmag.*phase;
```

Alternatively we can define frequencies going up to 2 pi, for example using $\omega(n) = n\ 2pi/L$. If using this formula, it's important to remember to start at n=0 to get $\omega = 0$ for the first frequency bin.

Part 2: Parks-McClellan

Hint: reading the help files for firpmord and firpm will help quite a bit.