```
close all; clear all;
% input my transfer function data. x contains the frequency, y the magnitude of the
% transfer function.  Note that y is on a linear scale.

f = [40 60 100 200 400 600 1000 2000 4000 6000 8000 10000]';

Hd = [0.2 0.31 0.579 1.487 0.618 0.329 0.421 0.107 0.455 1.069 1.995 1.387]';

nf = 12;  % the number of discrete frequencies

% Compute the transfer function in dB's:

Hdb = 20*log10(Hd);

% plot the data with a semilog-plot

h=semilogx(f,Hdb,'LineWidth',2);
xlabel('f(Hz)','FontSize',14);
ylabel('dB','FontSize',14);
legend('Ideal Transfer Function','Location','NorthWest');
set(gca,'FontSize',14);

% In this section, it is assumed that the filter can be represented by the
% sum of two bandpass filters.  Observing the graph, it appears that there
% are resonant frequencies at f = 200 Hz, and f = 8000 Hz, which are the center
% frequencies of the two band-pass filters.  We will start out fixing these
% falues.  Subsequently, we will scan through 4 loops changing the gain and
% the Q of each of the two filters.  Based on initial estimations
% (and some trial and error), it is determined that a good range for the
% gain and Q of the first filter is 1.1<A1<1.6, and 1 < Q1 < 3.5.  For
% the second filter, 1.8 < A2 < 2.3, and 1 < Q2 < 3.5

f1 = 200;
f2 = 1000;
f3 = 8000;

A10 = 1.4;
dA1 = 0.1;
A20 = 0.2;
dA2 = 0.05;
A30 = 1.8;
dA3 = 0.1;
Q10 = 1.5;
dQ1 = 0.5;
Q20 = 12;
dQ2 = 0.5;
Q30 = 1;
dQ3 = 0.5;


%  Next, write the "for" loops to sweep through the gains and Q's.  At the inner-
```

```matlab
% most loop, compute the mean and the variance.  Then, determine if the variance
% is less than the current minimum value.  IF it is, update the minimum variance
% and store the current values of A1, Q1, A2, and Q2.

svm = 9999;

for i1=1:6
   A1=A10+dA1*(i1-1);
   for i2=1:6
      A2=A20+dA2*(i2-1);
      for j1=1:6
         Q1=Q10+dQ1*(j1-1);
         for j2=1:6
            Q2=Q20+dQ2*(j2-1);
            for i3=1:6
                A3=A30+dA3*(i3-1);
                for j3=1:6
                    Q3=Q30+dQ3*(j3-1);
                    u=0;
                    for k1=1:nf
                       ht=hbp(f(k1),A1,f1,Q1)+hbp(f(k1),A2,f2,Q2)...
                         +hbp(f(k1),A3,f3,Q3);
                       htm = 20*log10(abs(ht));
                       u = u+(htm-Hdb(k1));
                    end
                    u = u/nf;
                    sv = 0;
                    for k2=1:nf
                      ht=hbp(f(k2),A1,f1,Q1)+hbp(f(k2),A2,f2,Q2)...
                         +hbp(f(k2),A3,f3,Q3);
                      htm = 20*log10(abs(ht));
                      sv = sv+(htm-Hdb(k2)-u)^2;
                   end
                   sv = sv/nf;
                   if sv < svm
                      a1m = A1;
                      a2m = A2;
                      a3m = A3;
                      q1m = Q1;
                      q2m = Q2;
                      q3m = Q3;
                      svm = sv;
                      um = u;
                   end
                end
            end
         end
      end
   end
end
```

```
% The above loop computed the gains and Q's that lead to the minimum
% variance error within the specified range of all the values (note that
% these values had to be played with so that a minimum fell within this range).
% Next, construct a vector with the new design values

hdsgn = zeros(nf,1);
for i1=1:nf
   hdsgn(i1)=20*log10(abs(hbp(f(i1),a1m,f1,q1m)+hbp(f(i1),a2m,f2,q2m)...
                          +hbp(f(i1),A3,f3,Q3)));
end

% plot out the new design (hdsgn) and the ideal design (Hdb) versus f:

figure(2);
h2=semilogx(f,Hdb,f,hdsgn,'LineWidth',2);
xlabel('f(Hz)','FontSize',14);
ylabel('dB','FontSize',14);
legend('Ideal Transfer Fn','Optimized Design','Location','SouthWest');
set(gca,'FontSize',14);

%finally, print out the values of the A1, A2, Q1, and Q2 that lead to the
% minimum variance error within the specified range of values.  Also print
% out the variance error and the mean error for these values.

a1m,a2m,a3m,q1m,q2m,q3m,svm,um
```