```
function  hh = dtmfdesign(fcent, L, fs)
%DTMFDESIGN
%     hh = dtmfdesign(fcent, L, fs)
%       returns a matrix where each column is the
%       impulse response of a BPF, one for each frequency
%  fcent = vector of center frequencies
%      L = length of FIR bandpass filters
%     fs = sampling freq
%
% The BPFs must be scaled so that the maximum magnitude
% of the frequency response is equal to one.
%==========================================
% [697;770;852;941;1209;1336;1477;1633]; list of centre frequencies

%%%% add your lines below to complete the code

% Stephen's work begins
% I'll design it with some flexibility and allow for different center
% frequencies and length of fcent

% setup matrix hh (for ease of calculations)
hh = zeros(L+1, length(fcent));

% run once for each center frequency
for i=1:1:length(fcent)
    % select frequency - this will run through every center freq
    fb = fcent(i);

    % setup input (cos(2pi*fb/fs*n))
    nn = 0:999; % time indices
    idealFilter = cos((2*pi*fb/fs)*nn); % 'ideal' filter (only 1-99)
    hamWindow = L+1; % hamming window
    windowedFilter = idealFilter(1:hamWindow);

    % calculate beta and apply
    [H, W] = freqz(windowedFilter, 1, 4096);
    maxVal = max(abs(H)); % finds maximum amplitude
    Beta = 1/maxVal; % gets beta
    finalFilter = Beta * windowedFilter; % normalise filter

    % add filter coefficients to hh
    hh(:, i) = finalFilter(:);
end
```

Published with MATLAB® R2023a