A simple scrambling circuit for voice communications works as follows. Consider the frequency band from 0 to 4 kHz. It is a known fact that the overwhelming majority of the power spectrum of human voice is concentrated in this frequency band. One way of scrambling this frequency band is to subdivide it into 4 equal sub-bands and interchange the sub-bands according to some pre-determined key. For example, let sub-band A correspond to frequencies between 0 and 1 kHz. Then, sub-band B corresponds to frequencies between 1 and 2 kHz, sub-band C corresponds to frequencies between 2 and 3 kHz, and sub-band D corresponds to frequencies between 3 and 4 kHz. The original order of the sub-bands is ABCD. A simple scrambling technique is to interchange this order, i.e. reorder the sub-bands to BCDA or DCBA or CABD or any other pre-determined order. Call the resulting signal the scrambled signal. This scrambled signal is not comprehensible unless you know the key and can rearrange the sub-bands back into the original order.

The goal of this problem is for you to design a MATLAB program that will descramble a given voice signal. You may obtain the scrambled signal ('scramble.wav') from the course web site. To transform the voice signal into frequency domain, you should use the MATLAB command `fft`; to rearrange the bands you should take the first half of the transform (beware of off-by-one), and rearrange it using the key given below; after that, use `ifft` to transform the signal back to time domain. You can play any sound clip in MATLAB using the command `sound`; remember to provide the sampling rate, and don't do it during class (`sound` doesn't respect Ctrl-c).

It has been scrambled in the above-described fashion by rearranging the bands ABCD into CBDA. Descramble this signal and transcribe the first and last 5 words. Turn in your code as well as your transcription.