The purpose of this assignment is to get further practice using interrupts, particularly concurrent interrupts.

A musical “note” is a waveform that has a certain fundamental frequency. For example, the “A” above middle C on a piano has a frequency of 440 Hz, and middle C itself is approximately 261 Hz. On an equally tempered scale, each of the twelve notes is related to each other by a factor of $2^{1/12}$. A note that is an octave above another note is double the frequency of the lower note.

A perfect sine wave consists only of the fundamental frequency, and sounds very “pure” or mellow, like a flute. Most waveforms consist of a fundamental and harmonics or overtones, which are additional frequencies above the fundamental. For instance, a square wave consists of the fundamental and multiples of the fundamental in decreasing amplitude. The distinctive sound of a violin is mostly a sawtooth wave (created as the rosin on the bow catches the string, then releases it), with other overtones (called dissonant overtones, due to the fact that they are not multiples of the fundamental) created by resonances in the wood.

For this assignment, you are to use interrupts to generate the notes of a musical scale, while at the same time using another interrupt to update a count. We will generate square waves by toggling a bit within a port on and off. Specifically, your program should play the notes from middle C to the next higher C, each of approximately one second in duration. These notes should be playing at the same time that an 8-bit binary count is being displayed in the LED’s, updating at approximately 1/4 second.

As before, determine the code sizes, and estimate the percentages of time your ISR’s and “main” program takes of the total execution time. Also, for each note, estimate its accuracy - that is, how close is the note your program plays to the calculated frequency for the note.