CMPSCI 119
LAB #4 – DTMF Tones
Professor William T. Verts

The goal of this Python program is to generate tones for dialing a Touchtone™ telephone. Indeed, placing the handset of a telephone near the speaker of your computer as it is playing the tones will cause the number to be dialed! The tones are in what is called Dual-Tone Multi-Frequency (DTMF) and each dialed digit is the combination of a pair of sine waves of certain frequencies, where none of the defined frequencies are harmonics of any of the other frequencies.

See page 331 of the Companion for a description of the frequencies used in DTMF. Each digit is a combination of a row frequency (697, 770, 852, or 941 Hz) and a column frequency (1209, 1336, 1447, or 1633 Hz). For example, the digit "1" is a combination of two sine waves, one at 697 Hz and the other at 1209 Hz.

On page 327 of the Companion are two functions that generate sound in JES. One is called **`SineWave`** (synthesize a single sine wave sound) and the other is called **`DualSineWave`** (synthesize a blend of two sine waves). We will need the **`DualSineWave`** function.

Part 1

Create a new Python program called **`DTMF.py`** and type in the **`DualSineWave`** function from page 327. Note that the function header is spread across five lines, the first four terminated by \ continuation characters. This was done to fit the code on the printed page. You may type in the function header all on one line as you are used to doing.

Once the function is typed in and successfully loaded, test it by typing in the black interactive area:

```
play(DualSineWave(697,1209))
```

You should hear the DTMF for the digit "1" play for one second. You may test duration and volume by putting in additional parameters. For example:

```
play(DualSineWave(697,1209,2,8000))
```

will play the digit "1" for two seconds at much lower volume.

Part 2

Write a **`Main`** function that allows the user to type in a telephone number (with or without special characters such as dashes or spaces or parentheses or anything else). For entering the telephone number, please use the **`requestString`** function from JES instead of the **`raw_input`** from Python. Your function then synthesizes and plays each digit, with short pauses in between each character, and longer pauses without making a sound for any character that is not a valid DTMF digit.

Implement the data structure as a ***dictionary*** called **`Tones`** that maps valid digit characters onto a record containing two frequencies. For example, **`Tones["1"]`** would return **`[697,1209]`** as its result, the two DTMF frequencies that are used to play the "1" digit. Put into the dictionary all 16 defined keys and their respective frequency pairs.

You will need to import the **`time`** library for determining the lengths of pauses. Each valid digit must play for 300ms followed by a 100ms pause, and for any invalid digit (such as a dash, a space, or parentheses) there must be a 400ms pause.

When playing the tones, use the JES function **`blockingPlay`** instead of the **`play`** function; if you use **`play`** then all the tones will overlap each other, but **`blockingPlay`** does not start a new tone until the old one has finished playing.

Submit this assignment as Lab #4. The graders will run your program by entering the string **`123A456B789C*0#D`** and will listen to make sure the sixteen tone sequence is correct. They will also type in **`1 (413) 555-1212`**, including spaces, dashes, and parentheses, to make sure that non-digits are handled correctly.