Programming Exam 1

Date/Time: 2023.03.14 08:10 – 09:00

(程式檔命名學號_quiz1.py，上傳至 Moodle PE1 上傳區)

Problem: Music Note Frequency

One of the oldest problems of music is how to map the notes of a musical piece to a set of audio frequencies. There are various "tuning approaches" that state slightly different ways of assigning notes to a particular frequency. This problem will require that you write a problem that does one particular kind of this mapping.

First, we must define some of musical note notations. The one common representation is the *octpch* note notation. This notation represents each note as a number pair, where the first number indicates which *octave* (八度) the note belongs to, and the second number is the position (or called *pitch class* (音高), *Pc*) that the note locates within the octave. There are 12 *semitones* (半音) within each octave on the keyboard, as listed in this table:

Note	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
Pc	0	1	2	3	4	5	6	7	8	9	10	11

Octpch representations are often written in decimal format. For example, 5.9 is the 5th octave, the 9th semitone. We must map sound frequencies to this representation. We start by choosing a reference note. The reference note frequency mapping is that 4.9 (note **A** in the fourth octave) is 440 Hz. Every tuning mapping must insure that each note in the next higher octave has a frequency that is twice the previous octave, therefore 5.9 ⇒ 880 Hz and 3.9 ⇒ 220 Hz. Our tuning system will assume that each of the semitones within an octave is equally spaced. In other words, the distance from one semitone to the next is the same within the octave. This is called a *Tempered Scale*. The formula we use to compute the frequency F can be expressed by the following formula:

$$F = x \times 2^{\left(o+\frac{m}{12}\right)}$$

where x is a reference Hz value for a known note, o is the *difference* (sign matters) in octave between the reference note and the note in question, and m is the *difference* (sign matters) between the reference semitone and the note semitone in question. Given the *octpch* value is 0.0 (note **C** in the first octave), which is 4 octaves and 9 semitones **below** 4.9 (A of the 4th octave). The resulting frequency will be:

$$F_{0.0} = 440 \times 2^{\left(-4+\frac{-9}{12}\right)} = 440 \times 2^{-4.75} = 16.3516\ \ Hz$$

Your program will ask the user to input an *octave* and a *pitch class* (Pc), and output and print out the corresponding frequency value in Hz.

<u>Sample Input/Output</u> (以下是你的程式在執行後所必須列印出來的東西)

```
Input an octave: 3
Input a pitch class: 7
The corresponding frequency = 195.99771799087463
```

助教在批改作業時，會用 python pe1.py 來執行你的程式，你的程式在被執行後，必須先跳出 Input an octave: 讓助教可以輸入數字，接著跳出 Input a pitch class: 讓助教再次輸入數字。譬如以下截圖中第一次執行時，助教在 octave 輸入 0 與在 pitch class 輸入 0。接著才會根據輸入的 octave 與 pitch class，列印出後面 frequency 之結果。

(繳交是交 pe1.py 檔，不是交截圖)

Note: You are required to write comments (註解) for each part in your code.

```
C:\Python35\workspace>python pe1.py
Input an octave: 0
Input a pitch class: 0
The corresponding frequency =  16.351597831287414

C:\Python35\workspace>python pe1.py
Input an octave: 10
Input a pitch class: 0
The corresponding frequency =  16744.036179238312

C:\Python35\workspace>python pe1.py
Input an octave: 5
Input a pitch class: 3
The corresponding frequency =  622.2539674441618

C:\Python35\workspace>_
```