22 Automatic Tuner

12 points

Introduction

Did you know that 'Autotune' is one of the most commonly used 'special effects' in the music industry? Among different applications, Autotune is used by many singers and artist to tune their pitch when singing, even in real-time!

As you may know, sounds are acoustic waves that propagate through air. The musical notes we hear are acoustic waves of a certain (and also well known!) vibration frequency. You may know that the seven notes are: Do (C), Re (D), Mi (E), Fa (F), Sol (G), La (A), Si (B). You can check the 'standard' frequency of those notes in the following table:

	C	D	E	F	G	A	B
Reference Frequency (Hz)	261.6	293.7	329.6	349.2	392.0	440.0	493.9

Musical notes have different tones. The same note can sound high-pitched or low-pitched, while still being the same note (those notes are called the octaves). In terms of frequency, it is multiplied or divided by 2 when going to a higher or a lower pitch. See the following table, where different pitches are shown for the note A, taking 440 Hz as the reference pitch:

Octave	Factor	A
2nd High Octave	4	1760.0
1st High Octave	2	880.0
Reference Frequency (Hz)		440.0
1st Low Octave	1/2	220.0
2nd Low Octave	1/4	110.0

Note that this table shows only some of the pitches for the note A, but there are many more.

But, when we sing or play an instrument, the note we sing or play may not be perfectly pitched. That means that if you try to sing a 'La', you may be generating a sound close to the 'La' frequency, but not exactly that note. Autotune corrects it to make it sound like a perfectly pitched note!

So, in this problem, you are asked to implement a simple autotune program. Given an input frequency, output the identified note with the proper octave and also the correction needed, in Hz, to make it sound perfectly pitched.

Input

One line of a decimal positive value bounded between 20.0 and 20,000.0.

Output

The output of this problem is made by three lines:

- First line the identified note corresponding to the input frequency
- Second line the closest note with the proper octave frequency
- Third line the correction to make the input frequency sound perfectly pitched. If the input frequency is already perfectly pitched, then the output may be slightly different. Check the examples!

In case the frequency given lays just in the middle of two notes, it is preferable to increase the frequency instead of decreasing.

All values must be rounded to one decimal.

Example 1

Input

```
440.0
```

Output

```
Input note: A (440.0 Hz)
Closest note: A (440.0 Hz)
Pitch Perfect!
```

Example 2

Input

```
1245.2
```

Output

```
Input note: D (1245.2 Hz)
Closest note: D (1174.8 Hz)
Decrease frequency in 70.4 Hz
```

Example 3

Input

```
15062.2
```

Output

```
Input note: B (15062.2 Hz)
Closest note: B (15804.8 Hz)
Increase frequency in 742.6 Hz
```