4939 Musical Road

There is an unusual road in Lancaster, California-it acts as a musical instrument! The road has grooves cut across it. As a car drives over the grooves the resulting vibrations are transmitted to the car. If the vibrations are at appropriate frequencies, musical notes at standard frequencies are formed. By cutting the proper number of grooves in a road with the proper distance between them, a car driven over the road at the proper speed will literally play a tune. The road in Lancaster is supposed to play the beginning of the "William Tell Overture."

Each groove cut into the road causes one vibration as the tires cross over it. By crossing the proper number of grooves per second, a note is played. For example, A below middle C has a frequency of 220 Hz. In order to play this note for one second, the car must cross 220 grooves in that second.

There are twelve notes on the musical scale, listed below in order of increasing frequency:

C C♯/D♭ D D♯/E♭ E F F♯/G♭ G G♯/A♭ A A♯/B♭ B

The seven natural notes are represented by the upper-case letters A through G. The other five notes are referred to by using either a note letter followed by a sharp symbol (♯-meaning one note higher), or a note letter followed by a flat symbol (♭-meaning one note lower). We will use '`#`' to represent the sharp symbol and lower case '`b`' to represent the flat symbol.

An *octave* is made up of the twelve notes listed above, from C to B. The note above B is C in the next octave. A note is represented by its letter followed by an optional sharp or flat symbol and its octave number. Octaves are represented by single digits.

The frequency of a given note is a factor of $\sqrt[12]{2}$ higher than the previous note. Frequencies are determined by using the A note in octave 4 as a base. A4 has a frequency of 440 Hz. Therefore, A♯4 has a frequency of $(440 * \sqrt[12]{2})$ Hz, and A3 has a frequency that is half of A4 (220 Hz). The musical road uses octaves 2 through 5.

Each note has a duration associated with it. These durations are referred to as full, half, quarter, or eighth notes. (A half note is played for half the duration of a full note, a quarter note for a quarter of the duration of a full note, and so on.) The specifications for the musical road will give the number of quarter notes to be played per minute. The duration of a note can be increased by 50% by placing a dot (`.`) after the note-so a dotted half-note would be played for 75% as long as a full note.

Rests occur in a song when no note is playing. No grooves would be cut in the road for the duration of a rest. Rest durations are also measured in terms of whole, half, quarter, and eighth notes, and can also be dotted to increase their duration by 50%.

Your team is to write a program that will, given a target road speed along with the notes and tempo for a song, will produce the required groove counts and distance between grooves needed for each note or rest in the song.

Input

Input to your program will be a series of songs ending with the end-of-file. The input for each song will be as follows: The first line will contain the targeted road speed in miles per hour as an integer, followed by a single space and the integer tempo in quarter notes per minute. The remaining lines will contain notes to be played on the road. Each note will be given as a capital letter, an optional sharp (`#`) or flat (`b`) sign, a single digit representing the octave, a single digit representing the note duration (1, 2, 4, or 8 for full, half, quarter, or eighth note respectively), and an optional dot. Rests will be represented by the letter '`R`', a note duration digit, and an optional dot. Notes will be separated from each other by one or more spaces. Each song will end with an empty line.

Output

For each note, your program is to print a line containing the number of grooves rounded to the nearest integer and the distance between the grooves in feet and inches to the nearest 1/8 inch. For a rest, print a groove count of zero followed by the proper distance. The rounding of the groove count and distance are to occur after both values are computed.

Each output line is to be printed in the following format:

- The output line is to begin with the groove count followed by a colon.
- The distance is to be printed after the colon using the following format, where the components in square brackets are only to be printed if needed:

  [(space)(*whole feet*)']][(space)(*whole inches*)][(space)(*fractional inches*)]["]
- Omit any elements that have zero values.
- Fractions are to be printed in reduced form.
- No trailing space is to appear on an output line.
- Print an empty line at the end of each song (including the last).

If two identical notes appear in the input, print the groove count and spacing for each note — your program is not to consider the spacing between notes.

Hint: There are 5280 feet in a mile.

Sample Input

```
40 120
F44 G48 E48 F44 G44 A44 Bb48 A48

60 120
E42. D42. C42. C42 R4
E42. D42.
```

Sample Input

```
175: 2"
98: 1 3/4"
82: 2 1/8"
175: 2"
196: 1 3/4"
220: 1 5/8"
117: 1 1/2"
110: 1 5/8"

494: 3 1/4"
440: 3 5/8"
392: 4"
262: 4"
0: 44'
494: 3 1/4"
440: 3 5/8"
```