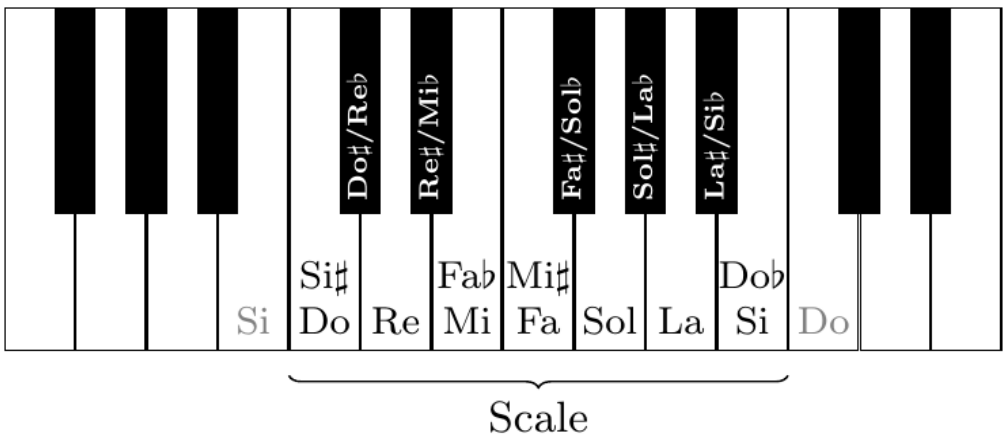
Iker is so exited about his new piano! Finally he will be able to play all the songs he likes. The seller has told him that this is a very high quality piano that will last for long, but Iker does not trust him entirely so he has decided to keep track of how many times he presses each key.

The piano keyboard has 7 octaves [Actually pianos have some more keys...] and each octave has 12 notes (7 white and 5 black keys) each one at half a tone (semitone) of the next one. The white keys correspond to the notes Do, Re, Mi, Fa, Sol, La and Si [Another different notation is C, D, E, F, G, A and B.] (and then the Do of the next octave). Do and Re are one tone away so there is a black key between them, but Mi and Fa are only a semitone away. The black keys correspond to those semitones and are named using two special symbols: sharp (♯) meaning higher in pitch by a semitone, and flat (♭) meaning lower in pitch by a semitone.

This way, the first black key in the octave corresponds to Do ♯ but also to Re ♭. Besides, Mi ♯ and Fa are the same key, and Do ♭ is the same than Si in the previous octave. What a mess!



Can you help Iker to count how many times he presses each key?

Input

The program will read from the standard input several songs, each one described with 2 lines. The first line shows the number of notes and the second line contains the specific notes. Notes are separated by blanks and they always have the same format: *name*, *accidental* (`#`, `b` or nothing), and the *octave number*. Octave 1 is the lowest pitch and octave 7 the highest.

The input ends with a song with 0 notes that must not be processed.

Output

The program will write a line for each song showing how many times each key was pressed. The keys will be sorted from the lowest to the highest pitch. The first number will correspond to the keystrokes of the lower pitch note in the song and the last number to the highest note in the song, i.e., the solution will never begin or end with zeros.

Sample Input

```
6
Do4 Do4 Re4 Do4 Fa4 Mi4
9
Mi5 Re#5 Mi5 Re#5 Mi5 Si4 Re5 Do5 La4
10
Do4 Do#4 Reb4 Re4 Re#4 Mib4 Mi4 Fab4 Mi#4 Fa4
0
```

Sample Output

```
3 0 1 0 1 1
1 0 1 1 0 1 2 3
1 2 1 2 2 2
```