Problem G
Queen of KTV II

Time limit: **3 seconds**
Memory limit: **1024 megabytes**

Problem Description

MIMI loves going to karaoke with her friends, and they all call her the "Queen of KTV." Recently, she went to sing karaoke with her crush. Back then, MIMI used the "restroom escape strategy" to avoid singing songs with high notes beyond her range. However, she realized that going to the restroom so often was even more awkward than hitting an off-pitch high note.

This time, she decided to change her plan. MIMI took the initiative to invite her crush to another karaoke session, and she came up with a brand new, perfect strategy:

- If the highest note of a song is higher than the highest note MIMI can sing, she will lower the song's key using the remote control before the song starts, until the highest note of the song becomes exactly the same as the highest note she can sing.
- If the highest note of a song is lower than the highest note MIMI can sing, she will raise the song's key before the song starts, until the highest note of the song becomes exactly the same as the highest note she can sing.
- If the highest note of a song is exactly the same as the highest note MIMI can sing, she will not adjust the key at all.

To fully showcase her high-note skills, MIMI wants the highest note of every song to match exactly with the highest note she can sing. However, since she is not very good at calculating pitch changes, she asks you to write a program that determines how many keys to raise or lower for each song to meet her requirement.

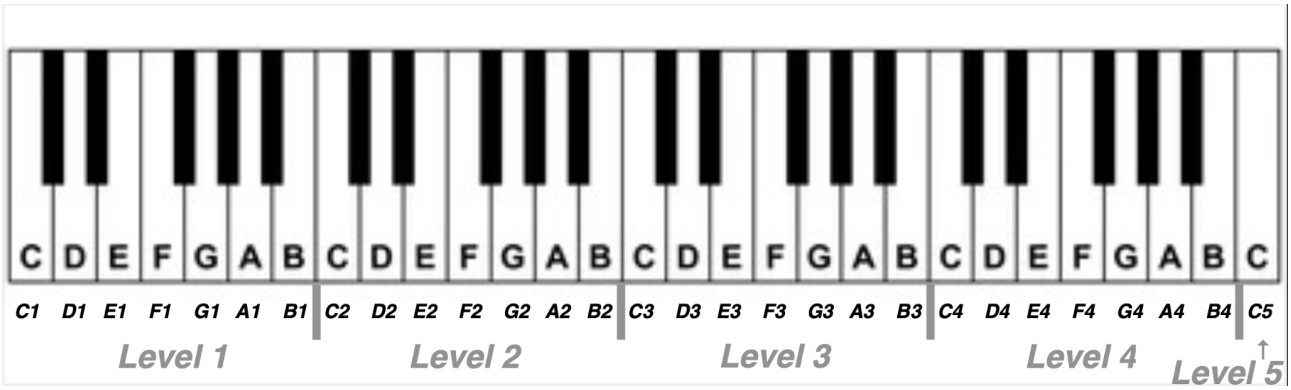
This problem guarantees that both MIMI's highest singable note and the song's highest note will always be one of `C`, `D`, `E`, `F`, `G`, `A` or `B`, with no sharps (`#`) appearing.

How to Calculate Key Changes

If there is a black key between two white keys, it means you need to go up 2 Keys (2 semitones) to reach the next note. If there is no black key in between, you only need to go up 1 Key to reach the next note.

For example, there is a black key between `C1` and `D1`, so `C1` needs to go up 2 Keys to reach `D1`. There is no black key between `E1` and `F1` or `B1` and `C2`, so going up 1 Key reaches the next note.

Every 12 Keys you go up (or down) will reach the same note in the next higher (or lower) Level(octave). For example, going up 12 Keys from `G1` will land on `G2`. (Please refer to the following figure.)



Notice that, on a piano keyboard, the keys on the right produce higher pitches, while the keys on the left produce lower pitches.

Input Format

Your program is to read from standard input. The first line contains a character $Note$ with an integer $Level$. This represents the highest note MIMI can sing on a keyboard. The second line contains a positive integer T, the number of songs to evaluate. The next T lines each contain a character $Note$ with an integer $Level$, representing the highest note of the next song in the karaoke playlist.

Output Format

Your program is to write to standard output. Output T lines. Each line contains a single integer which represents the number of Keys MIMI needs to adjust before singing the song. If MIMI needs to raise the song's key, output a **Positive integer**. If MIMI needs to lower the song's key, output a **Negative integer**. If no adjustment is needed, output `0`. Please see the sample output.

Technical Specification

- $Note \in \{C, D, E, F, G, A, B\}$
- $1 \leq Level \leq 5$
- $1 \leq T \leq 1,000$

Sample Input 1

```
G4
4
C4
B4
G2
C5
```

Sample Output 1

```
7
-4
24
-5
```