I have written two songs, one is called “烟雨” (This song was written in 2005, but did not get any serious treatment until in 2007, when I participated in Tsinghua University's Campus singers' competition) and the other one “沧浪” (This song was written especially for the final round of Tsinghua University's Campus singers' competition, April 2008).

Writing songs requires a lot of work, especially when you need to give a performance on a stage, in a serious competition. You need to polish your lyrics, melody, orchestration (The lyrics of both songs were written by Bing Song, harmony primarily by Kai Chung Tam and Zhen Shang, orchestration was done by Jun Huang, and the initial piano score was written by Qindi Li. I thank them from the bottom of my heart) again and again, and you need a lot of rehearsals with your friends (who play the piano/flute etc, or sing harmony).

Here is the photo of “my team” before the performance of “沧浪”.

You may wonder how I wrote songs. The short answer is: I didn't “write” songs, I just “select” from music fragments that automatically came into my mind. So it is essential to note down the melody quickly when it suddenly appears.

Unfortunately, I don't have absolute pitch or well-developed long-term relative pitch, so I have to stick to the movable do system (It's called “首调唱法” in Chinese. See, for example the web `http://en.wikipedia.org/wiki/Solf%C3%A8ge`).

I use 1..7 to represent do, re, mi, fa, sol, la to si, and #1 to represent #Do etc. When I'm changing the key, I'll write syllables in both keys (formatted as “$s1 = s2$” that means “this note is syllable $s1$ in the old key, which is also syllable $s2$ in the new key”), like this:

5 5 6 5 1=4 3 | 1 1 2 1 5 4 | 1=5 5 5 3 1 7=3 2 | b7 b7 6 4 5=2 1 ||

Well, I admit this is a weird way to transcript the birthday song, but ... You know what I mean, right?

However, after I wrote down the whole thing, I often find a lot of ridiculous modulations (i.e. changing key) — I just couldn't understand why I changed the key. Luckily, I don't need to know why. All I need is a small tool that rewrites the (possibly weird) melody in a “reasonable” way. By “reasonable”, I mean a good balance between the number of modulations and the number of accidentals (i.e. sharp or flats).

Given the maximum number of accidentals, find a transcript with minimal modulations.

Notes:

- Don't worry about the missing octaves information (e.g. the “5 5 5 3” part actually contains three “so”'s in two different octaves) and rhythm information. I can always remember them.
- I always explicitly written down every accidental (both before and after rewriting). For example, if I write “b7 7”, the second note is a normal “si”. Also, I never write things like “b4” or “#7” (both before and after conversion), I'll write “3” and “1” instead.
- When counting accidentals, “#5=b3” contains two accidentals, though it's only one note.
- If there are multiple solutions, print the lexicographically smallest one (the melody should be regarded as a sequence of strings). For example, “1 2 b3 ||” can be rewritten to “1=2 3 4 ||” or “1 2=3 4”. The second one is better. However, “6 7 1” is better than both, because there is no modulation at all!
- You may change the first note. For example, the optimal way to express both “b7 b7 b7 b7 ||” and “5 5 5 5 ||” is “1 1 1 1 ||”.

Input

The first line contains the number of test cases T ($T \leq 100$). Each test case contains two lines. The first line contains the maximal number of accidentals, and the second line contains the initial transcript, ending with a double barline (`||`). Adjacent symbols are separated by a single space. There will be no more than 100 notes/barlines in each transcript.

Output

For each test case, print the transcript with minimal modulations. If there are multiple solutions, print the lexicographically smallest one (don't forget the transcript is regarded as a sequence of strings, not a big string). Barlines (‘`|`’ and ‘`||`’) should be output as-is.

Background

For those who are not familiar with music, here is some information:

- There are 12 different syllables in the movable do system: 1, #1, 2, #2, 3, 4, #4, 5, #5, 6, #6, 7. The pitch interval of adjacent syllables is one semitone. You can label them 0 11, and the “pitch interval” calculation is done mod 12. For example, the pitch interval of #6 and 2 is 2-10=4 (mod 12), you can also get this by counting: #6 ->7 -> 1 -> #1 -> 2. Four semitones.
- When we hear “1 2 3”, we may also consider it “4 5 6”, because the sequence of ”adjacent pitch interval” of both melody is (2, 2). Similarly, “2 3 4 5” and “6 7 1 2” (the last “1 2” is in a higher octave) are similar, because their sequence of “adjacent pitch interval” are both (2, 1, 2). Here “similar” means “we can rewrite either melody to the other”.
- Now consider the last example, “1 #1 2 #2 3 #4”, the pitch interval sequence is (1, 1, 1, 1, 2). The rewritten transcript has one modulation, so it can be divided into two parts: “2 #2 3 4=3” and “4=3 4 5”, the pitch interval sequence of the first part is (1, 1, 1), and the second sequence is (1, 2). Note that in the first part “4=3” uses its old syllable “4”, and in the second part, “4=3” uses its new syllable, “3”. Another solution with minimal number of modulation but lexicographically larger, is “3 4 #4 5=3 4 5 ||”

Sample Input

```
5
0
5 5 6 5 1=4 3 | 1 1 2 1 5 4 |
1=5 5 5 3 1 7=3 2 | b7 b7 6 4
5=2 1 ||
1
b7 b7 b7 b7 7 ||
0
b7 b7 b7 b7 7 ||
5
6 7 6 7 7 7 ||
1
1 #1 2 #2 3 #4 ||
```

Sample Output

```
Case 1: 5 5 6 5 1 7 | 5 5 6 5 2 1 |
5 5 5 3 1 7 6 | 4 4 3 1 2 1 ||
Case 2: 1 1 1 1 #1 ||
Case 3: 3 3 3 3 4 ||
Case 4: #2 4 #2 4 4 4 ||
Case 5: 2 #2 3 4=3 4 5 ||
```