

Problem A: First Composed, Then Transposed

Problem A: First Composed, Then Transposed

The conventional western musical scale is based on 12 tones:

A A# B C C# D D# E F F# G G#



The sequence repeats indefinitely - the next note higher than G# is also called A.

Each step from one note to another in this sequence is called a half-step. The # symbol is called a "sharp" and actually means "raised a half step". Hence "A#" is "A raised a half-step". There is a similar symbol "flat" meaning "lowered a half-step" that we will render as "b" in this problem, though it is normally written as b.

Given that meaning for # and b, it is possible for notes to actually be referred to by more than one name. A# and Bb refer to the same note. B# means the same as C. Fb means the same as E.

A common task for musical arrangers is to transpose a work of music - to raise or lower the entire work by some number of half-steps in order to fit better into the range of a particular singer or musical instrument. A proper transposition will preserve the number of half-steps between any two successive notes in the work.

Input

Input consists of one or more input sets.

An input set consists of a line containing a sequence of 0 or more notes with sharp and flat marks, each note separated from the others by one or more blanks. This is followed by a line containing a single integer indicating the number of half-steps to transpose the piece (positive numbers indicating the notes should be transposed up, negative indicating it should be transposed down).

The end of input is signaled by a line consisting of the string "****".

Output

For each input set, you should print one line containing the sequence of transposed notes, each represented as in the list of 12 given at the start of this problem. Notes should be printed with a single blank space separating them.

Example

Input:

```
C# E Db G#
1
D E# D A
-1
***
```

Problem A: First Composed, Then Transposed

Output:

D F D A
C# E C# G#