

Problem DIFF: Diff

The lecture on compiler construction has attracted many students, and now their compiler implementations have to be graded. These implementations are based on a code skeleton, so that the students have less boring busywork to do and can focus on the interesting parts of a compiler.

As Jakob is too lazy to read through hundreds of lines of Java code, he is looking for a way to only examine modified parts of the source – i.e. he is looking for the *diff*¹ between the students' versions and the skeleton. Jakob remembers that in the Wikipedia article about *diff* there was a comment that the longest common subsequence could be used to construct the *diff* between two files. Two semesters ago he posed an exercise in the course on algorithms and data structures, where students had to determine the longest common subsequence of two strings, so he already has (several) implementations to determine the longest common subsequence. All he needs now is a way to compute the *diff*, given the old and new versions of the file and their longest common subsequence.

Your task is to compute exactly this (linewise) *diff*, given two versions of a file and a longest common subsequence of their lines.²

Input

Input starts with three integers o , n , and c on one line ($0 \leq c \leq o, n \leq 2048$) giving the number of lines for the **old**, **new** and longest **common subsequence** files.

Then several lines follow giving the old version (o lines), the new version (n lines), and the longest common subsequence (c lines). The two versions of the file and the longest common subsequence are separated each by one blank line.

Every line of the old version, the new version and the longest common subsequence will consist of at least one (and up to 2048) non-blank characters (followed by a newline character), and will contain neither a space (" "), nor plus sign ("+") nor a minus sign ("-").

Output

Print the *diff* between the old and the version of the file in the following format:

- Every line of the output contains one line of the old or the new version of the file, prefixed by either a space (" "), a plus ("+") or a minus sign ("-").
- If you take the output, remove lines marked with a plus sign and remove the prefix characters (space or minus sign) from the remaining lines, the result shall be equal to the *old* version.
- If you take the output, remove lines marked with a minus sign and remove the prefix characters (space or plus sign) from the remaining lines, the result shall be equal to the *new* version.
- If you take the output, remove lines marked with a plus *or* a minus sign and remove the prefix characters (space character) from the remaining lines, the result shall be equal to a longest common subsequence of the old and the new version.
- The solution is not necessarily unique. If several valid versions of the diff exist, any of them will be accepted.

¹A *diff* is a short comparison between two versions of a file, highlighting the common parts, and those parts unique to either version.

²Remember: A *subsequence* of lines of a file is a sequence of lines that you get if you delete zero or more lines from the file (*without* reordering the remaining lines). A *common* subsequence of the lines of two files is a sequence of lines that is a subsequence of both files at the same time. A *longest common subsequence* is a common subsequence that is as long as possible, i.e. every sequence of lines that is longer than that longest common subsequence can not be a common subsequence.

Sample Input 1

```
7 8 5
this_is_a_test
@_
this_line_gets_deleted
this_line_stays_the_same
this_line_gets_changed
@_
end_of_test

this_is_a_test
@_
this_line_stays_the_same
this_line_gets_changed,_like_so
@_
these_two_lines_are_added
@_
end_of_test

this_is_a_test
@_
this_line_stays_the_same
@_
end_of_test
```

Sample Output 1

```
this_is_a_test
@_
-this_line_gets_deleted
this_line_stays_the_same
-this_line_gets_changed
+this_line_gets_changed,_like_so
@_
+these_two_lines_are_added
+@_
end_of_test
```