NOI '97 P3 - File Matching

Time limit: 0.6s **Memory limit:** 16M

National Olympiad in Informatics, China, 1997

In the day-to-day operations of computers, one often needs to operate on specific files from a larger directory of files. For example: copy all files of a particular extension from the current directory to another directory, delete all files that start with the letter `a` from the current directory, and so on.

Many operating systems support the feature of regular expressions for filename matching. Simple regular expressions are made up of (case-sensitive) English letters, numbers, and the symbols `*` and `?`. `?` represents any one character, and `*` represents any 0 or more characters. For example:

- `a*b`
  - matches `acb` (`*` represents `c`)
  - matches `aabb` (`*` represents `ab`)
  - matches `asdsfdfdb` (`*` represents `sdsfdfd`)
  - matches `ab` (`*` does not represent anything)
- `a*b`
  - does not match `ac` (we lack the rightmost letter `b`)
  - does not match `bb` (we lack the leftmost letter `a`)
  - does not match `abbc` (the rightmost letter is not a `b`)
- `a?b`
  - matches `acb` (`?` represents `c`)
  - does not match `ab` (we lack the middle character)
  - does not match `accb` (`?` can only represent one character)

We would like to operate on a subset of files from the current directory. Write a program to find a regular expression that matches as many files as possible from this desired subset of files, without matching any of the files we do not want to operate on. Note that the length of the regular expression must be as short as possible. If there are multiple shortest optimally matching regular expressions, any one of them is accepted.

Input Specification

The input contains no more than 250 lines. Each line contains one file name made up of only alphabetical letters and numerical digits. The file name is case-sensitive and will be at most 8 characters long. The name will be followed by a space, and after that will be one character (either `+` or `-`). `+` after a file indicates that we would like to operate on it, and `-` after a file indicates that we would not like to operate on it.

Output Specification

The output should contain two lines. The first line should contain one integer - the number of files your optimal regular expression can match. The second line should contain the regular expression that your program has found.

Sample Input

```
EXCHANGE +
EXT +
HARDWARE +
MOUSE -
NETWORK -
```

Sample Output

```
2
E*
```

Problem translated to English by **Alex**.