

In the world of telecommunications phone calls to different phone numbers have to be charged using different rate or different billing plan. International Carrier of Phone Communications (ICPC) has an antique billing table that determines which phone call has to be charged using which billing plan.

Each international phone number has 11 digits. The billing table has  $n$  lines. Each line specifies a range of prefixes of phone numbers like '7919 - 921'. This specification means that all phone numbers starting from 7919, 7920, and 7921 match this line. A billing plan name is specified for each prefix. To determine a billing plan for a call, the table is scanned from top to bottom and the first matching line determines the billing plan. If no match is found, the phone number is invalid and no billing plan is needed. A special billing plan named 'invalid' (without quotes) is used as an alternative way to define invalid phone numbers. Some billing plans are used for quite differently looking phone numbers and their names may be specified on different lines in different places of the table.

ICPC's billing table is old and contains many entries. Some of those entries may not be even used anymore. It is very hard to figure out which phone numbers each billing plan is actually used for. The ICPC's management has reached a decision to transform this billing table into a more legible format. In this new format table consists of the lexicographically ordered list of simple prefixes (without the '-' range feature of the old format) with a billing plan name for each prefix. No prefix of this new billing table should be a prefix of any other prefix from the table. Thus, a simple dictionary lookup (binary search, for example) will be sufficient to figure out a billing plan for a given phone number. Finding all phone numbers for a given billing plan will also become quite a simple task. The number of lines in the new billing table should be minimized. Billing plan named 'invalid' should not be present in the new billing table at all, since invalid phone numbers will be denoted by absence of the corresponding prefix in the new billing table.

## Input

The input will contain several test cases, each of them as described below. Consecutive test cases are separated by a single blank line.

The first line of the input contains a single integer number  $n$  ( $1 \leq n \leq 100$ ) — the number of lines in the old billing table. The following  $n$  lines describe the old billing table with one rule on a line. Each rule contains four tokens separated by spaces — prefix  $A$ , minus sign ('-'), prefix  $B$ , and billing plan name. Prefixes contain from 1 to 11 digits each, and the billing plan name contains from 1 to 20 lower case letters.

Further, let us denote with  $|A|$  the number of digits in the prefix  $A$ . It is true that  $1 \leq |B| \leq |A| \leq 11$ . Moreover, last  $|B|$  digits of prefix  $A$  form a string that is lexicographically equal or precedes  $B$ .

Such pair of prefixes  $A$  and  $B$  matches all phone numbers with the first  $|A| - |B|$  digits matching the first digits of  $A$  and with the following  $|B|$  digits being lexicographically between the last  $|B|$  digits of  $A$  and  $B$  (inclusive).

## Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

Write to the output file a single integer number  $k$  — the minimal number of lines that the new table should contain to describe the given old billing table. Then write  $k$  lines with the lexicographically ordered new billing table. Write two tokens separated by a space on each line — the prefix and the billing plan name. Note, that the prefix in the new billing table shall contain at least one digit.

If all phone numbers are invalid (every phone number has no matching line or matches line with billing plan 'invalid') then the output file should contain just number zero.

## Sample Input

```
8
7919 - 921 cell
7921800 - 999 priv
1 - 1 usa
760 - 9 rsv
7928 - 29 rsv
7600 - 7899 spec
73 - 77 invalid
7 - 7 cis
```

## Sample Output

```
35
1 usa
70 cis
71 cis
72 cis
76 rsv
77 spec
78 spec
790 cis
7910 cis
7911 cis
7912 cis
7913 cis
7914 cis
7915 cis
7916 cis
7917 cis
7918 cis
7919 cell
7920 cell
7921 cell
7922 cis
7923 cis
7924 cis
7925 cis
7926 cis
7927 cis
7928 rsv
7929 rsv
793 cis
794 cis
795 cis
796 cis
797 cis
798 cis
799 cis
```