Optimal Expedition Pairs

Quido has a list of various events organized by CTU Travel Society. Some events in the list are jungle forest expeditions. Quido is going to take part in exactly two expeditions.
There must be at least 7 free days between the last day of the first expedition and the first day of the second expedition.
Moreover, not to interfere with the members of the other expeditions, he has to follow one additional rule:
In the free days between the first and the second Quido's expedition, there must not be a day in which some other expedition starts or ends.
All expeditions take place in one calendar year, that year is not a leap year.

The list of events has a special structure. It is a list of text lines, each line describes exactly one event. An event is an expedition if and only if the corresponding text line contains at least one correctly formed date. A correctly formed date is an unempty sequence of digits followed by a period, followed by another sequence of digits and another period. The second digit sequence is interpreted as a number of month and the first digit sequence is interpreted as a number of the day in that month. The whole date defined by this two numbers must be a date in a non-leap year. The examples of correctly formed dates are:

```
  1.1.  31.1.   1.2.  28.2.  9.9.  10.10.  11.10.  20.12.  31.12.
```

A correctly formed date can be immediately preceded only by a space character and also it can be immediately followed only by a space character.. A correctly formed date cannot be a substring of a bigger string which does not contain spaces. For example, "day16.9." is not a correctly formed date and it does not contain a correctly formed date. There is a single exception to this rule, a correctly formed date may be immediately followed by a comma and a space, like e.g. "12.8., 16.11., and similar". Even in this case, the correctly formed date may be preceded only by a space.

When the line describing the expedition contains exactly one correctly formed date the duration of the expedition is exactly one day.
When the line describing the expedition contains more than one correctly formed date the duration of the expedition is equal to the difference, measured in days, between the latest correctly formed date and the earliest correctly formed date, increased by one.
For example, the duration of an expedition described by a line with dates 31.5., 5.6., 1.6. and 29.5. is 8 days.

Quido wants to choose such pair of expeditions which satisfy the given rules and which total duration is maximum possible. There might be more pairs of such expeditions on the list, Quido calls each such pair *optimal expedition pair*.

The task

You are given a list of text lines. Find all optimal expedition pairs in the list.

Input

The first input line contains one integer N, the number of following lines in the input. Next, the list of N lines follows, each line consists of one or more words separated by spaces and it is terminated by a dot (period, full stop) symbol.
Characters in the input lines are small and capital letters of English alphabet, ('abcd...xyz', 'ABCD...XYZ'), digits ('1234567890'), comma and period (dot, full stop). Each line contains at most 40 words, separated by spaces, each word length is at most 100 characters and it contains no spaces. The label of the first line in the list is considered to be 1, the label of the second line in the list is considered to be 2, and so on, The label of the last line in the list is considered to be N.
It holds, $2 \le N \le 1000$.

Output

The first output line contains the total number of days in any pair of optimal expedition pair in the input. The next output lines contain the list of all optimal expedition pairs in the input, each pair occupies one line.

Each optimal pair P is described by six values P(1), P(2), P(3), P(4), P(5), P(6), separated by spaces. The values P(2), P(3) are the dates of the first and the last day in the first expedition in the pair, The values P(5), P(6) are the dates of the first and the last day in the second expedition in the pair. The value P(1) is the label of the input line with the description of the first expedition in the pair. The value P(4) is the label of the input line with the description of the second expedition in the pair.

The output list is sorted first by values P(2) (start day of the first expedition), then, if those values are equal, it is sorted by values P(5) (start day of the second expedition), then, if those values are equal, it is sorted by values P(1) (label of the line describing the first expedition), then, if those values are equal, it is sorted by values P(4) (label of the line describing the second expedition).
All dates in the output are correctly formed dates.
It is guaranteed that there is at least one optimal expedition pair in the input.

Example 1

Input

```
14
That year winter weather lasted till early May.
Jill left in February, she spent 8 days in Berlin.
The newly planted trees were held up by wooden frames.
There were only seventeen players in the club in August.
She said it happened on 23.12. and others supported her view.
Tom, Pat and Sue had no time in November, however.
April and May were her favourite names and 27 was her favourite number.
In August it was possible to collect 23, 24, and 25 specimens, respectively.
They intended to arrive on 23.8. or on 27.8.
We could not find any documents in the 4th drawer.
Sand was blown on the beach on 1.1. in Perth.
Numbers 11, 12 or 14 may be brought in focus in April this year.
Another example is 5 and 10 of February.
And finally, check 1 2 3 in July against 5 6 7 in other months.
```

Output

```
6
```

```
11 1.1. 1.1. 9 23.8. 27.8.
9 23.8. 27.8. 5 23.12. 23.12.
```

Example 2

Input

```
7
trip1 20.9. 24.9. is a short one.
trip1 20.8. 24.8. is a short one.
trip1 20.7. 24.7. is a short one.
trip2 20.6. 24.6. is a short one.
trip3 20.5. 24.5. is a short one.
trip4 20.4. 24.4. is a short one.
trip5 20.3. 24.3. is a short one.
```

Output

```
10
7 20.3. 24.3. 6 20.4. 24.4.
6 20.4. 24.4. 5 20.5. 24.5.
5 20.5. 24.5. 4 20.6. 24.6.
4 20.6. 24.6. 3 20.7. 24.7.
3 20.7. 24.7. 2 20.8. 24.8.
2 20.8. 24.8. 1 20.9. 24.9.
```

Example 3

Input

```
8
January February.
March 3 April.
March 3 4 April.
April 4 5 may.
32.32. 16.7. 31.31.
32.32. 18.7. July 18 31.31.
28.2. February 29.
29.2. February 28.
```

Output

```
2
7 28.2. 28.2. 5 16.7. 16.7.
```