

## 2016 Charity Begins at Home-page

Dozens of charity WWW sites have appeared recently. These sites donate advertising payments they receive to charitable organizations. But advertisers don't want to pay for people who come to the WWW site dozens of time a day, so require a list of unique visitors to the site. Some sites, such as The Hunger Site ([thehungersite.com](http://thehungersite.com)), the original "Click for Charity" site, allow only one donation per address per day. Others, such as the Meals On Wheels site ([givemeals.com](http://givemeals.com)), allow donations as frequently as every 6 hours.

You are to write a program to take a list of visits for a month and list the visitors to the site and the number of visits, both valid and invalid, they made in that month.

### Input

The input data file contains zero or more records of visitors to a charity WWW site, where each record appears on single line. Each visit will be in the following format:

*dd:hh:mm:ss:address*

where:

*dd* is an integer that represents the day of the visit, with  $01 \leq dd \leq 32$

*hh* is an integer that represents the hour of the visit, with  $00 \leq hh \leq 23$  Should it be  $00 \leq hh \leq 23$ ?)

*mm* is an integer that represents the minute of the visit, with  $00 \leq mm \leq 59$

*ss* is an integer that represents the second of the visit, with  $00 \leq ss \leq 59$

*address* is a string that represents the email address of the visitor. The address will consist of the upper and lower case letters, digits, periods ('.'), and the at sign ('@') only. It will have exactly one '@' in it (that will not be the first character) and at least one period after the '@'. The '@' and '.' will not appear next to each other. An address will not end with a '.'. There will be at most 65 characters in the address. Case is not significant in addresses, so 'a@b.c' and 'A@B.C' should be considered to be the same address.

The visits will be arranged in time order, from earliest to latest. While there may be two or more visits at a given time, there will not be two visits for the same address at the same time.

There will be at most 1000 unique addresses in the input file.

The end of input will be indicated by an entry for day 32. This line should not be processed.

### Output

Output the number of valid and invalid visits for each address in the file. A visit is valid if it is the first visit in the file for the address or if it is 6 hours or more since the most recent previous valid visit for that address. A visit is invalid if it is not valid (that is, if it is less than 6 hours since the last valid visit for that email address in the file).

Output should be grouped by top-level domain, and within a top-level domain by the second-level domain. The top-level domain of the address is the character or characters after the last period in the address. The second-level domain is the collection of letters and digits immediately before the last period.

For example, the address ‘programmer@acm.cs.gcsu.edu’ has the top-level domain edu and the second-level domain ‘gcsu’. Top-level domains should be alphabetized, then the second-level domains alphabetized under them. For each second-level domain, all the donors in that domain should be listed in ASCII order of address, followed by the number of valid donations and invalid donations.

There should be one blank line between second-level domains and two blank lines between top-level domains. Use the format in the sample output.

All email addresses in the output should be in lower case.

### Sample Input

```
01:00:24:18:joe@gcsu.edu
01:00:24:18:jane3@cobra.cs.mercer.edu
01:01:00:29:joe@gcsu.edu
01:07:00:29:JOE@GCSU.EDU
01:08:30:15:JANE3@COBRA.CS.MERCER.EDU
01:12:02:19:THOMAS@USA.EDU
02:08:30:15:jane@cobra.cs.mercer.edu
04:07:20:00:bob@ist.ucf.EDU
04:08:18:39:mickey@disney.com
04:13:19:59:bob@ist.ucf.edu
04:18:20:19:thomas@usa.EDU
05:19:19:45:rudy@survivor.cbs.tv
06:00:18:24:jane3@cobra.cs.mercer.edu
06:01:19:45:Rudy@Survivor.CBS.TV
10:00:24:18:joe@gcsu.edu
32:01:02:03:last.name@the.list
```

### Sample Output

```
TOP-LEVEL DOMAIN = com
Second level domain = disney
mickey@disney.com                1  0

TOP-LEVEL DOMAIN = edu
Second level domain = gcsu
joe@gcsu.edu                      3  1

Second level domain = mercer
jane3@cobra.cs.mercer.edu         3  0
jane@cobra.cs.mercer.edu         1  0

Second level domain = ucf
bob@ist.ucf.edu                   1  1

Second level domain = usa
thomas@usa.edu                    2  0

TOP-LEVEL DOMAIN = tv
Second level domain = cbs
```

rudy@survivor.cbs.tv

2 0