

Sorting Bank Accounts

In one of the internet banks thousands of operations are being performed every day. Since certain customers do business more actively than others, some of the bank accounts occur many times in the list of operations. Your task is to sort the bank account numbers in ascending order. If an account appears twice or more in the list, write the number of repetitions just after the account number. The format of accounts is as follows: **2** control digits, an **8**-digit code of the bank, **16** digits identifying the owner (written in groups of four digits), for example (at the end of each line there is exactly one space):

```
30 10103538 2222 1233 6160 0142
```

Banks are real-time institutions and they need FAST solutions. If you feel you can meet the challenge within a very stringent time limit, go ahead! A well designed sorting algorithm in a fast language is likely to succeed.

Input

t [the number of tests ≤ 5]
 n [the number of accounts $\leq 100\,000$]
[list of accounts]
[empty line]
[next test cases]

Output

[sorted list of accounts with the number of repeated accounts]
[empty line]
[other results]

Example

Input:

```
2
6
03 10103538 2222 1233 6160 0142
03 10103538 2222 1233 6160 0141
30 10103538 2222 1233 6160 0141
30 10103538 2222 1233 6160 0142
30 10103538 2222 1233 6160 0141
30 10103538 2222 1233 6160 0142
```

```
5
30 10103538 2222 1233 6160 0144
30 10103538 2222 1233 6160 0142
30 10103538 2222 1233 6160 0145
30 10103538 2222 1233 6160 0146
30 10103538 2222 1233 6160 0143
```

Output:

03 10103538 2222 1233 6160 0141 1
03 10103538 2222 1233 6160 0142 1
30 10103538 2222 1233 6160 0141 2
30 10103538 2222 1233 6160 0142 2

30 10103538 2222 1233 6160 0142 1
30 10103538 2222 1233 6160 0143 1
30 10103538 2222 1233 6160 0144 1
30 10103538 2222 1233 6160 0145 1
30 10103538 2222 1233 6160 0146 1