

# Amplitude Hackathon Summer '24 Problem 2 - Feature Requests

---

**Time limit:** 5.0s    **Memory limit:** 1G

---

$n$  customers have various features that they want to see implemented. Amplitude is curious to see which features are the most requested. Your job is to aggregate all the feature requests and determine, for each one, how many customers requested the given feature. On top of that, to determine what features should be prioritized for implementation, you should present them in decreasing order of the number of customers that requested them.

## Constraints

---

$$1 \leq n \leq 10^5$$

$$1 \leq k \leq 5$$

A given customer will not request the same feature more than once.

Every feature request will be a string of lowercase ASCII letters that has length at most 20.

## Input Specification

---

The first line contains a single integer,  $n$ .

$n$  lines follow. Each line starts with an integer  $k$ , the number of feature requests that the given customer has.  $k$  feature requests follow.

## Output Specification

---

Output the feature requests in sorted order. More specifically, output feature requests, one per line. A feature request should be formatted first with the name of the feature request followed by the number of customers that requested that feature. If feature  $x$  goes before feature  $y$ , either  $x$  was requested by more customers compared to  $y$ , or  $x$  and  $y$  were requested by the same number of customers and  $x$  is alphabetically before  $y$ .

## Sample Input 1

---

```
3
2 chatgptsupport anthropicsupport
1 anthropicsupport
1 fastqueries
```

## Sample Output 1

---

```
anthropicsupport 2  
chatgptsupport 1  
fastqueries 1
```

## Sample Explanation 1

---

Two customers requested anthropic support, so it is first on the list. One customer requested ChatGPT support and one customer requested fast queries, so since ChatGPT support is alphabetically earlier, it was printed first.