

## Challenge 11 - Pheasant

There has been an outage on our event feed server, so all the generation code and the main db have been lost.

Fortunately, we have some backups of certain tables, but they are not complete. We've managed to recover all the user feed data, which is in the following format:

```
user_id timestamp_newest event_id user_id
timestamp_newer event_id ... user_id timestamp_oldest
event_id
```

This data is encrypted with AES in ECB mode with a 32-byte key, with only uppercase and lowercase Latin alphabet characters.

Unfortunately, the key field of the backup table was defined with a 29 character length, so the last three digits of each key have been truncated and cannot be recovered.

We also have the last event time data for each user, which is just the `timestamp_newest`, and it's unencrypted.

We have not been able to recover our old generation algorithm, so we are asking you to write a new one. As you may know, our event feed is generated by ordering all friend events according to their timestamp in descending order, so you must generate a feed just like the one for Twitter. Customers don't need your entire feed, so there is an events number limit as an input, along with your friends list.

```
...
( '>
/
/ |
/ ,_//
~~~~~//'- '~~~~~~
~~~~~//~~~~~
```

The feeds and the last times are available in the following file: [feeds.tar.gz](https://feeds.tar.gz)

## Input

The input contains a list of cases to solve, one per line.

Each line (or case) consists in the number of events we need  $N$  ( $1 \leq N \leq 300$ ), followed by the list of user id's of your friends and the keys of each of them with at most 3000 users in that list.

## Output

You have to print the solution for each case in a separated line.

For each case, the output should be the list of the event ids requested, separated by spaces. If users don't have enough events to reach the limit, simply return the events they have. However, keep in mind that they must be sorted by timestamp in descending order.

## Input format

```
events_number; user_1_id,user_1_key;  
user_2_id,user_2_key; ...
```

## Output format

```
event_id_1 event_id_2 event_id_3...
```

## Sample input

```
7; 17023,hNMqffpeMSqUqNfbvSDImDRQmtSbU;  
57970,ZWIiwjrkhh1EJcSn0uCANXQqexSxC;  
88916,aMLtuoZ0wrydHyXBJCexjkzeBGKvF;  
94293,CItCenDzXLglVHJqZSrkdGwatYnrg;  
21533,mtGlWjHkfkthdDXzRCfsjtVssCND0;  
89112,fXMAPXYluCCUHpef0rNTvIOUtyrJQ;  
97879,rFUvNLbFhqeqglWWYztSgRXVoHlcn  
20; 24969,MjFhBiAdhApDjxPUPTAYEmkipNdjl;  
40446,GrkLwoykYYLDKhLJmuDxEIA00pQwa;  
52107,CbsYzqgwTHIfMlPKKfeekviPYAmRE;  
51797,PzXWkkJZrbyiBlACjaVlRNJtCStQZ;  
18991,yzcVwTbZhPZNKEbdEusitxTfiGPIg;  
53847,bcothVZiQBNRhmyEbpepjdPuYlkYq;  
93538,h1IuekdFLMwOfXqoXNZhUEGQhcfrs;  
101158,PVjneTkBAPTuvePutVpUcmsjsyVyV;
```

78771,tHCREWafBRhzTXbhNleVkeSgWLVYW;  
55239,gTPVujaEJFIetgVbfXOVLSltKpVRw;  
86644,CiugbjSOnyPmMoXAHAXEQCvweErbJ;  
24416,ZlsDCLXYUaDTpnybiJOAH0thWYdrL;  
94329,IrMYwnrdwqqCQuAnSHQHhHwXGNzHM;  
83021,ldSSaLpogUfneQXtxdkXtWWbHIPjv;  
64346,zskHxtWYdlxKagqxZOY0YlylGATCH;  
56253,RwJDSFjUCadVLnZpALcJLKZZblWke;  
74977,ccuOqhjoCDSUKamYEsKuXdNqqlhjk;  
103658,EKjXiUZcytVrLAqDmIjZsqhjMMgLG;  
43021,UYvTkcqhmZSHaknsQpNTgkggscJtX;  
101157,ybQJ0oPdPaophiGhTJpqfkdHSzrNV;  
77118,dWwpXTxuomGKVhRnKfVgOucoLOPwq;  
31585,TZrYqgkEqakJQntAOGPMOqWrGrzBX;  
101618,mzKJmsDhrgMUZaSloGePrUmljnRzu;  
102425,YrftsSyQlbBvQihtPZjEllmQIOMaw;  
66314,oRmtDeavEYfoqHdViFMPKsppzfdYI;  
95219,ompwtQaKZCfRZWAmoNMImWMeIEzCn;  
93822,xWzvFiDomRbQbzuqHcpLBA1KlKyoY;  
28868,vPJAdmuPeQQoPTmTRyFuhefsmHbTM;  
80171,QLJzvVAwBlCrYGhBYMFjfUtPcrmmB;  
42535,OrqGLsKETQPCSzPbRqoQKouWwsyug  
5; 63321,YwsSygLoCYQvyrnMrqmZZigmwMdyq;  
79648,CsBZurtvHMsdqIWcsUsgEldfbhYSR;  
50262,dsJIQeLJNhVeeBJEwIfGbpERHSelS;  
89350,fRaJaPBUiNkvSljalovDkpytCsFYS;  
43756,IaJoLdozmQVcVxCReEQlFGuPbtpkj

## Sample output

2379660 2356841 2356830 2323784 2293275 2268941  
2240228  
2524521 2524518 2524517 2524514 2524509 2524497  
2517572 2517558 2517551 2510502 2510492 2510491  
2510486 2510479 2510478 2510473 2510469 2510458  
2503510 2503509  
2519439 2492300 2487373 2472350 2424300

## Submit & test your code

---

To test and submit code we provide a set of tools to help you. Download [contest tools](#) if you haven't already done that. You will then be able to test and submit your solution to this challenge with the challenge token.

Challenge token: iTMed1pXbt4o3ZTLRbwS

### To test your program

```
./test_challenge iTMed1pXbt4o3ZTLRbwS path/program
```

A nice output will tell you if your program got the right solution or not. You can try as many times as you need.

### To submit your program to the challenge

```
./submit_challenge iTMed1pXbt4o3ZTLRbwS  
path/source_pkg.tgz path/program
```

Note that you first need to solve the test phase before submitting the code. During the submit phase, in some problems, we might give your program harder questions, so try to make your program failsafe.

**Important:** In this phase, you must provide the source code used to solve the challenge and, if necessary, a brief explanation of how you solved it.

Remember **you can only submit once!** Once your solution is submitted you won't be able to amend it to fix issues or make it faster, so please be sure your solution is finished before submitting it.

If you have any doubts, please check the [info section](#).

## Go ahead

### I'm done! :)

Once you have submitted your code, hit refresh and continue to next challenge.

### I'm stuck! :(

Be sure you follow the [Tuenti Engineering](#) twitter for updates and possible hints during the contest.

If this challenge is too hard and you are blocked, you will be able to skip it after two hours. Note that **you won't be able to complete it later**, and you have a limited number of challenges to skip.

Finally, if you run out of skips but are still really stuck with one problem, you will be able to skip it after 24 hours.

### Challenge status:

Test case	Not done
Solution submitted	Not done
Skip	You still have to wait 0h, 30m and 0s to be able to skip this challenge

Refresh status

Tweet about this! [#TuentiChallenge4](#)



Follow @Tuentieng