

## Problem K. Duplicate Files

Input file:           files.in  
Output file:         standard output  
Balloon Color:      Red

Many of us have this problem on their computer, where we make several copies of the exact same files, which uses more memory. So you finally decided to write a program to remove all duplicate files.

Here's how your computer works. Whenever you create a new file, it gets a unique ID (when you make a copy of an existing file, the new copy gets a new ID). The IDs are relative to the time, so older files get smaller IDs (but the IDs are not necessary sequential). Also each file has a name, but multiple files can have the same name, and when 2 or more files get the same name, this means they are all exactly the same and they are just identical copies of the one of them with the smallest ID.

You are given the list of all files with their names and IDs, your task is to delete all duplicates and just keep the oldest copy of each file.

### Input

Your program will be tested on one or more test cases. The first line of the input will be a single integer  $T$  ( $1 \leq T \leq 100$ ) representing the number of test cases. Followed by  $T$  test cases.

Each test case starts with a line containing an integer  $N$  ( $1 \leq N \leq 10^5$ ) representing the number of files followed by  $N$  lines, each line will contain the file name followed by a space then the file ID. The file name is a non-empty string of at most 10 lower case English letters, and the ID is a positive integer which is at most  $10^5$ . All IDs will be distinct within each test case.

### Output

For each test case print a single line containing the IDs of the files which won't get deleted, the IDs should be sorted in increasing order and separated by a single space.

### Example

files.in	standard output
2	5 6
2	1
aaa 6	
aa 5	
3	
file 3	
file 2	
file 1	