**Problem: Duplicate Flies**

Input file: files.in

Output file: standard output

Many of us have this problem on our computer where we make several copies of the 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** representing the number of test cases. Followed by **T** test cases.

Each test case test with a line containing an integer **N** (1 ≤ **N** ≥ 105) 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 105 .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  2  aaa 6  aa5  3  file 3  file2  file1 | 5 6  1 |

**Problem C. Self Describing Numbers**

Input file: self.in

Output file: standard output

In one of the classes, Dr.Edsger explained that an integer is said to be “self-describing” if it has the property that: the first digit represents the number of zeroes in the integer; the second digit represents the number of ones in the integer, and so on.

For example, 1210 is a self-describing integer since:

* The first digit is 1 and there is 1 zero in the number &
* The second digit is 2 and there are 2 ones in the number &
* The third digit is 1 and there is 1 two in the number &
* The fourth digit is 0 and there are 0 threes in the number

At the end of the class, Dr.Edsger gave a small homework: create a program to check whether an integer is self-describing or not.

**Input**

The first line of the input contains an integer T, the number of test cases.

Each test case consists of a single integer N (0 ≤ N < 1010 ), without leading zeroes.

**Output**

For each test case, print a single line containing the answer which is either “self-describing” or “not self-describing” without the quotes.

**Example**

|  |  |
| --- | --- |
| Self.in | Standard output |
| 2  1210  2017 | Self-describing  Not self-describing |