J. Smart Warehouse Product Sorting

Constraint: Time Limit: 1 seconds, Memory: 64MB

Context

A logistics center needs to organize product inventory for warehouse storage. Each product belongs to **one Category**(e.g., CatA, CatB, ...) and has **a code number within that specific Category** (positive integer).

Important notes:

- **Two products can have the same code number but belong to different categories, making them different products**
  (Example: "CatA 5" and "CatB 5" are two different products because they have different categories)
- Category names are **case-insensitive** when comparing (e.g., "CatA", "cata", "CATA" are all the same category)
- When outputting results, **preserve** the original case of Category names as they appear in the input

Requirements

Sort **all products** by:

- **Category** in ascending order (lexicographic comparison, case-insensitive)
- Within **the same Category**, sort by **code number** in ascending order

Constraints

- **$1 \le n \le 100{,}000$** (number of products)
- **$1 \le len(category) \le 50$** (category name length)
- **$1 \le code \le 10^9$** (code value)
- Category contains only letters (a-z, A-Z)
- All duplicate products (same category and code) are preserved in the output

Input/Output Format

Input

- First line: integer **n** - number of products **$(1 \le n \le 100{,}000)$**
- Next **n** lines, each line contains:
  - **category**: string containing only letters (a-z, A-Z), length **$(1 \le len \le 50)$**

      o **code**: positive integer **(1 ≤ code ≤ 10^9)**

Output

Print **n** lines after sorting, each line in the format:

Input	Output
n category1 code1 category2 code2 ... categoryn coden	category code

Example

Input	Output
7 CatA 10 CatB 2 CatA 4 CatC 5 CatB 2 CatB 1 CatA 3	CatA 3 CatA 4 CatA 10 CatB 1 CatB 2 CatB 2 CatC 5

Explanation

- **Group by Category** (lexicographic order): CatA → CatB → CatC
- **Within CatA**: codes [10, 4, 3] → sorted as [3, 4, 10]
- **Within CatB**: codes [2, 2, 1] → sorted as [1, 2, 2]
- **CatC**: only has code 5