Class Photo

Rar the Cat wants to take a class photo with all the students. To make the photo nice, he wants to arrange the students in **non-decreasing height** from left to right. Rar the Cat stands at the **leftmost position** of the photo and since he is a cat, he is very short and **effectively has height 0**.

He sees the students arrive one by one and orders each of them to stand **to the right** of the correct person. He wants to ensure that the line of students have non-decreasing height from left to right at all times. If there are multiple possible positions, the student will stand at the position **as close to Rar the Cat as possible** (i.e. choose the leftmost possible position).

Input
The first line will be a single integer **N**. The next **N** lines each contains a name and an integer representing the name and the height of each student in the order they enter the classroom.

Output
For each student, print the name of the student they should stand directly to the right of. You are to ensure that the students are always ordered in non-decreasing height from left to right.

Once all **N** students have arrived, print the names of the students in the photo from **left to right** separated by spaces on a single line. **Do NOT print a space at the end of this line.** Instead, print a newline after the last name.

Limits
- $0 < \mathbf{N} \leq 100{,}000$
- The names of students will only contain lowercase and uppercase English alphabets. It will also not be longer than 20 characters.
- No 2 students will have the same name.
- All the heights will range from 1 to 10^9 inclusive.

Sample Testcase

Sample Input (**classphoto1.in**)	Sample Output (**classphoto1.out**)
5 Gary 3 Steven 6 Panda 6 Shark 10 Turtle 3	Rar Gary Gary Steven Rar Rar Turtle Gary Panda Steven Shark

Explanation for Sample Testcase

1. Initially, the classroom only has Rar the Cat with height 0:
   [Rar, 0]
2. After *Gary* enters the classroom, he should stand on the right of *Rar* since *Gary* is taller:
   [Rar, 0], **[Gary, 3]**
3. After *Steven* enters the classroom, he should stand on the right of *Gary* since *Steven* is taller:
   [Rar, 0], [Gary, 3], **[Steven, 6]**
4. *Panda*'s height of 6 is higher than *Gary* but same as *Steven*. He can either stand directly to the right of *Gary* or directly to the right of Steven to keep the line in increasing order. However, since standing on the right of *Gary* is closer to *Rar* the Cat than standing on the right of *Steven*, you should ask *Panda* to stand on the right of *Gary* instead:
   [Rar, 0], [Gary, 3], **[Panda, 6]**, [Steven, 6]
5. *Shark* should stand on the right of *Steven* since *Shark* is taller:
   [Rar, 0], [Gary, 3], [Panda, 6], [Steven, 6], **[Shark, 10]**
6. *Turtle*'s height of 3 is the same as *Gary*. He can either stand directly to the right of *Rar* the Cat or directly to the right of *Gary*. Since standing on the right of *Rar* the Cat is closer to *Rar* the Cat than standing on the right of *Gary*, he should be asked to stand on the right of *Rar* instead:
   [Rar, 0], **[Turtle, 3]**, [Gary, 3], [Panda, 6], [Steven, 6], [Shark, 10]

Notes:

1. You should develop your program in the subdirectory **ex4** and use the skeleton java file provided. You should not create a new file or rename the file provided.
2. You are free to define your own helper methods and classes (or remove existing ones).
3. Please be reminded that the marking scheme is:
   a. Public Test Cases (1%) - 1% for passing **all** test cases, 0% otherwise
   b. Hidden Test Cases (1%) - Partial scoring depending on test cases passed
   c. Manual Grading (1%)
      i. Overall Correctness (correctness of algorithm, severity of bugs)
      ii. Coding Style (meaningful comments, modularity, proper indentation, meaningful method and variable names)
4. Your program will be tested with a time limit of not less than **2 sec** on Codecrunch.

Skeleton File – Classphoto.java

You are given the skeleton file `Classphoto.java`. You should see a non-empty file when you open the skeleton file. Otherwise, you might be in the wrong working directory.
You should see the following contents when you open the skeleton file:

```
import java.util.*;
public class Classphoto {
    private void run() {
        //implement your "main" method here
    }
    public static void main(String[] args) {
        Classphoto newClassphoto = new Classphoto();
        newClassphoto.run();
    }
}
```

Source

CS2040C AY2018/19 Semester 1 Practical Exam