

Pair GTE

จงเขียนฟังก์ชันเพิ่มเติมให้กับคลาส `CP::pair` เพื่อทำการเปรียบเทียบว่า `CP::pair` สองตัวนั้น มากกว่าหรือเท่ากัน หรือไม่ โดยให้เขียนฟังก์ชันเพื่อ override `operator>=` ของคลาส `CP::pair` โดยมีกฎในการเปรียบเทียบดังนี้

- กำหนดให้ `pair a` เท่ากับ `pair b` ก็ต่อเมื่อทั้ง `first` ของทั้ง `a` และ `b` เท่ากัน และ `second` ของทั้ง `a` และ `b` เท่ากัน
- กำหนดให้ `pair a` มากกว่า `b` ก็ต่อเมื่อเงื่อนไขใดต่อไปนี้เงื่อนไขหนึ่งเป็นจริง
 - `first` ของ `a` นั้นมากกว่า `first` ของ `b`
 - `first` ของ `a` เท่ากับ `first` ของ `b` และ `second` ของ `a` มากกว่า `second` ของ `b`

ข้อบังคับ

- `operator>=` จะต้องไม่ทำการแก้ไขข้อมูลใด ๆ ใน `pair` ทั้งคู่
- `operator>=` จะต้องไม่ทำการเรียกใช้ `operator<` หรือ `operator==` ของ `pair`
- โจทย์ข้อนี้จะมีไฟล์โปรเจ็คของ `Code::Blocks` ให้ ซึ่งในไฟล์โปรเจ็คดังกล่าวจะมีไฟล์ `pair.h`, `main.cpp` และ `student.h` อยู่ ให้นักศึกษาเขียน code เพิ่มเติมลงในไฟล์ `student.h` เท่านั้น และการส่งไฟล์เข้าสู่ระบบ grader ให้ส่งเฉพาะไฟล์ `student.h` เท่านั้น
 - ในไฟล์ `student.h` ดังกล่าวจะต้องไม่ทำการอ่านเขียนข้อมูลใด ๆ ไปยังหน้าจอ หรือคีย์บอร์ดหรือไฟล์ใด ๆ

คำอธิบายฟังก์ชัน `main()`

`main` จะอ่านข้อมูล `int` และ `string` จำนวนสองชุดจาก `cin` แล้วนำข้อมูลดังกล่าวมาสร้างตัวแปร `CP::pair<int,string> a` และ `b` หลังจากนั้นจะแสดงผลจากการเรียก `a >= b` และ `b >= a` ออกทาง `cout`

ตัวอย่าง

<code>CP::pair<int,string> a</code>	<code>CP::pair<int,string> b</code>	<code>a >= b</code>
<code>{5, "yes"}</code>	<code>{1, "haha"}</code>	<code>true</code>
<code>{1, "haha"}</code>	<code>{1, "haha"}</code>	<code>true</code>
<code>{9, "no"}</code>	<code>{9, "zzz"}</code>	<code>false</code>