

Queue Min Element

เราต้องการเพิ่มฟังก์ชัน `T min_element(std::vector<size_t> pos, Comp comp) const` ให้กับคลาส `CP::queue` โดยฟังก์ชันนี้จะต้องคืนค่าของตัวแปรที่มีค่าน้อยที่สุดใน queue โดยพิจารณาเฉพาะข้อมูลที่อยู่ตำแหน่งที่ระบุใน pos เท่านั้น โดยค่าตำแหน่งใน pos จะเป็นจำนวนเต็มไม่ลบที่ระบุตำแหน่งจากหัวของคิว

ตัวอย่างเช่น หาก pos มีค่าเป็น [0,5,2] หมายถึงให้พิจารณาตัวที่มีค่าน้อยที่สุดจากข้อมูลตัวที่อยู่หัวคิว, ตัวที่ควรจะอยู่ที่หัวคิวเมื่อ pop ไปห้าครั้ง (จาก queue ก่อนเรียกฟังก์ชัน) และ ตัวที่ควรจะอยู่ที่หัวคิวเมื่อมีการ pop ไปสองครั้ง (จาก queue ก่อนเรียกฟังก์ชัน ไม่ใช่ queue ที่โดน pop ไปแล้วห้าครั้งจากค่าของ pos ตัวก่อนหน้า)

ค่าใน pos นี้อาจจะระบุถึงตำแหน่งที่ไม่มีอยู่ใน queue ก็เป็นไปได้ ในกรณีดังกล่าวให้ข้ามการพิจารณาตำแหน่งนั้นไป ตัวอย่างเช่นหาก pos มีค่าเป็น [0,5,2] และขนาดของ queue เป็น 4 นั้น ฟังก์ชันนี้จะทำงานเหมือนกับการเรียกด้วยค่า pos เป็น [0,2]

รับประกันว่าจะมีอย่างน้อย 1 ค่าใน pos ที่ระบุถึงตำแหน่งที่มีอยู่ใน queue จริง ๆ

ในการเปรียบเทียบค่ามากหรือน้อยนั้น ให้ทำการเปรียบเทียบโดยใช้ฟังก์ชัน `comp` โดยการเรียกฟังก์ชัน `comp(a,b)` นั้นจะคืนค่า `true` ก็ต่อเมื่อ `a` มีค่าน้อยกว่า `b` โดยที่ทั้ง `a` และ `b` ต้องเป็นข้อมูลประเภท `T`

ข้อบังคับ

- โจทย์ข้อนี้จะมีไฟล์โปรเจกต์ของ `Code::Blocks` ให้ ซึ่งในไฟล์โปรเจกต์ดังกล่าวจะมีไฟล์ `queue.h`, `main.cpp` และ `student.h` อยู่ ให้นิสิตเขียน code เพิ่มเติมลงในไฟล์ `student.h` เท่านั้น และการส่งไฟล์เข้าสู่ระบบ grader ให้ส่งเฉพาะไฟล์ `student.h` เท่านั้น
 - ในไฟล์ `student.h` ดังกล่าวจะต้องไม่ทำการอ่านเขียนข้อมูลใด ๆ ไปยังหน้าจอหรือคีย์บอร์ดหรือไฟล์ใด ๆ

คำอธิบายฟังก์ชัน `main()`

`main` จะอ่านข้อมูลมา 3 บรรทัด ตามรูปแบบนี้

- บรรทัดแรกประกอบด้วยจำนวนเต็ม 2 ตัวคือ `n` และ `m`
- บรรทัดที่สองรับจำนวนเต็ม `n` ตัว และ `main` จะนำข้อมูล `n` ตัวนั้นใส่เข้าไปใน queue
- บรรทัดที่สามรับจำนวนเต็ม `m` ตัว และ `main` จะนำข้อมูลเหล่านั้นมาใส่ไว้ใน vector `pos` หลังจากนั้น `main` จะแสดงผลลัพธ์ของการเรียก `min_element(pos)`

**** main ที่ใช้จริงใน grader นั้นจะแตกต่างจาก main ที่ได้รับในไฟล์โปรเจกต์เริ่มต้น แต่จะทำการทดสอบในลักษณะเดียวกัน ****

ตัวอย่างข้อมูลนำเข้า

Input	ผลลัพธ์
4 3 10 20 30 40 0 5 2	10
5 5 90 50 40 100 20 4 100 2 2 2	20

ชุดข้อมูลทดสอบ

- 20% pos.size() = 1 และ T เป็น int และ comp เป็น std::less<int>()
- 20% mSize <= 10,000 และ T เป็น int
- 20% mSize <= 10,000
- 40% mSize <= 200,000