

Queue Back to Front

จงเพิ่มบริการให้กับคลาส `CP::queue` โดยให้เพิ่มฟังก์ชัน `back to front()` ซึ่งจะทำการย้ายข้อมูลที่อยู่ท้ายคิวไปยังหัวคิว ตัวอย่างเช่น หากคิวมีข้อมูลเป็น `<10,20,30,40,50,60>` เมื่อให้หัวคิวอยู่ทางด้านซ้าย การเรียก `back to front()` จะทำให้ข้อมูลในคิวกลายเป็น `<60,10, 20, 30, 50>` และหากเรียกฟังก์ชันนี้ขณะที่คิวไม่มีข้อมูลอยู่ให้ฟังก์ชันนี้ไม่ต้องทำงานใด ๆ

ข้อบังคับ

โจทย์ข้อนี้จะมีไฟล์โปรเจ็คของ `Code::Blocks` ให้ ซึ่งในโปรเจ็คดังกล่าวจะมีไฟล์ `queue.h`, `main.cpp` และ `student.h` อยู่ ให้นักเขียน code เพิ่มเติมลงในไฟล์ `student.h` เท่านั้น และการส่งไฟล์ขึ้น grader ให้ส่งเฉพาะไฟล์ `student.h` ניתสามารถแก้ไข `student.h` ได้โดยอิสระ สามารถ include และเรียกใช้ data structure หรือ ฟังก์ชันอื่นใดของ `queue` ได้

***** ห้ามทำการพิมพ์ข้อมูลทางจอภาพหรืออ่านข้อมูลจากคีย์บอร์ดในไฟล์ `student.h` ที่ส่งมายัง grader โดยเด็ดขาด *****

คำอธิบายฟังก์ชัน `main()`

โปรแกรมจะเริ่มต้นจาก `CP::queue<int> q` ซึ่งเป็น queue ว่าง หลังจากนั้น `main` จะทำการเรียกใช้ฟังก์ชันต่าง ๆ ของ `queue` ของเราตามข้อมูลคำสั่งที่ได้รับจาก keyboard มาทีละบรรทัด แต่ละบรรทัดนั้นจะระบุการทำงานต่าง ๆ ที่จะกระทำกับคิวของเรา การทำงานมีหลายแบบ โดยขึ้นอยู่กับตัวอักษรตัวแรกในบรรทัด กล่าวคือ `u` เป็นการ push ข้อมูลเข้าไปใน `queue`, `o` เป็นการ pop ข้อมูล, `m` เป็นการเรียกใช้บริการ `back_to_front` และสุดท้าย `p` จะเป็นการพิมพ์ข้อมูลใน `queue` ออกมา สุดท้าย `q` เป็นการจบการทำงาน

- บรรทัดที่มีคำสั่ง `u` จะตามด้วยตัวเลข 1 ตัว ซึ่งเป็น argument ของฟังก์ชันดังกล่าว
- คำสั่ง `p` จะพิมพ์ข้อมูลออกมาจาก `queue` โดยพิมพ์ข้อมูลจำนวนข้อมูลในคิว และ พิมพ์จากตัวแรกไปยังข้อมูลตัวสุดท้ายใน `queue`

***** `main` ใน grader นั้นจะแตกต่างจาก `main` ที่นิสิตได้รับ แต่จะเป็นการทดสอบในลักษณะเดียวกัน ขอให้เขียนฟังก์ชันเพิ่มเติมให้ตรงตามนิยามที่กำหนดไว้ข้างต้น *****

ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
u 1	Size 3: 4 2 3
u 2	Size 3: 3 4 2
u 3	
u 4	
o	
m	
p	
m	
p	
q	

คำแนะนำ

การจะทำข้อนี้ให้ได้คะแนนเต็ม `back_to_front` ควรจะใช้เวลาเป็น $O(1)$