

Queue Move To Front

จะเพิ่มบริการให้กับคลาส CP::queue โดยให้เพิ่มฟังก์ชัน move_to_front(size_t pos) ซึ่งจะทำการ “ลัดคิว” โดยให้ข้อมูลที่อยู่ตำแหน่งที่ pos นับจากหัวคิว (กำหนดให้เรียกหัวคิวว่าตำแหน่งที่ 0, ข้อมูลที่ถัดจากหัวคิวเป็นตำแหน่งที่ 1, ..., ข้อมูลที่อยู่ท้ายคิวคือตำแหน่งที่ size() - 1) นั้น “ลัดคิว” ขึ้นมาอยู่ที่หัวคิวแทน โดยที่ลำดับของข้อมูลอื่น ๆ ยังเรียงตามเดิม ตัวอย่างเช่น หากคิวมีข้อมูลเป็น $<10,20,30,40,50,60>$ เมื่อให้หัวคิวอยู่ทางด้านซ้าย การเรียก move_to_front(3) จะทำให้ข้อมูลในคิวกลายเป็น $<40, 10, 20, 30, 50, 60>$

ข้อบังคับ

โจทย์ข้อนี้จะมีไฟล์โปรแกรม Code::Blocks ให้ ซึ่งในโปรเจ็คดังกล่าวจะมีไฟล์ queue.h, main.cpp และ student.h อยู่ ให้นิสิตเขียน code เพิ่มเติมลงไปในไฟล์ student.h เท่านั้น และการส่งไฟล์ขึ้น grader ให้ส่งเฉพาะไฟล์ student.h นิสิตสามารถแก้ไข student.h ได้โดยอิสระ สามารถ include และเรียกใช้ data structure หรือ ฟังก์ชัน อื่นใดของ queue ได้

*** ห้ามทำการพิมพ์ข้อมูลทางจอภาพหรืออ่านข้อมูลจากคีย์บอร์ดในไฟล์ student.h ที่ส่งมาจัง grader โดยเด็ดขาด ***

คำอธิบายฟังก์ชัน main()

โปรแกรมจะเริ่มต้นจาก CP::queue<int> q ซึ่งเป็น queue ว่า หลังจากนั้น main จะทำการเรียกใช้ ฟังก์ชันต่าง ๆ ของ queue ของเรามาตามข้อมูลคำสั่งที่ได้รับจาก keyboard มาทีละบรรทัด แต่ละบรรทัดนั้นจะระบุ การทำงานต่าง ๆ ที่จะกระทำกับคิวของเรา การทำงานมีหลายแบบ โดยขึ้นอยู่กับตัวอักษรตัวแรกในบรรทัด กล่าวคือ บ เป็นการ push ข้อมูลเข้าไปใน queue, o เป็นการ pop ข้อมูล, m เป็นการเรียกใช้บริการ move_to_front และ สุดท้าย p จะเป็นการพิมพ์ข้อมูลใน queue ออกมา สุดท้าย q เป็นการจบการทำงาน

- บรรทัดที่มีคำสั่ง บ และ m จะตามด้วยตัวเลข 1 ตัว ซึ่งเป็น argument ของฟังก์ชันดังกล่าว
- คำสั่ง p จะพิมพ์ข้อมูลออกมาจาก queue โดยพิมพ์ข้อมูลจำนวนข้อมูลในคิว และ พิมพ์จากตัวแรกไปยัง ข้อมูลตัวสุดท้ายใน queue

*** main ใน grader นั้นจะแตกต่างจาก main ที่นิสิตได้รับ แต่จะเป็นการทดสอบในลักษณะเดียวกัน ขอให้ เขียนฟังก์ชันเพิ่มเติมให้ตรงตามนิยามที่กำหนดไว้ข้างต้น ***

ตัวอย่าง

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