

List Check

จงเพิ่มบริการให้กับคลาส `CP::list<T>` โดยให้เพิ่มฟังก์ชัน `bool check()` เพื่อตรวจสอบว่า linked list นี้มีโครงสร้างเป็น linked list ที่ถูกต้องหรือไม่ กล่าวคือตัวแปร `next` และ `prev` ของปมต่าง ๆ มีการชี้อย่างถูกต้องตามรูปแบบของ linked list ครบถ้วนตามจำนวนที่ควรเป็น กล่าวคือต้องเห็น circular doubly linked list with header ที่มีจำนวนปมครบเท่ากับ `mSize` และการเชื่อมต่อของ `next` ตั้งแต่ของ `mHeader` จะเรียงลำดับจากตัวแรกถึงตัวสุดท้ายและตัวสุดท้ายชี้กลับมาที่ `mHeader` และ การเชื่อมต่อของ `prev` ตั้งแต่ `mHeader` จะเรียงลำดับจากตัวสุดท้ายถึงตัวแรกและตัวแรกจุดจะชี้กลับมาที่ `mHeader`

ฟังก์ชันนี้ต้องคืนค่า `true` เมื่อ linked list นี้มีโครงสร้างที่ถูกต้อง และคืนค่า `false` เมื่อไม่ใช่ รับประกันว่าค่า `next`, `prev` ของทุก ๆ ปม (รวมถึง header) จะชี้ไปยังตัวแปรประเภท node ที่มีอยู่จริง ๆ (แต่อาจจะไม่ถูกต้อง) หรือไม่ก็เป็นค่า `NULL` เท่านั้น

ฟังก์ชันนี้ควรใช้เวลาเป็น $O(mSize)$

ข้อบังคับ

โจทย์ข้อนี้จะมีไฟล์โปรเจ็คของ `Code::Blocks` ให้ ซึ่งในโปรเจ็คดังกล่าวจะมีไฟล์ `list.h`, `main.cpp` และ `student.h` อยู่ ให้นักเขียน code เพิ่มเติมลงในไฟล์ `student.h` เท่านั้น และการส่งไฟล์ขึ้น grader ให้ส่งเฉพาะไฟล์ `student.h`

คำอธิบายฟังก์ชัน main

โปรแกรมจะรับค่า `n` และสร้าง `CP::list<int> l` แล้วใส่ค่า 1 ถึง `n` ลงไปใน list นี้ แล้วเรียกฟังก์ชัน `messup` ของ `l` ซึ่งฟังก์ชันนี้จะทำการแก้ค่า `next` ของบาง node ให้ผิด แล้วจะเรียก `check` เพื่อตรวจสอบ พร้อมแสดงผลออกทางหน้าจอ

นิสิตสามารถแก้ไขฟังก์ชัน `messup` ได้ตามต้องการเพื่อทำการทดสอบใด ๆ โดยแก้ไขในไฟล์ `list.h` ได้เลย อย่างไรก็ตาม ใน grader นั้นจะมีวิธีการแก้ไข list ให้พังที่แตกต่างจากวิธีที่ใช้ใน `messup` นี้

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

ชุดข้อมูลทดสอบ

- 10% ความผิดพลาด (ถ้ามี) จะเกิดที่ `next` ของ `mHeader` แน่ ๆ
- 10% ความผิดพลาด (ถ้ามี) จะเกิดที่ `next` ของ `mHeader` หรือของตัวแรก เพียงหนึ่งเดียวแน่ ๆ
- 10% ความผิดพลาด (ถ้ามี) จะเกิดที่ `next` หรือ `prev` ของ `mHeader` หรือไม่กี่ของตัวแรก เพียงหนึ่งเดียวแน่ ๆ
- 20% รับประกันว่า list ที่ผิดพลาดเกิดจากการแก้ไข pointer เพียงจุดที่เดียวแน่ ๆ
- 50% ไม่มีข้อจำกัดอื่นใด

คำอธิบายเพิ่มเติม

ในฟังก์ชัน `messup` นั้นมีการเพิ่มส่วนของโปรแกรมเพื่อป้องกันมิให้การที่ `messup` ทำให้ link ต่าง ๆ เสียไปมีผลกระทบต่อดestructor หากนิสิตแก้ไข `messup` ในส่วนดังกล่าวอาจจะทำให้โปรแกรมทำงานผิดพลาดได้ในเครื่องของนิสิต หากต้องการแก้ไข `mess up` ให้แก้ไขเฉพาะบรรทัดที่อยู่หลังจากบรรทัดที่เขียน `//ONLY EDIT BELOW THIS LINE` เท่านั้น