

# List Iterator Position

(1 sec, 512mb)

จงเพิ่มบริการ `int data_iterator(CP::list<T>::iterator it)` ให้กับ `CP::list<T>` เพื่อตรวจสอบว่า `it` เป็น iterator ที่ชี้ไปยังข้อมูลของ `CP::list` ที่เรียกใช้ฟังก์ชันดังกล่าวหรือไม่ โดยฟังก์ชันนี้จะต้องคืนค่าตามกฎดังต่อไปนี้

- ในกรณีที่ `it` ชี้ไปยังข้อมูล `CP::list` นั้น ให้คืนหมายเลขลำดับของ iterator ดังกล่าว โดยให้ข้อมูลตัวแรกสุดที่มีของ `list` (ข้อมูลที่อยู่ ณ `begin()`) คือตำแหน่งที่ 0 ให้สังเกตว่าในกรณีนี้ ค่าที่คืนมาจะเป็นไปได้ตั้งแต่ 0 ถึง `size() - 1`
- ในกรณีที่ `it` ชี้ไปยัง `end()` ให้คืนค่า -1
- ในกรณีที่ `it` ไม่ได้ชี้ไปยังข้อมูลหรือ `end()` ของ `list` ที่เรียก ให้คืนค่า -2

## คำอธิบายฟังก์ชัน main

`main()` จะสร้าง `list<int>` มาสองตัว คือ `l1` และ `l2` หลังจากนั้นจะรับข้อมูลจำนวนเต็มเข้ามา 4 ตัวคือ `n1`, `p1`, `n2` และ `p2` แล้วกระทำการดังต่อไปนี้

- ใส่ข้อมูลเข้าไปใน `l1` เป็นจำนวน `n1` ตัว
- สร้าง iterator ขึ้นมา 3 ตัวคือ `it1a`, `it1b` `it1c` โดยให้ iterator ทั้งสามตัวนั้นชี้ไปยัง `begin()`, ข้อมูลตำแหน่งที่ `p1` และ `end()` ของ `l1` ตามลำดับ
- ทำซ้ำกับ `l2` แต่ใช้ข้อมูลเป็น `n2` และ `p2` เพื่อสร้าง `it2a`, `it2b`, `it2c`
- แสดงผลการเรียก `l1.data_iterator` ด้วย iterator ทั้ง 6 ตัวที่สร้างขึ้นมา
- แสดงผลการเรียก `l2.data_iterator` ด้วย iterator ทั้ง 6 ตัวที่สร้างขึ้นมา

## ชุดข้อมูลทดสอบ

- 10% `list` ที่เรียกมีข้อมูล 2 ตัว และ `it` ชี้ไปยังข้อมูลใดข้อมูลหนึ่งใน 2 ตัวนั้นแน่ ๆ
- 15% `list` ที่เรียกมีข้อมูล 1 ตัว และ `it` ชี้ไปยังข้อมูลนั้น หรือไม่ก็ `end()` ของ `list` ที่เรียกแน่ ๆ
- 15% `it` ชี้ไปยังข้อมูลใดข้อมูลหนึ่งของ `list` ที่เรียก หรือ ชี้ไปยัง `begin()` ของ `list` อื่น
- 20% `it.ptr` ชี้ไปยัง `node` ที่ใช้งานได้แน่ ๆ
- 40% ไม่มีข้อบังคับอื่นใด (`it.ptr` อาจจะมีค่าเป็น `null` ก็ได้)

## ข้อบังคับ

- โจทย์ข้อนี้จะมีไฟล์โปรเจกต์ของ `Code::Blocks` ให้ ซึ่งในไฟล์โปรเจกต์ดังกล่าวจะมีไฟล์ `list.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
- หากใช้ VS Code ให้ทำการ compile ที่ไฟล์ `main.cpp`

**\*\* main ที่ใช้จริงใน grader นั้นจะแตกต่างจาก main ที่ได้รับในไฟล์โปรเจกต์เริ่มต้นแต่จะทำการทดสอบในลักษณะเดียวกัน \*\***

## ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
2 1 4 3	l1 with begin() of l1 : 0 l1 with p1 of l1 : 1 l1 with end() of l1 : -1 l1 with begin() of l1 : -2 l1 with p1 of l1 : -2 l1 with end() of l1 : -2 l2 with begin() of l1 : -2 l2 with p1 of l1 : -2 l2 with end() of l1 : -2 l2 with begin() of l1 : 0 l2 with p1 of l1 : 3 l2 with end() of l1 : -1
0 5 7 9	l1 with begin() of l1 : -1 l1 with p1 of l1 : -2 l1 with end() of l1 : -1 l1 with begin() of l1 : -2 l1 with p1 of l1 : -2 l1 with end() of l1 : -2 l2 with begin() of l1 : -2 l2 with p1 of l1 : -2 l2 with end() of l1 : -2 l2 with begin() of l1 : 0 l2 with p1 of l1 : -2 l2 with end() of l1 : -1
1 1 6 6	l1 with begin() of l1 : 0 l1 with p1 of l1 : -1 l1 with end() of l1 : -1 l1 with begin() of l1 : -2 l1 with p1 of l1 : -2 l1 with end() of l1 : -2 l2 with begin() of l1 : -2 l2 with p1 of l1 : -2 l2 with end() of l1 : -2 l2 with begin() of l1 : 0 l2 with p1 of l1 : -1 l2 with end() of l1 : -1