

Vector Multi Unique

(1 sec, 512mb)

จงเพิ่มบริการ `uniq(std::vector<CP::vector<T>::iterator> itr)` ให้กับ `CP::vector<T>` ซึ่งฟังก์ชันนี้จะรับ `itr` ซึ่งเป็นรายการของ iterator ของ vector นี้ โดยฟังก์ชันนี้จะพิจารณาข้อมูลทุกตัวที่ระบุด้วย `itr` แล้วทำให้ข้อมูลดังกล่าวไม่มีข้อมูลใดซ้ำกันเลย โดยหากมีข้อมูลใดซ้ำกันแล้วจะต้องทำการลบข้อมูลตัวที่ซ้ำที่ไม่ได้อยู่ในช่องที่อยู่ใกล้ `begin()` มากที่สุดออกไป

ตัวอย่างเช่น หาก `itr` ชี้ไปยังข้อมูล 5 ตัวคือ

- ข้อมูล 5 ในช่อง 3
- ข้อมูล 5 ในช่อง 6
- ข้อมูล 10 ในช่อง 9
- ข้อมูล 5 ในช่อง 13
- ข้อมูล 10 ในช่อง 17

เราจะต้องทำให้ข้อมูลในช่อง 6, 13, 17 ถูกลบไป

รับประกันว่าการเรียกฟังก์ชันนี้ จะมีค่า `itr` ที่เป็น iterator ที่มีค่าอยู่ในช่วงตั้งแต่ `begin()` จนถึง `end()-1` แน่แน่นอน และค่าใน `itr` ในการเรียกแต่ละครั้งจะไม่มีตำแหน่งที่ซ้ำกันเลย

คำอธิบายฟังก์ชัน main

`main()` จะเป็นการทดลองใช้งาน `CP::vector` ด้วยคำสั่งต่าง ๆ โดย `main` จะ สร้าง `CP::vector<int>` ชื่อ `vec` มาแล้วอ่านคำสั่งที่ละบรรทัด ซึ่งแต่ละบรรทัดจะเริ่มต้นด้วย string ที่ระบุคำสั่ง และอาจจะตามด้วยค่าต่าง ๆ ที่จำเป็นสำหรับคำสั่งนั้น โดยมีรูปแบบของแต่ละคำสั่งดังต่อไปนี้ (ให้ `n`, `X` และ `Y` หมายถึงค่าประเภท `int` ใด ๆ)

- `p` เป็นการพิมพ์ค่าของ `vec` ออกมาพร้อมกับขนาดของ `vec`
- `a X Y` จะเป็นการ insert ค่า `Y` ไปยังตำแหน่ง `X` ใน `vec`
- `u n` แล้วตามด้วย `X` จำนวน `n` ตัว จะเป็นการเรียก `uniq` ของ `vec` โดยให้ `itr` เป็น iterator ที่ชี้ไปยังตำแหน่ง `X` ต่าง ๆ ตามที่ระบุ
- `q` เป็นการหยุดการทำงาน

ชุดข้อมูลทดสอบ

รับประกันว่าจำนวนคำสั่งที่กระทำต่อ `vec` จะไม่เกิน 1,000,000 คำสั่งแน่นอน

- 10% คำสั่ง `u` จะมีค่า `n` เป็น 2 เสมอ
- 20% คำสั่ง `u` จะมีค่า `n` ไม่เกิน 5 และ ขนาดของ `vec` ไม่เกิน 100 เสมอ และ ค่า `X` ต่าง ๆ ของคำสั่งนั้นจะเรียงจากน้อยไปมาก
- 10% คำสั่ง `u` จะมีค่า `n` ไม่เกิน 5 และ ขนาดของ `vec` ไม่เกิน 100 เสมอ
- 20% ขนาดของ `vec` ไม่เกิน 100 เสมอ
- 40% ไม่มีข้อจำกัดอื่นใด

ข้อบังคับ

- โจทย์ข้อนี้จะมีไฟล์โปรเจกต์ของ Code::Blocks ให้ ซึ่งในไฟล์โปรเจกต์ดังกล่าวจะมีไฟล์ vector.h, main.cpp และ student.h อยู่ให้นิสิตเขียน 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 ****

ตัวอย่างการทำงานของ main

ข้อมูลนำเข้า	ข้อมูลส่งออก
a 0 5 a 1 5 a 2 10 a 3 5 a 4 10 p u 4 0 2 3 4 p a 2 3 p u 4 2 3 1 0 p q	5 5 5 10 5 10 3 5 5 10 4 5 5 3 10 3 5 3 10
a 0 1 a 1 2 a 2 3 a 0 1 a 1 2 a 2 3 p u 3 2 0 1 p u 2 4 1 p q	6 1 2 3 1 2 3 6 1 2 3 1 2 3 5 1 2 3 1 3