Union, Intersection of Ordered Set

256MB เวลาไม่เกิน 5 วินาที

ในข้อนี้ขอนิยามให้เซตแบบมีลำดับ (Ordered Set) คือ รายการของข้อมูล**ที่ไม่มีข้อมูลซ้ำ** และลำดับของข้อมูลมีนัยสำคัญ เช่น <3,2,9,1> คือเซตแบบมีลำดับที่มีข้อมูล 3,2,9,1 ตามลำดับซึ่งไม่เหมือนกับ <3,9,1,2> ซึ่งเราสามารถจัดเก็บได้ด้วย std::vector โดยเรานิยามฟังก์ชัน Union(A,B) และ Intersect(A,B) เมื่อ A และ B เป็น Ordered Set ดังนี้

- Union(A,B) ให้คืนเซตแบบมีลำดับ ที่มีข้อมูลที่อยู่ใน A หรือ B โดยที่ข้อมูลที่อยู่ใน A ต้องมาก่อนหน้าข้อมูลที่อยู่แต่ใน B เสมอ เช่น Union(<4,6,9,0,7>, <0,3,2,4,5>) = <4,6,9,0,7,3,2,5>
- Intersect(A,B) ให้คืนเซตแบบมีลำดับ ที่มีข้อมูลที่อยู่ทั้งใน A และ B โดยข้อมูลจะต้องอยู่ในลำดับก่อนหลังเหมือนกับ A เช่น Intersect(<3,9,0,1,6>,<4,6,9,1,7>) = <9,1,6>

ข้อสังเกต Union(A,B) อาจจะไม่เท่ากับ Union(B,A) ก็ได้, Intersect(A,B) อาจจะไม่เท่ากับ Intersect(B,A)

โดยในข้อนี้ให้นิสิตเติมฟังก์ชันดังต่อไปนี้ใน student.h ให้สมบูรณ์ (นิสิตสามารถประกาศฟังก์ชั่น หรือ include เพิ่มได้ตามต้องการ ใน student.h)

```
vector<T> Union(const vector<T>& A, const vector<T>& B);
vector<T> Intersect(const vector<T>& A, const vector<T>& B);
```

โดยไฟล์ main.cpp ที่ให้มาแสดงตัวอย่างการเรียกใช้ฟังก์ชัน (ซึ่งเป็นคนละตัวกับ main.cpp ที่จะใช้ทดสอบใน grader) โดย**ให้นิสิตส่งเฉพาะ student.h เท่านั้น** นิสิตสามารถแก้ไข main.cpp ได้เพื่อทดสอบฟังก์ชันตามต้องการแต่ห้ามส่ง main.cpp

ผลลัพท์ที่คาดหวังเมื่อรัน main.cpp

```
4 6 9 0 7 3 2 5
9 1 6
```

หมายเหตุ ชุดทดสอบแต่ละชุดจะทดสอบเฉพาะฟังก์ชันใดฟังก์ชันหนึ่งเท่านั้น และ 50% จะเป็น Union อีก 50% เป็น Intersect โดยหากนิสิตทำได้ข้อใดข้อหนึ่งก็ขอให้ส่งมาได้ (แต่ต้อง compile ผ่าน) และรับประกันว่าทั้ง A และ B มีข้อมูลไม่เกิน 100000 ตัว