

Vector Block Swap

จงเขียนฟังก์ชัน `bool block_swap(iterator a, iterator b, size_t m)` เพิ่มเติมให้กับคลาส `CP::vector` โดยเพิ่มฟังก์ชัน สำหรับการสลับข้อมูลจำนวน `m` ตัวจากช่องที่ระบุด้วย iterator `a` ถึง `a+m-1` กับข้อมูลจำนวน `m` ตัวในช่องที่ระบุด้วย iterator `b` ถึง `b+m-1` ใน vector ของเรา ตัวอย่างเช่น เมื่อกำหนดให้ vector `v` มีค่าเป็น `{10,20,30,40,50,60,70,80}` และ `a, b, m` มีค่าเป็น `v.begin()+5, v.begin()+1, 2` ตามลำดับ การเรียก `v.block_swap(a,b,m)` จะทำให้ `v` มีค่าเป็น `{10,60,70,40,50,20,30,80}`

ฟังก์ชันนี้จะทำงานได้ก็ต่อเมื่อ `a, b, a+m-1, b+m-1` เป็น iterator ที่ถูกต้องของ `v` และเมื่อช่วงของข้อมูล `a` ถึง `a+m-1` ไม่ทับกับช่วงของข้อมูล `b` ถึง `b+m-1` และ `m` มีค่ามากกว่า 0 ฟังก์ชันนี้จะคืนค่า `true` ก็ต่อเมื่อเงื่อนไขข้างต้นเป็นจริง แต่ถ้าหากเงื่อนไขดังกล่าวไม่เป็นจริง ให้คืนค่า `false` โดยไม่ต้องทำการเปลี่ยนแปลงข้อมูลใด ๆ ใน vector

ข้อบังคับ

- โจทย์ข้อนี้จะมีไฟล์โปรเจกต์ของ `Code::Blocks` ให้ ซึ่งในไฟล์โปรเจกต์ดังกล่าวจะมีไฟล์ `vector.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

คำอธิบายฟังก์ชัน `main()`

`main` จะอ่านข้อมูลมาสองบรรทัด ตามรูปแบบนี้

- บรรทัดแรกประกอบด้วยจำนวนเต็ม `n`
- บรรทัดที่สองประกอบด้วยจำนวนเต็มจำนวน `n` ตัว ซึ่งจะใช้เป็นข้อมูลของ vector
- บรรทัดที่สามประกอบด้วยจำนวนเต็ม 3 ตัว คือ `a, b, m`

หลังจากนั้น `main` จะสร้าง vector ตามข้อมูลดังกล่าวแล้วเรียก `block_move` ของ vector นั้นด้วยค่า `v.begin()+a, v.begin()+b, m` แล้วแสดงผลข้อมูลทั้งหมดใน vector ดังกล่าวออกมา รวมถึงขนาดของ vector และค่าที่คืนมาจากการเรียกฟังก์ชันดังกล่าวด้วย

ข้อควรระวัง

***** `main` ที่ใช้จริงใน grader นั้นจะแตกต่างจาก `main` ที่ได้รับในไฟล์โปรเจกต์เริ่มต้น *****

ตัวอย่าง

ค่าใน vector <code>v</code>	<code>a</code>	<code>b</code>	<code>m</code>	ค่าที่คืนจากการเรียก	<code>v</code> หลังเรียก
{1,2,3,4,5}	0	1	0	false	{1,2,3,4,5}
{1,2,3,4,5}	0	1	1	true	{2,1,3,4,5}
{1,2,3,4,5}	1	3	2	true	{1,4,5,2,3}
{1,2,3,4,5}	1	4	2	false	{1,2,3,4,5}
{1,2,3,4,5}	3	0	2	true	{4,5,3,1,2}
{1,2,3,4,5}	0	3	3	false	{1,2,3,4,5}
{1,2,3,4,5,6,7}	0	2	3	false	{1,2,3,4,5,6,7}