Split Stack

เราต้องการเพิ่มฟังก์ชัน std::vector<std::vector<T>> split_stack(int k) const ลงใน CP::stack โดยฟังก์ชันนี้จะต้องคืนตัวแปรประเภท vector ขนาด k โดยที่ vector[x] จะต้องมีค่าภายในลำดับดังนี้ {T[x], T[x + k], T[x + 2k], ...} **ที่เรียงย้อนกลับ** ซึ่งขอนิยาม T[x] ว่าหมายถึงตัวที่จะเป็น Top of stack หลังจากการ Pop ออกมาแล้ว x ครั้ง (แต่เราไม่ได้ทำการ pop จริง ๆ)

ยกตัวอย่างเช่นเรามี Stack ที่มีข้อมูลภายในตามลำดับดังนี้ [10, 20, 30, 40, 50, 60] **(กำหนดให้ด้านขวาสุดคือ Top of stack)** T[0] คือ 60 และ T[2] คือ 40 เป็นต้น ดังนั้น เมื่อทำการเรียกใช้ split_stack(4) จะได้ผลลัพธ์ดังนี้

- vector[0] = {20, 60}
- vector[1] = {10, 50}
- vector[2] = {40}
- vector[3] = {30}

เป็นไปได้ว่า k อาจมีค่ามากกว่าขนาด stack ดังนั้นในกรณีที่ T[x] เป็นค่าที่ไม่มีอยู่จริง ให้ข้ามค่าดังกล่าวไป

ข้อบังคับ

- โจทย์ข้อนี้จะมีไฟล์โปรเจ็คของ Code::Blocks ให้ซึ่งในไฟล์โปรเจ็คดังกล่าวจะมีไฟล์ stack.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
- **สามารถ include library ใดเพิ่มก็ได้ในไฟล์ student.h แต่จะต้องไม่มีการใช้ compiler directive อื่น grader จะไม่ทำการตรวจสอบเรื่องนี้ แต่จะมีการตรวจสอบภายหลัง การผิดเงื่อนไขจะได้คะแนนในข้อนี้เป็น 0**

คำอธิบายฟังก์ชัน main

main จะอ่านข้อมูลมา 3 บรรทัด ตามรูปแบบนี้

- บรรทัดแรกประกอบด้วยจำนวนเต็ม 2 ตัวคือ n และ m
- บรรทัดที่สองรับจำนวนเต็ม n ตัวและ main จะนำข้อมูล n ตัวนั้นใส่เข้าไปใน stack ตามลำดับ
- บรรทัดที่สามรับจำนวนเต็ม m ตัว โดยจะหมายถึงค่า k ที่จะใส่ในฟังก์ชัน split_stack โดย main จะทำการ print ผลลัพธ์จากการเรียก split_stack ในแต่ละครั้ง

ชุดข้อมูลทดสอบ

ให้ mSize คือ ขนาดของข้อมูลใน stack ก่อนเรียกฟังก์ชัน split_stack

- 10% 1 <= mSize <= 10 และ k = 1
- 20% 1 <= mSize <= 1,000 และ mSize % k = 0
- 20% 1 <= mSize <= 1,000 และ k <= 1,000
- 50% 1 <= mSize <= 100,000 และ k <= 200,000
- **รับประกันว่าข้อมูลทดสอบถูกออกแบบมาให้สามารถทำงานได้ใน 1 วินาที**

**** main ที่ใช้จริงใน grader นั้นจะแตกต่างจาก main ที่ได้รับในไฟล์โปรเจ็กต์เริ่มต้น แต่จะทำการทดสอบในลักษณะเดียวกัน ****

ตัวอย่างข้อมูลนำเข้า

Input	ผลลัพธ์
6 2 10 20 30 40 50 60 4 2	vector 0(2): 20 60 vector 1(2): 10 50 vector 2(1): 40 vector 3(1): 30 vector 0(3): 20 40 60 vector 1(3): 10 30 50
4 2 1 22 333 4444 5 7	vector 0(1): 4444 vector 1(1): 333 vector 2(1): 22 vector 3(1): 1 vector 4(0): vector 0(1): 4444 vector 1(1): 333 vector 2(1): 22 vector 3(1): 1 vector 4(0): vector 5(0): vector 6(0):