

Stack Distribute

จงเขียนฟังก์ชันเพิ่มเติมให้กับคลาส `CP::stack` ซึ่งจะทำให้การ “กระจาย” ข้อมูลใน stack ให้เป็น vector หลาย ๆ ตัวที่มีขนาดพอ ๆ กัน โดยข้อมูลใน vector แต่ละตัวนั้นจะต้องเรียงตามลำดับการ pop ข้อมูลออกมาจาก stack ตัวอย่างเช่น ถ้าให้ stack `s` มีข้อมูลเป็น `[1,2,3,4,5,6,7,8]` โดยที่ top of stack อยู่ด้านซ้ายในรายการนี้ การเรียก `s.distribute(4)` จะเป็นการกระจายข้อมูลใน `s` ออกมาเป็น vector 4 ตัว โดยที่ตัวแรกจะเป็น `{1,2}` ตัวถัดมาเป็น `{3,4}` ตัวถัดมาเป็น `{5,6}` และตัวสุดท้ายเป็น `{7,8}`

กำหนดให้ฟังก์ชันดังกล่าว คือ `vector<vector<T>> distribute(size_t k)` ซึ่งฟังก์ชันนี้จะต้องคืน vector จำนวน `k` ตัว (โดยคืนมาเป็น vector ของ vector) กำหนดให้ `v` คือ `vector<vector<T>>` ที่คืนมาจากฟังก์ชันนี้ ผลลัพธ์ที่ได้จะต้องมีคุณสมบัติดังนี้

- `v.size()` ต้องเท่ากับ `k` (กล่าวคือ ข้อมูลถูกแบ่งออกเป็น `k` vector)
- ผลรวมของ `v[i].size()` แต่ละตัวจะต้องเท่ากับ `s.size()` (ข้อมูลถูกแบ่งออกมาทั้งหมด โดยไม่ขาดและไม่ซ้ำกัน)
- ขนาดของ `v[i].size()` ต้องเท่ากัน หาก `s.size()`หารด้วย `k` ไม่ลงตัว เราจะยอมให้ `v[a].size() - v[b].size()` มีค่าเป็น 1 ได้เฉพาะเมื่อ `a < b` เท่านั้น (กล่าวคือ กรณีแบ่งไม่ลงตัว จะยอมให้ vector แรก ๆ มีข้อมูลมากกว่า vector หลัง ๆ ได้ไม่เกิน 1 ตัว)
- ข้อมูลใน `v[i]` แต่ละตัวต้องเรียงลำดับตามการ pop ออกจาก `s` และข้อมูลใน `v[a]` ต้องเป็นข้อมูลที่ถูก pop ออกมาก่อนข้อมูลใน `v[b]` ก็ต่อเมื่อ `a < b`

นอกจากนี้ ฟังก์ชันนี้ถูกประกาศเป็นแบบ `const` ซึ่งหมายความว่าเราไม่สามารถทำการแก้ไขข้อมูลใด ๆ ของ stack ได้

ข้อบังคับ

- โจทย์ข้อนี้จะมีไฟล์โปรเจ็คของ `Code::Blocks` ให้ ซึ่งในไฟล์โปรเจ็คดังกล่าวจะมีไฟล์ `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

คำอธิบายฟังก์ชัน `main()`

ใน `main` จะรับค่าจำนวนเต็ม `n` และ `k` จากคีย์บอร์ด หลังจากนั้นจะรับข้อมูลจำนวน `n` ตัวมาเก็บไว้ใน stack `s` แล้วเรียก `v = s.distribute(k)` และพิมพ์ค่าใน `v` ออกมาทางหน้าจอ

***** main ที่ใช้จริงใน grader นั้นจะแตกต่างจาก main ที่ได้รับในไฟล์โปรเจ็คเริ่มต้น แต่จะทำการทดสอบในลักษณะเดียวกัน ****

ตัวอย่างการใช้งาน

<code>s</code> (top of stack อยู่ด้านซ้าย)	<code>k</code>	ผลลัพธ์จากการเรียก <code>V = s.distribute(k)</code>
<code>[1,2,3,4,5,6,7,8]</code>	2	<code>V[0] = {1,2,3,4}</code> <code>V[1] = {5,6,7,8}</code>
<code>[1,2,3,4,5,6,7,8]</code>	3	<code>V[0] = {1,2,3}</code> <code>V[1] = {4,5,6}</code>

		$V[2] = \{7, 8\}$
[1,2,3,4]	4	$V[0] = \{1\}$ $V[1] = \{2\}$ $V[2] = \{3\}$ $V[3] = \{4\}$
[1,2,3,4,5,6,7,8,9,10]	4	$V[0] = \{1, 2, 3\}$ $V[1] = \{4, 5, 6\}$ $V[2] = \{7, 8\}$ $V[3] = \{9, 10\}$