

## Multiple erase from Stack

จงเขียนฟังก์ชันเพิ่มเติมความสามารถในการลบข้อมูลของ `CP::stack` โดยให้เพิ่มฟังก์ชันสองฟังก์ชันให้กับคลาสดังกล่าว คือ

- `void multi_pop( size_t K )` ซึ่งจะกระทำการลบข้อมูลจำนวน `K` ตัวทิ้งออกไปจาก `stack` ของเรา โดยลบจากข้อมูลตัวที่อยู่บนสุดใน `stack` ไหลลงมาเรื่อย ๆ ตามลำดับ ในกรณีที่มีข้อมูลใน `stack` น้อยกว่า `K` ตัว ให้ทำการลบเฉพาะข้อมูลที่มีอยู่เท่านั้น
- `std::stack<T> remove_top( size_t K )` ฟังก์ชันนี้จะทำงานคล้ายกับ `multi_pop` คือจะทำการลบข้อมูลจำนวน `K` ตัวทิ้งออกไปจาก `stack` ของเรา โดยลบจากข้อมูลตัวที่อยู่บนสุดใน `stack` ไหลลงมาเรื่อย ๆ ตามลำดับ แต่ให้คืนข้อมูลที่ลบมาทั้งหมดมาใน `std::stack` อีกอันหนึ่งด้วย โดยให้ข้อมูลใน `stack` ที่คืนมานั้น มีลำดับเหมือนกับข้อมูลที่อยู่ใน `stack` ของเราก่อนที่จะลบด้วย กล่าวคือ ตัวบนสุดของ `stack` ที่คืนมา จะเท่ากับ ตัวบนสุดของ `stack` ของเรา และ ตัวที่อยู่ถัดจากตัวบนสุดของ `stack` ที่คืนมา จะเท่ากับ ตัวที่อยู่ถัดจากตัวบนสุดของ `stack` ของเรา เป็นเช่นนี้ไปเรื่อย ๆ

ตัวอย่างเช่น สมมติให้ `stack` ของเรามีค่าเป็น `<1, 2, 3, 4, 5>` โดยให้ตัวซ้ายสุดคือตัวบนสุดของ `stack` การสั่ง `remove_top( 3 )` จะทำให้ `stack` ของเรากลายเป็น `<4, 5>` และฟังก์ชันจะต้องคืน `stack` ที่มีค่า `<1, 2, 3>` กลับมา

ในกรณีที่มีข้อมูลใน `stack` น้อยกว่า `K` ตัว ให้ทำการลบเฉพาะข้อมูลที่มีอยู่เท่านั้น

## ข้อบังคับ

โจทย์ข้อนี้จะมีไฟล์โปรเจ็คของ `code::block` ให้ ซึ่งในโปรเจ็คดังกล่าวจะมีไฟล์ `vector.h`, `main.cpp` และ `student.h` อยู่ ให้นิสิตเขียน `code` เพิ่มเติมลงไปไฟล์ `student.h` เท่านั้น และการส่งไฟล์ขึ้น `grader` ให้ส่งเฉพาะไฟล์ `student.h`

## คำอธิบายฟังก์ชัน `main()`

`main` จะทำการเรียกใช้ฟังก์ชันต่าง ๆ ของ `stack` ของเรา โดยโปรแกรมจะเริ่มจาก `stack<int>` มา 1 อัน และเรียกใช้งาน `stack` ดังกล่าวตามข้อมูลคำสั่งที่ได้รับจาก `keyboard` มาทีละบรรทัด แต่ละบรรทัดนั้นจะมีความทำงานต่าง ๆ ซึ่งขึ้นอยู่กับตัวอักษรตัวแรกในบรรทัด โดยที่ `a` เป็นการ `push` ข้อมูลเข้าไปใน `stack`, `d` เป็นการ `pop` ข้อมูล, `m` เป็นการเรียกใช้บริการ `multi_pop`, `r` เป็นการเรียกใช้ `remove_top` ส่วน `p` จะเป็นการพิมพ์ข้อมูลใน `stack` ออกมา และ `q` เป็นการจบการทำงาน

- บรรทัดที่มีคำสั่ง `a` และ `m` และ `r` จะตามด้วยตัวเลข 1 ตัว ซึ่งเป็น `parameter` ของฟังก์ชันดังกล่าว
- คำสั่ง `r` เมื่อทำงานเสร็จแล้ว จะพิมพ์ "ขนาดของ `stack` ตั้งต้นที่เหลือ" และจะพิมพ์ข้อมูลออกจาก `stack` ที่

return กลับมาด้วย ไล่จาก top of stack ไปเรื่อย ๆ

- คำสั่ง p จะพิมพ์ข้อมูลออกมาจาก Stack โดยจะพิมพ์ข้อมูลจาก top of stack ไล่ไปเรื่อย ๆ

ตัวอย่าง

ข้อมูลที่พิมพ์เข้าทาง <b>keyboard</b>	ข้อมูลที่เป็นผลจากการทำงานของโปรแกรม
a 1 a 2 a 3 a 4 a 5 p d d p q	Stack size = 5 Data = 5 4 3 2 1 Stack size = 3 Data = 3 2 1
a 10 a 20 a 30 a 40 a 50 m 3 p r 3 p a 100 a 200 a 300 r 2 p m 10 p q	Stack size = 2 Data = 20 10 remove_top Result: Size = 0 Data = 20 10 Stack size = 0 Data = remove_top Result: Size = 1 Data = 300 200 Stack size = 1 Data = 100 Stack size = 0 Data =

คำแนะนำ

- ถ้าทำได้เพียงฟังก์ชันเดียวจากทั้งสองฟังก์ชัน ก็สามารถส่งได้
- ฟังก์ชัน remove\_top นั้นต้อง return เป็น std::stack<T> ไม่ใช่ CP::stack<T>