

# Queue Vertical Split

(1 sec, 512mb)

จงเพิ่มบริการ void CP::queue<T>::v\_split(std::vector<std::queue<T>> &output, size\_t k) const ให้กับ CP::queue โดยฟังก์ชันนี้จะ “แบ่ง” ข้อมูลใน CP::queue ปัจจุบันออกเป็น std::queue จำนวน k ตัว ให้กับ output ซึ่งเป็น std::vector ที่ใช้เก็บผลลัพธ์ของการแบ่ง กล่าวคือ output[i] สำหรับ  $(0 \leq i < k)$  เป็น queue ที่เก็บผลลัพธ์ของการแบ่ง โดยเมื่อเริ่มต้นโปรแกรม output จะเป็น vector เปล่า (output.size() = 0)

การแบ่งของเราจะทำให้ข้อมูลใน CP::queue นั้นแบ่งออกไปยัง std::queue แต่ละตัวในจำนวนพอ ๆ กัน โดยลำดับของข้อมูลใน std::queue (เริ่มจาก front) นั้นจะเรียงตามลำดับข้อมูลใน CP::queue (เริ่มจาก front เช่นกัน) กล่าวคือ หากให้ cp คือ CP::queue ที่เรียกใช้บริการแบ่งนี้ ค่าของ v ในส่วนของโปรแกรมข้างล่างนี้จะต้องเหมือนกัน

```
vector<T> v;
while (!cp.empty()) {
    v.push_back(cp.front());
    cp.pop();
}
```

```
vector<T> v;
for (int i = 0; i < k; i++) {
    while (!output[i].empty()) {
        v.push_back(output[i].front());
        output[i].pop();
    }
}
```

นอกจากนี้ ผลลัพธ์ของการแบ่งจะต้องเป็นดังนี้

- ขนาดของ queue ผลลัพธ์ของการแบ่งแต่ละตัวต้องมีขนาดเท่ากัน ยกเว้นเมื่อขนาดของ CP::queue หาด้วย k ไม่ลงตัว ขนาดของผลลัพธ์แต่ละตัวจะต่างกันได้ไม่เกิน 1
- $\text{output}[a].\text{size}() \geq \text{output}[b].\text{size}()$  เสมอ เมื่อ  $a < b$  กล่าวคือ std::queue ตัวที่มีขนาดใหญ่กว่าจะต้องเป็น output ตัวแรก ๆ เท่านั้น
- ข้อมูลทุกตัวของ CP::queue จะต้องไม่เปลี่ยนแปลง

ตัวอย่างต่อไปนี้จะแสดงผลลัพธ์ของการเรียก v\_split โดยให้ [x,x,x,x] หมายถึงข้อมูลใน queue และให้ front อยู่ด้านซ้าย

CP::queue	k	output
[1,2,3,4,5,6,7,8]	2	{ [1,2,3,4], [5,6,7,8] }
[1,2,3,4,5,6,7,8]	3	{ [1,2,3], [4, 5, 6], [7, 8] }
[1,2,3,4,5,6]	4	{ [1,2], [3,4], [5], [6] }
[1,2,3,4,5,6,7,8]	10	{ [1], [2], [3], [4], [5], [6], [7], [8], [], [] }
[]	4	{ [], [], [], [] }
[1,2,3]	1	{ [1,2,3] }

## คำอธิบายฟังก์ชัน main

main() จะอ่านข้อมูลมา 1 บรรทัด คือ

- บรรทัดแรกประกอบด้วยจำนวนเต็ม n และ k ซึ่งระบุจำนวนข้อมูลใน queue และค่า k ที่ต้องการแบ่ง

หลังจากนั้น main จะใส่ข้อมูลเริ่มที่ 1 จนถึง n เข้าไปใน CP::queue ตามลำดับแล้วทำการเรียก v\_split แล้วพิมพ์ผลลัพธ์ของ output[i] ออกมา

## ชุดข้อมูลทดสอบ

- 20%  $n \leq 100$ ,  $k \leq 2$ ,  $T$  เป็น int
- 30%  $n \% k == 0$
- 50% ไม่มีเงื่อนไขอื่น ๆ

## ข้อบังคับ

- โจทย์ข้อนี้จะมีไฟล์โปรเจกต์ของ Code::Blocks ให้ ซึ่งในไฟล์โปรเจกต์ดังกล่าวจะมีไฟล์ queue.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
- หากใช้ VS Code ให้ทำการ compile ที่ไฟล์ main.cpp
  - \*\* main ที่ใช้จริงใน grader นั้นจะแตกต่างจาก main ที่ได้รับในไฟล์โปรเจกต์เริ่มต้นแต่จะทำการทดสอบในลักษณะเดียวกัน \*\***