

## Split Queue

ในข้อนี้จะให้เขียนบริการของคลาส `CP::queue<T>` ที่ชื่อว่า `std::vector<CP::queue<T> >` `split_queue(int k)` เพื่อการแบ่ง แถวคอยปัจจุบันเป็น  $k$  แถวคอยโดยข้อมูลหน้าสุดจะไปอยู่แถวคอยที่ 0 ข้อมูลตัวถัดไปจะไปอยู่แถวคอยที่ 1 ... ข้อมูลตัวที่  $k-1$  อยู่แถวคอยที่  $k-1$  ข้อมูลตัวที่  $k$  อยู่แถวคอยที่ 0 ..... ไปเรื่อยๆ จนกว่าข้อมูลจะหมดจากแถวคอยและจะทำการนำข้อมูลทั้งหมดออกจากแถวคอยปัจจุบันด้วย ตัวอย่างเช่น หาก `queue<int> q` เก็บข้อมูล {3,9,8,6,7,2,1,3,8,4} ตามลำดับจากหน้าไปหลังอยู่ หากเราเรียก

```
vector<queue<int>> qs = q.split_queue(3)
```

ผลที่ได้คือ `qs[0]` จะเป็นแถวคอยที่เก็บ {3,6,1,4}

`qs[1]` จะเป็นแถวคอยที่เก็บ {9,7,3}

`qs[2]` จะเป็นแถวคอยที่เก็บ {8,2,8}

และ `q` จะไม่มีข้อมูลอยู่

โดยเวลาการทำงานของโปรแกรมจะต้องเป็น  $O(n+k)$  เมื่อ  $n$  คือจำนวนของข้อมูลใน `queue`

### ข้อบังคับ

นิสิตสามารถใช้บริการใดๆที่มีใน `CP::queue` หรือ STL ได้โดยไม่มีข้อจำกัด โจทย์ข้อนี้จะมีไฟล์โปรเจกต์ของ `code::block` ให้ ซึ่งในโปรเจกต์ดังกล่าวจะมีไฟล์ `queue.h`, `student.h` และ `main.cpp` อยู่ ให้นิสิตเขียน code เพิ่มเติมลงไปในไฟล์ `student.h` ในฟังก์ชัน `split_queue` เท่านั้น และการส่งไฟล์ขึ้น grader ให้ส่งเฉพาะไฟล์ `student.h` โดยใน grader จะมี main ต่างหากที่ใช้ในการตรวจซึ่งจะทดสอบความถูกต้องของ `split_queue` ในกรณีการใช้งานที่ซับซ้อนกว่านี้

### คำอธิบายฟังก์ชัน `main()`

โปรแกรมจะรับจำนวนข้อมูล  $n$  จากคีย์บอร์ดจากนั้นจะรับตัวเลขมา  $n$  ตัว เอาใส่ไว้ใน `queue<int> q` แล้วจึงรับค่า  $k$  แล้วเรียก `split_queue` โดยผลรับที่ได้ใส่ไว้ใน `qs` แล้วทำการแสดงผล `qs` แต่ละอัน และ แสดง `q`

### ตัวอย่าง

ข้อมูลที่พิมพ์เข้าทาง keyboard	ข้อมูลที่เป็นผลจากการทำงานของโปรแกรม
8 5 7 1 10 45 3 6 4 3	qs[0] = [5 10 6] qs[1] = [7 45 4] qs[2] = [1 3] q = []

### คำแนะนำ

1. ทุกชุดข้อมูลมี  $0 < k \leq n \leq 30000$  โดยมี 70% ที่  $0 < k \leq n \leq 100$
2. split\_queue จะต้องทำงานถูกต้องในทุกๆค่า  $k > 0$  ที่อาจจะมากกว่าจำนวนข้อมูลก็ได้
3. split\_queue จะต้องทำงานถูกต้องไม่ว่า queue จะเคยโดยเพิ่มหรือลดข้อมูลไปแล้วกี่ครั้งก็ตาม