

Heap K-th

จงเขียนฟังก์ชัน `T get_kth(size_t k) const` เพิ่มเติมให้กับคลาส `CP::priority_queue` ซึ่งฟังก์ชันนี้จะคืนค่าข้อมูลที่ “จะถูก pop ออกมาในครั้งที่ k” จาก priority queue นี้ ตัวอย่างเช่น เมื่อ k มีค่าเป็น 1 ฟังก์ชันนี้จะต้องคืนค่าเดียวกับ `top()` กลับมาให้ หรือเมื่อ k มีค่าเป็น 3 ฟังก์ชันนี้ควรจะคืนค่า `top()` ที่เกิดขึ้นหลังจากการเรียก `pop()` สองครั้งนั่นเอง (กล่าวอีกนัยหนึ่ง `get_kth` จะคืนค่าที่มากที่สุดเป็นลำดับที่ k จาก priority_queue นี้) เมื่อพิจารณาค่ามากน้อยตาม comparator ที่ใช้ใน priority_queue นี้)

อย่างไรก็ตาม รับประกันว่าค่า k ที่เรียกนั้นจะอยู่ในช่วง 1 ถึง 3 เท่านั้น และจะไม่มีการเรียกค่า k ที่มากกว่า `mSize`แน่นอน

ให้สังเกตว่าฟังก์ชันนี้ถูกประกาศเป็นแบบ `const` ซึ่งหมายความว่าฟังก์ชันนี้จะต้องไม่ทำการเปลี่ยนแปลงค่าใด ๆ ใน priority_queue ของเรา

ข้อบังคับ

- โจทย์ข้อนี้จะมีไฟล์โปรเจ็คของ `Code::Blocks` ให้ ซึ่งในไฟล์โปรเจ็คดังกล่าวจะมีไฟล์ `vector.h`, `main.cpp` และ `student.h` อยู่ ให้คุณเขียน code เพิ่มเติมลงในไฟล์ `student.h` เท่านั้น และการส่งไฟล์เข้าสู่ระบบ grader ให้ส่งเฉพาะไฟล์ `student.h` เท่านั้น
 - ในไฟล์ `student.h` ดังกล่าวจะต้องไม่ทำการอ่านเขียนข้อมูลใด ๆ ไปยังหน้าจอหรือคีย์บอร์ดหรือไฟล์ใด ๆ

คำอธิบายฟังก์ชัน `main()`

`main` จะอ่านข้อมูลมาสองบรรทัด ตามรูปแบบนี้

- บรรทัดแรกประกอบด้วยจำนวนเต็ม `n` และ `k`
- บรรทัดที่สองประกอบด้วยจำนวนเต็มจำนวน `n` ตัว ซึ่งจะถูกนำไปใส่ใน `priority_queue` หลังจากนั้น `main` จะเรียก `get_kth(k)` แล้วทำการพิมพ์ค่าดังกล่าวออกมาทางหน้าจอ

ข้อควรระวัง

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

- ฟังก์ชัน `get_kth` นั้นจะพิจารณาค่าตาม comparator ที่ใช้ใน `priority_queue` ด้วย

ตัวอย่างการใช้งาน `get_kth()`

```
int main() {
    int n = 10;
    CP::priority_queue<int> pq;
    while (n--) pq.push(n);
    cout << pq.get_kth(3) << endl; // แสดงค่า 7
    cout << pq.get_kth(2) << endl; // แสดงค่า 8
    cout << pq.get_kth(1) << endl; // แสดงค่า 9

    n = 10;
    CP::priority_queue<int, greater<int>> pq2;
    while (n--) pq2.push(n);
    cout << pq2.get_kth(3) << endl; // แสดงค่า 2
    cout << pq2.get_kth(2) << endl; // แสดงค่า 1
    cout << pq2.get_kth(1) << endl; // แสดงค่า 0
}
```