Leaves Count

จงเพิ่มบริการให้กับคลาส CP::map_bst โดยให้เพิ่มฟังก์ชัน size_t leaves_count( ) ซึ่งจะทำการคืนค่าจำนวนปมที่เป็นปมใบ (ปมที่ไม่มีลูก) ในต้นไม้ Binary Search Tree นี้

ข้อบังคับ

โจทย์ข้อนี้จะมีไฟล์โปรเจ็คของ Code::Blocks ให้ ซึ่งในโปรเจ็คดังกล่าวจะมีไฟล์ map_bst.h, main.cpp และ student.h อยู่ ให้นิสิตเขียน code เพิ่มเติมลงไปในไฟล์ student.h เท่านั้น และการส่งไฟล์ขึ้น grader ให้ส่งเฉพาะไฟล์ student.h นิสิตสามารถแก้ไข student.h ได้โดยอิสระ และสามารถเรียกใช้ฟังก์ชันใด ๆ ใน stl รวมถึงของ map_bst ได้

***** ห้ามทำการพิมพ์ข้อมูลทางจอภาพหรืออ่านข้อมูลจากคีย์บอร์ดในไฟล์ student.h ที่ส่งมายัง grader โดยเด็ดขาด *****

คำแนะนำ

ข้อนี้สามารถทำได้โดยง่ายโดยเขียนโปรแกรมแบบ Recursive และเพื่อให้การเขียนโปรแกรมแบบ recursive ทำได้สะดวก นิสิตสามารถเขียนฟังก์ชัน size_t leaves_count(node* n) เพื่อนับจำนวนใบจาก bst ที่มีรากเป็น n ได้ ฟังก์ชันดังกล่าวนั้นมีโครงอยู่ใน student.h แล้ว ถ้าหากนิสิตต้องการจะใช้ สามารถเขียนรายละเอียดของฟังก์ชันดังกล่าวได้เลย รวมถึงสามารถให้ leaves_count( ) นั้นเรียกใช้ leaves_count(node* n) ได้ด้วย

คำอธิบายฟังก์ชัน main()

โปรแกรมจะเริ่มต้นจาก CP::map_bst<int,int> m ซึ่งเป็น bst ว่าง หลังจากนั้น main จะทำการอ่านข้อมูลจากคีย์บอร์ดดังรูปแบบต่อไปนี้

บรรทัดแรกประกอบด้วยจำนวนเต็มหนึ่งค่าคือ n ซึ่งระบุจำนวนข้อมูลที่ต้องการใส่เข้าไปใน bst

บรรทัดที่สองประกอบด้วยจำนวนเต็ม n ค่า คือข้อมูลที่จะอยู่ใน bst โดย main จะเพิ่มข้อมูลดังกล่าวเข้าไปใน bst ตามลำดับ โดยก่อนที่จะใส่ข้อมูลดังกล่าวเข้าไปในแต่ละค่านั้น main จะเรียก leaves_count และพิมพ์ค่าดังกล่าวออกมาก่อน และเมื่อใส่ข้อมูลจนครบหมดแล้ว main จะเรียก leaves_count อีกครั้งด้วย

***** main ใน grader นั้นจะแตกต่างจาก main ที่นิสิตได้รับ แต่จะเป็นการทดสอบในลักษณะเดียวกัน ขอให้เขียนฟังก์ชั่นเพิ่มเติมให้ตรงตามนิยามที่กำหนดไว้ข้างต้น *****

ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
1 10	Leaves = 0 Leaves = 1
5 3 1 2 4 5	Leaves = 0 Leaves = 1 Leaves = 1 Leaves = 1 Leaves = 2 Leaves = 2