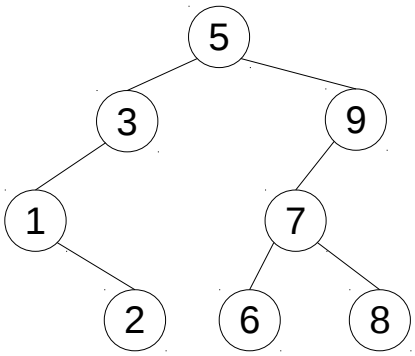
Count Unary Node



จงเขียนฟังก์ชัน size_t count_unary() สำหรับคลาส CPP::map_bst ซึ่งจะนับจำนวนปมประเภท unary ใน ต้นไม้ของเรา โดยปม unary นั้นก็คือปมที่มีลูกเพียงลูกเดียวเท่านั้น ตัวอย่างเช่นต้นไม้ในรูปด้านข้างนี้ มีปม unary อยู่ทั้งหมด 3 ปมซึ่งได้แก่ปมที่มีค่า 3, 1 และ 9

ฟังก์ชันดังกล่าวจะต้องคืนค่าจำนวนปมที่เป็น unary มา ในกรณีที่เป็นต้นไม้ว่าง ให้คืนค่า 0

ข้อบังคับ

ฟังก์ชันดังกล่าวจะต้องไม่ทำการเปลี่ยนแปลงข้อมูลของต้นไม้ดังกล่าว เนื่องจาก count_nary ประกาศเป็น const ไว้ โจทย์ข้อนี้จะมีไฟล์โปรเจ็คของ code::block ให้ ซึ่งในโปรเจ็คดังกล่าวจะมีไฟล์ map_bst.h, main.cpp และ student.h อยู่ ให้นิสิตเขียน code เพิ่มเติมลงไปในไฟล์ student.h เท่านั้น และการส่งไฟล์ขึ้น grader ให้ส่งเฉพาะไฟล์ student.h เท่านั้น

การเขียน count_unary นั้นสามารถทำได้ง่ายด้วยการเขียนโปรแกรมแบบ recursive ในไฟล์ map_bst.h และ student.h นั้นจะมีโครงฟังก์ชัน size_t process(node *ptr) เตรียมไว้ให้ใช้สำหรับการเขียนโปรแกรมแบบ recursive แล้ว นิสิตไม่จำเป็นต้องใช้ฟังก์ชันดังกล่าว แต่การใช้ฟังก์ชันดังกล่าวจะทำให้การทำงานนั้นง่ายขึ้นมาก

เนื่องจากนิสิตไม่สามรถแก้ไขไฟล์ map_bst.h ได้ ดังนั้น นิสิตจึงไม่สามารถเพิ่มฟังก์ชันอื่นใดให้กับคลาสนี้ได้

คำอธิบายฟังก์ชัน main()

ฟังก์ชัน main จะอ่านค่าจำนวนข้อมูล n จากคีย์บอร์ด หลังจากนั้นจะทำการเข้าสู่วงวนจำนวน n รอบ โดยที่แต่ละรอบจะอ่านค่าข้อมูลจากคีย์บอร์ด และนำเอาค่าดังกล่าวใส่ลงไปใน map_bst และเรียกฟังก์ชัน count_unary พร้อมกับแสดงค่า จำนวนข้อมูลในต้นไม้ และ count_unary หลังจากใส่ข้อมูลดังกล่าวเข้าไป

ข้อกำหนดขนาดข้อมูล

รับประกันว่าขนาดของข้อมูลใน vector ทั้งสองตัวจะไม่เกิน 5,000 ตัว

ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
5 1 2 3 4 5	Size = 1 unary_count = 0 Size = 2 unary_count = 1 Size = 3 unary_count = 2 Size = 4 unary_count = 3 Size = 5 unary_count = 4
8 5 3 1 2 9 7 6 8 //หมายเหตุ: ข้อมูลในตัวอย่างนี้จะสร้างต้นไม้เหมือนรูปด้านบน	Size = 1 unary_count = 0 Size = 2 unary_count = 1 Size = 3 unary_count = 2 Size = 4 unary_count = 3 Size = 5 unary_count = 2 Size = 6 unary_count = 3 Size = 7 unary_count = 4 Size = 8 unary_count = 3