

(1 Second, 16 MiB)

ปรมาจารย์มารสวรค์วิหคเพลิงบรรพกาลผู้มีตำนาน



หลังจากนักล่าอสูรได้มาถึงประสาทไร่ของเซต ด้วยความที่นักล่าอสูรในยุคนี้ป่าเถื่อนเป็นอย่างมาก แค่ว่าพวกเขาเดินผ่านอรัญแสงตะวันอันร้อนแรง ก็ได้ทำการแผดเผาทุกสิ่งทุกอย่าง มารสวรค์เทพพระเจ้าบรรพชนแห่งไตรธรรณีโลกที่เป็นอสูรมารผจญข้างแรม Ryzen 5 3600 นั้นไม่อยากโดนนักล่าอสูรผู้โหดร้ายแผดเผา จึงไปขอนาคิเมะที่สามารถควบคุมประสาทไร่ขอบเขต เพื่อที่จะยื้อเวลาก่อนนักล่าอสูรผู้มีอรัญแสงตะวันร้อนแรงจะเข้ามาถึงตัวของมารสวรค์เทพพระเจ้าบรรพชนแห่งไตรธรรณีโลก ซึ่งมารสวรค์เทพพระเจ้าบรรพชนแห่งไตรธรรณีโลกรู้ว่่านักล่าอสูรจะบุกเข้ามาเป็นเส้นตรง โดยคุณสามารถออกคำสั่งแก่นาคิเมะได้ แต่ว่าการที่เขาจะออกคำสั่งเขาต้องเข้าใจถึงแก่นแท้ของมนต์อสูรโลหิต ซึ่งเขาใช้เวลาเหลือไม่มาก จึงทำให้เขาเข้าใจแค่วิธีการสอดส่องตำแหน่งของเหล่านักล่าอสูร (ฟังค์ชัน peek) และรู้ว่าข้อมูลของมนต์อสูรโลหิตของนาคิเมะจะเก็บเป็น node ด้วยความที่คุณเป็นเพื่อนรักของเขาและไม่อยากโดนอรัญของนักล่าอสูรจึงจำเป็นต้องทำความเข้าใจและสร้างฟังค์ชันเพื่อลดระยะเวลาการใช้มนต์อสูรโลหิตในแต่ละครั้ง

ภายในข้อมูลของนักล่าอสูรแต่ละคนจะมีข้อมูลดังนี้

Slayer: ชื่อของนักล่าอสูรแต่ละคน

โดยที่คุณต้องทำความเข้าใจมนต์อสูรโลหิตทั้งหมดอย่างถ่องแท้และเขียนออกมาเป็นฟังก์ชันทั้งหมด ดังนี้

ฟังก์ชัน createNode	รับ parameter (const string &Slayer)	คืนค่า node*
คุณต้องสร้างที่เก็บข้อมูลของนักล่าอสูรแต่ละคนในรูปแบบของ node และคืนค่า node* กลับไปเช่น createNode("Tanjiro"); สร้าง node ใหม่ที่มีค่า Slayer คือ "Tanjiro" และคืนค่ากลับไป		

ฟังก์ชัน insert	รับ parameter (const string &Slayer)	คืนค่า ไม่มี
ทำการเพิ่มข้อมูลของนักล่าอสูรที่เขามาสอบพบเข้าไปในชุดข้อมูลของมนต์อสูรโลหิต โดยในแต่ละคนจะมีชื่อของตัวเอง (Slayer) เช่น insert("Tanjiro"); insert("Zenitsu"); insert("Inosuke"); insert("Kanao"); insert("Giyu"); เพิ่มข้อมูลชื่อว่า "Tanjiro" เข้าไปในชุดข้อมูลของมนต์อสูรโลหิต และเพิ่มข้อมูล "Zenitsu", "Inosuke", "Kanao", "Giyu" ตามเข้าไปในชุดข้อมูลของมนต์อสูรโลหิต		

ฟังก์ชัน push	รับ parameter (const string &Slayer) การันตีว่าจะเลือกข้อมูลที่มีอยู่ในชุดข้อมูลเสมอ	คืนค่า ไม่มี
<p>นาคิเมะจะทำการผลักดันนักล่าอสูรหนึ่งคนไปอยู่ข้างหลังของนักล่าอสูรที่อยู่ข้างหลังก่อนที่จะโดนผลัก เช่น</p> <p>push("Zenitsu");</p> <p>push("Giyu");</p> <p>นาคิเมะทำการผลัก "Zenitsu" ให้ไปอยู่ข้างหลัง "Inosuke" ทำให้ชุดข้อมูลกลายเป็น "Tanjiro", "Inosuke", "Zenitsu", "Kanao", "Giyu" และผลัก "Giyu" ทำให้ชุดข้อมูลกลายเป็น "Tanjiro", "Inosuke", "Zenitsu", "Kanao", "Giyu"</p>		

ฟังก์ชัน shuffle	รับ parameter (const string &Slayer, const string &Slayer2) การันตีว่าจะเลือกข้อมูลที่มีอยู่ในชุดข้อมูลเสมอและ Slayer ไม่เท่ากับ Slayer2	คืนค่า ไม่มี
<p>นาคิเมะจะทำการเลือกนักล่าอสูรสองคนโดยให้นักล่าอสูรคนแรกไปอยู่ข้างหลังของนักล่าอสูรคนที่ 2 เช่น</p> <p>shuffle("Tanjiro","Kanao");</p> <p>นาคิเมะทำการเลือก "Tanjiro" และ "Kanao" ทำให้ชุดข้อมูลกลายเป็น "Inosuke", "Zenitsu", "Kanao", "Tanjiro", "Giyu"</p>		

ฟังก์ชัน deleteRange	รับ parameter (int l, int r) ($0 \leq l \leq r < \text{ขนาดของชุดข้อมูลในช่วงขณะนั้น}$)	คืนค่า ไม่มี
<p>นาคิเมะจะทำการกำจัดนักล้าอสูรตั้งแต่ตำแหน่งที่ l ไปจนถึงตำแหน่งที่ r เช่น</p> <p>deleteRange(0,2);</p> <p>นาคิเมะจะจัดการนักล้าอสูรตั้งแต่ช่วงที่ตำแหน่งที่ 0 ไปจนถึงตำแหน่งที่ 2 ทำให้ชุดข้อมูลกลายเป็น "Tanjiro", "Giyu"</p>		

ตัวอย่าง

หากเขียนโปรแกรมให้กับฟังก์ชันต่างๆ ถูกต้อง

```
insert("Tanjiro");
insert("Zenitsu");
insert("Inosuke");
insert("Kanao");
insert("Giyu");
peek();
push("Zenitsu");
peek();
push("Giyu");
peek();
shuffle("Tanjiro", "Kanao");
peek();
deleteRange(0,2);
peek();
```

ข้อมูลนำเข้า	ข้อมูลนำออก
	Tanjiro Zenitsu Inosuke Kanao Giyu Tanjiro Inosuke Zenitsu Kanao Giyu Tanjiro Inosuke Zenitsu Kanao Giyu Inosuke Zenitsu Kanao Tanjiro Giyu Tanjiro Giyu

ปัญหาย่อย

1. (10 คะแนน) ทำฟังก์ชัน **insert** ได้
2. (20 คะแนน) ทำฟังก์ชัน **push** และ **insert** ได้
3. (20 คะแนน) ทำฟังก์ชัน **shuffle** และ **insert** ได้
4. (20 คะแนน) ทำฟังก์ชัน **deleteRange** และ **insert** ได้
5. (30 คะแนน) ทำฟังก์ชันทุกฟังก์ชัน

คำแนะนำ

ให้โหลดไฟล์ `singlefile.cpp` จากนั้นเขียนโปรแกรมแล้วทดสอบให้เรียบร้อย จากนั้นโหลดไฟล์ `function.cpp` แล้วเติมเฉพาะฟังก์ชันให้เรียบร้อยจึง submit ขึ้น grader และไม่ต้องเขียนฟังก์ชัน `peek`