

ปัญหา ตรวจสอบการสะกดคำ2 [SpellChecking2] (6 คะแนน)

ในปัญหานี้ นอกจากจะตรวจคำผิดแล้วเรายังจะแก้คำให้ถูกต้องด้วย โดยมีเงื่อนไขการแก้ไขดังนี้

- (1) เราจะแก้ไขเฉพาะคำที่ไม่มีอยู่ในพจนานุกรม
- (2) เราจะเลือกคำในพจนานุกรมที่ใกล้เคียงกับคำที่จะแก้มากที่สุดมาแทนที่
- (3) ความใกล้เคียงวัดจากจำนวนตัวอักษรที่ตรงกันตามตำแหน่งของตัวอักษร เช่น ถ้าในพจนานุกรมมีคำว่า **ant** และ **bog** ส่วนคำที่จะแก้ขึ้นนั้นคือ **btn** เราจะถือว่าคำนี้ใกล้เคียงกับ **bog** มากกว่าเพราะตัวอักษรตรงตามตำแหน่งกันมากที่สุด¹ และจะแก้คำว่า **btn** เป็นคำว่า **bog**

แต่หากคำที่จะแก้ไขคือ **bnt** เราจะถือว่ามันใกล้เคียงกับคำว่า **ant** มากกว่า และจะแก้คำว่า **bnt** ไปเป็น **ant**

ในกรณีที่คำในพจนานุกรมที่ใกล้เคียงกับคำผิดมากกว่าหนึ่งคำ เราจะถือตามลำดับการปรากฏ และเลือกคำศัพท์ที่มาก่อน เช่น ถ้าหากพจนานุกรมมีสองคำคือ **ant** และ **anti** ตามลำดับ ส่วนคำผิดคือ **bat** คำนี้จะถูกแก้เป็น **ant** แต่ถ้าลำดับคำศัพท์ในพจนานุกรมเปลี่ยนไปเป็น **anti** และ **ant** คำว่า **bat** ก็จะถูกเปลี่ยนไปเป็น **anti** แทน

รูปแบบข้อมูลเข้า

บรรทัดแรก	เป็นค่า N ซึ่งก็คือจำนวนคำศัพท์ในพจนานุกรม โดยที่ $1 < N \leq 10,000$
บรรทัดที่สอง	เป็นคำศัพท์ทั้งหมดในพจนานุกรม ซึ่งคำศัพท์แต่ละคำคั่นด้วยช่องว่างหนึ่งช่อง และคำศัพท์ทุกคำจะไม่มีช่องว่างอยู่ (ทำให้มั่นใจได้ว่าเมื่อเจอช่องว่างหมายถึงสิ้นสุดคำศัพท์นั้น ๆ)
บรรทัดที่สาม	เป็นค่า K ซึ่งคือจำนวนคำที่ต้องการตรวจสอบและแก้ไข (ถ้าหากพบว่าเป็นคำผิด) โดยที่ $1 < K \leq 300$
บรรทัดที่เหลือ	เป็นคำที่ต้องการตรวจสอบและแก้ไข หนึ่งคำต่อบรรทัด

¹ คือมีตัว **b** ตรงกันหนึ่งตำแหน่ง ในขณะที่ **ant** ไม่มีอักษรตำแหน่งใดที่ตรงกับ **btn** เลย

หมายเหตุ เช่นเดียวกับข้อที่แล้ว ตัวอักษรในคำศัพท์และคำที่ต้องการตรวจสอบทั้งหมดเป็นภาษาอังกฤษตัวพิมพ์เล็ก

รูปแบบผลลัพธ์

มี K บรรทัด แต่ละบรรทัดเป็นคำที่ผ่านการตรวจสอบและแก้ไขแล้ว (ถ้าเป็นคำผิด) นั่นคือถ้าหากว่าคำที่ใส่เข้ามาถูกต้องตามพจนานุกรมโปรแกรมก็จะพิมพ์คำนั้นออกมา แต่หากคำที่ใส่เข้ามาผิด โปรแกรมก็จะพิมพ์คำที่ใกล้เคียงที่สุดออกมา นอกจากนี้ลำดับผลลัพธ์ในแต่ละบรรทัดจะเรียงตามลำดับในข้อมูลเข้า

ตัวอย่าง

ข้อมูลเข้า	ผลลัพธ์
5 ant bee cat dog elephant 13 cat cot dug dog cne ant egg elep phant b see foggy zzzzzzz	 cat cat dog dog ant ant dog elephant ant bee bee dog ant

หมายเหตุ 1 ไม่ต้องพิมพ์บรรทัดเปล่า 3 อันแรกตามที่อยู่ในผลลัพธ์ตัวอย่าง เพราะในตัวอย่างใส่มาไว้เพียงเพื่อให้อ่านง่ายว่าคำแต่ละคำจะให้ผลลัพธ์เป็นอะไร

หมายเหตุ 2 ตอนที่ไม่มีตัวอักษรในคำศัพท์ใดที่ตรงกับอักษรในคำที่กำลังตรวจสอบ แสดงว่าจำนวนตัวอักษรที่ตรงกันมากที่สุดคือ 0 และโปรแกรมจะเลือกเอาคำศัพท์ตัวแรกมาเป็นคำตอบ (ดูตัวอย่างอันสุดท้ายที่เป็น zzzzzzz) [ยังมีอีกตัวอย่างในหน้าถัดไป]

ข้อมูลเข้า	ผลลัพธ์
6 jojo naruto kenshiro goku vegita ash 16 jonathan boruto kenshiro kensington kenkennarutovegitash goruto gokuto naku vekuto ash ashita gojo joku naruhiro kensto kenuto	jojo naruto kenshiro kenshiro kenshiro kenshiro naruto goku naruto naruto ash vegita jojo goku naruto kenshiro naruto

คำแนะนำ 1 การจะตรวจว่าคำศัพท์ไหนใกล้เคียงกับคำที่กำลังตรวจสอบมากที่สุด ทำได้ด้วยการเทียบตัวอักษรในคำศัพท์และคำที่กำลังตรวจสอบทีละตัว ถ้าพบว่าตัวอักษรที่ตำแหน่งเดียวกันตรงกัน ก็ให้นับหนึ่ง และต้องระวังไม่ให้การตรวจล้นออกไปข้างนอกสตริงอันใดอันหนึ่ง ซึ่งเหตุที่ต้องระวังก็เพราะว่าความยาวของสตริงทั้งสองอาจจะไม่เท่ากัน

คำแนะนำ 2 เวลาที่จะเทียบสตริงว่าเท่ากันหรือไม่ เราใช้ `equals` แต่ถ้าเป็นตัวอักษรซึ่งเป็น `primitive data type` เราใช้เครื่องหมาย `==` ได้เลยโดยตรง
[ยังมีคำแนะนำอีกข้อในหน้าถัดไป]

คำแนะนำ 3 การแยกเมธอดตรวจสอบความใกล้เคียงออกมา จะทำให้โค้ดเราดูง่ายกว่าเดิมมาก ซึ่งหากไม่แยกเมธอดออกมา เราจะต้องเขียนลูปสามชั้น แต่ถ้าแยกออกมาลูปจะเป็นสองชั้นที่ใช้วิธีการคล้ายกับโจทย์หลายข้อที่เคยมีมาก่อนหน้า สำหรับเมธอดสำหรับการนับจำนวนตัวอักษรที่ตรงกันนั้น เราอาจจะประกาศเป็น

```
static int countMatch(String vocab, String word) {  
    int match = 0;  
  
    // YOUR CODE HERE  
  
    return match;  
}
```