

ปัญหา จัดจ้องอักขระ 2 (CharWatch2) [6 คะแนน]

เช่นเดิมกับข้อ CharWatch อันแรก แต่คราวนี้สตริงหลักยาว ๆ ที่รับเข้ามามีมากกว่า 1 สตริง และเราไม่สนใจการนับการอ้างอิงตำแหน่งที่ผิดพลาด ซึ่งการอ้างอิงตำแหน่งในข้อนี้ เราจะใช้เลขเดียวกับทุกสตริง ถ้าตำแหน่งนั้นตรงกับอักขระที่สนใจในสตริงหลักใด เราก็นับการพบตัวอักขระนั้นให้ 1 ครั้งสำหรับแต่ละสตริง (ถ้าหากมีหลายสตริงที่พบอักขระที่สนใจในตำแหน่งดังกล่าว เราก็นับให้หลายครั้ง และถ้าหากตำแหน่งที่อ้างอิงอยู่นอกขอบเขตของสตริงอันใด เราก็ไม่ต้องพิจารณาสตริงนั้น ส่วนสตริงอื่น ๆ ให้พิจารณาตามปกติ)

จงเขียนโปรแกรมที่นับจำนวนการอ้างอิงอักขระที่สนใจแต่ละตัว

ข้อมูลเข้า

บรรทัดแรก	เป็นอักขระ 3 ตัวที่สนใจ คั่นด้วยช่องว่าง
บรรทัดที่ 2	เป็นจำนวนเต็มบวก K โดยที่ $2 \leq K \leq 30$ ซึ่งแทนจำนวนสตริงหลัก
อีก K บรรทัดถัดมา	เป็นสตริงหลักความยาวไม่เกิน 100,000 ตัวอักขระ หนึ่งสตริงต่อบรรทัด ไม่มีช่องว่างในสตริงดังกล่าว ภายในสตริงนี้จะมีเฉพาะตัวพิมพ์ใหญ่และอักขระขีดเส้นใต้ _
บรรทัดที่ K + 3	จำนวนเต็มบวก N โดยที่ $3 \leq N \leq 100,000$
บรรทัดที่ K + 4	เลขตำแหน่งที่อ้างอิงทั้ง N ค่า คั่นด้วยช่องว่าง

ผลลัพธ์

มี 3 บรรทัด สามบรรทัดแรกเป็นจำนวนครั้งที่การอ้างอิงตรงกับอักขระที่สนใจ โดยบรรทัดแรกเป็นของอักขระตัวแรกที่สนใจและบรรทัดที่ 2 และ 3 เป็นของอักขระตัวที่ 2 และ 3 ตามลำดับ

ตัวอย่าง

ข้อมูลเข้า	<p>S I L 2 SILPA PHEERASRI 11 0 1 2 3 4 5 6 7 8 9 10</p> <p>[หมายเหตุ ข้อมูลเข้านี้เป็นแบบง่าย เอาไว้ทดสอบว่าเราป้องกันการอ้างถึงตำแหน่งนอกสตริงได้อย่างถูกต้องหรือไม่]</p>
ผลลัพธ์	<p>2 2 1</p>
ข้อมูลเข้า	<p>P O S 5 TWO_POKEMON_STOPS SLOWPOKE POLIWAG GLOOM_SANDSHREW BUS_STOPS 20 -1 2 4 5 1 1 3 6 7 8 9 10 11 12 13 14 15 16 17 18</p>
ผลลัพธ์	<p>6 10 9</p>

ตัวช่วย

```
public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);

    // Read inputs as strings of one character
    String s1 = scan.next();
    String s2 = scan.next();
    String s3 = scan.next();

    // Extract the first char to obtain the characters we need.
    char char1 = s1.charAt(0);
    char char2 = s2.charAt(0);
    char char3 = s3.charAt(0);

    // Read the main string
    final int K = scan.nextInt();
    String[] major = new String[K];
    for(int i = 0; i < K; ++i) {
        major[i] = scan.next();
    }

    final int N = scan.nextInt();

    ... ที่เหลือคิดต่อเอาเอง แต่มีแนวโน้มว่าคุณต้องซ้อนลูปหรือไม่ก็เขียนแยกเมธอด ...
}
```