

ปัญหา โควิด19ชนะ 1 [Covid19Wins1]

แม้ประเทศไทยดูเหมือนจะจัดการโรคโควิด-19 ได้ดี ทว่าอยู่ดี ๆ ก็มีกลุ่มบุคคลจากต่างประเทศที่ได้รับสิทธิพิเศษ เช่น ทหารหรือนักการทูตที่เข้าประเทศไทยมาแต่ไม่ได้กักตัวตามมาตรฐานสาธารณสุขที่ดี ทำให้ประเทศไทยตกอยู่ในภาวะเสี่ยงของการระบาดระลอกที่สอง

เพื่อแก้ไขปัญหา เราจึงต้องนำข้อมูลการเดินทางหรือการเข้าใช้บริการสถานที่ต่าง ๆ ของแต่ละคนมาเทียบกัน ซึ่งหากมีใครที่เข้าใช้บริการช่วงเดียวกับคนที่ติดโควิด ก็จะต้องเรียกมาตรวจสอบการติดเชื้อทันที

การเก็บข้อมูลจะเริ่มจากการเข้าใช้สถานที่และเวลา โดยจะสร้างคลาส Visitation ซึ่งจะเก็บข้อมูล 3 อย่างคือ 1. รหัสประจำสถานที่ (คล้าย ๆ รหัสประจำตัวนักศึกษาที่จะไม่ซ้ำกันระหว่างนักศึกษาสองคนหรือสถานที่สองแห่ง) 2. วันที่เข้าใช้สถานที่ และ 3. เวลาที่เข้าใช้สถานที่ ทั้งนี้กำหนดให้รหัสประจำสถานที่และวันที่เข้าใช้สถานที่เป็นจำนวนเต็มบวก ส่วนเวลาที่เข้าใช้สถานที่จะระบุเป็นชุดตัวเลขจำนวนเต็ม 0 ถึง 23 (อาจจะมากกว่าหนึ่งตัว) เพราะใช้แทนว่าเวลาที่นาฬิกาที่ไปใช้สถานที่ดังกล่าว เช่น ถ้าระบุว่า 0, 7, 9, 21, 22, 23 แสดงว่าไปใช้สถานที่ดังกล่าวเวลา 0, 7, 9, 21, 22, 23 นาฬิกา

เพื่อบรรลุมิติวัตถุประสงค์ของการเก็บข้อมูลดังกล่าวเราจะสร้างคลาส Visitation ซึ่งมีข้อกำหนด¹ เพิ่มเติมจากที่กล่าวไปแล้วดังนี้

1. ตัวสร้างของคลาสจะรับค่ามาสองค่าคือรหัสประจำสถานที่และวันที่เข้าใช้สถานที่ ซึ่งตัวสร้างจะต้องเก็บข้อมูลเหล่านี้ไว้ในตัวแปรสมาชิกคลาส นอกจากนี้ตัวสร้างจะต้องเตรียมพื้นที่เก็บข้อมูลว่าสถานที่นี้ถูกเข้าใช้ ณ เวลาใดบ้าง (แนะนำว่าใช้อาเรย์ที่มี 24 ช่อง เพราะเวลากำหนดเป็นรายชั่วโมงจาก 0 ถึง 23 นาฬิกา แต่จะใช้วิธีอื่นก็ไม่ว่ากัน)
2. มีเมธอด **visit** ซึ่งรับพารามิเตอร์หนึ่งตัวเป็นเวลา (หน่วยเป็นนาฬิกา) ที่ระบุว่าคนนี้เข้าใช้สถานที่ ณ ชั่วโมงใดบ้างในวันที่ระบุมาให้ตัวสร้าง ซึ่งหากเวลาติดลบหรือเกิน 23

¹ คำว่า “ข้อกำหนด” ก็คือ “สิ่งที่กำหนดให้ทำตาม ถ้าไม่ทำตามถือว่าผิด” ไม่ใช่ไปเขียนอะไรอย่างอื่นมาแล้วได้คำตอบถูกแต่ผิดข้อกำหนด พอโดนหักคะแนนก็มาโวยวายว่า “อาจารย์ไม่ได้ห้าม จะมาหักคะแนนกันได้ไง” ดังนั้นขอให้เข้าใจคำว่าข้อกำหนดตามนี้ด้วย

จะไม่เก็บข้อมูลนั้นไว้ เพราะถือว่าข้อมูลไม่ถูกต้อง และในกรณีที่ปัญหานี้ เมธอด **visit** จะพิมพ์ข้อความเตือนว่า “invalid hour: x” โดยให้แทน x ด้วยเลข ชั่วโมงที่ผิดพลาด เช่น **invalid hour: 27** เป็นต้น

- มีเมธอด **count** ซึ่งไม่รับพารามิเตอร์ใด แต่จะคืนจำนวนเต็มที่แสดงจำนวนชั่วโมงในการเข้าใช้สถานที่ที่บันทึกได้ เช่นถ้าเข้าใช้เวลา 0, 7, 9, 21, 22, 23 นาฬิกา เมธอดก็จะคืนค่า 6 กลับไปเป็นผลลัพธ์ (**return** ค่า 6) ถ้าไม่มีการเข้าใช้สถานที่เลยก็คืนเลข 0 เป็นผลลัพธ์ (อนุญาตให้นับจำนวนล่วงหน้าในเมธอด **visit** ได้ และ **หากต้องการสร้างตัวแปรสมาชิกคลาสสำหรับการนับเพิ่มเติมก็สามารถทำได้**)
- มีเมธอด **printVisitHours()** ซึ่งจะพิมพ์ข้อมูลออกมา 4 บรรทัดตามลำดับดังนี้ (1) รหัสสถานที่, (2) วันที่, (3) จำนวนชั่วโมงที่เข้าใช้สถานที่ในวันนั้น² และ (4) รายการชั่วโมงที่เข้าใช้ เรียงจากน้อยไปมาก ซึ่งหากแท้จริงไม่มีการเข้าใช้งานเลย ให้พิมพ์ว่า “no visit”

สำหรับตัวเมธอด **main** จะแยกอยู่นอกคลาส **Visitation** ไปอยู่ในคลาสหลักของ โจทย์ ซึ่งจะรับข้อมูลการเข้าใช้สถานที่ สร้างวัตถุ **Visitation** และเรียกใช้เมธอดต่าง ๆ ที่เกี่ยวข้องตามแนวคิดที่กำหนดไว้ รวมถึงการเรียกใช้เมธอด **visit** เพื่อบันทึกการเข้าใช้สถานที่ด้วย³ เมื่อรับข้อมูลการเข้าใช้สถานที่ครบถ้วนแล้ว เมธอด **main** จะต้องเรียกเมธอด **printVisitHours** ของวัตถุ **Visitation** เพื่อพิมพ์สรุปการเข้าใช้สถานที่
จงเขียนโปรแกรมที่ทำตามข้อกำหนดที่ระบุไว้ข้างบน

รูปแบบข้อมูลเข้า

บรรทัดแรก	เป็นค่ารหัสประจำสถานที่เป็นจำนวนเต็มบวกมีค่าไม่เกิน 300
บรรทัดที่สอง	เป็นเลขวันที่เข้าใช้สถานที่เป็นจำนวนเต็มบวกมีค่าไม่เกิน 1,000
บรรทัดที่สาม	เป็นค่า $K \leq 50$ ซึ่งแทนจำนวนข้อมูลเกี่ยวกับการเข้าใช้สถานที่

² สำหรับการนับจำนวนชั่วโมงให้เรียกใช้เมธอด **count** หามันใน **printVisitHours**

ไม่เช่นนั้นจะถูกตัดคะแนนอย่างหนัก เนื่องจากเป็นประเด็นหลักในการทดสอบ

³ ถ้าไม่เขียนเมธอด **visit** ให้ถูกต้องหรือไม่เรียกใช้เมธอดนี้อย่างที่ควรเป็น คะแนนจะลดลงอย่างหนักเช่นกัน

บรรทัดที่ 4	เป็นเลขชั่วโมงที่เข้าใช้สถานที่ มีจำนวนเต็มหน่วยเป็นนาฬิกา ซึ่งอาจจะติดลบหรือมีค่าเกิน 23 ก็ได้ (กรณีเช่นนี้คือข้อมูลเวลาผิด ซึ่งโปรแกรมจะต้องทำการตรวจสอบในเมธอด visit อย่าไปตรวจสอบที่อื่น) จำนวนเลขชั่วโมงนี้มีทั้งหมด K ค่า
-------------	---

รูปแบบผลลัพธ์

มีอย่างน้อย 4 บรรทัดซึ่งมาจาก **printVisitHours** และมีบรรทัดอื่น ๆ มาก่อนหน้าหากมีข้อมูลเวลาที่ผิดพลาดปนอยู่ด้วย

ตัวอย่าง

ข้อมูลเข้า	ผลลัพธ์
5 7 8 10 5 9 24 23 -2 0 -1	invalid hour: 24 invalid hour: -2 invalid hour: -1 5 7 5 0 5 9 10 23
5 7 8 -1 -5 28 24 250 -2 27 -1	invalid hour: -1 invalid hour: -5 invalid hour: 28 invalid hour: 24 invalid hour: 250 invalid hour: -2 invalid hour: 27 invalid hour: -1 5 7 0 no visit

เกณฑ์การให้คะแนน

โปรแกรมจะต้องทำงานถูกอย่างน้อย 60% ของชุดทดสอบ จึงจะได้คะแนน
[มีโครงสร้างโค้ดให้ในหน้าถัดไป]

โครงสร้างโค้ดสำหรับการทำข้อสอบ

```
import java.util.Scanner;

class Visitation {
    // Enter your code here
}

public class Covid19Wins1 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int id = scan.nextInt();
        int date = scan.nextInt();
        final int K = scan.nextInt();

        // Create object and correctly use it here
    }
}
```