

## ปัญหา วางแผนทำคุกกี้ [CookiePlan] (5 คะแนน)

จากที่ทราบว่าคุณก็หนึ่งถุงนั้นจะใช้แป้ง 6 ถ้วยและเนย 3 ถ้วย เราทำการสำรวจปริมาณวัตถุดิบทั้งสองที่มีในแต่ละวันเพื่อวางแผนการผลิตและทำให้ทราบล่วงหน้าว่าวันใดที่วัตถุดิบจะขาดจากที่ควรเป็น โดยเราจะรับข้อมูลเกี่ยวกับปริมาณแป้งที่จะมีในแต่ละวันมาหนึ่งชุดและปริมาณเนยที่จะมีในแต่ละวันมาอีกชุด เราต้องการทราบว่าวันใดบ้างที่ปริมาณวัตถุดิบจะมีไม่เพียงพอต่อการผลิตให้ได้คุกกี้อย่างน้อย 10 ถุง (ถ้าหากมีแป้ง 60 ถ้วยและเนย 30 ถ้วยก็จะผลิตได้ 10 ถุงพอดี)

ในงานนี้ เราจะสร้างเมธอด `planCookie` แบบสถิต (`static method`) ที่เรียกจากเมธอด `main` ได้โดยตรง โดยห้ามสร้างวัตถุที่มีเมธอดนี้ก่อนการเรียกใช้งาน (มีโค้ดโครงสร้างให้ท้ายโจทย์) เมธอดนี้จะรับพารามิเตอร์มาหลายค่าจาก `main` เพื่อนำมาคำนวณค่าและแสดงผล ซึ่งมีข้อกำหนดเกี่ยวกับตัวเมธอดและการเรียกใช้งานดังต่อไปนี้<sup>1</sup>

1. เมธอดนี้ชื่อ `planCookie` คืนจำนวนเต็มกลับไปยังผู้เรียก จำนวนเต็มนี้แทนจำนวนวันที่วัตถุดิบไม่เพียงพอต่อการผลิตคุกกี้ถึง 10 ถุง (สมมติมีข้อมูล 10 วัน และมี 3 วันที่วัตถุดิบไม่พอ เมธอดนี้จะคืนค่า 3 ไปให้)

2. เมธอด `planCookie` รับพารามิเตอร์มาสองตัวดังนี้

- 2a. อาร์เรย์ของจำนวนแป้งที่มีในแต่ละวันเป็นจำนวนเต็ม (หน่วยเป็นถ้วย)

- 2b. อาร์เรย์ของจำนวนเนยที่มีในแต่ละวันเป็นจำนวนเต็ม (หน่วยเป็นถ้วย)

อาร์เรย์ทั้งสองนี้มีช่องแรก (ช่องหมายเลข 0) แทนข้อมูลของวัตถุดิบในวันแรกของการวางแผนการผลิต ส่วนช่องต่อ ๆ มาจะแทนข้อมูลวัตถุดิบของวันที่ 2, 3, 4, ... ไปเรื่อย ๆ เป็นคู่ข้อมูลจนหมดอาร์เรย์ และอาร์เรย์ทั้งสองนี้จะมีจำนวนช่องที่เท่ากัน

3. เมธอดนี้นำข้อมูลของแต่ละวันไปวิเคราะห์ และวันใดที่จำนวนวัตถุดิบมีไม่พอผลิต 10 ถุง เมธอดจะพิมพ์หมายเลขวันนั้นออกมาหนึ่งวันต่อบรรทัด เช่น ถ้าเป็นวันที่ 3, 4 และ 7 ที่วัตถุดิบไม่พอก็จะพิมพ์ว่า

---

<sup>1</sup> คำว่า ข้อกำหนด หมายถึงสิ่งที่ต้องปฏิบัติตาม ถ้าไม่ปฏิบัติตามจะโดนหักคะแนน ซึ่งในกรณีที่ทำผิดวัตถุประสงค์การทดสอบไปมาก ผู้เข้าสอบอาจจะได้ศูนย์คะแนนแม้โปรแกรมจะให้ผลลัพธ์ที่ถูกต้อง เช่น ไม่ได้สร้างเมธอดขึ้นมา หรือสร้างขึ้นมาแต่ไม่ได้เรียกใช้งาน เป็นต้น

3  
4  
7

4. เมื่อ `main` รับค่าที่คืนจาก `planCookie` แล้วจะนำผลลัพธ์มาพิมพ์ในรูปแบบประโยค  
There are P days that ingredients are insufficient เช่น  
There are 3 days that ingredients are insufficient หรือ  
There are 0 days that ingredients are insufficient  
(ต่อให้ไม่มีเลยสักวันหรือแค่วันเดียว `days` ก็เติม `s` และ ต่อให้มีเพียงวัตถุดิบเดียวที่ไม่  
เพียงพอ เราก็จะใช้รูปพหูพจน์ของ `ingredients` เช่นเดิม)

#### รูปแบบข้อมูลเข้า

บรรทัดแรก	เป็นจำนวนเต็มบวก $N \leq 10000$ แทนจำนวนวันที่จะรับข้อมูล ปริมาณวัตถุดิบของแต่ละวันมาพิจารณา
อีก N บรรทัด ถัดมา	แต่ละบรรทัดเป็นค่าในรูปแบบ <code>Flour Butter</code> ซึ่ง <code>Flour</code> และ <code>Butter</code> เป็นปริมาณแป้งและเนยในวันนั้น ๆ เป็นจำนวนเต็มที่ไม่ เป็นลบทั้งคู่และมีค่าไม่เกิน 100,000

#### รูปแบบผลลัพธ์

มีสองส่วน

ส่วนแรกมาจากเมธอด `planCookie` ซึ่งจะเป็นหมายเลขวันที่วัตถุดิบไม่พอ (หมายเลข  
วันเริ่มนับจากหนึ่ง) หนึ่งวันต่อบรรทัด ถ้าทุกวันมีวัตถุดิบครบ จะไม่มีผลลัพธ์จากส่วนนี้

ส่วนที่สองคือส่วนสรุปจำนวนวันในรูปแบบ

There are P days that ingredients are insufficient  
ซึ่งจะต้องพิมพ์โดยเมธอด `main` และค่า P นี้คือค่าที่คืนมาจากเมธอด `planCookie`  
(ห้าม `main` หาคำนี้อย่างตัวเอง)

## ตัวอย่าง

ข้อมูลเข้า	ผลลัพธ์
10 60 30 61 30 59 35 62 29 600 5000 778 120 1 2 100 200 300 500 63 39	3 4 7 There are 3 days that ingredients are insufficient  [หมายเหตุ ข้อความภาษาอังกฤษที่จริงเป็นบรรทัดเดียว ไม่ได้มีการขึ้นบรรทัดใหม่จากโปรแกรม ในที่นี้ขึ้นบรรทัดใหม่เพราะพื้นที่ในการแสดงข้อความมีจำกัด]
3 800 1000 555 432 685 586	There are 0 days that ingredients are insufficient
4 20 30 50 60 20 0 50 1000	1 2 3 4 There are 4 days that ingredients are insufficient

## เกณฑ์การให้คะแนน

โปรแกรมจะต้องทำงานถูกต้องอย่างน้อย 30% ของกรณีในชุดทดสอบ จึงจะได้คะแนน และข้อนี้เน้นที่ตัวโค้ดมากกว่าผลลัพธ์ จะมีการตรวจโค้ดจากผู้สอนอีกครั้ง โดยเฉพาะเรื่องการเขียนและเรียกใช้เมธอด นอกจากนี้ ห้ามสร้างฟิลด์ข้อมูลหรือเมธอดแบบสแตติกอื่น ๆ เพิ่มเติมอีก ยกเว้นตัวเมธอดที่ระบุให้สร้างในข้อนี้

[มีโครงสร้างโค้ดให้ทางด้านล่าง]

## โครงสร้างโค้ดคำตอบ

```
import java.util.Scanner;

public class CookiePlan {
    static ??? ???(???) {
        ???
    }

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        final int N = scan.nextInt();
        int[] flours = new int[N];
        int[] butters = new int[N];
        ???

        ??? = ??? (???);
        ???
    }
}
```