

## ปัญหา โรงงานคุกกี้ [CookieFactory] (10 คะแนน)

การคำนวณการผลิตคุกกี้ของโรงงานนี้มีวัตถุดิบสองตัวที่ต้องติดตามเป็นพิเศษคือแป้งและเนย เพราะปริมาณวัตถุดิบสองตัวนี้คือตัวกำหนดว่าผลิตคุกกี้ได้กี่ถุง ซึ่งแต่ละถุงจะใช้แป้ง  $F$  ถ้วยและเนย  $B$  ถ้วย ซึ่งเหตุที่ไม่ได้ระบุจำนวนไว้ตายตัวเพราะปริมาณที่ต้องใช้ขึ้นอยู่กับขนาดถุงและชนิดของคุกกี้ที่ต้องการผลิต เพื่อให้การติดตามวัตถุดิบและการผลิตเป็นไปอย่างมีระบบ เราจึงสร้างคลาส **CookiePlanner** ขึ้นมา โดยมีข้อกำหนดต่าง ๆ ดังนี้

1. ให้ **CookiePlanner** มีตัวสร้าง (constructor) ซึ่งจะรับพารามิเตอร์มาสองตัวตามลำดับดังนี้ (a) ปริมาณแป้งที่ต้องใช้ในการผลิตคุกกี้หนึ่งถุง (ค่า  $F$ ) และ (b) ปริมาณเนยที่ต้องใช้ในการผลิตคุกกี้หนึ่งถุง (ค่า  $B$ ) ทั้งสองค่าเป็นจำนวนเต็มบวก ซึ่งตัวสร้างจะนำค่าเหล่านี้ไปเก็บไว้ในตัวแปรสมาชิกคลาสเพื่อใช้ในระยะยาว
2. คลาส **CookiePlanner** มีตัวแปรสมาชิกคลาสอีกสามค่าคือ **flour** และ **butter** ซึ่งแทนปริมาณแป้งและเนยที่โรงงานมีอยู่ ณ ปัจจุบัน และอีกตัวคือ **bags** ซึ่งแทนจำนวนคุกกี้ที่ผลิตไปทั้งหมดจนถึงปัจจุบัน ทั้งสามค่านี้เริ่มต้นจะมีค่าเป็นศูนย์เพื่อแทนว่าเริ่มต้นโรงงานยังไม่มีแป้งและเนยเก็บไว้ รวมถึงยังไม่มีการผลิตคุกกี้ใด ๆ เกิดขึ้น
3. มีเมธอด **calculate** ทำหน้าที่คำนวณว่าจำนวนคุกกี้ที่จะผลิตได้มากที่สุดจากวัตถุดิบที่มีอยู่คือเท่าใด ซึ่งจำนวนคุกกี้ที่คำนวณได้จะถูกคืนกลับไปเป็นผลลัพธ์ เมธอดนี้ไม่รับพารามิเตอร์ใด ๆ
4. มีเมธอด **produce** ซึ่งรับพารามิเตอร์มาเป็นจำนวนคุกกี้ที่ต้องการผลิต ซึ่งหากวัตถุดิบเพียงพอที่จะผลิตได้ตามที่ต้องการ เมธอดจะทำการนับจำนวนคุกกี้ที่ผลิตได้เพิ่มเข้าไป พิมพ์จำนวนคุกกี้ที่ผลิตมาทั้งหมด รวมถึงหักปริมาณวัตถุดิบตามที่ใช้ในการผลิตนั้น และคืนค่า **true** กลับไป เช่นหากจำนวนคุกกี้ที่ต้องการผลิตคือ 10,  $F = 6$ ,  $B = 3$  และจำนวนแป้งและเนยปัจจุบันที่มีคือ 100 และ 80 ส่วนจำนวนคุกกี้ที่ผลิตไปก่อนหน้านี้คือ 25 การผลิตนี้จะใช้แป้ง 60 และเนย 30 ถ้วย ทำให้เหลือแป้ง 40 และเนย 50 ส่วนจำนวนคุกกี้ที่ผลิตจะกลายเป็น 35 และเมธอดจะพิมพ์เลข 35 ออกมา
5. กรณีที่วัตถุดิบมีไม่เพียงพอต่อการผลิตตามที่ต้องการ เมธอด **produce** จะคืนค่า **false** กลับไป และก่อนคืนค่ากลับ เมธอด **produce** จะเรียกเมธอด **calculate**

เพื่อให้ทราบจำนวนถุงสูงสุดที่ผลิตได้เป็นเท่าใด และพิมพ์ข้อความรายงานในรูปแบบ **Max capacity = C bags, but R bags requested** โดยค่า **C** คือจำนวนที่ได้มาจากเมธอด **calculate** และ **R** คือจำนวนถุงที่ต้องการผลิต (พารามิเตอร์ของ **produce**)

6. มีเมธอด **addFlour** ซึ่งรับพารามิเตอร์หนึ่งตัวเป็นจำนวนเต็ม แทนปริมาณแป้งที่จะเพิ่มเข้าไปในคลังวัตถุดิบ ถ้าหากค่าพารามิเตอร์เป็นศูนย์หรือติดลบ เมธอดจะพิมพ์ว่า **Invalid flour quantity** และคืนค่า **false** แต่ถ้าเป็นค่าบวก เมธอดนี้จะเพิ่มปริมาณแป้งที่โรงงานมีเข้าไปตามจำนวนที่ระบุมาในพารามิเตอร์ และพิมพ์ปริมาณแป้งที่มีอยู่ในคลังหลังการเพิ่ม หลังจากนั้นจะคืนค่า **true**
7. มีเมธอด **addButter** มีลักษณะเดียวกันกับ **addFlour** เพียงแต่จะจัดการเกี่ยวกับปริมาณเนยในคลังวัตถุดิบ ซึ่งหากเป็นเลขศูนย์หรือติดลบจะพิมพ์ข้อความว่า **Invalid butter quantity** ส่วนพฤติกรรมอื่นเป็นทำนองเดียวกันกับ **addFlour**
8. มีเมธอด **reportProduction** ซึ่งไม่รับพารามิเตอร์และไม่คืนค่าใด ๆ แต่จะพิมพ์รายงานจำนวนถุงคุกกี้ทั้งหมดที่ผลิตไปและวัตถุดิบที่เหลือในรูปแบบ

<b>Cookie bags:</b> จำนวนถุงคุกกี้ที่ผลิตไปทั้งหมด <b>Flour left:</b> จำนวนแป้งที่เหลือ <b>Butter left:</b> จำนวนเนยที่เหลือ
--

เช่น

<b>Cookie bags:</b> 70 <b>Flour left:</b> 23 <b>Butter left:</b> 90
---

สำหรับตัวเมธอด **main** จะรับข้อมูลเกี่ยวกับคุกกี้ การเติมวัตถุดิบ และการผลิต เมื่อจบกระบวนการทุกอย่างแล้วจะเรียกเมธอด **reportProduction** เพื่อรายงานสรุปสถานะการผลิต

### รูปแบบข้อมูลเข้า

บรรทัดแรก	เป็นค่า <b>F</b> และ <b>B</b> ซึ่งเป็นจำนวนเต็มบวก
-----------	--

บรรทัดที่สอง	เป็นจำนวนเต็มบวก <b>N</b> ซึ่งแทนจำนวนคำสั่งเพิ่มวัตถุดิบและการผลิตที่จะมีตามมา โดยที่ $N \leq 10,000$
อีก <b>N</b> บรรทัดถัดมา	เป็นคำสั่งที่ใช้ในการเพิ่มวัตถุดิบหรือผลิตคุกกี้ในรูปแบบ <b>K A</b> โดยที่ <b>K</b> เป็นเลข 1, 2, หรือ 3 ซึ่งแทนการเพิ่มแป้ง เพิ่มเนย หรือสั่งผลิตคุกกี้ตามลำดับ ส่วน <b>A</b> คือจำนวนที่เพิ่มหรือผลิตเป็นจำนวนเต็มบวกสำหรับคำสั่งผลิต แต่จะเป็นจำนวนเต็มที่สามารถเป็นบวก ลบ หรือศูนย์สำหรับคำสั่งเพิ่มวัตถุดิบ

### รูปแบบผลลัพธ์

ส่วนคำสั่งเพิ่มวัตถุดิบและผลิตคุกกี้	จากข้อมูลคำสั่งเพิ่มวัตถุดิบหรือสั่งผลิตคุกกี้ เมธอด <b>main</b> จะเรียกใช้ <b>addFlour</b> , <b>addButter</b> หรือ <b>produce</b> และพิมพ์ค่าที่คืนมาจากการเรียกเป็นผลลัพธ์ (คือจะพิมพ์ค่า <b>true</b> หรือ <b>false</b> ออกมาจาก <b>main</b> ) และหากตัวเลขพารามิเตอร์ผิดหรือสั่งผลิตมากกว่าที่วัตถุดิบจะทำได้ จะมีข้อความแจ้งออกมาจากเมธอดที่ถูกเรียกเป็นผลลัพธ์ก่อนพิมพ์ค่า <b>true/false</b> ดังกล่าวด้วย
ส่วนสรุป	เป็นผลจากการเรียกคำสั่ง <b>reportProduction</b> ซึ่งจะถูกเรียกใช้งานภายหลังจากสิ้นสุดคำสั่งเพิ่มวัตถุดิบและการผลิตทั้งหมด

## ตัวอย่าง

ข้อมูลเข้า	ผลลัพธ์
6 3 11 1 60 2 50 3 5 3 6 3 3 1 30 1 0 2 -5 2 3 3 7 3 1	60 true 50 true 5 true Max capacity = 5 bags, but 6 bags requested false 8 true 42 true Invalid flour quantity false Invalid butter quantity false 29 true 15 true Max capacity = 0 bags, but 1 bags requested false Cookie bags: 15 Flour left: 0 Butter left: 8
5 2 13 1 0 1 10 3 1 2 2 3 1 2 2 3 1 1 10 3 1 2 0 1 -5 2 10 1 15	Invalid flour quantity false 10 true Max capacity = 0 bags, but 1 bags requested false 2 true 1 true 2 true 2 true 10 true Max capacity = 0 bags, but 1 bags requested false Invalid butter quantity false Invalid flour quantity false 10 true 25 true Cookie bags: 2 Flour left: 25 Butter left: 10

## เกณฑ์การให้คะแนน

โปรแกรมจะต้องทำงานถูกต้องอย่างน้อย 20% ของชุดทดสอบ จึงจะได้คะแนน

[มีโครงสร้างโค้ดให้ในหน้าถัดไป]

โครงสร้างโค้ดสำหรับใช้ในการทำข้อสอบ แนะนำให้เติมเต็มคลาส CookiePlanner ตามข้อกำหนด ส่วนในเมธอด main ให้เน้นแก้โค้ดตรงที่เป็น ???

```
import java.util.Scanner;

class CookiePlanner {
    // Enter your code here
}

public class CookieFactory {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        final int F = scan.nextInt();
        final int B = scan.nextInt();
        ??? planner = ???;

        final int N = scan.nextInt();
        for(int i = 0; i < N; ++i) {
            int K = scan.nextInt();
            int A = scan.nextInt();
            ???
        }
        ???;
    }
}
```