

ปัญหา ข้อมูลโปรเซส [ProcessInfo] (5 คะแนน)

โปรเซส (Process) มักจะหมายถึงโปรแกรมอันหนึ่งที่ทำงานอยู่¹ ซึ่งแต่ละโปรแกรมก็จะมีปริมาณหน่วยความจำแรม (RAM) ที่ต้องใช้แตกต่างกันไป ในงานนี้เราต้องการแสดงรายการโปรเซสออกมาเป็นชื่อและปริมาณแรมที่โปรเซสนั้นใช้คู่กันไป พร้อมกับรายงานสรุปเกี่ยวกับการใช้งานแรมทั้งหมด

เพื่อบรรลุวัตถุประสงค์ดังกล่าว เราจะสร้างเมธอดที่รายงานการใช้ทรัพยากรของเครื่อง ซึ่งเป็นเมธอดแบบสถิต (static method) ที่เรียกจากเมธอด main ได้โดยตรง ไม่ต้องสร้างวัตถุเพิ่มเติมก่อนการเรียกใช้งาน (มีโค้ดโครงสร้างให้ท้ายโจทย์) เมธอดนี้จะรับพารามิเตอร์มาจาก main เพื่อนำมาจัดรูปแบบการแสดงผล และมีข้อกำหนดเกี่ยวกับตัวเมธอดและการเรียกใช้งานดังต่อไปนี้²

1. เมธอดนี้ชื่อ `listProcesses` ซึ่งคืน (return) บูลีนแสดงสถานะการใช้หน่วยความจำแรมกลับไปยังผู้เรียก นั่นคือหากหน่วยความจำที่ใช้รวมกันทุกโปรเซสมีค่ามากกว่าที่เครื่องมี เครื่องจะคืนค่า `false` แต่ถ้าหากรวมกันไม่มากกว่าที่เครื่องมี เครื่องจะคืนค่า `true` (อาจจะมองง่าย ๆ ได้ว่า `true` ใช้กับเหตุการณ์ที่หน่วยความจำมีเพียงพอ)

2. เมธอด `listProcesses` รับพารามิเตอร์ตามลำดับดังนี้ อาเรย์ของชื่อโปรเซส, อาเรย์ปริมาณแรมที่โปรเซสใช้, และ ปริมาณแรมที่เครื่องมี ซึ่งอาเรย์ของชื่อโปรเซสและปริมาณแรมจะสอดคล้องกันทางอินเด็กซ์ คือช่องแรกของอาเรย์ทั้งสองจะเป็นข้อมูลของโปรเซสแรก ช่องที่สองของอาเรย์ทั้งสองจะเป็นข้อมูลของโปรเซสที่สอง และจะเป็นคู่เช่นนี้ไปเรื่อย ๆ โดยชื่อโปรเซสเป็นสตริง ส่วนค่าปริมาณแรมเป็นจำนวนเต็ม

¹ ความหมายในทางวิชาการซับซ้อนกว่านี้มาก ตรงนี้เป็นมุมมองแบบง่าย ๆ

² คำว่า ข้อกำหนด หมายถึงสิ่งที่ต้องปฏิบัติตาม ถ้าไม่ปฏิบัติตามจะโดนหักคะแนน ซึ่งในกรณีที่ทำผิดวัตถุประสงค์การทดสอบไปมาก ผู้เข้าสอบอาจจะได้ศูนย์คะแนนแม้โปรแกรมจะให้ผลลัพธ์ที่ถูกต้อง เช่น ไม่ได้สร้างเมธอดขึ้นมา หรือสร้างขึ้นมาแต่ไม่ได้เรียกใช้งาน เป็นต้น

3. เมธอดนี้จะพิมพ์ข้อมูลชื่อโปรเซสกับปริมาณแรมที่โปรเซสนั้นใช้อย่างคร่าวๆ เช่น
Word 3

อย่างไรก็ตาม โปรเซสที่ใช้หน่วยความจำมากถึง **10** หน่วยจะมีเครื่องหมายดอกจัน * ตามมาด้วย เพื่อให้ผู้ใช้งานมองเห็นได้โดยง่ายว่าโปรเซสใดที่ใช้หน่วยความจำมากเป็นพิเศษ เช่น **Chrome* 12**

4. เมื่อแสดงรายการข้อมูลโปรเซสทั้งหมดเสร็จ เมธอดนี้จะรายงานสรุปปริมาณแรมในรูปแบบ

RAM used = U out of M เช่น

RAM used = 17 out of 32

5. เมื่อเมธอด **main** ได้รับค่าที่คืนมาจาก **listProcesses** เมธอด **main** จะนำผลมาพิมพ์ว่า

Sufficient RAM ถ้าหากว่าแรมที่มีอยู่ในเครื่อง มีปริมาณที่เพียงพอต่อการใช้งานของโปรเซสทั้งหมดรวมกัน (พิจารณาค่า **true/false** ที่ได้ออกมา ว่ามีความหมายอย่างไร)

แต่หากแรมไม่เพียงพอต่อการใช้งาน เมธอด **main** จะพิมพ์ว่า

Not enough RAM

รูปแบบข้อมูลเข้า

บรรทัดแรก	เป็นจำนวนเต็มบวก N <= 10000 แทนจำนวนโปรเซสที่จะรายงาน
อีก N บรรทัดถัดมา	แต่ละบรรทัดเป็นค่าในรูปแบบ Name Mem ซึ่ง Name เป็นชื่อโปรเซส ซึ่งเป็นสตริงที่ไม่มีช่องว่าง และ Mem คือปริมาณแรมที่โปรเซสนี้ใช้
บรรทัดสุดท้าย	เป็นจำนวนเต็มบวกแทนปริมาณแรมที่เครื่องมี

รูปแบบผลลัพธ์

มีสองส่วน

ส่วนแรกเป็นรายการโปรเซสทั้งหมดและหน่วยความจำที่แต่ละโปรเซสใช้ หนึ่งโปรเซสต่อบรรทัด เรียงตามลำดับการปรากฏในข้อมูลเข้า (ข้อ 3 ด้านบน) และมีสรุปยอดรวมหน่วยความจำที่ใช้ (ข้อ 4 ด้านบน) ส่วนนี้เป็นการพิมพ์ด้วยเมธอด `listProcesses`

ส่วนที่สองคือสรุปว่ามีหน่วยความจำเพียงพอหรือไม่ด้วยข้อความว่า

Sufficient RAM หรือ

Not enough RAM

ซึ่งจะต้องพิมพ์โดยเมธอด `main` และสิ่งที่นำมาสรุปนี้คือค่าบูลีนที่คืนมาจากเมธอด

`listProcesses` (ห้าม `main` หาคำนี้อยู่ด้วยตัวเอง)

ตัวอย่าง

ข้อมูลเข้า	ผลลัพธ์
8 Word 2 Moodle 3 Civ 6 Chrome 12 Firefox 10 Edge 9 Octave 1 PDF 2 32	Word 2 Moodle 3 Civ 6 Chrome* 12 Firefox* 10 Edge 9 Octave 1 PDF 2 RAM used = 45 out of 32 Not enough RAM
8 Word 12 Moodle 1 Civ 3 Chrome 3 Firefox 10 Edge 11 Octave 1 PDF 2 128	Word* 12 Moodle 1 Civ 3 Chrome 3 Firefox* 10 Edge* 11 Octave 1 PDF 2 RAM used = 43 out of 128 Sufficient RAM

เกณฑ์การให้คะแนน

โปรแกรมจะต้องทำงานถูกอย่างน้อย **20%** ของกรณีในชุดทดสอบ จึงจะได้คะแนน
[มีโครงสร้างโค้ดให้ในหน้าถัดไป]

โครงสร้างโค้ดคำตอบ

```
import java.util.Scanner;

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

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

        ??? = ??? (???);
        ???

    }
}
```