

### ปัญหา 3 ผลการเลือกตั้ง (Vote)

ในการเลือกตั้งครั้งหนึ่งมีผู้สมัครทั้งหมด  $N$  คน โดยผู้สมัครจะได้หมายเลข  $1, 2, 3, \dots, N$  โดยที่ไม่ซ้ำกันเลย ในการเลือกตั้งครั้งนี้มีผู้ออกเสียงทั้งหมด  $K$  คน โดยผู้ออกเสียงแต่ละคนจะเลือกออกเสียงโหวตผู้สมัครหมายเลขใดหมายเลขหนึ่งเพียงคนเดียวเท่านั้น กำหนดให้การเลือกตั้งครั้งนี้ไม่มีผู้ใดที่งดออกเสียง และไม่มีบัตรเสียเลย กล่าวคือผู้ออกเสียงเลือกหมายเลขใดหมายเลขหนึ่งจาก  $1$  ถึง  $N$  อย่างถูกต้องทุกคน จงเขียนโปรแกรมที่ทำการนับคะแนนโหวต จากนั้นให้โปรแกรมรายงานว่าผู้ชนะคือหมายเลขใดและได้รับคะแนนโหวตเท่าใด

#### ข้อมูลเข้า

- บรรทัดแรกระบุจำนวนเต็ม  $N$  และ  $K$  ตามลำดับ คั่นด้วยช่องว่าง โดยที่  $N$  มีค่าไม่เกิน  $100$  และ  $K$  ไม่น้อยกว่า  $1$
- บรรทัดที่สองประกอบด้วยตัวเลขจำนวนเต็ม  $K$  ตัว แต่ละตัวคือหมายเลขผู้สมัครที่ผู้ออกเสียงแต่ละคนเลือก เลขแต่ละตัวคั่นด้วยช่องว่างหนึ่งช่อง

#### ผลลัพธ์

- บรรทัดแรกระบุหมายเลขผู้สมัครที่ได้รับคะแนนโหวตมากที่สุด
- บรรทัดที่สองระบุคะแนนของผู้สมัครที่ได้รับคะแนนมากที่สุด

หมายเหตุ ข้อมูลเข้ารับประกันว่าผู้สมัครที่ได้รับคะแนนมากที่สุดจะมีเพียงคนเดียวเท่านั้น

## คำแนะนำ

(เป็นคำแนะนำสำหรับคนที่ยังเขียนโปรแกรมไม่คล่องนัก ใครที่เขียนคล่องแล้วลองดูตัวอย่างแล้วทำเลยจะดีกว่า)

1. ให้สร้างอาร์เรย์เก็บผลลัพธ์ขนาด 101 ช่องขึ้นมา สำหรับเก็บคะแนนโหวต (เราสร้างอาร์เรย์ขึ้นมาตามจำนวนช่องสูงสุดที่อาจจะได้ใช้ เพราะค่า N สูงสุดคือ 100)
2. ให้ระวังว่าหมายเลขผู้สมัครเริ่มจากเลข 1 ไม่ใช่เลขศูนย์นั่นเป็นเหตุผลว่าทำไมจำนวนช่องของอาร์เรย์มันถึงเกินออกมาหนึ่งช่อง (อาจจะมีบางช่องที่ไม่ได้ใช้เลยก็ได้ ไม่ต้องประหลาดใจ)
3. อาร์เรย์ที่สร้างขึ้นมานี้เก็บคะแนนโหวตที่ผู้สมัครแต่ละคนได้รับ ดังนั้นอย่าลืมว่าหลังจากสร้างอาร์เรย์ขึ้นมาแล้วเราต้องทำให้คะแนนโหวตของผู้สมัครทุกคนเป็นศูนย์ก่อนจากนั้นจึงค่อยเริ่มนับคะแนน
4. ในกรณีที่ผลลัพธ์จากโปรแกรมนักศึกษาผิด แนะนำให้ลองพิมพ์ค่าในอาร์เรย์จากช่องหมายเลข 1 ถึง N ออกมาดู
5. พอนับคะแนนโหวตเสร็จแล้วให้วนดูว่าช่องไหนในอาร์เรย์ที่มีค่ามากที่สุด ช่องนั้นแหละที่นำไปสู่คำตอบของเรา

## ตัวอย่าง

ข้อมูลเข้า	ผลลัพธ์
5 10 3 2 3 2 3 1 2 3 4 5	3 4
4 20 4 3 2 4 3 2 1 2 3 1 1 1 2 2 1 1 4 4 1 1	1 8
6 25 4 3 2 4 5 1 6 1 4 6 5 1 3 6 6 6 6 6 6 6 3 2 2 1	6 10

1. compare duplicate number
2. keep duplicate max