

1. เขียนโปรแกรมรับจำนวนเต็ม 1 จำนวน (n) แทนจำนวนพนักงานในบริษัทแห่งหนึ่งซึ่งทำงานสัปดาห์ละ 6 วัน แล้วรับข้อมูลจำนวนชั่วโมงการทำงานในแต่ละวันของพนักงานแต่ละคนตั้งแต่วันจันทร์ ถึง วันเสาร์ จากนั้นให้ทำการคำนวณหา
  - a. พนักงานแต่ละคนมีจำนวนชั่วโมงในการทำงานรวมต่อสัปดาห์เป็นเท่าไร แล้วแสดงผลลัพธ์ที่ได้ทางหน้าจอ
  - b. แต่ละวันมีพนักงานทุกคนทำงานรวมกันกี่ชั่วโมง แล้วแสดงผลลัพธ์ที่ได้ทางหน้าจอ

หมายเหตุ: ในนิสิตเขียนโปรแกรมโดยทำเป็น method ต่อไปนี้

- public static void inputArray(int [ ] [ ] hours, Scanner kb) { ... }
  - ทำหน้าที่รับข้อมูลเข้าเป็นจำนวนชั่วโมงมาใส่ใน array
  - main เป็นผู้เรียก method
- public static int sumRow(int [ ] [ ] hours, int row) { .... }
  - ทำหน้าที่หาจำนวนชั่วโมงรวมต่อสัปดาห์ของพนักงานคนที่ row (หาผลรวมตัวเลขทุกตัวในแถวที่ row ของ array ชื่อ hours), แล้ว return ผลรวม
  - main เป็นผู้เรียก method นี้, หลังจากที่ method return ให้ main นำค่าที่ return มาแสดงผลทางหน้าจอ เช่น Employee 1 : 29 hr.
- public static int sumCol(int [ ] [ ] hours, int col) { .... }
  - ทำหน้าที่หาจำนวนชั่วโมงรวมการทำงานของวันที่ col (หาผลรวมตัวเลขทุกตัวในหลักที่ col), แล้ว return ผลรวม
  - main เป็นผู้เรียก method นี้, หลังจากที่ method return ให้ main นำค่าที่ return มาแสดงผลทางหน้าจอ เช่น Day 1 : 12 hr.

ตัวอย่าง

Input	Output
3	Employee 1 : 29 hr.
2 3 4 5 7 8	Employee 2 : 24 hr.
7 3 4 3 3 4	Employee 3 : 18 hr.
3 3 4 3 3 2	Day 1 : 12 hr.
	Day 2 : 9 hr.
	Day 3 : 12 hr.
	Day 4 : 11 hr.
	Day 5 : 13 hr.
	Day 6 : 14 hr.

6	Employee 1 : 29 hr.
2 3 4 5 7 8	Employee 2 : 24 hr.
7 3 4 3 3 4	Employee 3 : 18 hr.
3 3 4 3 3 2	Employee 4 : 36 hr.
9 3 4 5 7 8	Employee 5 : 23 hr.
6 3 4 3 3 4	Employee 6 : 19 hr.
4 3 4 3 3 2	Day 1 : 31 hr.
	Day 2 : 18 hr.
	Day 3 : 24 hr.
	Day 4 : 22 hr.
	Day 5 : 26 hr.
	Day 6 : 28 hr.

2. จงเขียนโปรแกรมเพื่อรับขนาดของ matrix ได้แก่ จำนวน row และ column จากนั้นรับข้อมูลสมาชิกของ matrix ทั้งสอง (matrix A, matrix B) แล้วนำ matrix A และ matrix B มาบวกกัน แล้วแสดงผลลัพธ์ออกมาทางหน้าจอ

ตัวอย่าง

Input	Output
2 3	19 28 37
10 20 30	41 52 63
40 50 60	
9 8 7	
1 2 3	

หมายเหตุ: ในนิสิตเขียนโปรแกรมโดยทำเป็น method ต่อไปนี้

- public static void inputArray(int [ ] [ ] A, Scanner kb) { ... }
  - ทำหน้าที่รับข้อมูลมาใส่ใน array
  - main เป็นผู้เรียก method 2 ครั้งเพื่อรับข้อมูลเข้ามาใน matrix A และ matrix B
- public static int [ ] [ ] addMatrix(int [ ] [ ] A, int [ ] [ ] B) { .... }
  - ทำหน้าที่หาผลรวมของ Matrix A และ B โดยเก็บผลรวมไว้ใน array ใหม่, แล้ว return array ผลรวม ออกมา
  - main เป็นผู้เรียก method นี้, หลังจากนี้ method return ให้ main นำค่าที่ return มาแสดงผลทางหน้าจอ โดยใช้ method printArray ที่นิสิตเคยทำในห้อง Lab

3. จงเขียนโปรแกรมเพื่อรับเลขจำนวนเต็ม 2 จำนวน (m และ n) เพื่อกำหนดขนาด ของ Matrix จตุรัส (m x m) จากนั้นรับค่าสมาชิกแต่ละตัวจากแป้นพิมพ์จนครบ พอรับค่าเสร็จ ให้ทำการหมุนค่าใน Matrix จำนวน n ครั้ง ะละ 90 องศา (ตามเข็มนาฬิกา) และให้แสดงค่าใน matrix จตุรัสหลังจากเสร็จสิ้นการหมุน (ดังตัวอย่าง)

หมายเหตุ: ในนิสิตเขียนโปรแกรมโดยทำเป็น method ต่อไปนี้

- public static void inputArray(int [ ] [ ] A, Scanner kb) { ... }
  - ทำหน้าที่รับข้อมูลมาใส่ใน array
  - main เป็นผู้เรียก method เพื่อรับข้อมูลเข้ามาใน array
- public static int [ ] [ ] rotateMatrix(int [ ] [ ] A) { .... }
  - ทำหน้าที่หมุนค่าใน Matrix A ไป 90 องศา (ตามเข็มนาฬิกา), แล้ว return array ใหม่ที่เป็นผลลัพธ์ออกมา
  - main เป็นผู้เรียก method นี้ เป็นจำนวน n ครั้งเพื่อที่จะหมุนค่าใน array n ครั้ง

ตัวอย่าง

Input	Output
3 0 11 22 33 44 55 66 77 88 99	11 22 33 44 55 66 77 88 99
3 1 11 22 33 44 55 66 77 88 99	77 44 11 88 55 22 99 66 33
3 2 11 22 33 44 55 66 77 88 99	99 88 77 66 55 44 33 22 11

4. จงเขียนโปรแกรมเพื่อช่วยหาจุดที่ใกล้จุด i มากที่สุด
- รับจำนวนเต็ม (n) มาจาก keyboard ซึ่งแทนจำนวนคู่จุด (x, y) ที่มีอยู่ทั้งหมด
  - จากนั้นให้วน loop n รอบ เพื่อรับค่าพิกัด x และ y ของแต่ละจุด (ค่าของพิกัดเป็นได้ทั้งจำนวนเต็มและทศนิยม) แล้วนำมาเก็บไว้ใน array สองมิติ
  - รับจำนวนเต็ม (i) จากแป้นพิมพ์ ให้เป็นเลขของจุดที่สนใจ (i == 0 หมายถึง จุดแรกคือจุดที่เราสนใจ)
  - คำนวณหาจุดที่ใกล้กับจุด i มากที่สุด แล้วพิมพ์พิกัด x y ของจุดนั้น ออกทางหน้าจอ (พร้อมระยะห่างจากจุด i เป็นทศนิยมสองตำแหน่ง)

หมายเหตุ: ให้นักเขียนโปรแกรมโดยทำเป็น method ต่อไปนี้

- public static void inputArray(int [ ] [ ] points, Scanner kb) { ... }
  - ทำหน้าที่รับข้อมูลมาใส่ใน array
  - main เป็นผู้เรียก method เพื่อรับข้อมูลเข้ามาใน array
- public static double findDistance(double [ ] [ ] points, int point1, int point2) { .... }
  - ทำหน้าที่หาระยะห่างระหว่างจุดแรก (point1) และจุดที่สอง (point2) ที่เก็บไว้ใน array ชื่อ points โดยใช้สมการด้านล่าง พร้อม return ระยะห่างออกมา
    - ระยะห่างระหว่างจุด 2 จุด ที่มีพิกัด x y ดังนี้ (x1, y1) และ (x2, y2) คำนวณได้จาก
 
$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$
  - main เป็นผู้เรียก method นี้เพื่อหาระยะห่างระหว่าง จุด i และ จุดอื่น ๆ ทุก ๆ จุดที่เหลือ

ตัวอย่าง

Input	Output
8	1 1
-1 3	2.83
-1 -1	
1 1	
2 0.5	
2 -1	
3 3	
4 2	
4 -0.5	
1	

5. จงเขียนโปรแกรมเพื่อช่วยหาจุดที่ใกล้จุดที่ใกล้กันมากที่สุดสองจุด

- รับจำนวนเต็ม (n) มาจาก keyboard ซึ่งแทนจำนวนคู่จุด (x, y) ที่มีอยู่ทั้งหมด
- จากนั้นให้วน loop n รอบ เพื่อรับค่าพิกัด x และ y ของแต่ละจุด (ค่าของพิกัดเป็นได้ทั้งจำนวนเต็มและทศนิยม) แล้วนำมาเก็บไว้ใน array สองมิติ
- คำนวณหาจุดที่ใกล้กันมากที่สุดสองจุด แล้วแสดงหมายเลขจุดของทั้งสองจุดนั้นออกทางหน้าจอ (พร้อมระยะห่างของทั้งสองจุด) (กำหนด หมายเลขของจุดแรก คือ 0)

หมายเหตุ: ให้นักเขียนโปรแกรมโดยทำเป็น method ต่อไปนี้

- การแสดงคู่จุดที่ใกล้กันที่สุดสองจุด ให้แสดงหมายเลขจุดที่น้อยกว่าก่อนเสมอ เช่น ถ้าโจทย์กำหนดให้มี 10 จุด (n เท่ากับ 10) ซึ่งจุดแรกมีหมายเลข 0 และจุดสุดท้ายมีหมายเลข 9 สมมติว่า จุดหมายเลข 3 กับ จุดหมายเลข 5 ใกล้กันที่สุด โปรแกรมต้องแสดง 3 5 ในบรรทัดแรก ไม่ใช่ 5 3

- public static void inputArray(int [ ] [ ] A, Scanner kb) { ... }
  - ทำหน้าที่รับข้อมูลมาใส่ใน array
  - main เป็นผู้เรียก method เพื่อรับข้อมูลเข้ามาใน array
- public static double findDistance(double [ ] [ ] points, int point1, int point2) { .... }
  - ทำหน้าที่หาระยะห่างระหว่างจุดแรก (point1) และจุดที่สอง (point2) ที่เก็บไว้ใน array ชื่อ points โดยใช้สมการด้านล่าง พร้อม return ระยะห่างออกมา
    - ระยะห่างระหว่างจุด 2 จุด ที่มีพิกัด x y ดังนี้ (x1, y1) และ (x2, y2) คำนวณได้จาก
 
$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$
  - findDistanceAllPairs จะเรียก method นี้เพื่อให้ช่วยหาระยะทางระหว่างคู่จุด
- public static double [ ] [ ] findDistanceAllPairs(double [ ] [ ] points) { .... }
  - ทำหน้าที่หาระยะห่างระหว่างทุก ๆ คู่จุด โดยเก็บระยะทางไว้ใน array 2 มิติที่สร้างขึ้นใหม่ ชื่อ distances โดย distances[ i ] [ j ] จะเก็บระยะทางระหว่างจุดที่ i และจุดที่ j พอได้ระยะห่างครบทุกคู่จุดแล้ว method จะ return array distances ออกมา
  - main เป็นผู้เรียก method นี้ หลังจากได้ distances ทุกคู่จุดแล้ว main จะ scan ให้ระยะทางที่น้อยที่สุดใน array distance เพื่อหาคู่จุดที่ห่างกันน้อยที่สุด

#### ตัวอย่าง

Input	Output
5	0 1
-1 3	1
-1 4	
1 10	[คำอธิบาย: จุดแรก (หมายเลข 0) และ จุดที่สอง (หมายเลข 1) เป็นคู่จุดที่ใกล้กันที่สุด
2 11	โดยจุดแรกอยู่ที่พิกัด (-1, 3) ส่วนจุดที่สองอยู่ที่พิกัด (-1, 4) ห่างกันแค่ 1 หน่วย]
3 12	