



Task Lab02 ข้อ 1 (Lab02_1)

แบบฝึกปฏิบัติการครั้งที่ 2 การเขียนโปรแกรมเพื่อใช้งานเมทอด

จุดประสงค์

เมื่อผ่านปฏิบัติการนี้แล้ว นักศึกษาจะสามารถเขียนโปรแกรมเพื่อนิยามและเรียกใช้เมทอดได้

การส่งงาน

บน Grader ของวิชา โดย login ด้วย user และ password ที่แจกให้

- เลือก Contest และ เลือก Task ที่ต้องการส่งงาน
- Upload ไฟล์ .java ที่มีชื่อเดียวกันกับชื่อ Task เช่น Lab02_1.java
- ให้เขียน comment เป็นรหัสนักศึกษาและชื่อไว้ด้านบนไฟล์

คำสั่ง

จงเขียนโปรแกรมเพื่อรับตัวอักษร 2 ตัว และตัวเลข 1 ตัว ใน main() แล้วส่งไปประมวลผลที่เมทอดชื่อ printStartUp() หรือ printStarDown() หรือ printStarBoth() ที่ผลลัพธ์ของการทำงานขึ้นอยู่กับตัวอักษรและตัวเลขที่รับเข้ามา

Input มี 1 บรรทัด

ประกอบด้วยตัวอักษร 2 ตัว และตัวเลข 1 ตัว แต่ละตัวจะคั่นด้วยช่องว่าง 1 ช่อง มีรายละเอียดดังนี้

- อักษรตัวแรกอาจเป็นตัวอักษรพิมพ์เล็ก ตัวอักษรพิมพ์ใหญ่ หรือตัวอักษรอื่นๆ
- อักษรตัวที่สองจะต้องเป็น # หรือ * เท่านั้น
- ตัวเลข 1 ตัว ($1 \leq N \leq 100$)

Output มี N บรรทัด หรือ 2N-1 บรรทัด ขึ้นอยู่กับ Input

รูปสามเหลี่ยมแสดงด้วยเครื่องหมาย # หรือ * ซึ่งเกิดจากการเรียกใช้เมทอดชื่อ printStartUp() หรือ printStartDown() หรือ printStarBoth() ทั้งสามเมทอดนี้จะมีการทำงานและแสดงผลภายในเมทอด โดยจะไม่มี การส่งค่ากลับมาที่เมทอด main()

รายละเอียดการประมวลผลของโปรแกรมมีดังนี้

- แบบที่ 1 กรณีอักษรตัวแรกเป็นตัวอักษรพิมพ์เล็กในช่วง 'a' ถึง 'z' ให้เรียกใช้เมทอดชื่อ printStartDown() ซึ่งจะพิมพ์รูปสามเหลี่ยมด้วยเครื่องหมาย # หรือ * และมีจำนวนแถว N แถว ดังตัวอย่าง

หากรับ a # 3 จะพิมพ์ 3 แถว ดังนี้

```
###
##
#
```

หากรับ z * 4 จะพิมพ์ 4 แถว ดังนี้

```
****
***
**
*
```

- แบบที่ 2 กรณีอักษรตัวแรกเป็นตัวพิมพ์ใหญ่ในช่วง 'A' ถึง 'Z' ให้เรียกใช้เมทอดชื่อ printStartUp() ซึ่งจะพิมพ์รูปสามเหลี่ยมด้วยเครื่องหมาย # หรือ * และมีจำนวนแถว N แถว ดังตัวอย่าง

หากรับ A # 4 จะพิมพ์ 4 แถว ดังนี้

```
#
##
###
####
```

หากรับ Z * 3 จะพิมพ์ 3 แถว ดังนี้

```
*
**
***
```

- แบบที่ 3 กรณีอักษรตัวแรกเป็นตัวอักษรอื่นๆ ให้เรียกใช้เมทอดชื่อ printStarBoth() ซึ่งจะพิมพ์รูปสามเหลี่ยมด้วยเครื่องหมาย # หรือ * และมีจำนวนแถว 2N-1 แถว ดังตัวอย่าง

หากรับ \$ * 3 จะพิมพ์ 5 แถว ดังนี้

```
*
**
***
**
*
```

หากรับ @ # 4 จะพิมพ์ จะพิมพ์ 7 แถว ดังนี้

```
#
##
###
####
####
###
##
#
```

ตัวอย่าง Input และ Output

ตัวอย่างที่	Input	Output	คำอธิบาย
1	a # 5	##### ##### #### ## #	พิมพ์แบบที่ 1 เนื่องจากอักขระตัวแรกเป็นตัวอักษรพิมพ์เล็ก
2	B # 5	# ## ### #### #####	พิมพ์แบบที่ 2 เนื่องจากอักขระตัวแรกเป็นตัวอักษรพิมพ์ใหญ่
3	@ * 5	* ** *** **** ***** ***** **** *** ** *	พิมพ์แบบที่ 3 เนื่องจากอักขระตัวแรกเป็นตัวอักขระอื่นๆ



Task Lab02 ข้อ 2 (Lab02_2)

แบบฝึกปฏิบัติการครั้งที่ 2
การเขียนโปรแกรมเพื่อใช้งานเมทอด

จุดประสงค์

เมื่อผ่านปฏิบัติการนี้แล้ว นักศึกษาจะสามารถเขียนโปรแกรมเพื่อนิยามและเรียกใช้เมทอดได้

การส่งงาน

บน Grader ของวิชา โดย login ด้วย user และ password ที่แจกให้

- เลือก Contest และ เลือก Task ที่ต้องการส่งงาน
- Upload ไฟล์ .java ที่มีชื่อเดียวกันกับชื่อ Task เช่น Lab02_2.java
- ให้เขียน comment เป็นรหัสนักศึกษาและชื่อไว้ด้านบนไฟล์

คำสั่ง

จงเขียนโปรแกรมเพื่อรับเลขจำนวนเต็มบวก 1 จำนวนใน main() แล้วส่งไปประมวลผลที่เมทอดชื่อ checkOddEven() เพื่อทำการตรวจสอบว่าเป็นเลขคี่หรือเลขคู่ และ checkPrime() เพื่อให้เมทอดทำการตรวจสอบว่าเลขที่รับเข้ามาเป็นจำนวนเฉพาะหรือไม่ (จำนวนเฉพาะคือจำนวนเต็มบวกที่มากกว่า 1 ซึ่งนอกจาก 1 และตัวมันเองแล้ว จะไม่มีเลขจำนวนใดหารลงตัว เช่นเลข 2 3 5 7 11 677 10301 เป็นต้น) โดยแสดงผลดังตัวอย่าง

Input มี 1 บรรทัด

เป็นเลขจำนวนเต็ม N 1 จำนวนโดยที่ $1 \leq N \leq 100000000$

Output มี 2 บรรทัด

บรรทัดแรกคือคำว่า odd หรือ even

บรรทัดที่สอง คือ คำว่า prime หรือ not prime

ตัวอย่าง Input และ Output

ตัวอย่างที่	Input	Output
1	677	odd prime
2	24	even not prime



Task Lab02 ข้อ 3 (Lab02_3)

แบบฝึกปฏิบัติการครั้งที่ 2 การเขียนโปรแกรมเพื่อใช้งานเมทอด

จุดประสงค์

เมื่อผ่านปฏิบัติการนี้แล้ว นักศึกษาจะสามารถเขียนโปรแกรมเพื่อนิยามและเรียกใช้เมทอดได้

การส่งงาน

บน Grader ของวิชา โดย login ด้วย user และ password ที่แจกให้

- เลือก Contest และ เลือก Task ที่ต้องการส่งงาน
- Upload ไฟล์ .java ที่มีชื่อเดียวกันกับชื่อ Task เช่น Lab02_3.java
- ให้เขียน comment เป็นรหัสนักศึกษาและชื่อไว้ด้านบนไฟล์

คำสั่ง

จงเขียนโปรแกรมเพื่อรับค่ารับค่า X_1, Y_1, X_2, Y_2 (ทุกค่าเป็นเลขจำนวนเต็ม 4 ค่า อยู่ในช่วง 1 ถึง 10000) แทนค่าจุด A (ค่า X_1 และ Y_1) และ B (ค่า X_2 และ Y_2) โดยที่ A และ B ไม่ใช่จุดเดียวกัน (ไม่ต้องตรวจสอบ) จากนั้นรับจำนวนเต็มบวก N (อยู่ในช่วง 1 ถึง 10000000) เพื่อรับค่า X_3, Y_3 (ทุกค่าเป็นเลขจำนวนเต็ม 2 ค่า อยู่ในช่วง 1 ถึง 10000) ของจุด C ใน main () จำนวน N รอบ

ในแต่ละรอบให้ส่งไปประมวลผลที่เมทอดชื่อ checkBoundary() เพื่อเช็คสถานะของจุด C จนครบ N รอบ โดยเมทอด checkBoundary() จะทำการตรวจสอบสถานะของ จุด C ว่าอยู่ภายในหรือภายนอกหรือบนเส้นขอบของกรอบสี่เหลี่ยมที่สร้างจากจุด A และ B จากนั้นจะส่งค่ากลับเป็นหมายเลขสถานะของจุด C ดังนี้

- ส่งค่ากลับเป็น 1 แสดงว่า จุด C อยู่ภายในกรอบสี่เหลี่ยม
- ส่งค่ากลับเป็น 2 แสดงว่าจุด C บนเส้นขอบของกรอบสี่เหลี่ยม
- ส่งค่ากลับเป็น 3 แสดงว่า จุด C อยู่ภายนอกกรอบสี่เหลี่ยม

แล้วให้โปรแกรมสรุปจำนวนผลลัพธ์ จำนวนจุด ในแต่ละสถานะ

Input มี N + 2 บรรทัด (ในแต่ละบรรทัด เป็นเลขจำนวนเต็ม แต่ละค่าคั่นด้วยช่องว่าง 1 ช่อง)

บรรทัดแรก รับค่าเลขจำนวนเต็ม 4 ค่า X_1, Y_1, X_2, Y_2 ของจุด A และ B โดย $-10000 \leq X_1, Y_1, X_2, Y_2 \leq 10000$

บรรทัดที่สอง รับค่าเลขจำนวนเต็ม N แทนจำนวนของจุด C ที่จะตรวจสอบสถานะ โดยที่ $1 \leq N \leq 10000000$

อีก N บรรทัด แต่ละบรรทัดรับค่าเลขจำนวนเต็ม 2 ค่า X_3, Y_3 ของจุด C โดย $-10000 \leq X_3, Y_3 \leq 10000$

Output มี 1 บรรทัด

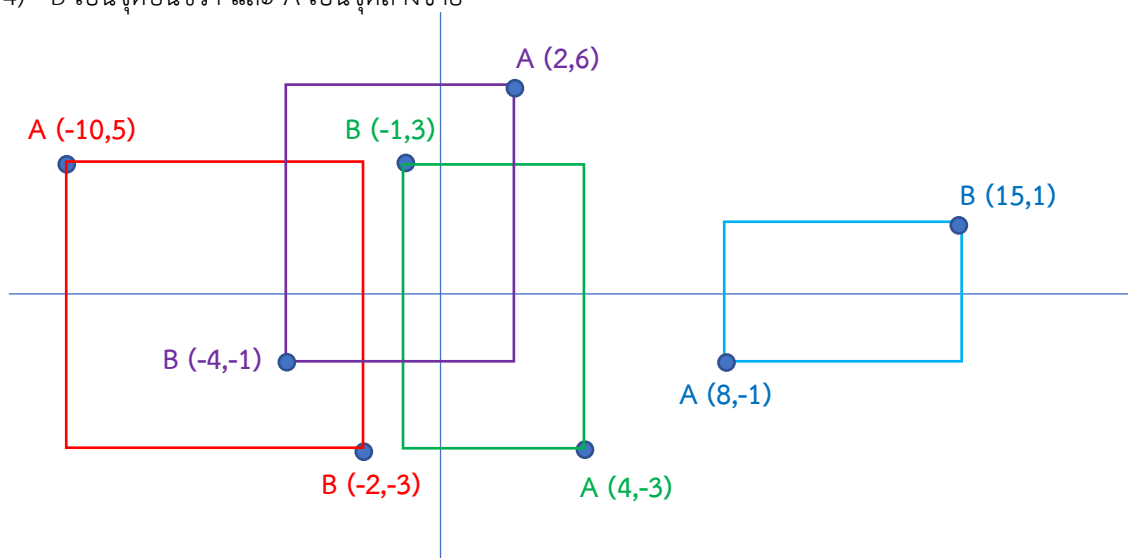
เป็นเลขจำนวนเต็ม 3 ค่า แทนจำนวนจุด C ที่อยู่ในสถานะภายในกรอบสี่เหลี่ยม บนเส้นขอบของกรอบสี่เหลี่ยม และอยู่ภายนอกกรอบสี่เหลี่ยม ตามลำดับ

ตัวอย่าง Input และ Output

ตัวอย่างที่	Input	Output	คำอธิบาย
1	-10 5 -2 -3 5 -5 5 //อยู่บนกรอบสี่เหลี่ยม -6 6 //อยู่นอกกรอบสี่เหลี่ยม -3 -2 //อยู่ในกรอบสี่เหลี่ยม -5 0 //อยู่ในกรอบสี่เหลี่ยม 0 3 //อยู่นอกกรอบสี่เหลี่ยม	2 1 2	ข้อมูลนำเข้า - รับจุด A และ B แทนกรอบสี่เหลี่ยม โดย A เป็นจุดบนซ้าย และ B เป็นจุดล่างขวา - รับค่า N = 5 แทนจำนวนจุด C - รับจุด C จำนวน 5 จุด ข้อมูลส่งออก - มีจำนวนจุด C ที่อยู่ อยู่ภายในกรอบสี่เหลี่ยม 2 จุด - มีจำนวนจุด C ที่อยู่ อยู่บนขอบของกรอบสี่เหลี่ยม 1 จุด - มีจำนวนจุด C ที่อยู่ อยู่ภายนอกกรอบสี่เหลี่ยม 2 จุด
2	4 -3 -1 3 10 2 3 //อยู่บนกรอบสี่เหลี่ยม -2 3 //อยู่นอกกรอบสี่เหลี่ยม -2 -3 //อยู่นอกกรอบสี่เหลี่ยม 2 -3 //อยู่บนกรอบสี่เหลี่ยม 0 3 //อยู่บนกรอบสี่เหลี่ยม 3 0 //อยู่ในกรอบสี่เหลี่ยม 0 -3 //อยู่บนกรอบสี่เหลี่ยม -3 0 //อยู่นอกกรอบสี่เหลี่ยม 0 0 //อยู่ในกรอบสี่เหลี่ยม 5 1 //อยู่นอกกรอบสี่เหลี่ยม	2 4 4	ข้อมูลนำเข้า - รับจุด A และ B แทนกรอบสี่เหลี่ยม โดย B เป็นจุดบนซ้าย และ A เป็นจุดล่างขวา - รับค่า N = 10 แทนจำนวนจุด C - รับจุด C จำนวน 10 จุด ข้อมูลส่งออก - มีจำนวนจุด C ที่อยู่ อยู่ภายในกรอบสี่เหลี่ยม 2 จุด - มีจำนวนจุด C ที่อยู่ อยู่บนขอบของกรอบสี่เหลี่ยม 4 จุด - มีจำนวนจุด C ที่อยู่ อยู่ภายนอกกรอบสี่เหลี่ยม 4 จุด

หมายเหตุ : A และ B อาจเป็นจุดในแนวทแยงของกรอบสี่เหลี่ยม ซึ่งเป็นไปได้ 4 แบบ

- 1) A เป็นจุดบนซ้าย และ B เป็นจุดล่างขวา
- 2) B เป็นจุดบนซ้าย และ A เป็นจุดล่างขวา
- 3) A เป็นจุดบนขวา และ B เป็นจุดล่างซ้าย
- 4) B เป็นจุดบนขวา และ A เป็นจุดล่างซ้าย





Task Lab02 ข้อ 4 (Lab02_4)

แบบฝึกปฏิบัติการครั้งที่ 2 การเขียนโปรแกรมเพื่อใช้งานเมท็อด

จุดประสงค์

เมื่อผ่านปฏิบัติการนี้แล้ว นักศึกษาจะสามารถเขียนโปรแกรมเพื่อนิยามและเรียกใช้เมท็อดได้

การส่งงาน

บน Grader ของวิชา โดย login ด้วย user และ password ที่แจกให้

- เลือก Contest และ เลือก Task ที่ต้องการส่งงาน
- Upload ไฟล์ .java ที่มีชื่อเดียวกันกับชื่อ Task เช่น Lab02_4.java
- ให้เขียน comment เป็นรหัสนักศึกษาและชื่อไว้ด้านบนไฟล์

คำสั่ง

จงเขียนโปรแกรม เพื่อ

- 1) รับข้อมูลเลขจำนวนเต็มซึ่งไม่ซ้ำกันมาเก็บไว้ในอาร์เรย์ A ทั้งหมด N จำนวน (ไม่ต้องเขียนส่วนการตรวจสอบว่าข้อมูลซ้ำหรือไม่ ให้สมมติว่า ผู้ใช้กรอกข้อมูลไม่ซ้ำกัน) ซึ่งการรับข้อมูลใส่อาร์เรย์ โดยเรียกใช้เมท็อดชื่อ `getData()` ซึ่งทำหน้าที่รับข้อมูลจากผู้ใช้ในอาร์เรย์ตามขนาดที่รับมาจากเมท็อด `main()` จากนั้นจึงส่งค่าอาร์เรย์กลับไปยังเมท็อด `main()`
- 2) เขียนเมท็อดใหม่ชื่อ `findAinB()` ทำหน้าที่
 - เรียกใช้ `getData()` อีกครั้ง แต่เพื่อรับข้อมูลจำนวนเต็มใส่ในอาร์เรย์ B ทั้งหมด M จำนวน (อาจมีค่าซ้ำกันได้)
 - ให้นับและแสดงว่ามีสมาชิกในอาร์เรย์ A ปรากฏอยู่ในอาร์เรย์ B อย่างละกี่ตัว ดังตัวอย่าง
- 3) หาค่าสูงสุดในอาร์เรย์ A คือ ค่าใด

เช่น กำหนดให้ ตัวอย่างเช่น กรณี $N = 3$ และ $M = 12$ และสมมติผู้ใช้กรอกข้อมูลใน อาร์เรย์ A และ B ดังนี้

อาร์เรย์ A

5	7	2
---	---	---

อาร์เรย์ B

1	7	7	7	2	8	7	2	9	20	15	7
---	---	---	---	---	---	---	---	---	----	----	---

ผลลัพธ์ที่ได้

0 5 2

7

เนื่องจากพบค่า 5 7 และ 2 ใน อาร์เรย์ B จำนวน 0 5 และ 2 ค่าตามลำดับ

Input มี 3 บรรทัด

บรรทัดแรก เป็นเลขจำนวนเต็มบวก M และ N แต่ละค่าเว้นด้วยช่องว่าง 1 ช่อง โดยที่ $1 \leq M, N \leq 10000$

บรรทัดที่ 2- 3 เป็นเลขจำนวนเต็ม A_i, B_j จำนวน M และ N ค่าสำหรับอาร์เรย์ A และอาร์เรย์ B

โดยแต่ละค่าเรียงลำดับจากน้อยไปมาก และแต่ละค่าเว้นด้วยช่องว่าง 1 ช่อง

ซึ่งที่ $-10000000 \leq A_i, B_j \leq 10000000$ โดยที่ $i = 1.. M$ และ $j = 1.. N$

Output มี N+1 บรรทัด

N บรรทัดแรก เป็นผลการนับจำนวนตัวเลขในอาร์เรย์ A ที่พบในอาร์เรย์ B

บรรทัดสุดท้าย คือ ค่าสูงสุดในอาร์เรย์ A

ตัวอย่าง Input และ Output

ตัวอย่างที่	Input	Output	คำอธิบาย
1	3 12 5 7 2 1 7 7 7 2 8 7 2 9 20 15 7	0 5 2 7	Finding 5 in array B พบ 0 จำนวน Finding 7 in array B พบ 5 จำนวน Finding 2 in array B พบ 2 จำนวน Maximum in A is 7
2	4 2 12 7 20 2 33 7	0 1 0 0 20	Finding 12 in array B พบ 0 จำนวน Finding 7 in array B พบ 1 จำนวน Finding 20 in array B พบ 0 จำนวน Finding 2 in array B พบ 0 จำนวน Maximum in A is 20 จำนวน