

ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร พระราชวังสนามจันทร์

ข้อสอบแข่งขันคอมพิวเตอร์โอลิมปิกระดับชาติ ครั้งที่ 8

ข้อสอบมี 3 ข้อ 14 หน้า ให้ทำทุกข้อ เวลา 9.00 – 12.00 น. วันพฤหัสบดี ที่ 17 พฤษภาคม 2555

หาทำเลตั้งศูนย์บริการลูกค้า (location)

ร้านคอมพิวเตอร์ K.I.B. ต้องการขยายฐานลูกค้าไปยังเมืองใหม่ โดยเมืองดังกล่าวมีการวางผังเมืองเป็นพื้นที่สี่เหลี่ยมย่อยจำนวน $M \times N$ พื้นที่ (M แถว N หลัก) และจากการสำรวจสำมะโนประชากรทำให้ทราบจำนวนประชากรในแต่ละพื้นที่ (ดูภาพประกอบด้านล่าง)

เนื่องจากร้าน K.I.B. ต้องการเปิดศูนย์บริการลูกค้าเพียงร้านเดียวในเมืองนี้ ยิ่งไปกว่านั้นพื้นที่บริการที่ร้านให้บริการลูกค้าได้จะครอบคลุมบริเวณที่ประกอบด้วยสี่เหลี่ยมย่อยจำนวน $K \times K$ พื้นที่ (K แถว K หลัก) เท่านั้น ทางร้านจึงพยายามหาพื้นที่บริการที่ดีที่สุด ซึ่งในที่นี้หมายถึงพื้นที่บริการที่มีประชากรรวมกันมากที่สุด

5	9	2	9	1	2	8	9	1	6
9	1	3	9	8	4	2	1	5	7
2	7	9	3	8	5	2	7	6	8
1	6	2	1	7	7	1	9	4	1
8	5	2	3	9	8	5	6	3	3

ภาพประกอบตัวอย่างโจทย์ แสดงผลการหาทำเลตั้งศูนย์บริการลูกค้าในพื้นที่ขนาด 2×2 ($K = 2$) ของผังเมืองขนาด 5×10 ในที่นี้บริเวณที่ถูกเน้นคือพื้นที่บริการที่ดีที่สุด

จงเขียนโปรแกรมที่มีประสิทธิภาพในการหาจำนวนประชากรรวมในทำเลพื้นที่บริการที่ดีที่สุด

ข้อมูลนำเข้า

- บรรทัดแรกเป็นเลขจำนวนเต็มบวกสองตัวบอกจำนวนแถว (M) และจำนวนหลัก (N) ตามลำดับ โดยที่ $2 \leq M, N \leq 1,000$
- บรรทัดที่สองระบุขนาดพื้นที่บริการของร้าน (K) โดยที่ $0 < K < M$ และ $0 < K < N$
- บรรทัดที่สามถึง $M + 2$ ระบุจำนวนประชากรในแถวที่ 1 ถึง M ตามลำดับ ข้อมูลแต่ละบรรทัดประกอบด้วยตัวเลขจำนวนเต็มบวก N จำนวน ซึ่งระบุจำนวนประชากรของพื้นที่ที่เสื่อมย่อย N หลัก เรียงจากซ้ายไปขวาในแถวนั้น ๆ แต่ละจำนวนถูกคั่นด้วยช่องว่าง โดยประชากรในแต่ละพื้นที่ที่เสื่อมย่อยมีจำนวนไม่เกิน 2,000 คน

ข้อมูลส่งออก

จำนวนประชากรภายในพื้นที่บริการที่ดีที่สุด

ตัวอย่างที่ 1

ข้อมูลนำเข้า	ข้อมูลส่งออก
5 10 2 5 9 2 9 1 2 8 9 1 6 9 1 3 9 8 4 2 1 5 7 2 7 9 3 8 5 2 7 6 8 1 6 2 1 7 7 1 9 4 1 8 5 2 3 9 8 5 6 3 3	31

ตัวอย่างที่ 2

ข้อมูลนำเข้า	ข้อมูลส่งออก
6 4 3 7 8 5 1 0 3 5 2 3 3 2 9 9 7 8 9 4 3 5 9 8 6 5 2	55

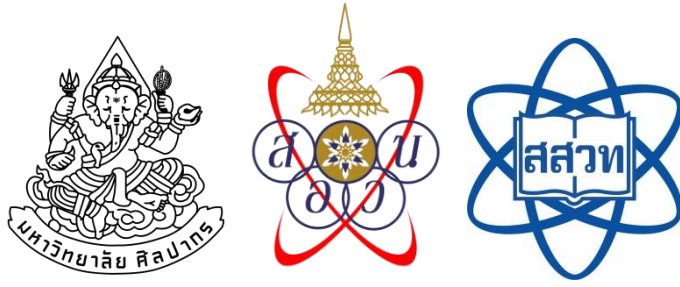
ข้อกำหนด

หัวข้อ	เงื่อนไข
ข้อมูลเข้า	Standard Input (คีย์บอร์ด)
ข้อมูลส่งออก	Standard Output (จอภาพ)
ระยะเวลาสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดลองหนึ่งชุด	1 วินาที
หน่วยความจำสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดลองหนึ่งชุด	32 MB
จำนวนชุดทดสอบ (โปรแกรมประมวลผลครั้งละชุดทดสอบ)	10
คะแนนสำหรับชุดทดสอบแต่ละชุด	10
คะแนนสูงสุดของโจทย์	100
เงื่อนไขการรันโปรแกรม	โปรแกรมจะต้องประมวลผลข้อมูลตามตัวอย่างที่ให้มาได้

ข้อมูลคำสั่งเพิ่มเติม

ผู้เข้าแข่งขันจะต้องระบุชื่อแฟ้มข้อมูลและส่วนหัวของโปรแกรมให้สอดคล้องกับภาษาและคอมไพเลอร์ที่ใช้ดังนี้

ภาษา C และ MinGW 4.4.1 (Code::Blocks บนวินโดวส์)	ภาษา C++ และ MinGW 4.4.1 (Code::Blocks บนวินโดวส์)
/* TASK: location.c LANG: C COMPILER: WCB AUTHOR: YourName YourLastName CENTER: YourCenter */	/* TASK: location.cpp LANG: C++ COMPILER: WCB AUTHOR: YourName YourLastName CENTER: YourCenter */
ภาษา C และ MinGW 3.4.2 (Dev-C++ บนวินโดวส์)	ภาษา C++ และ MinGW 3.4.2 (Dev-C++ บนวินโดวส์)
/* TASK: location.c LANG: C COMPILER: WDC AUTHOR: YourName YourLastName CENTER: YourCenter */	/* TASK: location.cpp LANG: C++ COMPILER: WDC AUTHOR: YourName YourLastName CENTER: YourCenter */
ภาษา C และ GCC 4.4.4 (คอมไพเลอร์บนลินุกซ์)	ภาษา C++ และ GCC 4.4.4 (คอมไพเลอร์บนลินุกซ์)
/* TASK: location.c LANG: C COMPILER: LINUX AUTHOR: YourName YourLastName CENTER: YourCenter */	/* TASK: location.cpp LANG: C++ COMPILER: LINUX AUTHOR: YourName YourLastName CENTER: YourCenter */



ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร พระราชวังสนามจันทร์

ข้อสอบแข่งขันคอมพิวเตอร์โอลิมปิกระดับชาติ ครั้งที่ 8

ข้อสอบมี 3 ข้อ 14 หน้า ให้ทำทุกข้อ เวลา 9.00 – 12.00 น. วันพฤหัสบดี ที่ 17 พฤษภาคม 2555

ฝ่าเขาวงกต (maze)

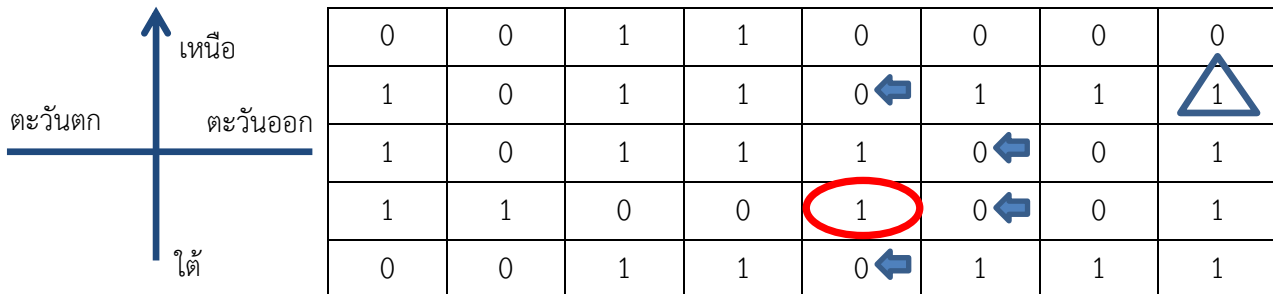
นักล่าขุมทรัพย์นามว่า “อินเดียนา เจ” พลادتพลังตกลงไปในหลุมพรางที่ส่งเขาไปอยู่ในเขาวงกตซึ่งมีทางออกอยู่เพียงตำแหน่งเดียวเท่านั้น เคราะห์ดีที่นายอินเดียนามีแผนที่เขาวงกตติดตัวมาด้วย ทำให้เขาทราบตำแหน่งปัจจุบันของเขาและตำแหน่งของทางออก จากแผนที่ อินเดียนาพบว่าพื้นที่เขาวงกตถูกแบ่งออกเป็นช่องจำนวน M แถว N หลัก โดยแต่ละช่องในแผนที่จะมีเลขหนึ่งหรือเลขศูนย์อย่างใดอย่างหนึ่ง ซึ่งเลขศูนย์แทนกำแพงและเลขหนึ่งแทนทางเดิน นอกจากนี้เขาวงกตยังวางตัวในทิศเหนือ-ใต้ ตะวันออก-ตะวันตกพอดี ดังแสดงในภาพตัวอย่างที่อยู่หน้าถัดไป

อย่างไรก็ตามปัญหาหนักใจมีอยู่ว่า บริเวณที่อินเดียนาตกลงมาไม่ได้เชื่อมต่อกับทางออก อินเดียนาจึงจำเป็นต้องจะต้องระเบิดกำแพงเขาวงกตด้วยระเบิดที่มีติดตัวอยู่เพียงลูกเดียวเท่านั้น นอกจากนี้อินเดียนาทราบว่าระเบิดนี้มีพลังทำลายกำแพงเขาวงกตได้เพียงหนึ่งช่องเท่านั้น

อินเดียนาจึงจำเป็นต้องวางแผนว่าเขาจะต้องเดินในเขาวงกตอย่างไร และใช้ระเบิดทำลายกำแพงตรงพื้นที่ช่องใด จึงจะสามารถเดินไปถึงทางออกได้ อินเดียนาทราบตำแหน่งเริ่มต้นของเขาและตำแหน่งทางออกเท่านั้น และเพื่อให้การวางแผนและประมาณระยะทางเดินเป็นไปโดยง่าย อินเดียนาจะเดินในทิศเหนือ ใต้ ตะวันออก หรือ ตะวันตก เท่านั้น อินเดียนาจะไม่เดินในทิศเฉียงเป็นอันขาด (เช่น ไม่เดินในทิศตะวันออกเฉียงเหนือ เป็นต้น)

ยกตัวอย่างจากแผนที่ในหน้าถัดไป เขาวงกตนี้ประกอบด้วยช่องจำนวนทั้งหมด 5 แถวและ 8 หลัก กำหนดให้อินเดียนาเริ่มต้นในช่องที่ถูกเน้นด้วยวงรี และทางออกอยู่ ณ ตำแหน่งที่เน้นด้วยสามเหลี่ยม หากอินเดียนาระเบิดกำแพงที่ช่องใดช่องหนึ่งที่ถูกเน้นด้วยลูกศรก็จะสามารถเดินไปถึงทางออกได้ การระเบิดกำแพงที่ช่องอื่น ๆ นอกจากหนึ่งในสี่ช่องนี้ จะไม่ทำให้อินเดียนาไปถึงทางออกได้

ยิ่งไปกว่านั้นอินเดียนายังสนใจด้วยว่าทางเดินจากจุดเริ่มต้นไปถึงทางออกที่ใกล้ที่สุดมีระยะทางเท่าใด (ระยะทางนับจากจำนวนช่องที่เดินผ่าน) จากตัวอย่างเดิม ถ้าอินเดียนาระเบิดกำแพงที่ช่อง ณ ตำแหน่งแถวที่สอง หลักที่ห้า หรือ ตำแหน่งแถวที่สาม หลักที่หก จะทำให้ได้ทางเดินที่ใกล้ที่สุดด้วย คือได้ทางเดินที่ผ่านจำนวนช่องทั้งหมด 6 ช่อง (นับช่องที่จุดเริ่มต้นและสิ้นสุดและช่องที่เป็นกำแพงที่ถูกระเบิดด้วย)



จงเขียนโปรแกรมที่มีประสิทธิภาพในการหาจำนวนช่องของกำแพงที่อินเดียนาสามารถทำการระเบิดเพื่อนำอินเดียนาไปสู่ทางออกได้ รวมทั้งหาระยะทางเดินที่สั้นที่สุดจากจุดเริ่มต้นไปจนถึงทางออก

ข้อมูลนำเข้า

- บรรทัดแรกระบุค่า M และ N ซึ่งแทนจำนวนแถวและจำนวนหลักของเขาวงกตตามลำดับ โดยที่ $1 \leq M, N \leq 150$ โดย M และ N ถูกคั่นด้วยช่องว่าง
- บรรทัดที่สองระบุแถว (r_s) และหลัก (c_s) ของช่องที่อินเดียนาเริ่มต้น โดยที่ $1 \leq r_s \leq M$ และ $1 \leq c_s \leq N$ โดย r_s และ c_s ถูกคั่นด้วยช่องว่าง
- บรรทัดที่สามระบุแถว (r_e) และหลัก (c_e) ของช่องที่เป็นทางออก โดยที่ $1 \leq r_e \leq M$ และ $1 \leq c_e \leq N$ โดย r_e และ c_e ถูกคั่นด้วยช่องว่าง
- อีก M บรรทัดถัดมา ในแต่ละบรรทัดจะประกอบไปด้วยเลขจำนวน N ตัว แต่ละตัวคั่นด้วยช่องว่าง โดยเลขศูนย์แทนกำแพง และเลขหนึ่งแทนทางเดิน บรรทัดแรกใน M บรรทัดนี้บอกลักษณะช่องของแถวแรกในเขาวงกต (แถวแรกคือแถวที่อยู่ทางเหนือสุด) เรียงจากหลักทางทิศตะวันตกไปตะวันออก (หลักแรกคือหลักทางทิศตะวันตก) บรรทัดถัดมาบอกลักษณะของแถวที่สอง และเป็นเช่นนี้ไปเรื่อย ๆ จนครบ M บรรทัด

ข้อมูลส่งออก

- บรรทัดแรกระบุจำนวนช่องกำแพงที่อินเดียนาสามารถวางระเบิดและพาอินเดียนาไปถึงทางออกได้
- บรรทัดที่สองระบุระยะทางที่น้อยที่สุดที่อินเดียนาสามารถเดินเพื่อไปถึงทางออก โดยระยะทางคือจำนวนช่องที่อินเดียนาเดินผ่านทั้งหมด ซึ่งนับรวมช่องที่เป็นจุดเริ่มต้นและจุดสิ้นสุด พร้อมทั้งนับรวมช่องกำแพงที่อินเดียนาระเบิดด้วย

ตัวอย่างที่ 1

ข้อมูลนำเข้า	ข้อมูลส่งออก
5 8 4 5 2 8 0 0 1 1 0 0 0 0 1 0 1 1 0 1 1 1 1 0 1 1 1 0 0 1 1 1 0 0 1 0 0 1 0 0 1 1 0 1 1 1	4 6

ตัวอย่างที่ 2

ข้อมูลนำเข้า	ข้อมูลส่งออก
6 8 1 4 2 7 0 0 1 1 0 0 0 0 1 0 1 1 0 0 1 1 1 0 1 1 1 0 0 1 1 1 0 0 1 0 0 1 0 0 1 1 0 1 1 1 0 1 0 1 1 1 1 1	4 13

ตัวอย่างที่ 3

ข้อมูลนำเข้า	ข้อมูลส่งออก
5 8 1 4 2 1 0 0 1 1 0 0 0 0 1 0 1 1 0 0 1 1 1 0 1 1 1 0 0 1 1 1 0 0 1 0 0 1 0 1 1 1 0 1 1 1	5 5

หมายเหตุ

1. สำหรับข้อมูลเข้าทุกชุด อินดิเค้นาจำเป็นต้องใช้ระเบิดหนึ่งลูกในการไปถึงทางออก
2. สำหรับข้อมูลเข้าทุกชุด ตำแหน่งเริ่มต้นและทางออกจะตรงกับช่องที่มีเลขหนึ่งอยู่ในแผนที่

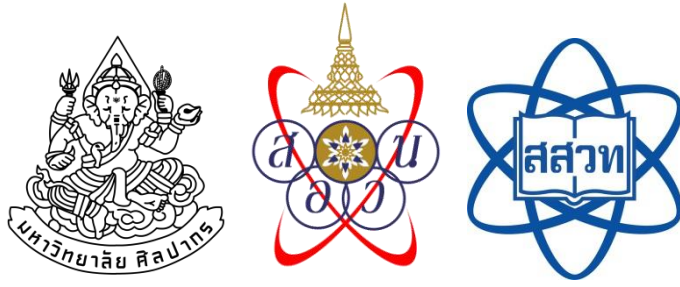
ข้อกำหนด

หัวข้อ	เงื่อนไข
ข้อมูลเข้า	Standard Input (คีย์บอร์ด)
ข้อมูลส่งออก	Standard Output (จอภาพ)
ระยะเวลาสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดลองหนึ่งชุด	1 วินาที
หน่วยความจำสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดลองหนึ่งชุด	32 MB
จำนวนชุดทดสอบ (โปรแกรมประมวลผลครั้งละชุดทดสอบ)	10
คะแนนสำหรับชุดทดสอบแต่ละชุด	10
คะแนนสูงสุดของโจทย์	100
เงื่อนไขการรันโปรแกรม	โปรแกรมจะต้องประมวลผลข้อมูลตามตัวอย่างที่ให้มาได้

ข้อมูลคำสั่งเพิ่มเติม

ผู้เข้าแข่งขันจะต้องระบุชื่อแฟ้มข้อมูลและส่วนหัวของโปรแกรมให้สอดคล้องกับภาษาและคอมไพเลอร์ที่ใช้ดังนี้

ภาษา C และ MinGW 4.4.1 (Code::Blocks บนวินโดวส์)	ภาษา C++ และ MinGW 4.4.1 (Code::Blocks บนวินโดวส์)
<pre>/* TASK: maze.c LANG: C COMPILER: WCB AUTHOR: YourName YourLastName CENTER: YourCenter */</pre>	<pre>/* TASK: maze.cpp LANG: C++ COMPILER: WCB AUTHOR: YourName YourLastName CENTER: YourCenter */</pre>
ภาษา C และ MinGW 3.4.2 (Dev-C++ บนวินโดวส์)	ภาษา C++ และ MinGW 3.4.2 (Dev-C++ บนวินโดวส์)
<pre>/* TASK: maze.c LANG: C COMPILER: WDC AUTHOR: YourName YourLastName CENTER: YourCenter */</pre>	<pre>/* TASK: maze.cpp LANG: C++ COMPILER: WDC AUTHOR: YourName YourLastName CENTER: YourCenter */</pre>
ภาษา C และ GCC 4.4.4 (คอมไพเลอร์บนลินุกซ์)	ภาษา C++ และ GCC 4.4.4 (คอมไพเลอร์บนลินุกซ์)
<pre>/* TASK: maze.c LANG: C COMPILER: LINUX AUTHOR: YourName YourLastName CENTER: YourCenter */</pre>	<pre>/* TASK: maze.cpp LANG: C++ COMPILER: LINUX AUTHOR: YourName YourLastName CENTER: YourCenter */</pre>



ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร พระราชวังสนามจันทร์

ข้อสอบแข่งขันคอมพิวเตอร์โอลิมปิกระดับชาติ ครั้งที่ 8

ข้อสอบมี 3 ข้อ 14 หน้า ให้ทำทุกข้อ เวลา 9.00 – 12.00 น. วันพฤหัสบดี ที่ 17 พฤษภาคม 2555

จัดลำดับการทดลอง (schedule)

นายเมธาต้องการทำการทดลองทางวิทยาศาสตร์อยู่สองงาน โดยที่แต่ละงานประกอบด้วยขั้นตอนทั้งหมด N ขั้นตอน คือขั้นตอน $J_1, J_2, J_3, \dots, J_N$ สำหรับงานแรก และ ขั้นตอน $K_1, K_2, K_3, \dots, K_N$ สำหรับงานที่สอง ซึ่งแต่ละขั้นตอนอาจใช้เวลาเท่ากันหรือต่างกันได้ อย่างไรก็ตามขั้นตอนในงานเดียวกันไม่สามารถสลับลำดับกันได้ กล่าวคือ สำหรับงานแรก ขั้นตอน J_1 จะต้องถูกทำเป็นอันดับแรก และขั้นตอน J_2, J_3, \dots, J_N จะถูกทำต่อมาตามลำดับดังกล่าว สำหรับงานที่สองก็เช่นกัน ขั้นตอน K_1 จะต้องถูกทำเป็นอันดับแรก และขั้นตอน K_2, K_3, \dots, K_N จะถูกทำตามลำดับ

แม้จะไม่สามารถสลับลำดับขั้นตอนในงานเดียวกันได้ แต่เมธาก็สามารถสลับลำดับขั้นตอนระหว่างงานแรกกับงานที่สองได้ เป็นต้นว่าถ้า $N = 3$ เมธาสามารถที่จะทำการทดลองในลำดับ $K_1, K_2, J_1, K_3, J_2, J_3$ เพราะลักษณะนี้เป็นการทำการทดลองแต่ละงานตามลำดับจากขั้นตอนแรกไปขั้นตอนสุดท้าย

โชคไม่ดีนัก เมธาพบว่าห้องปฏิบัติการมีเครื่องมือสำหรับทำการทดลองอยู่เพียงชุดเดียว และงานทั้งสองก็ต้องใช้เครื่องมือชุดเดียวกันนี้ ยิ่งไปกว่านั้นเครื่องมือสามารถทำงานได้เพียง M นาฬิกาในแต่ละวัน และการทดลองแต่ละขั้นตอนก็ต้องดำเนินการอย่างต่อเนื่องให้สำเร็จภายในวันเดียวเท่านั้น

ยกตัวอย่างเช่น หากงานแต่ละงานมีสองขั้นตอน ($N = 2$) และใช้เครื่องได้ 300 นาฬิกาต่อวัน ($M = 300$) เมื่อ $J_1 = 200, J_2 = 150, K_1 = 50$ และ $K_2 = 150$ ถ้าหากเมธাজัดลำดับการทดลองเป็น J_1, J_2, K_1, K_2 ตามลำดับ ขั้นตอน J_2 จะไม่สามารถทำได้ในวันแรกเพราะเวลารวมในวันแรกจะเกิน 300 นาฬิกา ทำให้ต้องเลื่อนไปทำในวันที่สอง และการทดลองตามลำดับนี้ จะใช้เวลาทั้งหมด 3 วัน โดยวันสุดท้าย (วันที่สาม) จะใช้เวลาทั้งหมด 150 นาฬิกา แต่หากเมธাজัดลำดับการทดลองใหม่เป็น J_1, K_1, K_2, J_2 การทดลองทั้งหมดจะแล้วเสร็จในเวลาเพียง 2 วัน โดยวันสุดท้าย (วันที่สอง) จะใช้เวลาทั้งหมด 300 นาฬิกา

จงเขียนโปรแกรมที่มีประสิทธิภาพในการจัดลำดับขั้นตอนการทดลองที่ทำให้การทดลองทั้งสองงานเสร็จด้วยเวลาที่น้อยที่สุด

ข้อมูลเข้า

1. บรรทัดแรกเป็นเลขจำนวนเต็ม M ระบุเวลาที่สามารถใช้เครื่องมือได้ในแต่ละวัน โดยที่ $1 \leq M \leq 600$ และ M มีหน่วยเป็นนาทีก
2. บรรทัดที่สองเป็นจำนวนเต็ม N ระบุจำนวนขั้นตอนในแต่ละงานโดยที่ $2 \leq N \leq 1000$
3. บรรทัดที่สามเป็นจำนวนเต็มบวก N จำนวน คือ $a_1, a_2, a_3, \dots, a_N$ แต่ละตัวคั่นด้วยช่องว่าง จำนวนแต่ละจำนวนนี้แทนเวลาที่ต้องใช้ทำการทดลองขั้นตอน $J_1, J_2, J_3, \dots, J_N$ ของงานแรกตามลำดับ มีหน่วยเป็นนาทีก จำนวนแต่ละจำนวนถูกคั่นด้วยช่องว่าง โดยที่ $1 \leq a_i \leq M, i = 1, \dots, N$
4. บรรทัดที่สี่เป็นจำนวนเต็มบวก N จำนวนในลักษณะเดียวกับบรรทัดที่สาม แต่จำนวนเหล่านี้แทนเวลาที่ต้องใช้ในการทดลองขั้นตอน $K_1, K_2, K_3, \dots, K_N$ สำหรับงานที่สอง ซึ่งเวลาเหล่านี้มีค่ามากกว่าหรือเท่ากับหนึ่งและน้อยกว่าหรือเท่ากับ M

ข้อมูลส่งออก

ข้อมูลส่งออกระบุจำนวนวันที่ต้องใช้ในการทดลองของเมธา และจำนวนนาทีกที่ใช้ในการทดลองวันสุดท้าย โดยข้อมูลส่งออกต้องอยู่ในรูปแบบดังต่อไปนี้

1. บรรทัดแรกระบุจำนวนวันที่ต้องใช้ในการทดลองเป็นจำนวนเต็ม
2. บรรทัดที่สองระบุจำนวนนาทีกที่ใช้สำหรับการทดลองในวันสุดท้าย โดยที่จำนวนนาทีกนี้มีค่าตั้งแต่หนึ่งและไม่เกิน M

หมายเหตุ เวลาในการทดลองที่ดีที่สุดถือตามจำนวนวันเป็นลำดับแรก ในกรณีที่การจัดลำดับขั้นตอนสองแบบใช้จำนวนวันเท่ากัน จะนับเวลาที่ดียุติสุดจากจำนวนนาทีกที่ใช้ในวันสุดท้าย

ตัวอย่างที่ 1

ข้อมูลนำเข้า	ข้อมูลส่งออก
8 4 4 5 6 4 3 3 2 4	4 8

ตัวอย่างที่ 2

ข้อมูลนำเข้า	ข้อมูลส่งออก
8 6 2 3 4 5 3 2 6 2 3 2 4 5	6 5

ตัวอย่างที่ 3

ข้อมูลนำเข้า	ข้อมูลส่งออก
10 12 1 7 5 4 3 6 2 3 4 5 1 8 3 4 4 8 3 9 1 7 3 2 4 5	11 8

ข้อกำหนด

หัวข้อ	เงื่อนไข
ข้อมูลเข้า	Standard Input (คีย์บอร์ด)
ข้อมูลส่งออก	Standard Output (จอภาพ)
ระยะเวลาสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดลองหนึ่งชุด	1 วินาที
หน่วยความจำสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดลองหนึ่งชุด	32 MB
จำนวนชุดทดสอบ (โปรแกรมประมวลผลครั้งละชุดทดสอบ)	10
คะแนนสำหรับชุดทดสอบแต่ละชุด	10
คะแนนสูงสุดของโจทย์	100
เงื่อนไขการรันโปรแกรม	โปรแกรมจะต้องประมวลผลข้อมูลตามตัวอย่างที่ให้มาได้

ข้อมูลคำสั่งเพิ่มเติม

ผู้เข้าแข่งขันจะต้องระบุชื่อแฟ้มข้อมูลและส่วนหัวของโปรแกรมให้สอดคล้องกับภาษาและคอมไพเลอร์ที่ใช้ดังนี้

ภาษา C และ MinGW 4.4.1 (Code::Blocks บนวินโดวส์)	ภาษา C++ และ MinGW 4.4.1 (Code::Blocks บนวินโดวส์)
/* TASK: schedule.c LANG: C COMPILER: WCB AUTHOR: YourName YourLastName CENTER: YourCenter */	/* TASK: schedule.cpp LANG: C++ COMPILER: WCB AUTHOR: YourName YourLastName CENTER: YourCenter */
ภาษา C และ MinGW 3.4.2 (Dev-C++ บนวินโดวส์)	ภาษา C++ และ MinGW 3.4.2 (Dev-C++ บนวินโดวส์)
/* TASK: schedule.c LANG: C COMPILER: WDC AUTHOR: YourName YourLastName CENTER: YourCenter */	/* TASK: schedule.cpp LANG: C++ COMPILER: WDC AUTHOR: YourName YourLastName CENTER: YourCenter */
ภาษา C และ GCC 4.4.4 (คอมไพเลอร์บนลินุกซ์)	ภาษา C++ และ GCC 4.4.4 (คอมไพเลอร์บนลินุกซ์)
/* TASK: schedule.c LANG: C COMPILER: LINUX AUTHOR: YourName YourLastName CENTER: YourCenter */	/* TASK: schedule.cpp LANG: C++ COMPILER: LINUX AUTHOR: YourName YourLastName CENTER: YourCenter */