

สี่เหลี่ยมบิน (flying rectangles)

ข้อจำกัด: เวลาการทำงาน 1 วินาที หน่วยความจำ 16 MB

■ ล้วนหนึ่งผู้คนที่ตกใจเมื่อพบว่าสี่เหลี่ยมบิน K รูป ($1 \leq K \leq 2$) มาอยู่เหนือหน้าต่างที่มีลักษณะเป็นตารางกริดขนาด $N \times N$ เพื่อความสะดวก เราวางตารางกริดดังกล่าวลงบนระนาบ โดยมีมุมล่างซ้ายอยู่ที่จุด $(0,0)$ ■ และมุมบนขวาที่จุด (N,N) รูปสี่เหลี่ยมบินทั้ง K รูป จะมีพิกัดของจุดมุมเป็นจำนวนเต็ม ■ และจะไม่ทับกัน หรือว่ามีขอบติดกัน รวมทั้งไม่ใช้จุดมุมร่วมกันด้วย

ให้เขียนโปรแกรมเพื่อหาพิกัดของรูปสี่เหลี่ยมบินเหล่านั้น โดยในการตรวจสอบตำแหน่งของรูปสี่เหลี่ยมบินให้ใช้ไลบรารีดังระบุด้านล่าง ส่วนในการตอบให้พิมพ์ผลลัพธ์ทาง standard output

การเขียนโปรแกรม

จะมีไลบรารีสำหรับเขียนโปรแกรมให้ โดยฟังก์ชันที่ให้มามีรายละเอียดดังนี้

- `int get_K()`
คืนค่า K ($1 \leq K \leq 2$) คุณต้องเรียกฟังก์ชันนี้ก่อนการทำงานอื่น ๆ
- `int get_N()`
คืนค่า N ($1 \leq N \leq 1,000,000$)
- `int check(int x1, int y1, int x2, int y2)`
พิจารณาสี่เหลี่ยมที่มีจุดมุมล่างซ้ายเป็น $(x1,y1)$ ■ และมุมบนขวา $(x2,y2)$ ■ แล้วคืนจำนวนของสี่เหลี่ยมบินที่ทับกับสี่เหลี่ยมดังกล่าว (จะไม่นับกรณีที่ ■ ซ้อนกัน)
สามารถเรียกใช้ฟังก์ชันนี้ได้ไม่เกิน 400 ครั้ง

ในการใช้งานฟังก์ชันดังกล่าว ในโปรแกรมให้คุณ `#include "rectlib.h"`

ข้อมูลส่งออก

ให้พิมพ์พิกัดของสี่เหลี่ยมบินทั้ง K รูป จะเรียงลำดับสี่เหลี่ยมบินอย่างไรก็ได้รูปละหนึ่งบรรทัด โดยในแต่ละรูปให้พิมพ์จำนวนเต็มสี่จำนวนคือ $x1\ y1\ x2\ y2$ โดยที่ $(x1,y1)$ ■ แทนมุมล่างซ้ายของสี่เหลี่ยมบิน ■ และ $(x2,y2)$ ■ แทนมุมบนขวาของสี่เหลี่ยมบิน

ตัวอย่างไลบรารีเพื่อทดสอบ

ตัวอย่างไลบรารีจะมีให้ดาวน์โหลดที่หน้า Tasks โดยจะมี ■ ไฟล์ `rectlib.h` ■ และ `rectlib.c` ตัวอย่างไลบรารีจะอ่านข้อมูลป้อนเข้าจาก standard input (ซึ่งคุณสามารถ ■ แก้ไขใน `rectlib.c` ให้อ่านจาก ■ ไฟล์ข้อมูลได้) โดยมีรูปแบบเป็นดังนี้ บรรทัด ■ ระบุจำนวนเต็มสองจำนวน $K\ N$ จากนั้นอีก K บรรทัดระบุข้อมูลของสี่เหลี่ยมบิน ■ แต่ละรูป โดยประกอบไปด้วยจำนวนเต็มสี่จำนวน $x1\ y1\ x2\ y2$ เพื่อระบุว่าสี่เหลี่ยมบินมีพิกัดมุมล่างซ้ายที่ $(x1,y1)$ ■ และมุมบนขวาเป็น $(x2,y2)$ ■ ไลบรารีที่ให้ ■ มีไว้เพื่อใช้ทดสอบเท่านั้น ในการตรวจโปรแกรมของคุณ เราสามารถใช้ไลบรารีตัวอื่นที่มี interface ตามที่กำหนดได้

ตัวอย่างการทำงาน

ตัวอย่างที่ 1

สมมติให้ $K = 1, N = 10$, สี่เหลี่ยมบินมีจุดมุมอยู่ที่ $(1,1)$ ■ และ $(9,9)$

```
get_K();           // return 1
get_N();           // return 10
```

```

check(0,0,1,1); // return 0
check(0,0,2,2); // return 1
check(0,0,1,3); // return 0
...

```

output:

1 1 9 9

ตัวอย่างที่สอง

สมมติให้ $K = 2$, $N = 10$ หน่วย, สีเหลี่ยมบิรูป■แรกมีจุดมุมอยู่ที่ (1,1) ■และ (4,4) รูปที่สองมีจุดมุมที่ (2,5) ■และ (9,9)

```

get_K(); // return 2
get_N(); // return 10
check(0,0,1,1); // return 0
check(0,0,2,2); // return 1
check(0,0,1,3); // return 0
check(0,0,6,6); // return 2
check(0,4,5,5); // return 0
...

```

output:

2 5 9 9

1 1 4 4

การให้คะแนน

ถ้าโปรแกรมตอบความยาวที่ถูกต้องจะได้คะแนนเต็ม ถ้าตอบความยาวผิด หรือเรียกใช้ไลบรารีมากกว่าที่กำหนด หรือเรียกใช้ผิด (เรียกออกนอกขอบเขต) จะได้คะแนนศูนย์

ขอบเขตเพิ่มเติม

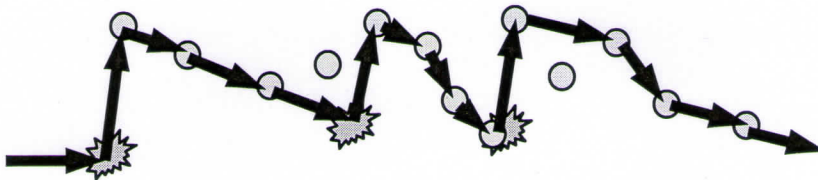
ไม่น้อยกว่า 30% ของข้อมูลชุดทดสอบ $K = 1$.

ก้านกล้วย, บิน! (flying elephant)

ข้อจำกัด: เวลาการทำงาน 1 วินาที หน่วยความจำ 16 MB

ก้านกล้วยเป็นช้างมหัศจรรย์ กล่าวคือ 1. เขาสามารถใช้หูกางเพื่อร่อนในอากาศได้ ■ ด้วยน้ำหนักที่มาก ความสูงที่เขาจะลดลงเรื่อย ๆ ■ และ 2. เขาสามารถใช้เครื่องยนต์เจ็ทติดที่หลังเพื่อบินได้ อย่างไรก็ตามด้วยน้ำหนักมหาศาล เครื่องยนต์จึงสามารถช่วยยกตัวเขาขึ้นได้ไม่เกินสามครั้ง

มีกล้วย N หวีห้อยอยู่จากเชือกที่ซึ่งเป็นเส้นตรงทางยาว กล้วยหวีที่ i ห้อยอยู่สูง h_i เมตร ก้านกล้วยต้องการจะเก็บกล้วยเหล่านี้ โดยมีเงื่อนไขว่าต้องเก็บเรียงไปตามลำดับหวีที่ห้อยอยู่ (■ ไม่จำเป็นต้องเก็บทุกหวี) ด้วยอัจฉริยภาพของก้านกล้วย เขาสามารถใช้เครื่องบินเจ็ทยกตัวขึ้น ■ และร่อนเพื่อเก็บกล้วยได้ อย่างไรก็ตามการบังคับเครื่องยนต์เจ็ทต้องใช้สมาธิมาก เขาจึงไม่สามารถยกตัวขึ้นพร้อม ๆ กับเก็บกล้วยได้ ตัวอย่างการเก็บกล้วยของก้านกล้วย ■ แสดงได้ดังรูปด้านล่าง



ในรูปดังกล่าวก้านกล้วยเก็บกล้วยได้ทั้งสิ้น 11 หวี โดยมีการใช้เครื่องยนต์เจ็ททั้งสิ้น 3 ครั้ง สังเกตว่าเขาไม่สามารถเก็บกล้วยหวีที่ 4 ได้ ■ แม้จะอยากก็ตาม เพราะว่าตอนนั้นกำลังบังคับเครื่องยนต์เจ็ทอยู่

กล่าวอย่างเป็นทางการก็คือ ถ้าก้านกล้วยเก็บกล้วยหวีที่ i ■ แล้วถ้าเขาจะเก็บกล้วยหวีที่ j ที่ $j > i$ เป็นหวีถัดไปได้ ก็ต่อเมื่อ $h_j < h_i$ หรือไม่ก็ต้องใช้เครื่องยนต์เจ็ทหนึ่งครั้งในการยกตัวขึ้นไป นอกจากนี้ในการเริ่มเก็บกล้วยหวี ■ แรก เขาก็ต้องใช้เครื่องยนต์เจ็ทด้วยอีกหนึ่งครั้งเสมอ (เพราะว่าเขาเริ่มเก็บจากพื้น)

ข้อมูลป้อนเข้า

บรรทัด ■ แรกระบุจำนวนเต็ม N ($1 \leq N \leq 10,000$) จากนั้นอีก N บรรทัด ระบุความสูงของกล้วย บรรทัดที่ $1 + i$ สำหรับค่า $i = 1, \dots, N$ ระบุจำนวนเต็ม h_i ($0 \leq h_i \leq 1,000,000,000$)

ข้อมูลส่งออก

มี 1 บรรทัด ■ ระบุจำนวนหวีของกล้วยที่ก้านกล้วยเก็บได้มากที่สุด

ตัวอย่าง

input:	output:
4	3
1	
2	
3	
4	

input:

9
2
1
4
3
6
5
8
7
3

output:

7

ขอบเขตเพิ่มเติม

ไม่น้อยกว่า 30% ของข้อมูลชุดทดสอบมี $N \leq 150$

ไม่น้อยกว่า 50% ของข้อมูลชุดทดสอบมี $N \leq 1000$