

Convex Hull (convexhull)

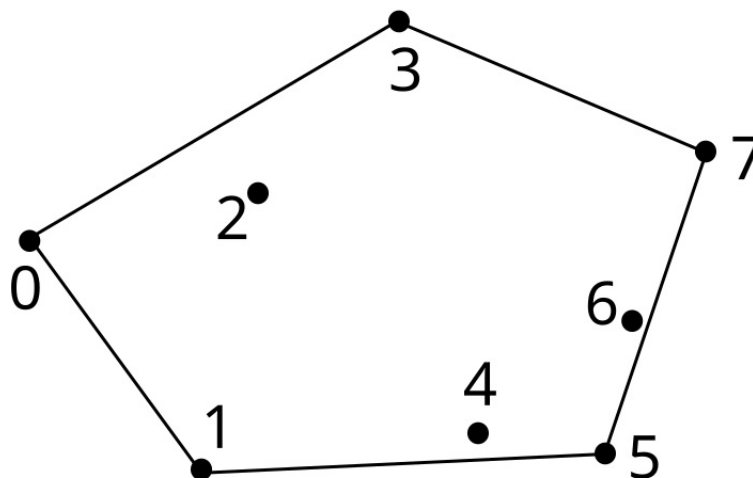
สำหรับเซตของจุด S ใด ๆ บนระนาบ **convex hull** C ของ S คือเซตของจุดบนระนาบที่มีทุกจุดใน S เหล่านั้น และสำหรับจุดสองจุดใด ๆ ใน C ทุก ๆ จุดในส่วนของเส้นตรงที่เชื่อมระหว่างจุดสองจุดนั้นจะต้องอยู่ใน C ด้วย

มีจุด N จุดบนระนาบที่เราไม่ทราบพิกัด (เรียกเป็นจุดที่ 0 ถึงจุดที่ $N - 1$) อย่างไรก็ตามเราทราบว่าจุดเหล่านี้เรียงลำดับตามพิกัดในแกน x จากน้อยไปหามาก ไม่มีจุดสองจุดใด ๆ ที่มีพิกัดบนแกน x เท่ากัน และไม่มีจุดสามจุดใด ๆ ที่อยู่บนเส้นตรงเดียวกัน

เราต้องการหาว่าจุดที่อยู่บนขอบของ convex hull ของจุด N จุดเหล่านี้ มีจุดใดบ้าง? โดยคุณสามารถเรียกใช้ฟังก์ชันเหล่านี้ประกอบการหาคำตอบ ทั้งสองฟังก์ชันจะรับอาร์กิวเมนต์เป็นเวกเตอร์ของหมายเลขของจุด

- ฟังก์ชัน `is_critical(vector<int> V)`
ฟังก์ชันนี้จะคืนค่า `true` ก็ต่อเมื่อจุดมุมของ convex hull ของเซตของจุด V เท่ากับ V (นั่นคือทุกจุดใน V อยู่ในขอบของ convex hull ของ V) และคืนค่า `false` ในกรณีอื่น ๆ คุณจะสามารเรียกฟังก์ชัน `is_critical` ได้ไม่เกิน Q ครั้ง
- ฟังก์ชัน `triangle(vector<int> V)`
สำหรับฟังก์ชันนี้ ก่อนที่คุณจะเรียกใช้ คุณต้องมั่นใจก่อนว่า convex hull ของเซตของจุด V นั้นประกอบไปด้วยจุด 3 จุดเท่านั้น ถ้าเงื่อนไขดังกล่าวเป็นจริง ฟังก์ชันจะคืนเซตของจุดสามจุดที่เป็น convex hull ของ V นอกจากนี้ ลำดับของจุดที่ได้รับคืนมาจะเรียงตามพิกัดในแกน y จากน้อยไปหามาก ในกรณีที่เงื่อนไขข้างต้นไม่จริง เกรดเดอร์จะถือว่าโปรแกรมของคุณทำงานผิด และจะจบการทำงานทันที คุณจะเรียกฟังก์ชันนี้กี่ครั้งก็ได้ (ถ้าเซตของจุด V ตรงตามเงื่อนไข)

พิจารณาตัวอย่างต่อไปนี้ที่ $N = 8$ รูปด้านล่างแสดงจุดและ convex hull



ด้านล่างเป็นตัวอย่างของการเรียกใช้ฟังก์ชัน `is_critical` และ `triangle` กับเซตของจุดในตัวอย่างนี้

การเรียกฟังก์ชัน	ค่าที่คืน	คำอธิบายเพิ่มเติม
<code>is_critical([0,1,4,2])</code>	true	
<code>is_critical([0,2,1,7,5])</code>	true	
<code>is_critical([2,3,7])</code>	true	
<code>is_critical([0,1,3])</code>	true	
<code>is_critical([1,2,4,5,7])</code>	false	
<code>is_critical([0,1,2,3])</code>	false	
<code>triangle([0,1,2,3])</code>	[0,1,3]	
<code>triangle([6,7,5,1,4])</code>	[1,5,7]	
<code>triangle([0,1,2,3,4])</code>	จบการทำงาน	เนื่องจากจุดมุมของ convex hull ของเซตดังกล่าวคือ [0, 1, 4, 3] การเรียกจึงไม่ตรงตามเงื่อนไข

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
vector<int> convexhull(int N)
```

- ฟังก์ชันนี้จะถูกเรียกหนึ่งครั้ง โดยที่ตัวแปร N จะระบุจำนวนจุดบนระนาบ
- ฟังก์ชันจะต้องคืนเวกเตอร์ของจุดมุมทั้งหมดของ convex hull ของเซตของจุด N จุดเหล่านี้
- จุดในเวกเตอร์จะเรียงลำดับอย่างไรก็ได้

ฟังก์ชัน `convexhull` จะสามารถเรียกใช้ฟังก์ชันสองฟังก์ชันต่อไปนี้ได้

```
bool is_critical(vector<int> V)
```

- คุณสามารถเรียกฟังก์ชันนี้ได้ไม่เกิน Q ครั้ง
- ฟังก์ชันจะคืนค่า true ก็ต่อเมื่อจุดมุมของ convex hull ของจุดทุกจุดในเวกเตอร์ V คือ V (นั่นคือทุกจุดในเซต V คือจุดมุมของ convex hull ของ V)
- การันตีว่าถ้า $V.size()$ น้อยกว่าหรือเท่ากับ 3 จะคืนค่า true

```
vector<int> triangle(vector<int> V)
```

- ในการเรียกฟังก์ชันนี้มีเงื่อนไขว่า convex hull ของจุดใน V จะต้องประกอบไปด้วยจุดมุม 3 จุดเท่านั้น
- ถ้าเงื่อนไขข้างต้นเป็นจริง ฟังก์ชันจะคืนเวกเตอร์ของจุดที่เป็นจุดมุมของ convex hull (ซึ่งจะมี 3 จุดพอดี)

โดยที่จุดในเวกเตอร์จะเรียงลำดับตามพิกัดในแกน y จากน้อยไปหามาก

- ถ้าเงื่อนไขข้างต้นไม่จริง เกรดเดอร์จะถือว่าโปรแกรมของคุณทำงานผิดพลาด และจบการทำงานทันที
- คุณสามารถเรียกฟังก์ชันนี้กี่ครั้งก็ได้ (ตราบเท่าที่ไม่ผิดเงื่อนไข)

เงื่อนไข

- $3 \leq N \leq 300$
- $Q = 20\,000$

ปัญหาย่อย

1. (27 คะแนน) $N \leq 20$
2. (35 คะแนน) $N \leq 100$
3. (38 คะแนน) $N \leq 300$

ในปัญหาย่อยที่ 3 จะมีการให้คะแนนบางส่วน กำหนดค่า M แทนจำนวนครั้งของการเรียกใช้ฟังก์ชัน `is_critical` โดยคะแนนที่ได้จะถูกประมาณเป็นทศนิยมสองตำแหน่งดังนี้

เงื่อนไข	คะแนนที่ได้
$M > 20\,000$	0
$10\,000 < M \leq 20\,000$	$76 - \frac{3.8M}{1000}$
$M \leq 10\,000$	38

ตัวอย่าง

จากตัวอย่างข้างต้น เกรดเดอร์จะเรียกฟังก์ชัน `convexhull` ดังนี้

```
convexhull(8)
```

ฟังก์ชันดังกล่าวถ้าเรียกใช้งานฟังก์ชัน `is_critical` จะมีค่าที่คืนกลับมาดังต่อไปนี้

```
is_critical([0,1,4,2]) // return true
is_critical([0,2,1,7,5]) // return true
is_critical([2,3,7]) // return true
is_critical([0,1,3]) // return true
is_critical([1,2,4,5,7]) // return false
is_critical([0,1,2,3]) // return false
```

ถ้าเรียกฟังก์ชัน `triangle` แบบที่ตรงตามเงื่อนไข จะมีค่าที่คืนกลับมาดังนี้

```
triangle([0,1,2,3]) // return [0,1,3]
triangle([6,7,5,1,4]) // return [1,5,7]
```

ถ้าเรียกฟังก์ชัน `triangle` ดังด้านล่างนี้

```
triangle([0,1,2,3,4])
```

จะไม่ตรงตามเงื่อนไข เพราะว่า convex hull ของเซตของจุดดังกล่าว มีจุดมุมคือ $[0, 1, 4, 3]$ ในกรณีนี้ โปรแกรมจะจบการทำงาน และถือว่าโปรแกรมของคุณให้คำตอบไม่ถูกต้อง

ฟังก์ชัน `convexhull` ที่ถูกต้องจะเป็นค่าเวกเตอร์ (จะเรียงลำดับอย่างไรก็ได้)

```
[0, 1, 3, 5, 7]
```

เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างจะอ่านข้อมูลนำเข้าดังนี้

- บรรทัด 1: $N \ Q$
- บรรทัด $2 + i$ สำหรับ $0 \leq i < N$: $X[i] \ Y[i]$
โดยที่ $(X[i], Y[i])$ จะเป็นพิกัดของจุดบนระนาบ และ $0 \leq X[i] \leq 1\,000\,000\,000, 0 \leq Y[i] \leq 1\,000\,000\,000$

ถ้าโปรแกรมตอบ convex hull ถูกต้อง โดยที่ใช้จำนวนครั้งในการเรียก `is_critical` ไม่เกิน Q ครั้ง เกรตเตอร์จะพิมพ์จำนวนครั้งที่เรียกใช้ฟังก์ชันดังกล่าว

ถ้าเรียก `is_critical` เกิน Q ครั้ง หรือมีการเรียกฟังก์ชัน `triangle` ที่ผิดเงื่อนไข หรือคำตอบจากฟังก์ชัน `convexhull` ไม่ถูกต้อง เกรตเตอร์จะจบการทำงานและพิมพ์ข้อผิดพลาด

ขีดจำกัด

- Time limit: 5 seconds
- Memory limit: 512 MB