

นำภัย

จงเขียนฟังก์ชันต่าง ๆ ข้างล่างนี้ ให้ทำงานตามหน้าที่ที่เขียนใน comment

จำนวนช่องทั้งหมดของอาร์เรย์ **A** หาได้จาก **A.shape** เช่น ถ้า **A.shape** มีค่า **(3,4,5)** **A** มี 60 ช่อง

```
import numpy as np
```

```
def eq(A, B, p):
```

```
# A และ B เป็นอาร์เรย์ที่มีขนาดเท่ากัน (ก็มิติก็ได้), p เป็นจำนวนระหว่าง 0 ถึง 100
```

```
# คืน True ถ้าข้อมูลใน A กับใน B ที่ตำแหน่งเดียวกันมีค่าเท่ากันอย่างน้อยร้อยละ p ของจำนวนช่องทั้งหมด
```

```
# ถ้าไม่มีถึงก็คืน False
```

$eq\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 0 \\ 7 & 0 & 9 \end{bmatrix}, 70\right)$ ได้ True

```
def closest_point_indexes(points, p):
```

```
# points คืออาร์เรย์สองมิติที่มี 2 คอลัมน์ คอลัมน์ 0 เก็บพิกัด x คอลัมน์ 1 เก็บพิกัด y
```

```
# p คืออาร์เรย์มิติเดียว 2 ช่อง เก็บพิกัด x และ y
```

```
# คืน อาร์เรย์ที่เก็บ index ของจุดต่าง ๆ ใน points ที่อยู่ใกล้กับจุด p มากสุด
```

```
# ถ้ามีหลายจุดที่ใกล้สุดเท่ากัน ให้เก็บ index ทั้งหมดเรียงจากน้อยไปมาก
```

$closest_point_indexes\left(\begin{bmatrix} 1 & 0 \\ 9 & 9 \\ -1 & 0 \\ 0 & 1 \end{bmatrix}, [0 \ 0]\right)$ ได้ [0 2 3]

```
def number_of_inversions(A):
```

```
# A เป็นอาร์เรย์หนึ่งมิติเก็บจำนวนเต็ม
```

```
# คืน จำนวนคู่ข้อมูลใน A ที่ตัวทางซ้ายมากกว่าตัวขวา
```

```
# (คือมีข้อมูล A[i] กับ A[j] ที่ i < j แต่ A[i] > A[j] อยู่กี่คู่)
```

```
# เช่น [1 2 9 4 8 7] มี 4 คู่คือ 9 กับ 4, 9 กับ 8, 9 กับ 7 และ 8 กับ 7
```

```
# [9 7 5 3 2] มี 10 คู่ เพราะทุกคู่มีตัวซ้ายมากกว่าตัวขวา ในขณะที่ [2 4 6 8] มี 0 คู่
```

```
exec(input().strip()) # ต้องมีคำสั่งนี้ ตรงนี้ ตอนส่งให้ Grader ตรวจ
```

ข้อมูลนำเข้า

คำสั่งภาษา Python ที่ใช้ทดสอบการทำงานของฟังก์ชัน

ข้อมูลส่งออก

ผลที่ได้จากการสั่งทำงานคำสั่งที่ได้รับ

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
<code>print(eq(np.array([1,2]), np.array([1,0]), 50))</code>	True
<code>print(eq(np.array([[1,2]]), np.array([[1,1]]), 51))</code>	False
<code>print(closest_point_indexes(np.array([9,9]), np.array([1,1])))</code>	[1]
<code>print(closest_point_indexes(np.array([0,3]), np.array([1,1])))</code>	[0 1]
<code>print(number_of_inversions(np.array([1,2,3,4,5])))</code>	0
<code>print(number_of_inversions(np.array([5,4,3,2,1])))</code>	10
<code>print(number_of_inversions(np.array([1,3,5,2,4,6])))</code>	3