

problem1 (example)

จงเขียนโปรแกรมเพื่อคำนวณค่า y จากค่า x และสูตรดังต่อไปนี้

$y = 5x^2 - 20x + 2$

ข้อมูลนำเข้า

จำนวนเต็ม **x**

ข้อมูลส่งออก

ค่า **y** ที่ได้จากสูตร

ตัวอย่าง

| input (จากแป้นพิมพ์) | output (ทางจอภาพ) |
|----------------------|-------------------|
| 0                    | 2                 |
| -1                   | 27                |
| 10                   | 302               |

## problem2 example

จงเขียนโปรแกรมเพื่อรับค่าคะแนนและแสดงข้อความตามเงื่อนไขดังนี้

- หากคะแนนมีค่ามากกว่าหรือเท่ากับ 90 ให้แสดงข้อความ **passed with honors**
- หากคะแนนมีค่ามากกว่าหรือเท่ากับ 70 แต่น้อยกว่า 90 ให้แสดงข้อความ **passed with merit**
- หากคะแนนมีค่ามากกว่าหรือเท่ากับ 50 แต่น้อยกว่า 70 ให้แสดงข้อความ **passed**
- หากคะแนนมีค่าน้อยกว่า 50 ให้แสดงข้อความ **failed**

## ข้อมูลนำเข้า

ค่าคะแนนในรูปแบบทศนิยม

## ข้อมูลส่งออก

ข้อความตามเงื่อนไข

## ตัวอย่าง

| input (จากแป้นพิมพ์) | output (ทางจอภาพ)  |
|----------------------|--------------------|
| 100.0                | passed with honors |
| 70                   | passed with merit  |
| 65.25                | passed             |
| 0                    | failed             |
| 50.0                 | passed             |
| 38.8                 | failed             |

## problem3 example

จงเขียนโปรแกรมที่

- รับค่า  $n$  ที่แทนจำนวนที่ต้องการแสดงผล ( $0 \leq n \leq 1000$ )
- รับค่า  $x$  ที่แทนค่าเริ่มต้น ( $-1000 \leq x \leq 1000$ )
- รับค่า  $m$  ที่แทนค่าจำนวนเท่าความแตกต่างของแต่ละจำนวน ( $-100 \leq m \leq 100$ )
- แสดงผลรวมของเลขทั้ง  $n$  ตัว

เช่น หาก  $n$  เป็น 4,  $x$  เป็น 2 และ  $m$  เป็น 3 ก็จะทำให้หมายถึงให้แสดงตัวเลข 4 ตัว เริ่มต้นที่ค่า 2 แล้วเพิ่มขึ้นทีละ 3 เท่าซึ่งก็คือ 2 6 18 54 และผลรวมก็คือ 80

## ข้อมูลนำเข้า

บรรทัดแรกเป็นจำนวนเต็ม  $n$

บรรทัดที่สองเป็นจำนวนเต็ม  $x$

บรรทัดที่สามเป็นจำนวนเต็ม  $m$

## ข้อมูลส่งออก

$n$  บรรทัดแรกเป็นจำนวนที่ให้แสดงผลบรรทัดละตัว

บรรทัดที่  $n+1$  เป็นผลรวมของเลขทั้ง  $n$  ตัว

## ตัวอย่าง

| input (จากแป้นพิมพ์) | output (ทางจอภาพ)                                                      |
|----------------------|------------------------------------------------------------------------|
| 4<br>2<br>3          | 2<br>6<br>18<br>54<br>80                                               |
| 10<br>-2<br>-2       | -2<br>4<br>-8<br>16<br>-32<br>64<br>-128<br>256<br>-512<br>1024<br>682 |

## Seven Up

เกม Seven Up เป็นกิจกรรม ice breaking ง่าย ๆ โดยให้ผู้เล่นนั่งเป็นวงกลม จากนั้นให้นับ 1, 2, 3 ต่อกันไปเรื่อย ๆ เมื่อตัวเลขที่จะพูดหารด้วย 7 ลงตัวหรือมีเลข 7 เช่น 7, 14, 17, 21, 27, 35, 37, ... ให้พูด Seven Up แทน

โจทย์ข้อนี้ให้เขียนฟังก์ชัน 2 ฟังก์ชัน ได้แก่

1. `isSevenUp` รับจำนวนเต็ม 1 ค่าและคืนค่า `True` หากจำนวนเต็มนั้นต้องพูดว่า Seven Up
2. `nextSevenUp` รับจำนวนเต็ม 1 ค่าและคืนค่าจำนวนเต็มตัวถัดไปที่ต้องพูดว่า Seven Up

```
def isSevenUp(x):  
    # write your code here  
    pass  
def nextSevenUp(x):  
    # write your code here  
    pass  
  
exec(input().strip()) # DON'T remove this line
```

## ข้อมูลนำเข้า

คำสั่งภาษา Python ที่ต้องการให้ทำงาน

## ข้อมูลส่งออก

ผลที่ได้จากคำสั่งที่ได้รับ

## ตัวอย่าง

| input (จากแป้นพิมพ์)                 | output (ทางจอภาพ)  |
|--------------------------------------|--------------------|
| <code>print(isSevenUp(14))</code>    | <code>True</code>  |
| <code>print(isSevenUp(72))</code>    | <code>True</code>  |
| <code>print(isSevenUp(95))</code>    | <code>False</code> |
| <code>print(nextSevenUp(60))</code>  | <code>63</code>    |
| <code>print(nextSevenUp(-10))</code> | <code>-7</code>    |

## Flowchart

จงเขียนโปรแกรมที่ทำงานตามผังงานทางขวานี้

### ข้อมูลนำเข้า

จำนวนเต็ม 4 จำนวนบนบรรทัดเดียวกัน คั่นด้วยช่องว่าง

### ข้อมูลส่งออก

ค่าของ **a b c** และ **d** ในผังงาน คั่นด้วยช่องว่างหนึ่งช่อง

### ตัวอย่าง

| input (จากแป้นพิมพ์) | output (ทางจอภาพ) |
|----------------------|-------------------|
| 1 2 3 4              | 9 2 7 4           |
| 4 3 5 6              | 5 4 5 4           |

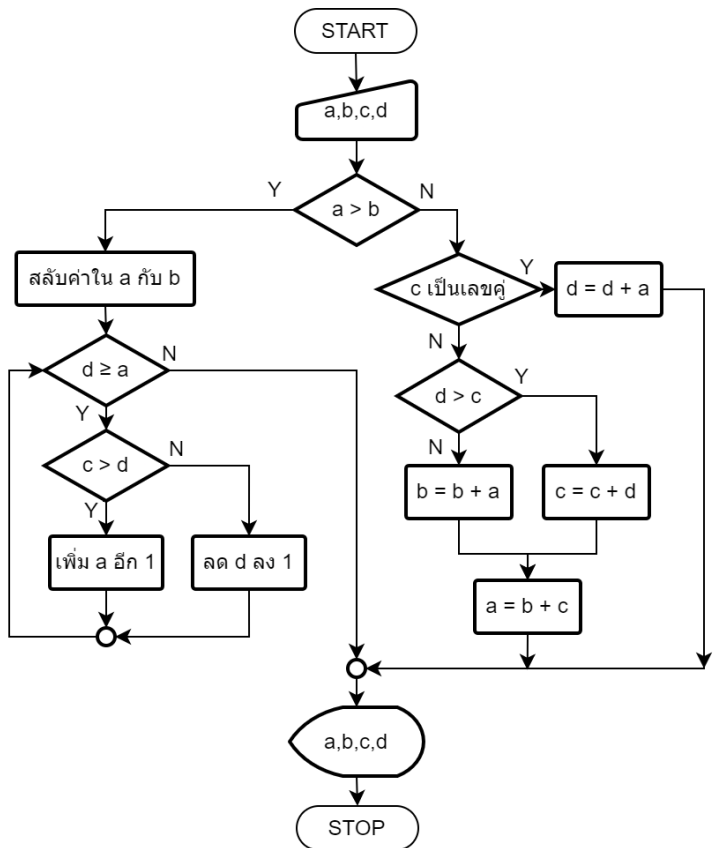
คำเตือน: ตัวอย่างข้างบนนี้ไม่ได้ครอบคลุมทุกกรณีของการทำงานในผังงาน

สำหรับคำสั่ง **input** เข้าเก็บในตัวแปร **a,b,c** และ **d** ในด้านบนผังงาน ให้ใช้คำสั่งข้างล่างนี้

```
a,b,c,d = [int(e) for e in input().split()]
```

สำหรับคำสั่งการแสดงค่าในตัวแปร **a,b,c** และ **d** ให้ใช้คำสั่งข้างล่างนี้

```
print(a,b,c,d)
```



ค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน

ค่าเฉลี่ย ( $\mu$ ) และค่าเบี่ยงเบนมาตรฐาน ( $\sigma$ ) ของข้อมูล  $x_1, x_2, \dots, x_n$  หาได้จากสูตรข้างล่างนี้

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2}$$

ให้เขียนโปรแกรมรับข้อมูลชุดหนึ่ง แล้วแสดงค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของข้อมูลชุดนี้

ข้อมูลนำเข้า

บรรทัดแรกเป็นจำนวนข้อมูล  $n$   
 $n$  บรรทัดต่อมา เป็นข้อมูล (จำนวนจริง) บรรทัดละหนึ่งจำนวน

ข้อมูลส่งออก

ค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของข้อมูลที่รับมาในรูปแบบที่แสดงในตัวอย่าง  
ให้แสดงค่าทั้งสองในรูปแบบที่มีเลขหลังจุดทศนิยมมากที่สุด 5 ตำแหน่ง ด้วยการใชฟังก์ชัน **round** (ค่าที่ต้องการแสดง , 5)

ตัวอย่าง

| input (จากแป้นพิมพ์)                 | output (ทางจอภาพ)          |
|--------------------------------------|----------------------------|
| 4<br>1.5<br>1.5<br>1.5<br>1.5        | average = 1.5 sd = 0.0     |
| 5<br>1.0<br>2.5<br>3.0<br>3.5<br>5.0 | average = 3.0 sd = 1.30384 |

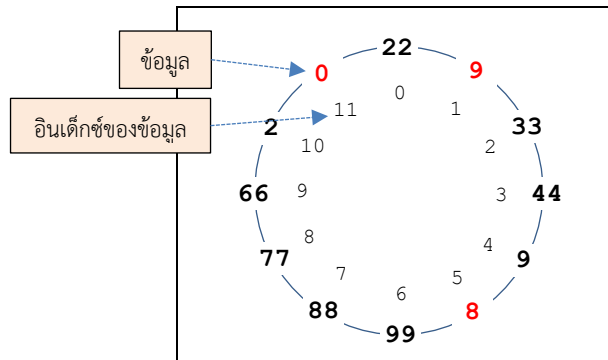
## หาหลุมในลิสต์วงกลม

**นิยาม** "หลุม" ในลิสต์ คือ ข้อมูลในลิสต์ที่มีค่าน้อยกว่าตัวที่ติดกันด้านซ้ายและด้านขวา

เช่น [22, 9, 33, 44, 9, 8, 99, 88, 77, 66, 2, 0] มีหลุมที่อินเด็กซ์ 1 กับ 5

**นิยาม** "ลิสต์วงกลม" คือ ลิสต์ที่เสมือนว่าตัวซ้ายสุดกับตัวขวาสุดของลิสต์นั้นติดกัน

ดังนั้น ถ้าคิดว่า [22, 9, 33, 44, 9, 8, 99, 88, 77, 66, 2, 0] เป็นลิสต์วงกลม จะมีหลุมที่อินเด็กซ์ 1, 5 และ 11



จึงเขียนโปรแกรมที่รับรายการของจำนวนที่แทนลิสต์วงกลม แล้วหาอินเด็กซ์ของหลุมทั้งหมดในลิสต์วงกลมนี้

## ข้อมูลนำเข้า

บรรทัดแรกเป็นจำนวนเต็ม  $n$  ตามด้วยอีก  $n$  บรรทัด บรรทัดละจำนวนแทนข้อมูลในลิสต์วงกลมตามลำดับ เริ่มตั้งแต่อินเด็กซ์ที่ 0 เป็นต้นไป

## ข้อมูลส่งออก

อินเด็กซ์ของหลุมทั้งหมดในลิสต์วงกลมที่ได้รับ เรียงอินเด็กซ์จากน้อยไปมาก อินเด็กซ์ละบรรทัด ถ้าไม่มีหลุมเลยให้แสดง **No pits**

## ตัวอย่าง

| input (จากแป้นพิมพ์)                                                  | output (ทางจอภาพ) |
|-----------------------------------------------------------------------|-------------------|
| 3<br>9<br>9<br>9<br><div>จำนวนข้อมูล</div>                            | No pits           |
| 5<br>1<br>2<br>3<br>2<br>1                                            | No pits           |
| 4<br>11<br>22<br>33<br>44                                             | 0                 |
| 12<br>22<br>9<br>33<br>44<br>9<br>8<br>99<br>88<br>77<br>66<br>2<br>0 | 1<br>5<br>11      |

## APEC Headache

โจได้รับหน้าที่ในการจัดโต๊ะกินเลี้ยงของงานประชุม APEC แต่ด้วยสถานการณ์สงครามในปัจจุบันทำให้การจัดโต๊ะนั้นเต็มไปด้วย

ข้อจำกัดมากมาย จงเขียนโปรแกรมเพื่อเช็คว่าการจัดโต๊ะกินเลี้ยงแต่ละโต๊ะเป็นไปได้หรือไม่

### ข้อจำกัดการจัดโต๊ะ

1. โต๊ะแต่ละโต๊ะเป็นโต๊ะกลม
2. ประเทศที่เป็นคู่สงครามกันไม่สามารถนั่งติดกัน ถ้าประเทศใดมีพันธมิตร ให้ถือว่าพันธมิตรนั้นเป็นคู่สงครามด้วย
3. พันมิตรของศัตรูคือศัตรู
4. ศัตรูของพันธมิตรคือศัตรู
5. ศัตรูของศัตรูไม่จำเป็นต้องเป็นพันธมิตร

### ข้อมูลนำเข้า

ข้อมูลนำเข้าจะรับจากแป้นพิมพ์ที่ละบรรทัดเริ่มต้นข้อมูลกลุ่มพันธมิตร ตามด้วยข้อมูลคู่สงคราม

Ally America England Ukraine France

Ally Russia China

Enemy China America

Enemy France Iran

Enemy Iran Iraq

ในกรณีนี้ จะมีกลุ่มประเทศที่เป็นคู่สงครามกันคือ

{America, England, Ukraine, France} vs {Russia, China} และ {America, England, Ukraine, France} vs {Iran} และ

{Iran} vs {Iraq}

ทั้งนี้ จะไม่มีประเทศใดอยู่สองกลุ่มพันธมิตรพร้อมกัน และจะไม่มีคู่ประเทศที่เป็นพันธมิตรและคู่สงครามในเวลาเดียวกัน

หลังจากข้อมูลคู่พันธมิตรและคู่สงคราม จะใส่ข้อมูลโต๊ะบรรทัดละโต๊ะ โดยเรียงที่นั่งไล่ลำดับไป

Table America England Iceland France Iran Russia Iraq

โต๊ะนี้ไม่สามารถจัดได้ เพราะ Iran นั่งติดกับ France

Table America England Russia

โต๊ะนี้ไม่สามารถจัดได้ เพราะ America นั่งติดกับ Russia

Table America England Thailand Russia China Iran China Japan

โต๊ะนี้สามารถจัดได้

จบโปรแกรมโดยการพิมพ์คำสั่ง End

จะไม่มีอินพุตที่ไม่ได้ขึ้นต้นด้วย Ally, Enemy, Table หรือ End กรณีทดสอบจะมีไม่เกิน 100 บรรทัด

### ข้อมูลส่งออก

ตอบ Okay สำหรับทุกโต๊ะที่จัดได้ ตอบ No สำหรับทุกโต๊ะที่ไม่สามารถจัดได้

| Input (จากแป้นพิมพ์)                                         | Output (ทางจอภาพ) |
|--------------------------------------------------------------|-------------------|
| Ally America England Ukraine France                          | No                |
| Ally Russia China                                            | No                |
| Enemy China America                                          | Okay              |
| Enemy France Iran                                            |                   |
| Enemy Iran Iraq                                              |                   |
| Table America England Iceland France Iran Russia Iraq        |                   |
| Table America England Russia                                 |                   |
| Table America England Thailand Russia China Iran China Japan |                   |
| End                                                          |                   |