

## ตัวอย่างการจัดการสตริง

ทำ s[2] = 'a' ไม่ได้ แต่สร้างใหม่ได้	s = s[:2] + 'a' + s[3:]
ตรวจว่าตัวแปร c เก็บตัวอักษร ตัวเดียวและเป็นสระในภาษาอังกฤษ หรือไม่	if len(c) == 1 and c.lower() in 'aeiou' : # c contains a vowel ...
ตรวจว่าตัวแปร c เก็บตัวอักษร ตัวเดียวและเป็นตัวอักษรภาษาอังกฤษ หรือไม่	if len(c) == 1 and 'a' <= c.lower() <= 'z' : # c contains an English alphabet ...

## รูปแบบการประมวลผลสตริงที่พบบ่อย

ต้องการหิบบัอักขระในสตริงจาก ซ้ายไปขวามาประมวลผลทีละตัว	ต้องการนับว่าสตริง s มีตัวเลขกี่ตัว c = 0 for e in s : if '0' <= e <= '9' : c += 1
ต้องการหิบบัอักขระทีละตัว พร้อมกับ index ของตัวนั้น ๆ	for i in range(len(s)): c = s[i] print(i,c)  หรือ for i,c in enumerate(s): print(i,c)
ต้องการหิบบัอักขระในสตริงจาก ขวามาซ้ายทีละตัว	for e in s[::-1] : ...  หรือ for k in range(-1,-(len(s)+1),-1): e = s[k] ...
ต้องการหิบบัอักขระในสตริงจาก ซ้ายไปขวามาประมวลผลทีละคู่ที่ ติดกัน	ต้องการนับว่าสตริง s มีตัวที่ติดกันเป็นตัวเลขทั้งคู่อยู่กี่คู่ c = 0 for k in range(len(s)-1) : # ต้องการตัวติดกัน จึงวนถึงตัวรองสุดท้าย if '0' <= s[k] <= '9' and '0' <= s[k+1] <= '9' : c += 1

<p>ถ้าต้องการนับว่ามี t ปรากฏอยู่ใน s ก็คือ</p> <pre> c = 0 while t in s :     c += 1 </pre> <p>แบบนี้ผิด ถ้า มี t ใน s เงื่อนไข t in s จะเป็น True ตลอด ก็จะไม่รู้จบ</p>	<pre> c = 0 k = s.find(t) while k != 0 :     c += 1     k = s.find(t,k+1)    # 'aaaaaa' มี 'aaa' 4 ครั้ง </pre> <p>หรือ</p> <pre> c = 0 for i in range(len(s)-len(t)+1) :     if s[i:i+len(t)] == t :         c += 1 </pre>
<p>ใช้สตริงสะสมข้อมูลเพื่อนำมาแสดงทีหลัง</p>	<p>หาจำนวนเฉพาะที่มีค่าน้อยกว่า 30</p> <pre> result = "" for n in range(2,30):     for k in range(2, n) :         if n % k == 0 : break     else:         result += str(n) + ", " print(result[:-2])   ได้ 2, 3, 5, 7, 11, 13, 17, 19, 23, 29 </pre>
<p>ต้องการหีบสตริงย่อยที่อยู่ระหว่างรูปแบบสตริงที่กำหนด</p>	<p>หาสตริงย่อยที่อยู่ระหว่าง "&lt;title&gt;" และ "&lt;/title&gt;" (ให้ถือว่า ถ้าสตริงที่ได้รับมี &lt;title&gt; ก็ต้องมี &lt;/title&gt; ด้วย)</p> <pre> t = input().strip() a = t.find("&lt;title&gt;") if a &gt;= 0 :     j = a+len("&lt;title&gt;")      # a      j      b     b = t.find("&lt;/title&gt;",j)  # ..&lt;title&gt;.....&lt;/title&gt;     print( t[j:b] ) else :     print( 'No title' ) </pre>
<p>ใช้สตริงเป็นที่เก็บข้อมูลเพื่อแปลงจากจำนวนเต็มเป็นสตริง</p>	<p>แปลงเลขเดือนใน m เป็นชื่อย่อเดือน</p> <pre> months = "JANFEBMARAPR MAYJUNJUL AUGSEP OCTNOVDEC" m = int(input()) if 1 &lt;= m &lt;= 12 :     print( months[(m-1)*3:m*3] ) else :     print( 'invalid month number' ) </pre>
<p>ใช้สตริงสองสายเป็นที่เก็บข้อมูลเพื่อการแปลงข้อมูลจากสตริงหนึ่งเป็นอีกสตริงหนึ่ง</p>	<p>หาชื่อย่อสกุลเงิน (3 ตัวอักษร) จากชื่อย่อประเทศ (2 ตัวอักษร)</p> <pre> c = input().strip().lower()    # country code # USA, Thailand, Japan, China, Singapore countries = "usthjp cnsg" currency_codes = "USDTHBJPYCNYSGD" k = countries.find(c) if k &gt;= 0:     k //= 2     print( currency_codes[3*k:3*(k+1)] ) else :     print( 'Not found' ) </pre>

# เรื่องพิศบอย

เครื่องหมายเปิดปิดสตริงไม่ตรงกัน หรือใส่ช่องว่างโดยไม่นำหน้าด้วย \	<pre>s = "String s = String' s = "string" s = "You shouted "Hey!" very loudly yesterday"</pre>
ถ้าต้องการรับสตริงจากแป้นพิมพ์ ควร strip ด้วย เพราะผู้ใช้อาจเผลอเพิ่มช่องว่างทางซ้ายหรือขวาเพิ่มเติม	<p>คำสั่งข้างล่างนี้ ถ้าผู้ใช้ป้อน ok ตามด้วยช่องว่างแล้วกด enter การเปรียบเทียบก็จะไม่เท่า 'ok'</p> <pre>s = input() if s == 'ok' :     print('OK')</pre> <p>จึงควรเขียน s = input().strip() (ยกเว้นกรณีที่เราต้องการคงทุกสิ่งที่ผู้ใช้ป้อนทั้งหมด)</p>
ใช้ index ที่เกินช่วงที่ใช้ได้ของสตริง อย่าลืมนะ index ของสตริง s อยู่ในช่วง 0 ถึง len(s)-1 จากซ้ายไปขวา และ -1 ถึง -len(s) ถอยจากขวามาซ้าย	<pre>s = input().strip() t = "" for i in range(10):     t = s[i] + t</pre> <p># ถ้า len(s) &lt; 10 ก็ผิดแน่</p>
เลข index ที่ใช้กับสตริงไม่ใช่จำนวนเต็ม	<pre>print( s[n/2] )</pre> <p># ผิดเพราะ n/2 ได้จำนวนจริง</p>
เปลี่ยนข้อมูลภายในสตริง	<pre>t = input().strip() t[0],t[-1] = t[-1],t[0]</pre> <p># สลับตัวแรกกับตัวท้ายแบบนี้ผิด</p> <p>ถ้าต้องการสลับตัวแรกกับตัวท้าย ต้องสร้างใหม่</p> <pre>t = t[-1] + t[1:-1] + t[0]</pre>
เขียนสตริงระบุตำแหน่งของแฟ้มข้อมูลที่มีเครื่องหมาย \ แต่ไม่ได้เขียน \ \	<pre>s = "c:\temp\data\input.txt" # ผิด ต้องเป็น s = "c:\\temp\\data\\input.txt"</pre>
นำสตริงไปบวกกับข้อมูลประเภทอื่น	<pre>print("average = " + avg) # ผิด ต้องเป็น print("average = " + str(avg)) หรือ print("average =", avg)</pre>



Problem	Code
<p><u>Input:</u> รับสตริงหนึ่งบรรทัด</p> <p><u>Process:</u> สร้างสตริงใหม่ที่มีอักขระในสตริงที่รับเข้ามาปรากฏซ้ำอีกตัว เช่น รับ 'pypy' จะได้สตริง 'ppyyppyy'</p> <p><u>Output:</u> สตริงผลลัพธ์</p>	
<p><u>Input:</u> รับสตริงหนึ่งบรรทัด</p> <p><u>Process:</u> สร้างสตริงใหม่ที่มีทุกอักขระในสตริงที่รับเข้ามาปรากฏซ้ำอีกตัว แต่ถ้ามีตัวซ้ำติดกันอยู่แล้ว ก็ไม่ต้องทำซ้ำ เช่น รับ 'pythonnaa' จะได้สตริง 'ppyytthhoonnaa'</p> <p><u>Output:</u> สตริงผลลัพธ์</p>	
<p><u>Input:</u> รับสตริงหนึ่งบรรทัด</p> <p><u>Process:</u> ตรวจสอบว่าสตริงนี้เป็น palindrome (ซึ่งคือสตริงที่กลับลำดับแล้วคือสตริงเดิม) หรือไม่</p> <p><u>Output:</u> ถ้าเป็น ก็แสดง Y ถ้าไม่เป็น ก็แสดง N</p>	
<p><u>Input:</u> รับจำนวนเต็มบวกสองจำนวนเก็บใส่ d กับ n</p> <p><u>Process:</u> สร้างสตริงใหม่จากจำนวน d ที่มี n หลัก โดยถ้า d มีไม่ครบ n หลัก ก็ต้องเติมเลข 0 ไว้ทางซ้ายให้ครบ n หลัก แต่ถ้า d มีจำนวนหลัก <math>\geq n</math> หลัก ก็ให้เป็นสตริงของ d เดิม</p> <p><u>Output:</u> สตริงใหม่ที่ต้องการ</p>	

Problem	Code
<p>Input: รับเลขฐานสิบหก 1 หลัก</p> <p>Process: แปลงเป็นจำนวนในฐานสิบ</p> <p>Output: จำนวนในฐานสิบที่แปลงได้</p> <p>Base 16 : 0, 1, ..., 9, A, B, C, D, E, F</p> <p>Base 10 : 0, 1, ..., 9, 10, 11, 12, 13, 14, 15</p>	
<p>Input: รับสตริงมาหนึ่งบรรทัด</p> <p>Process: นับจำนวนตัวเลขที่ปรากฏในสตริงที่รับเข้ามา</p> <p>Output: จำนวนที่นับได้</p>	
<p>Input: รับสตริงมาหนึ่งบรรทัด</p> <p>Process: นับว่ามีตัวอักษรติดกันกี่คู่ที่เป็นสระภาษาอังกฤษ</p> <p>Output: จำนวนที่นับได้</p>	
<p>Input: รับสตริงที่มีแต่เลข 0 กับ 1</p> <p>Process: แปลงสตริงเลขฐานสองที่ได้รับให้เป็นจำนวนเต็มในฐานสิบ (คือ int นั่นเอง) ตัวอย่างเช่น</p> $\text{"01101"} = 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$ $= 13$ <p>Output: จำนวนเต็มที่ได้</p>	

## ตัวอย่างการแก้ไข้ปัญหา

### ตรวจสอบเลขซ้ำ

จงเขียนโปรแกรมที่รับสตริง เพื่อตรวจสอบว่าสตริงนี้มีเลขซ้ำกันหรือไม่ เช่น ถ้ารับ ...102...89...3... แบบนี้ไม่มีเลขซ้ำ แต่ถ้ารับ ...102...89...2... แบบนี้มีเลขซ้ำ (มีเลข 2 ซ้ำ)

#### ► ข้อมูลนำเข้า

รับสตริงหนึ่งบรรทัด

#### ► ข้อมูลส่งออก

ถ้าสตริงที่รับมามีเลขซ้ำ แสดง True แต่ถ้าไม่มีเลขซ้ำ แสดง False

#### ► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
..125.9.0.	False
..125.9.2.	True

### ตัวอย่างการเขียนโปรแกรม

โปรแกรม	คำอธิบาย
<pre>t = '' for e in input().strip() :     if '0' &lt;= e &lt;= '9' :         t += e print(t)</pre>	อาจเริ่มด้วยการลบอักขระที่ไม่ใช่ตัวเลขออกจากสตริงที่รับมาก่อน หลังจากนั้นก็ค่อยหาวิธีตรวจสอบอักขระซ้ำ แต่เราเรียนมาแล้ว ไม่สามารถเปลี่ยนแปลงอักขระในสตริงได้ ดังนั้นต้องใช้วิธีการสร้างสตริงใหม่ เช่น ถ้าจะลบอักขระตัวที่ index 2 ออก ก็ใช้การสร้างใหม่แล้วทับตัวเดิมด้วยคำสั่ง <code>s = s[:2]+s[3:]</code> วิธีนี้ค่อนข้างยุ่ง จึงขอเปลี่ยนเป็นการเลือกเฉพาะตัวเลขในสตริงที่รับมา เพิ่มใส่สตริงใหม่อีกตัวหนึ่ง จะง่ายกว่า โปรแกรมทางซ้าย เริ่มด้วยสตริงว่าง <code>t</code> จากนั้นวนหยาบอักขระออกมาทีละตัวจากสตริงที่รับเข้ามา ถ้าอักขระนั้นเป็นตัวเลข ก็เพิ่มใส่ <code>t</code> สั่ง run, ใส่ 12a34bce5, ได้ผลเป็น 12345, ถูกต้อง
<pre>t = '' for e in input().strip() :     if '0' &lt;= e &lt;= '9' :         t += e b = '' for e in t :     if e in b : break     b += e</pre>	เราได้ <code>t</code> เก็บสตริงที่มีตัวเลขอย่างเดียว ก็มาถึงขั้นตอนตรวจว่า <code>t</code> ไม่มีเลขซ้ำหรือไม่ ? จะทำอย่างไร ? ถ้าเราค่อย ๆ ดูอักขระทีละตัวในสตริง ก็ต้องตรวจว่าอักขระตัวใหม่นี้ปรากฏในอักขระที่ดูผ่านมาหรือไม่ ถ้าเราใช้วงวน <code>for e in t</code> ก็จำตัวที่ดูผ่านมาด้วยการใช้ตัวแปรใหม่ <code>b</code> เริ่มด้วยสตริงว่าง ถ้าตัวใหม่ไม่มีใน <code>b</code> เพิ่มตัวใหม่ต่อท้ายเข้าใน <code>b</code> ดัง code ทางซ้ายนี้ เมื่อใดพบว่า <code>e in b</code> เป็นจริง คือตัวใหม่ซ้ำกับตัวที่ดูผ่านมา ก็ <code>break</code> ออกจากวงวนได้ (คือรู้ว่ามีเลขซ้ำแล้ว)

โปรแกรม	คำอธิบาย
<pre> t = '' for e in input().strip() :     if '0' &lt;= e &lt;= '9' :         t += e b = '' for e in t :     if e in b :         print(True)         break     b += e else:     print(False) </pre>	<p>แต่การ break ออกจากวงวนทันทีที่พบตัวซ้ำนั้น พอออกจากวงวนแล้ว เราจะไม่ทราบว่า การหลุดออกจากวงวนมาจากกรณี break หรือมาจากกรณีที่วนครบทุกอักขระแล้ว ในโจทย์บอกว่า ถ้าพบซ้ำ ให้แสดง True ถ้าไม่ซ้ำเลย ให้แสดง False ดังนั้นควร print(True) เลยเมื่อพบซ้ำ แล้ว break ออกจากวงวน ส่วนการตรวจว่าได้วนครบทุกตัวโดยไม่ break ก็ทำได้โดยเพิ่มคำสั่ง print(False) หลัง else ของ for นั้นแสดงว่าวนครบทุกตัวแล้วไม่พบเลขซ้ำเลย ได้ code ทางซ้าย</p> <p>สั่ง run, ใส่ ..125.9.0., ได้ผลเป็น False, แสดงว่าไม่มีซ้ำ, ถูกต้อง</p> <p>สั่ง run, ใส่ ..125.9.2., ได้ผลเป็น True, แสดงว่ามีซ้ำ, ถูกต้อง</p>
<pre> t = '' for e in input().strip() :     if '0' &lt;= e &lt;= '9' :         t += e for k in range(len(t)) :     if t[k] in t[:k] :         print(True)         break else:     print(False) </pre>	<p>ถ้าคิดอีกนิต พบว่า หากเราวนถึงรอบที่ k การตรวจว่าตัวที่ k ซ้ำกับตัวที่ผ่านมาหรือไม่สามารถดูตัวที่ผ่านมาจากส่วนทางซ้ายของ t ไม่เห็นจำเป็นต้องสร้าง b ในแบบที่ทำมา ด้วยคำสั่ง if t[k] in t[:k] จึงเปลี่ยนจาก for e in t เป็น for k in range(len(t)) ได้ตั้ง code ทางซ้าย</p> <p>สั่ง run, ใส่ ..125.9.0., ได้ผลเป็น False, แสดงว่าไม่มีซ้ำ, ถูกต้อง</p> <p>สั่ง run, ใส่ ..125.9.2., ได้ผลเป็น True, แสดงว่ามีซ้ำ, ถูกต้อง</p>
<pre> t = input().strip() for k in range(len(t)) :     if '0' &lt;= t[k] &lt;= '9' and \         t[k] in t[:k] :         print(True)         break else:     print(False) </pre>	<p>เราสามารถปรับปรุงต่อได้อีก โดยยุบรวมวงวน for สองวงวนเข้าด้วยกัน for แรกจะจัดอักขระที่ไม่ใช่ตัวเลข for หลังตรวจเรื่องตัวซ้ำ เราก็ยุบรวมให้มี for เดียว ดูไล่ไปทีละตัว ถ้าไม่ใช่ตัวเลขก็ข้ามไป ถ้าใช่ก็ตรวจว่าซ้ำกับที่ผ่านมาหรือไม่ในทำนองเดียวกับที่ทำมา ได้ code ทางซ้าย</p> <p>สั่ง run, ใส่ ..125.9.0., ได้ผลเป็น False, แสดงว่าไม่มีซ้ำ, ถูกต้อง</p> <p>สั่ง run, ใส่ ..125.9.2., ได้ผลเป็น True, แสดงว่ามีซ้ำ, ถูกต้อง</p>
<pre> t = input().strip() for k,e in enumerate(t) :     if '0' &lt;= e &lt;= '9' and \         e in t[:k] :         print(True)         break else:     print(False) </pre>	<p>หรือจะเปลี่ยนมาใช้ for k,e in enumerate(t): ก็เป็นแบบที่นิยมกว่า แบบบน for แบบนี้จะได้ทั้ง index (เก็บใน k) และตัวข้อมูล (เก็บใน e) มาใช้งานในวงวน</p> <p>สั่ง run, ใส่ ..125.9.0., ได้ผลเป็น False, แสดงว่าไม่มีซ้ำ, ถูกต้อง</p> <p>สั่ง run, ใส่ ..125.9.2., ได้ผลเป็น True, แสดงว่ามีซ้ำ, ถูกต้อง</p>

## ตัวอย่างโจทย์ปัญหา

### แปลงวันที่

ให้อ่าน วันเดือนปีในรูปแบบ เดือน/วันที่/ปี (mm/dd/yyyy) โดยรับค่าทางแป้นพิมพ์ และแปลงวันที่ที่ได้รับเป็นรูปแบบ วันที่ เดือน ปี (dd MMM yyyy) ค่าของเดือนที่รับมาเป็นตัวเลขจำนวนเต็ม ต้องแปลงให้อยู่ในรูปตัวย่อภาษาอังกฤษของเดือนที่รับเข้ามา

#### ► ข้อมูลนำเข้า

มี 1 บรรทัด เป็นข้อความสตริง แทนวันที่ในรูปแบบ เดือน/วันที่/ปี (mm/dd/yyyy)

#### ► ข้อมูลส่งออก

มี 1 บรรทัด เป็นวันที่ในรูปแบบ วันที่ เดือน ปี (dd MMM yyyy)

#### ► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
08/27/2014	27 AUG 2014
02/01/2018	01 FEB 2018



# นับตัวอักษรพิมพ์ใหญ่

ให้เขียนโปรแกรมอ่านข้อมูลที่เป็นสตริงเข้ามาหนึ่งบรรทัด ประกอบด้วยอักขระอะไรก็ได้ เว้นวรรคก็ได้ จากนั้นให้นับเฉพาะตัวอักษรที่เป็นตัวใหญ่เท่านั้น และแสดงผลลัพธ์ออกมาทางหน้าจอ

## ► ข้อมูลนำเข้า

มี 1 บรรทัด ประกอบด้วยสตริงที่ประกอบไปด้วยตัวอักขระอะไรก็ได้ เว้นวรรคก็ได้

## ► ข้อมูลส่งออก

มี 1 บรรทัด แสดงจำนวนตัวอักษรที่เป็นตัวใหญ่ทั้งหมดที่นับได้จากข้อมูลนำเข้า

## ► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
HeLLo WoRLd	5
PYTHON123	6
yes!	0

# กลุ่มสระ

คำภาษาอังกฤษประกอบด้วยพยัญชนะและสระ (a, e, i, o, u) ขอเรียกสระที่อยู่ติดกันว่าเป็น กลุ่มสระ เช่น beautiful มีกลุ่มสระ eau, i, และ u จึงมีกลุ่มสระ 3 กลุ่ม, vowel มีกลุ่มสระ 2 กลุ่ม, group มีกลุ่มสระ 1 กลุ่ม และ rhythm มีกลุ่มสระ 0 กลุ่ม

ให้เขียนโปรแกรมเพื่อับจำนวนกลุ่มสระในคำภาษาอังกฤษ

## ► ข้อมูลนำเข้า

มี 1 บรรทัด แทนคำภาษาอังกฤษ ประกอบด้วยอักขรตัวพิมพ์เล็กเท่านั้น

## ► ข้อมูลส่งออก

มี 1 บรรทัด แสดงจำนวนกลุ่มสระของคำที่กำหนด

## ► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
beautiful	3
vowel	2
group	1
rhythm	0



วรวิธ สัตยวินิจ (Product Manager)  
ภัทรารุณ ชื้อสัตยาศิลป์ (CTO)  
ยอด ชินสุภักกุล (CEO)  
ศุภฤกษ์ กฤตยาเกียรติ (Software Architect)  
เอกลักษณ์ วิริยะโกวิทยา (COO)

(จากซ้ายไปขวา)

Intania 84

Wongnai

ไม่ว่าคุณจะเรียนคณะอะไรหรือภาควิชาอะไร **Computer Programming** ได้กลายมาเป็นความรู้ที่ทุกคนจำเป็นต้องมีไปแล้ว เพราะเทคโนโลยีเกิดใหม่ในธุรกิจและอุตสาหกรรมต่าง ๆ ไม่ว่าจะเป็นทางด้านการค้าขาย ธนาคาร การแพทย์ โฆษณา บ้านเหิง กีฬา เครื่องจักรและหุ่นยนต์ ฯลฯ ล้วนแล้วแต่ต้องใช้ความรู้ด้าน **Computer Programming** มาช่วยพัฒนาผลิตภัณฑ์หรือบริการให้ดีขึ้นกว่าเดิม โดยความต้องการนี้จะยิ่งมากขึ้นมากในอนาคต เมื่อผลิตภัณฑ์ บริการ หรือแม้กระทั่งเงินที่เราใช้ซื้อสินค้าต่าง ๆ ถูกแปลงไปอยู่ในรูปแบบดิจิทัล จึงเป็นเรื่องสำคัญมากที่เราจะใส่ใจกับความรู้ด้าน **Computer Programming** เพราะมันคือ “ทักษะที่จำเป็น” ของคนที่จะสามารถรับมือกับความเปลี่ยนแปลงของโลกดิจิทัล และพร้อมเก็บเกี่ยวโอกาสที่จะมาพร้อมกับคลื่นยักษ์ลูกใหม่ในอนาคตอันใกล้

## สรุปเนื้อหา

การอ่านข้อมูลเก็บเป็นไฟล์ข้อมูลเข้ามาประมวลผลทำได้หลายวิธีหลายคำสั่ง วิชานี้ใช้วิธีง่ายสุด ๆ คือ อ่านเพิ่มข้อความ (text file) ทีละบรรทัดมาเก็บในสตริง มีรูปแบบดังนี้

<p>อ่านเพิ่มข้อความทีละบรรทัด ใช้ <code>readline</code></p> <pre>infile = open(filename, 'r') first_line = infile.readline() second_line = infile.readline() for line in infile :     # คำสั่ง for อ่านจาก infile รอบละหนึ่งบรรทัด     # มาเก็บเป็นสตริงใน line จนหมดเพิ่ม     ... infile.close()      # ไม่อ่านแล้ว ก็ปิดเพิ่ม</pre> <p>ถ้าบรรทัดที่อ่านเข้ามาเป็นบรรทัดว่าง ๆ เก็บใน <code>t</code> จะได้ <code>t = '\n'</code> หรือ <code>len(t)</code> เป็น 1 แต่ถ้าอ่านตอนที่เพิ่มไม่มีข้อมูลให้อ่านแล้ว จะได้ <code>t = ''</code> หรือ <code>len(t)</code> เป็น 0</p> <pre>for line in infile :     ... เหมือนกับ line = infile.readline() while len(line) &gt; 0 :     ...     line = infile.readline()</pre>	<p>อ่านข้อมูลในเพิ่มมาหาค่าเฉลี่ย แต่ละบรรทัดเก็บรหัสกับคะแนน</p> <pre>infile = open('c:/temp/data.txt', 'r') n = 0 s = 0 for line in infile :     sid, sc = line.strip().split()     s += float(sc)     n += 1 infile.close() print('average =', (s/n))</pre> <p>อีกแบบ อ่านข้อมูลในเพิ่มมาหาค่าเฉลี่ย แต่ละบรรทัดเก็บรหัสกับคะแนน บรรทัดแรกบอกจำนวนบรรทัดที่ต้องอ่าน</p> <pre>infile = open('c:/temp/data.txt', 'r') n = int(infile.readline()) # อ่านบรรทัดแรก s = 0 for k in range(n) :     sid, sc = infile.readline().strip().split()     s += float(sc) infile.close() print('average =', (s/n))</pre>
<p>การอ่านจากเพิ่มหนึ่งบรรทัด อาจมีรหัสขึ้นบรรทัดใหม่ <code>\n</code> อยู่ทีปลายบรรทัด (กรณีอ่านบรรทัดสุดท้ายของเพิ่มอาจไม่มีรหัส)</p> <p>ถ้าไม่ต้องการรหัส <code>\n</code> ก็อย่าลืมลบทิ้ง แต่ถ้าอ่านมาแล้วมีการ <code>strip()</code> รหัส <code>\n</code> จะถูกขจัดไปด้วย ไม่ต้องทำอะไรเพิ่ม</p>	<pre>f = open('data.txt', 'r') t = f.readline() if len(t)&gt;0 and t[-1]=='\n' :     t = t[:-1]  for line in f :     if line[-1] == '\n' :         line = line[:-1]     ...</pre>

บันทึกข้อมูลลงแฟ้มข้อความ

```
outfile = open('c:/temp/out.txt','w')
outfile.write(any_string)
outfile.write(any_string + '\n')
...
outfile.close()
```

อย่าลืม : write ไม่เพิ่ม '\n'  
เพื่อขึ้นบรรทัดใหม่ ให้ต้องเพิ่มเองเมื่อต้องการ

หาจำนวนเฉพาะที่มีค่าน้อยกว่า m บันทึกลงแฟ้ม  
บรรทัดละ 5 ตัว

```
m = int(input())
outfile = open('D:/primes.txt','w')
c = 0
result = ""
for n in range(2,m):
    for k in range(2,n):
        if n % k == 0: break
    else:
        result += str(n) + ", "
        c += 1
        if c % 5 == 0:
            outfile.write(result[:-2] + '\n')
            result = ""
if len(result) > 0:
    outfile.write(result[:-2] + '\n')
outfile.close()
```

## เรื่องพิศบอย

เมื่ออ่านข้อมูลจากแฟ้มมาหนึ่งบรรทัด  
ลึ้มลบริหัส \n ทำให้การประมวลผล  
ผิดพลาด

```
t = input().strip()
f = open('names.txt', 'r')
for line in f :
    if line == t :      # ผิด เพราะ line อาจมีรหัส \n
        print(t, ': found in names.txt')
        break
else :
    print(t, ': not found')
```

ชื่อแฟ้มมีเครื่องหมาย \ แต่ใส่แค่ตัวเดียว

```
of = open('c:\temp\data.txt', 'w')
ผิด เพราะ \t คือ tab ต้องเป็น
of = open('c:\\temp\\data.txt', 'w')
หรือใช้ / ก็ได้ เพราะระบบรับชื่อแฟ้มที่เขียนแบบ / ได้
(ไม่ได้หมายความว่า \ เหมือนกับ /)
of = open('c:/temp/data.txt', 'w')
```

เปิดแฟ้มที่ไม่มีอยู่ในเครื่องมาอ่าน

```
infile = open('h:\\file.data')
ถ้าไม่มีแฟ้ม h:/file.data ในเครื่อง ก็จะมี
fn = input()
infile = open(fn)
อาจผิดได้ ถ้า fn มี blank หน้าหรือหลัง จึงควร strip ก่อน
infile = open( input().strip() )
```

พิมพ์ readline เป็น readlines readlines อ่านทีละบรรทัดเพิ่ม (อ่าน เสร็จแล้วได้เป็น list of strings ที่ ยังไม่ได้นำเสนอ จึงขอไม่ลงในรายละเอียด)	infile = open('c:/temp/data.txt') first_line = infile.readlines() # ระวัง second_line = infile.readline() # ควรใช้แบบบรรทัดนี้
---	--



Problem	Code								
<p>Input: หนึ่งบรรทัดเป็นชื่อแฟ้ม</p> <p>Process: อ่านข้อความในแฟ้มมากลับลำดับบรรทัด</p> <p>Output: แสดงข้อความในแฟ้มแบบกลับลำดับบรรทัด ออกทางจอภาพ เช่น</p> <table border="1"> <thead> <tr> <th>ข้อมูลในแฟ้ม</th><th>ผลลัพธ์ (ออกทางจอ)</th></tr> </thead> <tbody> <tr> <td>line1</td><td>line3</td></tr> <tr> <td>line2</td><td>line2</td></tr> <tr> <td>line3</td><td>line1</td></tr> </tbody> </table>	ข้อมูลในแฟ้ม	ผลลัพธ์ (ออกทางจอ)	line1	line3	line2	line2	line3	line1	
ข้อมูลในแฟ้ม	ผลลัพธ์ (ออกทางจอ)								
line1	line3								
line2	line2								
line3	line1								
<p>Input: หนึ่งบรรทัดเป็นชื่อแฟ้ม</p> <p>Process: อ่านข้อความในแฟ้มแล้วกลับลำดับบรรทัด แต่มีเงื่อนไขว่าจะไม่เอาบรรทัดที่ว่าง ๆ หรือมีแต่ blank</p> <p>Output: บันทึกข้อความแบบกลับลำดับบรรทัดลงแฟ้ม ชื่อ reverse.txt</p>									
<p>Input: หนึ่งบรรทัดเป็นชื่อแฟ้ม</p> <p>Process: แสดงหัวข้อข่าวทั้งหมดในแฟ้ม หัวข้อข่าวเป็น ข้อความที่อยู่ระหว่าง &lt;headline&gt; กับ &lt;/headline&gt; ในแฟ้มนี้ (ทั้ง &lt;headline&gt; กับ &lt;/headline&gt; อยู่ในบรรทัดเดียวกันแน่ ๆ และ แต่ละบรรทัดมีไม่เกิน 1 หัวข้อข่าว)</p> <p>Output: บรรทัดละหนึ่งหัวข้อออกทางจอภาพ ให้ครบทุกหัวข้อ</p>									

Problem	Code
<p>Input: สองบรรทัด แต่ละบรรทัดเป็นชื่อแฟ้ม</p> <p>Process: เปรียบเทียบว่าสองแฟ้มนี้มีค่าเหมือนกันหรือไม่</p> <p>Output: ถ้าแฟ้มทั้งสองมีข้อมูลเหมือนกัน แสดง True ต่างก็แสดง False</p>	

## ตัวอย่างการแก้ไข้ปัญหา

### ตรวจสอบคำตอบ

จึงเขียนโปรแกรมอ่านแฟ้มเก็บคำตอบแบบปรนัยของนักเรียน มาตรวจให้คะแนน

#### ► ข้อมูลนำเข้า

ข้อมูลจากแฟ้ม c:\t\answers.txt รูปแบบแฟ้มเป็นดังนี้

บรรทัดแรกเก็บเฉลย เป็นสตริงของตัวอักษร A, B, C หรือ D

บรรทัดต่อมาจนหมดแฟ้ม แต่ละบรรทัด เก็บเลขประจำตัวนักเรียน ตามด้วยช่องว่างหนึ่งช่อง ตามด้วยคำตอบแบบปรนัย ซึ่งเป็นสตริงของตัวอักษร A,B,C,D หรือเป็นช่องว่าง (กรณีไม่ตอบข้อนั้น) หรือเป็นตัวอื่น (กรณีกรอกมากกว่าหนึ่งคำตอบ)

#### ► ข้อมูลส่งออก

แฟ้มใหม่ c:\t\results.txt เก็บผลการตรวจ แต่ละบรรทัดประกอบด้วยเลขประจำตัวนักเรียนตามด้วยคะแนนที่ได้ คั่นด้วยเครื่องหมายจุลภาค (comma)

#### ► ตัวอย่าง

Input (อ่านจากแฟ้ม)	Output (บันทึกลงแฟ้ม)
AABBCCCBCCDDABABDDCCDDCC	5630120421,25
5630120421 AABBCCCBCCDDABABDDCCDDCA	5631010121,14
5631010121 A BB CD BDBAAABA ABDBCCCDCC	563102121,16
563102121 ABABCCNNAADDDABAB CCDDAC	5630121821,24
5630121821 AABBCCCBCCDDABABDDCCDD	

## ตัวอย่างการเขียนโปรแกรม

โจทย์กำหนดให้อ่านข้อมูลจากแฟ้ม และบันทึกผลการทำงานลงแฟ้ม เพื่อให้การเขียนโปรแกรมและหาที่ผิดได้ง่ายขึ้น จะขอเขียนแบบแสดงผลออกหน้าจอก่อน เมื่อทบทวนถูกต้อง ค่อยเปลี่ยนให้บันทึกลงแฟ้ม

โปรแกรม	คำอธิบาย
<pre> fin = open('c:\\t\\answers.txt', 'r') soln = fin.readline().strip() for line in fin:     sid,ans = line.strip().split()     point = 0     for k in range(len(soln)):         if ans[k] == soln[k]:             point += 1     print(sid, point) fin.close() </pre>	<p>เริ่มด้วยการเปิดแฟ้ม, ใช้ <code>readline</code> หลังเปิดแฟ้มทันทีจะได้บรรทัดแรกของแฟ้มซึ่งเป็นเฉลย (<code>strip</code> เพื่อลบช่องว่างซ้ายขวาและรหัสขึ้นบรรทัดใหม่) จากนั้นใช้ <code>for</code> อ่านบรรทัดที่เหลือ ใช้ <code>split</code> แยกเลขประจำตัวกับคำตอบออกจากกัน แล้วใช้อีก <code>for</code> นำคำตอบทีละตัวเปรียบเทียบกับเฉลย (เราใช้ <code>for</code> แบบเปลี่ยนค่า <code>k</code> เป็น index ของทั้งคำตอบกับเฉลย) ถ้าตรงกันก็เพิ่มคะแนน วนตรวจครบทุกข้อก็แสดงผลทางจอภาพ วนครบทุกบรรทัดก็ปิดแฟ้ม ดัง code ทางซ้าย ก่อนสั่งทำงาน ก็ต้องสร้างแฟ้ม <code>answers.txt</code> จากนั้นสั่ง <code>run</code>, ได้</p> <p>OSError: [Errno 22] Invalid argument: 'c:\\t\\x07nswers.txt'</p> <p>เกิดอะไรแปลก ๆ กับชื่อแฟ้ม</p>
<pre> fin = open('c:\\t\\answers.txt','r') soln = fin.readline().strip() for line in fin:     sid,ans = line.strip().split()     point = 0     for k in range(len(soln)):         if ans[k] == soln[k]:             point += 1     print(sid, point) fin.close() </pre>	<p>ถ้ายังจำได้ เครื่องหมาย \ ที่ปรากฏในสตริงจะถูกตีความหลายแบบ ถ้าต้องการสัญลักษณ์ \ ในสตริง ต้องเขียน \\ แก่ให้ถูกต้อง</p> <p>สั่ง <code>run</code>, ได้ 5630120421 25, ผิดบรรทัดที่ 4 ของโปรแกรม</p> <p>ValueError: too many values to unpack (expected 2)</p> <p>แปลว่าหลัง <code>split</code> แล้ว ได้สตริงมากกว่าตัวแปรที่จะมารับผล ซึ่งน่าจะเกิดหลังอ่านบรรทัดที่ 3 ของแฟ้ม</p> <p>5631010121 A BB CD BDBAAABA ABDBCCCDCC</p> <p>บรรทัดนี้มีช่องว่างในคำตอบ (ช่องว่างแปลว่าข้อนั้นไม่ตอบ) จึงทำให้ <code>split</code> แล้วได้สตริงมากกว่า 2 สตริง จึงต้องหาทางแยกบรรทัดให้เป็นสองสตริง เลขประจำตัว กับ คำตอบทั้งหลาย</p>



โปรแกรม	อธิบาย
<pre> fin = open('c:\\t\\answers.txt', 'r') soln = fin.readline().strip() for line in fin:     line = line.strip()     j = line.find(' ')     if j &gt; 0 :         sid = line[:j]         ans = line[j:]         point = 0         for k in range(len(soln)):             if ans[k] == soln[k] :                 point += 1         print(sid, point) fin.close() </pre>	<p>เราไม่ควรแก้ปัญหานี้โดยคิดว่าเลขประจำตัวมี 10 หลัก ก็แยกด้วย</p> <pre> sid = line[:10] ans = line[10:] </pre> <p>เพราะถ้าดูในแฟ้ม พบว่าเลขประจำตัวบางคนมีน้อยกว่า 10 หลัก จึงควรใช้วิธีหาช่องว่างแรกจากทางซ้ายด้วยบริการ find ของสตริง (แต่ต้องอย่าลืม strip ก่อน ไม่เช่นนั้น ถ้า line เริ่มต้นด้วยช่องว่าง ก็จะผิด) เมื่อหาช่องว่างพบที่ index j ก็แยกได้ด้วย</p> <pre> sid = line[:j] ans = line[j:] </pre> <p>แล้วก็ตรวจสอบคำตอบด้วยวิธีที่ทำมา สั่ง run, ได้</p> <pre> 5630120421 12 5631010121 7 563102121 6 </pre> <p>แล้วเกิดข้อผิดพลาดที่คำสั่ง if ans[k] == soln[k] IndexError: string index out of range</p> <p>ถ้าดูผลคะแนนที่ได้ก่อนเกิดข้อผิดพลาด จะพบว่าได้คะแนนรวมผิดด้วย</p>
<pre> fin = open('c:\\t\\answers.txt', 'r') soln = fin.readline().strip() for line in fin:     line = line.strip()     j = line.find(' ')     if j &gt; 0 :         sid = line[:j]         ans = line[j:].strip()         point = 0         for k in range(len(soln)):             if ans[k] == soln[k] :                 point += 1         print(sid, point) fin.close() </pre>	<p>ผลที่ผิด ได้คะแนนลดลงมาก เหมือนกับว่า คำตอบผิดมีมาก ผิดปกติ ถ้าแทรกคำสั่ง print(ans) กับ print(soln) ออกมาดูเทียบกัน จะได้ (ขอตัดมาให้ดูแค่ 2 บรรทัดแรก)</p> <pre> AABBCCCBCCDDDABABDDCCDDCA AABBCCCBCCDDDABABDDCCDDCC </pre> <p>เห็นได้ว่าบรรทัดบน ans มันเลื่อนไปทางขวา เพราะว่ามีช่องว่างทางซ้าย แก้ปัญหานี้ด้วย</p> <pre> ans = line[j:].strip() </pre> <p>เพื่อตัดช่องว่างออก</p> <p>สั่ง run, ได้</p> <pre> 5630120421 25 5631010121 14 </pre> <p>แล้วเกิดข้อผิดพลาดที่คำสั่ง</p> <pre> if ans[k] == soln[k] </pre> <p>IndexError: string index out of range</p> <p>ทำงานผิดเหมือนครั้งที่แล้ว แต่ได้คะแนนรวมถูกต้อง</p> <p>ส่วนที่ทำงานผิดพลาดกลายมาเกิดกับบรรทัดที่ 4 ในแฟ้ม ซึ่งมีข้อมูล</p> <pre> 563102121 ABABCCNNAADDDABAB CCDDAC </pre>

โปรแกรม	คำอธิบาย
<pre> fin = open('c:\\t\\answers.txt', 'r') soln = fin.readline().strip() for line in fin:     line = line.strip()     j = line.find(' ')     if j &gt; 0 :         sid = line[:j]         ans = line[j+1:]         point = 0         for k in range(len(soln)):             if ans[k] == soln[k] :                 point += 1         print(sid, point) fin.close() </pre>	<p>เนื่องจากโจทย์กำหนดว่าเลขประจำตัวกับคำตอบขึ้นด้วยช่องว่าง 1 ช่อง แต่ในบรรทัดที่ 4 ห่างกัน 2 ช่อง แสดงว่าช่องว่างตัวที่ 2 นั้นแทนคำตอบ (ที่ไม่มี) ของข้อที่ 1 คำสั่ง <code>line[j:].strip()</code> จะลบช่องว่างออกหมด ทำให้ประมวลผลผิด ทำให้มีข้อมูลไม่ครบ และทำให้เมื่อนำ <code>ans[k]</code> มาเทียบคำตอบก็ผิด เพราะ <code>k</code> มีค่าเกินช่องในสตริง จึงแก้ไขคำสั่ง <code>line[j:].strip()</code> เป็น <code>line[j+1:]</code> สั่ง run, ได้</p> <pre> 5630120421 25 5631010121 14 563102121 16 </pre> <p>แล้วเกิดข้อผิดพลาดที่คำสั่ง</p> <pre> if ans[k] == soln[k] </pre> <p><code>IndexError: string index out of range</code> ไม่ผิดบรรทัดที่ 4 ของแฟ้มข้อมูลแล้ว แต่ผิดบรรทัดที่ 5 ของแฟ้ม</p>
<pre> fin = open('c:\\t\\answers.txt', 'r') soln = fin.readline().strip() for line in fin:     line = line.strip()     j = line.find(' ')     if j &gt; 0 :         sid = line[:j]         ans = line[j+1:]         if len(ans) &lt; len(soln) :             ans += ' '*(len(soln)-len(ans))         point = 0         for k in range(len(soln)):             if ans[k] == soln[k] :                 point += 1         print(sid, point) fin.close() </pre>	<p><code>index out of range</code> แปลว่าค่าของ <code>index</code> อยู่นอกช่วงที่ถูกต้อง คำสั่ง <code>ans[k] == soln[k]</code> ผิดได้ที่ <code>ans[k]</code> หรือไม่ก็ <code>soln[k]</code> คำสั่ง <code>soln[k]</code> ไม่น่าผิด เพราะ <code>k</code> มีค่าใน <code>range(len(soln))</code> แต่ <code>ans[k]</code> อาจผิดได้ถ้า <code>ans</code> มีขนาดน้อยกว่า <code>soln</code> ถ้ากลับไปดูที่บรรทัดที่ 5 ในแฟ้มพบว่าคำตอบมีไม่ครบ นักเรียนไม่ตอบคำตอบท้าย ๆ จะเป็นช่องว่าง และถูก <code>strip</code> ทิ้ง</p> <p>วิธีแก้ไข ก็แค่ตรวจว่า ถ้า <code>len(ans) &lt; len(soln)</code> จะเติมช่องว่างต่อทางขวาของ <code>ans</code> เป็นจำนวนเท่ากับผลต่างของความยาวทั้งสองสตริง</p> <p>สั่ง run, ได้</p> <pre> 5630120421 25 5631010121 14 563102121 16 5630121821 24 </pre> <p>ถูกต้อง ภาระที่เหลือก็แค่เปลี่ยนจากการแสดงผลออกหน้าจอเป็นการบันทึกลงแฟ้ม ซึ่งขอให้ผู้อ่านลองเขียนต่อเอง</p>

# ตัวอย่างโจทย์ปัญหา

## คะแนนเฉลี่ยของนักเรียน

จงเขียนโปรแกรมเพื่ออ่านแฟ้ม data.txt แฟ้มนี้เก็บข้อมูล  
คะแนนของนิสิตโดยมีรูปแบบ id:name:section:score

```
5913842721:Somsak Rakrian:1:56.6
5913845921:Somsri Deeying:2:78.0
5913856821:Rakchard Yingcheep:2:89.0
5913861321:Thumdee Tong Daidee:2:99
591387721:Somrak Rakrian:10:84.25
```

จากนั้นรับข้อมูลจากแป้นพิมพ์ เป็นตอนเรียนที่ต้องการหาค่าคะแนนเฉลี่ย  
หากไม่พบนิสิตในตอนเรียนนั้น ให้พิมพ์ Not Found

### ► ข้อมูลนำเข้า

จำนวนเต็มหนึ่งจำนวน เป็นตอนเรียนที่ต้องการหาค่าคะแนนเฉลี่ย

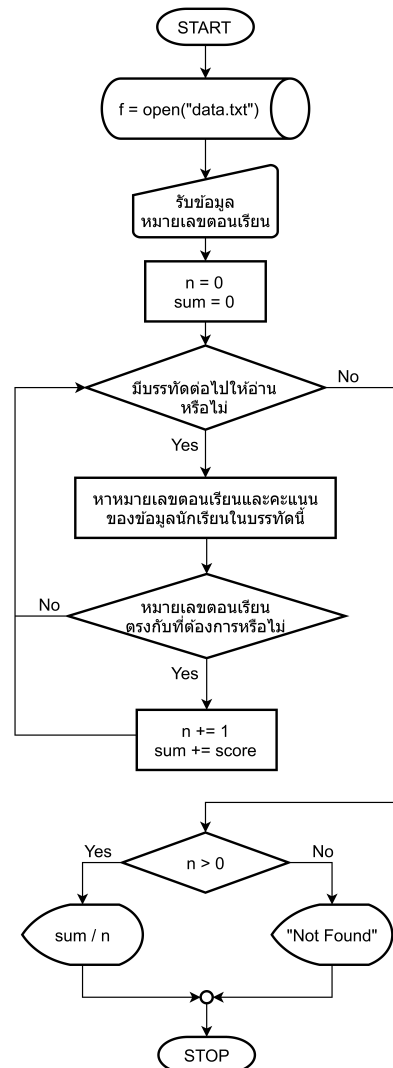
### ► ข้อมูลส่งออก

แสดงคะแนนเฉลี่ยของนิสิตในตอนเรียนที่ต้องการ

### ► ตัวอย่าง

สมมติให้แฟ้ม data.txt มีข้อความข้างต้น

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
1	56.6
2	88.66666666666667
3	Not Found



# Find Student's Grade

ให้เขียนโปรแกรมเพื่ออ่านแฟ้ม `score.txt` ซึ่งมีรหัสனிสิตและเกรดของนิสิตแต่ละคน (0-4) คั่นด้วยช่องว่าง และ  
รับค่ารหัสนิสิตจากแป้นพิมพ์ แล้วแสดงเกรดของนิสิตคนนั้น หากไม่พบรหัสนิสิตในแฟ้ม ให้แสดง `Not Found`

## ► ข้อมูลนำเข้า

มี 1 บรรทัด รับรหัสนิสิตเป็นจำนวนเต็ม

## ► ข้อมูลส่งออก

มี 1 บรรทัด แสดงเกรดของนิสิตเป็นจำนวนเต็ม หากไม่พบรหัสนิสิตในแฟ้ม ให้แสดงคำว่า `Not Found`

## ► ตัวอย่าง

ข้อมูลในแฟ้ม <code>score.txt</code>	Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
1111 3 12345 2 9999 4 89431 0 76238 1	12345	2
	9999	4
	1234	Not Found

# รหัสลับ

จงเขียนโปรแกรมอ่านข้อมูลรหัสลับจากแฟ้มหนึ่ง ซึ่งระบุว่า จะต้องใช้รหัสตามลำดับอย่างไร จึงจะสามารถปล่อยอาวุธอานุภาพรุนแรงได้ วิธีการถอดรหัสจากแฟ้มคือ ต้องรับอินพุตเป็นตัวอักษรสามตัว จากนั้นนับว่า ตัวอักษรทั้งสามตัวนั้น ปรากฏเป็นจำนวนเท่าใดในแฟ้ม แล้วจึงเรียงลำดับตัวอักษรทั้งสามตามจำนวนครั้งที่ปรากฏในแฟ้มจากมากที่น้อยที่สุด โดยถือว่าตัวอักษรตัวพิมพ์ใหญ่กับตัวพิมพ์เล็ก ไม่เหมือนกัน (case sensitive) และจำนวนตัวอักษรทั้งสามตัวนั้นจะไม่เท่ากัน ตัวอย่างเช่น หากมีแฟ้ม data.txt เป็นอินพุตดังต่อไปนี้

```
agAbggggDf
ffgFFFaA
DaADDFF
FFDFFF
```

จะเห็นว่า ตัวอักษร a มี 4 ตัว, A มี 2 ตัว, b มี 1 ตัว, D มี 6 ตัว, f มี 3 ตัว, F มี 10 ตัว และ g มี 6 ตัว

หากอินพุตเป็น a b f จะได้ผลลัพธ์เป็น afb

หากอินพุตเป็น F A f จะได้ผลลัพธ์เป็น FfA

## ► ข้อมูลนำเข้า

ข้อความสี่บรรทัด บรรทัดแรกแทนชื่อแฟ้ม อีกสามบรรทัดถัดมาแทนตัวอักษรสามตัว บรรทัดละหนึ่งตัว

## ► ข้อมูลส่งออก

แสดงข้อความผลลัพธ์ตามต้องการ

## ► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
data.txt a b f	Afb
data.txt F A f	FfA

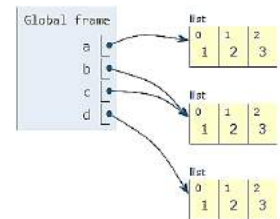
## 06.1 : List

### สรุปเนื้อหา

รายการหรือลิสต์ (list) เป็นที่เก็บกลุ่มของข้อมูลที่มีลำดับ แต่ละตัวมีเลข index กำกับลำดับ ตัวซ้ายสุดมี index 0

- การสร้าง list

- o `x = []` หรือ `x = list()` ได้ลิสต์ว่าง ๆ มี `len(x)` เป็น 0
- o `x = [0]*10` ได้ `x = [0,0,0,0,0,0,0,0,0,0]`
- o `x = list(a)` ได้ลิสต์ที่มีข้อมูลตามที่หยิบออกมาจาก a (ด้วย `for e in a`)  
`x = list('abcde')` ได้ `x = ['a', 'b', 'c', 'd', 'e']`  
`x = list(range(1,10,2))` ได้ `x = [1,3,5,7,9]`
- o `a = [1,2,3]; b = [1,2,3]; c = b; d = list(b)`  
ได้ผลดังรูปขวา ที่น่าสนใจคือ คำสั่ง `c = b` ทำให้ `c` กับ `b` เป็นลิสต์เดียวกัน



- ใช้ `+` เพื่อต่อ list และ `*` เพื่อ + หลาย ๆ ครั้ง

`x = 2*([1,2,3] + [3,4,5])` ได้ `x = [1,2,3,3,4,5,1,2,3,3,4,5]`

- `x.append(e)` เพิ่ม `e` ต่อท้าย (ทางขวา) ของลิสต์ `x`
- `x.insert(i, e)` แทรก `e` ไว้ที่ index `i` ของลิสต์ `x`
- `x.pop(i)` ลบข้อมูลตัวที่ index `i` ของลิสต์ `x` และคืนข้อมูลที่ถูกลบเป็นผลลัพธ์
- `x.sort()` ทำให้ข้อมูลในลิสต์ `x` เรียงจากน้อยไปมาก คำสั่งนี้ไม่มีผลคืนกลับมา
- `sorted(x)` คืนลิสต์ที่มีค่าเหมือนกับที่แฉงออกจาก `x` แต่เรียงลำดับข้อมูลจากน้อยไปมากให้เรียบร้อย (`x` ไม่เปลี่ยนแปลง)
- `sum(x)` คืนผลรวมของจำนวนในลิสต์ `x`
- `max(x)` คืนค่ามากสุดในลิสต์ `x`, `min(x)` คืนค่าน้อยสุดในลิสต์ `x`
- `x.count(e)` คืนจำนวนครั้งที่ `e` ปรากฏในลิสต์ `x`
- `if e in x` ใช้ตรวจสอบว่ามี `e` ในลิสต์ `x` หรือไม่
- `x.index(e)` คืน index น้อยสุดที่พบ `e` ในลิสต์ `x` ถ้าไม่พบจะทำงานผิดพลาด จึงต้องตรวจสอบ  
`if e in x :`  
    `k = x.index(e)`  
    `...`  
`else :`  
    `...`

หมายเหตุ : ลิสต์ไม่มี `find` เหมือนกับของสตริง

- บริการ `split` กับ `join` ไม่ใช่บริการของลิสต์ แต่เป็นของสตริง ที่เกี่ยวข้องกับลิสต์
  - o `t.split()` คืนลิสต์ของสตริงย่อยที่แยกออกจากสตริง `t` โดยใช้ช่องว่างเป็นตัวคั่นสตริงย่อย
  - o `t.split(s)` คืนลิสต์ของสตริงย่อยที่แยกออกจากสตริง `t` โดยใช้สตริง `s` เป็นตัวคั่นสตริงย่อย
  - o `s.join(x)` คืนสตริงที่ได้จากการนำสตริงในลิสต์ `x` มาต่อกันคั่นด้วยสตริง `s`

ตัวอย่างการเข้าใช้ข้อมูลและลิสต์ย่อยในลิสต์ (สมมติให้  $x = [11, 12, 13, 14, 15]$ )

- มีวิธีการเข้าใช้และการเลือกลิสต์ย่อยด้วย index เหมือนกับวิธีของสตริง
  - ใช้ `for e in x` : ในการแจกแจงข้อมูลในลิสต์  $x$  จากซ้ายไปขวาออกมาเก็บใน  $e$  เพื่อนำไปใช้ในวงวน `for` ได้  
คำเตือน : ไม่ควรเปลี่ยนแปลงลิสต์ ในวงวนที่กำลังแจกแจงข้อมูลในลิสต์ออกมาใช้งาน อาจมีพฤติกรรมที่ไม่ตรงกับที่คาดไว้
  - $x[0]$  เหมือน  $x[-len(x)+1]$  ได้ 11
  - $x[-1]$  เหมือน  $x[len(x)-1]$  ได้ 15
  - อย่าลืมว่า index ของลิสต์  $x$  ต้องอยู่ในช่วง
    - 0 ถึง  $len(x)-1$  จากซ้ายไปขวา
    - $-1$  ถึง  $-len(x)$  ถอยจากขวามาซ้าย
- ดังนั้นเราเขียน  $x[k]$  ได้ โดยที่  $-len(x) \leq k \leq (len(x)-1)$  เพราะฉะนั้น  $x[-6]$  กับ  $x[5]$  ผิด
- $x[a:b]$  (slice ของลิสต์) ได้ลิสต์เสมอ
  - เขียน  $x[a:b]$  ค่า  $a$  กับ  $b$  เป็นอะไรก็ได้ ไม่ผิด
    - $x[2:50000]$  ได้  $[13, 14, 15]$ ,  $x[4999:50000]$  ได้  $[]$
    - $x[-500:-2]$  ได้  $[11, 12, 13]$ ,  $x[-3:-500:-1]$  ได้  $[13, 12, 11]$ ,  $x[-500:-300]$  ได้  $[]$
  - $x$  มีค่าเหมือน  $x[:]$  เหมือน  $x[0:]$  เหมือน  $x[:len(s)]$  เหมือน  $x[::]$  เหมือน  $x[::1]$
  - $x[::2]$  ได้ลิสต์ย่อยเฉพาะ index คู่  $x[1::2]$  ได้ลิสต์ย่อยเฉพาะ index คี่
  - $x[::-1]$  เหมือน  $x[-1::-1]$  เหมือน  $x[-1:-len(x)+1:-1]$  ได้  $[15, 14, 13, 12, 11]$

ตัวอย่างการเปลี่ยนข้อมูลในลิสต์ (เปลี่ยนอักขระภายในสตริงไม่ได้ แต่เปลี่ยนข้อมูลภายในลิสต์ได้)

- $x[k] = e$  เหมือนกับ  $x[k:k+1] = [e]$
- $x[a:b:c] = y$  นำข้อมูลในลิสต์  $y$  ไปแทนข้อมูลใน  $x[a:b:c]$   
ถ้า  $|c| > 1$   $len(y)$  ต้องเท่ากับ  $len(x[a:b:c])$  เช่น  
 $x = [1, 2, 3, 4, 5]$ ;  $x[::2] = [0, 0, 0]$  ทำให้  $x$  เปลี่ยนเป็น  $[0, 2, 0, 4, 0]$
- $x[len(x):] = [e]$  เหมือนกับ  $x.append(e)$
- $x[i:i] = [e]$  เหมือนกับ  $x.insert(i, e)$
- $x[i:i+1] = []$  เหมือนกับ  $x.pop(i)$
- $x += [1, 2]$  กับ  $x = x + [1, 2]$  ทำให้  $x$  มีสมาชิกเพิ่มอีก 2 ตัวคือ 1 กับ 2 เหมือนกัน แต่สองคำสั่งนี้มีการทำงานต่างกัน
  - $x += [1, 2]$  หมายความว่า ให้นำ 1 กับ 2 ต่อท้ายลิสต์  $x$   
เหมือนกับทำ  $x.append(1)$  ตามด้วย  $x.append(2)$
  - $x = x + [1, 2]$  หมายความว่า ให้  $x$  เก็บลิสต์ใหม่ที่สร้างจากการนำค่าในลิสต์  $x$  เดิมมาต่อกับลิสต์  $[1, 2]$
  - ดังนั้น  $x = [3]$ ;  $y = x$  ทำให้  $x$  กับ  $y$  เป็นลิสต์เดียวกัน  
แต่ถ้าต่อด้วย  $x = x + [9]$  จะทำให้  $x$  กับ  $y$  เป็นลิสต์คนละตัว
  - ในขณะที่  $x = [3]$ ;  $y = x$  เมื่อทำ  $x.append(9)$  แล้ว  $x$  กับ  $y$  ก็ยังเป็นลิสต์เดียวกัน
- สรุปความแตกต่างของ  $x = y$ ,  $x = y[:]$ ,  $x[:] = y$ , และ  $x[:] = y[:]$  เมื่อทั้ง  $x$  และ  $y$  เป็นลิสต์
  - $x = y$   $x$  เปลี่ยนไปอ้างอิงลิสต์ตัวเดียวกับของ  $y$  (หมายความว่า  $x$  และ  $y$  อ้างอิงลิสต์เดียวกัน)
  - $x = y[:]$   $x$  เปลี่ยนไปอ้างอิงลิสต์ใหม่ที่ถูกสร้างขึ้นเหมือนกับของ  $y$  เหมือนคำสั่ง  $x = list(y)$
  - $x[:] = y$   $x$  ยังอ้างอิงลิสต์ตัวเดิม แต่ข้อมูลในลิสต์  $x$  เปลี่ยนไปเหมือนกับข้อมูลของ  $y$  ( $x$  กับ  $y$  เป็นคนละลิสต์)
  - $x[:] = y[:]$  ได้ผลเหมือน  $x[:] = y$

# รูปแบบการประมวลผลข้อมูลที่พบบ่อย

ใช้ลิสต์เก็บข้อมูลเพื่อนำมาใช้ภายหลัง	อ่านข้อมูลเข้ามา n ตัว <pre>n = int(input()) data = [] for k in range(n):     data.append( float(input()) )</pre>
ต้องการหิบบข้อมูลในลิสต์จากซ้ายไปขวา ประมวลผลทีละตัว  ใช้ for e in x	ต้องการนับว่าลิสต์ x มีข้อมูลเท่ากับ e กี่ตัว <pre>c = 0 for d in x :     if d == e : c += 1</pre> หรือแบบสั้น ๆ <pre>c = x.count(e)</pre>
ต้องการหิบบข้อมูลของลิสต์ในช่วงที่สนใจมา ประมวลผล  ใช้ for e in x[a:b:c]	ต้องการหาผลรวมของคะแนนที่เก็บในลิสต์ x โดยขอไม่รวมคะแนน ที่น้อยสุดและมากที่สุด (ตัดออกอย่างละหนึ่งตัว) <pre>s = 0 for d in sorted(x)[1:-1] :     s += d</pre> หรือแบบสั้น ๆ <pre>s = sum( sorted(x)[1:-1] )</pre>
ต้องการปรับเปลี่ยนค่าในลิสต์ ใช้  <pre>for i in range(len(x)) :     x[i] = ...</pre>	x เป็นลิสต์เก็บคะแนน ต้องการปรับช่องที่มีค่าน้อยกว่า 30 ให้มีค่าเพิ่มอีก 10% <pre>for i in range(len(x)) :     if x[i] &lt; 30 :         x[i] += 0.1*x[i]</pre> เขียนแบบข้างล่างนี้ไม่ได้ เพราะ e ที่แจงออกมาเป็นที่เก็บคนละที่กับที่อยู่ ในลิสต์ <pre>for e in x :     if e &lt; 30 :         e += 0.1*e</pre>
ต้องการหิบบข้อมูลแต่ละตัว พร้อมกับ index ของตัวนั้น ๆ	<pre>for i in range(len(t)) :     c = t[i]     print(i,c)</pre> หรือ <pre>for i,c in enumerate(t):     print(i,c)</pre>
ต้องการหิบบข้อมูลในลิสต์จากขวามาซ้าย ทีละตัว	<pre>for e in x[::-1] :     ...</pre> หรือ <pre>for k in range(-1,-(len(x)+1),-1) :     # อ่านยาก โอกาสผิดสูง     e = x[k]     ...</pre>



ต้องการหุบข้อมูลในลิสต์มาประมวลผลจนกว่าเงื่อนไขหนึ่งจะเป็นจริง	<pre> for e in x :     if เงื่อนไขที่ต้องการ :         ...         break     ... else :     ... # มาทำที่นี่ ถ้าไม่พบเงื่อนไขที่ต้องการเลย </pre>
ต้องการหุบข้อมูลในลิสต์ เรียงลำดับที่เรียงจากข้อมูลน้อยสุดไปข้อมูลมากสุดในลิสต์	<pre> for e in sorted(x) :     ... </pre>
ต้องการหุบข้อมูลในลิสต์ เรียงลำดับที่เรียงจากข้อมูลมากสุดไปข้อมูลน้อยสุดในลิสต์	<pre> for e in sorted(x)[::-1] :     ... </pre>
ต้องการหุบข้อมูลในลิสต์จากซ้ายไปขวา มาประมวลผลทีละคู่ข้อมูลที่ติดกัน	<pre> ต้องการตรวจดูว่า ข้อมูลในลิสต์ s เรียงลำดับจากน้อยไปมากหรือไม่ for k in range(len(s)-1) :     if s[k] &gt; s[k+1] :         print("False")         break else:     print("True") </pre>
ต้องการหุบข้อมูลทุก ๆ คู่ในลิสต์ (ไม่จำเป็นต้องติดกัน) มาประมวลผล	<pre> ต้องการนับว่า มีข้อมูลกี่คู่ในลิสต์ที่ตัวทางซ้ายมีค่ามากกว่าตัวทางขวา (ไม่จำเป็นต้องอยู่ติดกัน) c = 0 for i in range(len(x)) :     for j in range(i+1,len(x)) :         if x[i] &gt; x[j] : c += 1 print(c) </pre>
เรียงลำดับข้อมูลในลิสต์ (ด้วยวิธี bubble sort)	<pre> # ต้องการเรียงลำดับข้อมูลในลิสต์ d จากน้อยไปมาก for k in range(len(d)-1) :     for i in range(len(d)-1) :         if d[i] &gt; d[i+1] :             d[i],d[i+1] = d[i+1],d[i] </pre>

## เรื่องพิศบอย

ใช้ index ที่เกินช่วงที่ใช้ได้ของลิสต์ อย่าลืมว่า index ของลิสต์ x อยู่ในช่วง 0 ถึง len(x)-1 จากซ้ายไปขวา และ -1 ถึง -len(x) ถอยจากขวามาซ้าย	<pre> x = [2,3,5,7,11,13,17,19,23] s = 0 for i in range(len(x),0,-1):     s += i*x[i]     # ผิด ค่าแรกของ i คือ len(x)อยู่นอกช่วง </pre>
เลข index ที่ใช้กับลิสต์ไม่ใช่จำนวนเต็ม	<pre> print( x[n/2] ) # ผิด เพราะ n/2 ได้จำนวนจริง </pre>

เมื่อ x เป็นลิสต์ อย่าสับสนระหว่าง x.append(y) กับ x+=y ซึ่งได้ผล ไม่เหมือนกัน	<pre> = [1,2,3,4]; x.append( [5] ) # ได้ [1,2,3,4,[5]] x = [1,2,3,4]; x += [5]      # ได้ [1,2,3,4,5] x = [1,2,3,4]; x.append( 5 ) # ได้ [1,2,3,4,5] x = [1,2,3,4]; x += 5        # ผิด </pre>
ต้องการลิสต์ y ที่มีค่าเหมือนกับลิสต์ x, ต้อง เลือกจะใช้ y = x หรือ y = list(x) ปกติไม่ค่อยน่าจะใช้ y = x	<pre> x = [1,2,3] y = x                # y เป็นลิสต์เดียวกับ x x[2] = 0             # y[2] ก็เปลี่ยนเป็น 0 ด้วย </pre>
เขียน x = list(str) เป็นการสร้างลิสต์ ที่ประกอบด้วย 'แต่ละอักขระ' ใน str	<pre> x = list('abc') ได้ x = ['a','b','c'] ไม่ใช่ x = ['abc'] </pre>
นำข้อมูลในลิสต์มา join กันให้เป็นสตริง แต่ลืมไปว่าข้อมูลในลิสต์นั้นต้องเป็นสตริง ถึงจะ join ได้	<pre> x = [1,2,3] s = ','.join(x)    # ผิด เพราะ 1,2,3 ไม่ใช่สตริง t = [] for e in x :     t.append( str(e) ) s = ','.join(t)    # ใช้ได้ เพราะ t เป็นลิสต์ของสตริง </pre>
สับสนคำสั่ง x.sort() กับ sorted(x) x.sort() เรียงลำดับข้อมูลในลิสต์ x คำสั่งนี้ไม่คืนผลใด ๆ ในขณะที่ sorted(x) นำข้อมูลที่ได้จาก x มาเรียง ลำดับแล้วคืนลิสต์ใหม่ที่มีข้อมูลเหมือนใน x แต่เรียงลำดับแล้ว โดยที่ x ไม่เปลี่ยนแปลง	<pre> x = [9,2,0,4] x = x.sort()        # ผิด แบบนี้ทำให้ x เก็บค่า None                     # เพราะ x.sort() ไม่คืนผลใด ๆ for e in x.sort() :     ...             # ผิด x.sort() ไม่คืนผล ไม่มีอะไรให้ e ถ้าต้องการเรียงลำดับข้อมูลใน x ใช้ x.sort() หรือ x = sorted(x) for e in sorted(x) :     ...             # แบบนี้ได้ sorted(x) คืนลิสต์ที่เรียงแล้ว </pre>
มีการลบหรือเพิ่มข้อมูลในลิสต์ระหว่างที่ มีการแจกแจงข้อมูลในลิสต์	<pre> x = [1,2,3,3,2,1] for e in x :     if e%2 == 1 :         x.pop(x.index(e)) หรือ for i in range(len(x)) :     if x[i]%2 == 1 :         x.pop(i) </pre> <p>จะไม่ได้ผลตามที่คาด และเกิดข้อผิดพลาดระหว่างการทำงานด้วย (ลอง run ดู) ควรใช้การสร้างลิสต์ชั่วคราวก่อน แล้วค่อยนำกลับไปใส่ในลิสต์เดิม</p> <pre> t = [] for e in x :     if e%2 != 1 :         t.append(e) x[:] = t </pre> <p>ให้สังเกตว่าคำสั่งสุดท้ายคือ x[:] = t แทนที่จะเป็น x = t เพราะต้องการ เปลี่ยนข้อมูลในลิสต์ x ให้เหมือน t ไม่ได้ต้องการให้ x ไปอ้างอิงลิสต์เดียวกับ t</p>

# แบบฝึกหัด

Problem	Code
<p><b>Input:</b> มี 2 บรรทัด แต่ละ บรรทัดเป็นสมาชิกของเวกเตอร์ ซึ่งเป็นจำนวนจริงหลายจำนวนโดยช่วยช่องว่าง อ่านทั้งสอง บรรทัดเก็บในลิสต์ <math>v1</math> และ <math>v2</math></p> <p><b>Process:</b> คำนวณ dot product ของเวกเตอร์ <math>v1</math> กับ <math>v2</math></p> <p><b>Output:</b> แสดงคำว่า Error ถ้า <math>v1</math> และ <math>v2</math> มีขนาด ไม่เท่ากัน แต่ถ้าเท่ากัน แสดงค่า dot product ที่คำนวณได้</p>	
<p><b>Input:</b> บรรทัดแรกเป็นจำนวนเต็ม <math>n</math> และมีอีก <math>n</math> บรรทัด แต่ละบรรทัดเป็นจำนวนเต็ม 1 จำนวน</p> <p><b>Process:</b> เรียงลำดับจำนวนเต็มทั้ง <math>n</math> ตัวจากน้อยไปมาก</p> <p><b>Output:</b> จำนวนเต็มทั้ง <math>n</math> ที่เรียงจากน้อยไปมากบนบรรทัด เดียวกันเรียงจากซ้ายขวา คั่นด้วยจุลภาค ,</p>	
<p><b>Input:</b> หนึ่งบรรทัดเป็นชื่อแฟ้ม แฟ้มนี้เก็บจำนวนเต็ม บรรทัดละจำนวน</p> <p><b>Process:</b> หาว่าจำนวนเต็มใดในแฟ้มปรากฏซ้ำกันมากที่สุด ถ้ามีซ้ำกันมากที่สุดหลายตัว ให้หาทุกตัว</p> <p><b>Output:</b> ข้อมูลทุกตัวที่ปรากฏซ้ำกันมากสุดในแฟ้ม เรียงตามลำดับที่ปรากฏในแฟ้ม</p>	

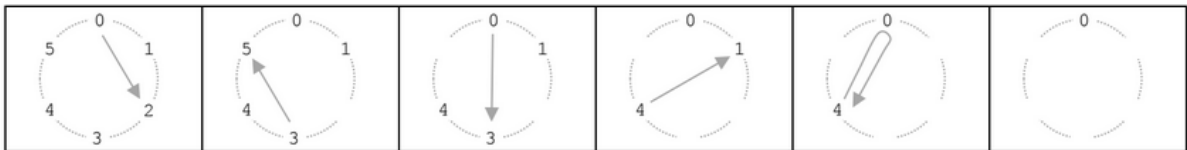
Problem	Code
<p><u>Input</u>: หนึ่งบรรทัดเป็นชื่อแฟ้ม</p> <p><u>Process</u>: แสดงหัวข้อข่าวทั้งหมดในแฟ้ม หัวข้อข่าวเป็นข้อความที่อยู่ระหว่าง <code>&lt;headline&gt;</code> กับ <code>&lt;/headline&gt;</code> ในแฟ้มนี้ (ทั้ง <code>&lt;headline&gt;</code> กับ <code>&lt;/headline&gt;</code> อยู่ในบรรทัดเดียวกันแน่ ๆ และแต่ละบรรทัดมีไม่เกิน 1 หัวข้อข่าว)</p> <p><u>Output</u>: หนึ่งบรรทัดหนึ่งหัวข้อ ให้ครบทุกหัวข้อ โดยแสดงเรียงหัวข้อตามตัวอักษรจากน้อยไปมาก</p>	
<p><u>Input</u>: หนึ่งบรรทัดเป็นชื่อแฟ้ม แฟ้มนี้เก็บชื่อบรรทัดละหนึ่งชื่อ</p> <p><u>Process</u>: เรียงลำดับชื่อที่อ่านจากแฟ้ม โดยเรียงลำดับตามความยาวของชื่อจากน้อยไปมาก ถ้ามีความยาวเท่ากันให้เรียงตามตัวอักษรแบบในพจนานุกรม</p> <p><u>Output</u>: ลำดับของชื่อตามที่เรียงได้</p>	

## ตัวอย่างการแก้โจทย์ปัญหา

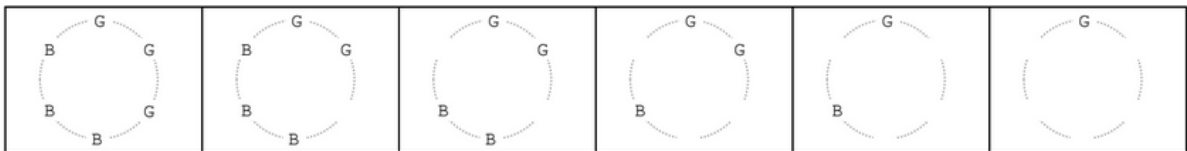
### ขจัดคนเลว

กำหนดให้  $n$  เป็นจำนวนเต็ม หากเรานำคนดี  $n$  คน กับคนเลว  $n$  คน มาเรียงเป็นวงกลม โดยให้คนดียืนเรียงติดกัน ตั้งแต่ตำแหน่งที่ 0 ถึง  $n-1$  แล้วคนเลวยืนเรียงต่อไปตั้งแต่ตำแหน่ง  $n$  ถึง  $2n-1$  จงหาจำนวนเต็มบวก  $d$  ที่มีค่าน้อยสุด ที่เมื่อนำคนทั้งหมดมาใช้กับปัญหา Josephus แล้วจะเหลือคนดีเป็นคนสุดท้าย

ปัญหา Josephus เป็นดังนี้ : ข้อมูลนำเข้าคือจำนวนเต็ม  $m$  กับ  $d$  ให้มี  $m$  คนยืนเรียงเป็นวงกลม เริ่มคนที่ 0 นับไปอีก  $d$  คน ก็ให้คนนั้นออกจากวง แล้วก็เริ่มจากคนถัดไปนับไปอีก  $d$  คน ก็ให้คนนั้นออก ทำเช่นนี้ไปเรื่อย ๆ จนเหลือคนสุดท้าย คนนั้นเป็นผู้ชนะ เช่น ให้  $m = 6$  และ  $d = 2$  การเปลี่ยนแปลงของคนที่ยืนในวงกลมแสดงได้ดังรูปข้างล่างนี้ หมายเลข 0 เป็นผู้ชนะ



ดังนั้นถ้าให้  $n = 3$  จะได้ค่า  $d = 2$  เป็นค่าที่เมื่อขจัดคนออกแล้ว จะได้คนดีเป็นคนสุดท้าย (G แทนคนดี, B แทนคนเลว)



#### ► ข้อมูลนำเข้า

จำนวนเต็มบวก 1 จำนวน แทนค่า  $n$  ข้างต้น

#### ► ข้อมูลส่งออก

ค่า  $d$  ที่เป็นจำนวนบวกน้อยสุดที่ทำให้ขจัดคนออกแล้วเหลือคนสุดท้ายเป็นคนดี ดังที่อธิบายไว้ข้างต้น

#### ► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
1	1
3	2
6	3
13	4
14	6

## ตัวอย่างการเขียนโปรแกรม

โปรแกรม	คำอธิบาย
<pre> n = int(input()) for d in range(1,2*n):     q = ['G']*n + ['B']*n     m = 2*n      # solve Josephus problem      if len(q) &gt; 0 and q[0] == 'G':         print(d)         break     else:         print('Not found') </pre>	<p>เนื่องจากคอมพิวเตอร์ทำงานได้รวดเร็ว ขอแก้ปัญหาด้วยการจำลองการค่อย ๆ ขจัดคนในวงกลมออกตามกฎเกณฑ์ที่กำหนดไว้ โดยใช้ลิสต์เก็บคนที่ยืนในวงกลม แล้วค่อย ๆ ลบข้อมูลในลิสต์ออก</p> <p>โปรแกรมทางซ้ายนี้รับค่า <math>n</math> จากนั้นลู่ทดสอบการขจัดคนด้วยค่า <math>d</math> ต่าง ๆ เริ่มที่ 1 ไปถึง <math>2n-1</math> ภายในแต่ละรอบของ <code>for</code> จะสร้างลิสต์ของตัวอักษร G จำนวน <math>n</math> ตัวตามด้วยตัวอักษร B อีก <math>n</math> ตัว ('G' แทนคนดี, 'B' แทนคนเลว) แล้วก็เริ่มขั้นตอนการแก้ปัญหา Josephus ด้วย <math>m = 2*n</math> และ <math>d</math> ตามค่าของ <code>for</code> หากผลการขจัดคนในวงกลมของปัญหา Josephus เหลือคนสุดท้ายเป็นคนดี ก็แสดงค่า <math>d</math> และ <code>break</code> ออกจาก <code>for</code> ได้เลย ถ้าไม่ได้คนดีเป็นคนสุดท้าย ก็วนกลับไปเพิ่มค่า <math>d</math> เพื่อทำรอบต่อไปของ <code>for</code></p>
<pre> m = int(input()) d = int(input()) q = list(range(m)) k = 0 while len(q) &gt; 1 :     k += d     if k &gt;= len(q) :         k -= len(q)     q.pop(k) print(q[0]) </pre>	<p>ขอเก็บโปรแกรมข้างบนนี้ไว้ก่อน และมาเขียนโปรแกรมสำหรับปัญหา Josephus ทางซ้ายนี้ ที่รับค่าจำนวนเต็ม <math>m</math> กับ <math>d</math> แล้วสร้างลิสต์ที่เก็บหมายเลข 0 ถึง <math>m-1</math> ด้วยคำสั่ง <code>list(range(m))</code> จากนั้นเข้าสู่ <code>while</code> ที่จะวนทำงานตราบเท่าที่ลิสต์ <code>q</code> ยังมีข้อมูลมากกว่าหนึ่งตัว มีตัวแปร <math>k</math> เก็บ <code>index</code> ของลิสต์ที่มีค่าเริ่มที่ 0 ในแต่ละรอบจะลบข้อมูลตัวที่ถัดจาก <math>k</math> ไปอีก <math>d</math> ตัว ซึ่งคือเพิ่มค่า <math>k</math> อีก <math>d</math> ถ้าค่า <math>k</math> เกินหรือเท่ากับขนาดของลิสต์ ก็ต้องวนกลับมาด้านซ้ายของลิสต์ ซึ่งก็คือการลดค่าของ <math>k</math> ด้วยขนาดของลิสต์ เช่น <code>q</code> มีข้อมูล 7 ตัว, <math>k = 5</math>, <math>d = 3</math> คำสั่ง <code>k += d</code> ทำให้ <math>k = 8</math> เกินขนาดของ <code>q</code> คำสั่ง <code>k -= len(q)</code> ทำให้ <math>k</math> กลับมามีค่าเป็น 1 สรุปคือ ถัดจากตำแหน่งที่ 5 ไปอีก 3 ตำแหน่งคือ 1 (<math>5 \rightarrow 6 \rightarrow 0 \rightarrow 1</math>)</p> <p>เมื่อได้ตำแหน่ง <math>k</math> ที่เราต้องลบข้อมูลออก ก็ทำคำสั่ง <code>q.pop(k)</code> วน <code>while</code> จะลบข้อมูลรอบละตัว เมื่อเหลือข้อมูลตัวเดียว ข้อมูลที่เหลือรอดนั้นก็คือ <code>q[0]</code></p> <p>สั่ง <code>run</code>, ใส่ <math>m</math> เป็น 6, <math>d</math> เป็น 2, ได้ผลเป็น 0 ถูกต้อง</p> <p>สั่ง <code>run</code>, ใส่ <math>m</math> เป็น 4, <math>d</math> เป็น 3, ทำงานผิด</p> <p><code>IndexError: pop index out of range</code></p> <p>ที่คำสั่ง <code>q.pop(k)</code> แปลว่า <math>k</math> เก็บ <code>index</code> ที่มีค่าเกินช่วงที่ลิสต์ <code>q</code> มีให้ลบ ถ้าลองแทรกคำสั่ง <code>print(k, len(q))</code> จะได้ 3 3 นั่นคือลิสต์ที่มี 3 ตัว ลบตัวที่ <code>index</code> 3 ไม่ได้ เพราะมีให้ใช้แค่ <code>index</code> 0 ถึง 2 เท่านั้น</p>
<pre> m = int(input()) d = int(input()) q = list(range(m)) k = 0 while len(q) &gt; 1 :     k = (k + d) % len(q)     q.pop(k) print(q[0]) </pre>	<p>แก้ปัญหาข้างต้นได้ด้วยการใช้คำสั่ง <math>k = (k + d) \% \text{len}(q)</math> เพื่อคำนวณ <code>index</code> ที่ถัดจาก <math>k</math> ไปอีก <math>d</math> ช่อง กรณีที่ <code>index</code> เลยไปทางขวาของลิสต์ การ <code>mod</code> ด้วยขนาดของลิสต์ จะได้ผลวนกลับมาทางซ้าย เช่น ถ้าลิสต์มีขนาด 4 ตัว ถัดจากตำแหน่ง 3 ไปอีก 5 ตำแหน่งก็คือ <math>(3+5) \% 4 = 0</math></p> <p>สั่ง <code>run</code>, ใส่ <math>m</math> เป็น 6, <math>d</math> เป็น 2, ได้ผลเป็น 0 ถูกต้อง</p> <p>สั่ง <code>run</code>, ใส่ <math>m</math> เป็น 4, <math>d</math> เป็น 3, ได้ผลเป็น 1</p> <p>ลองจำลองการทำงานได้</p> <p>0,1,2,3 <math>\rightarrow</math> 0,1,2 <math>\rightarrow</math> 1,2 <math>\rightarrow</math> 1 ถูกต้อง</p>

โปรแกรม	คำอธิบาย
<pre> n = int(input()) for d in range(1,2*n) :     q = ['G']*n + ['B']*n     k = 0     while len(q) &gt; 1 :         k = (k + d) % len(q)         q.pop(k)         if q[0] == 'G' :             print(d)             break     else:         print('Not round') </pre>	<p>แนวคิดสิ่งในการหาคำตอบของปัญหา Josephus ข้างต้นไปแทรกในโปรแกรมตอนที่เขียน ได้โปรแกรมทางซ้ายนี้</p> <p>สั่ง run, ใส่ 1, ได้ 1, ถูกต้อง</p> <p>สั่ง run, ใส่ 3, ได้ 2, ถูกต้อง</p> <p>สั่ง run, ใส่ 6, ได้ 3, ถูกต้อง</p> <p>สั่ง run, ใส่ 13, ได้ 4, ถูกต้อง</p> <p>สั่ง run, ใส่ 14, ได้ 6, ถูกต้อง</p>

## 06.2 : Nested List

### สรุปเนื้อหา

ลิสต์เกือบอะไรก็ได้ จำนวนเต็ม จำนวนจริง สตริง หรืออื่น ๆ หรือแม้กระทั่งลิสต์ ก็ได้ ให้  $x$  เป็นลิสต์ และถ้า  $x[k]$  ก็เป็นลิสต์ จะได้  $x[k][i]$  ค่าข้อมูลอยู่ที่ index  $j$  ของลิสต์  $x[k]$

```
x = []
x.append( 1 )           # [1]
x.append( [2,3] )       # [1,[2,3]]
x.append( [4,5,6] )     # [1,[2,3],[4,5,6]]
x.append( [[7,8]] )     # [1,[2,3],[4,5,6],[[7,8]]]
x.append( [] )          # [1,[2,3],[4,5,6],[[7,8]],[]]
len(x) คือ 5, x[1][1] คือ 3, x[2][1] คือ 5, x[3][0][1] คือ 8, len(x[3][0]) คือ 2, len(x[4]) คือ 0
```

### รูปแบบการประมวลผลลิสต์ซ้อนลิสต์ที่พบบ่อย

ใช้ลิสต์ซ้อนลิสต์แทนการเก็บลิสต์ของข้อมูล โดยที่ข้อมูลแต่ละตัวเป็นลิสต์ที่เก็บข้อมูลย่อย ๆ จำนวนเท่า ๆ กัน

ใช้ลิสต์ซ้อนลิสต์เก็บข้อมูลของวงกลมหลาย ๆ วง ลิสต์ข้างในคือจำนวนสามจำนวนที่แทนพิกัด  $x$ ,  $y$  และรัศมีของวงกลม เช่น

```
[[0.0, 0.0, 10.0], [1.0, 5.0, 3.0]]
```

โปรแกรมข้างล่างนี้อ่านข้อมูลวงกลมจากแฟ้มมาเก็บในลิสต์

```
circles = []
file1 = open('c:/temp/circles.txt')
for line in file1 :
    x,y,r = line.split()
    circles.append( [float(x),float(y),float(r)] )
```

ข้างล่างนี้หาวงกลมที่ไม่ทับหรือแตะวงกลมอื่นเลย

```
free = []
for i in range(len(circles)) :
    for j in range(i+1,len(circles)) :
        dx = circles[i][0] - circles[j][0]
        dy = circles[i][1] - circles[j][1]
        sumr = circles[i][2] + circles[j][2]
        if dx**2 + dy**2 <= sumr**2 : break
    else :
        free.append(circles[i])
```

```
for c in free :
    print(c)
```



<p>ใช้ลิสต์ซ้อนลิสต์ สร้างเมทริกซ์ ถ้าต้องการเมทริกซ์ขนาด <math>m \times n</math> ก็ สร้างลิสต์ A ที่ <code>len(A)</code> มีค่าเท่ากับ <math>m</math> (จำนวนแถว) และ <code>len(A[0]) = len(A[1])</code> <code>= ... = len(A[m-1]) =</code> (จำนวนคอลัมน์) เช่นสร้างเมทริกซ์ ที่มีค่า 0 ขนาด <math>5 \times 1</math> <code>A = []</code> <code>for k in range(5):</code> <code>    A.append([0]*1)</code></p>	<p>ตัวอย่างการหาผลบวกของเมทริกซ์ A กับ B</p> $A = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 \\ 4 & 3 \end{bmatrix} \quad A + B = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 4 & 3 \end{bmatrix}$ <pre> C = [] for i in range(len(A)) :     C.append([0]*len(A[i]))     for j in range(len(A[i])) :         C[i][j] = A[i][j] + B[i][j]</pre>
<p>ใช้ลิสต์ซ้อนลิสต์ โดยที่ลิสต์ข้างในเก็บ ข้อมูลที่ไม่จำเป็นต้องมีขนาดเท่ากัน</p>	<p>ใช้ลิสต์ซ้อนลิสต์ เก็บข้อมูลที่ประกอบด้วย username (ที่ช่อง 0), และลิสต์ของ usernames รายอื่นที่ขอติดตามข่าวสาร (ที่ช่อง 1 โดยลิสต์นี้ไม่จำเป็นต้องมี ขนาดคงตัว) (ตัวอย่างนี้ซ้อนตั้งสามชั้น) เช่น</p> <pre> f = [ ['noon',['pat','koi']], ['wii',['noon','koi']],       ['pat',['koi','noon','wii']], ['koi',[]] ]</pre> <p>ต้องการหาว่า username ไหนมีคนติดตามเป็นจำนวนมากสุด</p> <pre> maxindex = 0 for k in range(1,len(f)) :     if len(f[k][1]) &gt; len(f[maxindex][1]) :         maxindex = k print(f[maxindex][0], 'has max.# of followers :', \       ', '.join(f[maxindex][1]))</pre> <p>หรือเขียน</p> <pre> max_followers = [] max_username = '' for [username,followers] in f :     if len(followers) &gt; len(max_followers) :         max_username = username         max_followers = followers print(max_username, 'has max.# of followers :', \       ', '.join(max_followers))</pre>

<p>ใช้ลิสต์ซ้อนลิสต์เป็นที่เก็บข้อมูลชั่วคราวเพื่อนำไป sort ตามข้อกำหนดที่ต้องการ โดยสร้างลิสต์ข้างในให้มีสมาชิกตัวแรกเป็นเกณฑ์ที่ใช้ในการ sort</p> <p>หมายเหตุ : การ sort ลิสต์จะเปรียบเทียบความน้อยกว่าของข้อมูลของลิสต์ข้างในทีละตัวจากซ้ายไปขวา เช่น</p> <pre>x=[[3,2],[3,1],[9],[2,5],[3]] x.sort() จะได้ x เปลี่ยนเป็น [[2,5],[3],[3,1],[3,2],[9]] นั่นคือ [2,5] &lt; [3] &lt; [3,1] &lt; ...</pre>	<p>จากตัวอย่างก่อนหน้า เมื่อมี f แล้ว ถ้าต้องการเรียงลำดับ usernames ให้หลายตามจำนวนผู้ติดตาม ก็ใช้</p> <pre>c = [] for [username,followers] in f :     c.append([len(followers),username]) c.sort() for [x,username] in c :     print(username)</pre> <p>จากตัวอย่างบน</p> <pre>f = [ ['noon','pat','koi'], ['wii',['noon','koi']],       ['pat',['koi','noon'],'wii'], ['koi',[]] ] จะได้ c = [[2,'noon'],[2,'wii'],[3,'pat'],[0,'koi']] c.sort() ได้ [[0,'koi'],[2,'noon'],[2,'wii'],[3,'pat']]</pre> <p>หากต้องการให้ sort ด้วยเกณฑ์ที่ซับซ้อนขึ้น เช่น ให้เรียงตามจำนวนผู้ติดตามจากน้อยไปมาก และในกรณีที่จำนวนผู้ติดตามเท่ากัน ให้เรียงตาม username จากน้อยไปมากเช่นกัน แบบที่ต้องการนี้ทั้งสองข้อมูลย่อยในลิสต์เรียงแบบน้อยไปมาก สามารถใช้ sort() ได้เลย แต่ถ้าเปลี่ยนเป็น ให้เรียงตามจำนวนผู้ติดตามจากมากไปน้อย สำหรับกรณีที่จำนวนผู้ติดตามเท่ากัน ให้เรียงตาม username จากน้อยไปมาก ก็อาจใช้กลวิธีเล็กน้อย เช่น ดิดลบจำนวนที่ต้องการเรียงจากมากไปน้อย การเรียงเลขลบจากน้อยไปมาก ก็คือเรียงเลขบวกจากมากไปน้อย</p> <pre>c = [] for [username,followers] in f :     c.append([-len(followers),username]) c.sort()</pre> <p>จากตัวอย่างบน</p> <pre>f = [ ['noon','pat','koi'], ['wii',['noon','koi']],       ['pat',['koi','noon'],'wii'], ['koi',[]] ] จะได้ c = [[-2,'noon'],[-2,'wii'],[-3,'pat'],[0,'koi']] sort ได้ [[-3,'pat'],[-2,'noon'],[-2,'wii'],[0,'koi']]</pre>
---	--

## เรื่องพิศบอย

<p>การเพิ่มลิสต์ย่อย a เข้าไปในลิสต์ใหญ่ x ต้องใช้ x.append(a) ไม่ใช่</p> <pre>x = x + a หรือ x += a</pre>	<pre>x = [[2,3],[3,4]] ได้มาจากการ append [2,3] กับ [3,4] เข้า x x = [] x.append( [2,3] ) x.append( [3,4] ) แต่ไม่เหมือนคำสั่งข้างล่างนี้ x = [] x = x + [2,3] + [3,4] # ได้ [2,3,3,4]</pre>
--	--

<p>อย่าสร้างลิสต์ซ้อนลิสต์ด้วย *</p> <p>อย่าเขียน <code>[a]*n</code> เมื่อ <code>a</code> เป็นลิสต์</p>	<p>ต้องสร้างเมทริกซ์ที่แทนด้วยลิสต์ซ้อนลิสต์ ที่มีขนาด <math>3 \times 3</math> มีค่าเป็น 0 ทั้งหมด</p> <p><code>x = [ [0,0,0] ] * 3</code></p> <p><code>print(x)</code> จะได้ <code>[[0,0,0],[0,0,0],[0,0,0]]</code></p> <p>แต่ลิสต์ข้างในทั้งสามตัวนี้คือลิสต์ตัวเดียวกัน ถ้าสั่ง <code>x[0][1] = 1</code> ทำงาน</p> <p><code>x</code> จะเปลี่ยนเป็น <code>[[0,1,0],[0,1,0],[0,1,0]]</code></p> <p>ใช้ <code>x = [[0]*3]*3</code> ก็ได้ผลที่แปลกแบบข้างบนเหมือนกัน</p> <p>เราต้องสร้างลิสต์ข้างในให้เป็นคนละตัว โดยเขียน</p> <p><code>x = [[0,0,0],[0,0,0],[0,0,0]]</code></p> <p>หรือใช้วงวนสร้าง</p> <pre>n = 3 x = [] for i in range(n):     x.append([0]*n)</pre> <p>สรุปคือ เมื่อใดเขียน <code>[a]*n</code> ต้องระวัง ถ้า <code>a</code> เป็นลิสต์ ให้ใช้วงวนสร้างแทน</p>
---	---



Problem	Code
<p>Input: บรรทัดแรกมีจำนวนเต็ม <code>r</code> กับ <code>c</code> ตามด้วยอีก <code>r</code> บรรทัด แต่ละบรรทัดมีจำนวนเต็ม <code>c</code> ตัว</p> <p>Process: สร้างเมทริกซ์ด้วยลิสต์ซ้อนลิสต์ ถ้ามีบรรทัดที่มีข้อมูลไม่ใช่ <code>c</code> ตัว ให้แสดงผลเป็นลิสต์ซ้อนลิสต์ว่าง <code>[]</code></p> <p>Output: เมทริกซ์ลิสต์ซ้อนลิสต์ที่สร้างได้</p>	
<p>Input: หนึ่งบรรทัดเป็นชื่อแฟ้ม แฟ้มนี้มีหลายบรรทัดเท่ากับจำนวน <code>username</code> แต่ละบรรทัดประกอบด้วย <code>username</code> ตามด้วย <code>usernames</code> อื่น ๆ ที่ติดตามข่าวสารของ <code>username</code> แรกต้นบรรทัด เช่น</p> <pre>noon pat koi wii noon pat koi noon wii koi</pre> <p>Process: สร้างลิสต์ซ้อนลิสต์ที่มีรูปแบบตามตัวอย่างนี้</p> <pre>f = [['noon', ['pat', 'koi']],       ['wii', ['noon']],       ['pat', ['koi', 'noon', 'wii']],       ['koi', []]]</pre> <p>Output: ลิสต์ที่สร้างได้</p>	

Problem	Code
<p>Input: จาก f ที่ได้ในข้อที่แล้ว</p> <p>Process: ต้องการรู้ว่า ใครบ้างที่ไม่มีใครติดต่อกับเลย</p> <p>Output: รายชื่อของผู้ที่ไม่มีใครติดต่อกับเลย</p>	
<p>Input: บรรทัดแรกมีจำนวนเต็ม k และอีก n บรรทัดที่ตามมาเป็นสตริง</p> <p>Process: เรียงลำดับสตริงที่อ่านเข้ามา จากน้อยไปมากตามความยาวล.เรียง ถ้าสตริงยาวเท่ากันให้เรียงตามตัวสตริงเอง เช่น 'xyz', 'xy', 'abc' เรียงแล้วได้ 'xy', 'abc', 'xyz'</p> <p>Output: สตริงที่เรียงแล้ว บรรทัดละสตริง</p>	

## ตัวอย่างการแก้โจทย์ปัญหา

### เรียงตามคะแนนรวม

จงเขียนโปรแกรมรับรหัสนักเรียนและรายการของคะแนนการสอบย่อยต่าง ๆ ของนักเรียนจำนวนหนึ่ง มาประมวลผลเพื่อแสดงรหัสนักเรียนและคะแนนรวมตามลำดับคะแนนรวมจากมากไปน้อย

#### ► ข้อมูลนำเข้า

บรรทัดแรกมีจำนวนเต็มบวก n กับ m (คั่นด้วยช่องว่าง) n คือจำนวนนักเรียน และ m คือจำนวนการสอบย่อยของนักเรียนแต่ละคน n บรรทัดต่อมา แต่ละบรรทัดประกอบด้วย รหัสนักเรียนตามด้วยรายการของคะแนนย่อย คั่นด้วยช่องว่าง

#### ► ข้อมูลส่งออก

ถ้ามีบรรทัดที่จำนวนคะแนนย่อยไม่ตรงกับ m ให้รวบรวมรหัสนักเรียนมาแสดงตามตัวอย่าง (เรียงตามที่ได้รับจากข้อมูลนำเข้า)

ถ้าทุกบรรทัดมีคะแนนครบจำนวนทุกคน ให้แสดงรหัสนักเรียนตามด้วยคะแนนรวมของนักเรียนบรรทัดละคน เรียงลำดับตามคะแนนรวมจากมากไปน้อย ในกรณีที่มียกคะแนนรวมเท่ากัน ให้เรียงตามรหัสนักเรียนจากน้อยไปมาก

#### ► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
3 2 A 3.0 2.0 B 4.0 5.0 C 2.0 3.0	B 9.0 A 5.0 C 5.0
3 3 AA 4.0 B 9.0 8.0 7.0 AC 8.0 5.0	Invalid data: AA AC

## ตัวอย่างการเขียนโปรแกรม

ขอเก็บข้อมูลในลิสต์ แบบลิสต์ซ้อนลิสต์ ลิสต์ข้างในแต่ละลิสต์ เก็บรหัสนักเรียนที่ช่อง 0 ตามด้วยคะแนนตั้งแต่ช่องที่ 1 ถึง m  
[ [รหัสนักเรียน, คะแนน, ..., คะแนน], [รหัสนักเรียน, คะแนน, ..., คะแนน], ... ]

ขอแบ่งการประมวลผลเป็นขั้นตอนที่ละเอียดดังนี้

1. อ่านข้อมูลนำเข้าเก็บเป็นลิสต์ซ้อนลิสต์ เนื่องจากข้อมูลที่เข้ามา รหัสเป็นสตริง ส่วนคะแนนต้องการเก็บเป็นจำนวนจริง แต่ขอเริ่มด้วยการอ่านเข้ามาจนเสร็จให้หมดก่อน ถ้าใช้ข้อมูลนำเข้าของตัวอย่างที่สอง ได้  
d0 = [['AA', '4.0'], ['B', '9.0', '8.0', '7.0'], ['AC', '8.0', '5.0']]
2. นำ d0 มาหาว่ารหัสนักเรียนใดที่มีจำนวนคะแนนไม่เท่ากับจำนวนที่กำหนดให้ (ตัวอย่างที่สองระบุว่าต้องมี 3 ข้อ) ได้ err = ['AA', 'AC']
3. ถ้า err มีขนาดเป็น 0 ก็แสดงว่ามีที่ผิด จึงแสดง Invalid data ตามด้วยรหัสนักเรียนที่เก็บใน err
4. ถ้า err มีขนาดเป็น 0 (คือไม่มีผิดเลย) ประมวลผลต่อ (คราวนี้ขอใช้ข้อมูลจากตัวอย่างแรก เพราะไม่มีที่ผิด) ได้ d0 = [['A', '3.0', '2.0'], ['B', '4.0', '5.0'], ['C', '2.0', '3.0']] และ err = []
  - 4.1. นำ d0 จากขั้นตอนที่แล้ว เปลี่ยนคะแนนให้เป็นจำนวนจริง แล้วหาผลรวม นำมาเก็บคู่กับรหัสนักเรียน ได้ d1 = [[5.0, 'A'], [9.0, 'B'], [5.0, 'C']]
  - 4.2. เรียงลำดับข้อมูลใน d1 ตามโจทย์ คือคะแนนรวมจากมากไปน้อย ถ้าคะแนนรวมเท่ากัน เรียงตามรหัส จากนั้นน้อยไปมาก ได้ d1 = [[9.0, 'B'], [5.0, 'A'], [5.0, 'C']]
  - 4.3. นำข้อมูลใน d1 มาแสดงเป็นผลลัพธ์

โปรแกรม	คำอธิบาย
<pre>n,m = [int(e) for e in input().split()] d0 = list() for k in range(n) :     d0.append(input().split())</pre>	<p>ขั้นตอนที่ 1 : อ่านค่า n (จำนวนนักเรียน) กับ m (จำนวนคะแนนต่อคน) ใช้วงวน for วงจำนวน n รอบ อ่านจากแป้นพิมพ์แล้ว split ได้เป็นลิสต์ของสตริง เพิ่มแต่ละลิสต์ที่ได้เข้าในลิสต์ d0</p> <p>สั่ง run, ใส่ข้อมูลตามตัวอย่างที่สอง, แล้ว print(d0) ได้</p> <pre>['AA', '4.0'], ['B', '9.0', '8.0', '7.0'], ['AC', '8.0', '5.0']]</pre>
<pre>err = list() for x in d0 :     if len(x[1:]) != m :         err.append( x[0] ) if len(err) &gt; 0 :     print('Invalid data: ')     for sid in err :         print(sid)</pre>	<p>ขั้นตอนที่ 2, 3 : นำแต่ละลิสต์ x ข้างในลิสต์ d0 จากขั้นตอนที่แล้ว มาตรวจ ถ้า x[1:] มีขนาดไม่เท่ากับ m คือมีจำนวนคะแนนไม่เท่ากับที่กำหนด จะนำ x[0] ใส่เพิ่มในลิสต์ err หลังจากสร้างเสร็จ ถ้า err มีขนาดเกิน 0 ก็แสดงรหัสนักเรียนที่มีความผิดพลาด</p> <p>สั่ง run, ใส่ข้อมูลตามตัวอย่างที่สอง, ได้</p> <p>Invalid data:</p> <pre>AA AC</pre>
<pre>else :     d1 = list()     for x in d0 :         s = 0         for e in x[1:] :             s += float(e)         d1.append( [s, x[0]] )</pre>	<p>ขั้นตอนที่ 4.1 : เป็นกรณีที่มีคะแนนครบ แจกแจงลิสต์ x ข้างใน d0 ส่วนที่เก็บคะแนน ซึ่งคือ x[1:] มาหาผลรวม โดยต้องแปลงเป็น float ก่อน จากนั้นนำผลรวมมาต่อกับ x[0] ได้ลิสต์ย่อยเก็บใส่ลิสต์ใหม่ d1</p> <p>สั่ง run, ใส่ข้อมูลตามตัวอย่างแรก, แล้ว print(d1) ได้</p> <pre>[[5.0, 'A'], [9.0, 'B'], [5.0, 'C']]</pre>

โปรแกรม	คำอธิบาย
<pre> for k in range(len(d1)-1):     for i in range(len(d1)-1):         if d1[i][0] &lt; d1[i+1][0] \         or (d1[i][0] == d1[i+1][0] \         and d1[i][1] &gt; d1[i+1][1]):             d1[i],d1[i+1] = \             d1[i+1],d1[i] for [total,sid] in d1:     print( sid, total ) </pre>	<p>ขั้นตอนที่ 4.2 และ 4.3 : ต้องการเรียงลำดับตามคะแนนรวม แต่ใช้คำสั่ง <code>d1.sort()</code> ไม่ได้ เพราะจะเรียงข้อมูลจากน้อยไปมาก สิ่งที่ต้องการคือเรียงตามคะแนนจากมากมาน้อย แต่ใช้คำสั่ง <code>d1.sort(reverse=True)</code> ไม่ได้ เพราะถ้ากรณีที่คะแนนเท่ากัน จะเรียงรหัสนักเรียนจากมากมาน้อย ซึ่งไม่ตรงที่โจทย์ต้องการ</p> <p>จึงขอเขียนการเรียงลำดับเอง (ด้วยวิธี bubble sort ที่เรียนมา) โดยปรับคำสั่งการเปรียบเทียบ คือ จะสลับข้อมูลสองตัวที่ติดกันเมื่อ</p> <ul style="list-style-type: none"> <li>• คะแนนรวมของตัวซ้ายน้อยกว่าคะแนนรวมของตัวขวา หรือ</li> <li>• ถ้าคะแนนรวมของทั้งสองตัวเท่ากัน รหัสนักเรียนของตัวซ้ายมากกว่ารหัสนักเรียนของตัวขวา</li> </ul> <p>เมื่อเรียงลำดับเสร็จ ก็นำข้อมูลใน <code>d1</code> มาแสดงตามที่โจทย์กำหนด สั่ง <code>run</code>, ใส่ข้อมูลตามตัวอย่างแรก, ได้ผลถูกต้อง</p> <p>B 9.0 A 5.0 C 5.0</p>
<pre> else :     d1 = list()     for x in d0 :         s = 0         for e in x[1:] :             s += float(e)         d1.append( [-s, x[0]] )     d1.sort()     for sumsc,sid in d1 :         print( sid, -sumsc ) </pre>	<p>สำหรับขั้นตอนที่ 4.1 ถึง 4.3 เราอาจเปลี่ยนการจัดเก็บข้อมูลของ <code>d1</code> คือแทนที่ลิสต์ข้างในจะเป็น [คะแนนรวม, รหัสனிสิต] เปลี่ยนมาเก็บเป็น [-คะแนนรวม, รหัสனிสิต] แล้วใช้คำสั่ง <code>d1.sort()</code> เพื่อเรียงข้อมูลจากน้อยไปมาก เมื่อคะแนนรวมมีค่ามากติดลบคะแนนรวมจะมีค่าน้อย กลายเป็นต้องการเรียงคะแนนรวมจากมากไปน้อย และถ้าเท่ากัน ก็เรียงรหัสனிสิตจากน้อยไปมาก ตามต้องการ ตอนแสดงคะแนนรวม ก็ต้องแสดงค่าติดลบของค่าที่เก็บด้วย (เพราะที่เก็บเป็นค่าติดลบ)</p>

ลองเขียนโปรแกรมนี้นี้ใหม่ ถ้าเราเปลี่ยนการจัดเก็บข้อมูลจาก

[ [รหัสนักเรียน, คะแนน, ..., คะแนน], [รหัสนักเรียน, คะแนน, ..., คะแนน], ... ]

เป็น

[ [รหัสนักเรียน, [คะแนน, ..., คะแนน] ], [รหัสนักเรียน, [คะแนน, ..., คะแนน] ], ... ]

CONFIDENTIAL

ดร. วิจารณ์  
จิรพัฒน์กุล

Intania 87

กรรมการผู้จัดการ  
บริษัท สคูติโอ จำกัด,

อดีตนักวิทยาศาสตร์ข้อมูล (Data Scientist) ที่ Facebook



ความสามารถในการเขียนโปรแกรมเป็นเหมือนพลังวิเศษที่ช่วยเพิ่มขีดความสามารถในการทำงานของเรา ไม่ว่าจะเป็นการแก้ปัญหาทางวิศวกรรมที่ซับซ้อน (simulation, optimization) การสั่งให้คอมพิวเตอร์ทำงานที่ซ้ำซากน่าเบื่อแทนเรา (automation) การสอนให้คอมพิวเตอร์มีความเฉลียวฉลาดและช่วยเราตัดสินใจได้ (artificial intelligence) รวมไปถึงการสร้างเว็บหรือแอปที่เป็นประโยชน์กับผู้คนในวงกว้าง เรียกได้ว่าเป็นทักษะที่ขาดไม่ได้สำหรับนวัตกรรมในยุคนี้

## 06.3 : List Comprehension

### สรุปเนื้อหา

List comprehension เป็นวิธีการสร้างลิสต์ที่เขียนได้สั้นและทำงานได้รวดเร็ว

### รูปแบบการสร้างลิสต์ด้วย list comprehension ที่พบบ่อย

สร้างลิสต์ด้วยวงวนเพิ่มข้อมูล	สร้างลิสต์ด้วย list comprehension
<b>map:</b> นำข้อมูลจากลิสต์หนึ่งมาประมวลผลเก็บใส่อีกลิสต์ เช่น สร้างลิสต์ b เก็บเฉพาะหลักหน่วยของจำนวนในลิสต์ a <pre>b = [] for e in a :     b.append( e%10 )</pre>	<b>map:</b> <pre>b = [ e%10 for e in a ]</pre>
<b>filter:</b> เลือกข้อมูลจากลิสต์หนึ่งมาใส่อีกลิสต์ เช่น สร้างลิสต์ b เก็บสตริงจากลิสต์ a เฉพาะตัวที่ยาวเกิน 5 <pre>b = [] for e in a :     if len(e) &gt; 5 :         b.append( e )</pre>	<b>filter:</b> <pre>b = [e for e in a if len(e)&gt;5]</pre>
<b>map &amp; filter:</b> ผสมการประมวลผลสองแบบ เช่น นำข้อมูลความสูง (เป็นนิ้ว) ที่เกิน 10 ในลิสต์ a มาแปลงเป็นเซนติเมตรเก็บใส่ลิสต์ b <pre>b = [] for e in a :     if e &gt; 10 :         b.append( 2.54*e )</pre>	<b>map &amp; filter:</b> <pre>b = [2.54*e for e in a if e&gt;10]</pre>
สร้างลิสต์ที่มีการแจกแจงด้วยหลายวงวนซ้อนกันก็ได้ เช่น เลือกจำนวนเต็มที่มีค่าระหว่าง 0 ถึง 20 สองตัวที่มีผลรวมเท่ากับผลคูณ <pre>c = [] for a in range(0,21) :     for b in range(a,21) :         if a+b == a*b :             c.append( [a,b] )</pre>	<pre>c = [[a,b] for a in range(0,21) \         for b in range(a,21) \         if a+b == a*b ]</pre>

### ตัวอย่าง list comprehension

```
x = [e for e in a] เหมือนกับ x = list(a)
ได้ลิสต์ x เป็นลิสต์ใหม่มีค่าภายในเหมือนกับของ a x กับ a เป็นลิสต์คนละตัวกันแต่มีค่าเหมือนกัน
แต่ถ้าเขียน x = a จะได้ x กับ a เป็นลิสต์เดียวกัน การเปลี่ยนค่าในลิสต์ x จะทำให้ a เปลี่ยนด้วย
หรือการเปลี่ยนค่าในลิสต์ a ก็เปลี่ยน x เช่นกัน
```



```
x = [int(e) for e in input().split()]
```

อ่านสตริงจากแป้นพิมพ์ ด้วย `input()` แยกออกเป็นลิสต์ของสตริงด้วย `split()`  
นำแต่ละสตริงในลิสต์มาเปลี่ยนเป็นจำนวนเต็ม เก็บใส่ลิสต์ `x`

```
t = ','.join( [str(e) for e in x] )
```

ใช้ list comprehension สร้างลิสต์ของสตริงที่นำข้อมูลจากลิสต์ `x` มาแปลง แล้วส่งผลลัพธ์ที่ได้ไป `join` กันอีกที

```
c = sum( [ 1 for e in x if e%2 == 0 ] )
```

นับว่าลิสต์ `x` มีจำนวนคู่กี่ตัว โดยสร้างลิสต์ที่เพิ่มเลข 1 ทุกครั้งที่พบจำนวนคู่ในลิสต์ `x` แล้วก็หาผลรวมด้วย `sum`

```
b = [ (1 if x[i] >= 0 else -1) for i in range(len(x)) ]
```

ให้สังเกตการใช้ค่าสลับ `(1 if x[i] >= 0 else -1)` ที่ได้ผลเป็น 1 ถ้า `x[i] >= 0` ไม่เช่นนั้นได้ผลเป็น -1  
จึงเป็นการสร้างลิสต์ `b` โดยให้ `b[i] = 1` ถ้า `x[i] >= 0` ไม่เช่นนั้นให้ `b[i] = -1`

```
x = [ [e for e in range(3)] for k in range(4) ]
```

 เหมือน  

```
x = [ list(range(3)) for k in range(4) ]
```

 เหมือน  

```
x = [ [0,1,2], [0,1,2], [0,1,2], [0,1,2] ]
```

 เหมือน  

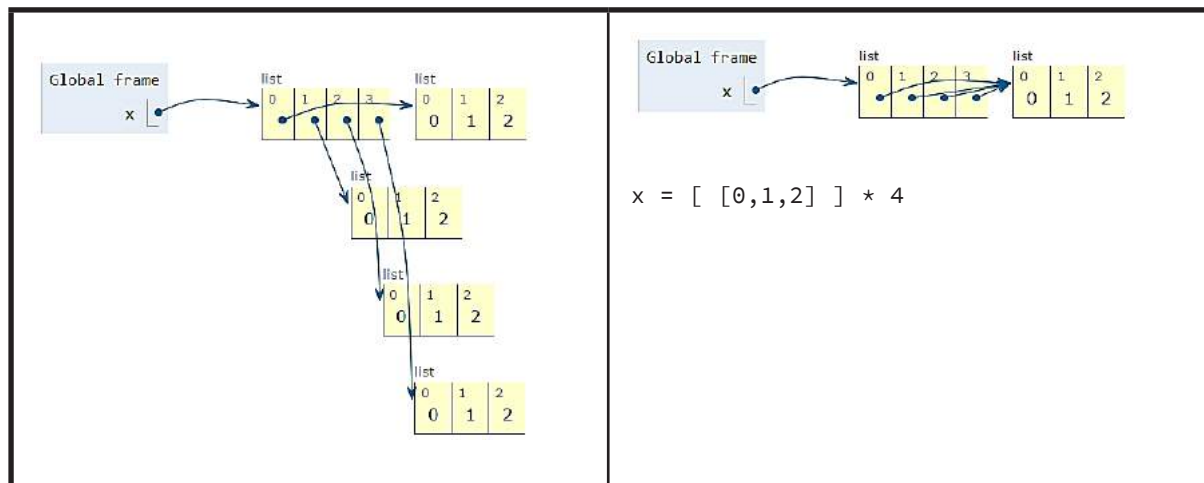
```
x = [ [0,1,2] for k in range(4) ]
```

 เหมือนรูปข้างล่างนี้ทางซ้าย แต่ไม่เหมือนกับ  

```
x = [ [0,1,2] ] * 4
```

 คือรูปข้างล่างนี้ทางขวา

แบบหลังสุดนี้แปลตรงที่ `x[0]`, `x[1]`, `x[2]` และ `x[3]` คือลิสต์ตัวเดียวกันหมด ถ้าทำ `x[0][0] = 9`  
จะได้ `x` มีค่าเป็น `[[9,1,2], [9,1,2], [9,1,2], [9,1,2]]` ในขณะที่แบบอื่น ๆ ลิสต์ข้างในเป็นคนละตัวกันหมด



```
a = [ [x**2+y**2, x, y] for [x,y] in data ]
```

```
a.sort()
```

```
d = [ [x,y] for [s,x,y] in a ]
```

`data` เป็นลิสต์ซ้อนลิสต์ โดยที่ลิสต์ภายในแต่ละตัวเก็บพิกัด `x` กับ `y`  
ต้องการเรียงลำดับจุดเหล่านี้ตามระยะห่างของจุดถึงจุดกำเนิดจากน้อยไปมาก  
สร้างลิสต์ซ้อนลิสต์ชั่วคราว `a` มีลิสต์ภายในเก็บระยะห่าง ตามด้วยพิกัด `x` กับ `y`  
จากนั้น `sort` ลิสต์ชั่วคราวนี้ ซึ่งคือการเรียงลำดับตามระยะห่างจากน้อยไปมาก  
ปิดท้ายด้วยการใช้ list comprehension หยิบข้อมูลตามลำดับใน `a` มาสร้างลิสต์ใหม่ที่มีแต่พิกัด `x, y` เก็บใส่ `d`

## เรื่องผิดพลาด

<p>ไม่ควรใช้ list comprehension เพื่อให้ทำงานแบบวงวน แต่ให้นำมาจากการสร้างลิสต์ด้วย list comprehension ไปใช้งานเลย</p>	<p>ไม่ควรเขียน <code>[print(k) for k in range(5)]</code> ถึงแม้ว่าจะแสดงค่า 0 1 2 3 4 (บรรทัดละค่า) ตามต้องการ เนื่องจากมีการสร้างลิสต์ แล้วไม่ได้ใช้ เสียเวลาเปล่า ๆ หรือเขียน <code>[t.append(e) for e in x if e not in t]</code> เพื่อนำข้อมูลใน x ที่ไม่ปรากฏใน t ไปเพิ่มใน t จะเกิดการสร้างลิสต์ <code>[None, None, ...]</code> แล้วทิ้งไป ควรเขียนแบบวงวนปกติ</p> <pre>for e in x :     if e not in t :         t.append(e)</pre>
<p>อย่านำค่าของตัวแปรใน for ของ list comprehension มาใช้งานนอกคำสั่ง อาจทำให้สับสน</p>	<pre>x = [e for e in range(10)] print(e)                # ผิด เพราะ e ไม่มีค่า</pre> <p>แต่ถ้าเขียน</p> <pre>x = [] for e in range(10) :     x.append(e) print(e)                # ได้ 9</pre> <p>แต่ถ้าแบบนี้</p> <pre>e = 99 x = [e for e in range(10)] print(e)                # ได้ 99</pre>

## แบบฝึกหัด

Problem	Code
<p><u>Input</u>: มีลิสต์ x เก็บสตริง และตัวแปร c เก็บตัวอักษร</p> <p><u>Process</u>: สร้างลิสต์ d เก็บจำนวนครั้งที่ตัวอักษรใน c ปรากฏในแต่ละสตริงของลิสต์ x</p> <p>เช่น <code>x = ['abba', 'babana', 'ann']; c = 'a'</code></p> <p>จะได้ <code>d = [2,3,1]</code></p>	
<p><u>Input</u>: ลิสต์ x เก็บจำนวนเต็ม</p> <p><u>Process</u>: ลบจำนวนเต็มใน x ทุกตัวที่ติดลบ</p>	

Problem	Code
<p>Input: x เป็น list of lists of integers</p> <p>Process: หาผลรวมของจำนวนเต็มใน x</p>	
<p>Input: รับหนึ่งบรรทัดที่มีจำนวนเต็มหลายจำนวน (คั่นด้วยช่องว่าง) จากแป้นพิมพ์</p> <p>Process: หาว่าที่รับมามีจำนวนคี่, เลขี่จำนวน</p>	
<p>Input: รับข้อความ หนึ่งบรรทัดจากแป้นพิมพ์</p> <p>Process: ตัดอักขระทุกตัวในข้อความที่รับมาที่ไม่ใช่พยัญชนะภาษาอังกฤษ</p>	
<p>Input: รับหนึ่งบรรทัดที่มีจำนวนเต็มหลายจำนวน (คั่นด้วยช่องว่าง) ชุดหนึ่ง เก็บใส่ลิสต์ x และรับอีกหนึ่งบรรทัดเก็บใส่ลิสต์ y ในทำนองเดียวกัน โดยลิสต์ทั้งสองมีจำนวนข้อมูลเท่ากัน</p> <p>Process: สร้างลิสต์ z โดยที่ <math>z[i]</math> มีค่าเท่ากับ <math>x[i]+y[i]</math></p>	
<p>Input: เมทริกซ์ m แทนด้วย list of lists of integers</p> <p>Process: แปลง m ให้กลายเป็น list of integers เช่น จาก <math>m = [[1,2,3],[4,5,6]]</math> กลายเป็น <math>[1,2,3,4,5,6]</math></p>	
<p>Input: รับหนึ่งบรรทัดที่มีจำนวนเต็มหลายจำนวน (คั่นด้วยช่องว่าง) เก็บใส่ลิสต์ x</p> <p>Process: สร้างลิสต์ใหม่จากข้อมูลใน x แต่ไม่มีตัวซ้ำ โดย</p> <ol style="list-style-type: none"> <li>เรียงลำดับข้อมูลใน x จากน้อยไปมาก</li> <li>สร้างลิสต์ใหม่โดยนำข้อมูลในลิสต์ตัวที่ติดกันมาพิจารณาจากซ้ายไปขวา ถ้าค่าของตัวติดกันตัวซ้ายไม่เท่ากับตัวขวาให้นำตัวซ้ายเก็บใส่ลิสต์ใหม่</li> <li>นำตัวขวาสุดใน x เพิ่มต่อท้ายลิสต์ใหม่นี้</li> </ol> <p>ลิสต์ใหม่นี้ก็จะเก็บข้อมูลใน x ที่ไม่มีตัวซ้ำ</p>	
<p>Input: รับจำนวนเต็ม n</p> <p>Process: สร้างลิสต์ c ที่เก็บจำนวนประกอบที่มีค่าน้อยกว่า n เริ่มจากสร้างลิสต์ใหม่ x โดย</p> <ul style="list-style-type: none"> <li>เพิ่มค่า 4,6,8,10,... (ไม่เกิน n-1) ใน x</li> <li>เพิ่มค่า 6,9,12,15,... (ไม่เกิน n-1) ใน x</li> <li>เพิ่มค่า 8,12,16,20,... (ไม่เกิน n-1) ใน x</li> <li>...</li> </ul> <p>ไปเรื่อย ๆ トラバเท่าที่ยังไม่เกิน n-1 นั่นคือ</p> <p>เพิ่มค่า 2k,3k,4k,5k,... (ไม่เกิน n-1) ใน x</p> <p>โดยแปรค่า <math>k = 2,3,4,\dots,N//2-1</math></p> <p>จากนั้นสร้างลิสต์ใหม่ c ที่ได้ข้อมูลจาก x แต่ไม่มีตัวซ้ำ (ใช้วิธีที่เขียนในข้อที่แล้ว)</p>	

Problem	Code
<p>Input: รับจำนวนเต็ม n</p> <p>Process: สร้างลิสต์ที่เก็บจำนวนเฉพาะที่มีค่าน้อยกว่า n โดย</p> <ol style="list-style-type: none"> <li>1. สร้างลิสต์ที่เก็บจำนวนประกอบที่มีค่าน้อยกว่า n</li> <li>2. สร้างลิสต์ที่เก็บจำนวนเฉพาะตั้งแต่ 2 ถึง n-1 แต่ไม่เอาจำนวนประกอบที่เข้าในขั้นตอนที่ 1</li> </ol>	



## เรียงตามคะแนนรวม

ขอใช้ปัญหาเรียงคะแนนรวมในหัวข้อลิสต์ซ้อนลิสต์ มาเป็นตัวอย่าง โดยนำโปรแกรมที่ได้เขียนไปแล้ว มาปรับปรุงโดยใช้ list comprehension

ตัวอย่างการเขียนโปรแกรม

โปรแกรมเดิม	โปรแกรมปรับปรุงแบบใช้ list comprehension
<pre> n,m = [int(e) for e in \         input().split()] d0 = list() for k in range(n) :     d0.append(input().split()) err = list() for x in d0 :     if len(x[1:]) != m :         err.append(x[0]) if len(err) &gt; 0 :     print('Invalid data: ')     for sid in err :         print(sid) else :     d1 = list()     for x in d0 :         s = 0         for e in x[1:] :             s += float(e)         d1.append( [-s, x[0]] )     d1.sort()     for ntotal,scode in d1 :         print( scode, -ntotal ) </pre>	<pre> n,m = [int(e) for e in \         input().split()] d0 = [input().split() for k in range(n)]  err = [x[0] for x in d0 if len(x[1:]) != m]  if len(err) &gt; 0 :     print('Invalid data: ')     print('\n'.join(err)) else :     # list comprehension ซ้อนใน     # list comprehension อีกชั้น     d1 = [[-sum([float(e) for e in x[1:])), \             x[0]] for x in d0]      d1.sort()     for ntotal,scode in d1 :         print( scode, -ntotal ) </pre>

## ตัวอย่างโจทย์ปัญหา

### Minus All

โจทย์ข้อนี้ให้นิสิตเขียนโปรแกรมรับชุดตัวเลขจำนวนเต็มชุดหนึ่ง ซึ่งจะจบด้วยเลขที่มีค่าติดลบ จากนั้นให้แสดงผลชุดตัวเลขใหม่ (ไม่รวมตัวเลขที่ติดลบ) ที่สมาชิกแต่ละตัวมีค่าเท่ากับค่าเดิมรวมกับค่าที่ติดลบนั้น

#### ► ข้อมูลนำเข้า

เป็นชุดของตัวเลข แต่ละบรรทัดจะประกอบด้วยตัวเลขจำนวนเต็มที่ไม่ติดลบ 1 จำนวน และบรรทัดสุดท้ายจะจบด้วยตัวเลขจำนวนเต็มที่มีค่าติดลบ (รับประกันว่า จะมีตัวเลขจำนวนเต็มที่ไม่ติดลบอย่างน้อย 1 จำนวน)

#### ► ข้อมูลส่งออก

ให้แสดงชุดตัวเลขใหม่ (ไม่รวมตัวเลขที่ติดลบ) ที่สมาชิกแต่ละตัวมีค่าเท่ากับค่าเดิมรวมกับค่าที่ติดลบ โดยแสดงผลบรรทัดละ 1 จำนวน

#### ► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
1 10 2 -5	-4 5 -3
100 0 -1	99 -1
3 -3	0

# คำนวณหา rank

จงเขียนโปรแกรมเพื่อคำนวณหา rank ของรหัสสินค้าที่กำหนด เมื่อเรียงลำดับตามคะแนน (rank 1 คือมีคะแนนมากที่สุด)

## ► ข้อมูลนำเข้า

แต่ละบรรทัดจะระบุข้อมูลสินค้าแต่ละตัว ประกอบด้วยรหัสสินค้า ตามด้วยคะแนนเป็นเลขทศนิยม

บรรทัดสุดท้าย ระบุรหัสสินค้าที่ต้องการคำนวณหา rank

ในการคำนวณ rank หากมีสินค้าได้คะแนนเท่ากัน ให้เรียงลำดับตามรหัสสินค้า (เรียงแบบจำนวนเต็ม)

## ► ข้อมูลส่งออก

มีบรรทัดเดียว แสดง rank ของสินค้าที่ต้องการค้นหา หากไม่พบรหัสสินค้านี้ดังกล่าว ให้แสดงว่า Not Found

## ► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
5931111121 87.25 5932222221 77.00 5933333321 82.50 5934444421 69.75 5935555521 66.00 5934444421	4
111 100 13 96 1234 96 555 99 2121 96 99 99 1234	5
429801 78 359124 89 902316 91.25 773842 45.75 264336	Not Found

# ความถี่เกินครึ่ง

ให้เขียนโปรแกรมเพื่อหาข้อมูลที่มีความถี่มากกว่าครึ่งหนึ่งของจำนวนข้อมูลทั้งหมดที่รับเข้ามา โดยอ่านค่าจำนวนเต็มจนกระทั่งพบ -1

## ► ข้อมูลนำเข้า

ให้อ่านข้อมูลจำนวนเต็มบรรทัดละ 1 จำนวน จนกระทั่งพบค่า -1

## ► ข้อมูลส่งออก

มีบรรทัดเดียว แสดงข้อมูลที่มีความถี่มากกว่าครึ่งหนึ่งของข้อมูลทั้งหมด (ไม่รวม -1) ถ้าไม่มีข้อมูลที่มีความถี่มากกว่าครึ่งหนึ่งเลย ให้แสดง Not found

## ► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
4 5 5 10 -1	Not found
4 5 5 -1	5
10 5 5 3 5 -1	5
2 4 2 4 2 4 -1	Not found

## 07 : Tuple, Dictionary and Set

### สรุปเนื้อหา

#### Tuple

สร้าง tuple เหมือนสร้าง list แต่ใช้วงเล็บโค้ง	<pre>my_list = [1, 2.5, 3, 'A']    ได้ list my_tuple = (1, 2.5, 3, 'A')   ได้ tuple my_tuple_2 = (1,)             ได้ tuple ที่มีตัวเดียว (สังเกตที่ comma) not_a_tuple = (1)             ได้จำนวนเต็ม เหมือน not_a_tuple = 1 t1 = tuple('abc')             สร้างจากสตริง ได้ ('a','b','c') t2 = tuple([1, 2, 3])         สร้างจากลิสต์ ได้ (1,2,3)</pre>
การเข้าถึงข้อมูลทำเหมือน list	<pre>my_tuple[0]    ได้ 1 my_tuple[1:3]   ได้ (2.5, 3) my_tuple[-2:-1] ได้ (3,)</pre>
เครื่องหมาย + ใช้ต่อ tuple เครื่องหมาย * ใช้ต่อ tuple หลายครั้ง	<pre>a = (2, 3, 5) b = (7,) p = a + b    ได้ (2, 3, 5, 7) q = p*2      ได้ (2, 3, 5, 7, 2, 3, 5, 7) a += (2,)    เหมือนกับเขียน a = a + (2,)               คือนำ a เดิมมาต่อกับ (2,) สร้าง tuple ใหม่               ได้ (2, 3, 5, 2) แล้วให้ค่า tuple นี้กับตัวแปร a</pre>
แก้ไขข้อมูลใน tuple ไม่ได้ (เหมือนสตริง)	<pre>my_tuple[3] = 'B' ผิด ถ้าต้องการเปลี่ยน ต้องตัดต่อสร้างใหม่ my_tuple = my_tuple[:3] + ('B',) + my_tuple[4:]</pre>
ระบบจะแปลงสิ่งที่มี comma คั่น ให้กลายเป็น tuple เช่น t = 1,2,3 เมื่อ print(t) ได้ (1, 2, 3)	<pre>a,b,c = (1,2,3)    ได้ a = 1, b = 2, c = 3 x,y = 'A','Z'      ได้ x = 'A', y = 'Z' a,x = x,a           เป็นการสลับค่าในตัวแปร a กับ x</pre>
สามารถสร้าง list ของ tuple ได้	<pre>L = [('A','B'), (1,20,300), (9.9,)] เป็น list ของ tuple L[1]    ได้ (1,20,300) L[1][2] ได้ 300</pre>



สามารถสร้าง tuple ของ list ได้	<pre>T = ('A','B', [1,20,300], [9.9]) เป็น tuple ของ list T[1] ได้ [1,20,300] T[1][2] ได้ 300 T เป็น tuple จึงเปลี่ยน T[1] = 9 ไม่ได้ แต่เนื่องจาก T[1] เป็นลิสต์ จึงเปลี่ยน T[1][2] = 9 ได้</pre>
สามารถสร้าง tuple ของ tuple ก็ได้	<pre>T = ((1,2), (3,4)) เป็น tuple ของ tuple T[1][0] ได้ 3</pre>
<p>ข้อแตกต่างระหว่าง</p> <p>tuple1 += (0,) กับ list1 += [0]</p> <p>tuple1 += (0,) เหมือน</p> <p>tuple1 = tuple1 + (0,)</p> <p>จะสร้าง tuple ใหม่ (ทำงานช้า)</p> <p>ในขณะที่ list1 += [0]</p> <p>ไม่เหมือนกับ list1 = list1 + [0]</p> <p>แต่จะเหมือนกับ list1.append(0)</p> <p>เป็นการเพิ่ม 0 ต่อท้ายลิสต์ ไม่สร้างลิสต์ใหม่</p> <p>จะทำงานได้เร็วกว่า</p>	<p>ลองสั่งทำงานโปรแกรมทางขวานี้ แล้วสังเกตเวลาการทำงาน</p> <pre>import time n = 100000  t0 = time.time() list1 = [] for k in range(n):     list1 += [0] print(time.time()-t0)  t0 = time.time() tuple1 = tuple() for k in range(n):     tuple1 += (0,) print(time.time()-t0)</pre>

## Dictionary

- ใช้เก็บข้อมูลแบบคู่อันดับ (key, value)
- ไม่มี key ที่ซ้ำกัน หนึ่ง key มี value ที่คู่กันเพียงค่าเดียว (แต่ value อาจเป็น list, tuple, ... ที่เก็บข้อมูลย่อย ๆ ได้)
- ถ้าเรามี key จะสามารถหา value ที่คู่กับ key ได้เร็วมาก ๆ

การสร้าง dict ทำได้หลายแบบ อาจจะระบุข้อมูลที่ต้องการเก็บด้วยก็ได้	<pre>dict1 = {} หรือ dict1 = dict() เป็นการสร้าง dict ว่าง ๆ  dict2 = {'Name':'Tom', 'Age':39} เป็นการสร้าง dict ที่เก็บคู่อันดับ ('Name','Tom') และ ('Age',39)</pre>
สามารถเรียกใช้ แก้ไข และเพิ่ม value โดยการอ้างถึงด้วย key	<pre>D = {'Name':'Sarah'} print(D['Name']) ได้ Sarah  D['Age'] = 20 เป็นการเพิ่มข้อมูล ('Age',20)  D['Name'] = 'Somsri' เป็นการแก้ไขข้อมูลให้เป็น ('Name','Somsri')</pre>

key ของ dict อาจเป็นจำนวนเต็ม จำนวนจริง สตริง หรือ tuple ก็ได้ แต่ห้ามเป็น list, dict หรือ set ส่วน value ที่คู่กับ key จะเป็นอะไรก็ได้	<pre> my_dict = {} my_dict[1] = 8.00 my_dict[3.75] = [3, 7, 5] my_dict['Hello'] = 'World' my_dict[(2,99)] = 999 my_dict[[1,2,3]] = '123' ผิด ห้ามใช้ list เป็น key </pre>
มีบริการ keys(), values(), items() ซึ่งใช้กับ for...in ได้	<pre> d = {1:7, 2:8, 3:9} for e in d :          แจกแจกแต่ละ key ของ d ให้กับ e for e in d.keys()     แจกแจกแต่ละ key ของ d ให้กับ e for e in d.values()   แจกแจกแต่ละ value ของ d ให้กับ e for e in d.items()    แจกแจกแต่ละคู่ข้อมูลของ d เป็น                     tuple ให้กับ e                     e[0] คือ key, e[1] คือ value for k,v in d.items()  แจกแจก key, value ของ d                     ให้กับ k, v </pre> <p>ลำดับของข้อมูลที่แจกแจกออกมาไม่จำเป็นต้องเหมือนกับที่เขียนหรือที่เพิ่ม</p>
บริการ keys(), values() และ items() ไม่ได้ผลเป็น list แต่เป็น อะไรบางอย่างที่คล้าย list นำไปใช้กับ for...in ได้ แต่ใช้ index เพื่อหยิบ ข้อมูลไม่ได้	<pre> d = {'aka':'as known as', 'so':'significant other'} x = d.keys()  print(x[1]) ผิด ทำไม่ได้ เพราะ x ไม่ใช่ลิสต์ ถ้าต้องการขอ key ทั้งหมดให้เป็นลิสต์จริง ๆ ต้องเขียน x = list( d.keys() ) ได้ x เป็นลิสต์ ใช้ x[1] ได้ print(x[1]) ทำได้ (แต่ไม่รู้ว่าจะได้ 'aka' หรือ 'so') </pre>
บริการ d.update(d1) เป็นการเพิ่ม ข้อมูลใหม่จาก dict d1 เข้าใน d ถ้ามี key ใน d1 ซ้ำกับใน d จะเป็นการแก้ไข value เดิม	<pre> A = {1:'one', 2:'two'} B = {1:'nueng', 3:'sarm'}  A.update(B) ได้ A เป็น {1:'nueng', 2:'two', 3:'sarm'} ส่วน B ไม่เปลี่ยนแปลง </pre>
บริการ pop(key) เป็นการลบคู่ข้อมูล ที่มี key นั้นออก ถ้าไม่มี key อยู่ จะเกิด error	<pre> C = {1: 'one', 2: 'two', 3: 'three'} x = C.pop(1) ได้ x = 'one', C = {2:'two', 3:'three'} y = C.pop(4) เกิด error </pre>
การใช้ in เพื่อตรวจสอบว่ามี key หรือ value หรือ คู่ (key,value) ใน dict อยู่หรือไม่ โดยการตรวจ value จะไม่เร็วเท่ากับอีกสองกรณี	<pre> D = {1: 'one', 2: 'two', 3: 'three'} x = 1 in D          ได้ x = True y = 4 in D.keys()   ได้ y = False z = 'two' in D.values() ได้ z = True w = (2,'two') in D.items() ได้ w = True </pre> <p>การค้นใน keys และ items เร็วมาก แต่การค้นใน values ไม่เร็วเท่า</p>