



python

၁၀၉



ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

CONFIDENTIAL

PYTHON ๙๐๙

สิงหาคม ๒๕๖๑

v1.0.2

ผู้อ่านสามารถดาวน์โหลดหนังสือรุ่นล่าสุด
และร่วมแสดงความคิดเห็น/ข้อแนะนำเกี่ยวกับหนังสือเล่มนี้ที่

www.cp.eng.chula.ac.th/books/python101

ขอบคุณครับ

กิตติภณ พลพาร, กิตติภพ พลพาร, สมชาย ประสิทธิจตระ, สุรี สินธุวิญญา

PYTHON ๑๐๑ / กิตติภณ พลพาร, กิตติภพ พลพาร, สมชาย ประสิทธิจตระกูล, สุรี สินธุวิญญา

1. การเขียนโปรแกรม (คอมพิวเตอร์)
2. ไฟฟอน (คอมพิวเตอร์)

005.133

ISBN 978-616-407-102-6

พิมพ์ครั้งที่ 1 จำนวน 1,000 เล่ม พ.ศ. 2560

พิมพ์ครั้งที่ 2 จำนวน 1,000 เล่ม พ.ศ. 2561

ส่วนนลิขสิทธิ์ตาม พ.ร.บ. ลิขสิทธิ์ พ.ศ. 2537/2540

การผลิตและการลอกเลียนหนังสือเล่มนี้ไม่ว่ารูปแบบใดทั้งสิ้น
ต้องได้รับอนุญาตเป็นลายลักษณ์อักษรจากเจ้าของลิขสิทธิ์

จัดพิมพ์โดย

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

พญาไท กรุงเทพฯ 10330

<https://www.cp.eng.chula.ac.th>

เข้าชมหนังสือเล่มอื่น ๆ ของภาควิชาได้ที่

<https://www.cp.eng.chula.ac.th/books>

ออกแบบปก : กมลพรรณ ลี้ประเสริฐ

ออกแบบรูปเล่ม : ภานุกร สุนทรเวชพงษ์

พิมพ์ที่ โรงพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย โทรศัพท์ 0-2218-3549-50 โทรสาร 0-2218-3551

คำนำ

วิชา ๒๑๑๐๑๐๑ ภาษาไทย โปรแกรมคอมพิวเตอร์ เป็นหนึ่งในวิชาพื้นฐานทางวิศวกรรมศาสตร์ ที่นิสิตชั้นปีที่ ๑ คณะวิศวกรรมศาสตร์ ทุกคนต้องลงทะเบียนเรียน วัตถุประสงค์หลักของวิชานี้คือ ให้นิสิตเข้าใจหลักการในการใช้คำสั่งต่าง ๆ ของภาษาโปรแกรม เนื่องจากเป็นภาษาที่ตรงตามข้อกำหนดที่ได้รับ การเขียนโปรแกรมเป็นความสามารถที่ต้องลงมือฝึกฝนฝึกปฏิบัติด้วยตนเอง เมื่อกันทักษะอื่นทางวิศวกรรมที่จำเป็นต้องฝึก ๆ ให้ชำนาญจะจะได้ผล ไม่สามารถได้มาด้วยการอ่าน ๆ ๆ ๆ ๆ

หนังสือ PYTHON ๑๐๑ เล่มนี้ถูกจัดทำขึ้น เพื่อให้นิสิตใช้ทบทวนเนื้อหาหลังจากพยนตร์บรรยายเนื้อหาด้วยตนเอง ในแต่ละบท ใช้เตรียมตัวก่อนเข้าเรียน และใช้ระหว่างการเขียนโปรแกรมจริงในห้องปฏิบัติการที่จัดขึ้นเป็นกิจกรรมประจำทุกสัปดาห์ โดยมีระบบ Grader ช่วยตรวจสอบความถูกต้องของผลการทำงานของโปรแกรมอย่างอัตโนมัติ นิสิตจะได้ฝึกเขียนโปรแกรมตามโจทย์ ฝึกหาที่ผิด และฝึกตรวจสอบความถูกต้องของโปรแกรมที่พัฒนาขึ้น นอกจากนี้ ยังมีแบบฝึกปฏิบัติเพิ่มเติมอีกมากmany ในระบบ Grader ให้นิสิตได้ทำเสริมอีกด้วย

ผู้เขียนต้องขอขอบคุณ ผศ. ดร. นพที นิภานนท์ ผู้ปรับปรุงระบบตรวจโปรแกรมอัตโนมัติ Grader เพื่อใช้ประกอบการเรียน การสอน และการสอบวิชาการเขียนโปรแกรมมาตั้งแต่ปีการศึกษา ๒๕๕๗ ขอขอบคุณบุคลากรของศูนย์คอมพิวเตอร์ คณะวิศวกรรมศาสตร์ ที่ให้บริการจัดเตรียมความพร้อมของอุปกรณ์และเครื่องข่ายในห้องปฏิบัติการ ขอขอบคุณคณาจารย์และนิสิตช่วยสอนที่ร่วมกันสร้างໂโจทย์ปัญหา สอน และปรับปรุงวิชา ๒๑๑๐๑๐๑ ตลอดมา ขอขอบคุณ ภาควิชางานสุนทรเวชพงษ์ และอดีตคุณ อว.ไอศรุตย์ ที่ช่วยจัดตั้งห้องเรียน และ กมลพรนน ลีวะประเสริฐ ที่ช่วยออกแบบหน้าปก ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ให้การสนับสนุนในสารพัดเรื่อง และท้ายสุดที่ต้องขอบคุณที่สุดก็คือ นิสิตคณะวิศวฯ กว่าหลายพันคนที่ขยันหมั่นศึกษาและฝ่าฟันอุปสรรคในการเรียนวิชาพื้นฐานบังคับที่ค่อนข้างไม่คุ้นเคยนี้จนสำเร็จ *

กิตติภรณ พลการ
กิตติภพ พลการ
สมชาย ประสีห์เจริญตระกูล^(ผู้เรียบเรียงและรวมเนื้อหา)
สุกฤtie สินธุภิญโญ^(ที่ปรึกษา)

ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย
๑๒ สิงหาคม ๒๕๖๐

* ภาควิชาวิศวกรรมคอมพิวเตอร์ขอขอบคุณ บริษัท เอ็กซอนโมบิล จำกัด ที่ให้การสนับสนุนค่าใช้จ่ายสำหรับการจัดพิมพ์ให้กับนิสิตทุกคนที่ลงทะเบียนเรียนวิชา ๒๑๑๐๑๐๑ ใช้ประกอบการเรียนในปีการศึกษา ๒๕๖๐ และ ๒๕๖๑

ขอขอบคุณ

คณาจารย์คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ผศ. ดร. ชัยเชษฐ์ สายวิจิตร

ศ. ดร. ไพรัล กิตติศุภกร

รศ. ดร. อาณัติ เรืองรัศมี

รศ. ดร. พิสุทธิ์ เพียรมานะ

อ. ดร. กรวิก ตนัคชร บุนท์

อ. ดร. พรรณี แสงแก้ว

รศ. ดร. ณัฐชา ทวีแสงสกุลไทย

อ. ดร. เชษฐา พันธ์เครือบุตร

อ. ดร. สุรัช ขวัญเมือง

ผศ. ดร. อนุรักษ์ ศรีอริยวัฒน์

ผศ. ดร. จิรวัฒน์ ชีรุ่งโรจน์

ผศ. ดร. ณัฐวุฒิ หนูโพโรจน์

ศิษย์เก่าภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

คุณโภรัส ใจจนะ

คุณธนาวัฒน์ มาลาบุปผา

คุณลิสา ตรงประกอบ

คุณภัทรารุธ ชื่อสัตยาศิลป์

คุณวีโรจน์ จิรพัฒนกุล

คุณสุภารัชช์ สุตตันทวิบูลย์

คุณศุภารักษ์ ชูชัยศรี

คุณณัฐชา ลีละศุภกุล

ที่กรุณาริห์ความรู้ถึงความสำคัญของการเขียนโปรแกรมกับงานทางวิศวกรรมในสาขาต่าง ๆ

ขอขอบคุณภาพประกอบจาก

<https://commons.wikimedia.org/wiki/File:WomaBallPython.jpg> (ภาพ)

<https://pixabay.com/en/snake-python-tree-pyton-terrarium-1184810/> (ภาพลายหนังงู)

<https://pixabay.com/photo-1745096/> (ภาพพื้นหลังหน้าปก)

<https://www.python.org/community/logos/> (สัญลักษณ์ Python)

<http://www.pythontutor.com> (ภาพประกอบคำอธิบายเนื้อหา)

สารบัญ

00 : Programming in Engineering	1
01 : Data Type, Variable and Expression	5
02 : Selection (if-elif-else)	15
03 : Repetition (while, for)	29
04 : String	43
05 : File	55
06.1 : List	65
06.2 : Nested List	75
06.3 : List Comprehension	83
07 : Tuple, Dictionary and Set	91
08 : Function and Recursion	109
09 : NumPy	121
10 : Class	133
11 : Solutions to Exercises	149
Appendix	157

CONFIDENTIAL

00 : Programming in Engineering

ศาสตร์ของวิศวกรรมทั่วไป คือ ศึกษาศาสตร์ที่ครอบคลุมความหลากหลาย มีการเปลี่ยนแปลงและพัฒนาอย่างมากในปัจจุบัน ในหลายโอกาส การสร้างต้นแบบเสมือนจริงอาจมีความสูง หรือแท็บเป็นไปไม่ได้เลยในช่วงเริ่มต้นของการต่าง ๆ ความสามัคคีในการใช้ทักษะการ Programming เพื่อวัตถุประสงค์ต่าง ๆ เช่น การสร้างแบบจำลองด้วยคอมพิวเตอร์เพื่อการออกแบบ การศึกษาความเป็นไปได้ในการใช้งาน การทดสอบสมรรถนะ หรือแม้กระทั่งการจำลองเพื่อหาเหตุการณ์ที่เกิดความคาดหมายของวิศวกร จึงเป็นสิ่งจำเป็นอย่างมากสำหรับวิศวกรไฟฟ้าทุกสาขาในปัจจุบันและอนาคต

ผศ. ดร. ชัยชาญ สายวิจิต
ภาควิชาวิศวกรรมไฟฟ้า



Programming เป็นเครื่องมือที่ช่วยในการหาคำตอบเพื่ออธิบายปรากฏการณ์ หรือพฤติกรรมพลวัตของหน่วยปฏิบัติการหรือกระบวนการต่าง ๆ นอกจากนี้ยังใช้ประมาณค่าตัวแปรหรือพารามิเตอร์ และพัฒนาไปสู่การทำนายปรากฏการณ์หรือพฤติกรรมพลวัตของหน่วยปฏิบัติการหรือกระบวนการต่อไป

ศ. ดร. ไพบูล กิตติศุภก
ภาควิชาวิศวกรรมเคมี

ในทางวิศวกรรมโยธา โปรแกรมมีใช้กันอย่างมากเพื่อใช้จำลองพฤติกรรมของโครงสร้าง เช่น อาคาร สะพาน ฐานราก ภายใต้การรับแรงซึ่งจะทำให้ทราบแรงที่เกิดกับโครงสร้างเพื่อที่วิศวกรโยธาจะทำการออกแบบโครงสร้างให้ปลอดภัยและประหยัด และในการก่อสร้างก็ต้องใช้โปรแกรมเพื่อวางแผนและจัดการก่อสร้าง นอกจากนั้นก็มีการใช้โปรแกรมในการจำลองระบบขนส่งเพื่อการออกแบบ หรือวิเคราะห์ระบบคมนาคมแบบต่าง ๆ

ศ. ดร. อาณัติ เรืองรัศมี
ภาควิชาวิศวกรรมโยธา



ก ารเรียนวิศวกรรมสิ่งแวดล้อม ปัจจุบัน เรายังต้องคำนวณและสรุปผลวิเคราะห์ค่าพารามิเตอร์ทางสิ่งแวดล้อม ไม่ใช่เรื่องที่ง่าย และก็ไม่ใช่ข้อสอนถึงขนาดต้องเรียนภาคคอมฯ การที่นิสิตได้เรียน Programming เป็นอย่างตื้น จะสามารถช่วยให้นิสิตสามารถต่อยอดความรู้ไปสู่งานวิชาชีวานิสิตได้อย่างง่ายดาย

ศ. ดร. พิสุทธิ์ เพียรบุนกุล
ภาควิชาวิศวกรรมสิ่งแวดล้อม



ก ารเขียนโปรแกรมมีความสำคัญกับงานทุกด้านของวิศวกรรมสำรวจ โดยเฉพาะอย่างยิ่งในด้าน GIS ที่ต้องการการเขียนโปรแกรมในการจัดการ ประมวลผลวิเคราะห์ และแสดงผลข้อมูล โดยเฉพาะข้อมูลขนาดใหญ่ที่ซอฟต์แวร์พื้นฐานโดยทั่วไป เช่น MS Excel ไม่สามารถรับได้ นอกจากนี้ ยังมีความสำคัญต่อการพัฒนาซอฟต์แวร์ด้านแผนที่ โดยเฉพาะระบบแผนที่บนเว็บที่ต้องอาศัยการพัฒนาด้วยการเขียนโปรแกรม และการเขียนโปรแกรมสำคัญมากที่สุดกับการศึกษาในระดับสูงเนื่องจากจะต้องพัฒนาซอฟต์แวร์ใหม่ขึ้นมาใหม่เอง

อ. ดร. กฤติก ตันกษ์ราบันทก
ภาควิชาวิศวกรรมสำรวจ

Programming มีความสำคัญต่อการเรียนในสาขาวิชาวิศวกรรมนิวเคลียร์ค่อนข้างมาก การประมวลผลการตรวจวัดรังสีแบบที่เป็นจำนวนนับรังสี จำนวนข้อมูลการนับรังสี มีจำนวนมากต่อการวัด เครื่องตรวจนับรังสีต้องใช้โปรแกรมคอมพิวเตอร์ในการรวบรวมและประมวลผลข้อมูลมาสร้างเป็นแบบสเปกตรัมข้อมูล สำหรับการประมวลผลที่ได้จากการฉายรังสีแบบภาคตอนทรารสขา-ดำ การถ่ายภาพรังสีแบบที่หล่อผ่านและมีสเต็ปการถ่ายหมุนรอบวัตถุนั้นที่เรียกว่า CT-scan เมื่อร่วมผลของแต่ละสเต็ปมารวมกันจะได้ภาพตัดขวางของวัตถุหรือถ้ามีการวัดในแนวตั้งด้วยทำให้ได้ภาพสุดท้ายเป็นภาพ 3D ซึ่งจะได้เป็นโมเดลที่มีรายละเอียดภายในด้วย โดยเทคนิคนี้ทำให้ได้ข้อมูลภายในของวัตถุหรือแม้แต่ภายในของร่างกายมนุษย์ได้ และยังออกแบบโปรแกรมในการควบคุมอุปกรณ์การตรวจวัดรังสีให้ทำงานได้อย่างอัตโนมัติ และควบคุมได้จากระยะไกล ทำให้ผู้ใช้รับปริมาณรังสีน้อยลง สำหรับระบบการควบคุมการทำงานทั้งหมดของโรงไฟฟ้านิวเคลียร์นั้น ทำโดยโปรแกรมคอมพิวเตอร์ทั้งหมด



อ. ดร. พรรดา แสงแก้ว
ภาควิชาวิศวกรรมนิวเคลียร์

บ บทบาทของวิศวกรรมอุตสาหการ คือ ก่อสร้างแบบ ดำเนินการ ปรับปรุงและสร้างสรรค์นวัตกรรมระบบ ที่ งาน บริการ และ ธุรกิจ

ทักษะ Programming เป็นปัจจัยตระหง่านการคิดอย่างมีเหตุผล อีกทั้งยังเป็นเครื่องมือในการวิเคราะห์และ iron มวลผลอย่างรวดเร็วและแม่นยำตามตัวแปรและข้อจำกัดในบริบทหนึ่ง ๆ ซึ่งมีความซับซ้อนมากขึ้นในอนาคต

ศ. ดร. ณัฐชา ทวีแสงสกุลไทย
ภาควิชาวิศวกรรมอุตสาหการ



วิทยาการทางด้านวิศวกรรมโลหการและวัสดุ เป็นความรู้ที่เกี่ยวข้องกับหลักการพื้นฐานของวัสดุต่าง ๆ กระบวนการแปรรูปและขั้นรูปโลหะ สมบัติของวัสดุ และการเลือกและออกแบบวัสดุที่เหมาะสมกับงานที่หลากหลาย ดังนั้น เพื่อให้สามารถพัฒนาวัสดุใหม่ให้มีสมบัติต่าง ๆ ที่ดีขึ้น Computer Programming จึงเข้ามามีบทบาทสำคัญในการสร้างแบบจำลองต่าง ๆ เพื่อเชื่อมโยงความสัมพันธ์ระหว่าง กระบวนการผลิต - โครงสร้างของวัสดุ - สมบัติของวัสดุ - ความสามารถในการใช้งาน ในปัจจุบันมีการนำ Computer Programming เข้ามาใช้อย่างแพร่หลายมากขึ้น อาทิเช่น การทำนายโครงสร้างจุลภาคของวัสดุภายหลังการขึ้นรูปด้วยกรรมวิธีต่าง ๆ เช่น Casting, 3D printing, Metal Forming จนสามารถทำให้ปรับปรุงให้วัสดุมีความแข็งแรงที่สูงขึ้นได้ หรือมีการนำมาใช้เพื่อทำนายการพังเสียหายของวัสดุจากการเกิด Fatigue และ Corrosion ทำให้สามารถวางแผนการซ่อมบำรุงได้อย่างเหมาะสมมากขึ้น นอกจากนั้นยังมีการนำ Artificial Intelligence เข้ามาใช้เพื่อหาส่วนผสมของวัสดุใหม่ ๆ ที่ยังไม่เคยมีการค้นพบอีกด้วย

อ. ดร. เชษฐา พันธุ์เครือบุตร
ภาควิชาวิศวกรรมโลหการ

Programming เป็นเครื่องมือที่ใช้ในการสร้าง คำนวนและวิเคราะห์ผลของแบบจำลองทางวิศวกรรมของระบบทางกล ความร้อน ของแข็งและของเหลว นอกจากแบบจำลองแล้ว ยังใช้ในการควบคุมระบบทางกลต่าง ๆ เช่น ระบบอัตโนมัติหุ่นยนต์ เป็นต้น

อ. ดร. สุรชัย ขวัญเมือง
ภาควิชาวิศวกรรมเครื่องกล





งานทางด้านวิศวกรรมแหล่งน้ำมีความจำเป็นที่จะต้องยุ่งเกี่ยวกับข้อมูลจำนวนมากทั้งข้อมูลน้ำฝน ข้อมูลน้ำท่า ข้อมูลสภาพพื้นที่ ข้อมูลความต้องการการใช้น้ำ Programming จึงเป็นเครื่องมือสำคัญในการช่วยจัดการ วิเคราะห์ และประมวลผลข้อมูลต่าง ๆ อย่างมีประสิทธิภาพ

พศ. ดร. อุบลรักเบิร์ก ศรีอรโยธัย^น
ภาควิชาวิศวกรรมแหล่งน้ำ

ก การใช้แบบจำลองแหล่งน้ำเก็บปีโตรเลียมเป็นสิ่งจำเป็นในงานวิศวกรรมปีโตรเลียม เพื่อทำความเข้าใจกับพฤติกรรมการผลิตในอดีตและปัจจุบัน และยังใช้ทำนายการผลิตในอนาคตอีกด้วย นอกจากนี้ปริมาณข้อมูลการผลิตที่ถูกจัดเก็บอย่างต่อเนื่อง จำเป็นต้องมีการบริหารจัดการที่ดี เพื่อให้วิศวกรสามารถนำไปใช้ประโยชน์ได้หลากหลายรูปแบบ ความสามารถของ การเข้าใจและสามารถเขียนโปรแกรมเพื่อใช้งานเป็นการเฉพาะ เป็นสิ่งจำเป็นสำหรับวิศวกรปีโตรเลียม

สำหรับงานวิศวกรรมเหมืองแร่มีการนำ Programming ไปใช้ในการออกแบบการทำเหมืองแร่ เช่น การเปิดหน้าดินเพื่อนำแร่ออกมานำ คำนวณเสถียรภาพของชั้นดิน การขนส่งวัตถุดิบจากบริเวณหน้าเหมือง การใช้ซอฟต์แวร์เพื่อประกอบการคำนวณปริมาณสำรอง เป็นต้น

พศ. ดร. จิรวัฒน์ ชีวอรุ่งโรจน์
ภาควิชาวิศวกรรมเหมืองแร่และปีโตรเลียม



การที่มนุษย์เราเป็นสัตว์สังคมที่อยู่ร่วมกัน ทำให้สิ่งที่เราจะเรียนรู้ตั้งแต่เด็ก ๆ คือภาษาที่เราใช้ในการสื่อสารระหว่างกัน เพื่อให้มนุษย์สามารถเข้าใจกันได้เป็นอย่างดี เลยก็ตามเดียวกัน การที่จะเข้าใจสิ่งต่าง ๆ ในโลกปัจจุบัน ที่เป็นโลกดิจิทัล จำเป็นอย่างยิ่งที่เราจะต้องเรียนรู้ภาษาที่คอมพิวเตอร์เข้าใจ ดังเช่นภาษา Python ซึ่งจะช่วยให้เราได้เข้าใจการทำงานของคอมพิวเตอร์มากยิ่งขึ้น ดังนั้น นิสิตที่ศึกษาในภาควิชาวิศวกรรมคอมพิวเตอร์ มีความจำเป็นอย่างยิ่งที่จะต้องเรียนรู้การเขียนโปรแกรม เพราะไม่เพียงแต่จะต้องเข้าใจกลไกและกระบวนการทำงานของคอมพิวเตอร์เท่านั้น แต่ยังจะต้องสามารถเขียนโปรแกรมเพื่อควบคุมการทำงานของคอมพิวเตอร์ได้อย่างเหมาะสมและมีประสิทธิภาพ

พศ. ดร. นรัฐวุฒิ หนูไฟโจร
ภาควิชาวิศวกรรมคอมพิวเตอร์

01 : Data Type, Variable and Expression

สรุปเนื้อหา

ตัวแปรเป็นที่เก็บข้อมูลในโปรแกรม ห้องมีชื่อกำกับ

- ชื่อตัวแปรจะกลบจ้ำงวอักษร ตัวเลข หรือเครื่องหมายขีดเส้นใต้ _ ตัวอังกฤษใหญ่ไม่เหมือนตัวเล็ก ห้ามขีด underscore ชื่อตัวเลข
- อย่างตั้งชื่อตัวแปรซ้ำกับชื่อฟังก์ชันใน Python เช่น int, str, max, sum, abs, ... (ไม่ห้ามถ้าจะตั้งช้า แต่ไม่ควรทำอย่างยิ่ง)

ข้อมูลใน Python ที่นำมาประมวลผลมีหลายประเภท ที่เราจะศึกษาในบทนี้มีดังต่อไปนี้

int	จำนวนเต็ม	-10 5000011 (Python ห้ามไม่ให้เขียน 0 นำหน้าจำนวนเต็ม เช่น 020)
float	จำนวนจริง	10.0 1.23e59 มีค่าเท่ากับ 1.23×10^{59}
str	ข้อความ	'Programming is easy' "Let's go shopping"

การให้ค่ากับตัวแปร

- a = b = c = 0.0 ให้ตัวแปร a b และ c เก็บจำนวนจริง 0.0
- a = 5; b = 6; a,b = b,a ตัวแปร a กับ ตัวแปร b สลับค่ากัน ได้ a เก็บ 6 และ b เก็บ 5
- a = x ถ้า x ไม่เคยมีการให้ค่ามาก่อน คำสั่งนี้จะผิด เพราะไม่รู้ว่า x มีค่าเท่าใด

ตัวดำเนินการ ลำดับการทำงาน และการแปลงประเภทข้อมูล

- ตัวดำเนินการ บวก (+), ลบ (-), คูณ (*), ยกกำลัง (**), หาร (/), หารปัดเศษ (//), เศษจากการหาร (%)
- การดำเนินการระหว่างจำนวนเต็มกับจำนวนจริงจะได้ผลเป็นจำนวนจริง (เช่น 2 + 1.0 ได้ 3.0)
- // กับจำนวนลบ : 1//2 ได้ 0, (-1)//2 ได้ -1, 11//10 ได้ 1, (-11)//10 เมื่อ 11//10 ได้ -2
- a = 3+97//2**3%8 a มีค่า $3+97//8 \% 8 = 3+12 \% 8 = 3+4 = 7$
- a = 12//3/2+2**3**2 a มีค่า $12//3/2+2**9 = 12//3/2+512 = 4/2+512 = 2.0+512 = 514.0$
- a += 2 ก็คือ a = a + 2, a /= 2 ก็คือ a = a // 2, a *= -1 ก็คือ a = a * -1
- ถ้า import math จะมีค่าคงตัวและฟังก์ชันทางคณิตศาสตร์ให้ใช้มากมาย
 - math.pi, math.e, math.sin(x), math.cos(x), math.sqrt(x), math.log(x,b), ...
- นำสตริงบวกกัน คือนำสตริงมาต่อกัน เช่น '12'+'23' คือ '1223'
- สตริงคูณกับจำนวนเต็ม คือนำสตริงนั้นมาต่อกันเป็นจำนวนครั้งเท่ากับค่าของจำนวนเต็มนั้น เช่น '12'*3 คือ '121212'
- ฟังก์ชัน int, float และ str มีไว้เปลี่ยนประเภทข้อมูล เช่น
 - int('12') ได้ 12, float('1.2') ได้ 1.2, str(12//2) ได้ '6', str('Aa') ได้ 'Aa'
- ข้อควรระวัง : รู้ความแตกต่างของ / กับ // และศึกษาลำดับการทำงานของ operator ให้ดี
ถ้าไม่มั่นใจ ใส่เงื่อนไข เช่น a/2*b เท่ากับ (a/2)*b แต่ a/(2*b) เท่ากับ a/2/b

คำสั่งการแสดงข้อมูลทางจอภาพ

- `print(a,b,c)`
 - `print(str(a)+str(b)+str(c))`
- ท่าค่าในตัวแปร a b และ c มาแสดงต่อ กันคันด้วยช่องว่างบรรทัดเดียว กัน
สำคัญ ว่า แบบที่ 2 แบบที่ 1 ท่าแปร a b และ c มาเปลี่ยนเป็นสตริงต่อ กัน แล้วแสดงบนบรรทัดเดียว กัน

คำสั่งการอ่านข้อมูลจากแป้นพิมพ์

- `a = input()`
- `a = input().strip()`
- `a = int(input())`
- `a = float(input())`
- ถ้าต้องการอ่านข้อมูลหลาย ๆ ตัวที่ผู้ใช้ป้อนเข้ามาในบรรทัดเดียวกัน โดยข้อมูลแต่ละตัวคันด้วยช่องว่าง
 - `a,b,c = [e for e in input().split()]`
◦ `a,b,c = input().split()` หรือ อ่านสตริง 3 ตัว
 - `x,y = [int(e) for e in input().split()]` อ่านจำนวนเต็ม 2 จำนวน
 - `a,b,c = [float(e) for e in input().split()]` อ่านจำนวนจริง 3 จำนวน
 - หากจะอ่านจำนวนจริงตามด้วยจำนวนเต็ม ก็อ่านเป็นสตริงก่อน โดยใช้คำสั่ง `f,n = input().split()` แล้วจึงค่อยแปลงเป็นจำนวนจริงกับจำนวนเต็ม โดยใช้คำสั่ง `f = float(f); n = int(n)`

*** ถ้าโจทย์บอกว่าข้อมูลที่รับมาคันด้วยช่องว่างในบรรทัดเดียวกัน อย่าใช้ `input().split(' ')`

แต่ควรใช้ `input().split()` แทน

เรื่องผิดบอย

รับข้อมูลจากแป้นพิมพ์แล้วลืมแปลง เป็นจำนวน ก่อนนำไปคำนวณ	$x = \text{input}()$ $y = x^{**2} + 7$ ผิด เพราะ x เป็นสตริง
จำลำดับการทำงานของตัวดำเนินการ + - * / // % ** ผิด $(**$ ทำก่อน * / // % ทำก่อน + -)	$y = x / 2*a$ จะได้ $y = (x/2)*a$ ถ้าต้องการคำนวณ $y = \frac{x}{2a}$ $y = x^{**1/3}$ ต้องเขียน $y = x/(2*a)$ จะได้ $(x^{**1})/3$ ถ้าต้องการหารากที่สามของ x ต้องเขียน $y = x^{**}(1/3)$
ลืมใส่ * สำหรับการคูณ	$y = 2x + 1$ ต้องเขียน $y = 2*x + 1$
$10e7$ มีค่าไม่เท่ากับ 10^7	$10e7$ มีค่าเท่ากับ 10×10^7 อยากได้ 10^7 ต้องเขียน $1e7$
ไม่ใช่จำนวนเต็ม 1000	$1e3$ มีค่าเท่ากับ 1000.0 ดังนั้น 2345 % $1e2$ ได้ 45.0

สำหรับผู้ที่เคยเรียนภาษา C อย่าเลือกเขียนคำสั่ง <code>++k</code> หรือ <code>--k</code>	<code>++, --</code> คือ การติดบวกค่าใน <code>k</code> สองครั้ง จึงมีค่าเท่ากับ <code>k</code> ค่าใน <code>k</code> ไม่เปลี่ยน <code>- k</code> คือการติดลบค่าใน <code>k</code> สองครั้ง จึงมีค่าเท่ากับ <code>k</code> ค่าใน <code>k</code> ไม่เปลี่ยน
ลืม <code>import math</code> เมื่อใช้ฟังก์ชัน <code>math</code>	<code>y = (-b+math.sqrt(b*b-4*a*c)) / (2*a)</code> จะพ้องว่าไม่รู้จัก <code>math</code>
ใส่ร่วงเล็บเปิดกับปิดไม่ครบ	<code>import math</code> <code>y = 2+(x*abs(y-z/2))</code> วงเล็บบิดมีน้อยไป <code>y = -b+math.sqrt(b*b-4*a*c)) / (2*a)</code> ขาดวงเล็บเปิด
สะกดชื่อตัวแปรผิด หักผิดเรื่องการใช้ตัวอังกฤษเล็กกับใหญ่	<code>count = 0</code> <code>Count = count + 1</code> Count กับ <code>count</code> เป็นคนละตัว
ตั้งชื่อตัวแปรซ้ำกับชื่อฟังก์ชั่นมาตรฐานใน IDLE ตัวแปรที่ถูกต้องมีสีดำ เป็นสีอินจะสร้างปัญหา	<code>int = 27</code> <code>print(int)</code> ได้ 27 แต่หลังจากนี้ ใช้คำสั่ง <code>a = int(input())</code> เพื่ออ่านข้อมูลจากแป้นพิมพ์ แล้วแปลงเป็นจำนวนเต็มได้แล้ว (IDLE แสดง <code>int</code> ด้วยสีม่วง)
นำข้อมูลที่ไม่ใช่สตริงมาบวกกับสตริง	<code>import math</code> <code>a = math.pi * r**2</code> <code>print('area = '+a)</code> ผิด <code>print('area = '+str(a))</code> แปลง <code>a</code> เป็นสตริงก่อน <code>print('area =',a)</code> แบบนี้ <code>print</code> แปลง <code>a</code> เป็นสตริงให้



แบบฝึกหัด

Problem	Code
<u>Input:</u> รับจำนวนเต็ม 3 จำนวนจากแป้นพิมพ์ (บรรทัดละจำนวน) เก็บในตัวแปร <code>h</code> , <code>m</code> และ <code>s</code> ซึ่งแทนจำนวน ชั่วโมง นาที และ วินาที <u>Process:</u> คำนวณจำนวนวินาทีรวมที่คิดจาก <code>h</code> , <code>m</code> และ <code>s</code> <u>Output:</u> จำนวนวินาทีรวมทั้งหมดที่คำนวณได้	
<u>Input:</u> รับจำนวนจริง 1 จำนวนจากแป้นพิมพ์ เก็บใน <code>x</code> <u>Process:</u> คำนวณ $y = 2 - x + \frac{3}{7}x^2 - \frac{5}{11}x^3 + \log_{10}(x)$ <u>Output:</u> ค่า <code>y</code> ที่คำนวณได้	

Problem	Code
<p><u>Input:</u> รับจำนวนจริง 1 จำนวนจากแป้นพิมพ์ กีบ a</p> <p><u>Process:</u> ให้ x มีค่าเป็น 1 จากนั้นทำคำสั่ง $x = (x + a/x)/2$ จำนวน 4 ครั้ง</p> <p><u>Output:</u> ค่า x ที่ได้จากการทำงานข้างบนนี้</p>	
<p><u>Input:</u> มี 2 บรรทัด แต่ละบรรทัดมีจำนวนจริง 3 จำนวน คั่นด้วย ช่องว่าง อ่านบรรทัดแรกเก็บ ใน $v1, v2, v3$ แทนเวกเตอร์ $v = (v1, v2, v3)$ อ่านบรรทัดที่สองเก็บใส่ $u1, u2, u3$ แทน เวกเตอร์ $u = (u1, u2, u3)$</p> <p><u>Process:</u> คำนวณ dot product ของเวกเตอร์ v กับ u</p> <p><u>Output:</u> ค่า dot product ที่คำนวณได้</p>	
<p><u>Input:</u> อ่านจำนวนจริง 4 จำนวนคั่นด้วยช่องว่างจากแป้นพิมพ์ เก็บใน $x1, y1, x2$ และ $y2$ ค่าของ $x1, y1$ แทนพิกัดของจุดที่ 1 และ $x2, y2$ แทนพิกัดของจุดที่ 2 บนระบบ x-y</p> <p><u>Process:</u> คำนวณระยะห่างสั้นสุดระหว่างจุดทั้งสอง</p> <p><u>Output:</u> ระยะห่างที่หาได้</p>	
<p><u>Input:</u> อ่านพิกัดเชิงข้าวของจุดบนระนาบ ซึ่งเป็นจำนวนจริง 2 จำนวนคั่นด้วยช่องว่าง เก็บในตัวแปร r และ θ (เป็นเรเดียน)</p> <p><u>Process:</u> คำนวณค่า x และ y ซึ่งเป็นพิกัดคาร์ทีเซียน ของจุด (r, θ) ที่อ่านเข้ามา</p> <p><u>Output:</u> ค่า x และ y (คั่นด้วยช่องว่าง)</p>	
<p><u>Input:</u> อ่านพิกัดคาร์ทีเซียนของจุดบนระนาบ ซึ่งเป็นจำนวนจริง 2 จำนวนคั่นด้วยช่องว่าง เก็บในตัวแปร x และ y</p> <p><u>Process:</u> คำนวณค่า r และ θ (เป็นเรเดียน) ซึ่งเป็น พิกัดเชิงข้าวของจุด (x, y)</p> <p><u>Output:</u> ค่า r และ θ (คั่นด้วยช่องว่าง)</p>	

Problem	Code
<u>Input:</u> อ่านจำนวนจริง 5 จำนวน คั่นด้วยตัวอ่าง <u>Process:</u> คำนวณค่าเฉลี่ยของจำนวนทั้งห้า <u>Output:</u> ค่าเฉลี่ยที่หาได้	
<u>Input:</u> รับข้อมูล 3 ตัว a, b กับ c คั่นด้วยช่องว่าง a และ b เป็น เวอักษร ตรง c เป็นจำนวนเต็ม <u>Output:</u> ตัวอักษรใน c ต่อ กับ ตัวอักษรใน b ต่อ กับ ค่าของจำนวนเต็มใน c ต่อ กับ ชุดของตัวอักษรใน a ต่อ กับ ตัวอักษร ใน b ที่ 3 ๆ กัน เป็นจำนวน c ชุด เช่น ผู้ใช้ป้อน v o 5 จะแสดง vo5vovovovovo	

ตัวอย่างการแก้โจทย์ปัญหา

Triangle

จะเขียนโปรแกรมคำนวณพื้นที่สามเหลี่ยมที่ทราบความยาวด้าน (a กับ b) และมุ่งหมายว่า ด้านสองด้านหนึ่ง (C) จากสูตร

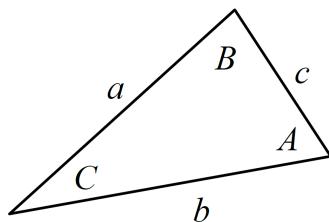
$$area = \frac{1}{2} ab \sin C$$

▶ ข้อมูลนำเข้า

บรรทัดแรกคือความยาวด้าน a (หน่วยเป็นเซนติเมตร)

บรรทัดที่สองคือความยาวด้าน b (หน่วยเป็นเซนติเมตร)

บรรทัดที่สามคือมุ่งหมายว่า ด้านที่สอง C (หน่วยเป็นองศา)



▶ ข้อมูลส่งออก

พื้นที่ของสามเหลี่ยมที่รับเป็นข้อมูลนำเข้า (หน่วยเป็นตารางเซนติเมตร) แสดงในรูปแบบที่แสดงตามตัวอย่างด้านล่าง

▶ ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
10.0 10 90.0	area = 50.0 (sq cm)
1e1 2e1 50.5	area = 77.162458338772 (sq cm)

ตัวอย่างการเขียนโปรแกรม

โปรแกรม	คำอธิบาย
<pre>a = input() b = input() c = input() area = 1/2 a*b*sin C print(area)</pre>	<p>Invalid Syntax ผิดที่บรรทัดที่ 4 ลืมเขียน * --> เปลี่ยนเป็น area = 1/2*a*b*sin C</p>
<pre>a = input() b = input() C = input() area = 1/2*a*b*sin C print(area)</pre>	<p>Invalid Syntax ผิดที่บรรทัดที่ 4 ลืมใส่วงเล็บ --> เปลี่ยนเป็น area = 1/2*a*b*sin(C)</p>
<pre>a = input() b = input() C = input() area = 1/2*a*b*sin(C) print(area)</pre>	<p>สั่ง run, ใส่ข้อมูล, ผิดที่บรรทัดที่ 4 TypeError: can't multiply sequence by non-int of type 'float' แปลว่า ระบบไม่สามารถคูณได้ เพราะ a เป็นสตริง (b และ C ด้วย) จากบรรทัดที่ 1, 2 และ 3 จึงต้องแปลงให้เป็นจำนวนก่อน ในโจทย์ ไม่ได้บอกว่าความยาวด้านและมุมเป็น int หรือ float แต่ถ้าต้องการ พบว่าใส่ได้ทั้ง int และ float จึงต้องแปลงสตริงจาก input() ให้เป็น float</p>
<pre>a = float(input()) b = float(input()) C = float(input()) area = 1/2*a*b*sin(C) print(area)</pre>	<p>Invalid Syntax บอกว่าผิดบรรทัดที่ 2 ระบบบอกผิดบรรทัดใด ให้ดูบรรทัดก่อนหน้าด้วย เพราะผิดก่อนหน้า บางทีตามมาบรรทัดถัดมา ในที่นี้ เห็นได้ว่า ลืมใส่วงเล็บปิด</p>
<pre>a = float(input()) b = float(input()) C = float(input()) area = 1/2*a*b*sin(C) print(area)</pre>	<p>สั่ง run, ใส่ข้อมูล, ผิดที่บรรทัดที่ 4 NameError: name 'sin' is not defined แปลว่า ระบบไม่รู้จักคำว่า sin ก็เพราะว่าต้องเขียน math.sin</p>
<pre>a = float(input()) b = float(input()) C = float(input()) area = 1/2*a*b*math.sin(C) print(area)</pre>	<p>สั่ง run, ใส่ข้อมูล, ผิดที่บรรทัดที่ 4 NameError: name 'math' is not defined แปลว่า ระบบไม่รู้จักคำว่า math ก็เพราะว่าต้อง import math</p>

โปรแกรม	ค่า, รูปท.
<pre>import math a = float(input()) b = float(input()) C = float(input()) area = 1/2*a*b*math.sin(C) print(area)</pre>	<p>สั่ง run, ใส่ข้อมูลตามตัวอย่างแรก</p> <p>10 10 90 ได้ผล</p> <p>44.699833180027895</p> <p>ไม่ตรงกับที่แสดง ต้องได้พื้นที่ 50.0 ได้ผลผิด ก็น่าจะผิดที่การคำนวณ ลองคำนวณเองดู</p> <p>$1/2 \times 10 \times 10 \times \text{math.sin}(90) = 1/2 \times 10 \times 10 \times 1$</p> <p>ก็น่าจะได้ 50.0 แล้วทำไมไม่ใช่ ไม่ใกล้เคียงด้วย ตัวที่น่าสงสัยก็น่าจะ เป็น <code>math.sin(90)</code> เมื่อเรียกใช้ฟังก์ชัน เราต้องเข้าใจกฎเกณฑ์ของ การเรียกใช้ด้วย ลองค้น python <code>math.sin</code> ในเน็ต จะพบข้อความว่า</p> <p><code>math.sin(x)</code> Return the sine of x radians.</p> <p>แสดงว่า ต้องแปลงองศาเป็นเรเดียนก่อนส่งไปให้ <code>math.sin</code> ก็ต้อง คิดวิธีแปลง: 180 องศา เท่ากับ π, C องศา ก็เท่ากับ $C \times \pi / 180$ แล้วจะใช้ค่า π เท่าไรดี จะใช้ $C \times (22/7) / 180$ หรือ $C \times 3.14159 / 180$ แต่น่าจะรู้ว่า ควรใช้ <code>math.pi</code> เพราะระบบเก็บค่า π ที่ละเอียดมากไว้ที่นี่ ดังนั้นใช้ <code>CR = C * math.pi / 180</code> เปลี่ยนเป็น เรเดียนก่อนแล้วค่อยไปใช้</p>
<pre>import math a = float(input()) b = float(input()) C = float(input()) CR = C * math.pi / 180 area = 1/2*a*b*math.sin(CR) print(area)</pre>	<p>สั่ง run, ใส่ข้อมูลตามตัวอย่างแรก, ได้ 50.0 ถูกต้อง run อีกครั้ง, ใส่ข้อมูลของอีกตัวอย่าง</p> <p>1e1 2e1 50.5 ได้ 77.162458338772 ถูกต้อง</p> <p>submit เข้า Grader ตรวจให้คะแนน --> ได้ 0, ทำไม ???? คำนวณพื้นที่ได้ถูกต้อง แต่แสดงผลไม่เหมือนกับที่โจทย์บอก ดูที่ตัวอย่าง <code>area = 50.0 (sq cm)</code> แต่โปรแกรมแสดงแค่พื้นที่ แก้ไขบรรทัดสุดท้ายให้ตรงตามตัวอย่าง</p>
<pre>import math a = float(input()) b = float(input()) C = float(input()) CR = C * math.pi / 180 area = 1/2*a*b*math.sin(CR) print("area =", area, "(sq cm.)")</pre>	<p>สั่ง run, ใส่ข้อมูลตามตัวอย่างแรก, ได้ <code>area = 50.0 (sq cm.)</code></p> <p>มันใจว่าถูก, submit เข้า Grader, แต่ตรวจแล้วได้ 0, ทำไม ???? ใจเย็น ๆ ดูให้มั่นใจว่าเหมือนที่โจทย์ต้องการไหม ?</p> <p><code>area = 50.0 (sq cm)</code> พบว่ามีจุดเกินมาหนึ่งตัว กีบจุดทิ้ง</p>
<pre>import math a = float(input()) b = float(input()) C = float(input()) CR = C * math.pi / 180 area = 1/2*a*b*math.sin(CR) print("area =", area, "(sq cm)")</pre>	<p>สั่ง run, ใส่ข้อมูลตามตัวอย่างแรก, ได้ <code>area = 50.0 (sq cm)</code></p> <p>มันใจว่าถูก ชัวร์, submit เข้า Grader ตรวจ ได้ 100 เต็ม</p>

ตัวอย่างโจทย์ปัญหา

แปลงอุณหภูมิ

สูตรในการเปลี่ยนค่าอุณหภูมิจากเซลเซียสไปเป็นองศา华เรนไฮต์และเคลวินมีดังนี้

$$F = \frac{9}{5}C + 32$$

$$K = C + 273.15$$

ให้อ่านข้อมูลอุณหภูมิ (หน่วยเป็นองศาเซลเซียส) จากนั้นคำนวณหาค่าองศา华เรนไฮต์และเคลวินด้วยสมการด้านบน เมื่อ C คือ องศาเซลเซียส F คือ องศา华เรนไฮต์ และ K คือ เคลวิน

▶ ข้อมูลนำเข้า

หนึ่งบรรทัดประกอบด้วยค่าองศาเซลเซียสเป็นจำนวนจริง

▶ ข้อมูลส่งออก

มีหนึ่งบรรทัดประกอบด้วยตัวเลขจำนวนจริงสองจำนวน ตัวแรกเป็นองศา华เรนไฮต์ และตัวที่สองเป็นเคลวิน

▶ ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
39.85	103.73 313.0

Triangle 2

จงเขียนโปรแกรมคำนวณหาความยาวด้านที่สามของสามเหลี่ยม เมื่อเราทราบความยาวด้านสองด้าน (a กับ b) และมุมระหว่างด้านสองด้านนั้น (C) ซึ่งค่ามาณ์จาก Law of Cosines

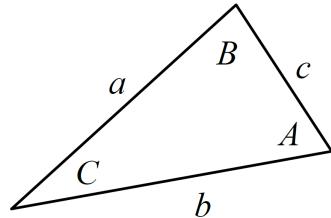
$$c^2 = a^2 + b^2 - 2ab \cos(C)$$

▶ ข้อมูลนำเข้า

บรรทัดแรกคือค่า ขนาด a (หน่วยเป็นเซนติเมตร)

บรรทัดที่สองคือความยาวด้าน b (หน่วยเป็นเซนติเมตร)

บรรทัดที่สามคือมุมระหว่างด้านทั้งสอง C (หน่วยเป็นองศา)



▶ ข้อมูลส่งออก

ความยาวด้านที่สามของสามเหลี่ยมที่รับเป็นข้อมูลนำเข้า (หน่วยเป็นเซนติเมตร) แสดงในรูปแบบที่แสดงตามตัวอย่างด้านล่าง

▶ ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
3 4 90	c = 5.0 cm.
7.0 24.0 90.0	c = 25.0 cm.
10 10 60	c = 9.99999999999998 cm.
3 3 60	c = 2.999999999999996 cm.

ขั้นตอนการทำงานของโปรแกรม

- รับข้อมูลจากแป้นพิมพ์ เป็นจำนวนจริง แล้วเก็บในตัวแปร a
- รับข้อมูลจากแป้นพิมพ์ เป็นจำนวนจริง แล้วเก็บในตัวแปร b
- รับข้อมูลจากแป้นพิมพ์ เป็นจำนวนจริง แล้วเก็บในตัวแปร D
- นำ D ที่มีหน่วยเป็นองศา แปลงเป็น เ雷เดียน เก็บในตัวแปร C
- คำนวณความยาวของด้านที่สาม ด้วยสูตร $c = \sqrt{a^2 + b^2 - 2ab \cos(C)}$
- แสดงความยาวด้านที่สามวนได้ทางจอภาพในรูปแบบที่แสดงตามตัวอย่าง

ISBN

ISBN (International Standard Book Number) เป็นตัวเลขจำนวน 10-13 หลักที่ใช้ระบุหนังสือแต่ละเล่ม โดยข้อนี้สนใจเฉพาะ ISBN ที่มี 10 หลัก การตรวจสอบความถูกต้องของ ISBN จะใช้ตัวเลขหลักสุดท้ายเป็น check digit ในการตรวจสอบความถูกต้องของตัวเลขอื่น ๆ โดยวิธีที่ใช้ร่วมกับคือ

$$10n_1 + 9n_2 + 8n_3 + 7n_4 + 6n_5 + 5n_6 + 4n_7 + 3n_8 + 2n_9 + n_{10} \text{ จะต้องหารด้วย } 11 \text{ ลงตัว}$$

ตัวอย่างเช่น หากตัวเลข 9 หลักแรกคือ 020131452 จะได้ว่า

$$10^0 \cdot 0 + 9 \cdot 2 + 8 \cdot 0 + 7 \cdot 1 + 6 \cdot 3 + 5 \cdot 1 + 4 \cdot 4 + 3 \cdot 5 + 2 \cdot 2 + n_{10} = 83 + n_{10} \text{ ต้องหารด้วย } 11 \text{ ลงตัว}$$

จะได้ว่า n_{10} ต้องมีค่าเท่ากับ 5 เพื่อให้ผลรวมเป็น 88 ซึ่งหารด้วย 11 ลงตัว และได้ ISBN คือ 0201314525

หากกำหนดตัวเลขหลักที่ 1-9 มาให้ จงคำนวณหา ISBN ทั้งสิบหลัก

▶ ข้อมูลนำเข้า

มีบรรทัดเดียว ระบุ ISBN หลักที่ 1-9

▶ ข้อมูลส่งออก

มีบรรทัดเดียว แสดง ISBN ทั้งสิบหลัก รับประกันว่ากรณีทดสอบจะไม่มีกรนีที่ n_{10} เท่ากับ 10

▶ ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
020131452	0201314525
1000000000	1000000001

02 : Selection (if-elif-else)

สรุปเนื้อหา

Flowchart	Code
<pre> graph TD C1{C1} -- Y --> P1[P1] P1 --> C1 C1 -- N --> Start(()) </pre>	<pre>if C1 : P1</pre>
<pre> graph LR C1{C1} -- Y --> P1[P1] P1 --> C1 C1 -- N --> Start(()) notC1{not C1} -- Y --> P1 P1 --> notC1 notC1 -- N --> Start(()) </pre>	<pre>if C1 : pass else : P1</pre>
<pre> graph TD C1{C1} -- Y --> P1[P1] C1 -- N --> P2[P2] P1 --> End(()) P2 --> End(()) </pre>	<pre>if C1 : P1 else : P2</pre>
<pre> graph TD C1{C1} -- Y --> P1[P1] C1 -- N --> C2{C2} C2 -- Y --> P2[P2] C2 -- N --> C3{C3} C3 -- Y --> P3[P3] C3 -- N --> P4[P4] P1 --> End(()) P2 --> End(()) P3 --> End(()) P4 --> End(()) C1{C1} -- Y --> P1[P1] C1 -- N --> C2{C2} C2 -- Y --> P2[P2] C2 -- N --> C3{C3} C3 -- Y --> P3[P3] C3 -- N --> P4[P4] P1 --> End(()) P2 --> End(()) P3 --> End(()) P4 --> End(()) </pre>	<pre>if C1 : P1 elif C2 : P2 elif C3 : P3 else : P4</pre>

Flowchart	Code
<pre> graph TD C1{C1} -- Y --> P1[P1] C1 -- N --> C2{C2} C2 -- Y --> P2[P2] C2 -- N --> C3{C3} C3 -- Y --> P3[P3] C3 -- N --> P4[P4] P2 --> P4 P3 --> P4 P4 --> O(()) </pre>	<pre> if C1 : P1 if C2 : P2 if C3 : P3 else : P4 </pre>
<pre> graph TD P1[P1] --> C1{C1} C1 -- Y --> P2[P2] C1 -- Y --> P4[P4] C1 -- N --> C4{C4} C4 -- Y --> P5[P5] C4 -- N --> C5{C5} C5 -- Y --> P6[P6] C5 -- N --> P7[P7] C2{C2} -- Y --> P2 C2 -- N --> C3{C3} C3 -- Y --> P3[P3] C3 -- N --> P4 P2 --> P4 P3 --> P4 P4 --> P7 P7 --> O(()) </pre>	<pre> if C1 : if C2 : P2 if C3 : P3 else : P4 else : P1 if C4 : P5 elif C5 : P6 P7 </pre>

ตัวอย่างเงื่อนไขที่เขียนแทนกันได้

not($x == 0$)	$x \neq 0$
not($x==2$ or $x==4$)	$x \neq 2$ and $x \neq 4$
not($x<2$ and $y>=4$)	$x \geq 2$ or $y < 4$
$3 \leq x$ and $x < 9$	$3 \leq x < 9$
$a < b$ and $b < c$ and $c < d$ and $d \leq e$	$a < b < c < d \leq e$
$c=='a'$ or $c=='e'$ or $c=='i'$ or $c=='o'$ or $c=='u'$	$c \in ('a', 'e', 'i', 'o', 'u')$ หรือ $c \in 'aeiou'$

<pre>if condition : t = value1 else : t = value2</pre>	$t = \text{value1 if condition else value2}$ ใช้เฉพาะกรณีการให้ค่ากับตัวแปร ถ้าเงื่อนไขเป็นจริงให้ค่าหนึ่ง เป็นเท็จให้อีกค่าหนึ่ง
<pre>if s > a + b % 7 : t = True else : t = False</pre>	$t = \text{True if } s > a+b\%7 \text{ else False}$ หรือเขียนสั้น ๆ $t = (s > a + b \% 7)$

การเปรียบเทียบที่ใช้บ่อย

```
if a%2 == 0 :
if a%100 == 0 :
if (a//100)%10 == 9 :
if (a%1000)//100 == 9:
if a <= x <= b :
if abs(a-b) <= max(abs(a),abs(b))*1e-10 :
```

a เป็นเลขคู่หรือไม่
 a หารด้วย 100 ลงตัวหรือไม่
 เลขหลักท้ายของ a คือ 9 หรือไม่ หรือ
 กี่หนึ่งในห้า
 x มีค่าในช่วงตั้งแต่ a ถึง b หรือไม่
 ตรวจสอบว่าจำนวนจริง a มีค่าใกล้กับ b หรือไม่ โดยตรวจสอบว่า a กับ b ต่างกัน
 เชิงล้มเหลวไม่เกิน 10^{-10} หรือไม่

```
mx = a
if b > mx : mx = b
if c > mx : mx = c
if d > mx : mx = d
mx = max(a,b,c,d)
```

mx เก็บค่ามากสุดของ a,b,c และ d หรือเขียนโดยใช้ฟังก์ชัน max
 $\max(a,b,c,d)$ หากค่ามากสุดของ a,b,c และ d



เรื่องพิดบ่อย

ต้องการเปรียบเทียบความเท่ากัน แต่ใช่ =	<code>if x = 0 :</code>	ต้องเขียน <code>if x == 0 :</code>
ใช้ and กับ or ผิดความหมาย ทำให้ได้ค่าจริงหรือเท็จตลอด	<code>if x != 2 or x != 3 :</code> <code>if x <= 3 and x == 4 :</code>	แบบนี้ได้จริงตลอด แบบนี้ได้เท็จตลอด
เขียนคำสั่งภายใน if หรือ else ไม่ตรงกัน	<code>if x > 0 :</code> <code>a = math.sqrt(x)</code> <code>print(a)</code>	ทุกบรรทัดใน if เขียนบรรทัดไม่ตรงกัน ผิด
ใช้ tab ผสมกับ blank ในการเขียนคำสั่ง	tab ผสมกับ blank ก็อาจดูว่าเขียนตรงกัน แต่ผิด (IDLE จัดการเรื่อง tab กับ blank ให้ถูกต้องโดยอัตโนมัติ)	(IDLE จัดการเรื่อง tab กับ blank ให้ถูกต้องโดยอัตโนมัติ)

เข้าใจผิดเกี่ยวกับการเปรียบเทียบสตริง
สตริงเปรียบเทียบกันตามลำดับแบบที่เขียนใน
พจนานุกรม โดย

'0' < '9' < 'A' < 'Z' < 'a' < '-'

'bc' < 'eg'	'< 'b'	เป็นจริง
'12' < '4'	'< '9'	เป็นจริง
จะเห็นว่า		
print(x)		ได้ 1234
print(y)		ได้ 9
print(x < y)		ได้ True ถ้า x = '1234', y = '9'
print(x < y)		ได้ False ถ้า x = 1234, y = 9

เรื่องที่ปรับปรุงได้

คำสั่งที่เหมือนกันทั้งในกลุ่มหลัง if และ
กลุ่มหลัง else อาจแยกออกมาข้างนอกก็ได้

การใช้ if-elif-else ที่ตรวจสอบค่าต่ออยู่ใน
ช่วงใด สามารถลดการเปรียบเทียบลงได้ ถ้าจัด
ลำดับการเปรียบเทียบให้เหมาะสม

```
if d > 0 :
    a = 9
    c += d - 5
    e = c
else:
    a = 9
    c -= d + 7
    e = c
```

```
if s >= 80 :
    g = 'A'
elif 70 <= s < 80 :
    g = 'B'
elif 60 <= s < 70 :
    g = 'C'
elif 50 <= s < 60 :
    g = 'D'
else :
    g = 'F'
```

```
a = 9
if d > 0:
    c += d - 5
else:
    c -= d + 7
e = c
```

```
if s >= 80 :
    g = 'A'
elif s >= 70 :
    g = 'B'
elif s >= 60 :
    g = 'C'
elif s >= 50 :
    g = 'D'
else :
    g = 'F'
```



Problem	Code
<p><u>Input:</u> รับจำนวนเต็ม 3 จำนวน คั่นด้วยช่องว่าง</p> <p><u>Process:</u> หาผลบวกของจำนวนทั้ง 3</p> <p><u>Output:</u> มีข้อความที่หาได้</p>	
<p><u>Input:</u> รับข้อมูลของวงกลม 2 วง บรรทัดละหนึ่งวง ประกอบด้วยจำนวนจริง 3 จำนวนคั่นด้วยช่องว่าง แทน พิกัด x กับ y ของจุดศูนย์กลาง และรัศมีของวงกลม</p> <p><u>Process:</u> ตรวจว่าวงกลมสองวงที่รับมาทับกันหรือแตะกัน หรือไม่</p> <p><u>Output:</u> แสดงคำว่า touch เมื่อขอบของทั้งสองวงแตะกัน พอดี แสดงคำว่า overlap เมื่อสองวงทับกัน ถ้าไม่แตะ หรือทับ ให้แสดงคำว่า free</p>	
<p><u>Input:</u> รับจำนวนจริง 2 จำนวน คั่นด้วยช่องว่าง แทนพิกัด (x, y) บนระบบสองมิติ</p> <p><u>Process:</u> ตรวจว่าพิกัด (x, y) อยู่บริเวณใดในระบบ</p> <p><u>Output:</u> ตำแหน่งของพิกัด (x, y) ว่า อยู่ในจตุภาคใด หรืออยู่บนแกน x หรือ y หรืออยู่ที่จุดกำเนิด</p>	
<p><u>Input:</u> รับจำนวนเต็ม 5 จำนวน คั่นด้วยช่องว่าง</p> <p><u>Process:</u> ตรวจว่าลำดับจากซ้ายไปขวาของจำนวนที่รับมา เรียงจากน้อยไปมากหรือไม่</p> <p><u>Output:</u> ผลการตรวจว่า True หรือ False</p>	

Problem	Code
<p><u>Input:</u> รับจำนวนเต็ม 4 จำนวน คั่นด้วยช่องว่าง</p> <p><u>Process:</u> หาผลรวมของจำนวนที่รับมา โดยไม่รวม จำนวนที่มากสุดหนึ่งจำนวน และจำนวนที่น้อยสุดหนึ่งจำนวน</p> <p><u>Output:</u> ผลรวมที่หาได้</p>	
<p><u>Input:</u> รับจำนวนหนึ่งจำนวนเก็บในตัวแปร a</p> <p><u>Process:</u> ตรวจสอบว่ามีจำนวนเต็ม x ที่ค่า x^3 เท่ากับ a หรือไม่</p> <p><u>Output:</u> ถ้ามี แสดงค่าของ x ถ้าไม่มี แสดง Not Found</p>	
<p><u>Input:</u> รับจำนวนเต็มแทนรอบอก (หน่วยเป็นนิ้ว)</p> <p><u>Process:</u> หาขนาดของเสื้อยืดโดยโภคตามรอบอกดังนี้</p> <p>น้อยกว่า 37 นิ้ว ขนาด XS</p> <p>ตั้งแต่ 37 แต่ไม่ถึง 41 นิ้ว ขนาด S</p> <p>ตั้งแต่ 41 แต่ไม่ถึง 43 นิ้ว ขนาด M</p> <p>ตั้งแต่ 43 แต่ไม่ถึง 46 นิ้ว ขนาด L</p> <p>ตั้งแต่ 46 นิ้วเป็นต้นไป ขนาด XL</p> <p><u>Output:</u> ขนาดเสื้อโดยโภคตามรอบอกที่ได้รับ</p>	

ตัวอย่างการแก้โจทย์ปัญหา

สลากกินแบ่ง

หากเราซื้อสลาก กินแบ่งเรียงหมายเลขตั้งแต่ที่หมายเลข n_1 ต่อเนื่องไปจนถึงหมายเลข n_2 (เช่นหมายเลข 10300 ถึง 13999) และลงวันที่ร่างวัลที่ 1 คือ หมายเลข p_1 เลขท้ายสองตัวคือหมายเลข p_2 และ เลขท้ายสามตัวคือหมายเลข p_3 เราจะได้ร่างวัลรวมเป็นเงินเท่าไร

กำหนดให้สลากกินแบ่งที่ขายนี้เป็นรุ่นพิเศษ เป็นเลข 5 หลัก รางวัลที่หนึ่ง 10,000 บาท หนึ่งรางวัล รางวัลเลขท้ายสองตัวหนึ่งหมายเลข 25 บาท และรางวัลเลขท้ายสามตัวหนึ่งหมายเลข 100 บาท

► ข้อมูลนำเข้า

หนึ่งบรรทัดประกอบด้วยจำนวนเต็ม 5 จำนวน $p_1 \ p_2 \ p_3 \ n_1 \ n_2$ คั่นด้วยช่องว่าง

► ข้อมูลส่งออก

เงินรางวัลรวมที่ได้รับ

► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอกาฟ)
01234 11 811 01000 01250	10075
99999 99 999 99950 99999	10125
19999 13 001 09015 13000	1275

ตัวอย่างการเขียนโปรแกรม

โปรแกรม	คำอธิบาย
<pre>p1,p2,p3,n1,n2 = \ [int(e) for e in input().split()] s = 0 if n1 <= p1 <= n2 : s += 10000 if n1 <= p2 <= n2 : s += 25 if n1 <= p3 <= n2 : s += 100 print(s)</pre>	บรรทัดแรกรับข้อมูลใส่ตัวแปร p_1, p_2, p_3, n_1 และ n_2 ให้ตัวแปร s เก็บเงินรางวัลรวม เริ่มด้วยการตรวจสอบ หมายเลขของรางวัลที่ 1 อยู่ในช่วงหมายเลขที่ซื้อหรือไม่ ($n_1 \leq p_1 \leq n_2$) ถ้าใช่ก็เพิ่มเงินรางวัล 10,000 บาท ตามด้วยการตรวจสอบเลขท้ายสองตัว แล้วก็สามตัว ในลักษณะเดียวกัน สิ้น run, ใส่ข้อมูลตามตัวอย่าง 01234 11 811 01000 01250, ได้ 10000 บาท ไม่ตรงตามตัวอย่าง ได้แค่รางวัลที่หนึ่ง เลขท้ายตรวจไม่พบ

โปรแกรม	คำอธิบาย
<pre data-bbox="87 178 619 355"> p1,p2,p3,n1,n2 = \ [int(e) for e in input().split()] s = 0 if n1 <= p1 <= n2 : s += 10000 print(s) </pre>	<p>การเขียนโปรแกรมที่มีหลาย ๆ กรณี ควรแยกทดสอบหากเขียนรวดเดียว จะทำให้ติดลำบาก ใจเย็น ๆ ขอเขียนและทดสอบกรณีรางวัลที่หนึ่งก่อน</p> <p>สั่ง run, ใส่ข้อมูล 12345 00 000 12000 13000 ได้ 10000 ถูกต้อง (ทดสอบกรณีอยู่ระหว่าง)</p> <p>สั่ง run, ใส่ข้อมูล 12345 00 000 12345 13000 ได้ 10000 ถูกต้อง (ทดสอบกรณีอยู่ที่ขอบล่าง)</p> <p>สั่ง run, ใส่ข้อมูล 12345 00 000 12000 12345 ได้ 10000 ถูกต้อง (ทดสอบกรณีอยู่ที่ขอบบน)</p> <p>สั่ง run, ใส่ข้อมูล 12345 00 000 12345 12345 ได้ 10000 ถูกต้อง (ทดสอบกรณีแคบไปเฉยและถูกรางวัล)</p> <p>สั่ง run, ใส่ข้อมูล 12345 00 000 12346 14000 ได้ 0 ถูกต้อง (ทดสอบกรณีอยู่นอกช่วงทางซ้าย)</p> <p>สั่ง run, ใส่ข้อมูล 12345 00 000 12000 12344 ได้ 0 ถูกต้อง (ทดสอบกรณีอยู่นอกช่วงทางขวา)</p> <p>สรุปว่า กรณีรางวัลที่หนึ่ง ถูกต้อง</p>
<pre data-bbox="87 798 612 1024"> p1,p2,p3,n1,n2 = \ [int(e) for e in input().split()] s = 0 #if n1 <= p1 <= n2 : # s += 10000 if n1%100 <= p2 <= n2%100 : s += 25 print(s) </pre>	<p>คราวนี้สนใจกรณีเลขท้ายสองตัว จะ comment คำสั่ง ตรวจสอบที่หนึ่งออก ถ้ากลับไปดูโปรแกรมแรกที่เขียน คำสั่ง if n1 <= p2 <= n2 ตรวจเลขท้ายไม่ครบ ทุกกรณี เช่นข้อหมายเลขอ 10000 ถึง 10099 เลขท้าย p2 = 50 การทดสอบ n1 <= p2 <= n2 เป็นเท็จ แต่ความจริงแล้วถูกเลขท้ายสองตัว จึงห้องเปลี่ยนเป็น ทดสอบเฉพาะสองหลักขวาเท่านั้น ด้วยคำสั่ง</p> <pre data-bbox="734 1053 1201 1313"> if n1%100 <= p2 <= n2%100 สั่ง run, ใส่ 00000 50 000 10000 10099 ได้ 25 ถูกต้อง สั่ง run, ใส่ 00000 50 000 10000 10199 ได้ 25 ผิด น่าจะได้ 50 สั่ง run, ใส่ 00000 50 000 10000 10299 ได้ 25 ผิด น่าจะได้ 75 </pre>

โปรแกรม	คำอธิบาย															
<pre>p1,p2,p3,n1,n2 = \ [int(e) for e in input().split()] s = 0 #if n1 <= p1 <= n2 : # s += 10000 if n1//100 == n2//100 : if n1%100 == p2 <= n2%100 : s += 25 else: s += .5*(n2//100 - n1//100 + 1) print(s)</pre>	<p>คำสั่ง</p> <p>ใช้ได้เฉพาะ กรณีที่ 3 หลักแรกของ n1 และ n2 เท่ากัน เท่านั้น (เมื่อ $n1/100$ มีค่าเท่ากับ $n2/100$)</p> <p>จึงขอจัดการเป็นสองกรณีคือ</p> <p>A. กรณี 3 หลักแรกเท่ากัน เช่น 10000 ถึง 10099 ทำเหมือนเดิม</p> <p>B. กรณี 3 หลักแรกไม่เท่ากัน เช่น 10000 ถึง 10299 เมื่อนำ $102 - 100 = 2$ แสดงว่า ถูกเลขท้ายสองตัว 2 หมายเลข ก็คูณด้วย 25</p> <p>สั่ง run, ใส่ $00000\ 50\ 000\ 10000\ 10199$ ได้ 50 ถูกต้อง</p> <p>สั่ง run, ใส่ $00000\ 50\ 000\ 10000\ 10299$ ได้ 75 ถูกต้อง</p> <p>สั่ง run, ใส่ $00000\ 50\ 000\ 10060\ 10680$ ได้ 175 ผิด น่าจะได้ 150</p>															
<pre>p1,p2,p3,n1,n2 = \ [int(e) for e in input().split()] s = 0 #if n1 <= p1 <= n2 : # s += 10000 if n1//100 == n2//100: if n1%100 <= p2 <= n2%100 : s += 25 else: if n1%100 <= p2 : # ช่วง 1 s += 25 if p2 <= n2%100 : # ช่วง 3 s += 25 s += 25*((n2//100-1) - \ # ช่วง 2 (n1//100+1) + 1) print(s)</pre>	<p>กรณีของ $p2 = 50$, $n1 = 10060$ และ $n2 = 10680$ การตรวจสอบน่าจะซับซ้อนเล็กน้อย แบ่งเป็นสามช่วง คือ</p> <table border="1"> <thead> <tr> <th>ช่วงที่ 1</th> <th>ช่วงที่ 2</th> <th>ช่วงที่ 3</th> </tr> </thead> <tbody> <tr> <td>$10060 - 10099$</td> <td>$10100 - 10599$</td> <td>$10600 - 10680$</td> </tr> <tr> <td>3 หลักซ้าย เท่ากัน ดูเฉพาะ 2 หลักขวา ดูเฉพาะสามหลักซ้าย</td> <td>2 หลักขวา 00 ถึง 99 ถูกเลขท้าย 2 หลักขวา</td> <td>3 หลักซ้าย 00 ถึง 99 ถูกเลขท้าย 2 หลักขวา</td> </tr> <tr> <td>$n1 \% 100 \leq p2$</td> <td>ดูเฉพาะสามหลักซ้าย</td> <td>$p2 \leq n2 \% 100$</td> </tr> <tr> <td>60 <= 50 เท็จ ไม่ถูกเลขท้าย</td> <td>จาก 101 ถึง 105 ถูก琅รัล $(105-101+1)$ ครั้ง</td> <td>50 <= 80 จริง ถูก 1 ครั้ง</td> </tr> </tbody> </table> <p>สั่ง run, ใส่ $00000\ 50\ 000\ 10000\ 10299$ ได้ 75 ถูกต้อง</p> <p>สั่ง run, ใส่ $00000\ 50\ 000\ 10060\ 10680$ ได้ 150 ถูกต้อง</p>	ช่วงที่ 1	ช่วงที่ 2	ช่วงที่ 3	$10060 - 10099$	$10100 - 10599$	$10600 - 10680$	3 หลักซ้าย เท่ากัน ดูเฉพาะ 2 หลักขวา ดูเฉพาะสามหลักซ้าย	2 หลักขวา 00 ถึง 99 ถูกเลขท้าย 2 หลักขวา	3 หลักซ้าย 00 ถึง 99 ถูกเลขท้าย 2 หลักขวา	$n1 \% 100 \leq p2$	ดูเฉพาะสามหลักซ้าย	$p2 \leq n2 \% 100$	60 <= 50 เท็จ ไม่ถูกเลขท้าย	จาก 101 ถึง 105 ถูก琅รัล $(105-101+1)$ ครั้ง	50 <= 80 จริง ถูก 1 ครั้ง
ช่วงที่ 1	ช่วงที่ 2	ช่วงที่ 3														
$10060 - 10099$	$10100 - 10599$	$10600 - 10680$														
3 หลักซ้าย เท่ากัน ดูเฉพาะ 2 หลักขวา ดูเฉพาะสามหลักซ้าย	2 หลักขวา 00 ถึง 99 ถูกเลขท้าย 2 หลักขวา	3 หลักซ้าย 00 ถึง 99 ถูกเลขท้าย 2 หลักขวา														
$n1 \% 100 \leq p2$	ดูเฉพาะสามหลักซ้าย	$p2 \leq n2 \% 100$														
60 <= 50 เท็จ ไม่ถูกเลขท้าย	จาก 101 ถึง 105 ถูก琅รัล $(105-101+1)$ ครั้ง	50 <= 80 จริง ถูก 1 ครั้ง														

โปรแกรม	คำอธิบาย
<pre>p1,p2,p3,n1,n2 = \ [int(e) for e in input().split()] s = 0 if n1 <= p1 <= n2 : # s += 10000 if n1%100 <= p2 : #ช่วง 1 s += 25 if p2 <= n2%100 : #ช่วง 3 s += 25 s += 25*((n2//100 - 1) - \ (n1//100+1) + 1) #ช่วง 2 print(s)</pre>	<p>แต่ถ้าพิจารณาให้ลับเอี้ยด จะพบว่า คำสั่งที่พิจารณาทั้ง 3 ช่วงนั้น ครอบคลุมกราฟนี้ <code>if n//100 == n2//100</code> ที่เราเขียนไว้ตอนต้น คือกรณีที่สามหลักซ้ายมีค่าเท่ากัน จึงลบทั้งได้กลายเป็นโปรแกรมข้างซ้ายนี้</p> <p>สั่ง run, ใส่ 00000 50 000 10000 10299 ได้ 75 ถูกต้อง</p> <p>สั่ง run, ใส่ 00000 50 000 10060 10680 ได้ 150 ถูกต้อง</p>
<pre>p1,p2,p3,n1,n2 = \ [int(e) for e in input().split()] s = 0 if n1 <= p1 <= n2 : s += 10000 if n1%100 <= p2 : s += 25 if p2 <= n2%100 : s += 25 s += 25*(n2//100 - n1//100 - 1) if n1%1000 <= p3 : s += 100 if p3 <= n2%1000 : s += 100 s += 100*(n2//1000 - n1//1000 - 1) print(s)</pre>	<p>การตรวจสอบการทำงานด้วยตัวอย่างที่โจทย์ให้มา</p> <p>สั่ง run, ใส่ 01234 11 811 01000 01250 ได้ 10075 ถูกต้อง</p> <p>สั่ง run, ใส่ 99999 99 999 99950 99999 ได้ 10125 ถูกต้อง</p> <p>สั่ง run, ใส่ 19999 13 001 09015 13000 ได้ 1275 ถูกต้อง</p>

ตัวอย่างໂຄງປົງຫາ

Days in Month

ให้เขียนโปรแกรมที่รับค่าเดือนและปีเป็นพุทธศักราช จากนั้นหาว่าในเดือนของปีนั้น จะมีจำนวนวันทั้งสิ้นกี่วัน

ตัวอย่าง: ปี ๒๕๖๗ มกราคม มี 29 วัน ก็ต่อเมื่อ

(ปี ค.ศ. หารด้วย 4 ลงตัว แต่หารด้วย 100 ไม่ลงตัว) หรือ (ปี ค.ศ. หารด้วย 400 ลงตัว)

▶ ข้อมูลนำเข้า

มี 1 บรรทัด ประกอบด้วยจำนวนเต็ม 2 ตัว คือ เดือนและปีเป็นพุทธศักราช

▶ ข้อมูลส่งออก

จำนวนวันของเดือนและปีของข้อมูลนำเข้า

▶ ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
10 2557	31
2 2557	28
4 2556	30
2 2547	29

ตัดเกรด

การตัดเกรดของวิชานี้เป็นไปตามตาราง ข้างล่าง ที่เขียนโปรแกรมเพื่อตัดเกรดตามเกณฑ์ที่ระบุ โดยในกรณีที่คะแนนมีข้อผิดพลาด ให้แสดงผลว่า ERROR

คะแนนรวม (x)	เกรด
$80 \leq x \leq 100$	A
$75 \leq x < 80$	B+
$70 \leq x < 75$	B
$65 \leq x < 70$	C+
$60 \leq x < 65$	C
$55 \leq x < 60$	D+
$50 \leq x < 55$	D
$0 \leq x < 50$	F
กรณีอื่น ๆ	ERROR

▶ ข้อมูลนำเข้า

มีบรรทัดเดียว แทนคะแนนที่จะตัดเกรด เป็นจำนวนจริง

▶ ข้อมูลส่งออก

มีบรรทัดเดียว ระบุเกรดที่ได้รับ โดยในกรณีที่คะแนนมีข้อผิดพลาด ให้แสดงผลว่า ERROR

▶ ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
87.25	A
69.95	C+
120	ERROR

คิดค่าที่จอดรถ

ให้รับเวลาเข้าและออกของรถคันหนึ่ง (เว็บบริการตั้งแต่ 7:00 - 23:00) จากนั้นคำนวณค่าที่จอดรถที่ต้องจ่ายโดยหลักเกณฑ์การคำนวนมีดังนี้

- จอดรถไม่เกิน 15 นาที ไม่มีค่าจ่าบีการ
- จอดรถเกิน 15 นาที แต่ไม่เกิน 4 ชั่วโมง คิดค่าบริการชั่วโมงละ 10 บาท เศษของชั่วโมงคิดเป็นหนึ่งชั่วโมง
- จอดรถตั้งแต่ 4 ชั่วโมง ถึง 6 ชั่วโมง คิดค่าบริการชั่วโมงที่ 4-6 ชั่วโมงละ 20 บาท เศษของชั่วโมงคิดเป็นหนึ่งชั่วโมง
- จอดรถเกิน 6 ชั่วโมง เป็นเวลาจ่ายวันละ 200 บาท

► ข้อมูลนำเข้า

มี 4 บรรทัด แต่ละบรรทัดมีจำนวนเต็มหนึ่งจำนวน

โดยบรรทัดที่ 1-2 เป็นชั่วโมงและนาทีของเวลาเข้า และบรรทัดที่ 3-4 เป็นชั่วโมงและนาทีของเวลาออก

► ข้อมูลส่งออก

มีบรรทัดเดียว เป็นค่าที่จอดรถที่ต้องจ่าย ให้แสดงผลลัพธ์เป็นจำนวนเต็ม

► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
7 0 7 15	0
7 0 7 16	10
7 30 10 30	30
7 30 10 31	50
7 30 13 31	200

CONFIDENTIAL

Cross rome: Intania 85

CEO & Co-Founder
Dek-D Interactive co., Ltd.

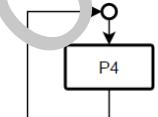
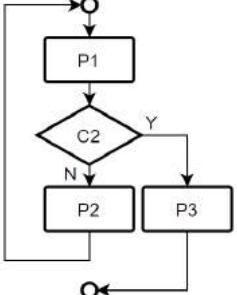
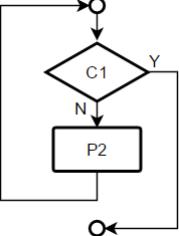
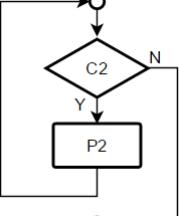


Programming จำเป็นมากในการเริ่มกิจการด้าน IT เพราะกิจการส่วนใหญ่ที่สามารถอยู่รอดได้ ผู้ก่อตั้งซึ่งเป็นผู้ที่เข้าใจผลิตภัณฑ์ที่จะสร้างมากที่สุด ต้องมีส่วนร่วมในการพัฒนาระบบเอง ถึงจะสามารถถ่ายทอดความคิด และผลักดันผลิตภัณฑ์ที่ดีออกมาได้ เรียกว่ามีเงินมากแค่ไหน ถ้าต้องไปจ้างให้คนอื่นมาเขียนโปรแกรมแทนให้ยังไงก็สู้คนที่มีทักษะ และทักษะลงมือเขียนโปรแกรมเองได้ยาก

นอกจากนี้ Programming ยังเป็นการฝึกวิธีคิดแก้ปัญหาอย่างเป็นขั้นตอน ทำให้ผู้เรียนมีระบบการคิดซึ่งสามารถประยุกต์ไปแก้ปัญหาได้ในหลายแขนงอย่างที่ผู้เรียนคาดไม่ถึงอีกด้วย (เทพในหลาย ๆ วงการจบวิศวฯ คอม)

03 : Repetition (while, for)

สรุปเนื้อหา

Flowchart	Code
	<pre>while True : P4</pre>
	<pre>while True : P1 if C2 : P3 break P2</pre>
	<pre>while True : if C1 : break P2</pre>
	<pre>while True : if not C2 : break P2</pre>

Flowchart	Code
	<pre> while C1 : P1 while C2 : P2 P3 </pre>

ให้สังเกตว่า ภายในวงวน while ควรมีคำสั่งที่เปลี่ยนแปลงเงื่อนไขของ while ไม่ เช่นนั้นจำนวนทำงานไม่ถ้วนสุด เช่น

`while i < j :`

`...`

`i += 2`

เงื่อนไข `i < j` ข้างบนนี้จะเป็นเท็จได้ (เพื่อให้ออกจากวงวน) ก็ต่อเมื่อการที่ค่าของ `i` เพิ่มขึ้น หรือค่าของ `j` ลดลง
คำสั่ง `i += 2` ในตัวอย่างข้างบนสร้างความมั่นใจว่า วงวนนี้ทำงานแล้วจะมีจุดถ้วนสุดและออกจากวงวน

Flowchart	Code	Flowchart	Code
	<pre> k = 0 while k < n : P1 k += 1 </pre>		<pre> for k in range(n) : P1 </pre>
			<pre> for p in range(m) : for k in range(n) : P1 </pre>

Flowchart	Code
	<pre> for k in range(m) : P1 if C : P3 break P2 else: P4 </pre> <p>จะมาทำหลัง else ของ for ก็เมื่อทำครบทุกรอบ หลังทำรอบที่ $k = m-1$ เสร็จ ก็มาทำที่ P4 ก่อนออกจากวงวน</p>
	<pre> while C1 : P1 if C2 : P3 break P2 else: P4 </pre> <p>จะมาทำหลัง else ของ while ก็เมื่อทำงานเงื่อนไข C1 ของ while เป็นเท็จ ก็มาทำที่ P4 ก่อนออกจากวงวน</p>

range(start, stop, step)

- start, stop และ step ต้องเป็นจำนวนเต็ม
- `for k in range(10) :` $k = 0, 1, 2, \dots, 9$
- `for k in range(2,10) :` $k = 2, 3, 4, \dots, 9$
- `for k in range(2,10,2) :` $k = 2, 4, 6, 8$
- `for k in range(10,1,-2) :` $k = 10, 8, 6, 4, 2$
- `for k in range(11,11) :` ไม่ทำการกรอง เพราะ step เป็นบวก และ $start \geq stop$
- `for k in range(9,10,-1) :` ไม่ทำการกรอง เพราะ step ติดลบ และ $start \leq stop$

*** `break` จะยกการทำงานออกจากวงวนที่ `break` นั้นอยู่เท่านั้น

```

for i in range(5):
    for j in range(6):
        if condition1 :
            break          # break นี้ออกจาก for j
        if condition2 :
            break          # break นี้ออกจาก for i

```

วงศ์บกีพบปอย

<p>เมื่อต้องการทำให้เรบางอย่างซ้ำกัน n ครั้ง</p> <p>ใช้ <code>for k in range(n)</code></p>	<p>เมื่อ ต้องการอ่านข้อมูลจำนวน 10 ตัว เพื่อหาค่าเฉลี่ย</p> <pre>s = 0 for k in range(10) : # เป็นแค่คำสั่งควบคุมจำนวนรอบการทำซ้ำ a = float(input()) s += a print('average =', (s/10))</pre>		
<p>เมื่อต้องการนำค่าที่ร่างจาก <code>range(start, stop, step)</code> มาใช้ในการประมวลผล</p> <p>ใช้ <code>for k in range(...)</code></p>	<p> เช่น ต้องการหาว่า q เป็นจำนวนเฉพาะหรือไม่ ใช้วงวน for หาว่า จำนวนเต็มมากกว่าหนึ่งขนาดเล็กสุดอะไร ที่หาร q ลงตัว</p> <pre>for k in range(2, q+1) : if q % k == 0 : break # มีการนำ k มาใช้ if k == q : print(q, 'is prime') else: print(q, '=', k, 'x', q//k)</pre>		
<p>เมื่อต้องการประมวลผลชุดคำสั่งซ้ำ ๆ จนกว่าเงื่อนไขหนึ่งจะเป็นจริง</p> <p>ใช้ <code>while</code> หรือใช้ <code>if break</code> ในวงวน</p>	<p> เช่น ต้องการหาค่าเฉลี่ยจากชุดข้อมูลที่ผู้ใช้ป้อนเข้ามาเรื่อย ๆ จนกว่าจะรับจำนวนติดลบ</p> <table border="1" data-bbox="543 880 1310 1224"> <tr> <td data-bbox="543 880 916 1224"> <pre>s = 0 n = 0 while True : t = float(input()) if t < 0 : break s += t n += 1 if n == 0 : print('No Data') else : print('avg =', (s/n))</pre> </td><td data-bbox="923 880 1310 1224"> <pre>s = 0 n = 0 t = float(input()) while t >= 0 : s += t n += 1 t = float(input()) if n == 0 : print('No Data') else : print('avg =', (s/n))</pre> </td></tr> </table>	<pre>s = 0 n = 0 while True : t = float(input()) if t < 0 : break s += t n += 1 if n == 0 : print('No Data') else : print('avg =', (s/n))</pre>	<pre>s = 0 n = 0 t = float(input()) while t >= 0 : s += t n += 1 t = float(input()) if n == 0 : print('No Data') else : print('avg =', (s/n))</pre>
<pre>s = 0 n = 0 while True : t = float(input()) if t < 0 : break s += t n += 1 if n == 0 : print('No Data') else : print('avg =', (s/n))</pre>	<pre>s = 0 n = 0 t = float(input()) while t >= 0 : s += t n += 1 t = float(input()) if n == 0 : print('No Data') else : print('avg =', (s/n))</pre>		
<p>เมื่อต้องการแยกแยะวิธีการเลือกหมายเลขอจากหมายเลข 0 ถึง n-1 จำนวน 2 หมายเลข แบบเลือกแล้วไม่เลือกอีก (คือแยกแยะการเลือกหมายเลขออกมาทีละคู่)</p> <pre>for i in range(n) : for j in range(i+1, n) : 'ได้ i < j ทุก ๆ กรณี' หรือถ้าต้องการให้ i = j ด้วย ก็เป็น for i in range(n) : for j in range(i, n) : 'ได้ i ≤ j ทุก ๆ กรณี'</pre>	<p> เช่น จะหาว่ามีจำนวนเต็ม x y และ z อะไรมากที่ทำให้สมการ $z^3 = x^2 + y^2$ เป็นจริง (x กับ y มีค่า 0 ถึง 19) เช่น $5^3 = 5^2 + 12^2$ แต่เราไม่ต้องการคำตอบ $5^3 = 10^2 + 5^2$ เพราะซ้ำ จึงต้องกำหนดว่า $x < y$ แต่ถ้าเราต้องการคำตอบ $8^3 = 16^2 + 16^2$ ด้วย ก็ต้องให้ $x \leq y$ โดยให้ y มีค่าเริ่มที่ x เป็นต้นไป ด้วย <code>for y in range(x,n)</code> ข้างล่างนี้</p> <pre>n = 20 for x in range(1,n) : for y in range(x,n): t = x**2 + y**2 z = int(round(t**(1/3),0)) # หารากที่สามแล้ว # ปัดเศษ if z**3 == t : print(z,x,y)</pre>		

เรื่องผิดบ่อ

เข้าใจผิดเรื่องตัวสุดท้ายของ range

ใช้วงวน ทำอะไรมากอย่างเดอสรุปว่า เป็น A หรือ เป็น B โดยจะ ถ้า A เทอเงื่อนไข C เป็นจริง อย่างน้อยหนึ่งครั้ง จะเป็น B เมื่อ C ต้องไม่เป็นจริงทุกครั้งทุกรอบ ถ้าเขียน

```
for k in range(...):
    if C:
        print('A')
    else:
        print('B')
```

แบบนี้ผิด เพราะสรุปว่าเป็น B เร็วไป ยังตรวจสอบทุกรอบทุกกรณี แก้ไขด้วยการใช้ตัวแปรเก็บสภาพการตรวจ

```
found = False
for k in range(...):
    if C:
        print('A')
        found = True
        break
if not found:
    print('B')
```

หรือใช้ for-else ก็ง่ายกว่า

```
for k in range(...):
    if C:
        print('A')
        break
else:
    print('B')
```

วงวน ทำจำนวนรอบน้อยไปหรือมากไปกว่าที่ต้องการ (โดยทั่วไปมากขาดหรือเกินไปหนึ่งรอบ)

`for k in range(1,5) k = 1,2,3,4 (ไม่รวม 5)`

เช่น ต้องการหาว่า q เป็นจำนวนเฉพาะหรือไม่

```
for k in range(2, q):
    if q % k == 0:
        print(q,'is composite') # สรุปถูก เพราะ
                                # หาตัวหารพบ
    else:
        print(q,'is prime')     # ผิด สรุปเร็วไป
                                # ต้องวนทดสอบต่อ
```

แก้โดยใช้ตัวแปรเก็บสภาพการตรวจ เป็นดังนี้

```
iscomposite = False
for k in range(2,q):
    if q % k == 0:
        print(q,'is composite')
        iscomposite = True
        break
if not iscomposite:
    print(q,'is prime')

หรือใช้ for-else ก็ง่ายกว่า

for k in range(2,q):
    if q % k == 0:
        print(q,'is composite')
        break
else:
    print(q,'is prime')
```

เช่น จากชุดคำสั่งตรวจสอบจำนวนเฉพาะข้างบนนี้ ถ้าเขียน

```
for k in range(2,q+1):    # เขียน q+1 แทนที่จะเป็น q
    if q % k == 0:
        print(q,'is composite')
        break
    else:
        print(q,'is prime')
```

แบบนี้เกินไปรอบ ทำให้ผลลูกคามาเป็น composite เช่น 0 เพราะอะไร ?

หรือ อยากรหาค่าของ $\sum_{k=1}^n k^3$ ถ้าเขียน

```
s = 0
for k in range(1,n):
    s += k**3
print(s)

ก็จะพบว่าขาดไปรอบ
```

ลีมปรับค่าของตัวแปรที่ใช้ในเงื่อนไขของ while หรือไม่ก็ปรับค่าผิด ความผิดพลาดแบบนี้อาจทำให้วันทำงานไม่สิ้นสุด

เช่น ต้องการวนรับจำนวนจากผู้ใช้ มาหาค่าเฉลี่ยจนกว่าจะพบจำนวนลบ

```
s = n = 0
t = float(input())
while t >= 0 :
    s += t
    n += 1
print('avg =', (s/n))
```

ใช้ t ใน การตรวจสอบเงื่อนไขของ while แต่ค่า t ไม่ได้เปลี่ยนแปลงเลยในวงวน ถ้าหลุดเข้ามานในวงวนได้จะเกิดอะไรขึ้น ?
ควรแก้เป็น

```
s = n = 0
t = float(input())
while t >= 0 :
    s += t
    n += 1
    t = float(input()) # เพิ่มบรรทัดนี้
print('avg =', (s/n))
```

ตั้งค่าให้กับตัวแปรที่ควรจะให้ค่าก่อนเข้าวงวน แต่กลับไปเปลี่ยนในวงวน

เช่น ต้องการวนรับจำนวนจากผู้ใช้ มาหาค่าเฉลี่ยจนกว่าจะพบจำนวนลบ

```
while True :
    s = n = 0 # บรรทัดนี้ไม่มีมาอยู่ในวงวน
    t = float(input())
    if t < 0 : break
    s += t
    n += 1
print('avg =', (s/n))
จะเกิดอะไรขึ้น ?
```

เรื่อง॥ลอกของ for ใน Python

หลังจากวนทำงานใน for k จะเรียบร้อยแล้ว ค่า k หลังออกจากวงวน จะมีค่าเท่ากับค่าสุดท้ายที่ทำงานในวงวน

```
for k in range(1,5):
    ...
print(k) # ได้ 4 เพราะเป็นค่าสุดท้ายที่ทำในวงวน

for m in range(4,10):
    if m % 3 == 0 : break
    ...
print(m) # ได้ 6 เพราะค่าสุดท้ายในวงวนคือ 6
ก่อนจะ break ออก

for w in range(10,10):
    ...
print(w) # ผิด เพราะไม่ได้เข้าไปทำในวงวน, w จึงไม่มีค่า
```

<p>การปรับค่า k ภายในวงวน</p> <pre>for k in range(...)</pre> <p>จะไม่มีผลต่อการเปลี่ยนค่า k ในรอบถัดไป</p> <p> เพราะฉะนั้น ถ้าต้องการปรับค่าของ k ในวงวน ต้องใช้วงวน while แทน</p>	<pre>for k in range(1, 4) : print(k) k += 2 print(k) # ได้ผลตามทางขวาบน</pre> <p>ถ้าต้องการเปลี่ยน k ต้องใช้ while</p> <pre>k = 1 while k < 4 : print(k) k += 2 print(k)</pre>
--	---



Problem	Code
<p><u>Input:</u> ไม่มี</p> <p><u>Process:</u> หาจำนวนเต็มบวก k ที่มีค่าน้อยสุดที่ทำให้มี $\left(\frac{1}{k}\right) k$ คำไม่เท่ากับ 1 (เนื่องจากความไม่แน่นยำ 100% ของจำนวนจริงในคอมพิวเตอร์)</p> <p><u>Output:</u> จำนวนเต็ม k ที่หาได้</p>	
<p><u>Input:</u> ไม่มี</p> <p><u>Process:</u> หาจำนวนเต็มบวก k ที่มีค่าน้อยสุดที่ทำให้ $1 - \left(\frac{365}{365}\right) \left(\frac{365-1}{365}\right) \left(\frac{365-2}{365}\right) \cdots \left(\frac{365-k}{365}\right) \geq 0.5$ เป็นจริง</p> <p><u>Output:</u> จำนวนเต็ม k ที่หาได้</p>	
<p><u>Input:</u> ไม่มี</p> <p><u>Process:</u> คำนวณค่าประมาณของ π จากสูตร $4 \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \cdots + \frac{1}{399997} - \frac{1}{399999}\right)$</p> <p><u>Output:</u> ค่าประมาณของ π ที่หาได้</p>	
<p><u>Input:</u> อ่านจำนวนเต็ม 2 จำนวน a กับ b</p> <p><u>Process:</u> คำนวณค่าจากสูตร $\sum_{i=a}^{b-1} \left((-1)^i \sum_{j=i+1}^b (i+j) \right)$</p> <p><u>Output:</u> ค่าที่คำนวณได้</p>	
<p><u>Input:</u> อ่านจำนวนเต็ม 2 จำนวน a กับ b</p> <p><u>Process:</u> คำนวณค่าจากสูตร $\sum_{a \leq i < j \leq b} (-1)^i (i+j)$</p> <p><u>Output:</u> ค่าที่คำนวณได้</p>	

Problem	Code
<p><u>Input:</u> บรรทัดแรกรับจำนวนเต็มเก็บในตัวแปร n (n จะมีค่ามากกว่า 0) และอีก n บรรทัด บันทึกจำนวนเต็มบรรทัดละจำนวน</p> <p><u>Process:</u> หาผลต่างของค่ามากสุดกับค่าขั้นต่ำ และหาร่วมกับจำนวนที่เป็นเลขลบ ให้ได้</p> <p><u>Output:</u> ผลต่าง และจำนวนเลขลบ ให้ได้</p>	
<p><u>Input:</u> จำนวนเต็มเก็บในตัวแปร n</p> <p><u>Process:</u> หาจำนวนเต็มบวก w, x, y และ z ทั้งหมดที่ $w^3 = x^2 + y^2 + z^2$ โดยที่ $1 \leq x \leq y \leq z \leq n$</p> <p><u>Output:</u> ค่าของ w, x, y และ z ให้ได้</p>	



Approximation of sine

ค่าของ $\sin(x)$ คำนวณได้ด้วยอนุกรมเทย์เลอร์ ดังแสดงข้างล่างนี้

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!}$$

จะเขียนโปรแกรมรับค่าของ x เพื่อคำนวณค่า $\sin(x)$ ให้ได้ความแม่นยำมากที่สุดเท่าที่จะมากได้ ด้วยสูตรข้างบนนี้

► ข้อมูลนำเข้า

หนึ่งบรรทัดประกอบด้วยจำนวนจริง x (หน่วยเป็นองศา)

► ข้อมูลส่งออก

ค่าประมาณของ $\sin(x)$ จากสูตรข้างต้น

► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
45	0.7071067811865475
36045	0.7071067811865475

ตัวอย่างการเขียนโปรแกรม

โปรแกรม	จำรีบาร
<pre>import math x = math.radians(float(input())) n = 100 s = f = x for k in range(1,n+1): f *= (-1)*x**2 / ((2*k)*(2*k+1)) s += f print(s)</pre>	<p>แล้วปัญหาโดยไม่ใช่จำนวนเต็มขนาดใหญ่ แต่เมื่อว่า การคำนวนหา พจน์ในแต่ละรอบนั้น สามารถคำนวนได้จากพจน์ในรอบก่อนได้ ไม่น่าต้องคำนวน $(-1)^k$, x^{2k+1} และ $(2k+1)!$ ทุกรอบ แต่สามารถ คำนวนจากความสัมพันธ์ดังนี้</p> $(-1)^k = (-1)((-1)^{(k-1)})$ $x^{2k+1} = x^2(x^{2(k-1)+1})$ $(2k+1)! = (2k+1)(2k)((2(k-1)+1)!)$ <p>ดังนั้น</p> $(-1)^k \frac{x^{2k+1}}{(2k+1)!} = \underbrace{(-1)}_{\text{พจน์ในรอบที่ } k} \underbrace{\frac{x^2}{(2k+1)(2k)}}_{\text{ตัวคูณ}} \underbrace{((-1)^{(k-1)} \frac{x^{2(k-1)+1}}{(2(k-1)+1)!})}_{\text{พจน์ในรอบที่ } k-1}$ <p>เริ่มด้วย $f = x$ (คือพจน์ที่ $k = 0$)</p> $\begin{aligned} k=1 \quad f &= (-1)*x**2 / ((2*k+1)*(2*k)) * f \\ &= -(x**2)/(3*2) * x \\ &= -(x**3)/(3!) \\ k=2 \quad f &= -(x**2)/(5*4) * -(x**3)/(3!) \\ &= +(x**5)/(5!) \\ k=3 \quad f &= -(x**2)/(7*6) * (x**5)/(5!) \\ &= -(x**7)/(7!) \\ &\dots \end{aligned}$ <p>ด้วยแนวคิดนี้เขียนโปรแกรมได้ดัง code ทางซ้าย สั่ง run, ใส่ 45 เป็นข้อมูล, ได้ผลเป็น 0.7071067811865475 ถูกต้อง ถ้าลองเพิ่มคำสั่ง <code>print(k, s)</code> ในวงวน for จะพบว่า ค่า s มีค่าไม่เปลี่ยนแปลงตั้งแต่ $k = 8$ เป็นต้นไป แสดงว่าการคำนวน ตั้งแต่รอบที่ 8 เป็นต้นไป ไม่มีประโยชน์เลย แต่ถ้าใส่ $x = 180$ เป็นข้อมูลน้ำเข้า พบว่า $k = 22$ เป็นต้นไปจะไม่เปลี่ยนแปลง</p>
<pre>import math x = math.radians(float(input())) s = f = x k = 1 while f != 0: f *= (-1)*x**2 / ((2*k)*(2*k+1)) s += f k += 1 print(s)</pre>	<p>จึงควรปรับการทำงานของโปรแกรมให้ตรวจสอบว่า หากค่าในตัวแปร s ไม่เปลี่ยนแปลง หรือใช้วิธีตรวจสอบว่า เมื่อ f เป็น 0 ก็สามารถ หยุดการคำนวนได้ โดยเปลี่ยนจากการใช้ วงวน for เป็นวงวน while จะเหมาะสมกว่า โดยมีเงื่อนไข while $f \neq 0$ เป็นตัวกำหนดว่า ต้องทำต่อในวงวน เมื่อ f ยังไม่เท่ากับ 0 ตัวแปร k เริ่มต้นที่ 1 ในรอบแรก และ เพิ่มทีละ 1 ในรอบถัด ๆ ไป</p> <p>สั่ง run, ใส่ 45 เป็นข้อมูล, ได้ผลเป็น 0.7071067811865475 ถูกต้อง และถ้าลอง <code>print(k)</code> ดูใน Python shell จะพบว่า $k = 8$ เมื่อออกจากวงวน ด้วยวิธีนี้เราไม่ต้องกำหนดจำนวนรอบให้กับวงวน มาลองตัวอย่างที่ 2 สั่ง run, ใส่ 36045 เป็นข้อมูล, ได้ผลเป็น 2.541635699930208e+252 ผิด ก็คงอะไรขึ้น ?</p>

โปรแกรม	คำอธิบาย
<pre>import math x = float(input()) x = math.radians(x%360) s = f = x k = 1 while f != 0: f *= (-1)*x**2/(2*k)*(2*k+1) s += f k += 1 print(s)</pre>	<p>ค่า 36045 เมื่อแปลงเป็นเรเดียนแล้ว x มีค่ามาก การคำนวณค่า f ในวงวนจะลดความแม่นยำลงด้วยเหตุที่จำนวนจุดหลังเศษนิยมของ float มีจำนวนจำกัด ทำให้การคำนวณผิดพลาด ด้วยคุณสมบัติของฟังก์ชัน sin ที่เป็นเชิงคاب เราชารลดขนาดของค่า x ลงด้วยการเปลี่ยนค่า 36045 องศาเป็น 36045 % 360 = 45 องศา ก่อนเปลี่ยนเป็นเรเดียน จะเพิ่มความแม่นยำและลดความผิดพลาดในการคำนวณ ได้ผลดัง code ทางซ้าย</p>

ตัวอย่างโจทย์ปัญหา

Multiples of 3 or 5

จงเขียนโปรแกรมที่คำนวณผลรวมของจำนวนเต็มบวกทุกจำนวนที่มีค่าต่ำกว่าจำนวนที่เป็นข้อมูลนำเข้าและมี 3 หรือ 5 เป็นตัวประกอบ เช่น หากข้อมูลนำเข้าคือ 20 คำตอบที่เราต้องการจะเท่ากับ $3 + 5 + 6 + 9 + 10 + 12 + 15 + 18 = 78$ (สังเกตว่าคำตอบของเรามิ่งรวมค่า 20 เนื่องจากเราสนใจเฉพาะจำนวนที่มีค่าต่ำกว่า 20)

▶ ข้อมูลนำเข้า

มีบรรทัดเดียว เป็นจำนวนเต็มบวก

▶ ข้อมูลส่งออก

มีบรรทัดเดียว แสดงผลรวมของจำนวนเต็มบวกทุกจำนวนที่มีค่าต่ำกว่าจำนวนที่เป็นข้อมูลนำเข้าและมี 3 หรือ 5 เป็นตัวประกอบ

▶ ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
20	78
25	143
3	0

Average until -1

ให้อ่านข้อมูลจากแป้นพิมพ์ที่เป็นจำนวน ริงๆ กว่าจะพบค่า -1 เพื่อหาค่าเฉลี่ยของจำนวนเหล่านั้นทั้งหมด (ไม่รวม -1)

▶ ข้อมูลนำเข้า

จำนวนจริงบรรทัดละ 1 จำนวนหลังบรรทัด 1 บรรทัดสุดท้ายคือ -1

▶ ข้อมูลส่งออก

เมื่อรับตัวเลข แสดงค่าเฉลี่ยของจำนวนทั้งหมด (ไม่รวม -1) ออกทางหน้าจอ
ในกรณีที่มีจำนวนข้อมูลเป็น 0 ให้แสดง No Data

▶ ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
2295.498850599365 8502.139421733784 515.0100470901091 3705.6829835190097 8722.343211974356 4446.712951812571 6375.086965715 3801.511489785674 7638.911577747659 -1	5111.43305555306
-1	No Data

สามเหลี่ยมมุมจาก

โจทย์นี้สนใจเฉพาะสามเหลี่ยมมุม钝角 กรณี จำนวนด้านทุกด้านเป็นจำนวนเต็ม

จงเขียนโปรแกรมอ่านค่าคงที่ a เส้นรอบรูปจากแป้นพิมพ์ เพื่อหาจำนวนเต็มมากสุดที่เป็นความยาวของด้านตรงข้ามมุมจากของสามเหลี่ยมมุมจากที่มีความยาว a รันรอบรูปตามที่ได้รับ เช่น ให้เส้นรอบรูปของสามเหลี่ยมมุมยาว 90 จะมีสามเหลี่ยมมุมจากตามข้อกำหนดอยู่สองรูปคือ 15, 36, 39 และ 9, 40, 41 คำตอบที่ต้องการคือ 41 เพราะเป็นความยาวด้านตรงข้ามมุมจากที่ยาวสุดของสามเหลี่ยมมุมจากตามข้อก หนท และมีเส้นรอบรูปยาว 90

► ข้อมูลนำเข้า

มีบรรทัดเดียวเป็น จำนวนเต็มบวก แทนความยาวเส้นรอบรูปของสามเหลี่ยมมุมจากตามข้อกำหนด
(รับประกันว่า มีสามเหลี่ยมมุมจากที่มีความยาวเส้นรอบรูปเท่ากับจำนวนที่เป็นข้อมูลนำเข้า)

► ข้อมูลส่งออก

มีบรรทัดเดียวแสดงความยาวของด้านตรงข้ามมุมจากที่ยาวที่สุดของสามเหลี่ยมมุมจากตามข้อกำหนด

► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
30	13
90	41

S นางสาวบุญมาลาบุปผา

Intania 84

CEO & Co-Founder of Priceza



Programming เป็นหนึ่งในวิชาที่ผมชอบเรียนมากที่สุดวิชานึงเลยตอนปี 1 เหตุผล เพราะว่ามันช่วยให้ผมมีกระบวนการคิดและการแก้ปัญหาต่าง ๆ อย่างเป็นระบบ จนมาถึงตอนที่อยากทำ Startup สร้าง Priceza ขึ้นมา ด้วยความเข้าใจใน Programming ทำให้ผมมีพื้นฐานวิธีคิดในการศึกษาพัฒนาระบบท่าง ๆ ที่เราไม่เคยพัฒนาขึ้นมาก่อน และทำให้เราศึกษาและวิจัยตั้งแต่ต้น จนพัฒนามันขึ้นมาได้ในที่สุด



ลีสา ตรงประกอบ

Intania 81

Google Inc.

Programming ทำให้เรารู้จักคิดและรู้จักสื่อสาร และไม่ใช่การคิดและการสื่อสารแบบทั่ว ๆ ไป แต่เป็น การคิดอย่างเป็นระบบ คิดอย่างรอบคอบ คิดในทุก ๆ ความเป็นไปได้ และเป็นการสื่อสารที่ต้องซัดเจนที่ทั้ง ผู้ส่งสารและผู้รับสารต้องเข้าใจไปในทางเดียวกัน ถ้ามีอะไรผิดพลาดแม้เพียงนิดเดียว ล้มคิดไปบินดีนึง สื่อสารผิดไปเล็กน้อย ผลก็จะออกมามาไม่ได้อย่างที่วางแผนไว้ การคิดและการสื่อสารแบบนี้สามารถนำไปใช้กับอะไรก็ได้ และสามารถสร้างผลลัพธ์เป็นอะไรได้ตามแต่ใจเรารอ York ให้เป็น

04 : String

สรุปเนื้อหา

สตริง (string) เก็บ 문자 (character) ตั้งแต่ศูนย์ตัวขึ้นไป เรียงจากซ้ายไปขวาต่อกันไป แต่ละตัวมีเลข index ระบุตำแหน่ง โดย 0 ของภาษาสุดคือ 0 อักษรแต่ละตัวในสตริงเป็นได้ทั้งอักษร ตัวเลข และสัญลักษณ์พิเศษต่าง ๆ การเขียนสตริงที่ได้หลักแบบดังนี้

- เขียนครอบด้วย อัญประกาศเดียว 'I am "Python".'
- เขียนครอบด้วย อัญประกาศคู่ "I'm Python."
- เขียนครอบด้วย อัญประกาศเดียวสามตัวติด '''I'm "Python'''
- เขียนครอบด้วย อัญประกาศคู่สามตัวติด """I'm "Python""""

หมายเหตุ : 12 เป็น int ไม่ใช่สตริง แต่ '12' คือสตริง ถ้าต้องการแปลงจำนวนในตัวแปร x ให้กลายเป็นสตริง ใช้ str(x)

ตัวอย่างการเข้าใช้อักษรและสตริงย่อยในสตริง (สมมติให้ s = "ABCDEFG")

- len(t) ได้จำนวนตัวอักษรใน t โดย len('') ได้ 0
- s[0] เมื่อ s[-len(s)] ได้ "A" ส่วน s[-1] เมื่อ s[len(s)-1] ได้ "G"
- อย่าลืมว่า index ของสตริง s ต้องอยู่ในช่วง 0 ถึง len(s)-1 จากซ้ายไปขวา และ -1 ถึง -len(s) ถอยจากขวาไปซ้าย ดังนั้นเราเขียน s[k] ได้โดยที่ $-len(s) \leq k \leq (len(s)-1)$ เพราะฉะนั้น "01234"[-6] กับ "01234"[5] ผิด
- s เมื่อ s[:] เมื่อ s[0:] เมื่อ s[:len(s)] เมื่อ s[::] เมื่อ s[::1]
- s[::-2] หยิบตัวที่ index คือได้ "ACEG", s[1::2] หยิบตัวที่ index คือได้ "BDF"
- s[::-1] เมื่อ s[-1::-1] เมื่อ s[-1:-(len(s)+1):-1] ได้ "GFEDCBA"
- ถ้าเขียน s[a:b] เพื่อเลือกสตริงย่อยออกมา ค่า a กับ b เป็นอะเรกเก็ตได้ ไม่ผิด
 - "01234"[2:50000] ได้ "234", "01234"[4999:50000] ได้ ""
 - "01234"[-500:-2] ได้ "012", "01234"[-3:-500:-1] ได้ "210", "01234"[-500:-300] ได้ ""
- ใช้ for c in s : เพื่อแจกแจงอักษรทีละตัวใน s จากซ้ายไปขวาเก็บในตัวแปร c นำไปใช้ในวงวนี้ได้

ตัวอย่างการจัดการสตริง (ให้ s = " Python 3.6 ")

- t = s.upper() ได้ t เก็บ " PYTHON 3.6 " s เมื่อันเดิม
- t = s.lower() ได้ t เก็บ " python 3.6 " s เมื่อันเดิม
- t = s.strip() ได้ t เก็บ "Python 3.6" s เมื่อันเดิม
- s = s.strip().upper() ได้ s เก็บ "PYTHON 3.6"
- k = s.find(c) คืน index น้อยสุดที่พบ c ใน s เริ่มค้นตั้งแต่ index 0 ถ้าไม่พบ จะได้ผลเป็น -1
k = "engineering".find("ng") ได้ k เก็บ 1 เพราะ "ng" ปรากฏเริ่มที่ index 1 ใน "engineering" คำสั่ง if c in s ก็เหมือนกับ if s.find(c) >= 0
- k = s.find(c,j) คืน index น้อยสุดที่พบ c ใน s เริ่มค้นตั้งแต่ index j เป็นต้นไป
- นำสตริงบวกกัน คือนำสตริงมาต่อกัน เช่น '12'+'23' คือ '1223'
- สตริงคูณกับจำนวนเต็ม คือนำสตริงนั้นมาต่อ กันเป็นจำนวนครั้งเท่ากับค่าของจำนวนเต็มนั้น เช่น '12'*3 คือ '121212'

ตัวอย่างการจัดการสตริง

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

รูปแบบการประมวลผลสตริงที่พบบ่อย

ต้องการหยิบอักขระในสตริงจาก ซ้ายไปขวา มาประมาณผลที่ละตัว	ต้องการนับว่าสตริง <code>s</code> มีตัวเลขกี่ตัว <code>c = 0</code> <code>for e in s :</code> <code> if '0' <= e <= '9' :</code> <code> c += 1</code>
ต้องการหยิบที่ละตัว พร้อมกับ index ของตัวนั้น ๆ	<code>for i in range(len(s)):</code> <code> c = s[i]</code> <code> print(i,c)</code> หรือ <code>for i,c in enumerate(s):</code> <code> print(i,c)</code>
ต้องการหยิบอักขระในสตริงจาก ขวามาซ้ายที่ละตัว	<code>for e in s[::-1] :</code> <code> ...</code> หรือ <code>for k in range(-1,-(len(s)+1),-1):</code> <code> e = s[k]</code> <code> ...</code>
ต้องการหยิบอักขระในสตริงจาก ซ้ายไปขวา มาประมาณผลที่ละคู่ที่ ติดกัน	ต้องการนับว่าสตริง <code>s</code> มีตัวที่ติดกันเป็นตัวเลขทั้งคู่อยู่กี่คู่ <code>c = 0</code> <code>for k in range(len(s)-1) : # ต้องการตัวติดกัน จึงวนถึงตัวรองลุดท้าย</code> <code> if '0' <= s[k] <= '9' and '0' <= s[k+1] <= '9' :</code> <code> c += 1</code>

<p>ถ้าต้องการนับว่ามี t ปรากฏอยู่ใน s กี่ครั้ง</p> <pre>c = 0 while t in s : c += 1</pre> <p>แบบนี้ผิด ถ้า มี t ใน s เนื่องจาก t in s จะเป็น True ตลอด ก็จะวนไม่รู้จบ</p>	<pre>c = 0 k = s.find(c) while k != -1 : c += 1 k = s.find(t,k+1) # 'aaaaaa' มี 'aaa' 4 ครั้ง หรือ c = 0 for i in range(len(s)-len(t)+1) : if s[i:i+len(t)] == t : c += 1</pre>
<p>ใช้สตริงสะสมข้อมูลพอนำมาแสดงทีหลัง</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>ต้องการหยิบสตริงย่ออย่างที่อยู่ระหว่างรูปแบบสตริงที่กำหนด</p>	<p>หาสตริงย่ออย่างที่อยู่ระหว่าง "<title>" และ "</title>" (ให้ถือว่า ถ้าสตริงที่ได้รับมี <title> ก็ต้องมี </title> ด้วย)</p> <pre>t = input().strip() a = t.find("<title>") if a >= 0 : j = a+len("<title>") # a j b b = t.find("</title>",j) print(t[j:b]) else : print('No title')</pre>
<p>ใช้สตริงเป็นที่เก็บข้อมูลเพื่อแปลงจากจำนวนเต็มเป็นสตริง</p>	<p>แปลงเลขเดือนใน m เป็นชื่อย่อเดือน</p> <pre>months = "JANFEBMARAPR MAY JUN JULAUG SEPOCT NOV DEC" m = int(input()) if 1 <= m <= 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 = "usthjpcnsg" currency_codes = "USDTHBJPYCNYSGD" k = countries.find(c) if k >= 0: k // 2 print(currency_codes[3*k:3*(k+1)]) else : print('Not found')</pre>

เรื่องผิดป่วย

เครื่องหมายเปิดปิดสตริง! ครบ
ไม่ตรงกัน หรือใส่ซ้อนกัน. โดย
ไม่นำหน้าด้วย \

```
s = "String"
s = String'
s = "string'
s = "You shouted "Hey!" very loudly yesterday"
```

ถ้าต้องการรับสตริงจาก
แป้นพิมพ์ ควร strip ด้วย
 เพราะผู้ใช้อาจแพลเเพมช่องว่าง
 ทางซ้ายหรือขวาเพิ่มเติม

คำสั่งข้างล่างนี้ ถ้าผู้ใช้ป้อน ok ตามด้วยช่องว่างแล้วกด enter
 การเปรียบเทียบก็จะไม่เท่า 'ok'

```
s = input()
if s == 'ok' :
    print('OK')

# จึงควรเขียน s = input().strip()
# (ยกเว้นกรณีที่เราต้องการคงทุกสิ่งที่ผู้ใช้ป้อนทั้งหมด)
```

ใช้ index ที่เกินช่วงที่ใช้ได้
 ของสตริง อย่าลืมว่า index
 ของสตริง s ออยู่ในช่วง 0 ถึง
 len(s)-1 จากซ้ายไปขวา
 และ -1 ถึง -len(s)
 ถอยจากขวามาซ้าย

```
s = input().strip()
t = ""
for i in range(10):
    t = s[i] + t # ถ้า len(s) < 10 ก็ผิดแน่
```

เลข index ที่ใช้กับสตริงไม่ใช่
 จำนวนเต็ม

```
print( s[n/2] ) # ผิด เพราะ n/2 ได้จำนวนจริง
```

เปลี่ยนข้อมูลภายในสตริง

```
t = input().strip()
t[0],t[-1] = t[-1],t[0] # ลับตัวแรกกับตัวท้ายแบบนี้ผิด
ถ้าต้องการลับตัวแรกกับตัวท้าย ต้องสร้างใหม่
t = t[-1] + t[1:-1] + t[0]
```

เขียนสตริงระบุตำแหน่งของ
 แฟ้มข้อมูลที่มีเครื่องหมาย \
 แต่ไม่ได้เขียน \\

```
s = "c:\temp\data\input.txt" # ผิด
ต้องเป็น
s = "c:\\temp\\\\data\\\\input.txt"
```

นำสตริงไปบวกกับข้อมูล
 ประเภทอื่น

```
print("average = " + avg) # ผิด
ต้องเป็น
print("average = " + str(avg)) หรือ
print("average =", avg)
```



แบบฝึกหัด

Problem	Code
<p><u>Input:</u> รับสตริงหนึ่ง บรรทัด</p> <p><u>Process:</u> สร้างสตริงใหม่ที่ทุกอักษรในสตริงที่รับเข้ามาปรากฏซ้ำ อีกด้วย เช่น รับ 'pypy' จะได้สตริง 'ppyyppyy'</p> <p><u>Output:</u> สตริงผลลัพธ์</p>	
<p><u>Input:</u> รับสตริงหนึ่งบรรทัด</p> <p><u>Process:</u> สร้างสตริงใหม่ที่ทุกอักษรในสตริงที่รับเข้ามาปรากฏซ้ำ อีกด้วย แต่ถ้ามีตัวซ้ำติดกันอยู่แล้ว ก็ไม่ต้องทำซ้ำ อีกต่อไป เช่น รับ 'pythonnnnaa' จะได้สตริง 'ppyytthhoonnnaa'</p> <p><u>Output:</u> สตริงผลลัพธ์</p>	
<p><u>Input:</u> รับสตริงหนึ่งบรรทัด</p> <p><u>Process:</u> ตรวจว่าสตริงนี้เป็น palindrome (ซึ่คือสตริงที่กลับลำดับแล้วคือสตริงเดิม) 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 $\geq n$ หลัก ก็ให้เป็นสตริงของ d เดิม</p> <p><u>Output:</u> สตริงใหม่ที่ต้องการ</p>	

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

ตัวอย่างการแก้โจทย์ปัญหา

ตรวจสอบเลข

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

▶ ข้อมูลนำเข้า

รับสตริงหนึ่งบรรทัด

▶ ข้อมูลส่งออก

ถ้าสตริงที่รับมาไม่มีเลขซ้ำ แสดง True แต่ถ้าไม่มีเลขซ้ำ แสดง False

▶ ตัวอย่าง

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

ตัวอย่างการเขียนโปรแกรม

โปรแกรม	คำอธิบาย
<pre>t = ''\nfor e in input().strip() :\n if '0' <= e <= '9' :\n t += e\nprint(t)</pre>	อาจเริ่มด้วยการลบอักขระที่ไม่ใช่ตัวเลขออกจากสตริงที่รับมาก่อน หลังจากนั้นค่อยหวิธีตรวจสอบอักขระซ้ำ แต่เราเรียนมาว่า ไม่สามารถเปลี่ยนแปลง อักขระในสตริงได้ ดังนั้นต้องใช้วิธีการสร้างสตริงใหม่ เช่น ถ้าจะลบอักขระตัวที่ index 2 ออก ก็ทำการสร้างใหม่แล้วทับตัวเดิมด้วยคำสั่ง s = s[:2]+s[3:] วิธีนี้คือข้างบุ้ง จึงขอเปลี่ยนเป็นการเลือกเฉพาะตัวเลขในสตริงที่รับมา เพิ่มไส้สตริงใหม่อีกด้วยนั่น จะง่ายกว่า โปรแกรมทางช้าย เริ่มด้วยสตริงว่า t จากนั้นวนรอบอักขระของมาทีละตัว จากสตริงที่รับเข้ามา ถ้าอักขระนั้นเป็นตัวเลข ก็เพิ่มใส่ t สิ้ง run, ใส่ 12a34bce5, ได้ผลเป็น 12345, ถูกต้อง
<pre>t = ''\nfor e in input().strip() :\n if '0' <= e <= '9' :\n t += e\nb = ''\nfor e in t :\n if e in b : break\n b += e</pre>	เรารีด t เก็บสตริงที่มีตัวเลขอย่างเดียว ก็มาถึงขั้นตอนตรวจสอบว่า t ไม่มีเลขซ้ำหรือไม่ ? จะทำอย่างไร ? ถ้าเราค่อย ๆ ดูอักขระทีละตัวในสตริง ก็ต้องตรวจว่าอักขระใหม่นี้ปรากฏในอักขระที่ดูผ่านมาหรือไม่ ถ้าเราใช้วงวน for e in t ก็จะตัวที่ผ่านมาด้วยการใช้ตัวแปรใหม่ b เริ่มด้วยสตริงว่าง ถ้าตัวใหม่ไม่มีใน b เพิ่มตัวใหม่ต่อท้ายเข้าใน b ดัง code ทางช้ายนี้ เมื่อได้พบว่า e in b เป็นจริง คือตัวใหม่ซ้ำกับตัวที่ผ่านมา ก็ break ออกจากวงวนได้ (คือรู้ว่ามีเลขซ้ำแล้ว)

โปรแกรม	คำอธิบาย
<pre>t = '' for e in input().strip() : if '0' <= e <= '9' : t += e b = '' for e in t : if e in b : print(True) break b += e else: print(False)</pre>	<p>ແຕ່ນາງ b ອກຈາກງວນທັນທີທີ່ພບຕົວໜັ້ນ ພອອກຈາກງວນແລ້ວ ລາຍລະອຽດວ່າ ການຫຼຸດອອກຈາກງວນມາຈາກຮູນ <code>break</code> ຮີ່ອມາຈາກຮູນທີ່ໄວ້ ຄຽບທຸກອັກຂະແລ້ວ ໃນໂຈທົບອກວ່າ ດ້ວຍບວ່າໜ້າ ໃຫ້ແສດງ <code>True</code> ດ້ວຍເຫັນວ່າເລີຍໃຫ້ແສດງ <code>False</code> ດັ່ງນັ້ນຄວບ <code>print(True)</code> ເລີຍມີ່ວິພບວ່າໜ້າ ແລ້ວ <code>break</code> ອອກຈາກງວນ ສ່ວນການຕຽບວ່າດ້ວຍຄຽບທຸກຕົວໄດ້ມ່ວນຕົວທຸກຕົວ ແລ້ວໄມ່ພົບເລີຍໃຫ້ເລີຍ ໄດ້ <code>code</code> ທາງໜ້າຍ ສິ່ງ <code>run</code>, ໄສ່ ..125.9.0., ໄດ້ຜລເປັນ <code>False</code>, ແສດງວ່າມີ່ມີ້ໜ້າ, ຖຸກຕ້ອງ ສິ່ງ <code>run</code>, ໄສ່ ..125.9.2., ໄດ້ຜລເປັນ <code>True</code>, ແສດງວ່າມີ້ໜ້າ, ຖຸກຕ້ອງ</p>
<pre>t = '' for e in input().strip() : if '0' <= e <= '9' : t += e for k in range(len(t)) : if t[k] in t[:k] : print(True) break else: print(False)</pre>	<p>ດ້າວີດອືນດີ ພບວ່າ ພາກເຮົາວຸນຄຶງຮອບທີ່ <code>k</code> ການຕຽບວ່າດ້ວຍທີ່ <code>k</code> ຈຳກັບດ້ວຍທີ່ ຜ່ານມາຫຼືອ່ານີ້ ສາມາຄັດຕູ້ວ່າທີ່ຜ່ານມາຈາກສ່ວນທາງໜ້າຍຂອງ <code>t</code> ໄນເຫັນຈຳເປັນຕົ້ນ ສ້າງ <code>b</code> ໃນແບບທີ່ທຳມາ ດ້ວຍຄຳສັ່ງ <code>if t[k] in t[:k]</code> ຈຶ່ງປັບປຸງຈາກ <code>for e in t</code> ເປັນ <code>for k in range(len(t))</code> ໄດ້ສັ່ງ <code>code</code> ທາງໜ້າຍ ສິ່ງ <code>run</code>, ໄສ່ ..125.9.0., ໄດ້ຜລເປັນ <code>False</code>, ແສດງວ່າມີ່ມີ້ໜ້າ, ຖຸກຕ້ອງ ສິ່ງ <code>run</code>, ໄສ່ ..125.9.2., ໄດ້ຜລເປັນ <code>True</code>, ແສດງວ່າມີ້ໜ້າ, ຖຸກຕ້ອງ</p>
<pre>t = input().strip() for k in range(len(t)) : if '0' <= t[k] <= '9' and \ t[k] in t[:k] : print(True) break else: print(False)</pre>	<p>ເຮົາສາມາຄັດປັບປຸງຕ່ອງໄດ້ອີກ ໂດຍບູນຮ່ວມງວນ <code>for</code> ສອງງວນເຂົ້າດ້ວຍກັນ <code>for</code> ແກ່ງຈັດອັກຂະທີ່ໄນ້ໃຫ້ຕົວເລີນ <code>for</code> ພັນຍາກວ່າເວັບໄວ້ຕົວໜັ້ນ ເຖິງ <code>for</code> ເຊິ່ງ ຫຼຸໄລ່ໄປທີ່ລະດ້ວຍ ດ້ວຍໃຫ້ຕົວເລີນກີ່ຂໍາມໄປ ດ້ວຍໃຫ້ກີ່ຕຽບວ່າໜ້າກັນທີ່ຜ່ານມາຫຼືອ່ານີ້ໃນການເຕີຍວັບທີ່ທຳມາ ໄດ້ <code>code</code> ທາງໜ້າຍ ສິ່ງ <code>run</code>, ໄສ່ ..125.9.0., ໄດ້ຜລເປັນ <code>False</code>, ແສດງວ່າມີ່ມີ້ໜ້າ, ຖຸກຕ້ອງ ສິ່ງ <code>run</code>, ໄສ່ ..125.9.2., ໄດ້ຜລເປັນ <code>True</code>, ແສດງວ່າມີ້ໜ້າ, ຖຸກຕ້ອງ</p>
<pre>t = input().strip() for k,e in enumerate(t) : if '0' <= e <= '9' and \ e in t[:k] : print(True) break else: print(False)</pre>	<p>ຫຼືອຈະປັບປຸງມາໃຫ້ <code>for k,e in enumerate(t) :</code> ກີ່ເປັນແບບທີ່ນີ້ມາກວ່າ <code>for</code> ແກ່ງຈັດອັກຂະທີ່ໄນ້ໃຫ້ຕົວເລີນ <code>for</code> ແບບນີ້ຈະໄດ້ທີ່ສັ່ງ <code>index</code> (ເກີບໃນ <code>k</code>) ແລະ ຕົວໜັ້ນມູນ (ເກີບໃນ <code>e</code>) ມາໃຫ້ງານໃນງວນ ສິ່ງ <code>run</code>, ໄສ່ ..125.9.0., ໄດ້ຜລເປັນ <code>False</code>, ແສດງວ່າມີ່ມີ້ໜ້າ, ຖຸກຕ້ອງ ສິ່ງ <code>run</code>, ໄສ່ ..125.9.2., ໄດ້ຜລເປັນ <code>True</code>, ແສດງວ່າມີ້ໜ້າ, ຖຸກຕ້ອງ</p>

ตัวอย่างໂຄສະນາ

॥ปลงວັນທີ

ให้อ่าน ວັນເດືອນປີໃນຮູບແບບ ເດືອນ/ວັນທີ/ປີ (mm/dd/yyyy) ໂດຍຮັບຄ່າກາງແບ່ນພິມພໍ ແລະ ແປລງວັນທີທີ່ໄດ້ຮັບເປັນຮູບແບບ ວັນທີ ເດືອນ ປີ (dd MMMM yyyy) ດ້ວຍອຳນວຍທີ່ຮັບມາເປັນຕົວເລີຂຈຳນວນເຕີມ ຕ້ອງແປລັງໃຫ້ຢູ່ໃນຮູບຕ້ວຍໆການອັກສູ່ຂອງເດືອນທີ່ຮັບເຂົ້າມາ

► ຂໍ້ມູນລຳເຊົາ

ມີ 1 ບຣතັດ ເປັນຂໍ້ມູນສຕຣີງ ແທນວັນທີໃນຮູບແບບ ເດືອນ/ວັນທີ/ປີ (mm/dd/yyyy)

► ຂໍ້ມູນລຳອອກ

ມີ 1 ບຣතັດ ເປັນວັນທີໃນຮູບແບບ ວັນທີ ເດືອນ ປີ (dd MMMM yyyy)

► ຕັວຢ່າງ

Input (ຈາກແປ່ນພິມພໍ)	Output (ທາງຈອກພັບ)
08/27/2014	27 AUG 2014
02/01/2018	01 FEB 2018

บันทึกการพิมพ์ใหญ่

ให้เขียนโปรแกรมอ่านข้อมูลที่เป็นสตริงเข้ามา นำร่องทัด ประกอบด้วยอักษรอะไรก็ได้ เว้นวรรคก็ได้ จากนั้นให้บันทึก
ตัวอักษรที่เป็นตัวใหญ่เท่านั้น และแสดงผลเพื่อกำหนดทางหน้าจอ

▶ ข้อมูลนำเข้า

มี 1 บรรทัด ประกอบด้วยสตริงที่ระบุไปด้วยตัวอักษรอะไรก็ได้ เว้นวรรคก็ได้

▶ ข้อมูลส่งออก

มี 1 บรรทัด แสดงจำนวนตัวอักษรที่เป็นตัวใหญ่ทั้งหมดที่นับได้จากข้อมูลนำเข้า

▶ ตัวอย่าง

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

กลุ่มสระ

คำภาษาอังกฤษประกอบด้วยพยัญชนะ ๕ ตัว คือ (a, e, i, o, u) ขอเรียกสรรที่อยู่ติดกันว่าเป็น กลุ่มสระ เช่น **beautiful** มีกลุ่มสระ **eau**, **i**, และ **u** ซึ่งมีกลุ่มสระ 3 กลุ่ม, **vowel** มีกลุ่มสระ 2 กลุ่ม, **group** มีกลุ่มสระ 1 กลุ่ม และ **rhythm** มีกลุ่มสระ 0 กลุ่ม

ให้เขียนโปรแกรมเพื่อบ่งชี้ ว่ามีกลุ่มสระในคำภาษาอังกฤษ

▶ ข้อมูลนำเข้า

มี 1 บรรทัด แทนคำภาษาอังกฤษ ประกอบด้วยอักษรตัวพิมพ์เล็กเท่านั้น

▶ ข้อมูลส่งออก

มี 1 บรรทัด แสดงจำนวนกลุ่มสระของคำที่กำหนด

▶ ตัวอย่าง

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



วรวีร์ สัตยวินิจ (Product Manager)
กัตตราวด ชื่อสัตยาศิลป (CTO)
ยอด ชินสุกัคกุล (CEO)
ศุภฤทธิ์ กฤตยาเกียรตโน (Software Architect)
เอกลักษณ์ วิริยะโภวิทยา (COO)

(จากซ้ายไปขวา)
Intania 84
Wongnai

ไม่ ว่าคุณจะเรียนคณะอะไรหรือภาคอะไร Computer Programming ได้กลายมาเป็นความรู้ที่ทุกคนจำเป็นต้องมีไปแล้ว เพราะเทคโนโลยีเกิดใหม่ในธุรกิจและอุตสาหกรรมต่าง ๆ ไม่ว่าจะทางด้านการค้าขาย ธนาคาร การแพทย์ โฆษณา บันเทิง กีฬา เครื่องจักรและหุ่นยนต์ ฯลฯ ล้วนแล้วแต่ต้องใช้ความรู้ด้าน Computer Programming มาช่วยพัฒนาผลิตภัณฑ์หรือบริการให้ดีขึ้นกว่าเดิม โดยความต้องการนี้ยังมีมากขึ้นมากในอนาคต เมื่อผลิตภัณฑ์ บริการ หรือแม้กระทั่งเงินที่เราใช้ซื้อสินค้าต่าง ๆ ถูกแปลงไปอยู่ในรูปแบบดิจิตอล จึงเป็นเรื่องสำคัญมากที่เราจะใส่ใจกับความรู้ด้าน Computer Programming เพราะมันคือ “ทักษะที่จำเป็น” ของคนที่จะสามารถรับมือกับความเปลี่ยนแปลงของโลกดิจิตอล และพร้อมเดินทางไปอีก一步ที่จะมาพร้อมกับคลื่นยักษ์ใหญ่ในอนาคตอันใกล้นี้

05 : File

สรุปเนื้อหา

การอ่านข้อมูลเก็บเข้ามาประมวลผลทำได้หลายวิธีหลายคำสั่ง วิชานี้ใช้วิธีง่ายสุด ๆ คือ อ่านแฟ้มข้อความ (text file) ที่ลับบรรทัดๆ กันมาใส่ใน列表 หรือ ตัวแปรที่มีรูปแบบดังนี้

```
อ่านแฟ้มข้อความทีล่ะบรรทัด ใช้ readline  
infile = open(filename, 'r')  
first_line = infile.readline()  
second_line = infile.readline()  
for line in infile :  
    # คำสั่ง for อ่านจาก infile รอบละหนึ่งบรรทัด  
    # มาเก็บเป็นสตริงใน line จนหมดแฟ้ม  
    ...  
infile.close()      # ไม่อ่านแล้ว ก็ปิดแฟ้ม  
  
ถ้าบรรทัดที่อ่านเข้ามาเป็นบรรทัดว่าง ๆ เก็บใน t  
จะได้ t = '\n' หรือ len(t) เป็น 1  
แต่ถ้าอ่านตอนที่แฟ้มไม่มีข้อมูลให้อ่านแล้ว  
จะได้ t = '' หรือ len(t) เป็น 0  
  
for line in infile :  
    ...  
    เปลี่ยนกับ  
    line = infile.readline()  
while len(line) > 0 :  
    ...  
    line = infile.readline()
```

การอ่านจากแฟ้มหนึ่งบรรทัด อาจมีรหัสขึ้นบรรทัดใหม่ \n อยู่ที่ปลายบรรทัด (กรณีอ่านบรรทัดสุดท้ายของแฟ้มอาจไม่มีรหัสนี้)
ถ้าไม่ต้องการรหัส \n ก็อย่าลืมลบติ้ง แต่ถ้าอ่านมาแล้วมีการ strip()
รหัส \n จะถูกขจัดไปด้วย ไม่ต้องทำอะไรมาก

อ่านข้อมูลในแฟ้มมาหาค่าเฉลี่ย แต่ละบรรทัดเก็บรหัสกับคะแนน

```
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 = ', (sc/n))
```

อีกแบบ อ่านข้อมูลในแฟ้มมาหาค่าเฉลี่ย แต่ละบรรทัดเก็บรหัสกับคะแนน บรรทัดแรกออกจำนวนบรรทัดที่ต้องอ่าน

```
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 = ', (sc/n))
```

```
f = open('data.txt', 'r')  
t = f.readline()  
if len(t)>0 and t[-1]=='\n' :  
    t = t[:-1]  
  
for line in f :  
    if line[-1] == '\n' :  
        line = line[:-1]  
    ...
```

บันทึกข้อมูลลงแฟ้มข้อความ

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

อย่าลืม : write ไม่ต้องมีสัก 'n'
เพื่อขับบรรทัดใหม่ ที่ต้องกรณีเมื่อต้องการ

หารจำนวนเฉพาะที่มีค่าน้อยกว่า 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()
```

เรื่องผิดป oy

เมื่ออ่านข้อมูลจากแฟ้มมาหนึ่งบรรทัด
ลีบ人民币 \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')
```

ซื้อแฟ้มมีเครื่องหมาย \ แต่ใส่แค่ตัวเดียว

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

เปิดแฟ้มที่ไม่มีอยู่ในเครื่องมาอ่าน

```
infile = open('h:\\file.data')
ถ้าไม่มีแฟ้ม h:\\file.data ในเครื่อง ก็จะผิด
fn = input()
infile = open(fn)
อาจผิดได้ถ้า fn มี blank หน้าหรือหลัง จึงควร strip ก่อน
infile = open( input().strip() )
```

พิมพ์ readline เป็น readlines readlines อ่านทีเดียวหมดเพิ่ม (อ่านเสร็จแล้วได้เป็น list of strings ที่ยังไม่ได้นำเสนอ จึงขอไม่ลงในรายละเอียด)	<pre>infile = open('c:/temp/data.txt') first_line = infile.readlines() # จะว่าง second_line = infile.readline() # ควรใช้แบบบรรทัดนี้</pre>
---	--



Problem	Code								
<p><u>Input:</u> หนึ่งบรรทัดเป็นชื่อไฟล์</p> <p><u>Process:</u> อ่านข้อความในไฟล์มากลับลำดับบรรทัด</p> <p><u>Output:</u> แสดงข้อความในไฟล์แบบกลับลำดับบรรทัดออกทางจอภาพ เช่น</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding: 5px;">ข้อมูลในไฟล์</td> <td style="padding: 5px;">ผลลัพธ์ (ออกทางจอ)</td> </tr> <tr> <td style="padding: 5px;">line1</td> <td style="padding: 5px;">line3</td> </tr> <tr> <td style="padding: 5px;">line2</td> <td style="padding: 5px;">line2</td> </tr> <tr> <td style="padding: 5px;">line3</td> <td style="padding: 5px;">line1</td> </tr> </table>	ข้อมูลในไฟล์	ผลลัพธ์ (ออกทางจอ)	line1	line3	line2	line2	line3	line1	
ข้อมูลในไฟล์	ผลลัพธ์ (ออกทางจอ)								
line1	line3								
line2	line2								
line3	line1								
<p><u>Input:</u> หนึ่งบรรทัดเป็นชื่อไฟล์</p> <p><u>Process:</u> อ่านข้อความในไฟล์แล้วกลับลำดับบรรทัดแต่มีเงื่อนไขว่าจะไม่เอาบรรทัดที่ว่าง ๆ หรือมีแต่ blank</p> <p><u>Output:</u> บันทึกข้อความแบบกลับลำดับบรรทัดลงไฟล์ชื่อ reverse.txt</p>									
<p><u>Input:</u> หนึ่งบรรทัดเป็นชื่อไฟล์</p> <p><u>Process:</u> แสดงหัวข้อข่าวทั้งหมดในไฟล์ หัวข้อข่าวเป็นข้อความที่อยู่ระหว่าง <headline> กับ </headline> ในไฟล์นี้ (ทั้ง <headline> กับ </headline> อยู่ในบรรทัดเดียวกันแน่น ๆ และแต่ละบรรทัดมีไม่เกิน 1 หัวข้อข่าว)</p> <p><u>Output:</u> บรรทัดละหนึ่งหัวข้อออกทางจอภาพให้ครบทุกหัวข้อ</p>									

Problem	Code
<p><u>Input:</u> ส่องบรรทัด แต่ละบรรทัดเป็นชื่อแฟ้ม</p> <p><u>Process:</u> เปรียบเทียบว่าส่องแฟ้มนี้มีค่าตามกันนั้น หรือไม่</p> <p><u>Output:</u> ถ้าเพิ่มทั้งสองมีข้อมูลเหมือนกัน แสดง True ต่างกันก็แสดง False</p>	

ตัวอย่างการแก้โจทย์ปัญหา

ตรวจคำตอบ

จะเขียนโปรแกรมอ่านแฟ้มเก็บคำตอบแบบปรนัยของนักเรียน มาตรวจให้คะแนน

▶ ข้อมูลนำเข้า

ข้อมูลจากแฟ้ม c:\t\answers.txt รูปแบบแฟ้มเป็นดังนี้

บรรทัดแรกเก็บเฉพาะ เป็นสตริงของตัวอักษร A, B, C หรือ D

บรรทัดต่อมาจานหมดแฟ้ม แต่ละบรรทัด เก็บเลขประจำตัวนักเรียน ตามด้วยซึ่งว่างหนึ่งช่อง 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)
AABBCCCBCCCCDDABABDDCCDDCC 5630120421 AABBCCCBCCCCDDABABDDCCDDCA 5631010121 A BBCD BDBAAABA ABDBCCCDCC 563102121 ABABCCNNAADDABAB CCDDAAC 5630121821 AABBCCCBCCCCDDABABDDCCDD	5630120421,25 5631010121,14 563102121,16 5630121821,24

ตัวอย่างการเขียนโปรแกรม

โจทย์กำหนดให้อ่านข้อมูลจากแฟ้ม ละบัน พลการทำงานลงแฟ้ม เพื่อให้การเขียนโปรแกรมและหาที่ผิดได้ง่ายขึ้น จขอเขียนแบบแสดงผลออกหน้าจอค่า เมื่อทุกอย่างถูกต้อง ค่อยเปลี่ยนให้บันทึกลงแฟ้ม

โปรแกรม	คำอธิบาย
<pre> fin = open('c:\t\x07answers.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> หลังเปิดแฟ้มทันทีจะได้บรรทัดแรกของแฟ้มซึ่งคือเฉลย (<code>strip</code> เพื่อลบช่องว่าง ซ้ายขวาและรหัสขีนบรรทัดใหม่) จากนั้นใช้ <code>for</code> อ่านบรรทัดที่เหลือ ใช้ <code>split</code> แยกเลขประจำตัวกับคำตอบออกจากกัน แล้วใช้อีก <code>for</code> นำคำตอบที่ละตัวเบรียบเทียบกับเฉลย (เราใช้ <code>for</code> แบบเปลี่ยนค่า <code>k</code> เป็น index ของทั้งคำตอบกับเฉลย) ถ้าตรงกันก็เพิ่มคะแนน วนตรวจสอบทุกข้อก็แสดงผลทางจอกาฟ วนครบทุกบรรทัดก็ปิดแฟ้ม ดัง code ทางซ้าย ก่อนสั่งทำงาน ก็ต้องสร้างแฟ้ม <code>answers.txt</code> จากนั้นสั่ง run, ได้</p> <p><code>OSSError: [Errno 22] Invalid argument: 'c:\t\x07answers.txt'</code> เกิด lorsqueแปลก 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 \\ แก้ให้ถูกต้อง สั่ง run, ได้ 5630120421 25, ผิดบรรทัดที่ 4 ของโปรแกรม</p> <p><code>ValueError: too many values to unpack (expected 2)</code> แปลว่าหลัง <code>split</code> แล้ว ได้สตริงมากกว่าตัวแปรที่จะมารับผลซึ่งน่าจะเกิดหลังอ่านบรรทัดที่ 3 ของแฟ้ม</p> <p>563010121 A BBCD BDBAAABA ABDBCCCDCC บรรทัดนี้มีช่องว่างในคำตอบ (ซ่องว่างแปลว่าข้อนี้ไม่ตอบ) จึงทำให้ <code>split</code> แล้วได้สตริงมากกว่า 2 สตริง จึงต้องหาทางแยกบรรทัดให้เป็นสองสตริง เลขประจำตัว กับ คำตอบทั้งหลาย</p>

โปรแกรม

```

fin = open('c:\\t\\answers.txt', 'r')
soln = fin.readline().strip()
for line in fin:
    line = line.strip()
    j = line.find(' ')
    if j > 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()

```

คำอธิบาย

เรามีเครื่องแก้ปัญหานี้โดยคิดว่าเลขประจำตัวมี 10 หลัก ก็แยกด้วย空格 sid = line[:10] ans = line[10:] เพราะถ้าดูในไฟล์พบว่าเลขประจำตัวบางคนมีน้อยกว่า 10 หลัก จึงควรใช้วิธีหาช่องว่างแรกจากทางซ้ายด้วยบริการ find ของสตริง (แต่ต้องอย่าลืม strip ก่อน ไม่เช่นนั้น ถ้า line เริ่มด้วยช่องว่าง ก็จะผิด) เมื่อหาช่องว่างพบที่ index j ก็แยกได้ด้วย sid = line[:j] ans = line[j:] แล้วก็ตรวจสอบด้วยวิธีที่ทำมา สิ่ง run, ได้ 5630120421 12 5631010121 7 563102121 6 แล้วเกิดข้อผิดพลาดที่คำสั่ง if ans[k] == soln[k] IndexError: string index out of range ถ้าดูผลคณ์แบบที่ได้ก่อนเกิดข้อผิดพลาด จะพบว่า ได้ค่าคะแนนรวมผิดด้วย

```

fin = open('c:\\t\\answers.txt', 'r')
soln = fin.readline().strip()
for line in fin:
    line = line.strip()
    j = line.find(' ')
    if j > 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()

```

ผลที่ผิด ได้คะแนนลดลงมาก เนื่องกับว่า คำตอบผิดมีมาก ผิดปกติ ถ้าแทรกคำสั่ง print(ans) กับ print(soln) ออกมากดูเทียบกัน จะได้ (ขอตัดมาให้ครึ่งแค่ 2 บรรทัดแรก)
AABBCCCBCCCCDDABABDDCCDDCA
AABBCCCBCCCCDDABABDDCCDDCA
เห็นได้ว่าบรรทัดบน ans มันเลื่อนไปทางขวา เพราะว่ามีช่องว่างทางซ้าย แก้ปัญหานี้ด้วย ans = line[j:].strip() เพื่อตัดช่องว่างออก สิ่ง run, ได้ 5630120421 25 5631010121 14 แล้วเกิดข้อผิดพลาดที่คำสั่ง if ans[k] == soln[k] IndexError: string index out of range ทำงานผิดเหมือนครั้งที่แล้ว แต่ได้คะแนนรวมถูกต้อง ส่วนที่ทำงานผิดพลาดกลับมากเกิดกับบรรทัดที่ 4 ในไฟล์ ซึ่งมีข้อมูล 563102121 ABABCCNNAADDABAB CCDDAC

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 > 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>เนื่องจากโจทย์กำหนดว่าเลขประจำตัวกับคำตอบคั่นด้วยช่องว่าง 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> มีค่าเกินซึ่งอยู่ในสตริง จึงแก้ไขคำสั่ง <code>line[j:].strip()</code> เป็น <code>line[j+1:]</code> สั่ง run, ได้</p> <p>5630120421 25 5631010121 14 563102121 16 แล้วเกิดข้อผิดพลาดที่คำสั่ง <code>if ans[k] == soln[k]</code> <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 > 0 : sid = line[:j] ans = line[j+1:] if len(ans) < 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> หรือไม่ก็ที่ <code>soln[k]</code> คำสั่ง <code>soln[k]</code> ไม่ดำเนินการ เพราะ <code>k</code> มีค่าใน <code>range(len(soln))</code> แต่ <code>ans[k]</code> อาจผิดได้ถ้า <code>ans</code> มีขนาดน้อยกว่า <code>soln</code> ถ้าลับไปดูที่บรรทัดที่ 5 ในแฟ้มพบว่าคำตอบมีไม่ครบ นักเรียนไม่ตอบคำตอบทั้งๆ จะเป็นช่องว่าง และถูก <code>strip</code> ทิ้ง</p> <p>วิธีแก้ไข ก็แค่ตรวจสอบ ถ้า <code>len(ans) < len(soln)</code> จะเดิมช่องว่างต่อท้ายของ <code>ans</code> เป็นจำนวนเท่ากับผลต่างของความยาวทั้งสองสตริง สั่ง run, ได้</p> <p>5630120421 25 5631010121 14 563102121 16 5630121821 24 ถูกต้อง ภาระที่เหลือก็แค่เปลี่ยนจากการแสดงผลออกหน้าจอ เป็นการบันทึกลงแฟ้ม ซึ่งขอให้ผู้อ่านลองเขียนต่อเอง</p>

ตัวอย่างโจทย์ปัญหา

คะแนนเฉลี่ยของตอนเรียน

จะเขียนโปรแกรมเพื่อป้อนไฟล์ 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

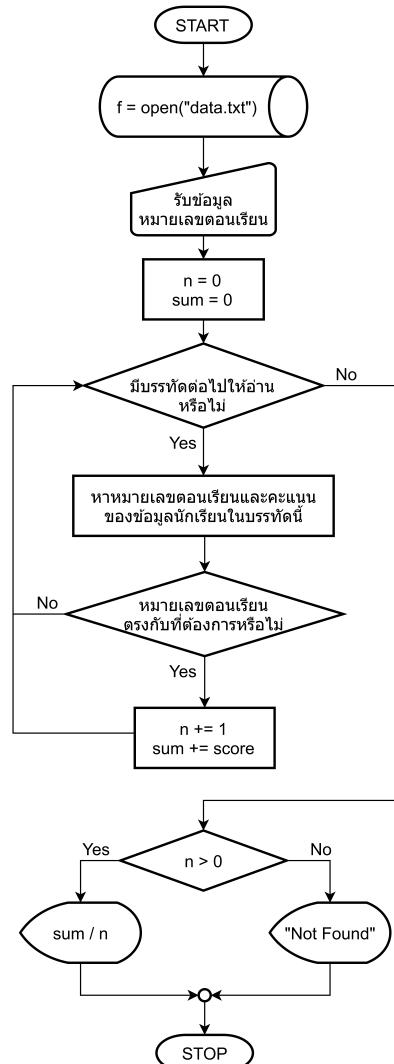
▶ ข้อมูลสองออก

แสดงคะแนนเฉลี่ยของนิสิตในตอนเรียนที่ต้องการ

▶ ตัวอย่าง

สมมติให้ไฟล์ data.txt มีข้อความข้างต้น

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



Find Student's Grade

ให้เขียนโปรแกรมเพื่ออ่านแฟ้ม score.txt ซึ่งมีรหัสนิสิตและเกรดของนิสิตแต่ละคน (0-4) ค้นด้วยช่องว่าง และรับค่ารหัสนิสิตจากแป้นพิมพ์ แล้วแสดงเกรดของนิสิตคนนั้น หากไม่พบรหัสนิสิตในแฟ้ม ให้แสดง Not Found

► ข้อมูลนำเข้า

มี 1 บรรทัด รับรหัสนิสิตเป็นจำนวนเต็ม

► ข้อมูลส่งออก

มี 1 บรรทัด แสดงเกรดของนิสิตเป็นจำนวนเต็ม หากไม่พบรหัสนิสิตในแฟ้ม ให้แสดงคำว่า Not Found

► ตัวอย่าง

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

sk сліб

จะเขียนโปรแกรมอ่านข้อมูลรหัสลับจากไฟล์名为 “file” ซึ่งระบุว่า จะต้องใช้รหัสตามลำดับอย่างไร จึงจะสามารถปล่อยอาชุดานภาพรุนแรงได้ วิธีการคือรหัสจากไฟล์นี้คือ ตัวอักษรที่บันทึกอยู่เป็นตัวอักษรสามตัว จากนั้นนับว่า ตัวอักษรทั้งสามตัวนั้น ปรากฏเป็นจำนวนเท่าใดในแฟ้ม และจึงเรียงลำดับของอักษรทั้งสามตามจำนวนครั้งที่ปรากฏในแฟ้มจากมากที่สุดไปน้อยที่สุด โดยถือว่า ตัวอักษรตัวพิมพ์ใหญ่กับตัวพิมพ์เล็ก เป็นตัวเดียวกัน (case sensitive) และจำนวนตัวอักษรทั้งสามตัวนั้นจะไม่เท่ากัน ตัวอย่างเช่น หากมีแฟ้ม data.txt เป็นดังนี้

```
agAbggggDf  
ffgFFFaaD  
DaADDFF  
FFDDFF
```

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

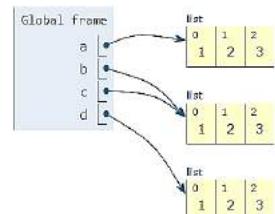
06.1 : List

สรุปเนื้อหา

รายการหรือลิสต์ (list) เป็นที่เก็บค่าของข้อมูลที่มีลำดับ แต่ละตัวมีเลข index กำกับลำดับ ตัวซ้ายสุดมี index 0

การสร้าง list

- `x = []` หรือ `x = list()` ได้ลิสต์ว่าง ๆ มี `len(x)` เป็น 0
- `x = [0]*10` ได้ `x = [0,0,0,0,0,0,0,0,0,0]`
- `x = list(a)` ได้ลิสต์ที่มีข้อมูลตามที่หยิบออกมานาจาก 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]`
- `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 คำสั่งนี้ไม่เปลี่ยนลิสต์มา

`sorted(x)` คืนลิสต์ที่มีค่าเหมือนกับที่แจงออกจาก 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 จึงต้องตรวจก่อน

```
if e in x :  
    k = x.index(e)  
    ...  
else :  
    ...
```

หมายเหตุ : ลิสต์ไม่มี `find` เมื่อมองกับของสตริง

บริการ `split` กับ `join` ไม่ใช่บริการของลิสต์ แต่เป็นของสตริง ที่เกี่ยวข้องกับลิสต์

- `t.split()` คืนลิสต์ของสตริงย่อยที่แยกออกจากสตริง t โดยใช้ช่องว่างเป็นตัวคั่นสตริงย่อย
- `t.split(s)` คืนลิสต์ของสตริงย่อยที่แยกออกจากสตริง t โดยใช้สตริง s เป็นตัวคั่นสตริงย่อย
- `s.join(x)` คืนสตริงที่ได้จากการนำสตริงในลิสต์ x มาต่อกันคันด้วยสตริง s

ตัวอย่างการเข้าใช้ข้อมูลและลิสต์ย่อยในลิสต์ (สมมติให้ $x = [1, 2, 12, 13, 14, 15]$)

- มีวิธีการเข้าใช้และการเลือกลิสต์ย่อยด้วย $len(x)$ เหนื่องบกับวิธีของสตริง
 - ใช้ $for e \text{ in } x :$ ในการแยกและนำข้อมูลในลิสต์ x จากช้ายไปขวาออกมาเก็บใน e เพื่อนำไปใช้ในวงวน for ได้คำเตือน: ไม่ควรเปลี่ยนแปลงลิสต์ x ในวนที่กำลังแยกแจงข้อมูลในลิสต์ออกมานำใช้งาน อาจมีพฤติกรรมที่ไม่ตรงกับที่คาดไว้
 - $x[0]$ เหมือน $x[-len(x)]$ ได้ 1
 - $x[-1]$ เหมือน $x[-len(x)-1]$ ได้ 15
 - อย่าลืมว่า $index$ ของลิสต์ x ต้องอยู่ในช่วง
 - 0 ถึง $len(x)-1$ จากช้ายไปขวา
 - 1 ถึง $-len(x)$ ถอยจากขวามาซ้าย
- ดังนั้นเราเขียน $x[k]$ ได้โดยที่ $-len(x) \leq k \leq (len(x)-1)$ เพราะฉะนั้น $x[-6]$ กับ $x[5]$ ผิด
- $x[a:b]$ (slice ของลิสต์) ได้ลิสต์เสมอ
 - เขียน $x[a:b]$ ค่า a กับ b เป็นอะไรมีได้ไม่ผิด
 - $x[2:50000]$ 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 (เปลี่ยนอักษรภาษาไทยในสตริงไม่ได้ แต่เปลี่ยนข้อมูลภาษาในลิสต์ได้)

- $x[k] = e$ 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 เปลี่ยนไปอ้างอิงลิสต์ใหม่ที่ถูกสร้างขึ้นเมื่อตอนกับของ y เหมือนคำสั่ง $x = list(y)$
 - $x[:] = y$ x ยังอ้างอิงลิสต์ตัวเดิม แต่ข้อมูลในลิสต์ x เปลี่ยนไปเมื่อตอนกับข้อมูลของ y (x กับ y เป็นคนละลิสต์)
 - $x[:] = y[:]$ ได้ผลเหมือน $x[:] = y$

รูปแบบการประมวลผลสตรีก์เพบปอย

ใช้ลิสต์เก็บข้อมูลเพื่อนำมาใช้ภายหลัง	อ่านข้อมูลเข้ามา n ตัว <pre>n = int(input()) data = [] for k in range(n): data.append(float(input()))</pre>
ต้องการหยิบข้อมูลในลิสต์จากซ้ายไปขวาตาม ประมวลผลที่ <u>ตั้ง</u> ใช้ <code>for e in x</code>	ต้องการนับว่าลิสต์ x มีข้อมูลเท่ากับ e กี่ตัว <pre>c = 0 for d in x : if d == e : c += 1 หรือแบบสั้น ๆ c = x.count(e)</pre>
ต้องการหยิบข้อมูลของลิสต์ในช่วงที่สนใจมา ประมวลผล ใช้ <code>for e in x[a:b:c]</code>	ต้องการหาระยะของคะแนนที่เก็บในลิสต์ x โดยขอไม่รวมคะแนน ที่น้อยสุดและมากสุด (ตัดออกอย่างละหนึ่งตัว) <pre>s = 0 for d in sorted(x)[1:-1] : s += d หรือแบบสั้น ๆ s = sum(sorted(x)[1:-1])</pre>
ต้องการปรับเปลี่ยนค่าในลิสต์ ใช้ <code>for i in range(len(x)) :</code> <code> x[i] = ...</code>	x เป็นลิสต์เก็บคะแนน ต้องการปรับซ่องที่มีค่าน้อยกว่า 30 ให้มีค่าเพิ่มอีก 10% <pre>for i in range(len(x)) : if x[i] < 30 : x[i] += 0.1*x[i] เจียนแบบข้างล่างนี้ไม่ได้ เพราะ e ที่จะออกมาเป็นที่เก็บคงจะที่กับที่อยู่ ในลิสต์ for e in x : if e < 30 : e += 0.1*e</pre>
ต้องการหยิบข้อมูลแต่ละตัว พร้อมกับ index ของตัวนั้น ๆ	<code>for i in range(len(t)) :</code> <code> c = t[i]</code> <code> print(i,c)</code> หรือ <code>for i,c in enumerate(t):</code> <code> print(i,c)</code>
ต้องการหยิบข้อมูลในลิสต์จากขวามาซ้าย ทีละตัว	<code>for e in x[::-1] :</code> <code> ...</code> หรือ <code>for k in range(-1,-(len(x)+1),-1) :</code> <code> # อ่านจาก ໄວກສິດສູງ</code> <code> e = x[k]</code> <code> ...</code>

ต้องการหยิบข้อมูลในลิสต์มาคำนวณ จนกว่าเงื่อนไขหนึ่งจะเป็นจริง	<pre>for e in x : # เงื่อนไขที่ต้องการ : ... break ... else : ... # มาทำที่นี่ ถ้าไม่พบเงื่อนไขที่ต้องการเลย</pre>
ต้องการหยิบข้อมูลในลิสต์ตามลำดับที่เรียง จากข้อมูลน้อยสุดไปข้อมูลมากสุดในลิสต์	<pre>for e in sorted(x) : ...</pre>
ต้องการหยิบข้อมูลในลิสต์ตามลำดับที่เรียง จากข้อมูลมากสุดไปข้อมูลน้อยสุดในลิสต์	<pre>for e in sorted(x)[::-1] : ...</pre>
ต้องการหยิบข้อมูลในลิสต์จากซ้ายไปขวา ประมวลผลทีละคู่ข้อมูลที่ติดกัน	<p>ต้องการตรวจสอบว่า ข้อมูลในลิสต์ s เรียงลำดับจากน้อยไปมากหรือไม่</p> <pre>for k in range(len(s)-1) : if s[k] > s[k+1] : print("False") break else: print("True")</pre>
ต้องการหยิบข้อมูลทุก ๆ คู่ในลิสต์ (ไม่จำเป็นต้องติดกัน) มาประมวลผล	<p>ต้องการนับว่า มีข้อมูลคู่ในลิสต์ที่ตัวทางซ้ายมีค่ามากกว่าตัวทางขวา (ไม่จำเป็นต้องอยู่ติดกัน)</p> <pre>c = 0 for i in range(len(x)) : for j in range(i+1,len(x)) : if x[i] > x[j] : c += 1 print(c)</pre>
เรียงลำดับข้อมูลในลิสต์ (ด้วยวิธี bubble sort)	<p># ต้องการเรียงลำดับข้อมูลในลิสต์ d จากน้อยไปมาก</p> <pre>for k in range(len(d)-1) : for i in range(len(d)-1) : if d[i] > d[i+1] : d[i],d[i+1] = d[i+1],d[i]</pre>

เรื่องผิดบอย

ใช้ index ที่เกินช่วงที่ใช้ได้ของลิสต์ อย่าลืมว่า index ของลิสต์ x อยู่ในช่วง 0 ถึง <code>len(x)-1</code> จากซ้ายไปขวา และ -1 ถึง <code>-len(x)</code> 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>

<p>เมื่อ x เป็นลิสต์ อย่าสับสนระหว่าง $x.append(y)$ กับ $x+=y$ ซึ่งได้ผล ไม่เหมือนกัน</p>	<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>
<p>ต้องการลิสต์ y ที่มีค่าเหมือนกับลิสต์ x ตัวเดียวกัน เลือกว่าจะใช้ $y = x$ หรือ $y = list(x)$ ปกติ เมื่อยกน้ำหน้าจะใช้ $y = x$</p>	<pre>x = [1,2,3] y = x # y เป็นลิสต์เดียวกับ x x[2] = 0 # y[2] ก็เปลี่ยนเป็น 0 ด้วย</pre>
<p>เขียน $x = list(str)$ เป็นการสร้างลิสต์ ที่ประกอบด้วย 'ตัวอักษร' ใน str</p>	<pre>x = list('abc') ได้ x = ['a','b','c'] ไม่ใช่ x = ['abc']</pre>
<p>นำข้อมูลในลิสต์มา $join$ กันให้เป็นสตริง แต่ถ้าไม่ไปว่าข้อมูลในลิสต์นั้นต้องเป็นสตริง ถึงจะ $join$ ได้</p>	<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>
<p>สับสนคำสั่ง $x.sort()$ กับ $sorted(x)$ $x.sort()$ เรียงลำดับข้อมูลในลิสต์ x คำสั่งนี้ไม่คืนผลใด ๆ ในขณะที่ $sorted(x)$ นำข้อมูลที่ได้จาก x มาเรียง ลำดับแล้วคืนลิสต์ใหม่ที่มีข้อมูลเหมือนใน x แต่เรียงลำดับแล้ว โดยที่ x ไม่เปลี่ยนแปลง</p>	<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>
<p>มีการลบหรือเพิ่มข้อมูลในลิสต์ระหว่างที่ มีการแยกแจงข้อมูลในลิสต์</p>	<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) จะไม่ได้ผลตามที่คาด และเกิดข้อผิดพลาดระหว่างการทำงานด้วย (ลอง run ดู) ควรใช้การสร้างลิสต์ชั่วคราวก่อน แล้วค่อยนำกลับไปใส่ในลิสต์เดิม t = [] for e in x : if e%2 != 1 : t.append(e) x[:] = t ให้สังเกตว่าคำสั่งสุดท้ายคือ x[:] = t แทนที่จะเป็น x = t เพราะต้องการ เปลี่ยนข้อมูลในลิสต์ x ให้เหมือน t ไม่ได้ต้องการให้ x ไปอ้างอิงลิสต์เดียวกับ t</pre>

แบบฝึกหัด

Problem	Code
<p><u>Input:</u> มี 2 บรรทัด แต่ละ บรรทัดเป็นสมาชิกของเวกเตอร์ ซึ่งเป็นจำนวนจริงที่อยู่ในรูปแบบดังนี้ คำย่อว่า อ่านทั้งสองบรรทัดเก็บในลิสต์ v1 และ v2</p> <p><u>Process:</u> คำนวณ dot product ของเวกเตอร์ v1 กับ v2</p> <p><u>Output:</u> แสดงคำว่า Error ถ้า v1 และ v2 มีขนาดไม่เท่ากัน แต่ถ้าเท่ากัน แสดงค่า dot product ที่คำนวณได้</p>	
<p><u>Input:</u> บรรทัดแรกเป็นจำนวนเต็ม n และมีอีก n บรรทัด แต่ละบรรทัดเป็นจำนวนเต็ม 1 จำนวน</p> <p><u>Process:</u> เรียงลำดับจำนวนเต็มทั้ง n ตัวจากน้อยไปมาก</p> <p><u>Output:</u> จำนวนเต็มทั้ง n ที่เรียงจากน้อยไปมากบนบรรทัดเดียว กันเรียงจากซ้ายขวา คั่นด้วยจุลภาค ,</p>	
<p><u>Input:</u> หนึ่งบรรทัดเป็นชื่อแฟ้ม แฟ้มนี้เก็บจำนวนเต็ม บรรทัดละจำนวน</p> <p><u>Process:</u> หาว่าจำนวนเต็มใดในแฟ้มประภูมิซักกันมากสุด ซึ่งมีซ้ำกันมากสุดหลายตัว ให้หาทุกตัว</p> <p><u>Output:</u> ข้อมูลทุกตัวที่ประภูมิซักกันมากสุดในแฟ้ม เรียงตามลำดับที่ประภูมิในแฟ้ม</p>	

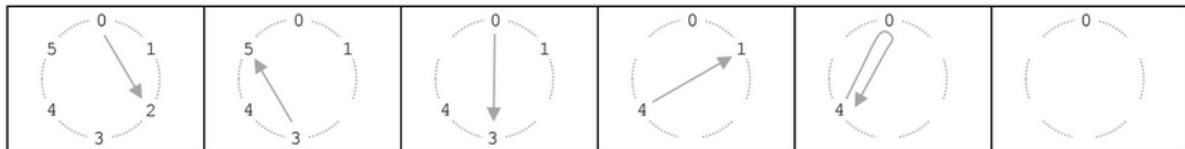
Problem	Code
<p><u>Input:</u> หนึ่งบรรทัดเป็นชื่อแฟ้ม</p> <p><u>Process:</u> แสดงหัวข้อข่าวทั้งหมดในไฟล์ฟิล์ม หัวข้อของ瓜 เป็นข้อความที่อยู่ระหว่าง <headline> และ </headline> ในแฟ้มนี้ (ทั้ง <headline> กับ </headline> อยู่ในบรรทัดเดียวกันแน่นอน และแต่ละบรรทัดมีไม่เกิน 1 หัวข้อข้างต้น)</p> <p><u>Output:</u> หนึ่งบรรทัด ขึ้นทั้งหัวข้อ ให้ครบถ้วนหัวข้อโดยแสดงเรียง หัวข้อตามตัวอักษรจากน้อยไปมาก</p>	
<p><u>Input:</u> หนึ่งบรรทัดเป็นชื่อแฟ้ม แฟ้มนี้เก็บชื่อบรรทัดละหนึ่งชื่อ</p> <p><u>Process:</u> เรียงลำดับชื่อที่อ่านจากแฟ้ม โดยเรียงลำดับตามความยาวของชื่อจากน้อยไปมาก ถ้ามีความยาวเท่ากันให้เรียงตามตัวอักษรแบบในพจนานุกรม</p> <p><u>Output:</u> ลำดับของชื่อตามที่เรียงได้</p>	

ตัวอย่างการแก้โจทย์ปัญหา

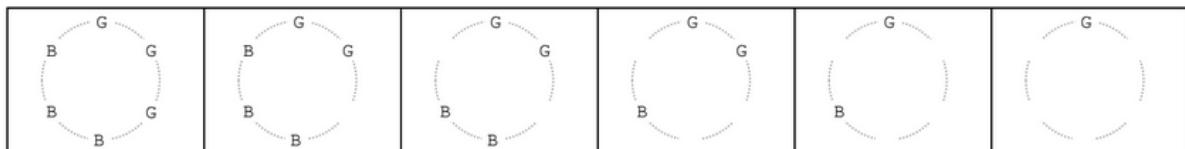
ขั้นตอนการ解

กำหนดให้ r เป็นจำนวนเต็ม หากเรานำคนดี n คน กับคนเลว m คน มาบินเรียงเป็นวงกลม โดยให้คนดียืนเรียงติดกัน ตั้งแต่ตำแหน่งที่ 0 ถึง $n-1$ และคนเลวยืนเรียงต่อไปตั้งแต่ตำแหน่ง n ถึง $2n-1$ จงหาจำนวนเต็มบวก d ที่มีค่าน้อยสุด ที่เมื่อนำคนทั้งหมดมาใช้กับปัญหา Josephus และจะเหลือคนดีเป็นคนสุดท้าย

ปัญหา Josephus เป็นดังนี้ : ข้อมูลนำเข้าคือจำนวนเต็ม m กับ d ให้มีคน m คนยืนเรียงเป็นวงกลม เริ่มนับที่ 0 นับไปอีก d คน ก็ให้คนนั้นออกจากรอบ แล้วก็เริ่มจากคนถัดไปนับไปอีก d คน ก็ให้คนนั้นออก ทำเช่นนี้ไปเรื่อยๆ จนเหลือคนสุดท้าย คนนั้นเป็นผู้ชนะ เช่น ให้ $m = 6$ และ $d = 2$ การเปลี่ยนแปลงของคนที่ยืนในวงกลมแสดงได้ดังรูปข้างล่างนี้ หมายเหตุ 0 เป็นผู้ชนะ



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



▶ ข้อมูลนำเข้า

จำนวนเต็มบวก 1 จำนวน แทนค่า n ข้างต้น

▶ ข้อมูลส่งออก

ค่า d ที่เป็นจำนวนบวกน้อยสุดที่ทำให้ขัดคนออกแล้วเหลือคนสุดท้ายเป็นคนดี 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)==1 and q[0]=='G': print(d) break else: print('Not found') </pre>	<p>เนื่องจากคอมพิวเตอร์ทำงานได้รวดเร็ว ขอแก้ปัญหานี้ด้วยการจำลอง การค่อย ๆ จัดคนในวงกลมออกตามกฎเกณฑ์ที่กำหนดไว้ โดยใช้ลิสต์เก็บ คนที่ยืนในวงกลม และค่อย ๆ ลบข้อมูลในลิสต์ออก</p> <p>โปรแกรมทางข้างนี้รับค่า n จากนั้นคุยกทดสอบการจัดคนด้วยค่า d ต่าง ๆ เริ่มที่ 1 ไปถึง $2n-1$ ภายในแต่ละรอบของ for จะสร้างลิสต์ของ ตัวอักษร G จำนวน n ตัวตามด้วยตัวอักษร B อีก n ตัว ('G' แทนคนดี, 'B' แทนคนเลว) และก็เริ่มขั้นตอนการแก้ปัญหา Josephus ด้วย m = 2*n และ d ตามค่าของ for หากผลการจัดคนใน วงกลมของปัญหา Josephus เหลือคนสุดท้ายเป็นคนดี ก็แสดงค่า d และ break ออกจาก for ได้เลย ถ้าไม่ได้คนดีเป็นคนสุดท้าย ก็วนกลับไปเพิ่ม ค่า d เพื่อทำรอบต่อไปของ for</p>
<pre> m = int(input()) d = int(input()) q = list(range(m)) k = 0 while len(q) > 1 : k += d if k >= len(q) : k -= len(q) q.pop(k) print(q[0]) </pre>	<p>ขอเก็บโปรแกรมข้างบนไว้ก่อน และมาเขียนโปรแกรมสำหรับปัญหา Josephus ทางข้างนี้ ที่รับค่าจำนวนเต็ม m กับ d และสร้างลิสต์ที่เก็บ หมายเลข 0 ถึง m-1 ด้วยคำสั่ง list(range(m)) จากนั้นเข้าสู่ vòng while ที่จะวนทำงานตราบเท่าที่ลิสต์ q ยังมีข้อมูลมากกว่าหนึ่งตัว มีตัวแปร k เก็บ index ของลิสต์มีค่าเริ่มที่ 0 ในแต่ละรอบจะลบข้อมูล ตัวที่ถัดจาก k ไปอีก d ตัว ซึ่งคือเพิ่มค่า k อีก d ถ้าค่า k เกินหรือเท่ากับ ขนาดของลิสต์ ก็ต้องวนกลับมาด้านข้างของลิสต์ ซึ่งก็คือการลดค่าของ k ด้วยขนาดของลิสต์ เช่น q มีข้อมูล 7 ตัว, k = 5, d = 3 คำสั่ง k += d ทำให้ k = 8 เกินขนาดของ q คำสั่ง k -= len(q) ทำให้ k กลับมามีค่าเป็น 1 สรุปคือ ถ้าหากตำแหน่งที่ 5 ไปอีก 3 ตำแหน่งคือ 1 (5 -> 6 -> 0 -> 1)</p> <p>เมื่อได้ตำแหน่ง k ที่เราต้องลบข้อมูลออก ก็ทำการคำสั่ง q.pop(k) วงวน while จะลบข้อมูลรอบตัวๆ เมื่อเหลือข้อมูลตัวเดียว ข้อมูลที่เหลือรอด นั้นก็คือ q[0]</p> <p>สั่ง run, ใส่ m เป็น 6, d เป็น 2, ได้ผลเป็น 0 ถูกต้อง สั่ง run, ใส่ m เป็น 4, d เป็น 3, ทำงานผิด IndexError: pop index out of range ที่คำสั่ง q.pop(k) แปลว่า k เก็บ index ที่มีค่าเกินช่วงที่ลิสต์ q มีให้ลบ ถ้าลองแทรกรคำสั่ง print(k, len(q)) จะได้ 3 3 นั่นคือลิสต์ที่มี 3 ตัว ลบตัวที่ index 3 ไม่ได้ เพราะมีให้ใช้แค่ index 0 ถึง 2 เท่านั้น</p>
<pre> m = int(input()) d = int(input()) q = list(range(m)) k = 0 while len(q) > 1 : k = (k + d) % len(q) q.pop(k) print(q[0]) </pre>	<p>แก้ปัญหาข้างต้นได้ด้วยการใช้คำสั่ง $k = (k + d) \% \text{len}(q)$ เพื่อคำนวณ index ที่ถัดจาก k ไปอีก d ซอง กรณีที่ index เลยไปทางขวาของลิสต์ การ mod ด้วยขนาดของ ลิสต์ จะได้ผลลัพธ์มาทางซ้าย เช่น ถ้าลิสต์มีขนาด 4 ตัว ถ้าหากตำแหน่ง 3 ไปอีก 5 ตำแหน่งก็คือ $(3+5) \% 4 = 0$</p> <p>สั่ง run, ใส่ m เป็น 6, d เป็น 2, ได้ผลเป็น 0 ถูกต้อง สั่ง run, ใส่ m เป็น 4, d เป็น 3, ได้ผลเป็น 1 ลองจำลองการทำงานได้ <u>0,1,2,3 -> 0,1,2 -> 1,2 -> 1</u> ถูกต้อง</p>

โปรแกรม

```
n = int(input())
for d in range(1,2*n) :
    q = ['G']*n + ['B']*n
    k = 0
    while len(q) > 1 :
        k = (k + d) % len(q)
        q.pop(k)
    if q[0] == 'G' :
        print(d)
        break
    else:
        print('Not Found')
```

คำอธิบาย

น าดคำสั่งในการหาคำตอบของปัญหา Josephus ข้างต้นไปแรกใน
โปรแกรมตอนแรกที่เขียน ได้โปรแกรมทางซ้ายนี้
สั่ง run, ใส่ 1, ได้ 1, ถูกต้อง
สั่ง run, ใส่ 3, ได้ 2, ถูกต้อง
สั่ง run, ใส่ 6, ได้ 3, ถูกต้อง
สั่ง run, ใส่ 13, ได้ 4, ถูกต้อง
สั่ง run, ใส่ 14, ได้ 6, ถูกต้อง

06.2 : Nested List

สรุปเนื้อหา

ลิสต์เก็บอะไรได้ จ. วน. ตีม จำนวนจริง สตริง หรือ อื่น ๆ หรือแม้กระทั่งลิสต์ ก็ได้ ให้ x เป็นลิสต์ และถ้า $x[k]$ ก็เป็นลิสต์ จะได้ $x[k][i]$ ค่าข้อมูลที่ 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
```

รูปแบบการประมวลผลลิสต์ซ้อนลิสต์กีบพับปวย

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]]
```

โปรแกรมข้างล่างนี้อ่านข้อมูลวงกลมจากไฟล์มาเก็บในลิสต์

```
circles = []
file1 = open('c:/temp/circles.txt')
for line in file1 :
    x,y,r = line.split()
    circles.append( [float(x),float(y),float(r)] )
# ข้างล่างนี้ทางกลมที่ไม่ทับหรือแตะวงกลมอื่นเลย
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>ใช้ลิสต์ช้อนลิสต์ สร้างเมทริกซ์ ถ้าต้องการเมทริกซ์ขนาด $m \times n$ ก็ สร้างลิสต์ A ที่ <code>len(A)</code> มีค่าเท่ากับ m (จำนวนแถว) และ $\text{len}(A[0]) = \text{len}(A[1])$ $= \dots = \text{len}(A[m-1]) =$ (จำนวนคอลัมน์) เช่นสร้างเมทริกซ์ ที่มีค่า 0 หมดขนาด 5×3 <code>A = []</code> <code>for k in range(5):</code> <code> A.append([0]*3)</code> </p>	<p>ตัวอย่างคำภาษาบ้าๆของเมทริกซ์ A กับ B</p> $A = [[1, 2], [1, 1]]$ $B = [[0, 1], [4, 3]]$ $C = []$ <pre> 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>ใช้ลิสต์ช้อนลิสต์ โดยที่ลิสต์ข้างในเก็บ ข้อมูลที่ไม่จำเป็นต้องมีขนาดเท่ากัน</p>	<p>ใช้ลิสต์ช้อนลิสต์ เก็บข้อมูลที่ประกอบด้วย <code>username</code> (ที่ช่อง 0), และลิสต์ของ <code>usernames</code> รายอื่นที่ขอติดตามข่าวสาร (ที่ช่อง 1 โดยลิสต์นี้ไม่จำเป็นต้องมี ขนาดคงตัว) (ตัวอย่างนี้ช้อนตั้งสามชั้น) เช่น</p> $f = [['noon', 'pat', 'koi']], ['wii', ['noon', 'koi']],$ $['pat', ['koi', 'noon', 'wii']], ['koi', []]]$ <p>ต้องการหาว่า <code>username</code> ใดมีคนติดตามเป็นจำนวนมากสุด</p> <pre> maxindex = 0 for k in range(1, len(f)) : if len(f[k][1]) > 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) > len(max_followers) : max_username = username max_followers = followers print(max_username, 'has max.# of followers :', ', '.join(max_followers)) </pre>

ใช้ลิสต์ซ้อนลิสต์เป็นที่เก็บข้อมูล
ชั่วคราวเพื่อนำไป sort ตาม
ข้อกำหนดที่ต้องการ โดยสร้างลิสต์
ข้างในให้มีสมาชิกตัวแรกเป็นเกณฑ์ที่
ใช้ในการ sort

หมายเหตุ : การ sort ลิสต์ที่มีลิสต์
จะเปรียบเทียบความ อย่างว่า ของ
ข้อมูลของลิสต์นั้นในที่ตัวแรกซ้าย
ไปขวา เช่น
 $x = [[3,2],[3,1],[9],[2,5],[3]]$
 $x.sort()$ จะได้ x เปลี่ยนเป็น
 $[[2,5],[3],[3,1],[3,2],[9]]$
นั่นคือ
 $[2,5] < [3] < [3,1] < \dots$

จากน้ำด้วย จ่อ ที่นี่ เมื่อมี f แล้ว ถ้าต้องการเรียงลำดับ usernames
ตามจำนวนผู้ติดตาม ก็ใช้
 $c = []$
 $for [username,followers] in f :$
 $c.append([len(followers),username])$
 $c.sort()$
 $for [x,username] in c :$
 $print(username)$

จากตัวอย่างบน

$f = [['noon','[pat','koi']], ['wii','[noon','koi']],$
 $['pat','[koi','noon','wii']], ['koi',[],[]]$

จะได้ $c = [[2,'noon'],[2,'wii'],[3,'pat'],[0,'koi']]$

$c.sort()$ ได้ $[[0,'koi'],[2,'noon'],[2,'wii'],[3,'pat']]$

หากต้องการให้ sort ด้วยเกณฑ์ที่ซับซ้อนขึ้น เช่น ให้เรียงตามจำนวนผู้ติดตาม
จากน้อยไปมาก และในกรณีที่จำนวนผู้ติดตามเท่ากัน ให้เรียงตาม username
จากน้อยไปมาก เช่นกัน แบบที่ต้องการนั่งสองข้อมูลอยู่ในลิสต์เรียงแบบน้อยไป
มาก สามารถใช้ sort() ได้เลย แต่ถ้าเปลี่ยนเป็น ให้เรียงตามจำนวนผู้ติดตาม
จากมากไปน้อย สำหรับกรณีที่จำนวนผู้ติดตามเท่ากัน ให้เรียงตาม username
จากน้อยไปมาก ก็อาจใช้กลวิธีเล็กน้อย เช่น ติดลบจำนวนที่ต้องการเรียงจากมาก
ไปน้อย การเรียงเลขลบจากน้อยไปมาก ก็คือเรียงเลขบวกจากมากไปน้อย

$c = []$
 $for [username,followers] in f :$
 $c.append([-len(followers),username])$
 $c.sort()$

จากตัวอย่างบน

$f = [['noon','[pat','koi']], ['wii','[noon','koi']],$
 $['pat','[koi','noon','wii']], ['koi',[],[]]$

จะได้ $c = [[-2,'noon'],[-2,'wii'],[-3,'pat'],[0,'koi']]$

sort ได้ $[[3,'pat'],[-2,'noon'],[-2,'wii'],[0,'koi']]$

เรื่องผิดบอຍ

การเพิ่มลิสต์ย่อย a เข้าไปในลิสต์ใหญ่ x
ต้องใช้ $x.append(a)$ ไม่ใช่

$x = x + a$ หรือ $x += a$

$x = [[2,3],[3,4]]$
'ได้มาจากการ append [2,3] กับ [3,4] เข้า x
 $x = []$
 $x.append([2,3])$
 $x.append([3,4])$
แต่เมื่อตอนคำสั่งข้างล่างนี้
 $x = []$
 $x = x + [2,3] + [3,4] # ได้ [2,3,3,4]$

อย่าสร้างลิสต์ซ้อนลิสต์ด้วย *
อย่าเขียน $[a]*n$ เมื่อ a เป็นลิสต์

ต้องทราบสิ่ง หมายความว่าที่แทนด้วยลิสต์ซ้อนลิสต์ ที่มีขนาด 3×3 มีค่าเป็น 0 หมด
 $x = [[0,0,0], [0,0,0], [0,0,0]]$
 $\text{print}(x)$ จะได้ $[[0,0,0], [0,0,0], [0,0,0]]$
 แต่ลิสต์ข้างในทั้งสามตัวนี้คือลิสต์ตัวเดียวกัน ถ้าสั่ง $x[0][1] = 1$ ทำงาน
 x จะเปลี่ยนเป็น $[[0,1,0], [0,1,0], [0,1,0]]$
 ใช้ $x = [[0]*3]*3$ ก็ได้ผลที่แปลกแบบข้างบนเหมือนกัน
 เราต้องสร้างลิสต์ข้างในให้เป็นคนละตัว โดยเขียน
 $x = [[0,0,0], [0,0,0], [0,0,0]]$
 หรือใช้วงวนสร้าง
 $n = 3$
 $x = []$
 $for i in range(n) :$
 $x.append([0]*n)$
 สรุปคือ เมื่อได้เขียน $[a]*n$ ต้องระวัง ถ้า a เป็นลิสต์ ให้ใช้วงวนสร้างแทน

แบบฝึกหัด

Problem	Code
<p><u>Input:</u> บรรทัดแรกมีจำนวนเต็ม r กับ c ตามด้วยอีก r บรรทัด แต่ละบรรทัดมีจำนวนเต็ม c ตัว</p> <p><u>Process:</u> สร้างเมทริกซ์ด้วยลิสต์ซ้อนลิสต์ ถ้ามีบรรทัดที่ มีข้อมูลไม่ใช่ c ตัว ให้แสดงผลเป็นลิสต์ซ้อนลิสต์ว่าง $[]$</p> <p><u>Output:</u> เมทริกซ์ลิสต์ซ้อนลิสต์ที่สร้างได้</p>	
<p><u>Input:</u> หนึ่งบรรทัดเป็นชื่อแฟ้ม แฟ้มนี้มีหลายบรรทัดเท่ากับ จำนวน <code>username</code> แต่ละบรรทัดประกอบด้วย <code>username</code> ตามด้วย <code>usernames</code> อีก ๆ ที่คิดตามข่าวสารของ <code>username</code> แรกต้นบรรทัด เช่น</p> <pre>noon pat koi wii noon pat koi noon wii koi</pre> <p><u>Process:</u> สร้างลิสต์ซ้อนลิสต์ที่มีรูปแบบตามตัวอย่างนี้</p> <pre>f = [['noon', 'pat', 'koi'], ['wii', 'noon'], ['pat', 'koi', 'noon', 'wii'], ['koi']]</pre> <p><u>Output:</u> ลิสต์ที่สร้างได้</p>	

Problem	Code
<u>Input:</u> จาก f ที่ได้มาในข้อที่แล้ว <u>Process:</u> ต้องการรู้ว่า โครงสร้างที่ไม่ใช่โครงติดห่วงเลย <u>Output:</u> รายชื่อของผู้ที่ไม่ใช่โครงติดห่วงเลย	
<u>Input:</u> บรรทัดแรกมี จำนวนหนึ่ง n และอีก m บรรทัดที่ ตามมาเป็นสตริง <u>Process:</u> เรียงลำดับสตริงที่อ่านเข้ามา จากน้อยไปมาก ตามความยาวแล้ว จึง วิ่ง สตริงยาวเท่ากันให้เรียงตาม ตัวสตริงเอง เช่น 'xyz', 'xy', 'abc' เรียงแล้วได้ 'xy', 'abc', 'xyz' <u>Output:</u> สตริงที่เรียงแล้ว บรรทัดละสตริง	

ตัวอย่างการแก้โจทย์ปัญหา

เรียงตามค่าแบบรวม

จะเขียนโปรแกรมรับรหัสนักเรียนและรายการของคะแนนการสอบย่อยต่าง ๆ ของนักเรียนจำนวนหนึ่ง มาประมวลผลเพื่อแสดงรหัสนักเรียนและคะแนนรวมตามลำดับคะแนนรวมจากมากไปน้อย

▶ ข้อมูลนำเข้า

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

▶ ข้อมูลส่งออก

ถ้ามีบรรทัดที่จำนวนคะแนนย่อยไม่ตรงกับ m ให้รวมรหัสนักเรียนมาแสดงตามตัวอย่าง (เรียงตามที่ได้รับจากข้อมูลนำเข้า)

ถ้าทุกบรรทัดมีคะแนนครบจำนวนทุกคน ให้แสดงรหัสนักเรียนตามด้วยคะแนนรวมของนักเรียนบรรทัดละคน เรียงลำดับตามคะแนนรวมจากมากไปน้อย ในกรณีที่มีคะแนนรวมเท่ากัน 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 ลิสต์ข้างในจะเป็นลิสต์เก็บรหัสนักเรียนที่ซ่อง 0 ตามด้วยคะแนนตั้งแต่ซ่องที่ 1 ถึง m
 [[รหัสนักเรียน, คะแนน, ..., จะเบ่ง], [รหัสนักเรียน, คะแนน, ..., คะแนน], ...]]

ขอแบ่งการประมวลผลเป็นขั้นตอนทีละขั้น ดังนี้

- อ่านข้อมูลนำเข้าเก็บเป็นลิสต์ซ้อนลิสต์ นี่องจากข้อมูลที่เข้ามา รหัสเป็นสตริง ส่วนคะแนนต้องการเก็บเป็นจำนวนจริง แต่ขอเริ่มด้วยการอ่านเข้ามา “นับครั้ง” ให้หมดก่อน ถ้าใช้ข้อมูลนำเข้าของตัวอย่างที่สอง ได้
 $d0 = [['AA', '4 ๐'], ['B', '9.0', '8.0', '7.0'], ['AC', '8.0', '5.0']]$
- นำ d0 มาหาว่ารหัสนักเรียนใด ที่มีจำนวนคะแนนไม่เท่ากับจำนวนที่กำหนดให้ (ตัวอย่างที่สองระบุว่าต้องมี 3 ข้อ)
 ได้ err = ['AA', 'AC']
- ถ้า err มีขนาดเท่ากับ 0 ก็แสดงว่ามีที่ผิด จึงแสดง Invalid data ตามด้วยรหัสนักเรียนที่เก็บใน err
- ถ้า err มีขนาดเป็น 0 (คือไม่ผิดเลย) ประมวลผลต่อ (คราวนี้ขอใช้ข้อมูลจากตัวอย่างแรก เพราะไม่มีที่ผิด)
 ได้ d0 = [['A', '3.0', '2.0'], ['B', '4.0', '5.0'], ['C', '2.0', '3.0']] และ err = []
 - นำ d0 จากขั้นตอนที่แล้ว เปลี่ยนคะแนนให้เป็นจำนวนจริง แล้วหาผลรวม นำมาเก็บคู่กับรหัสนักเรียน
 ได้ d1 = [[5.0, 'A'], [9.0, 'B'], [5.0, 'C']]
 - เรียงลำดับข้อมูลใน d1 ตามโจทย์ คือคะแนนรวมจากมากไปน้อย ถ้าคะแนนรวมเท่ากัน เรียงตามรหัส
 จากน้อยไปมาก ได้ d1 = [[9.0, 'B'], [5.0, 'A'], [5.0, 'C']]
 - 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 วนจำนวน n รอบ อ่านจากแป้นพิมพ์แล้ว split ได้เป็นลิสต์ของสตริง เพิ่มแต่ละลิสต์ที่ได้เข้าในลิสต์ d0 สั่ง run, ใส่ข้อมูลตามตัวอย่างที่สอง, แล้ว print(d0) ได้ [[['AA', '4.0'], ['B', '9.0', '8.0', '7.0'], ['AC', '8.0', '5.0']]]</p>
<pre>err = list() for x in d0 : if len(x[1:]) != m : err.append(x[0]) if len(err) > 0 : print('Invalid data: ') for sid in err : print(sid)</pre>	<p>ขั้นตอนที่ 2, 3 : นำแต่ละลิสต์ x ข้างในลิสต์ d0 จากขั้นตอนที่แล้ว มาตรวจสอบ ถ้า x[1:] มีขนาดไม่เท่ากับ m คือมี จำนวนคะแนนไม่เท่ากับที่กำหนด จะนำ x[0] ใส่เพิ่มในลิสต์ err หลังจากสร้างเสร็จ ถ้า err มีขนาดเกิน 0 ก็แสดง รหัสนักเรียนที่มีความผิดพลาด สั่ง run, ใส่ข้อมูลตามตัวอย่างที่สอง, ได้ Invalid data: AA AC</p>
<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 แจกลิสต์ x ข้างใน d0 ส่วนที่เก็บคะแนน ซึ่งคือ x[1:] มาหาผลรวม โดยต้อง แปลงเป็น float ก่อน จากนั้นนำผลรวมมาต่อกับ x[0] ได้ลิสต์ย่อยเก็บใส่ลิสต์ใหม่ d1 สั่ง run, ใส่ข้อมูลตามตัวอย่างแรก, แล้ว print(d1) ได้ [[5.0, 'A'], [9.0, 'B'], [5.0, 'C']]</p>

โปรแกรม	คำอธิบาย
<pre> for k in range(len(d1)-1) : for i in range(len(d1)-1) : if d1[i][0] < d1[i+1][0] \ or (d1[i][0] = d1[i+1][0]) \ and d1[i][1]>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 : ต้องการเรียงลำดับตามค่าคะแนนรวม แต่ใช้คำสั่ง <code>d1.sort()</code> ไม่ได้ เพราะจะเรียงข้อมูลจากน้อยไปมาก สิ่งที่โจทย์ต้องการคือเรียงตามค่าคะแนนจากมากน้อย แต่ก็ใช้คำสั่ง <code>d1.sort(reverse=True)</code> ไม่ได้ เพราะถ้ากราฟีล์ค่าคะแนนเท่ากัน จะเรียงรหัสนักเรียนจากมากน้อยซึ่งไม่ตรงที่โจทย์ต้องการ</p> <p>จึงขอเขียนการเรียงลำดับเอง (ด้วยวิธี bubble sort ที่เรียนมา) โดยปรับคำสั่งการเปรียบเทียบ คือ จะสลับข้อมูลสองตัวที่ติดกันเมื่อ</p> <ul style="list-style-type: none"> คะแนนรวมของตัวข้างน้อยกว่าคะแนนรวมของตัวขวา หรือ ถ้าคะแนนรวมของห้องสองตัวเท่ากัน รหัสนักเรียนของตัวข้างมากกว่ารหัสนักเรียนของตัวขวา <p>เมื่อเรียงลำดับเสร็จ ก็นำข้อมูลใน <code>d1</code> มาแสดงตามที่โจทย์กำหนด สิ่ง <code>run</code>, ใส่ข้อมูลตามตัวอย่างแรก, ได้ผลลัพธ์ดัง</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 <code>[คะแนนรวม, รหัสนิสิต]</code> เปลี่ยนมาเก็บเป็น <code>[-คะแนนรวม, รหัสนิสิต]</code> แล้วใช้คำสั่ง <code>d1.sort()</code> เพื่อเรียงข้อมูลจากน้อยไปมาก เมื่อคะแนนรวมมีค่าน้อย ติดลบคะแนนรวมจะมีค่าน้อย กล้ายเป็นต้องการเรียงคะแนนรวมจากมากไปน้อย และถ้าเท่ากัน ก็เรียงรหัสนิสิตจากน้อยไปมาก ตามต้องการ ตอนแสดงคะแนนรวม ก็ต้องแสดงค่าติดลบของค่าที่เก็บด้วย (เพราะที่เก็บเป็นค่าติดลบ)</p>

ลองเขียนโปรแกรมนี้ใหม่ ถ้าเราเปลี่ยนการจัดเก็บข้อมูลจาก

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

เป็น

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

DS • วิรจัน จิรพัฒนกุล

Intania 87

กรรมการผู้จัดการ
บริษัท สคูลดิโว จำกัด,
อดีตนักวิทยาศาสตร์ข้อมูล (Data Scientist) ที่ Facebook



ความสามารถในการเขียนโปรแกรมเป็นเหมือนพลังวิเศษที่ช่วยเพิ่มขีดความสามารถในการทำงานของเรา ไม่ว่าจะเป็นการแก้ปัญหาทางวิศวกรรมที่ซับซ้อน (simulation, optimization) การสั่งให้คอมพิวเตอร์ทำงานที่ซ้ำๆ กันไปอัตโนมัติ (automation) การสอนให้คอมพิวเตอร์มีความเฉลียวฉลาดและช่วยเราตัดสินใจได้ (artificial intelligence) รวมไปถึงการสร้างเว็บหรือแอปที่เป็นประโยชน์กับผู้คนในวงกว้าง เรียกได้ว่าเป็นทักษะที่ขาดไม่ได้สำหรับนวัตกรในยุคนี้

06.3 : List Comprehension

สรุปเนื้อหา

List comprehension เป็นวิธีการสร้างลิสต์ที่เขียนได้สั้นและทำงานได้รวดเร็ว

รูปแบบการสร้างลิสต์ด้วย list comprehension กี่พับบอย

สร้างลิสต์ด้วยวนเพิ่มข้อมูล	สร้างลิสต์ด้วย list comprehension
<pre>map: นำข้อมูลจากลิสต์หนึ่งมาประมวลผลเก็บใส่อีกลิสต์ เช่น สร้างลิสต์ b เก็บเฉพาะหลักหน่วยของจำนวนในลิสต์ a b = [] for e in a : b.append(e%10)</pre>	<pre>map: b = [e%10 for e in a]</pre>
<pre>filter: เลือกข้อมูลจากลิสต์หนึ่งมาใส่อีกลิสต์ เช่น สร้างลิสต์ b เก็บสตริงจากลิสต์ a เอกพาะตัวที่ยาวเกิน 5 b = [] for e in a : if len(e) > 5 : b.append(e)</pre>	<pre>filter: b = [e for e in a if len(e)>5]</pre>
<pre>map & filter: ผสมการประมวลผลสองแบบ เช่น นำข้อมูลความสูง (เป็นนิ้ว) ที่เกิน 10 ในลิสต์ a มาแปลง เป็นเซนติเมตรเก็บใส่ลิสต์ b b = [] for e in a : if e > 10 : b.append(2.54*e)</pre>	<pre>map & filter: b = [2.54*e for e in a if e>10]</pre>
<pre>สร้างลิสต์ที่มีการแจกแจงด้วยหลายวนซ้อนกันก็ได้ เช่น เลือกจำนวนเต็มที่มีค่าระหว่าง 0 ถึง 20 สองตัวที่มีผลรวม เท่ากับผลคูณ 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]` เมื่ອันกับ `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() แยกอ่าเป็นลิสต์ของสตริงด้วย split()
นำแต่ละสตริงในลิสต์มาเปลี่ยนเป็นจำนวนเต็ม เว็บไซต์ x
```

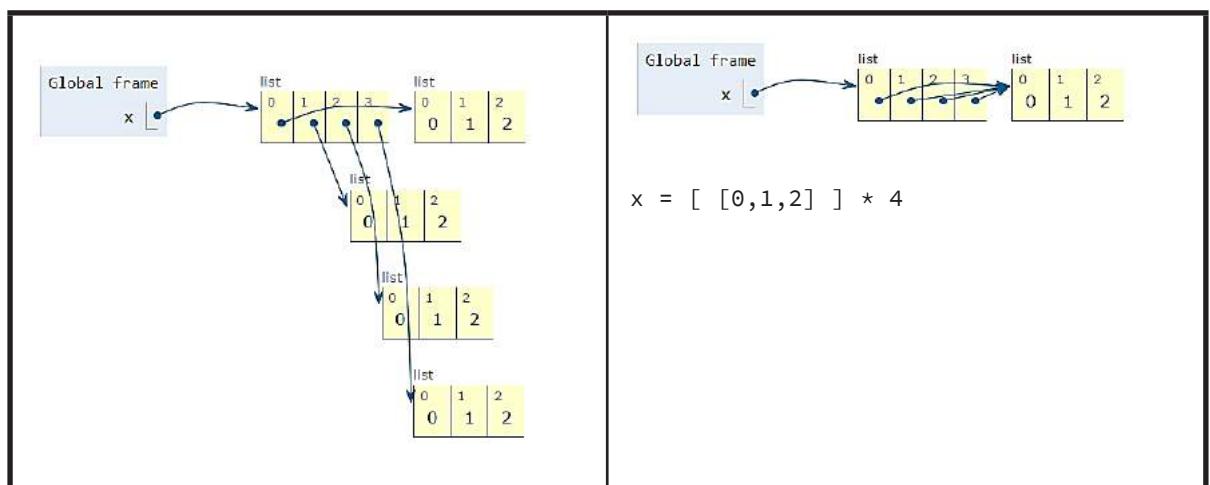
```
t = ','.join( [str(e) for e in x] )
ใช้ list comprehension สร้างลิสต์ของ ตัวรีง ที่นำข้อมูลจากลิสต์ x มาแปลง แล้วส่งผลลัพธ์ที่ได้ไป join กันอีกที
```

```
c = sum( [ 1 for e in x if e%2 == 0 ] )
นับว่าลิสต์ x มีจำนวนคี่ๆ ด้วยส่วน割ิสต์ที่เพิ่มเลข 1 ทุกครั้งที่พบจำนวนคู่ในลิสต์ x และก็หาผลรวมด้วย sum
```

```
b = [ (1 if x[i] >= 0 else -1) for i in range(len(x)) ]
ให้สังเกตการใช้คำสั่ง (-1 if x[i] >= 0 else -1) ที่ได้ผลเป็น 1 ถ้า x[i] >= 0 ไม่ เช่นนั้นได้ผลเป็น -1
จึงเป็นการสร้างลิสต์ b โดยให้ b[i] = 1 ถ้า x[i] >= 0 ไม่ เช่นนั้นให้ 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) ] เมื่อเรียกข้างล่างนี้ทางซ้าย แต่ไม่เมื่อเรียกกับ
x = [ [0,1,2] ] * 4 คือรูปข้างล่างนี้ทางขวา
```

แบบหลังสุดนี้แบกตรงที่ 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]] 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 เป็นลิสต์ช้อนลิสต์ โดยที่ลิสต์ภายในแต่ละตัวเก็บพิกัด x กับ y
ต้องการเรียงลำดับจุดเหล่านี้ตามระยะห่างของจุดถึงจุดกำเนิดจากน้อยไปมาก

สร้างลิสต์ช้อนลิสต์ชั่วคราว a มีลิสต์ภายในเก็บระยะห่าง ตามด้วยพิกัด x กับ y
จากนั้น sort ลิสต์ชั่วคราวนี้ ซึ่งคือการเรียงลำดับตามระยะห่างจากน้อยไปมาก

ปิดท้ายด้วยการใช้ list comprehension หยิบข้อมูลตามลำดับใน a มาสร้างลิสต์ใหม่ที่มีแต่พิกัด x, y เก็บไว้ใน d

เรื่องผิดบ่อๆ

ไม่ควรใช้ list comprehension เพื่อให้ทำงานแบบวนๆ แต่ใช้ลูปน้ำๆ จากการสร้างลิสต์ด้วย list comprehension ไปใช้งานเลย

ไม่ควรเขียน [print(k) for k in range(5)]
ถึงแม้ว่าจะแสดงค่า 0 1 2 3 4 (บรรทัดละค่า) ตามต้องการ
เนื่องจากมีการสร้างลิสต์ แล้วไม่ได้ใช้ เสียเวลาเปล่า ๆ หรือเขียน

[t.append(e) for e in x if e not in t]
เพื่อนำข้อมูลใน x ที่ไม่ปรากฏใน t ไปเพิ่มใน t
จะเกิดการสร้างลิสต์ [None, None, ...] แล้วทีหลัง
ควรเขียนแบบวนๆ ก็ได้

```
for e in x :  
    if e not in t :  
        t.append(e)
```

อย่านำค่าของตัวแปรใน for ของ
list comprehension มาใช้งานนอกคำสั่ง
อาจทำให้สับสน

```
x = [e for e in range(10)]  
print(e) # ผิด เพราะ e ไม่มีค่า
```

แต่ถ้าเขียน

```
x = []  
for e in range(10) :  
    x.append(e)  
print(e) # ได้ 9
```

แต่ถ้าแบบนี้

```
e = 99  
x = [e for e in range(10)]  
print(e) # ได้ 99
```

แบบฝึกหัด

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

Problem	Code
<p><u>Input:</u> x เป็น list of lists of integers</p> <p><u>Process:</u> หาผลรวมของจำนวนเต็มใน x</p>	
<p><u>Input:</u> รับหนึ่งบรรทัดมีจำนวนเต็มหลายจำนวน (คั่นด้วยช่องว่าง) จากแป้นพิมพ์</p> <p><u>Process:</u> หาว่าที่รับมามีจำนวนหล. 黎 จำนวน</p>	
<p><u>Input:</u> รับข้อความ หนึ่งบรรทัดจากแป้นพิมพ์</p> <p><u>Process:</u> ตัดอักขระทุก... เนื้อความที่รับมาที่ไม่ใช่พยัญชนะภาษาอังกฤษ</p>	
<p><u>Input:</u> รับหนึ่งบรรทัดมีจำนวนเต็มหลายจำนวน (คั่นด้วยช่องว่าง) ชุดหนึ่ง เก็บใส่ลิสต์ x และรับอีกหนึ่งบรรทัดเก็บใส่ลิสต์ y ในทำนองเดียวกัน โดยลิสต์ทั้งสองมีจำนวนข้อมูลเท่ากัน</p> <p><u>Process:</u> สร้างลิสต์ z โดยที่ $z[i]$ มีค่าเท่ากับ $x[i] + y[i]$</p>	
<p><u>Input:</u> เมทริกซ์ m แทนด้วย list of lists of integers</p> <p><u>Process:</u> แปลง m ให้กลายเป็น list of integers เช่น จาก $m = [[1,2,3],[4,5,6]]$ กลายเป็น $[1,2,3,4,5,6]$</p>	
<p><u>Input:</u> รับหนึ่งบรรทัดมีจำนวนเต็มหลายจำนวน (คั่นด้วยช่องว่าง) เก็บใส่ลิสต์ x</p> <p><u>Process:</u> สร้างลิสต์ใหม่จากข้อมูลใน x แต่ไม่มีตัวซ้ำ โดย <ol style="list-style-type: none"> เรียงลำดับข้อมูลใน x จากน้อยไปมาก สร้างลิสต์ใหม่โดยนำข้อมูลในลิสต์ตัวที่ติดกันมาพิจารณาจาก ซ้ายไปขวา ถ้าค่าของตัวติดกันตัวซ้ายไม่เท่ากับตัวขวา ให้นำ ตัวซ้ายเก็บใส่ลิสต์ใหม่ นำตัวขวาสดใน x เพิ่มต่อท้ายลิสต์ใหม่นี้ ลิสต์ใหม่นี้ก็จะเก็บข้อมูลใน x ที่ไม่มีตัวซ้ำ </p>	
<p><u>Input:</u> รับจำนวนเต็ม n</p> <p><u>Process:</u> สร้างลิสต์ c ที่เก็บจำนวนประกอบที่มีค่าน้อยกว่า n เริ่มจากสร้างลิสต์ใหม่ x โดย <ul style="list-style-type: none"> เพิ่มค่า $4, 6, 8, 10, \dots$ (ไม่เกิน $n-1$) ใน x เพิ่มค่า $6, 9, 12, 15, \dots$ (ไม่เกิน $n-1$) ใน x เพิ่มค่า $8, 12, 16, 20, \dots$ (ไม่เกิน $n-1$) ใน x ... <p>ไปเรื่อย ๆ ตราบเท่าที่ยังไม่เกิน $n-1$ นั่นคือ เพิ่มค่า $2k, 3k, 4k, 5k, \dots$ (ไม่เกิน $n-1$) ใน x โดยประค่า $k = 2, 3, 4, \dots, N//2-1$</p> <p>จากนั้นสร้างลิสต์ใหม่ c ที่ได้ข้อมูลจาก x แต่ไม่มีตัวซ้ำ (ใช้วิธีที่เขียนในข้อที่แล้ว)</p> </p>	

Problem	Code
<p><u>Input:</u> รับจำนวนเต็ม n</p> <p><u>Process:</u> สร้างลิสต์ที่เก็บจำนวนเฉพาะที่มีค่าอยู่กว่า n โดย</p> <ol style="list-style-type: none"> สร้างลิสต์ที่เก็บจำนวนบวกๆ ที่มีค่าน้อยกว่า n สร้างลิสต์ที่เก็บจำนวนที่หารด้วย 2 ถึง $n-1$ แต่ไม่เอาจำนวนประกอบที่ทางในขั้นตอนที่ 1 	

ตัวอย่างการแก้โจทย์ปัญหา

เรียงตามค่าแบบรวม

ขอใช้ปัญหาระบบแบบรวมในหัวข้อลิสต์ช้อนลิสต์ 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) > 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) > 0 : print('Invalid data: ') print('\n'.join(err)) else : # list comprehension ช้อนใน # 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

โจทย์ข้อนี้ กำหนดให้เป็นโปรแกรมรับชุดตัวเลขจำนวนเต็มชุดหนึ่ง ซึ่งจะจับด้วยเลขที่มีค่าติดลบ 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 ของรหัสนิสิตที่กำหนด เมื่อเรียงลำดับตามคะแนน (rank 1 คือมีคะแนนมากที่สุด)

► ข้อมูลนำเข้า

แต่ละบรรทัดจะระบุข้อมูลนิสิตแต่ละคน ประกอบด้วยรหัสนิสิต ตามด้วยคะแนนเป็นเลขศูนย์

บรรทัดสุดท้าย ระบุรหัสไว้เพื่อตัด การคำนวณหา rank

ในการคำนวณ rank หากมีนิสิตใดคะแนนเท่ากัน ให้เรียงลำดับตามรหัสนิสิต (เรียงแบบจำนวนเต็ม)

► ข้อมูลส่งออก

มีบรรทัดเดียว แสดง รากแรก ของนิสิตที่ต้องการค้นหา หากไม่บรรทัดนิสิตตั้งกล่าว 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

ความถี่เก็บครึ่ง

ให้เขียนโปรแกรมเพื่อหาข้อมูลที่มีความว่าๆ กัน จำนวนหนึ่งของจำนวนข้อมูลทั้งหมดที่รับเข้ามา โดยอ่านค่าจำนวนเต็มจำนวนหนึ่ง พบค่า -1

▶ ข้อมูลนำเข้า

ให้อ่านข้อมูลจำนวนเต็มบรรทัดละ 1 จำนวน จำนวน จนกระทั่งพบค่า -1

▶ ข้อมูลส่งออก

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

สามารถสร้าง tuple ของ list ได้	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
สามารถสร้าง tuple ของ tuple ก็ได้	T = ((1,2), (3,4)) เป็น tuple ของ tuple T[1][0] ได้ 3
ข้อแตกต่างระหว่าง tuple1 += (0,) กับ list1 += [0] tuple1 += (0,) เมื่อตอน tuple1 = tuple1 + (0,) จะสร้าง tuple ใหม่ (ทำงานช้า) ในขณะที่ list1 += [0] ไม่เหมือนกับ list1 = list1 + [0] แต่จะเมื่อตอนกับ list1.append(0) เป็นการเพิ่ม 0 ต่อท้ายลิสต์ ไม่สร้างลิสต์ใหม่ จะทำงานได้เร็วกว่า	ลองสัง.eng ทำงานโปรแกรมทางขวาแล้วสังเกตเวลาการทำงาน 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)

Dictionary

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

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

<p>key ของ dict อาจเป็นจำนวนเต็ม จำนวนจริง สตริง หรือ tuple ก็ได้ แต่ห้ามเป็น list, dict หรือ set และ value ที่คู่กับ key จะเป็นประเภทใดๆ ก็ได้</p>	<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>
<p>มีบริการ keys(), values(), items() ซึ่งใช้กับ for...in ได้</p>	<pre>d = {1:7, 2:8, 3:9} for e in d : แจกแจงแต่ละ key ของ d ให้กับ e for e in d.keys() แจกแจงแต่ละ key ของ d ให้กับ e for e in d.values() แจกแจงแต่ละ value ของ d ให้กับ e for e in d.items() แจกแจงแต่ละคู่ข้อมูลของ d เป็น tuple ให้กับ e e[0] คือ key, e[1] คือ value for k,v in d.items() แจกแจง key, value ของ d ให้กับ k, v ลำดับของข้อมูลที่แจกแจงออกมากไม่จำเป็นต้องเหมือนกับที่เขียนหรือที่เพิ่ม</pre>
<p>บริการ keys(), values() และ items() ไม่ได้ผลเป็น list แต่เป็น อะไรมากอย่างที่คล้าย list นำไปใช้กับ for...in ได้ แต่ใช้ index เพื่อหยิบ ข้อมูลไม่ได้</p>	<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 เป็นลิสต์ใช้ x[1] ได้ print(x[1]) ทำได้ (แต่เมื่อรู้ว่าได้ 'aka' หรือ 'so')</pre>
<p>บริการ d.update(d1) เป็นการเพิ่ม ข้อมูลใหม่จาก dict d1 เข้าใน d ถ้ามี key ใน d1 ซ้ำกับใน d จะเป็นการแก้ไข value เดิม</p>	<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>
<p>บริการ pop(key) เป็นการลบคู่ข้อมูล ที่มี key นั้นออก ถ้าไม่มี key อยู่ จะเกิด error</p>	<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>
<p>การใช้ in เพื่อตรวจสอบว่ามี key หรือ value หรือคู่ (key,value) ใน dict อยู่หรือไม่ โดยการตรวจ value จะไม่เร็วเท่ากับอีกสองกรณี</p>	<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 การค้นใน keys และ items เร็วมาก แต่การค้นใน values ไม่เร็วเท่า</pre>

Set

- เหมือนกับเซตทางคณิตศาสตร์ เชตไม่เก็บข้อมูลซ้ำ และมี operation union, intersect ได้
- เหมือนกับ dict แบบที่เก็บแค่ key, ไม่มี value (เพราะ key ไม่ซ้ำกัน)
- การค้นด้วย in สามารถทำได้เร็วมาก

<p>การสร้างเซต สามารถสร้างเซตจากข้อมูลใน string, tuple, list, set หรือ dict ได้</p>	<pre>set_1 = set() set_2 = {1, 2, 3} set_3 = set('Hello') set_4 = set(['oh', 'no']) set_5 = set(set_2) set_6 = set({1:2, 3:4})</pre>	<p>ได้เซตว่าง ได้เซตที่มีสมาชิกเป็น 1, 2, 3 ได้เซต {'H', 'e', 'l', 'o'} ได้เซต {'oh', 'no'} ได้เซตใหม่มีสมาชิก เหมือนของ set_2 ได้เซตของ key ของ dict คือ {1,3}</p>
<p>บริการ add(e) ใช้เพิ่มข้อมูล 1 ตัว ถ้าเพิ่มตัวที่ซ้ำกับที่มีอยู่ใน set ก็ไม่เพิ่มให้</p>	<pre>S = {1} S.add((2, 3)) S.add('Hello')</pre>	<p>ได้ {1, (2, 3)} ได้ {1, (2, 3), 'Hello'}</p>
<p>บริการ update(t) ใช้เพิ่มข้อมูลที่แยกแจง มาจาก t ที่เป็นกลุ่มข้อมูล ซึ่งเป็นได้ทั้ง string, tuple, list, set และ dict (ในกรณีของ dict จะเพิ่ม key) คำสั่ง s.update(t) ทำงานเหมือนคำสั่ง <code>for e in t : s.add(e)</code></p>	<pre>S = {1} S.update([2, 3]) S.update('Hello') S.update(6)</pre>	<p>ได้ {1, 2, 3} ได้ {1, 2, 3, 'H', 'e', 'l', 'o'} เกิด error</p>
<p>การใช้ e in S เพื่อตรวจสอบความเป็น สมาชิกของ e ในเซต S</p>	<pre>S = {1, 2, 3, 'H', 'e', 'l', 'o'} x = 1 in S y = 'h' in S</pre>	<p>ได้ x = True ได้ y = False</p>
<p>บริการ remove(e) และ discard(e) เพื่อลบข้อมูล ถ้าไม่มีข้อมูลที่ต้องการจะลบ remove() จะเกิด error ดังนั้นควรใช้ discard() <code>S.remove('e')</code></p>	<pre>S = {1, 2, 3, 'H', 'e', 'l', 'o'} S.discard(1) S.discard(4) S.remove('e')</pre>	<p>เหลือ {2, 3, 'H', 'e', 'l', 'o'} เหลือเท่าเดิม ไม่เกิด error เหลือ {2, 3, 'H', 'l', 'o'} เกิด error เพราะไม่มี 'h'</p>

<p>การดำเนินการของ set</p> <p><code>union()</code> หรือใช้สัญลักษณ์ <code> </code></p> <p><code>intersection()</code> หรือใช้สัญลักษณ์ <code>&</code></p> <p><code>difference()</code> หรือใช้สัญลักษณ์ <code>-</code></p> <p>$(A \cup B) - (A \cap B)$ หรือใช้วิจารณ์ลักษณ์ <code>^</code></p> <p>โดยเชตที่มากกว่า ก็จะ จำกัด ไปยัง ปริญนแปลง</p>	<pre>a = {1, 2, 3} b = {2, 3, 4} c = a.union(b) หรือ c = a b ได้ c = {1, 2, 3, 4} c = a.intersection(b) หรือ c = a & b ได้ c = {2, 3} c = a.difference(b) หรือ c = a - b ได้ c = {1} c = b.difference(a) หรือ c = b - a ได้ c = {4} c = a ^ b ได้ c = {1, 4}</pre>
<p>บริการ <code>issubset()</code> และ <code>issuperset()</code> ใช้ตรวจสอบความเป็น subset และ superset</p>	<pre>a = {1, 2, 3} b = {1, 2} a.issubset(b) ได้ False b.issubset(a) ได้ True a.issuperset(b) ได้ True b.issuperset(a) ได้ False</pre>

สรุปการใช้งาน list, tuple, dict, set

- ใช้คำสั่ง `len, sum, max, min, sorted` ได้ทั้งหมด
- ใช้ `x = sorted(q)` ได้โดยที่ `q` เป็นวิธีทั้ง `list, tuple, dict, set`
ผลที่ได้เป็น `list` ที่นำข้อมูลที่แจ้ง "เจ้า" จาก `q` ไปเรียงลำดับ

	list	tuple	dict	set
การใช้งาน	ลำดับมีความหมาย สร้างแล้วแก้ไขได้	ลำดับมีความหมาย สร้างแล้วแก้ไขไม่ได้	เก็บคู่ลำดับ (key, value) key ไม่ซ้ำ, ไม่สนล้ำดับ	เก็บชื่ออยู่ในเซ็ต ไม่สนลำดับ สามารถใช้ set operation ได้
ประเภทข้อมูลที่เก็บ	อะไรมีได้	อะไรมีได้	key เป็น int, float, str, tuple, bool ส่วน value เป็นอะไรมีได้	int, float, str, tuple, bool
การเข้าถึงข้อมูล	ใช้จำนวนเต็มระบุตำแหน่งหรือช่วง <code>x[i]</code> <code>x[a:b:c]</code>	ใช้จำนวนเต็มระบุตำแหน่งหรือช่วง <code>t[i]</code> <code>t[a:b:c]</code>	ใช้ key ระบบตำแหน่ง <code>d[key]</code> ได้ value ที่คู่กัน ไม่มีแบบรับ value แล้วได้ key	ไม่มี
การแจกแจงข้อมูล	<code>for e in x</code> ได้ลำดับจากซ้ายไปขวา	<code>for e in t</code> ได้ลำดับจากซ้ายไปขวา	<code>for k in d</code> <code>for k in d.keys()</code> <code>for v in d.values()</code> <code>for k,v in d.items()</code> ได้ลำดับไม่แน่นอน	<code>for e in s</code> ได้ลำดับไม่แน่นอน
การค้นด้วย <code>in, not in</code>	ค้นจากซ้ายไปขวา (ซ้ำ) ใช้ <code>x.index(e)</code> หา index ของ <code>e</code> ใน <code>x</code>	ค้นจากซ้ายไปขวา (ซ้ำ) ใช้ <code>x.index(e)</code> หา index ของ <code>e</code> ใน <code>x</code>	ค้น key เร็วมาก	ค้นข้อมูลเร็วมาก

	list	tuple	dict	set
การสร้าง	x = [1,2,3,4] x = list() x = [] x = list(q) เมื่อ q เป็นสิ่งที่ใช้กับ for in ได้	t = (1, 2, 3, 4) t = () t = tuple() t = tuple(q) เมื่อ q เป็นสิ่งที่ใช้กับ for in ได้	d = {'k1':1,'k2':2} d = dict() d = {}	s = {1, 2, 3, 4} s = set() s = set(q) เมื่อ q เป็นสิ่งที่ใช้กับ for in ได้
	ใช้ x = list(x1) เมื่อ x1 เป็น list ไม่ควรเขียน x = x1 จะเป็น list ตัวเดียวกัน	ใช้ t = t1 ได้ เมื่อ t1 เป็น tuple เพรา tuple ไม่เปลี่ยนค่า	ใช้ d = dict(d1) เมื่อ d1 เป็น dict ไม่ควรเขียน d = d1 จะเป็น dict ตัวเดียวกัน	ใช้ s = set(s1) เมื่อ s1 เป็น set ไม่ควรเขียน s = s1 จะเป็น set ตัวเดียวกัน
การเพิ่ม ข้อมูล	x.append(9) x.insert(1,9)	ต้องสร้างตัวใหม่ t=t + (9,) t=t[:1]+(9,)+t[1:]	d['k3'] = 9 d.update({'k3':9})	s.add(9) s.update({9})
การลบ ข้อมูล	x.pop(2)	ต้องสร้างตัวใหม่ t = t[:2] + t[3:]	d.pop('k3')	s.discard(99)
การแก้ไข ข้อมูล	x[2] = 7	ต้องสร้างตัวใหม่ t=t[:2]+(7,)+t[3:]	d['k3'] = 7	ต้องลบแล้วเพิ่ม

การใช้ tuple, dict, set ที่พบบ่อย

ใช้ dict เพื่อจับคู่ข้อมูล หรือสร้างความสัมพันธ์ระหว่างคู่ข้อมูล key กับ value โดยหวังจะขอ value จากค่า key	month = {'JAN':1, 'FEB':2, 'MAR':3} ใช้ month[k] เพื่อขอเลขเดือนจากชื่อย่อเดือนที่เก็บในตัวแปร k num2en = {11:'eleven', 2:'two', 3:'three'} ใช้ num2en[a] เพื่อขอคำภาษาอังกฤษจากจำนวนเต็มในตัวแปร a
ใช้ set เพื่อกีบกลุ่มข้อมูลที่ไม่ซ้ำ ต้องการค้นข้อมูลว่ามีอยู่หรือไม่ในเซตโดยรอดเร็ว หรือต้องการบริการ intersection, union, issubset และอื่น ๆ ของ set	ต้องการหาจำนวนประกอบที่มีค่าระหว่าง 2 ถึง 12 n = 13; c = set() for i in range(2, n//2) : for j in range(2*i, n, i) : c.add(j) # มีการเพิ่ม j ที่ซ้ำกัน แต่ add ไม่เพิ่มข้อมูลซ้ำ # c = {4, 6, 8, 9, 10, 12}

<p>ใช้ tuple เมื่อต้องการเก็บข้อมูลมีลำดับคล้าย list แต่มั่นใจว่าหลังสร้างแล้วจะไม่เปลี่ยนค่าภายใน tuple ประหรับด้หน่วยความจำ</p> <p>ใช้คำสั่ง <code>t1 = t2</code> ได้โดยมัตต้องพิจารณา การใช้ tuple ร่วมกับ (ตรวจสอบค่าภายในเปลี่ยนไม่ได้) และสามารถใช้เป็น key ของ dict และเป็นข้อมูลที่เก็บใน set ได้</p>	<p>เมื่อจัด <code>x, y, z</code> จำนวนหนึ่งที่ต้องเก็บ ถ้าต้องการหา <code>z</code> จาก <code>x, y</code> ที่กำหนด ห้ามอยู่ ๆ ก็ไม่ควรเก็บเป็น list of tuples <code>(x,y,z)</code> เช่น</p> <pre>d = [(1,1,3), (2,1,8), (3,1,2)]</pre> <p>แบบนี้การหา <code>z</code> จาก <code>x, y</code> ก็ต้องเป็น</p> <pre>for a,b,z in d : if x == a and y == b : print('z =', z) break else: print('Not Found')</pre> <p>ควรใช้ dict <code>{(x,y):z}</code> เช่น <code>d={(1,1):3,(2,1):8,(3,1):2}</code></p> <p>แบบนี้ การหา <code>z</code> จาก <code>x, y</code> ก็ง่ายและที่สำคัญคือเร็ว</p> <pre>if (x,y) in d : print('z =', d[(x,y)]) else : print('Not Found')</pre>
<p>การแยกแจงข้อมูลต่าง ๆ ใน dict</p>	<pre>d = {1:7, 2:8, 3:9} for k in d: print(k) for k in d.keys(): print(k) for v in d.values(): print(v) for k,v in d.items(): print(k,v)</pre>
<p>การแยกแจงข้อมูลใน dict เรียงตาม key หรือเรียงตาม value จากน้อยไปมาก</p>	<pre>d = {2:8, 1:8, 3:9} for k in sorted(d.keys()): print(k,d[k]) for v in sorted(d.values()): print(v)</pre>
<p>การรับคู่อันดับข้อมูล แล้วเพิ่มลงใน dict</p>	<pre>d = [] n = int(input()) for i in range(n): k,v = input().split() d[k] = v</pre>
<p>การรับคู่อันดับข้อมูล แล้วเพิ่มลงใน dict ที่มี value เป็น list หรือ set</p>	<pre>d = [] n = int(input()) for i in range(n): k,v = input().split() if k not in d: d[k] = [v] # ถ้า <code>d[k]</code> เป็นชุด, ใช้ <code>d[k] = {v}</code> else : d[k].append(v) # ถ้า <code>d[k]</code> เป็นชุด, ใช้ <code>d[k].add(v)</code></pre> <p>ข้อควรระวัง : คำสั่ง <code>d[k] = [v]</code> ถ้าเขียนเป็น <code>d[k] = list(v)</code> จะมีปัญหา ถ้า <code>v</code> เป็นสตริง (เพราะอะไร ?)</p>

<p>สร้างลิสต์ของ tuple เก็บข้อมูลชั่วคราว เพื่อการประมวลผล (เช่น เรียงลำดับข้อมูล)</p>	<p>d เป็น dict ที่ key เป็นรหัสนิสิต ส่วน value เป็นลิสต์ของคะแนน ต้องการแสดงชื่อเรียงลำดับตามค่าคะแนนรวมจากน้อยไปมาก <code>x = [(sum(scores), sid) for sid, scores in d.items()] t = [sid for sum_scores, sid in sorted(x)] print('\n'.join(t))</code></p>
<p>dict มี key เป็น a, value เป็นเซ็ตของ b ต้องการสร้างอีก dict : ที่กลับกันนะ การจัดเก็บคือว่า key เป็น b, value เป็นเซ็ต ของ a</p>	<p>c เป็น dict ที่ key เป็นรหัสนิสิต ส่วน value เป็นเซ็ตของรหัสวิชา จึงใช้ c ตอบคำถามว่ารหัสนิสิต sid เรียนวิชาอะไร ได้อย่างรวดเร็ว ถ้า อยากรู้ด้วยว่า รหัสวิชา cid มีรหัสนิสิตใดเรียนบ้าง ก็ต้องสร้างอีก dict stu เป็น dict ที่ key เป็นรหัสวิชา ส่วน value เป็นเซ็ตของรหัสนิสิต <code>stu = dict() for (sid, cids) in c.items() : for cid in cids : if cid not in stu : stu[cid] = {sid} else : stu[cid].add(sid)</code></p>
<p>นำชุดข้อมูลที่อาจมีค่าซ้ำกันมาเพิ่มใส่ set แล้วได้ชุดข้อมูลที่ไม่มีค่าซ้ำ</p>	<p>จาก dict c ในข้อที่แล้ว อยากรายบ่าว่ามีรหัสวิชาอะไรบ้างที่มีนิสิตเรียน <code>d = set() for cids in c.values() : d.update(cids)</code></p>

เรื่องผิดป่วย

<p>คำสั่ง a = b โดยที่ b คือ set, dict หรือ list จะทำให้ a กับ b เป็นตัวแปร ของที่เก็บข้อมูลที่เดียวกัน ถ้าต้องการเป็น คุณลักษณะ แต่เก็บข้อมูลเหมือนกัน ต้องใช้ <code>a = set(b)</code> หรือ <code>a = dict(b)</code> หรือ <code>a = list(b)</code></p>	<p>A = {1:'one', 2:'two', 3:'three', 10:'ten'} B = A B[4] = 'four' จะเป็นการแก้ทั้ง A และ B เพราะเป็น dict เดียวกัน ถ้าใช้คำสั่ง C = dict(A) จะได้ A และ C เป็นคุณลักษณะ dict กัน แต่มีข้อมูลเหมือนกัน</p>
<p>ไม่สามารถใช้ d.sort() เมื่อ d เป็น tuple, dict หรือ set ได้ แต่สามารถ ใช้ sorted(d) ได้ โดย d คือ tuple, dict, set หรือ list</p>	<p>S = {3, 1, 2}; D = {'A':5, 'C':2, 'B':7} <code>S.sort()</code> ผิด เพราะ set ไม่มีบริการ sort <code>L = sorted(S)</code> ได้ L = [1, 2, 3] <code>L = sorted(D)</code> ได้ L = ['A', 'B', 'C'] <code>L = sorted(D.values())</code> ได้ L = [2, 5, 7]</p>

tuple ที่มีตัวเดียวต้องมี comma ต่อท้าย	<pre>my_tuple = (1,) # ได้ tuple ที่มีตัวเดียว (ลั่งเกตที่ comma) my_tuple += (4,) # ได้ (1, 4) my_tuple += 5 # ผิด นำ 5 ไปรวมกับ tuple ไม่ได้ my_tuple += (5) # ผิด เชิญ (5) ก็เหมือน 5 not_a_tuple = (1) # ได้จำนวนเต็มธรรมด้า</pre>
tuple แก้ข้อมูลไม่ได้ tuple ไม่มีบริการ append, insert, add, pop, remove, discard	<pre>my_tuple[3] = 'B' ผิด ถ้าจะแก้ข้อมูล ต้องสร้างใหม่ my_tuple = my_tuple[:3] + ('B',) + my_tuple[4:]</pre>
การอ้างถึงข้อมูลใน dict ที่ไม่มีมาก่อน จะผิด	<pre>D = {'Name': 'Tom', 'Age': 39} print(D['Gender']) ผิด เพราะอ้างได้แค่ 'Name' และ 'Age' หรือ ต้องการนับจำนวนตัวอักษรภาษาอังกฤษแต่ละตัวว่าปรากฏกี่ครั้งในสตริง t c = dict() for e in t : c[e] += 1 # ผิด เพราะอาจไม่มี key e ใน t ต้องเปลี่ยนเป็น for e in t : if e not in c : c[e] = 1 else : c[e] += 1</pre>
key ของ dict ห้ามเป็น list, dict หรือ set	<pre>my_dict = {} my_dict[[1,2,3]] = 'list' ผิด เพราะ [1,2,3] ใช้เป็น key ไม่ได้</pre>
การเก็บคู่อันดับใน dict อาจจะไม่เรียง ลำดับตามลำดับการใส่ข้อมูล	<pre>D = {} D[1] = 1.00; D[2] = 2.00 for k in D.keys(): # อาจได้ 1 แล้ว 2 หรือ 2 แล้ว 1 ก็ได้ ไม่แน่นอน</pre>
ใช้งานหา key เพื่อให้ได้ value ที่คุ้นเคยแบบนี้ไม่ได้ใช้ความสามารถของ dict เลย	<p>ต้องการหา value ของ key ที่มีค่าเท่ากับ key1 ใน d ไม่ควรเขียนแบบข้างล่างนี้</p>
	<pre>for k,v in d.items() : if k == key1 : print('value =', v) break else : print('Not found')</pre> <p>เขียนแบบข้างล่างนี้ง่ายกว่า และเร็วกว่ามาก</p> <pre>if key1 in d : print('value =', d[key1]) else : print('Not found')</pre>

วงเล็บปีกกว่าง คือ dict ไม่ใช่ set	= {} เป็นการสร้าง dict ว่าง ๆ
การเก็บข้อมูลใน set อาจจะไม่เรียงลำดับ ตามลำดับการใส่ข้อมูล	S = {1, 2} print(S) อาจได้ {1, 2} หรือ {2, 1} ก็ได้ ไมแน่นอน
การสร้างเซตจากสตริง ต่างจากการสร้างชุด จากลิสต์ของสตริง	set_1 = {'Hello'} ได้เซต {'Hello'} set_2 = set('Hello') ได้เซต {'H', 'e', 'l', 'o'} set_3 = set(['Hello', 'World']) ได้เซต {'Hello', 'World'} การเขียน s = set(q) เหมือนการเขียน s = set() for e in q : s.add(e)
ห้ามเก็บ list, dict หรือ set ใน set	my_set = {1} หรือ my_set.add((1,2)) ทำได้ my_set.add([2,3,4]) ผิด
การใช้ตัวดำเนินการของ set ไม่เปลี่ยนค่า ในเซตเดิม	a = {1, 2, 3} b = {2, 3, 4} c = a.union(b) ได้ c = {1, 2, 3, 4} แต่ค่าของ a = {1, 2, 3} และ b = {2, 3, 4} ไม่เปลี่ยนแปลง
ใช้ index เพื่อเข้าใช้ข้อมูลใน set ไม่ได้ เพราะข้อมูลใน set ไม่มีลำดับ ไม่มี index	A = ['one', 'two', 'three'] for i in range(len(A)) : print(A[i]) แบบนี้ผิด ควรใช้ for...in แทน for e in A : print(e)



Problem	Code
<p>ลองคิดดูว่าการจะตอบคำถามแบบนี้ ควรเก็บข้อมูลด้วยอะไร</p> <ol style="list-style-type: none"> 1) ถามเกรดวิชา comp prog จากรหัสนิสิต 2) มีรายชื่อนิสิตที่ลงทะเบียนเรียนวิชา comp prog เพื่อถามว่านิสิตที่มี รหัส x ลงทะเบียน comp prog หรือไม่ 3) ให้ข้อภาควิชา แล้วถามว่า มีนิสิตคนใดอยู่ภาควิชานั้นบ้าง 4) ต้องการเก็บข้อมูลรุ่นโทรศัพท์มือถือที่เขาเคยใช้ เรียงจากอดีตจนถึง ปัจจุบัน ของนิสิตคนหนึ่ง 5) อยากรู้ว่าหมายเลขนิสิตที่ส่วนใหญ่ลงท้ายด้วยเลขอะไร 	

Problem	Code
<p><u>Input:</u> รับจำนวนเต็มบวก 1 จำนวนจากแป้นพิมพ์เข้าใน x</p> <p><u>Process:</u> สร้าง tuple ของจำนวนเต็มคู่ที่แต่ 0 และน้อยกว่า x</p> <p><u>Output:</u> tuple ที่สร้าง</p> <p>เช่น รับค่า 10 ให้แสดงผล (0, 2, 4, 6, 8)</p>	
<p><u>Input:</u> รับจำนวนเต็มบวก 1 จำนวนจากแป้นพิมพ์ เก็บใน x</p> <p><u>Process:</u> สร้าง tuple ของการแยกหลักของ x</p> <p><u>Output:</u> tuple ที่สร้าง</p> <p>เช่น รับค่า 12803 ให้แสดงผล (1, 2, 8, 0, 3)</p>	
<p><u>Input:</u> รับสตริงจากแป้นพิมพ์ เก็บใน x</p> <p><u>Process:</u> สร้าง dict และการนับตัวอักษรของ x</p> <p><u>Output:</u> dict ที่สร้าง (ไม่สนใจลำดับที่แสดงผล)</p> <p>เช่น รับค่า book ให้แสดงผล {'b':1, 'k':1, 'o':2}</p>	
<p><u>Input:</u> รับสตริง 2 บรรทัดจากแป้นพิมพ์ เก็บใน x และ y</p> <p><u>Process:</u> สร้าง set ของตัวอักษรใน x และ set ของตัวอักษรใน y จากนั้นนำมาหาตัวอักษรที่ปรากฏในทั้งสองสตริง</p> <p><u>Output:</u> set ของ ตัวอักษรที่ปรากฏในทั้งสองสตริง (ไม่สนใจลำดับที่แสดงผล) เช่น รับค่า book และ bank ให้แสดงผล {'b', 'k'}</p>	

ตัวอย่างการแก้โจทย์ปัญหา

ทงเบียนบันลือต

จงเขียนโปรแกรมรับรายการของข้อมูล ซึ่งประกอบด้วยชื่อของนิสิตและคณะที่นิสิตคนนั้นอยู่ จำนวนจะกำหนดชื่อคณะมาให้จำนวนหนึ่ง เนื้อหาว่า นิสิตในคณะเหล่านั้นมีชื่ออะไรบ้าง ถ้ามีชื่อซ้ำกัน ให้ตอบเพียงครั้งเดียว

▶ ข้อมูลนำเข้า

บรรทัดแรกมีจำนวนเต็มบวก n คือจำนวนรายการข้อมูลทั้งหมด

n บรรทัดต่อมา แต่ละบรรทัดประกอบด้วย ชื่อของนิสิตและคณะที่นิสิตอยู่ โดยคั่นด้วยช่องว่าง

บรรทัดสุดท้ายจะเป็นรายชื่อคณะที่ต้องการถาม ถ้ามีหลายชื่อจะคั่นด้วยช่องว่าง

▶ ข้อมูลส่งออก

แสดงชื่อนิสิตในคณะเหล่านั้น โดยเรียงตามตัวอักษร ให้คั่นแต่ละชื่อด้วยช่องว่าง ถ้ามีชื่อซ้ำกัน ให้ตอบเพียงครั้งเดียว

▶ ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
3 Tom Engineering Pam Arts Jim Engineering Engineering	Jim Tom
5 Tom Engineering Pam Arts Jim Engineering Tom Arts Jenny Science Engineering Arts	Jim Pam Tom

ตัวอย่างการเขียนโปรแกรม

เนื่องจากโจทย์จะกำหนดชื่อคณะมาให้ และให้เราหาชื่อนิสิตที่อยู่ในคณะเหล่านั้น ดังนั้นเราควรเก็บข้อมูลโดยใช้ dict ซึ่งมี key คือชื่อคณะ และ value เป็น set ของชื่อนิสิต (เพราะในแต่ละคณะ มีนิสิตได้หลายคน) ตัวอย่างการเก็บข้อมูล เช่น

```
{  
    'Engineering' : {'Tom', 'Jim'},  
    'Arts'        : {'Pam', 'Tom'},  
    'Science'     : {'Jenny'}  
}
```

ข้อแบบการประมวลผลเป็นขั้นตอนที่ละเอียดดังนี้

- อ่านข้อมูลนำเข้าเก็บเป็น dict
- รับรายการของชื่อคณะที่ต้องการรวมมาเก็บใน list
- เนื่องจากเราต้องการพิมพ์ชื่อที่ไม่ซ้ำกัน ดังนั้นเราจะใช้ set มาช่วย ให้สร้าง set คำตอบเริ่มต้นเป็นเซตว่าง
- วนลูปทีละคณะที่ต้องการหา ทึ้งน้ำเราจะใช้ set มาช่วย ให้สร้าง set คำตอบเริ่มต้นเป็นเซตว่าง
- พิมพ์คำตอบ โดยใช้ list จับชื่อตามตัวอักษร

โปรแกรม	คำอธิบาย
ขั้นตอนที่ 1	
<pre>n = int(input()) fac2name = {} for i in range(n): name, fac = input().split() fac2name[fac] = {name}</pre>	อ่านค่า n ซึ่งเป็นจำนวนข้อมูลเข้ามา สร้าง dict ว่าง ๆ ชื่อ fac2name ซึ่งจะใช้เก็บว่า คณะนี้ มีนิสิตชื่ออะไรบ้าง จำนวนนับลูป n รอบ เพื่อเก็บข้อมูลคู่ (key, value) คือ ชื่อคณะและ set ของชื่อนิสิต แต่ตัวอย่างนี้ผิด เพราะแต่ละ key ของ dict จะเก็บ value ได้แค่ค่าเดียว ถ้าใส่ค่าแบบนี้จะทำให้ค่าที่เก็บใหม่ไปทับค่าเดิม
<pre>n = int(input()) fac2name = {} for i in range(n): name, fac = input().split() fac2name[fac].add(name)</pre>	เปลี่ยนมาใช้ add เพิ่มใน value ของ dict แต่การเก็บค่า แบบนี้ยังผิดอยู่ เพราะไม่ได้ตั้งค่าเริ่มต้นของ dict ไว้ก่อน จึงทำให้ add ไม่ได้
<pre>n = int(input()) fac2name = {} for i in range(n): name, fac = input().split() if fac in fac2name: fac2name[fac].add(name) else: fac2name[fac] = {name}</pre>	แบบที่ถูกต้อง ต้องมีการตรวจสอบก่อนว่า dict ของเรา มี key นี้เก็บไว้หรือยัง ถ้ายังไม่มี ให้สร้าง set ขึ้นมาก่อน แต่ ถ้ามีแล้ว จะสามารถให้คำสั่ง add เพิ่มได้
ขั้นตอนที่ 2	
<pre>ask_fac = input().split()</pre>	รับข้อมูลรายชื่อคณะที่ต้องการถาม มาเก็บไว้ใน list ชื่อ ว่า ask_fac
ขั้นตอนที่ 3	
<pre>ans_set = {}</pre>	สร้างเซตคำตอบเริ่มต้นเป็นเซตว่าง แต่แบบนี้ผิด เพราะจะได้เป็น dict ว่างแทน
<pre>ans_set = set()</pre>	สร้างแบบนี้ถึงจะได้เซตว่างที่ถูกต้อง

โปรแกรม	คำอธิบาย
ขั้นตอนที่ 4	
<pre>for f in ask_fac: ans_set.union(fac2name[f])</pre>	นำชื่อนิสิตในคณะมาเพิ่มในเซต ans_set แบบ union แต่แบบนี้ผิด เพราะการใช้คำสั่ง union แบบนี้ไม่ได้ทำให้ค่า ของ ans_set เปลี่ยนไปเป็นอย่างไร
<pre>for f in ask_fac: ans_set = ans_set.union(fac2name[f])</pre>	การใช้คำสั่ง union แบบนี้จะทำให้ค่าใน ans_set เปลี่ยน แต่แบบนี้ยังผิดอยู่ เพราะคณะที่ถูกมา อาจจะไม่มีอยู่ใน dict ของเราก็ได้ ต้องตรวจสอบก่อน
<pre>for f in ask_fac: if f in fac2name: ans_set = ans_set.union(fac2name[f])</pre>	แบบนี้ถูกต้องแล้ว
ขั้นตอนที่ 5	
<pre>print(' '.join(ans_set.sort()))</pre>	พิมพ์ค่าใน ans_set โดยเรียงลำดับตามตัวอักษร แต่แบบนี้ผิด เพราะ sort() ใช้กับ set ไม่ได้
<pre>print(' '.join(list(ans_set).sort()))</pre>	ใช้วิธีแปลงเป็น list แล้วค่อยใช้ sort() แบบที่แสดง ทางซ้ายนี้ก็ไม่ได้ เพราะ list(ans_set).sort() เรียงลำดับได้ แต่ไม่ได้คืนค่าอะไรมากับไปให้ join
<pre>ans_list = list(ans_set) ans_list.sort() print(' '.join(ans_list))</pre>	ต้องแปลงเป็น list เก็บใส่ในตัวแปร แล้วค่อยเรียก sort กับตัวแปร แล้วส่งตัวแปรนั้นให้ join ไปใช้
<pre>print(' '.join(sorted(ans_set)))</pre>	หรือใช้ sorted แทน เพราะ sorted รับ set ได้ และให้ ผลเป็นลิสต์ที่เรียงลำดับแล้วกลับคืนมาส่งให้ join

ตัวอย่างโจทย์ปัญหา

Union & Intersection

เขียนโปรแกรมเพื่อหา union และ intersection ของเซตที่กำหนด

▶ ข้อมูลนำเข้า

บรรทัดแรก ระบุจำนวนเต็ม n แทนจำนวนเซต

น บรรทัดถัดมา ระบุเซตของจำนวนเต็มบรรทัดละ 1 เซต โดยระบุจำนวนเต็มที่อยู่ในเซต คั่นด้วยช่องว่าง

▶ ข้อมูลส่งออก

บรรทัดแรก แสดงขนาดของเซตที่เกิดจากการ union ทุกเซต

บรรทัดที่ 2 แสดงขนาดของเซตที่เกิดจากการ intersect ทุกเซต

▶ ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
3 1 2 1 2 3 1 2 1 2 1 2 3 2 5 4 3	5 2
6 100 1000 101 123 200 201 -1 -2 -3	9 0
6 -1 0 1 -1 1 0 0 -1 1 0 1 -1 1 -1 0 1 0 -1	3 3

อัកชรสองตัวหน้ากิพบนาในประเภทของคำภาษาอังกฤษ

จากข้อมูลคำศัพท์ภาษาอังกฤษเบย สาม ราชทายของคำ จงเขียนโปรแกรมเพื่อหาตัวอัកชรสองตัวแรกยอดฮิตของคำศัพท์ที่เป็นข้อมูลนำเข้า จำนวนคำศัพท์ และ รายการคำศัพท์ที่ขึ้นต้นด้วยอัកชรสองตัวแรกยอดฮิตนี้

▶ ข้อมูลนำเข้า

บรรทัดแรก บอกจำนวน รายการ ข้อมูลที่ต้องอ่านเข้ามา

บรรทัดที่เหลือ เป็นรายชื่อข้อมูล โดยข้อมูลแรกเป็นประเภทของคำศัพท์ ข้อมูลที่สองเป็นคำศัพท์ คันด้วยเครื่องหมายแท็บ

▶ ข้อมูลส่งออก

ให้พิมพ์อัកชรสองตัวหน้ายอดฮิต จำนวนคำศัพท์ และรายการคำศัพท์ที่ขึ้นต้นด้วยตัวอัកชรสองตัวหน้ายอดฮิตนี้ พร้อมประเภทของคำศัพท์ คันด้วยเว้นวรค โดยเรียงลำดับตามลำดับเดียวกันกับข้อมูลนำเข้า ถ้ามีอัកชรสองตัวที่ปรากฏบ่อยมากที่สุดเท่ากัน ให้เลือกอัកชรสองตัวแรกที่มาก่อนเรียงตามพจนานุกรม

▶ ตัวอย่าง

Input (จากแป้นพิมป์)	Output (ทางจอภาพ)
10 Adjective good Noun goose Verb wrap Verb write Noun wrinkle Noun wreck Noun wrangler Noun hall Adjective happy Noun hobby	wr 5 wrap Verb write Verb wrinkle Noun wreck Noun wrangler Noun
6 Adjective good Verb wrap Verb write Noun hall Noun hobby Noun goose	go 2 good Adjective goose Noun

การเรียนของ

ให้อ่านข้อมูลจากแป้นพิมพ์ โดยอ่านเข้ามาแล้วบันทึก แต่ละบรรทัดมีรหัสนักเรียนคั่นด้วยเว้นวรรค ตามด้วยรหัสวิชา (อาจมีมากกว่า 1 วิชา) เมื่อพบว่า รหัสนักเรียนเป็น -1 ท้ายๆ ตัวอ่าน จำนวนนักเรียนที่เรียนทั้งหมดจะอ่านจากนั้นให้อ่านรหัสวิชาสองรหัสวิชา แล้วให้แสดงจำนวนนักเรียนที่เรียนทั้งสองวิชา จำนวนนักเรียนทั้งหมดซึ่งเรียนวิชาใดวิชาหนึ่งหรือทั้งสองวิชา

▶ ข้อมูลนำเข้า

รหัสนักเรียนคั่นด้วยเว้นวรรค ตามด้วยรหัสวิชา (อาจมีมากกว่า 1 วิชา คั่นด้วยเว้นวรรค)

รับประกันว่าจะไม่มีนักเรียนที่ รหัษากัน และนักเรียน 1 คน จะไม่มีรหัสวิชาซ้ำกัน

เมื่อได้ข้อมูลนักเรียนครบถ้วนแล้ว บรรทัดต่อไปจะเป็นข้อมูล -1

บรรทัดสุดท้ายเป็นรหัสสองรหัสวิชา คั่นด้วยเว้นวรรค

▶ ข้อมูลส่งออก

มี 3 จำนวนคั่นด้วยเว้นวรรค เรียงลำดับดังนี้

จำนวนนักเรียนที่เรียนทั้งสองวิชานั้น

จำนวนนักเรียนที่เรียนวิชาใดวิชาหนึ่งเท่านั้น

จำนวนนักเรียนซึ่งเรียนวิชาใดวิชาหนึ่งหรือทั้งสองวิชา

▶ ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
001 c001 c002 c003 002 c002 c003 c004 003 c003 c005 -1 c002 c003	2 1 3
5931111121 2110101 2109101 5932222221 2109101 -1 2110101 5500101	0 1 1

▶ คำแนะนำ (ถ้ารู้วิธีทำแล้ว ไม่ต้องอ่านกีด้วย)

ควรเก็บข้อมูลด้วย dict ที่มี key คือรหัสวิชา และ value เป็นเซตของนักเรียน (ลองคิดดูว่าทำไม่) เช่น ตัวอย่างที่ 1 จะได้ dict ดังนี้

```
{  
    'c001': {'001'},  
    'c002': {'001', '002'},  
    'c003': {'001', '002', '003'},  
    'c004': {'002'},  
    'c005': {'003'}  
}
```

สรุปเนื้อหา

การใช้งานฟังก์ชัน

- อาจเรียกว่า subroutine หรือ subroutine
- เป็นการແນ່າສ່າງ ກໍາສັ່ງທີ່ໜ້າ ຖ້າ ກັນ ຢ້ອງເຂົາໃຈຢາກ ອອກມາຈາກໂປຣແກຣມຫລັກ
- ทำໄທໂປຣແກຣມອ່ານຈ່າຍ ເຂົາໃຈຈ່າຍ
- ทำໄທໂປຣແກຣມຫລັກເຮັດໃຫ້ຝັ້ງກໍາໃດ ໂດຍໄມ້ຕ້ອງເຂົ້າໃຈຢາກຄໍາສັ່ງຫລາຍຮອບ
- ຝັ້ງກໍາຄວາມມື້ນ້າທີ່ການທຳງານສັດເຈນ ເຊັ່ນ ຝັ້ງກໍາຫາຄ່າເລີ່ມຂອງຈຳນວນໃນລິສົດ ຝັ້ງກໍາກັບສຕຽງ ເປັນຕົ້ນ
- ຕ້ອງເຂົ້າໃຈຝັ້ງກໍາໄວ້ກ່ອນສ່ວນທີ່ຈະເຮັດໃຫ້

องค์ประกอบของຝັ້ງກໍາ

- ຊື່ຝັ້ງກໍາ ມີຂໍອກາມຫຼັດໜ້າມີອານຸການຕັ້ງຊື່ອຕັວແປ່
- ຄ່າທີ່ຮັບເຂົາມາ ຮ້ອງ "ພາຣາມີເຕົອຣ" (ໄມ້ຈໍາເປັນຕ້ອງມີກີ່ໄດ້)
- ກາຣຶນຄ່າຈາກຝັ້ງກໍາຕ້ວຍຄໍາສັ່ງ return (ໄມ້ຈໍາເປັນຕ້ອງມີກີ່ໄດ້)

```
def average3(x,y,z):  
    s = x+y+z  
    return s/3
```

ກາຣຶນການທຳງານຈາກຝັ້ງກໍາແລະກາຣຶນຄ່າຈາກຝັ້ງກໍາ

- ສາມາລືໃຫ້ຄໍາສັ່ງ return ໄດ້ຫລາຍທີ່ໃນຝັ້ງກໍາ
- ເນື້ອທຳຄໍາສັ່ງ return ແລ້ວ ຈະຫຼຸດການທຳງານຂອງຝັ້ງກໍາທັນທີ ແລະກັບປັບປຸງການຕ່ອ້ວລັງຈຸດທີ່ເຮັດໃຫ້ຝັ້ງກໍາ
- ຝັ້ງກໍາຄືນຄືນຄ່າໄດ້ 1 ດ້ວຍກັນ ຄ້າຕ້ອງກາຣຶນຫລາຍຄ່າ ໃຫ້ໃຊ້ tuple ເຊັ່ນ return (answer1, answer2)
- หากຄໍາສັ່ງສຸດທ້າຍຂອງຝັ້ງກໍາໄປໃໝ່ return ຮະບບະຈະເພີ່ມຄໍາສັ່ງ return (ໄມ້ຄືນຄ່າໄດ້) ທີ່ທ້າຍຝັ້ງກໍາ
- ຄໍາສັ່ງ return ທີ່ໄມ້ໄດ້ກຳຫຼັດໃຫ້ຄືນຄ່າໄດ້ ຖ້າ ຮະບບະຈະຄືນຄ່າ None (None ເປັນຄ່າປີເສີ່ງໃນຮະບບະ ໄນໃໝ່ສຕຽງ 'None')

ຕັວແປ່ໃນຝັ້ງກໍາ (local variables)

- ຕັ້ງຊື່ໜ້າກັບຕັວແປ່ໃນຝັ້ງກໍາອື່ນໄດ້ ຄື່ອວ່າເປັນຄຸນລະຕັວແປ່ກັນ
 - ເຮັດໃຫ້ຕັວແປ່ໃນຝັ້ງກໍາອື່ນໄໝໄດ້
 - หากສ່າງຕັວແປ່ປະເທດ int, float, string, boolean ເຂົ້າມາໃນຝັ້ງກໍາ ຈະຄື່ອວ່າເປັນຄຸນລະຕັວກັນ
 - หากສ່າງຕັວແປ່ປະເທດ list, dict, set ເຂົ້າມາໃນຝັ້ງກໍາ
- ຄ້າມີການແກ້ຄ່າໃນຝັ້ງກໍາ ຕັວແປ່ເປົ້າເຖິງຜູ້ເຮັດໃຫ້ຝັ້ງກໍາຈະເປັນຄ່າດ້ວຍ

ฟังก์ชันเวียนเกิด

- ฟังก์ชันเวียนเกิดมี 2 ส่วนคือ ส่วนการคำนวณแบบพื้นฐาน และส่วนที่มีการเรียกซ้ำ
- บางครั้งเขียนง่ายกว่า loop เหมือนการทำซ้ำที่ไม่รู้จำนวนรอบ
- ทำงานช้ากว่า loop และใช้หน่วยทางจำมากกว่า

การแปลงความสัมพันธ์ทาง ณิตศาสตร์เป็นฟังก์ชันเวียนเกิด

- เขียนกรณีพิเศษ ก่อน คือกรณีที่รู้คำตอบทันที ไม่มีการเรียกซ้ำ
- แล้วค่อยเขียนกรณีที่ต้องเรียกซ้ำ

$$f_n = \begin{cases} 0, & n = 0 \\ 1, & 1 \leq n \leq 2 \\ f_{n-1} + f_{n-2}, & \text{otherwise} \end{cases}$$

```
def factorial(n):
    if n < 2: return 1
    return n * factorial(n-1)
```

เรื่องผิดป oy

เขียนฟังก์ชันไว้หลังส่วนที่เรียกใช้	print(median(3,1,2)) def median(x,y,z): return (x+y+z)-min(x,y,z)-max(x,y,z)	ผิด เพราะหาฟังก์ชัน median ไม่เจอ ต้องย้ายฟังก์ชันนี้ไปไว้ก่อนบรรทัดนี้
ตั้งชื่อฟังก์ชันซ้ำกันเอง หรือซ้ำกับชื่อตัวแปร	def average(x,y): return (x+y)/2 def average(x,y,z): return (x+y+z)/3 print(average(1,2,3)) print(average(4,5)) average = 0 print(average(1,2,3))	average เป็นฟังก์ชันที่รับ 2 ตัวแปร average กลายเป็นฟังก์ชันที่รับ 3 ตัวแปร ยังเรียกได้อยู่ ผิด เพราะต้องใช้แบบ 3 พารามิเตอร์ average กลายเป็นตัวแปรแล้ว ผิด เพราะ average เป็นตัวแปรแล้ว

<p>ถ้าพารามิเตอร์ของฟังก์ชันเป็น list, set หรือ dict การเปลี่ยนแปลงข้อมูลที่เก็บในพารามิเตอร์นี้ จะส่งผลให้ข้อมูลที่ “ไป” ในตัวแปรของผู้เรียกที่ส่งมาให้พารามิเตอร์เปลี่ยนแปลงด้วย (เพราะเป็น “ตัวแปรอ้างอิง” จึงวักัน)</p> <pre> def f(x) : ... # ลิสต์ x เปลี่ยนແບບลง ลิสต์ y จะเปลี่ยน x.append(e) x[i] = e x[:] = [0,0,0] x.pop(0) ... f(y) ← </pre>	<pre> def sum_double(x): for i in range(len(x)): x[i] *= 2 # x เป็นที่เก็บเดียวกับตัวแปรที่ส่งให้ x return sum(x) y = [1, 2, 3] print(sum_double(y)) ได้ 12 print(y) ได้ [2, 4, 6] </pre>
<p>ถ้าตั้งค่าใหม่ให้กับพารามิเตอร์ของฟังก์ชันจะไม่ส่งผลถึงตัวแปรของผู้เรียกที่ส่งมาให้พารามิเตอร์</p> <pre> def f(x) : ... # ตัวแปร x เปลี่ยนค่า ลิสต์ y ไม่เปลี่ยน x = [1,2,3] ... f(y) ← </pre>	<pre> def double(x) : x *= 2 def sum_double(x): x = [2*e for e in x] # x ถูกเปลี่ยนเป็นลิสต์ตัวใหม่แล้ว return sum(x) a = 8 double(a) print(a) ได้ 8 เมื่อตอนเดิม ไม่ใช่ 16 y = [1, 2, 3] print(sum_double(y)) ได้ 12 print(y) จะได้ [1, 2, 3] </pre>
<p>ถ้า return ผลลัพธ์</p>	<pre> def square(x): x = x**2 ถ้าคืนค่า จะทำให้ได้ None def clip(x): if x > 255 : return 255 กรณี x ไม่เกิน 255 จะได้ None </pre>
<p>ถ้า () เมื่อเรียกใช้ฟังก์ชัน</p>	<pre> def read_next_positive_int(): n = int(input()) while n <= 0 : n = int(input()) return n a = read_next_positive_int ผิด ต้องมี () หลังชื่อฟังก์ชัน a = read_next_positive_int() ถูกต้อง </pre>

สั่งค่าหรือตัวแปรที่เก็บค่าผิดประเภทให้กับพารามิเตอร์ของฟังก์ชัน	<pre> def f(): n ต้องเป็นจำนวนเต็ม เพราะใช้ใน range x = 1 for k in range(n): g = 1 + 1/(1+g) return g </pre> <p>a = int(input()) b = f(a/2) ถ้า float ไป จะทำงานผิดในฟังก์ชัน f c = f(10*a) ถ้าจำนวนเต็ม ทำงานได้ถูกต้อง</p>
เล่มกรณีพื้นฐานของ recursive	<pre> def factorial(n): return n * factorial(n-1) จะเรียกฟังก์ชันวนไปเรื่อยๆ </pre>
เขียนตรวจสอบกรณีพื้นฐานไว้ทีหลัง	<pre> def factorial(n): return n * factorial(n-1) if n == 0: return 1 บรรทัดนี้ไม่เคยทำงานเลย </pre>
ถ้ามีการคำนวณค่า recursive ที่ซ้ำกัน ควรคำนวณที่เดียว แล้วเก็บไว้ในตัวแปร	<pre> def f(n): if n == 0: return 1 return f(n-1) + f(n-1)**2 </pre> <p>มีการคำนวณซ้ำ โปรแกรมจะซ้ำ</p> <pre> def f(n): if n == 0: return 1 x = f(n-1) return x + x**2 </pre> <p>คำนวณครั้งเดียว จะทำงานเร็วกว่า</p>

แบบฝึกหัด

Problem	Code
<u>ชื่อฟังก์ชัน:</u> f1 <u>Parameter:</u> รับข้อมูล a เป็นสตริง และ รับ b เป็นจำนวนเต็ม <u>Process:</u> พิมพ์ a ออกทางหน้าจอจำนวน b บรรทัด <u>Return:</u> ไม่ต้องคืนค่า	
<u>ชื่อฟังก์ชัน:</u> f2 <u>Parameter:</u> รับข้อมูล a เป็นสตริง และ รับ b เป็นจำนวนเต็ม <u>Process:</u> สร้างลิสต์ที่เก็บสตริง a จำนวน b ตัว <u>Return:</u> ลิสต์ที่สร้าง	

Problem	Code
<p><u>ชื่อฟังก์ชัน:</u> g</p> <p><u>Parameter:</u> รับจำนวนจริงสี่จำนวน m n c และ c</p> <p><u>Return:</u> คืนค่าจุดตัดของเส้นตรง $y = mx+b$ และ $y = nx+c$ เป็น tuple ของจุดตัด (x,y)</p> <p>หากเป็นเส้นตรงที่ขนานกัน ให้คืนค่าจำนวนเต็ม 1</p> <p>หากเป็นเส้นตรงเดียวกัน ให้คืนค่าจำนวนเต็ม 2</p>	
<p><u>ชื่อฟังก์ชัน:</u> h</p> <p><u>Parameter:</u> รับลิสต์ของจำนวนเต็ม x</p> <p><u>Return:</u> คืนลิสต์ใหม่ ที่มีสมาชิกจาก x เฉพาะที่เป็นเลขคู่เท่านั้น โดยห้ามแก้ไขค่าในลิสต์ x</p>	
<p>เขียนฟังก์ชันเรียนเกิดเพื่อคำนวณค่าดังนี้</p> $a_n = \begin{cases} 1, & n = 0 \\ -2, & n = 1 \\ a_{n-2} \times n, & \text{otherwise} \end{cases}$	
<p>เขียนฟังก์ชันเรียนเกิดเพื่อคำนวณค่าดังนี้</p> $\begin{aligned} k(2n) &= k(n) + (k(n)\%10) \quad \text{if } n > 0 \\ k(2n+1) &= k(n-1)*n \quad \text{if } n > 0 \\ k(0) &= 1 \\ k(1) &= 2 \end{aligned}$	
<p>เขียนฟังก์ชันเรียนเกิดเพื่อคำนวณค่าดังนี้</p> $\begin{aligned} s_{i,k} &= \begin{cases} 0, & i \geq k \\ k + t_{i+1,k}, & i < k \end{cases} \\ t_{j,k} &= \begin{cases} 0, & j \geq k \\ j + s_{j,k-1}, & j < k \end{cases} \end{aligned}$	
<p><u>ชื่อฟังก์ชัน:</u> is_palindrome</p> <p><u>Parameter:</u> รับข้อมูล s เป็นสตริง</p> <p><u>Process:</u> เขียนฟังก์ชันเรียนเกิดเพื่อตรวจสอบว่า s เป็น palindrome (อ่านจากหน้าไปหลังเหมือนหลังไปหน้า) หรือไม่</p> <p><u>Return:</u> ถ้า s เป็น palindrome ให้คืนค่า True ถ้าไม่เป็น ให้คืนค่า False</p>	

ตัวอย่างการแก้ไข源ปัญหา

Function Refactoring

กำหนดโปรแกรมคำนวณจำนวนวันที่ตั้งแต่วันที่ A จนถึงวันที่ B มาให้ จงแยกเป็นฟังก์ชันต่อไปนี้ และเติมคำสั่งให้ครบสมบูรณ์

เส้นประ ฟังก์ชัน `is_leap_year(y)` คืน `True` เมื่อปีคริสตศักราช `y` เป็นปีอิจสุหริบิน
ถ้าไม่เป็นให้คืน `False`

เส้นไข่ปลา ฟังก์ชัน `day_of_year(d,m,y)` คืนว่าวันที่ `d m y` เป็นวันที่ลำดับที่เท่าใดของปีคริสตศักราช `y`

เส้นประยາ ฟังก์ชัน `days_in_year(y)` คืนจำนวนวันในปีคริสตศักราช `y`

► ข้อมูลนำเข้า

มี 2 บรรทัด บรรทัดแรกเป็นวันที่ A ในรูปแบบ `d1 m1 y1` เป็นปีคริสตศักราช

บรรทัดที่ 2 เป็นวันที่ B ในรูปแบบ `d2 m2 y2` เป็นปีคริสตศักราช

รับประทานว่าวันที่หั้งสองวันจะเป็นวันที่ที่ถูกต้อง และวันที่ A จะมาก่อนวันที่ B เสมอ และอยู่คนละปีกัน

► ข้อมูลส่งออก

จำนวนวันระหว่างวันที่ A และ B โดยรวมวันที่ A และ B ด้วย

► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
1 1 2018 1 1 2020	731
25 12 1999 9 3 2000	76

โปรแกรมเริ่มต้น

```
d1,m1,y1 = [int(e) for e in input().split()]
d2,m2,y2 = [int(e) for e in input().split()]

# คำนวณว่าวันที่ A เป็นวันที่เท่าไรของปี
D = [31,28,31,30,31,30,31,31,30,31,30,31]
if y1%4==0 and (y1%100!=0 or y1%400==0):      # ตรวจสอบปีอธิกสุรทิน
    D[1] += 1
A = 0
for i in range(m1-1):
    A += D[i]
A += d1

# คำนวณว่าวันที่ B เป็นวันที่เท่าไรของปี
D = [31,28,31,30,31,30,31,31,30,31,30,31]
if y2%4==0 and (y2%100!=0 or y2%400==0):      # ตรวจสอบปีอธิกสุรทิน
    D[1] += 1
B = 0
for i in range(m2-1):
    B += D[i]
B += d2

# คำนวณจำนวนวันระหว่างปี A และปี B
C = 0
for i in range(y1+1,y2):
    # หาจำนวนวันในปีนั้น ๆ
    if i%4==0 and (i%100!=0 or i%400==0):      # ตรวจสอบปีอธิกสุรทิน
        C += 366
    else:
        C += 365

print( ?? )

# [จำนวนวันในปี A] - [(ลำดับที่ของวัน A)] + 1 + (จำนวนวันระหว่างปีทั้งสอง) + [(ลำดับที่ของวัน B)]
```

ตัวอย่างการเขียนโปรแกรม

โปรแกรม	คำอธิบาย
ฟังก์ชัน is_leap_year จากโปรแกรมในกล่องสีน้ำเงิน เป็นฟังก์ชันได้ดังนี้ <pre>def is_leap_year(y): if y%4==0 and (y%100!=0 or y%400==0): return True else: return False</pre>	ฟังก์ชัน is_leap_year จะตรวจสอบว่าเป็นปีอิกซูรทินหรือไม่ ตามเงื่อนไขในกล่องสีน้ำเงินตามเงื่อนไขให้คืนค่า True แต่ถ้าไม่ตรงให้คืนค่า False
หรือสามารถเขียนให้สั้นลงได้เป็นแบบนี้ <pre>def is_leap_year(y): if y%4==0 and (y%100!=0 or y%400==0): return True return False</pre>	ปรับปรุง : ในกรณีนี้ เราสามารถตัดคำว่า else ออกได้ เนื่องจากถ้าเงื่อนไขเป็นจริงแล้วจะ return True และจากการทำงานฟังก์ชันทันที จะไม่มาทำงานที่บรรทัด return False อีก
หรือสามารถเขียนให้สั้นลงได้อีกเป็นแบบนี้ <pre>def is_leap_year(y): return y%4==0 and (y%100!=0 or y%400==0)</pre>	ปรับปรุง : เนื่องจากประยุกต์ตรวจสอบเงื่อนไข มีค่าความจริงเป็น True หรือ False อญญาติ เราสามารถคืนผลการเปรียบเทียบนี้โดยตรงได้เลย
ฟังก์ชัน day_of_year จากโปรแกรมในกล่องสีน้ำเงิน สามารถเขียนเป็นฟังก์ชันได้ดังนี้ <pre>def day_of_year(d,m,y): D = [31,28,31,30,31,30,31,31,30,31,30,31] if y%4==0 and (y%100!=0 or y%400==0): D[1] += 1 A = 0 for i in range(m-1) A += D[i] A += d</pre>	คัดลอกโปรแกรมจากกล่องสีน้ำเงินมา อย่าลืมเปลี่ยนชื่อตัวแปรด้วย (d,m,y เป็น d,m,y) แต่ฟังก์ชันนี้ยังไม่ถูกต้อง เพราะลืมคืนค่า
<pre>def day_of_year(d,m,y): D = [31,28,31,30,31,30,31,31,30,31,30,31] if y%4==0 and (y%100!=0 or y%400==0): D[1] += 1 A = 0 for i in range(m-1) A += D[i] A += d return A</pre>	เติมคำสั่ง return A ทำให้ฟังก์ชันทำงานได้ถูกต้องแล้ว

โปรแกรม	คำอธิบาย
<pre>def day_of_year(d,m,y): D = [31,28,31,30,31,30,31,31,30,31,30,31] if is_leap_year(y): D[1] += 1 A = 0 for i in range(m-1) A += D[i] A += d return A</pre> <p>ฟังก์ชัน days_of_year(y)</p>	ปรับปรุง: นำฟังก์ชัน is_leap_year ที่เขียนไว้แล้ว มาใช้ได้
<p>จากโปรแกรมในกล่องเด็นประยุวฯ สามารถเขียนเป็นฟังก์ชันได้ดังนี้</p> <pre>def days_in_year(y): if y%4==0 and (y%100!=0 or y%400==0): return 366 else: return 365</pre>	คัดลอกโปรแกรมจากกล่องเด็นประยุวฯ อย่าลืมเปลี่ยนชื่อตัวแปร และคืนค่าให้ถูกต้อง
<pre>def days_in_year(y): if is_leap_year(y): return 366 else: return 365</pre>	ปรับปรุง: นำฟังก์ชัน is_leap_year ที่เขียนไว้แล้ว มาใช้
<pre>def days_in_year(y): return day_of_year(31,12,y)</pre>	ปรับปรุง: นำฟังก์ชัน day_of_year ที่เขียนไว้แล้ว มาใช้ โดยขอคำดับของวันที่ 31 ธันวาคม ปี y (กรณีนี้ได้โปรแกรมสั้นลง แต่อาจทำให้ฟังก์ชัน เข้าใจยากขึ้น)
ส่วนโปรแกรมหลัก	
<pre>d1,m1,y1 = [int(e) for e in input().split()] d2,m2,y2 = [int(e) for e in input().split()] A = day_of_year(d1,m1,y1) B = day_of_year(d2,m2,y2) C = 0 for i in range(y1+1,y2): C += days_in_year(i) print(days_in_year(y1) - A + 1 + C + B)</pre>	เปลี่ยนกล่องต่าง ๆ เป็นการเรียกฟังก์ชันและส่งค่าไปให้แต่ละฟังก์ชันให้ถูกต้อง

ตัวอย่างโจทย์ปัญหา

Four Functions

จะเขียน 4 ฟังก์ชัน ให้สามารถตามที่เขียนอธิบายกำกับแต่ละฟังก์ชัน ในโครงข้องโปรแกรมข้างล่างนี้

```
def make_int_list(x):
    # รับสตริง x มาแยกและแปลงเป็น int เก็บใน list และคืนเป็นผลลัพธ์
    # เช่น x = '12 34 5' ได้ผลเป็น [12, 34, 5]

def is_odd(e):
    # คืนค่าจริงเมื่อ e เป็นจำนวนคี่ ถ้าไม่ใช่ คืนค่าเท็จ

def odd_list(alist):
    # คืน list ที่มีค่าเท่านั้นใน alist แต่มีเฉพาะตัวที่เป็นจำนวนคี่
    # เช่น alist = [10, 11, 13, 24, 25] จะได้ [11, 13, 25]

def sum_square(alist):
    # คืนผลรวมของกำลังสองของแต่ละค่าใน alist
    # เช่น alist = [1,3,4] ได้ผลเป็น (1*1 + 3*3 + 4*4) = 26

exec(input().strip()) # ต้องมีบรรทัดนี้เมื่อส่งไป Grader
```

► ข้อมูลนำเข้า

คำสั่งในการทดสอบฟังก์ชันที่เขียน

► ข้อมูลส่งออก

ผลที่ได้จากคำสั่งที่ป้อนเป็นข้อมูลนำเข้า

► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
print(make_int_list('1 2 3 4 5'))	[1, 2, 3, 4, 5]
print(is_odd(2222))	False
print(odd_list([1,2,3,4,5,6,7]))	[1, 3, 5, 7]
print(sum_square([1,1,2,3]))	15

Recursive C(n,k)

เราสามารถหาค่าของจำนวนวิธีนั้นกันได้โดย ซึ่งเปลี่ยน $C(n,k)$ ได้จากการด้านล่าง จงใช้สมการต่อไปนี้ในการเขียนโปรแกรมแบบ recursive เพื่อคำนวณค่า $C(n,k)$

$$C(n, k) = \begin{cases} C(n - 1, k) + C(n - 1, k - 1), & \text{if } 0 < k < n \\ 1, & \text{if } k = 0 \text{ or } n = k \\ 0, & \text{otherwise} \end{cases}$$

► ข้อมูลนำเข้า

มี 2 บรรทัด ประกอบด้วยจำนวนเต็ม n และ k

► ข้อมูลส่งออก

มี 1 บรรทัด แสดงค่า $C(n,k)$ ที่คำนวณได้

► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
6 2	15
10 8	45
0 1	0
3 7	0
10 10	1

Recursive SumList

โจทย์ข้อนี้สั้น ๆ จะเขียนฟังก์ชัน รูปแบบ `def sumlist(x)` ของโครงสร้างข้างล่างนี้ `sumlist` รับ `x` เป็นลิสต์เก็บจำนวนเต็ม แล้วคืนผลรวมของจำนวนเต็มทุกตัวใน `x` โดย `x` เป็นลิสต์ที่ภายในเป็นลิสต์ซ้อนลิสต์กี่ชั้นก็ได้ ดังตัวอย่างที่แสดงข้างล่างนี้

หมายเหตุ : สามารถใช้คำสั่ง `if type(x) == list` เพื่อทดสอบว่า `x` เป็นข้อมูลประเภทลิสต์หรือไม่

```
def sumlist( x ):  
    # ???  
  
    print(eval(input().strip())) # do not remove this line
```

► ข้อมูลนำเข้า

ลิสต์ที่เก็บจำนวนเต็ม (อาจเป็นลิสต์แบบบลิสต์ซ้อนลิสต์กี่ชั้นก็ได้)

► ข้อมูลสองออก

ผลรวมของจำนวนเต็มทุกตัวในลิสต์ที่เป็นข้อมูลนำเข้า

► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
<code>sumlist([] ,[])</code>	0
<code>sumlist([1,1,2,2])</code>	6
<code>sumlist([1,[1],[2,[2],0,[0,0]]])</code>	6
<code>sumlist([0,[1,[2,[3,[4,5],6],7],8],[9,10]])</code>	55

`import numpy as np`

สรุปเนื้อหา

- NumPy เป็นคลังคำสั่งที่หัวริการามากมายเกี่ยวกับการคำนวณทางวิทยาศาสตร์ โดยมีที่เก็บข้อมูลเรียกว่า อาเรย์ n มิติ (ndarray) มีไว้เก็บข้อมูลเพื่อการประมวลผลที่มีประสิทธิภาพมาก ๆ
- อาเรย์มีลักษณะคล้ายลิสต์ แต่สร้างแล้วเปลี่ยนขนาดไม่ได้
- ค่าในอาเรย์ทุกช่องต้องเป็นประเภทเดียวกันทั้งหมด เช่น เป็น int ทุกช่อง หรือ float ทุกช่อง (ผูกกันไม่ได้ ถ้าเป็นลิสต์ผสมได้)
- เราสร้าง เวกเตอร์ ได้ด้วยอาเรย์ 1 มิติ และสร้าง เมทริกซ์ ได้ด้วยอาเรย์ 2 มิติ

การสร้างอาเรย์ด้วยฟังก์ชัน

สร้างอาเรย์จากลิสต์ L	<code>np.array(L)</code>	ขนาดของอาเรย์ M	<code>M.shape</code>
[1 0 2 3 -1]	<code>np.array([1,0,2,3,-1])</code>	[[2 3] [4 1] [7 5]]	<code>np.array([[2,3],[4,1],[7,5]])</code>
[[0. 0. 0.] [0. 0. 0.]]	<code>np.zeros((2,3))</code>	[[1 1] [1 1] [1 1]]	<code>np.ones((3,2),int)</code>
[[1 0 0 0] [0 1 0 0] [0 0 1 0] [0 0 0 1]]	<code>np.identity(4,int)</code>	[[1. 0. 0.] [0. 1. 0.] [0. 0. 1.]]	<code>np.identity(3)</code>
สร้างอาเรย์ที่มีขนาดเท่ากับอาเรย์ x และเป็น 0 (int) ทุกช่อง	<code>np.zeros_like(x,int)</code> (x อาจเก็บ int หรือ float ก็ได้)	สร้างอาเรย์ที่มีขนาดเท่ากับอาเรย์ y และเป็น 1.0 ทุกช่อง	<code>np.ones_like(y,float)</code> (y อาจเก็บ int หรือ float ก็ได้)
[4 5 6 7 8 9]	<code>np.arange(4,10)</code>	[8. 7. 6. 5.]	<code>np.arange(8.0,4.0,-1.0)</code> <code>np.arange(8,4,-1,float)</code>
[2.3 2.33 2.36 2.39 2.42 2.45 2.48]			<code>np.arange(2.3,2.5,0.03)</code>

Element-wise operations, Broadcasting และ Indexing

- เป็นสามแนวคิดของ NumPy (ที่ต้องรู้ในวิชาฯ) ที่ช่วยให้การประมวลผลอาเรย์ทำได้สะดวก
 - การดำเนินการอาเรย์กับอาเรย์ (เช่น A+B) ก็ดำเนินการหรือทำฟังก์ชันกับอาเรย์ จะเป็นแบบซองต่อซอง (element-wise) เช่น
$$\begin{aligned} \text{np.array([1, 2, 3])} + \text{np.array([1, 1, 1])} &\rightarrow \text{np.array([2, 3, 4])} \\ 1 / (\text{np.array([1, 2, 4]))} &\rightarrow \text{np.array([1.0, 0.5, 0.25])} \end{aligned}$$
 - การดำเนินการอาเรย์กับอาเรย์ที่มีขนาดไม่เท่ากัน อาจมีการขยายอาเรย์ให้มีขนาดเท่ากันก่อน (broadcasting)
โดยพิจารณา mimic ของทั้งสองอาเรย์จากขวาไปซ้าย ให้เป็นไปตามกฎการ broadcast (ขอนำเสนอด้วยตัวอย่าง)
 - $A.shape = (2, 3)$, $B.shape = (3,)$ ดูจากวามาซ้าย 3 เท่ากัน จึง broadcast B ให้มีขนาดเท่ากับ A ได้ เช่น $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} + \begin{bmatrix} 9 & 8 & 7 \end{bmatrix}$ broadcast แล้วกลายเป็น $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} + \begin{bmatrix} 9 & 8 & 7 \end{bmatrix} = \begin{bmatrix} 10 & 10 & 10 \\ 4 & 5 & 6 \end{bmatrix} \quad [9 & 8 & 7] \quad [13 & 13 & 13]$
 - มิติไหนขนาดเป็น 1 ถือว่า มิตินั้น broadcast ได้ (แต่มิติอื่นที่ไม่ใช่ 1 ต้องเท่ากัน)
 - $A.shape = (2, 3)$, $B.shape = (2, 1)$ broadcast B ให้เป็น $(2, 3)$ มีขนาดเท่ากับ A
 - $A.shape = (1, 3)$, $B.shape = (2, 1)$ broadcast ทั้ง A และ B ให้เป็น $(2, 3)$
เช่น $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} + \begin{bmatrix} 6 \end{bmatrix}$ broadcast แล้วกลายเป็น $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} + \begin{bmatrix} 6 & 6 & 6 \end{bmatrix} = \begin{bmatrix} 7 & 8 & 9 \\ 1 & 2 & 3 \end{bmatrix} \quad [7 & 7 & 7] \quad [8 & 9 & 10]$
 - $A.shape = (2, 3)$, $B.shape = (1,)$ broadcast B ได้ (ถ้า B เป็นสเกลาร์ ให้ถือว่ามีมิติเป็น $(1,)$)
เช่น $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} + 9$ broadcast แล้วกลายเป็น $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} + \begin{bmatrix} 9 & 9 & 9 \end{bmatrix} = \begin{bmatrix} 10 & 11 & 12 \\ 4 & 5 & 6 \end{bmatrix} \quad [9 & 9 & 9] \quad [13 & 14 & 15]$
 - $A.shape = (3, 5)$, $B.shape = (5, 1)$ broadcast B ไม่ได้ (ปรับ 1 ได้ แต่ก็มิติไม่เท่ากัน เลยทำไม่ได้)
 - การอ้างอิงข้อมูลในอาเรย์ (Indexing ของ NumPy)
 - การอ้างอิงข้อมูลในอาเรย์ 1 มิติ
 - การเข้าใช้ข้อมูล
 - $V[k]$ เหมือนการใช้ลิสต์ คือเลือกข้อมูลหนึ่งช่อง โดยที่ k เป็นจำนวนเต็ม
 - $V[a:b:c]$ เหมือนการใช้ลิสต์ คือเลือกข้อมูลเป็นช่วง ได้ผลเป็นอาเรย์
 - การใส่ค่าใหม่
 - $V[k] = 99$ เหมือนกับการใช้ลิสต์
 - $V[a:b:c] = d$ แตกต่างจากลิสต์
 - ถ้า d เป็น int , float หรือลิสต์/อาเรย์ขนาด 1 ช่อง จะ broadcast ให้มีขนาดเท่ากับช่วงของ $a:b:c$
 - ถ้า d เป็นลิสต์/อาเรย์ขนาดมากกว่า 1 ช่อง ต้องมีขนาดเท่ากับขนาดของช่วง $a:b:c$ เช่น
 - $V = \text{np.zeros}(5, \text{int})$
 - $V[2:4] = 1$ ทำได้, V เปลี่ยนเป็น $[0 \ 0 \ 1 \ 1 \ 0]$
 - $V[::2] = 3$ ทำได้, V เปลี่ยนเป็น $[3 \ 0 \ 3 \ 1 \ 3]$
 - $V[::2] = [9]$ ทำได้, V เปลี่ยนเป็น $[9 \ 0 \ 9 \ 1 \ 9]$
 - $V[::2] = [8, 8, 8]$ ทำได้, V เปลี่ยนเป็น $[8 \ 0 \ 8 \ 1 \ 8]$
 - $V[1:5] = [1, 2]$ ทำไม่ได้, เพราะขนาดของข้อมูลไม่ตรงกัน
 - $V[7:7] = [1, 2]$ ไม่เกิดการเปลี่ยนแปลงใด ๆ

- การอ้างอิงข้อมูลในอาร์เรย์ 2 มิติ

- สามารถอ้างอิงถึงตัวข้อมูล อาร์เรย์ 1 มิติ ที่แทนแถว (row) ของข้อมูล อาร์เรย์ 1 มิติที่แทนหลัก (column) ของข้อมูล หรืออาร์เรย์อยู่ 2 มิติที่ “ทันท่วง” ของแถวและหลัก ได้หลากหลายแบบ ดังนี้

$M[r, c]$	ข้อมูล ณ แควร์ที่ r หลักที่ c
$M[r, :]$ หรือ $M[:, r]$	อาร์เรย์ 1 มิติของแถวที่ r ทั้งหมด
$M[:, c]$	อาร์เรย์ 1 มิติของหลักที่ c ทั้งหมด (ไม่ใช้อาร์เรย์ 2 มิติที่มี 1 หลัก)
$“[r1:r2:c]”$ หรือ $M[r1:r2]$	อาร์เรย์ 2 มิติประกอบด้วย แควร์ที่ $r1$ ถึง $r2-1$
$M[:, c1:c2]$	อาร์เรย์ 2 มิติประกอบด้วย หลักที่ $c1$ ถึง $c2-1$
$M[r1:r2, c1:c2]$	อาร์เรย์ 2 มิติ ในช่วงแควร์ที่ $r1$ ถึง $r2-1$ และช่วงหลักที่ $c1$ ถึง $c2-1$
$M[r1:r2:a, c1:c2:b]$	อาร์เรย์ 2 มิติ ในช่วงแควร์และหลักที่กำหนดโดย $r1:r2:a$, $c1:c2:b$
เช่น ให้ $M = np.array([[1,2,3,4],[5,6,7,8],[9,10,11,12]])$	
$M[1:3]$	ได้ $np.array([[5,6,7,8],[9,10,11,12]])$
$M[1:2, 1:3]$	ได้ $np.array([[6,7]])$
$M[:, 2]$	ได้ $np.array([3,7,11])$
$M[1:, :2]$	ได้ $np.array([[5,6],[9,10]])$
$M[0:3:2]$	ได้ $np.array([[1,2,3,4],[9,10,11,12]])$
$M[0:3:2, 1:4:2]$	ได้ $np.array([[2,4],[10,12]])$
$M[::-1]$	ได้ $np.array([[9,10,11,12],[5,6,7,8],[1,2,3,4]])$

- สามารถให้ค่ากับหลาย ๆ ช่องในอาร์เรย์พร้อมกันได้ เช่น

$M[2:4, 1:4] = [[-1, 1, -1], [-1, 1, -1]]$	ขนาดเท่ากันพอดี
$M[2:4, 1:4] = [-1, 1, -1]$	ขนาดไม่เท่า ระบบ broadcast ให้ M ต้องมี 4 แควร์
$M[:, 0] = [1, 2, 3, 4]$	
$M[::2, ::3] = 0$	broadcast 0

คำสั่ง dot (ใช้ว่า $x.dot(y)$ หรือ $np.dot(x,y)$ ก็ได้)

dot ใช้กับเวกเตอร์ คือการหา dot product

```
x = np.array([ 1, 2, 3, 4])
y = np.array([ 0, -1, 1, 2])
x.dot(y) #ได้ค่าเท่ากับ y.dot(x) เท่ากับ
1*0 + 2*(-1) + 3*1 + 4*2 = 9
```

dot ใช้กับเมทริกซ์ คล้ายกับการคูณเมทริกซ์

```
x = np.array([[1,2],[3,4],[5,6]]) x.shape คือ (3,2)
y = np.array([[2],[3]]) y.shape คือ (2,1)
z = np.array([[2,3]]) z.shape คือ (1,2)
w = np.array([-2,3]) w.shape คือ (2,)
```

$x.dot(y)$ ได้ $np.array([[4],[6],[8]])$
 มีมิติเป็น (3,2) กับ (2,1) คูณได้ ได้อาร์เรย์ขนาด (3,1)
 $y.dot(x)$ ทำไม่ได้ เพราะมิติไม่ถูกต้อง (2,1) กับ (3,2)
 $x.dot(z)$ ทำไม่ได้ เพราะมิติไม่ถูกต้อง (3,2) กับ (1,2)
 $x.dot(w)$ ได้ $np.array([4,6,8])$
 อันนี้แปลก (3,2) กับ (2,) คูณได้ ได้อาร์เรย์ขนาด (3,)

ฟังก์ชันที่ระบุแกนได้, การ transpose และการเปลี่ยนเที่ยบ

($M = \text{np.array}([[1,2,3,4],[5,6,7,8]])$)

- ฟังก์ชันที่ระบุแกนได้ เช่น sum, max, min, mean, std, argmax, argmin (คืนตำแหน่งตัวมาก/น้อยสุด)
- $\text{np.sum}(M)$ ได้ผลรวมของทุกช่อง $\text{np.sum}(M)$ ได้ 36
- $\text{np.sum}(M, \text{axis}=0)$ ได้ผลรวมตามแนวตั้ง $\text{np.sum}(M, \text{axis}=0)$ ได้ $\text{np.array}([6,8,10,12])$
- $\text{np.sum}(M, \text{axis}=1)$ ได้ผลรวมตามแนวนอน $\text{np.sum}(M, \text{axis}=1)$ ได้ $\text{np.array}([10,26])$
- matrix transpose ชี้ว่า M. $M.T$ ได้ $\text{np.array}([[1,5],[2,6],[3,7],[4,8]])$
- $M > 3$ ได้ $\text{array}([[\text{True},\text{False},\text{False},\text{True}],[\text{True},\text{True},\text{True},\text{True}]]$)
- ใช้ np.sum นับจำนวน element ที่ตรงตามเงื่อนไขได้ $\text{np.sum}(M > 3)$ ได้ 5
- ใช้เงื่อนไขในการเลือกบาง element ของอาเรย์ได้ $M[M \% 2 == 0]$ ได้ $\text{np.array}([2,4,6,8])$

เรื่องผิดบอย

ลีม import numpy	โดยทั่วไปมักใช้ <code>import numpy as np</code> คือการ import คลังคำสั่ง numpy และตั้งชื่อใหม่ว่า np จะได้เขียนสั้น ๆ
ลีมระบุประเภทของข้อมูลที่จะเก็บว่าเป็น int หรือ float	<code>x = np.zeros((3,4))</code> # ได้เป็น float <code>x = np.zeros((3,4),int)</code> # ได้เป็น int <code>x = np.zeros_like(y)</code> # ขึ้นกับประเภทข้อมูลของ y
ถ้าเก็บค่า float ในเมทริกซ์ที่เก็บ int ระบบจะปัดค่าหลังจุดศูนย์มิถึง	<code>x = np.array([1,2,3],int)</code> <code>x[0] = 4.8</code> # ได้ x เท่ากับ <code>np.array([4,2,3])</code>
ใช้ <code>np.array</code> กับ <code>np.ndarray</code> ผิด	ต้องการสร้างเมทริกซ์ขนาด 4×5 แต่เขียน <code>a = np.array((4,5))</code> # ได้ [4,5] ต้องการสร้างเวกเตอร์ $[2,3]$ แต่เขียน <code>a = np.ndarray([2,3])</code> # ได้เมทริกซ์ขนาด 2×3
การทำงานของเวกเตอร์ไม่เหมือนเมทริกซ์ (อาเรย์ 1 มิติ vs 2 มิติ)	<code>x = np.array([1,2,3])</code> ได้อารей 1 มิติ 3 ช่อง <code>y = np.array([[1,2,3]])</code> ได้อารей 2 มิติ 1 แถว 3 คอลัมน์ <code>x.shape</code> ได้ $(3,)$ <code>y.shape</code> ได้ $(1,3)$ <code>x.T</code> ได้อารейเหมือนกับ x <code>y.T</code> ได้อารей 3 แถว 1 คอลัมน์ <code>print(x.dot(x.T))</code> <code>print(x.T.dot(x))</code> <code>print(y.dot(y.T))</code> <code>print(y.T.dot(y))</code> ได้ $\text{np.array}([[1,2,3],[2,4,6],[3,6,9]])$

broadcasting
& element-wise operations

```
x = np.array([[1,2,3],[4,5,6]])
y = x+2                                # [[3,4,5],[6,7,8]]
` = x+[1,2,3]                          # [[2,4,6],[5,7,9]]
y = x+np.array([1,2,3])                  # [[2,4,6],[5,7,9]]
y = x+np.array([[1,2,3]])                # [[2,4,6],[5,7,9]]
y = x+np.array([[1],[2],[3]]).T          # [[2,4,6],[5,7,9]]
y = x+np.array([[1],[2],[3]])            # ผิด
y = x+np.array([[1,2,3]]).T              # ผิด
y = x+np.array([[1,2]]).T                # [[2,3,4],[6,7,8]]
```

แบบฝึกหัด

(พยายามเขียนให้สั้น ๆ และทำงานได้เร็วสุด ๆ)

Problem	Code
สมมติว่ามีอาเรย์ 2 มิติ M มาให้ Input: จำนวนเต็มบวก 1 จำนวนจากแป้นพิมพ์ เก็บใน k Process: เปลี่ยนค่าของซ่องใน M ที่หมายเลขແກວและหลัก หารด้วย k ลงตัวให้มีค่าเป็น 0	
สมมติว่ามีอาเรย์ 2 มิติ M มาให้ Input: จำนวนเต็มบวก 1 จำนวนจากแป้นพิมพ์ เก็บใน k Process: เปลี่ยนค่าของซ่องใน M ที่หมายเลขແກວและหลัก หารด้วย k ลงตัวให้มีค่าเป็น 2 เท่าของค่าเดิม	
สมมติว่ามีอาเรย์ 2 มิติ M มาให้ ให้คำนวนหาผลต่างของค่ามากสุดและค่าน้อยสุดในแต่ละหลัก เช่น ถ้า M = np.array([[3,2],[5,6],[7,1]]) จะได้คำตอบ A เท่ากับ np.array([4,5]) (4 มาจาก 7-3 และ 5 มาจาก 6-1)	
กำหนดอาเรย์ X เก็บความยาวของด้านประกอบมุมจากของ รูปสามเหลี่ยมนุ่มจากหลายรูป เช่น X = np.array([[3,4],[5,12],[24,7]]) ให้สร้างอาเรย์ Y ซึ่งเก็บความยาวของด้านตรงข้ามมุมจาก เช่น จาก X ด้านบน จะได้ Y เท่ากับ array([5.,13.,25.])	
Input: จำนวนเต็มบวก 1 จำนวนจากแป้นพิมพ์ เก็บใน k Process: สร้างเมทริกซ์ขนาด $k \times k$ ซึ่งเก็บค่า 0 และ 1 เป็น ตารางมากรุก (มุมชี้ยบนเป็น 0) เช่น ถ้า $k = 5$ จะได้ C = np.array([[0,1,0,1,0] [1,0,1,0,1] [0,1,0,1,0] [1,0,1,0,1] [0,1,0,1,0]])	

Problem	Code
<p><u>Input:</u> จำนวนเต็มบวก 1 จำนวนจากแป้นพิมพ์ เว็บไซต์ k</p> <p><u>Process:</u> สร้างเมทริกซ์ขนาด $k \times k$ ซึ่งเก็บค่าเป็น</p> <p>ตารางหมากลูกอิฐแบบหนึ่ง (มุมซ้ายบนเป็น 1) โดยแทน 1 ด้วย เลขແຄວ (ให้เลขແຄວเริ่มที่ 1) เช่น ถ้า $k = 5$ จะได้</p> <pre>C = np.array([[1,0,1,0,1] [0,1,0,2,1] [1,0,2,0,1] [0,4,0,4,0] [5,0,5,0,5]])</pre>	



Weighted Score

รายการประมวลผลกำลังหาผู้ชนะจากการแข่งขัน โดยคะแนนของผู้เข้าแข่งขันมาจากการคะแนนรวมของผู้เข้าแข่งขันจาก 3 ส่วน คือ คะแนนของกรรมการ คะแนนจากผู้ชมในห้องส่ง และคะแนนจากผู้ชมทางบ้าน หากรายการได้กำหนดนำหนักของคะแนนแต่ละส่วนมาให้แล้ว ให้คำนวณคะแนนรวมของผู้เข้าแข่งขันแต่ละคน

► ข้อมูลนำเข้า

บรรทัดแรกคือจำนวนผู้เข้าแข่งขัน n เป็นจำนวนเต็ม

บรรทัดถัดมา รับจำนวนเต็ม 3 จำนวน คือคะแนนของกรรมการ คะแนนจากผู้ชมในห้องส่ง และคะแนนจากผู้ชมทางบ้าน บรรทัดสุดท้ายคืออัตน้ำหนักของคะแนนแต่ละส่วน เป็นจำนวนทศนิยม 3 จำนวน

► ข้อมูลส่งออก

มี n บรรทัด แต่ละบรรทัดแสดงคะแนนของผู้เข้าแข่งขันแต่ละคน

► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
4 10 15 10 20 5 15 14 8 7 12 12 12 0.5 0.25 0.25	11.25 15.0 10.75 12.0
2 10 30 20 20 10 30 0.6 0.3 0.1	17.0 18.0

ตัวอย่างการเขียนโปรแกรม

โปรแกรม	คำอธิบาย
<pre> n = int(input()) S = [] for i in range(n): S.append([int(e) for e in input().split()]) W = [float(e) for e in input().split()] for i in range(n): score = [S[i][j]*W[j] for j in range(3)] print(sum(score)) </pre>	<p>โปรแกรมส่วนบนเป็นส่วนสำหรับรับข้อมูล ได้ S เป็นลิสต์ช้อนลิสต์ของคะแนนผู้เข้าแข่งขัน แต่ละคน และ W เป็นลิสต์ของน้ำหนัก เช่น $S = [[10, 30, 20], [20, 10, 30]]$ $W = [0.6, 0.3, 0.1]$</p> <p>โปรแกรมด้านขวาทำงานถูกต้อง แต่ถ้ามีจำนวน ข้อมูลมาก จะช้า เพราะใช้ลิสต์ในการประมวลผล</p>
<pre> n = int(input()) S = [] for i in range(n): S.append([int(e) for e in input().split()]) W = [float(e) for e in input().split()] score = [sum([S[i][j]*W[j] \ for j in range(3)]) \ for i in range(n)] for i in score: print(i) </pre>	<p>ยุบ for วงล่างให้เป็น list comprehension ทำงานถูกต้อง แต่ก็ยังช้าอยู่ ลองเปลี่ยนมาใช้ numpy</p>
<pre> n = int(input()) S = [] for i in range(n): S.append([int(e) for e in input().split()]) W = [float(e) for e in input().split()] S = np.array(S) W = np.array(W) for i in range(n): print(sum(S[i]*W)) </pre>	<p>เปลี่ยนลิสต์ S และ W มาใช้ numpy array สั่ง run, ใส่ข้อมูลตามตัวอย่างแรก, ผิดที่บรรทัดที่ 7 NameError: name 'np' is not defined แปลว่า ระบบไม่รู้จักคำว่า np เนื่องจากไม่ได้ import numpy as np</p>
<pre> import numpy as np n = int(input()) S = [] for i in range(n): S.append([int(e) for e in input().split()]) W = [float(e) for e in input().split()] S = np.array(S) W = np.array(W) for i in range(n): print(sum(S[i]*W)) </pre>	<p>ทำงานได้ถูกต้อง (ใช้การคูณแบบ element-wise) แต่ไม่ได้เร็วกว่าของเดิมนัก เพราะยังใช้คำสั่งของ numpy ได้ไม่เต็มที่</p>

โปรแกรม	คำอธิบาย
<pre> import numpy as np n = int(input()) S = [] for i in range(n): S.append([int(e) for e in input().split()]) W = [float(e) for e in input().split()] S = np.array(S) W = np.array(W) S *= W total_score = np.sum(S, axis = 0) for i in total_score: print(i) </pre>	<p>เราสามารถเขียน $S *= W$ ได้โดย เพราะ numpy จะ broadcast อาร์เรย์ W ให้โดยอัตโนมัติ สัง run, ใส่ข้อมูลตามตัวอย่างแรก, ได้ 28 9 9 ไม่ตรงกับที่แสดง พบว่าจำนวนตัวเลขไม่ครบ 4 ตัว น่าจะผิดที่ axis</p>
<pre> import numpy as np n = int(input()) S = [] for i in range(n): S.append([int(e) for e in input().split()]) W = [float(e) for e in input().split()] S = np.array(S) W = np.array(W) S *= W total_score = np.sum(S, axis = 1) for i in total_score: print(i) </pre>	<p>เปลี่ยนเป็น axis = 1 สัง run, ใส่ข้อมูลตามตัวอย่างแรก, ได้ 10 14 10 12 ไม่ตรงกับที่แสดง พบว่าสิ่งที่แสดงเป็นจำนวนเต็ม แสดงว่าจะมีปัญหาที่การเก็บค่า พบว่าคำสั่ง $S *= W$ จะเก็บผลลูปในอาร์เรย์ S ซึ่งเก็บจำนวนเต็ม ดังนั้น ผลลัพธ์จึงเป็นจำนวนเต็ม ซึ่งไม่ถูกต้อง</p>
<pre> import numpy as np n = int(input()) S = [] for i in range(n): S.append([int(e) for e in input().split()]) W = [float(e) for e in input().split()] S = np.array(S) W = np.array(W) T = S * W total_score = np.sum(T, axis = 1) for i in total_score: print(i) </pre>	<p>เปลี่ยนเป็น $T = S * W$ สัง run, ใส่ข้อมูลตามตัวอย่างแรก, ได้ 11.25 15.0 10.75 12.0 ถูกต้อง</p>
<pre> import numpy as np n = int(input()) S = [] for i in range(n): S.append([int(e) for e in input().split()]) W = [float(e) for e in input().split()] S = np.array(S) W = np.array(W) for i in S.dot(W): print(i) </pre>	<p>อีกวิธีที่ทำได้ และง่ายกว่าคือ ใช้คำสั่ง $S.dot(W)$ ซึ่งได้ผลลัพธ์เป็นอาร์เรย์ 1 มิติ มีข้อมูลแต่ละตัวคือ ค่าแบบรวมที่ถ่วงน้ำหนักแล้ว</p>

ตัวอย่างໂທນີ້ບໍ່ຢາ

ค่าเช่าหนังสือ

จะเขียนโปรแกรมคำนวณหาราคาค่าเช่าหนังสือของร้านเช่าหนังสือแห่งหนึ่ง ที่มีประเภทหนังสือให้เช่า 4 ประเภท คือ นิยาย สารคดี ท่องเที่ยว และการ์ตูน

เจ้าของร้านเช่าหนังสือต้องการทราบว่าในหนึ่งสัปดาห์ (คิด 5 วันจันทร์ถึงศุกร์) วันใดให้เช่าหนังสือเป็นจำนวนเล่มมากที่สุด เป็นจำนวนกี่เล่ม และค่าเช่าหนังสือรวมทุกประเภทในแต่ละวันเป็นจำนวนเท่าไร

► ข้อมูลนำเข้า

บรรทัดที่ 1 เป็นจำนวนเต็ม 4 จำนวนคันด้วยซึ่งว่าง แทนค่าเช่า นิยาย สารคดี ท่องเที่ยว และการ์ตูน

บรรทัดที่ 2 เป็นจำนวนหนังสือนิยายที่ถูกเช่าในวัน จ. อ. พ. พฤ. และ ศ. (คันด้วยซึ่งว่าง)

บรรทัดที่ 3 เป็นจำนวนหนังสือสารคดีที่ถูกเช่าในวัน จ. อ. พ. พฤ. และ ศ. (คันด้วยซึ่งว่าง)

บรรทัดที่ 4 เป็นจำนวนหนังสือท่องเที่ยวที่ถูกเช่าในวัน จ. อ. พ. พฤ. และ ศ. (คันด้วยซึ่งว่าง)

บรรทัดที่ 5 เป็นจำนวนหนังสือการ์ตูนที่ถูกเช่าในวัน จ. อ. พ. พฤ. และ ศ. (คันด้วยซึ่งว่าง)

► ข้อมูลส่งออก

บรรทัดแรกแสดง ชื่อวันที่มีจำนวนหนังสือถูกเช่ารวมมากสุด และจำนวนหนังสือรวมนั้น (ให้ลือว่ามีวันเดียวเท่านั้นที่ให้เช่ามากสุด)

บรรทัดที่สองแสดงค่าเช่ารวมของหนังสือทุกประเภทในแต่ละวัน (เรียงตั้งแต่วันจันทร์ถึงศุกร์ คันด้วยซึ่งว่าง)

► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
50 30 40 20 20 50 10 15 20 30 40 20 65 35 75 30 42 70 45 40 25 35 22 55	Thu 172 5700 5400 3480 5940 4950

ขั้นตอนการทำงานของโปรแกรม

1. x = ลิสต์ที่สร้างจากข้อมูลในบรรทัดแรกจากแป้นพิมพ์ เป็นจำนวนเต็ม 4 จำนวนแทนค่าเช่าหนังสือแต่ละประเภท
2. $rentalrates$ = สร้าง numpy array ที่มีค่าเริ่มต้นจากลิสต์ x
3. $sales$ = สร้าง numpy array ขนาด 4 แถว 5 คอลัมน์ (ແນהພະເທດหนังສຶກ ຄອລົມນີ້ແນວວັນ)
4. วนลặp 5 โดยเปลี่ยนค่าของตัวแปร $k = 0, 1, 2, 3$ (k ແນໝາຍເລີ່ມປະເທດหนังສຶກ)
5. $sales[k, :] =$ list ที่สร้างจากข้อมูลหนึ่งบรรทัดจากแป้นพิมพ์ เป็นจำนวนเต็ม 5 จำนวน แทนจำนวนหนังสือ ประเภทที่ k ของแต่ละวันที่ขายได้
6. $totalsales$ = ผลรวมของจำนวนหนังสือที่ถูกเช่าในแต่ละวันคำนวณจาก $sales$ (ในข้อ 3)
7. d = ตำแหน่งใน $totalsales$ ที่มีค่ามากสุด ($d = 0$ ແນວັນຈັນທີ, 1 ແນວັນອັງຄານ, ..., 4 ແນວັນຫຼຸກ)
8. หาชื่อวันจาก d และ tuple ('Mon', 'Tue', 'Wed', 'Thu', 'Fri')
9. แสดง ชื่อวัน ตามด้วย $totalsales[d]$
10. $salesvalues$ = ค่าเช่าหนังສຶກรวมของหนังສຶກทุกประเภทในแต่ละวัน (นำ $rentalrates$ มา dot กับ $sales$)
11. แสดงรายการของยอดเงินที่ขายได้จาก $salesvalues$ มาแสดง (คันด้วยซึ่งว่าง)

BMI

ฟังก์ชัน `read_height_weight()` ชี้ ล่าสุด ว่า จำนวนข้อมูลความสูง (หน่วยเป็นเซนติเมตร) และน้ำหนัก (หน่วยเป็นกิโลกรัม) มาสร้าง `numpy array` แบบสองมิติ ดังตัวอย่างในตารางข้างล่างนี้ (บรรทัดแรกคือจำนวนข้อมูล บรรทัดที่ 2 ตามมาคือ ความสูงกับน้ำหนัก)

Input	ผลที่ได้จาก <code>read_height_weight()</code>
4 160 60 155 62 170 54 180 55	array([[160, 60], [155, 62], [170, 54], [180, 55]])

จะเขียนฟังก์ชัน `cm_to_m(x)` และ `cal_bmi(hw)` ในโปรแกรมข้างล่างนี้ ที่มีข้อกำหนดของพารามิเตอร์ และผลลัพธ์ ที่ได้ตามตารางนี้

Function	Input parameter	Return value
<code>cm_to_m(x)</code>	<code>array</code> หนึ่งมิติ เก็บความสูงหน่วยเป็นเซนติเมตร เช่น <code>array([160, 155, 170, 180])</code>	<code>array</code> หนึ่งมิติเก็บความสูงหน่วยเป็นเมตร เช่น <code>array([1.6, 1.55, 1.7, 1.8])</code>
<code>cal_bmi(hw)</code>	<code>array</code> สองมิติ ขนาด n และ 2 คอลัมน์ แต่ละแถว แทนข้อมูลหนึ่งคู่ คอลัมน์ 0 เก็บความสูง (เซนติเมตร) คอลัมน์ 1 เก็บน้ำหนัก (กิโลกรัม) เช่น <code>array([[160, 60], [155, 62], [170, 54], [180, 55]])</code>	<code>array</code> หนึ่งมิติเก็บ bmi ที่คำนวณจาก ความสูงและน้ำหนักใน Input parameter ที่ได้รับ เช่น <code>array([23.4375, 25.80645161, 18.68512111, 16.97530864])</code>

และเขียนคำสั่ง

- หาค่าเฉลี่ยของ bmi ทั้งหมดที่คำนวณได้ เก็บใส่ตัวแปร `avg_bmi` และ
- นับจำนวน bmi ที่คำนวณได้ที่มีค่าน้อยกว่า 18.5

โปรแกรมเริ่มต้น

```
import numpy as np

def read_height_weight():
    list_hw = []
    for k in range(int(input())):
        h,w = input().split()
        list_hw.append((int(h),int(w)))
    return np.array(list_hw)

def cm_to_m(x):
    # ???

def cal_bmi(hw):
    # ???

def main():
    hw = read_height_weight()
    bmi = cal_bmi(hw)
    avg_bmi = _____
    count_underweight = _____
    print('average bmi =', avg_bmi)
    print('#bmi < 18.5 =', count_underweight)

exec(input().strip())
```

► ข้อมูลนำเข้า

คำสั่งในการทดสอบฟังก์ชันที่เขียน

► ข้อมูลสองออก

ผลที่ได้จากคำสั่งที่ป้อนเป็นข้อมูลนำเข้า

► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
x=np.array([160,150,140]); print(cm_to_m(x)); print(x)	[1.6 1.5 1.4] [160 150 140]
d=np.array([[100,30],[120,36]]); print(cal_bmi(d))	[30. 25.]
main() 4 160 60 155 62 170 54 180 55	average bmi = 21.2260953405 #bmi < 18.5 = 1

การคำนวณจำนวนพีโบนัชชีโดยใช้การยกกำลังเมทริกซ์อย่างรวดเร็ว

0, 1, 1, 2, 3, 5, 8, ... เป็นลำดับของ จำนวน พีโอนัชชี ($F_0 = 0, F_1 = 1, F_2 = 1, \dots$) วิธีหนึ่งในการหา F_n คือคำนวณผลการยกกำลัง n ของเมทริกซ์ A ที่แสดงด้านล่าง จะเห็นได้ว่าผลเป็นเมทริกซ์ขนาด 2×2 ที่มี F_n อยู่ที่มุมขวาบนของเมทริกซ์ เช่น

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

$$\text{หา } F_3 \text{ คำนวณ } A^3 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}^3 = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} \text{ ได้ } F_3 = 2 \quad \text{หา } F_4 \text{ คำนวณ } A^4 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}^4 = \begin{bmatrix} 2 & 3 \\ 3 & 5 \end{bmatrix} \text{ ได้ } F_4 = 3$$

ถ้าคิดดูดี ๆ จะพบว่าการหาด้วยวิธีข้างต้นนี้คือการหาค่ายกกำลัง ซึ่งหากไม่นำมาแบบค่อย ๆ คูณไปทีละครั้ง เช่น การหา A^{10} ก็ไม่น่าใช้วิธีที่เริ่มด้วยเมทริกซ์เอกลักษณ์ I และคูณด้วย A ไป 10 ครั้ง น่าจะใช้วิธีการหา A^5 แล้วจับมาคูณกับตัวเอง ที่จะได้ A^{10} นั่นคือ

$$A^n = \begin{cases} I & n = 0 \\ (A^{\lfloor n/2 \rfloor})^2 & n \text{ is even} \\ A(A^{\lfloor n/2 \rfloor})^2 & n \text{ is odd} \end{cases} \quad A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

จะเขียนฟังก์ชัน `fib(n, k)` เพื่อคำนวณ $F_n \% k$ ด้วยวิธีข้างต้นนี้ โดยใช้คำสั่งของ numpy เพื่อคูณเมทริกซ์ (หมายเหตุ: หลังการคูณเมทริกซ์ทุกครั้ง ให้นำผลที่ได้มา % k numpy จะทำ % k แบบ element-wise ในเมทริกซ์)

```
import numpy as np

def fib(n,k):
    # ???

n,k = [int(e) for e in input().split()]
print( fib(n,k) )
```

► ข้อมูลนำเข้า

จำนวนเต็ม 2 ค่า n กับ k ($0 \leq n \leq 10^{**13}$, $0 \leq k \leq 100000$)

► ข้อมูลส่งออก

แสดงค่า $F_n \% k$

► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
0 10	0
2 10	1
89 10	9
11111 111	55
1234567890 1234	162
1000000000000000 999	600

สรุปเนื้อหา

- คลาส ใช้สร้างປະເກດເນື່ອມໃໝ່ທີ່ຕ້ອງການ โดยเราກຳຫັນໄດ້ວ່າ ຈະໃຫ້ເກີບຂໍ້ອມຸລຍ່ອຍຂອງໄຣບ້າງ ແລະທໍານານອະໄຮໄດ້ບ້າງ
- คลາສ ດື່ມ ຊົ່ວໂມງ, ອົບເຈກຕໍ່ ດື່ມ ຕັ້ງຂໍ້ອມຸລ ເຊັ່ນ `b1 = Book(...)` ໄດ້ວ່າ `b1` ເປັນອົບເຈກຕໍ່ຂອງคลາສ `Book`
- ແມ່ທົດ ຜັກ໌ ຈຶ່ງເປັນບໍລິການຂອງคลາສ

ຕ້າວອ່າງ: ຄລາສ `Book`

- ຄລາສ `Book` ຂ້າງລ່າງນີ້ ໃຊ້ສ້າງອົບເຈກຕໍ່ທີ່ເກີບຂໍ້ອມຸລຂອງໜັນສື່ອ (ຊ່ອ ຮາຄາ ແລະຄໍາສຳຄັນຕ່າງໆ ຂອງໜັນສື່ອ)
- ທຸກເມື່ອດູດຂອງຄລາສຕ້ອງມີຕ້າແປຣ `self` ເປັນພາຣາມີເຕີເຮັກແກ້ໄຂແທນອົບເຈກຕໍ່ທີ່ຈະໃຫ້ບໍລິການ
ເນື່ອເຮັກໃຫ້ຕັ້ງແປຣປະຈຳອົບເຈກຕໍ່ກາຍໃນຄລາສ ຈະຕ້ອງມີ `self`. ນຳທັນເສົມອ
- ແມ່ທົດ `__init__` (ເວີກວ່າ constructor) ໃຊ້ສ້າງອົບເຈກຕໍ່ຂອງຄລາສ ໂດຍບອກວ່າອົບເຈກຕໍ່ຈະເກີບຂໍ້ອມຸລຂອງໄຣບ້າງ
ແລະຈະຖຸກເຮັກເນື່ອມີການສ້າງອົບເຈກຕໍ່ ໂດຍ `self` ຈະແທນອົບເຈກຕໍ່ທີ່ເພີ່ມສ້າງ
- ແມ່ທົດ `__str__` ສິນຄ່າຕຽງຂອງອົບເຈກຕໍ່ ຈະຖຸກເຮັກເນື່ອໃຫ້ຈານພັກໜັນ `str` ຢ່ອງ `print`
- ແມ່ທົດ `__lt__` ໃຊ້ສໍາໜັບເປີຍບໍ່ເປີຍອົບເຈກຕໍ່ຂອງ `Book` ໃນທີ່ນີ້ຈະເປີຍບໍ່ເປີຍໃຫ້ຮາຄາກ່ອນ ຄ້າຮາຄາເທົ່າກັນ
ຈະໃຫ້ຊ່ອໜັນສື່ອເປັນຕັ້ງເປີຍບໍ່ເປີຍ ແມ່ທົດນີ້ຈະຖຸກເຮັກເນື່ອເປີຍບໍ່ເປີຍອົບເຈກຕໍ່ດ້ວຍ < ຢ່ອງໃຫ້ພັກໜັນ `sort`
- ນອກຈາກນີ້ຍັງມີບໍລິການ ປັບຮາຄາ ແລະຂໍຄໍາສຳຄັນຮ່ວມຂອງໜັນສື່ອ 2 ເລີ່ມ ພ່ານເມື່ອດູດ `update_price` ແລະ
`get_common_keywords` (ສາມາດເຂົ້າຍິນເມື່ອດູດນີ້ ທ່ານໄດ້ຕ້າມຕ້ອງການ ເປັນການເພີ່ມຄວາມສາມາດຂອງອົບເຈກຕໍ່)
- ແມ່ທົດອາຈະຈຶ່ງຄືນຄ່າຫຼືໄມ້ໄດ້ ໂດຍທ້າວໄປນີ້ທົດທີ່ໄມ້ຄືນພລັພັນມັກເປັນນີ້ທົດທີ່ມີການເປີຍແປງຄ່າກາຍໃນອົບເຈກຕໍ່

```
class Book:
    def __init__(self, title, price, keywords):
        self.title = title; self.price = price; self.keywords = set(keywords)

    def __lt__(self, rhs):
        if self.price != rhs.price: return self.price < rhs.price
        else: return self.title < rhs.title

    def __str__(self):
        return self.title + ' (' + str(self.price) + ')'

    def update_price(self, new_price):
        self.price = new_price

    def get_common_keywords(self, other):
        return self.keywords & other.keywords

b1 = Book('Python', 99, ['code', 'computer'])           # using __init__
b2 = Book('Calculus', 199, ['maths'])
b3 = Book('Physics', 99, ['science', 'maths'])

b1.update_price(199)
print(Book.get_common_keywords(b2, b3))                  # {'maths'}
if b3 < b2: print(b1)                                   # using __lt__ & __str__
books = [b1, b2, b3]
books.sort()                                           # using __lt__
print(books[0], ', ', books[1], ', ', books[2])       # using __str__
# 'Physics ($99)', 'Calculus ($199)', 'Python ($199)'
```

- การเรียกเมท็อดที่ได้ 2 แบบ
 - เรียกผ่านชื่อคลาส เช่น `Book.get_common_keywords(b2,b3)`
 - เรียกผ่านอ้อบเจกต์ เช่น `b2.get_common_keywords(b3)`
(`b2` จะถูกแทนใน `self` และ `b3` จะถูกแทนใน `other` โดยอัตโนมัติ และทั้งสองคำสั่งนี้ทำงานเหมือนกัน)
- `str(b1)` เมื่อกดกับการเรียก `Book.__str__(b1)` หรือ `b1.__str__()`
- `print(b1)` เมื่อกดกับ การเรียก `print(str(b1))`
- `b1 < b2` เมื่อกดกับการเรียก `Book.__lt__(b1,b2)` หรือ `b1.__lt__(b2)`
- การเรียก `books.sort()` จะเรียงลำดับอ้อบเจกต์ของ `Book` จากน้อยไปมาก โดยเปลี่ยนเทียบด้วย `__lt__`

เรื่องผิดป oy

<p>ลืม <code>self</code> ในเมท็อด ไม่ได้ประกาศ <code>self</code> เป็นพารามิเตอร์ ไม่มี <code>self</code>. นำหน้าตัวแปรของอ้อบเจกต์</p>	<pre>class A: def __init__(self, x, y): d = dict() # แก้ไขเป็น self.d = dict() d[x] = y # แก้ไขเป็น self.d[x] = y def total(): # แก้ไขเป็น def total(self): return sum(d.values()) # แก้ไขเป็น return sum(self.d.values())</pre>
<p>การกำหนดค่าอ้อบเจกต์ให้กับตัวแปรด้วย เครื่องหมายเท่ากับ จะทำให้ตัวแปรนั้นซึ่งไปที่ อ้อบเจกต์เดียวกัน</p>	<pre>class B: def __init__(self, b): self.b = b b1 = B(10) b2 = b1 # b1, b2 อ้างอิงอ้อบเจกต์เดียวกัน b2.b = 5 print(b1.b) # 5</pre>
<p>indent ผิด</p>	<pre>class C: def __init__(self, c): self.c = c def double(self): # indent ผิด กล้ายเป็นพังก์ชันที่ไม่ใช่ self.c *= 2 # เรียก double(c1) ได้ # แต่จะเรียก c1.double() ไม่ได้ # เรียก C.double(c1) ก็ไม่ได้</pre>

แบบฝึกหัด

1. เติมเมทีออดของคลาสนิสิตตัว Component ที่กำหนดให้สมบูรณ์

```
class Nisit:  
    def __init__(self, name, year, faculty):  
        # n = Nisit('Krit', 4, 'Engineering')  
  
    def __str__(self):  
        # คืนสตริงของนิสิต เช่น 'Krit (year 4) Engineering'  
  
    def __lt__(self, rhs):  
        # เรียงลำดับนิสิตด้วยคณะตามพจนานุกรม ถ้าอยู่คณะเดียวกัน ให้เรียงลำดับด้วยชื่อตามพจนานุกรม  
        # ถ้าอยู่คณะและชื่อเป็นเดียวกัน ให้เรียงลำดับด้วยชื่อตามพจนานุกรม  
        # เช่น Nisit('Krit', 4, 'Engineering') < Nisit('Boy', 3, 'Science')  
        #       Nisit('Prim', 2, 'Engineering') < Nisit('Krit', 4, 'Engineering')  
        #       Nisit('Joey', 2, 'Engineering') < Nisit('Prim', 2, 'Engineering')
```

2. เดิมเมื่อต้องคลาสรถยนต์ตาม comment ที่กำหนดให้ บูรณา

```
class Car:  
    def __init__(self, license, brand, color):  
        # c = Car('AA1234', 'Honda', 'White')  
        # มีตัวแปร report สำหรับใช้ข้อมูลประวัติการซ่อมบำรุง โดยกำหนดค่าเริ่มต้นเป็นลิสต์ว่าง  
  
    def __str__(self):  
        # คืนสตริงของรถยนต์ เช่น 'AA1234 - White Honda'  
  
    def __lt__(self, rhs):  
        # เรียงลำดับรถยนต์โดยเบรียบเทียบป้ายทะเบียนรถแบบสตริง  
  
    def add_report(self, new_report):  
        # เพิ่มประวัติการซ่อมบำรุง โดยไม่ต้องคืนค่า  
        # ตัวแปร new_report เป็น tuple (วันที่, คำอธิบาย, ราคา)  
        # เช่น c.add_report( ('25 May 2017', 'change tires', 1500) )  
  
    def total_payment(self):  
        # คืนค่าใช้จ่ายทั้งหมดที่ใช้ในการซ่อมบำรุงที่ผ่านมา  
  
    def max_payment(self):  
        # คืนลิสต์ของประวัติการซ่อมบำรุง (วันที่, คำอธิบาย, ราคา) ทุกรายการ ที่มีค่าใช้จ่ายมากที่สุด  
        # กรณีที่รถยนต์ไม่มีประวัติการซ่อมบำรุงเลย ให้คืนค่าลิสต์ว่าง
```

3. จาก class Book ให้เดิมเมื่อต้องของ class ShoppingCart สำหรับการซื้อหนังสือผ่านเว็บไซต์ ดังนี้

```
class ShoppingCart:  
    def __init__(self, id):  
        self.id = id  
        self.books = []  
        # books เก็บ [หนังสือ] ที่อยู่ในตะกร้าพร้อมจำนวน เช่น [[b1,2],[b3,7]]  
  
    def add_book(self, book, n):  
        # เพิ่มชิ้นส่วนของการซื้อหนังสือ book เพิ่มอีก n เล่ม โดยไม่ต้องคืนค่า  
        # หากไม่มีหนังสือเล่มนี้ในตะกร้า ให้เพิ่มลิสต์ [book, n] ต่อท้าย books  
        # หากเคยมีข้อมูลหนังสือเล่มนี้ในตะกร้าแล้ว ให้เพิ่มจำนวนที่ซื้ออีก n เล่ม  
        # เช่น ถ้า books = [[b1,2]] และเราลั่ง add_book(b1,3) จะได้ books = [[b1,5]]  
  
    def delete_book(self, book):  
        # ลบข้อมูลการซื้อหนังสือ book ออกจากตะกร้า โดยไม่ต้องคืนค่า  
        # ถ้าในตะกร้าไม่มีหนังสือ book ไม่ต้องทำอะไร  
  
    def get_total(self):  
        # คืนค่าราคารวมของหนังสือทั้งหมดในตะกร้า  
  
    def __lt__(self, rhs):  
        # ตะกร้าที่มีราคารวมของหนังสือน้อยกว่า จะเป็นตะกร้าที่น้อยกว่า
```

4. ข้างล่างนี้แสดงคลาส Station และคลาส BTScard (อ่านคำอธิบายของแต่ละคลาสจาก comment ที่เขียน)

สถานีรถไฟเป็นอ็อบเจกต์ของคลาส Station และบัตรโดยสารแบบเติมเงินแต่ละใบเป็นอ็อบเจกต์ของคลาส BTScard จงเติมคำสั่งในเมท็อด add_value, enter, leave และ __lt__ ของคลาส BTScard ให้ทำงานตาม comment ที่เขียน (เมท็อดอื่นที่ได้เขียนคำสั่งไว้ ทำงานถูกต้อง (ลัว) ภูตัวย่อ่งการใช้งานข้างล่างนี้ประกอบ)

```
s1 = Station(1,'Siam'); s2 = Station(3,'Mo Chit'); s3 = Station(5,'Asok')
c1 = BTScard(123, 5), c2 = BTScard(999, 10)
# -----
c1.add_value(10)      # c1 มีเงินในบัตร 105 บาท
p = c1.enter(s1)     # p = True
p = c1.enter(s3)     # p = False (แตะเข้าสถานีหลังจากแตะเข้าไปแล้ว)
p = c1.leave(s2)     # c1 เหลือเงินในบัตร 95 บาท โดย p = (95, 0)
# -----
p = c2.enter(s3)     # p = True
p = c2.leave(s1)     # c2 มีเงินในบัตร 10 บาทไม่พอจ่ายค่าโดยสาร โดย p = (10, -1)
c2.add_value(50)     # c2 มีเงินในบัตร 60 บาท
p = c2.leave(s1)     # c2 เหลือเงินในบัตร 40 บาท โดย p = (40, 0)
p = c2.leave(s2)     # p = (40, -2) (ยังไม่ได้แตะเข้าสถานี จึงไม่มีสถานีต้นทาง)
p = c2.enter(s2)     # p = True
p = c1 < c2          # p = False
```

```
class Station:                      # คลาสของสถานีรถไฟ
    def __init__(self, id, name):   # สร้างสถานีที่มีหมายเลข (id) และชื่อสถานี (name)
        self.sid = int(id)           # กำหนดให้หมายเลขสถานีเป็นจำนวนเต็ม โดยสถานีที่ติดกัน
        self.name = name             # มีค่าห่างกัน 1

    def get_price(self, other):     # คืนค่าโดยสารระหว่างสถานี self และ other
        return abs(self.sid - other.sid)*5
# ----

class BTScard:                     # คลาสของบัตรโดยสารแบบเติมเงิน
    def __init__(self, id, value): # สร้างบัตรโดยสารที่มีเลขบัตร (id) และเงินเริ่มต้น (value)
        self.cid = id              # self.station เก็บว่าสถานีต้นทางคือสถานีอะไร
        self.value = value           # โดยถ้าบัตรไม่ได้อยู่ระหว่างการเดินทาง จะเก็บค่า
        self.station = ''            # สถานีต้นทางนี้เป็นสตริงว่าง ๆ

    def __str__(self):
        return '('+str(self.cid)+','+str(self.value)+')'

    def add_value(self, x):        # เพิ่มเงินในบัตรโดยสารเท่ากับ x โดยไม่ต้อง return
```

```

def enter(self, station):
    # แตะบัตรเพื่อเข้าสู่สถานีรถไฟฟ้า ให้เช็คว่า บัตรนี้ไม่ได้แตะเข้าที่สถานีอื่นมาก่อน
    # ถ้าไม่มีการแตะเข้ามาก่อน ให้เป็น False และสถานีต้นทางเป็น station และ return True
    # แต่ถ้ามีการแตะเข้าสถานีอื่นมาก่อน ให้ return False โดยไม่เปลี่ยนข้อมูลสถานีต้นทางของบัตรโดยสาร

def leave(self, station):
    # แตะบัตรเพื่ออกจากสถานีรถไฟฟ้า ให้เช็คว่า บัตรนี้มีข้อมูลสถานีต้นทางอยู่
    # ถ้าไม่มีข้อมูลสถานีต้นทาง ให้ return tuple ของเงินในบัตรและ -2
    # ถ้ามีสถานีต้นทาง แต่จำนวนเงินในบัตรไม่พอจ่ายค่าโดยสาร ให้ return tuple ของเงินในบัตรและ -1
    # ถ้ามีสถานีต้นทาง และจำนวนเงินในบัตรพอจ่ายค่าโดยสาร ให้ลบค่าโดยสารออกจากจำนวนเงินในบัตร
    # เปลี่ยนสถานีต้นทางเป็นตริงว่าง และ return tuple ของเงินในบัตรหลังหักค่าโดยสารและ 0

def __lt__(self, rhs):
    # บัตรโดยสารที่มีเงินในบัตรน้อยกว่า จะถือว่ามีอยู่กว่า

```

ตัวอย่างการแก้โจทย์ปัญหา

Bus

ให้เขียนคลาสของรถเมล์ซึ่งมีเมื่อวันนี้

1. `__init__` สร้างรถเมล์ 1 คัน รับพารามิเตอร์ จำนวนคนบนรถ `people` และค่าโดยสาร `fare`
2. `__str__` คืนค่าสตริงซึ่งบอกจำนวนคนบนรถและค่าโดยสาร
3. `__lt__` เปรียบเทียบรถเมล์โดยพิจารณาค่าโดยสารรวมของรถ (จำนวนคนบนรถคูณค่าโดยสารต่อคน)
4. `people_in` เพิ่มจำนวนคนบนรถ `k` คน ไม่คืนค่า
5. `people_out` ลดจำนวนคนบนรถ `k` คน (หากจำนวนคนน้อยกว่า 0 จะต้องแก้ไขจำนวนคนเป็น 0) ไม่คืนค่า
6. `change_fare` เปลี่ยนค่าโดยสารเป็นค่าโดยสารใหม่ `new_fare` ไม่คืนค่า

▶ ตัวอย่าง

```
b1 = Bus(10, 5)
b2 = Bus( 8, 7)
if b1 < b2:
    print(b1)
else:
    print(b2)
b1.people_in(3)
b1.people_out(6)
b1.change_fare(12)
print(b1)                                # b1 has 10 people with fare = 5
                                            # b2 has 8 people with fare = 7
                                            # b1 < b2 is True (10*5 < 8*7)
                                            this bus has 10 people with fare = 5

                                            # b1 has 13 people with fare = 5
                                            # b1 has 7 people with fare = 5
                                            # b1 has 7 people with fare = 12
                                            this bus has 7 people with fare = 12
```

ตัวอย่างการเขียนโปรแกรม

โปรแกรม	คำอธิบาย
<pre>class Bus: def __init__(people, fare): people = people fare = fare def __str__(): return 'this bus has ' + str(people) \ + ' people with fare = ' + str(fare) def __lt__(rhs): return people*fare < \ rhs.people*rhs.fare def people_in(k): return people += k def people_out(k): return people -= k def change_fare(new_fare): return fare = new_fare</pre>	<p>ทำงานไม่ถูกต้อง มีจุดผิดดังนี้</p> <ul style="list-style-type: none">- ไม่มีการใช้ <code>self</code>- เมท็อด <code>people_in</code>, <code>people_out</code> และ <code>change_fare</code> ต้องไม่คืนค่า- เมท็อด <code>people_out</code> ไม่ได้ตรวจสอบว่า จำนวนคนน้อยกว่าศูนย์หรือไม่

โปรแกรม	คำอธิบาย
<pre>class Bus: def __init__(self, people, fare): self.people = people self.fare = fare def __str__(self): return 'this bus has ' + \ str(self.people) + \ ' people with fare = ' + \ str(self.fare) def __lt__(self, rhs): return self.people*self.fare < \ rhs.people*rhs.fare def people_in(self, k): self.people += k def people_out(self, k): self.people = max(0,self.people-k) def change_fare(self, new_fare): self.fare = new_fare</pre>	<p>ทำงานถูกต้อง สังเกตการใช้งาน <code>self.people = max(0,self.people-k)</code> ว่าทำการแก้ไขจำนวนคนหากน้อยกว่าศูนย์ให้แล้ว</p>
<pre>class Bus: def __init__(self, people, fare): self.fare = fare self.total = people*fare def __str__(self): return 'this bus has ' + \ str(self.total//self.fare) + \ ' people with fare = ' + \ str(self.fare) def __lt__(self, rhs): return self.total < rhs.total def people_in(self, k): self.total += self.fare*k def people_out(self, k): self.total = max(0,self.total- \ self.fare*k) def change_fare(self, new_fare): self.total = \ self.total//self.fare*new_fare self.fare = new_fare</pre>	<p>ความสามารถเขียนคลาส Bus แบบอื่นได้ เช่น แทนที่จะเก็บค่าโดยสารต่ำคนและค่าโดยสารรวม ก็ได้ แต่ก็ต้องปรับการทำงานของคลาสให้ถูกต้อง (ในเมื่อต้อง <code>change_fare</code> ถ้าทำการเปลี่ยนค่า <code>self.fare</code> ก่อน จะทำให้เมื่อต้องทำงานผิดได้)</p>

ตัวอย่างโจทย์ปัญหา

เศษส่วน

กำหนดคลาสของเศษส่วน ประกอบด้วยเศษ (numerator) และส่วน (denominator) และมีเมท็อด 4 เมท็อดคือ เมท็อดสำหรับการแสดงผลในสตริง ที่ทำการทำเศษส่วนอย่างต่อเนื่อง เมท็อดการบวก และเมท็อดการคูณ ดังนี้ (ขอให้สังเกตการเรียกใช้งาน เมท็อด ว่าสามารถเรียกแบบเดียวกัน)

```
def gcd(x,y):
    if x%y == 0: return y
    return gcd(y,x%y)

class Fraction:
    def __init__(self,a,b):
        self.numerator = _____
        self.denominator = _____

    def __str__(self):
        # ???

    def simplify(self):
        g = gcd(self.numerator,self.denominator)
        return Fraction(self.numerator//g,self.denominator//g)

    def add(self,other):
        # ???

    def multiply(self,other):
        ans_numer = self.numerator * other.numerator
        ans_denom = self.denominator * other.denominator
        return Fraction(ans_numer,ans_denom).simplify()

a,b,c,d = [int(e) for e in input().split()]
fraction1 = _____
fraction2 = _____

print(fraction1.add(fraction2))
print(Fraction.multiply(fraction1,fraction2))
```

โจทย์ได้เขียนเมท็อดการทำเศษส่วนอย่างต่อเนื่อง และเมท็อดการคูณมาให้แล้ว ให้เขียนเติมส่วนอื่น ๆ ให้สมบูรณ์ สำหรับ การบวก เมื่อบวกแล้วให้ตอบเป็นเศษส่วนอย่างต่อตัว สามารถเรียกใช้เมท็อด `simplify` ได้

► ข้อมูลนำเข้า

มีบรรทัดเดียว เป็นจำนวนเต็มบวก $a \ b \ c \ d$ ซึ่งแทนเศษส่วน a/b และ c/d

► ข้อมูลส่งออก

มี 2 บรรทัด แสดงผลบวกและผลคูณของเศษส่วนที่กำหนดให้

► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
1 7 3 7	$4/7$ $3/49$
1 2 1 3	$5/6$ $1/6$
1 8 3 8	$1/2$ $3/64$
2 3 1 2	$7/6$ $1/3$

คลาสของไพ่

ข้างล่างนี้แสดงการเรียกใช้คลาส Card ซึ่งແນ່ນໄຟ' 1 ໃປ ປະກອບດ້ວຍ ຄໍາຂອງໄຟ (value) ຊື່ເປັນສຕຣີງ ('A', '2', '3', ..., '10', 'J', 'Q', 'K') ແລະ ດອກ ຈົກເຟ (suit) ຊື່ເປັນສຕຣີງເຊັ່ນ ('club', 'diamond', 'heart', 'spade') ໂປຣແກຣມຂ່າຍມີຮູບແບບຂອງຄະດີ (cards) ແລະ ມີການເຮັດວຽກໃຫ້ສະບຸງຮູບແບບຂອງຄະດີ

```
class Card:  
    def __init__(self, value, suit):  
        # ???  
  
    def __str__(self):  
        # ???  
  
    def getScore(self):  
        # ???  
  
    def sum(self, other):  
        # ???  
  
    def __lt__(self, rhs):  
        # ???  
  
n = int(input())  
cards = []  
for i in range(n):  
    value, suit = input().split()  
    cards.append(Card(value, suit))  
for i in range(n):  
    print(cards[i].getScore())  
print("-----")  
for i in range(n-1):  
    print(Card.sum(cards[i], cards[i+1]))  
print("-----")  
cards.sort()  
for i in range(n):  
    print(cards[i])
```

รายละเอียดต่าง ๆ ของคลาส Card และเมท็อดของคลาส Card

- เมท็อด getScore จะคืนค่าคะแนนของไพ่เป็นจำนวนเต็ม ตามกฎดังนี้
 - ไพ่ที่มีค่า A จะมีคะแนน + ค่าหนึ่น
 - ไพ่ที่มีค่า 2 ถึง 10 จะมีจำนวนเท่ากับค่าของไพ่ คือ 2 ถึง 10 คะแนน ตามลำดับ
 - ไพ่ที่มีค่า J, Q, K จะมีคะแนน 10 คะแนน
- เมท็อด sum จะคืนค่าจำนวนคะแนนของไพ่สองใบและ mod ด้วย 10 เช่น
 - Card.sum(Card('7', 'club'), Card('2', 'heart')) ได้ผลลัพธ์เป็น 9
 - Card.sum(Card('J', 'spade'), Card('5', 'diamond')) ได้ผลลัพธ์เป็น 5
- การเรียงลำดับของไพ่เป็นดังนี้
 - ค่าของไพ่เรียงตามลำดับดังนี้ 3 < 4 < 5 < ... < 10 < J < Q < K < A < 2
 - ดอกของไพ่เรียงตามลำดับดังนี้ club < diamond < heart < spade
 - ถ้าไพ่สองใบมีค่าไม่เท่ากัน ไพ่ที่มีค่ามากกว่าจะเป็นไพ่ที่มากกว่า
 - ถ้าไพ่สองใบมีค่าเท่ากัน ไพ่ที่มีดอกสูงกว่าจะเป็นไพ่ที่มากกว่า

▶ ข้อมูลนำเข้า

บรรทัดแรกมีจำนวนเต็ม n แทนจำนวนไพ่ที่จะรับเข้ามา
 n บรรทัดต่อมา แต่ละบรรทัดมีค่าและดอกของไพ่แต่ละใบ คั่นด้วยช่องว่าง

▶ ข้อมูลส่งออก

มี $3n+1$ บรรทัด

n บรรทัดแรก แสดงคะแนนของไพ่แต่ละใบ ตามด้วยขีดคิ้น 1 บรรทัด

$n-1$ บรรทัดถัดมา แสดงคะแนนรวมของไพ่ 2 ใบที่ติดกันในลำดับ ตามด้วยขีดคิ้น 1 บรรทัด

n บรรทัดสุดท้าย แสดงไพ่เรียงตามลำดับ

▶ ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
5 A spade K heart K club 7 diamond 2 spade	1 10 10 7 2 ----- 1 0 7 9 ----- (7 diamond) (K club) (K heart) (A spade) (2 spade)

กระบุกออมสิน

จะเขียนคลาส piggybank เพื่อผลิตย์ฯ บache ตัวกําบุกออมสิน ที่สามารถหยดเหรียญมูลค่าอะไรก็ได้ โดยจำกัดจำนวนเหรียญรวมทุกแบบในกระบุกแล้วห้ามเกิน 100 เหรียญ (ถ้ายอดแล้วเกินไม่รับเพิ่ม) ตามโครงของคลาสและตัวอย่างการใช้งานข้างล่างนี้

โครงของคลาส piggybar'.	ตัวอย่างการใช้งาน piggybank
<pre>class piggybank: def __init__(self): # มีตัวแปร self.coins เป็น dict เริ่มต้นให้ว่าง ๆ # มี key เป็นมูลค่าเหรียญ และ value เป็นจำนวนเหรียญ def add(self, v, n): # ถ้าเพิ่มจำนวนเหรียญในกระบุกอีก n เหรียญแล้วเกิน 100 # จะไม่ให้เพิ่ม ให้คืน False และว่า เพิ่มไม่สำเร็จ # แปลง v เป็น float ก่อน (เพิ่ม 5 กับ 5.0 จะได้เหมือนกัน) # ถ้ากระบุกไม่เคยมีเหรียญ v ทำ self.coins[v] = 0 # ทำการลั่ง self.coins[v] += n # คืน True และว่าเพิ่มสำเร็จ def __float__(self): # นำค่าของเหรียญดูณกับจำนวนเหรียญ ของเหรียญทุกแบบ # ต้องคืนจำนวนแบบ float เท่านั้น อย่างคืนคุณย์ ก็ต้อง 0.0 def __str__(self): # คืนสตริงที่แสดงจำนวนเหรียญแต่ละแบบตามตัวอย่าง # โดยให้เรียงเหรียญตามมูลค่าจากน้อยไปมาก</pre>	<pre>p1 = piggybank() print(float(p1)) # 0.0 p1.add(0.25, 4) # เพิ่มเหรียญ 25 สตางค์ 4 เหรียญ print(float(p1)) # 1.0 p1.add(0.50, 1) # เพิ่มเหรียญ 50 สตางค์ 1 เหรียญ print(float(p1)) # 1.5 p1.add(10, 1) # เพิ่มเหรียญ 10 บาท 1 เหรียญ print(float(p1)) # 11.5 print(p1) # {0.25:4, 0.5:1, 10.0:1} print(p1.add(10, 1)) # True เพิ่มได้ print(float(p1)) # 21.5 print(p1.add(1,94)) # False เพิ่มไม่ได้ เกิน 100 เหรียญ print(float(p1)) # 21.5</pre>

เมื่อ `__float__` ถูกเรียกเมื่อ `float(p)` ทำงาน โดยที่ `p` เป็น `piggybank` ได้ผลลัพธ์เป็น `float` แทนค่าของ `p`
เมื่อ `__str__` ถูกเรียกเมื่อ `str(p)` ทำงาน โดยที่ `p` เป็น `piggybank` ได้ผลลัพธ์เป็นสตริงแทนค่าของ `p`

► การส่งตรวจ

ให้นำโปรแกรมข้างล่างนี้ ต่อท้าย `Class piggybank` ที่เขียนข้างบนนี้ แล้วจึงส่งให้ Grader ตรวจ

```
cmd1 = input().split(';')
cmd2 = input().split(';')
p1 = piggybank(); p2 = piggybank()
for c in cmd1: eval(c)
for c in cmd2: eval(c)
```

► ข้อมูลนำเข้า

คำสั่งต่อ ๆ กันเพื่อทดสอบคลาส

► ข้อมูลส่งออก

ผลการทำงานของโปรแกรมข้างบนที่อ้างอิงคลาส `piggybank`

► ตัวอย่าง

Input (จากแป้นพิมพ์)	Output (ทางจอภาพ)
<code>p1.add(1.11,2); print(float(p1), p1) print(float(p2), p2)</code>	<code>2.22 {1.11:2} 0.0 {}</code>
<code>p1.add(0.25,1);p1.add(5,1);p1.add(0.25,2);p1.add(5.0,1) print(float(p1), str(p1))</code>	<code>10.75 {0.25:3, 5.0:2}</code>
<code>p1.add(0.25,1); print(p1.add(0.25,100)) print(p1.add(0.25,99)); print(float(p1))</code>	<code>False True 25.0</code>

CONFIDENTIAL



AS. สุชาติ สุตันทวีบูลย์

Intania 63

Principal Member of Technical Staff
Advanced Micro Devices Inc. (AMD)
California, USA

การใช้คอมพิวเตอร์ช่วยออกแบบ (Computer Aid Design : CAD) มีความสำคัญมากสำหรับการออกแบบ System on Chips ซึ่งประกอบด้วยทรานซิสเตอร์หลายพันล้านตัว วิศวกรออกแบบ chips ที่ประสบความสำเร็จส่วนใหญ่มีความรู้ความสามารถในการใช้โปรแกรม CAD และสามารถเขียนโปรแกรมเสริมเพื่อให้ทนทานทำงานออกแบบได้เร็วขึ้น การเขียนโปรแกรมให้ได้ดีจึงต้องเริ่มด้วยทักษะพื้นฐานด้าน Programming และเสริมด้วยองค์ความรู้เรื่อง Algorithms และ Data Structures จึงจะทำให้เขียนโปรแกรมที่มีประสิทธิภาพ เช่น ทำงานได้เร็ว ใช้หน่วยความจำน้อย และใช้พลังงานต่ำ เป็นต้น

11 : Solutions to Exercises

บทที่ 1

► ข้อที่ 1

```
h = int(input())
m = int(input())
s = int(input())
total = 60*t*h + 60*m + s
print(total)
```

► ข้อที่ 2

```
import math
x = float(input())
y = 2 - x + 3/7*x**2 - 5/11*x**3 + \
    math.log(x,10)
print(y)
```

► ข้อที่ 3

```
a = float(input())
x = 1
x = (x + a/x)/2
print(x)
```

► ข้อที่ 4

```
v1,v2,v3 = [float(e) \
             for e in input().split()]
u1,u2,u3 = [float(e) \
             for e in input().split()]
dotp = v1*u1 + v2*u2 + v3*u3
print(dotp)
```

► ข้อที่ 5

```
import math
x1,y1,x2,y2 = \
    [float(e) for e in input().split()]
d = math.sqrt((x1-x2)**2+(y1-y2)**2)
print(d)
```

► ข้อที่ 6

```
import math
r,theta = [float(e) \
            for e in input().split()]
x = r*math.cos(theta)
y = r*math.sin(theta)
print(x, y)
```

► ข้อที่ 7

```
import math
x,y = [float(e) for e in input().split()]
r = math.sqrt(x**2+y**2)
theta = math.atan2(y,x)
print(r,theta)
```

► ข้อที่ 8

```
a,b,c,d,g = \
    [float(e) for e in input().split()]
avg = (a+b+c+d+g)/5
print(avg)
```

► ข้อที่ 9

```
a,b,c = [e for e in input().split()]
c = int(c)
out = a + b + str(c) + (a + b) * c
print(out)
```

บทที่ 2

► ข้อที่ 1

```
# ex. 3 4 1 --> median is 3
a,b,c = [int(e) for e in input().split()]
if b <= a <= c or c <= a <= b :
    print(a)
elif a <= b <= c or c <= b <= a :
    print(b)
else :
    print(c)
```

► ข้อที่ 2

```
x1,y1,r1 = [float(e) \
              for e in input().split()]
x2,y2,r2 = [float(e) \
              for e in input().split()]
d = (x1-x2)**2 + (y1-y2)**2
sumr2 = (r1+r2)**2
if d < sumr2 :
    print('overlap')
elif d == sumr2 :
    print('touch')
else :
    print('free')
```

► ข้อที่ 3

```
x,y = [float(e) \
        for e in input().split()]
if x == 0 and y == 0 :
    print('At the origin')
elif x == 0 :
    print('On y-axis')
elif y == 0 :
    print('On x-axis')
elif x > 0 and y > 0 :
    print('Quadrant I')
elif x < 0 and y > 0 :
    print('Quadrant II')
elif x < 0 and y < 0 :
    print('Quadrant III')
else :
    print('Quadrant IV')
```

► ข้อที่ 4

```
a,b,c,d,e = [int(e) \
               for e in input().split()]
if a <= b <= c <= d <= e :
    print('True')
else :
    print('False')

# print( a <= b <= c <= d <= e )
```

► ข้อที่ 5

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

mx = a
if b > mx : mx = b
if c > mx : mx = c
if d > mx : mx = d
mn = a
if b < mn : mn = b
if c < mn : mn = c
if d < mn : mn = d
s = (a+b+c+d) - mx - mn
print(s)

# s = (a+b+c+d)-max(a,b,c,d)-min(a,b,c,d)
```

► ข้อที่ 6

```
a = int(input())
x = int(round(a***(1/3),0))
if x**3 == a :
    print(x)
else :
    print('Not Found')
```

► ข้อที่ 7

```
c = int(input())
if c < 37 :
    s = 'XS'
elif c < 41 :
    s = 'S'
elif c < 43 :
    s = 'M'
elif c < 46 :
    s = 'L'
else :
    s = 'XL'
print(s)
```

บทที่ 3

► ข้อที่ 1

```
k = 1
while 1/k*k == 1 :
    k += 1
print(k)
```

► ข้อที่ 2

```
k = 1
p = 1.0
while (1-p) < 0.5 :
    p *= (365-k)/365
    k += 1
print(k)
```

► ข้อที่ 3

```
p = 0.0
for k in range(1,400000,4):
    p += 1/k
    p -= 1/(k+2)
print(4*p)
```

► ข้อที่ 4

```
a,b = [int(e) for e in input().split()]
s = 0
for i in range(a,b):
    t = 0
    for j in range(i+1,b+1):
        t += (i+j)
        s += (-1)**i * t
print(s)
```

► ข้อที่ 5

```
a,b = [int(e) for e in input().split()]
s = 0
for i in range(a,b):
    for j in range(i+1,b+1):
        s += (-1)**i * (i+j)
print(s)
```

► ข้อที่ 6

```

n = int(input())
a = int(input())
mn = mx = a
c = 0
if a < 0 : c = 1
for k in range(n-1):
    a = int(input())
    if a > mx : mx = a
    if a < mn : mn = a
    if a < 0 : c += 1
print( (mx - mn), c)

```

► ข้อที่ 7

```

n = int(input())
for x in range(1,n+1):
    for y in range(x,n+1):
        for z in range(y,n+1):
            t = x**2+y**2+z**2
            w = int(round(t***(1/3),0))
            if w**3 == t:
                print(w,x,y,z)

```

บทที่ 4

► ข้อที่ 1

```

s = input().strip()
t = ''
for e in s:
    t += e*2
print(t)

```

► ข้อที่ 2

```

s = input().strip()
t = ''
s = ' ' + s + ' '
for i in range(1,len(s)-1) :
    t += s[i]
    if s[i-1] != s[i] != s[i+1] :
        t += s[i]
print(t)

```

► ข้อที่ 3

```

s = input().strip()
if s == s[::-1] :
    print('Y')
else:
    print('N')

```

► ข้อที่ 4

```

d,n = [int(e) for e in input().split()]
t = "0"*n + str(d)
t = t[-max(n,len(str(d))):]
print(t)

```

► ข้อที่ 5

```

h = input().strip()
d = '0123456789ABCDEF'.find(h)
if d >= 0 :
    print(d)
else :
    print('invalid hex digit')

```

► ข้อที่ 6

```

t = input().strip()
c = 0
for e in t :
    if e in '13579' :
        c += 1
print(c)

```

► ข้อที่ 7

```

t = input().strip()
c = 0
for k in range(len(t)-1) :
    if t[k] in 'aeiou' and \
       t[k+1] in 'aeiou' :
        c += 1
print(c)

```

► ข้อที่ 8

```

b = input().strip()
b = b[::-1]
d = 0
for i in range(len(b)) :
    d += int(b[i])*2**i
print(d)

```

บทที่ 5

► ข้อที่ 1

```

file1 = open( input().strip() )
s = ''
for line in file1 :
    if line[-1] != '\n' : # กรณีบรรทัดสุดท้ายไม่มี \n
        line = line + '\n'
    s = line + s # นำบรรทัดใหม่มาต่อท้ายชี้อีก
file1.close()
print(s[:-1]) # ลบ \n ที่บรรทัดสุดท้าย

```

► ข้อที่ 2

```
file1 = open( input().strip() )
s = ''
for line in file1 :
    if len(line.strip()) > 0 :
        if line[-1] != '\n'
            line = line + '\n'
        s = line + s
file1.close()
out = open('reverse txt', 'w')
out.write(s[:-1])
out.close()
```

► ข้อที่ 3

```
file1 = open( input().strip() )
for line in file1 :
    a = line.find("<headline>")
    if a >= 0 :
        j = a+len("<headline>")
        b = line.find("</headline>",j)
        print( line[j:b] )
file1.close()
```

► ข้อที่ 4

```
file1 = open( input().strip() )
file2 = open( input().strip() )
for line1 in file1 :
    line2 = file2.readline()
    if line1 != line2 :
        print( False )
        break
    else :
        print( len(file2.readline()) == 0 )
file1.close()
file2.close()
```

หรือแบบสั้น ๆ ใช้ readlines

```
file1 = open( input().strip() )
file2 = open( input().strip() )
print( file1.readlines() == \
      file2.readlines() )
```

บทที่ 6.1

► ข้อที่ 1

```
v1 = [float(e) for e in input().split()]
v2 = [float(e) for e in input().split()]
if len(v1) != len(v2) :
    print( 'Error' )
else :
    dotp = 0
    for k in range(len(v1)) :
        dotp += v1[k]*v2[k]
    print(dotp)
```

► ข้อที่ 2

```
n = int(input())
d = []
for k in range(n):
    d.append( int(input()) )
d.sort()
t = []
for e in d :
    t.append(str(e))
print( ','.join(t) )
```

► ข้อที่ 3

```
file1 = open( input().strip() )
d = []
for line in file1:
    d.append( int(line) )
file1.close()
c = []
for e in d:
    c.append( d.count(e) )
maxc = max(c)
out = []
for k in range(len(d)):
    if c[k]==maxc and d[k] not in out :
        out.append(d[k])
for e in out :
    print(e)
```

► ข้อที่ 4

```
file1 = open( input().strip() )
h = []
for line in file1 :
    a = line.find("<headline>")
    if a >= 0 :
        j = a+len("<headline>")
        b = line.find("</headline>",j)
        h.append( line[j:b] )
file1.close()
h.sort()
for e in h :
    print(e)
```

► ข้อที่ 5

```
file1 = open( input().strip() )
d = []
for line in file1 :
    d.append( line.strip() )
file1.close()

for k in range(len(d)-1) :
    for i in range(len(d)-1) :
        if len(d[i]) > len(d[i+1]) or \
           len(d[i]) == len(d[i+1]) and \
           d[i] > d[i+1] :
            d[i],d[i+1] = d[i+1],d[i]

for e in d :
    print(e)
```

บทที่ 6.2

► ข้อที่ 1

```
r,c = [int(e) for e in \
        input().split()]
m = []
for k in range(r) :
    m.append( [int(e) for e in \
                input().split()] )
if len(m[k]) != c :
    m = [[]]
    break
print( m )
```

► ข้อที่ 2

```
infile = open(input().strip())
f = []
for line in infile :
    usernames = line.split()
    f.append( [usernames[0], \
                usernames[1:]] )
infile.close()
print( f )
```

► ข้อที่ 3

```
# สมมติว่ามี f มาแล้ว
nofollowers = []
for [username,followers] in f :
    if len(followers) == 0 :
        nofollowers.append( username )
print( 'No followers :', \
      ','.join(nofollowers) )
```

► ข้อที่ 4

```
n = int(input())
d = []
for k in range(n)
    s = input().strip()
    d.append( [len(s),s] )
d.sort()
for [c,s] in d :
    print(s)
```

บทที่ 6.3

► ข้อที่ 1

```
d = [ e.count(c) for e in x ]
```

► ข้อที่ 2

แบบแรก

```
x = [e for e in x if e >= 0]
คือการสร้างลิสต์ใหม่ที่ไม่มีเลขลบ แล้วให้ x อ้างอิงลิสต์ใหม่นี้
ถ้าเขียนอีกแบบ
x[:] = [e for e in x if e >= 0]
คือการสร้างลิสต์ใหม่ที่ไม่มีเลขลบ แล้วนำค่าในลิสต์ใหม่นี้ไปใส่
ในที่เก็บเดิมของลิสต์ x สองวิธีนี้ให้ผลคล้ายกัน ต่างกันตรงที่
ถ้าก่อนหน้านี้มีการทำคำสั่ง y = x คือให้ y อ้างอิงลิสต์เดียวกับ x
การทำแบบแรกจะทำให้ y อ้างอิงลิสต์เดิม และ x อ้างอิงลิสต์ใหม่
ในขณะที่แบบหลัง y ยังคงอ้างอิงลิสต์เดียวกับ x
```

► ข้อที่ 3

```
# t = [ [1,2,3], [33], [3,3,3,4] ]
t = sum( [ sum(e) for e in x ] )
```

► ข้อที่ 4

```
c = sum( [1 for e in input().split() \
           if int(e) < 0])
```

► ข้อที่ 5

```
t = ''.join([e for e in input() \
             if 'a' <= e.lower() <= 'z'])
```

► ข้อที่ 6

```
x = [float(e) for e in input().split()]
y = [float(e) for e in input().split()]
z = [x[i]+y[i] for i in range(len(x))]
```

► ข้อที่ 7

```
f = [e for row in m for e in row]
```

► ข้อที่ 8

```
x = [int(e) for e in input().split()]
x.sort()
t = [x[i] for i in range(len(x)-1) \
      if x[i] != x[i+1] ]
t.append(x[-1])
```

เขียนแบบนี้ก็ได้ (แต่ช้ากว่าเยอะ)

```
x = [int(e) for e in input().split()]
t = []
for e in x :
    if e not in t :
        t.append(e)
```

► ข้อที่ 9

```

n = int(input())
x = [j for i in range(2, n//2) \
      for j in range(2*i, n, i)]
x.sort()
c = [x[i] for i in range(len(x)-1) \
      if x[i] != x[i+1]]
c.append(x[-1])

```

► ข้อที่ 10

```

n = int(input())
x = [j for i in range(2, n//2) \
      for j in range(2*i, n, i)]
x.sort()
c = [x[i] for i in range(len(x)-1) \
      if x[i] != x[i+1]]
c.append(x[-1])
p = [e for e in range(2,n) if e not in c]

```

บทที่ 7

► ข้อที่ 1

- 1) dict มี key คือ studentID,
value คือ grade
- 2) set ของ studentID
- 3) dict มี key คือ dept,
value คือ set ของ studentID
- 4) list ของ phone
- 5) dict มี key คือ เลขท้าย TelNo,
value คือ count

► ข้อที่ 2

```

x = int(input())
t = ()
for i in range(0,x,2):
    # คำสั่งข้างล่างนี้ไม่ได้เปลี่ยน tuple ของ t แต่สร้าง
    # tuple ใหม่ แล้วแทนที่ t ด้วยกัน
    t += (i,) # เมื่อ n t = t + (i,)
print(t)
หรือ
t = tuple([e for e in range(0,x,2)])
หรือ
t = tuple(range(0,x,2))

```

► ข้อที่ 3

```

x = int(input())
t = ()
while x > 0:
    t = (x%10,) + t
    x //= 10
print(t)
หรือ เปลี่ยนมาใช้สตริง, นำแต่ละหลักมาเปลี่ยนเป็น int,
เก็บใส่ list, แล้วส่งไปสร้างเป็น tuple
x = input().strip()
print(tuple([int(e) for e in x]))

```

► ข้อที่ 4

```

x = input().strip()
d = {}
for e in x:
    if e not in d: d[e] = 1
    else: d[e] += 1
print(d)

```

► ข้อที่ 5

```

x = input().strip()
y = input().strip()
set_x = set(x)
set_y = set(y)
print(set_x.intersection(set_y))

```

บทที่ 8

► ข้อที่ 1

```

def f1(a,b):
    for i in range(b): print(a)

```

► ข้อที่ 2

```

def f2(a,b):
    return [a]*b

```

► ข้อที่ 3

```

def g(m,b,n,c):
    if m==n and b!=c: return 1
    if m==n and b==c: return 2
    x = (c-b)/(m-n)
    y = m*x + b
    return (x,y)

```

► ข้อที่ 4

```

def h(x):
    return [e for e in x if e%2==0]

```

► ข้อที่ 5

```

def a(n):
    if n==0: return 1
    if n==1: return -2
    return a(n-2)*n

```

► ข้อที่ 6

```
def k(n):
    if n==0: return 1
    if n==1: return 2
    if n%2==0:
        x = k(n//2)
        return x + x%10
    return k(n//2-1)*(n%2)
```

► ข้อที่ 7

```
def s(i,k):
    if i>=k: return 0
    return k - t(i+1,k)
def t(j,k):
    if j>=k: return 0
    return j + s(j,k-1)
```

► ข้อที่ 8

```
def is_palindrome(s):
    if len(s) <= 1: return True
    if s[0]!=s[-1]: return False
    return is_palindrome(s[1:-1])
```

บทที่ 9

► ข้อที่ 1

```
k = int(input())
M[:,k] = 0
```

► ข้อที่ 2

แบบนี้ก็ยังช้าอยู่

```
k = int(input())
for i in range(M.shape[0]):
    for j in range(M.shape[1]):
        if i%k==0 and j%k==0:
            M[i][j]*=2
```

แบบนี้เร็วขึ้น

```
M = np.array([[2*M[i][j] if i%k==0 \
    and j%k==0 else M[i][j] \
    for j in range(M.shape[1])] \
    for i in range(M.shape[0])])
```

แบบนี้เร็ว

```
k = int(input())
N = np.zeros_like(M)
N[:,k] = 1
M += M*N
```

แบบนี้เร็วและลื้น

```
k = int(input())
M[:,k] *= 2
```

► ข้อที่ 3

```
MAX = np.max(M, axis=0)
MIN = np.min(M, axis=0)
A = MAX-MIN
```

► ข้อที่ 4

```
Y = (X[:,0]**2+X[:,1]**2)**0.5
```

► ข้อที่ 5

```
k = int(input())
C = np.zeros((k,k),int)
C[::2, ::2] = C[1::2, ::2] = 1
```

► ข้อที่ 6

```
k = int(input())
C = np.zeros((k,k),int)
C[::2, ::2] = C[1::2, 1::2] = 1
C = (C*np.arange(1,k+1)).T
# ใช้ C*(np.arange(1,k+1).T) ไม่ได้ (หาระยะไร?)
```

บทที่ 10

► ข้อที่ 1

```
def __init__(self, name, year, faculty):
    self.name = name
    self.year = year
    self.faculty = faculty
```

def __str__(self):
 return self.name + \
 ' (' + str(self.year) + ') ' + \
 self.faculty

def __lt__(self, rhs):
 if self.faculty != rhs.faculty:
 return self.faculty < rhs.faculty
 if self.year != rhs.year:
 return self.year < rhs.year
 return self.name < rhs.name

► ข้อที่ 2

```
def __init__(self, license, brand, color):
    self.license = license
    self.brand = brand
    self.color = color
    self.report = []

def __str__(self):
    return self.license + '-' + \
        self.color + ' ' + self.brand

def __lt__(self, rhs):
    return self.license < rhs.license

def add_report(self, new_report):
    self.report.append(new_report)

def total_payment(self):
    return sum([r[2] for r in self.report])

def max_payment(self):
    if self.report == []: return []
    max_p = max([r[2] for r in self.report])
    return [r for r in self.report \
            if r[2] == max_p]
```

► ข้อที่ 3

```
def add_book(self, book, n):
    for b in self.books:
        if b[0] == book:
            b[1] += n
            break
    else:
        self.books.append([book,n])

def delete_book(self, book):
    self.books = [[b,n] for [b,n] \
                 in self.books if b != book]

    หรือ
    for k in range(len(self.books)):
        if self.books[k][0] == book:
            self.books.pop(k)
            break

def get_total(self):
    return sum([b.price*n for \
               [b,n] in self.books])

def __lt__(self, rhs):
    return self.get_total() < \
        rhs.get_total()
```

► ข้อที่ 4

```
def add_value(self, x):
    self.value += x

def enter(self, station):
    if self.station == '':
        self.station = station
        return True
    else:
        return False

def leave(self, station):
    if self.station == '':
        return (self.value, -2)
    price = Station.get_price \
        (self.station, station)
    if price > self.value:
        return (self.value, -1)
    else:
        self.value -= price
        self.station = ''
        return (self.value, 0)

def __lt__(self, rhs):
    return self.value < rhs.value
```

เว็บไซต์ที่เป็นประโยชน์

- youtu.be/U2lxgpVsbc?list=PL0R0NaCzUGB4ieaQndKybT9xyoq2n9NGq (ภาพยนตร์บรรยายเนื้อหาวิชา)
- en.wikibooks.org/wiki/Python_Programming (เว็บไซต์อ้างอิงสำหรับการเขียนโปรแกรมภาษา Python)
- www.pythontutor.org (เว็บไซต์ทางการของภาษา Python)
- docs.python.org/3/tutorial/index.html, docs.python.org/3/library/index.html, docs.python.org/3/reference/index.html (เว็บไซต์ทางการของภาษา Python ส่วนเอกสารอ้างอิง)
- www.numpy.org (เว็บไซต์ทางการของคลังคำสั่ง NumPy)
- www.lfd.uci.edu/~gohlke/pythonlibs (เว็บไซต์สำหรับดาวน์โหลดคลังคำสั่งภาษา Python)
- www.pythontutor.com (เว็บไซต์สำหรับการ visualize โปรแกรมภาษา Python)
- repl.it/languages/python3 (เว็บไซต์สำหรับเขียนโปรแกรมภาษา Python ออนไลน์ สามารถใช้ NumPy ได้)
- openbookproject.net/thinkcs/python/english3e (เว็บไซต์อ้างอิงสำหรับการเขียนโปรแกรมภาษา Python)

Error ที่สามารถพบได้

แบบขึ้นเป็นกล่องข้อความ

ข้อความ error	สาเหตุ	แนวทางการแก้ไข
unexpected EOF while parsing	ใส่ร่องเล็บไม่ครบ เช่น x = (1+(2**3))	ตรวจสอบว่างเล็บให้ครบ
EOL while scanning string literal	ใส่อัญประกาศจบสตริงไม่ครบ เช่น s = 'hello	เดินอัญประกาศปิดสตริงให้ครบ
invalid syntax	มีการเขียนโปรแกรมผิดกฎของภาษา เช่น ลืม : หลัง if หรือ while หรือเรียง else มาก่อน if หรือเบรียบเทียบด้วย = ตัวเดียว เป็น if x = 1 : หรืออาจจะมี อักษรพิเศษซ่อนอยู่ ซึ่งมาจากการ copy-paste คำสั่งจากไฟล์ pdf	ตรวจสอบการเขียนโปรแกรมให้เป็นไปตามกฎของภาษา
unexpected indent	มีการเยื่องที่ไม่ตรงกัน เช่น if a>0: a+=1 a+=2	จัดการเยื่องให้ตรงกัน

หมายเหตุ ควรตรวจสอบข้อผิดพลาดทั้งบรรทัดที่เกิด error และบรรทัดก่อนหน้าบรรทัดที่เกิด error

แบบขึ้นเป็นตัวอักษรสีแดงใน shell

ข้อความ error	สาเหตุ	แนวทางการแก้ไข
name 'x' is not defined	ไม่ได้กำหนดค่าตัวแปร x ก่อนใช้งาน เช่น <code>print(x+1)</code> หรือพิมพ์ชื่อฟังก์ชันผิด เช่น <code>pintt('cat')</code> หรือลืม <code>import</code>	กำหนดค่าตัวแปรก่อนใช้งานให้เรียบร้อย ตรวจสอบการพิมพ์ชื่อฟังก์ชันทำการ <code>import</code> ให้เรียบร้อย
ZeroDivisionError: division by zero	มีการหารด้วย 0 เช่น <code>print(5/0)</code>	แก้ไขตัวหารให้ไม่เป็น 0
ImportError: No module named 'maht'	พิมพ์ชื่อ module ที่ต้องการ import ผิด เช่น <code>import maht</code>	แก้ไขชื่อ module ที่ต้องการ import ให้ถูกต้อง
AttributeError: module 'math' has no attribute 'arcsin'	module ที่เรียก ไม่มีฟังก์ชันที่ต้องการ เช่น <code>print(math.arcsin(0))</code>	แก้ไขชื่อฟังก์ชันให้ถูกต้อง เช่น ในที่นี้ต้องใช้ <code>math.asin(0)</code>
ValueError: math domain error	มีการใส่ค่าที่ฟังก์ชันไม่รองรับ เช่น <code>print(math.asin(1000))</code> <code>print(math.log(10,0))</code>	ตรวจสอบค่าที่ฟังก์ชันนั้นรองรับ และใส่ค่าที่ถูกต้อง
invalid literal for int() with base 10	มีการใส่ค่าที่แปลงไม่ได้ลงไปใน <code>int()</code> เช่น <code>x = int('cat')</code>	ตรวจสอบค่าที่ต้องการแปลงว่าเป็นจำนวนเต็มหรือไม่ และแก้ไขให้ถูกต้อง
could not convert string to float	มีการใส่ค่าที่แปลงไม่ได้ลงไปใน <code>float()</code> เช่น <code>x = float('cat')</code>	ตรวจสอบค่าที่ต้องการแปลงว่าเป็นจำนวนทศนิยมหรือไม่ และแก้ไขให้ถูกต้อง
Can't convert 'int' object to str implicitly	มีการบวก <code>int</code> กับ <code>string</code> เช่น <code>print('cat'+3)</code>	แก้ไขให้ถูกต้อง อาจใช้ comma แทนเครื่องหมาย + หรือเพิ่มอัญประกาศเป็น '3'
sqrt() takes exactly one argument (3 given)	ฟังก์ชัน <code>sqrt()</code> ต้องใส่ข้อมูล 1 ค่า แต่ตอนนี้ใส่ไป 3 ค่า เช่น <code>x = sqrt(4,9,16)</code>	เปลี่ยนให้ใส่ข้อมูลตามจำนวนที่ฟังก์ชันกำหนด
ValueError: too many values to unpack	ค่าที่มิให้ มากกว่าจำนวนตัวแปร เช่น <code>a,b,c = 1,2,3,4,5</code>	แก้ไขจำนวนทั้งสองฝ่ายให้เท่ากัน
ValueError: not enough values to unpack	ค่าที่มิให้ น้อยกว่าจำนวนตัวแปร เช่น <code>a,b,c = 1,2</code>	แก้ไขจำนวนทั้งสองฝ่ายให้เท่ากัน
TypeError: unorderable types: int() < str()	มีการเปรียบเทียบจำนวนกับสตริง เช่น <code>if 9 < '9':</code>	แก้ไขประเภทข้อมูลให้ตรงกัน
'float' object cannot be interpreted as an integer	มีการใส่ค่า <code>float</code> ในส่วนที่ควรจะเป็นค่า <code>int</code> เช่น <code>for i in range(2.5):</code>	แก้ไขค่าให้เป็นจำนวนเต็ม

ข้อความ error	สาเหตุ	แนวทางการแก้ไข
'int' object is not iterable	มีการใช้คำสั่ง <code>in</code> กับสิ่งที่ไม่สามารถเดินแบบ <code>for</code> ได้ (<code>int</code> หรือ <code>float</code>) เช่น <code>for i in 5:</code>	ใช้ <code>in</code> กับสิ่งที่สามารถเดินแบบได้ เช่น <code>range</code> , <code>string</code> , <code>list</code> , <code>tuple</code> , <code>set</code> , <code>dict</code>
index out of range	มีการเรียก <code>index</code> เกินจากที่มี เช่น <code>x = 'cat'; print(x[100])</code> <code>y = [1,2,3]; print(y[100])</code>	ตรวจสอบว่า <code>index</code> ที่ต้องการเรียกไม่เกินความยาวที่มี
indices must be integers	มีการเรียก <code>index</code> ของสตริงหรือลิสต์ด้วยค่าที่ไม่เป็นจำนวนเต็ม เช่น <code>x = 'cat'; print(x[2/1])</code> <code>y = [1,2,3]; print(y['a'])</code>	ตรวจสอบว่า <code>index</code> ที่เรียก เป็นจำนวนเต็มหรือไม่ และแก้ไขให้ถูกต้อง
FileNotFoundException: No such file or directory	ต้องการเปิดไฟล์ แต่ไม่เจอกับไฟล์นั้น	ตรวจสอบว่ามีไฟล์อยู่จริง และพิมพ์ชื่อไฟล์ถูกต้อง
OSError: Invalid argument: 'D:\\x0cile.txt'	พิมพ์ชื่อไฟล์ด้วยเครื่องหมาย \ เช่น <code>f = open('D:\\file.txt')</code>	พิมพ์ชื่อไฟล์ด้วย \\ แทน เป็น <code>f = open('D:\\\\file.txt')</code>
KeyError	<code>dict</code> ไม่มี <code>key</code> ที่ต้องการ เช่น <code>d = {};</code> <code>print(d[1])</code> หรือ <code>set</code> ไม่มีสมาชิกที่ต้องการ เช่น <code>s = {1,2,3}; s.remove(4)</code>	ถูくる้าใช้ <code>key</code> ถูกต้องหรือไม่ เพิ่ม code เพื่อตรวจสอบว่ามี <code>key</code> อยู่ใน <code>dict</code> หรือไม่
'set' object has no attribute 'delete' (อาจเกิด error กับ <code>list</code> , <code>string</code> , <code>tuple</code> และ <code>dict</code> ได้ด้วย)	เรียกใช้บริการที่ไม่มีกำหนดไว้ เช่น <code>s = {1,2,3}; s.delete(1)</code> ซึ่ง <code>set</code> ไม่มีบริการที่ชื่อว่า <code>delete</code>	แก้ไขการใช้คำสั่งให้ถูกต้อง
TypeError: unhashable type	มีการใช้ <code>list</code> , <code>set</code> , <code>dict</code> เป็น <code>key</code> ของ <code>dict</code> หรือเก็บ <code>list</code> , <code>set</code> , <code>dict</code> ใน <code>set</code> เช่น <code>d = {};</code> <code>d[{1}] = 'cat'</code>	ตรวจสอบประเภทข้อมูลที่นำมาใช้ว่าไม่ใช่ <code>list</code> , <code>set</code> , <code>dict</code>
unsupported operand type(s) for +: 'NoneType' and 'int'	มีการนำ <code>None</code> ไปประมวลผล เช่น <code>def func(x):</code> <code>x += 1</code> <code>print(func(3)+1)</code> ซึ่งอาจเกิดจากการลืม <code>return</code>	<code>return</code> ค่าให้ถูกต้อง
'int' object is not callable	มีการเรียกฟังก์ชันด้วยตัวแปร เช่น <code>a = 3</code> <code>b = a(5)</code> อาจเกิดจากการตั้งชื่อฟังก์ชันซ้ำกับตัวแปร	ตรวจสอบการตั้งชื่อฟังก์ชันและตัวแปร และแก้ไขให้ถูกต้อง
RecursionError: maximum recursion depth exceeded	มีการเรียกฟังก์ชันตัวเองซ้ำมากเกินไป อาจเกิดจากการลืมกรณีพัฒนาของ การ recursive เช่น <code>def f(n):</code> <code>return n*f(n-1)</code>	เติมกรณีพิเศษของ การ recursive และเช็คเงื่อนไขการหยุดทำงานให้ถูกต้อง

ក្រសួងបច្ចេកទេស
Intania ៩៦

Solutions Engineer, Facebook และ บายกสมาคมโปรแกรมเมอร์ไทย



พ มคิดว่าการเรียนเขียนโปรแกรมจะเป็นเหมือนการเรียนภาษาที่ 3 ในอนาคตอันใกล้ ยิ่งเทคโนโลยีแห่งรุ่มปีในทุกสิ่งเรื่องเท่าไหร่ ความต้องการคนที่เข้าใจและควบคุมเทคโนโลยีได้ก็จะมีมากขึ้นเท่านั้น นอกจากนั้นการเรียนเขียนโปรแกรมยังช่วยให้เราสร้างคิดแก้ปัญหาอย่างเป็นระบบ และให้พลังในการสร้างสรรค์สิ่งต่าง ๆ ได้อย่างไม่รู้จบ

ນ້າມສະເໜີ ສືບສຸກກຸລ Intania 95

Intania 95

ตลาดหลักทรัพย์แห่งประเทศไทย



๗ เทียบการสร้าง software กับการสร้างตึกตึกหนึ่ง **programming** ก็เหมือนเป็นเสาเข็ม ที่รองรับแรงของหั้งตึกให้ดั้งอยู่ได้ และไม่พังทลาย ตรงๆ ให้ไม่มีเสาเข็ม ก็ไม่สามารถสร้างตึกได้ ดังนั้น **programming** ก็เป็นเหมือนฐานที่สำคัญของ software ถ้าเรามีทักษะ **programming** ที่ดี เราจะสามารถพัฒนา software ที่มีคุณภาพได้เช่นกัน

<h3>คำสั่งพื้นฐาน</h3> <p><code>x = int(input())</code> รับข้อมูลจำนวนเต็ม 1 ตัว</p> <p><code>x,y,z = [float(e) for e in input().split()]</code> รับข้อมูลจำนวนจริง 3 จำนวนมาทางตัวอักษร กันคั่นด้วยช่องว่าง</p> <p><code>int(x), float(x), str(x):</code> คืนค่า x ที่ถูกเปลี่ยนเป็นเลขตัวอักษร เป็นจำนวนเต็ม จำนวนจริง และสตริง</p> <p><code>abs(n):</code> คืนค่าส่วนบวกของ n</p> <p><code>round(f):</code> คืนค่าจำนวนเต็มที่เกิดจากการปัดเศษจำนวนจริง f โดยถ้าเศษของ f มีค่าตัวทศนิยมมากกว่า 0.5 จะปัดขึ้น ถ้าไม่กว่า 0.5 จะปัดลง</p> <p><code>round(f,d):</code> คืนค่าจำนวนจริงที่เกิดจากการปัดเศษจำนวนจริง f โดยปัดให้เข้าจำนวนวัดเลขหลังจุดทศนิยม d หลัก</p> <p><code>range(start,stop [,step]):</code> หรือ <code>range(stop):</code> คืนค่าเป็นลิสต์ของตัวเลขตามลำดับตัวต่อๆ แต่ start ถึง stop-step และ เพิ่มขึ้นทีละ step (ถ้าไม่ระบุ start จะมีค่า 0 และ step จะมีค่า 1)</p> <p><code>enumerate(L):</code> คืนลิสต์ของ tuple (index, element) ของแต่ละ ข้อมูลในลิสต์ L</p> <p><code>len(a):</code> คืนค่าเป็นจำนวนข้อมูลใน a ซึ่ง a อาจเป็นลิสต์ ติกขันนารี เช่น ที่มาลิสต์ หรือ numpy array ก็ได้</p> <p><code>max(a), min(a):</code> คืนค่าที่มาก/น้อยที่สุดของข้อมูลใน a ซึ่ง a อาจเป็น ลิสต์ ติกขันนารี ทูเพิล หรือสตริงได้</p> <p>(numpy array ใช้ <code>np.max(a), np.min(a)</code>) ถ้า a เป็นติกขันนารี จะวนค่านามาทีละอย่างที่สุดของ key ของติกขันนารี</p> <p><code>type(a):</code> คืนค่าประเภทของ a เช่น <code>type([1,2])</code> ได้ <class 'list'></p> <p><code>list(), dict(), tuple(), set():</code> สร้างลิสต์/ติกขันนารี/ทูเพิล/เซต</p>	<pre>import math</pre> <p><code>math.exp(x):</code> คืนค่า e ยกกำลัง x</p> <p><code>math.cos(x):</code> คืนค่า cosine ของ x เเรเดียน</p> <p><code>math.sin(x):</code> คืนค่า sine ของ x เเรเดียน</p> <p><code>math.sqrt(x):</code> คืนค่ารากที่สองของ x</p> <p><code>math.log(x,base):</code> คืนค่าลอการิทึมของ x ฐาน base</p> <p><code>math.degrees(x):</code> แปลง x จากเรเดียนเป็นองศา</p> <p><code>math.radians(x):</code> แปลง x จากองศาเป็นเรเดียน</p> <p><code>math.pi, math.e:</code> ค่าคงที่ pi และ e</p>
<h3>string s</h3>	<pre>string s</pre>
	<p><code>s.lower():</code> คืนสตริงใหม่ที่มีค่าเท่ากับ s แต่เป็นตัวพิมพ์เล็กทั้งหมด</p>
	<p><code>s.upper():</code> คืนสตริงใหม่ที่มีค่าเท่ากับ s แต่เป็นตัวพิมพ์ใหญ่ทั้งหมด</p>
	<p><code>s.find(sub):</code> คืน index แรกสุดที่พบ sub ใน s ถ้าไม่พบคืนค่า -1</p>
	<p><code>s.find(sub,i):</code> คืน index แรกสุดที่พบ sub ใน s โดยเริ่มต้นที่ index i</p>
	<p><code>s.count(sub):</code> คืนจำนวนครั้งที่ sub ปรากฏในสตริง s</p>
	<p><code>s.split(sep):</code> คืนลิสต์ของสตริงที่แยกด้วย sep (หรือ space ถ้าไม่ระบุ)</p>
	<p><code>s.strip():</code> คืนสตริงใหม่ที่มีค่าเท่ากับ s แต่ตัด spaces หัวท้ายออก</p>
	<p><code>s.join(L):</code> คืนสตริงที่สร้างจากการนำแต่ละ element ในลิสต์ L มาต่อกัน</p>
	<p>โดยมี s เป็นตัวคั่นระหว่างข้อมูลที่ต้องกัน (L ต้องเป็นลิสต์ของสตริง)</p>
<h3>import numpy as np</h3>	<pre>import numpy as np</pre>
	<p><code>np.array(L):</code> คืนค่า numpy array ที่สร้างจากลิสต์ L</p>
	<p><code>np.arange(start,stop,step):</code> คืนอาร์เรย์ 1 มิติของจำนวนที่มีค่าตาม</p>
	<p>start,stop,step</p>
	<p><code>np.ones(shape):</code> คืนอาร์เรย์ที่มีค่า 1 ทั้งหมด มีขนาดตาม tuple shape</p>
	<p><code>np.zeros(shape):</code> คืนอาร์เรย์ที่มีค่า 0 ทั้งหมด มีขนาดตาม tuple shape</p>
	<p><code>np.identity(size):</code> คืนอาร์เรย์ขนาด size x size ซึ่งมีข้อมูลในแนว</p>
	<p>เดือนที่แรกจะเป็น 1 และค่านิ่งแห่งอื่น ๆ เป็น 0</p>
	<p><code>np.empty_like(a):</code> คืนอาร์เรย์ใหม่ที่มีขนาดเหมือน a แต่ไม่มีการกำหนดค่า</p>
	<p>ค่าข้อมูลในอาร์เรย์ใหม่</p>
	<p><code>np.add(a,b), np.subtract(a,b), np.multiply(a,b), np.divide(a,b):</code></p>
	<p>คืนค่าอาร์เรย์ใหม่ที่เป็นผลลัพธ์ของ operation ของ a และ b</p>
	<p><code>np.dot(a,b):</code> คืนอาร์เรย์ที่เป็นผลคูณแบบแมทริกซ์ของ a และ b</p>
	<p><code>np.sin(a), np.cos(a), np.sqrt(a), np.abs(a):</code></p>
	<p>คืนอาร์เรย์ที่มีค่าของข้อมูลในแต่ละตำแหน่งที่เป็นผลจากการเรียกฟังก์ชัน</p>
	<p>sine, cosine, sqrt, abs กับข้อมูลในอาร์เรย์ a ที่ตำแหน่งเดียวกัน</p>
	<p><code>np.max(a, axis), np.min(a, axis):</code> คืนอาร์เรย์ของค่ามาก/น้อยที่สุด</p>
	<p>ใน a ตาม axis ที่กำหนด</p>
	<p><code>np.argmax(a, axis), np.argmin(a, axis):</code> คืนอาร์เรย์ของ index</p>
	<p>ที่มีค่ามาก/น้อยที่สุดใน a ตาม axis ที่กำหนด ด้วยอ้างอิง</p>
	<p>a = np.array([[2, 4, 6], [8, 10, 12]])</p>
	<p><code>np.max(a):</code> คืนค่า 12, <code>np.argmax(a):</code> คืนค่า 5</p>
	<p><code>np.max(a, axis=0):</code> คืนค่า array([8,10,12])</p>
	<p><code>np.argmax(a, axis=0):</code> คืนค่า array([1,1,1])</p>
	<p><code>np.argmax(a, axis=1):</code> คืนค่า array([2,2])</p>
	<p><code>np.sum(), np.std(), np.mean():</code> มีการใช้งานเหมือน np.max()</p>
	<p><code>np.ndenumerate(a):</code> คืนลิสต์ของ tuple (position,element) ของ</p>
	<p>แต่ละข้อมูลใน a โดย position เป็น tuple ที่เก็บตำแหน่งของข้อมูล</p>

เปลี่ยนความเชี่ยวกรุขอยากเห็น มาสู่งานอาชีพ

ในอนาคต ประเทศไทยจำเป็นต้องมีบุคลากรทางด้านวิศวกรรมและเทคโนโลยีเพิ่มขึ้นอีกมาก เพื่อให้ทันต่อความต้องการ ด้วยเหตุนี้ เอสโซ่ และ บริษัทในเครือเอ็กซอนเมบิลในประเทศไทย จึงให้ความสนับสนุนโครงการเรียนรู้ที่เกี่ยวข้องกับคณิตศาสตร์และวิทยาศาสตร์ เพื่อเตรียมความพร้อมทั้งด้านศักยภาพและทักษะให้เยาวชนรุ่นใหม่ สำหรับงานในศตวรรษที่ 21 เพราะการส่งเสริมสนับสนุนเยาวชนของเราวันนี้ จะส่งผลถึงความก้าวหน้าของประเทศต่อไปในวันข้างหน้า

Energy lives here™

ExxonMobil

Mobil ESSO Mobil



Find job opportunities at jobs.exxonmobil.com



CONFIDENTIAL

“
ทุกอย่างใน **python** จะดูซอฟต์
เมื่อเป็นสีพ้าสเกต
”

- โปรแกรมเมอร์นิรนาม -

สนับสนุนการพิมพ์โดย

ExxonMobil

ISBN 978-616-407-189-6



9 786164 071896