

CitizenID

เลขบัตรประจำตัวประชาชนของคนไทยประกอบด้วยเลข 13 หลัก กำหนดให้ n_0 คือเลขตัวซ้ายสุด ไล่ไปจนถึง n_{12} คือเลขตัวขวาสุด เลขตัวขวาสุดนี้มีค่าที่คำนวณได้จากเลข 12 ตัวทางซ้าย มีไว้เพื่อตรวจสอบว่า มีการป้อนเลขบัตรผิดหรือไม่ (ซึ่งตรวจได้ระดับหนึ่ง) ในวงการเรียกเลขนี้ว่า check digit มีสูตรการคำนวณดังนี้

$$n_{12} = (11 - (13n_0 + 12n_1 + 11n_2 + 10n_3 + 9n_4 + 8n_5 + 7n_6 + 6n_7 + 5n_8 + 4n_9 + 3n_{10} + 2n_{11})) \bmod 11 \bmod 10$$

จงเขียนโปรแกรมเพื่อหา check digit ของ เลข 12 หลักแรกของเลขบัตรประจำตัวประชาชน และแสดงเลขบัตรตามรูปแบบมาตรฐาน

ข้อมูลนำเข้า

เลข 12 หลักแรก (จากซ้าย) ของเลขที่บัตรประชาชน

ข้อมูลส่งออก

เลข 12 หลักที่รับมา พร้อมกับ เลข check digit ในรูปแบบมาตรฐานที่ปรากฏในบัตรประชาชน (ดูตัวอย่างประกอบ)

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
123456789012	1 2345 67890 12 1
310030011214	3 1003 00112 14 2
110070234512	1 1007 02345 12 9

Arabic numerals

จงเขียนโปรแกรมที่อ่านเลข 1 ตัว แล้วแสดงคำอ่านในภาษาอังกฤษ ตามตารางข้างล่างนี้

0	1	2	3	4	5	6	7	8	9
zero	one	two	three	four	five	six	seven	eight	nine

ข้อมูลนำเข้า

เลข 1 ตัว

ข้อมูลส่งออก

คำอ่านเลขที่ได้รับในภาษาอังกฤษ

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
5	5 --> five
1	1 --> one

US Date

จงเขียนโปรแกรมที่อ่านวันเดือนปีในรูปแบบ เลขวัน/เลขเดือน/เลขปี เพื่อเปลี่ยนและแสดงในรูปแบบ ชื่อเดือน เลขวัน, เลขปี

ข้อมูลนำเข้า

วันที่ในรูปแบบ เลขวัน/เลขเดือน/เลขปี

ข้อมูลส่งออก

วันที่ในรูปแบบ ชื่อเดือน เลขวัน, เลขปี

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
12/1/2019	January 12, 2019
31/12/2020	December 31, 2020

นำหน้าด้วย 0

จงเขียนโปรแกรมรับจำนวนเต็มบวก **M** กับ **N** เพื่อแสดงค่า **M** ทางจอภาพ โดยถ้าค่า **M** มีจำนวนหลักน้อยกว่า **N** ก็ให้เติม 0 ด้านซ้ายให้จำนวนหลักทั้งหมดครบ **N** ตัว เช่น ถ้า **M=123, N=5** จะแสดง **00123** ถ้า **M=12345, N=3** จะแสดง **12345**

ข้อมูลนำเข้า

จำนวนเต็มบวก 2 จำนวน สำหรับ **M** กับ **N** จำนวนละบรรทัด

ข้อมูลส่งออก

ค่าของ **M** ที่อาจมีการเติม 0 ด้านซ้ายเพื่อให้มีจำนวนหลักอย่างน้อย **N** หลัก

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
123 5	00123
123 3	123
123 2	123

ข้อแนะนำ: Python มี built-in function ชื่อ **max** มีไว้หาค่ามากที่สุดของข้อมูล เช่น **max (5 , 9)** ได้ 9

ยอดขายทั้งสัปดาห์

จงเขียนโปรแกรมรับยอดขายของแต่ละวันในหนึ่งสัปดาห์ เพื่อหาและแสดงยอดขายรวมของทั้งสัปดาห์

ข้อมูลนำเข้า

จำนวนเต็ม 7 ตัว เรียงกันในบรรทัดเดียว คั่นด้วยช่องว่าง

ข้อมูลส่งออก

แสดงผลรวมของจำนวนทั้ง 7 ที่รับเข้ามา

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
10 20 30 40 50 60 70	280
0 0 0 100 0 0 0	100

ผลบวกเวกเตอร์ 3 มิติ

ให้ $\mathbf{u} = [u_1, u_2, u_3]$ และ $\mathbf{v} = [v_1, v_2, v_3]$ เป็นเวกเตอร์ ผลบวกของเวกเตอร์ทั้งสองมีค่าเท่ากับ

$$\mathbf{u} + \mathbf{v} = [u_1 + v_1, u_2 + v_2, u_3 + v_3]$$

จงเขียนโปรแกรมรับเวกเตอร์สามมิติสองตัว แล้วแสดงผลบวกของเวกเตอร์ทั้งสอง

ข้อมูลนำเข้า

เวกเตอร์สามมิติสองตัว บรรทัดละตัว ในรูปแบบ [จำนวน, จำนวน, จำนวน]

ข้อมูลส่งออก

ผลบวกเวกเตอร์ในรูปแบบเดียวกับที่รับ (ดูตัวอย่างข้างล่าง)

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
[1, 2, 3] [2, 3, 4]	[1.0, 2.0, 3.0] + [2.0, 3.0, 4.0] = [3.0, 5.0, 7.0]
[1.25, 2.5, 3.0] [-2.5, 1.3, -2.5]	[1.25, 2.5, 3.0] + [-2.5, 1.3, -2.5] = [-1.25, 3.8, 0.5]
[0, 10, 0] [10, 0, 10]	[0.0, 10.0, 0.0] + [10.0, 0.0, 10.0] = [10.0, 10.0, 10.0]

ข้อแนะนำ: ถ้า `d = [1.2, 3.4, 5.6]` เป็นลิสต์ คำสั่ง `print(d)` จะได้แสดง `[1.2, 3.4, 5.6]` ทางจอภาพ

Decimal → Fraction

เราสามารถเขียนจำนวนตรรกยะในรูปแบบเศษส่วนหรือแบบทศนิยมได้ เช่น $\frac{1}{8} = 0.125$ แต่ก็มีจำนวนตรรกยะที่เขียนออกมาได้เป็นเลขหลังจุดทศนิยมไม่รู้จบแบบซ้ำ เช่น $\frac{3221}{555} = 5.8036036036036036...$ (เลข 036 จะซ้ำไปเรื่อย ๆ ไม่รู้จบ) ในกรณีนี้ ขอเขียนเป็น 5.8(036) แสดงให้เห็นว่า เลขในวงเล็บ 036 จะซ้ำไม่รู้จบ จึงเขียนโปรแกรมที่รับจำนวนในรูปแบบทศนิยม แล้วแสดงในรูปแบบเศษส่วน

ข้อมูลนำเข้า

จำนวนไม่ติดลบแบบทศนิยม ที่แบ่งทศนิยมเป็นสามส่วนคั่นด้วยจุลภาคคือ เลขหน้าจุด เลขหลังจุดที่ไม่อยู่ในวงเล็บ และเลขในวงเล็บ (ดูตัวอย่าง)

ข้อมูลส่งออก

จำนวนในรูปแบบเศษส่วนที่มีค่าเดียวกับจำนวนที่รับเข้ามา โดยที่ค่าของเศษและส่วนมี ห.ร.ม. เป็น 1 (ดูตัวอย่าง)

ตัวอย่าง

จำนวนในรูปแบบทศนิยม	input (จากแป้นพิมพ์)	output (ทางจอภาพ)
7.	7,,0	7 / 1
0.	0,,0	0 / 1
0.5	0,5,0	1 / 2
0.08(3)	0,08,3	1 / 12
0.02(27)	0,02,27	1 / 44
123.456(789)	123,456,789	41111111 / 333000
987.(987)	987,,987	329000 / 333

พยายามเขียน code โดยใช้เฉพาะคำสั่งในบทที่ 2 (คือไม่ใช่คำสั่ง if ...)

ข้อแนะนำ

เราสามารถใช้บริการ `math.gcd(a,b)` ในการหา ห.ร.ม ของ `a` กับ `b` เช่น คำสั่ง `math.gcd(2431, 13277)` ได้ผลเป็น 187 ดังนั้น

$$\frac{2431}{13277} = \frac{2431/187}{13277/187} = \frac{13}{71}$$