CitizenID

เลขบัตรประจำตัวประชาชนของคนไทยประกอบด้วยเลข 13 หลัก กำหนดให้ n_0 คือเลขตัวซ้ายสุด ไล่ไปจนถึง n_{12} คือเลขตัวขวาสุด เลขตัวขวา สุดนี้มีค่าที่คำนวณได้จากเลข 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 ในรูปแบบมาตรฐานที่ปรากฏในบัตรประชาขน (ดูตัวอย่างประกอบ)

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

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

US Date

จงเขียนโปรแกรมที่อ่านวันเดือนปีในรูปแบบ <mark>เลขวัน/เลขเดือน/เลขปี</mark> เพื่อเปลี่ยนและแสดงในรูปแบบ <mark>ชื่อเดือน เลขวัน, เลขปี</mark>

ข้อมูลนำเข้า

วันที่ในรูปแบบ <mark>เลขวัน/เลขเดือน/เลขปี</mark>

ข้อมูลส่งออก

วันที่ในรูปแบบ <mark>ชื่อเดือน เลขวัน, เลขปี</mark>

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 ด้านซ้ายให้จำนวน หลักทั้งหมดครบ **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	00123
5	
123	123
3	
123	123
2	

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

ยอดขายทั้งสัปดาห์

-จงเขียนโปรแกรมรับยอดขายของแต่ละวันในหนึ่งสัปดาห์ เพื่อหาและแสดงยอดขายรวมของทั้งสัปดาห์

ข้อมูลนำเข้า

จำนวนเต็ม 7 ตัว เรียงกันในบรรทัดเดียว คั่นด้วยช่องว่าง

ข้อมูลส่งออก

แสดงผลรวมของจำนวนทั้ง 7 ที่รับเข้ามา

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

ผลบวกเวกเตอร์ 3 มิติ

ให้ ${f u}=[u_1,u_2,u_3]$ และ ${f v}=[v_1,v_2,v_3]$ 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]	[1.0, 2.0, 3.0] + [2.0, 3.0, 4.0] = [3.0, 5.0, 7.0]
[2, 3, 4]	
[1.25, 2.5, 3.0]	[1.25, 2.5, 3.0] + [-2.5, 1.3, -2.5] = [-1.25, 3.8, 0.5]
[-2.5, 1.3, -2.5]	
[0, 10, 0]	[0.0, 10.0, 0.0] + [10.0, 0.0, 10.0] = [10.0, 10.0, 10.0]
[10, 0, 10]	

<u>ข้อแนะนำ</u>: ถ้า d = [1.2, 3.4, 5.6] เป็นลิสต์ คำสั่ง print(d) จะได้แสดง [1.2, 3.4, 5.6] ทางจอภาพ

ถอดรหัสลับ

นาย ก ต้องการส่งรหัสลับให้นาย ข โดยส่งเป็นเลข 0 ถึง 9 จำนวน 32 ตัว แล้วตกลงกับนาย ข ว่า วิธีถอดรหัสลับเป็นดังนี้ (ดูตัวอย่างประกอบด้วย)

กำหนดให้ เลขซ้ายสุดคือหลักที่ 1

- 1 หยิบเลขหลักที่ 4,11,18,25 และ 32 (คือเริ่มที่ 4 แล้วข้ามไปทีละ
 7) มาเขียนติดกัน
- 2 หยิบเลขหลักที่ 8,13,18,23,28 (คือเริ่มที่ 8 แล้วข้ามไปทีละ 5) มา เขียนติดกัน
- 3 นำเลขทั้งสองที่ได้จาก 2 ข้อแรกมาบวกกันแล้วบวกอีก 10000
- 4 เลือกเฉพาะหลักพัน หลักร้อย และหลักสิบของจำนวนที่ได้ในข้อ 3
- 5 นำแต่ละหลักของจำนวนในข้อ 4 มารวมกัน เลือกหลักหน่วย แล้วเพิ่มอีก 1
- 6 แปลงเลขที่ได้ในข้อที่ 5 เป็นตัวอังกฤษตัวใหญ่ โดย 1 คือ A, 2 คือ B, 3 คือ C, ..., 9 คือ I, 10 คือ J
- 7 รหัสลับที่ต้องการส่งคือ เลขจากข้อที่ 4 ต่อด้วยตัวอักษรจากข้อที่ 6

```
12345678901234567890123456789012 <-- หลักที่
92813912398100282033745980018127 <-- ข้อมูล
928<mark>1</mark>391239<mark>8</mark>100282<mark>0</mark>337459<mark>8</mark>0018127 ได้ 18087
```

$$18087 + 20051 + 10000 = 48138$$

$$813 --> 8 + 1 + 3 = 12 --> 2 + 1 = 3$$

813C

ข้อมูลนำเข้า

ตัวเลขจำนวน 32 หลัก

ข้อมูลส่งออก

รหัสที่ถอดได้จากข้อมูลนำเข้าด้วยวิธีที่กำหนดให้ข้างต้น

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
92813912398100282033745980018127	813C
000000000000000000000000000000000000000	A000
9999999999999999999999999	999н

Decimal → Fraction

เราสามารถเขียนจำนวนตรรกยะในรูปแบบเศษส่วนหรือแบบทศนิยมได้ เช่น $\frac{1}{8}=0.125$ แต่ก็มีจำนวนตรรกยะที่เขียนออกมาได้เป็นเลขหลังจุด ทศนิยมไม่รู้จบแบบซ้ำ เช่น $\frac{3221}{555}=5.8036036036036036036...$ (เลข 036 จะซ้ำไปเรื่อย ๆ ไม่รู้จบ) ในกรณีนี้ ขอเขียนเป็น 5.8(036) แสดงให้ เห็นว่า เลขในวงเล็บ 036 จะซ้ำไม่รู้จบ จงเขียนโปรแกรมที่รับจำนวนในรูปแบบทศนิยม 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 พยายามเขียน code โดยใช้เฉพาะคํ	ำสั่ง
0.5	0,5,0	1 / 2 ในบทที่ 2 (คือไม่ใช้คำสั่ง if)	
0.08(3)	0,08,3	1 / 12	l
0.02(27)	0,02,27	1 / 44	
123.456(789)	123,456,789	4111111 / 333000	
987.(987)	987,,987	329000 / 333	

ข้อแนะนำ

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

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