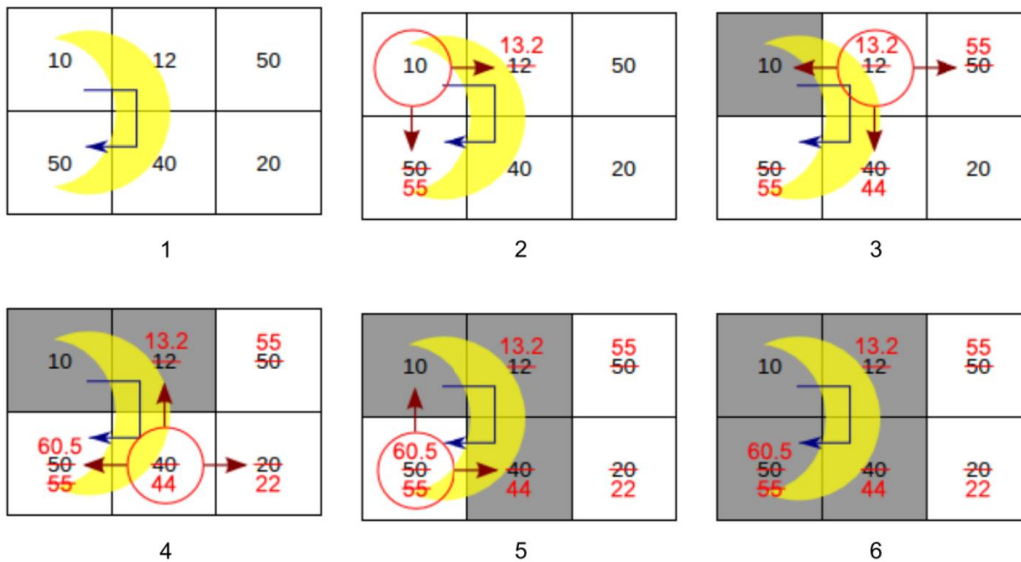


ชื่อ-สกุล _____ รหัสสนิต _____

1. [huangjui2] ฮวงจุ้ยเศรษฐี

เศรษฐีต้องการซื้อที่ดิน 4 แปลงตามฮวงจุ้ยที่ซินแสจินแนะนำ โดยต้องซื้อที่ดินที่ละแปลงติดกันจำนวน 4 แปลง เป็นรูปพระจันทร์เสี้ยวหันซ้าย โดยสามารถซื้อเป็นลำดับวนตาม หรือ ทวน เข็มนาฬิกาเท่านั้น แต่ผู้ขายที่ดินทราบว่าจะมีเศรษฐีมาซื้อที่ดินจึงตกลงร่วมกันว่า จะเพิ่มราคาที่ดินที่ติดกับที่ดินที่ถูกซื้อไปทั้ง 4 ทิศ (เหนือ ใต้ ตะวันออก ตะวันตก) 10% เสมอ ให้นิสิตเขียนโปรแกรมเพื่อช่วยให้เศรษฐีซื้อที่ดิน 4 แปลงที่ติดกันเป็นรูปพระจันทร์เสี้ยวตามที่ซินแสจินให้คำแนะนำไว้ ได้ในราคาถูกที่สุด



*** จากรูปด้านบนเป็นตัวอย่างการคำนวณค่าใช้จ่ายเมื่อเลือกซื้อที่ดินตามคำแนะนำของซินแส จากที่ดิน 6 แปลงที่อยู่ติดกันแบบลิส 2x3 ซึ่งหากซื้อณลักษณะนี้ จะมีค่าใช้จ่าย $10 + 13.2 + 44 + 60.5 = 127.7$ (แต่หากซื้อบริเวณเดียวกันแต่ซื้อตามลำดับทวนเข็มนาฬิกาจะมีค่าใช้จ่าย $50 + 44 + 13.2 + 12.1 = 119.3$) ***

ข้อมูลนำเข้าเป็นตารางของจำนวนเต็มบวกแทนราคาที่ดินแต่ละผืน รับประกันว่าที่ดินจะมีรูปร่างเป็นสี่เหลี่ยมผืนผ้า หรือ จตุรัสเท่านั้น

ข้อมูลนำออกเป็นราคาที่น้อยที่สุดที่เศรษฐีต้องจ่ายเงินเพื่อซื้อที่ดินตามที่ซินแสจินแนะนำ

Sample Output 1

```
10 12 50
50 40 20

119.3
```

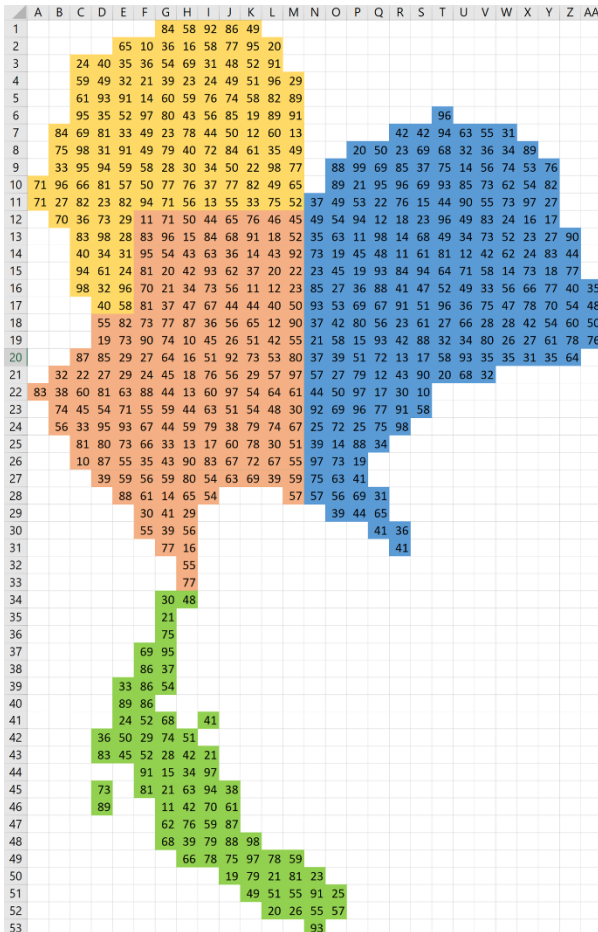
Sample Output 2

```
90 31 14 67 84
71 25 43 60 64
11 97 13 92 28
73 30 47 56 54
23 55 20 58 16
33 49 95 90 22
42 55 38 52 36

123.95
```

2. [census] สำมะโนประชากร

แผนที่ข้างล่างเป็นแผนที่ของประเทศหนึ่ง ซึ่งสามารถแบ่งออกเป็น 4 ภูมิภาคตามสีที่กำหนดให้ ได้แก่ ภาคเหนือ (สีเหลือง) ภาคตะวันออกเฉียงเหนือ (สีน้ำเงิน) ภาคกลาง (สีส้ม) และภาคใต้ (สีเขียว) โดยตัวเลขที่ระบุไว้ในแผนที่คือจำนวนประชากร (หมื่นคน) ที่อยู่ในขอบเขตรูปร่างสี่เหลี่ยมจัตุรัสขนาด 10 ตารางกิโลเมตร อยากทราบว่าภูมิภาคไหนมีจำนวนประชากรมากที่สุด และมีความหนาแน่น (จำนวนประชากรต่อตารางกิโลเมตร) เป็นเท่าไรตอบเป็นทศนิยม 2 ตำแหน่ง ข้อมูลเก็บอยู่ในไฟล์ /tmp/census.txt ในเครื่อง nobita.mikelab.net ซึ่งสามารถใช้ฟังก์ชัน load_census_file ตามที่กำหนดไว้ข้างล่างเพื่อเรียกใช้งานไฟล์ดังกล่าว



```
def load_census_file():
    lines = open('/tmp/census.txt').read().splitlines()
    data = []
    for line in lines:
        line_list = line.split(',')
        int_list = []
        for i in line_list:
            if i == '':
                int_list.append(0)
            else:
                int_list.append(int(i))
        data.append(int_list)
    return data
```

ข้อมูลนำออก 3 บรรทัด

บรรทัดแรก แสดงภูมิภาคที่มีประชากรมากที่สุดโดยแสดงเป็น North North-East Middle หรือ South

บรรทัดที่สอง แสดงจำนวนประชากรของภาคนั้น

บรรทัดที่สาม แสดงความหนาแน่นของภูมิภาคนั้น

3. [court] สร้างสนามบาส

เนื่องจากสนามบาสตรงข้ามภาควิชาวิศวกรรมคอมพิวเตอร์มักจะเต็มไปด้วยผู้คน หากต้องการเล่นก็ต้องรอคิว ด้วยความที่ไซโต้ไม่
 ยากรอคิวจึงตัดสินใจสร้างสนามบาสเอง โดยเขาได้สำรวจที่ดินแล้วพบว่าที่ดินที่เขา มีนั้นบางส่วนเป็นแอ่งน้ำทำให้เขาไม่สามารถ
 นำมาใช้สร้างสนามบาสได้จึงเขียนโปรแกรมเพื่อสร้างสนามบาสขนาด 2×4 ในแนวนั่ง โดยที่สนามบาสจะไม่ติดขอบของพื้นที่
 เพราะไซโต้มักชู้ตลูกบาสออกเป็นประจำ กำหนดให้ 0 แทนบริเวณที่ไม่สามารถสร้างสนามได้ และ 1 แทนบริเวณที่สร้างสนามได้
 จากตัวอย่างด้านล่าง บริเวณสีเขียวคือบริเวณที่สามารถสร้างสนามได้ และ บริเวณสีแดงคือบริเวณที่ไม่สามารถสร้างสนามได้

0	0	0	0
0	1	0	1
0	1	1	1
1	1	1	1
0	1	1	1
0	1	1	1
0	1	1	0
0	0	0	0

0	0	0	0
0	1	0	1
0	1	1	1
1	1	1	1
0	1	1	1
0	1	1	1
0	1	1	0
0	0	0	0

0	0	0	0
0	1	0	1
0	1	1	1
1	1	1	1
0	1	1	1
0	1	1	1
0	1	1	0
0	0	0	0

0	0	0	0
0	1	0	1
0	1	1	1
1	1	1	1
0	1	1	1
0	1	1	1
0	1	1	0
0	0	0	0

Sample Output 1

Length: 8
 Width: 4
0 0 0 0
0 1 0 1
0 1 1 1
1 1 1 1
0 1 1 1
0 1 1 1
0 1 1 0
0 0 0 0
 2 possible court(s)
 0 0 0 0
 0 1 0 1
 0 x x 1
 1 x x 1
 0 x x 1
 0 x x 1
 0 1 1 0
 0 0 0 0

Sample Output 1 (ต่อ)

0 0 0 0
 0 1 0 1
 0 1 1 1
 1 x x 1
 0 x x 1
 0 x x 1
 0 x x 0
 0 0 0 0

Sample Output 2

Length: 7
 Width: 5
0 1 0 0 0
0 1 0 1 1
0 0 1 1 1
1 1 0 1 1
0 1 0 1 1
0 0 1 1 1
0 1 0 0 0
 0 possible court(s)

4. [bifid_cipher] Bifid cipher

ไซโตได้ถูกจับเข้าสถานกักกัน แต่ไซโตจะสามารถสื่อสารกับคนภายนอกได้ตามปกติ ไม่ว่าจะเป็นการส่งข้อความ, โทร แต่จะมีผู้คอยตรวจพบทศนาและข้อความตลอด เนื่องจากไซโตไม่ต้องการให้ผู้คุมได้รับทศนาของเขา เขาจึงได้ทำการเข้ารหัสข้อความของเขาด้วย Bifid cipher

Bifid cipher เป็นวิธีการเข้ารหัสวิธีหนึ่ง โดยเริ่มจากการสร้างตารางขนาด 5x5 ซึ่งในแต่ละช่องจะบรรจุตัวอักษรภาษาอังกฤษ A-Z โดย I และ J จะอยู่ในช่องเดียวกันดังตารางที่ 1 จากนั้นจึงแทนค่า ตัวอักษรแต่ละตัวในข้อความด้วยเลขของแถวและคอลัมน์ตามลำดับโดยจะตัดตัวอักษรพิเศษทิ้ง (!, #, @, ' , " , , .) เช่น

“I’m Saito” แทนด้วย 24 32 43 11 24 44 34

จากนั้นนำตัวเลขที่ได้มาเขียนในแนวตั้งตามตารางที่ 2 แล้วอ่านตัวเลขจากซ้ายไปขวา และจับคู่เลขที่อยู่ติดกัน จากตารางที่ 2 จะได้

23 41 24 34 23 14 44

และเมื่อนำตัวเลขที่ได้ไปเปลี่ยนเป็นตัวอักษรจากตารางที่ 1 จะได้ “hqiohdt”

- 23 -> แถวที่ 2 คอลัมน์ที่ 3 -> h
- 41 -> แถวที่ 4 คอลัมน์ที่ 1 -> q
- 24 -> แถวที่ 2 คอลัมน์ที่ 4 -> i / j (i หรือ j ก็ได้)

ให้นิสิตพัฒนาโปรแกรมเพื่อเข้ารหัสข้อความแบบ Bifid cipher และในกรณีที่เจอช่องที่ 24 (I หรือ J) ให้แสดงข้อความที่เป็นไปได้ทั้งหมด

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I/J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

ตารางที่ 1

I	m	S	a	i	t	o
2	3	4	1	2	4	3
4	2	3	1	4	4	4

ตารางที่ 2

Sample Output 1

Text : **I'm Saito**
"hqiohdt"

Sample Output 2

Text : **Hi**
"go"

Sample Output 3

Text : **hqiohdt**
"imsaito"
"imsajto"
"jmsaito"
"jmsajto"

5. [columnar_cipher] Columnar cipher

เนื่องจากผู้คุมสงสัยในข้อความที่ไซโต้ส่งโดยการเข้ารหัสด้วย Bifid cipher ผู้คุมจึงได้ให้นักเรียนวิศวกรรมคอมพิวเตอร์สร้างโปรแกรมถอดรหัส Bifid cipher ให้ เมื่อไซโต้รู้ดังนั้นก็เปลี่ยนรูปแบบการเข้ารหัสข้อความเป็นแบบ Columnar cipher โดยจะมีคำที่ต้องการเข้ารหัสและคีย์เวิร์ด เช่น

Text: Meet me at restroom second floor.

Keyword: python

ขั้นตอนการเข้ารหัสแบบ Columnar cipher

- สร้างตารางความกว้างเท่ากับ keyword และบรรจุตัวอักษรลงไปในแต่ละช่องจากซ้ายไปขวาทีละแถว โดยแทนช่องว่างด้วย 'x'
- อ่านตารางทีละคอลัมน์จากบนลงล่าง โดยเริ่มจากคอลัมน์ที่ลำดับตัวอักษรใน keyword มาก่อน (เริ่มจากคอลัมน์ h เพราะ h เป็นตัวอักษรลำดับแรก ตามด้วย n o p t y ตามลำดับ)

ตัวอย่างการอ่านตาราง

- คอลัมน์ h จะได้ temdx
- คอลัมน์ n จะได้ etelx
- คอลัมน์ o จะได้ mssfx
- คอลัมน์ p จะได้ marco
- คอลัมน์ t จะได้ eronr
- คอลัมน์ y จะได้ etooo

และนำมาเรียงต่อกันจะได้ temdxetelxmssfxmarcoeronretooo

keyword	p	y	t	h	o	n
Alphabetical order	4	6	5	1	3	2
m	e	e	t	m	e	
a	t	r	e	s	t	
r	o	o	m	s	e	
c	o	n	d	f	l	
o	o	r	x	x	x	

Sample Output 1

Text: Meet me at restroom second floor.
Keyword: python
"temdxetelxmssfxmarcoeronretooo"

Sample Output 2

Text: Good luck for your exam.
Keyword: cpe
"gdcooemoufyraxolkruxx"

6. [ninja] ยอดนินจาไซโต้

ไซโต้ได้รับการถ่ายทอดวิชานินจาจากตระกูลของเขาที่ญี่ปุ่น และเขาต้องการถ่ายทอดวิชานินจาให้แก่เพื่อนๆ จึงได้ให้เพื่อนของเขาอยู่ในห้องขนาด H แถว W คอลัมน์ โดยแต่ละคนจะมีดาวกระจายคนละ 1 อัน เมื่อเริ่มฝึก (เวลา: $t = 0$) ทุกคนจะขว้างดาวกระจายออกไปในทิศทางที่หันหน้าอยู่ ด้วยความเร็วเท่ากับ 1 (เคลื่อนที่ไป 1 ช่อง เมื่อ t เพิ่มขึ้น 1) หากดาวกระจายชนกัน (อยู่ในตำแหน่งเดียวกัน) ดาวกระจายเหล่านั้นจะหายไป

ให้นิสิตเขียนโปรแกรมเพื่อหาจำนวนช่องที่ดาวกระจายลอยผ่านไป โดยกำหนดข้อมูลนำเข้าดังนี้

บรรทัดที่ 1 W : จำนวนคอลัมน์

บรรทัดที่ 2 H : จำนวนแถว

บรรทัดที่ 3 N : จำนวนนินจา

บรรทัดที่ i ($4 \leq i \leq 3 + N$): X_i Y_i D_i โดยที่

X_i : ตำแหน่งของนินจาในแนวแกน x (คอลัมน์ที่อยู่ 1, 2, 3, ...)

Y_i : ตำแหน่งของนินจาในแนวแกน y (แถวที่อยู่ 1, 2, 3, ...)

D_i : ทิศทางที่หันหน้าอยู่ (r = right, l = left, a = above, b = below)

Sample Output 1

```
W: 5
H: 4
N: 5
3 3 l
3 2 r
4 2 l
5 4 a
1 1 b
11
```

Sample Output 2

```
W: 7
H: 6
N: 12
3 2 b
6 3 l
7 1 b
1 5 r
3 6 a
6 6 a
4 5 l
1 3 r
6 5 l
5 1 l
6 4 b
4 1 b
29
```

ตัวอย่างการเดินทางของดาวกระจายใน Sample Output 1

