

problem1 (example)

จงเขียนโปรแกรมเพื่อคำนวณค่า y จากค่า x และสูตรดังต่อไปนี้

$$y = 5x^2 - 20x + 2$$

ข้อมูลนำเข้า

จำนวนเต็ม x

ข้อมูลส่งออก

ค่า y ที่ได้จากสูตร

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
0	2
-1	27
10	302

problem2 example

จงเขียนโปรแกรมเพื่อรับค่าคะแนนและแสดงข้อความตามเงื่อนไขดังนี้

- หากคะแนนมีค่ามากกว่าหรือเท่ากับ 90 ให้แสดงข้อความ **passed with honors**
- หากคะแนนมีค่ามากกว่าหรือเท่ากับ 70 แต่น้อยกว่า 90 ให้แสดงข้อความ **passed with merit**
- หากคะแนนมีค่ามากกว่าหรือเท่ากับ 50 แต่น้อยกว่า 70 ให้แสดงข้อความ **passed**
- หากคะแนนมีค่าน้อยกว่า 50 ให้แสดงข้อความ **failed**

ข้อมูลนำเข้า

ค่าคะแนนในรูปแบบทศนิยม

ข้อมูลส่งออก

ข้อความตามเงื่อนไข

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
100.0	passed with honors
70	passed with merit
65.25	passed
0	failed
50.0	passed
38.8	failed

problem3 example

จงเขียนโปรแกรมที่

- รับค่า n ที่แทนจำนวนที่ต้องการแสดงผล ($0 \leq n \leq 1000$)
- รับค่า x ที่แทนค่าเริ่มต้น ($-1000 \leq x \leq 1000$)
- รับค่า m ที่แทนค่าจำนวนเท่าความแตกต่างของแต่ละจำนวน ($-100 \leq m \leq 100$)
- แสดงผลรวมของเลขทั้ง n ตัว

เช่น หาก n เป็น 4, x เป็น 2 และ m เป็น 3 ก็จะทำให้หมายถึงให้แสดงตัวเลข 4 ตัว เริ่มต้นที่ค่า 2 แล้วเพิ่มขึ้นทีละ 3 เท่าซึ่งก็คือ 2 6 18 54 และผลรวมก็คือ 80

ข้อมูลนำเข้า

บรรทัดแรกเป็นจำนวนเต็ม n

บรรทัดที่สองเป็นจำนวนเต็ม x

บรรทัดที่สามเป็นจำนวนเต็ม m

ข้อมูลส่งออก

n บรรทัดแรกเป็นจำนวนที่ให้แสดงผลบรรทัดละตัว

บรรทัดที่ $n+1$ เป็นผลรวมของเลขทั้ง n ตัว

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
4 2 3	2 6 18 54 80
10 -2 -2	-2 4 -8 16 -32 64 -128 256 -512 1024 682

Student Ranking

จงเขียนฟังก์ชันตามรายละเอียดใน comment

```
def calculate_average(scores):
    """
    รับ scores เป็นลิสต์ของคะแนนสอบ 4 ครั้ง เช่น [70, 72, 92, 85]

    คืน จำนวนจริง เป็นค่าคะแนนเฉลี่ยของคะแนนสอบทั้ง 4 ครั้ง
    เช่น calculate_average([70, 72, 92, 85]) คืน 79.75
        calculate_average([0, 0, 0, 0]) คืน 0.0
    """

def get_grade(average_score):
    """
    รับ average_score เป็นจำนวนจริงที่เป็นคะแนนเฉลี่ยของการสอบ

    คืน สตริง ที่เป็นเกรดที่เกิดจากเกณฑ์ดังนี้
    - มากกว่าหรือเท่ากับ 80 ได้ A
    - มากกว่าหรือเท่ากับ 75 แต่น้อยกว่า 80 ได้ B+
    - มากกว่าหรือเท่ากับ 70 แต่น้อยกว่า 75 ได้ B
    - มากกว่าหรือเท่ากับ 65 แต่น้อยกว่า 70 ได้ C+
    - มากกว่าหรือเท่ากับ 60 แต่น้อยกว่า 65 ได้ C
    - มากกว่าหรือเท่ากับ 55 แต่น้อยกว่า 60 ได้ D+
    - มากกว่าหรือเท่ากับ 50 แต่น้อยกว่า 55 ได้ D
    - น้อยกว่า 50 ได้ F

    เช่น get_grade(80.0) คืน 'A'
        get_grade(55.5) คืน 'D+'
        get_grade(0.0) คืน 'F'
    """

def process_student_data(stu_data):
    """
    รับ stu_data เป็นลิสต์ที่ประกอบด้วยสมาชิกสองตัว
    - สมาชิกแรกจะเป็นสตริงแทนชื่อนักเรียน
    - สมาชิกที่สองจะเป็นลิสต์ที่เก็บคะแนนสอบ 4 ครั้ง
    เช่น ['Nana', [70, 72, 92, 85]]

    คืนลิสต์ที่มีสองค่า
    - ค่าแรกเป็นสตริงแทนชื่อนักเรียน
    - ค่าที่สองเป็นสตริงแทนเกรดที่ได้จากการคำนวณคะแนนเฉลี่ย

    เช่น process_student_data(['Nana', [70, 72, 92, 85]]) คืน ['Nana', 'B+']
        process_student_data(['Malee', [78, 71, 89, 91]]) คืน ['Malee', 'A']
        process_student_data(['Lisa', [74, 71, 89, 91]]) คืน ['Lisa', 'A']
        process_student_data(['Usa', [71, 69, 75, 80]]) คืน ['Usa', 'B']
    """

def print_students_ranked_by_grades(list_of_stu_data):
    """
    รับ list_of_stu_data เป็นลิสต์ที่สมาชิกทุกตัวเป็น stu_data
    เหมือนที่ประกาศไว้ใน process_student_data

    ให้แสดงชื่อนักเรียนและเกรดที่ได้ โดยให้แสดงนักเรียนที่เกรดสูงที่สุดก่อน
    หากเกรดเท่ากัน ให้แสดงรายชื่อตามลำดับพจนานุกรม

    ฟังก์ชันนี้ไม่คืนค่า

    ตัวอย่าง ให้ list_of_stu_data = [['Nana', [70, 72, 92, 85]],
                                      ['Malee', [78, 71, 89, 91]],
                                      ['Lisa', [74, 71, 89, 91]],
                                      ['Usa', [71, 69, 75, 80]]]

    หากเรียก print_students_ranked_by_grades(list_of_stu_data) จะแสดงผลดังนี้
    Lisa A
    Malee A
    Nana B+
    Usa B
    """
    # ห้ามแก้ไขบรรทัดข้างล่างทั้งหมดนี้
    while (cmd:=input().strip()):
        exec(cmd)
        if cmd[-1]=='\n': break
```

ข้อมูลนำเข้า

คำสั่งภาษา Python ที่ต้องการให้ทำงาน

ข้อมูลส่งออก

ผลที่ได้จากคำสั่งที่ได้รับ

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
<pre>print(calculate_average([70, 72, 92, 85]));</pre>	79.75
<pre>print(type(calculate_average([0, 0, 0, 0]));</pre>	<class 'float'>
<pre>print(get_grade(55.5));</pre>	D+
<pre>print(type(get_grade(80.0)));</pre>	<class 'str'>
<pre>print(process_student_data(['Usa', [71, 69, 75, 80]]));</pre>	['Usa', 'B']
<pre>print(type(process_student_data(['Nana', [70, 72, 92, 85]]));</pre>	<class 'list'>
<pre>list_of_stu_data = [] list_of_stu_data.append(['Nana', [70,72,92,85]]) list_of_stu_data.append(['Malee', [78,71,89,91]]) list_of_stu_data.append(['Lisa', [74,71,89,91]]) list_of_stu_data.append(['Usa', [71,69,75,80]]) print_students_ranked_by_grades(list_of_stu_data);</pre>	Lisa A Malee A Nana B+ Usa B
<pre>list_of_stu_data = [] list_of_stu_data.append(['Somchai', [0,0,0,0]]) list_of_stu_data.append(['Somchai', [100,100,100,100]]) print_students_ranked_by_grades(list_of_stu_data);</pre>	Somchai A Somchai F
<pre>list_of_stu_data = [] list_of_stu_data.append(['A', [59, 58, 57, 56]]) list_of_stu_data.append(['A', [29, 28, 27, 26]]) list_of_stu_data.append(['B', [65, 66, 67, 68]]) list_of_stu_data.append(['C', [0, 15, 55, 1]]) list_of_stu_data.append(['D', [42, 52, 100, 0]]) list_of_stu_data.append(['E', [0, 0, 0, 0]]) list_of_stu_data.append(['F', [89, 88, 87, 86]]) list_of_stu_data.append(['GG', [80, 80, 80, 80]]) list_of_stu_data.append(['G', [80, 80, 80, 80]]) list_of_stu_data.append(['J', [79, 78, 77, 76]]) list_of_stu_data.append(['H', [75, 75, 75, 75]]) list_of_stu_data.append(['I', [79.9, 79.9, 79.9, 79.9]]) list_of_stu_data.append(['K', [74, 73, 72, 71]]) list_of_stu_data.append(['L', [70, 70, 70, 70]]) list_of_stu_data.append(['M', [69, 68, 67, 66]]) list_of_stu_data.append(['N', [65, 65, 65, 65]]) list_of_stu_data.append(['P', [60, 60, 60, 60]]) list_of_stu_data.append(['Q', [62, 63, 64, 61]]) list_of_stu_data.append(['R', [55, 55, 55, 55]]) list_of_stu_data.append(['S', [58, 58, 58, 58]]) list_of_stu_data.append(['T', [54, 54, 54, 54]]) list_of_stu_data.append(['U', [50, 50, 50, 50]]) list_of_stu_data.append(['V', [90, 90, 0, 0]]) list_of_stu_data.append(['W', [0, 1, 2, 3]]) list_of_stu_data.append(['X', [84, 53, 72, 63]]) list_of_stu_data.append(['Y', [49, 50, 50, 50]]) list_of_stu_data.append(['B', [100, 100, 100, 100]]) list_of_stu_data.append(['Z', [100, 100, 100, 100]]) print_students_ranked_by_grades(list_of_stu_data);</pre>	B A F A G A GG A Z A H B+ I B+ J B+ K B L B B C+ M C+ N C+ X C+ P C Q C A D+ R D+ S D+ T D U D A F C F D F E F V F W F Y F

ค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน

ค่าเฉลี่ย (μ) และค่าเบี่ยงเบนมาตรฐาน (σ) ของข้อมูล x_1, x_2, \dots, x_n หาได้จากสูตรข้างล่างนี้

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2}$$

ให้เขียนโปรแกรมรับข้อมูลชุดหนึ่ง แล้วแสดงค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของข้อมูลชุดนี้

ข้อมูลนำเข้า

บรรทัดแรกเป็นจำนวนข้อมูล n

n บรรทัดต่อมา เป็นข้อมูล (จำนวนจริง) บรรทัดละหนึ่งจำนวน

ข้อมูลส่งออก

ค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของข้อมูลที่ได้รับมาในรูปแบบที่แสดงในตัวอย่าง

ให้แสดงค่าทั้งสองในรูปแบบที่มีเลขหลังจุดทศนิยมมากที่สุด 5 ตำแหน่ง ด้วยการใช้ฟังก์ชัน **round** (ค่าที่ต้องการแสดง, 5)

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
4 1.5 1.5 1.5 1.5	average = 1.5 sd = 0.0
5 1.0 2.5 3.0 3.5 5.0	average = 3.0 sd = 1.30384

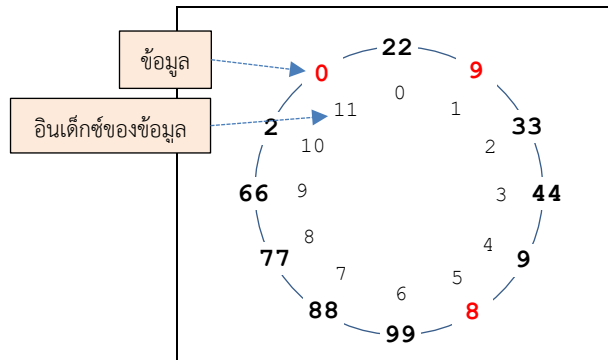
หาหลุมในลิสต์วงกลม

นิยาม "หลุม" ในลิสต์ คือ ข้อมูลในลิสต์ที่มีค่าน้อยกว่าตัวที่ติดกันด้านซ้ายและด้านขวา

เช่น [22, 9, 33, 44, 9, 8, 99, 88, 77, 66, 2, 0] มีหลุมที่อินเด็กซ์ 1 กับ 5

นิยาม "ลิสต์วงกลม" คือ ลิสต์ที่เสมือนว่าตัวซ้ายสุดกับตัวขวาสุดของลิสต์นั้นติดกัน

ดังนั้น ถ้าคิดว่า [22, 9, 33, 44, 9, 8, 99, 88, 77, 66, 2, 0] เป็นลิสต์วงกลม จะมีหลุมที่อินเด็กซ์ 1, 5 และ 11



จงเขียนโปรแกรมที่รับรายการของจำนวนที่แทนลิสต์วงกลม แล้วหาอินเด็กซ์ของหลุมทั้งหมดในลิสต์วงกลมนี้

ข้อมูลนำเข้า

บรรทัดแรกเป็นจำนวนเต็ม n ตามด้วยอีก n บรรทัด บรรทัดละจำนวนแทนข้อมูลในลิสต์วงกลมตามลำดับ เริ่มตั้งแต่อินเด็กซ์ที่ 0 เป็นต้นไป

ข้อมูลส่งออก

อินเด็กซ์ของหลุมทั้งหมดในลิสต์วงกลมที่ได้รับ เรียงอินเด็กซ์จากน้อยไปมาก อินเด็กซ์ละบรรทัด ถ้าไม่มีหลุมเลยให้แสดง **No pits**

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
3 9 9 9 <div style="margin-left: 100px;">← จำนวนข้อมูล</div>	No pits
5 1 2 3 2 1	No pits
4 11 22 33 44 <div style="margin-left: 100px;">←</div>	0
12 22 9 33 44 9 8 99 88 77 66 2 0 <div style="margin-left: 100px;">←</div>	1 5 11

Coding Contest

การแข่งขันเขียนโปรแกรมที่จัดขึ้นบนระบบเกรดเดอรัลครั้งหนึ่ง จัดอันดับทีมที่เข้าแข่งขันโดย

- ทีมที่สามารถแก้ปัญหาโจทย์ได้ถูกต้องมากกว่าจะได้อันดับที่ดีกว่า
 - หากมีจำนวนโจทย์ที่แก้ปัญหามาเท่ากัน ทีมที่มี **penalty** น้อยกว่าจะมีอันดับที่ดีกว่าโดย **penalty** จะคิดจาก
 1. เวลารวมที่ใช้ในการแก้ปัญหาโจทย์ที่ทำได้อีกต้อง โดยจะนับจากตั้งแต่เริ่มแข่ง และนับรวมทุกข้อ (หน่วยเป็นนาที)
 2. หากข้อที่ทำได้อีกต้องมีการส่งก่อนหน้าทีเกรดเดอรัลแจ้งว่าผิด จะคิดเวลาเพิ่มของข้อดังกล่าวไปครั้งละ 20 นาที
 3. หากทีมใดมีการส่งโค้ดของข้อที่ทีมนั้นเคยทำถูกแล้วเข้ามาอีก การส่งนี้จะไม่นำไปคิด **penalty** ไม่ว่าผลครั้งนี้จะถูกหรือไม่
- ตัวอย่าง การแข่งขันระหว่างทีม A และทีม B โดย Input จะประกอบด้วย เวลาที่ส่ง ข้อทีม หมายเลขโจทย์ และผลลัพธ์ โดยที่ผลลัพธ์จะมีค่าได้เป็น **T** หรือ **F** (**T** แปลว่าถูก และ **F** แปลว่าผิด)

input	A		B		คำอธิบาย
	#ข้อที่ได้	penalty ข้อ1+ข้อ2+ข้อ3+ข้อ4	#ข้อที่ได้	penalty ข้อ1+ข้อ2+ข้อ3+ข้อ4	
10 A 1 T	1	10+0+0+0			นาทีที่ 10 ทีม A ส่งโค้ดข้อ 1 และผลลัพธ์ถูกต้อง
23 A 3 F	1	10+0+0+0			นาทีที่ 23 ทีม A ส่งโค้ดข้อ 3 และผลลัพธ์ผิด
30 A 3 F	1	10+0+0+0			นาทีที่ 30 ทีม A ส่งโค้ดข้อ 3 และผลลัพธ์ผิด
45 A 2 F	1	10+0+0+0			นาทีที่ 45 ทีม A ส่งโค้ดข้อ 2 และผลลัพธ์ผิด
60 B 2 T			1	0+60+0+0	นาทีที่ 60 ทีม B ส่งโค้ดข้อ 2 และผลลัพธ์ถูกต้อง
65 A 3 T	2	10+0+(65+20+20)+0			นาทีที่ 65 ทีม A ส่งโค้ดข้อ 3 และผลลัพธ์ถูกต้อง
84 B 1 T			2	84+60+0+0	นาทีที่ 84 ทีม B ส่งโค้ดข้อ 1 และผลลัพธ์ถูกต้อง
115 B 4 T			3	84+60+0+115	นาทีที่ 115 ทีม B ส่งโค้ดข้อ 4 และผลลัพธ์ถูกต้อง
120 A 4 T	3	10+0+(65+20+20)+120			นาทีที่ 120 ทีม A ส่งโค้ดข้อ 4 และผลลัพธ์ถูกต้อง
125 B 4 F			3	84+60+0+115	นาทีที่ 125 ทีม B ส่งโค้ดข้อ 4 และผลลัพธ์ผิด
125 B 4 T			3	84+60+0+115	นาทีที่ 125 ทีม B ส่งโค้ดข้อ 4 และผลลัพธ์ถูกต้อง
125 A 4 F	3	10+0+(65+20+20)+120			นาทีที่ 125 ทีม A ส่งโค้ดข้อ 4 และผลลัพธ์ผิด

ทีม A มี penalty ของการคิดเวลารวมเท่ากับ $10+65+120 \rightarrow 195$ นาที และ penalty จากการทำผิดก่อนจะส่งได้ถูกต้อง 2 ครั้งเป็น $20+20$ นาที ฉะนั้นทีม A จะมี penalty รวม **235** นาที

ทีม B มี penalty ของการคิดเวลารวมเท่ากับ $60+84+115 \rightarrow 259$ นาที โดยที่ไม่มีผลลัพธ์ว่าผิดก่อนจะส่งได้ถูกต้องเลย ฉะนั้นทีม B จะมี penalty รวม **259** นาที

การแข่งขันนี้จะถือว่าทีม A ชนะทีม B เพราะว่าทั้งคู่ทำโจทย์ได้ 3 ข้อเท่ากัน จึงต้องไปวัดที่ penalty ซึ่งทีม A มี penalty น้อยกว่าโปรดสังเกตว่า

1. ผลลัพธ์ที่ผิดของทีม A ที่นาทีที่ 45 จะไม่นำมาคิด penalty เพราะไม่สามารถทำโจทย์ข้อ 2 ได้ถูกต้อง
2. การส่งของทีม A และ B ที่นาทีที่ 125 จะไม่นำมาคิด penalty เพราะว่าได้ทำโจทย์ข้อนั้นได้ถูกต้องไปแล้ว
3. ณ นาทีเดียวกัน อาจจะมีการส่งเข้ามาหลายครั้งได้ และอาจจะเป็นของทีมเดียวกันก็ได้
4. เวลาของ submission จะบรรทัดก่อนหน้าจะน้อยกว่าหรือเท่ากับของบรรทัดปัจจุบันเสมอ

ข้อมูลนำเข้า

บรรทัดแรกเป็นจำนวนเต็ม **N** ระบุจำนวน submissions ทั้งหมด

อีก **N** บรรทัดถัดมาเป็นนาที ข้อทีม หมายเลขโจทย์ และผลลัพธ์ คั่นด้วยช่องว่าง

ข้อมูลส่งออก

ทีมที่เป็นผู้ชนะ 3 อันดับแรกโดย

- แสดงผลบรรทัดละทีม โดยให้ทีมที่ผลการแข่งขันดีกว่ามาก่อน โดยให้แสดงชื่อทีม จำนวนโจทย์ที่ทำได้และ penalty คั่นด้วยช่องว่าง
- หากมีอย่างน้อยสองทีมที่ผลการแข่งขันเสมอกันให้เรียงลำดับด้วยชื่อทีมตามพจนานุกรม
- หากมีทีมลำดับที่ 3 หลายทีมให้แสดงทุกทีมที่ผลการแข่งขันเท่ากับทีมที่ 3

- หากมีทีมน้อยกว่า 3 ทีมให้แสดงผลเท่าที่มี
- จะแสดงเฉพาะทีมที่ทำได้อย่างน้อยหนึ่งข้อเท่านั้น

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)	คำอธิบาย
12 10 A 1 T 23 A 3 F 30 A 3 F 45 A 2 F 60 B 2 T 65 A 3 T 84 B 1 T 115 B 4 T 120 A 4 T 125 B 4 F 125 B 4 T 125 A 4 F	A 3 235 B 3 259	เหมือนตัวอย่างข้างบน
9 8 Nattee 1 F 10 Nattee 1 T 12 Somchai 7 T 13 Jessada 2 F 15 Chate 1 T 20 Kamonluk 2 T 35 Chate 7 F 45 Chate 7 T 50 Somchai 2 T	Somchai 2 62 Chate 2 80 Kamonluk 1 20	ถึงแม้ทีม Nattee จะได้ 1 ข้อเท่ากับทีม Kamonluk แต่ว่า penalty ของทีม Nattee มากกว่าทำให้ทีม Nattee ไม่อยู่ 3 อันดับแรก
7 3 J 1 F 5 J 1 T 10 D 1 T 15 C 1 F 20 C 1 T 25 K 2 T 25 B 3 T	D 1 10 B 1 25 J 1 25 K 1 25	ทีม K ได้จำนวนข้อและ penalty เท่ากับทีม J ที่ได้ลำดับที่ 3 จึงรวมทีม K เป็นผู้ชนะ 3 ลำดับแรกด้วย
4 10 D 1 T 15 C 1 F 20 K 2 F 20 J 3 F	D 1 10	ทีม C J K ทำโจทย์ไม่ถูกต้องเลยจึงไม่ถือเป็นผู้ชนะ

Items sale

เกม Comprog RPG เป็นเกมที่ไอเทมมีหน่วยเป็นทอง (G) โดยไอเทมแต่ละชนิดจะมีมูลค่าที่ขายได้ไม่เท่ากัน โจทย์ข้อนี้ให้หาว่า ถ้าเราเทขายไอเทมในเกมเพื่อไปซื้อไอเทมอีกชิ้นที่มีมูลค่า N ได้พอดีมันจะมีกี่วิธี

ตัวอย่าง สมมุติเรามีไอเทมมูลค่า 8G, 5G, 4G, 2G, 1G อย่างละ 3 ชิ้น และ N = 10G จะมีวิธีขายได้ 10 วิธีตามตารางข้างล่างนี้

8G	5G	4G	2G	1G
0	0	1	2	2
0	0	1	3	0
0	0	2	0	2
0	0	2	1	0
0	1	0	1	3
0	1	0	2	1
0	1	1	0	1
0	2	0	0	0
1	0	0	0	2
1	0	0	1	0

หาก N = 0G จะมีวิธีขายได้ 1 วิธี

8G	5G	4G	2G	1G
0	0	0	0	0

ถ้ายังทำไม่ได้สามารถอ่านคำแนะนำได้ที่ท้าย pdf

ข้อมูลนำเข้า

บรรทัดแรกเป็นจำนวนเต็ม $M N$ คั่นด้วยช่องว่าง $1 \leq M, N \leq 1000$

M บรรทัดถัดมาจะเป็นจำนวนเต็ม $V C$ คั่นด้วยช่องว่าง โดย V จะเป็นมูลค่าไอเทม ($1 \leq V \leq 200$) และ C เป็นจำนวนไอเทม ($0 \leq C \leq 1000$) และจะไม่มี V ที่มีค่าเท่ากัน

ข้อมูลส่งออก

จำนวนวิธีที่จะขายไอเทมให้ได้ค่า N ภายใต้เงื่อนไขที่เสนอข้างต้น

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
5 10 8 3 5 3 4 3 2 3 1 3 ตามตัวอย่าง	10
5 0 8 3 5 3 4 3 2 3 1 3 ตามตัวอย่าง	1
7 55 7 7 6 6 5 5 4 4 3 3 2 2 1 1 1 4	558
	1

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
2 7	
2 5 2 3 1 3	2
2 11 2 7 1 9	5
3 13 5 3 2 3 1 3	3
3 10 5 4 2 7 1 7	8
4 15 10 3 5 3 2 3 1 3	6
4 15 10 10 5 9 2 9 1 9	18
8 10 200 0 50 0 8 3 6 0 5 3 4 3 2 3 1 3	10
8 10 200 1000 50 1000 8 999 6 0 5 3 4 3 2 3 1 3	10

คำแนะนำเพิ่มเติม

สามารถใช้ข้อมูลข้างล่างนี้ช่วยเขียน base case ได้ หรือไม่ใช้ก็ได้

- ไอเทมมูลค่า 0 แลกได้ 1 วิธี
- ไอเทมมูลค่าลบ ไม่มีวิธีแลก
- ถ้าไม่มีไอเทมให้แลกเลย ก็ไม่มีวิธีแลก (ยกเว้นกรณีต้องการแลกมูลค่า 0)

ชุดข้อมูลทดสอบ

ข้อมูลต่อไปนี้อธิบายว่าข้อมูลทดสอบของโจทย์ข้อนี้มีลักษณะอย่างไรบ้าง นิสิตสามารถใช้ข้อมูลเหล่านี้ช่วยในการทำโจทย์ได้

10% M=2

10% M=3

10% M=4

70% ไม่มีเงื่อนไข