

1. 교점에 별 만들기 - Level 2

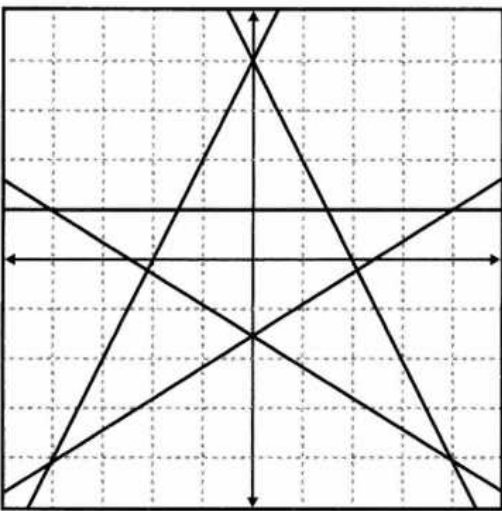
URL <https://programmers.co.kr/learn/courses/30/lessons/87377>

$Ax + By + C = 0$ 으로 표현할 수 있는 n 개의 직선이 주어질 때, 이 직선의 교점 중 정수 좌표에 별을 그리려 합니다

예를 들어 다음과 같은 직선 5개를

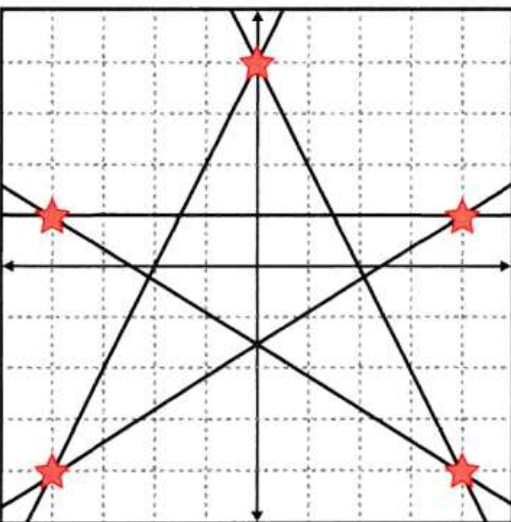
- $2x - y + 4 = 0$
- $-2x - y + 4 = 0$
- $-y + 1 = 0$
- $5x - 8y - 12 = 0$
- $5x + 8y + 12 = 0$

좌표 평면 위에 그리면 다음 그림과 같습니다.

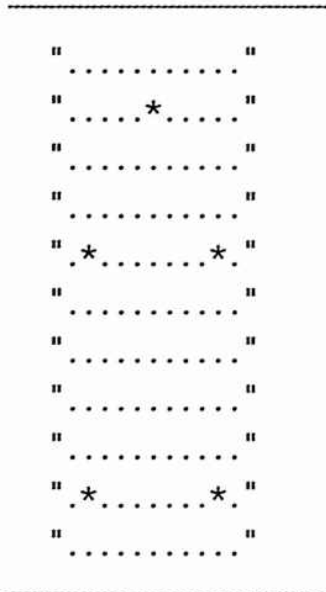


이때 모든 교점의 좌표는 $(4, 1)$, $(4, -4)$, $(-4, -4)$, $(-4, 1)$, $(0, 4)$, $(1.5, 1.0)$, $(2.1, -0.19)$, $(0, -1.5)$, $(-2.1, -0.19)$, $(-1.5, 1.0)$ 입니다. 이 중 정수로만 표현되는 좌표는 $(4, 1)$, $(4, -4)$, $(-4, -4)$, $(-4, 1)$, $(0, 4)$ 입니다.

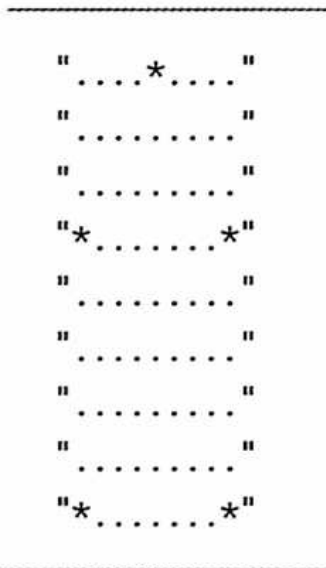
만약 정수로 표현되는 교점에 별을 그리면 다음과 같습니다.



위의 그림을 문자열로 나타낼 때, 별이 그려진 부분은 $*$, 빈 공간(격자선이 교차하는 지점)은 $.$ 으로 표현하면 다음과 같습니다.



이때 격자판은 무한히 넓으니 모든 별을 포함하는 최소한의 크기만 나타내면 됩니다.
따라서 정답은



입니다.

직선 A, B, C에 대한 정보가 담긴 배열 line이 매개변수로 주어집니다 이때 모든 별을 포함하는 최소 사각형을 return하도록 solution 함수를 완성해주세요.

제한사항

- line의 세로(행) 길이는 2 이상 1,000 이하인 자연수입니다
 - line의 가로(열) 길이는 3입니다
 - line의 각 원소는 [A, B, C] 형태입니다.
 - A, B, C는 -100,000 이상 100,000 이하인 정수입니다
 - 무수히 많은 교점이 생기는 직선 쌍은 주어지지 않습니다.
 - A = 0이면서 B = 0인 경우는 주어지지 않습니다.
- 정답은 1,000*1,000 크기 이내에서 표현됩니다.
- 별이 한 개 이상 그려지는 입력만 주어집니다.

입출력 예

line	result
[[2, -1, 4], [-2, -1, 4], [0, -1, 1], [5, -8, -12], [5, 8, 12]]	["....*....", ".....", ".....", "*.....*", ".....", ".....", ".....", ".....", "*.....*"]
[[0, 1, -1], [1, 0, -1], [1, 0, 1]]	["*. *"]
[[1, -1, 0], [2, -1, 0]]	["*"]
[[1, -1, 0], [2, -1, 0], [4, -1, 0]]	["*"]

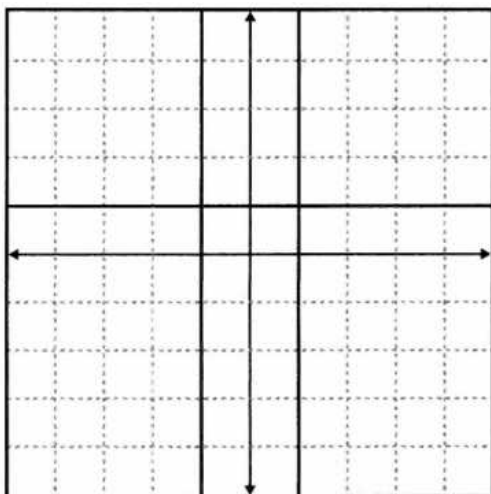
입출력 예 설명

입출력 예 #1

문제 예와 같습니다

입출력 예 #2

직선 $y = 1$, $x = 1$, $x = -1$ 은 다음과 같습니다.



$(-1, 1)$, $(1, 1)$ 에서 교점이 발생합니다.

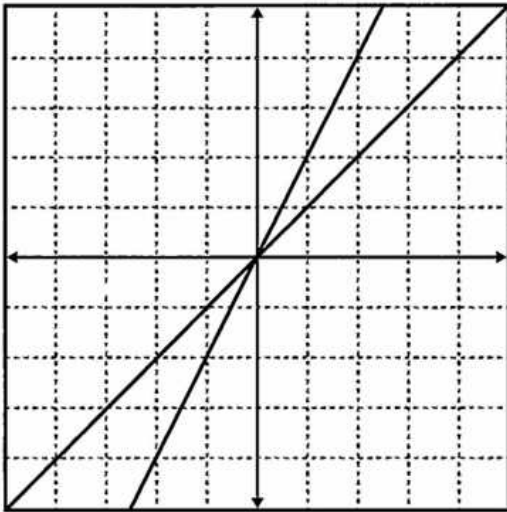
따라서 정답은

["*. *"]

입니다

입출력 예 #3

직선 $y = x$, $y = 2x$ 는 다음과 같습니다.



(0, 0)에서 교점이 발생합니다.

따라서 정답은

"★"

입니다.

입출력 예 #4

직선 $y = x$, $y = 2x$, $y = 4x$ 는 다음과 같습니다.

(0, 0)에서 교점이 발생합니다

따라서 정답은

"★"

입니다.

참고사항

$$Ax + By + E = 0$$

$$ex + Dy + F = 0$$

두 직선의 교점이 유일하게 존재할 경우, 그 교점은 다음과 같습니다.

$$x = \frac{BF - ED}{AD - BC} \quad y = \frac{EC - AF}{AD - BC}$$

또, $AD - BC = 0$ 인 경우 두 직선은 평행 또는 일치합니다.

2. 행렬 테두리 회전하기 - Level 2

URL <https://programmers.co.kr/learn/courses/30/lessons/77485>

rows x columns 크기인 행렬이 있습니다. 행렬에는 1부터 rows x columns까지의 숫자가 한 줄씩 순서대로 적혀있습니다. 이 행렬에서 직사각형 모양의 범위를 여러 번 선택해, 테두리 부분에 있는 숫자들을 시계방향으로 회전시키려 합니다. 각 회전은 (x1, y1, x2, y2)인 정수 4개로 표현하며, 그 의미는 다음과 같습니다.

x1 행 y1 열부터 x2 행 y2 열까지의 영역에 해당하는 직사각형에서 테두리에 있는 숫자들을 한 칸씩 시계방향으로 회전합니다.

다음은 6 x 6 크기 행렬의 예시입니다.

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

이 행렬에 (2, 2, 5, 4) 회전을 적용하면, 아래 그림과 같이 2행 2열부터 5행 4열까지 영역의 테두리가 시계방향으로 회전합니다. 이때, 중앙의 15와 21이 있는 영역은 회전하지 않는 것을 주의하세요.

1	2	3	4	5	6
7	14	8	9	11	12
13	20	15	10	17	18
19	26	21	16	23	24
25	27	28	22	29	30
31	32	33	34	35	36

행렬의 세로 길이(행 개수) rows, 가로 길이(열 개수) columns, 그리고 회전들의 목록 queries가 주어질 때, 각 회전들을 배열에 적용한 뒤, 그 회전에 의해 위치가 바뀐 숫자들 중 가장 작은 숫자들을 순서대로 배열에 담아 return 하도록 solution 함수를 완성해주세요.

제한사항

- rows는 2 이상 100 이하인 자연수입니다.
- columns는 2 이상 100 이하인 자연수입니다.
- 처음에 행렬에는 가로 방향으로 숫자가 1부터 하나씩 증가하면서 적혀있습니다.
 - 즉, 아무 회전도 하지 않았을 때, i 행 j 열에 있는 숫자는 $(i-1) \times \text{columns} + j$ 입니다.
- queries의 행의 개수(회전의 개수)는 1 이상 10,000 이하입니다.
- queries의 각 행은 4개의 정수 [x1, y1, x2, y2]입니다.
 - x1 행 y1 열부터 x2 행 y2 열까지 영역의 테두리를 시계방향으로 회전한다는 뜻입니다.
 - $1 \leq x1 < x2 \leq \text{rows}$, $1 \leq y1 < y2 \leq \text{columns}$ 입니다.
 - 모든 회전은 순서대로 이루어집니다.
 - 예를 들어, 두 번째 회전에 대한 답은 첫 번째 회전을 실행한 다음, 그 상태에서 두 번째 회전을 실행했을 때 이동한 숫자 중 최솟값을 구하면 됩니다.

입출력 예

rows	columns	queries	result
6	6	[[2, 2, 5, 4],[3, 3, 6, 6],[5, 1, 6, 3]]	[8, 10, 25]
3	3	[[1, 1, 2, 2],[1, 2, 2, 3],[2, 1, 3, 2],[2, 2, 3, 3]]	[1, 1, 5, 3]
100	97	[[1, 1, 100, 97]]	[1]

입출력 예 설명

입출력 예 #1

- 회전을 수행하는 과정을 그림으로 표현하면 다음과 같습니다.

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

1	2	3	4	5	6
7	14	8	9	11	12
13	20	15	10	17	18
19	26	21	16	23	24
25	27	28	22	29	30
31	32	33	34	35	36

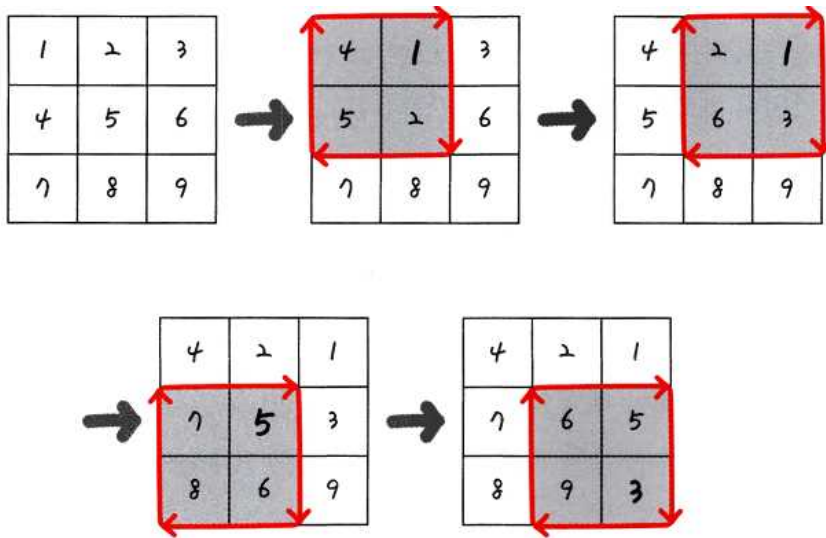
1	2	3	4	5	6
7	14	8	9	11	12
13	20	21	15	10	17
19	26	28	16	23	18
25	27	33	22	29	24
31	32	34	35	36	30

1	2	3	4	5	6
7	14	8	9	11	12
13	20	21	15	10	17
19	26	28	16	23	18
25	27	33	22	29	24
31	32	34	35	36	30

1	2	3	4	5	6
7	14	8	9	11	12
13	20	21	15	10	17
19	26	28	16	23	18
25	27	33	22	29	24
31	25	27	22	29	24
32	34	33	35	36	30

입출력 예# 2

- 회전을 수행하는 과정을 그림으로 표현하면 다음과 같습니다.



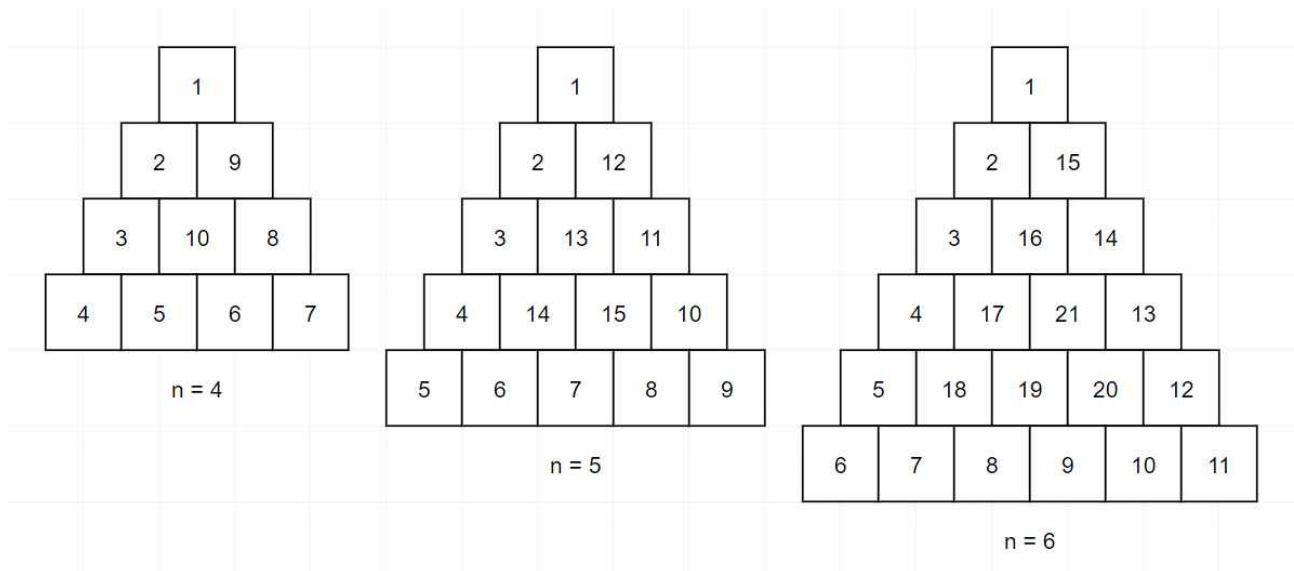
입출력 예 #3

- 이 예에서는 행렬의 테두리에 위치한 모든 칸들이 움직입니다. 따라서 행렬의 테두리에 있는 수 중 가장 작은 숫자인 1이 바로 답이 됩니다.

3. 삼각 달팽이 - Level 2

URL <https://programmers.co.kr/learn/courses/30/lessons/68645>

정수 n 이 매개변수로 주어집니다. 다음 그림과 같이 밑변의 길이와 높이가 n 인 삼각형에서 맨 위 꼭짓점부터 반시계 방향으로 달팽이 채우기를 진행한 후, 첫 행부터 마지막 행까지 모두 순서대로 합친 새로운 배열을 return 하도록 solution 함수를 완성해주세요.



제한사항

n 은 1 이상 1,000 이하입니다.

입출력 예

n	result
4	[1,2,9,3,10,8,4,5,6,7]
5	[1,2,12,3,13,11,4,14,15,10,5,6,7,8,9]
6	[1,2,15,3,16,14,4,17,21,13,5,18,19,20,12,6,7,8,9,10,11]

입출력 예 설명

입출력 예 #1

문제 예시와 같습니다.

입출력 예 #2

문제 예시와 같습니다.

입출력 예 #3

문제 예시와 같습니다.

4. 거리두기 확인하기 - Level 2

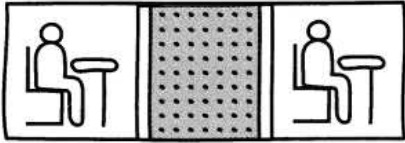

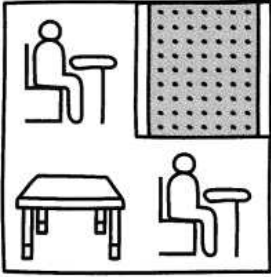


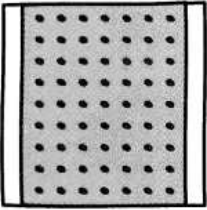
URL <https://programmers.co.kr/learn/courses/30/lessons/81302>

개발자를 희망하는 조르디가 카카오에 면접을 보러 왔습니다.

코로나 바이러스 감염 예방을 위해 응시자들은 거리를 뒤편 대기해야 하는데 개발 직군 면접인 만큼 아래와 같은 규칙으로 대기실에 거리를 두고 앉도록 안내하고 있습니다.

1. 대기실은 5개이며, 각 대기실은 5x5 크기입니다.
2. 거리두 기를 위하여 응시자들 끼리는 맨해튼 거리가 2 이하로 앉지 말아주세요.
3. 단 응시자가 앉아있는 자리 사이가 파티션으로 막혀 있을 경우에는 허용합니다.

예를 들어,

 <p>위 그림처럼 자리 사이에 파티션이 존재한다면 맨해튼 거리가 2여도 거리두기를 지킨 것입니다.</p>	 <p>위 그림처럼 파티션을 사이에 두고 앉은 경우도 거리두기를 지킨 것입니다.</p>	 <p>위 그림처럼 자리 사이가 맨해튼 거리 2이고 사이에 빈 테이블이 있는 경우는 거리두기를 지키지 않은 것입니다.</p>
 <p>응시자가 앉아있는 자리(P)를 의미합니다.</p>	 <p>빈 테이블(O)을 의미합니다.</p>	 <p>파티션(X)을 의미합니다.</p>

5개의 대기실을 본 조르디는 각 대기실에서 응시자들이 거리두기를 잘 기키고 있는지 알고 싶어했습니다. 자리에 앉아있는 응시자들의 정보와 대기실 구조를 대기실별로 담은 2차원 문자열 배열 places가 매개변수로 주어집니다. 각 대기실별로 거리두기를 지키고 있으면 1을, 한 명이라도 지키지 않고 있으면 0을 배열에 담아 return 하도록 solution 함수를 완성해 주세요

제한사항

- places의 행 길이(대기실 개수) = 5
 - places의 각 행은 하나의 대기실 구조를 나타냅니다.
- places의 열 길이(대기실 세로 길이) = 5

- places의 원소는 P,O,X로 이루어진 문자열입니다.
 - places 원소의 길이(대기실 가로 길이) = 5
 - P는 응시자가 앉아있는 자리를 의미합니다.
 - O는 빈 테이블을 의미합니다.
 - X는 파티션을 의미합니다.
- 입력으로 주어지는 5개 대기실의 크기는 모두 5x5 입니다.
- return 값 형식
 - 1차원 정수 배열에 5개의 원소를 담아서 return 합니다.
 - places에 담겨 있는 5개 대기실의 순서대로, 거리두기 준수 여부를 차례대로 배열에 담습니다.
 - 각 대기실 별로 모든 응시자가 거리두기를 지키고 있으면 1을, 한 명이라도 지키지 않고 있으면 0을 담습니다

입출력 예

places	result
<pre>[["P000P", "0XX0X", "0PXPX", "00X0X", "P0XXP"], ["P00PX", "0XPXP", "PXXX0", "0XXX0", "000PP"], ["PX0PX", "0X0XP", "0XP0X", "0XX0P", "PXP0X"], ["000XX", "X000X", "000XX", "0X00X", "00000"], ["PXPXP", "XPXPX", "PXPXP", "PXPXP"]]</pre>	[1, 0, 1, 1, 1]

입출력 예 설명

입출력 예 #1

첫 번째 대기실

No.	0	1	2	3	4
0	P	O	O	O	P
1	O	X	X	O	X
2	O	P	X	P	X
3	O	O	X	O	X
4	P	O	X	X	P

- 모든 응시자가 거리두기를 지키고 있습니다.

두 번째 대기실

No.	0	1	2	3	4
0	P	O	O	P	X
1	O	X	P	X	P
2	P	X	X	X	O
3	O	X	X	X	O
4	O	O	O	P	P

- (0, 0) 자리의 응시자와 (2, 0) 자리의 응시자가 거리두기를 지키고 있지 않습니다.
- (1, 2) 자리의 응시자와 (0, 3) 자리의 응시자가 거리두기를 지키고 있지 않습니다.
- (4, 3) 자리의 응시자와 (4, 4) 자리의 응시자가 거리두기를 지키고 있지 않습니다.

세 번째 대기실

No.	0	1	2	3	4
0	P	X	O	P	X
1	O	X	O	X	P
2	O	X	P	O	X
3	O	X	X	O	P
4	P	X	P	O	X

- 모든 응시자가 거리두기를 지키고 있습니다.

네 번째 대기실

No.	0	1	2	3	4
0	O	O	O	X	X
1	X	O	O	O	X
2	O	O	O	X	X
3	O	X	O	O	X
4	O	O	O	O	O

- 대기실에 응시자가 없으므로 거리두기를 지키고 있습니다.

다섯 번째 대기실

No.	0	1	2	3	4
0	P	X	P	X	P
1	X	P	X	P	X
2	P	X	P	X	P
3	X	P	X	P	X
4	P	X	P	X	P

- 모든 응시자가 거리두기를 지키고 있습니다.

두 번째 대기실을 제외한 모든 대기실에서 거리두기가 지켜지고 있으므로, 배열 [1, 0, 1, 1, 1]을 return 합니다.

제한시간 안내

정확성 테스트 : 10초

공지 - 2022년 4월 25일 테스트케이스가 추가되었습니다.

두 테이블 T1, T2가 행렬 (r1, c1), (r2, c2)에 각각 위치하고 있다면, T1, T2 사이의 맨해튼 거리는 $|r1 - r2| + |c1 - c2|$ 입니다.

5. 행렬의 곱셈 - Level 2

URL <https://programmers.co.kr/learn/courses/30/lessons/12949>

2차원 행렬 arr1과 arr2를 입력받아, arr1에 arr2를 곱한 결과를 반환하는 함수, solution을 완성해주세요.

제한 조건

- 행렬 arr1, arr2의 행과 열의 길이는 2 이상 100 이하입니다.
- 행렬 arr1, arr2의 원소는 -10 이상 20 이하인 자연수입니다.
- 곱할 수 있는 배열만 주어집니다.

입출력 예

arr1	arr2	return
[[1, 4], [3, 2], [4, 1]]	[[3, 3], [3, 3]]	[[15, 15], [15, 15], [15, 15]]
[[2, 3, 2], [4, 2, 4], [3, 1, 4]]	[[5, 4, 3], [2, 4, 1], [3, 1, 1]]	[[22, 22, 11], [36, 28, 18], [29, 20, 14]]

6. 시저 암호 - Level 1

URL <https://programmers.co.kr/learn/courses/30/lessons/12926>

어떤 문장의 각 알파벳을 일정한 거리만큼 밀어서 다른 알파벳으로 바꾸는 암호화 방식을 시저 암호라고 합니다. 예를 들어 "AB"는 1만큼 밀면 "BC"가 되고, 3만큼 밀면 "DE"가 됩니다. "z"는 1만큼 밀면 "a"가 됩니다. 문자열 s와 거리 n을 입력받아 s를 n만큼 밀 암호문을 만드는 함수, solution을 완성해 보세요.

제한 조건

- 공백은 아무리 밀어도 공백입니다.
- s는 알파벳 소문자, 대문자, 공백으로만 이루어져 있습니다.
- s의 길이는 8000 이하입니다.
- n은 1 이상, 25 이하인 자연수입니다.

입출력 예

s	n	result
"AB"	1	"BC"
"z"	1	"a"
"a B z"	4	"e F d"

7. 이상한 문자 만들기 - Level 1

URL <https://programmers.co.kr/learn/courses/30/lessons/12930>

문자열 `s`는 한 개 이상의 단어로 구성되어 있습니다. 각 단어는 하나 이상의 공백문자로 구분되어 있습니다. 각 단어의 짝수번째 알파벳은 대문자로, 홀수번째 알파벳은 소문자로 바꾼 문자열을 리턴하는 함수, `solution`을 완성하세요.

제한 사항

문자열 전체의 짝/홀수 인덱스가 아니라, 단어(공백을 기준)별로 짝/홀수 인덱스를 판단해야 합니다.
첫 번째 글자는 0번째 인덱스로 보아 짝수번째 알파벳으로 처리해야 합니다.

입출력 예

s	Return
"try hello world"	"TrY HeLlO WoRlD"

입출력 예 설명

"try hello world"는 세 단어 "try", "hello", "world"로 구성되어 있습니다. 각 단어의 짝수번째 문자를 대문자로, 홀수번째 문자를 소문자로 바꾸면 "TrY", "HeLlO", "WoRlD"입니다. 따라서 "TrY HeLlO WoRlD"를 리턴합니다.

8. 튜플-Level2

URL <https://programmers.co.kr/learn/courses/30/lessons/64065>

셀수있는 수량의 순서있는 열거 또는 어떤 순서를 따르는 요소들의 모음을 튜플(tuple)이라고 합니다. n개의 요소를 가진 튜플을 n-튜플(n-tuple)이라고 하며, 다음과 같이 표현할 수 있습니다.

(a1, a2, a3, ..., an)

튜플은 다음과 같은 성질을 가지고 있습니다.

- 중복된 원소가 있을 수 있습니다. ex : (2, 3, 1, 2)
- 원소에 정해진 순서가 있으며, 원소의 순서가 다르면 서로 다른 튜플입니다. ex : (1, 2, 3) \neq (1, 3, 2)
- 튜플의 원소 개수는 유한합니다.
-

원소의 개수가 n개이고, 중복되는 원소가 없는 튜플 (a1, a2, a3, ..., an)이 주어질 때(단, a1, a2, ..., an은 자연수), 이는 다음과 같이 집합 기호 '{, }'를 이용해 표현할 수 있습니다.

- {{a1}, {a1, a2}, {a1, a2, a3}, {a1, a2, a3, a4}, ... {a1, a2, a3, a4, ..., an}}

예를 들어 튜플이 (2, 1, 3, 4)인 경우 이는

- {{2}, {2, 1}, {2, 1, 3}, {2, 1, 3, 4}}

와 같이 표현할 수 있습니다. 이때, 집합은 원소의 순서가 바뀌어도 상관없으므로

- {{2}, {2, 1}, {2, 1, 3}, {2, 1, 3, 4}}
- {{2, 1, 3, 4}, {2}, {2, 1, 3}, {2, 1}}
- {{1, 2, 3}, {2, 1}, {1, 2, 4, 3}, {2}}
-

는 모두 같은 튜플 (2, 1, 3, 4)를 나타냅니다.

특정 튜플을 표현하는 집합이 담긴 문자열 s가 매개변수로 주어질 때, s가 표현하는 튜플을 배열에 담아 return 하도록 solution 함수를 완성해주세요.

[제한사항]

- s의 길이는 5 이상 1,000,000 이하입니다.
- s는 숫자와 '{', '}', ',' 로만 이루어져 있습니다.
- 숫자가 0으로 시작하는 경우는 없습니다.
- s는 항상 중복되는 원소가 없는 튜플을 올바르게 표현하고 있습니다.
- s가 표현하는 튜플의 원소는 1 이상 100,000 이하인 자연수입니다.
- return 하는 배열의 길이가 1 이상 500 이하인 경우만 입력으로 주어집니다.

[입출력 예]

s	result
"{{2},{2, 1},{2, 1, 3},{2, 1, 3, 4}}"	[2, 1, 3, 4]
"{{1, 2, 3},{2, 1},{1, 2, 4, 3},{2}}"	[2, 1, 3, 4]
"{{20, 111},{111}}"	[111, 20]
"{{123}}"	[123]
"{{4, 2, 3},{3},{2, 3, 4, 1},{2, 3}}"	[3, 2, 4, 1]

입출력 예에 대한 설명

입출력 예 #1

문제 예시와 같습니다.

입출력 예 #2

문제 예시와 같습니다.

입출력 예 #3

(111, 20)을 집합 기호를 이용해 표현하면 {{111}, {111,20}}이 되며, 이는 {{20,111},{111}}과 같습니다.

입출력 예 #4

(123)을 집합 기호를 이용해 표현하면 {{123}} 입니다.

입출력 예 #5

(3, 2, 4, 1)을 집합 기호를 이용해 표현하면 {{3},{3,2},{3,2,4},{3,2,4,1}}이 되며, 이는 {{4,2,3}, {3}, {2,3,4,1}, {2,3}}과 같습니다.

9. 짝지어 제거하기 - Level 2

URL <https://programmers.co.kr/learn/courses/30/lessons/12973>

짝지어 제거하기는, 알파벳 소문자로 이루어진 문자열을 가지고 시작합니다. 먼저 문자열에서 같은 알파벳이 2개 붙어 있는 짝을 찾습니다. 그다음, 그 둘을 제거한 뒤, 앞뒤로 문자열을 이어 붙입니다. 이 과정을 반복해서 문자열을 모두 제거한다면 짝지어 제거하기가 종료됩니다. 문자열 S가 주어졌을 때, 짝지어 제거하기를 성공적으로 수행할 수 있는지 반환하는 함수를 완성해 주세요. 성공적으로 수행할 수 있으면 1을, 아닐 경우 0을 리턴해주면 됩니다.

예를 들어, 문자열 S = baabaa 라면

b aa baa → bb aa → aa →

의 순서로 문자열을 모두 제거할 수 있으므로 1을 반환합니다.

제한사항

문자열의 길이 : 1,000,000이하의 자연수

문자열은 모두 소문자로 이루어져 있습니다.

입출력 예

s	result
baabaa	1
cdcd	0

입출력 예 설명

입출력 예 #1

위의 예시와 같습니다.

입출력 예 #2

문자열이 남아있지만 짝지어 제거할 수 있는 문자열이 더 이상 존재하지 않기 때문에 0을 반환합니다.

10. 문자열 압축 - Level 2

URL <https://programmers.co.kr/learn/courses/30/lessons/60057>

데이터 처리 전문가가 되고 싶은 "어피치"는 문자열을 압축하는 방법에 대해 공부를 하고 있습니다. 최근에 대량의 데이터 처리를 위한 간단한 비손실 압축 방법에 대해 공부를 하고 있는데, 문자열에서 같은 값이 연속해서 나타나는 것을 그 문자의 개수와 반복되는 값으로 표현하여 더 짧은 문자열로 줄여서 표현하는 알고리즘을 공부하고 있습니다.

간단한 예로 "aabbaccc"의 경우 "2a2ba3c"(문자가 반복되지 않아 한번만 나타난 경우 1은 생략함)와 같이 표현할 수 있는데, 이러한 방식은 반복되는 문자가 적은 경우 압축률이 낮다는 단점이 있습니다. 예를 들면, "abcabcdede"와 같은 문자열은 전혀 압축되지 않습니다. "어피치"는 이러한 단점을 해결하기 위해 문자열을 1개 이상의 단위로 잘라서 압축하여 더 짧은 문자열로 표현할 수 있는지 방법을 찾아보려고 합니다.

예를 들어, "abababcdcdabababcdcd"의 경우 문자를 1개 단위로 자르면 전혀 압축되지 않지만, 2개 단위로 잘라서 압축한다면 "2ab2cd2ab2cd"로 표현할 수 있습니다. 다른 방법으로 8개 단위로 잘라서 압축한다면 "2abababcdcd"로 표현할 수 있으며, 이때가 가장 짧게 압축하여 표현할 수 있는 방법입니다.

다른 예로, "abcabcdede"와 같은 경우, 문자를 2개 단위로 잘라서 압축하면 "abcabc2de"가 되지만, 3개 단위로 자른다면 "2abcdede"가 되어 3개 단위가 가장 짧은 압축 방법이 됩니다. 이때 3개 단위로 자르고 마지막에 남는 문자열은 그대로 붙여주면 됩니다.

압축할 문자열 `s`가 매개변수로 주어질 때, 위에 설명한 방법으로 1개 이상 단위로 문자열을 잘라 압축하여 표현한 문자열 중 가장 짧은 것의 길이를 return 하도록 solution 함수를 완성해주세요.

제한사항

`s`의 길이는 1 이상 1,000 이하입니다.

`s`는 알파벳 소문자로만 이루어져 있습니다.

입출력 예

s	result
"aabbaccc"	7
"abababcdcdabababcdcd"	9
"abcabcdede"	8
"abcabcabcababcdededededede"	14
"xabababcdcdabababcdcd"	17

입출력 예에 대한 설명

입출력 예 #1

문자열을 1개 단위로 잘라 압축했을 때 가장 짧습니다.

입출력 예 #2

문자열을 8개 단위로 잘라 압축했을 때 가장 짧습니다.

입출력 예 #3

문자열을 3개 단위로 잘라 압축했을 때 가장 짧습니다.

입출력 예 #4

문자열을 2개 단위로 자르면 "abcabcabcabc6de" 가 됩니다.

문자열을 3개 단위로 자르면 "4abcdededededede" 가 됩니다.

문자열을 4개 단위로 자르면 "abcabcabcabc3dede" 가 됩니다.

문자열을 6개 단위로 자를 경우 "2abcabc2dedede"가 되며, 이때의 길이가 14로 가장 짧습니다.

입출력 예 #5

문자열은 제일 앞부터 정해진 길이만큼 잘라야 합니다.

따라서 주어진 문자열을 x / ababcdcd / ababcdcd 로 자르는 것은 불가능 합니다.

이 경우 어떻게 문자열을 잘라도 압축되지 않으므로 가장 짧은 길이는 17이 됩니다.

11. 3진법 뒤집기 - Level 1

URL <https://programmers.co.kr/learn/courses/30/lessons/68935>

자연수 n 이 매개변수로 주어집니다. n 을 3진법 상에서 앞뒤로 뒤집은 후, 이를 다시 10진법으로 표현한 수를 return 하도록 solution 함수를 완성해주세요.

제한사항

n 은 1 이상 100,000,000 이하인 자연수입니다.

입출력 예

n	result
45	7
125	229

입출력 예 설명

입출력 예 #1

답을 도출하는 과정은 다음과 같습니다.

n (10진법)	n (3진법)	앞뒤 반전(3진법)	10진법으로 표현
45	1200	0021	7

따라서 7을 return 해야 합니다.

입출력 예 #2

답을 도출하는 과정은 다음과 같습니다.

n (10진법)	n (3진법)	앞뒤 반전(3진법)	10진법으로 표현
125	11122	22111	229

따라서 229를 return 해야 합니다.

12. 이진 변환 반복하기 -Level 2

URL <https://programmers.co.kr/learn/courses/30/lessons/70129>

0과 1로 이루어진 어떤 문자열 x 에 대한 이진 변환을 다음과 같이 정의합니다.

1. x 의 모든 0을 제거합니다.
2. x 의 길이를 c 라고 하면, x 를 " c 를 2진법으로 표현한 문자열"로 바꿉니다.

예를 들어, $x = "0111010"$ 이라면, x 에 이진 변환을 가하면 $x = "0111010" \rightarrow "1111" \rightarrow "100"$ 이 됩니다.

0과 1로 이루어진 문자열 s 가 매개변수로 주어집니다. s 가 "1"이 될 때까지 계속해서 s 에 이진 변환을 가했을 때, 이진 변환의 횟수와 변환 과정에서 제거된 모든 0의 개수를 각각 배열에 담아 return 하도록 solution 함수를 완성해주세요.

제한사항

s 의 길이는 1 이상 150,000 이하입니다.

s 에는 '1'이 최소 하나 이상 포함되어 있습니다.

입출력 예

s	result
"110010101001"	[3, 8]
"01110"	[3, 3]
"1111111"	[4, 1]

입출력 예 설명

입출력 예 #1

"110010101001"이 "1"이 될 때까지 이진 변환을 가하는 과정은 다음과 같습니다.

회차	이진 변환 이전	제거할 0의 개수	0 제거 후 길이	이진 변환 결과
1	"110010101001"	6	6	"110"
2	"110"	1	2	"10"
3	"10"	1	1	"1"

3번의 이진 변환을 하는 동안 8개의 0을 제거했으므로, [3,8]을 return 해야 합니다.

입출력 예 #2

"01110"이 "1"이 될 때까지 이진 변환을 가하는 과정은 다음과 같습니다.

회차	이진 변환 이전	제거할 0의 개수	0 제거 후 길이	이진 변환 결과
1	"01110"	2	3	"11"
2	"11"	0	2	"10"
3	"10"	1	1	"1"

3번의 이진 변환을 하는 동안 3개의 0을 제거했으므로, [3,3]을 return 해야 합니다.

입출력 예 #3

"1111111"이 "1"이 될 때까지 이진 변환을 가하는 과정은 다음과 같습니다.

회차	이진 변환 이전	제거할 0의 개수	0 제거 후 길이	이진 변환 결과
1	"1111111"	0	7	"111"
2	"111"	0	3	"11"
3	"11"	0	2	"10"
4	"10"	1	1	"1"

4번의 이진 변환을 하는 동안 1개의 0을 제거했으므로, [4,1]을 return 해야 합니다.

13. 신규 아이디 추천 -Level 1

URL <https://programmers.co.kr/learn/courses/30/lessons/72410>

카카오에 입사한 신입 개발자 네오는 "카카오계정개발팀"에 배치되어, 카카오 서비스에 가입하는 유저들의 아이디를 생성하는 업무를 담당하게 되었습니다. "네오"에게 주어진 첫 업무는 새로 가입하는 유저들이 카카오 아이디 규칙에 맞지 않는 아이디를 입력했을 때, 입력된 아이디와 유사하면서 규칙에 맞는 아이디를 추천해주는 프로그램을 개발하는 것입니다.

다음은 카카오 아이디의 규칙입니다.

아이디의 길이는 3자 이상 15자 이하여야 합니다.

아이디는 알파벳 소문자, 숫자, 빼기(-), 밑줄(_), 마침표(.) 문자만 사용할 수 있습니다.

단, 마침표(.)는 처음과 끝에 사용할 수 없으며 또한 연속으로 사용할 수 없습니다.

"네오"는 다음과 같이 7단계의 순차적인 처리 과정을 통해 신규 유저가 입력한 아이디가 카카오 아이디 규칙에 맞는 지 검사하고 규칙에 맞지 않은 경우 규칙에 맞는 새로운 아이디를 추천해 주려고 합니다.

신규 유저가 입력한 아이디가 `new_id` 라고 한다면,

1단계 `new_id`의 모든 대문자를 대응되는 소문자로 치환합니다.

2단계 `new_id`에서 알파벳 소문자, 숫자, 빼기(-), 밑줄(_), 마침표(.)를 제외한 모든 문자를 제거합니다.

3단계 `new_id`에서 마침표(.)가 2번 이상 연속된 부분을 하나의 마침표(.)로 치환합니다.

4단계 `new_id`에서 마침표(.)가 처음이나 끝에 위치한다면 제거합니다.

5단계 `new_id`가 빈 문자열이라면, `new_id`에 "a"를 대입합니다.

6단계 `new_id`의 길이가 16자 이상이면, `new_id`의 첫 15개의 문자를 제외한 나머지 문자들을 모두 제거합니다.

만약 제거 후 마침표(.)가 `new_id`의 끝에 위치한다면 끝에 위치한 마침표(.) 문자를 제거합니다.

7단계 `new_id`의 길이가 2자 이하라면, `new_id`의 마지막 문자를 `new_id`의 길이가 3이 될 때까지 반복해서 끝에 붙입니다.

예를 들어, `new_id` 값이 "...!@BaT#*..y.abcdefghijklm" 라면, 위 7단계를 거치고 나면 `new_id`는 아래와 같이 변경됩니다.

1단계 대문자 'B'와 'T'가 소문자 'b'와 't'로 바뀌었습니다.

"...!@BaT#*..y.abcdefghijklm" → "...!@bat#*..y.abcdefghijklm"

2단계 '!', '@', '#', '*' 문자가 제거되었습니다.

"...!@bat#*..y.abcdefghijklm" → "...bat..y.abcdefghijklm"

3단계 '...'와 '..' 가 '.'로 바뀌었습니다.

"...bat..y.abcdefghijklm" → ".bat.y.abcdefghijklm"

4단계 아이디의 처음에 위치한 '.'가 제거되었습니다.

".bat.y.abcdefghijklm" → "bat.y.abcdefghijklm"

5단계 아이디가 빈 문자열이 아니므로 변화가 없습니다.

"bat.y.abcdefghijklm" → "bat.y.abcdefghijklm"

6단계 아이디의 길이가 16자 이상이므로, 처음 15자를 제외한 나머지 문자들이 제거되었습니다.

"bat.y.abcdefghijklm" → "bat.y.abcdefghi"

7단계 아이디의 길이가 2자 이하가 아니므로 변화가 없습니다.

"bat.y.abcdefghi" → "bat.y.abcdefghi"

따라서 신규 유저가 입력한 new_id가 "...!@BaT#*..y.abcdefghijklm"일 때, 네오의 프로그램이 추천하는 새로운 아이디는 "bat.y.abcdefghi" 입니다.

[문제]

신규 유저가 입력한 아이디를 나타내는 new_id가 매개변수로 주어질 때, "네오"가 설계한 7단계의 처리 과정을 거친 후의 추천 아이디를 return 하도록 solution 함수를 완성해 주세요.

[제한사항]

new_id는 길이 1 이상 1,000 이하인 문자열입니다.

new_id는 알파벳 대문자, 알파벳 소문자, 숫자, 특수문자로 구성되어 있습니다.

new_id에 나타날 수 있는 특수문자는 -_.~!@#\$%^&*()=+[{ }:?,<>/ 로 한정됩니다.

[입출력 예]

no	new_id	result
예1	"...!@BaT#*..y.abcdefghijklm"	"bat.y.abcdefghi"
예2	"z-+.^.."	"z--"
예3	"=,=."	"aaa"
예4	"123_.def"	"123_.def"
예5	"abcdefghijklmn.p"	"abcdefghijklmn"

입출력 예에 대한 설명

입출력 예 #1

문제의 예시와 같습니다.

입출력 예 #2

7단계를 거치는 동안 new_id가 변화하는 과정은 아래와 같습니다.

1단계 변화 없습니다.

2단계 "z-+.^.." → "z-.."

3단계 "z-.." → "z-."

4단계 "z-." → "z-"

5단계 변화 없습니다.

6단계 변화 없습니다.

7단계 "z-" → "z--"

입출력 예 #3

7단계를 거치는 동안 new_id가 변화하는 과정은 아래와 같습니다.

1단계 변화 없습니다.

2단계 "=". → "."

3단계 변화 없습니다.

4단계 "." → "" (new_id가 빈 문자열이 되었습니다.)

5단계 "" → "a"

6단계 변화 없습니다.

7단계 "a" → "aaa"

입출력 예 #4

1단계에서 7단계까지 거치는 동안 new_id("123_.def")는 변하지 않습니다. 즉, new_id가 처음부터 카카오의 아이디 규칙에 맞습니다.

입출력 예 #5

1단계 변화 없습니다.

2단계 변화 없습니다.

3단계 변화 없습니다.

4단계 변화 없습니다.

5단계 변화 없습니다.

6단계 "abcdefghijklmn.p" → "abcdefghijklmn." → "abcdefghijklmn"

7단계 변화 없습니다.

14. 문자열 다루기 기본 - Level 1

URL <https://programmers.co.kr/learn/courses/30/lessons/12918>

문자열 `s`의 길이가 4 혹은 6이고, 숫자로만 구성돼있는지 확인해주는 함수, `solution`을 완성하세요. 예를 들어 `s`가 "a234"이면 `False`를 리턴하고 "1234"라면 `True`를 리턴하면 됩니다.

제한 사항

`s`는 길이 1 이상, 길이 8 이하인 문자열입니다.

`s`는 영문 알파벳 대소문자 또는 0부터 9까지 숫자로 이루어져 있습니다.

입출력 예

s	return
"a234"	false
"1234"	true

15. 핸드폰 번호 가리기 - Level 1

URL <https://school.programmers.co.kr/learn/courses/30/lessons/12948>

프로그래머스 모바일은 개인정보 보호를 위해 고지서를 보낼 때 고객들의 전화번호의 일부를 가립니다. 전화번호가 문자열 `phone_number`로 주어졌을 때, 전화번호의 뒷 4자리를 제외한 나머지 숫자를 전부 *으로 가린 문자열을 리턴하는 함수, `solution`을 완성해주세요.

제한 조건

`phone_number`는 길이 4 이상, 20이하인 문자열입니다.

입출력 예

phone_number	return
"01033334444"	"*****4444"
"027778888"	"*****8888"