

문제 소개 및 해설

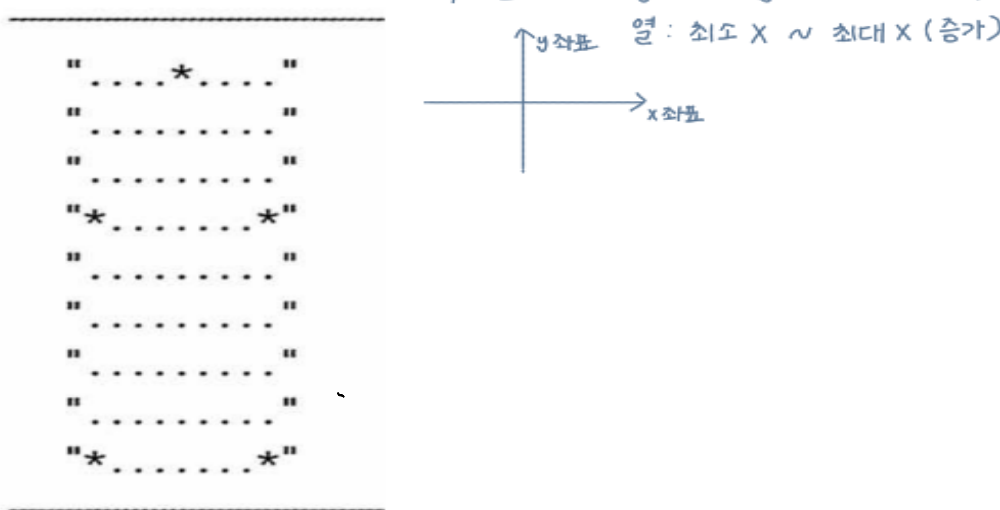
$Ax + By + C = 0$ 으로 표현할 수 있는 n 개의 직선이 주어질 때, 이 직선의 교점 중 정수 좌표에 별을 그리려 합니다

예를 들어 다음과 같은 직선 5개를

- $2x - y + 4 = 0$
- $-2x - y + 4 = 0$
- $-y + 1 = 0$

n 개 $[A, B, C]$ 2개의 직선 사이의 교점 찾기
↳ 2중 for문

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



참고사항 교점구하기

$$Ax + By + E = 0$$

$$Cx + Dy + F = 0$$

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.

↳ 무한한 교점

∴ $AD - BC = 0 \rightarrow$ 교점 X 직선

→ 제한사항: 무한한 교점이 생기는 쌍 주어지지 X

교점에 별 만들기

(요약)

1. 주어진 직선의 교점 중 정수 좌표 구하기
2. 구한 좌표에는 *, 아닌 부분은 .을 그려 출력
3. 격자판은 모든 별을 포함하는 최소한의 크기만 나타냄

입출력 예

line	result
$[[2, -1, 4], [-2, -1, 4], [0, -1, 1], [5, -8, -12], [5, 8, 12]]$	<pre>["....*....", ".....", ".....", "*.....*", ".....", ".....", ".....", ".....", "*.....*"]</pre>
$[[0, 1, -1], [1, 0, -1], [1, 0, 1]]$	<pre>["*.*."]</pre>
$[[1, -1, 0], [2, -1, 0]]$	<pre>["*"]</pre>
$[[1, -1, 0], [2, -1, 0], [4, -1, 0]]$	<pre>["*"]</pre>

<https://velog.io/@byhazy/Python-%ED%8A%9C%ED%94%8C-%EC%A7%91%ED%95%A>

<https://school.programmers.co.kr/learn/courses/30/lessons/87377?language=python3>

코드 해설

```
def solution(line):
    answer = []
    points = set() #교점(x,y)을 추가할 비어있는 집합 자료형

    # 교점 구하기
    for i in range(len(line)):
        for j in range(i + 1, len(line)):
            a, b, e = line[i]
            c, d, f = line[j]

            if (a * d) - (b * c) != 0: #평행하는 직선이 아니라면
                x = (b * f - e * d) / (a * d - b * c) # x 좌표 = BF - ED / AD - BC
                y = (e * c - a * f) / (a * d - b * c) # y 좌표 = EC - AF / AD - BC

            # 정수 확인
            if int(x) == x and int(y) == y:
                x = int(x)
                y = int(y)
                points.add((x, y)) #집합에 튜플(좌표) 추가

    # 그림 영역 구하기
    min_x = min(point[0] for point in points) # 집합에 추가된 튜플(좌표)에서 x좌표의 최소값
    max_x = max(point[0] for point in points)
    min_y = min(point[1] for point in points)
    max_y = max(point[1] for point in points)

    # 그림 그리기
    # x좌표는 왼쪽에서 오른쪽으로 증가, y좌표는 위에서 아래로 감소
    for y in range(max_y, min_y - 1, -1):
        row = ""
        for x in range(min_x, max_x + 1):
            if (x, y) in points:
                row += "*"
            else:
                row += "."
        answer.append(row)

    return answer
```

answer []
points ()

line[i] line[j]
① : ② ~ ⑦
② : ③ ~ ⑦
⋮

입력값 ①[A,B,C]
②
⋮
⑦

) n

points ((x₁,y₁), (x₂,y₂), ...)

↑ ↑
point

point[0] = x 좌표
point[1] = y 좌표

결과값 "....*...."

⋮

max-y
min-y

min-x max-x