

CODE FESTIVAL



문제 1

단체사진 찍기



가을을 맞아 카카오프렌즈는 단체로 소풍을 떠났다. 즐거운 시간을 보내고 마지막에 단체사진을 찍기 위해 카메라 앞에 일렬로 나란히 섰다. 그런데 각자가 원하는 배치가 모두 달라 어떤 순서로 설지 정하는데 시간이 오래 걸렸다. 네오는 프로도와 나란히 서기를 원했고, 튜브가 뽀은 불을 맞은 적이 있던 라이언은 튜브에게서 적어도 세 칸 이상 떨어져서 서기를 원했다. 사진을 찍고 나서 돌아오는 길에, 무지는 모두가 원하는 조건을 만족하면서도 다르게 서는 방법이 있지 않았을까 생각해보게 되었다. 각 프렌즈가 원하는 조건을 입력으로 받았을 때 모든 조건을 만족할 수 있도록 서는 경우의 수를 계산하는 프로그램을 작성해보자.

입력 형식

입력은 조건의 개수를 나타내는 정수 n 과 n 개의 원소로 구성된 문자열 배열 `data`로 주어진다. `data`의 원소는 각 프렌즈가 원하는 조건이 `N~F=0`과 같은 형태의 문자열로 구성되어 있다. 제한조건은 아래와 같다.

- $1 \leq n \leq 100$
- `data`의 원소는 다섯 글자로 구성된 문자열이다. 각 원소의 조건은 다음과 같다.
 - 첫 번째 글자와 세 번째 글자는 다음 8개 중 하나이다. {A, C, F, J, M, N, R, T} 각 각 어피치, 콘, 프로도, 제이지, 무지, 네오, 라이언, 튜브를 의미한다. 첫 번째 글자는 조

CODE FESTIVAL

건을 제시한 프렌즈, 세 번째 글자는 상대방이다. 첫 번째 글자와 세 번째 글자는 항상 다르다.

- 두 번째 글자는 항상 ~이다.
- 네 번째 글자는 다음 3개 중 하나이다. {=, <, >} 각각 같음, 미만, 초과를 의미한다.
- 다섯 번째 글자는 0 이상 6 이하의 정수의 문자형이며, 조건에 제시되는 간격을 의미한다. 이때 간격은 두 프렌즈 사이에 있는 다른 프렌즈의 수이다.

출력 형식

모든 조건을 만족하는 경우의 수를 리턴한다.

예제 입출력

N	data	answer
2	["N~F=0", "R~T>2"]	3648
2	["M~C<2", "C~M>1"]	0

예제에 대한 설명

첫 번째 예제는 문제에 설명된 바와 같이, 네오는 프로도와 의 간격이 0이기를 원하고 라이언은 튜브와의 간격이 2보다 크기를 원하는 상황이다.

두 번째 예제는 무지가 콘과의 간격이 2보다 작기를 원하고, 반대로 콘은 무지와 의 간격이 1보다 크기를 원하는 상황이다. 이는 동시에 만족할 수 없는 조건이므로 경우의 수는 0이다.



문제 2

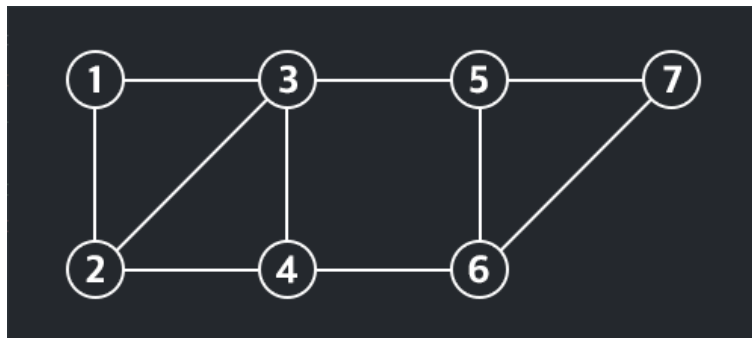
GPS

카카오 택시 개발자 Jay-G는 다음 업데이트를 준비하기 위해 개선사항을 위한 여러 피드백을 받았다. 그중에서 손님이 자주 탑승하는 위치를 추천해주었으면 한다는 의견이 많았다.

다음 업데이트 준비를 위해 Jay-G는 택시의 승하차 및 이동 경로를 수집하여 분석하기 시작하였다. 데이터를 분석하던 Jay-G는 몇 가지 특이사항을 발견했다. 택시의 이동 경로를 GPS를 통해 수집하게 되는데, GPS 신호 불량, 통신 오류 등 다양한 원인으로 위치의 오류가 발생한 것을 알게 되었다. 다만 승차 위치와 하차 위치는 오류가 없는 것으로 확인이 되었다.

개발자 Jay-G는 수집한 이동 경로의 오류를 최소한으로 수정하여 좀 더 정확한 이동 경로를 구하고 싶어 한다.

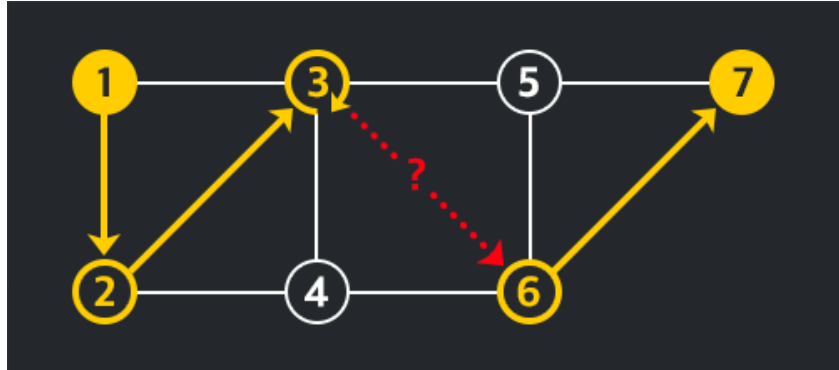
택시는 다음과 같은 조건으로만 이동한다. 먼저 택시는 거점을 이동해 다니며, 거점 간의 이동은 해당하는 도로가 있는 경우에만 가능하다. 또한, 교통 상황에 따라 택시는 한 거점에 머무를 수 있고, 왔던 길을 되돌아갈 수 있다. 모든 도로는 방향이 별도로 없는 왕복 도로이다.



예를 들어, 위 그래프에서 택시가 다음과 같이 시간대별로 이동 경로를 보내왔다.

CODE FESTIVAL

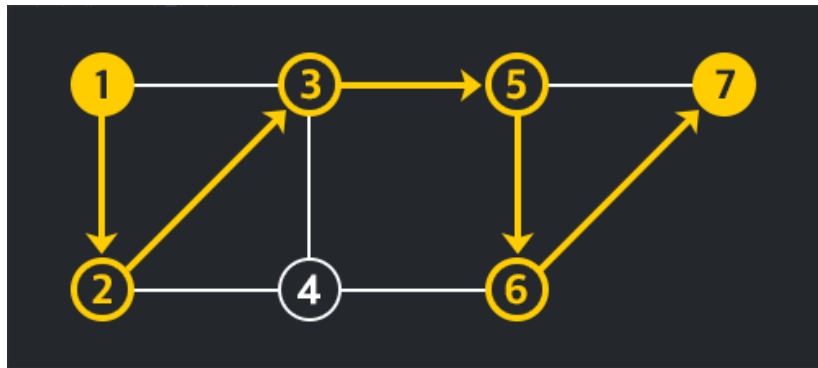
t	위치
1	1
2	2
3	3
4	3
5	6
6	7



하지만 위의 택시가 보내온 경로에는 거점 3에서 거점 6으로 이동할 수 있는 도로가 없으므로 이동 경로에 오류가 있다.

이러한 오류를 최소한으로 수정하여 이동 가능한 경로로 만들고 싶다. 이 경우 1회의 오류를 수정하여 다음과 같이 이동 가능한 경로를 만들 수 있다. 시간 $t=4$ 의 위치를 거점 5로 한 번 수정하면 이동 가능한 경로가 된다.

t	위치
1	1
2	2
3	3
4	5
5	6
6	7



이와 비슷하게 시간 $t=4$ 의 위치를 거점 4로 바꾸거나, 시간 $t=5$ 위치를 거점 5로 바꾸면 이동 가능한 경로로 만들 수 있다. 위의 경우 수정한 오류의 개수는 1개이다.

CODE FESTIVAL

(t=4 위치를 거점 4로 변경)

t	위치
1	1
2	2
3	3
4	4
5	6
6	7

(t=5 위치를 거점 5로 변경)

t	위치
1	1
2	2
3	3
4	3
5	5
6	7

위와 같이 택시가 보내온 경로에서 이동 가능한 경로로 만드는 최소의 오류 수정 횟수를 구하여라.

입력 형식

주어지는 입력은 총 다섯 가지로, 거점 개수 n 과 도로의 개수 m , 각 거점 간의 연결된 도로 정보 `edge_list`, 택시가 시간대별로 보내오는 거점 정보의 총 개수 k , 그리고 머물렀던 거점의 정보 `gps_log`이다. 제한조건은 아래와 같다.

- $2 \leq n \leq 200$
- $1 \leq m \leq 10,000$
- $2 \leq k \leq 100$
- `edge_list`는 $m \times 2$ 크기의 2차원 배열로, 각 행의 두 값은 도로가 있는 두 거점의 번호를 의미한다.
- 거점의 번호는 1부터 n 까지 숫자이다.
- 모든 도로는 양방향 통행이 가능하다.
- 입력되는 데이터에서 항상 모든 거점 간 경로가 있음이 보장되지 않는다.
- `gps_log`의 시작 거점과 도착 거점은 바뀔 수 없다.

출력 형식

이동 가능한 경로로 만들 수 있는 최소의 오류 수정 횟수를 리턴한다. 올바른 경로로 수정하는 것이 불가능할 경우 -1을 리턴한다.

예제 입출력

변수명	값
n	7
m	10
edge_list	[[1, 2], [1, 3], [2, 3], [2, 4], [3, 4], [3, 5], [4, 6], [5, 6], [5, 7], [6, 7]]
k	6
gps_log	[1, 2, 3, 3, 6, 7]
answer	1

변수명	값
n	7
m	10
edge_list	[[1, 2], [1, 3], [2, 3], [2, 4], [3, 4], [3, 5], [4, 6], [5, 6], [5, 7], [6, 7]]
k	6
gps_log	[1, 2, 4, 6, 5, 7]
answer	0

예제에 대한 설명

두 예제 모두 `edge_list`의 데이터는 본문의 그림과 같은 예이다.

첫 번째 테스트 케이스에서 `gps_log`로 주어진 경로 중 거점 3에서 거점 6으로 가는 도로가 없다. 여기서 시간 `t=4`의 위치를 거점 5로 한 번 수정하면 이동 가능한 경로가 된다.

두 번째 테스트 케이스는 `gps_log`로 주어진 경로가 모두 도로로 연결된 경우이므로 수정이 필요 없다.



문제 3

리틀 프렌즈 사천성

언제나 맛있는 음식들이 가득한 평화로운 푸드 타운.
푸드 타운에서 행복하게 사는 리틀 프렌즈들은
마을에 있는 매직 스푼을 보물처럼 보관하고 있다.

매직 스푼은 재료만 준비해서 냄비에 넣고 휘젓기만 하면
순식간에 최고의 요리로 만들어주는 신비의 아이템.
어느 날 매직 스푼을 호시탐탐 노리는 악당들이 보물을 훔쳐간다.

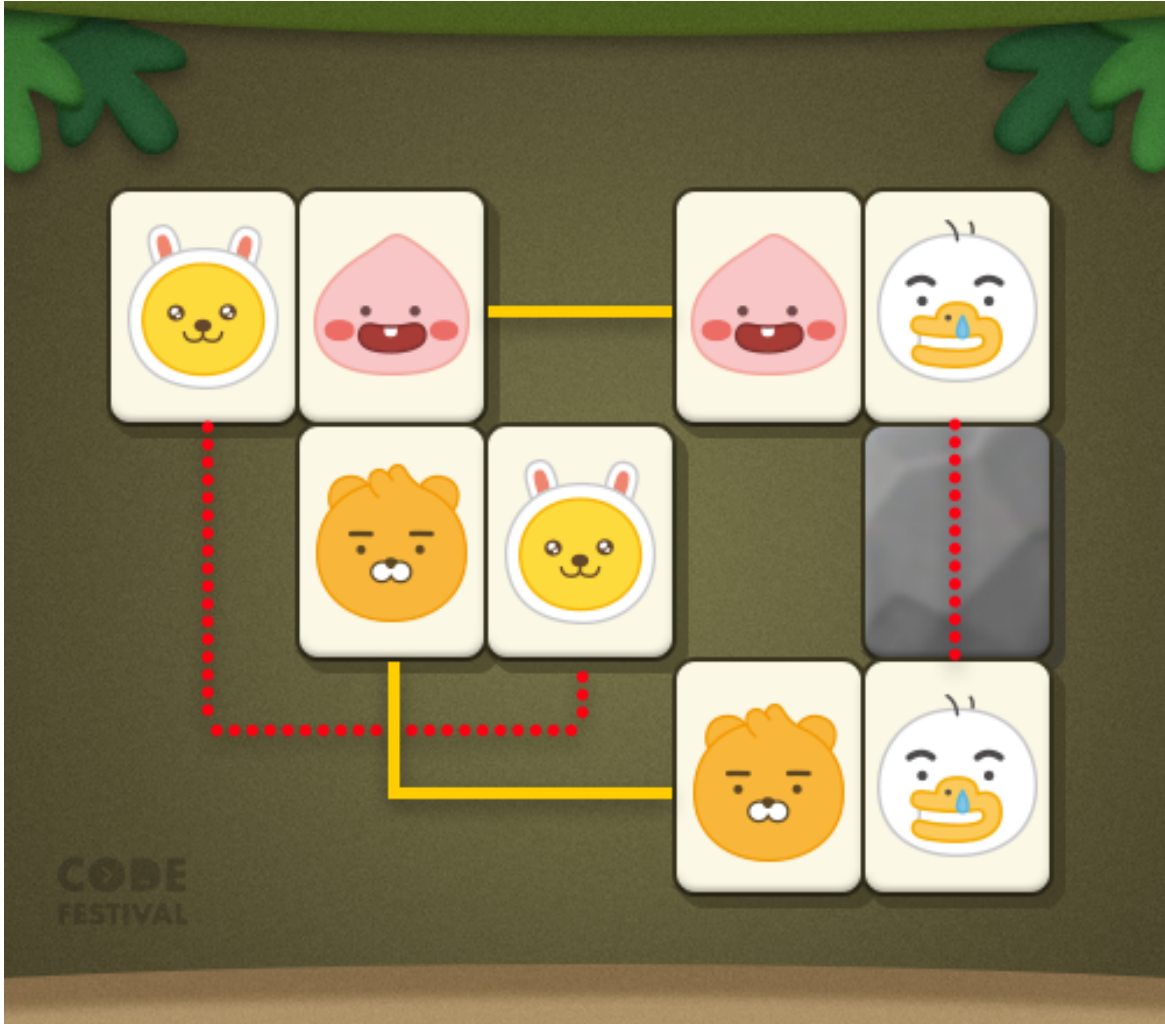
매직 스푼을 되찾고 다시 마을에 평화를 가져오기 위해
프렌즈의 대모험이 시작되는데...

리틀 프렌즈 사천성은 프렌즈 사천성과 유사한 게임이다. 게임은 2차원 배열에서 진행되는데, 여러 가지 무늬로 구성된 타일이 배치되어 있으며 같은 모양의 타일은 두 개씩 배치되어 있다. 게임의 목적은 배치된 모든 타일을 제거하는 것으로, 같은 모양의 타일을 규칙에 따라 제거하면 된다. 타일을 제거할 수 있는 경우는 다음과 같다.

다음 조건을 만족하는 경로가 있을 때 두 타일을 제거할 수 있다.

- 경로의 양 끝은 제거하려는 두 타일이다.
- 경로는 두 개 이하의 수평/수직 선분으로 구성되어 있고, 이들은 모두 연결되어 있다. (즉, 경로를 한 번 이하로 꺾을 수 있다)
 - 참고: 프렌즈 사천성은 경로가 세 개 이하의 선분으로 구성되어야 한다는 점이 다르다. (즉, 경로를 두 번 이하로 꺾을 수 있다)
- 경로 상에는 다른 타일 또는 장애물이 없어야 한다.

CODE FESTIVAL



위의 배열에서 어피치 타일은 직선의 경로로 이을 수 있으므로 제거 가능하다. 라이언 타일 역시 한번 꺾인 경로로 연결되므로 제거 가능하다. 무지 타일의 경우 다른 타일을 지나지 않는 경로는 두 번 꺾여야 하므로 제거할 수 없는 타일이며, 튜브 타일 역시 직선의 경로 사이에 장애물이 있으므로 제거 가능하지 않다.

타일 배열이 주어졌을 때, 규칙에 따라 타일을 모두 제거할 수 있는지, 그 경우 어떤 순서로 타일을 제거하면 되는지 구하는 프로그램을 작성해보자.

입력 형식

입력은 게임판의 크기를 나타내는 m 과 n , 그리고 배치된 타일의 정보를 담은 문자열 배열 `board`로 주어진다. 이 배열의 크기는 m 이며, 각각의 원소는 n 글자의 문자열로 구성되어 있다. 입력되는 값의 제

한조건은 다음과 같다.

- $1 \leq m, n \leq 100$
- board의 원소는 아래 나열된 문자로 구성된 문자열이다. 각 문자의 의미는 다음과 같다.
 - . : 빈칸을 나타낸다.
 - * : 막힌 칸을 나타낸다.
 - 알파벳 대문자(A-Z) : 타일을 나타낸다. 이 문제에서, 같은 글자로 이루어진 타일은 한 테스트 케이스에 항상 두 개씩만 존재한다.
 - board에는 알파벳 대문자가 항상 존재한다. 즉, 타일이 없는 입력은 주어지지 않는다.

출력 형식

해가 존재하는 경우 타일을 제거하는 순서대로 한 글자씩 이루어진 문자열을, 그렇지 않은 경우 IMPOSSIBLE을 리턴한다. 해가 여러 가지인 경우, 알파벳 순으로 가장 먼저인 문자열을 리턴한다.

예제 입출력

m	n	board	answer
3	3	["DBA", "C*A", "CDB"]	"ABCD"
2	4	["NRYN", "ARYA"]	"RYAN"
4	4	[".ZI.", "M.**", "MZU.", ".IU."]	"MUZI"
2	2	["AB", "BA"]	"IMPOSSIBLE"

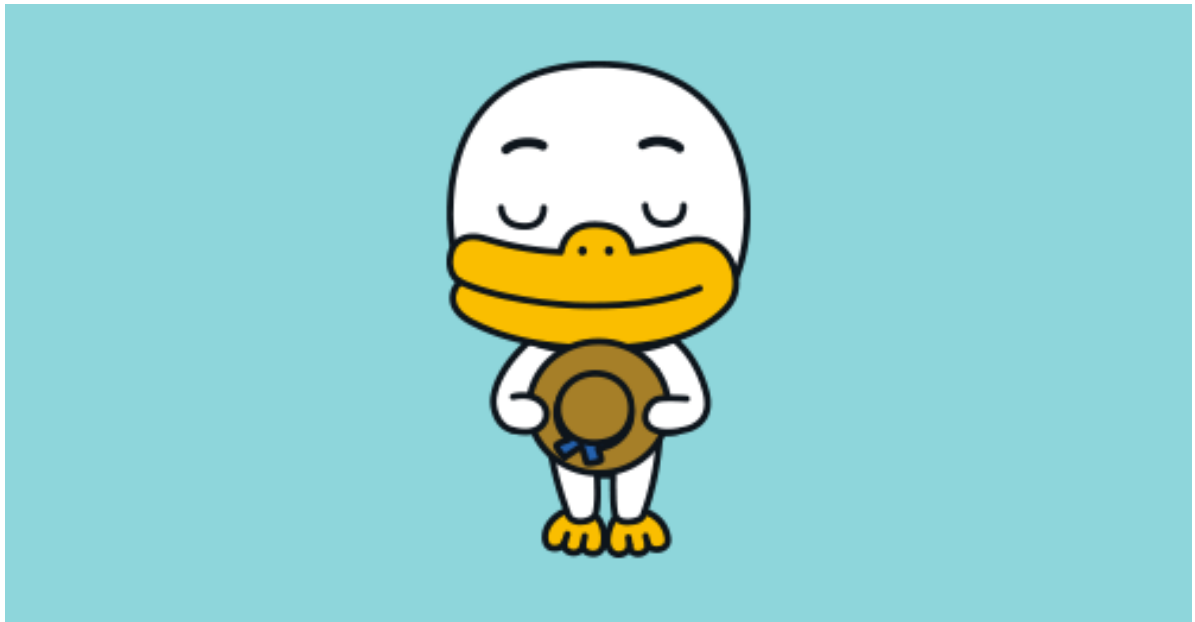
예제에 대한 설명

첫 번째 테스트 케이스에서 처음으로 제거 가능한 타일은 A와 C이다. 그리고 모든 가능한 경우를 나열하면 ABCD, ACBD, ACDB, CABD, CADB, CDAB이다. 이 중 알파벳 순으로 가장 먼저인 ABCD가 정답이다. 네 번째 테스트 케이스는 초기 상태에서 제거할 수 있는 타일이 없으므로 타일을 모두 제거하는 것이 불가능하다. 따라서 정답은 IMPOSSIBLE이다.



문제 4

튜브의 소개팅



얼마 전 소개팅에 실패해 낙담한 튜브를 안타깝게 여긴 친구들은 튜브에게 새로운 사람을 소개해주기로 했다. 좀 더 자연스러운 자리를 만들기 위해, 튜브와 소개팅녀를 파티에 초대하기로 했다.

친구들은 튜브에게 파티에 초대된 모든 사람의 위치를 알려주었다. 사각형 모양의 파티장 입구는 왼쪽 맨 위였고, 만나야 하는 상대의 좌석은 파티장의 오른쪽 맨 아래였다. 파티장의 가로 길이가 n 이라고 하고, 세로 길이를 m 이라 할 때, 튜브와 소개팅 상대의 위치는 각각, $(0, 0)$, $(m - 1, n - 1)$ 이 된다. 튜브는 $(0, 0)$ 에서 $(m - 1, n - 1)$ 까지 가는 가장 짧은 길을 알고 싶다. 이동은 좌/우/상/하로만 가능하다. 또한, 조금이라도 더 빠르게 이동하고 싶은 튜브는 이동하는 중에 가능한 덜 수다스러운 친구들만 만나고 싶다. 다시 말해, 목표 지점까지 길이가 같은 여러 개의 경로가 존재할 경우, 경로상에 위치한 친구들과 나눠야 하는 대화 시간의 합이 적은 것을 더 선호한다.

튜브에게는 스트레스를 심하게 받을 경우 미친 오리로 변하는 비밀이 있다. 경로 상에 위치한 친구들과 나눠야 하는 대화 시간의 합이 s 를 초과한다면 미친 오리로 변하고 소개팅에 실패하고 말 것이다! 그러므로 아무리 짧은 경로라도 친구들과 나눠야 하는 대화 시간의 합이 s 를 초과한다면 그 경로를 진행할 수는 없다.

CODE FESTIVAL



튜브가 소개팅 상대에게 무사히 갈 수 있는 경로를 알려주자.

입력 형식

입력은 파티장의 크기 m 과 n 그리고 튜브가 참을 수 있는 대화 시간의 총합 s , 친구와의 대화에 필요한 시간 t 가 담긴 2차원 배열 `time_map`으로 주어진다. t 가 -1 인 경우는 파티 테이블이 위치한 경우라 지나갈 수 없다. 그리고 시작 지점과 도착 지점에서는 수다에 필요한 시간이 없다, 즉 $t = 0$ 이다. 추가 제한조건은 아래와 같다.

- $2 \leq n, m \leq 50$
- $1 \leq s \leq 2^{31}-1$
- $-1 \leq t \leq 2^{31}-1$
- 모든 입력에는 문제의 조건을 만족하는 경로가 하나 이상 존재한다.

출력 형식

리턴 타입은 원소가 두 개인 정수 배열이다. 튜브가 이동해야 하는 경로의 길이와 해당 경로를 지나갈 때 친구와 나눠야 하는 대화 시간의 총합을 리턴한다.

CODE FESTIVAL

예제 입출력

m	n	s	time_map	answer
3	3	150	[[0, 2, 99], [100, 100, 4], [1, 2, 0]]	[4, 103]
4	6	25	[[0, 1, 1, -1, 2, 4], [-1, 7, 2, 1, 5, 7], [-1, 1, -1, 1, 6, 3], [-1, 1, -1, -1, 7, 0]]	[8, 15]
5	5	12	[[0, 1, 1, 1, 1], [9, 9, 9, 1, 9], [1, 1, 1, 1, 9], [1, 1, 5, 9, 9], [1, 1, 1, 1, 0]]	[12, 11]



문제 5

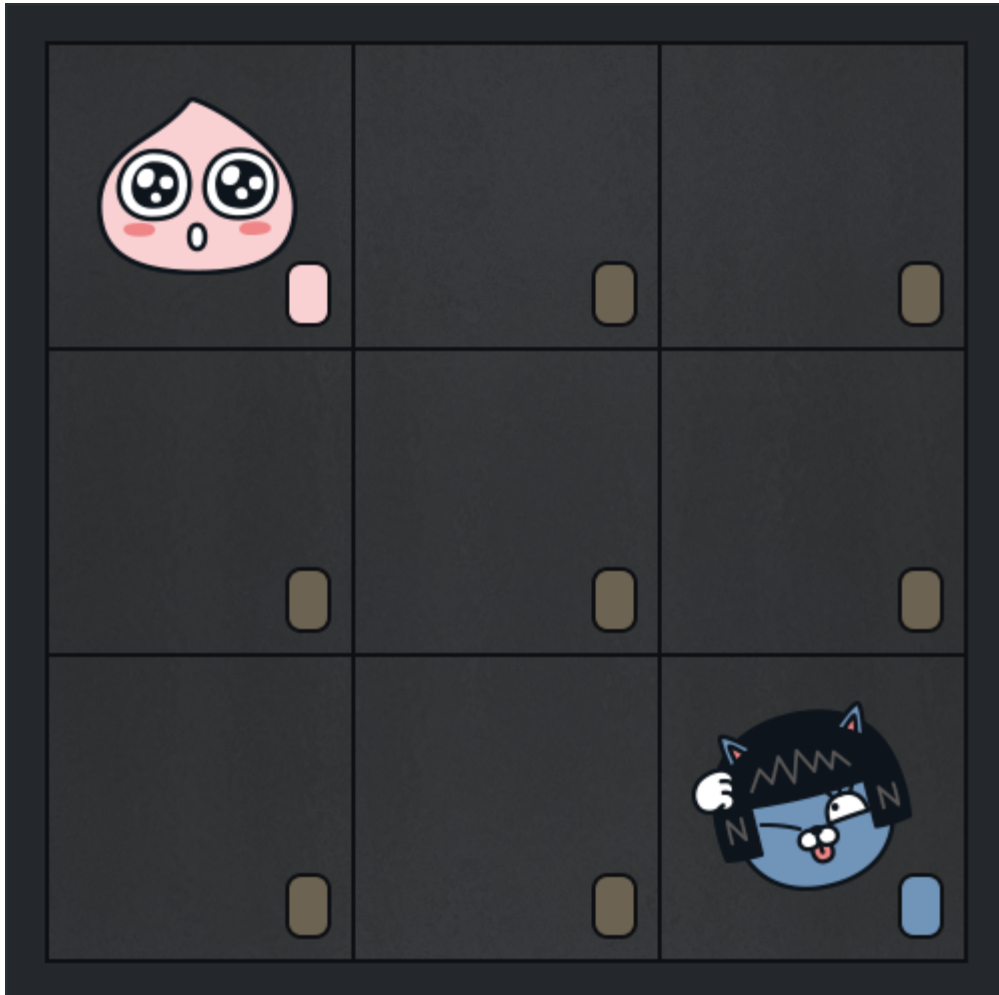
몸짱 트레이너 라이언의 고민



헬스장에서 일하는 몸짱 트레이너 라이언은 최근 손님들에게 불평을 많이 들었다. 그것은 옷을 갈아입는데 다른 사람과 너무 가까워서 옷을 갈아입기가 불편하다는 것이었다.

불만을 해결하기 위해 고민하던 라이언은 손님들의 예약시간을 참고해서 되도록이면 서로 멀리 떨어지도록 키를 나눠주기로 마음먹었다. 예를 들어, 락커들이 3x3 정사각형 모양으로 배치되어 있고, 동시간대에 2명의 손님이 예약되어 있다면 아래와 같이 락커를 할당하는 것을 고려해볼 수 있다.

C>DE FESTIVAL



라이언이 일하는 헬스장은 아래와 같은 상황이라고 가정하자.

- 락커는 정사각형으로 배치되어있고, 락커 사이에 옷을 갈아입을 공간이 있다. 단, 이 공간은 계산에서 제외된다.
- 락커 간 거리는 상하좌우는 1로, 대각선은 2로 계산한다.
- 손님들은 철저히 예약시간에 맞춰 락커를 이용하고, 퇴실하는 시간까지 락커를 차지한다.
- 영업시간은 오전 10시부터 오후 10시까지이다.
- 헬스장은 예약제로 운영되므로 락커의 개수보다 더 많은 손님들이 동시간대에 몰리는 경우는 없다.

이런 조건에서 헬스장을 이용한 손님들 중 가장 가까웠던 손님 간의 거리는 얼마일까?

CODE FESTIVAL

입력 형식

입력은 정사각형 한 변의 길이 n 과 손님수 m , 그리고 각 손님의 예약된 입실/퇴실 시간 `timetable`로 주어진다. 제한조건은 다음과 같다.

- $0 < n \leq 10$
- $0 \leq m \leq 1,000$
- `timetable`은 $m \times 2$ 크기의 2차원 배열이다. 각 행은 손님의 입실시각과 퇴실시각이 분 단위로 환산된 값 ($t1, t2$)가 들어있다.
 - $t1, t2$ 에 대해서는 다음 조건이 성립한다. $600 \leq t1 < t2 \leq 1,320$

출력 형식

할당된 락커들 간 거리 중 최소거리를 리턴한다. 손님 간에 이용 시간이 한 번도 겹치지 않을 경우에는 0을 리턴한다.

예제 입출력

n	m	timetable	answer
3	2	[[1170,1210], [1200,1260]]	4
2	1	[[840,900]]	0
2	2	[[600,630],[630,700]]	2
4	5	[[1140,1200],[1150,1200],[1100,1200], [1210,1300],[1220,1280]]	4

예제에 대한 설명

- 첫 번째 예제: 손님 2명이 20:00부터 20:10까지 이용시간이 겹치고, 이 상황은 본문에 소개된 예와 같다.
- 두 번째 예제: 손님이 1명으로 겹치는 시간이 없기 때문에 0을 출력한다.
- 세 번째 예제: 손님 2명이 10:30에 겹친다. 2×2 정사각형에 2명을 배치해야 하므로 정답은 2가 된다. (ex. (0,0), (1,1)에 배치)
- 네 번째 예제: 손님 3명이 19:10부터 20:00까지 이용시간이 겹친다. 이 경우 아래와 같이 배치해서 할당된 모든 락커 간 거리가 4가 되도록 할 수 있다. (할당된 락커는 #으로 표시)

CODE FESTIVAL

0 0 0

0 0 0 #

0 0 0 0

0 # 0 0

- 네 번째 예제에서 손님을 모서리부터 배치하는 건 좋은 전략이 아니다. 2명까지는 거리가 6으로 가장 먼 거리가 되지만 예약된 다음 손님이 들어오면 할당된 락커 간 거리는 3이 돼버린다. 이런 경우 라이언은 예약시간을 고려해서 거리가 4가 되도록 배치한다.

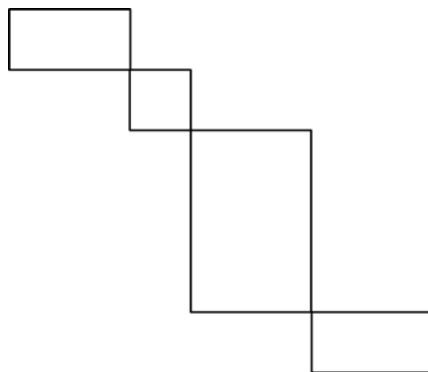


문제 6

네오의 귀걸이



귀걸이를 좋아하는 네오는 자신만을 위한 귀걸이를 만들기로 하였다. 기하학적으로 아름다운 귀걸이를 만들고 싶던 네오는 아래와 같은 모습의 귀걸이를 구상 중이다.



2차원 좌표 평면에 N 개의 점이 주어져 있다. 점들의 x 나 y 좌표가 겹치는 경우는 없다고 하자. 이들 점 중 가장 왼쪽에 위치한 점은 항상 가장 위쪽에 위치한 점과 같다. 이 점을 S 라고 부르자. 마찬가지로, 가장 오른쪽에 위치한 점은 항상 가장 아래쪽에 위치한 점과 같다. 이 점을 E 라고 부르자.

CODE FESTIVAL

기하학적으로 아름다운 귀결이는 다음의 조건을 만족해야 한다. 주어진 점들을 왼쪽 위와 오른쪽 아래 꼭짓점으로 가지는 축에 평행한 직사각형들이 순서대로 놓여있다. 이때, 첫 번째 직사각형은 S를 왼쪽 위 꼭짓점으로 가져야 한다. 순서의 마지막 직사각형은 E를 오른쪽 아래 꼭짓점으로 가져야 한다. 순서에서, 어떤 직사각형 A 바로 다음에 직사각형 B가 등장한다면, A의 오른쪽 아래 꼭짓점은 B의 왼쪽 위 꼭짓점과 같은 점이어야 한다.

주어진 점 각각은 자신의 특징적인 k값을 가지고 있다. 이 값의 의미는 해당 점이 어떤 직사각형의 오른쪽 아래 점이 되는 경우 그 직사각형의 가로 폭과 세로 높이의 합이 정확히 k가 되어야 한다는 뜻이다.

네오는 주어진 점들로 만들 수 있는 기하학적인 귀결이가 몇 가지인지 궁금하다. 또, 점을 하나 추가했을 때 혹은 점을 하나 제거했을 때의 종류도 몇 가지인지 궁금하다. 네오가 궁금증을 해결하기 위해 여러분의 도움이 필요하다.

S와 E는 처음 상태에서 고정된다. S나 E가 제거되는 경우와 추가되는 점이 일정한 영역을 벗어나는 경우에 주의하라.

입력 형식

입력은 점의 개수 n , 점의 정보를 담은 배열 `point`, 점의 추가 또는 삭제 정보를 담은 배열 `query`로 구성된다. `point` 배열은 $n \times 3$ 크기의 2차원 배열로, 각각의 행은 점의 x좌표, y좌표, 그리고 그 점의 k값으로 이루어져 있다. `query` 배열은 $n \times 4$ 크기의 2차원 배열로, 점을 추가하거나 삭제하는 변화의 정보를 담고 있다. 각 변화는 독립적으로 초기 상태에 대해서 적용됨에 주의하라. 즉, 변화가 누적되는 것이 아니다. 각 행의 첫 번째 값은 변화의 종류를 담은 값으로, 0인 경우 점을 추가하는 경우를 의미하고 1인 경우 최초에 주어진 n 개의 점들 중 하나를 제거하는 경우를 의미한다. 점을 추가하는 경우는 새로 추가되는 점의 x좌표, y좌표, k값이 이후의 원소로 주어진다. 점을 삭제하는 경우는 삭제하려는 점의 x좌표, y좌표가 이후의 원소로 주어지며, 마지막 원소는 배열의 크기를 맞추기 위한 것으로 의미 없는 값이다. 제한조건은 다음과 같다.

- $2 \leq n \leq 100,000$
- 주어진 점들 중 가장 왼쪽에 위치한 점은 항상 가장 위쪽에 위치한 점과 같음이 보장된다. 또한, 가장 오른쪽에 위치한 점은 항상 가장 아래쪽에 위치한 점과 같음이 보장된다.
- 추가되는 점은 최초에 주어진 점과 x좌표나 y좌표가 겹치지 않음이 보장된다.
- 제거되는 점은 최초에 주어진 점들 중 하나임이 보장된다.
- 모든 좌표 값은 $-1,000,000,000$ 이상 $1,000,000,000$ 이하의 정수이다.

CODE FESTIVAL

출력 형식

리턴 타입은 정수의 배열로, 총 $n+1$ 개의 원소로 이루어진 배열을 리턴한다. 첫 번째 원소는 초기 상태에서 가능한 기하학적 귀결의 개수를 $1,000,000,007$ 로 나눈 나머지로 한다. 다음 n 개의 원소는 각 변화에 대해 그 변화를 적용한 이후 가능한 귀결의 개수를 $1,000,000,007$ 로 나눈 나머지가 입력의 순서에 맞게 들어있으면 된다.

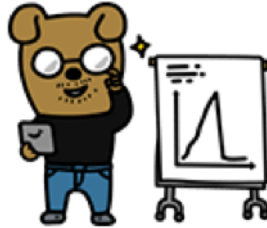
예제 입출력

n	point	query	answer
3	[[0, 4, 1], [2, 2, 4], [4, 0, 4]]	[[0, 3, 3, 4], [0, 3, 1, 2], [1, 2, 2, 0]]	[1, 2, 1, 0]



문제 7

스마트한 프로도



직장인 프로도는 오늘도 열심히 일한다. 회사에서 인정받는 프로도는 직장 상사에게 문제를 풀어 달라는 요청을 받았다. 문제를 읽은 프로도는 혼자 풀긴 어려운 문제로 보여 여러분에게 도움을 요청하였다. 상사가 전달한 문제는 아래와 같다.

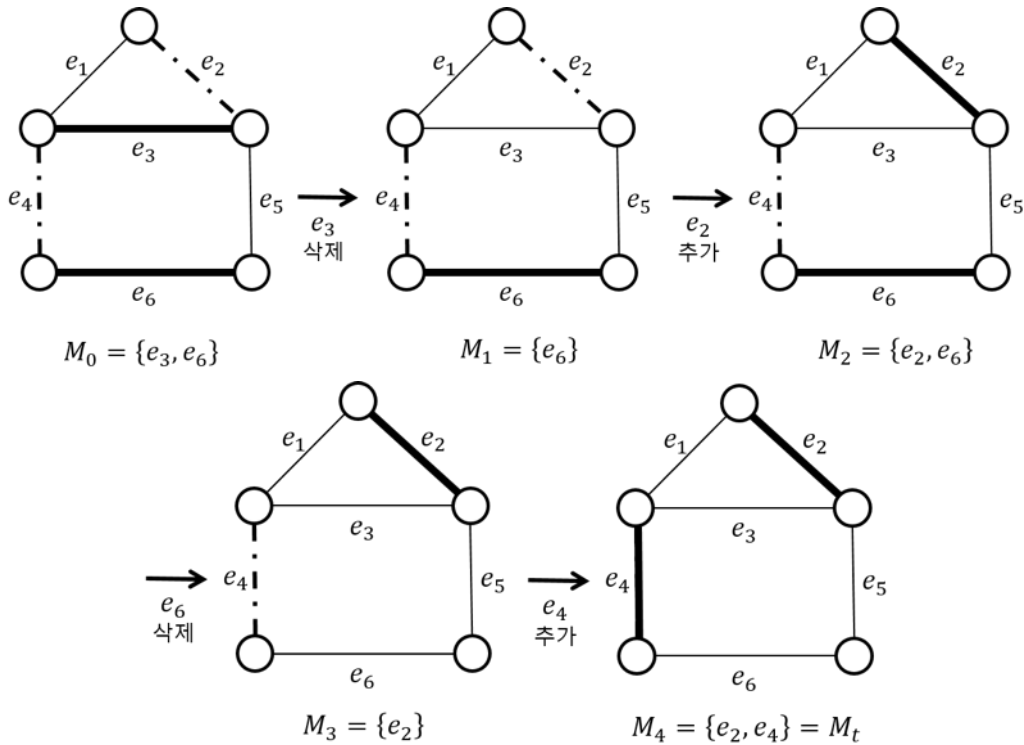
그래프 G 에 대해서, 서로 다른 두 정점 사이에 간선이 존재한다면 단지 한 간선만 존재한다. 또한 동일한 정점을 연결하는 간선(셀프 루프)은 존재하지 않는다. G 의 두 간선 e_1 과 e_2 가 인접하다면, e_1 이 연결하는 두 정점 중 하나는 e_2 가 연결하는 정점과 일치한다. 그래프 G 의 매칭 M 은 간선들의 집합이고 M 에 속하는 임의의 두 간선은 서로 인접하지 않다. 여기서, 공집합은 매칭임에 주목하자.

그래프 G 와 정수 k 에 대해서, 초기에 $|M_0| \geq k$ 이고 $|M_t| \geq k$ 인 두 매칭 M_0 와 M_t 가 주어진다. 우리는 매칭 M_0 에서 G 의 간선을 추가하거나 또는 삭제해서 M_0 를 변환한다. 변환의 한 단계에서는 하나의 간선을 추가하거나 삭제할 수 있다. 이렇게 해서 변환된 간선들의 집합 M_1 은 다시 매칭이 되어야 한다. 다시 말해서, 각 변환 단계의 결과물은 매칭이어야 한다. 따라서 i 번째 단계에서는 매칭 M_{i-1} 를 매칭 M_i 로 변환하게 된다.

이런 식으로 M_0 에서 시작해서 중간 매칭들로의 변환 단계들을 거쳐서 마지막에 M_t 를 만들어야 한다. 다시 말해서, p 번의 단계를 거쳐 만들어진 매칭 M_p 에 대해서, $M_p = M_t$ 를 만족하면 된다. 하지만 중간에 만들어지는 M_i ($0 < i < p$)는 크기에 제한이 있어서 $|M_i| \geq k-2$ 를 만족해야만 한다.

매칭 M_0 에서 M_t 로 위의 조건들을 만족하면서 항상 변환할 수 있다는 것이 잘 알려져 있다. 따라서 여러분은 그 변환 과정을 리턴하는 프로그램을 작성하여야 한다.

예를 들어, 아래 그림에서 초기 매칭 $M_0 = \{e_3, e_6\}$ 이고 마지막 매칭 $M_t = \{e_2, e_4\}$ 이고, $k = 2$ 인 경우이다. 그림에서와 같이 간선들을 추가하거나 삭제하면, M_0 에서 M_t 로 변환할 수 있다.



입력 형식

입력은 총 9가지의 변수로 주어진다. n 과 m 은 각각 그래프 G 의 정점과 간선의 수이다. 그래프 G 의 정점들은 1부터 n 까지의 정수로, 간선들은 1부터 m 까지의 정수로 나타낸다. a 와 b 는 각각 크기가 m 인 1차원 배열로, 간선이 있는 두 점을 나타낸다. i 번째 간선이 있는 두 정점의 번호가 a 와 b 의 i 번째 원소가 된다. k 는 문제에서 설명된 것과 같다. m_1 과 m_2 는 각각 초기 매칭 M_0 의 크기, 마지막 매칭 M_t 의 크기이다. e_1 과 e_2 는 각각 M_0 와 M_t 에 속하는 간선의 정보를 나타내는 1차원 배열이며, 각각의 원소의 개수는 m_1 과 m_2 이다. 제한조건은 다음과 같다.

- $1 \leq n \leq 100,000$
- $1 \leq m \leq 1,000,000$
- $1 \leq k \leq 50,000$
- $m_1 \geq k, m_2 \geq k$

출력 형식

리턴 타입은 2차원 정수 배열이다. 매칭 M_0 에서 M_t 로 변환하는데 p 단계가 걸린다고 할 때, 배열의 크기는 $p \times 2$ 가 되어야 한다. 각 행의 첫 번째 원소는 0 또는 1로, 0이면 간선을 삭제했음을 의미하며

1이면 간선을 추가했음을 의미한다. 두 번째 원소는 추가 또는 삭제하는 간선의 번호이다. 답이 될 수 있는 변환이 여러 가지인 경우에는 그중 한 가지를 리턴하면 된다.

예제 입출력

변수명	값
n	5
m	6
a	[1, 1, 2, 2, 3, 4]
b	[2, 3, 3, 4, 5, 5]
k	2
m1	2
m2	2
e1	[3, 6]
e2	[2, 4]
answer	[[0, 3], [1, 2], [0, 6], [1, 4]]

알림

이 문제의 경우 같은 입력에 대해 정답이 여러 개 존재할 수 있으나, '실행'을 눌러 예제 입출력에 대해 테스트를 진행할 때 등록된 한 가지 답만 정답으로 처리되고 있습니다. '코드 채점'을 눌러 제출할 시에는 올바르게 채점되니 참고하여 주시기 바랍니다.

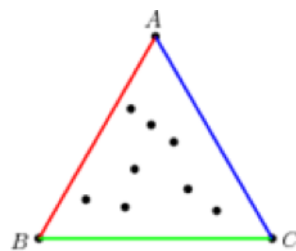


문제 8

IU와 콘의 보드게임



2017년 8월 5일, 다양한 방법으로 3단 고음을 마스터한 아이유는 신나게 놀기로 했다. 무엇을 할까 고민하던 그녀는 콘이 보드게임을 플레이하고 있는 것을 보고 흥미가 생겼다. 보드판은 다음과 같이 생겼다.



게임판에는 점 A, B, C 위치에 핀이 꼽혀 있으며 이 세 점은 게임판에서 가장 큰 삼각형을 이루고 있다. 고무줄이 세 개가 있는데 각각이 A와 B, B와 C, C와 A에 묶여 있어 삼각형 ABC를 그리고 있다. 이 커다란 삼각형 ABC 안에는 n 개의 핀이 무작위로 꼽혀 있다.

이 게임은 세 개의 고무줄을 삼각형 ABC의 안쪽으로 이동시켜 새로운 도형을 만들어 내는 것이 목표이며, 그 과정에서 핀의 위치는 고정되어 움직일 수 없다. 삼각형 ABC 내부의 핀들과 고무줄의 물리적

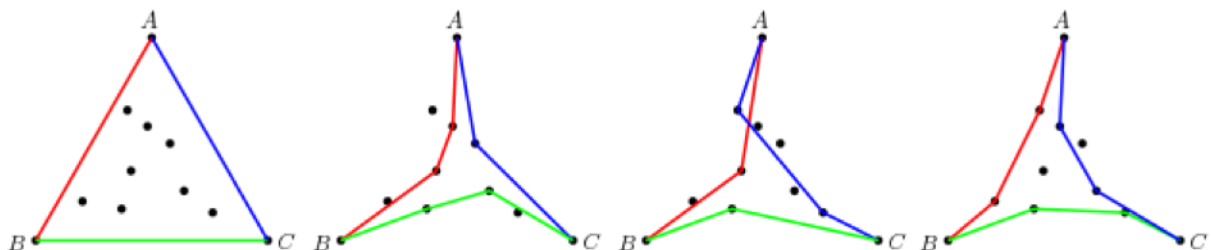
CODE FESTIVAL

특성에 의해 이러한 방식으로 만들어지는 도형들은 다음과 같은 성질을 갖는다.

- 꼭짓점은 모두 어떤 핀에 의해 만들어지며 변은 핀과 핀 사이의 선분이 된다.
- 끝점이 A와 B에 고정된 고무줄은 A에서 시작하여 B로 끝나는 안쪽으로 오목한 모양을 그리게 된다.
- 끝점이 B와 C에 고정된 고무줄은 B에서 시작하여 C로 끝나는 안쪽으로 오목한 모양을 그리게 된다.
- 끝점이 C와 A에 고정된 고무줄은 C에서 시작하여 A로 끝나는 안쪽으로 오목한 모양을 그리게 된다.

여기에 아래의 추가 조건을 만족하는 도형을 모두 찾고 싶다.

- 임의의 두 고무줄이 서로 교차하지 않으며, 점 A, B, C를 제외한 곳에서 겹치지 않는다. 따라서 만들어지는 도형의 내부와 외부가 명확하게 구분된다.
- 그 도형의 내부에 다른 핀이 존재하지 않는다.



위의 첫 번째 그림은 게임판의 시작 형태를 보여주고 있다. 여기에는 삼각형 ABC 안에 8개의 핀이 꽂혀 있으며, 세 개의 고무줄은 서로 다른 색으로 표현되어 있다. 두 번째 그림은 추가 조건 두 가지를 모두 만족하는 도형의 예를 보여준다. 세 번째와 네 번째 그림은 고무줄로 만들 수 있는 모양이지만 추가 조건을 만족시키지 못하는 예이다. 세 번째 도형은 두 고무줄이 교차하여 추가 조건 1을 만족하지 못하고 네 번째 도형은 추가 조건 1을 만족하지만 내부에 핀이 존재하여 조건 2를 만족하지 못한다.

위의 두 번째 도형과 같이 추가 조건 1과 2를 모두 만족하는 도형이 몇 개인지 맞추는 사람이 이기는 게임이다. 승부욕이 강한 아이유는 콘에게 지고 싶지 않아 여러분에게 도움을 요청했다. 삼각형 ABC 내부에 위치한 n 개의 핀의 좌표가 주어질 때 위의 조건을 만족하는 도형의 개수를 세는 프로그램을 작성하시오.

CODE FESTIVAL

입력 형식

입력은 삼각형 내부의 핀의 개수 n , 삼각형의 좌표 `triangle`, 핀의 좌표 `v` 배열로 구성되어 있다. `triangle`은 3×2 크기의 2차원 배열이며, `v`는 $n \times 2$ 크기의 2차원 배열이다. 제한조건은 다음과 같다.

- $0 \leq n \leq 500$
- `triangle`과 `v`의 원소인 x 좌표와 y 좌표는 모두 $-100,000,000$ 이상 $100,000,000$ 이하이다.
- `triangle`과 `v`로 주어진 모든 점을 통틀어, 같은 좌표를 가진 두 핀이 입력으로 주어지지 않는다.
- `triangle`과 `v`로 주어진 모든 점을 통틀어, 임의의 세 점이 하나의 직선 위에 놓이는 경우는 없다.

출력 형식

문제에 설명된 조건을 모두 만족하는 도형의 개수를 리턴한다.

예제 입출력

n	triangle	v	answer
2	[[0, 8], [-4, 0], [4, 0]]	[[-1, 3], [1, 4]]	6
10	[[[-13, 40], [-27, -48], [31, 6]]	[[2, 21], [15, -1], [6, -13], [-20, -27], [5, 9], [5, 11], [-18, -26], [-15, 19], [7, 4], [-5, -18]]	73

예제에 대한 설명

첫 번째 예제에는 삼각형의 꼭짓점을 제외하고 총 2개의 핀의 위치가 주어졌다. 삼각형의 좌표를 `A(0, 8)`, `B(-4, 0)`, `C(4, 0)`, 그리고 삼각형 내부의 핀을 `D(-1, 3)`, `E(1, 4)`라고 하자. 고무줄로 만들 수 있으면서 추가 조건 1, 2를 만족하는 도형은 총 6개로 아래와 같다.

사각형 `ABCD`, 사각형 `AEBC`, 오각형 `ADBCE`, 오각형 `ABDCE`, 오각형 `ADBEC`, 오각형 `ABDEC`

그 외의 도형은 모두 추가 조건 1 혹은 추가 조건 2를 만족하지 못한다.