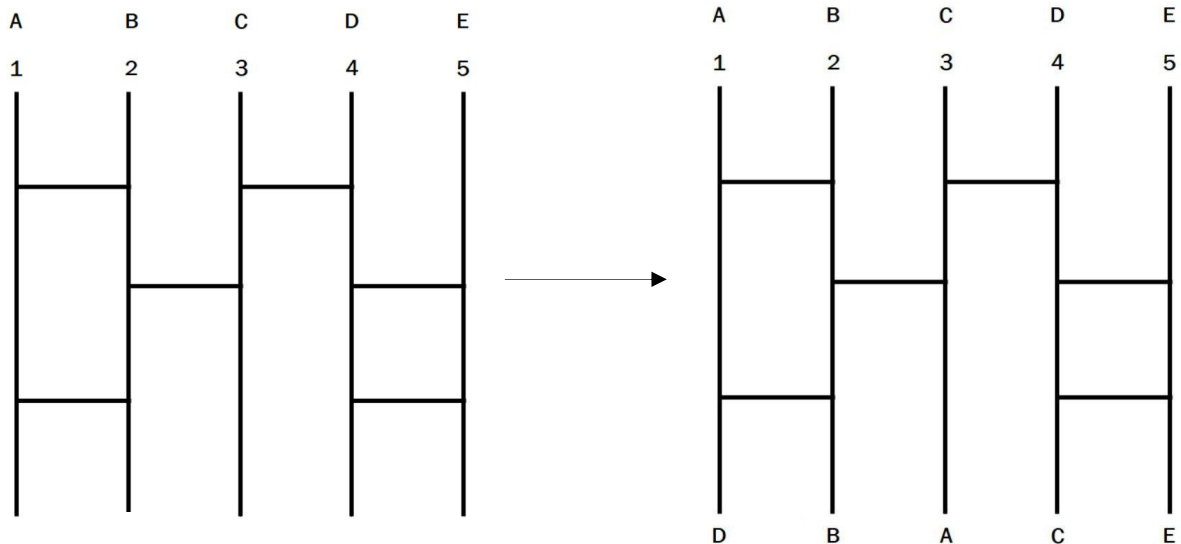


사다리 타기

현수네 반에는 n 명의 학생이 있습니다. 선생님은 n 명의 학생이 모두 사다리타기를 한 다음 당첨된 학생을 이 번주 학급회장으로 선출하려고 합니다.

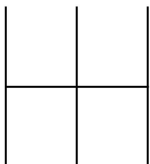
각 학생은 알파벳 대문자로 표시됩니다.

만약 $n=5$ 이고 아래와 같은 사다리라면



위에 사다리는 세로 라인이 1부터 5까지로 표현는 5개의 세로줄과 3개의 가로줄을 가지고 있습니다. 첫 번째 가로줄은 1번 세로줄과 2번 세로줄을 연결한 가로막대와 3번 세로줄과 4번 세로줄을 연결한 가로막대 2개가 있는데 이를 표현하는 방법은 $[1, 3]$ 으로 표현합니다. 즉 가로막대가 연결하고 있는 세로줄 중 왼쪽 세로줄 번호만 알려주는 형식입니다. 예를 들어 어떤 가로줄의 입력정보가 $[1, 3, 5]$ 로 표현된다면 이 가로줄에는 1번 세로줄과 2번 세로줄은 연결한 가로막대, 3번 세로줄과 4번 세로줄은 연결한 가로막대, 5번 세로줄과 6번 세로줄은 연결한 가로막대 이렇게 3개의 가로막대가 존재한다는 것입니다.

아래 그림처럼 가로줄의 정보는 $[1, 2]$ 와 같이 두 가로막대가 직접연결되는 경우는 입력되지 않습니다.



위에 사다리의 정보는 $[[1, 3], [2, 4], [1, 4]]$ 와 같이 첫 번째 가로줄부터 순서대로 입력정보가 2차원 배열로 주어집니다.

사다리를 타는 학생은 알파벳순으로 1번 세로줄부터 순서대로 사다리를 탑니다.

매개변수 n 에 학생수, ladder에 사다리의 정보가 주어지면, 모든 학생이 사다리를 탄 결과를 담은 배열을 반환하는 프로그램을 작성하세요.

입출력 예:

n	ladder	answer
5	[[1, 3], [2, 4], [1, 4]]	['D', 'B', 'A', 'C', 'E']
7	[[1, 3, 5], [1, 3, 6], [2, 4]]	['A', 'C', 'B', 'F', 'D', 'G', 'E']
8	[[1, 5], [2, 4, 7], [1, 5, 7], [2, 5, 7]]	['C', 'A', 'B', 'F', 'D', 'E', 'H', 'G']
12	[[1, 5, 8, 10], [2, 4, 7], [1, 5, 7, 9, 11], [2, 5, 7, 10], [3, 6, 8, 11]]	['C', 'A', 'F', 'B', 'D', 'T', 'E', 'K', 'G', 'L', 'J', 'H']

제한사항:

- $3 \leq n \leq 25$ 입니다.
- 매개변수 ladder의 길이(사다리 가로줄의 개수)는 1,000을 넘지 않습니다.
- 매개변수 ladder[i]의 길이는 10을 넘지 않습니다.

청소

청소로봇이 방을 청소하려고 합니다. 방은 $n \times n$ 격자판 지도로 표현됩니다.

방에는 장애물이 있고, 장애물이 있는 지점은 로봇이 지나갈 수 없습니다.

로봇은 지도의 왼쪽 가장 위 격자에서 3시 방향(오른쪽)을 보고 있습니다.

로봇이 한 격자를 이동하는데 걸리는 시간은 1초입니다.

로봇은 매초 한 칸씩 보고 있는 방향으로 이동합니다. 만약 지도 끝으로 이동해 더 이상 전진할 수 없거나 또는 장애물을 만나면 제자리에서 시계방향으로 90도 회전합니다. 회전하는데도 1초의 시간이 필요합니다.

매개변수 board에 방의 지도정보가 주어지고, k에 초시간이 주어지면 로봇이 움직이기 시작해서 k초 후에 멈춥니다. k초 후 로봇의 위치를 반환하는 프로그램을 작성하세요.

입출력 예:

board	k	answer
[[0, 0, 0, 0, 0], [0, 1, 1, 0, 0], [0, 0, 0, 0, 0], [1, 0, 1, 0, 1], [0, 0, 0, 0, 0]]	10	[2, 2]
[[0, 0, 0, 1, 0, 1], [0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 1], [1, 1, 0, 0, 1, 0], [0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0]]	20	[4, 5]

제한사항:

- board의 크기 ($3 \leq n \leq 100$)
- board에서 0은 빈 공간이고, 1은 장애물이다.
- board에서 로봇의 시작위치는 0행 0열(가장 왼쪽 가장 위)이다.
- 변수 k는 1,000이하의 자연수이다.

입력예제 1 설명 :

로봇이 0행 0열에서 출발해 10초 후에 2행 2열에서 멈춥니다.

	0	1	2	3	4
0	0	0	0	0	0
1	0	1	1	0	0
2	0	0	0	0	0
3	1	0	1	0	1
4	0	0	0	0	0

잃어버린 강아지

현수는 농사지를 땅을 찾아 강아지를 데리고 산으로 들로 땅을 찾아 다니고 있었다. 숲속에서 낮잠을 자던 현수는 강아지가 도망가버려 강아지를 잃게 되었다. 강아지가 어디로 갔는지 모르는 현수는 강아지를 찾아 나섰다. 다행히 강아지에게 위치 추적기가 달려 있어 핸드폰 실시간 위성지도로 현수의 위치와 강아지의 위치, 그리고 근처의 지도를 현수는 알 수 있습니다. 지도의 크기는 항상 10*10이며, 각각의 칸에는 각각 나무, 빈칸, 강아지, 그리고 현수가 있을 수 있습니다. 지도는 다음과 같이 주어진다.

0 - 빈칸, 1 - 나무, 2 - 현수, 3 - 강아지

강아지와 현수는 항상 고정된 방법으로 지도를 다닌다. 먼저 북쪽(지도에서 위쪽)으로 출발하되, 계속 한쪽방향으로 가다가 나무나 지도의 끝에 이르면 90도 시계방향으로 회전하게 된다. 한 칸을 이동하거나, 방향을 회전할 때에는 1분이 소요된다.

만약 이동, 또는 회전을 한 후 현수와 강아지가 같은 칸에 있게 되면 현수가 강아지를 찾게 된다. 현수와 강아지가 있는 숲의 지도정보가 board에 주어지면 몇 분 후에 현수가 강아지를 찾을 수 있는지 구하는 프로그램을 작성하세요. 10,000분 후에도 찾을 수 없으면 0을 반환합니다.

입출력 예:

board	answer
<pre>[[0, 0, 0, 0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0, 2, 0, 0], [1, 0, 0, 0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 3, 0, 0, 0, 1], [0, 0, 0, 1, 0, 1, 0, 0, 0, 0], [0, 1, 0, 1, 0, 0, 0, 0, 0, 0]]</pre>	51
<pre>[[1, 0, 0, 0, 1, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 1, 0, 0, 0], [0, 0, 1, 1, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 1, 0, 1, 0, 0, 0, 0], [1, 0, 0, 0, 0, 0, 1, 0, 1, 0], [0, 0, 0, 1, 0, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0, 0, 2, 1], [0, 0, 0, 1, 0, 1, 0, 0, 0, 1], [0, 1, 0, 1, 0, 0, 0, 0, 0, 3]]</pre>	17

좌석 번호

세계 최고의 알고리즘 전문가인 현수의 강연을 보기위해 많은 사람들이 찾아왔습니다.

강연장에는 가로로 c 개, 세로로 r 개의 좌석이 $c \times r$ 격자형태로 배치되어 있다. 각 좌석의 번호는 해당 격자의 좌표 (x,y) 로 표시된다.

아래 그림은 가로 6개, 세로 5개 좌석으로 구성된 6×5 격자형 좌석배치입니다. 각 격자에 표시된 (x,y) 는 해당 좌석의 번호를 말합니다. 가장 왼쪽 아래의 좌석번호는 $(1,1)$ 이며, 가장 오른쪽 위 좌석의 번호는 $(6, 5)$ 이다.

(1, 5)	(2, 5)	(3, 5)	(4, 5)	(5, 5)	(6, 5)
(1, 4)	(2, 4)	(3, 4)			(6, 4)
(1, 3)	(2, 3)				(6, 3)
(1, 2)	(2, 2)				(6, 2)
(1, 1)	(2, 1)	(3, 1)	(4, 1)	(5, 1)	(6, 1)

사람들은 온 순서대로 $(1,1)$ 위치 좌석부터 시작하여 시계방향으로 돌아 들어가면서 빈 좌석에 앉습니다. 만약 5번째로 온 사람은 $(1, 5)$ 좌석에 앉고, 8번째로 온 사람은 $(4, 5)$ 좌석에 앉으며, 12번째 온 사람은 $(6, 3)$ 좌석에, 20번째 온 사람은 $(2, 3)$ 좌석에 앉게됩니다.

매개변수 c 와 r 에 강연장의 크기가 주어지면, k 번째로 온 사람이 앉을 좌석번호를 반환하는 프로그램을 작성하세요.

만일 모든 좌석이 배정되어 k 번째 온 사람이 앉을 좌석이 없을 경우 $[0, 0]$ 을 반환합니다.

입출력 예:

c	r	k	answer
6	5	12	[6, 3]
6	5	20	[2, 3]
6	5	30	[4, 3]
6	5	31	[0, 0]

제한사항:

- $5 \leq c, r \leq 1,000$ 이다.
- $1 \leq k \leq 100,000,000$ 이다.

과일 가져가기

현수네 반에는 1번부터 n번까지 학생번호를 가진 n명의 학생이 있습니다.

모든 학생의 책상에는 사과, 배, 귤이 담긴 A, B, C 세 바구니가 놓여 있습니다.

A바구니는 사과만 담고, B바구니는 배만 담고, C바구니는 귤만 담습니다.

각 학생은 책상에 있는 세 바구니 중 하나를 가질 수 있습니다. 단 이 세 바구니 중 가장 적게 과일이 담겨있는 바구니를 가집니다.

예를 들어 현수의 책상에 10, 10, 12개가 각각 바구니에 담겨 있다면 현수는 사과(10개) 또는 배(10개) 중 하나를 가져갑니다. 즉 현수는 10개의 과일을 가져가는 것입니다.

모든 학생은 딱 한 번 특정 바구니의 과일 한 개를 다른 학생과 교환할 수 있는 기회가 있습니다.

교환하는 규칙은 다음과 같습니다.

- 1) 1번 학생부터 번호 순으로 교환을 할 건지 결정합니다.
- 2) 교환을 하는 양쪽 학생이 서로 이득이 되면 무조건 교환을 합니다. 즉 양쪽이 모두 가져가는 과일의 개수가 원래 가져가려고 했던 것보다 증가한다면 교환을 무조건 합니다.
- 3) 교환을 할 때는 A바구니는 사과만, B바구니에는 배만, C바구니에는 귤만 담아야 합니다.
- 4) 교환 가능한 학생이 여러명일 경우 가장 번호가 작은 학생과 교환합니다.
- 5) 서로가 이득이 생기는 경우가 존재하지 않으면 교환하지 않는 학생도 있습니다.

매개변수 fruit 에 1번 학생부터 n번 학생까지의 A, B, C 세 바구니에 들어있는 과일의 개수 정보가 주어지면 모든 교환이 끝난 후 모든 학생이 가져가는 과일의 총 개수를 반환하는 프로그램을 작성하세요.

입출력 예:

fruit	answer
[[10, 20, 30], [12, 15, 20], [20, 12, 15], [15, 20, 10], [10, 15, 10]]	58
[[10, 9, 11], [15, 20, 25]]	24
[[0, 3, 27], [20, 5, 5], [19, 5, 6], [10, 10, 10], [15, 10, 5], [3, 7, 20]]	32
([[3, 7, 20], [10, 15, 5], [19, 5, 6], [10, 10, 10], [15, 10, 5], [3, 7, 20], [12, 12, 6], [10, 20, 0], [5, 10, 15]])	48

제한사항:

- fruit 의 길이는 10,000을 넘지 않습니다. 즉 n은 10,000을 넘지 않습니다.
- 각 바구니에 담기는 과일이 개수는 0부터 50개 까지입니다. ($0 \leq \text{과일개수} \leq 50$)

입력예제 1 설명 :

1번 학생은 3번 학생과 사과를 받고 배를 주는 교환을 해서 1번 학생은 [11, 19, 30], 3번 학

생은 [19, 13, 15]로 변한다. 즉 1번 학생은 최소값이 10에서 11로 증가하고, 3번 학생은 최소값이 12에서 13으로 증가한다. 2번 학생은 4번 학생과 사과를 받고 귤을 주는 교환하여 2번 학생은 [13, 15, 19], 4번 학생은 [14, 20, 11]이 된다. 5번 학생은 교환을 할 학생이 없다.

모든 학생이 가져가는 과일의 총 개수는 $11 + 13 + 13 + 11 + 10 = 58$ 입니다.

비밀번호

현수는 자신의 홍채 지문을 읽으면 비밀번호가 자동으로 입력되는 소프트웨어를 만들고 있습니다. 이 소프트웨어는 1부터 9까지의 숫자가 3 * 3 격자모양으로 되어 있는 키패드에서 비밀번호의 순서대로 이동하면서 입력되는 방식입니다.

키패드의 숫자배치는 항상 변합니다.

소프트웨어는 비밀번호의 첫 숫자에서 시작하여 이웃한(상하좌우, 대각선) 8개의 방향으로 이동하면서 입력됩니다. 이웃한 번호로의 이동시간은 1초가 걸립니다. 그리고 이웃하지 않은 숫자로의 이동은 이웃한 숫자를 통해서 이동하는 형태를 취하며, 이웃한 숫자로 이동때마다 1초씩 걸립니다.

즉 키패드에 숫자가 아래와 같이 배치되고,

1	2	3
4	5	6
7	8	9

시작위치가 2라면 1, 3, 4, 5, 6 으로 이동은 1초가 걸리고, 시작위치 2에서 7, 8, 9로는 2초가 걸립니다. 물론 번호가 입력되는 시간은 무시합니다.

매개변수 keypad에 키패드의 숫자배열이 주어지고, password에 입력해야 할 비밀번호가 주어지면 비밀번호가 모두 입력되는데 걸리는 총 시간을 반환하는 프로그램을 작성하세요.

입출력 예:

keypad	password	answer
[2, 5, 3, 7, 1, 6, 4, 9, 8]	"7596218"	8
[1, 5, 7, 3, 2, 8, 9, 4, 6]	"63855526592"	12
[2, 9, 3, 7, 8, 6, 4, 5, 1]	"323254677"	13
[1, 6, 7, 3, 8, 9, 4, 5, 2]	"3337772122"	8

제한사항:

- password의 길이는 200,000을 넘지 않습니다.

입력예제 1 설명 :

키패드의 배열은 아래와 같습니다. 비밀번호 "7596218" 를 입력하는데 걸리는 시간은

2	5	3
7	1	6
4	9	8

7(0초), 5(1초), 9(2초), 6(1초), 2(2초), 1(1초), 8(1초) 가 걸립니다.

총 시간은 0 + 1 + 2 + 1 + 2 + 1 + 1 = 8 초입니다.