

오픈채팅방

카카오톡 오픈채팅방에서는 친구가 아닌 사람들과 대화를 할 수 있는데, 본래 닉네임이 아닌 가상의 닉네임을 사용하여 채팅방에 들어갈 수 있다.

신입사원인 김크루는 카카오톡 오픈 채팅방을 개설한 사람을 위해, 다양한 사람들이 들어오고, 나가는 것을 지켜볼 수 있는 관리자창을 만들기로 했다. 채팅방에 누군가 들어오면 다음 메시지가 출력된다.

[닉네임]님이 들어왔습니다.

채팅방에서 누군가 나가면 다음 메시지가 출력된다.

[닉네임]님이 나갔습니다.

채팅방에서 닉네임을 변경하는 방법은 다음과 같이 두 가지이다.

- 채팅방을 나간 후, 새로운 닉네임으로 다시 들어간다.
- 채팅방에서 닉네임을 변경한다.

닉네임을 변경할 때는 기존에 채팅방에 출력되어 있던 메시지의 닉네임도 전부 변경된다.

예를 들어, 채팅방에 Muzi와 Prodo라는 닉네임을 사용하는 사람이 순서대로 들어오면 채팅방에는 다음과 같이 메시지가 출력된다.

Muzi님이 들어왔습니다. Prodo님이 들어왔습니다.

채팅방에 있던 사람이 나가면 채팅방에는 다음과 같이 메시지가 남는다.

Muzi님이 들어왔습니다. Prodo님이 들어왔습니다. Muzi님이 나갔습니다.

Muzi가 나간후 다시 들어올 때, Prodo 라는 닉네임으로 들어올 경우 기존에 채팅방에 남아있던 Muzi도 Prodo로 다음과 같이 변경된다.

Prodo님이 들어왔습니다. Prodo님이 들어왔습니다. Prodo님이 나갔습니다. Prodo님이 들어왔습니다.

채팅방은 중복 닉네임을 허용하기 때문에, 현재 채팅방에는 Prodo라는 닉네임을 사용하는 사람이 두 명이 있다. 이제, 채팅방에 두 번째로 들어왔던 Prodo가 Ryan으로 닉네임을 변경하면 채팅방 메시지는 다음과 같이 변경된다.

Prodo님이 들어왔습니다. Ryan님이 들어왔습니다. Prodo님이 나갔습니다. Prodo님이 들어왔습니다.

채팅방에 들어오고 나가거나, 닉네임을 변경한 기록이 담긴 문자열 배열 record가 매개변수로 주어질 때, 모든 기록이 처리된 후, 최종적으로 방을 개설한 사람이 보게 되는 메시지를 문자열 배열 형태로 return 하도록 solution 함수를 완성하라.

제한사항

- record는 다음과 같은 문자열이 담긴 배열이며, 길이는 1 이상 100,000 이하이다.
- 다음은 record에 담긴 문자열에 대한 설명이다.
 - 모든 유저는 [유저 아이디]로 구분한다.
 - [유저 아이디] 사용자가 [닉네임]으로 채팅방에 입장 - Enter [유저 아이디][닉네임] (ex. Enter uid1234 Muzi)

- [유저 아이디] 사용자가 채팅방에서 퇴장 - Leave [유저 아이디] (ex. Leave uid1234)
- [유저 아이디] 사용자가 닉네임을 [닉네임]으로 변경 - Change [유저 아이디][닉네임] (ex. Change uid1234 Muzi)
- 첫 단어는 Enter, Leave, Change 중 하나이다.
- 각 단어는 공백으로 구분되어 있으며, 알파벳 대문자, 소문자, 숫자로만 이루어져있다.
- 유저 아이디와 닉네임은 알파벳 대문자, 소문자를 구별한다.
- 유저 아이디와 닉네임의 길이는 1 이상 10 이하이다.
- 채팅방에서 나간 유저가 닉네임을 변경하는 등 잘못 된 입력은 주어지지 않는다.

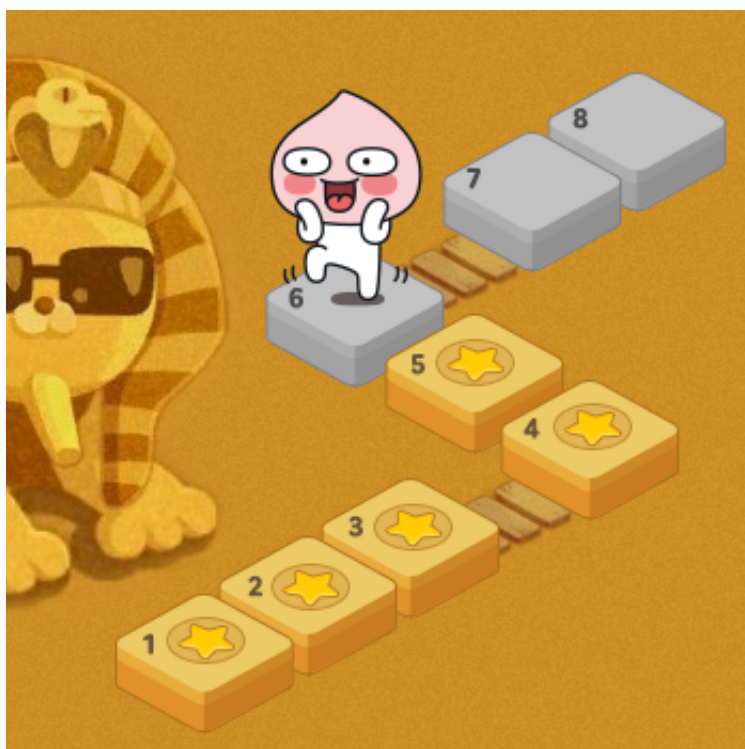
입출력 예

record	result
[Enter uid1234 Muzi, Enter uid4567 Prodo,Leave uid1234,Enter uid1234 Prodo,Change uid4567 Ryan]	[Prodo님이 들어왔습니다., Ryan님이 들어왔습니다., Prodo님이 나갔습니다., Prodo님이 들어왔습니다.]

입출력 예 설명

입출력 예 #1 문제의 설명과 같다.

실패율



슈퍼 게임 개발자 오렐리는 큰 고민에 빠졌다. 그녀가 만든 프랜즈 오천성이 대성공을 거뒀지만, 요즘 신규 사용자의 수가 급감한 것이다. 원인은 신규 사용자와 기존 사용자 사이에 스테이지 차이가 너무 큰 것이 문제였다.

이 문제를 어떻게 할까 고민 한 그녀는 동적으로 게임 시간을 늘려서 난이도를 조절하기로 했다. 역시 슈퍼 개발자라 대부분의 로직은 쉽게 구현했지만, 실패율을 구하는 부분에서 위기에 빠지고 말았다. 오렐리를 위해 실패율을 구하는 코드를 완성하라.

- 실패율은 다음과 같이 정의한다.
 - 스테이지에 도달했으나 아직 클리어하지 못한 플레이어의 수 / 스테이지에 도달한 플레이어 수

전체 스테이지의 개수 N , 게임을 이용하는 사용자가 현재 멈춰있는 스테이지의 번호가 담긴 배열 `stages`가 매개변수로 주어질 때, 실패율이 높은 스테이지부터 내림차순으로 스테이지의 번호가 담겨있는 배열을 return 하도록 `solution` 함수를 완성하라.

제한사항

- 스테이지의 개수 N 은 1 이상 500 이하의 자연수이다.
- `stages`의 길이는 1 이상 200,000 이하이다.
- `stages`에는

1

이상

$N + 1$

이하의 자연수가 담겨있다.

- 각 자연수는 사용자가 현재 도전 중인 스테이지의 번호를 나타낸다.
- 단, $N + 1$ 은 마지막 스테이지(N 번째 스테이지) 까지 클리어 한 사용자를 나타낸다.
- 만약 실패율이 같은 스테이지가 있다면 작은 번호의 스테이지가 먼저 오도록 하면 된다.
- 스테이지에 도달한 유저가 없는 경우 해당 스테이지의 실패율은 0 으로 정의한다.

입출력 예

N	stages	result
5	[2, 1, 2, 6, 2, 4, 3, 3]	[3,4,2,1,5]
4	[4,4,4,4,4]	[4,1,2,3]

입출력 예 설명

입출력 예 #1 1번 스테이지에는 총 8명의 사용자가 도전했으며, 이 중 1명의 사용자가 아직 클리어하지 못했다. 따라서 1번 스테이지의 실패율은 다음과 같다.

- 1 번 스테이지 실패율 : 1/8

2번 스테이지에는 총 7명의 사용자가 도전했으며, 이 중 3명의 사용자가 아직 클리어하지 못했다. 따라서 2번 스테이지의 실패율은 다음과 같다.

- 2 번 스테이지 실패율 : 3/7

마찬가지로 나머지 스테이지의 실패율은 다음과 같다.

- 3 번 스테이지 실패율 : 2/4
- 4번 스테이지 실패율 : 1/2
- 5번 스테이지 실패율 : 0/1

각 스테이지의 번호를 실패율의 내림차순으로 정렬하면 다음과 같다.

- [3,4,2,1,5]

입출력 예 #2

모든 사용자가 마지막 스테이지에 있으므로 4번 스테이지의 실패율은 1이며 나머지 스테이지의 실패율은 0이다.

- [4,1,2,3]

후보키

프렌즈대학교 컴퓨터공학과 조교인 제이지는 네오 학과장님의 지시로, 학생들의 인적사항을 정리하는 업무를 담당하게 되었다.

그의 학부 시절 프로그래밍 경험을 되살려, 모든 인적사항을 데이터베이스에 넣기로 하였고, 이를 위해 정리를 하던 중에 후보키(Candidate Key)에 대한 고민이 필요하게 되었다.

후보키에 대한 내용이 잘 기억나지 않던 제이지는, 정확한 내용을 파악하기 위해 데이터베이스 관련 서적을 확인하여 아래와 같은 내용을 확인하였다.

- 관계 데이터베이스에서 릴레이션(Relation)의 튜플(Tuple)을 유일하게 식별할 수 있는 속성(Attribute) 또는 속성의 집합 중, 다음 두 성질을 만족하는 것을 후보 키(Candidate Key)라고 한다.
 - 유일성(uniqueness) : 릴레이션에 있는 모든 튜플에 대해 유일하게 식별되어야 한다.
 - 최소성(minimality) : 유일성을 가진 키를 구성하는 속성(Attribute) 중 하나라도 제외하는 경우 유일성이 깨지는 것을 의미한다. 즉, 릴레이션의 모든 튜플을 유일하게 식별하는 데 꼭 필요한 속성들만으로 구성되어야 한다.

제이지를 위해, 아래와 같은 학생들의 인적사항이 주어졌을 때, 후보 키의 최대 개수를 구하라.

학번	이름	전공	학년
100	ryan	music	2
200	apeach	math	2
300	tube	computer	3
400	con	computer	1
500	muzi	music	3
600	apeach	music	2

위의 예를 설명하면, 학생의 인적사항 릴레이션에서 모든 학생은 각자 유일한 학번을 가지고 있다. 따라서 학번은 릴레이션의 후보 키가 될 수 있다. 그다음 이름에 대해서는 같은 이름(peach)을 사용하는 학생이 있기 때문에, 이름은 후보 키가 될 수 없다. 그러나, 만약 [이름, 전공]을 함께 사용한다면 릴레이션의 모든 튜플을 유일하게 식별 가능하므로 후보 키가 될 수 있게 된다. 물론 [이름, 전공, 학년]을 함께 사용해도 릴레이션의 모든 튜플을 유일하게 식별할 수 있지만, 최소성을 만족하지 못하기 때문에 후보 키가 될 수 없다. 따라서, 위의 학생 인적사항의 후보키는 학번, [이름, 전공] 두 개가 된다.

릴레이션을 나타내는 문자열 배열 relation이 매개변수로 주어질 때, 이 릴레이션에서 후보 키의 개수를 return 하도록 solution 함수를 완성하라.

제한사항

- relation은 2차원 문자열 배열이다.
- relation의 컬럼(column)의 길이는 1 이상 8 이하이며, 각각의 컬럼은 릴레이션의 속성을 나타낸다.
- relation의 로우(row)의 길이는 1 이상 20 이하이며, 각각의 로우는 릴레이션의 튜플을 나타낸다.
- relation의 모든 문자열의 길이는 1 이상 8 이하이며, 알파벳 소문자와 숫자로만 이루어져 있다.
- relation의 모든 튜플은 유일하게 식별 가능하다.(즉, 중복되는 튜플은 없다.)

입출력 예

relation	result
[[100,ryan,music,2],[200,apeach,math,2],[300,tube,computer,3],[400,con,computer,4],[500,muzi,music,3],[600,apeach,music,2]]	2

입출력 예 설명

입출력 예 #1 문제에 주어진 릴레이션과 같으며, 후보 키는 2개이다.

무지의 먹방 라이브

* 효율성 테스트에 부분 점수가 있는 문제입니다.

평소 식욕이 왕성한 무지는 자신의 재능을 뽐내고 싶어 족고 고민 끝에 카카오 TV 라이브로 방송을 하기로 마음먹었다.



그냥 먹방을 하면 다른 방송과 차별성이 없기 때문에 무지는 아래와 같이 독특한 방식을 생각해냈다.

회전판에 먹어야 할 N 개의 음식이 있다. 각 음식에는 1부터 N 까지 번호가 붙어있으며, 각 음식을 섭취하는데 일정 시간이 소요된다. 무지는 다음과 같은 방법으로 음식을 섭취한다.

- 무지는 1번 음식부터 먹기 시작하며, 회전판에 번호가 증가하는 순서대로 음식을 무지 앞으로 가져다 놓는다.
- 마지막 번호의 음식을 섭취한 후에는 회전판에 의해 다시 1번 음식이 무지 앞으로 온다.
- 무지는 음식 하나를 1초 동안 섭취한 후 남은 음식은 그대로 두고, 다음 음식을 섭취한다.
 - 다음 음식이란, 아직 남은 음식 중 다음으로 섭취해야 할 가장 가까운 번호의 음식을 말한다.
- 회전판이 다음 음식을 무지 앞으로 가져오는데 걸리는 시간은 없다고 가정한다.

무지가 먹방을 시작한 지 K 초 후에 네트워크 장애로 인해 방송이 잠시 중단되었다. 무지는 네트워크 정상화 후 다시 방송을 이어갈 때, 몇 번 음식부터 섭취해야 하는지를 알고자 한다. 각 음식을 모두 먹는데 필요한 시간이 담겨있는 배열 `food_times`, 네트워크 장애가 발생한 시간 K 초가 매개변수로 주어질 때 몇 번 음식부터 다시 섭취하면 되는지 return 하도록 solution 함수를 완성하라.

제한사항

- `food_times` 는 각 음식을 모두 먹는데 필요한 시간이 음식의 번호 순서대로 들어있는 배열이다.
- `k` 는 방송이 중단된 시간을 나타낸다.
- 만약 더 섭취해야 할 음식이 없다면 `-1` 을 반환하면 된다.

정확성 테스트 제한 사항

- `food_times` 의 길이는 1 이상 2,000 이하이다.
- `food_times` 의 원소는 1 이상 1,000 이하의 자연수이다.
- `k` 는 1 이상 2,000,000 이하의 자연수이다.

효율성 테스트 제한 사항

- `food_times` 의 길이는 1 이상 200,000 이하이다.

- food_times 의 원소는 1 이상 100,000,000 이하의 자연수이다.
- k는 1 이상 2×10^{13} 이하의 자연수이다.

입출력 예

food_times	k	result
[3, 1, 2]	5	1

입출력 예 설명

입출력 예 #1

- 0~1초 동안에 1번 음식을 섭취한다. 남은 시간은 [2,1,2] 이다.
- 1~2초 동안 2번 음식을 섭취한다. 남은 시간은 [2,0,2] 이다.
- 2~3초 동안 3번 음식을 섭취한다. 남은 시간은 [2,0,1] 이다.
- 3~4초 동안 1번 음식을 섭취한다. 남은 시간은 [1,0,1] 이다.
- 4~5초 동안 (2번 음식은 다 먹었으므로) 3번 음식을 섭취한다. 남은 시간은 [1,0,0] 이다.
- 5초에서 네트워크 장애가 발생했다. 1번 음식을 섭취해야 할 때 중단되었으므로, 장애 복구 후에 1번 음식부터 다시 먹기 시작하면 된다.

길 찾기 게임

전무로 승진한 라이언은 기분이 너무 좋아 프렌즈를 이끌고 특별 휴가를 가기로 했다. 내친김에 여행 계획까지 구상하던 라이언은 재미있는 게임을 생각해냈고 역시 전무로 승진할만한 인재라고 스스로에게 감탄했다.

라이언이 구상한(그리고 아마도 라이언만 즐거울만한) 게임은, 카카오 프렌즈를 두 팀으로 나누고, 각 팀이 같은 곳을 다른 순서로 방문하도록 해서 먼저 순회를 마친 팀이 승리하는 것이다.

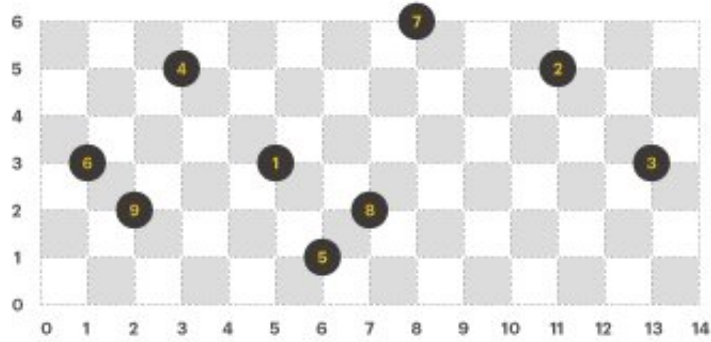
그냥 지도를 주고 게임을 시작하면 재미가 덜해지므로, 라이언은 방문할 곳의 2차원 좌표 값을 구하고 각 장소를 이진트리의 노드가 되도록 구성한 후, 순회 방법을 힌트로 주어 각 팀이 스스로 경로를 찾도록 할 계획이다.

라이언은 아래와 같은 특별한 규칙으로 트리 노드들을 구성한다.

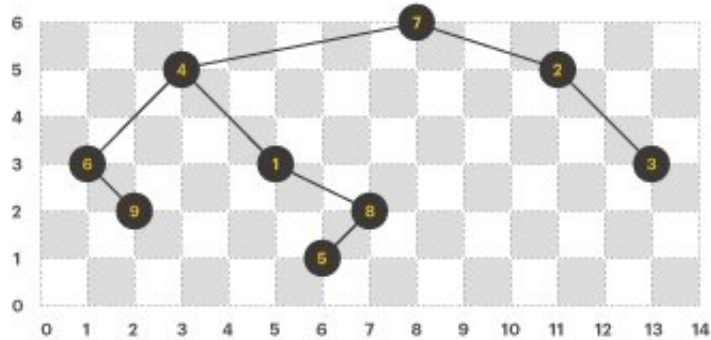
- 트리를 구성하는 모든 노드의 x, y 좌표 값은 정수이다.
- 모든 노드는 서로 다른 x값을 가진다.
- 같은 레벨(level)에 있는 노드는 같은 y 좌표를 가진다.
- 자식 노드의 y 값은 항상 부모 노드보다 작다.
- 임의의 노드 V의 왼쪽 서브 트리(left subtree)에 있는 모든 노드의 x값은 V의 x값보다 작다.
- 임의의 노드 V의 오른쪽 서브 트리(right subtree)에 있는 모든 노드의 x값은 V의 x값보다 크다.

아래 예시를 확인해보자.

라이언의 규칙에 맞게 이진트리의 노드만 좌표 평면에 그리면 다음과 같다. (이진트리의 각 노드에는 1부터 N까지 순서대로 번호가 붙어있다.)



이제, 노드를 잇는 간선(edge)을 모두 그리면 아래와 같은 모양이 된다.



위 이진트리에서 전위 순회(preorder), 후위 순회(postorder)를 한 결과는 다음과 같고, 이것은 각 팀이 방문해야 할 순서를 의미한다.

- 전위 순회 : 7, 4, 6, 9, 1, 8, 5, 2, 3
- 후위 순회 : 9, 6, 5, 8, 1, 4, 3, 2, 7

다행히 두 팀 모두 머리를 모아 분석한 끝에 라이언의 의도를 간신히 알아차렸다.

그러나 여전히 문제는 남아있다. 노드의 수가 예시처럼 적다면 쉽게 해결할 수 있겠지만, 예상대로 라이언은 그렇게 할 생각이 전혀 없었다.

이제 당신이 나설 때가 되었다.

곤경에 빠진 카카오 프렌즈를 위해 이진트리를 구성하는 노드들의 좌표가 담긴 배열 `nodeinfo`가 매개변수로 주어질 때, 노드들로 구성된 이진트리를 전위 순회, 후위 순회한 결과를 2차원 배열에 순서대로 담아 return 하도록 `solution` 함수를 완성하자.

제한사항

- `nodeinfo`는 이진트리를 구성하는 각 노드의 좌표가 1번 노드부터 순서대로 들어있는 2차원 배열이다.
 - `nodeinfo`의 길이는 1 이상 10,000 이하이다.
 - `nodeinfo[i]` 는 $i + 1$ 번 노드의 좌표이며, [x축 좌표, y축 좌표] 순으로 들어있다.
 - 모든 노드의 좌표 값은 0 이상 100,000 이하인 정수이다.
 - 트리의 깊이가 1,000 이하인 경우만 입력으로 주어진다.
 - 모든 노드의 좌표는 문제에 주어진 규칙을 따르며, 잘못된 노드 위치가 주어지는 경우는 없다.

입출력 예

nodeinfo	result
[[5,3],[11,5],[13,3],[3,5],[6,1],[1,3],[8,6],[7,2],[2,2]]	[[7,4,6,9,1,8,5,2,3],[9,6,5,8,1,4,3,2,7]]

입출력 예 설명

입출력 예 #1

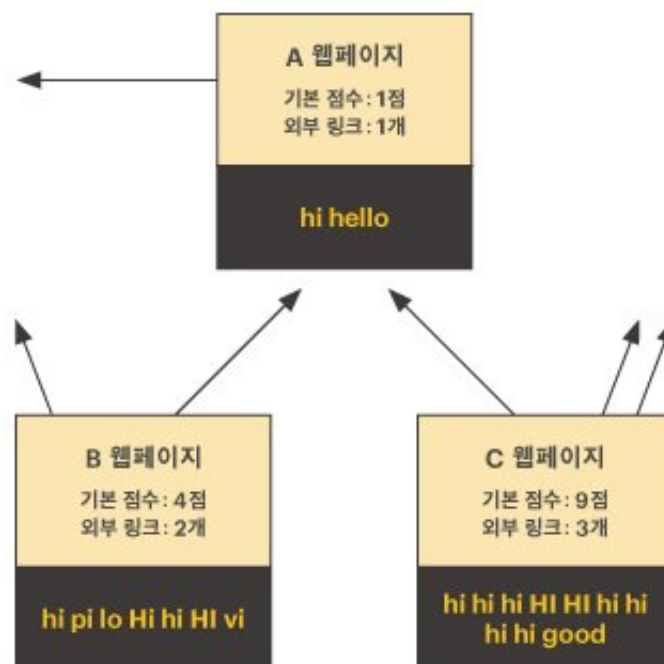
문제에 주어진 예시와 같다.

매칭 점수

프렌즈 대학교 조교였던 제이지는 허드렛일만 시키는 네오 학과장님의 마수에서 벗어나, 카카오에 입사하게 되었다. 평소에 관심있어하던 검색에 마침 결원이 발생하여, 검색개발팀에 편입될 수 있었고, 대망의 첫 프로젝트를 맡게 되었다. 그 프로젝트는 검색어에 가장 잘 맞는 웹페이지를 보여주기 위해 아래와 같은 규칙으로 검색어에 대한 웹페이지의 매칭점수를 계산 하는 것이었다.

- 한 웹페이지에 대해서 기본점수, 외부 링크 수, 링크점수, 그리고 매칭점수를 구할 수 있다.
- 한 웹페이지의 기본점수는 해당 웹페이지의 텍스트 중, 검색어가 등장하는 횟수이다. (대소문자 무시)
- 한 웹페이지의 외부 링크 수는 해당 웹페이지에서 다른 외부 페이지로 연결된 링크의 개수이다.
- 한 웹페이지의 링크점수는 해당 웹페이지로 링크가 걸린 다른 웹페이지의 기본점수 ÷ 외부 링크 수의 총합이다.
- 한 웹페이지의 매칭점수는 기본점수와 링크점수의 합으로 계산한다.

예를 들어, 다음과 같이 A, B, C 세 개의 웹페이지가 있고, 검색어가 hi라고 하자.



이때 A 웹페이지의 매칭점수는 다음과 같이 계산할 수 있다.

- 기본 점수는 각 웹페이지에서 h_i 가 등장한 횟수이다.
 - A,B,C 웹페이지의 기본점수는 각각 1점, 4점, 9점이다.
- 외부 링크수는 다른 웹페이지로 링크가 걸린 개수이다.
 - A,B,C 웹페이지의 외부 링크 수는 각각 1점, 2점, 3점이다.
- A 웹페이지로 링크가 걸린 페이지는 B와 C가 있다.
 - A 웹페이지의 링크점수는 B의 링크점수 2점($4 \div 2$)과 C의 링크점수 3점($9 \div 3$)을 더한 5점이 된다.
- 그러므로, A 웹페이지의 매칭점수는 기본점수 1점 + 링크점수 5점 = 6점이 된다.

검색어 word와 웹페이지의 HTML 목록인 pages가 주어졌을 때, 매칭점수가 가장 높은 웹페이지의 index를 구하라. 만약 그런 웹페이지가 여러 개라면 그중 번호가 가장 작은 것을 구하라.

제한사항

- pages는 HTML 형식의 웹페이지가 문자열 형태로 들어있는 배열이고, 길이는 1 이상 20 이하이다.
- 한 웹페이지 문자열의 길이는 1 이상 1,500 이하이다.
- 웹페이지의 index는 pages 배열의 index와 같으며 0부터 시작한다.
- 한 웹페이지의 url은 HTML의 태그 내에 태그의 값으로 주어진다.
 - 예를들어, 아래와 같은 meta tag가 있으면 이 웹페이지의 url은 <https://careers.kakao.com/index>이다.
-
- 한 웹페이지에서 모든 외부 링크는 `<a href=
https://careers.kakao.com/index
>`의 형태를 가진다.
 - 내에 다른 attribute가 주어지는 경우는 없으며 항상 href로 연결할 사이트의 url만 포함된다.
 - 위의 경우에서 해당 웹페이지는 <https://careers.kakao.com/index>로 외부링크를 가지고 있다고 볼 수 있다.
- 모든 url은 https:// 로만 시작한다.
- 검색어 word는 하나의 영어 단어로만 주어지며 알파벳 소문자와 대문자로만 이루어져 있다.
- word의 길이는 1 이상 12 이하이다.
- 검색어를 찾을 때, 대소문자 구분은 무시하고 찾는다.
 - 예를들어 검색어가 blind일 때, HTML 내에 Blind라는 단어가 있거나, BLIND라는 단어가 있으면 두 경우 모두 해당된다.
- 검색어는 단어 단위로 비교하며, 단어와 완전히 일치하는 경우에만 기본 점수에 반영한다.
 - 단어는 알파벳을 제외한 다른 모든 문자로 구분한다.
 - 예를들어 검색어가 aba 일 때, abab abababa는 단어 단위로 일치하는게 없으니, 기본 점수는 0점이 된다.
 - 만약 검색어가 aba 라면, aba@aba aba는 단어 단위로 세개가 일치하므로, 기본 점수는 3점이다.
- 결과를 돌려줄때, 동일한 매칭점수를 가진 웹페이지가 여러 개라면 그중 index 번호가 가장 작은 것을 리턴한다
 - 즉, 웹페이지가 세개이고, 각각 매칭점수가 3,1,3 이라면 제일 적은 index 번호인 0을 리턴하면 된다.

입출력 예 #1

- word : blind
- pages :

```
[ "<html lang=\"ko\" xml:lang=\"ko\"
xmlns=\"http://www.w3.org/1999/xhtml\">\n<head>\n  <meta charset=\"utf-8\">\n  <meta property=\"og:url\" content=\"https://a.com\"/>\n</head>
\n<body>\nBlind Lorem Blind ipsum dolor Blind test sit amet,
consectetur adipiscing elit. \n<a href=\"https://b.com\"> Link to b
</a>\n</body>\n</html>", "<html lang=\"ko\" xml:lang=\"ko\"
xmlns=\"http://www.w3.org/1999/xhtml\">\n<head>\n  <meta charset=\"utf-8\">\n  <meta property=\"og:url\" content=\"https://b.com\"/>\n</head>
\n<body>\nSuspendisse potenti. Vivamus venenatis tellus non turpis
bibendum, \n<a href=\"https://a.com\"> Link to a </a>\nblind sed congue
urna varius. Suspendisse feugiat nisl ligula, quis malesuada felis
hendrerit ut.\n<a href=\"https://c.com\"> Link to c
</a>\n</body>\n</html>", "<html lang=\"ko\" xml:lang=\"ko\"
xmlns=\"http://www.w3.org/1999/xhtml\">\n<head>\n  <meta charset=\"utf-8\">\n  <meta property=\"og:url\" content=\"https://c.com\"/>\n</head>
\n<body>\nUt condimentum urna at felis sodales rutrum. Sed dapibus
cursus diam, non interdum nulla tempor nec. Phasellus rutrum enim at
orci consectetur blind\n<a href=\"https://a.com\"> Link to a
</a>\n</body>\n</html>" ]
```

- pages는 다음과 같이 3개의 웹페이지에 해당하는 HTML 문자열이 순서대로 들어있다.

```
<html lang="ko" xml:lang="ko" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8">
  <meta property="og:url" content="https://a.com"/>
</head>
<body>
Blind Lorem Blind ipsum dolor Blind test sit amet, consectetur adipiscing
elit.
<a href="https://b.com"> Link to b </a>
</body>
</html>

<html lang="ko" xml:lang="ko" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8">
  <meta property="og:url" content="https://b.com"/>
</head>
<body>
Suspendisse potenti. Vivamus venenatis tellus non turpis bibendum,
<a href="https://a.com"> Link to a </a>
blind sed congue urna varius. Suspendisse feugiat nisl ligula, quis
malesuada felis hendrerit ut.
<a href="https://c.com"> Link to c </a>
</body>
```

```

</html>
<html lang="ko" xml:lang="ko" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8">
  <meta property="og:url" content="https://c.com"/>
</head>
<body>
  Ut condimentum urna at felis sodales rutrum. Sed dapibus cursus diam, non
  interdum nulla tempor nec. Phasellus rutrum enim at orci consetetu blind
  <a href="https://a.com"> Link to a </a>
</body>
</html>

```

위의 예를 가지고 각각의 점수를 계산해보자.

- 기본점수 및 외부 링크수는 아래와 같다.
 - a.com의 기본점수는 3, 외부 링크 수는 1개
 - b.com의 기본점수는 1, 외부 링크 수는 2개
 - c.com의 기본점수는 1, 외부 링크 수는 1개
- 링크점수는 아래와 같다.
 - a.com의 링크점수는 b.com으로부터 0.5점, c.com으로부터 1점
 - b.com의 링크점수는 a.com으로부터 3점
 - c.com의 링크점수는 b.com으로부터 0.5점
- 각 웹 페이지의 매칭 점수는 다음과 같다.
 - a.com : 4.5 점
 - b.com : 4 점
 - c.com : 1.5 점

따라서 매칭점수가 제일 높은 첫번째 웹 페이지의 index인 0을 리턴 하면 된다.

입출력 예 #2

- word : Muzi
- pages :

```

[ "<html lang=\"ko\" xml:lang=\"ko\"
xmlns=\"http://www.w3.org/1999/xhtml\">\n<head>\n  <meta charset=\"utf-
8\">\n  <meta property=\"og:url\"
content=\"https://careers.kakao.com/interview/list\"/>\n</head>
\n<body>\n<a href=\"https://programmers.co.kr/learn/courses/4673\">
</a>#!MuziMuzi!) jayg07con&&\n\n</body>\n</html>", "<html lang=\"ko\"
xml:lang=\"ko\" xmlns=\"http://www.w3.org/1999/xhtml\">\n<head>\n
<meta charset=\"utf-8\">\n  <meta property=\"og:url\"
content=\"https://www.kakaocorp.com\"/>\n</head>
\n<body>\ncon&tmuzI92apeach&2<a href=\"https://hashcode.co.kr/tos\">
</a>\n\n\t^n</body>\n</html>" ]

```

- pages는 다음과 같이 2개의 웹페이지에 해당하는 HTML 문자열이 순서대로 들어있다.

```

<html lang="ko" xml:lang="ko" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8">
  <meta property="og:url"
content="https://careers.kakao.com/interview/list"/>
</head>
<body>
<a href="https://programmers.co.kr/learn/courses/4673">
</a>#!MuziMuzi!)jayg07con&&

</body>
</html>
<html lang="ko" xml:lang="ko" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8">
  <meta property="og:url" content="https://www.kakaocorp.com"/>
</head>
<body>
con%      muzI92apeach&2<a href="https://hashcode.co.kr/tos"></a>

  ^
</body>
</html>

```

- 기본점수 및 외부 링크수는 아래와 같다.
 - `careers.kakao.com/interview/list` 의 기본점수는 0, 외부 링크 수는 1개
 - `www.kakaocorp.com` 의 기본점수는 1, 외부 링크 수는 1개
- 링크점수는 아래와 같다.
 - `careers.kakao.com/interview/list` 의 링크점수는 0점
 - `www.kakaocorp.com` 의 링크점수는 0점
- 각 웹 페이지의 매칭 점수는 다음과 같다.
 - `careers.kakao.com/interview/list` : 0점
 - `www.kakaocorp.com` : 1 점

따라서 매칭점수가 제일 높은 두번째 웹 페이지의 index인 1을 리턴 하면 된다.

블록게임

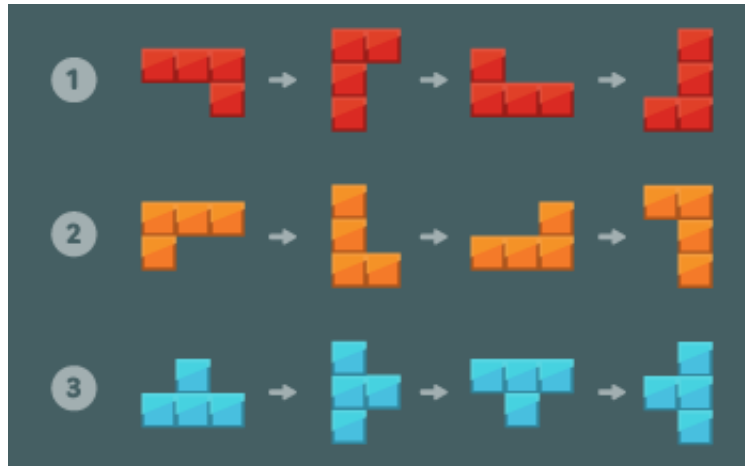
프렌즈 블록이라는 신규 게임이 출시되었고, 어마어마한 상금이 걸린 이벤트 대회가 개최 되었다.

이 대회는 사람을 대신해서 플레이할 프로그램으로 참가해도 된다는 규정이 있어서, 게임 실력이 형편없는 프로도는 프로그램을 만들어서 참가하기로 결심하고 개발을 시작하였다.

프로도가 우승할 수 있도록 도와서 빠르고 정확한 프로그램을 작성해 보자.

게임규칙

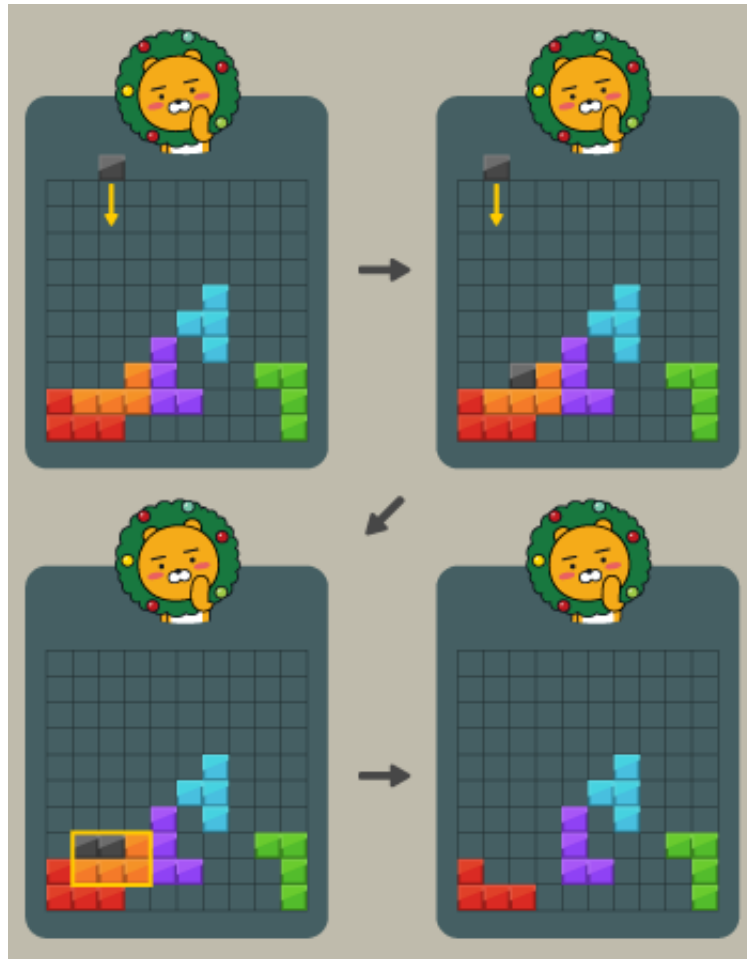
아래 그림과 같이 1×1 크기의 블록을 이어 붙여 만든 3 종류의 블록을 회전해서 총 12가지 모양의 블록을 만들 수 있다.



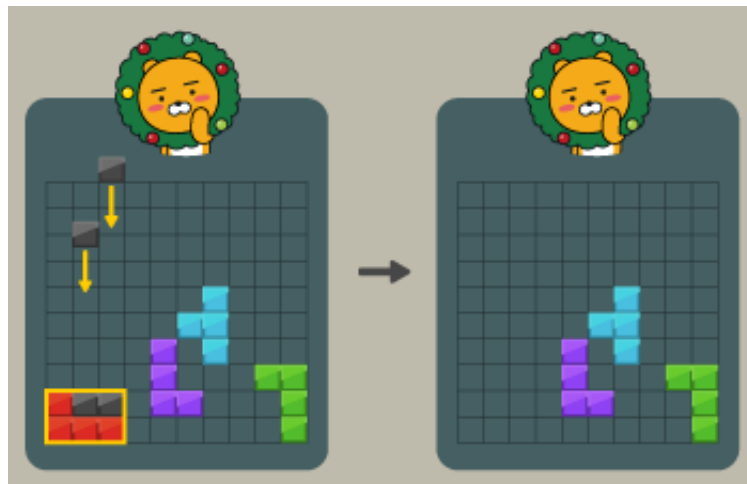
1 x 1 크기의 정사각형으로 이루어진 N x N 크기의 보드 위에 이 블록들이 배치된 채로 게임이 시작된다. (보드 위에 놓인 블록은 회전할 수 없다). 모든 블록은 블록을 구성하는 사각형들이 정확히 보드 위의 사각형에 맞도록 놓여있으며, 선 위에 걸치거나 보드를 벗어나게 놓여있는 경우는 없다.

플레이어는 위쪽에서 1 x 1 크기의 검은 블록을 떨어뜨려 쌓을 수 있다. 검은 블록은 항상 맵의 한 칸에 딱 차게 떨어뜨려야 하며, 줄에 걸치면 안된다. 이때, 검은 블록과 기존에 놓인 블록을 합해 **속이 꽉 채워진** 직사각형을 만들 수 있다면 그 블록을 없앨 수 있다.

예를 들어 검은 블록을 떨어뜨려 아래와 같이 만들 경우 주황색 블록을 없앨 수 있다.



빨간 블록을 가로막던 주황색 블록이 없어졌으므로 다음과 같이 빨간 블록도 없앨 수 있다.



그러나 다른 블록들은 검은 블록을 떨어뜨려 직사각형으로 만들 수 없기 때문에 없앨 수 없다.

따라서 위 예시에서 없앨 수 있는 블록은 최대 2개이다.

보드 위에 놓인 블록의 상태가 담긴 2차원 배열 board가 주어질 때, 검은 블록을 떨어뜨려 없앨 수 있는 블록 개수의 최댓값을 구하라.

제한사항

- board는 블록의 상태가 들어있는 $N \times N$ 크기 2차원 배열이다.
 - N 은 4 이상 50 이하이다.
- board의 각 행의 원소는

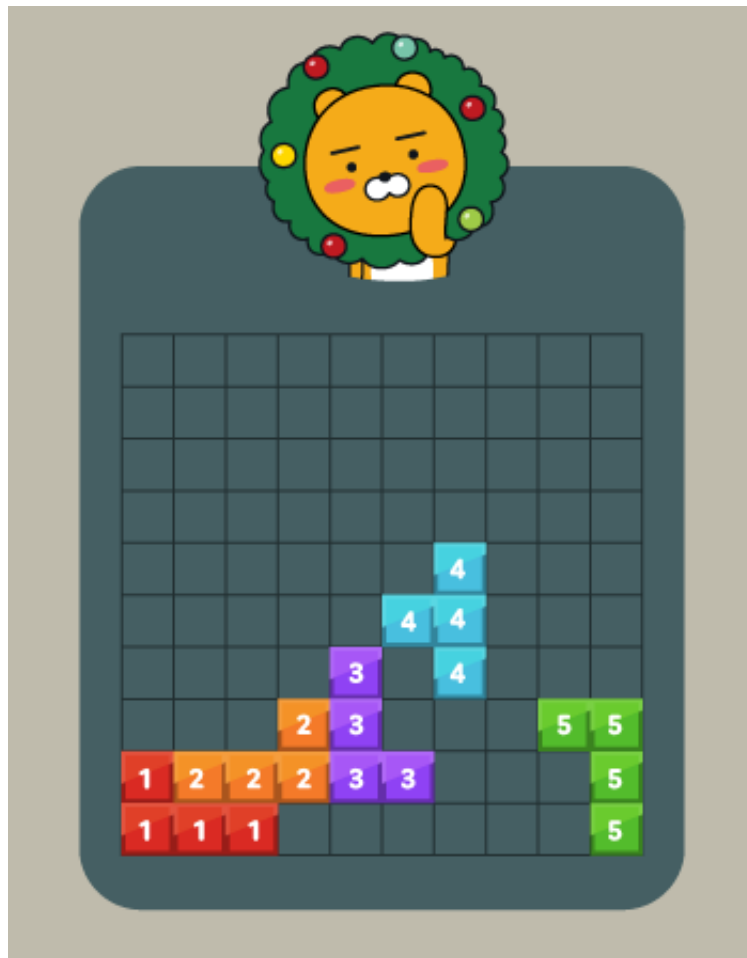
0

이상

200

이하의 자연수이다.

- 0 은 빈 칸을 나타낸다.
- board에 놓여있는 각 블록은 숫자를 이용해 표현한다.
- 잘못된 블록 모양이 주어지는 경우는 없다.
- 예를 들어 문제에 주어진 예시의 경우 다음과 같이 주어진다.



입출력 예

board	result
<pre>[[0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0], [0,0,0,0,0,0,4,0,0,0],[0,0,0,0,0,4,4,0,0,0],[0,0,0,0,3,0,4,0,0,0],[0,0,0,2,3,0,0,0,5,5], [1,2,2,2,3,3,0,0,0,5],[1,1,1,0,0,0,0,0,0,5]]</pre>	2

입출력 예 설명

입출력 예 #1 문제에 주어진 예시와 같음