**문제 설명**

신작 모바일 게임 "인형 맞추기"는 화면의 랜덤한 위치에 나타났다 사라지는 인형들을 손으로 터치해서 점수를 획득하는 간단한 게임입니다. 게임 규칙은 다음과 같습니다.

**기본 규칙**

* 인형은 1초, 3초, 10초 등 **정확한 초 단위 시각**에 나타나며 **한 번에 하나만** 나타났다가 0.3초가 지나면 자동으로 사라집니다.
* 플레이어는 인형이 사라지기 전에 정확히 터치해야 점수를 획득할 수 있습니다.
* 터치한 인형은 화면에서 즉시 사라집니다.

**기본 점수 계산**

* 인형 하나를 터치할 때마다 기본점수 1점을 획득합니다.

**피버 카드(Fever Card) 아이템**

* 몇몇 인형은 터치 시 피버 카드 아이템을 획득할 수 있습니다.
* 아이템은 획득한 후 **10초 안에 사용**해야하며, 사용하지 않을 경우 소멸됩니다.
* 아이템은 하나만 가질 수 있으며 추가로 아이템을 획득한 경우 사용 제한 시간이 **10초로 초기화** 됩니다.
* 아이템을 사용하면 즉시 지금까지 **맞힌 인형의 수 x 10점**의 **아이템 보너스 점수**를 획득하게 됩니다.
* 아이템은 x.5초와 같이 소수점 이하의 시각이 0.5인 순간에만 사용할 수 있습니다.(1.5초, 2.5초, 3.5초,....등)

**피버 타임(Fever Time) 보너스**

* 피버 카드 아이템을 사용해서 피버 타임을 발동할 수 있습니다.
* 피버 타임동안 터치한 인형은 **기본점수 1점 외에 추가로 50점**의 **보너스 점수**를 획득합니다.
* 피버 타임은 피버 카드 아이템을 사용한 순간 부터 **30초 동안 유지**됩니다.
* 피버 타임 중 피버 카드 아이템을 추가로 사용하면 피버 타임 유지 시간이 30초로 초기화 됩니다.

마지막 인형이 나타난 후, 1초 뒤에 게임은 자동으로 종료되어 더이상 인형을 터치하거나 아이템을 사용할 수 없습니다.

위와 같은 규칙에서 각 인형이 화면에 나타나는 시각과 각 인형을 터치했을 때, 피버 카드 아이템 획득 여부를 담은 2차원 배열 dolls가 매개변수로 주어질 때, 플레이어가 획득할 수 있는 **최고 점수**를 return 하도록 solution 함수를 완성해주세요.

제한사항

* dolls의 세로(행) 길이는 1 이상 1,000 이하입니다.
* dolls의 가로(열) 길이는 항상 2입니다.
* dolls의 원소는 [A, B] 형태입니다.
  + A는 인형이 나오는 시각, B는 각 인형을 터치했을 때, 피버 카드 아이템 획득 여부를 나타냅니다.
  + 인형이 나오는 시각은 1 이상 10,000 이하입니다.
  + 인형의 아이템 획득 여부는 0, 1로 나타내며 1인 경우에 아이템을 획득합니다.
* dolls는 인형이 나오는 시간 순서대로 정렬되어 있으며, 시각이 중복되는 경우는 없습니다.

입출력 예

| **dolls** | **result** |
| --- | --- |
| [[5,0],[7,1],[12,0],[16,0],[20,1],[29,1],[35,0],[38,0],[60,0],[72,0]] | 510 |

입출력 예 설명

입출력 예 #1

각 인형이 나오는 시각과 인형의 아이템 획득 여부는 다음과 같습니다.

| **인형 번호** | **나오는 시각(초)** | **아이템 획득 여부** |
| --- | --- | --- |
| 1 | 5 | X |
| 2 | 7 | O |
| 3 | 12 | X |
| 4 | 16 | X |
| 5 | 20 | O |
| 6 | 29 | O |
| 7 | 35 | X |
| 8 | 38 | X |
| 9 | 60 | X |
| 10 | 72 | X |

먼저 모든 인형을 맞히면 1 x 10 = 10점의 기본점수를 획득할 수 있습니다. 아래는 보너스 점수를 계산하는 예시입니다.

* 7초에 "피버 카드" 아이템을 획득 후 7.5초에 아이템을 사용하면 맞힌 인형 수 2 x 10 = 20점(아이템 보너스)을 획득합니다.
  + 37.5초까지 "피버 타임" 이 유지됩니다.
* 12초, 16초에 나오는 인형을 맞히면 2 x 50 = 100점(피버 타임 보너스)을 획득합니다.
* 20초에 나오는 인형을 맞히면 50점(피버 타임 보너스)을 추가 획득하며 아이템을 얻습니다.
  + 20.5 ~ 28.5초에 아이템을 사용하면 맞힌 인형 수 5 x 10 = 50점(피버 타임 보너스)을 획득합니다.
* 29초에 나오는 인형을 맞히면 50점(피버 타임 보너스)을 추가 획득하며 아이템을 얻습니다.
* 35, 38초에 나오는 인형을 맞히면 2 x 50 = 100점(피버 타임 보너스)을 획득합니다.
* 38.5초에 아이템을 사용하면 맞힌 인형 수 8 x 10 = 80점(아이템 보너스)을 획득합니다.
  + 68.5초까지 "피버 타임"이 유지됩니다.
* 60초에 나오는 인형을 맞히면 50점(피버 타임 보너스)을 획득합니다.
* 72초에 나오는 인형은 "피버 타임" 지속 시간이 끝났으므로 보너스 점수를 획득할 수 없습니다.

따라서 플레이어가 획득할 수 있는 **최고 점수**는 **510**점입니다.

###### 문제 설명

당신이 작성한 게시글을 다양한 유저들이 공유(share)하거나 댓글(comment)을 달았습니다. 유저들이 게시글을 공유하거나 댓글을 작성할 때마다 알림(notification)창에 해당 내용이 표시됩니다.

예를 들어 john이 당신의 게시글을 공유하고, mary가 댓글을 단 후, 다시 jay가 게시글을 공유한 경우 알림창 상태는 다음과 같습니다.

* 알림창

| **알림 내용** |
| --- |
| "john shared your post" |
| "mary commented on your post" |
| "jay shared your post" |

알림은 한 줄에 하나씩 표시되며, 나중에 생성된(즉, 더 최근에 발생한) 알림일 수록 아래쪽에 표시됩니다.

이때, 당신은 알림창에서 가장 최근(가장 밑에 있는) 알림부터 한 번에 하나씩 보관함으로 이동할 수 있습니다. 예를 들어 위 예시에서 알림을 보관함으로 **두 번** 이동한 후 알림창과 보관함의 상태는 다음과 같습니다.

* 알림창

| **알림 내용** |
| --- |
| "john shared your post" |

* 보관함

| **알림 내용** |
| --- |
| "jay shared your post" |
| "mary commented on your post" |

알림창에서 가장 밑에 있는(즉, 더 최근에 받은) 알림 부터 하나씩 보관함으로 이동하며, 따라서, 먼저 이동된 알림이 보관함에서 더 위쪽에 위치합니다.

위 상태에서 이번에는 sally가 댓글을 작성하고, james가 게시글을 공유한 후, 알림을 보관함으로 **한 번** 이동한 후 알림창과 보관함 상태는 다음과 같습니다.

* 알림창

| **알림 내용** |
| --- |
| "john shared your post" |
| "sally commented on your post" |

* 보관함

| **알림 내용** |
| --- |
| "jay shared your post" |
| "mary commented on your post" |
| "james shared your post" |

보관함으로 한번 이동된 알림 내용은 바뀌지 않지만, 알림창의 알림 내용은 다음과 같은 규칙으로 바뀌게 됩니다.

* 알림창의 가장 밑에서 부터 연속으로 두 알림이 같은 종류인 경우  
  + 알림을 보낸 두 사람의 닉네임을 A, B라 할 때 A and B commented on your post, A and B shared your post와 같이 표시합니다.
  + A가 더 위에 있는 알림을 보낸 닉네임 이어야 합니다.
* 알림창의 가장 밑에서 부터 연속으로 세 개 이상의 알림이 같은 종류인 경우  
  + 연속된 알림 중, 가장 위에 있는 알림의 닉네임 A이고, A의 알림 밑으로 N명의 알림 종류가 같으면 A and N others shared your post, A and N others commented on your post와 같이 표시합니다.

예를 들어 lee와, laura, will이 순서대로 게시글을 공유한 후 알림창 상태는 다음과 같습니다. [1](https://programmers.co.kr/tryouts/15219/challenges#fn1)

* 알림창

| **알림 내용** |
| --- |
| "john shared your post" |
| "sally commented on your post" |
| "lee and 2 others shared your post" |

* 보관함

| **알림 내용** |
| --- |
| "jay shared your post" |
| "mary commented on your post" |
| "james shared your post" |

이제 알림을 보관함으로 **한 번** 이동한 후 alice가 댓글을 작성하면 알림창 상태는 다음과 같습니다.

* 알림창

| **알림 내용** |
| --- |
| "john shared your post" |
| "sally and alice commented on your post" |

* 보관함

| **알림 내용** |
| --- |
| "jay shared your post" |
| "mary commented on your post" |
| "james shared your post" |
| "lee and 2 others shared your post" |

유저들이 당신의 게시글을 공유하거나, 또는 댓글을 달은 기록과 당신이 알림을 보관함으로 이동한 기록이 담긴 문자열 배열 records가 매개변수로 주어질 때, 모든 기록을 처리한 후 보관함의 상태를 배열에 담아 reutrn 하도록 solution 함수를 완성해주세요.

##### 제한사항

* records의 길이는 2 이상 100 이하입니다.
* records의 원소는 다음과 같은 형식의 문자열입니다.
  + [닉네임] share → [닉네임]이 공유 버튼을 누름
  + [닉네임] comment → [닉네임]이 댓글을 담
  + check notification → 알림창에서 가장 아래쪽에 있는 알림 하나를 보관함으로 이동
  + [닉네임]과 share, comment는 공백(스페이스) 하나로 구분되어 있습니다.
  + [닉네임]은 길이가 1 이상 10 이하인 문자열로, 알파벳 소문자로만 이루어져 있습니다.
  + 모든 [닉네임]은 정확히 한 번씩만 등장합니다.
  + check notification는 적어도 한 번 이상 등장합니다.
  + 알림창이 빈 상태에서 check notification이 주어지는 경우는 없습니다.
* 정답으로 return 하는 문자열의 단어와 단어 사이에 공백(스페이스)을 하나씩만 넣어주세요.
* 보관함에서 가장 위에 있는(즉, 가장 먼저 보관함으로 이동된) 알림 내용부터 순서대로 배열에 담아 return 하면 됩니다.

##### 입출력 예

| **records** | **result** |
| --- | --- |
| ["john share", "mary comment", "jay share", "check notification", "check notification", "sally comment", "james share", "check notification", "lee share", "laura share", "will share", "check notification", "alice comment", "check notification"] | ["jay shared your post", "mary commented on your post", "james shared your post", "lee and 2 others shared your post", "sally and alice commented on your post"] |
| ["john share", "mary share", "jay share", "james comment", "lee share", "check notification", "sally comment", "laura comment", "check notification", "will share", "ruby share", "check notification"] | ["lee shared your post", "james and 2 others commented on your post", "john and 4 others shared your post"] |

##### 입출력 예 설명

입출력 예 #1

문제의 예시에서 마지막에 알림을 보관함으로 **한 번** 이동한 기록이 추가됐습니다. 따라서 알림창과 보관함 상태는 다음과 같습니다.

* 알림창

| **알림 내용** |
| --- |
| "john shared your post" |

* 보관함

| **알림 내용** |
| --- |
| "jay shared your post" |
| "mary commented on your post" |
| "james shared your post" |
| "lee and 2 others shared your post" |
| "sally and alice commented on your post" |

입출력 예 #2

먼저 john, mary, jay가 게시글을 공유하고, james가 댓글을 단 다음, lee가 다시 게시글을 공유한 후 알림창과 보관함 상태는 다음과 같습니다.

* 알림창

| **알림 내용** |
| --- |
| "john and 2 others shared your post" |
| "james commented on your post" |
| "lee shared your post" |

* 보관함

| **알림 내용** |
| --- |
|  |

알림을 보관함으로 **한 번** 이동하고, sally와 laura가 댓글을 단 후 알림창과 보관함 상태는 다음과 같습니다.

* 알림창

| **알림 내용** |
| --- |
| "john and 2 others shared your post" |
| "james and 2 others commented on your post" |

* 보관함

| **알림 내용** |
| --- |
| "lee shared your post" |

알림을 보관함으로 **한 번** 이동하고, will과 ruby가 게시글을 공유한 후 알림창과 보관함 상태는 다음과 같습니다.

* 알림창

| **알림 내용** |
| --- |
| "john and 4 others shared your post" |

* 보관함

| **알림 내용** |
| --- |
| "lee shared your post" |
| "james and 2 others commented on your post" |

마지막으로 알림을 보관함으로 **한 번** 이동한 후 알림창과 보관함 상태는 다음과 같습니다.

* 알림창

| **알림 내용** |
| --- |
|  |

* 보관함

| **알림 내용** |
| --- |
| "lee shared your post" |
| "james and 2 others commented on your post" |
| "john and 4 others shared your post" |

###### 문제 설명

여러 개의 주차장이 하나의 트리[1](https://programmers.co.kr/tryouts/15219/challenges" \l "fn1) 형태로 연결되어 있습니다. 아래 그림은 13개의 주차장이 연결된 예시입니다.

지도이(가) 표시된 사진

자동 생성된 설명

1. 하나의 노드는 하나의 주차장을 나타냅니다.
2. 노드 위의 검은 숫자는, 해당 주차장의 고유 번호(ID)를 나타냅니다.
3. 노드 안의 빨간 숫자는, 해당 주차장의 수용 가능한 자동차 수를 나타냅니다.

3-1. 위의 그림에서, 11번 주차장에는 10대, 5번 주차장에는 8대, 7번 주차장에는 20대를 주차할 수 있습니다.

1. 노드 사이의 간선은 해당 주차장이 연결되어 있음을 의미합니다.

4-1. 위의 그림에서, 11번 주차장은 8번 주차장과, 8번 주차장은 7번 주차장과 연결되어 있습니다.

주차장 사이를 연결한 간선 하나를 끊어서, 전체 주차장을 두 개의 그룹으로 나누려고 합니다. 단, 두 개의 그룹이 수용 가능한 자동차 수의 차이가 최소가 되도록 나누어야 합니다. 위의 그림에서, 8번과 7번 주차장을 잇는 간선을 끊는다면, 전체 주차장은 A그룹(7번, 4번, 1번 주차장)과 B그룹(11번, 8번, 5번, 2번, 9번, 10번, 6번, 13번, 3번, 12번 주차장)으로 나누어집니다. A그룹이 수용 가능한 자동차 수는 57대(20+15+22), B그룹이 수용 가능한 자동차 수는 62대(10+7+8+9+11+5+6+1+1+4)입니다. 따라서, 두 그룹이 수용 가능한 자동차 수의 차이(절댓값)는 5입니다. 다른 어떤 간선을 끊더라도, 두 그룹의 차이가 5보다 작게 분할할 수는 없습니다.

전체 주차장 개수를 나타내는 n, 각 주차장의 수용 가능한 자동차 수를 나타내는 cars, 주차장의 연결 상태를 나타내는 links가 매개변수로 주어집니다. 수용 가능한 자동차 수의 차이가 최소가 되도록 간선 하나를 끊어서 두 개의 그룹으로 나누었을 때, 그 차이의 절댓값을 return 하도록 solution 함수를 완성해주세요.

##### 제한사항

* n은 2 이상 100,000 이하인 자연수입니다.
* cars는 길이가 n인 정수형 배열입니다.
  + cars[i] 에는 i+1번 주차장이 수용 가능한 자동차의 수가 담겨있습니다.
  + 각 주차장이 수용 가능한 자동차의 수는 1 이상 10,000 이하인 자연수입니다.
* links는 길이가 n-1인 정수형 2차원 배열입니다.
  + links는 [a, b] 형태입니다.
    - 고유 번호가 a, b인 두 개의 주차장이 간선으로 연결되어 있음을 의미합니다.
    - a, b는 1 이상 n 이하인 자연수입니다.
    - a ≠ b 입니다.
    - 같은 의미를 가지는 중복된 원소는 주어지지 않습니다. 예를 들어, links에 [8, 11]라는 원소가 존재한다면, 또 다른 인덱스에 [8, 11] 혹은 [11, 8]인 원소는 존재하지 않습니다.

##### 입출력 예

| **n** | **cars** | **links** | **result** |
| --- | --- | --- | --- |
| 13 | [22, 9, 1, 15, 8, 6, 20, 7, 11, 5, 10, 4, 1] | [[4, 7], [13, 10], [6, 3], [7, 1], [6, 12], [5, 11], [5, 6], [5, 10], [9, 8], [8, 11], [8, 2], [7, 8]] | 5 |
| 6 | [6, 4, 10, 9, 8, 4] | [[4, 1], [3, 2], [1, 6], [3, 5], [5, 1]] | 3 |

##### 입출력 예 설명

입출력 예 #1  
문제 예시와 같습니다.

입출력 예 #2  
텍스트, 지도이(가) 표시된 사진

자동 생성된 설명

* 5번 주차장과 1번 주차장을 잇는 간선을 끊으면, A그룹(1번, 4번, 6번 주차장)과 B그룹(5번, 3번, 2번 주차장)으로 나누어집니다.
* A그룹의 수용 가능 대수는 19(6+9+4), B그룹의 수용 가능 대수는 22(8+10+4)입니다.
* |19 - 22 | = 3, 따라서 3을 return 하여야 합니다.