**문제 설명**

당신에게 직사각형 흑백 그림이 여러 개 주어집니다. 당신은 이 흑백 그림이 정확히 하나의 검은색 ㅁ (미음, Consonant 'ㅁ' in the Korean alphabet) 모양을 나타내고 있는지 알고 싶습니다. ㅁ 모양이란 외부는 검은색 직사각형 형태이며, 내부에 흰색 직사각형 구멍이 하나 뚫린 모양을 말합니다. 이때, ㅁ의 **각 변의 굵기와 길이는 얼마가 되든 상관없습니다.** 즉, ㅁ이 직사각형 모양을 하고 있다던가, 또는 각 변의 굵기가 제각각이더라도 온전한 ㅁ 모양입니다.

그림이 여러 개 담긴 2차원 문자열 배열 grids가 주어졌을 때, 각 그림이 정확히 하나의 검은색 ㅁ 모양을 나타내고 있으면 참값, 아니면 거짓값을 return 하도록 solution 함수를 완성해주세요.

제한사항

* 1 ≤ grids의 행의 개수 ≤ 10
  + 1 ≤ grids의 열의 길이 ≤ 500
  + 1 ≤ grids의 모든 문자열의 길이 ≤ 500
  + grids의 모든 문자열은 '.'과 'X'로만 이루어져 있습니다.
  + grids[i][j][k]는 i+1번째 그림의 j+1번째 가로줄의 k+1번째 세로줄의 색상을 의미합니다. 'X'면 해당 위치가 검은색임을, '.'면 해당 위치가 흰색임을 의미합니다.
  + 같은 그림 내의 모든 문자열의 길이는 전부 같습니다.

입출력 예

| **grids** | **result** |
| --- | --- |
| [[".....",".XXX.",".X.X.",".XXX.","....."],["XXXXX","XXXXX","XXX.X","XXX.X","XXXXX"],["XXXXX","X...X","X.X.X","X...X","XXXXX"],["....X",".....","XXX..","X.X..","XXX.."],[".......","XXX.XXX","X.X.X.X","XXX.XXX","......."],["XXXXX","XX.XX","X...X","XX.XX","XXXXX"]] | [true,true,false,false,false,false] |

입출력 예 설명

**입출력 예 #1**

* 주어진 예시에는 다음과 같은 6개의 그림이 있습니다.

텍스트, 낱말맞추기게임, 구급 상자이(가) 표시된 사진

자동 생성된 설명

1. 1번째 그림은 온전한 ㅁ 모양을 나타내고 있습니다.
2. 2번째 그림도 온전한 ㅁ 모양을 나타내고 있습니다. 다만 각 변의 굵기가 조금씩 다를 뿐입니다.
3. 3번째 그림은 온전한 ㅁ 모양을 나타내지 않습니다. ㅁ의 내부 공백 안에 검은색으로 칠해진 칸이 있기 때문입니다.
4. 4번째 그림도 온전한 ㅁ 모양을 나타내지 않습니다. ㅁ의 외부에 검은색으로 칠해진 칸이 있기 때문입니다.
5. 5번째 그림은 **하나의** 온전한 ㅁ 모양을 나타내지 않습니다. ㅁ 모양을 **2개** 가지고 있기 때문입니다.
6. 6번째 그림은 온전한 ㅁ 모양을 나타내지 않습니다. 내부 공백이 직사각형 형태가 아니기 때문입니다.

**문제 설명**

처음으로 저축한 날짜부터 마지막으로 저축한 날짜까지 매일 1회 이상 저축을 했다면 1일 1저축에 성공했다고 판단합니다. 예를 들어,  
첫 번째 저축을 2021년 4월 12일 오후 4시 8분 35초에 하고,  
두 번째 저축을 1일 6시간 30분뒤인 2021년 4월 13일 오후 10시 38분 35초에 하고,  
마지막 저축을 1일 4시간 12분뒤인 2021년 4월 15일 오전 2시 50분 35초에 했다면,

* 4월 14일에는 저축을 하지 않았기 때문에 1일 1저축은 실패입니다.
* 첫 저축 날짜는 4월 12일이고, 마지막 저축 날짜는 4월 15일이므로, 저축 기간은 4일입니다.
  + 저축 기간은 첫 저축 날짜부터 마지막 저축 날짜까지 포함하여 그 사이 모든 날짜를 **저축 유무와 상관없이** 한 번씩 센 기간입니다.

만약,  
첫 번째 저축을 2021년 4월 12일 오후 4시 8분 35초에 하고,  
두 번째 저축을 1일 6시간 30분뒤인 2021년 4월 13일 오후 10시 38분 35초에 하고,  
마지막 저축을 0일 1시간 12분뒤인 2021년 4월 13일 오후 11시 50분 35초에 했다면,

* 4월 12월부터 4월 13일까지 매일 1회 이상 저축을 했기 때문에, 1일 1저축은 성공입니다.
* 첫 저축 날짜는 4월 12일이고, 마지막 저축 날짜는 4월 13일이므로, 저축 기간은 2일입니다.

첫 저축을 시작한 시각인 문자열 s, 다음 저축까지 걸린 기간을 담은 문자열 배열 times가 주어집니다. 이때 1일 1저축의 성공 여부와 저축 기간(일)을 배열에 담아서 return 하도록 solution 함수를 완성해주세요.

제한사항

* **문제를 간단하게 하기 위해, 윤년은 없으며 모든 달은 30일이 마지막 날이라고 가정합니다. 따라서, 1년은 360일입니다.**
* return 값 형식
  + 길이 2인 정수형 배열을 return 합니다.
  + [1일 1저축의 성공 여부, 저축한 기간]을 return 합니다.
  + 첫 번째 원소로 1일 1저축에 성공했다면 1을, 성공하지 못했다면 0을 담습니다.
  + 두 번째 원소로 저축 기간(일)을 담습니다.
* s는 YYYY:MM:DD:HH:mm:SS 형태로, 길이가 11인 문자열입니다.
  + YYYY는 년도를 뜻합니다. 1년은 0001:MM:DD:HH:mm:SS 형태로 주어집니다.
    - 1 ≤ YYYY ≤ 9999
  + MM은 월을 뜻합니다. 1월은 YYYY:01:DD:HH:mm:SS 형태로 주어집니다.
    - 1 ≤ MM ≤ 12
  + DD는 일을 뜻합니다. 1일은 YYYY:MM:01:HH:mm:SS 형태로 주어집니다.
    - 1 ≤ DD ≤ 30
  + HH는 시를 뜻합니다. 오전 1시는 YYYY:MM:DD:01:mm:SS 형태로 주어집니다.
    - 0 ≤ HH ≤ 23
  + mm은 분을 뜻합니다. 1분은 YYYY:MM:DD:HH:01:SS 형태로 주어집니다.
    - 0 ≤ mm ≤ 59
  + SS는 초를 뜻합니다. 1초는 YYYY:MM:DD:HH:mm:01 형태로 주어집니다.
    - 0 ≤ SS ≤ 59
  + 예를 들어, 1년 1월 1일 오전 1시 1분 1초는 0001:01:01:01:01:01 형태로 주어집니다. 2021년 11월 30일 오후 10시 4분 24초는 2021:11:30:22:04:24형태로 주어집니다.
* 0 ≤ times의 길이 ≤ 1,000
* times의 원소는 DD:HH:mm:SS의 형태로 주어집니다.
  + DD는 일을 뜻합니다. 1일은 01:HH:mm:SS 형태로 주어집니다.
    - 0 ≤ DD ≤ 99
  + HH는 시를 뜻합니다. 1시간은 DD:01:mm:SS 형태로 주어집니다.
    - 0 ≤ HH ≤ 23
  + mm은 분을 뜻합니다. 1분은 DD:HH:01:SS 형태로 주어집니다.
    - 0 ≤ mm ≤ 59
  + SS는 초를 뜻합니다. 1초는 DD:HH:mm:01 형태로 주어집니다.
    - 0 ≤ SS ≤ 59
  + 예를 들어, 0일 2시간 5분 18초는 00:02:05:18 형태로 주어집니다.
* 마지막 저축 시각이 9999년을 초과하는 입력은 주어지지 않습니다.

입출력 예

| **s** | **times** | **result** |
| --- | --- | --- |
| "2021:04:12:16:08:35" | ["01:06:30:00", "01:04:12:00"] | [0, 4] |
| "2021:04:12:16:08:35" | ["01:06:30:00", "00:01:12:00"] | [1, 2] |
| "2021:04:12:16:10:42" | ["01:06:30:00"] | [1, 2] |
| "2021:04:12:16:08:35" | ["01:06:30:00", "01:01:12:00", "00:00:09:25"] | [1, 4] |

입출력 예 설명

**입출력 예 #1**

문제 첫 번째 예시와 같습니다.

1일 1저축을 실패했습니다. 저축 기간은 4월 12일 ~ 4월 15일, 총 4일입니다.

따라서 정답은 [0, 4]입니다.

**입출력 예 #2**

문제 두 번째 예시와 같습니다.

1일 1저축을 성공했습니다. 저축 기간은 4월 12일 ~ 4월 13일, 총 2일입니다.

따라서 정답은 [1, 2]입니다.

**입출력 예 #3**

* 첫 번째 저축은 2021년 4월 12일 오후 4시 10분 42초에 합니다.
* 두 번째 저축은 1일 6시간 30분뒤인 2021년 4월 13일 오후 10시 40분 42초에 합니다.

4월 12일과 4월 13일 모두 저축을 했기 때문에 1일 1저축에 성공했습니다. 저축기간은 4월 12일 ~ 4월 13일, 총 2일입니다.

따라서 정답은 [1, 2]입니다.

**입출력 예 #4**

* 첫 번째 저축은 2021년 4월 12일 오후 4시 8분 35초에 합니다.
* 두 번째 저축은 1일 6시간 30분뒤인 2021년 4월 13일 오후 10시 38분 35초에 합니다.
* 세 번째 저축은 1일 1시간 12분뒤인 2021년 4월 14일 오후 11시 50분 35초에 합니다.
* 네 번째 저축은 0일 0시간 9분 25초뒤인 2021년 4월 15일 오전 0시 0분 0초에 합니다.

4월 12일부터 4월 15일까지 모두 저축을 했기 때문에 1일 1저축에 성공했습니다. 저축기간은 4월 12일 ~ 4월 15일, 총 4일입니다.

따라서 정답은 [1, 4]입니다.

**문제 설명**

코드 품질 향상을 위해, 코드 리뷰를 실시하고자 합니다. 모든 개발자는 자신과 같은 프로젝트를 수행하는 동료들 중, n명에게 코드 리뷰를 받아야 합니다. 단, 코드 리뷰에 너무 많은 시간을 투자하면 업무에 지장이 있을 수 있으므로, 한 명의 개발자가 리뷰할 수 있는 동료의 수는 최대 k명으로 제한합니다.

프로젝트 1 : D, B, E, X

프로젝트 2 : A, C, E

프로젝트 3 : C, D, X

프로젝트 4 : A, B, G, X

위와 같이 알파벳 대문자로 표현된 7명(A, B, C, D, E, G, X)의 개발자가 4개의 프로젝트를 수행하고 있고, n = 2, k = 3 인 경우, 다음과 같이 코드 리뷰어를 지정해줄 수 있습니다.

| **개발자** | **A** | **B** | **C** | **D** | **E** | **G** | **X** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **리뷰어 지정 방법 1** | B,C | A,D | A,D | B,C | A,C | B,X | D,E |
| 리뷰어 지정 방법 2 | E,C | D,E | E,A | X,B | D,B | A,B | C,D |
|  | B,C | **A**,D | **A**,D | B,C | **A**,C | **A**,X | D,E |
|  | B,C | A,**C** | A,D | B,C | A,C | B,X | D,E |
|  | E,C | **D,D** | E,A | X,B | D,B | A,B | **C** |

* 취소선이 그어진 방법 3은 A가 4명의 코드를 리뷰하기로 지정되었으므로, 잘못된 방법입니다.
  + k = 3 이므로, 한 명이 리뷰할 수 있는 최대 동료의 수는 3명입니다.
* 취소선이 그어진 방법 4는 **B와 같은 프로젝트를 수행하고 있지 않는 C**가 B의 코드를 리뷰하기로 지정되었습니다. 따라서 잘못된 방법입니다.
* 취소선이 그어진 방법 5는 X의 리뷰어로 두 명이 아닌 한 명만 지정되었습니다. 뿐만 아니라, B의 리뷰어로 서로 다른 두 명이 지정되어야 하는데, D만 두 번 지정되었습니다. 따라서 잘못된 방법입니다.
  + n = 2 이므로, 모든 개발자는 같은 프로젝트를 수행하고 있는 n명의 개발자들에게 코드 리뷰를 받아야 합니다.
* 방법 1, 2는 모두 올바른 방법입니다. 이 중에서, 알파벳 순서가 빠른 개발자부터 자신의 코드를 리뷰할 동료들을 차례대로 이어 붙여서 문자열을 만들었을 때, 사전 순으로 빠른 방법을 정답으로 간주합니다. 즉, 올바른 리뷰어 지정 방법을 나타내는 행에서, 쉼표를 제외하고 알파벳만 차례대로 이어 붙였을 때 사전 순으로 빠른 것을 정답으로 간주합니다.
  + 리뷰어 지정 방법 1을 나타내는 행에서 알파벳만 차례대로 이어 붙여서 만든 문자열은 "BCADADBCACBXDE"입니다.
  + 리뷰어 지정 방법 2를 나타내는 행에서 알파벳만 차례대로 이어 붙여서 만든 문자열은 "ECDEEAXBDBABCD"입니다.
  + "BCADADBCACBXDE" < "ECDEEAXBDBABCD" 이므로, 방법 1이 정답입니다.
* 방법 1, 2 말고도 주어진 조건에 맞게 코드 리뷰어를 지정하는 방법은 더 있습니다. 하지만, 사전 순으로 "BCADADBCACBXDE" 보다 빠른 방법으로 리뷰어를 할당하는 방법은 없습니다.

개발자들이 진행하고 있는 프로젝트를 나타내는 문자열 배열 prj, 각 개발자들이 리뷰를 받아야 하는 동료의 수를 나타내는 정수 n, 한 명의 개발자가 리뷰를 할 수 있는 최대 인원을 나타내는 정수 k가 매개변수로 주어집니다. 알파벳 순서가 빠른 개발자부터 자신의 코드를 리뷰할 동료를 차례대로 이어 붙여서 문자열을 만들었을 때 사전 순으로 가장 빠른 문자열을 return 하도록 solution 함수를 완성해주세요.

제한사항

* 1 ≤ prj의 길이(프로젝트의 개수) ≤ 30
* 1 ≤ prj의 원소의 길이 ≤ 7
  + 즉, 한 프로젝트에 최대 7명의 개발자가 포함될 수 있습니다.
  + prj의 원소는 알파벳 대문자로만 이루어진 문자열입니다.
  + prj의 원소는 같은 알파벳을 2개 이상 포함하고 있지 않습니다.
  + 예를 들어, "AABC", "GHWGGH" 등은 prj의 원소가 될 수 없습니다.
* 2 ≤ prj에서 나타날 수 있는 서로 다른 알파벳의 개수 ≤ 7
  + 즉, 전체 개발자는 최소 2명에서 최대 7명입니다.
* 1 ≤ n ≤ 3
* 1 ≤ k ≤ 3
* **문제의 조건에 맞게 코드 리뷰어를 지정하는 방법이 있는 경우만 입력으로 주어집니다.**

입출력 예

| **prj** | **n** | **k** | **result** |
| --- | --- | --- | --- |
| ["DBEX", "ACE", "CDX", "ABGX"] | 2 | 3 | "BCADADBCACBXDE" |
| ["OR", "QO", "R", "OQ", "OR"] | 1 | 2 | "QOO" |

입출력 예 설명

**입출력 예 #1**

문제 예시와 같습니다.

**입출력 예 #2**

전체 개발자 3명을 알파벳 순으로 오름차순 정렬하면 [O, Q, R]입니다.  
O, Q, R에 대한 리뷰어를 지정하는 방법은 다음 표와 같이 두 가지입니다.

| **개발자** | **O** | **Q** | **R** |
| --- | --- | --- | --- |
| **리뷰어 지정 방법 1** | Q | O | O |
| 리뷰어 지정 방법 2 | R | O | O |

"QOO" < "ROO" 이므로, "QOO"가 정답입니다.