**문제 설명**

어떤 은행에서는 고객들이 송금할 때, 다음의 두 가지 조건 중 **한 가지 이상** 해당하면 보이스피싱 주의 문자를 발송합니다.

**[문자 발송 조건]**

1. 같은 사람에게 k번 이상 연속으로 송금할 때
   * 사람 이름을 비교할 때, 대소문자는 구분하지 않습니다.
2. 송금액이 m원 이상일 때

아래의 표는 k = 3, m = 50000 인 경우에 고객들의 송금 내역에 따른 문자 발송 결과입니다.

| **받는 사람 이름** | **송금액** | **결과** |
| --- | --- | --- |
| msLEE | 950 |  |
| jsKim | 52524 | 송금액이 50000원 이상이므로, 문자 발송 |
| jsKIM | 1400 |  |
| jskiM | 6055 | 3번 연속으로 같은 사람에게 송금하므로, 문자 발송 |
| jskim | 10000 | 4번 연속으로 같은 사람에게 송금하므로, 문자 발송 |
| John | 4512 |  |
| john | 512 |  |
| John | 52000 | 두 가지 조건 모두 해당하므로, 문자 발송 |
| msLEE | 9000 |  |
| msLEE | 49000 |  |
| jsKIM | 1400 |  |
| roy | 50000 | 송금액이 50000원 이상이므로, 문자 발송 |

* 위의 예시에서는 총 5회 문자가 발송됩니다.

고객들에게 문자를 발송하는 기준이 되는 정수 k와 m, 송금을 받는 사람의 이름을 담은 문자열 배열 names, 송금액을 담은 정수 배열 amounts가 매개변수로 주어집니다. 이때, 문자가 발송되는 횟수를 return 하도록 solution 함수를 완성해주세요. 만약 문자가 한 번도 발송되지 않으면, 0을 return 합니다.

제한사항

* 2 ≤ k ≤ 10
* 1 ≤ m ≤ 1,000,000
* 1 ≤ names의 길이 ≤ 1,000
  + 3 ≤ names의 원소의 길이 ≤ 10
  + names의 원소는 알파벳 소문자와 알파벳 대문자로만 이루어진 문자열입니다.
* amounts의 길이 = names의 길이
  + 1 ≤ amounts의 원소 ≤ 1,000,000
* names[i]와 amounts[i]는 i+1 번째 송금 내역에서 받는 사람 이름과 송금액을 의미합니다.

입출력 예

| **k** | **m** | **names** | **amounts** | **result** |
| --- | --- | --- | --- | --- |
| 3 | 50000 | ["msLEE", "jsKim", "jsKIM", "jskiM", "jskim", "John", "john", "John", "msLEE", "msLEE", "jsKIM", "roy"] | [950, 52524, 1400, 6055, 10000, 4512, 512, 52000, 9000, 49000, 1400, 50000] | 5 |
| 2 | 3451 | ["abcd", "aBCd", "jsKIM", "rrr", "rrr"] | [950, 1000, 1400, 4000, 10000] | 3 |

입출력 예 설명

**입출력 예 #1**

* 문제 예시와 같습니다. 총 5회 문자가 발송되므로, 5를 return 합니다.

**입출력 예 #2**

| **받는 사람 이름** | **송금액** | **결과** |
| --- | --- | --- |
| abcd | 950 |  |
| aBCd | 1000 | 2회 연속으로 같은 사람에게 송금하므로, 문자 발송 |
| jsKIM | 1400 |  |
| rrr | 4000 | 송금액이 3451원 이상이므로, 문자 발송 |
| rrr | 10000 | 두 가지 조건 모두에 해당하므로, 문자 발송 |

* 총 3회 문자가 발송되므로, 3을 return 합니다.

**문제 설명**

당신은 검색 기능을 가진 사이트를 운영하고 있습니다. 당신은 매월 초마다 전월의 **최고의 이슈 검색어**를 조사하고 있습니다. **최고의 이슈 검색어**를 조사하는 방법은 아래와 같습니다.

* 전월의 사용자들의 검색 기록을 일 기준으로 나눈 research가 주어집니다.
* research를 일 기준으로 어떤 검색어가 몇 번 검색되었는지 정리합니다.
* 어떤 검색어가 연속된 n일 동안 매일 최소 k 번 이상 검색되고, 같은 연속된 n일 동안 총 2 x n x k번 이상 검색되었을 경우 이슈 검색어가 됩니다.
  + 예를 들어, n이 3이고 k가 50이면 그 전월에서 임의의 연속된 3일을 골랐을 때, 매일 50번 이상씩 검색되고 같은 기간 동안 총 300번 이상 검색되면 이슈 검색어가 됩니다.
* 이슈 검색어 중에서 가장 여러 번 이슈 검색어가 된 검색어가 **최고의 이슈 검색어**가 됩니다.
  + 가장 여러 번 이슈 검색어가 된 검색어가 여러 개일 경우 사전 순으로 빠른 검색어가 **최고의 이슈 검색어**가 됩니다.
  + 예를 들어, "a"가 2번 이슈 검색어가 되었고 "c"도 2번 이슈 검색어가 되었다면 "a"가 **최고의 이슈 검색어**가 됩니다.

그 전월의 사용자들의 검색 기록을 일 기준으로 나눈 1차원 문자열 배열 research, 정수 n, k가 매개변수로 주어집니다. 그 전월의 **최고의 이슈 검색어**를 조사하여 return 하도록 solution 함수를 완성해 주세요. **최고의 이슈 검색어**가 없으면 "None"을 return해 주세요.

제한사항

* 2 ≤ research의 길이(=그 전월의 일 수) ≤ 30
  + 2 ≤ research의 원소의 길이 ≤ 100
  + research의 원소는 "검색어1검색어2..." 형태입니다. 하루 동안 검색된 검색어를 순서 없이 붙여 놓은 형태입니다.
    - 검색어의 길이 = 1
    - 검색어는 알파벳 소문자 중 하나입니다.
    - 검색어는 서로 겹칠 수 있습니다.
    - 예를 들어, "abaa"는 하루에 "a"가 3번 "b"가 1번 검색됐다는 뜻입니다.
  + research에는 1일부터 하루 동안 검색된 검색어들이 차례대로 담겨져있습니다.
* 1 ≤ n ≤ research의 길이
* 1 ≤ k ≤ 100

입출력 예

| **research** | **n** | **k** | **result** |
| --- | --- | --- | --- |
| ["abaaaa","aaa","abaaaaaa","fzfffffffa"] | 2 | 2 | "a" |
| ["yxxy","xxyyy"] | 2 | 1 | "x" |
| ["yxxy","xxyyy","yz"] | 2 | 1 | "y" |
| ["xy","xy"] | 1 | 1 | "None" |

입출력 예 설명

**입출력 예 #1**

총 4일 동안의 검색 기록이 주어졌습니다.  
아래 표는 전월의 모든 검색어를 날마다 몇 번 검색되었는지 정리한 표입니다.

| **검색어** | **1일** | **2일** | **3일** | **4일** |
| --- | --- | --- | --- | --- |
| a | **5** | **3** | **7** | 1 |
| b | 1 | 0 | 1 | 0 |
| f | 0 | 0 | 0 | 8 |
| z | 0 | 0 | 0 | 1 |

연속된 2(=n)일 동안 날마다 2(=k)번 이상 검색되고, 같은 연속된 2(=n)일 동안 총 8번(=2 x 2 x 2) 이상 검색된 검색어는 1일부터 2일까지 그리고 2일부터 3일까지 검색된 "a"뿐입니다. (4일에 k번 미만으로 검색되었으므로 3일부터 4일까지의 "a"는 이슈 검색어가 아닙니다.) 1일부터 2일까지 그리고 2일부터 3일까지 두 번 이슈 검색어가 된 "a"를 제외한 다른 검색어는 한 번도 이슈 검색어가 되지 못했습니다.  
따라서 "a"를 return 합니다.

**입출력 예 #2**

총 2일 동안의 검색 기록이 주어졌습니다.  
아래 표는 전월의 모든 검색어를 날마다 몇 번 검색되었는지 정리한 표입니다.

| **검색어** | **1일** | **2일** |
| --- | --- | --- |
| x | **2** | **2** |
| y | **2** | **3** |

연속된 2(=n)일 동안 날마다 1(=k)번 이상 검색되고, 같은 연속된 2(=n)일 동안 총 4번(=2 x 2 x 1) 이상 검색된 검색어는 "x"와 "y"이고 두 검색어 모두 1일부터 2일까지 한 번만 이슈 검색어가 되었습니다.  
따라서 사전 순으로 더 빠른 "x"를 return 합니다.

**입출력 예 #3**

총 3일 동안의 검색 기록이 주어졌습니다.  
아래 표는 전월의 모든 검색어를 날마다 몇 번 검색되었는지 정리한 표입니다.

| **검색어** | **1일** | **2일** | **3일** |
| --- | --- | --- | --- |
| x | **2** | **2** | 0 |
| y | **2** | **3** | **1** |
| z | 0 | 0 | 1 |

연속된 2(=n)일 동안 날마다 1(=k)번 이상 검색되고, 같은 연속된 2(=n)일 동안 총 4번(=2 x 2 x 1) 이상 검색된 검색어는 "x"와 "y"입니다. 이때 "x"는 1일부터 2일까지 단 한 번만 이슈 검색어가 되었고 "y"는 1일부터 2일까지 그리고 2일부터 3일까지 2번 이슈 검색어가 되었습니다.  
따라서 더 많이 이슈 검색어가 된 "y"를 return 합니다.

**입출력 예 #4**

총 2일 동안의 검색 기록이 주어졌습니다.  
아래 표는 전월의 모든 검색어를 날마다 몇 번 검색되었는지 정리한 표입니다.

| **검색어** | **1일** | **2일** |
| --- | --- | --- |
| x | 1 | 1 |
| y | 1 | 1 |

연속된 1(=n)일 동안 날마다 1(=k)번 이상 검색되고, 같은 연속된 1(=n)일 동안 총 2번(=2 x 1 x 1) 이상 검색된 검색어는 없습니다.  
따라서, 이슈 검색어가 없으므로 "None"을 return 합니다.

**문제 설명**

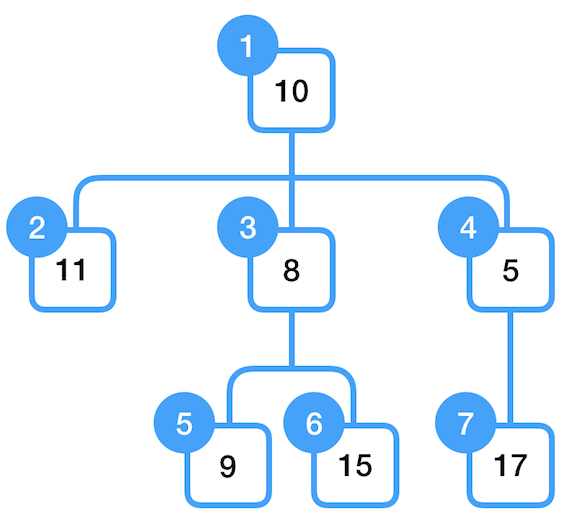
철수는 크라우드 펀딩을 이용한 프로젝트 투자 계획을 세우려 합니다. 철수가 투자하려는 프로젝트는 다음과 같은 특징을 가지고있습니다.

1. 철수가 투자하려는 프로젝트는 최상위 프로젝트라고 부릅니다.
2. 최상위 프로젝트는 여러개의 하위 프로젝트를 가질 수 있습니다.  
   2-1. 각각의 하위 프로젝트는 다시 여러개의 하위 프로젝트를 가질 수 있습니다.  
   2-2. 하위 프로젝트를 가지지 않는 프로젝트들을 최하위 프로젝트라고 부릅니다.
3. 모든 하위 프로젝트와 최상위 프로젝트는 기대 이익값을 가지고있습니다.

위 특징에 맞춰 철수는 해당 프로젝트에 투자했을 때의 투자 이익값을 다음과 같이 구합니다.

* 각 프로젝트의 투자 이익값은 [프로젝트의 기대 이익값] + [하위 프로젝트의 투자 이익값 중 최댓값]
* 단, 최하위 프로젝트의 투자 이익값은 해당 프로젝트의 기대 이익값과 같습니다.

다음은 철수가 투자하려는 프로젝트의 예시입니다.



위 그림에서 원 안에 적힌 숫자는 프로젝트 번호를, 사각형 안에 적힌 숫자는 각 프로젝트의 기대 이익값을 나타냅니다. 항상 1번 프로젝트가 철수가 투자하려는 최상위 프로젝트입니다. 1번 프로젝트는 2, 3, 4번의 하위 프로젝트를 가지며, 3번 프로젝트는 5, 6번의 하위 프로젝트를 가지고, 4번 프로젝트는 7번의 하위 프로젝트를 가집니다. 각 프로젝트의 투자 이익값을 구하는 예시는 다음과 같습니다.

* 2, 5, 6, 7번 프로젝트의 투자 이익값 = 프로젝트의 기대 이익값(각각 11, 9, 15, 17)
* 3번 프로젝트의 투자 이익값 = 기대 이익값(8) + 하위 프로젝트의 투자 이익값 중 최댓값(15) = 23
* 4번 프로젝트의 투자 이익값 = 기대 이익값(5) + 하위 프로젝트의 투자 이익값 중 최댓값(17) = 22
* 1번 프로젝트의 투자 이익값 = 기대 이익값(10) + 하위 프로젝트의 투자 이익값 중 최댓값(23) = 33

따라서 위 예시의 경우 최상위 프로젝트의 투자 이익값은 33이 됩니다.

각 프로젝트의 기대 이익값이 번호 순서대로 담긴 배열 value, 프로젝트의 관계가 담긴 2차원 배열 projects가 매개변수로 주어질 때, 최상위 프로젝트의 투자 이익값을 return 하도록 solution 함수를 완성해주세요.

제한사항

* 전체 프로젝트개수는 value의 길이와 같습니다.
* value의 길이는 2 이상 10,000 이하입니다.
* value의 원소는 1 이상 10,000 이하인 자연수입니다.
* 모든 프로젝트에는 1부터 value의 길이에 해당하는 숫자까지 하나씩 번호가 붙어있습니다.
  + 기대 이익값은 프로젝트 번호가 증가하는 순서대로 배열에 담겨있습니다.
* projects의 길이는 [value의 길이 - 1] 입니다.
* projects의 각 원소는 [a, b]형태입니다.
  + b가 a의 하위 프로젝트 임을 나타냅니다.
* 1번 프로젝트가 항상 최상위 프로젝트입니다.

입출력 예

| **value** | **projects** | **result** |
| --- | --- | --- |
| [10, 11, 8, 5, 9, 15, 17] | [[1, 2], [1, 3], [1, 4], [3, 5], [3, 6], [4, 7]] | 33 |

입출력 예 설명

입출력 예 #1

문제의 예시와 같습니다.