**모의고사**

수포자는 수학을 포기한 사람의 준말입니다. 수포자 삼인방은 모의고사에 수학 문제를 전부 찍으려 합니다. 수포자는 1번 문제부터 마지막 문제까지 다음과 같이 찍습니다.

1번 수포자가 찍는 방식: 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, ...

2번 수포자가 찍는 방식: 2, 1, 2, 3, 2, 4, 2, 5, 2, 1, 2, 3, 2, 4, 2, 5, ...

3번 수포자가 찍는 방식: 3, 3, 1, 1, 2, 2, 4, 4, 5, 5, 3, 3, 1, 1, 2, 2, 4, 4, 5, 5, ...

1번 문제부터 마지막 문제까지의 정답이 순서대로 들은 배열 answers가 주어졌을 때, 가장 많은 문제를 맞힌 사람이 누구인지 배열에 담아 return 하도록 solution 함수를 작성해주세요.

제한 조건

시험은 최대 10,000 문제로 구성되어있습니다.

문제의 정답은 1, 2, 3, 4, 5중 하나입니다.

가장 높은 점수를 받은 사람이 여럿일 경우, return하는 값을 오름차순 정렬해주세요.

입출력 예

answers return

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

[1,3,2,4,2] [1,2,3]

입출력 예 설명

입출력 예 #1

수포자 1은 모든 문제를 맞혔습니다.

수포자 2는 모든 문제를 틀렸습니다.

수포자 3은 모든 문제를 틀렸습니다.

따라서 가장 문제를 많이 맞힌 사람은 수포자 1입니다.

입출력 예 #2

모든 사람이 2문제씩을 맞췄습니다.

**소수 찾기**

한자리 숫자가 적힌 종이 조각이 흩어져있습니다. 흩어진 종이 조각을 붙여 소수를 몇 개 만들 수 있는지 알아내려 합니다.

각 종이 조각에 적힌 숫자가 적힌 문자열 numbers가 주어졌을 때, 종이 조각으로 만들 수 있는 소수가 몇 개인지 return 하도록 solution 함수를 완성해주세요.

제한사항

numbers는 길이 1 이상 7 이하인 문자열입니다.

numbers는 0~9까지 숫자만으로 이루어져 있습니다.

013은 0, 1, 3 숫자가 적힌 종이 조각이 흩어져있다는 의미입니다.

입출력 예

numbers return

17 3

011 2

입출력 예 설명

예제 #1

[1, 7]으로는 소수 [7, 17, 71]를 만들 수 있습니다.

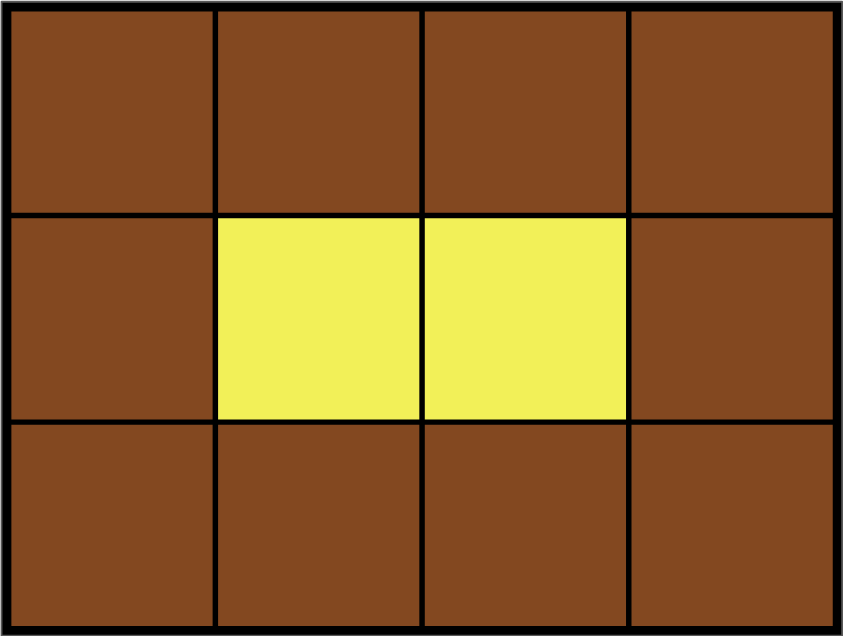
예제 #2

[0, 1, 1]으로는 소수 [11, 101]를 만들 수 있습니다.

11과 011은 같은 숫자로 취급합니다.

**카펫**

Leo는 카펫을 사러 갔다가 아래 그림과 같이 중앙에는 노란색으로 칠해져 있고 테두리 1줄은 갈색으로 칠해져 있는 격자 모양 카펫을 봤습니다.



Leo는 집으로 돌아와서 아까 본 카펫의 노란색과 갈색으로 색칠된 격자의 개수는 기억했지만, 전체 카펫의 크기는 기억하지 못했습니다.

Leo가 본 카펫에서 갈색 격자의 수 brown, 노란색 격자의 수 yellow가 매개변수로 주어질 때 카펫의 가로, 세로 크기를 순서대로 배열에 담아 return 하도록 solution 함수를 작성해주세요.

제한사항

갈색 격자의 수 brown은 8 이상 5,000 이하인 자연수입니다.

노란색 격자의 수 yellow는 1 이상 2,000,000 이하인 자연수입니다.

카펫의 가로 길이는 세로 길이와 같거나, 세로 길이보다 깁니다.

입출력 예

brown yellow return

10 2 [4, 3]

8 1 [3, 3]

24 24 [8, 6]

**네트워크**

문제 설명

네트워크란 컴퓨터 상호 간에 정보를 교환할 수 있도록 연결된 형태를 의미합니다. 예를 들어, 컴퓨터 A와 컴퓨터 B가 직접적으로 연결되어있고, 컴퓨터 B와 컴퓨터 C가 직접적으로 연결되어 있을 때 컴퓨터 A와 컴퓨터 C도 간접적으로 연결되어 정보를 교환할 수 있습니다. 따라서 컴퓨터 A, B, C는 모두 같은 네트워크 상에 있다고 할 수 있습니다.

컴퓨터의 개수 n, 연결에 대한 정보가 담긴 2차원 배열 computers가 매개변수로 주어질 때, 네트워크의 개수를 return 하도록 solution 함수를 작성하시오.

제한사항

컴퓨터의 개수 n은 1 이상 200 이하인 자연수입니다.

각 컴퓨터는 0부터 n-1인 정수로 표현합니다.

i번 컴퓨터와 j번 컴퓨터가 연결되어 있으면 computers[i][j]를 1로 표현합니다.

computer[i][i]는 항상 1입니다.

입출력 예

n computers return

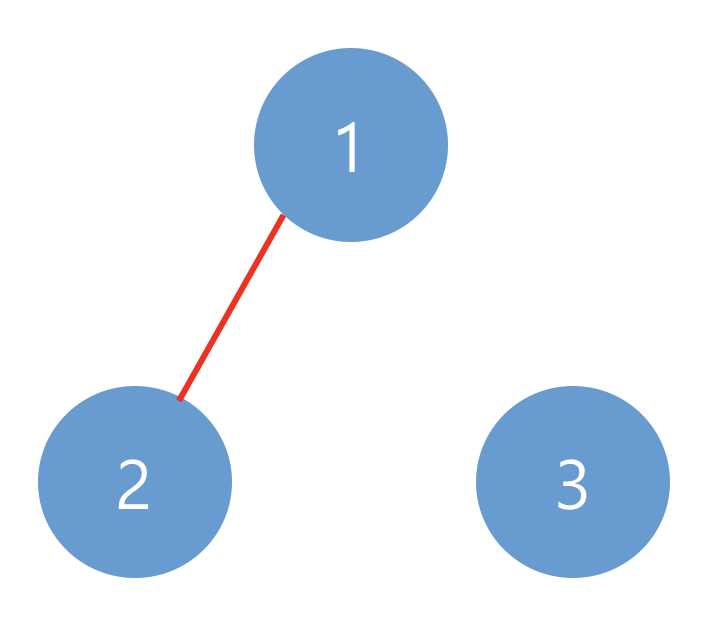
3 [[1, 1, 0], [1, 1, 0], [0, 0, 1]] 2

3 [[1, 1, 0], [1, 1, 1], [0, 1, 1]] 1

입출력 예 설명

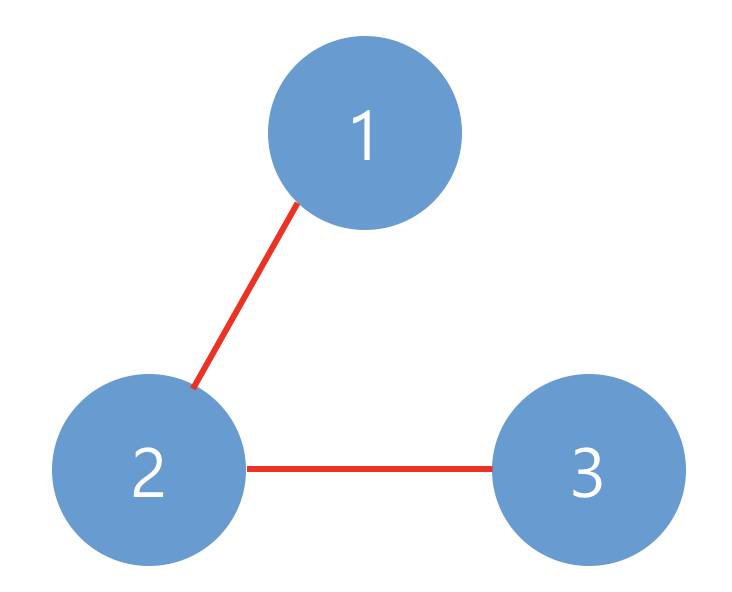
예제 #1

아래와 같이 2개의 네트워크가 있습니다.



예제 #2

아래와 같이 1개의 네트워크가 있습니다.



**단어 변환**

문제 설명

두 개의 단어 begin, target과 단어의 집합 words가 있습니다. 아래와 같은 규칙을 이용하여 begin에서 target으로 변환하는 가장 짧은 변환 과정을 찾으려고 합니다.

1. 한 번에 한 개의 알파벳만 바꿀 수 있습니다.

2. words에 있는 단어로만 변환할 수 있습니다.

예를 들어 begin이 hit, target가 cog, words가 [hot,dot,dog,lot,log,cog]라면 hit -> hot -> dot -> dog -> cog와 같이 4단계를 거쳐 변환할 수 있습니다.

두 개의 단어 begin, target과 단어의 집합 words가 매개변수로 주어질 때, 최소 몇 단계의 과정을 거쳐 begin을 target으로 변환할 수 있는지 return 하도록 solution 함수를 작성해주세요.

제한사항

각 단어는 알파벳 소문자로만 이루어져 있습니다.

각 단어의 길이는 3 이상 10 이하이며 모든 단어의 길이는 같습니다.

words에는 3개 이상 50개 이하의 단어가 있으며 중복되는 단어는 없습니다.

begin과 target은 같지 않습니다.

변환할 수 없는 경우에는 0를 return 합니다.

입출력 예

begin target words return

hit cog [hot, dot, dog, lot, log, cog] 4

hit cog [hot, dot, dog, lot, log] 0

입출력 예 설명

예제 #1

문제에 나온 예와 같습니다.

예제 #2

target인 cog는 words 안에 없기 때문에 변환할 수 없습니다.

**여행경로**

문제 설명

주어진 항공권을 모두 이용하여 여행경로를 짜려고 합니다. 항상 ICN 공항에서 출발합니다.

항공권 정보가 담긴 2차원 배열 tickets가 매개변수로 주어질 때, 방문하는 공항 경로를 배열에 담아 return 하도록 solution 함수를 작성해주세요.

제한사항

모든 공항은 알파벳 대문자 3글자로 이루어집니다.

주어진 공항 수는 3개 이상 10,000개 이하입니다.

tickets의 각 행 [a, b]는 a 공항에서 b 공항으로 가는 항공권이 있다는 의미입니다.

주어진 항공권은 모두 사용해야 합니다.

만일 가능한 경로가 2개 이상일 경우 알파벳 순서가 앞서는 경로를 return 합니다.

모든 도시를 방문할 수 없는 경우는 주어지지 않습니다.

입출력 예

tickets return

[[ICN, JFK], [HND, IAD], [JFK, HND]] [ICN, JFK, HND, IAD]

[[ICN, SFO], [ICN, ATL], [SFO, ATL], [ATL, ICN], [ATL,SFO]] [ICN, ATL, ICN, SFO, ATL, SFO]

입출력 예 설명

예제 #1

[ICN, JFK, HND, IAD] 순으로 방문할 수 있습니다.

예제 #2

[ICN, SFO, ATL, ICN, ATL, SFO] 순으로 방문할 수도 있지만 [ICN, ATL, ICN, SFO, ATL, SFO] 가 알파벳 순으로 앞섭니다.

**타겟 넘버**

문제 설명

n개의 음이 아닌 정수가 있습니다. 이 수를 적절히 더하거나 빼서 타겟 넘버를 만들려고 합니다. 예를 들어 [1, 1, 1, 1, 1]로 숫자 3을 만들려면 다음 다섯 방법을 쓸 수 있습니다.

-1+1+1+1+1 = 3

+1-1+1+1+1 = 3

+1+1-1+1+1 = 3

+1+1+1-1+1 = 3

+1+1+1+1-1 = 3

사용할 수 있는 숫자가 담긴 배열 numbers, 타겟 넘버 target이 매개변수로 주어질 때 숫자를 적절히 더하고 빼서 타겟 넘버를 만드는 방법의 수를 return 하도록 solution 함수를 작성해주세요.

제한사항

주어지는 숫자의 개수는 2개 이상 20개 이하입니다.

각 숫자는 1 이상 50 이하인 자연수입니다.

타겟 넘버는 1 이상 1000 이하인 자연수입니다.

입출력 예

numbers target return

[1, 1, 1, 1, 1] 3 5

입출력 예 설명

문제에 나온 예와 같습니다.

**섬 연결하기**

문제 설명

n개의 섬 사이에 다리를 건설하는 비용(costs)이 주어질 때, 최소의 비용으로 모든 섬이 서로 통행 가능하도록 만들 때 필요한 최소 비용을 return 하도록 solution을 완성하세요.

다리를 여러 번 건너더라도, 도달할 수만 있으면 통행 가능하다고 봅니다. 예를 들어 A 섬과 B 섬 사이에 다리가 있고, B 섬과 C 섬 사이에 다리가 있으면 A 섬과 C 섬은 서로 통행 가능합니다.

제한사항

섬의 개수 n은 1 이상 100 이하입니다.

costs의 길이는 ((n-1) \* n) / 2이하입니다.

임의의 i에 대해, costs[i][0] 와 costs[i] [1]에는 다리가 연결되는 두 섬의 번호가 들어있고, costs[i] [2]에는 이 두 섬을 연결하는 다리를 건설할 때 드는 비용입니다.

같은 연결은 두 번 주어지지 않습니다. 또한 순서가 바뀌더라도 같은 연결로 봅니다. 즉 0과 1 사이를 연결하는 비용이 주어졌을 때, 1과 0의 비용이 주어지지 않습니다.

모든 섬 사이의 다리 건설 비용이 주어지지 않습니다. 이 경우, 두 섬 사이의 건설이 불가능한 것으로 봅니다.

연결할 수 없는 섬은 주어지지 않습니다.

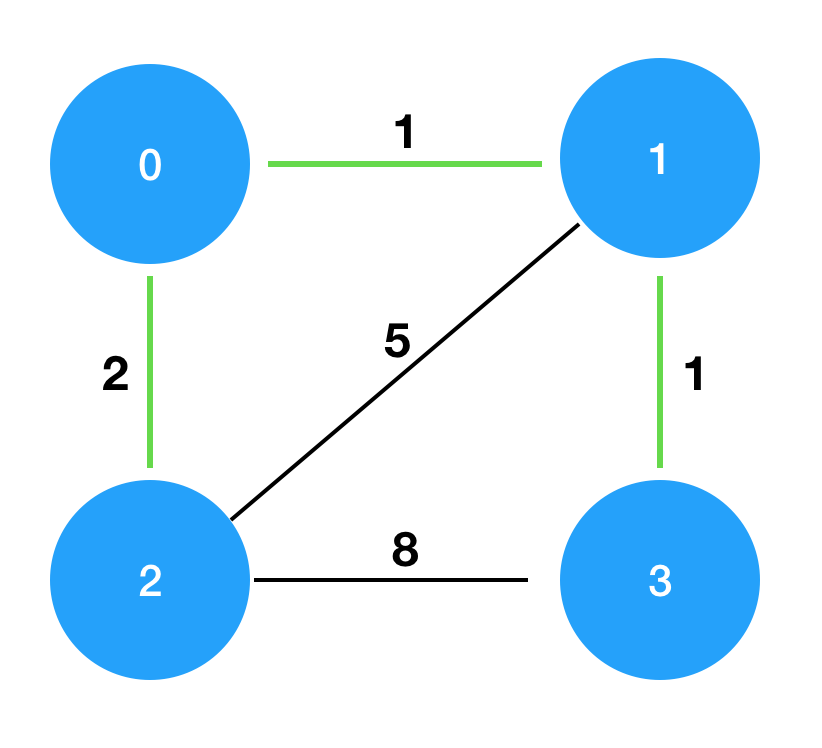
입출력 예

n costs return

4 [[0,1,1],[0,2,2],[1,2,5],[1,3,1],[2,3,8]] 4

입출력 예 설명

costs를 그림으로 표현하면 다음과 같으며, 이때 초록색 경로로 연결하는 것이 가장 적은 비용으로 모두를 통행할 수 있도록 만드는 방법입니다.



**단속카메라**

문제 설명

고속도로를 이동하는 모든 차량이 고속도로를 이용하면서 단속용 카메라를 한 번은 만나도록 카메라를 설치하려고 합니다.

고속도로를 이동하는 차량의 경로 routes가 매개변수로 주어질 때, 모든 차량이 한 번은 단속용 카메라를 만나도록 하려면 최소 몇 대의 카메라를 설치해야 하는지를 return 하도록 solution 함수를 완성하세요.

제한사항

차량의 대수는 1대 이상 10,000대 이하입니다.

routes에는 차량의 이동 경로가 포함되어 있으며 routes[i][0]에는 i번째 차량이 고속도로에 진입한 지점, routes[i][1]에는 i번째 차량이 고속도로에서 나간 지점이 적혀 있습니다.

차량의 진입/진출 지점에 카메라가 설치되어 있어도 카메라를 만난것으로 간주합니다.

차량의 진입 지점, 진출 지점은 -30,000 이상 30,000 이하입니다.

입출력 예

routes return

[[-20,15], [-14,-5], [-18,-13], [-5,-3]] 2

입출력 예 설명

-5 지점에 카메라를 설치하면 두 번째, 네 번째 차량이 카메라를 만납니다.

-15 지점에 카메라를 설치하면 첫 번째, 세 번째 차량이 카메라를 만납니다.

**구명보트**

문제 설명

무인도에 갇힌 사람들을 구명보트를 이용하여 구출하려고 합니다. 구명보트는 작아서 한 번에 최대 2명씩 밖에 탈 수 없고, 무게 제한도 있습니다.

예를 들어, 사람들의 몸무게가 [70kg, 50kg, 80kg, 50kg]이고 구명보트의 무게 제한이 100kg이라면 2번째 사람과 4번째 사람은 같이 탈 수 있지만 1번째 사람과 3번째 사람의 무게의 합은 150kg이므로 구명보트의 무게 제한을 초과하여 같이 탈 수 없습니다.

구명보트를 최대한 적게 사용하여 모든 사람을 구출하려고 합니다.

사람들의 몸무게를 담은 배열 people과 구명보트의 무게 제한 limit가 매개변수로 주어질 때, 모든 사람을 구출하기 위해 필요한 구명보트 개수의 최솟값을 return 하도록 solution 함수를 작성해주세요.

제한사항

무인도에 갇힌 사람은 1명 이상 50,000명 이하입니다.

각 사람의 몸무게는 40kg 이상 240kg 이하입니다.

구명보트의 무게 제한은 40kg 이상 240kg 이하입니다.

구명보트의 무게 제한은 항상 사람들의 몸무게 중 최댓값보다 크게 주어지므로 사람들을 구출할 수 없는 경우는 없습니다.

입출력 예

people limit return

[70, 50, 80, 50] 100 3

[70, 80, 50] 100 3

**조이스틱**

문제 설명

조이스틱으로 알파벳 이름을 완성하세요. 맨 처음엔 A로만 이루어져 있습니다.

ex) 완성해야 하는 이름이 세 글자면 AAA, 네 글자면 AAAA

조이스틱을 각 방향으로 움직이면 아래와 같습니다.

▲ - 다음 알파벳

▼ - 이전 알파벳 (A에서 아래쪽으로 이동하면 Z로)

◀ - 커서를 왼쪽으로 이동 (첫 번째 위치에서 왼쪽으로 이동하면 마지막 문자에 커서)

▶ - 커서를 오른쪽으로 이동

예를 들어 아래의 방법으로 JAZ를 만들 수 있습니다.

- 첫 번째 위치에서 조이스틱을 위로 9번 조작하여 J를 완성합니다.

- 조이스틱을 왼쪽으로 1번 조작하여 커서를 마지막 문자 위치로 이동시킵니다.

- 마지막 위치에서 조이스틱을 아래로 1번 조작하여 Z를 완성합니다.

따라서 11번 이동시켜 "JAZ"를 만들 수 있고, 이때가 최소 이동입니다.

만들고자 하는 이름 name이 매개변수로 주어질 때, 이름에 대해 조이스틱 조작 횟수의 최솟값을 return 하도록 solution 함수를 만드세요.

제한 사항

name은 알파벳 대문자로만 이루어져 있습니다.

name의 길이는 1 이상 20 이하입니다.

입출력 예

name return

JEROEN 56

JAN 23

**큰 수 만들기**

문제 설명

어떤 숫자에서 k개의 수를 제거했을 때 얻을 수 있는 가장 큰 숫자를 구하려 합니다.

예를 들어, 숫자 1924에서 수 두 개를 제거하면 [19, 12, 14, 92, 94, 24] 를 만들 수 있습니다. 이 중 가장 큰 숫자는 94 입니다.

문자열 형식으로 숫자 number와 제거할 수의 개수 k가 solution 함수의 매개변수로 주어집니다. number에서 k 개의 수를 제거했을 때 만들 수 있는 수 중 가장 큰 숫자를 문자열 형태로 return 하도록 solution 함수를 완성하세요.

제한 조건

number는 1자리 이상, 1,000,000자리 이하인 숫자입니다.

k는 1 이상 number의 자릿수 미만인 자연수입니다.

입출력 예

number k return

1924 2 94

1231234 3 3234

4177252841 4 775841