

16. 콜라츠 추측 - level 1

URL <https://school.programmers.co.kr/learn/courses/30/lessons/12943>

문제 설명

1937년 Collatz란 사람에 의해 제기된 이 추측은, 주어진 수가 1이 될 때까지 다음 작업을 반복하면, 모든 수를 1로 만들 수 있다는 추측입니다. 작업은 다음과 같습니다.

- 1-1. 입력된 수가 짝수라면 2로 나눕니다.
- 1-2. 입력된 수가 홀수라면 3을 곱하고 1을 더합니다.
- 2. 결과로 나온 수에 같은 작업을 1이 될 때까지 반복합니다.

예를 들어, 주어진 수가 6이라면 $6 \rightarrow 3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$ 이 되어 총 8번 만에 1이 됩니다. 위 작업을 몇 번이나 반복해야 하는지 반환하는 함수, solution을 완성해 주세요. 단, 주어진 수가 1인 경우에는 0을, 작업을 500번 반복할 때까지 1이 되지 않는다면 -1을 반환해 주세요.

제한 사항

입력된 수, num은 1 이상 8,000,000 미만인 정수입니다.

입출력 예

n	result
6	8
16	4
626331	-1

입출력 예 설명

입출력 예 #1

문제의 설명과 같습니다.

입출력 예 #2

$16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$ 이 되어 총 4번 만에 1이 됩니다.

입출력 예 #3

626331은 500번을 시도해도 1이 되지 못하므로 -1을 리턴해야 합니다.

17. 하노이의 탑 - level 3

URL <https://school.programmers.co.kr/learn/courses/30/lessons/12946>

문제 설명

하노이 탑(Tower of Hanoi)은 퍼즐의 일종입니다. 세 개의 기둥과 이 기둥에 꽂을 수 있는 크기가 다양한 원판들이 있고, 퍼즐을 시작하기 전에는 한 기둥에 원판들이 작은 것이 위에 있도록 순서대로 쌓여 있습니다. 게임의 목적은 다음 두 가지 조건을 만족시키면서, 한 기둥에 꽂힌 원판들을 그 순서 그대로 다른 기둥으로 옮겨서 다시 쌓는 것입니다.

한 번에 하나의 원판만 옮길 수 있습니다.

큰 원판이 작은 원판 위에 있어서는 안됩니다.

하노이 탑의 세 개의 기둥을 왼쪽 부터 1번, 2번, 3번이라고 하겠습니다. 1번에는 n 개의 원판이 있고 이 n 개의 원판을 3번 원판으로 최소 횟수로 옮기려고 합니다.

1번 기둥에 있는 원판의 개수 n 이 매개변수로 주어질 때, n 개의 원판을 3번 원판으로 최소로 옮기는 방법을 return하는 solution를 완성해주세요.

제한사항

n 은 15이하의 자연수입니다.

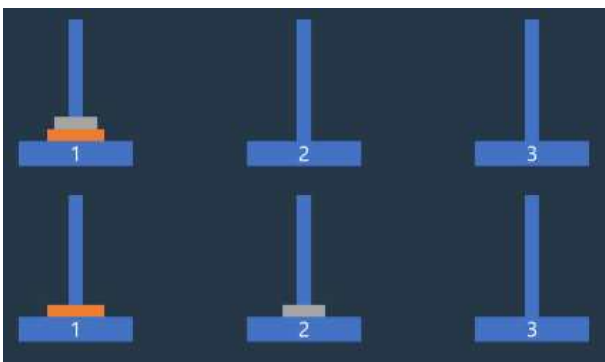
입출력 예

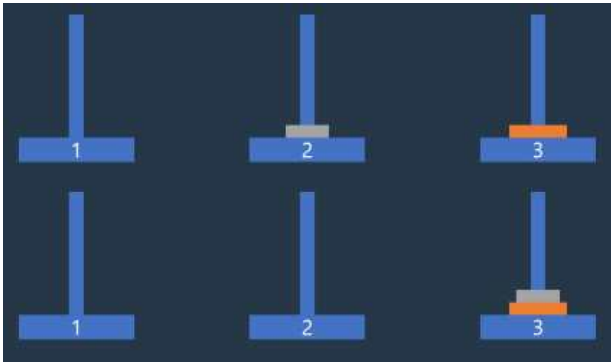
n	result
2	[[1,2], [1,3], [2,3]]

입출력 예 설명

입출력 예 #1

다음과 같이 옮길 수 있습니다.





18. 모음 사전 - level 2

URL <https://school.programmers.co.kr/learn/courses/30/lessons/84512>

문제 설명

사전에 알파벳 모음 'A', 'E', 'I', 'O', 'U'만을 사용하여 만들 수 있는, 길이 5 이하의 모든 단어가 수록되어 있습니다. 사전에서 첫 번째 단어는 "A"이고, 그다음은 "AA"이며, 마지막 단어는 "UUUUU"입니다.

단어 하나 word가 매개변수로 주어질 때, 이 단어가 사전에서 몇 번째 단어인지 return 하도록 solution 함수를 완성해주세요.

제한 사항

word의 길이는 1 이상 5 이하입니다.

word는 알파벳 대문자 'A', 'E', 'I', 'O', 'U'로만 이루어져 있습니다.

입출력 예

word	result
"AAAAE"	6
"AAAE"	10
"I"	1563
"EIO"	1189

입출력 예 설명

입출력 예 #1

사전에서 첫 번째 단어는 "A"이고, 그다음은 "AA", "AAA", "AAAA", "AAAAA", "AAAAE", ... 와 같습니다. "AAAAE"는 사전에서 6번째 단어입니다.

입출력 예 #2

"AAAE"는 "A", "AA", "AAA", "AAAA", "AAAAA", "AAAAE", "AAAAI", "AAAAO", "AAAAU"의 다음인 10 번째 단어입니다.

입출력 예 #3

"I"는 1563번째 단어입니다.

입출력 예 #4

"EIO"는 1189번째 단어입니다.

19. 호텔 방 배정 - level 4

URL <https://school.programmers.co.kr/learn/courses/30/lessons/64063>

문제 설명

[본 문제는 정확성과 효율성 테스트 각각 점수가 있는 문제입니다.]

"스노우타운"에서 호텔을 운영하고 있는 "스카피"는 호텔에 투숙하려는 고객들에게 방을 배정하려 합니다. 호텔에는 방이 총 k 개 있으며, 각각의 방은 1번부터 k 번까지 번호로 구분하고 있습니다. 처음에는 모든 방이 비어 있으며 "스카피"는 다음과 같은 규칙에 따라 고객에게 방을 배정하려고 합니다.

1. 한 번에 한 명씩 신청한 순서대로 방을 배정합니다.
2. 고객은 투숙하기 원하는 방 번호를 제출합니다.
3. 고객이 원하는 방이 비어 있다면 즉시 배정합니다.
4. 고객이 원하는 방이 이미 배정되어 있으면 원하는 방보다 번호가 크면서 비어있는 방 중 가장 번호가 작은 방을 배정합니다.

예를 들어, 방이 총 10개이고, 고객들이 원하는 방 번호가 순서대로 [1, 3, 4, 1, 3, 1] 일 경우 다음과 같이 방을 배정받게 됩니다.

원하는 방 번호	배정된 방 번호
1	1
3	3
4	4
1	2
3	5
1	6

전체 방 개수 k 와 고객들이 원하는 방 번호가 순서대로 들어있는 배열 `room_number`가 매개변수로 주어질 때, 각 고객에게 배정되는 방 번호를 순서대로 배열에 담아 `return` 하도록 `solution` 함수를 완성해주세요.

[제한사항]

- k 는 1 이상 1012 이하인 자연수입니다.
- `room_number` 배열의 크기는 1 이상 200,000 이하입니다.
- `room_number` 배열 각 원소들의 값은 1 이상 k 이하인 자연수입니다.
- `room_number` 배열은 모든 고객이 방을 배정받을 수 있는 경우만 입력으로 주어집니다.
예를 들어, $k = 5$, `room_number = [5, 5]` 와 같은 경우는 방을 배정받지 못하는 고객이 발생하므로 이런 경우는 입력으로 주어지지 않습니다.

[입출력 예]

k	room_number	result
10	[1,3,4,1,3,1]	[1,3,4,2,5,6]

입출력 예에 대한 설명

입출력 예 #1

문제의 예시와 같습니다.

첫 번째 ~ 세 번째 고객까지는 원하는 방이 비어 있으므로 즉시 배정받을 수 있습니다. 네 번째 고객의 경우 1번 방을 배정받기를 원했는데, 1번 방은 빈 방이 아니므로, 1번 보다 번호가 크고 비어 있는 방 중에서 가장 번호가 작은 방을 배정해야 합니다. 1번 보다 번호가 크면서 비어있는 방은 [2번, 5번, 6번...] 방이며, 이중 가장 번호가 작은 방은 2번 방입니다. 따라서 네 번째 고객은 2번 방을 배정받습니다. 마찬가지로 5, 6번째 고객은 각각 5번, 6번 방을 배정받게 됩니다.

20. 모의고사 - level 1

URL <https://school.programmers.co.kr/learn/courses/30/lessons/42840>

문제 설명

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 `answers`가 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

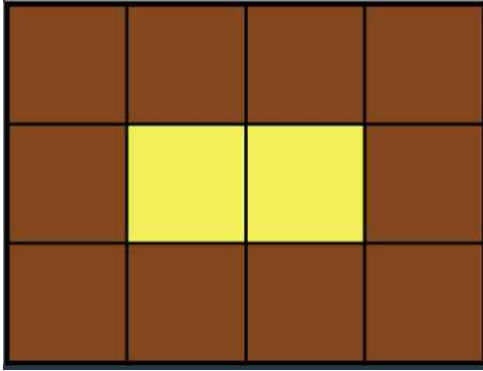
모든 사람이 2문제씩을 맞혔습니다.

21. 카펫 - level 2

URL <https://school.programmers.co.kr/learn/courses/30/lessons/42842>

문제 설명

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.

제한 사항

- 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]

22. 소수 찾기 - level 2

URL <https://school.programmers.co.kr/learn/courses/30/lessons/42839>

문제 설명

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.

제한 사항

numbers는 길이 1 이상 7 이하인 문자열입니다.

numbers는 0~9까지 숫자만으로 이루어져 있습니다.

"013"은 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.

23. 불량 사용자 - level 3

URL <https://school.programmers.co.kr/learn/courses/30/lessons/64064>

문제 설명

개발팀 내에서 이벤트 개발을 담당하고 있는 "무지"는 최근 진행된 카카오이모티콘 이벤트에 비정상적인 방법으로 당첨을 시도한 응모자들을 발견하였습니다. 이런 응모자들을 따로 모아 불량 사용자라는 이름으로 목록을 만들어서 당첨 처리 시 제외하도록 이벤트 당첨자 담당자인 "프로도" 에게 전달하려고 합니다. 이 때 개인정보 보호을 위해 사용자 아이디 중 일부 문자를 '*' 문자로 가려서 전달했습니다. 가리고자 하는 문자 하나에 '*' 문자 하나를 사용하였고 아이디 당 최소 하나 이상의 '*' 문자를 사용하였습니다. "무지"와 "프로도"는 불량 사용자 목록에 매핑된 응모자 아이디를 제재 아이디 라고 부르기로 하였습니다.

예를 들어, 이벤트에 응모한 전체 사용자 아이디 목록이 다음과 같다면

응모자 아이디
frodo
fradi
crodo
abc123
frodoc

다음과 같이 불량 사용자 아이디 목록이 전달된 경우,

불량 사용자
fr*d*
abc1**

불량 사용자에게 매핑되어 당첨에서 제외되어야 할 제재 아이디 목록은 다음과 같이 두 가지 경우가 있을 수 있습니다.

제재 아이디
frodo
abc123

제재 아이디
fradi
abc123

이벤트 응모자 아이디 목록이 담긴 배열 `user_id`와 불량 사용자 아이디 목록이 담긴 배열 `banned_id`가 매개변수로 주어질 때, 당첨에서 제외되어야 할 제재 아이디 목록은 몇 가지 경우의 수가 가능한지 return 하도록 solution 함수를 완성해주세요.

[제한 사항]

- `user_id` 배열의 크기는 1 이상 8 이하입니다.
- `user_id` 배열 각 원소들의 값은 길이가 1 이상 8 이하인 문자열입니다.
 - 응모한 사용자 아이디들은 서로 중복되지 않습니다.
 - 응모한 사용자 아이디는 알파벳 소문자와 숫자로만 구성되어 있습니다.
- `banned_id` 배열의 크기는 1 이상 `user_id` 배열의 크기 이하입니다.

- banned_id 배열 각 원소들의 값은 길이가 1 이상 8 이하인 문자열입니다.
불량 사용자 아이디는 알파벳 소문자와 숫자, 가리기 위한 문자 '*' 로만 이루어져 있습니다.
불량 사용자 아이디는 '*' 문자를 하나 이상 포함하고 있습니다.
불량 사용자 아이디 하나는 응모자 아이디 중 하나에 해당하고 같은 응모자 아이디가 중복해서 제재 아이디 목록에 들어가는 경우는 없습니다.
- 제재 아이디 목록들을 구했을 때 아이디들이 나열된 순서와 관계없이 아이디 목록의 내용이 동일하다면 같은 것으로 처리하여 하나로 세면 됩니다.

[입출력 예]

user_id	banned_id	result
["frodo", "fradi", "crodo", "abc123", "frodoc"]	["fr*d*", "abc1**"]	2
["frodo", "fradi", "crodo", "abc123", "frodoc"]	["*rodo", "*rodo", "*****"]	2
["frodo", "fradi", "crodo", "abc123", "frodoc"]	["fr*d*", "*rodo", "*****", "*****"]	3

입출력 예에 대한 설명

입출력 예 #1

문제 설명과 같습니다.

입출력 예 #2

다음과 같이 두 가지 경우가 있습니다.

제재 아이디
frodo
crodo
abc123

제재 아이디
frodo
crodo
frodoc

입출력 예 #3

다음과 같이 세 가지 경우가 있습니다.

제재 아이디
frodo
crodo
abc123
frodoc

제재 아이디
fradi
crodo
abc123
frodoc

제재 아이디
fradi
frodo
abc123
frodoc

24. 수식 최대화 - level 2

URL <https://school.programmers.co.kr/learn/courses/30/lessons/67257>

문제 설명

IT 벤처 회사를 운영하고 있는 라이언은 매년 사내 해커톤 대회를 개최하여 우승자에게 상금을 지급하고 있습니다.

이번 대회에서는 우승자에게 지급되는 상금을 이전 대회와는 다르게 다음과 같은 방식으로 결정하려고 합니다.

해커톤 대회에 참가하는 모든 참가자들에게는 숫자들과 3가지의 연산문자(+, -, *) 만으로 이루어진 연산 수식이 전달되며, 참가자의 미션은 전달받은 수식에 포함된 연산자의 우선순위를 자유롭게 재정의하여 만들 수 있는 가장 큰 숫자를 제출하는 것입니다.

단, 연산자의 우선순위를 새로 정의할 때, 같은 순위의 연산자는 없어야 합니다. 즉, $+ > - > *$ 또는 $- > * > +$ 등과 같이 연산자 우선순위를 정의할 수 있으나 $+, * > -$ 또는 $* > +, -$ 처럼 2개 이상의 연산자가 동일한 순위를 가지도록 연산자 우선순위를 정의할 수는 없습니다. 수식에 포함된 연산자가 2개라면 정의할 수 있는 연산자 우선순위 조합은 $2! = 2$ 가지이며, 연산자가 3개라면 $3! = 6$ 가지 조합이 가능합니다.

만약 계산된 결과가 음수라면 해당 숫자의 절댓값으로 변환하여 제출하며 제출한 숫자가 가장 큰 참가자를 우승자로 선정하며, 우승자가 제출한 숫자를 우승상금으로 지급하게 됩니다.

예를 들어, 참가자 중 네오가 아래와 같은 수식을 전달받았다고 가정합니다.

"100-200*300-500+20"

일반적으로 수학 및 전산학에서 약속된 연산자 우선순위에 따르면 더하기와 빼기는 서로 동등하며 곱하기는 더하기, 빼기에 비해 우선순위가 높아 $* > +, -$ 로 우선순위가 정의되어 있습니다.

대회 규칙에 따라 $+ > - > *$ 또는 $- > * > +$ 등과 같이 연산자 우선순위를 정의할 수 있으나 $+, * > -$ 또는 $* > +, -$ 처럼 2개 이상의 연산자가 동일한 순위를 가지도록 연산자 우선순위를 정의할 수는 없습니다.

수식에 연산자가 3개 주어졌으므로 가능한 연산자 우선순위 조합은 $3! = 6$ 가지이며, 그 중 $+ > - > *$ 로 연산자 우선순위를 정한다면 결과값은 22,000원이 됩니다.

반면에 $* > + > -$ 로 연산자 우선순위를 정한다면 수식의 결과값은 -60,420 이지만, 규칙에 따라 우승 시 상금은 절댓값인 60,420원이 됩니다.

참가자에게 주어진 연산 수식이 담긴 문자열 expression이 매개변수로 주어질 때, 우승 시 받을 수 있는 가장 큰 상금 금액을 return 하도록 solution 함수를 완성해주세요.

[제한 사항]

- expression은 길이가 3 이상 100 이하인 문자열입니다.
- expression은 공백문자, 괄호문자 없이 오로지 숫자와 3가지의 연산자(+, -, *) 만으로 이루어진 올바른 중위표기법(연산의 두 대상 사이에 연산기호를 사용하는 방식)으로 표현된 연산식입니다. 잘못된 연산식은 입력으로 주어지지 않습니다.

즉, "402+-561*"처럼 잘못된 수식은 올바른 중위표기법이 아니므로 주어지지 않습니다.

- expression의 피연산자(operand)는 0 이상 999 이하의 숫자입니다.
즉, "100-2145*458+12"처럼 999를 초과하는 피연산자가 포함된 수식은 입력으로 주어지지 않습니다.
"-56+100"처럼 피연산자가 음수인 수식도 입력으로 주어지지 않습니다.
- expression은 적어도 1개 이상의 연산자를 포함하고 있습니다.
- 연산자 우선순위를 어떻게 적용하더라도, expression의 중간 계산값과 최종 결과값은 절댓값이 263 - 1 이하가 되도록 입력이 주어집니다.
- 같은 연산자끼리는 앞에 있는 것의 우선순위가 더 높습니다.

입출력 예

expression	result
"100-200*300-500+20"	60420
"50*6-3*2"	300

입출력 예에 대한 설명

입출력 예 #1

* > + > - 로 연산자 우선순위를 정했을 때, 가장 큰 절댓값을 얻을 수 있습니다.

연산 순서는 아래와 같습니다.

$$\begin{aligned}
 &100-200*300-500+20 \\
 &= 100-(200*300)-500+20 \\
 &= 100-60000-(500+20) \\
 &= (100-60000)-520 \\
 &= (-59900-520) \\
 &= -60420
 \end{aligned}$$

따라서, 우승 시 받을 수 있는 상금은 $|-60420| = 60420$ 입니다.

입출력 예 #2

- > * 로 연산자 우선순위를 정했을 때, 가장 큰 절댓값을 얻을 수 있습니다.

연산 순서는 아래와 같습니다.

(expression에서 + 연산자는 나타나지 않았으므로, 고려할 필요가 없습니다.)

$$\begin{aligned}
 &50*6-3*2 \\
 &= 50*(6-3)*2 \\
 &= (50*3)*2 \\
 &= 150*2 \\
 &= 300
 \end{aligned}$$

따라서, 우승 시 받을 수 있는 상금은 300입니다.

25. 두 개 뽑아서 더하기 - level 1

URL <https://school.programmers.co.kr/learn/courses/30/lessons/68644>

문제 설명

정수 배열 numbers가 주어집니다. numbers에서 서로 다른 인덱스에 있는 두 개의 수를 뽑아 더해서 만들 수 있는 모든 수를 배열에 오름차순으로 담아 return 하도록 solution 함수를 완성해주세요.

제한 사항

numbers의 길이는 2 이상 100 이하입니다.

numbers의 모든 수는 0 이상 100 이하입니다.

입출력 예

numbers	result
[2,1,3,4,1]	[2,3,4,5,6,7]
[5,0,2,7]	[2,5,7,9,12]

입출력 예 설명

입출력 예 #1

2 = 1 + 1 입니다. (1이 numbers에 두 개 있습니다.)

3 = 2 + 1 입니다.

4 = 1 + 3 입니다.

5 = 1 + 4 = 2 + 3 입니다.

6 = 2 + 4 입니다.

7 = 3 + 4 입니다.

따라서 [2,3,4,5,6,7] 을 return 해야 합니다.

입출력 예 #2

2 = 0 + 2 입니다.

5 = 5 + 0 입니다.

7 = 0 + 7 = 5 + 2 입니다.

9 = 2 + 7 입니다.

12 = 5 + 7 입니다.

따라서 [2,5,7,9,12] 를 return 해야 합니다.

26. H-index - level 2

URL <https://school.programmers.co.kr/learn/courses/30/lessons/42747>

문제 설명

H-Index는 과학자의 생산성과 영향력을 나타내는 지표입니다. 어느 과학자의 H-Index를 나타내는 값인 h 를 구하려고 합니다. 위키백과1에 따르면, H-Index는 다음과 같이 구합니다.

어떤 과학자가 발표한 논문 n 편 중, h 번 이상 인용된 논문이 h 편 이상이고 나머지 논문이 h 번 이하 인용되었다면 h 의 최댓값이 이 과학자의 H-Index입니다.

어떤 과학자가 발표한 논문의 인용 횟수를 담은 배열 `citations`가 매개변수로 주어질 때, 이 과학자의 H-Index를 `return` 하도록 `solution` 함수를 작성해주세요.

제한 사항

과학자가 발표한 논문의 수는 1편 이상 1,000편 이하입니다.

논문별 인용 횟수는 0회 이상 10,000회 이하입니다.

입출력 예

citations	return
[3, 0, 6, 1, 5]	3

입출력 예 설명

이 과학자가 발표한 논문의 수는 5편이고, 그중 3편의 논문은 3회 이상 인용되었습니다. 그리고 나머지 2편의 논문은 3회 이하 인용되었기 때문에 이 과학자의 H-Index는 3입니다.

27. 문자열 내 마음대로 정렬하기 - level 1

URL <https://school.programmers.co.kr/learn/courses/30/lessons/12915>

문제 설명

문자열로 구성된 리스트 strings와, 정수 n이 주어졌을 때, 각 문자열의 인덱스 n번째 글자를 기준으로 오름차순 정렬하려 합니다. 예를 들어 strings가 ["sun", "bed", "car"]이고 n이 1이면 각 단어의 인덱스 1의 문자 "u", "e", "a"로 strings를 정렬합니다.

제한 조건

strings는 길이 1 이상, 50이하인 배열입니다.

strings의 원소는 소문자 알파벳으로 이루어져 있습니다.

strings의 원소는 길이 1 이상, 100이하인 문자열입니다.

모든 strings의 원소의 길이는 n보다 큼니다.

인덱스 1의 문자가 같은 문자열이 여럿 일 경우, 사전순으로 앞선 문자열이 앞쪽에 위치합니다.

입출력 예

strings	n	return
["sun", "bed", "car"]	1	["car", "bed", "sun"]
["abce", "abcd", "cdx"]	2	["abcd", "abce", "cdx"]

입출력 예 설명

입출력 예 1

"sun", "bed", "car"의 1번째 인덱스 값은 각각 "u", "e", "a" 입니다. 이를 기준으로 strings를 정렬하면 ["car", "bed", "sun"] 입니다.

입출력 예 2

"abce"와 "abcd", "cdx"의 2번째 인덱스 값은 "c", "c", "x"입니다. 따라서 정렬 후에는 "cdx"가 가장 뒤에 위치합니다. "abce"와 "abcd"는 사전순으로 정렬하면 "abcd"가 우선하므로, 답은 ["abcd", "abce", "cdx"] 입니다.

28. 가장 큰 수 - level 2

URL <https://school.programmers.co.kr/learn/courses/30/lessons/42746>

문제 설명

0 또는 양의 정수가 주어졌을 때, 정수를 이어 붙여 만들 수 있는 가장 큰 수를 알아내 주세요.

예를 들어, 주어진 정수가 [6, 10, 2]라면 [6102, 6210, 1062, 1026, 2610, 2106]를 만들 수 있고, 이중 가장 큰 수는 6210입니다.

0 또는 양의 정수가 담긴 배열 numbers가 매개변수로 주어질 때, 순서를 재배치하여 만들 수 있는 가장 큰 수를 문자열로 바꾸어 return 하도록 solution 함수를 작성해주세요.

제한 사항

numbers의 길이는 1 이상 100,000 이하입니다.

numbers의 원소는 0 이상 1,000 이하입니다.

정답이 너무 클 수 있으니 문자열로 바꾸어 return 합니다.

입출력 예

numbers	return
[6, 10, 2]	"6210"
[3, 30, 34, 5, 9]	"9534330"

29. 입국 심사 - level 3

URL <https://school.programmers.co.kr/learn/courses/30/lessons/43238>

문제 설명

n명이 입국심사를 위해 줄을 서서 기다리고 있습니다. 각 입국심사대에 있는 심사관마다 심사하는데 걸리는 시간은 다릅니다.

처음에 모든 심사대는 비어있습니다. 한 심사대에서는 동시에 한 명만 심사를 할 수 있습니다. 가장 앞에 서 있는 사람은 비어 있는 심사대로 가서 심사를 받을 수 있습니다. 하지만 더 빨리 끝나는 심사대가 있으면 기다렸다가 그곳으로 가서 심사를 받을 수도 있습니다.

모든 사람이 심사를 받는데 걸리는 시간을 최소로 하고 싶습니다.

입국심사를 기다리는 사람 수 n, 각 심사관이 한 명을 심사하는데 걸리는 시간이 담긴 배열 times가 매개 변수로 주어질 때, 모든 사람이 심사를 받는데 걸리는 시간의 최솟값을 return 하도록 solution 함수를 작성해주세요.

제한 사항

입국심사를 기다리는 사람은 1명 이상 1,000,000,000명 이하입니다.

각 심사관이 한 명을 심사하는데 걸리는 시간은 1분 이상 1,000,000,000분 이하입니다.

심사관은 1명 이상 100,000명 이하입니다.

입출력 예

n	times	return
6	[7, 10]	28

입출력 예 설명

가장 첫 두 사람은 바로 심사를 받으러 갑니다.

7분이 되었을 때, 첫 번째 심사대가 비고 3번째 사람이 심사를 받습니다.

10분이 되었을 때, 두 번째 심사대가 비고 4번째 사람이 심사를 받습니다.

14분이 되었을 때, 첫 번째 심사대가 비고 5번째 사람이 심사를 받습니다.

20분이 되었을 때, 두 번째 심사대가 비지만 6번째 사람이 그곳에서 심사를 받지 않고 1분을 더 기다린 후에 첫 번째 심사대에서 심사를 받으면 28분에 모든 사람의 심사가 끝납니다.

30. 순위 검색 level 2

URL <https://school.programmers.co.kr/learn/courses/30/lessons/72412>

문제 설명

[본 문제는 정확성과 효율성 테스트 각각 점수가 있는 문제입니다.]

카카오는 하반기 경력 개발자 공개채용을 진행 중에 있으며 현재 지원서 접수와 코딩테스트가 종료되었습니다. 이번 채용에서 지원자는 지원서 작성 시 아래와 같이 4가지 항목을 반드시 선택하도록 하였습니다.

- 코딩테스트 참여 개발언어 항목에 cpp, java, python 중 하나를 선택해야 합니다.
- 지원 직군 항목에 backend와 frontend 중 하나를 선택해야 합니다.
- 지원 경력구분 항목에 junior와 senior 중 하나를 선택해야 합니다.
- 선호하는 소울푸드로 chicken과 pizza 중 하나를 선택해야 합니다.

인재영입팀에 근무하고 있는 니니즈는 코딩테스트 결과를 분석하여 채용에 참여한 개발팀들에 제공하기 위해 지원자들의 지원 조건을 선택하면 해당 조건에 맞는 지원자가 몇 명인 지 쉽게 알 수 있는 도구를 만들고 있습니다.

예를 들어, 개발팀에서 궁금해하는 문의사항은 다음과 같은 형태가 될 수 있습니다.

코딩테스트에 java로 참여했으며, backend 직군을 선택했고, junior 경력이면서, 소울푸드로 pizza를 선택한 사람 중 코딩테스트 점수를 50점 이상 받은 지원자는 몇 명인가?

물론 이 외에도 각 개발팀의 상황에 따라 아래와 같이 다양한 형태의 문의가 있을 수 있습니다.

- 코딩테스트에 python으로 참여했으며, frontend 직군을 선택했고, senior 경력이면서, 소울푸드로 chicken을 선택한 사람 중 코딩테스트 점수를 100점 이상 받은 사람은 모두 몇 명인가?
- 코딩테스트에 cpp로 참여했으며, senior 경력이면서, 소울푸드로 pizza를 선택한 사람 중 코딩테스트 점수를 100점 이상 받은 사람은 모두 몇 명인가?
- backend 직군을 선택했고, senior 경력이면서 코딩테스트 점수를 200점 이상 받은 사람은 모두 몇 명인가?
- 소울푸드로 chicken을 선택한 사람 중 코딩테스트 점수를 250점 이상 받은 사람은 모두 몇 명인가?
- 코딩테스트 점수를 150점 이상 받은 사람은 모두 몇 명인가?

즉, 개발팀에서 궁금해하는 내용은 다음과 같은 형태를 갖습니다.

* [조건]을 만족하는 사람 중 코딩테스트 점수를 X점 이상 받은 사람은 모두 몇 명인가?

[문제]

지원자가 지원서에 입력한 4가지의 정보와 획득한 코딩테스트 점수를 하나의 문자열로 구성한 값의 배열 info, 개발팀이 궁금해하는 문의조건이 문자열 형태로 담긴 배열 query가 매개변수로 주어질 때, 각 문의 조건에 해당하는 사람들의 숫자를 순서대로 배열에 담아 return 하도록 solution 함수를 완성해주세요.

[제한 사항]

- info 배열의 크기는 1 이상 50,000 이하입니다.
- info 배열 각 원소의 값은 지원자가 지원서에 입력한 4가지 값과 코딩테스트 점수를 합친 "개발언어 직군 경력 소울푸드 점수" 형식입니다.
 - 개발언어는 cpp, java, python 중 하나입니다.
 - 직군은 backend, frontend 중 하나입니다.
 - 경력은 junior, senior 중 하나입니다.
 - 소울푸드는 chicken, pizza 중 하나입니다.
 - 점수는 코딩테스트 점수를 의미하며, 1 이상 100,000 이하인 자연수입니다.
 - 각 단어는 공백문자(스페이스 바) 하나로 구분되어 있습니다.
- query 배열의 크기는 1 이상 100,000 이하입니다.
- query의 각 문자열은 "[조건] X" 형식입니다.
 - [조건]은 "개발언어 and 직군 and 경력 and 소울푸드" 형식의 문자열입니다.
 - 언어는 cpp, java, python, - 중 하나입니다.
 - 직군은 backend, frontend, - 중 하나입니다.
 - 경력은 junior, senior, - 중 하나입니다.
 - 소울푸드는 chicken, pizza, - 중 하나입니다.
 - '-' 표시는 해당 조건을 고려하지 않겠다는 의미입니다.
 - X는 코딩테스트 점수를 의미하며 조건을 만족하는 사람 중 X점 이상 받은 사람은 모두 몇 명인지를 의미합니다.
 - 각 단어는 공백문자(스페이스 바) 하나로 구분되어 있습니다.
 - 예를 들면, "cpp and - and senior and pizza 500"은 "cpp로 코딩테스트를 봤으며, 경력은 senior 이면서 소울푸드로 pizza를 선택한 지원자 중 코딩테스트 점수를 500점 이상 받은 사람은 모두 몇 명인가?"를 의미합니다.

[입출력 예]

info	query	result
["java backend junior pizza 150","python frontend senior chicken 210","python frontend senior chicken 150","cpp backend senior pizza 260","java backend junior chicken 80","python backend senior chicken 50"]	["java and backend and junior and pizza 100","python and frontend and senior and chicken 200","cpp and - and senior and pizza 250","- and backend and senior and - 150","- and - and - and chicken 100","- and - and - and - 150"]	[1,1,1,1,2,4]

입출력 예에 대한 설명

지원자 정보를 표로 나타내면 다음과 같습니다.

언어	직군	경력	소울 푸드	점수
java	backend	junior	pizza	150
python	frontend	senior	chicken	210
python	frontend	senior	chicken	150
cpp	backend	senior	pizza	260
java	backend	junior	chicken	80
python	backend	senior	chicken	50

- "java and backend and junior and pizza 100" : java로 코딩테스트를 봤으며, backend 직군을 선택했고 junior 경력이면서 소울푸드로 pizza를 선택한 지원자 중 코딩테스트 점수를 100점 이상 받은 지원자는 1명 입니다.
- "python and frontend and senior and chicken 200" : python으로 코딩테스트를 봤으며, frontend 직군을 선택했고, senior 경력이면서 소울 푸드로 chicken을 선택한 지원자 중 코딩테스트 점수를 200점 이상 받은 지원자는 1명 입니다.
- "cpp and - and senior and pizza 250" : cpp로 코딩테스트를 봤으며, senior 경력이면서 소울푸드로 pizza를 선택한 지원자 중 코딩테스트 점수를 250점 이상 받은 지원자는 1명 입니다.
- "- and backend and senior and - 150" : backend 직군을 선택했고, senior 경력인 지원자 중 코딩테스트 점수를 150점 이상 받은 지원자는 1명 입니다.
- "- and - and - and chicken 100" : 소울푸드로 chicken을 선택한 지원자 중 코딩테스트 점수를 100점 이상을 받은 지원자는 2명 입니다.
- "- and - and - and - 150" : 코딩테스트 점수를 150점 이상 받은 지원자는 4명 입니다.