

31. 징검다리 - Level 4

URL <https://school.programmers.co.kr/learn/courses/30/lessons/43236>

문제 설명

출발지점부터 distance만큼 떨어진 곳에 도착지점이 있습니다. 그리고 그사이에는 바위들이 놓여있습니다. 바위 중 몇 개를 제거하려고 합니다.

예를 들어, 도착지점이 25만큼 떨어져 있고, 바위가 [2, 14, 11, 21, 17] 지점에 놓여있을 때 바위 2개를 제거하면 출발지점, 도착지점, 바위 간의 거리가 아래와 같습니다.

제거한 바위의 위치	각 바위 사이의 거리	거리의 최솟값
[21, 17]	[2, 9, 3, 11]	2
[2, 21]	[11, 3, 3, 8]	3
[2, 11]	[14, 3, 4, 4]	3
[11, 21]	[2, 12, 3, 8]	2
[2, 14]	[11, 6, 4, 4]	4

위에서 구한 거리의 최솟값 중에 가장 큰 값은 4입니다.

출발지점부터 도착지점까지의 거리 distance, 바위들이 있는 위치를 담은 배열 rocks, 제거할 바위의 수 n이 매개변수로 주어질 때, 바위를 n개 제거한 뒤 각 지점 사이의 거리의 최솟값 중에 가장 큰 값을 return 하도록 solution 함수를 작성해주세요.

제한 사항

도착지점까지의 거리 distance는 1 이상 1,000,000,000 이하입니다.

바위는 1개 이상 50,000개 이하가 있습니다.

n 은 1 이상 바위의 개수 이하입니다.

입출력 예

distance	rocks	n	return
25	[2, 14, 11, 21, 17]	2	4

입출력 예 설명

문제에 나온 예와 같습니다.

32. 징검다리 건너기 - Level 3

URL <https://school.programmers.co.kr/learn/courses/30/lessons/64062>

문제 설명

[본 문제는 정확성과 효율성 테스트 각각 점수가 있는 문제입니다.]

카카오 초등학교의 "니니즈 친구들"이 "라이언" 선생님과 함께 가을 소풍을 가는 중에 징검다리가 있는 개울을 만나서 건너편으로 건너려고 합니다. "라이언" 선생님은 "니니즈 친구들"이 무사히 징검다리를 건널 수 있도록 다음과 같이 규칙을 만들었습니다.

- 징검다리는 일렬로 놓여있고 각 징검다리의 디딤돌에는 모두 숫자가 적혀 있으며 디딤돌의 숫자는 한 번 밟을 때마다 1씩 줄어듭니다.
- 디딤돌의 숫자가 0이 되면 더 이상 밟을 수 없으며 이때는 그다음 디딤돌로 한 번에 여러 칸을 건너뛸 수 있습니다.
- 단, 다음으로 밟을 수 있는 디딤돌이 여러 개인 경우 무조건 가장 가까운 디딤돌로만 건너뛸 수 있습니다.

"니니즈 친구들"은 개울의 왼쪽에 있으며, 개울의 오른쪽 건너편에 도착해야 징검다리를 건넌 것으로 인정합니다.

"니니즈 친구들"은 한 번에 한 명씩 징검다리를 건너야 하며, 한 친구가 징검다리를 모두 건넌 후에 그 다음 친구가 건너기 시작합니다.

디딤돌에 적힌 숫자가 순서대로 담긴 배열 stones와 한 번에 건너뛸 수 있는 디딤돌의 최대 칸수 k가 매 개변수로 주어질 때, 최대 몇 명까지 징검다리를 건널 수 있는지 return 하도록 solution 함수를 완성해주세요.

[제한사항]

- 징검다리를 건너야 하는 니니즈 친구들의 수는 무제한 이라고 간주합니다.
- stones 배열의 크기는 1 이상 200,000 이하입니다.
- stones 배열 각 원소들의 값은 1 이상 200,000,000 이하인 자연수입니다.
- k는 1 이상 stones의 길이 이하인 자연수입니다.

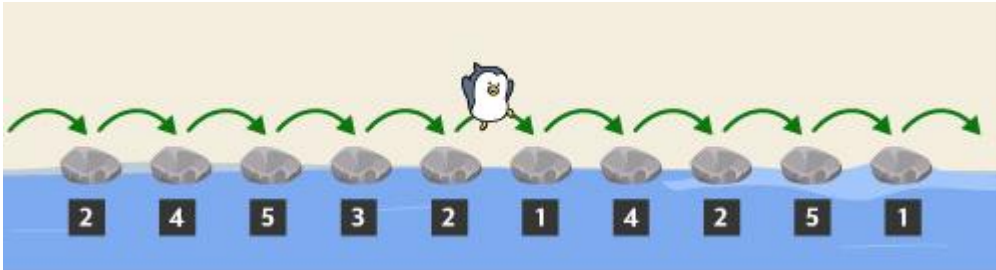
[입출력 예]

stones	k	result
[2, 4, 5, 3, 2, 1, 4, 2, 5, 1]	3	3

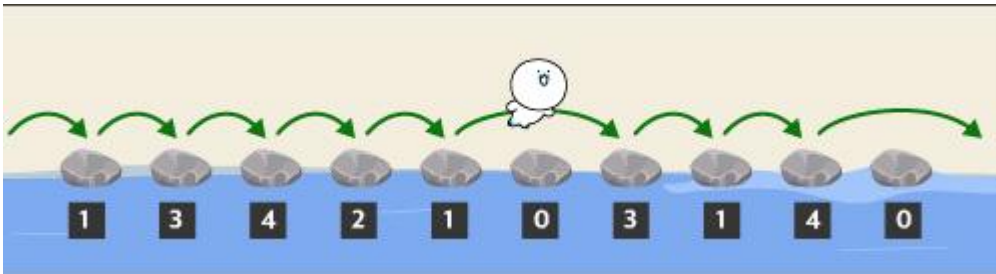
입출력 예에 대한 설명

입출력 예 #1

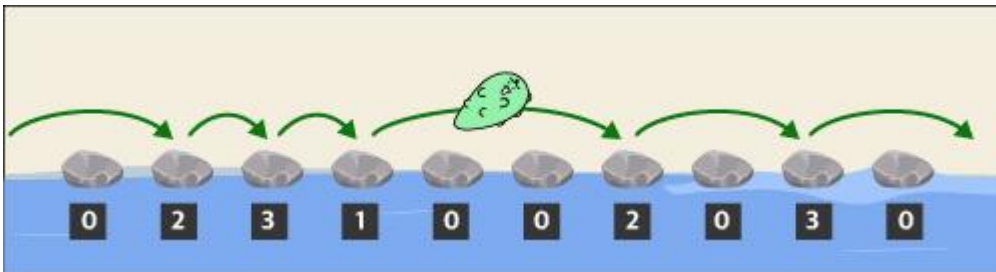
첫 번째 친구는 다음과 같이 징검다리를 건널 수 있습니다.



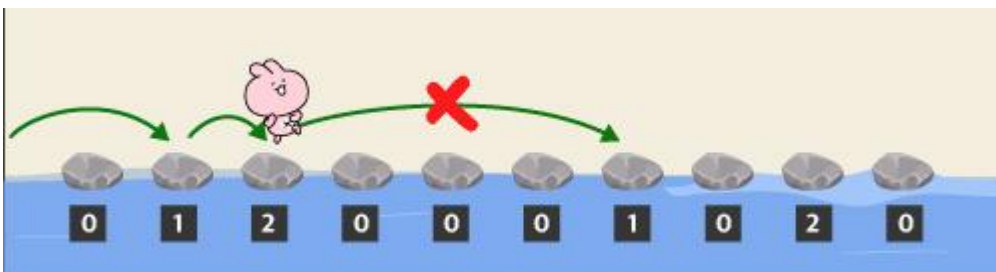
첫 번째 친구가 징검다리를 건넌 후 디딤돌에 적힌 숫자는 아래 그림과 같습니다.
두 번째 친구도 아래 그림과 같이 징검다리를 건널 수 있습니다.



두 번째 친구가 징검다리를 건넌 후 디딤돌에 적힌 숫자는 아래 그림과 같습니다.
세 번째 친구도 아래 그림과 같이 징검다리를 건널 수 있습니다.



세 번째 친구가 징검다리를 건넌 후 디딤돌에 적힌 숫자는 아래 그림과 같습니다.
네 번째 친구가 징검다리를 건너려면, 세 번째 디딤돌에서 일곱 번째 디딤돌로 네 칸을 건너뛰어야 합니다. 하지만 $k = 3$ 이므로 건너뛸 수 없습니다.



따라서 최대 3명이 디딤돌을 모두 건널 수 있습니다.

33. 완주하지 못한 선수 - Level 1

URL <https://programmers.co.kr/learn/courses/30/lessons/42576>

문제 설명

수많은 마라톤 선수들이 마라톤에 참여하였습니다. 단 한 명의 선수를 제외하고는 모든 선수가 마라톤을 완주하였습니다.

마라톤에 참여한 선수들의 이름이 담긴 배열 `participant`와 완주한 선수들의 이름이 담긴 배열 `completion`이 주어질 때, 완주하지 못한 선수의 이름을 `return` 하도록 `solution` 함수를 작성해주세요.

제한 사항

마라톤 경기에 참여한 선수의 수는 1명 이상 100,000명 이하입니다.

`completion`의 길이는 `participant`의 길이보다 1 작습니다.

참가자의 이름은 1개 이상 20개 이하의 알파벳 소문자로 이루어져 있습니다.

참가자 중에는 동명이인이 있을 수 있습니다.

입출력 예

participant	completion	return
["leo", "kiki", "eden"]	["eden", "kiki"]	"leo"
["marina", "josipa", "nikola", "vinko", "filipa"]	["josipa", "filipa", "marina", "nikola"]	"vinko"
["mislav", "stanko", "mislav", "ana"]	["stanko", "ana", "mislav"]	"mislav"

입출력 예 설명

예제 #1

"leo"는 참여자 명단에는 있지만, 완주자 명단에는 없기 때문에 완주하지 못했습니다.

예제 #2

"vinko"는 참여자 명단에는 있지만, 완주자 명단에는 없기 때문에 완주하지 못했습니다.

예제 #3

"mislav"는 참여자 명단에는 두 명이 있지만, 완주자 명단에는 한 명밖에 없기 때문에 한명은 완주하지 못했습니다.

34. 전화번호 목록 - Level 2

URL <https://programmers.co.kr/learn/courses/30/lessons/42577>

문제 설명

전화번호부에 적힌 전화번호 중, 한 번호가 다른 번호의 접두어인 경우가 있는지 확인하려 합니다. 전화번호가 다음과 같을 경우, 구조대 전화번호는 영석이의 전화번호의 접두사입니다.

- 구조대 : 119
- 박준영 : 97 674 223
- 지영석 : 11 9552 4421

전화번호부에 적힌 전화번호를 담은 배열 `phone_book` 이 `solution` 함수의 매개변수로 주어질 때, 어떤 번호가 다른 번호의 접두어인 경우가 있으면 `false`를 그렇지 않으면 `true`를 `return` 하도록 `solution` 함수를 작성해주세요.

제한 사항

- `phone_book`의 길이는 1 이상 1,000,000 이하입니다.
 - 각 전화번호의 길이는 1 이상 20 이하입니다.
 - 같은 전화번호가 중복해서 들어있지 않습니다.

입출력 예제

phone_book	return
["119", "97674223", "1195524421"]	false
["123", "456", "789"]	true
["12", "123", "1235", "567", "88"]	false

입출력 예 설명

입출력 예 #1

앞에서 설명한 예와 같습니다.

입출력 예 #2

한 번호가 다른 번호의 접두사인 경우가 없으므로, 답은 `true`입니다.

입출력 예 #3

첫 번째 전화번호, “12”가 두 번째 전화번호 “123”의 접두사입니다. 따라서 답은 `false`입니다.

35. 위장 - Level 2

URL <https://school.programmers.co.kr/learn/courses/30/lessons/42578>

문제 설명

코니는 매일 다른 옷을 조합하여 입는것을 좋아합니다.

예를 들어 코니가 가진 옷이 아래와 같고, 오늘 코니가 동그란 안경, 긴 코트, 파란색 티셔츠를 입었다면 다음날은 청바지를 추가로 입거나 동그란 안경 대신 검정 선글라스를 착용하거나 해야합니다.

종류	이름
얼굴	동그란 안경, 검정 선글라스
상의	파란색 티셔츠
하의	청바지
겉옷	긴 코트

- 코니는 각 종류별로 최대 1가지 의상만 착용할 수 있습니다. 예를 들어 위 예시의 경우 동그란 안경과 검정 선글라스를 동시에 착용할 수는 없습니다.
- 착용한 의상의 일부가 겹치더라도, 다른 의상이 겹치지 않거나, 혹은 의상을 추가로 더 착용한 경우에는 서로 다른 방법으로 옷을 착용한 것으로 계산합니다.
- 코니는 하루에 최소 한 개의 의상은 입습니다.

•

코니가 가진 의상들이 담긴 2차원 배열 clothes가 주어질 때 서로 다른 옷의 조합의 수를 return 하도록 solution 함수를 작성해주세요.

제한 사항

- clothes의 각 행은 [의상의 이름, 의상의 종류]로 이루어져 있습니다.
- 코니가 가진 의상의 수는 1개 이상 30개 이하입니다.
- 같은 이름을 가진 의상은 존재하지 않습니다.
- clothes의 모든 원소는 문자열로 이루어져 있습니다.
- 모든 문자열의 길이는 1 이상 20 이하인 자연수이고 알파벳 소문자 또는 '_' 로만 이루어져 있습니다.

입출력 예

clothes	return
[["yellow_hat", "headgear"], ["blue_sunglasses", "eyewear"], ["green_turban", "headgear"]]	5
[["crow_mask", "face"], ["blue_sunglasses", "face"], ["smoky_makeup", "face"]]	3

입출력 예 설명

예제 #1

headgear에 해당하는 의상이 yellow_hat, green_turban이고 eyewear에 해당하는 의상이 blue_sunglasses이므로 아래와 같이 5개의 조합이 가능합니다.

1. yellow_hat
2. blue_sunglasses
3. green_turban
4. yellow_hat + blue_sunglasses
5. green_turban + blue_sunglasses

예제 #2

face에 해당하는 의상이 crow_mask, blue_sunglasses, smoky_makeup이므로 아래와 같이 3개의 조합이 가능합니다.

1. crow_mask
2. blue_sunglasses
3. smoky_makeup

36. 오픈 채팅방 -Level 2

URL <https://school.programmers.co.kr/learn/courses/30/lessons/42888>

문제 설명

오픈채팅방

카카오톡 오픈채팅방에서는 친구가 아닌 사람들과 대화를 할 수 있는데, 본래 닉네임이 아닌 가상의 닉네임을 사용하여 채팅방에 들어갈 수 있다.

신입사원인 김크루는 카카오톡 오픈 채팅방을 개설한 사람을 위해, 다양한 사람들이 들어오고, 나가는 것을 지켜볼 수 있는 관리자창을 만들기로 했다. 채팅방에 누군가 들어오면 다음 메시지가 출력된다.

"[닉네임]님이 들어왔습니다."

채팅방에서 누군가 나가면 다음 메시지가 출력된다.

"[닉네임]님이 나갔습니다."

채팅방에서 닉네임을 변경하는 방법은 다음과 같이 두 가지이다.

- 채팅방을 나간 후, 새로운 닉네임으로 다시 들어간다.
- 채팅방에서 닉네임을 변경한다.

닉네임을 변경할 때는 기존에 채팅방에 출력되어 있던 메시지의 닉네임도 전부 변경된다.

예를 들어, 채팅방에 "Muzi"와 "Prodo"라는 닉네임을 사용하는 사람이 순서대로 들어오면 채팅방에는 다음과 같이 메시지가 출력된다.

"Muzi님이 들어왔습니다."

"Prodo님이 들어왔습니다."

채팅방에 있던 사람이 나가면 채팅방에는 다음과 같이 메시지가 남는다.

"Muzi님이 들어왔습니다."

"Prodo님이 들어왔습니다."

"Muzi님이 나갔습니다."

Muzi가 나간후 다시 들어올 때, Prodo 라는 닉네임으로 들어올 경우 기존에 채팅방에 남아있던 Muzi도 Prodo로 다음과 같이 변경된다.

"Prodo님이 들어왔습니다."

"Prodo님이 들어왔습니다."

"Prodo님이 나갔습니다."

"Prodo님이 들어왔습니다."

채팅방은 중복 닉네임을 허용하기 때문에, 현재 채팅방에는 Prodo라는 닉네임을 사용하는 사람이 두 명이 있다. 이제, 채팅방에 두 번째로 들어왔던 Prodo가 Ryan으로 닉네임을 변경하면 채팅방 메시지는 다음과 같이 변경된다.

"Prodo님이 들어왔습니다."

"Ryan님이 들어왔습니다."

"Prodo님이 나갔습니다."

"Prodo님이 들어왔습니다."

채팅방에 들어오고 나가거나, 닉네임을 변경한 기록이 담긴 문자열 배열 record가 매개변수로 주어질 때, 모든 기록이 처리된 후, 최종적으로 방을 개설한 사람이 보게 되는 메시지를 문자열 배열 형태로 return 하도록 solution 함수를 완성하라.

제한 사항

- record는 다음과 같은 문자열이 담긴 배열이며, 길이는 1 이상 100,000 이하이다.
- 다음은 record에 담긴 문자열에 대한 설명이다.
 - ▷ 모든 유저는 [유저 아이디]로 구분한다.
 - ▷ [유저 아이디] 사용자가 [닉네임]으로 채팅방에 입장 - "Enter [유저 아이디] [닉네임]" (ex. "Enter uid1234 Muzi")
 - ▷ [유저 아이디] 사용자가 채팅방에서 퇴장 - "Leave [유저 아이디]" (ex. "Leave uid1234")
 - ▷ [유저 아이디] 사용자가 닉네임을 [닉네임]으로 변경 - "Change [유저 아이디] [닉네임]" (ex. "Change uid1234 Muzi")
 - ▷ 첫 단어는 Enter, Leave, Change 중 하나이다.
 - ▷ 각 단어는 공백으로 구분되어 있으며, 알파벳 대문자, 소문자, 숫자로만 이루어져 있다.
 - ▷ 유저 아이디와 닉네임은 알파벳 대문자, 소문자를 구별한다.
 - ▷ 유저 아이디와 닉네임의 길이는 1 이상 10 이하이다.
 - ▷ 채팅방에서 나간 유저가 닉네임을 변경하는 등 잘못된 입력은 주어지지 않는다.

입출력 예

record	result
["Enter uid1234 Muzi", "Enter uid4567 Prodo", "Leave uid1234", "Enter uid1234 Prodo", "Change uid4567 Ryan"]	["Prodo님이 들어왔습니다.", "Ryan님이 들어왔습니다.", "Prodo님이 나갔습니다.", "Prodo님이 들어왔습니다."]

입출력 예 설명

입출력 예 #1

문제의 설명과 같다.

37. 베스트 앨범 - Level 3

URL <https://school.programmers.co.kr/learn/courses/30/lessons/42579>

문제 설명

스트리밍 사이트에서 장르 별로 가장 많이 재생된 노래를 두 개씩 모아 베스트 앨범을 출시하려 합니다. 노래는 고유 번호로 구분하며, 노래를 수록하는 기준은 다음과 같습니다.

1. 속한 노래가 많이 재생된 장르를 먼저 수록합니다.
2. 장르 내에서 많이 재생된 노래를 먼저 수록합니다.
3. 장르 내에서 재생 횟수가 같은 노래 중에서는 고유 번호가 낮은 노래를 먼저 수록합니다.

노래의 장르를 나타내는 문자열 배열 `genres`와 노래별 재생 횟수를 나타내는 정수 배열 `plays`가 주어질 때, 베스트 앨범에 들어갈 노래의 고유 번호를 순서대로 return 하도록 `solution` 함수를 완성하세요.

제한 사항

- `genres[i]`는 고유번호가 `i`인 노래의 장르입니다.
- `plays[i]`는 고유번호가 `i`인 노래가 재생된 횟수입니다.
- `genres`와 `plays`의 길이는 같으며, 이는 1 이상 10,000 이하입니다.
- 장르 종류는 100개 미만입니다.
- 장르에 속한 곡이 하나라면, 하나의 곡만 선택합니다.
- 모든 장르는 재생된 횟수가 다릅니다.

입출력 예

genres	plays	return
["classic", "pop", "classic", "classic", "pop"]	[500, 600, 150, 800, 2500]	[4, 1, 3, 0]

입출력 예 설명

classic 장르는 1,450회 재생되었으며, classic 노래는 다음과 같습니다.

- 고유 번호 3: 800회 재생
- 고유 번호 0: 500회 재생
- 고유 번호 2: 150회 재생

pop 장르는 3,100회 재생되었으며, pop 노래는 다음과 같습니다.

- 고유 번호 4: 2,500회 재생
- 고유 번호 1: 600회 재생

따라서 pop 장르의 [4, 1]번 노래를 먼저, classic 장르의 [3, 0]번 노래를 그다음에 수록합니다.

- 장르 별로 가장 많이 재생된 노래를 최대 두 개까지 모아 베스트 앨범을 출시하므로 2번 노래는 수록되지 않습니다.

38. 피보나치 수 - Level 2

URL <https://school.programmers.co.kr/learn/courses/30/lessons/12945>

문제 설명

피보나치 수는 $F(0) = 0$, $F(1) = 1$ 일 때, 1 이상의 n 에 대하여 $F(n) = F(n-1) + F(n-2)$ 가 적용되는 수입니다.

예를들어

$$F(2) = F(0) + F(1) = 0 + 1 = 1$$

$$F(3) = F(1) + F(2) = 1 + 1 = 2$$

$$F(4) = F(2) + F(3) = 1 + 2 = 3$$

$$F(5) = F(3) + F(4) = 2 + 3 = 5$$

와 같이 이어집니다.

2 이상의 n 이 입력되었을 때, n 번째 피보나치 수를 1234567으로 나눈 나머지를 리턴하는 함수, solution을 완성해 주세요.

제한 사항

n 은 2 이상 100,000 이하인 자연수입니다.

입출력 예

n	return
3	2
5	5

입출력 예 설명

피보나치수는 0번째부터 0, 1, 1, 2, 3, 5, ... 와 같이 이어집니다.

39. N으로 표현 - Level 3

URL <https://school.programmers.co.kr/learn/courses/30/lessons/42895>

문제 설명

아래와 같이 5와 사칙연산만으로 12를 표현할 수 있습니다.

$$12 = 5 + 5 + (5 / 5) + (5 / 5)$$

$$12 = 55 / 5 + 5 / 5$$

$$12 = (55 + 5) / 5$$

5를 사용한 횟수는 각각 6,5,4 입니다. 그리고 이중 가장 작은 경우는 4입니다.

이처럼 숫자 N과 number가 주어질 때, N과 사칙연산만 사용해서 표현 할 수 있는 방법 중 N 사용횟수의 최솟값을 return 하도록 solution 함수를 작성하세요.

제한 사항

N은 1 이상 9 이하입니다.

number는 1 이상 32,000 이하입니다.

수식에는 괄호와 사칙연산만 가능하며 나누기 연산에서 나머지는 무시합니다.

최솟값이 8보다 크면 -1을 return 합니다.

입출력 예

N	number	return
5	12	4
2	11	3

입출력 예 설명

예제 #1

문제에 나온 예와 같습니다.

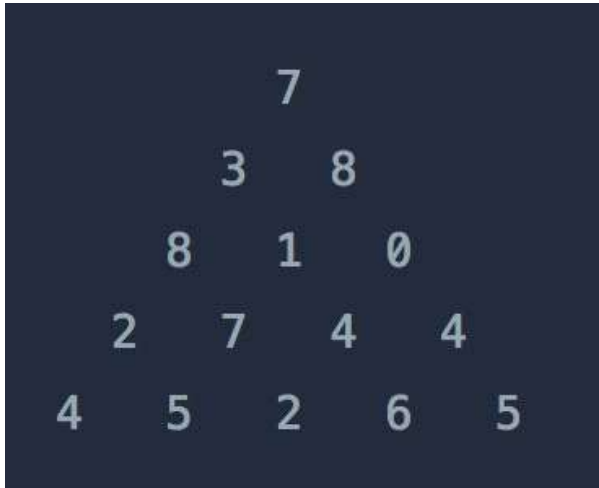
예제 #2

$11 = 22 / 2$ 와 같이 2를 3번만 사용하여 표현할 수 있습니다.

40. 정수 삼각형 - Level 1

URL <https://school.programmers.co.kr/learn/courses/30/lessons/43105>

문제 설명



위와 같은 삼각형의 꼭대기에서 바닥까지 이어지는 경로 중, 거쳐간 숫자의 합이 가장 큰 경우를 찾아보려고 합니다. 아래 칸으로 이동할 때는 대각선 방향으로 한 칸 오른쪽 또는 왼쪽으로만 이동 가능합니다. 예를 들어 3에서는 그 아래칸의 8 또는 1로만 이동이 가능합니다.

삼각형의 정보가 담긴 배열 `triangle`이 매개변수로 주어질 때, 거쳐간 숫자의 최댓값을 `return` 하도록 `solution` 함수를 완성하세요.

제한 사항

삼각형의 높이는 1 이상 500 이하입니다.

삼각형을 이루고 있는 숫자는 0 이상 9,999 이하의 정수입니다.

입출력 예

triangle	result
[[7], [3, 8], [8, 1, 0], [2, 7, 4, 4], [4, 5, 2, 6, 5]]	30

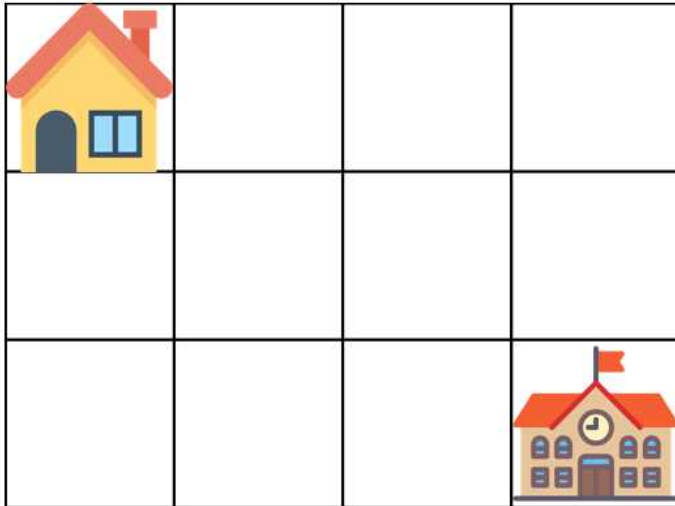
41. 등굣길 - Level3

URL <https://school.programmers.co.kr/learn/courses/30/lessons/42898>

문제 설명

계속되는 폭우로 일부 지역이 물에 잠겼습니다. 물에 잠기지 않은 지역을 통해 학교를 가려고 합니다. 집에서 학교까지 가는 길은 $m \times n$ 크기의 격자모양으로 나타낼 수 있습니다.

아래 그림은 $m = 4, n = 3$ 인 경우입니다.



가장 왼쪽 위, 즉 집이 있는 곳의 좌표는 (1, 1)로 나타내고 가장 오른쪽 아래, 즉 학교가 있는 곳의 좌표는 (m, n)으로 나타냅니다.

격자의 크기 m, n 과 물이 잠긴 지역의 좌표를 담은 2차원 배열 `puddles`이 매개변수로 주어집니다. 오른쪽과 아래쪽으로만 움직여 집에서 학교까지 갈 수 있는 최단경로의 개수를 1,000,000,007로 나눈 나머지를 `return` 하도록 `solution` 함수를 작성해주세요.

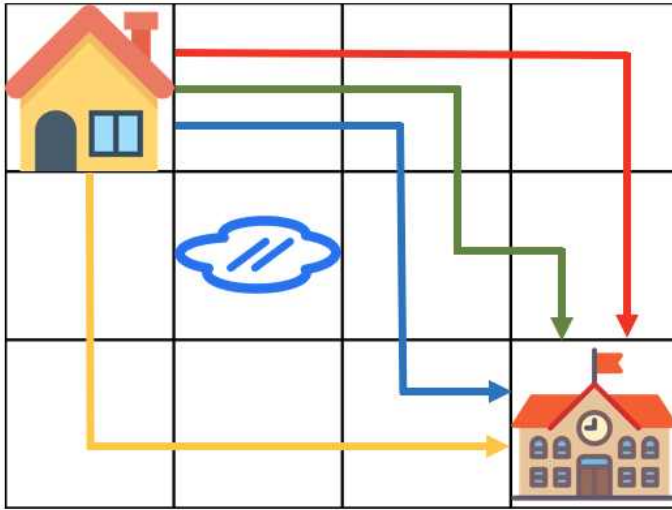
제한 사항

- 격자의 크기 m, n 은 1 이상 100 이하인 자연수입니다.
 - m 과 n 이 모두 1인 경우는 입력으로 주어지지 않습니다.
- 물에 잠긴 지역은 0개 이상 10개 이하입니다.
- 집과 학교가 물에 잠긴 경우는 입력으로 주어지지 않습니다.

입출력 예

m	n	puddles	return
4	3	[[2, 2]]	4

입출력 예 설명

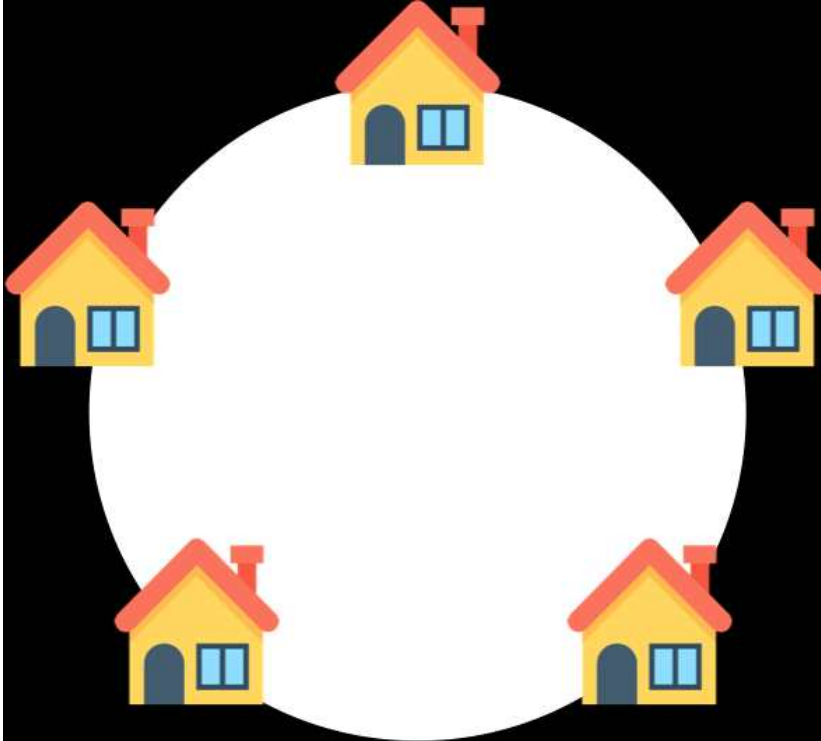


42. 도둑질 - Level4

URL <https://school.programmers.co.kr/learn/courses/30/lessons/42897>

문제 설명

도둑이 어느 마을을 털 계획을 하고 있습니다. 이 마을의 모든 집들은 아래 그림과 같이 동그랗게 배치되어 있습니다.



각 집들은 서로 인접한 집들과 방범장치가 연결되어 있기 때문에 인접한 두 집을 털면 경보가 울립니다.

각 집에 있는 돈이 담긴 배열 `money`가 주어질 때, 도둑이 훔칠 수 있는 돈의 최댓값을 `return` 하도록 `solution` 함수를 작성하세요.

제한 사항

이 마을에 있는 집은 3개 이상 1,000,000개 이하입니다.
`money` 배열의 각 원소는 0 이상 1,000 이하인 정수입니다.

입출력 예

money	return
[1, 2, 3, 1]	4

43. 주식 가격 - Level 2

URL <https://school.programmers.co.kr/learn/courses/30/lessons/42584>

문제 설명

초 단위로 기록된 주식가격이 담긴 배열 `prices`가 매개변수로 주어질 때, 가격이 떨어지지 않은 기간은 몇 초인지를 `return` 하도록 `solution` 함수를 완성하세요.

제한 사항

`prices`의 각 가격은 1 이상 10,000 이하인 자연수입니다.

`prices`의 길이는 2 이상 100,000 이하입니다.

입출력 예

prices	return
[1, 2, 3, 2, 3]	[4, 3, 1, 1, 0]

입출력 예 설명

- 1초 시점의 ₩1은 끝까지 가격이 떨어지지 않았습니다.
- 2초 시점의 ₩2은 끝까지 가격이 떨어지지 않았습니다.
- 3초 시점의 ₩3은 1초뒤에 가격이 떨어집니다. 따라서 1초간 가격이 떨어지지 않은 것으로 봅니다.
- 4초 시점의 ₩2은 1초간 가격이 떨어지지 않았습니다.
- 5초 시점의 ₩3은 0초간 가격이 떨어지지 않았습니다.

44. 기능 개발 - Level 2

URL <https://school.programmers.co.kr/learn/courses/30/lessons/42586>

문제 설명

프로그래머스 팀에서는 기능 개선 작업을 수행 중입니다. 각 기능은 진도가 100%일 때 서비스에 반영할 수 있습니다.

또, 각 기능의 개발속도는 모두 다르기 때문에 뒤에 있는 기능이 앞에 있는 기능보다 먼저 개발될 수 있고, 이때 뒤에 있는 기능은 앞에 있는 기능이 배포될 때 함께 배포됩니다.

먼저 배포되어야 하는 순서대로 작업의 진도가 적힌 정수 배열 progresses와 각 작업의 개발 속도가 적힌 정수 배열 speeds가 주어질 때 각 배포마다 몇 개의 기능이 배포되는지를 return 하도록 solution 함수를 완성하세요.

제한 사항

- 작업의 개수(progresses, speeds배열의 길이)는 100개 이하입니다.
- 작업 진도는 100 미만의 자연수입니다.
- 작업 속도는 100 이하의 자연수입니다.
- 배포는 하루에 한 번만 할 수 있으며, 하루의 끝에 이루어진다고 가정합니다. 예를 들어 진도율이 95%인 작업의 개발 속도가 하루에 4%라면 배포는 2일 뒤에 이루어집니다.

입출력 예

progresses	speeds	return
[93, 30, 55]	[1, 30, 5]	[2, 1]
[95, 90, 99, 99, 80, 99]	[1, 1, 1, 1, 1, 1]	[1, 3, 2]

입출력 예 설명

입출력 예 #1

첫 번째 기능은 93% 완료되어 있고 하루에 1%씩 작업이 가능하므로 7일간 작업 후 배포가 가능합니다.

두 번째 기능은 30%가 완료되어 있고 하루에 30%씩 작업이 가능하므로 3일간 작업 후 배포가 가능합니다. 하지만 이전 첫 번째 기능이 아직 완성된 상태가 아니기 때문에 첫 번째 기능이 배포되는 7일째 배포됩니다.

세 번째 기능은 55%가 완료되어 있고 하루에 5%씩 작업이 가능하므로 9일간 작업 후 배포가 가능합니다.

따라서 7일째에 2개의 기능, 9일째에 1개의 기능이 배포됩니다.

입출력 예 #2

모든 기능이 하루에 1%씩 작업이 가능하므로, 작업이 끝나기까지 남은 일수는 각각 5일, 10일, 1일, 1일, 20일, 1일입니다. 어떤 기능이 먼저 완성되었더라도 앞에 있는 모든 기능이 완성되지 않으면 배포가 불가능합니다.

따라서 5일째에 1개의 기능, 10일째에 3개의 기능, 20일째에 2개의 기능이 배포됩니다.

45. 가장먼노드 - Level 1

URL <https://school.programmers.co.kr/learn/courses/30/lessons/49189>

문제 설명

n 개의 노드가 있는 그래프가 있습니다. 각 노드는 1부터 n 까지 번호가 적혀있습니다. 1번 노드에서 가장 멀리 떨어진 노드의 갯수를 구하려고 합니다. 가장 멀리 떨어진 노드란 최단경로로 이동했을 때 간선의 개수가 가장 많은 노드들을 의미합니다.

노드의 개수 n , 간선에 대한 정보가 담긴 2차원 배열 `vertex`가 매개변수로 주어질 때, 1번 노드로부터 가장 멀리 떨어진 노드가 몇 개인지를 return 하도록 `solution` 함수를 작성해주세요.

제한 사항

노드의 개수 n 은 2 이상 20,000 이하입니다.

간선은 양방향이며 총 1개 이상 50,000개 이하의 간선이 있습니다.

`vertex` 배열 각 행 `[a, b]`는 a 번 노드와 b 번 노드 사이에 간선이 있다는 의미입니다.

입출력 예

n	vertex	return
6	[[3, 6], [4, 3], [3, 2], [1, 3], [1, 2], [2, 4], [5, 2]]	3

입출력 예 설명

예제의 그래프를 표현하면 아래 그림과 같고, 1번 노드에서 가장 멀리 떨어진 노드는 4,5,6번 노드입니다.

