

[5 차] 문제 1**문제 설명**

계단 n 칸을 올라가는 방법의 수를 구하려고 합니다. 계단은 한 번에 1 계단, 2 계단, 3 계단씩 오를 수 있습니다.

예를 들어, 계단 3 칸을 오르는 방법은 다음과 같이 4 가지가 있습니다.

1. 1 계단 + 1 계단 + 1 계단
2. 1 계단 + 2 계단
3. 2 계단 + 1 계단
4. 3 계단

계단 수 n 이 매개변수로 주어질 때, 계단을 오르는 경우의 수를 return 하도록 solution 함수를 작성하려 합니다. 빈칸을 채워 전체 코드를 완성해 주세요.

매개변수 설명

계단 수 n 이 solution 함수의 매개변수로 주어집니다.

- ♦ n 은 3 이상 30 이하인 정수입니다.

Return 값 설명

계단을 오르는 경우의 수를 return 합니다.

예시

n	return
3	4
4	7

예시 설명

예시 #1

문제에 나온 예와 같습니다.

예시 #2

계단 4 칸을 오르는 방법은 다음과 같이 7 가지가 있습니다.

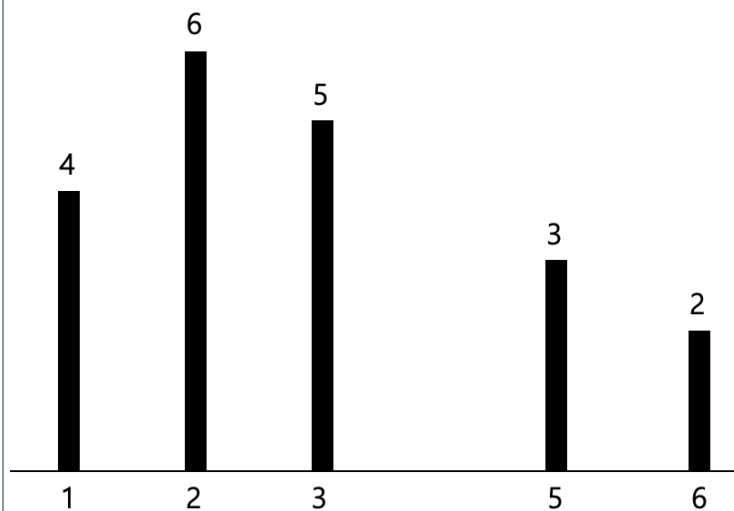
1. 1 계단 + 1 계단 + 1 계단 + 1 계단
2. 1 계단 + 1 계단 + 2 계단
3. 1 계단 + 2 계단 + 1 계단
4. 2 계단 + 1 계단 + 1 계단
5. 1 계단 + 3 계단
6. 3 계단 + 1 계단

7. 2 계단 + 2 계단

[5 차] 문제 2

문제 설명

아래와 같이 벽이 여러 개 있을 때, 벽 2 개를 제외한 나머지 벽을 제거하여 물을 최대한 담으려 합니다. 물은 '두 벽 사이의 거리 x 두 벽 중 낮은 벽의 높이' 리터 만큼 담을 수 있으며, 두 벽의 거리는 두 벽의 위치 차이입니다.



모든 벽의 위치와 높이를 담은 2 차원 리스트 walls 가 매개변수로 주어질 때, 물을 최대 몇 리터나 담을 수 있는지 return 하도록 solution 함수를 작성했습니다. 그러나, 일부 코드가 잘못되어 코드가 바르게 동작하지 않습니다. 주어진 코드가 모든 입력을 바르게 처리하도록 코드를 수정해 주세요. 코드는 한 줄만 수정해야 합니다.

매개변수 설명

모든 벽의 위치와 높이를 담은 2차원 리스트 walls가 solution 함수의 매개변수로 주어집니다.

- ◆ walls는 길이가 3이상 100이하인 2차원 리스트입니다.
- ◆ walls의 각 원소는 [벽의 위치, 벽의 높이] 이고, 벽의 위치를 기준으로 오름차순으로 정렬되어 있습니다.
 - 벽의 위치는 1 이상 100 이하입니다.
 - 벽의 높이는 1 이상 100 이하입니다..

Return 값 설명

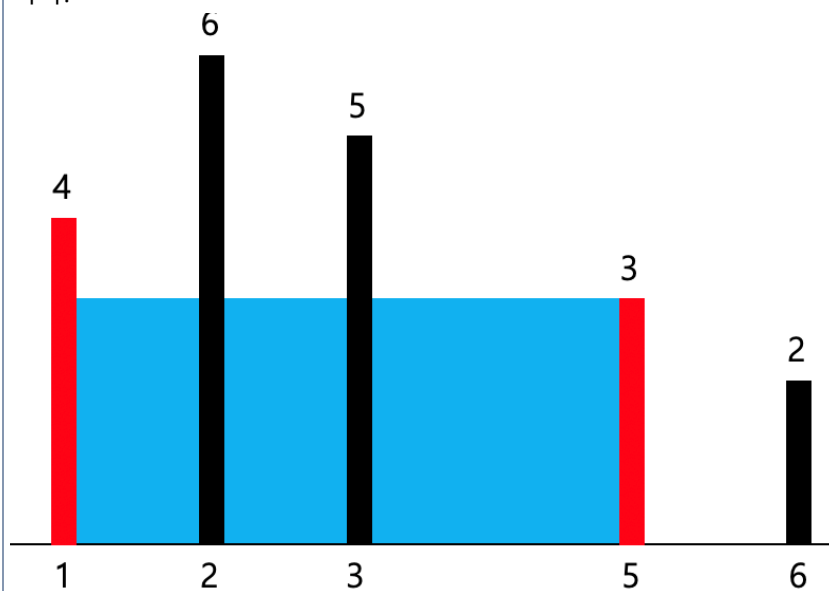
물을 최대 몇 리터나 담을 수 있는지 return 합니다.

예시

walls	return
[[1, 4], [2, 6], [3, 5], [5, 3], [6, 2]]	12

예시 설명

아래와 같이 벽을 남겼을 때 물을 최대로 담을 수 있습니다. 이 때 최대 12 리터를 담을 수 있습니다.

**[5 차] 문제 3****문제 설명**

서로 다른 정수 n 개가 담긴 리스트가 있습니다. 이 리스트를 앞 $(n+1)/2$ 개 원소는 증가하고 뒤 $(n+1)/2$ 개 원소는 감소하도록 정렬하려고 합니다. 이때, 조건을 만족하는 리스트가 여럿인 경우 사전 순으로 가장 먼저 나오는 리스트를 답으로 합니다. 예를 들어, 주어진 리스트가 [7, 3, 4, 1, 2, 5, 6]라면 정렬한 이후에는 [1, 2, 3, 7, 6, 5, 4]가 됩니다.

정수 리스트 numbers 가 매개변수로 주어질 때, 문제의 조건에 맞게 정렬하여 return 하도록 solution 함수를 작성했습니다. 그러나, 코드 일부분이 잘못되어 있기 때문에 몇몇 입력에 대해서는 올바르게 동작하지 않습니다. 주어진 코드에서 한 줄만 변경해서 모든 입력에 대해 올바르게 동작하도록 수정하세요.

매개변수 설명

정수 리스트 numbers가 solution 함수의 매개변수로 주어집니다.

- numbers의 각 원소는 -1,000 이상 1,000 이하인 정수이며, 중복되는 수는 없습니다.
- numbers의 길이는 3 이상 10,000 이하인 홀수입니다.

Return 값 설명

주어진 리스트를 앞 $(n+1)/2$ 개 원소는 증가하고 뒤 $(n+1)/2$ 개 원소는 감소하도록 정렬하여 return 합니다.

조건을 만족하는 리스트가 여럿인 경우 사전 순으로 가장 먼저 나오는 리스트를 답으로 합니다.

예시

numbers	return
[7, 3, 4, 1, 2, 5, 6]	[1, 2, 3, 7, 6, 5, 4]

예시 설명

문제에 나온 예와 같습니다.

[5 차] 문제 4**문제 설명**

주어진 숫자를 각 숫자와 숫자의 개수로 읽으려 합니다. 이때, 값이 큰 숫자를 먼저 읽어야 합니다.

예를 들어, 2433은 2 한개 4 한개 3 두개로 이루어져 있기 때문에 "413221"로 읽습니다.

숫자 number 가 매개변수로 주어질 때, 숫자를 읽어 문자열로 return 하도록 solution 함수를 작성했습니다. 그러나, 일부 코드가 잘못되어 코드가 바르게 동작하지 않습니다. 주어진 코드가 모든 입력을 바르게 처리하도록 코드를 수정해주세요. 코드는 한 줄만 수정해야 합니다.

매개변수 설명

숫자 number가 solution 함수의 매개변수로 주어집니다.

- number는 0 이상 2,000,000,000 이하인 정수입니다.

Return 값 설명

주어진 숫자를 읽어 문자열 형태로 return 합니다.

예시

number	return
2433	"413221"
662244	"624222"

예시 설명

예시 #1

문제에 나온 예와 같습니다.

예시 #2

662244 는 6 두개 2 두개 4 두개로 이루어져 있기 때문에 "624222"로 읽습니다.

[5 차] 문제 5**문제 설명**

XX 게임에선 캐릭터는 자신과 공격력이 같거나 자신보다 공격력이 작은 몬스터에게 이깁니다. 내가 가진 캐릭터가 최대 몬스터 몇 마리를 이길 수 있는지 구하려 합니다. 단, 한 캐릭터는 한 번만 싸울 수 있습니다.

예를 들어, 세 몬스터의 공격력이 각각 [1, 4, 3]이고, 내가 가진 두 캐릭터의 공격력이 각각 [1, 3]이라면 첫 번째 캐릭터는 첫 번째 몬스터와, 두 번째 캐릭터는 세 번째 몬스터와 싸워서 이길 수 있습니다. 따라서 이길 수 있는 몬스터 수는 최대 2 마리입니다.

모든 몬스터의 공격력을 담은 리스트 `enemies`, 내가 가진 모든 캐릭터의 공격력을 담은 리스트 `armies` 가 매개변수로 주어질 때, 내 캐릭터로는 최대 몬스터 몇 마리를 이길 수 있는지 `return` 하도록 `solution` 함수를 작성해 주세요.

매개변수 설명

모든 몬스터의 공격력을 담은 리스트 `enemies`, 내가 가진 모든 캐릭터의 공격력을 담은 리스트 `armies`가 `solution` 함수의 매개변수로 주어집니다.

- ◆ 몬스터 수는 1마리 이상, 500마리 이하입니다.
- ◆ 각 몬스터의 공격력은 1 이상 100 이하입니다.
- ◆ 내가 가진 캐릭터 수는 1개 이상 500개 이하입니다.
- ◆ 각 캐릭터의 공격력은 1 이상 100 이하입니다.

Return 값 설명

내가 가진 캐릭터로는 최대 몬스터 몇 마리를 이길 수 있는지 return 해 주세요.

예시

enemies	armies	return
[1, 4, 3]	[1, 3]	2
[1, 1, 1]	[1, 2, 3, 4]	3

예시 설명

입출력 예 #1

문제에 나온 예와 같습니다.

입출력 예 #2

캐릭터를 어떻게 보내도 모든 몬스터를 이길 수 있습니다.

[5 차] 문제 6

문제 설명

p 진법으로 표현한 수란, 각 자리를 0 부터 p-1 의 숫자로만 나타낸 수를 의미합니다. p 진법으로 표현한 자연수 두개를 더한 결과를 q 진법으로 표현하려 합니다.

예를 들어, 3 진법 수 112001 과 12010 을 더한 결과를 8 진법으로 나타내면 1005 입니다.

solution 함수의 매개변수로 p 진법 자연수를 담은 문자열 s1, s2 와 두 수를 나타내는 진법의 기수 p, 두 수의 덧셈 결과를 표현할 진법의 기수 q 가 매개변수로 주어집니다. p 진법으로 표현된 두 수를 더한 결과를 q 진법으로 나타낸 값을 return 하도록 solution 함수를 완성해 주세요.

매개변수 설명

p 진법으로 자연수를 담은 문자열 s1, s2와 두 수를 표현한 진법의 기수 p, 두 수의 덧셈 결과를 표현할 진법의 기수 q가 solution 함수의 매개변수로 주어집니다.

- p와 q는 2 이상 10 이하인 자연수입니다.
- s1과 s2의 길이는 1 이상 9 이하입니다.
- s1과 s2의 원소는 '0', '1', '2', ..., 'p-1'로만 구성됩니다.
- s1이나 s2가 '0'인 경우는 주어지지 않습니다.

Return 값 설명

두 수를 더한 결과를 q 진법으로 나타낸 값을 문자열로 return 하도록 solution 함수를 완성해 주세요.

~~~~~

#### 예시

| s1       | s2      | p | q | return |
|----------|---------|---|---|--------|
| "112001" | "12010" | 3 | 8 | "1005" |

#### 예시 설명

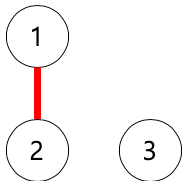
문제에 나온 예와 같습니다.

## [5 차] 문제 7

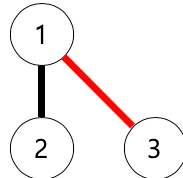
#### 문제 설명

그래프의 노드 수와 노드 연결 순서가 주어질 때, 몇 번째 연결에 사이클이 생기는지 알고 싶습니다. 예를 들어, 노드가 3 개이고 노드를 [[1, 2], [1, 3], [2, 3]] 순으로 연결한다면 아래 그림과 같습니다.

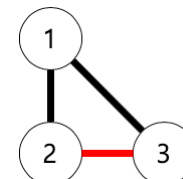
##### ◆ 1 번째 연결



##### ◆ 2 번째 연결



##### ◆ 3 번째 연결



따라서 3 번째 연결에서 사이클이 생깁니다.

그래프의 노드 수  $n$ , 노드 연결 순서 `connections` 가 매개변수로 주어질 때, 몇 번째 연결에 사이클이 생기는지 `return` 하도록 `solution` 함수를 작성하려 합니다. 빈칸을 채워 전체 코드를 완성해 주세요.

#### 매개변수 설명

그래프의 노드 수  $n$ , 노드 연결 순서 `connections`가 `solution` 함수의 매개변수로 주어집니다.

- ◆ 그래프의 노드 수  $n$ 은 3 이상 10 이하입니다.
- ◆ `connections`은 길이가 3 이상 50 이하인 리스트입니다.
- ◆ `connections` 리스트의 각 행은 `[a, b]`는  $a$ 번 노드와  $b$ 번 노드를 연결한다는 의미입니다.
- ◆ 항상 사이클이 생기는 경우만 주어집니다.

#### Return 값 설명

몇 번째 연결에서 사이클이 생기는지 `return` 합니다.

#### 예시

| <b>n</b> | <b>connections</b>       | <b>return</b> |
|----------|--------------------------|---------------|
| 3        | [[1, 2], [1, 3], [2, 3]] | 3             |

#### 예시 설명

문제에 나온 예와 같습니다.

## [5 차] 문제 8

#### 문제 설명

세 수  $a, b, c$ 의 공약수가 몇 개인지 구하려고 합니다. 공약수란, 동시에 모든 정수의 약수인 정수를 뜻합니다. 예를 들어, 세 수 24, 9, 15의 공약수는 1, 3 이고, 따라서 양의 공약수는 2 개입니다.

세 수의 공약수가 몇 개인지 구하기 위해 다음과 같이 프로그램 구조를 작성했습니다.

1. 세 수의 최대공약수를 구합니다.
2. 앞서 구한 최대공약수의 약수가 몇 개인지 구합니다.

세 수  $a, b, c$ 가 매개변수로 주어질 때, 세 수의 약수가 몇 개인지 `return` 하도록 `solution` 함수를 작성하려 합니다. 위 구조를 참고하여 코드가 올바르게 동작할 수 있도록 빈칸에 주어진 `func_a`, `func_b`, `func_c` 함수와 매개변수를 알맞게 채워주세요.

#### 매개변수 설명



세 수 a, b, c가 매개변수로 주어집니다.

- ◆ 세 수 a, b, c는 1 이상 1,000 이하인 정수입니다..

#### Return 값 설명

세 수의 약수가 몇 개인지 return 합니다.

#### 예시

| a  | b | c  | return |
|----|---|----|--------|
| 24 | 9 | 15 | 2      |

#### 예시 설명

문제에 나온 예와 같습니다.

## [5 차] 문제 9

#### 문제 설명

정수 number 와 target 이 주어졌을 때, 다음 세 연산을 이용해 number 를 target 으로 만들려 합니다.

연산 1. 1 을 더합니다.

연산 2. 1 을 뺍니다.

연산 3. 2 를 곱합니다.

정수 number 와 target 이 매개변수로 주어질 때 number 를 target 으로 만들려면 연산을 최소 몇 번 해야 하는지 return 하도록 solution 함수를 작성해 주세요.

#### 매개변수 설명

두 정수 number와 target이 solution 함수의 매개변수로 주어집니다.

- ◆ number와 target은 0 이상 10,000 이하입니다.

#### Return 값 설명

number 를 target 으로 만들려면 연산을 최소 몇 번 해야 하는지 return 합니다.

**예시**

| number | target | return |
|--------|--------|--------|
| 5      | 9      | 2      |
| 3      | 11     | 3      |

**예시 설명**

예시 #1

1.  $5 \times 2 = 10$ 2.  $10 - 1 = 9$ 

따라서 연산을 최소 2 번 해야 합니다.

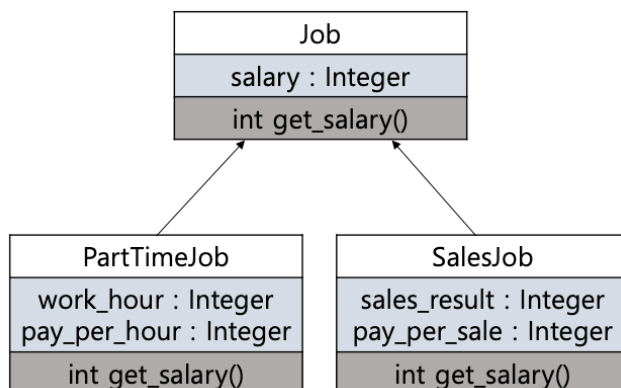
예시 #2

1.  $3 \times 2 = 6$ 2.  $6 \times 2 = 12$ 3.  $12 - 1 = 11$ 

따라서 연산을 최소 3 번 해야 합니다.

**[5 차] 문제 10****문제 설명**

직업에 아르바이트, 판매사원이 있습니다. 직원들 급여의 총합을 계산하기 위해 아래와 같이 Job, PartTimeJob, SalesJob 클래스를 작성했습니다.



\* Job :

\* Job : 직업을 나타내는 클래스입니다.

\* salary : 직업의 급여를 나타냅니다. 초기 급여는 0 입니다.

\* get\_salary : 직업의 급여를 return 합니다.

\* PartTimeJob :

- \* PartTimeJob : 아르바이트를 나타내는 클래스이며 Job 을 상속합니다.
- \* work\_hour : 아르바이트를 한 시간입니다.
- \* pay\_per\_hour : 아르바이트의 시간 당 급여입니다.
- \* get\_salary : 아르바이트 급여를 계산하여 return 합니다.

\* SalesJob :

- \* SalesJob : 판매사원을 나타내는 클래스이며 Job 을 상속합니다.
- \* sales\_result : 판매사원의 판매실적입니다.
- \* pay\_per\_sale : 판매실적 당 급여입니다.
- \* get\_salary : 판매사원의 급여를 계산하여 return 합니다.

주어진 아르바이트, 판매사원 급여의 총합을 계산하려 합니다.

- \* 아르바이트는 기본적으로 `아르바이트를 한 시간 X 시간 당 급여`를 받으며 40 시간 이상 근무 시 8 시간만큼의 급여를 추가로 받습니다.
- \* 판매사원은 기본적으로 `판매실적 \* 판매실적 당 급여`를 받으며 판매실적이 10 건이 넘으면 급여를 2 배로, 20 건이 넘으면 급여를 3 배로 받습니다.

아르바이트의 정보가 담긴 2 차원 리스트 part\_time\_jobs, 판매사원의 정보가 담긴 2 차원 리스트 sales\_jobs 가 매개변수로 주어질 때, 모든 직원들 급여의 총합을 return 하도록 solution 함수를 작성하려 합니다. 위 클래스 구조를 참고하여 주어진 코드의 빈 칸을 적절히 채워 전체 코드를 완성해 주세요.

~~~~~

매개변수 설명

아르바이트의 정보가 담긴 2차원 리스트 part_time_jobs, 판매사원의 정보가 담긴 2차원 리스트 sales_jobs가 solution 함수의 매개변수로 주어집니다.

- ◆ part_time_jobs은 길이가 1이상 10이하인 2차원 리스트입니다.
- ◆ part_time_jobs의 각 원소는 [아르바이트를 한 시간, 시간 당 급여]입니다.
 - 아르바이트를 한 시간은 1이상 100이하입니다.
 - 시간 당 급여는 1,000이상 20,000이하입니다.
- ◆ sales_jobs은 길이가 1이상 10이하인 2차원 리스트입니다.
- ◆ sales_jobs의 각 원소는 [판매실적, 판매실적 당 급여]입니다.
 - 판매실적은 1이상 30이하입니다.
 - 판매실적 당 급여는 1,000이상 20,000이하입니다..

~~~~~

### Return 값 설명

모든 직원들 급여의 총합을 return 합니다.

**예시**

| <b>part_time_jobs</b>                | <b>sales_jobs</b>        | <b>return</b> |
|--------------------------------------|--------------------------|---------------|
| [[10, 5000], [43, 6800], [5, 12800]] | [[3, 18000], [12, 8500]] | 718800        |

**예시 설명**

첫번째 아르바이트에게 50,000( $10 * 5,000$ )을 지급합니다.

두번째 아르바이트에게 346,800( $43 * 6,800 + 8 * 6,800$ )을 지급합니다.

세번째 아르바이트에게 64,000( $5 * 12,800$ )을 지급합니다.

첫번째 영업사원에게 54,000( $3 * 18,000$ )을 지급합니다.

두번째 영업사원에게 204,000( $12 * 8,500 * 2$ )을 지급합니다.

따라서 급여의 총합은 718,800 입니다.