

[4 차] 문제 1

문제 설명

어떤 단어가 XX 사전의 몇 번째 단어인지 알고 싶습니다. XX 사전에는 대문자 알파벳 'A', 'E', 'I', 'O', 'U'를 사용해 만들 수 있는 길이가 5 이하인 모든 단어가 수록되어 있습니다.

예를 들어, 사전의 첫 번째 단어는 "A"이고, 그 다음은 "AA"입니다. 마지막 단어는 "UUUUU"입니다.

문자열 word 가 매개변수로 주어질 때, word 가 사전의 몇 번째 단어인지 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

문자열 word가 solution 함수의 매개변수로 주어집니다.

- word는 'A', 'E', 'I', 'O', 'U'로만 구성됩니다.
- word의 길이는 5 이하입니다.

Return 값 설명

사전에서 word 가 몇 번째 단어인지 return 해 주세요.

예시

word	return
"AAAAE"	6
"AAAE"	10

예시 설명

사전엔 단어가 다음과 같이 수록됩니다. "A", "AA", "AAA", "AAAA", "AAAAA", "AAAAE", "AAAAI", "AAAAO", "AAAAU", "AAAE", ...

예시 #1

"AAAAE"는 여섯 번째에 나옵니다.

예시 #2

"AAAE"는 열 번째에 나옵니다.

[4 차] 문제 2

문제 설명

알파벳 소문자와 대문자로 구성된 문자열을 압축하려 합니다. 압축이란 대표 문자와 대표 문자가 연속되는 개수를 함께 표현하는 것입니다. 이때, 대문자와 소문자는 구분하지 않으며, 대표 문자는 소문자로 표현합니다.

예를 들어, 문자열 "YYYYYbbbBbbBBBMmmM"을 압축하면 "y5b9m4"입니다.

문자열 s 가 매개변수로 주어질 때 s 를 압축한 문자열을 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

문자열 s 가 solution 함수의 매개변수로 주어집니다.

- s 의 길이는 1 이상 100000 이하입니다.
- s 는 알파벳 소문자, 대문자로만 구성됩니다.
- 알파벳이 반복되는 횟수는 9회 이하로만 주어집니다.
 - 즉, "sssSSSSsss" 와 같은 경우는 주어지지 않습니다.

Return 값 설명

s 를 압축한 결과를 return 해 주세요.

예시

s	return
"YYYYYbbbBbbBBBMmmM"	"y5b9m4"

예시 설명

문제의 예와 같습니다.

[4 차] 문제 3

문제 설명

정확히 n 일 연속으로 스키장을 이용하는데 필요한 최소 비용을 계산하려 합니다. 다음은 스키장에서 판매하는 이용권입니다.

이용권 종류	스키장을 사용할 수 있는 일수	가격
one_day	구매한 날 하루 동안 사용 가능	one_day_price
multi_day	구매한 날부터 multi_day 일간 사용 가능	multi_day_price

예를 들어 one_day_price = 3, multi_day = 5, multi_day_price = 14 라면, 1 일 이용권은 3 원, 5 일 이용권은 14 원입니다. n = 6 일 때 정확히 6 일 연속으로 스키장을 이용하는데 필요한 비용은 다음과 같이 계산합니다.

- ♦ 1 일 이용권 x 6 장 → 6 x 3 원 = 18 원
- ♦ 1 일 이용권 x 1 장 + 5 일 이용권 x 1 장 → 1 x 3 원 + 1 x 14 원 = 17 원

따라서 정확히 6 일 연속 스키장을 이용하려면 최소 17 원이 듭니다.

※ 스키장을 정확히 6 일간 이용해야 하므로 5 일 이용권을 두 장 살 수는 없습니다.

solution 함수의 매개변수로 one_day_price, multi_day, multi_day_price, n 가 주어집니다. 이때 정확히 n 일 연속 스키장을 이용하는데 필요한 최소 금액을 계산해서 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

one_day_price, multi_day, multi_day_price, n이 순서대로 solution 함수의 매개변수로 주어집니다.

- ♦ one_day_price는 1일 이용권의 가격을 나타냅니다.
- ♦ one_day_price는 1 이상 1000 이하인 자연수입니다.
- ♦ multi_day_price는 multi_day일 이용권의 가격을 나타냅니다.
- ♦ multi_day_price는 1 이상 1000 이하인 자연수입니다.
- ♦ multi_day는 3 이상 5 이하인 자연수입니다.
- ♦ n은 1 이상 10¹² 이하인 자연수입니다.

Return 값 설명

정확히 n 일 연속으로 스키장을 이용하는데 필요한 최소 금액을 return 해 주세요.

예시

one_day_price	multi_day	multi_day_price	n	return
3	5	14	6	17
2	3	5	11	19

예시 설명

예시 #1

앞서 설명한 예와 같습니다.

예시 #2

1 일 이용권은 2 원, 3 일 이용권은 5 원입니다. 정확히 11 일 동안 스키장을 이용할 때 드는 최소 비용은 다음과 같습니다.

- ♦ 1 일 이용권 x 2 장 + 3 일 이용권 x 3 장 → 2 x 2 원 + 3 x 5 원 = 19 원

[4 차] 문제 4

문제 설명

마방진이란 가로, 세로, 대각선 방향의 수를 더한 값이 모두 같은 정사각형 행렬입니다. 마방진에는 '1'부터 '정사각형 넓이'까지, 수가 하나씩 배치되어야 합니다. 아래는 가로, 세로, 대각선 방향의 수를 더한 값이 모두 34 인 4 x 4 마방진입니다.

16	2	3	13
5	11	10	8
9	7	6	12
4	14	15	1

4 x 4 행렬의 두 빈칸을 채워 행렬을 마방진으로 완성하려 합니다. 빈칸은 0 으로 표시합니다. 이를 위해 다음과 같이 프로그램 구조를 작성했습니다.

1. 두 빈칸의 위치를 찾습니다.
2. 숫자 1 ~ 16 중 존재하지 않는 숫자 2 개를 찾습니다.
3. 첫 번째 빈칸에 작은 숫자를, 두 번째 빈칸에 큰 숫자를 넣어 행렬이 마방진이 되는지 검사합니다.
 - 4-1. 마방진이라면 [작은 숫자의 행 번호, 작은 숫자의 열 번호, 작은 숫자, 큰 숫자의 행 번호, 큰 숫자의 열 번호, 큰 숫자]를 return 합니다.
 - 4-2. 마방진이 아니라면 [큰 숫자의 행 번호, 큰 숫자의 열 번호, 큰 숫자, 작은 숫자의 행 번호, 작은 숫자의 열 번호, 작은 숫자]를 return 합니다.

4 x 4 크기 2 차원 리스트 matrix 가 매개변수로 주어질 때, 이때 빈칸의 위치와 각 칸에 들어갈 수를 담은 배열을 return 하도록 solution 함수를 작성하려 합니다. 위 구조를 참고하여 코드가 올바르게 동작하도록 빈칸에 주어진 func_a, func_b, func_c 함수와 매개변수를 알맞게 채워주세요.

- ♦ 주어진 행렬은 빈칸을 채우면 반드시 마방진으로 완성할 수 있습니다.

~~~~~

### 매개변수 설명

2차원 리스트 matrix가 solution 함수의 매개변수로 주어집니다.

- ♦ matrix는 크기가 4 x 4인 2차원 배열입니다.
- ♦ matrix에는 0이 반드시 두 개 들었습니다.

- matrix에는 1 이상 16 이하인 자연수 14개가 겹치지 않게 들었습니다.

### Return 값 설명

빈칸의 위치와 각 칸에 들어갈 수를 담은 배열을 return 해 주세요.

### 예시

| matrix                                           | return        |
|--------------------------------------------------|---------------|
| [[16,2,3,13],[5,11,10,0],[9,7,6,12],[0,14,15,1]] | [4,1,4,2,4,8] |

### 예시 설명

|             |             |             |             |
|-------------|-------------|-------------|-------------|
| 16<br>(1,1) | 2<br>(1,2)  | 3<br>(1,3)  | 13<br>(1,4) |
| 5<br>(2,1)  | 11<br>(2,2) | 10<br>(2,3) | 0<br>(2,4)  |
| 9<br>(3,1)  | 7<br>(3,2)  | 6<br>(3,3)  | 12<br>(3,4) |
| 0<br>(4,1)  | 14<br>(4,2) | 15<br>(4,3) | 1<br>(4,4)  |

위 그림에서 빈칸은 (4, 1)과 (2, 4)입니다. 각 칸에 4와 8을 채우면 마방진이 완성됩니다. 따라서 [작은 숫자의 행 번호, 작은 숫자의 열 번호, 작은 숫자, 큰 숫자의 행 번호, 큰 숫자의 열 번호, 큰 숫자]를 return 해야 합니다. 답은 [4, 1, 4, 2, 4, 8]입니다.

|             |             |             |             |
|-------------|-------------|-------------|-------------|
| 16<br>(1,1) | 2<br>(1,2)  | 3<br>(1,3)  | 13<br>(1,4) |
| 5<br>(2,1)  | 11<br>(2,2) | 10<br>(2,3) | 8<br>(2,4)  |
| 9<br>(3,1)  | 7<br>(3,2)  | 6<br>(3,3)  | 12<br>(3,4) |
| 4<br>(4,1)  | 14<br>(4,2) | 15<br>(4,3) | 1<br>(4,4)  |

## [4 차] 문제 5

### 문제 설명

다음과 같이 새로운 숫자를 생성할 때, n 번째 생성할 숫자를 구하려고 합니다.

1. 첫 번째 수는 1 입니다.
2. 다음 수를 1의 자리에 추가합니다.
3. 만든 숫자를 뒤집습니다.

4. 숫자 1 부터 9 까지 추가하면서 2, 3 번째 단계를 반복합니다.
5. 숫자 9 까지 추가했다면 다시 숫자 1 부터 추가합니다.

$n$  이 매개변수로 주어질 때,  $n$  번째로 생성할 수를 문자열 형태로 return 하는 solution 함수를 작성하려 합니다. 위 구조를 참고하여 코드가 바르게 동작하도록 빈칸을 채워주세요.

#### 매개변수 설명

숫자  $n$ 이 solution 함수의 매개변수로 주어집니다.

- ♦  $n$ 은 1 이상 1,000 이하인 자연수입니다.

#### Return 값 설명

solution 함수는  $n$  번째 수를 문자열 형태로 return 합니다.

#### 예시

| n | return  |
|---|---------|
| 5 | "53124" |

#### 예시 설명

첫 번째 숫자: 1

두 번째 숫자: 21

세 번째 숫자: 312

네 번째 숫자: 4213

다섯 번째 숫자: 53124

따라서 답은 "53124"입니다.

## [4 차] 문제 6

#### 문제 설명

어떤 자리 수  $k$  가 주어졌을 때 각 자릿수의  $k$  제곱의 합이 원래 수가 되는 수를 자아도취 수라고 합니다. 예를 들어 153 은 세 자리 자아도취 수입니다.

$$153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27$$

자연수  $k$  가 매개변수로 주어질 때,  $k$  자리 자아도취 수들을 리스트에 오름차순으로 담아 return 하도록 solution 함수를 작성하려 합니다. 빈칸을 채워 전체 코드를 완성해 주세요.

**매개변수 설명**

k가 solution 함수의 매개변수로 주어집니다.

- ◆ k는 3 이상 6 이하인 자연수입니다.

**Return 값 설명**

k 자리 자아도취 수를 오름차순으로 정렬한 뒤 리스트에 담아 return 합니다.

**예시**

| k | return               |
|---|----------------------|
| 3 | [153, 370, 371, 407] |

**예시 설명**

\*  $153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$

\*  $370 = 3^3 + 7^3 + 0^3 = 27 + 343 + 0 = 370$

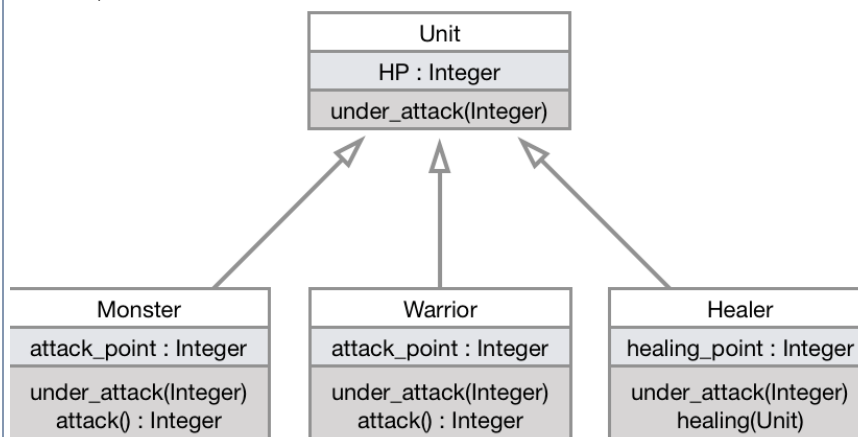
\*  $371 = 3^3 + 7^3 + 1^3 = 27 + 343 + 1 = 371$

\*  $407 = 4^3 + 0^3 + 7^3 = 64 + 0 + 343 = 407$

**[4 차] 문제 7****문제 설명**

게임에 몬스터, 전사, 힐러 세 종류의 유닛이 있습니다.

유닛들의 공격과 힐링에 따른 체력치의 변화를 계산하기 위해 아래와 같이 Unit, Monster, Warrior, Healer 클래스를 작성했습니다.



\* Unit :

\* Unit : 유닛을 나타내는 클래스입니다.

\* HP : 유닛의 체력을 나타냅니다. 유닛의 초기 체력은 1,000 입니다.

\* under\_attack : 유닛이 공격을 받으면 유닛의 체력을 감소시킵니다.

\* Monster :

\* Monster : 몬스터를 나타내는 클래스이며 Unit 을 상속합니다.

\* attack\_point : 몬스터의 공격력 입니다.

\* under\_attack : 몬스터가 공격을 받으면 몬스터의 체력을 감소시킵니다.

\* attack : 몬스터의 attack\_point 를 return 합니다.

\* Warrior :

\* Warrior : 전사를 나타내는 클래스이며 Unit 을 상속합니다.

\* attack\_point : 전사의 공격력 입니다.

\* under\_attack : 전사가 공격을 받으면 전사의 체력을 감소시킵니다.

\* attack : 전사의 attack\_point 를 return 합니다.

\* Healer :

\* Healer : 힐러를 나타내는 클래스이며 Unit 을 상속합니다.

\* healing\_point : 힐러의 회복치 입니다.

\* under\_attack : 힐러가 공격을 받으면 힐러의 체력을 감소시킵니다.

\* healing : 대상 유닛의 체력을 healing\_point 만큼 회복시킵니다. 회복 후 대상 유닛의 체력은 기본 체력 1,000 보다 커질 수 있습니다.

게임에서 각 유닛들이 다음과 같이 행동했을 때, 몬스터, 전사, 힐러의 남은 체력을 구하려 합니다.

\* 몬스터가 전사와 힐러를 한 번씩 공격했습니다.

\* 전사가 몬스터를 한 번 공격했습니다.

\* 힐러가 몬스터와 전사의 체력을 한 번씩 회복했습니다.

몬스터의 공격력 monster\_attack\_point, 전사의 공격력 warrior\_attack\_point, 힐러의 회복치 healing\_point 가 매개변수로 주어질 때, 각 유닛이 행동을 마친 후 몬스터, 전사, 힐러의 남은 체력을 순서대로 리스트에 담아 return 하도록 solution 함수를 작성하려 합니다. 위 클래스 구조를 참고하여 주어진 코드의 빈칸을 적절히 채워 전체 코드를 완성해 주세요.

#### 매개변수 설명

몬스터의 공격력 monster\_attack\_point, 전사의 공격력 warrior\_attack\_point, 힐러의 회복치 healing\_point가 solution 함수의 매개변수로 주어집니다.

- ♦ monster\_attack\_point는 1 이상 500 이하의 자연수입니다.



- ♦ warrior\_attack\_point는 1 이상 500 이하의 자연수입니다.
- ♦ healing\_point는 1 이상 500 이하의 자연수입니다.

### Return 값 설명

각 유닛이 문제에 주어진 행동을 마친 후 몬스터, 전사, 힐러의 남은 체력을 순서대로 리스트에 담아 return 해 주세요.

### 예시

| monster_attack_point | warrior_attack_point | healing_point | return          |
|----------------------|----------------------|---------------|-----------------|
| 100                  | 90                   | 30            | [940, 930, 900] |

### 예시 설명

- \* 몬스터 공격력 : 100
- \* 전사 공격력 : 90
- \* 힐러 회복치 : 30

몬스터가 전사와 힐러를 한 번씩 공격하면 각 유닛의 남은 체력은 다음과 같습니다.

| 몬스터 체력 | 전사 체력 | 힐러 체력 |
|--------|-------|-------|
| 1000   | 900   | 900   |

전사가 몬스터를 한 번 공격하면 각 유닛의 남은 체력은 다음과 같습니다.

| 몬스터 체력 | 전사 체력 | 힐러 체력 |
|--------|-------|-------|
| 910    | 900   | 900   |

힐러가 몬스터와 전사의 체력을 한 번씩 회복하면 각 유닛의 남은 체력은 다음과 같습니다.

| 몬스터 체력 | 전사 체력 | 힐러 체력 |
|--------|-------|-------|
| 940    | 930   | 900   |

따라서 [940, 930, 900]을 return 하면 됩니다.

**[4 차] 문제 8****문제 설명**

1 이상 9 이하 숫자가 적힌 카드를 이어 붙여 숫자를 만들었습니다. 이때, 숫자 카드를 조합해 만든 수 중에서  $n$  이 몇 번째로 작은 수인지 구하려 합니다.

예를 들어, 숫자 카드 1, 2, 1, 3 으로 만들 수 있는 수를 작은 순으로 나열하면 [1123, 1132, 1213, 1231, 1312, ... , 3121, 3211]입니다.  $n$  이 1312 라면, 숫자 카드를 조합해 만든 수 중  $n$  은 5 번째로 작은 수입니다.

숫자 카드를 담은 리스트 `card`, 수  $n$  이 매개변수로 주어질 때 숫자 카드를 조합해 만든 수 중에서  $n$  이 몇 번째로 작은 수인지 `return` 하도록 `solution` 함수를 완성해 주세요.

**매개변수 설명**

카드에 적힌 숫자를 담은 리스트 `card`, 수  $n$  이 `solution` 함수의 매개변수로 주어집니다.

- `card`는 길이가 9 이하인 리스트입니다.
- `card`의 원소는 1 이상 9 이하인 자연수입니다.
- $n$ 은 999,999,999 이하인 자연수입니다.
- $n$ 의 자릿수는 리스트 `card`의 길이와 같습니다.
- $n$ 의 각 자리의 숫자는 1 이상 9 이하입니다.

**Return 값 설명**

숫자 카드를 조합해 만든 수 중에서  $n$  이 몇 번째로 작은 수인지 `return` 해 주세요.

- 만약,  $n$  을 만들 수 없다면 -1 을 `return` 해 주세요.

**예시**

| card         | n    | return |
|--------------|------|--------|
| [1, 2, 1, 3] | 1312 | 5      |
| [1, 1, 1, 2] | 1122 | -1     |

**예시 설명**

예시 #1

앞서 설명한 예와 같습니다.

예시 #2

숫자 카드를 조합하면 [1112, 1121, 1211, 2111]를 만들 수 있습니다. 따라서 1122 는 만들 수 없습니다.

## [4 차] 문제 9

## 문제 설명

hour 시 minute 분에 아날로그 시계의 시침과 분침이 몇 도를 이루는지 계산하려 합니다. 예를 들어, 3 시 00 분에 시침과 분침은  $90^\circ$ 를 이룹니다.

어떤 시점의 시 hour, 분 minute 이 매개변수로 주어질 때, hour 시 minute 분에 아날로그 시계의 시침과 분침이 이루는 각도를 소숫점 첫 번째 자리까지 표현한 문자열을 return 하도록 solution 함수를 작성해 주세요.

## 매개변수 설명

어떤 시점의 시 hour, 분 minute이 solution 함수의 매개변수로 주어집니다.

- hour는 1 이상 12 이하인 자연수입니다.
- minute은 0 이상 59 이하인 정수입니다.

## Return 값 설명

hour 시 minute 분에 아날로그 시계의 시침과 분침이 이루는 각도를 소숫점 첫 번째 자리까지 표현한 문자열을 return 하세요.

- 단, 각도는 소수점 이하 첫째 자리까지 표현하세요.

## 예시

| hour | minute | return |
|------|--------|--------|
| 3    | 0      | "90.0" |

## 예시 설명

앞서 설명한 예와 같습니다.

## [4 차] 문제 10

## 문제 설명

자연수를 제공한 수는 제곱수, 세 제곱한 수는 세제곱 수라고 합니다. 예를 들어  $2^2 = 4$  는 제곱수,  $3^3 = 27$  은 세제곱수 입니다.

두 자연수 a, b 가 주어질 때 a 이상 b 이하인 자연수 중 소수의 제곱수와 세제곱수의 개수를 구하려 합니다. 예를 들어 a = 6, b = 30 일 때 소수의 제곱수는 [9, 25]로 2 개, 소수의 세제곱수는 [8, 27]로 2 개로 총 4 개입니다.

두 자연수  $a$ ,  $b$  가 매개변수로 주어질 때,  $a$  이상  $b$  이하인 제곱수와 세제곱수의 개수의 합을 return 하도록 solution 함수를 완성해 주세요.

#### 매개변수 설명

두 자연수  $a$ ,  $b$ 가 solution 함수의 매개변수로 주어집니다.

- $a$ ,  $b$ 는 각각 1 이상 1,000,000,000 이하인 자연수입니다.
- $a \leq b$ 인 경우만 입력으로 주어집니다.

#### Return 값 설명

$a$  이상  $b$  이하인 제곱수와 세제곱수의 개수의 합을 return 해 주세요.

#### 예시

| a | b  | return |
|---|----|--------|
| 6 | 30 | 4      |

#### 예시 설명

6 이상 30 이하인 수 중 소수의 제곱수는 다음과 같습니다.

$$* 3^2 = 9$$

$$* 5^2 = 25$$

소수의 세제곱 수는 다음과 같습니다.

$$* 2^3 = 8$$

$$* 3^3 = 27$$

따라서 4 를 return 하면 됩니다.