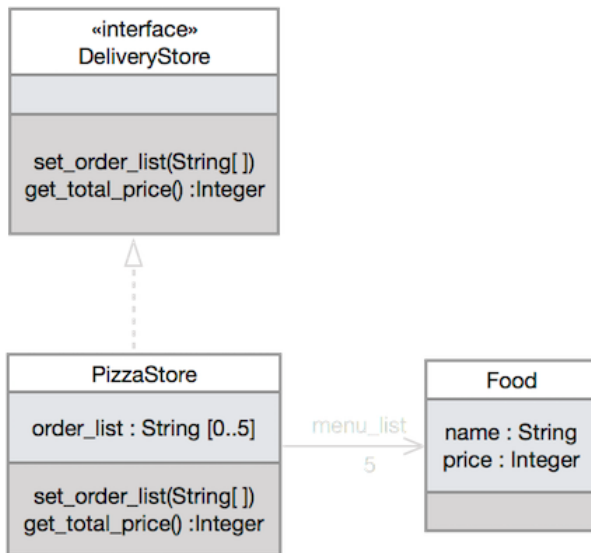


## [1 차] 문제 1

## 문제 설명

배달음식 전문점 운영을 위해 다음과 같이 DeliveryStore 인터페이스와 PizzaStore, Food 클래스를 작성했습니다.



## \* DeliveryStore :

- \* DeliveryStore 는 배달 음식점의 인터페이스입니다.
- \* 배달 음식점은 set\_order\_list 와 get\_total\_price 함수를 구현해야 합니다.
- \* set\_order\_list 함수는 주문 메뉴의 리스트를 매개변수로 받아 저장합니다.
- \* get\_total\_price 함수는 주문받은 음식 가격의 총합을 return 합니다.

## \* Food :

- \* Food 는 음식을 나타내는 클래스입니다.
- \* 음식은 이름(name)과 가격(price)으로 구성되어 있습니다.

## \* PizzaStore

\* PizzaStore 는 피자 배달 전문점을 나타내는 클래스이며 DeliveryStore 인터페이스를 구현합니다.

- \* menu\_list 는 피자 배달 전문점에서 주문 할 수 있는 음식의 리스트를 저장합니다.
- \* order\_list 는 주문 받은 음식들의 이름을 저장합니다.
- \* set\_order\_list 함수는 주문 메뉴를 받아 order\_list 에 저장합니다.
- \* get\_total\_price 함수는 order\_list 에 들어 있는 음식 가격의 총합을 return 합니다.

주문 메뉴가 들어 있는 리스트 `order_list` 가 매개변수로 주어질 때, 주문한 메뉴의 전체 가격을 `return` 하도록 `solution` 함수를 작성하려고 합니다. 위의 클래스 구조를 참고하여 주어진 코드의 빈칸을 적절히 채워 전체 코드를 완성해 주세요.

#### 매개변수 설명

주문 메뉴가 들어 있는 리스트 `order_list` 가 `solution` 함수의 매개변수로 주어집니다.

- ♦ `order_list` 의 길이는 1 이상 5 이하입니다.
- ♦ `order_list` 에는 주문하려는 메뉴의 이름들이 문자열 형태로 들어 있습니다.
- ♦ `order_list` 에는 같은 메뉴의 이름이 중복해서 들어 있지 않습니다.
- ♦ 메뉴의 이름과 가격은 `PizzaStore`의 생성자에서 초기화해 줍니다.

#### Return 값 설명

주문한 메뉴의 전체 가격을 `return` 해 주세요.

#### 예시

order_list	return
["Cheese","Pineapple","Meatball"]	51600

## [1 차] 문제 2

#### 문제 설명

해밍 거리(Hamming distance)란 같은 길이를 가진 두 개의 문자열에서 같은 위치에 있지만 서로 다른 문자의 개수를 뜻합니다. 예를 들어 두 2 진수 문자열이 "10010"과 "110"이라면, 먼저 두 문자열의 자릿수를 맞추기 위해 "110"의 앞에 0 두개를 채워 "00110"으로 만들어 줍니다. 두 2 진수 문자열은 첫 번째와 세 번째 문자가 서로 다르므로 해밍 거리는 2 입니다.

```
* `1`0`0`1 0
```

```
* `0`0`1`1 0
```

두 2 진수 문자열 `binaryA`, `binaryB` 의 해밍 거리를 구하려 합니다. 이를 위해 다음과 같이 간단히 프로그램 구조를 작성했습니다

1. 길이가 더 긴 2 진수 문자열의 길이를 구합니다.
2. 첫 번째 2 진수 문자열의 길이가 더 짧다면 문자열의 앞에 0 을 채워 넣어 길이를 맞춰줍니다.
3. 두 번째 2 진수 문자열의 길이가 더 짧다면 문자열의 앞에 0 을 채워 넣어 길이를 맞춰줍니다.

#### 4. 길이가 같은 두 2 진수 문자열의 해밍 거리를 구합니다.

두 2 진수 문자열 binaryA 와 binaryB 가 매개변수로 주어질 때, 두 2 진수의 해밍 거리를 return 하도록 solution 함수를 작성했습니다. 이때, 위 구조를 참고하여 중복되는 부분은 func\_a 라는 함수로 작성했습니다. 코드가 올바르게 동작할 수 있도록 빈칸을 알맞게 채워 전체 코드를 완성해 주세요.

##### 매개변수 설명

두 2진수 문자열 binaryA와 binaryB가 solution 함수의 매개변수로 주어집니다.

- \* binaryA의 길이는 1 이상 10 이하입니다.
- \* binaryA는 0 또는 1로만 이루어진 문자열이며, 0으로 시작하지 않습니다.
- \* binaryB의 길이는 1 이상 10 이하입니다.
- \* binaryB는 0 또는 1로만 이루어진 문자열이며, 0으로 시작하지 않습니다.

##### Return 값 설명

두 2 진수 문자열의 해밍 거리를 return 해 주세요.

##### 예시

binary A	binary B	return
10010	110	2

##### 예시 설명

두 2 진수의 자릿수는 각각 5 와 3 입니다. 자릿수를 맞추기 위해 "110" 앞에 0 두 개를 채워주면 "00110"이 됩니다. 이제 두 2 진수 문자열의 해밍 거리를 구하면 다음과 같습니다.

- ♦ 10010
- ♦ 00110

위와 같이 첫 번째와 세 번째 문자가 서로 다르므로, 해밍 거리는 2 가 됩니다.

## [1 차] 문제 3

##### 문제 설명

문자열 형태의 식을 계산하려 합니다. 식은 2 개의 자연수와 1 개의 연산자('+', '-', '\*') 중 하나로 이루어져 있습니다. 예를 들어 주어진 식이 "123+12"라면 이를 계산한 결과는 135 입니다.

문자열로 이루어진 식을 계산하기 위해 다음과 같이 간단히 프로그램 구조를 작성했습니다.

1. 주어진 식에서 연산자의 위치를 찾습니다.
2. 연산자의 앞과 뒤에 있는 문자열을 각각 숫자로 변환합니다.
3. 주어진 연산자에 맞게 연산을 수행합니다.

문자열 형태의 식 `expression` 이 매개변수로 주어질 때, 식을 계산한 결과를 `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

문자열 형태의 식 `expression`이 `solution` 함수의 매개변수로 주어집니다.

- `expression`은 연산자 1개와 숫자 2개가 결합한 형태입니다.
- 연산자는 '+', '-', '\*'만 사용됩니다.
- 숫자는 1 이상 10,000 이하의 자연수입니다.

#### Return 값 설명

`expression` 을 계산한 결과를 `return` 해 주세요.

계산 결과는 문자열로 변환하지 않아도 됩니다.

#### 예시

expression	return
"123+12"	135

#### 예시 설명

'+'를 기준으로 앞의 숫자는 123 이고 뒤의 숫자는 12 이므로 두 숫자를 더하면 135 가 됩니다.

## [1 차] 문제 4

#### 문제 설명

어느 누군가가 타임머신을 타고 과거로 가서 숫자 0 이 없는 수 체계를 전파했습니다. 역사가 바뀌어 이제 사람들의 의식 속엔 0 이란 숫자가 사라졌습니다. 따라서, 현재의 수 체계는 1, 2, 3, ..., 8, 9, 11, 12, ...와 같이 0 이 없게 바뀌었습니다.

0 을 포함하지 않은 자연수 num 이 매개변수로 주어질 때, 이 수에 1 을 더한 수를 return 하도록 solution 함수를 완성해 주세요.

#### 매개변수 설명

자연수 num이 solution 함수의 매개변수로 주어집니다.

- ♦ num은 1 이상 999,999,999,999,999 이하의 0을 포함하지 않는 자연수입니다.

#### Return 값 설명

자연수 num 에 1 을 더한 수를 return 해 주세요.

#### 예시

num	return
9949999	9951111

#### 예시 설명

9,949,999 에 1 을 더하면 9,950,000 이지만 0 은 존재하지 않으므로 9,951,111 이 됩니다.

## [1 차] 문제 5

#### 문제 설명

다음과 같이  $n \times n$  크기의 격자에 1 부터  $n \times n$  까지의 수가 하나씩 있습니다.

1	2	3
8	9	4
7	6	5

이때 수가 다음과 같은 순서로 배치되어 있다면 이것을 n-소용돌이 수라고 부릅니다. 소용돌이 수에서 1 행 1 열부터 n 행 n 열까지 대각선 상에 존재하는 수들의 합을 구해야 합니다.

1	2	3
8	9	4
7	6	5

위의 예에서 대각선 상에 존재하는 수의 합은 15 입니다.

격자의 크기 n 이 주어질 때 n-소용돌이 수의 대각선 상에 존재하는 수들의 합을 return 하도록 solution 함수를 완성해 주세요

**매개변수 설명**

격자의 크기  $n$ 이 `solution` 함수의 매개변수로 주어집니다.

- $n$ 은 1 이상 100 이하의 자연수입니다

**Return 값 설명**

$n$ -소용돌이 수의 대각선 상에 존재하는 수들의 합을 `return` 해 주세요.

**예시**

n	return
3	15
2	4

**예시 설명****예시 #1**

문제의 예와 같습니다.

**예시 #2.**

1 과 3 을 더하여 4 가 됩니다.

1	2
4	3

**[1 차] 문제 6****문제 설명**

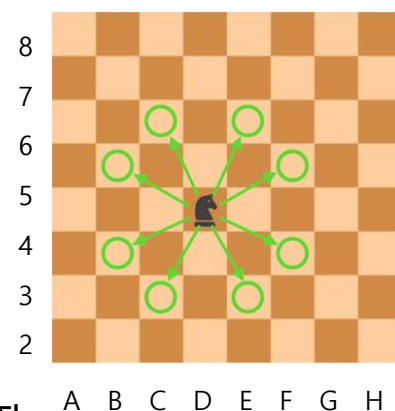
체스에서 나이트(knight)는 아래 그림과 같이 동그라미로 표시된 8 개의 방향 중 한 곳으로 한 번에 이동이 가능합니다.

단, 나이트는 체스판 밖으로는 이동할 수 없습니다.

체스판의 각 칸의 위치는 다음과 같이 표기합니다.

예를 들어, A 번줄과 1 번줄이 겹치는 부분은 'A1'이라고 합니다.

나이트의 위치 `pos` 가 매개변수로 주어질 때, 나이트를 한 번 움직여서 이동할 수 있는 칸은 몇 개인지 `return` 하도록 `solution` 함수를 완성해 주세요.



## 매개변수 설명

나이트의 위치 pos가 solution 함수의 매개변수로 주어집니다.

- ♦ pos는 A부터 H까지의 대문자 알파벳 하나와 1 이상 8이하의 정수 하나로 이루어진 두 글자 문자열입니다.
- ♦ 잘못된 위치가 주어지는 경우는 없습니다.

## Return 값 설명

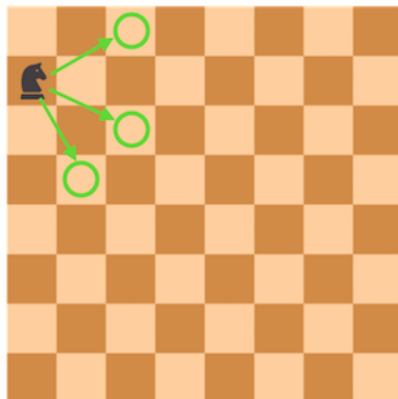
나이트를 한 번 움직여서 이동할 수 있는 칸의 개수를 return 해 주세요.

## 예시

pos	return
"A7"	3

## 예시 설명

나이트가 A7 위치에 있으면 아래 그림과 같이 왼쪽으로는 이동하지 못하고, 오른쪽으로는 맨 위를 제외한 나머지 세 칸으로 이동 가능합니다.



따라서 3 을 return 하면 됩니다.

## [1 차] 문제 7

## 문제 설명

오름차순으로 정렬되어 있는 두 리스트 arrA, arrB 를 하나의 리스트로 합치려 합니다. 단, 합친 후의 리스트도 오름차순으로 정렬되어 있어야 합니다.

예를 들어 arrA = [-2, 3, 5, 9], arrB = [0, 1, 5]인 경우 두 리스트를 오름차순으로 정렬된 하나의 리스트로 합치면 [-2, 0, 1, 3, 5, 5, 9]가 됩니다.

오름차순으로 정렬된 두 리스트 arrA 와 arrB 가 주어졌을 때, 두 리스트를 오름차순으로 정렬된 하나의 리스트로 합쳐서 return 하도록 solution 함수를 작성하려 합니다. 빈칸을 채워 전체 코드를 완성해 주세요.

#### 매개변수 설명

오름차순으로 정렬된 두 리스트 arrA와 arrB가 solution 함수의 매개변수로 주어집니다.

- arrA의 길이는 1 이상 200,000 이하입니다.
- arrA의 원소는 -1,000,000 이상 1,000,000 이하의 정수입니다.
- arrB의 길이는 1 이상 200,000 이하입니다.
- arrB의 원소는 -1,000,000 이상 1,000,000 이하의 정수입니다.

#### Return 값 설명

두 리스트 arrA, arrB 를 오름차순으로 정렬된 하나의 리스트로 합쳐서 return 해 주세요.

#### 예시

arrA	arrB	return
[-2, 3, 5, 9]	[0, 1, 5]	[-2, 0, 1, 3, 5, 5, 9]

## [1 차] 문제 8

#### 문제 설명

1 번부터 N 번까지 후보에 대해서 투표를 진행했습니다. 예를 들어 투표 결과가 [1, 5, 4, 3, 2, 5, 2, 5, 5, 4]라면 순서대로 [1 번, 5 번, 4 번, 3 번, 2 번, 5 번, 2 번, 5 번, 5 번, 4 번] 후보에 투표했음을 나타냅니다. 이때, 가장 많은 표를 받은 후보의 번호를 구하려고 합니다.

주어진 solution 함수는 후보의 수 N 과 투표를 진행한 결과가 담긴 리스트 votes 가 매개변수로 주어졌을 때, 가장 많은 표를 받은 후보의 번호를 return 하는 함수입니다. 그러나, 코드 일부가 잘못되어 있기 때문에 몇몇 입력에 대해서는 올바르게 동작하지 않습니다. 주어진 코드에서 한 줄만 변경해서 모든 입력에 대해 올바르게 동작하도록 수정하세요.

#### 매개변수 설명

후보의 수 N과 투표 결과가 들어 있는 리스트 votes가 solution 함수의 매개변수로 주어집니다.

- N은 1 이상 10 이하의 자연수입니다.



- ♦ votes의 길이는 1 이상 100 이하입니다.
- ♦ votes의 원소는 1 이상 N이하의 자연수입니다

#### Return 값 설명

가장 많은 표를 받은 후보의 번호를 리스트에 담아 return 해 주세요.

- ♦ 만약 가장 많은 표를 받은 후보가 2 개 이상이라면, 그 후보들의 번호를 모두 리스트에 담아 오름차순 정렬하여 return 해 주세요

#### 예시

N	votes	return
5	[1,5,4,3,2,5,2,5,5,4]	[5]
4	[1,3,2,3,2]	[2,3]

#### 예시 설명

.

#### 예시 #1

1 번부터 5 번까지 5 개의 후보가 있으며, 투표 결과는 [1, 5, 4, 3, 2, 5, 2, 5, 5, 4]입니다. 각 후보의 득표수는 다음과 같습니다.

- \* 1 번 후보가 1 표
- \* 2 번 후보가 2 표
- \* 3 번 후보가 1 표
- \* 4 번 후보가 2 표
- \* 5 번 후보가 4 표

이 경우 5 번 후보가 4 표로 가장 많은 표를 얻었습니다.

#### 예시 #2

1 번 후보가 1 표, 2 번 후보가 2 표, 3 번 후보가 2 표씩 받았습니다. 2 번과 3 번 후보가 공동으로 가장 많은 표를 받았기 때문에 [2, 3]을 오름차순 정렬하여 return 하면 됩니다.

## [1 차] 문제 9

#### 문제 설명

두 학생 A 와 B 는 계단 게임을 하였습니다.

계단 게임의 규칙은 아래와 같습니다.

1. 계단 제일 아래에서 게임을 시작합니다. (0 번째 칸)
2. 가위바위보를 합니다.
3. 이기면 계단 세 칸을 올라가고, 지면 한 칸을 내려가고, 비기면 제자리에 있습니다.
4. 계단 제일 아래에서 지면 제자리에 있습니다.
5. 2~4 과정을 열 번 반복합니다.

A 와 B 가 계단 게임을 완료한 후에 A 가 계단 위 몇 번째 칸에 있는지 파악하려고 합니다.

A 와 B 가 낸 가위바위보 기록이 순서대로 들어 있는 리스트 recordA 와 recordB 가 매개변수로 주어질 때, 게임을 마친 후의 A 의 위치를 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

A와 B가 낸 가위바위보 기록이 순서대로 들어 있는 리스트 recordA와 recordB가 매개변수로 주어집니다.

- recordA와 recordB의 원소는 0, 1, 2중 하나이고 순서대로 가위, 바위, 보를 의미합니다.
- recordA와 recordB의 길이는 10입니다.

#### Return 값 설명

solution 함수는 계단 게임을 마친 후에 A 가 계단 위 몇 번째 칸에 위치하는지를 return 합니다.

- 계단 제일 아래 칸은 0 번째 칸입니다.

#### 예시

recordA	recordB	return
[2,0,0,0,0,0,1,1,0,0]	[0,0,0,0,2,2,0,2,2,2]	14

#### 예시 설명

recordA	보	가위	가위	가위	가위	가위	바위	바위	가위	가위
recordB	가위	가위	가위	가위	보	보	가위	보	보	보
result	0	0	0	0	+3	+6	+9	+8	+11	+14

## [1 차] 문제 10

## 문제 설명

지난 연속된  $n$  일 동안의 주식 가격이 순서대로 들어 있는 리스트가 있습니다. 이때, 다음 규칙에 따라 주식을 사고 팔았을 때의 최대 수익을 구하려 합니다.

- $n$  일 동안 주식을 단 한 번 살 수 있습니다.
- $n$  일 동안 주식을 단 한 번 팔 수 있습니다.
- 주식을 산 날에 바로 팔 수는 없으며, 최소 하루가 지나야 팔 수 있습니다.
- 적어도 한 번은 주식을 사야 하며, 한 번은 팔아야 합니다.

주식을 팔 때는 반드시 이전에 주식을 사야 하며, 최대 수익은 양수가 아닐 수도 있습니다.

연속된  $n$  일 동안의 주식 가격이 순서대로 들어 있는 리스트 `prices` 가 매개변수로 주어질 때, 주식을 규칙에 맞게 한 번만 사고 팔았을 때 얻을 수 있는 최대 수익을 `return` 하도록 `solution` 함수를 작성했습니다. 그러나, 코드 일부분이 잘못되어 있기 때문에 코드가 올바르게 동작하지 않습니다. 주어진 코드에서 한 줄만 변경해서 모든 입력에 대해 올바르게 동작하도록 수정해 주세요.

## 매개변수 설명

연속된  $n$  일 동안의 주식 가격이 순서대로 들어 있는 리스트 `prices`가 `solution` 함수의 매개변수로 주어집니다

- `prices`의 길이는 2 이상 1,000,000 이하입니다.
- `prices`의 각 원소는 1 이상 1,000 이하의 자연수입니다.

## Return 값 설명

주식을 규칙에 맞게 한 번만 사고 팔았을 때 얻을 수 있는 최대 수익을 `return` 해 주세요.

## 예시

prices	return
[1, 2, 3]	2
[3, 1]	-2

## 예시 설명

## 예시 #1

연속된 3 일의 주가가 차례로 [1, 2, 3] 이며, 첫째 날에 주식을 사서 셋째 날에 팔면 수익은 2 이고, 이때가 최대입니다.

**예시 #2**

문제에서 설명한 것처럼 무조건 한 번은 매수하고, 한 번은 매도해야 합니다. 첫째 날에 매수하여 둘째 날에 매도하는 방법밖에 없기 때문에 수익은 -2, 즉 2 만큼 손실을 보게 됩니다.