

문제1

학생들의 시험 점수가 주어졌을 때, n번 학생이 몇 등인지 구하려 합니다.
학번은 0번부터 시작하며, 시험 점수는 학번순으로 주어집니다.

예를 들어 학생 네 명의 시험 점수가 다음과 같은 경우

학생 번호	수학 점수
0번	20
1번	60
2번	98
3번	59

3번 학생은 3등을 했습니다.

이를 위해 다음과 같이 프로그램 구조를 작성했습니다.

1. n번 학생의 점수를 변수에 저장합니다.
 2. 점수를 내림차순으로 정렬합니다.
 3. 리스트의 첫 번째 원소부터 마지막 원소까지 순회하며 n번 학생의 점수를 찾습니다.
3-1. 1번 단계에서 저장해둔 점수와 같은 점수를 찾으면 등수를 return 합니다.

학생들의 시험 점수가 번호순으로 들은 리스트 scores와 학번 n이 solution 함수의 매개변수로 주어질 때, n번 학생의 등수를 return 하도록 solution 함수를 작성하려 합니다. 위 구조를 참고하여 코드가 올바르게 동작하도록 빈칸에 주어진 func_a, func_b, func_c 함수와 매개변수를 알맞게 채워주세요.

매개변수 설명

학생들의 시험 점수가 번호순으로 들은 리스트 scores와 학번 n이 solution 함수의 매개변수로 주어집니다.

- * scores의 길이는 1 이상 100 이하입니다.
- * 점수는 0점 이상 100점 이하이며 동점자는 없습니다.
- * n은 0 이상 (scores의 길이 - 1) 이하의 정수입니다.

return 값 설명

n번 학생의 등수를 return 해주세요.

예시

scores	n	result
[20, 60, 98, 59]	3	3

예시 설명

- * 3번 학생의 점수는 59점입니다.
- * 점수를 내림차순으로 정렬하면 [98, 60, 59, 20]입니다.
- * 정렬된 점수에서 59는 3번째에 있습니다.

따라서 3번 학생은 3등입니다.

```
def func_a(scores, score):
    rank = 1
    for s in scores:
        if s == score:
            return rank
        rank += 1

    return 0

def func_b(arr):
    arr.sort(reverse=True)

def func_c(arr, n):
    return arr[n]

def solution(scores, n):
    score = func_@@@(@@@)
    func_@@@(@@@)
    answer = func_@@@(@@@)
    return answer

#아래는 테스트케이스 출력을 해보기 위한 코드입니다.
scores = [20, 60, 98, 59]
n = 3
ret = solution(scores, n)

#[실행] 버튼을 누르면 출력 값을 볼 수 있습니다.
print("solution 함수의 반환 값은", ret, "입니다.")
```

문제2

모 학교에서는 학기가 끝날 때마다 장학금을 줍니다. 이때 장학생이 몇 명인지 구하려고 합니다. 장학금을 주는 조건은 다음과 같습니다.

- 1. 이번 학기 성적이 80점 이상(100점 만점)이면서 석차가 상위 10% 이내인 학생
- 2. 이번 학기 성적이 80점 이상이면서 1등인 학생
- 3. 직전 학기 대비 성적이 가장 많이 오른 학생(여러 명인 경우 해당 학생 전부)

단, 동점인 학생들은 등수가 같으며, 중복 수혜는 불가합니다.

장학생이 몇 명인지 구하기 위해 다음과 같이 프로그램 구조를 작성했습니다.

1. 이번 학기 성적을 기준으로 학생별 석차를 구합니다.

2. 각 학생의 (이번 학기 성적 - 직전 학기 성적) 중 최댓값을 구합니다.

3. 아래 조건을 만족하는 학생을 발견하면, 장학생 수를 1 증가시킵니다.

3-1. 이번 학기 성적이 80점 이상이고, 석차가 상위 10% 이내인 경우

3-2. 또는 이번 학기 성적이 80점 이상이고, 석차가 1등인 경우

3-3. 또는 (이번 학기 성적 - 직전 학기 성적)이 2단계에서 구한 값과 같고, 그 값이 양수인 경우

4. 장학생 수를 return 합니다.

학생들의 이번 학기 성적을 담고 있는 리스트 `current_grade`, 직전 학기 성적을 담고 있는 리스트 `last_grade`가 매개 변수로 주어질 때, 장학생 수를 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

학생들의 이번 학기 성적, 지난 학기 성적이 들어있는 리스트 `current_grade`, `last_grade`가 `solution` 함수의 매개변수로 주어집니다.

- * `current_grade`의 길이와 `last_grade`의 길이는 같으며, 길이는 1 이상 200 이하입니다.
- * `current_grade`와 `last_grade`의 원소는 0 이상 100 이하인 정수입니다.

return 값 설명

장학생 수를 return 해주세요.

예시

<code>current_grade</code>	<code>last_grade</code>	<code>return</code>
[70, 100, 70, 80, 50, 95]	[35, 65, 80, 50, 20, 60]	3

예시 설명

학생 수가 10명보다 적으므로, 1등이 장학금을 받습니다.

직전 학기 대비 성적이 가장 많이 오른 학생은 다음과 같이 3명입니다.

- * 35 → 70점
- * 65 → 100점
- * 60 → 95점

이때, 두 번째 학생은 이번 학기 1등 장학금을 이미 받아 중복 수혜가 불가하고, 나머지 두 학생은 장학금을 받을 수 있습니다. 따라서 장학금을 받는 학생은 총 3명입니다.

```

def func_a(current_grade, last_grade, rank, max_diff_grade):
    arr_length = len(current_grade)
    count = 0

    for i in range(arr_length):
        if current_grade[i] >= 80 and rank[i] <= arr_length // 10:
            count += 1
        elif current_grade[i] >= 80 and rank[i] == 1:
            count += 1
        elif max_diff_grade > 0 and max_diff_grade == current_grade[i] - last_grade[i]:
            count += 1

    return count

def func_b(current_grade):
    arr_length = len(current_grade)
    rank = [1] * arr_length

    for i in range(arr_length):
        for j in range(arr_length):
            if current_grade[i] < current_grade[j]:
                rank[i] += 1

    return rank

def func_c(current_grade, last_grade):
    max_diff_grade = 1

    for i in range(len(current_grade)):
        max_diff_grade = max(max_diff_grade, current_grade[i] - last_grade[i])

    return max_diff_grade

def solution(current_grade, last_grade):
    rank = func_@@@(@@@)
    max_diff_grade = func_@@@(@@@)
    answer = func_@@@(@@@)
    return answer

```

아래는 테스트케이스 출력을 해보기 위한 코드입니다.

```

current_grade = [70, 100, 70, 80, 50, 95]
last_grade = [35, 65, 80, 50, 20, 60]
ret = solution(current_grade, last_grade)

```

#[실행] 버튼을 누르면 출력 값을 볼 수 있습니다.

```

print("solution 함수의 반환 값은", ret, "입니다.")

```

문제3

체조선수는 여러 심사위원의 점수 중 가장 높은 점수 하나와 가장 낮은 점수 하나를 제외하고 나머지 점수들의 평균을 계산하여 최종점수를 받습니다. 단, 이때 소수점 이하의 수는 버립니다.

예를 들어, A 선수가 10명의 심사위원에게 [35, 28, 98, 34, 20, 50, 85, 74, 71, 7]의 점수를 받았습니다. 가장 높은 점수인 98점과 가장 낮은 점수인 7점을 제외하고 나머지 8명의 심사위원 점수 평균을 구하면 49.625점입니다. 여기서 소수점 이하의 수를 버리면 49점입니다.

각 심사위원이 매긴 점수가 담긴 리스트 scores가 매개 변수로 주어질 때, 이 선수가 받은 점수를 return 하도록 solution 함수를 작성해주세요.

매개변수 설명

심사위원이 준 점수가 들어있는 리스트 scores가 solution 함수의 매개변수로 주어집니다.

- * scores의 길이는 3 이상 100 이하입니다.
- * 심사위원이 부여하는 점수의 범위는 0 이상 100 이하의 정수입니다.

return 값 설명

심사위원이 준 점수 중 가장 높은 점수와 가장 낮은 점수를 제외한 점수의 평균에서 소수점을 버린 값을 return 합니다.

예시

scores	return
[35, 28, 98, 34, 20, 50, 85, 74, 71, 7]	49
[1, 1, 1, 1, 1]	1

예시 설명

예시 #1

문제에 나온 예와 같습니다.

예시 #2

가장 높은 점수는 1점이며, 가장 낮은 점수도 1점입니다. 1점을 두 개 제외한 나머지 점수의 합은 3점이며, 평균은 1점이 됩니다.

#다음과 같이 import를 사용할 수 있습니다.

```
#import math
```

```
def solution(scores):
```

```
    answer = 0
```

```
    #여기에 코드를 작성해주세요.
```

```
    return answer
```

#아래는 테스트케이스 출력을 해보기 위한 코드입니다.

```
scores1 = [35, 28, 98, 34, 20, 50, 85, 74, 71, 7]
```

```
ret1 = solution(scores1)
```

#[실행] 버튼을 누르면 출력 값을 볼 수 있습니다.

```
print("solution 함수의 반환 값은", ret1, "입니다.")
```

```
scores2 = [1, 1, 1, 1, 1]
```

```
ret2 = solution(scores2)
```

#[실행] 버튼을 누르면 출력 값을 볼 수 있습니다.

```
print("solution 함수의 반환 값은", ret2, "입니다.")
```

문제4

영어 단어를 외우기 위해 단어를 반복하여 타이핑합니다. 그런데, 타이핑하고 보니 오타가 많습니다. 오타를 수정하려면 문자를 몇 개 바꿔야 하는지 구하려 합니다.

예를 들어, "CODE"라는 단어를 3번 타이핑했더니 ["CODE", "COED", "CDEO"] 가 적혀있었습니다.

1. "CODE"는 바르게 적혔습니다.
2. "COED"는 E와 D를 각각 D와 E로 바꾸면 됩니다.
3. "CDEO"는 D, E, O를 각각 O, D, E로 바꾸면 됩니다.

따라서 바꿔야 하는 문자는 총 5개입니다.

타이핑 한 단어를 담은 리스트 words와 원래 치려 한 단어인 word가 주어질 때 바꿔야 하는 문자 개수를 return 하도록 solution 함수를 완성해주세요.

매개변수 설명

타이핑한 단어를 담은 리스트 words와 원래 치려 한 단어인 word가 solution 함수의 매개변수로 주어집니다.

- * word는 10개 이하의 알파벳 대문자로만 이루어진 단어입니다.
- * words에 담긴 문자열의 길이는 word의 길이와 같으며 모든 문자열은 알파벳 대문자로만 이루어져 있습니다.
- * words에 담긴 문자열의 개수는 15 이하 자연수입니다.

return 값 설명

바꾸어야 하는 문자 수를 return 해주세요.

예시

words	word	return
["CODE", "COED", "CDEO"]	"CODE"	5

예시 설명

문제에 나온 예와 같습니다.

#다음과 같이 import를 사용할 수 있습니다.

```
#import math
```

```
def solution(words, word):
```

```
    count = 0
```

```
    #여기에 코드를 작성해주세요.
```

```
    return count
```

#아래는 테스트케이스 출력을 해보기 위한 코드입니다.

```
words = ["CODE", "COED", "CDEO"]
```

```
word = "CODE"
```

```
ret = solution(words, word)
```

#[실행] 버튼을 누르면 출력 값을 볼 수 있습니다.

```
print("solution 함수의 반환 값은", ret, "입니다.")
```

문제5

여행객들의 총 교통비를 계산하려고 합니다. 교통편은 "Bus", "Ship", "Airplane" 총 3가지입니다. 나이가 20살 이상이면 어른 요금을, 그렇지 않으면 어린이 요금을 받습니다. 각 교통편별 가격은 다음과 같습니다.

	어른	어린이
Bus	40,000원	15,000원
Ship	30,000원	13,000원
Airplane	70,000원	45,000원

여행객들이 10명 이상인 경우 연령에 따라 할인을 받습니다.

어른	어린이
10%	20%

여행객들의 나이를 담고 있는 리스트 member_age와 교통편인 transportation이 매개변수로 주어질 때, 총 교통비를 return 하도록 solution 함수를 작성하려 합니다. 빈칸을 채워 전체 코드를 완성해주세요.

매개변수 설명

여행객들의 나이를 담고 있는 리스트 member_age와 교통편인 transportation이 solution 함수의 매개변수로 주어 집니다.

- * member_age의 길이는 1 이상 1,000 이하입니다.
- * member_age의 원소는 1 이상 100 이하입니다.
- * transportation은 "Bus" 또는 "Ship" 또는 "Airplane" 입니다.

return값 설명

교통비 총액을 return 해주세요.

예시

member_age	transportation	return
[13, 33, 45, 11, 20]	"Bus"	150000
[25, 11, 27, 56, 7, 19, 52, 31, 77, 8]	"Ship"	203600

예시 설명

예시 #1

여행객들의 나이는 [13, 33, 45, 11, 20]이고 교통수단으로는 "Bus"를 이용하려합니다. 나이가 20살 미만인 인원은 2명으로 어린이 요금은 30,000원이고 나이가 20살 이상인 인원은 3명으로 어른 요금은 120,000원입니다. 따라서 교통비 총액은 150,000원입니다.

예시 #2

여행객들의 나이는 [25, 11, 27, 56, 7, 19, 52, 31, 77, 8]이고 교통수단으로는 "Ship"을 이용하려합니다. 여행객들이 10명 이상이므로 교통비가 할인됩니다. 나이가 20살 미만인 인원은 4명으로 어린이 요금은 41,600원이고 나이가 20살 이상인 인원은 6명으로 어른 요금은 162,000원입니다. 따라서 교통비 총액은 203,600원입니다.

```
def solution(member_age, transportation):
    if transportation == 'Bus':
        adult_expense = 40000
        child_expense = 15000
    elif transportation == 'Ship':
        adult_expense = 30000
        child_expense = 13000
    elif transportation == 'Airplane':
        adult_expense = 70000
        child_expense = 45000

    if len(member_age) >= 10:
        adult_expense = @@@
        child_expense = @@@

    total_expenses = 0

    for age in member_age:
        if @@@:
            total_expenses += adult_expense
        else:
            total_expenses += child_expense

    return total_expenses
```

#아래는 테스트케이스 출력을 해보기 위한 코드입니다.

```
member_age1 = [13, 33, 45, 11, 20]
transportation1 = "Bus"
ret1 = solution(member_age1, transportation1)
```

#[실행] 버튼을 누르면 출력 값을 볼 수 있습니다.

```
print("solution 함수의 반환 값은", ret1, "입니다.")
```

```
member_age2 = [25, 11, 27, 56, 7, 19, 52, 31, 77, 8]
transportation2 = "Ship"
ret2 = solution(member_age2, transportation2)
```

#[실행] 버튼을 누르면 출력 값을 볼 수 있습니다.

```
print("solution 함수의 반환 값은", ret2, "입니다.")
```

문제6

타일을 'R', 'G', 'B' 색으로 칠하려 합니다. R 색으로는 3칸을 한 번에, G 색으로는 2칸을 한 번에 칠할 수 있으며, B 색으로는 1칸을 칠할 수 있습니다. 색은 R, G, B 순서로 한 번씩 번갈아 가면서 사용해야 하며, 타일의 길이를 넘겨서 칠할 수는 없습니다.

예를 들어, 타일 길이가 11이면 "RRRGGBRRRG"의 색으로 칠할 수 있습니다.

타일 길이가 매개변수 `tile_length`로 주어질 때, 타일을 색칠한 순서를 문자열로 return하는 `solution` 함수를 작성하려 합니다. 빈칸을 채워 전체 코드를 완성해주세요.

순서에 맞게 타일을 칠할 수 없다면 -1을 return 해주세요.

매개변수 설명

타일 길이 `tile_length`가 `solution` 함수의 매개변수로 주어집니다.

* `tile_length`는 1,000 이하의 자연수입니다.

return 값 설명

타일을 색칠한 순서를 문자열로 return 해주세요.

* 순서에 맞게 타일을 칠할 수 없다면 -1을 return 해주세요.

예시

tile_length	answer
11	"RRRGGBRRRG"
16	-1

예시 설명

예시 #1

다음과 같은 순서로 타일을 칠하면 됩니다.

- * 'R' - 3장
- * 'G' - 2장
- * 'B' - 1장
- * 'R' - 3장
- * 'G' - 2장

따라서 "RRRGGBRRRG"를 return 하면 됩니다.

예시 #2

타일 길이는 16입니다.

'R', 'G', 'B' 순서대로 타일을 색칠하면 다음과 같습니다.

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

15번째 타일까지 칠하니 타일이 한 칸 남았습니다. G 색으로는 두 칸을 칠해야 하므로 타일을 더 칠할 수 없습니다. 따라서 "-1"을 return 합니다.

```
def solution(tile_length):
    answer = ''
    com = 'RRRGGB'

    if tile_length%6 == 1 or tile_length%6 == 2 or @@@:
        answer = '-1'
    else:
        for i in range(tile_length):
            answer += com[i % 6]

    return answer
```

#아래는 테스트케이스 출력을 해보기 위한 코드입니다.

```
tile_length1 = 11
ret1 = solution(tile_length1)
```

#[실행] 버튼을 누르면 출력 값을 볼 수 있습니다.

```
print("solution 함수의 반환 값은 ", ret1, " 입니다.")
```

```
tile_length2 = 16
```

```
ret2 = solution(tile_length2)
```

#[실행] 버튼을 누르면 출력 값을 볼 수 있습니다.

```
print("solution 함수의 반환 값은 ", ret2, " 입니다.")
```

문제7

주스 1잔을 만들려면 사과 3개와 당근 1개가 필요합니다. 그런데 키우는 토끼에게 먹이를 주기 위해 사과와 당근 종류에 상관없이 k개를 빼놓으려고 합니다. 주스는 최대한 많이 만들수록 좋습니다.

사과 개수 num_apple과 당근 개수 num_carrot, 토끼에게 줄 먹이 개수 k가 주어질 때 주스를 최대 몇 잔 만들 수 있는지 return 하도록 solution 함수를 작성했습니다. 그러나, 코드 일부가 잘못되어있기 때문에, 몇몇 입력에 대해서는 올바르게 동작하지 않습니다. 주어진 코드에서 `_**한 줄**_`만 변경해서 모든 입력에 대해 올바르게 동작하도록 수정하세요.

매개변수 설명

사과 개수 num_apple과 당근 개수 num_carrot, 토끼에게 줄 먹이 개수 k가 solution 함수의 매개변수로 주어집니다.

- * 사과 개수 num_apple과 당근 개수 num_carrot는 0 이상 200 이하인 정수입니다.
- * 토끼에게 줄 먹이 개수 k는 0 이상 `num_apple+num_carrot` 이하인 정수입니다.

return 값 설명

주스를 최대 몇 잔 만들 수 있는지 return 해주세요.

예시

num_apple	num_carrot	k	return
5	1	2	1
10	5	4	2

예시 설명

예시 #1

사과 2개를 먹이로 주면 사과 3개, 당근 1개가 남습니다. 남은 재료로 주스를 1잔 만들 수 있습니다.

예시 #2

사과 2개와 당근 2개를 먹이로 주면 사과 8개, 당근 3개가 남습니다. 남은 재료로 주스를 2잔 만들 수 있습니다.

```
def solution(num_apple, num_carrot, k):
    answer = 0

    if num_apple < num_carrot * 3:
        answer = num_apple // 3
    else:
        answer = num_carrot

    num_apple -= answer * 3
    num_carrot -= answer

    i = 0

    while k - (num_apple + num_carrot + i) > 0:
        if i % 4 == 0:
            answer += 1
        i = i + 1

    return answer
```

아래는 테스트케이스 출력을 해보기 위한 코드입니다.

아래 코드는 잘못된 부분이 없으니, solution함수만 수정하세요.

```
num_apple1 = 5
num_carrot1 = 1
k1 = 2
ret1 = solution(num_apple1, num_carrot1, k1)
```

[실행] 버튼을 누르면 출력 값을 볼 수 있습니다.

```
print("solution 함수의 반환 값은", ret1, "입니다.")
```

```
num_apple2 = 10
num_carrot2 = 5
k2 = 4
ret2 = solution(num_apple2, num_carrot2, k2)
```

[실행] 버튼을 누르면 출력 값을 볼 수 있습니다.

```
print("solution 함수의 반환 값은", ret2, "입니다.")
```


문제8

A씨가 하루에 TV를 두 대 이상 트는 시간을 알아내려합니다. A씨는 매일 세 프로그램을 시청합니다. 프로그램 방송 시간이 겹칠 때는 TV를 여러 대 켜서 모든 프로그램을 봅니다.

예를 들어 두 프로그램 방송 시간대가 겹치면 TV를 두 대 켜고, 세 프로그램 방송 시간이 겹치면 TV를 세 대 켵니다.

세 프로그램 방영 시작 시각과 끝 시각이 담긴 2차원 리스트 `programs`가 매개변수로 주어질 때, 하루에 TV를 2대 이상 트는 총 시간을 `return` 하도록 `solution` 함수를 작성했습니다. 그러나, 코드 일부분이 잘못되어있기 때문에, 몇몇 입력에 대해서는 올바르게 동작하지 않습니다. 주어진 코드에서 `_**한 줄**_`만 변경해서 모든 입력에 대해 올바르게 동작하도록 수정하세요.

매개변수 설명

3개의 프로그램이 시작되는 시각과 끝나는 시각이 담긴 2차원 리스트 `programs`가 `solution` 함수의 매개변수로 주어 집니다.

- * `programs`의 행 길이는 항상 3입니다.
- * `programs` 리스트의 각 원소는 각 프로그램의 방송 시간이 [시작 시각, 끝 시각] 형태로 들어있습니다.
- * 프로그램의 시작 시각과 끝 시각은 0 이상 24 이하의 정수입니다.
- * 프로그램의 시작 시각은 끝 시각보다 항상 빠릅니다.

return 값 설명

A 씨가 TV를 2대 이상 트는 총 시간을 `return` 해주세요.

예시

programs	return
[[1, 6], [3, 5], [2, 8]]	4

예시 설명

2대의 TV를 트는 시간이 2시부터 3시까지, 5시부터 6시까지 총 2시간이고
3대의 TV를 트는 시간이 3시부터 5시까지 총 2시간입니다.
따라서 TV를 2대 이상 트는 시간은 총 4시간입니다.

```
def solution(programs):
    answer = 0
    used_tv = [0] * 25

    for program in programs:
        for i in range(program[0], program[1]):
            used_tv[i] = used_tv[i] + 1

    for i in used_tv:
        if i >= 1:
            answer = answer + 1

    return answer
```

아래는 테스트케이스 출력을 해보기 위한 코드입니다.

아래 코드는 잘못된 부분이 없으니, solution함수만 수정하세요.

```
programs = [[1, 6], [3, 5], [2, 8]]
```

```
ret = solution(programs)
```

#[실행] 버튼을 누르면 출력 값을 볼 수 있습니다.

```
print("solution 함수의 반환 값은", ret, "입니다.")
```

문제9

관공서 주차장에서는 차량 2부제를 실시합니다. 차량 2부제는 차량 번호 끝자리가 홀수인 차량은 홀수 일에만, 짝수인 차량은 짝수 일에만 주차장에 들어올 수 있도록 하는 제도입니다.

며칠인지를 나타내는 day와 그날 주차장에 들어오려고 하는 차들의 번호를 담고 있는 리스트 numbers가 매개변수로 주어질 때, 주차장에 들어올 수 있는 차량의 수를 return 하도록 solution 함수를 작성했습니다. 그러나, 코드 일부가 잘못되어있기 때문에, 코드가 올바르게 동작하지 않습니다. 주어진 코드에서 한 줄만 변경해서 모든 입력에 대해 올바르게 동작하도록 수정해주세요.

단, 중복되는 번호판 번호는 주어지지 않습니다.

매개변수 설명

며칠인지를 나타내는 day와 그날 주차장에 들어오려고 하는 차들의 번호를 담고 있는 리스트 numbers가 solution 함수의 매개변수로 주어집니다.

- * day는 1 이상 31이하의 자연수입니다.
- * numbers의 길이는 1 이상 9,000 이하입니다.
- * numbers의 원소는 1,000 이상 9,999 이하의 자연수입니다.
- * numbers에는 차량 번호가 중복되어 들어있지 않습니다.

return값 설명

주차장에 들어올 수 있는 차량의 수를 return 해주세요.

예시

day	numbers	return
17	[3285, 1724, 4438, 2988, 3131, 2998]	2

예시 설명

17일은 홀수 일이므로 홀수 번호 차량인 1724, 3131 총 2대의 차량만 주차장에 들어올 수 있습니다.

```
def solution(day, numbers):  
    count = 0  
    for number in numbers:  
        if number%2 != day%2:  
            count += 1  
    return count
```

아래는 테스트케이스 출력을 해보기 위한 코드입니다.

아래 코드는 잘못된 부분이 없으니, solution함수만 수정하세요.

```
day = 17  
numbers = [3285, 1724, 4438, 2988, 3131, 2998]  
ret = solution(day, numbers)
```

#[실행] 버튼을 누르면 출력 값을 볼 수 있습니다.

```
print("solution 함수의 반환 값은", ret, "입니다.")
```

문제10

리스트 원소인 자신을 2로 나눈 값이 리스트에 있는 수의 개수를 구하려고 합니다.

예를 들어, 리스트가 [4, 8, 3, 6, 7]인 경우, $6/2 = 3$, $8/2 = 4$ 이므로 자신을 2로 나눈 값이 리스트에 들어있는 수의 개수는 총 2개입니다.

숫자가 담긴 리스트 arr가 주어졌을 때, 자신을 2로 나눈 값이 리스트에 들어있는 수가 몇 개인지 return 하는 solution 함수를 작성했습니다. 그러나, 코드 일부분이 잘못되어있기 때문에, 코드가 올바르게 동작하지 않습니다. 주어진 코드에서 한 줄만 변경해서 모든 입력에 대해 올바르게 동작하도록 수정해주세요.

매개변수 설명

숫자가 담긴 리스트 arr가 solution 함수의 매개변수로 주어집니다.

- * arr의 크기는 1 이상 1,000 이하입니다.
- * arr의 원소는 1,000 이하의 자연수입니다.

return 값 설명

자신을 2로 나눈 값이 리스트에 들어있는 수가 몇 개인지 return 해주세요.

예시

arr	return
[4, 8, 3, 6, 7]	2

예시 설명

$4/2=2$ 이고, 2는 리스트에 없습니다.

$8/2=4$ 이고 4는 4 이고 4는 리스트에 존재합니다.

$3/2=1.5$ 이고 1.5는 리스트에 없습니다.

$6/2=3$ 이고 3은 리스트에 존재합니다.

$7/2=3.5$ 이고 3.5는 리스트에 없습니다.

따라서 자신을 2로 나눈 값이 리스트에 들어있는 수가 총 2개이므로 2를 return 합니다.

```
def solution(arr):  
    answer = 0  
  
    for i in arr:  
        for i/2 in arr:  
            answer += 1  
  
    return answer
```

아래는 테스트케이스 출력을 해보기 위한 코드입니다.

아래 코드는 잘못된 부분이 없으니, solution함수만 수정하세요.

```
arr = [4, 8, 3, 6, 7]
```

```
ret = solution(arr)
```

#[실행] 버튼을 누르면 출력 값을 볼 수 있습니다.

```
print("solution 함수의 반환 값은", ret, "입니다.")
```