

바르게 살자

코딩테스트 week2

로또의 최고 순위와 최저 순위

순위	당첨 내용
1	6개 번호가 모두 일치
2	5개 번호가 일치
3	4개 번호가 일치
4	3개 번호가 일치
5	2개 번호가 일치
6(낙첨)	그 외

제한사항

- lottos는 길이 6인 정수 배열입니다.
- lottos의 모든 원소는 0 이상 45 이하인 정수입니다.
 - 0은 알아볼 수 없는 숫자를 의미합니다.
 - 0을 제외한 다른 숫자들은 lottos에 2개 이상 담겨있지 않습니다.
 - lottos의 원소들은 정렬되어 있지 않을 수도 있습니다.
- win_nums는 길이 6인 정수 배열입니다.
- win_nums의 모든 원소는 1 이상 45 이하인 정수입니다.
 - win_nums에는 같은 숫자가 2개 이상 담겨있지 않습니다.
 - win_nums의 원소들은 정렬되어 있지 않을 수도 있습니다.

당첨 번호	31	10	45	1	6	19	결과
최고 순위 번호	<u>31</u>	0→10	44	1	0→6	25	4개 번호 일치, 3등
최저 순위 번호	<u>31</u>	0→11	44	1	0→7	25	2개 번호 일치, 5등

- 순서와 상관없이, 구매한 로또에 당첨 번호와 일치하는 번호가 있으면 맞힌 걸로 인정됩니다.
- 알아볼 수 없는 두 개의 번호를 각각 10, 6이라고 가정하면 3등에 당첨될 수 있습니다.
 - 3등을 만드는 다른 방법들도 존재합니다. 하지만, 2등 이상으로 만드는 것은 불가능합니다.
- 알아볼 수 없는 두 개의 번호를 각각 11, 7이라고 가정하면 5등에 당첨될 수 있습니다.
 - 5등을 만드는 다른 방법들도 존재합니다. 하지만, 6등(낙첨)으로 만드는 것은 불가능합니다.

민우가 구매한 로또 번호를 담은 배열 lottos, 당첨 번호를 담은 배열 win_nums가 매개변수로 주어집니다. 이때, 당첨 가능한 최고 순위와 최저 순위를 차례대로 배열에 담아서 return 하도록 solution 함수를 완성해주세요.

로또의 최고 순위와 최저 순위

CODE

```
def solution(lottos, win_nums):
    cnt = 0
    dict = {0:6, 1:6, 2:5, 3:4, 4:3, 5:2, 6:1}
    joker = lottos.count(0)

    for num in lottos:
        if num in win_nums:
            cnt += 1

    print(joker, cnt)

    if cnt == 0 != joker:
        joker -= 1

    answer = [dict[cnt]-joker, dict[cnt]]

    return answer
```

BEST CODE

```
def solution(lottos, win_nums):

    rank=[6,6,5,4,3,2,1]

    cnt_0 = lottos.count(0)
    ans = 0
    for x in win_nums:
        if x in lottos:
            ans += 1
    return rank[cnt_0 + ans],rank[ans]
```

폰켓몬

당신은 폰켓몬을 잡기 위한 오랜 여행 끝에, 홍 박사님의 연구실에 도착했습니다. 홍 박사님은 당신에게 자신의 연구실에 있는 총 N 마리의 폰켓몬 중에서 N/2마리를 가져가도 좋다고 했습니다.

홍 박사님 연구실의 폰켓몬은 종류에 따라 번호를 붙여 구분합니다. 따라서 같은 종류의 폰켓몬은 같은 번호를 가지고 있습니다. 예를 들어 연구실에 총 4마리의 폰켓몬이 있고, 각 폰켓몬의 종류 번호가 [3번, 1번, 2번, 3번]이라면 이는 3번 폰켓몬 두 마리, 1번 폰켓몬 한 마리, 2번 폰켓몬 한 마리가 있음을 나타냅니다. 이때, 4마리의 폰켓몬 중 2마리를 고르는 방법은 다음과 같이 6가지가 있습니다.

1. 첫 번째(3번), 두 번째(1번) 폰켓몬을 선택
2. 첫 번째(3번), 세 번째(2번) 폰켓몬을 선택
3. 첫 번째(3번), 네 번째(3번) 폰켓몬을 선택
4. 두 번째(1번), 세 번째(2번) 폰켓몬을 선택
5. 두 번째(1번), 네 번째(3번) 폰켓몬을 선택
6. 세 번째(2번), 네 번째(3번) 폰켓몬을 선택

NC_2

nums	result
[3,1,2,3]	2
[3,3,3,2,2,4]	3
[3,3,3,2,2,2]	2

폰켓몬

CODE

```
def solution(nums):  
    choice = len(nums)//2  
    answer = max_class = len(set(nums))  
    if choice <= max_class:  
        answer = choice  
  
    return answer
```

BEST CODE

```
def solution(ls):  
    return min(len(ls)/2, len(set(ls)))
```

4주차 직업군 추천하기

점수	SI	CONTENTS	HARDWARE	PORTAL	GAME
5	JAVA	JAVASCRIPT	C	JAVA	C++
4	JAVASCRIPT	JAVA	C++	JAVASCRIPT	C#
3	SQL	PYTHON	PYTHON	PYTHON	JAVASCRIPT
2	PYTHON	SQL	JAVA	KOTLIN	C
1	C#	C++	JAVASCRIPT	PHP	JAVA

table	languages	preference	result
["SI JAVA JAVASCRIPT SQL PYTHON C#", "CONTENTS JAVASCRIPT JAVA PYTHON SQL C++", "HARDWARE C C++ PYTHON JAVA JAVASCRIPT", "PORTAL JAVA JAVASCRIPT PYTHON KOTLIN PHP", "GAME C++ C# JAVASCRIPT C JAVA"]	["PYTHON", "C++", "SQL"]	[7, 5, 5]	"HARDWARE"
["SI JAVA JAVASCRIPT SQL PYTHON C#", "CONTENTS JAVASCRIPT JAVA PYTHON SQL C++", "HARDWARE C C++ PYTHON JAVA JAVASCRIPT", "PORTAL JAVA JAVASCRIPT PYTHON KOTLIN PHP", "GAME C++ C# JAVASCRIPT C JAVA"]	["JAVA", "JAVASCRIPT"]	[7, 5]	"PORTAL"

	PYTHON	C++	SQL
선호도	7	5	5

아래 사진은 개발자가 선호하는 언어의 직업군 언어 점수 를 나타낸 표입니다.

	SI	CONTENTS	HARDWARE	PORTAL	GAME
PYTHON	2	3	3	3	0
C++	0	1	4	0	5
SQL	3	2	0	0	0
직업군별 점수 식	$7 \times 2 + 5 \times 0 + 5 \times 3$	$7 \times 3 + 5 \times 1 + 5 \times 2$	$7 \times 3 + 5 \times 4 + 5 \times 0$	$7 \times 3 + 5 \times 0 + 5 \times 0$	$7 \times 0 + 5 \times 5 + 5 \times 0$
점수 총합	29	36	41	21	25

따라서 점수 총합이 41로 가장 높은 "HARDWARE" 를 return 해야 합니다.

	JAVA	JAVASCRIPT
선호도	7	5

아래 사진은 개발자가 선호하는 언어의 직업군 언어 점수 를 나타낸 표입니다.

	SI	CONTENTS	HARDWARE	PORTAL	GAME
JAVA	5	4	2	5	1
JAVASCRIPT	4	5	1	4	3
직업군별 점수 식	$7 \times 5 + 5 \times 4$	$7 \times 4 + 5 \times 5$	$7 \times 2 + 5 \times 1$	$7 \times 5 + 5 \times 4$	$7 \times 1 + 5 \times 3$
점수 총합	55	53	19	55	22

점수 총합이 55로 가장 높은 직업군은 "SI" 와 "PORTAL"입니다.

따라서 사전 순으로 먼저 오는 "PORTAL" 을 return 해야 합니다.

4주차 직업군 추천하기

CODE

```
def solution(table, languages, preference):
    answer = dict()

    for idx, job in enumerate(table):
        table_split = job.split(" ")
        score = [0] * 6

        for lang, weight in zip(languages, preference):
            if lang in table_split:
                k = table_split.index(lang)
                score[k] = (6-k)*weight

        print(sum(score))
        answer[table_split[0]] = sum(score)

    answer = sorted(answer.items(), key=lambda x: (-x[1], x[0]))
    return answer[0][0]
```

BEST CODE

```
def solution(table, languages, preference):
    score = {}
    for t in table:
        for lang, pref in zip(languages, preference):
            if lang in t.split():
                score[t.split()[0]] = score.get(t.split()[0], 0) + (6 -
t.split().index(lang)) * pref

    return sorted(score.items(), key = lambda item: [-item[1],
item[0]])[0][0]
```