

완득하지 못한 선수

leo kiki eden	eden, kiki	동명이인인 경우		
/		mislav stanko	mislav ana	stanko ana mislav
/		1	← 함께	
.....				
0 / /	0인 것만			/
.....				
		2		
.....				
		2 1	1	2인 것만

1이 아닌 것만 남았으면 됨

- 코딩 구조
1. 동명이인 체크 코딩
 2. 완득자 체크 코딩
 3. 완득하지 못한 선수 반환 코딩

1. 동명이인 체크 코딩

HashMap이 [이름, 점수]로 넣으면 됨
 이름이 HashMap에 없으면 추가하고
 있으면 점수를 +1 해주면 동명이인은 이 때에 됨

eg. ⇒ leo kiki eden mislav stanko ana
 0 0 0 1 0 0

동명이인

2. 완득자 체크 코딩

이름을 HashMap에서 찾아 점수 +1

3. 완득하지 못한 선수 반환

완득하지 못한 선수는 0이므로 값이 0인 선수 이름 반환

동명이인은 1이 아닌 값이 있는 사람 반환

leo leo leo ana ana leo leo ana ana

leo ana ⇒ leo ana
 2 / 4 3

문자 변경

참여자 - 만크라

$$1-1=0$$

$$2-2=0$$

$$1-1=0$$

$$3-2 \neq 1 \rightarrow \text{만크}$$

$$1-1=0$$

$$2-1 \neq 1 \rightarrow \text{만크}$$

1. 동등성인 체크

그대로 진행

2. 만크라 체크

가중치 +2, -2 해시

만크라는 결국 0이 되고

만크라지 못한 사람은 값이 1.

3. 만크라지 못한 사람 반환

값이 1인 키를 반환하면 됨

⇒ `map.entrySet()` 메서드

키-값 쌍을 `map.Entry`로 넣어

값이 1인 키 반환.

1. 선수인들 - 완주여부를 담은 HashMap 선언

```
Map<String, Integer> hashMap = new HashMap<>();
```

2. 동영이인 체크 로직

```
for (String name : participant) {
```

2-1. 참가자가 hashMap에 들어있다면 동영이인이라면 +1

```
if (hashMap.containsKey(name)) {
```

```
    hashMap.put(name, hashMap.get(name) + 1);
```

2-2. 들어있지 않다면 값을 1로 초기화

```
    } else {
```

```
        hashMap.put(name, 1);
```

```
    }
```

```
}
```

3. 완주자 체크 로직

```
for (String name : completion) {
```

3-1. 완주자 명단에 있는 name이 hashMap에 있다면 -1

```
if (hashMap.containsKey(name)) {
```

```
    hashMap.put(name, hashMap.get(name) - 1);
```

```
}
```

```
}
```

4. 완주하지 못한 선수 반환

String result;

4-1. Map의 entrySet() 메서드와 entry를 사용해 키, 값 조회

```
for (Map.Entry<String, Integer> entry : hashMap.entrySet()) {
```

```
    if (entry.getValue() == 1) { // 값이 1이면 완주 못한 의미
```

```
        result = entry.getKey();
```

```
        break;
```

```
    }
```

```
}
```

```
return result;
```