

# Report

프로그래머스 완주하지 못한 선수 문  
제(42576)



과목 : 코딩테스트지도

담당교수 : 이승진 교수님

학부 : IT융합자율학부

학번 : 202114136

이름 : 장준희

제출일 : 2025-03-14

## 문제 설명

수많은 마라톤 선수들이 마라톤에 참여하였습니다. 단 한 명의 선수를 제외하고는 모든 선수가 마라톤을 완주하였습니다.

마라톤에 참여한 선수들의 이름이 담긴 배열 `participant`와 완주한 선수들의 이름이 담긴 배열 `completion`이 주어질 때, 완주하지 못한 선수의 이름을 `return` 하도록 `solution` 함수를 작성해주세요.

## 제한사항

마라톤 경기에 참여한 선수의 수는 1명 이상 100,000명 이하입니다.

`completion`의 길이는 `participant`의 길이보다 1 작습니다.

참가자의 이름은 1개 이상 20개 이하의 알파벳 소문자로 이루어져 있습니다.

참가자 중에는 동명이인이 있을 수 있습니다.

## 1차 시도

`arrayList`를 통해 차집합을 구해보겠습니다.

```
participantArray.addAll(Arrays.asList(participant));  
completionArray.addAll(Arrays.asList(completion));
```

```
participantArray.removeAll(completionArray);
```

하지만 이 방법을 사용하니 동명이인까지 모두 삭제되는 문제가 생깁니다.

(동명이인이 있으면 빈 리스트가 나옴)

## 2차 시도

동명이인 문제를 해결하기 위해 HashMap을 활용했습니다.

각 참가자의 등장 횟수를 value로 저장합니다.

```
professor.put(p, professor.getOrDefault(p, 0) + 1);
```

기존에 존재하는 이름이면 +1을 증가한다.

처음 등장하는 이름이면 초기값 0 + 1 증가

## 완주 카운트

완주한 사람은 카운트에서 1을 뺍니다.

이렇게 하는 이유는 동명이인은 카운트에서 남기기 위함이다.

```
professor.put(c, professor.get(c) - 1);
```

Key를 불러와 value가 0이 아닌(0보다 큰) 사람을 걸러낸다.

```
if(professor.get(s) != 0) {  
    answer = s;
```

그러면 미완주자를 찾을 수 있습니다.

## 배운 점

처음에는 단순히 차집합을 이용하면 된다고 생각했지만, 동명이인을 고려하지 못한 실수를 했습니다.

이를 해결하기 위해 HashMap을 활용하여 각 참가자의 개수를 저장하고, 완주 여부를 반영하는 방식으로 해결할 수 있었습니다.

이제 동명이인이 존재하더라도  $O(N)$  시간 복잡도로 효율적인 해결이 가능합니다.