1_완주하지 못한 선수

first code

```
def solution(participant, completion):
    hashTable = dict()
    for part in participant :
        if part in hashTable :
            hashTable[part] += 1
        else :
            hashTable[part] = 1
    for com in completion :
        hashTable[com] -= 1
    return list(filter(lambda key : hashTable[key] != 0, hashTable)).pop()
```

I use dictionary structure because using list can cause the fault for efficiency test. (1 ≤ N ≤ 100,000 but implementation using list is O(N^2)

other code

```
import collections

def solution(participant, completion):
    answer = collections.Counter(participant) - collections.Counter(completion)
    return list(answer.keys())[0]
```

This implementation uses Counter object.

Counter object can subtract each other.

best code

```
def solution(participant, completion) :
    tmp = 0
    hashTable = dict()
    for part in participant :
        hashTable[hash(part)] = part
```

```
        tmp += hash(part)
    for com in completion :
        tmp -= hash(com)
    return hashTable[tmp]
```