

신고결과받기

`id_list`, `report`, `k`가 주어질 때 처리메일을 받는 횟수를 담은 배열 `answer` 반환

신고당한 횟수가 `k`를 넘으면 이용자는 정지 처리되며 해당 이용자를 신고한 이용자에게 처리 메일이 발송 됨.

+ `answer`의 순서는 `id_list`와 같음.

1. 신고한 사람/신고당한 사람 저장할 빈리스트 생성
2. 중복되는 신고 1회로 처리
3. `report`에서 신고자와 신고당한자를 나누기 위한 `split`함수
4. 처리메일을 받을 이용자 딕셔너리 생성
5. 신고당한 사람과 횟수를 딕셔너리로 저장
6. 신고된 이용자의 신고횟수가 `k`보다 클 때 정지될 이용자 리스트로 저장
7. `id_list`의 순서에 따라 이용자가 처리메일을 몇 번 받을 것인가를 저장함.
8. `value`값만 꺼내서 `answer`에 추가함.

```
In [1]: from collections import Counter

def solution(id_list, report, k):
    answer = []
    lista, listb = [], []    #신고한 사람/신고당한 사람 저장할 빈리스트 생성
    report = set(report)     #중복되는 신고 1회로 처리

    for str in report:       #report에서 신고자와 신고당한자를 나누기 위한 split함수
        a, b = str.split(' ')
        lista.append(a)
        listb.append(b)

    dic2 = {}                #처리메일을 받을 이용자 딕셔너리 생성
    for name in id_list:
        dic2[name] = 0       #0으로 초기화해둠

    cnt = dict(Counter(listb)) # 신고당한 사람과 횟수를 딕셔너리로 저장

    stop_user = []          #정리 처리될 예정인 이용자 저장할 리스트
    for key, val in cnt.items(): #신고된 이용자의 신고횟수가 k보다 클 때 정지될 이용자 리스트
        if val >= k:
            stop_user.append(key)

    for i in range(len(stop_user)):    #id_list의 순서에 따라 이용자가 처리메일을 몇 번 받을
        for j in range(len(listb)):
            if stop_user[i] == listb[j]:
                dic2[lista[j]] += 1

    for v in dic2.values():             #value값만 꺼내서 answer에 추가함.
        answer.append(v)

    return answer
```

```
In [ ]: def solution(id_list, report, k):
        answers = [0]* len(id_list)
        report = set(report)
        reporter = {x:0 for x in id_list}

        for i in report:
            reporter [i.split()[1]] +=1
        for i in report:
            if reporter[i.split()[1]] >=k:
                answers[id_list.index(i.split()[0])] += 1

        return answers
```