


NO. 36 오픈채팅방

김다빈

NO. 36

코딩테스트 연습 > 2019 KAKAO BLIND RECRUITMENT > 오픈채팅방

도움말컴파일 옵션

오픈채팅방제출 내역

문제 설명

오픈채팅방

카카오톡 오픈채팅방에서는 친구가 아닌 사람들과 대화를 할 수 있는데, 본래 닉네임이 아닌 가상의 닉네임을 사용하여 채팅방에 들어갈 수 있다.

신입사원인 김크루는 카카오톡 오픈 채팅방을 개설한 사람을 위해, 다양한 사람들이 들어오고, 나가는 것을 지켜볼 수 있는 관리자창을 만들기로 했다. 채팅방에 누군가 들어오면 다음 메시지가 출력된다.

"[닉네임]님이 들어왔습니다."

채팅방에서 누군가 나가면 다음 메시지가 출력된다.

"[닉네임]님이 나갔습니다."

채팅방에서 닉네임을 변경하는 방법은 다음과 같이 두 가지이다.

- 채팅방을 나간 후, 새로운 닉네임으로 다시 들어간다.
- 채팅방에서 닉네임을 변경한다.

닉네임을 변경할 때는 기존에 채팅방에 출력되어 있던 메시지의 닉네임도 전부 변경된다.

예를 들어, 채팅방에 "Muzi"와 "Prodo"라는 닉네임을 사용하는 사람이 순서대로 들어오면 채팅방에는 다음과 같이 메시지가 출력된다.

solution.py

```
1 def solution(record):
2     dict = {}
3     log = []
4     record = [inst.split(' ') for inst in record]
5
6     for inst in record :
7         if (inst[0] == 'Enter') or (inst[0] == 'Change') :
8             dict[inst[1]] = inst[2]
9
10    for inst in record :
11        if inst[0] == 'Enter' :
12            log.append( '%s님이 들어왔습니다.'%dict[inst[1]])
13        elif inst[0] == 'Leave' :
14            log.append( '%s님이 나갔습니다.'%dict[inst[1]])
15
16    return log
```

실행 결과

테스트 28 / 28개 통과 (176.25ms, 65.0MB)

질문하기 (103)

테스트 케이스 추가하기

다른 사람의 풀이

초기화

코드 실행

제출 후 채점하기

문제 소개

카카오톡 오픈채팅방에서는 친구가 아닌 사람들과 대화를 할 수 있는데, 본래 닉네임이 아닌 가상의 닉네임을 사용하여 채팅방에 들어갈 수 있다. 신입사원인 김크루는 카카오톡 오픈 채팅방을 개설한 사람을 위해, 다양한 사람들이 들어오고, 나가는 것을 지켜볼 수 있는 관리자창을 만들기로 했다.

채팅방에 누군가 들어오면 다음 메시지가 출력된다.
"[닉네임]님이 들어왔습니다."

채팅방에서 누군가 나가면 다음 메시지가 출력된다.
"[닉네임]님이 나갔습니다."

채팅방에서 닉네임을 변경하는 방법은 다음과 같이 두 가지이다.

- 채팅방을 나간 후, 새로운 닉네임으로 다시 들어간다.
- 채팅방에서 닉네임을 변경한다.

닉네임을 변경할 때는 기존에 채팅방에 출력되어 있던 메시지의 닉네임도 전부 변경되고, 채팅방은 중복 닉네임을 허용한다.

... 중략 ...

채팅방에 들어오고 나가거나, 닉네임을 변경한 기록이 담긴 문자열 배열 record가 매개변수로 주어질 때, 모든 기록이 처리된 후, 최종적으로 방을 개설한 사람이 보게 되는 메시지를 문자열 배열 형태로 return 하도록 solution 함수를 완성하라.

Prodo 님이 들어왔습니다.

Ryan 님이 들어왔습니다.

Prodo 님이 나갔습니다.

Prodo 님이 들어왔습니다.

NO. 36

예제

[입력]
["Enter uid1234 Muzi", "Enter uid4567 Prodo", "Leave uid1234", "Enter uid1234 Prodo", "Change uid4567 Ryan"]

[Result]
["Prodo님이 들어왔습니다.", "Ryan님이 들어왔습니다.", "Prodo님이 나갔습니다.", "Prodo님이 들어왔습니다."]

‘uid1234’
‘Muzi’

‘uid1234’	‘uid4567’
‘Muzi’	‘Prodo’

NO. 36

예제

[입력]
["Enter uid1234 Muzi", "Enter uid4567 Prodo", "Leave uid1234", "Enter uid1234 Prodo", "Change uid4567 Ryan"]

[Result]
["Prodo님이 들어왔습니다.", "Ryan님이 들어왔습니다.", "Prodo님이 나갔습니다.", "Prodo님이 들어왔습니다."]

‘uid1234’	‘uid4567’
‘Prodo’	‘Prodo’

‘uid1234’	‘uid4567’
‘Prodo’	‘Ryan’

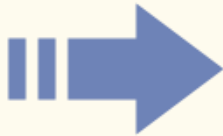
NO. 36

예제

[입력]
["Enter uid1234 Muzi", "Enter uid4567 Prodo", "Leave uid1234", "Enter uid1234 Prodo", "Change uid4567 Ryan"]

[Result]
["Prodo님이 들어왔습니다.", "Ryan님이 들어왔습니다.", "Prodo님이 나갔습니다.", "Prodo님이 들어왔습니다."]

‘uid1234’	‘uid4567’
‘Prodo’	‘Prodo’



‘uid1234’	‘uid4567’
‘Prodo’	‘Ryan’

파이썬의 딕셔너리(dictionary)를 이용해 표현 가능하다

NO. 36

```
def solution(record):
    dict = {}
    log = []

    # record에 저장된 문자열들을 공백 기준으로 split한다
    record = [inst.split(' ') for inst in record]

    for inst in record :
        # 첫번째 명령어가 Enter이거나 Change라면 딕셔너리에 데이터를 저장
        # +) Change의 경우 데이터를 덮어쓰기
        if (inst[0] == 'Enter') or (inst[0] == 'Change') :
            dict[inst[1]] = inst[2]

    # record에 저장된 문자열을 하나씩 꺼내오면서
    for inst in record :
        # 첫번째 명령어가 Enter이면
        if inst[0] == 'Enter' :
            log.append('%s님이 들어왔습니다.'%dict[inst[1]])

        # 첫번째 명령어가 Leave이면
        elif inst[0] == 'Leave' :
            log.append('%s님이 나갔습니다.'%dict[inst[1]])

    # 만들어낸 로그를 리턴
    return log
```

감사합니다
