나도 문제에서 원하는 의도대로 잘 코딩하였다.

그런데 다른사람들 코드를 보면 더간단하게 구현할수 있었다.

|  |  |
| --- | --- |
| 2 3 4 5 6 7 8 9 10 11 12 13 14 15 1  6 17 18 19 | #include <string> #include <vector> #include <algorithm>  using namespace std;  string solution(vector<string> participant, vector<string> completion) {  string answer = "";  sort(participant.begin(), participant.end());  sort(completion.begin(), completion.end());  for(int i=0;i<completion.size();i++)  {  if(participant[i] != completion[i])  return participant[i];  }  return participant[participant.size() - 1];  //return answer; } |

이 코드를 보면 <algorithm> 라이브러리의 sort를 사용했다.

sort (iterator start, iterator end); 이런식으로 사용하면 되나보다.

기본은 오름차순 이란다.

저렇게하면 이문제에서 말하는 중복 문제도 쉽게 해결 할 수있다.

그리고 또 하나는

for(auto elem : participant)

이런 코드인데

아직 C++을 제대로 공부하지 않아서 몰랐던 부분인거같다.

다음에는 저렇게도 사용해보자.