문제 설명

직사각형을 만드는 데 필요한 4개의 점 중 3개의 좌표가 주어질 때, 나머지 한 점의 좌표를 구하려고 합니다. 점 3개의 좌표가 들어있는 배열 v가 매개변수로 주어질 때, 직사각형을 만드는 데 필요한 나머지 한 점의 좌표를 return 하도록 solution 함수를 완성해주세요. 단, 직사각형의 각 변은 x축, y축에 평행하며, 반드시 직사각형을 만들 수 있는 경우만 입력으로 주어집니다.

제한사항

* v는 세 점의 좌표가 들어있는 2차원 배열입니다.
* v의 각 원소는 점의 좌표를 나타내며, 좌표는 [x축 좌표, y축 좌표] 순으로 주어집니다.
* 좌표값은 1 이상 10억 이하의 자연수입니다.
* 직사각형을 만드는 데 필요한 나머지 한 점의 좌표를 [x축 좌표, y축 좌표] 순으로 담아 return 해주세요.

입출력 예

| **v** | **result** |
| --- | --- |
| [[1, 4], [3, 4], [3, 10]] | [1, 10] |
| [[1, 1], [2, 2], [1, 2]] | [2, 1] |

입출력 예 설명

입출력 예 #1  
세 점이 [1, 4], [3, 4], [3, 10] 위치에 있을 때, [1, 10]에 점이 위치하면 직사각형이 됩니다.

입출력 예 #2  
세 점이 [1, 1], [2, 2], [1, 2] 위치에 있을 때, [2, 1]에 점이 위치하면 직사각형이 됩니다.

#include <iostream>

#include <vector>

using namespace std;

vector<int> solution(vector<vector<int> > v) {

int count = 0;

vector<int> ans;

while (count < 2)

{

for (int i = 0; i < v.size(); i++)

{

bool isAnswer = false;

for (int j = 0; j < v.size(); j++)

{

if (i == j) continue;

if (v[i][count] == v[j][count])

{

isAnswer = true;

break;

}

}

if (!isAnswer)

ans.push\_back(v[i][count]);

}

count++;

}

return ans;

}

/\* 과정 설명

[i][j]에서 j열은 0과 1로 반복되고 i열은 0 ~ 2로 반복된다

그러면 [0][0], [1][0], [2][0]의 값을 서로 비교해서 짝이 없는 값이 x 좌표 값이고

[0][1], [1][1], [2][1]의 값에도 짝이 없는 값이 y좌표값이 된다.

반복문으로 비교해서 나온 2개의 값이 나머지 점의 좌표값이된다.

\*/