

<https://www.acmicpc.net/problem/16986>



유행하는 인싸들의 가위바위보입니다. 지우, 경희, 민호 순으로 가위바위보를 할 때 지우가 모든 손동작을 다르게 하여 우승할 수 있는지 판단하는 문제입니다.

## 풀이

문제를 풀면서 코드를 3번 정도 엮었는데 알고 보니 모두 문제를 잘못 이해했습니다...

제가 헛갈렸던 조건은

1. 무승부 일 때 진행 순서가 뒤인 사람이 이긴다. (문제를 잘못 읽어서 다르게 했었습니다...)
2. 지우는 여태까지 냈던 손동작이 다르면 된다. (모든 N 개의 손동작을 해야 하는 줄 알았습니다.)

이 문제를 풀면서 문제를 제대로 읽어야 하는구나 다시 한번 느꼈네요.

지우가 다른 손동작을 낼 수 있는 경우의 수를 따져봤을 때 지우가 우승할 수 있다면 1 못한다면 0을 출력하면 됩니다.

생각은 쉽게 하지만 구현이 저는 까다로웠습니다... 때문에 코드가 엉망...

예제가 다양해서 다행입니다.

제 코드가 조금이나마 도움이 됐으면 좋겠네요...

## 소스 코드

```
#include <bits/stdc++.h>
#define FIO ios::sync_with_stdio(0), cin.tie(0), cout.tie(0)
#define FOR(i, a, b) for(int i=a; i<=b; i++)
using namespace std;

int main() {
    FIO;
    int n, k;
    cin >> n >> k;
    int t[10][10], rps[3][21], cnt[3] = {}, p[3] = {};
    bool v[10];
    FOR(i, 1, n) FOR(j, 1, n) cin >> t[i][j];
    FOR(i, 1, 20) cin >> rps[1][i];
    FOR(i, 1, 20) cin >> rps[2][i];

    function<bool(int, int, int)> ans = [&](int i, int f, int s) { // 0 - 지우, 1 - 경희, 2 - 민호
        if (p[0] == k) return 1;
        if (p[1] == k || p[2] == k) return 0;
        cnt[f]++, cnt[s]++;
        int ff = rps[f][cnt[f]], ss = rps[s][cnt[s]];
        if (f && s) { // 경희, 민호
            int win = (t[ff][ss] == 2 || (t[ff][ss] == 1 && f > s)) ? f : s;
            p[win]++;
            if (ans(i + 1, win, 0)) return 1;
            p[win]--;
        }
        else {
            FOR(j, 1, n) {
                if (v[j]) continue;
                v[j] = 1;
                int x = rps[f | s][cnt[f | s]]; // f | s - 경희 or 민호
                int win = (t[j][x] == 2) ? 0 : f | s;
                p[win]++;
                if (ans(i + 1, win, 3 - f - s)) return 1;
                p[win]--;
                v[j] = 0;
            }
        }
        cnt[f]--, cnt[s]--;
        return 0;
    };

    cout << ans(1, 0, 1) << '\n';
    return 0;
}
```

