

# Soluciones entrenos OIE

## Poi

### Solución

Primero, calcularemos a partir de la tabla de entrada cuántos concursantes han resuelto cada problema. Sabiendo esto, sumaremos la puntuación de cada jugador y contaremos el número de problemas que ha resuelto.

Usando un comparador que simula el ordenamiento descrito en el enunciado, ordenamos a los jugadores. Finalmente, buscamos a Phillip en el ranking e imprimimos su puntuación y clasificación final.

### Código

C++

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  using namespace std;
5
6  vector<int> puntos, solucionesPorProblema, problemasResueltos;
7
8  bool comp(int i, int j) {
9      // comparador para ordenar los concursantes
10     if (puntos[i] != puntos[j]) return puntos[i] > puntos[j];
11     else if (problemasResueltos[i] != problemasResueltos[j]) return
12         ↪ problemasResueltos[i] > problemasResueltos[j];
13     return i < j;
}
```

```

1 void ordenarConcursantes(vector<vector<int>> & tabla, vector<int> & concursantes)
  ↪ {
2     int T = (int) tabla[0].size(), N = (int) concursantes.size();
3     // guardamos en un vector cuántos concursantes han resuelto cada problema
4     for (int i = 0; i < T; i++) {
5         for (int j = 0; j < N; j++) solucionesPorProblema[i] +=
        ↪ tabla[j][i];
6     }
7     // calculamos jugador a jugador cuántos puntos ha obtenido y cuántos
    ↪ problemas ha resuelto
8     for (int i = 0; i < N; i++) {
9         for (int j = 0; j < T; j++) {
10            if (tabla[i][j]) {
11                puntos[i] += (N-solucionesPorProblema[j]);
12                problemasResueltos[i]++;
13            }
14        }
15    }
16    // ordenamos a los concursantes con el algoritmo descrito en las
    ↪ instrucciones
17    sort(concursantes.begin(), concursantes.end(), comp);
18 }
19
20 int main() {
21     int N, T, P;
22     cin >> N >> T >> P;
23     vector<vector<int>> tabla(N, vector<int>(T));
24     vector<int> concursantes(N);
25     puntos.assign(N, 0);
26     problemasResueltos.assign(N, 0);
27     solucionesPorProblema.assign(T, 0);
28     for (int i = 0; i < N; i++) {
29         concursantes[i] = i;
30         for (int j = 0; j < T; j++) {
31             cin >> tabla[i][j];
32         }
33     }
34
35     ordenarConcursantes(tabla, concursantes);
36
37     // Buscamos a Phillip en la clasificación e imprimimos sus puntos y su
    ↪ puesto final
38     for (int i = 0; i < N; i++) {
39         if (concursantes[i] == P-1) {
40             cout << puntos[P-1] << " " << i+1 << "\n";
41             break;
42         }
43     }
44
45     return 0;
46 }

```