

Soluciones entrenos OIE

Pizzas

En las anteriores reuniones de la Organización Independiente de Epizootias, también conocida como la Organización Mundial de Sanidad Animal, se reunían todos en Barcelona para discutir sobre los temas más relevantes. Pero cada año ocurrían varios problemas.

El problema llegaba cuando tocaba cenar, siempre se proponían varias opciones para cenar, un comedor en el lugar de reuniones, una cafetería un poco al sur del lugar o la pizzería de la esquina, pero este no era el problema, pues los asistentes a la reunión no podían elegir y siempre se iban a la pizzería.

El problema que se encontraban es que nunca sabían si podían comprar pizzas para satisfacer a todos los comensales sin que sobrase.

La pizzería vende M toppings de pizzas diferentes numeradas desde 0 hasta $M - 1$, cada pizza se divide en 8 porciones y una pizza solo tiene un topping.

En la reunión asisten N personas, y cada persona quiere comer p_i porciones de pizza.

La organización quiere saber si dadas las porciones que quieren comer los asistentes, es posible comprar un número de pizzas que haga que no se malgaste comida.

Input Format

La entrada consiste en varios casos. Cada caso consiste en varias líneas.

Cada caso empieza con dos enteros N, M . El número de asistentes y el número toppings de pizza que nos ofrece la pizzería.

Viene seguido de N líneas, cada línea empieza con un número p_i : el número de porciones que quiere comer la i -ésima persona, seguido de p_i enteros t_{ij} indicando los toppings que quiere para sus porciones. Una persona puede querer más de una porción con el mismo topping.

Constraints

$$1 \leq N \leq 100$$

$$1 \leq M \leq 100$$

$$1 \leq p_i \leq 10^5$$

$$0 \leq t_{i,j} \leq M - 1$$

Para cada caso la suma de p_i no es mayor que 10^5 .

- 15 Puntos: $M = N = 1$

- 20 Puntos: $M = 1$
- 65 Puntos: No hay restricciones adicionales

Output format

Para cada caso debes imprimir **SI**, si es posible comprar pizzas sin que sobre comida o **NO**, si no es posible. Cada caso debe imprimirse en una línea diferente.

Solución

Tenemos que contar cuántas porciones de pizza vamos a necesitar de cada topping. Si la cantidad que necesitamos de cada uno de los toppings es un múltiplo de 8, no va a sobrar pizza, si hay algún topping cuya cantidad no es un múltiplo de 8 sobraré pizza.

Código

C++

```

1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  int main() {
7      int n, m;
8      while (cin >> n >> m){
9          vector <int> cantidad(m, 0); //cantidad[i] será la cantidad que
          ↪ necesitamos del topping i
10
11         for (int persona = 0; persona < n; ++persona){
12             int porciones;
13             cin >> porciones;
14
15             for (int porcion = 0; porcion < porciones; ++porcion){
16                 int topping;
17                 cin >> topping;
18                 ++cantidad[topping];
19             }
20         }
21
22         //miramos si la cantidad de cada topping es múltiplo de 8
23         bool posible = true;
24         for (int topping = 0; topping < m; ++topping){
25             if (cantidad[topping] % 8 != 0) posible = false;
26         }
27
28         if (posible) cout << "SI" << endl;
29         else cout << "NO" << endl;
30     }
31 }

```

Python

```
1 import sys
2
3 for line in sys.stdin:
4     n, m = map(int, line.split())
5
6     cantidad = m*[0] #cantidad[i] será la cantidad que necesitamos del topping i
7
8     for persona in range(n):
9         toppings = list(map(int, sys.stdin.readline().strip().split(' ')))
10
11         for porcion in range(1, len(toppings)):
12             topping = toppings[porcion]
13             cantidad[topping] += 1
14
15         #miramos si la cantidad de cada topping es múltiplo de 8
16         posible = True
17         for topping in range(m):
18             if (cantidad[topping] % 8 != 0): posible = False
19
20         if (posible): print("SI")
21         else: print("NO")
```