
Filas Permutadas

S24188_es

Sea una matriz M cuadrada de $n \times n$ enteros ($n \geq 1$), en la que sabemos que los valores de una misma fila siempre son todos diferentes entre sí (es decir, no hay valores repetidos), y $M_{ij} \in [0, 1000]$, o sea, los valores de la matriz están entre 0 y 1000, ambos incluidos.

Decimos que la matriz satisface la propiedad de **filas permutadas** si y solo si:

- Para cada par de filas i y j , con $i \neq j$, la fila i es una permutación de la fila j . Es decir, cada fila contiene los mismos valores que las otras filas, pero en un orden diferente. (En particular, esto implica que todas las filas son diferentes.)

Por ejemplo, consideremos la matriz de 3×3 :

1	2	3
2	3	1
3	1	2

Esta matriz satisface la propiedad porque todas las filas contienen los valores $\{1, 2, 3\}$ y son todas diferentes.

Escribe una función `te_files_permutadas` con la siguiente declaración y especificación:

```
/*
 * @pre M es cuadrada y M.size() > 0
 *       1 <= M[i][j] <= 1000
 *       Los valores de una misma fila son todos diferentes.
 *
 * @post Devuelve `true` si y solo si la matriz satisface las
 *       siguientes condiciones:
 *       - Todas las filas son diferentes.
 *       - Todas las filas son permutaciones entre sí.
 */
bool te_files_permutadas(const vector<vector<int>>& M);
```

Observación

El ícono llamado '.CPP' contiene el programa principal para hacer las pruebas. **Solo** hace falta enviar la función que se pide (y las funciones que hayáis declarado vosotros). Podéis empezar con `main.cc` y comentar o borrar las funciones `llegeix_matriu` y `main` para hacer el envío.

En este problema es muy ventajoso utilizar la comparación de vectores (es decir, compararlos con los operadores `==` y `!=`). Por ejemplo:

```
vector<int> a = {1, 2, 3}, b = {1, 2, 3}, c = {4, 5, 6};
if (a == b) { cout << "Iguales!" << endl; }
if (a != c) { cout << "Diferentes!" << endl; }
```

Entrada

La entrada es una secuencia de matrices cuadradas. De la lectura ya se encarga el programa principal.

Salida

La salida son los resultados para cada matriz: "yes" si satisface la propiedad, "no" en caso contrario. De la escritura también se encarga el programa principal.

Ejemplo de entrada 1

```
3
1 2 3
2 3 1
3 1 2
```

```
4
1 2 3 4
2 3 4 1
3 4 1 2
1 2 3 4
```

```
4
0 5 1 8
5 1 8 0
1 8 0 5
8 5 1 0
```

```
4
5 0 9 7
0 9 7 5
9 7 5 0
7 4 5 0
```

```
5
1 2 3 4 5
2 3 4 5 1
3 4 5 1 2
4 5 1 2 3
5 1 2 3 4
```

Ejemplo de salida 1

```
yes
no
yes
no
yes
```

Información del problema

Autoría: Pau Fernández

Generación: 2026-01-25T12:42:59.748Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>