Matrices dispersas (3): suma — T33423_es

Tal como vimos en el problema *Matrices dispersas (1): conversión*, una **matriz dispersa** solo almacena los elementos no nulos. Los tipos usados (definidos en `matrius.hh`) son los mismos:

```
struct Casella {
    int pos;    // índice de columna
    int valor;  // valor del elemento
};

struct MatriuEsparsa {
    int ncols;                    // número de columnas
    vector<list<Casella>> files; // filas de la matriz dispersa
};
```

Realiza la siguiente función:

```
/**
 * @brief Calcula la suma de dos matrices de tipo `MatriuEsparsa`.
 *
 * @param A MatriuEsparsa de A.files.size() filas y A.ncols columnas.
 * @param B MatriuEsparsa de B.files.size() filas y B.ncols columnas.
 *
 * @pre A y B son matrices dispersas correctas.
 *      Las dimensiones de A y B son iguales.
 *
 * @post Retorna una nueva MatriuEsparsa C donde C.files[i] es la suma de
 *       A.files[i] y B.files[i], conservando solo los elementos no nulos,
 *       ordenados por columna ascendente.
 */
MatriuEsparsa suma_matrius_esparses(const MatriuEsparsa& A,
                                    const MatriuEsparsa& B);
```

Para verificar la precondición sobre las dimensiones, hay que usar el `assert` de `assert.hh` de la siguiente manera:

```
assert(condicio, "suma_matrius_esparses: dimensions diferents!");
```

Observación

Los ficheros públicos (icono del gatito) contienen:

`main.cc`	el programa principal, con la entrada/salida hecha
`matrius.hh`	los tipos `Casella` y `MatriuEsparsa`
`assert.hh`	la función `assert` para verificar precondiciones
`Makefile`	para compilar con `make` fácilmente
`.vscode`	carpeta para compilar y depurar con VSCode

Se debe enviar únicamente la implementación de la función `suma_matrius_esparses` en un fichero `.cc`, poniendo los includes de `matrius.hh` y `assert.hh`. El Juez copia el fichero enviado en una carpeta como la proporcionada y compila haciendo `make`.

Entrada

La entrada contiene dos matrices dispersas consecutivas. Cada matriz empieza con una línea con dos enteros n y m ($0 \leq n, m \leq 1000$), el número de filas y columnas. A continuación hay n líneas, cada una con las casillas no nulas de la fila como parejas `(col,val)` separadas por espacios, seguidas de un punto `.`. Las filas sin elementos no nulos se representan como `.`.

Salida

Si las dos matrices tienen las mismas dimensiones, se imprime la matriz suma en el mismo formato que la entrada: la primera línea contiene las dimensiones $n \times m$, seguida de n líneas con las casillas no nulas de cada fila, seguidas de un punto `.`.
Si las dimensiones no coinciden, se imprime: `suma_matrius_esparses: dimensions diferents!`

Ejemplo de entrada 1

```
3 4
(1,3) (2,5).
.
(0,1) (3,7).

3 4
(1,-3).
(2,2).
(0,1) (3,4).
```

Ejemplo de salida 1

```
3 4
(2,5).
(2,2).
(0,2) (3,11).
```

Ejemplo de entrada 2

```
2 3
(0,4) (2,-1).
(1,3).

2 3
(0,-4) (1,2).
(1,-3) (2,5).
```

Ejemplo de salida 2

```
2 3
(1,2) (2,-1).
(2,5).
```

Ejemplo de entrada 3

```
2 3
(0,1).
(1,2).

3 2
(0,3).
(1,4).
(0,5).
```

Ejemplo de salida 3

```
`assert` disparat i amb el missatge correcte
```

Información del problema

Autoría: Pau Fernández
Traducción: Pau Fernández