
Matrius esparses (3): suma

T33423_ca

Tal com vam veure al problema *Matrius esparses (1): conversió*, una **matriu esparsa** només emmagatzema els elements no nuls. Els tipus usats (definit a `matrius.hh`) són els mateixos:

```
struct Casella {
    int pos;    // índex de columna
    int valor;  // valor de l'element
};

struct MatriuEsparsa {
    int ncols;           // nombre de columnes
    vector<list<Casella>> files; // files de la matriu esparsa
};
```

Fes la funció següent:

```
/**
 * @brief Calcula la suma de dues MatriuEsparsa.
 *
 * @param A MatriuEsparsa de A.files.size() files i A.ncols columnes.
 * @param B MatriuEsparsa de B.files.size() files i B.ncols columnes.
 *
 * @pre A i B són matrius esparses correctes. Les dimensions d'A i B són iguals.
 *
 * @post Retorna una nova MatriuEsparsa C on C.files[i] és la suma de
 *       A.files[i] i B.files[i], conservant només els elements no nuls,
 *       ordenats per columna ascendentment.
 */
MatriuEsparsa suma_matrius_esparses(const MatriuEsparsa& A,
                                     const MatriuEsparsa& B);
```

Per verificar la precondició sobre les dimensions, cal usar el `assert` de `assert.hh` de la forma:

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

Observació

Els fitxers públics (icona del gatet) contenen:

<code>main.cc</code>	el programa principal, amb la entrada/sortida feta
<code>matrius.hh</code>	els tipus <code>Casella</code> i <code>MatriuEsparsa</code>
<code>assert.hh</code>	la funció <code>assert</code> per verificar precondicions
<code>Makefile</code>	per compilar amb <code>make</code> còmodament
<code>.vscode</code>	carpeta per compilar i depurar amb VSCode

Cal enviar únicament la implementació de la funció `suma_matrius_esparses` en un fitxer `verb|.cc|`, posant els `includes` de `matrius.hh` i `assert.hh`. El Jutge copia el fitxer enviat en una carpeta com la proporcionada i compila fent `make`.

Entrada

L'entrada conté dues matrius esparses seguides. Cada matriu comença amb una línia amb dos enters n i m ($0 \leq n, m \leq 1000$), el nombre de files i columnes. A continuació hi ha n línies, cadascuna amb les caselles no nul·les de la fila com a parelles `(col, val)` separades per espais, seguides d'un punt `.`. Les files sense elements no nuls es representen com `.`.

Sortida

Si les dues matrius tenen les mateixes dimensions, s'imprimeix la matriu suma en el mateix format que l'entrada: la primera línia conté les dimensions $n \times m$, seguida de n línies amb les caselles no nul·les de cada fila, seguides d'un punt `.`.

Si les dimensions no coincideixen, s'imprimeix: `suma_matrius_esparses: dimensions diferents!`

Exemple d'entrada 1

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

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

Exemple de sortida 1

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

Exemple d'entrada 2

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

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

Exemple de sortida 2

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

Exemple d'entrada 3

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

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

Exemple de sortida 3

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

Informació del problema

Autoria: Pau Fernández

Generació: 2026-02-25T09:48:03.957Z

© Jutge.org, 2006–2026.

<https://jutge.org>