

---

## Laberint

S14772\_ca

---

Sigui una matriu  $n \times m$  d'enters. Si un valor de la matriu és negatiu, direm que en aquella posició hi ha una **pedra**. Si el valor és més gran o igual que zero, llavors direm que en aquella posició hi ha un **premi**.

Heu de fer una funció que rebi una matriu `M`, un enter `potencia > 0`, un enter `objectiu > 0` i un string `recorregut` que podrà contenir només els caràcters `N`, `S`, `E`, `O`. Aquests valors determinen un recorregut per la matriu: partint de la posició inicial `(0,0)`, si rebem una `N` haurem de pujar una fila, si rebem una `S` haurem de baixar una fila, si rebem una `O` haurem d'anar una columna a l'esquerra, i si rebem una `E` haurem d'anar una columna a la dreta. Òbviament, no podem sortir de la matriu. Si el caràcter ens fa sortir de la matriu, ens haurem de quedar a la mateixa posició.

La funció haurà de dir si amb el `recorregut` podem aconseguir recollir prou premis per tal d'arribar a l'`objectiu` o no.

La posició de partida és `(0,0)`. A més, aquesta posició no contindrà mai cap pedra, només un premi.

Cada vegada que passem per una posició `(i,j)` podrà passar

1. Que hi hagi una pedra (un valor estrictament més petit que zero). En aquest cas, disminuïm la `potencia` en tantes unitats com el valor absolut d'aquesta posició.
2. Que hi hagi un premi. En aquest cas, acumulem aquest valor.

Podem passar per la mateixa posició més d'una vegada, depenent del que ens mani fer el `recorregut`. I cada vegada que hi passem, podem acumular el premi. Ara bé, si ens trobem en una posició donada i el següent moviment de `recorregut` ens fa anar fora de la matriu, el que farem serà no moure'ns de posició, i no hi haurà ni acumulació de premi (si ens trobem en un premi), ni reducció de potència (si ens trobem en una pedra).

La funció tornarà `ACONSEGUIT` si el `recorregut` és capaç de recollir almenys tants premis com `objectiu`. En canvi, si ha passat per tantes pedres que ha esgotat tota la seva `potencia`, haurà de tornar `ESGOTAT`. Si després d'exhaurir el `recorregut`, no ha pogut recollir prou premis com `objectiu`, llavors haurà de tornar `NO HI ARRIBEM`.

Per exemple, si tenim la matriu

```
1  2  -1  3
-1  1   1 -4
2  4  -1  2
3  -1 -1  2
1  0   1  1
```

i tenim que `potencia = 6`, `objectiu = 10` i `recorregut = EESESSS`, començarem a la posició `(0,0)`. El `recorregut` serà aquest:

Recorregut	Posició	M[i][j]	Potència	Acumulat
E	(0,1)	2	6	3
E	(0,2)	-1	5	3
S	(1,2)	1	5	4
E	(1,3)	-4	1	4
S	(2,3)	2	1	6
S	(3,3)	2	1	8
S	(4,3)	1	1	9

El resultat serà NO HI ARRIBEM perquè haurem exhaurit el recorregut, però no haurem aconseguit prou premis com l'objectiu ens demanava.

Fes una funció `matriu_laberint` amb la següent declaració i especificació:

```

/*
 * PRE: M.size() > 0 and M[0].size() > 0, és una matriu d'enters.
 * M[0][0] >= 0.
 * potencia > 0, objectiu > 0.
 * recorregut.size() > 0 i recorregut només conté 'N','S','E','O'.
 *
 * POST: Retorna ACONSEGUIT si el recorregut és capaç de recollir
 *       almenys tants premis com objectiu.
 *       Retorna ESGOTAT si recull tantes pedres o més que potència.
 *       Retorna NO HI ARRIBEM si el recorregut no aconsegueix recollir
 *       tants premis com objectiu.
 */

string matriu_laberint(const Matriu& M,int potencia, int objectiu,
                      const string& recorregut)

```

## Observació

**Només** cal enviar la funció que us demanem (i les funcions que hagueu pogut declarat vosaltres).

A més, al fitxer que envieu **cal** que també hi hagi això:

```

#include <iostream>
#include <vector>
using namespace std;

typedef vector<int> Vector;
typedef vector<Vector> Matriu;

```

No es pot fer servir l'ordenació del C++: `std::sort`. Tampoc no es pot fer servir el mètode `push_back()` de la classe `vector`.

Si voleu, podeu fer servir les funcions `min`, `max` o `swap`.

## Entrada

L'entrada és una matriu d'enters no buida, un enter potencia, un enter objectiu i un string recorregut. A la posició `M[0][0]` no hi ha una pedra. *potencia* > 0, *objectiu* > 0. *recorregut.size()* > 0 i recorregut només conté 'N','S','E','O'.

## Sortida

Retorna ACONSEGUIT si el recorregut és capaç de recollir almenys tants premis com objectiu. Retorna ESGOTAT si recull tantes pedres o més que potència. Retorna NO HI ARRIBEM si el recorregut no aconsegueix recollir tants premis com objectiu.

### Exemple d'entrada 1

```
5 4
 1 2 -1 3
-1 1 1 -4
 2 4 -1 2
 3 -1 -1 2
 1 0 1 1

10
10
ESENOSSSSOONNE

2
10
EESESSOOOONN

6
10
EESESS
```

### Exemple de sortida 1

```
POTENCIA: 10 OBJECTIU: 10 RECORREGUT: ESENOSSSSOONNE RESULTAT: ACONSEGUIT
POTENCIA: 2 OBJECTIU: 10 RECORREGUT: EESESSOOOONN RESULTAT: ESGOTAT
POTENCIA: 6 OBJECTIU: 10 RECORREGUT: EESESS RESULTAT: NO HI ARRIBEM
```

### Exemple d'entrada 2

```
5 4
0 2 4 8
29 -1 4 -1
28 39 35 15
27 -7 36 -6
25 22 20 20

10
90

SEOSSSOESEEENNN
```

### Exemple de sortida 2

```
POTENCIA: 10 OBJECTIU: 90 RECORREGUT: SEOSSSOESEEENNN RESULTAT: ACONSEGUIT
```

## Informació del problema

Autoria: PRO1

Generació: 2026-01-25T12:42:30.018Z

© *Jutge.org*, 2006–2026.  
<https://jutge.org>