

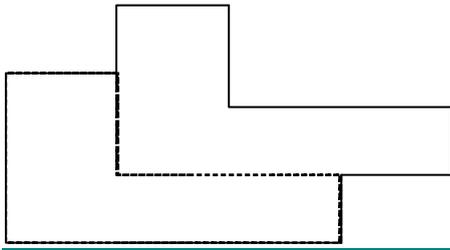
Dividiendo el Campo

Descripción del problema

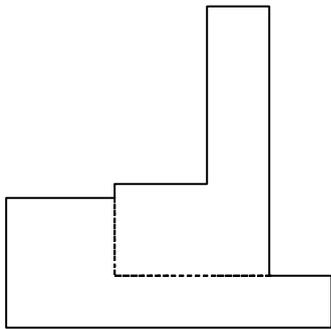
El dueño de un campo ha decidido dividirlo entre sus dos hijos. Como quiere ser muy justo con ellos, le ha pedido a un agrimensor que efectúe las mediciones necesarias para que la subdivisión cumpla con las siguientes restricciones:

- las dos partes deben ser idénticas, es decir, deben tener la misma superficie y la misma forma;
- las dos partes deben estar dispuestas con la misma orientación, es decir, una no puede estar rotada respecto de la otra.

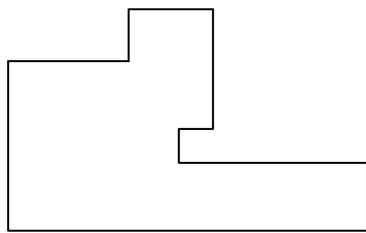
Por ejemplo, este campo puede ser subdividido:



Pero este campo no, pues existen dos subdivisiones idénticas, pero las dos partes están rotadas una respecto de la otra.



Este campo tampoco puede ser subdividido, ya que no pueden encontrarse dos subdivisiones idénticas y con la misma orientación.



Nota: Para efectuar correctamente las mediciones, el agrimensor ajusta el plano del campo en una cuadrícula de $m \times n$ casillas. Se

supone que los lados del campo son paralelos a los bordes de la cuadrícula, y que el plano se ajusta a los bordes cuando es posible.

Se debe escribir un programa **CAMPOS** en C, C++ o Pascal que determine si el campo del padre puede subdividirse en dos campos idénticos, y en tal caso indicar las subdivisiones.

Datos de entrada

Se recibe por la entrada estándar:

- Primera línea: los números m y n indicando el alto y ancho de la cuadrícula ($2 = m, n = 100$), separados por un blanco.
- m líneas con n números:
 - **1** si es un punto perteneciente al campo
 - **0** en otro caso

Datos de salida

El programa debe imprimir en la salida estándar dos cuadrículas de $m \times n$, cada una con una subdivisión, separadas por una línea en blanco. Cada cuadrícula debe conservar el formato de entrada excepto por las dimensiones de la cuadrícula que no debe aclararse.

En caso de no existir subdivisión posible, se debe imprimir "NO EXISTE SOLUCION"

Ejemplo

En el caso de que la entrada fuera:

```
10 21
00000111110000000000
00000111110000000000
00000111110000000000
11111111110000000000
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111100000
11111111111111100000
11111111111111100000
```

La salida debería ser:

```
000001111100000000000
000001111100000000000
000001111100000000000
000001111100000000000
000001111111111111111
000001111111111111111
000001111111111111111
000001111111111111111
000000000000000000000
000000000000000000000
000000000000000000000

000000000000000000000
000000000000000000000
000000000000000000000
111110000000000000000
111110000000000000000
111110000000000000000
111110000000000000000
111110000000000000000
111111111111111110000
111111111111111110000
111111111111111110000
```