



Unlock the door (unlock)

A cybersecurity firm has to develop an access control device to unlock a door. Unlocking requires a magnetic square card to be placed at an appropriate position over a square pad next to the door.

The problem is modeled as follows:

- The **card** is a $n \times n$ 0/1 matrix, with $n \geq 1$
- The **pad** is a $m \times m$ 0/1 matrix, with $m \geq 1$
- The card unlocks the door if it aligns with the pad, i.e., it appears as a submatrix of the pad up to rotations.

A $m \times m$ matrix cell has coordinate (i, j) if it lies on row i and column j , for $0 \leq i < m$ and $0 \leq j < m$.

Example:

A 3×3 card:	A 5×5 pad:
0 1 0	0 0 0 0 0
0 1 1	0 0 0 1 0
0 1 0	0 0 1 1 1
	0 0 0 0 0
	0 0 0 0 0

The card unlocks the door if it is rotated by 270 deg clockwise and its upper-left corner is placed at coordinates $(1, 2)$ of the pad:

```

0 0 0 0 0
0 0 0 1 0 ← Card aligns here after 270 deg clockwise rotation
0 0 1 1 1
0 0 0 0 0
0 0 0 0 0

```

Write a program that checks whether a card aligns with a pad. In case there were more possibilities of unlock the pad you must return the one with the lowest coordinate and the lowest rotation.

Implementation

You should submit a single file, with either a `.c`, `.cpp`, `.java` or `.py` extension.

Your program must read input data from `stdin` and write the output data into `stdout`.

`stdin` consists of $1 + n + m$ lines:

- Line 1: The integers n and m , space separated, the size of the card and the pad.
- Next n lines: n consecutive chars of 0/1.
- Next m lines: m consecutive chars of 0/1.

`stdout` consists of only one line:

- Line 1: Three integer i j r if the card unlocks the pad if placed in (i, j) rotated by r deg, the string **err** if the card does not align with the pad.

Constraints

- $1 \leq n \leq 16$.
- $1 \leq m \leq 16$.

Scoring

Your program will be tested against 10 testcases, each of which is worth 10 points.

Examples

stdin	stdout
<pre>3 5 010 011 010 00000 00010 00111 00000 00000</pre>	<pre>1 2 270</pre>
<pre>2 3 11 11 010 011 110</pre>	<pre>err</pre>