I - Matrix Convolutions Resurrections

Context

Certain problems have the strange ability to resurrect if they are not solved. They are like the monster of the last phase. It will keep appearing once and again, until you manage to kill it! If the monster is a big matrix, then we have a Matrix Resurrection.

Are you prepared to kill it?

The Problem

Given an image and a convolutional mask, your task is to compute the corresponding convolution of the image.

Suppose that we have an image *A* with a size of *W* pixels width and *H* pixels height. The image pixels are *A(x,y)* for $0 \le x < W$, and $0 \le y < H$. And we have a convolutional kernel *K*, of $2w+1$ pixels width and $2h+1$ pixels height. The values of the kernel are *K(x,y)* for $-w \le x \le w$, and $-h \le y \le h$.

Then, we define the convolution of *A* with the kernel *K*, as an image *C* with the same size of *A*, where each pixel is given by,

$$C(x,y) = \sum_{a=-w}^{w} \sum_{b=-h}^{h} K(a,b) \cdot A\big((x+a) \bmod W, (y+b) \bmod H\big)$$

The Input

The input contains an input image and a convolutional kernel.

The first line of the input contains two integer numbers, *W* and *H*, indicating the width and height of the image in pixels, between 10 and 1000, inclusive.

Then, there are *H* lines, each with *W* characters. Each character represents a pixel value, that can be from -40 to 40. The value is given by the ASCII code of the character minus 80. For example, '(' is -40, 'P' is 0, and 'x' is 40.

The next line contains two integers, *w* and *h*, indicating the size of the convolutional kernel, between 1 and 400, inclusive. Then, there are $2h+1$ lines, each with $2w+1$ characters. Each character represents a value of the kernel, that can be from -40 to 40. As in the case of the images, the value is given by

the ASCII code of the character minus 80. The kernel is not larger than the image, i.e., $2w+1 \leq W$, and $2h+1 \leq H$.

The Output

The output contains the resulting image, represented in the same format as the input image. The first line must contain two integers, *W* and *H*, indicating the width and height of the image in pixels.

Then, there are *H* lines, each with *W* characters. Each character represents a pixel value, that can be from -40 to 40. The value is given by the ASCII code of the character minus 80. The output value is saturated to this range, -40 to 40. So, if a pixel result is less than -40, it is set to -40; and if it is greater than 40, it is set to 40.

Sample Input

Sample Input

```
20 15
LHHKDaJBEZOGO_cOI<BV
\NI?b>Y`d[JK?WZGT?M]
^a?H[>\?>bZ_L>cI@XE?
NSNQED]]GFUABYNa\E<_
CZZT]XbaKB<^ZOKcWC\c
MM?TBRPc[Q^bGSQ?FOdV
U<OJO^IB=A@ORD>Y[SXQ
?QOJ?<MJOMd?NNNaG[?I
`D\NM^CCL_XPLC_OCH]W
Z\[CFD@QdC_`KVJ\ORc`
MWK]^EH=R@YGaONBXRXX
YRT?Cc`WP^RaQbZKC=MY
BAd>UMEIda<XNU\VSXH>
R_d>]ZMd[^YbdYdTBDaA
Ab^TRUE@HRC^LCaDaJDX
2 1
PQJWL
MNLPR
ONNQJ
```

Sample Output

```
20 15
xq(xx((<x(x((x(x(Px(
x(Mxxxx(G((s:xb(hxx(
(((x(x(xjxx(xE(x((xx
x((x(x((x(xx(x((((xb
x(x((2(((xx(x(x(((x(
Oxbixxx(x(xLu(xx(x@(
xYxxx(x\(xx(xrx8xxD(
(x/(xxxpx(5x(xx9(0(x
(^(xxx(x(x((`(x(x(x(
(((6xxxxxx(l(x((5x(0
(x:(((9x(xp(((xxOx((
KA(xxx(x(c(x((((4xxx
]x((x(6(((((9((;(x>(
(x(x(x(xx(x((_((x(xx
xx([Q(x(xxx((x(x(x(x
```