Problem 2 – String Matrix Rotation

You are given a **sequence of text lines**. Assume these text lines form a **matrix of characters** (pad the missing positions with spaces to build a rectangular matrix). Write a program to **rotate the matrix** by 90, 180, 270, 360, … degrees. Print the result at the console as sequence of strings. Examples:

Input	Rotate(90)	Rotate(180)	Rotate(270)

`hello`
`softuni`
`exam`
	(rotated matrix)	(rotated matrix)	(rotated matrix)

Input

The input is read from the console:

- The first line holds a command in format "**Rotate(X)**" where **X** are the degrees of the requested rotation.
- The next lines contain the **lines of the matrix** for rotation.
- The input ends with the command "END".

The input data will always be valid and in the format described. There is no need to check it explicitly.

Output

Print at the console the **rotated matrix** as a sequence of text lines.

Constraints

- The rotation **degrees** is positive integer in the range [0…90000], where **degrees** is **multiple of 90**.
- The number of matrix lines is in the range [1…**1 000**].
- The matrix lines are **strings** of length 1 … 1 000.
- Allowed working time: 0.2 seconds. Allowed memory: 16 MB.

Examples

Input	Output

`Rotate(90)`
`hello`
`softuni`
`exam`
	`esh`
	`xoe`
	`afl`
	`mtl`
	` uo`
	` n`
	` i`

Input	Output

`Rotate(180)`
`hello`
`softuni`
`exam`
	`  maxe`
	`inutfos`
	`  olleh`

Input	Output

`Rotate(270)`
`hello`
`softuni`
`exam`
	`  i`
	`  n`
	`ou`
	`ltm`
	`lfa`
	`eox`
	`hse`

Input	Output

`Rotate(720)`
`js`
`exam`
	`js`
	`exam`

Input	Output

`Rotate(810)`
`js`
`exam`
	`ej`
	`xs`
	`a`
	`m`

Input	Output

`Rotate(0)`
`js`
`exam`
	`js`
	`exam`