Problem 2 – Target Practice

Cotton-eye Gosho has a problem. Snakes! An infestation of snakes! Gosho is a red-neck which means he doesn't really care about animal rights, so he bought some ammo, loaded his shotgun and started shooting at the poor creatures. He has a favorite spot – rectangular stairs which the snakes like to climb in order to reach Gosho's stash of whiskey in the attic. You're tasked with the horrible cleanup of the aftermath.

A **snake** is represented by **a string**. The **stairs** are a **rectangular matrix of size NxM**. A snake starts climbing the stairs from the **bottom-right corner** and slithers its way up in a **zigzag** – first it moves left until it reaches the left wall, it climbs on the next row and starts moving right, then on the third row, moving left again and so on. The first cell (bottom-right corner) is filled with the first symbol of the snake, the second cell (to the left of the first) is filled with the second symbol, etc. The snake is as long as it takes in order to **fill the stairs completely** – if you reach the end of the string representing the snake, start again at the beginning. Gosho is patient and a sadist, he'll wait until the stairs are completely covered with snake and will then fire a shot.

The shot has three parameters – **impact row, impact column and radius**. When the projectile lands on the specified coordinates of the matrix it **destroys all symbols in the given radius (turns them into a space)**. You can check whether a cell is inside the blast radius using the Pythagorean Theorem (very similar to the "point inside a circle" problem).

The symbols above the impact area start **falling down until they land on other symbols (meaning a symbol moves down a row as long as there is a space below)**. When the horror ends, print on the console the **resulting staircase, each row on a new line**. You should really check out the examples at this point.

Input

- The input data should be read from the console. It consists of exactly three lines.
- On the first line, you'll receive the **dimensions** of the stairs in format: **"N M"**, where **N** is the number of **rows**, and **M** is the number of **columns**. They'll be separated by a single space.
- On the second line you'll receive the string representing the **snake**.
- On the third line, you'll receive the **shot parameters (impact row, impact column and radius)**, all separated by a **single space**.
- The input data will always be valid and in the format described. There is no need to check it explicitly.

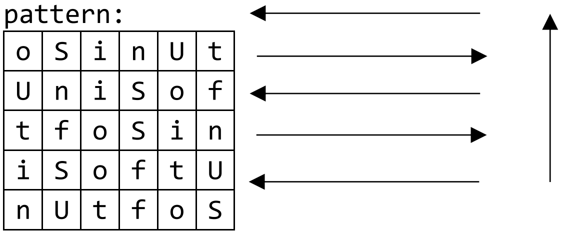
Output

- The output should be printed on the console. It should consist of **N lines**.
- Each line should contain a string representing the respective row of the final matrix.

Constraints

- The **dimensions** N and M of the matrix will be integers in the range [1 … 12].
- The **snake** will be a string with length in the range [1 … 20] and **will not contain any whitespace characters**.
- The shot's **impact row and column** will always be **valid coordinates** in the matrix – they will be integers in the range [0 … N – 1] and [0 … M – 1] respectively.
- The shot's **radius** will be an integer in the range [0 … 4].
- Allowed working time for your program: 0.1 seconds. Allowed memory: 16 MB.

Examples

Input	Output	Comments
5 6 SoftUni 2 3 2	o US t tn f iSi UU nUt oS	The matrix has 5 rows and 6 columns. Fill it in the described pattern:  o S i n U t U n i S o f t f o S i n i S o f t U n U t f o S The shot lands on cell (2,3). It has a radius of 2 cells. The impact

cell is shaded black and the other cells within the shot radius are shaded grey.

o	S	i	n	U	t
U	n	i	S	o	f
t	f	o	S	i	n
i	S	o	f	t	U
n	U	t	f	o	S

Replace all characters in the blast area with a space:

o	S	i		U	t
U	n				f
t					
i	S				U
n	U	t		o	S

All characters start falling down until they land on other characters:

o	S	i		U	t
U	n				f
t	↓				↓
i	S	↓		↓	U
n	U	t		o	S

The resulting matrix is:

o					
U	S				t
t	n				f
i	S	i		U	U
n	U	t		o	S