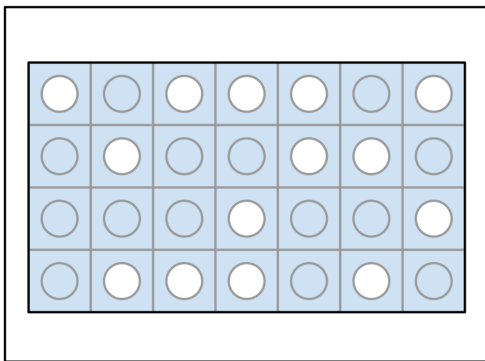


# Card Folding

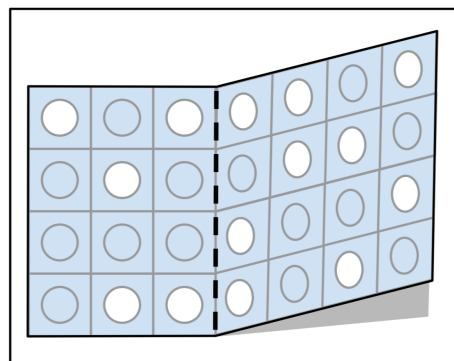
Time limit: 2 seconds

In the prehistoric era of computing, programs were represented as punch cards. A punch card is an  $R \times C$  grid of cells. A cell has a tiny piece of paper that can be removed to make a hole. Each cell of the grid is either a hole or not.

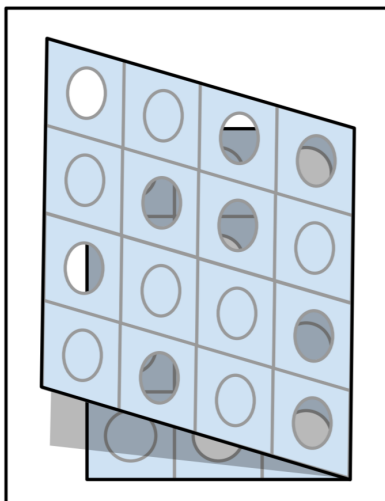
Savine is trying to program a punch card machine. Unfortunately, she does not have the ability to punch any new holes. Luckily, she already has a punch card with holes punched in it. Her plan is to fold the punch card vertically between two columns of the grid in such a way that the rightmost column is folded to the left. After folding, some of the cells of the grid will line up. For example, say Savine has a punch card with 7 columns. If she folds between columns 3 and 4, then the cells in columns 3 and 4 will line up, the cells in columns 2 and 5 will line up, the cells in columns 1 and 6 will line up, and the cells in column 7 will overhang to the left of column 1, not lining up with any cells in the original punchcard. This new punchcard has 4 columns (from left to right: the 7 column, the (1,6) column, the (2,6) column, and the (3,4) column). There is a hole in this new punchcard only if both of the lined-up cells were holes in the original punchcard.



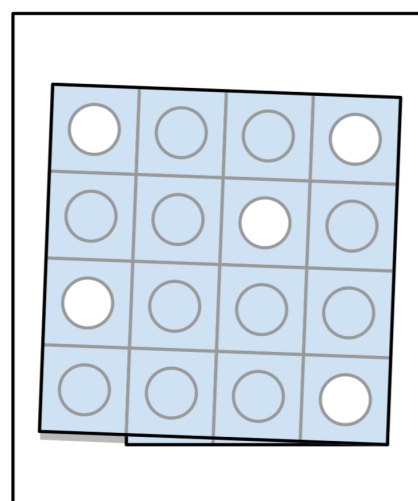
(1)



(2)



(2)



(4)

Given the original punchcard and the column where she is planning on doing the fold, what will the new punchcard look like?

## Input

The first line of input contains three integers  $R$  ( $1 \leq R \leq 50\,000$ ), which is the number of rows,  $C$  ( $2 \leq C \leq 10^5$ ) and  $(R \times C \leq 10^5)$ , which is the number of columns, and  $Y$  ( $1 \leq Y \leq C - 1$ ), which indicates that Savine plans to fold the punch between column  $Y$  and column  $Y + 1$ .

The next  $R$  lines describe the punch card. Each of these lines contains a string of length  $C$  containing only the characters 'o' and 'x'. An 'o' represents a cell with a hole and an 'x' represents a cell without a hole.

## Output

Display the folded punch card.

Sample Input 1

Sample Output 1

4 7 3	
OXOOOXO	OXXO
XOXXOOX	XXOX
XXXOXXO	OXXX
XOOOXOX	XXXO

Sample Input 2

Sample Output 2

3 3 1	
OXX	XX
XOX	XX
XXO	OX

Sample Input 3

Sample Output 3

3 3 2	
OXX	OX
XOX	XX
XXO	XX

Sample Input 3

Sample Output 3

2 3 1	
OOX	XO
XOO	OX

**Submission guidelines:** You need to upload two files in the system.

- (1) Write one page document (upload pdf version of the doc) describing your algorithm or pseudocode. You should describe ***why and how*** your algorithm design should be ***efficient*** (the corresponding program should run fast).
- (2) One program file (**you can upload the zip file for it**) – actual C/C++, Java, or Python code file. Make sure your code finishes its execution within 2 seconds for the largest possible input.
- (3) Zip the above two files and submit your zip file in the system.