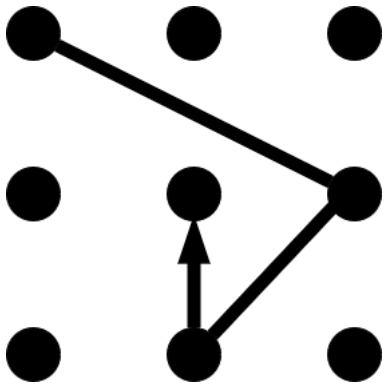
Pattern Lock

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Usually, smartphones can be locked by either a password, fingerprint sensing, or facial recognition. There's also another commonly used method: *Pattern Lock*.

Pattern Lock allows you to lock and unlock your device by drawing a pattern onto the screen. The pattern is drawn by connecting a series of dots on a grid by a polygonal path. The grid of dots on the lock screen has n rows and m columns. The rows have equal spacing between each other, and so do the columns. We denote the dots in the x-th row and y-th column by (x, y). And we use a sequence of dots to denote the pattern, i.e. the polygonal path. For example, the sequence $\{(1,1),(2,3),(3,2),(2,2)\}$ can denote the pattern shown in the picture below.



Let's denote a pattern with k dots as $\{A_1, A_2, \ldots, A_k\}$. A valid pattern to lock a smartphone should meet the following conditions:

- Each dot is visited no more than once. That is, for each $1 \le i < j \le k$, $A_i \neq A_j$.
- For each $1 \le i < k$, the segment connecting A_i and A_{i+1} cannot pass through other dots. For example, $A_i = (1,1), A_{i+1} = (3,3)$ is invalid because the segment passes through $(2,2)$.

Little Rabbit wants his smartphone to be as secure as possible. Therefore, he needs a **strong pattern** to lock his device. A **strong pattern** is a valid pattern that meets some extra conditions:

- Each dot is visited exactly once. That is, $k = n \times m$.
- For each $1 < i < k$, the angle formed by segment A_iA_{i-1} and segment A_iA_{i+1} must be an acute angle (less than 90°).

Can you construct a **strong pattern** for him?

Input

The input contains two integers n and m ($2 \le n, m \le 500$), representing the number of rows and columns of the grid.

Output

Output $n \times m$ lines. The i-th line contains two integers x_i and y_i ($1 \le x_i \le n$, $1 \le y_i \le m$), representing the i-th dot of the pattern is (x_i, y_i).

It can be proved that the answer always exists. If there are multiple answers, output any.

Example

standard input	standard output
2 2	1 1 2 1 1 2 2 2

Note

Please note that if the length or format of your output does not match the answer, you will possibly get a Presentation Error verdict.