

A. Strange Table

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Polycarp found a rectangular table consisting of n rows and m columns. He noticed that each cell of the table has its number, obtained by the following algorithm **"by columns"**:

- cells are numbered starting from one;
- cells are numbered from left to right by columns, and inside each column from top to bottom;
- number of each cell is an integer one greater than in the previous cell.

For example, if $n = 3$ and $m = 5$, the table will be numbered as follows:

1	4	7	10	13
2	5	8	11	14
3	6	9	12	15

However, Polycarp considers such numbering inconvenient. He likes the numbering **"by rows"**:

- cells are numbered starting from one;
- cells are numbered from top to bottom by rows, and inside each row from left to right;
- number of each cell is an integer one greater than the number of the previous cell.

For example, if $n = 3$ and $m = 5$, then Polycarp likes the following table numbering:

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

Polycarp doesn't have much time, so he asks you to find out what would be the cell number in the numbering **"by rows"**, if in the numbering **"by columns"** the cell has the number x ?

Input

The first line contains a single integer t ($1 \leq t \leq 10^4$). Then t test cases follow.

Each test case consists of a single line containing three integers n, m, x ($1 \leq n, m \leq 10^6, 1 \leq x \leq n \cdot m$), where n and m are the number of rows and columns in the table, and x is the cell number.

Note that the numbers in some test cases do not fit into the 32-bit integer type, so you must use at least the 64-bit integer type of your programming language.

Output

For each test case, output the cell number in the numbering **"by rows"**.

Example

input
5 1 1 1 2 2 3 3 5 11 100 100 7312 1000000 1000000 1000000000000
output
1 2 9 1174 1000000000000