

# Save the Prisoner!

A jail has  $N$  prisoners, and each prisoner has a unique id number,  $S$ , ranging from  $1$  to  $N$ . There are  $M$  sweets that must be distributed to the prisoners.

The jailer decides the fairest way to do this is by sitting the prisoners down in a circle (ordered by ascending  $S$ ), and then, starting with some random  $S$ , distribute one candy at a time to each sequentially numbered prisoner until all  $M$  candies are distributed. For example, if the jailer picks prisoner  $S = 2$ , then his distribution order would be  $(2, 3, 4, 5, \dots, n - 1, n, 1, 2, 3, 4, \dots)$  until all  $M$  sweets are distributed.

But wait—there's a catch—the very last sweet is poisoned! Can you find and print the ID number of the last prisoner to receive a sweet so he can be warned?

## Input Format

The first line contains an integer,  $T$ , denoting the number of test cases.  
The  $T$  subsequent lines each contain **3** space-separated integers:  
 $N$  (the number of prisoners),  $M$  (the number of sweets), and  $S$  (the prisoner ID), respectively.

## Constraints

- $1 \leq T \leq 100$
- $1 \leq N \leq 10^9$
- $1 \leq M \leq 10^9$
- $1 \leq S \leq 10^9$

## Output Format

For each test case, print the ID number of the prisoner who receives the poisoned sweet on a new line.

## Sample Input

```
1
5 2 1
```

## Sample Output

```
2
```

## Explanation

There are  $N = 5$  prisoners and  $M = 2$  sweets. Distribution starts at ID number  $S = 1$ , so prisoner **1** gets the first sweet and prisoner **2** gets the second (last) sweet. Thus, we must warn prisoner **2** about the poison, so we print **2** on a new line.