# **Save the Prisoner!**



A jail has a number of prisoners and a number of treats to pass out to them. Their jailer decides the fairest way to divide the treats is to seat the prisoners around a circular table in sequentially numbered chairs. A chair number will be drawn from a hat. Beginning with the prisoner in that chair, one candy will be handed to each prisoner sequentially around the table until all have been distributed.

The jailer is playing a little joke, though. The last piece of candy looks like all the others, but it tastes *awful*. Determine the chair number occupied by that prisoner so he can be warned.

For example, there are 4 prisoners and 6 pieces of candy. The prisoners arrange themselves in seats numbered 0 to 3. A two is drawn from the hat. Prisoners receive candy at positions 2, 3, 0, 1, 2, 3. The prisoner to be warned sits in chair number 3.

# **Input Format**

The first line contains an integer, t, denoting the number of test cases.

The next t lines each contain 3 space-separated integers:

- n: the number of prisoners
- m: the number of sweets
- s: the chair number to start passing out treats at

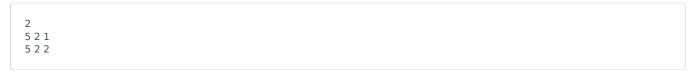
#### **Constraints**

- $1 \le t \le 100$
- $1 < n < 10^9$
- $1 \le m \le 10^9$
- $1 \le s \le N$

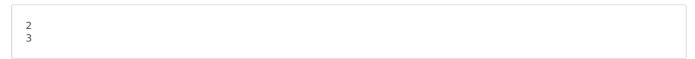
### **Output Format**

For each test case, print the chair number of the prisoner who receives the awful treat on a new line.

#### Sample Input 0



## Sample Output 0



### **Explanation 0**

In first query, there are n=5 prisoners and m=2 sweets. Distribution starts at seat number s=1. Prisoners in seats numbered 1 and 2 get sweets. Warn prisoner 2.

In the second query, distribution starts at seat  ${f 2}$  so prisoners in seats  ${f 2}$  and  ${f 3}$  get sweets. Warn prisoner  ${f 3}$