Practice   Compete   Jobs   Rank   Leaderboard

Dashboard > Algorithms > Implementation > Save the Prisoner!

Badge Progress (Details)

Points: 126 Rank: 270065

Save the Prisoner! 🔖

by sd5869

| Problem | Submissions | Leaderboard | Discussions | Editorial 🔒 |

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 $(2, 3, 4, 5, \ldots, n - 1, n, 1, 2, 3, 4, \ldots)$ 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

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

Output Format

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

Sample Input 0

```
2
5 2 1
5 2 2
```

Sample Output 0

```
2
3
```

Explanation 0

In first query, 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.

Rate This Challenge:

☆ ☆ ☆ ☆ ☆

More

Current Buffer (saved locally, editable)   ⑁ ⟲   Java 7 ⌄   ⤢ ⚙

```java
1  import java.io.*;
2  import java.util.*;
3  import java.text.*;
4  import java.math.*;
5  import java.util.regex.*;
6
7  public class Solution {
8
9      static int saveThePrisoner(int n, int m, int s){
10         // Complete this function
11     }
12
13     public static void main(String[] args) {
14         Scanner in = new Scanner(System.in);
15         int t = in.nextInt();
16         for(int a0 = 0; a0 < t; a0++){
17             int n = in.nextInt();
18             int m = in.nextInt();
19             int s = in.nextInt();
20             int result = saveThePrisoner(n, m, s);
21             System.out.println(result);
22         }
23     }
24 }
25
```

Line: 1 Col: 1

⬆ Upload Code as File   ☐ Test against custom input   Run Code   Submit Code

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature