



Save the Prisoner! ★

1294.8 more points to get your next star!

Rank: 218129 | Points: 905.2/2200

**Your Save the Prisoner! submission got 15.00 points.**[Share](#)[Post](#)

You are now 1294.8 points away from the 6th star for your problem solving badge.

[Try the next challenge](#) | [Try a Random Challenge](#)[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

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 the prisoner who will receive that candy.

Example

 $n = 4$ $m = 6$ $s = 2$

There are **4** prisoners, **6** pieces of candy and distribution starts at chair **2**. The prisoners arrange themselves in seats numbered **1** to **4**. Prisoners receive candy at positions **2, 3, 4, 1, 2, 3**. The prisoner to be warned sits in chair number **3**.

Function Description

Complete the `saveThePrisoner` function in the editor below. It should return an integer representing the chair number of the prisoner to warn.

`saveThePrisoner` has the following parameter(s):

- `int n`: the number of prisoners
- `int m`: the number of sweets
- `int s`: the chair number to begin passing out sweets from

Returns

- `int`: the chair number of the prisoner to warn

Input Format

The first line contains an integer, t , 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

- $1 \leq t \leq 100$
- $1 \leq n \leq 10^9$
- $1 \leq m \leq 10^9$
- $1 \leq s \leq n$

Sample Input 0

```
2
5 2 1
```

5 2 2

Sample Output 0

2
3

Explanation 0

In the 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 **2** so prisoners in seats **2** and **3** get sweets. Warn prisoner **3**.

Sample Input 1

2
7 19 2
3 7 3

Sample Output 1

6
3

Explanation 1

In the first test case, there are $n = 7$ prisoners, $m = 19$ sweets and they are passed out starting at chair $s = 2$. The candies go all around twice and there are **5** more candies passed to each prisoner from seat **2** to seat **6**.

In the second test case, there are $n = 3$ prisoners, $m = 7$ candies and they are passed out starting at seat $s = 3$. They go around twice, and there is one more to pass out to the prisoner at seat **3**.

[Change Theme](#)

Language

Python 3



```

8
9 #
10 # Complete the 'saveThePrisoner' function below.
11 #
12 # The function is expected to return an INTEGER.
13 # The function accepts following parameters:
14 # 1. INTEGER n
15 # 2. INTEGER m
16 # 3. INTEGER s
17 #
18
19 def saveThePrisoner(n, m, s):
20     # Write your code here
21     m = m % n
22     s = (s + m - 1) % n
23     if s == 0:
24         s = n
25     return s
26
27 if __name__ == '__main__':
28     fptr = open(os.environ['OUTPUT_PATH'], 'w')
29
30     t = int(input().strip())
31
32     for i in range(t):

```

```
32     for c in range(c):
33         first_multiple_input = input().rstrip().split()
34
35         n = int(first_multiple_input[0])
36
37         m = int(first_multiple_input[1])
38
39         s = int(first_multiple_input[2])
40
41         result = saveThePrisoner(s, m, c)
```

EMACS

Line: 25 Col: 13

[Upload Code as File](#)☐ Test against custom input[Run Code](#)[Submit Code](#)

You have earned 15.00 points!

You are now 1294.8 points away from the 6th star for your problem solving badge.

4%

905.2/2200



Congratulations

You solved this challenge. Would you like to challenge your friends?

[Next Challenge](#)

✓ **Test case 0**

Compiler Message

✓ Test case 1

Success

✓ Test case 2

✓ Test case 3

✓ Test case 4

Hidden Test Case

Unlock this testcase for 5 hackos.

✓ Test case 5

Unlock

✓ Test case 6