

2. Sprint Training

Pat is an ordinary kid who works hard to be a great runner. As part of training, Pat must run sprints of different intervals on a straight trail. The trail has numbered markers that the coach uses as goals. Pat's coach provides a list of goals to reach in order. Each time Pat starts at, stops at, or passes a marker it is considered a *visit*. Determine the lowest numbered marker that is visited the most times during Pat's day of training.

Example

$n = 5$
 $sprints = [2, 4, 1, 3]$

if the number of markers on the trail, $n = 5$, and assigned $sprints = [2, 4, 1, 3]$, Pat first sprints from position $2 \rightarrow 4$. The next sprint is from position $4 \rightarrow 1$, and then $1 \rightarrow 3$. A marker numbered position p is considered to be *visited* each time Pat either starts or ends a sprint there and each time it is passed while sprinting. The total number of visits to each position in the example is calculated like so:

Total Visits Per Position					
Sprint	1	2	3	4	5
$2 \rightarrow 4$		$\ominus \rightarrow$	\rightarrow	$\rightarrow \ominus$	
$4 \rightarrow 1$	$\ominus \leftarrow$	\leftarrow	\leftarrow	$\leftarrow \ominus$	
$1 \rightarrow 3$	$\ominus \rightarrow$	\rightarrow	$\rightarrow \ominus$		
Total Visits	2	3	3	2	0

Pat has visited markers 2 and 3 a total of 3 times each. Since $2 < 3$, the lowest numbered marker that is visited the most times during Pat's day of training is 2.

Function Description

Complete the function *getMostVisited* in the editor below.

getMostVisited has the following parameter(s):

- int n*: an integer denoting the number of markers along the trail
- int sprints[m]*: an array of integers denoting the sequence of markers to reach, beginning at the marker shown in *sprints[0]*.

Returns:

int: an integer denoting Pat's *most visited* position on the trail after performing all $m - 1$ sprints. If there are multiple such answers, return the smallest one.

Constraints

- $1 \leq n \leq 10^5$
- $2 \leq m \leq 10^5$
- $1 \leq sprints[i] \leq m$ (where $0 \leq i < m$)
- $sprints[i-1] \neq sprints[i]$ (where $0 < i < m$)

▼ Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer n , the number of markers along the path.
The second line contains an integer m , the number of markers in the list of goals.
The next m lines each contain an element $sprints[i]$ where $0 \leq i < m$.

▼ Sample Case 0

Sample Input 0

```
STDIN      Function Parameters
-----
10         → n = 10
4          → sprints[] size m = 4
1          → sprints = [1, 5, 10, 3]
5
10
3
```

Sample Output 0

5

Explanation 0

Given $sprints = [1, 5, 10, 3]$, Pat performs the following sequence of sprints:

Sprint	1	2	3	4	5	6	7	8	9	10
$1 \rightarrow 5$	$\ominus \rightarrow$	\rightarrow	\rightarrow	\rightarrow	$\rightarrow \ominus$					
$5 \rightarrow 10$					$\ominus \rightarrow$	\rightarrow	\rightarrow	\rightarrow	\rightarrow	$\rightarrow \ominus$

Language Python 3

Autocomplete Ready

```
1 > #!/bin/python3 ...
10
11 #
12 # Complete the 'getMostVisited' function below.
13 #
14 # The function is expected to return an INTEGER.
15 # The function accepts following parameters:
16 # 1. INTEGER n
17 # 2. INTEGER_ARRAY sprints
18 #
19
20 def getMostVisited(n, sprints):
21     # Write your code here
22
23 > if __name__ == '__main__': ...
```

Line: 10 Col: 1

Test Results Custom Input Run Code Run Tests Submit

10 → 3			⊙ ←	←	←	←	←	←	←	← ⊙
Total Visits	1	1	2	2	3	2	2	2	2	2

In the table above, Pat visited marker 5 the most.

▼ Sample Case 1

Sample Input 1

```

STDIN      Function Parameters
-----
5          → n = 5
2          → sprints[] size m = 2
1          → sprints = [1, 5]
5

```

Sample Output 1

1

Explanation 1

Given *sprints* = [1, 5], Pat performs the following sprint:

Sprint	1	2	3	4	5
1 → 5	⊙ →	→	→	→	→ ⊙
Total Visits	1	1	1	1	1

In the table above, every marker is visited the same number of times. Return the smallest of these, which is 1.

▼ Sample Case 2

Sample Input 2

```

STDIN      Function Parameters
-----
9          → n = 9
4          → sprints[] size m = 4
9          → sprints = [9, 7, 3, 1]
7
3
1

```

Sample Output 2

3

Explanation 2

Given *sprints* = [9, 7, 3, 1], Pat performs the following sequence of sprints:

Sprint	1	2	3	4	5	6	7	8	9
9 → 7							⊙ ←	←	← ⊙
7 → 3			⊙ ←	←	←	←	← ⊙		
3 → 1	⊙ ←	←	← ⊙						
Total Visits	1	1	2	1	1	1	2	1	1

In the table above, Pat visited positions 3 and 7 the most. Return the smallest of these, which is 3.