



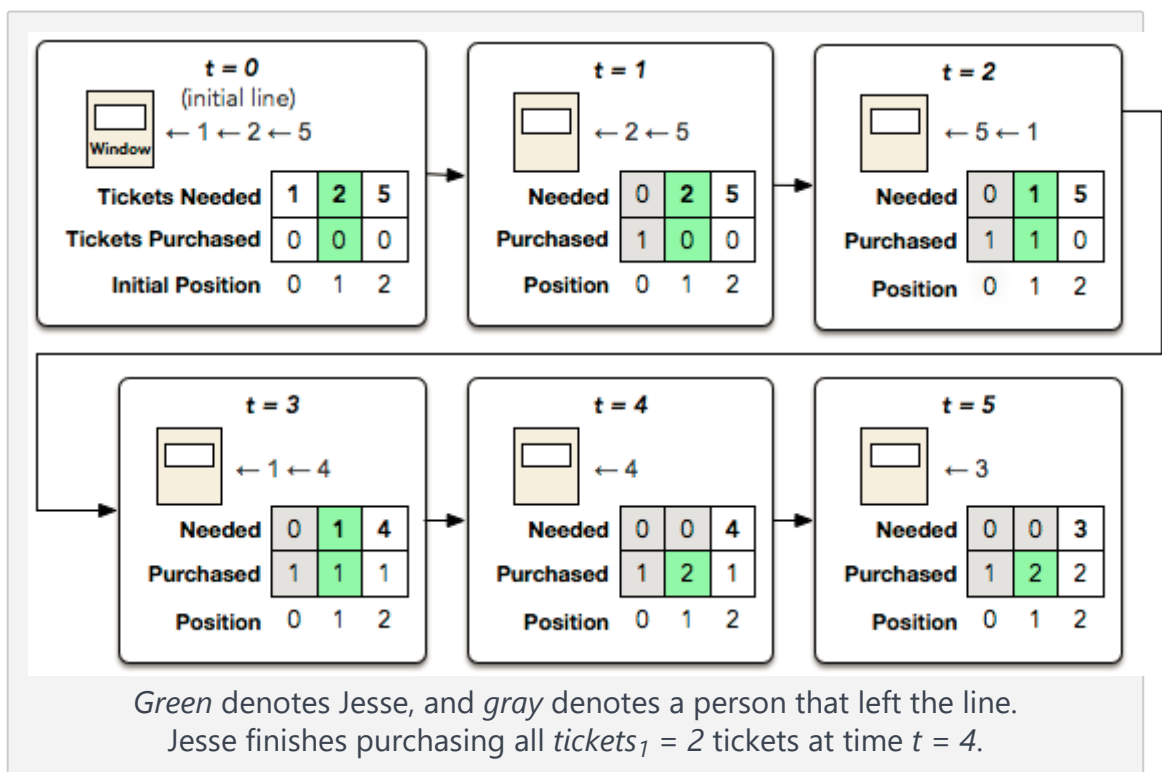
☆ Buying Show Tickets

There are n people standing in line to buy show tickets. Due to high demand, the venue sells tickets according to the following rules:

- The person at the head of the line can buy exactly 1 ticket and must then exit the line.
- If a person needs to purchase additional tickets, they must re-enter the *end* of the line and wait to be sold their next ticket (assume exit and re-entry takes zero seconds).
- Each ticket sale takes exactly 1 second.

We express the initial line of n people as an array, $tickets = [tickets_0, tickets_1, \dots, tickets_{n-1}]$, where each $tickets_i$ denotes the number of tickets that person i wishes to buy. If Jesse is located at position p , how many seconds will it take for him to purchase $tickets_p$ tickets?

For example, if $tickets = [1, 2, 5]$ and $p = 1$, the first five seconds of ticket sales look like this:



Complete the *waitingTime* function in the editor below. It has two parameters:

1. An array, *tickets*, of n positive integers describing the initial sequence of people standing in line. Each $tickets_i$ describes the number of tickets that the person waiting at initial position i needs to purchase.
2. An integer, p , denoting Jesse's position in *tickets*.

The function must return an integer denoting the number of seconds it takes for Jesse to finish purchasing exactly $tickets_p$ tickets.



3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

function:

The first line contains an integer, n , denoting the number of elements in *tickets*.

Each line i of the n subsequent lines contains an integer describing $tickets_i$.

The last line contains an integer, p .

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq tickets[i] \leq 10^9$, where $0 \leq i < n$.
- $0 \leq p < n$

Output Format

Return an integer denoting the number of seconds it takes for Jesse to finish purchasing exactly $tickets_p$ tickets.

Sample Input 0

```
5
2
6
3
4
5
2
```

Sample Output 0

```
12
```

Explanation 0

Given $tickets = [2, 6, 3, 4, 5]$, Jesse's wait time looks like this:

0. $window \leftarrow 2 \leftarrow 6 \leftarrow 3 \leftarrow 4 \leftarrow 5$
1. $window \leftarrow 6 \leftarrow 3 \leftarrow 4 \leftarrow 5 \leftarrow 1$
2. $window \leftarrow 3 \leftarrow 4 \leftarrow 5 \leftarrow 1 \leftarrow 5$
3. $window \leftarrow 4 \leftarrow 5 \leftarrow 1 \leftarrow 5 \leftarrow 2$
4. $window \leftarrow 5 \leftarrow 1 \leftarrow 5 \leftarrow 2 \leftarrow 3$
5. $window \leftarrow 1 \leftarrow 5 \leftarrow 2 \leftarrow 3 \leftarrow 4$
6. $window \leftarrow 5 \leftarrow 2 \leftarrow 3 \leftarrow 4$ (the person at the head of the line in the previous step purchased their last ticket and does not re-enter the line)
7. $window \leftarrow 2 \leftarrow 3 \leftarrow 4 \leftarrow 4$
8. $window \leftarrow 3 \leftarrow 4 \leftarrow 4 \leftarrow 1$
9. $window \leftarrow 4 \leftarrow 4 \leftarrow 1 \leftarrow 2$
10. $window \leftarrow 4 \leftarrow 1 \leftarrow 2 \leftarrow 3$
11. $window \leftarrow 1 \leftarrow 2 \leftarrow 3 \leftarrow 3$
12. $window \leftarrow 2 \leftarrow 3 \leftarrow 3$ (Jesse purchased his last ticket and does not re-enter the line)



3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

Sample Input 1

```
4
1
1
1
1
1
0
```

Sample Output 1

```
1
```

Explanation 1

Given $tickets = [1, 1, 1, 1]$, Jesse's wait time looks like this:

0. $window \leftarrow 1 \leftarrow 1 \leftarrow 1 \leftarrow 1$

1. $window \leftarrow 1 \leftarrow 1 \leftarrow 1$ (Jesse purchased his ticket and does not re-enter the line)

Because it took a total of one second for Jesse to purchase $tickets_0 = 1$ ticket, we return 1.

Sample Input 2

```
4
5
5
2
3
3
```

Sample Output 2

```
11
```

Given $tickets = [5, 5, 2, 3]$, Jesse's wait time looks like this:

0. $window \leftarrow 5 \leftarrow 5 \leftarrow 2 \leftarrow 3$

1. $window \leftarrow 5 \leftarrow 2 \leftarrow 3 \leftarrow 4$

2. $window \leftarrow 2 \leftarrow 3 \leftarrow 4 \leftarrow 4$

3. $window \leftarrow 3 \leftarrow 4 \leftarrow 4 \leftarrow 1$

4. $window \leftarrow 4 \leftarrow 4 \leftarrow 1 \leftarrow 2$

5. $window \leftarrow 4 \leftarrow 1 \leftarrow 2 \leftarrow 3$

6. $window \leftarrow 1 \leftarrow 2 \leftarrow 3 \leftarrow 3$

7. $window \leftarrow 2 \leftarrow 3 \leftarrow 3$ (the person at the head of the line in the previous step purchased their last ticket and does not re-enter the line)



11. window = 2 + 2 (Jesse purchased his last ticket and does not re-enter the line)
Because it took a total of eleven second for Jesse to purchase all $tickets_3 = 3$ tickets, we return 11.

YOUR ANSWER

We recommend you take a quick tour of our editor before you proceed.
The timer will pause up to 90 seconds for the tour.

[Start tour](#)

Draft saved 08:05 am

Original code

Java 7

```
1 ▶ import ↔;
6
7 public class Solution {
8
9 ▼ /*
10  * Complete the function below.
11  */
12
13 ▼ static long waitingTime(int[] tickets, int p) {
14
15
16 }
17
18
19 ▶ public static void main(String[] args) throws IOException{↔}
43 }
```

Line: 12 Col: 1

☐ Test against custom input[Run Code](#)[Submit code & Continue](#)

(You can submit any number of times)

[Download sample test cases](#)
Notepad to edit them on windows.

The input/output files have Unix line endings. Do not use



3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21