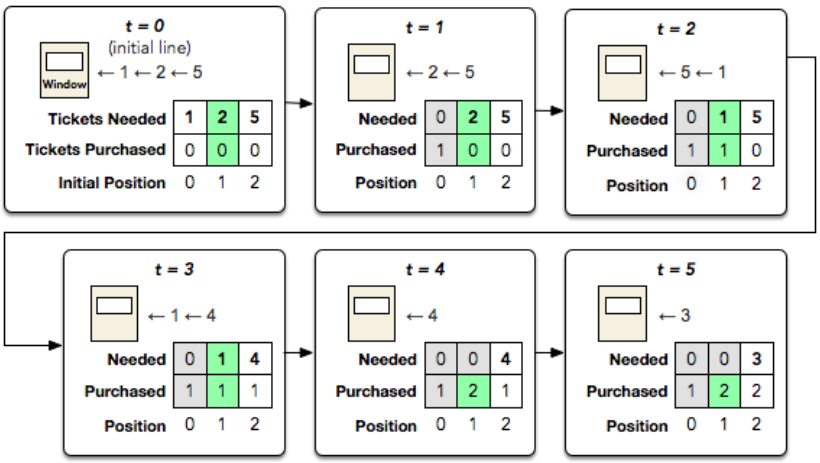
A line has formed to buy tickets for a concert. In order to delay a shortage caused by brokers buying large blocks of tickets, venue management has decided to sell only one ticket at a time. Buyers have to wait through the line again if they want to buy more tickets. Jesse is standing in line and has a number of tickets to purchase.

Given a list of ticket buyers with their numbers of desired tickets, determine how long it will take Jesse to purchase his tickets. Jesse's position in line will be stated, and each transaction takes 1 unit of time. For your purposes, no time is spent moving to the back of the line.

For example, if the zero indexed array of ticket requirements, *tickets = [1, 2, 5]* , and Jesse's position *p = 1*, the first five seconds of ticket sales look like this:



Green denotes Jesse, and *gray* denotes a person that left the line.
Jesse finishes purchasing all $tickets_1 = 2$ tickets at time *t = 4*.

Function Description

Complete the function *waitingTime* in the editor below. The function must return an integer representing the units of time it takes Jesse to purchase his desired number of tickets.

waitingTime has the following parameter(s):

tickets[tickets[0],...tickets[n-1]]: an array of tickets desired by each person at position *tickets[i]*

p: Jesse's position in line

Constraints

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

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 size of the array *tickets*.
The next *n* lines each contain an element *tickets[i]* where 0 ≤ i < n.
The next line contains an integer *p*, Jesse's position in line.

Sample Case 0

Sample Input 0