

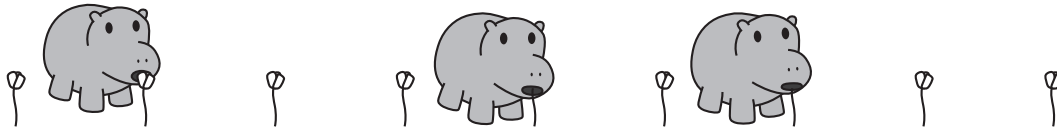
# Giant Hippos

**Input File:** *hippoin.txt*

**Output File:** *hippoout.txt*

**Time Limit:** 1 second

Your backyard has been overrun by preposterously large hippos, and they are eating the tulips!



You have a line of  $N$  tulips, labelled from 1 to  $N$ . There are  $H$  hippos standing next to different tulips ... and they are already eating them! If you don't act fast, they will eat all the tulips in your garden.

You have enough plywood to build up to  $F$  fences. Each fence will be built between two tulips that are next to each other. Hippos cannot go through fences (or over them, or under them, or around them).

Once you have built your fences, the hippos will roam back and forth along the line of tulips, eating all the tulips they can reach without crossing your fences. What is the greatest number of tulips you can save?

## Input

Your program should read from the file `hippoin.txt`.

- The first line of input will contain three space-separated integers  $N$ ,  $H$  and  $F$ : the number of tulips, hippos, and fences you can build respectively.
- The next  $H$  lines will each contain an integer describing which tulip one hippo is eating. These will be in increasing order. No two hippos start at the same tulip.

## Output

Your program should write to the file `hippoout.txt`. Your output file should contain one line with one integer: the greatest number of tulips you can save.

**Sample Input 1**

9 3 3  
2  
5  
7

**Sample Output 1**

4

**Sample Input 2**

7 2 1  
1  
7

**Sample Output 2**

0

**Explanation**



In the first sample case, as illustrated above, if we put a fence between tulips 2 and 3, between tulips 4 and 5 and between tulips 7 and 8, no hippo will be able to reach tulips 3, 4, 8 or 9. There is no placement of fences that will save more tulips, so 4 is the correct answer.

In the second sample case, no matter where you place the one fence you will not be able to save any tulips.

**Constraints**

To evaluate your solution, the judges will run your program against several different input files. All of these files will adhere to the following bounds:

- $1 \leq N \leq 1\,000\,000$
- $1 \leq H \leq 1\,000$  and  $H \leq N$
- $1 \leq F < N$

As some of the test cases will be quite large, you may need to think about how well your solution scales for larger input values. However, not all the cases will be large. In particular:

- For 50% of the available marks,  $H = F = 3$ .