Dan is playing a video game in which his character competes in a hurdle race. Hurdles are of varying heights, and Dan has a maximum height he can jump. There is a magic potion he can take that will increase his maximum height by 1 unit for each dose. How many doses of the potion must he take to be able to jump all of the hurdles.

Given an array of hurdle heights $height$, and an initial maximum height Dan can jump, $k$, determine the minimum number of doses Dan must take to be able to clear all the hurdles in the race.

For example, if $height = [1, 2, 3, 3, 2]$ and Dan can jump 1 unit high naturally, he must take $3 - 1 = 2$ doses of potion to be able to jump all of the hurdles.

**Input Format**

Complete the function $hurdleRace$ in the editor below. The code stub reads the input at passes it to the function. Inputs are in the following format:

The first line contains two space-separated integers $n$ and $k$, the number of hurdles and the maximum height Dan can jump naturally.

The second line contains $n$ space-separated integers $height[i]$ where $0 \leq i < n$.

**Constraints**

- $1 \leq n, k \leq 100$
- $1 \leq height[i] \leq 100$

**Output Format**

Print an integer denoting the minimum doses of magic potion Dan must drink to complete the hurdle race.

**Sample Input 0**

```
5 4
1 6 3 3 2
```

**Sample Output 0**

```
2
```

**Explanation 0**

Dan's character can jump a maximum of $k = 4$ units, but the tallest hurdle has a height of $h_1 = 6$:

To be able to jump all the hurdles, Dan must drink $6 - 4 = 2$ doses.
Sample Input 1

5 7
2 5 4 5 2

Sample Output 1

0

Explanation 1

Dan's character can jump a maximum of $k = 7$ units, which is enough to cross all the hurdles:

Because he can already jump all the hurdles, Dan needs to drink 0 doses.