Dashboard > Algorithms > Implementation > The Hurdle Race

Badge Progress (Details)   Points: 126   Rank: 270060

The Hurdle Race

by ma5termind

| Problem | Submissions | Leaderboard | Discussions | Editorial |

Dan is playing a video game in which his character competes in a hurdle race by jumping over n hurdles with heights $h_0, h_1, \ldots, h_{n-1}$. He can initially jump a maximum height of k units, but he has an unlimited supply of magic beverages that help him jump higher! Each time Dan drinks a magic beverage, the maximum height he can jump during the race increases by 1 unit.

Given n, k, and the heights of all the hurdles, find and print the *minimum* number of magic beverages Dan must drink to complete the race.

Input Format

The first line contains two space-separated integers describing the respective values of n (the number of hurdles) and k (the maximum height he can jump without consuming any beverages).
The second line contains n space-separated integers describing the respective values of $h_0, h_1, \ldots, h_{n-1}$.

Constraints

- $1 \le n, k \le 100$
- $1 \le h_i \le 100$

Output Format

Print an integer denoting the *minimum* number of magic beverages Dan must drink to complete the hurdle race.
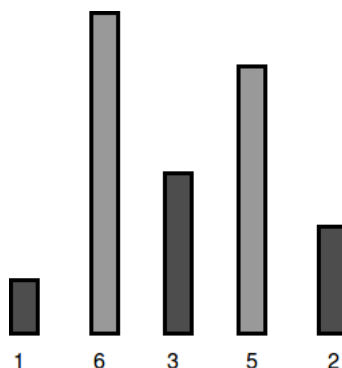
Sample Input 0

```
5 4
1 6 3 5 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 $h_1 = 6$:

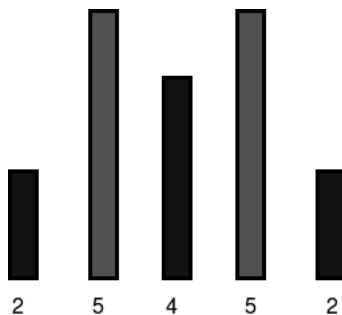To be able to jump all the hurdles, Dan must drink $6 - 4 = 2$ magic beverages.

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 magic beverages.

f   y   in

Submissions:32494

Max Score:15
Difficulty: Easy

Rate This Challenge:
☆ ☆ ☆ ☆ ☆

More

Current Buffer (saved locally, editable)   Java 7

```java
1  import java.io.*;
2  import java.util.*;
3  import java.text.*;
4  import java.math.*;
5  import java.util.regex.*;
6
7  public class Solution {
8
9      public static void main(String[] args) {
10         Scanner in = new Scanner(System.in);
11         int n = in.nextInt();
12         int k = in.nextInt();
13         int[] height = new int[n];
14         for(int height_i=0; height_i < n; height_i++){
15             height[height_i] = in.nextInt();
16         }
17         // your code goes here
18     }
19 }
20
```

Line: 1 Col: 1

⬆ Upload Code as File   ☐ Test against custom input

| Run Code | | Submit Code |

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature