



# Jesse and Cookies

by [vatsalchanana](#)

Problem

Submissions

Leaderboard

Discussions

Jesse loves cookies. He wants the sweetness of all his cookies to be greater than value  $K$ . To do this, Jesse repeatedly mixes two cookies with the least sweetness. He creates a special combined cookie with:

$\text{sweetness} = (1 \times \text{Least sweet cookie} + 2 \times \text{2nd least sweet cookie})$ .

He repeats this procedure until all the cookies in his collection have a sweetness  $\geq K$ .

You are given Jesse's cookies. Print the number of operations required to give the cookies a sweetness  $\geq K$ . Print  $-1$  if this isn't possible.

## Input Format

The first line consists of integers  $N$ , the number of cookies and  $K$ , the minimum required sweetness, separated by a space. The next line contains  $N$  integers describing the array  $A$  where  $A_i$  is the sweetness of the  $i^{\text{th}}$  cookie in Jesse's collection.

## Constraints

$$1 \leq N \leq 10^6$$

$$0 \leq K \leq 10^9$$

$$0 \leq A_i \leq 10^6$$

## Output Format

Output the number of operations that are needed to increase the cookie's sweetness  $\geq K$ .

Output  $-1$  if this isn't possible.

## Sample Input

```
6 7
1 2 3 9 10 12
```

## Sample Output

```
2
```

## Explanation

Combine the first two cookies to create a cookie with  $\text{sweetness} = 1 \times 1 + 2 \times 2 = 5$

After this operation, the cookies are **3, 5, 9, 10, 12**.

Then, combine the cookies with sweetness **3** and sweetness **5**, to create a cookie with resulting  $\text{sweetness} = 1 \times 3 + 2 \times 5 = 13$

Now, the cookies are **9, 10, 12, 13**.

All the cookies have a sweetness  $\geq 7$ .

Thus, **2** operations are required to increase the sweetness.



Submissions: [9248](#)




Max Score: 25

Difficulty: Easy

Rate This Challenge:

[More](#)

Current Buffer (saved locally, editable)  

Java 7   

```
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         /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
11     }
12 }
```

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](#)