

# CCSC:MW 2020 Programming Competition

## Problem 5: Juiciness of juices

Sunny loves packet juice. He is a picky eater and wants to make sure that all the juices have a juiciness level greater than a certain value,  $K$ . In order to achieve this, Sunny repeatedly mixes two juices with the least juiciness. This creates a cocktail juice with juiciness expressed by the following equation:

$$\text{Juiciness} = (1 \times \text{Least juicy juice} + 2 \times \text{2nd least juicy juice}).$$

This process is repeated until all the juices in his storage have a juiciness level  $\geq K$ .

Can you find out the number of operations (number of times the aforementioned process needs to be repeated) required to make sure all the juices have a juiciness level  $\geq K$ ? Print -1 if this isn't possible.

### Input:

First, you need to take input two integers:  $N$ : the number of juices and  $K$ : the minimum required juiciness. Next input  $N$  integers describing the juiciness of each juice in Sunny's collection.

### Output

Output should be the minimum number of operations needed to increase all juiciness of all juices  $\geq K$ . Print -1 if this isn't possible.

### Example 1

6 7  
9 3 1 2 10 12

The following is the correct output for the input above:

2

### Explanation:

Combine the two juices with least juiciness to create a juice with juiciness =  $1 \times 1 + 2 \times 2 = 5$ .

After this operation, the juiciness of juices are (in non-decreasing order) 3 5 9 10 12.

Then, combine the juices with juiciness 3 and juiciness 5, to create a juice with resulting juiciness =  $1 \times 3 + 2 \times 5 = 13$ .

Now, the juices are 9 10 12 13.

All the juices have a juiciness  $\geq 7$ .

Thus, 2 operations are required to increase the juiciness.