

The store nearby has a new deal for the holidays - one of every **K** products you buy is free! In other words - if you buy **N** items, the cheapest **N/K** (rounded down) from them are given to you for free. But this deal has a critical bug - with proper splitting of the items into different groups, and then buying these groups independently from each other, you can make the store give you some items for free that it wouldn't otherwise.

For example, if **K=2** and you want to buy items with prices \$100, \$20, \$10 and \$50 at once, the store would give you the items priced \$20 and \$10 for free (\$30 saved). On the other hand, if you first bought items priced \$50 and \$100 you would get the \$50 for free. Then you can buy the \$20 and \$10 items and get another \$10 for free. This would be \$50 + \$10 = \$60 saved instead of \$30.

Make a program that calculates the **biggest** amount you can save while splitting the items into **as few groups as possible**.

Input

From the first line of the file `shopcheat.in` **N** and **K** are entered. On the next line **N** numbers are entered - the prices of each item. All numbers are integers.

Output

On a single line in the file `shopcheat.out`, output the biggest amount you can save and the lowest number of groups you have to split your items into.

Constraints

$$1 \leq N \leq 2000$$

$$1 \leq \text{price of item} \leq \text{sum of the prices of all items} \leq 10^9$$

Time limit: 0.2 seconds

Memory limit: 256 MB

Example

Вход (<code>shopcheat.in</code>)	Исход (<code>shopcheat.out</code>)
4 2 100 20 10 50	60 2
19 3 7 9 7 3 7 10 2 10 10 7 6 5 1 4 8 7 7 6 5	38 6