Bots buying beer

In the year 3210, humans are so lazy to shop for themselves, so bots are doing it instead. Today, exactly N cans of beer are in the marketplace. M bots are willing to buy exactly one can of beer.

Suppose that the price of one can of beer is X. Each bot i has a *desired price* p_i for a can of beer. The bot will buy a can of beer if the price is less than or equal to p_i i.e. $X \leq p_i$. However, if the number of bots that are going to buy a can of beer is larger than N, we have no choice but only sell N cans. (so we can sell at most N cans)

The marketplace wants to *maximize* the revenue, which is defined by (price of the beer) × (number of bots that will actually buy a can of beer). As marketplace is a robot, it knows all values of p_i, and it needs to choose the *smallest* X such that the revenue is the largest.

Given N, M, and the desired prices $p_1, \cdots, p_n$, write a program that calculates the smallest price he should set for one can of beer to *maximize* the amount of money the marketplace gets.

Input

Your input consists of an arbitrary number of records, but no more than 10.

For each input record, the first line contains two integers N $(1 \leq N \leq 100{,}000)$ and M $(1 \leq M \leq 100{,}000)$ The next line contain $p_1, p_2, \cdots, p_M$ $(1 \leq p_i \leq 10^{12})$, each separated by a space.

The end of input is indicated by a line containing only the value -1.

Output

For each input record, print two integers separated by a space. The first integer should be the smallest price that the marketplace should choose to maximize the revenue. The second integer should be the amount of money that the marketplace earns.

Example

Standard input	Standard output

5 4
2 8 10 7
6 6
2 2 3 3 3 3
-1
	7 21
	2 12

Notes

Note that in the second example, if the marketplace sets the price as 2, the revenue is $2\times 6 = 12$, and if the price is 3, the revenue is also $3\times 4 = 12$. Both have the same revenue, but you have to print the *smallest price* that *maximize the revenue*, so the price should be 2.

Time Limit

2 seconds.