Codeforces Round #251 (Div. 2)

A. Devu, the Singer and Churu, the Joker

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Devu is a renowned classical singer. He is invited to many big functions/festivals. Recently he was invited to "All World Classical Singing Festival". Other than Devu, comedian Churu was also invited.

Devu has provided organizers a list of the songs and required time for singing them. He will sing n songs, i^{th} song will take t_i minutes exactly.

The Comedian, Churu will crack jokes. All his jokes are of 5 minutes exactly.

People have mainly come to listen Devu. But you know that he needs rest of 10 minutes after each song. On the other hand, Churu being a very active person, doesn't need any rest.

You as one of the organizers should make an optimal schedule for the event. For some reasons you must follow the conditions:

- The duration of the event must be no more than d minutes;
- Devu must complete all his songs;
- With satisfying the two previous conditions the number of jokes cracked by Churu should be as many as possible.

If it is not possible to find a way to conduct all the songs of the Devu, output -1. Otherwise find out maximum number of jokes that Churu can crack in the grand event.

Input
The first line contains two space separated integers n, d ($1 \le n \le 100$; $1 \le d \le 10000$). The second line contains n space-separated integers: $t_1, t_2, ..., t_n$ ($1 \le t_i \le 100$).

Output
If there is no way to conduct all the songs of Devu, output -1. Otherwise output the maximum number of jokes that Churu can crack in the grand event.

Examples

input
3 30 2 2 1
output
5

input
3 20 2 1 1
output
-1

Note
Consider the first example. The duration of the event is 30 minutes. There could be maximum 5 jokes in the following way:

- First Churu cracks a joke in 5 minutes.
- Then Devu performs the first song for 2 minutes.
- Then Churu cracks 2 jokes in 10 minutes.
- Now Devu performs second song for 2 minutes.
- Then Churu cracks 2 jokes in 10 minutes.
- Now finally Devu will perform his last song in 1 minutes.

Total time spent is $5 + 2 + 10 + 2 + 10 + 1 = 30$ minutes.

Consider the second example. There is no way of organizing Devu's all songs. Hence the answer is -1.

B. Devu, the Dumb Guy

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Devu is a dumb guy, his learning curve is very slow. You are supposed to teach him n subjects, the i^{th} subject has c_i chapters. When you teach him, you are supposed to teach all the chapters of a subject continuously.

Let us say that his initial per chapter learning power of a subject is x hours. In other words he can learn a chapter of a particular subject in x hours.

Well Devu is not complete dumb, there is a good thing about him too. If you teach him a subject, then time required to teach any chapter of the next subject will require exactly 1 hour less than previously required (see the examples to understand it more clearly). Note that his per chapter learning power can not be less than 1 hour.

You can teach him the n subjects in any possible order. Find out minimum amount of time (in hours) Devu will take to understand all the subjects and you will be free to do some enjoying task rather than teaching a dumb guy.

Please be careful that answer might not fit in 32 bit data type.

Input

The first line will contain two space separated integers n, x ($1 \leq n, x \leq 10^5$). The next line will contain n space separated integers: $c_1, c_2, \ldots, c_n$ ($1 \leq c_i \leq 10^5$).

Output

Output a single integer representing the answer to the problem.

Examples

input
2 3 4 1
output
11

input
4 2 5 1 2 1
output
10

input
3 3 1 1 1
output
6

Note

Look at the first example. Consider the order of subjects: 1, 2. When you teach Devu the first subject, it will take him 3 hours per chapter, so it will take 12 hours to teach first subject. After teaching first subject, his per chapter learning time will be 2 hours. Now teaching him second subject will take $2 \times 1 = 2$ hours. Hence you will need to spend $12 + 2 = 14$ hours.

Consider the order of subjects: 2, 1. When you teach Devu the second subject, then it will take him 3 hours per chapter, so it will take $3 \times 1 = 3$ hours to teach the second subject. After teaching the second subject, his per chapter learning time will be 2 hours. Now teaching him the first subject will take $2 \times 4 = 8$ hours. Hence you will need to spend 11 hours.

So overall, minimum of both the cases is 11 hours.

Look at the third example. The order in this example doesn't matter. When you teach Devu the first subject, it will take him 3 hours per chapter. When you teach Devu the second subject, it will take him 2 hours per chapter. When you teach Devu the third subject, it will take him 1 hours per chapter. In total it takes 6 hours.

C. Devu and Partitioning of the Array

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Devu being a small kid, likes to play a lot, but he only likes to play with arrays. While playing he came up with an interesting question which he could not solve, can you please solve it for him?

Given an array consisting of distinct integers. Is it possible to partition the whole array into k disjoint non-empty parts such that p of the parts have even sum (each of them must have even sum) and remaining k - p have odd sum? (note that parts need not to be continuous).

If it is possible to partition the array, also give any possible way of valid partitioning.

Input

The first line will contain three space separated integers n, k, p ($1 \le k \le n \le 10^5$; $0 \le p \le k$). The next line will contain n space-separated distinct integers representing the content of array a: $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^9$).

Output

In the first line print "YES" (without the quotes) if it is possible to partition the array in the required way. Otherwise print "NO" (without the quotes).

If the required partition exists, print k lines after the first line. The i^{th} of them should contain the content of the i^{th} part. Print the content of the part in the line in the following way: firstly print the number of elements of the part, then print all the elements of the part in arbitrary order. There must be exactly p parts with even sum, each of the remaining k - p parts must have odd sum.

As there can be multiple partitions, you are allowed to print any valid partition.

Examples

input
5 5 3 2 6 10 5 9
output
YES 1 9 1 5 1 10 1 6 1 2

input
5 5 3 7 14 2 9 5
output
NO

input
5 3 1 1 2 3 7 5
output
YES 3 5 1 3 1 7 1 2

D. Devu and his Brother

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Devu and his brother love each other a lot. As they are super geeks, they only like to play with arrays. They are given two arrays a and b by their father. The array a is given to Devu and b to his brother.

As Devu is really a naughty kid, he wants the minimum value of his array a should be at least as much as the maximum value of his brother's array b.

Now you have to help Devu in achieving this condition. You can perform multiple operations on the arrays. In a single operation, you are allowed to decrease or increase any element of any of the arrays by 1. Note that you are allowed to apply the operation on any index of the array multiple times.

You need to find minimum number of operations required to satisfy Devu's condition so that the brothers can play peacefully without fighting.

Input

The first line contains two space-separated integers n, m ($1 \le n, m \le 10^5$). The second line will contain n space-separated integers representing content of the array a ($1 \le a_i \le 10^9$). The third line will contain m space-separated integers representing content of the array b ($1 \le b_i \le 10^9$).

Output

You need to output a single integer representing the minimum number of operations needed to satisfy Devu's condition.

Examples

input
2 2 2 3 3 5
output
3

input
3 2 1 2 3 3 4
output
4

input
3 2 4 5 6 1 2
output
0

Note

In example 1, you can increase a_1 by 1 and decrease b_2 by 1 and then again decrease b_2 by 1. Now array a will be [3; 3] and array b will also be [3; 3]. Here minimum element of a is at least as large as maximum element of b. So minimum number of operations needed to satisfy Devu's condition are 3.

In example 3, you don't need to do any operation, Devu's condition is already satisfied.

E. Devu and Birthday Celebration

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Today is Devu's birthday. For celebrating the occasion, he bought n sweets from the nearby market. He has invited his f friends. He would like to distribute the sweets among them. As he is a nice guy and the occasion is great, he doesn't want any friend to be sad, so he would ensure to give at least one sweet to each friend.

He wants to celebrate it in a unique style, so he would like to ensure following condition for the distribution of sweets. Assume that he has distributed n sweets to his friends such that i^{th} friend is given a_i sweets. He wants to make sure that there should not be any positive integer $x > 1$, which divides every a_i.

Please find the number of ways he can distribute sweets to his friends in the required way. Note that the order of distribution is important, for example [1, 2] and [2, 1] are distinct distributions. As the answer could be very large, output answer modulo 1000000007 $(10^9 + 7)$.

To make the problem more interesting, you are given q **queries**. Each query contains an n, f pair. For each query please output the required number of ways modulo 1000000007 $(10^9 + 7)$.

Input

The first line contains an integer q representing the number of queries $(1 \le q \le 10^5)$. Each of the next q lines contains two space space-separated integers n, f $(1 \le f \le n \le 10^5)$.

Output

For each query, output a single integer in a line corresponding to the answer of each query.

Examples

input
5 6 2 7 2 6 3 6 4 7 4
output
2 6 9 10 20

Note

For first query: $n = 6, f = 2$. Possible partitions are [1, 5] and [5, 1].

For second query: $n = 7, f = 2$. Possible partitions are [1, 6] and [2, 5] and [3, 4] and [4, 3] and [5, 3] and [6, 1]. So in total there are 6 possible ways of partitioning.