Problem A. Maximum Median

Time Limit 2000 ms
Mem Limit 262144 kB

You are given an array a of n integers, where n is odd. You can make the following operation with it:

- Choose one of the elements of the array (for example a_i) and increase it by 1 (that is, replace it with $a_i + 1$).

You want to make the median of the array the largest possible using at most k operations.

The median of the odd-sized array is the middle element after the array is sorted in non-decreasing order. For example, the median of the array $[1, 5, 2, 3, 5]$ is 3.

Input

The first line contains two integers n and k ($1 \le n \le 2 \cdot 10^5$, n is odd, $1 \le k \le 10^9$) — the number of elements in the array and the largest number of operations you can make.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

Output

Print a single integer — the maximum possible median after the operations.

Examples

Input	Output
3 2 1 3 5	5

Input	Output
5 5 1 2 1 1 1	3

Input	Output
7 7 4 1 2 4 3 4 4	5

Note

In the first example, you can increase the second element twice. Than array will be $[1, 5, 5]$ and it's median is 5.

In the second example, it is optimal to increase the second number and than increase third and fifth. This way the answer is 3.

In the third example, you can make four operations: increase first, fourth, sixth, seventh element. This way the array will be $[5, 1, 2, 5, 3, 5, 5]$ and the median will be 5.

Peter received money from his parents this week and wants to spend it all buying books. But he does not read a book so fast, because he likes to enjoy every single word while he is reading. In this way, it takes him a week to finish a book.

As Peter receives money every two weeks, he decided to buy two books, then he can read them until receive more money. As he wishes to spend all the money, he should choose two books whose prices summed up are equal to the money that he has. It is a little bit difficult to find these books, so Peter asks your help to find them.

Input

Each test case starts with $2 \leq N \leq 10000$, the number of available books. Next line will have N integers, representing the price of each book, a book costs less than 1000001. Then there is another line with an integer M, representing how much money Peter has. There is a blank line after each test case. The input is terminated by end of file (EOF).

Output

For each test case you must print the message: '`Peter should buy books whose prices are` i `and` j`.`', where i and j are the prices of the books whose sum is equal do M and $i \leq j$. You can consider that is always possible to find a solution, if there are multiple solutions print the solution that minimizes the difference between the prices i and j. After each test case you must print a blank line.

Sample Input

```
2
40 40
80

5
10 2 6 8 4
10
```

Sample Output

```
Peter should buy books whose prices are 40 and 40.

Peter should buy books whose prices are 4 and 6.
```

Problem C. Factory Machines

Time Limit 1000 ms
Mem Limit 524288 kB

A factory has n machines which can be used to make products. Your goal is to make a total of t products.

For each machine, you know the number of seconds it needs to make a single product. The machines can work simultaneously, and you can freely decide their schedule.

What is the shortest time needed to make t products?

Input

The first input line has two integers n and t: the number of machines and products.

The next line has n integers $k_1, k_2, \ldots, k_n$: the time needed to make a product using each machine.

Output

Print one integer: the minimum time needed to make t products.

Constraints

- $1 \leq n \leq 2 \cdot 10^5$
- $1 \leq t \leq 10^9$
- $1 \leq k_i \leq 10^9$

Example

Input	Output
3 7 3 2 5	8

Explanation: Machine 1 makes two products, machine 2 makes four products and machine 3 makes one product.

Problem D. Magic Powder - 1

Time Limit 1000 ms
Mem Limit 262144 kB

This problem is given in two versions that differ only by constraints. If you can solve this problem in large constraints, then you can just write a single solution to the both versions. If you find the problem too difficult in large constraints, you can write solution to the simplified version only.

Waking up in the morning, Apollinaria decided to bake cookies. To bake one cookie, she needs n ingredients, and for each ingredient she knows the value a_i — how many grams of this ingredient one needs to bake a cookie. To prepare one cookie Apollinaria needs to use all n ingredients.

Apollinaria has b_i gram of the i-th ingredient. Also she has k grams of a magic powder. Each gram of magic powder can be turned to exactly 1 gram of any of the n ingredients and can be used for baking cookies.

Your task is to determine the maximum number of cookies, which Apollinaria is able to bake using the ingredients that she has and the magic powder.

Input

The first line of the input contains two positive integers n and k ($1 \le n, k \le 1000$) — the number of ingredients and the number of grams of the magic powder.

The second line contains the sequence $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 1000$), where the i-th number is equal to the number of grams of the i-th ingredient, needed to bake one cookie.

The third line contains the sequence $b_1, b_2, \ldots, b_n$ ($1 \le b_i \le 1000$), where the i-th number is equal to the number of grams of the i-th ingredient, which Apollinaria has.

Output

Print the maximum number of cookies, which Apollinaria will be able to bake using the ingredients that she has and the magic powder.

Examples

Input	Output
3 1 2 1 4 11 3 16	4

Input	Output
4 3 4 3 5 6 11 12 14 20	3

Note

In the first sample it is profitably for Apollinaria to make the existing 1 gram of her magic powder to ingredient with the index 2, then Apollinaria will be able to bake 4 cookies.

In the second sample Apollinaria should turn 1 gram of magic powder to ingredient with the index 1 and 1 gram of magic powder to ingredient with the index 3. Then Apollinaria will be able to bake 3 cookies. The remaining 1 gram of the magic powder can be left, because it can't be used to increase the answer.

Problem E. Magic Powder - 2

Time Limit 1000 ms
Mem Limit 262144 kB

The term of this problem is the same as the previous one, the only exception — increased restrictions.

Input

The first line contains two positive integers n and k ($1 \le n \le 100\,000$, $1 \le k \le 10^9$) — the number of ingredients and the number of grams of the magic powder.

The second line contains the sequence $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$), where the i-th number is equal to the number of grams of the i-th ingredient, needed to bake one cookie.

The third line contains the sequence $b_1, b_2, \ldots, b_n$ ($1 \le b_i \le 10^9$), where the i-th number is equal to the number of grams of the i-th ingredient, which Apollinaria has.

Output

Print the maximum number of cookies, which Apollinaria will be able to bake using the ingredients that she has and the magic powder.

Examples

Input	Output
1 1000000000 1 1000000000	2000000000

Input	Output
10 1 1000000000 1000000000 1000000000 1000000000 1000000000 1000000000 1000000000 1000000000 1000000000 1000000000 1 1 1 1 1 1 1 1 1 1	0

Input	Output
3 1 2 1 4 11 3 16	4

Input	Output
4 3 4 3 5 6 11 12 14 20	3

Problem F. Binary search - 1

Time Limit 2000 ms
Mem Limit 131072 kB

A sorted array of n integers is given. You need to answer q queries: how many times a given number x appears in the array.

Input

The first line contains two numbers n and q $(n, q \leq 10^6)$. The second line contains n integers sorted in increasing order. Each of the following q lines contains a single value x. All numbers in the array do not exceed 10^9 in absolute value.

Output

For each value of x, print the number of times it occurs in the array in a separate line.

Examples

Input	Output
6 3 2 4 4 8 11 14 10 4 2	0 2 1

Input	Output
10 5 0 0 1 1 2 3 4 5 6 8 8 2 0 12 10	1 1 2 0 0

Problem G. Queries about less or equal elements

Time Limit 2000 ms
Mem Limit 262144 kB

You are given two arrays of integers a and b. For each element of the second array b_j you should find the number of elements in array a that are less than or equal to the value b_j.

Input

The first line contains two integers n, m ($1 \le n, m \le 2 \cdot 10^5$) — the sizes of arrays a and b.

The second line contains n integers — the elements of array a ($-10^9 \le a_i \le 10^9$).

The third line contains m integers — the elements of array b ($-10^9 \le b_j \le 10^9$).

Output

Print m integers, separated by spaces: the j-th of which is equal to the number of such elements in array a that are less than or equal to the value b_j.

Examples

Input	Output
5 4 1 3 5 7 9 6 4 2 8	3 2 1 4

Input	Output
5 5 1 2 1 2 5 3 1 4 1 5	4 2 4 2 5

Problem H. Burning Midnight Oil

Time Limit	2000 ms
Mem Limit	262144 kB
Input File	stdin
Output File	stdout

One day a highly important task was commissioned to Vasya — writing a program in a night. The program consists of n lines of code. Vasya is already exhausted, so he works like that: first he writes v lines of code, drinks a cup of tea, then he writes as much as $\lfloor \frac{v}{k} \rfloor$ lines, drinks another cup of tea, then he writes $\lfloor \frac{v}{k^2} \rfloor$ lines and so on: $\lfloor \frac{v}{k^3} \rfloor$, $\lfloor \frac{v}{k^4} \rfloor$, $\lfloor \frac{v}{k^5} \rfloor$, ...

The expression $\lfloor \frac{a}{b} \rfloor$ is regarded as the integral part from dividing number a by number b.

The moment the current value $\lfloor \frac{v}{k^p} \rfloor$ equals 0, Vasya immediately falls asleep and he wakes up only in the morning, when the program should already be finished.

Vasya is wondering, what minimum allowable value v can take to let him write **not less** than n lines of code before he falls asleep.

Input

The input consists of two integers n and k, separated by spaces — the size of the program in lines and the productivity reduction coefficient, $1 \le n \le 10^9$, $2 \le k \le 10$.

Output

Print the only integer — the minimum value of v that lets Vasya write the program in one night.

Examples

Input	Output
7 2	4

Input	Output
59 9	54

Note

In the first sample the answer is $v = 4$. Vasya writes the code in the following portions: first 4 lines, then 2, then 1, and then Vasya falls asleep. Thus, he manages to write $4 + 2 + 1 = 7$ lines in a night and complete the task.

In the second sample the answer is $v = 54$. Vasya writes the code in the following portions: 54, 6. The total sum is $54 + 6 = 60$, that's even more than $n = 59$.

Problem I. Binary search

Time Limit 5000 ms
Mem Limit 262144 kB

A sorted array of n integers is given. It is necessary to answer q queries: whether the given number x is present in the array.

Input

The first line contains two integers n and q $(n, q \leq 10^6)$. The second line contains n integers sorted in ascending order. Each of the following q lines contains one number x. All numbers in the array do not exceed 10^9 by absolute value.

Output

For each query, print **"YES"** on a separate line if the number x is present in the array, and **"NO"** otherwise.

Examples

Input	Output
6 3 2 4 4 8 11 14 10 4 2	NO YES YES

Input	Output
10 5 0 0 1 1 2 3 4 5 6 8 8 2 0 12 10	YES YES YES NO NO

Problem J. Eko

Time Limit	1000 ms
Mem Limit	1572864 kB
Code Length Limit	50000 B
OS	Linux

Lumberjack Mirko needs to chop down **M** metres of wood. It is an easy job for him since he has a nifty new woodcutting machine that can take down forests like wildfire. However, Mirko is only allowed to cut a single row of trees.

Mirko"s machine works as follows: Mirko sets a height parameter **H** (in metres), and the machine raises a giant sawblade to that height and cuts off all tree parts higher than **H** (of course, trees not higher than **H** meters remain intact). Mirko then takes the parts that were cut off. For example, if the tree row contains trees with heights of 20, 15, 10, and 17 metres, and Mirko raises his sawblade to 15 metres, the remaining tree heights after cutting will be 15, 15, 10, and 15 metres, respectively, while Mirko will take 5 metres off the first tree and 2 metres off the fourth tree (7 metres of wood in total).

Mirko is **ecologically** minded, so he doesn"t want to cut off more wood than necessary. That"s why he wants to set his sawblade as high as possible. Help Mirko find the **maximum integer height** of the sawblade that still allows him to cut off **at least M** metres of wood.

Input

The first line of input contains two space-separated positive integers, **N** (the number of trees, $1 \leq N \leq 1\,000\,000$) and **M** (Mirko"s required wood amount, $1 \leq M \leq 2\,000\,000\,000$).

The second line of input contains **N** space-separated positive integers less than 1 000 000 000, the heights of each tree (in metres). The sum of all heights will exceed **M**, thus Mirko will always be able to obtain the required amount of wood.

Output

The first and only line of output must contain the required height setting.

Example

Input	Output
4 7 20 15 10 17	15

Input	Output
5 20 4 42 40 26 46	36

Problem K. Copying Books

Time Limit	5000 ms
Mem Limit	1572864 kB
Code Length Limit	50000 B
OS	Linux

Before the invention of book-printing, it was very hard to make a copy of a book. All the contents had to be re-written by hand by so called *scribers*. The scriber had been given a book and after several months he finished its copy. One of the most famous scribers lived in the 15th century and his name was Xaverius Endricus Remius Ontius Xendrianus (*Xerox*). Anyway, the work was very annoying and boring. And the only way to speed it up was to hire more scribers.

Once upon a time, there was a theater ensemble that wanted to play famous Antique Tragedies. The scripts of these plays were divided into many books and actors needed more copies of them, of course. So they hired many scribers to make copies of these books. Imagine you have m books (numbered $1, 2 \ldots m$) that may have different number of pages ($p_1, p_2 \ldots p_m$) and you want to make one copy of each of them. Your task is to divide these books among k scribes, $k \le m$. Each book can be assigned to a single scriber only, and every scriber must get a continuous sequence of books. That means, there exists an increasing succession of numbers $0 = b_0 < b_1 < b_2 \ldots < b_{k-1} \le b_k = m$ such that i-th scriber gets a sequence of books with numbers between $b_{i-1}+1$ and b_i. The time needed to make a copy of all the books is determined by the scriber who was assigned the most work. Therefore, our goal is to minimize the maximum number of pages assigned to a single scriber. Your task is to find the optimal assignment.

Input

The input consists of N cases (equal to about 200). The first line of the input contains only positive integer N. Then follow the cases. Each case consists of exactly two lines. At the first line, there are two integers m and k, $1 \le k \le m \le 500$. At the second line, there are integers $p_1, p_2 \ldots p_m$ separated by spaces. All these values are positive and less than 10000000.

Output

For each case, print exactly one line. The line must contain the input succession $p_1, p_2, \ldots p_m$ divided into exactly k parts such that the maximum sum of a single part should be as

small as possible. Use the slash character ('/') to separate the parts. There must be exactly one space character between any two successive numbers and between the number and the slash.

If there is more than one solution, print the one that minimizes the work assigned to the first scriber, then to the second scriber etc. But each scriber must be assigned at least one book.

Example

Input	Output
2 9 3 100 200 300 400 500 600 700 800 900 5 4 100 100 100 100 100	100 200 300 400 500 / 600 700 / 800 900 100 / 100 / 100 / 100 100