CODEFORCES β
Sponsored by Telegram

Manthan, Codefest 16

A. Ebony and Ivory

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Dante is engaged in a fight with "The Savior". Before he can fight it with his sword, he needs to break its shields. He has two guns, Ebony and Ivory, each of them is able to perform any non-negative number of shots.

For every bullet that hits the shield, Ebony deals a units of damage while Ivory deals b units of damage. In order to break the shield Dante has to deal **exactly** c units of damage. Find out if this is possible.

Input

The first line of the input contains three integers a, b, c $(1 \le a, b \le 100, 1 \le c \le 10\,000)$ — the number of units of damage dealt by Ebony gun and Ivory gun, and the total number of damage required to break the shield, respectively.

Output

Print "`Yes`" (without quotes) if Dante can deal exactly c damage to the shield and "`No`" (without quotes) otherwise.

Examples

input
```
4 6 15
```
output
```
No
```

input
```
3 2 7
```
output
```
Yes
```

input
```
6 11 6
```
output
```
Yes
```

Note

In the second sample, Dante can fire 1 bullet from Ebony and 2 from Ivory to deal exactly $1 \cdot 3 + 2 \cdot 2 = 7$ damage. In the third sample, Dante can fire 1 bullet from ebony and no bullets from ivory to do $1 \cdot 6 + 0 \cdot 11 = 6$ damage.

B. A Trivial Problem

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Mr. Santa asks all the great programmers of the world to solve a trivial problem. He gives them an integer m and asks for the number of positive integers n, such that the factorial of n ends with exactly m zeroes. Are you among those great programmers who can solve this problem?

Input

The only line of input contains an integer m $(1 \le m \le 100\,000)$ — the required number of trailing zeroes in factorial.

Output

First print k — the number of values of n such that the factorial of n ends with m zeroes. Then print these k integers in increasing order.

Examples

input

```
1
```

output

```
5
5 6 7 8 9
```

input

```
5
```

output

```
0
```

Note

The factorial of n is equal to the product of all integers from 1 to n inclusive, that is $n! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot n$.

In the first sample, $5! = 120$, $6! = 720$, $7! = 5040$, $8! = 40320$ and $9! = 362880$.

C. Spy Syndrome 2

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

After observing the results of Spy Syndrome, Yash realised the errors of his ways. He now believes that a super spy such as Siddhant can't use a cipher as basic and ancient as Caesar cipher. After many weeks of observation of Siddhant's sentences, Yash determined a new cipher technique.

For a given sentence, the cipher is processed as:

1. Convert all letters of the sentence to lowercase.
2. Reverse each of the words of the sentence individually.
3. Remove all the spaces in the sentence.

For example, when this cipher is applied to the sentence

```
Kira is childish and he hates losing
```

the resulting string is

```
ariksihsidlihcdnaehsetahgnisol
```

Now Yash is given some ciphered string and a list of words. Help him to find out any original sentence composed using only words from the list. Note, that any of the given words could be used in the sentence multiple times.

Input

The first line of the input contains a single integer n ($1 \le n \le 10\,000$) — the length of the ciphered text. The second line consists of n lowercase English letters — the ciphered text t.

The third line contains a single integer m ($1 \le m \le 100\,000$) — the number of words which will be considered while deciphering the text. Each of the next m lines contains a non-empty word w_i ($|w_i| \le 1\,000$) consisting of uppercase and lowercase English letters only. It's guaranteed that the total length of all words doesn't exceed $1\,000\,000$.

Output

Print one line — the original sentence. It is guaranteed that at least one solution exists. If there are multiple solutions, you may output any of those.

Examples

input

```
30
ariksihsidlihcdnaehsetahgnisol
10
Kira
hates
is
he
losing
death
childish
L
and
Note
```

output

```
Kira is childish and he hates losing
```

input

```
12
iherehtolleh
5
HI
Ho
there
HeLLo
hello
```

output

```
HI there HeLLo
```

Note

In sample case 2 there may be multiple accepted outputs, "HI there HeLLo" and "HI there hello" you may output any of them.

D. Fibonacci-ish

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Yash has recently learnt about the Fibonacci sequence and is very excited about it. He calls a sequence Fibonacci-ish if

1. the sequence consists of at least two elements
2. f_0 and f_1 are arbitrary
3. $f_{n+2} = f_{n+1} + f_n$ for all $n \geq 0$.

You are given some sequence of integers $a_1, a_2, ..., a_n$. Your task is rearrange elements of this sequence in such a way that its longest possible prefix is Fibonacci-ish sequence.

Input

The first line of the input contains a single integer n $(2 \leq n \leq 1000)$ — the length of the sequence a_i.

The second line contains n integers $a_1, a_2, ..., a_n$ $(|a_i| \leq 10^9)$.

Output

Print the length of the longest possible Fibonacci-ish prefix of the given sequence after rearrangement.

Examples

input

```
3
1 2 -1
```

output

```
3
```

input

```
5
28 35 7 14 21
```

output

```
4
```

Note

In the first sample, if we rearrange elements of the sequence as $-1, 2, 1$, the whole sequence a_i would be Fibonacci-ish.

In the second sample, the optimal way to rearrange elements is , , , , 28.

E. Startup Funding

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

An e-commerce startup pitches to the investors to get funding. They have been functional for n weeks now and also have a website!

For each week they know the number of unique visitors during this week v_i and the revenue c_i. To evaluate the potential of the startup at some range of weeks from l to r inclusive investors use the minimum among the maximum number of visitors multiplied by 100 and the minimum revenue during this period, that is:

The truth is that investors have no idea how to efficiently evaluate the startup, so they are going to pick some k random distinct weeks l_i and give them to managers of the startup. For each l_i they should pick some $r_i \geq l_i$ and report maximum number of visitors and minimum revenue during this period.

Then, investors will calculate the potential of the startup for each of these ranges and take minimum value of $p(l_i, r_i)$ as the total evaluation grade of the startup. Assuming that managers of the startup always report the optimal values of r_i for some particular l_i, i.e., the value such that the resulting grade of the startup is maximized, what is the expected resulting grade of the startup?

Input

The first line of the input contains two integers n and k ($1 \leq k \leq n \leq 1\ 000\ 000$).

The second line contains n integers v_i ($1 \leq v_i \leq 10^7$) — the number of unique visitors during each week.

The third line contains n integers c_i ($1 \leq c_i \leq 10^7$) —the revenue for each week.

Output

Print a single real value — the expected grade of the startup. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Namely: let's assume that your answer is a, and the answer of the jury is b. The checker program will consider your answer correct, if .

Examples

input

```
3 2
3 2 1
300 200 300
```

output

```
133.3333333
```

Note

Consider the first sample.

If the investors ask for $l_i = 1$ onwards, startup will choose $r_i = 1$, such that max number of visitors is 3 and minimum revenue is 300. Thus, potential in this case is $min(3 \cdot 100, 300) = 300$.

If the investors ask for $l_i = 2$ onwards, startup will choose $r_i = 3$, such that max number of visitors is 2 and minimum revenue is 200. Thus, potential in this case is $min(2 \cdot 100, 200) = 200$.

If the investors ask for $l_i = 3$ onwards, startup will choose $r_i = 3$, such that max number of visitors is 1 and minimum revenue is 300. Thus, potential in this case is $min(1 \cdot 100, 300) = 100$.

We have to choose a set of size 2 equi-probably and take minimum of each. The possible sets here are : $\{200, 300\}$, $\{100, 300\}$, $\{100, 200\}$, effectively the set of possible values as perceived by investors equi-probably: $\{200, 100, 100\}$. Thus, the expected value is $(100 + 200 + 100) / 3 = 133.(3)$.

F. The Chocolate Spree

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Alice and Bob have a tree (undirected acyclic connected graph). There are a_i chocolates waiting to be picked up in the i-th vertex of the tree. First, they choose two different vertices as their starting positions (Alice chooses first) and take all the chocolates contained in them.

Then, they alternate their moves, selecting one vertex at a time and collecting all chocolates from this node. To make things more interesting, they decided that one can select a vertex only if he/she selected a vertex adjacent to that one at his/her previous turn and this vertex has not been already chosen by any of them during other move.

If at any moment one of them is not able to select the node that satisfy all the rules, he/she will skip his turns and let the other person pick chocolates as long as he/she can. This goes on until both of them cannot pick chocolates any further.

Due to their greed for chocolates, they want to collect as many chocolates as possible. However, as they are friends they only care about the total number of chocolates they obtain **together**. What is the maximum total number of chocolates they may pick?

Input

The first line of the input contains the single integer n $(2 \le \text{n} \le 100\,000)$ — the number of vertices in the tree.

The second line contains n integers a_i $(1 \le a_i \le 10^9)$, i-th of these numbers stands for the number of chocolates stored at the node i.

Then follow $n - 1$ lines that describe the tree. Each of them contains two integers u_i and v_i $(1 \le u_i, v_i \le n)$ — indices of vertices connected by the i-th edge.

Output

Print the number of chocolates Alice and Bob can collect together if they behave optimally.

Examples

input

```
9
1 2 3 4 5 6 7 8 9
1 2
1 3
1 4
1 5
1 6
1 7
1 8
1 9
```

output

```
25
```

input

```
2
20 10
1 2
```

output

```
30
```

Note

In the first sample, Alice may start at the vertex 9 and Bob at vertex 8. Alice will select vertex 1 and Bob has no options now. Alice selects the vertex 7 and they both stop.

In the second sample, both of them will pick either of the nodes alternately.

G. Yash And Trees

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Yash loves playing with trees and gets especially excited when they have something to do with prime numbers. On his 20th birthday he was granted with a rooted tree of n nodes to answer queries on. Hearing of prime numbers on trees, Yash gets too intoxicated with excitement and asks you to help out and answer queries on trees for him. Tree is rooted at node 1. Each node i has some value a_i associated with it. Also, integer m is given.

There are queries of two types:

1. for given node v and integer value x, increase all a_i in the subtree of node v by value x
2. for given node v, find the number of prime numbers p less than m, for which there exists a node u in the subtree of v and a non-negative integer value k, such that $a_u = p + m \cdot k$.

Input

The first of the input contains two integers n and m ($1 \le n \le 100\ 000, 1 \le m \le 1000$) — the number of nodes in the tree and value m from the problem statement, respectively.

The second line consists of n integers a_i ($0 \le a_i \le 10^9$) — initial values of the nodes.

Then follow n - 1 lines that describe the tree. Each of them contains two integers u_i and v_i ($1 \le u_i, v_i \le n$) — indices of nodes connected by the i-th edge.

Next line contains a single integer q ($1 \le q \le 100\ 000$) — the number of queries to proceed.

Each of the last q lines is either `1 v x` or `2 v` ($1 \le v \le n, 0 \le x \le 10^9$), giving the query of the first or the second type, respectively. It's guaranteed that there will be at least one query of the second type.

Output

For each of the queries of the second type print the number of suitable prime numbers.

Examples

input

```
8 20
3 7 9 8 4 11 7 3
1 2
1 3
3 4
4 5
4 6
4 7
5 8
4
2 1
1 1 1
2 5
2 4
```

output

```
3
1
1
```

input

```
5 10
8 7 5 1 0
1 2
2 3
1 5
2 4
3
1 1 0
1 1 2
2 2
```

output

```
2
```

H. Fibonacci-ish II

time limit per test: 5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Yash is finally tired of computing the length of the longest Fibonacci-ish sequence. He now plays around with more complex things such as Fibonacci-ish potentials.

Fibonacci-ish potential of an array a_i is computed as follows:

1. Remove all elements j if there exists $i < j$ such that $a_i = a_j$.
2. Sort the remaining elements in ascending order, i.e. $a_1 < a_2 < ... < a_n$.
3. Compute the potential as $P(a) = a_1 \cdot F_1 + a_2 \cdot F_2 + ... + a_n \cdot F_n$, where F_i is the i-th Fibonacci number (see notes for clarification).

You are given an array a_i of length n and q ranges from l_j to r_j. For each range j you have to compute the Fibonacci-ish potential of the array b_i, composed using all elements of a_i from l_j to r_j inclusive. Find these potentials modulo m.

Input

The first line of the input contains integers of n and m $(1 \le n, m \le 30\ 000)$ — the length of the initial array and the modulo, respectively.

The next line contains n integers a_i $(0 \le a_i \le 10^9)$ — elements of the array.

Then follow the number of ranges q $(1 \le q \le 30\ 000)$.

Last q lines contain pairs of indices l_i and r_i $(1 \le l_i \le r_i \le n)$ — ranges to compute Fibonacci-ish potentials.

Output

Print q lines, i-th of them must contain the Fibonacci-ish potential of the i-th range modulo m.

Example

input

```
5 10
2 1 2 1 2
2
2 4
4 5
```

output

```
3
3
```

Note

For the purpose of this problem define Fibonacci numbers as follows:

1. $F_1 = F_2 = 1$.
2. $F_n = F_{n-1} + F_{n-2}$ for each $n > 2$.

In the first query, the subarray [1,2,1] can be formed using the minimal set {1,2}. Thus, the potential of this subarray is 1*1+2*1=3.