A. Removing Columns

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

You are given an $n \times m$ rectangular table consisting of lower case English letters. In one operation you can completely remove one column from the table. The remaining parts are combined forming a new table. For example, after removing the second column from the table

```
abcd
edfg
hijk
```

we obtain the table:

```
acd
efg
hjk
```

A table is called good if its rows are ordered from top to bottom lexicographically, i.e. each row is lexicographically no larger than the following one. Determine the minimum number of operations of removing a column needed to make a given table good.

Input

The first line contains two integers — n and m ($1 \le n, m \le$ **100**).

Next n lines contain m small English letters each — the characters of the table.

Output

Print a single number — the minimum number of columns that you need to remove in order to make the table good.

Examples

input

```
1 10
codeforces
```

output

```
0
```

input

```
4 4
case
care
test
code
```

output

```
2
```

input

```
5 4
code
forc
esco
defo
rces
```

output

```
4
```

Note

In the first sample the table is already good.

In the second sample you may remove the first and third column.

In the third sample you have to remove all the columns (note that the table where all rows are empty is considered good by definition).

Let strings s and t have equal length. Then, s is lexicographically larger than t if they are not equal and the character following the largest common prefix of s and t (the prefix may be empty) in s is alphabetically larger than the corresponding character of t.

B. Antipalindrome

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

A string is a palindrome if it reads the same from the left to the right and from the right to the left. For example, the strings "kek", "abacaba", "r" and "papicipap" are palindromes, while the strings "abb" and "iq" are not.

A substring $s[l \ldots r]$ ($1 \leq l \leq r \leq |s|$) of a string $s = s_1 s_2 \ldots s_{|s|}$ is the string $s_l s_{l+1} \ldots s_r$.

Anna does not like palindromes, so she makes her friends call her Ann. She also changes all the words she reads in a similar way. Namely, each word s is changed into its longest substring that is not a palindrome. If all the substrings of s are palindromes, she skips the word at all.

Some time ago Ann read the word s. What is the word she changed it into?

Input

The first line contains a non-empty string s with length at most 50 characters, containing lowercase English letters only.

Output

If there is such a substring in s that is not a palindrome, print the maximum length of such a substring. Otherwise print 0.

Note that there can be multiple longest substrings that are not palindromes, but their length is unique.

Examples

input
```
mew
```
output
```
3
```

input
```
wuffuw
```
output
```
5
```

input
```
qqqqqqqq
```
output
```
0
```

Note

"mew" is not a palindrome, so the longest substring of it that is not a palindrome, is the string "mew" itself. Thus, the answer for the first example is 3.

The string "uffuw" is one of the longest non-palindrome substrings (of length 5) of the string "wuffuw", so the answer for the second example is 5.

All substrings of the string "qqqqqqqq" consist of equal characters so they are palindromes. This way, there are no non-palindrome substrings. Thus, the answer for the third example is 0.

C. Kyoya and Colored Balls

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

Kyoya Ootori has a bag with n colored balls that are colored with k different colors. The colors are labeled from 1 to k. Balls of the same color are indistinguishable. He draws balls from the bag one by one until the bag is empty. He noticed that he drew the last ball of color i before drawing the last ball of color $i + 1$ for all i from 1 to $k - 1$. Now he wonders how many different ways this can happen.

Input

The first line of input will have one integer k ($1 \le k \le 1000$) the number of colors.

Then, k lines will follow. The i-th line will contain c_i, the number of balls of the i-th color ($1 \le c_i \le 1000$).

The total number of balls doesn't exceed 1000.

Output

A single integer, the number of ways that Kyoya can draw the balls from the bag as described in the statement, modulo $1\,000\,000\,007$.

Examples

input

```
3
2
2
1
```

output

```
3
```

input

```
4
1
2
3
4
```

output

```
1680
```

Note

In the first sample, we have 2 balls of color 1, 2 balls of color 2, and 1 ball of color 3. The three ways for Kyoya are:

```
1 2 1 2 3
1 1 2 2 3
2 1 1 2 3
```

D. Cloud of Hashtags

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

Vasya is an administrator of a public page of organization "Mouse and keyboard" and his everyday duty is to publish news from the world of competitive programming. For each news he also creates a list of hashtags to make searching for a particular topic more comfortable. For the purpose of this problem we define hashtag as a string consisting of lowercase English letters and exactly one symbol '#' located at the beginning of the string. The *length* of the hashtag is defined as the number of symbols in it **without** the symbol '#'.

The head administrator of the page told Vasya that hashtags should go in lexicographical order (take a look at the notes section for the definition).

Vasya is lazy so he doesn't want to actually change the order of hashtags in already published news. Instead, he decided to delete some suffixes (consecutive characters at the end of the string) of some of the hashtags. He is allowed to delete any number of characters, even the whole string except for the symbol '#'. Vasya wants to pick such a way to delete suffixes that the total number of deleted symbols is **minimum** possible. If there are several optimal solutions, he is fine with any of them.

Input

The first line of the input contains a single integer n $(1 \leq n \leq 500\,000)$ — the number of hashtags being edited now.

Each of the next n lines contains exactly one hashtag of positive length.

It is guaranteed that the total length of all hashtags (i.e. the total length of the string except for characters '#') won't exceed $500\,000$.

Output

Print the resulting hashtags in any of the optimal solutions.

Examples

input

```
3
#book
#bigtown
#big
```

output

```
#b
#big
#big
```

input

```
3
#book
#cool
#cold
```

output

```
#book
#co
#cold
```

input

```
4
#car
#cart
#art
#at
```

output

```
#
#
#art
#at
```

input

```
3
#apple
#apple
#fruit
```

output

```
#apple
#apple
#fruit
```

Note

Word $a_1, a_2, ..., a_m$ of length m is *lexicographically not greater* than word $b_1, b_2, ..., b_k$ of length k, if one of two conditions hold:

- at first position i, such that $a_i \neq b_i$, the character a_i goes earlier in the alphabet than character b_i, i.e. a has smaller character than b in the first position where they differ;
- if there is no such position i and $m \le k$, i.e. the first word is a prefix of the second or two words are equal.

The sequence of words is said to be sorted in lexicographical order if each word (except the last one) is lexicographically not greater than the next word.

For the words consisting of lowercase English letters the lexicographical order coincides with the alphabet word order in the dictionary.

According to the above definition, if a hashtag consisting of one character '#' it is lexicographically not greater than any other valid hashtag. That's why in the third sample we can't keep first two hashtags unchanged and shorten the other two.

E. Maxim and Array

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

Recently Maxim has found an array of n integers, needed by no one. He immediately come up with idea of changing it: he invented positive integer x and decided to add or subtract it from arbitrary array elements. Formally, by applying single operation Maxim chooses integer i ($1 \le i \le n$) and replaces the i-th element of array a_i either with $a_i + x$ or with $a_i - x$. Please note that the operation may be applied more than once to the same position.

Maxim is a curious minimalis, thus he wants to know what is the minimum value that the product of all array elements (i.e. $\prod_{i=1}^{n} a_i$) can reach, if Maxim would apply no more than k operations to it. Please help him in that.

Input

The first line of the input contains three integers n, k and x ($1 \le n, k \le 200\,000$, $1 \le x \le 10^9$) — the number of elements in the array, the maximum number of operations and the number invented by Maxim, respectively.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($|a_i| \le 10^9$) — the elements of the array found by Maxim.

Output

Print n integers $b_1, b_2, \ldots, b_n$ in the only line — the array elements after applying no more than k operations to the array. In particular, $a_i \equiv b_i \mod x$ should stay true for every $1 \le i \le n$, but the product of all array elements should be **minimum possible**.

If there are multiple answers, print any of them.

Examples

input

```
5 3 1
5 4 3 5 2
```

output

```
5 4 3 5 -1
```

input

```
5 3 1
5 4 3 5 5
```

output

```
5 4 0 5 5
```

input

```
5 3 1
5 4 4 5 5
```

output

```
5 1 4 5 5
```

input

```
3 2 7
5 4 2
```

output

```
5 11 -5
```

F. Greg and Friends

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

One day Greg and his friends were walking in the forest. Overall there were n people walking, including Greg. Soon he found himself in front of a river. The guys immediately decided to get across the river. Luckily, there was a boat by the river bank, just where the guys were standing. We know that the boat can hold people with the total weight of at most k kilograms.

Greg immediately took a piece of paper and listed there the weights of all people in his group (including himself). It turned out that each person weights either 50 or 100 kilograms. Now Greg wants to know what minimum number of times the boat needs to cross the river to transport the whole group to the other bank. The boat needs at least one person to navigate it from one bank to the other. As the boat crosses the river, it can have any non-zero number of passengers as long as their total weight doesn't exceed k.

Also Greg is wondering, how many ways there are to transport everybody to the other side in the minimum number of boat rides. Two ways are considered distinct if during some ride they have distinct sets of people on the boat.

Help Greg with this problem.

Input

The first line contains two integers n, k $(1 \le n \le 50, 1 \le k \le 5000)$ — the number of people, including Greg, and the boat's weight limit. The next line contains n integers — the people's weights. A person's weight is either 50 kilos or 100 kilos.

You can consider Greg and his friends indexed in some way.

Output

In the first line print an integer — the minimum number of rides. If transporting everyone to the other bank is impossible, print an integer `-1`.

In the second line print the remainder after dividing the number of ways to transport the people in the minimum number of rides by number 1000000007 $(10^9 + 7)$. If transporting everyone to the other bank is impossible, print integer 0.

Examples

input

```
1 50
50
```

output

```
1
1
```

input

```
3 100
50 50 100
```

output

```
5
2
```

input
2 50 50 50

output
-1 0

Note

In the first test Greg walks alone and consequently, he needs only one ride across the river.

In the second test you should follow the plan:

1. transport two 50 kg. people;
2. transport one 50 kg. person back;
3. transport one 100 kg. person;
4. transport one 50 kg. person back;
5. transport two 50 kg. people.

That totals to 5 rides. Depending on which person to choose at step 2, we can get two distinct ways.

G. Codehorses T-shirts

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

Codehorses has just hosted the second Codehorses Cup. This year, the same as the previous one, organizers are giving T-shirts for the winners.

The valid sizes of T-shirts are either "`M`" or from 0 to 3 "`X`" followed by "`S`" or "`L`". For example, sizes "`M`", "`XXS`", "`L`", "`XXXL`" are valid and "`XM`", "`Z`", "`XXXXL`" are not.

There are n winners to the cup for both the previous year and the current year. Ksenia has a list with the T-shirt sizes printed for the last year cup and is yet to send the new list to the printing office.

Organizers want to distribute the prizes as soon as possible, so now Ksenia is required not to write the whole list from the scratch but just make some changes to the list of the previous year. In one second she can choose arbitrary position in any word and replace its character with some uppercase Latin letter. Ksenia can't remove or add letters in any of the words.

What is the minimal number of seconds Ksenia is required to spend to change the last year list to the current one?

The lists are unordered. That means, two lists are considered equal if and only if the number of occurrences of any string is the same in both lists.

Input

The first line contains one integer n ($1 \le n \le 100$) — the number of T-shirts.

The i-th of the next n lines contains a_i — the size of the i-th T-shirt of the list for the previous year.

The i-th of the next n lines contains b_i — the size of the i-th T-shirt of the list for the current year.

It is guaranteed that all the sizes in the input are valid. It is also guaranteed that Ksenia can produce list b from the list a.

Output

Print the minimal number of seconds Ksenia is required to spend to change the last year list to the current one. If the lists are already equal, print `0`.

Examples

input

```
3
XS
XS
M
XL
S
XS
```

output

```
2
```

input

```
2
XXXL
```

```
XXL
XXL
XXXS
```

output

```
1
```

input

```
2
M
XS
XS
M
```

output

```
0
```

Note

In the first example Ksenia can replace "M" with "S" and "S" in one of the occurrences of "XS" with "L".

In the second example Ksenia should replace "L" in "XXXL" with "S".

In the third example lists are equal.

H. Help Far Away Kingdom

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

In a far away kingdom lived the King, the Prince, the Shoemaker, the Dressmaker and many other citizens. They lived happily until great trouble came into the Kingdom. The ACMers settled there.

Most damage those strange creatures inflicted upon the kingdom was that they loved high precision numbers. As a result, the Kingdom healers had already had three appointments with the merchants who were asked to sell, say, exactly 0.273549107 beer barrels. To deal with the problem somehow, the King issued an order obliging rounding up all numbers to the closest integer to simplify calculations. Specifically, the order went like this:

- If a number's integer part does not end with digit 9 and its fractional part is strictly less than 0.5, then the rounded up number coincides with the number's integer part.
- If a number's integer part does not end with digit 9 and its fractional part is not less than 0.5, the rounded up number is obtained if we add 1 to the last digit of the number's integer part.
- If the number's integer part ends with digit 9, to round up the numbers one should go to Vasilisa the Wise. In the whole Kingdom she is the only one who can perform the tricky operation of carrying into the next position.

Merchants found the algorithm very sophisticated and they asked you (the ACMers) to help them. Can you write a program that would perform the rounding according to the King's order?

Input

The first line contains a single number to round up — the integer part (a non-empty set of decimal digits that do not start with 0 — with the exception of a case when the set consists of a single digit — in this case 0 can go first), then follows character «`.`» (a dot), and then follows the fractional part (any non-empty set of decimal digits). The number's length does not exceed 1000 characters, including the dot. There are no other characters in the input data.

Output

If the last number of the integer part is not equal to 9, print the rounded-up number without leading zeroes. Otherwise, print the message "`GOTO Vasilisa.`" (without the quotes).

Examples

input
```
0.0
```
output
```
0
```

input
```
1.49
```
output
```
1
```

input

```
1.50
```

output

```
2
```

input

```
2.71828182845904523536
```

output

```
3
```

input

```
3.14159265358979323846
```

output

```
3
```

input

```
12345678901234567890.1
```

output

```
12345678901234567890
```

input

```
123456789123456789.999
```

output

```
GOTO Vasilisa.
```

I. Good String

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

You have a string s of length n consisting of only characters > and <. You may do some operations with this string, for each operation you have to choose some character that still remains in the string. If you choose a character >, the character that comes right after it is deleted (if the character you chose was the last one, nothing happens). If you choose a character <, the character that comes right before it is deleted (if the character you chose was the first one, nothing happens).

For example, if we choose character > in string > **>** < >, the string will become to > > >. And if we choose character < in string > **<**, the string will become to <.

The string is good if there is a sequence of operations such that after performing it only one character will remain in the string. For example, the strings >, > > are good.

Before applying the operations, you may remove any number of characters from the given string (possibly none, possibly up to $n-1$, but not the whole string). You need to calculate the minimum number of characters to be deleted from string s so that it becomes good.

Input

The first line contains one integer t ($1 \le t \le 100$) – the number of test cases. Each test case is represented by two lines.

The first line of i-th test case contains one integer n ($1 \le n \le 100$) – the length of string s.

The second line of i-th test case contains string s, consisting of only characters > and <.

Output

For each test case print one line.

For i-th test case print the minimum number of characters to be deleted from string s so that it becomes good.

Example

input

```
3
2
<>
3
><<
1
>
```

output

```
1
0
0
```

Note

In the first test case we can delete any character in string <>.

In the second test case we don't need to delete any characters. The string > < < is good, because we can perform the following sequence of operations: > **<** < $\rightarrow$ < **<** $\rightarrow$ <.

J. Primes on Interval

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

You've decided to carry out a survey in the theory of prime numbers. Let us remind you that a prime number is a positive integer that has exactly two distinct positive integer divisors.

Consider positive integers a, $a + 1$, ..., b $(a \le b)$. You want to find the minimum integer l $(1 \le l \le b - a + 1)$ such that for any integer x $(a \le x \le b - l + 1)$ among l integers x, $x + 1$, ..., $x + l - 1$ there are at least k prime numbers.

Find and print the required minimum l. If no value l meets the described limitations, print -1.

Input

A single line contains three space-separated integers a, b, k $(1 \le a, b, k \le 10^6; a \le b)$.

Output

In a single line print a single integer — the required minimum l. If there's no solution, print -1.

Examples

input

```
2 4 2
```

output

```
3
```

input

```
6 13 1
```

output

```
4
```

input

```
1 4 3
```

output

```
-1
```

K. Eugeny and Play List

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

Eugeny loves listening to music. He has n songs in his play list. We know that song number i has the duration of t_i minutes. Eugeny listens to each song, perhaps more than once. He listens to song number i c_i times. Eugeny's play list is organized as follows: first song number 1 plays c_1 times, then song number 2 plays c_2 times, ..., in the end the song number n plays c_n times.

Eugeny took a piece of paper and wrote out m moments of time when he liked a song. Now for each such moment he wants to know the number of the song that played at that moment. The moment x means that Eugeny wants to know which song was playing during the x-th minute of his listening to the play list.

Help Eugeny and calculate the required numbers of songs.

Input

The first line contains two integers n, m $(1 \le n, m \le 10^5)$. The next n lines contain pairs of integers. The i-th line contains integers c_i, t_i $(1 \le c_i, t_i \le 10^9)$ — the description of the play list. It is guaranteed that the play list's total duration doesn't exceed 10^9 $(\sum_{i=1}^{n} c_i \cdot t_i \le 10^9)$.

The next line contains m positive integers $v_1, v_2, ..., v_m$, that describe the moments Eugeny has written out. It is guaranteed that there isn't such moment of time v_i, when the music doesn't play any longer. It is guaranteed that $v_i < v_{i+1}$ $(i < m)$.

The moment of time v_i means that Eugeny wants to know which song was playing during the v_i-th munite from the start of listening to the playlist.

Output

Print m integers — the i-th number must equal the number of the song that was playing during the v_i-th minute after Eugeny started listening to the play list.

Examples

input

```
1 2
2 8
1 16
```

output

```
1
1
```

input

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

output

```
1
1
```

```
2
2
3
4
4
4
4
```

L. Kefa and Park

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

Kefa decided to celebrate his first big salary by going to the restaurant.

He lives by an unusual park. The park is a rooted tree consisting of n vertices with the root at vertex 1. Vertex 1 also contains Kefa's house. Unfortunaely for our hero, the park also contains cats. Kefa has already found out what are the vertices with cats in them.

The leaf vertices of the park contain restaurants. Kefa wants to choose a restaurant where he will go, but unfortunately he is very afraid of cats, so there is no way he will go to the restaurant if the path from the restaurant to his house contains more than m **consecutive** vertices with cats.

Your task is to help Kefa count the number of restaurants where he can go.

Input

The first line contains two integers, n and m ($2 \le n \le 10^5$, $1 \le m \le n$) — the number of vertices of the tree and the maximum number of consecutive vertices with cats that is still ok for Kefa.

The second line contains n integers $a_1, a_2, ..., a_n$, where each a_i either equals to 0 (then vertex i has no cat), or equals to 1 (then vertex i has a cat).

Next $n - 1$ lines contains the edges of the tree in the format "x_i y_i" (without the quotes) ($1 \le x_i, y_i \le n$, $x_i \ne y_i$), where x_i and y_i are the vertices of the tree, connected by an edge.

It is guaranteed that the given set of edges specifies a tree.

Output

A single integer — the number of distinct leaves of a tree the path to which from Kefa's home contains at most m consecutive vertices with cats.

Examples

input

```
4 1
1 1 0 0
1 2
1 3
1 4
```

output

```
2
```

input

```
7 1
1 0 1 1 0 0 0
1 2
1 3
2 4
2 5
3 6
3 7
```

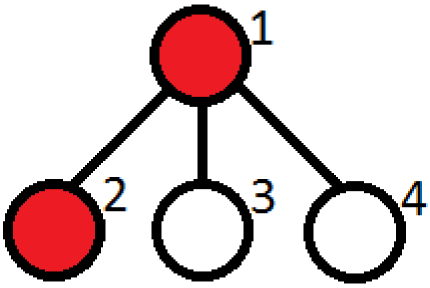
output

```
2
```

Note

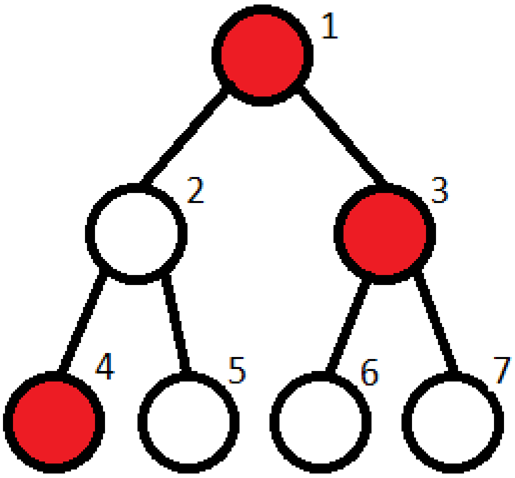
Let us remind you that a *tree* is a connected graph on n vertices and $n - 1$ edge. A *rooted* tree is a tree with a special vertex called *root*. In a rooted tree among any two vertices connected by an edge, one vertex is a parent (the one closer to the root), and the other one is a child. A vertex is called a *leaf*, if it has no children.

Note to the first sample test:



The vertices containing cats are marked red. The restaurants are at vertices 2, 3, 4. Kefa can't go only to the restaurant located at vertex 2.

Note to the second sample test:



The restaurants are located at vertices 4, 5, 6, 7. Kefa can't go to restaurants 6, 7.