TASK	ZAMKA	MULTIGRAM	PERICA	POPLAVA	OOP	PODNIZOVI
input	standard input (*stdin*)					
output	standard output (*stdout*)					
time limit	1 second	1 second	1 second	1 second	2 seconds	1 second
memory limit	64 MB	64 MB	64 MB	64 MB	512 MB	256 MB
score	**50**	**80**	**100**	**120**	**140**	**160**
	total 650					

The impossible has happened. Bear G. has fallen into his own trap. Lured by a delicious box of Domaćica, without even thinking, he rushed and fell into his trap. In order to get out of the trap, he must solve the following task with your help. You are given three integers L, D i X.

- determine the minimal integer N such that $L \leqslant N \leqslant D$ and the sum of its digits is X
- determine the maximal integer M such that $L \leqslant M \leqslant D$ and the sum of its digits is X

Bear will be able to escape from the trap if he correctly determines numbers N and M. The numbers N and M will always exist.

INPUT

The first line of input contains the integer L ($1 \leqslant L \leqslant 10\,000$), the number from the task.
The second line of input contains the integer D ($1 \leqslant D \leqslant 10\,000, L \leqslant D$), the number from the task.
The third line of input contains the integer X ($1 \leqslant X \leqslant 36$), the number from the task.

OUTPUT

The first line of output must contain the integer N from the task.
The second line of output must contain the integer M from the task.

SAMPLE TESTS

input	**input**	**input**
1 100 4	100 500 12	1 10000 1
output	**output**	**output**
4 40	129 480	1 10000

Pero is a passionate lover of riddles. The newest type of riddles he has come across requires the solver to check whether the given word is a **multigram**.
A multigram is a word that consists of concatenating two or more words that are all mutually **anagrams**. The first of these words is called the **root** of the multigram. For instance, the word bbabab is a multigram with the root bba because it consists of anagrams bba and bab.

Help Pero solve the riddle by determining whether his word is a multigram and determining its root in case it is. If there are multiple possible roots of the multigram, output the **shortest**.

Please note: Two words are mutually anagrams if one of them can be obtained from the other by changing the letter order.

INPUT

The first and only line of input contains a word of length at most 100 000 **lowercase English** characters.

OUTPUT

If the given word is not a multigram, output -1.
Otherwise, output the shortest root of the given word in one line.

SAMPLE TESTS

input	**input**	**input**
aaaa	ab	bbabab
output	**output**	**output**
a	-1	bba

Clarification of the first example: Notice that the word "aa" could also be the root, but "a" is shorter.
Clarification of the second example: The word is not a multigram because "a" and "b" are not mutually anagrams.

- "I'm stopping by Žnidaršić's house, you play the piano, Perica."
- "Ok, dad, I will!"

And so, Perica began playing the piano. His piano consists of N keys. Each key has a value written on it, a_i. When Perica plays the piano, he presses exactly K different keys at the same time. The piano is a bit strange because, after pressing K keys at the same time, it will play only the key with the largest value. Perica is going to play each combination of K keys on the piano and he wants to know the sum of values of the keys that will be played.
Help Perica determine the **remainder** of that number modulo 1 000 000 007.

INPUT

The first line of input contains two integers N and K ($1 \leqslant N \leqslant 100\,000$, $1 \leqslant K \leqslant 50$).
The following line of input contains N integers a_i ($0 \leqslant a_{ij} \leqslant 10^9$).

OUTPUT

The first and only line of output must contain the required number from the task.

SCORING

In test cases worth 40% of total points, it will additionally hold $1 \leqslant N \leqslant 1000$.

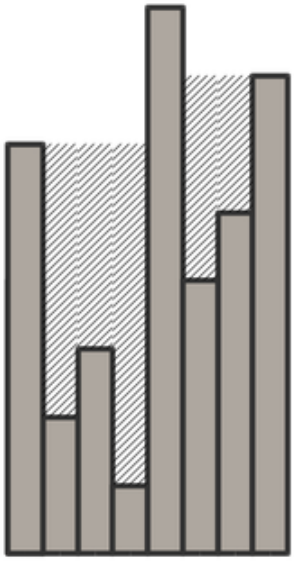
SAMPLE TESTS

input	**input**	**input**
5 3 2 4 2 3 4	5 1 1 0 1 1 1	5 2 3 3 4 0 0
output	**output**	**output**
39	4	31

Pojašnjenje prvog primjera: All selections of K keys are: [2, 4, 2], [2, 4, 3], [2, 4, 4], [2, 2, 3], [2, 2, 4], [2, 3, 4], [4, 2, 3], [4, 2, 4], [4, 3, 4], [2, 3, 4].

Mirko dreamt of a histogram last night that consists of N columns. Each column is one meter wide and the heights of the columns in meters are $h_1, h_2, ..., h_N$.

The capacity of a histogram is the maximal amount of water that a histogram can hold so that the configuration of the water is "stable", or, in other words, that it doesn't move under the influence of gravity. The image on the right depicts an example of a stable configuration.



Formally, let us denote the heights of water above the columns with $v_1, v_2, ..., v_N$.

The configuration of the water is stable if the following holds:

- $h_i + v_i \leqslant h_{i-1} + v_{i-1}$, for each $i \geqslant 2$ such that $v_i > 0$
- $h_i + v_i \leqslant h_{i+1} + v_{i+1}$, for each $i \leqslant N - 1$ such that $v_i > 0$
- $v_1 = 0$ and $v_N = 0$

When Mirko woke up, he wanted to know whether he could somehow choose the heights of columns that are a **permutation** of the set $\{1, 2, ..., N\}$ such that the capacity of such histogram is equal to its lucky number X? Help Mirko and find one histogram that meets his requirements.

INPUT

The first line of input contains integers N and X ($1 \leqslant N \leqslant 1\,000\,000, 1 \leqslant X \leqslant 10^{15}$).

OUTPUT

If a histogram of capacity exactly X does not exist, output -1. Otherwise, output numbers $h_1, h_2, ..., h_N$ that meet the given requirements in the first line separated by space. If there are multiple such solutions, output any.

SAMPLE TESTS

input	**input**	**input**
3 1	4 1	8 17
output	**output**	**output**
3 1 2	4 3 1 2	6 2 3 1 8 4 5 7

Clarification of the first example: In this configuration, it holds $v_1 = 0$, $v_2 = 1$, $v_3 = 0$.
Clarification of the second example: In this configuration, it holds $v_1 = 0$, $v_2 = 0$, $v_3 = 1$, $v_4 = 0$.
Clarification of the third example: The sample corresponds to the image from the task.

Little Matej is solving an OOP (Object-oriented programming) laboratory exercise and he's having trouble with solving one subtask.
He is given a set that contains N words. He is also given Q queries where each query is one pattern. A pattern consists of a single character "*" and lowercase letters of the English alphabet. For example, "*", "kul*to", "ana*".
A pattern is said to cover a word if such an array of letters (**which can be empty**) exists that, when replacing the character '*', the pattern and the word become completely identical. It is necessary to output how many words each pattern covers.

INPUT

The first line of input contains two integers N and Q ($1 \leqslant N, Q \leqslant 100\,000$).
Each of the following N lines contains a word that consists of lowercase letters of the English alphabet.
Each of the following Q lines contains a pattern for which you need to output how many words from the first set it covers.
The total number of characters will be less than $3\,000\,000$.

SCORING

In test cases worth 40% of total points, it will additionally hold $1 \leqslant N, Q \leqslant 1000$.

OUTPUT

Output Q lines, the k^{th} line containing the number of words that the k^{th} pattern covers.

SAMPLE TESTS

input

```
3 3
aaa
abc
aba
a*a
aaa*
*aaa
```

output

```
2
1
1
```

input

```
5 3
eedecc
ebdecb
eaba
ebcddc
eb
e*
*dca
e*c
```

output

```
5
0
2
```

You are given an array of integers of length N. Let $s_1, s_2, ..., s_q$ be the **lexicographically sorted** array of all its non-empty subsequences. A subsequence of the array is an array obtained by removing zero or more elements from the initial array. Notice that some subsequences can be equal and that it holds $q = 2^N - 1$.

An array A is lexicographically smaller than array B if $A_i < B_i$ where i is the first position at which the arrays differ, or if A is a strict prefix of array B.

Let us define the hash of an array that consists of values $v_1, v_2, ..., v_p$ as:

$$h(s) = (v_1 \cdot B^{p-1} + v_2 \cdot B^{p-2} + ... + v_{p-1} \cdot B + v_p) \mod M$$

where B, M are given integers.

Calculate $h(s_1), h(s_2), ..., h(s_K)$ for a given K.

INPUT

The first line contains integers N, K, B, M ($1 \leqslant N \leqslant 100\,000$, $1 \leqslant K \leqslant 100\,000$, $1 \leqslant B, M \leqslant 1\,000\,000$).
The second line contains integers $a_1, a_2, a_3, ..., a_N$ ($1 \leqslant a_i \leqslant 100\,000$).
In all test cases, it will hold $K \leqslant 2^N - 1$.

OUTPUT

Output K lines, the j^{th} line containing $h(s_j)$.

SCORING

In test cases worth 60% of total points, it will additionally hold $1 \leqslant a_1, a_2, ..., a_N \leqslant 30$.

SAMPLE TESTS

input

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

output

```
1
3
2
```

input

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

output

```
1
1
0
2
```

input

```
5 6 23 1000
1 2 4 2 3
```

output

```
1
25
25
577
274
578
```

Clarification of the first example: It holds: $s_1 = [1], s_2 = [1, 2], s_3 = [2]$. $h(s_1) = 1 \mod 5 = 1, h(s_2) = (1 + 2) \mod 5 = 3, h(s_3) = 2 \mod 5 = 2$.
Clarification of the second example: It holds: $s_1 = [1], s_2 = [1], s_3 = [1, 1], s_4 = [1, 3]$. $h(s_1) = 1 \mod 3 = 1, h(s_2) = 1 \mod 3 = 1, h(s_3) = (1 \cdot 2 + 1) \mod 3 = 0, h(s_4) = (1 \cdot 2 + 3) \mod 3 = 2$.