CodeCraft-19 and Codeforces Round #537 (Div. 2)

A. Superhero Transformation

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

We all know that a superhero can transform to certain other superheroes. But not all Superheroes can transform to any other superhero. A superhero with name s can transform to another superhero with name t if s can be made equal to t by changing any vowel in s to any other vowel and any consonant in s to any other consonant. Multiple changes can be made.

In this problem, we consider the letters 'a', 'e', 'i', 'o' and 'u' to be vowels and all the other letters to be consonants.

Given the names of two superheroes, determine if the superhero with name s can be transformed to the Superhero with name t.

Input

The first line contains the string s having length between 1 and 1000, inclusive.

The second line contains the string t having length between 1 and 1000, inclusive.

Both strings s and t are guaranteed to be different and consist of lowercase English letters only.

Output

Output "Yes" (without quotes) if the superhero with name s can be transformed to the superhero with name t and "No" (without quotes) otherwise.

You can print each letter in any case (upper or lower).

Examples

input
```
a
u
```
output
```
Yes
```

input
```
abc
ukm
```
output
```
Yes
```

input
```
akm
ua
```
output
```
No
```

Note

In the first sample, since both 'a' and 'u' are vowels, it is possible to convert string s to t.

In the third sample, 'k' is a consonant, whereas 'a' is a vowel, so it is not possible to convert string s to t.

B. Average Superhero Gang Power

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

Every superhero has been given a power value by the Felicity Committee. The avengers crew wants to maximize the average power of the superheroes in their team by performing certain operations.

Initially, there are n superheroes in avengers team having powers $a_1, a_2, \ldots, a_n$, respectively. In one operation, they can remove one superhero from their team (if there are at least two) or they can increase the power of a superhero by 1. They can do at most m operations. Also, on a particular superhero at most k operations can be done.

Can you help the avengers team to maximize the average power of their crew?

Input

The first line contains three integers n, k and m ($1 \le n \le 10^5$, $1 \le k \le 10^5$, $1 \le m \le 10^7$) — the number of superheroes, the maximum number of times you can increase power of a particular superhero, and the total maximum number of operations.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^6$) — the initial powers of the superheroes in the cast of avengers.

Output

Output a single number — the maximum final average power.

Your answer is considered correct if its absolute or relative error does not exceed 10^{-6}.

Formally, let your answer be a, and the jury's answer be b. Your answer is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-6}$.

Examples

input

```
2 4 6
4 7
```

output

```
11.00000000000000000000
```

input

```
4 2 6
1 3 2 3
```

output

```
5.00000000000000000000
```

Note

In the first example, the maximum average is obtained by deleting the first element and increasing the second element four times.

In the second sample, one of the ways to achieve maximum average is to delete the first and the third element and increase the second and the fourth elements by 2 each.

C. Creative Snap

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

Thanos wants to destroy the avengers base, but he needs to destroy the avengers along with their base.

Let we represent their base with an array, where each position can be occupied by many avengers, but one avenger can occupy only one position. Length of their base is a perfect power of 2. Thanos wants to destroy the base using minimum power. He starts with the whole base and in one step he can do either of following:

- if the current length is at least 2, divide the base into 2 equal halves and destroy them separately, or
- burn the current base. If it contains no avenger in it, it takes A amount of power, otherwise it takes his $B \cdot n_a \cdot l$ amount of power, where n_a is the number of avengers and l is the length of the current base.

Output the minimum power needed by Thanos to destroy the avengers' base.

Input

The first line contains four integers n, k, A and B ($1 \le n \le 30$, $1 \le k \le 10^5$, $1 \le A, B \le 10^4$), where 2^n is the length of the base, k is the number of avengers and A and B are the constants explained in the question.

The second line contains k integers $a_1, a_2, a_3, \ldots, a_k$ ($1 \le a_i \le 2^n$), where a_i represents the position of avenger in the base.

Output

Output one integer — the minimum power needed to destroy the avengers base.

Examples

input

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

output

```
6
```

input

```
3 2 1 2
1 7
```

output

```
8
```

Note

Consider the first example.

One option for Thanos is to burn the whole base $1-4$ with power $2 \cdot 2 \cdot 4 = 16$.

Otherwise he can divide the base into two parts $1-2$ and $3-4$.

For base $1-2$, he can either burn it with power $2 \cdot 1 \cdot 2 = 4$ or divide it into 2 parts $1-1$ and $2-2$.

For base $1-1$, he can burn it with power $2 \cdot 1 \cdot 1 = 2$. For $2-2$, he can destroy it with power 1, as there are no avengers. So, the total power for destroying $1-2$ is $2+1=3$, which is less than 4.

Similarly, he needs 3 power to destroy $3-4$. The total minimum power needed is 6.

D. Destroy the Colony

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

There is a colony of villains with several holes aligned in a row, where each hole contains exactly one villain.

Each colony arrangement can be expressed as a string of **even** length, where the i-th character of the string represents the type of villain in the i-th hole.

Iron Man can destroy a colony only if the colony arrangement is such that all villains of a certain type either live in the first half of the colony or in the second half of the colony.

His assistant Jarvis has a special power. It can swap villains of any two holes, i.e. swap any two characters in the string; he can do this operation any number of times.

Now Iron Man asks Jarvis q questions. In each question, he gives Jarvis two numbers x and y. Jarvis has to tell Iron Man the number of distinct colony arrangements he can create from the original one using his powers such that all villains having the same type as those originally living in x-th hole or y-th hole live in the same half and the Iron Man can destroy that colony arrangement.

Two colony arrangements are considered to be different if there exists a hole such that different types of villains are present in that hole in the arrangements.

Input

The first line contains a string s ($2 \le |s| \le 10^5$), representing the initial colony arrangement. String s can have both lowercase and uppercase English letters and its length is **even**.

The second line contains a single integer q ($1 \le q \le 10^5$) — the number of questions.

The i-th of the next q lines contains two integers x_i and y_i ($1 \le x_i, y_i \le |s|$, $x_i \ne y_i$) — the two numbers given to the Jarvis for the i-th question.

Output

For each question output the number of arrangements possible modulo 10^9+7.

Examples

input

```
abba
2
1 4
1 2
```

output

```
2
0
```

input

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

output

```
2
0
```

input

```
abcd
```

```
1
1 3
```

output

```
8
```

Note

Consider the first example. For the first question, the possible arrangements are "aabb" and "bbaa", and for the second question, index 1 contains 'a' and index 2 contains 'b' and there is no valid arrangement in which all 'a' and 'b' are in the same half.

E. Tree

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a tree with n nodes and q queries.

Every query starts with three integers k, m and r, followed by k nodes of the tree $a_1, a_2, \ldots, a_k$. To answer a query, assume that the tree is rooted at r. We want to divide the k given nodes into **at most** m groups such that the following conditions are met:

- Each node should be in exactly one group and each group should have at least one node.
- In any group, there should be no two distinct nodes such that one node is an ancestor (direct or indirect) of the other.

You need to output the number of ways modulo $10^9 + 7$ for every query.

Input

The first line contains two integers n and q ($1 \le n, q \le 10^5$) — the number of vertices in the tree and the number of queries, respectively.

Each of the next $n - 1$ lines contains two integers u and v ($1 \le u, v \le n, u \ne v$), denoting an edge connecting vertex u and vertex v. It is guaranteed that the given graph is a tree.

Each of the next q lines starts with three integers k, m and r ($1 \le k, r \le n$, $1 \le m \le min(300, k)$) — the number of nodes, the maximum number of groups and the root of the tree for the current query, respectively. They are followed by k distinct integers $a_1, a_2, \ldots, a_k$ ($1 \le a_i \le n$), denoting the nodes of the current query.

It is guaranteed that the sum of k over all queries does not exceed 10^5.

Output

Print q lines, where the i-th line contains the answer to the i-th query.

Examples

input

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

output

```
2
0
```

input

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

output

```
1
1
```

input

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

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

output

```
2
1
```

Note

Consider the first example.

In the first query, we have to divide the three given nodes (7, 4 and 3), into the maximum of three groups assuming that the tree is rooted at 2. When the tree is rooted at 2, 4 is an ancestor of both 3 and 7. So we can't put all the nodes into one group. There is only 1 way to divide the given nodes into two groups, which are $[4]$ and $[3, 7]$. Also, there is only one way to divide the given nodes into three groups, which are $[7]$, $[4]$ and $[3]$. So, there are total 2 ways to divide the given nodes into a maximum of three groups.

In the second query, when the tree is rooted at 4, 6 is an ancestor of 2 and 2 is an ancestor of 1. So, we can't put all the given nodes into one group.