Codecraft-17 and Codeforces Round #391 (Div. 1 + Div. 2, combined)

A. Gotta Catch Em' All!

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

Bash wants to become a Pokemon master one day. Although he liked a lot of Pokemon, he has always been fascinated by Bulbasaur the most. Soon, things started getting serious and his fascination turned into an obsession. Since he is too young to go out and catch Bulbasaur, he came up with his own way of catching a Bulbasaur.

Each day, he takes the front page of the newspaper. He cuts out the letters one at a time, from anywhere on the front page of the newspaper to form the word "`Bulbasaur`" (without quotes) and sticks it on his wall. Bash is very particular about case — the first letter of "`Bulbasaur`" must be upper case and the rest must be lower case. By doing this he thinks he has caught one Bulbasaur. He then repeats this step on the left over part of the newspaper. He keeps doing this until it is not possible to form the word "`Bulbasaur`" from the newspaper.

Given the text on the front page of the newspaper, can you tell how many Bulbasaurs he will catch today?

Note: **uppercase and lowercase letters are considered different.**

Input

Input contains a single line containing a string s ($1 \le |s| \le 10^5$) — the text on the front page of the newspaper without spaces and punctuation marks. $|s|$ is the length of the string s.

The string s contains lowercase and uppercase English letters, i.e. .

Output

Output a single integer, the answer to the problem.

Examples

input
```
Bulbbasaur
```
output
```
1
```

input
```
F
```
output
```
0
```

input
```
aBddulbasaurrgndgbualdBdsagaurrgndbb
```
output
```
2
```

Note

In the first case, you could pick: **`Bulb`**`b`**`asaur`**.

In the second case, there is no way to pick even a single Bulbasaur.

In the third case, you can rearrange the string to **`BulbasaurBulbasaur`**`addrgndgddgargndbb` to get two words "`Bulbasaur`".

B. Bash's Big Day

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

Bash has set out on a journey to become the greatest Pokemon master. To get his first Pokemon, he went to Professor Zulu's Lab. Since Bash is Professor Zulu's favourite student, Zulu allows him to take as many Pokemon from his lab as he pleases.

But Zulu warns him that a group of $k > 1$ Pokemon with strengths $\{s_1, s_2, s_3, ..., s_k\}$ tend to fight among each other if $gcd(s_1, s_2, s_3, ..., s_k) = 1$ (see notes for gcd definition).

Bash, being smart, does not want his Pokemon to fight among each other. However, he also wants to maximize the number of Pokemon he takes from the lab. Can you help Bash find out the maximum number of Pokemon he can take?

Note: A Pokemon cannot fight with itself.

Input

The input consists of two lines.

The first line contains an integer n $(1 \le n \le 10^5)$, the number of Pokemon in the lab.

The next line contains n space separated integers, where the i-th of them denotes s_i $(1 \le s_i \le 10^5)$, the strength of the i-th Pokemon.

Output

Print single integer — the maximum number of Pokemons Bash can take.

Examples

input

```
3
2 3 4
```

output

```
2
```

input

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

output

```
3
```

Note

gcd (greatest common divisor) of positive integers set $\{a_1, a_2, ..., a_n\}$ is the maximum positive integer that divides all the integers $\{a_1, a_2, ..., a_n\}$.

In the first sample, we can take Pokemons with strengths $\{2, 4\}$ since $gcd(2, 4) = 2$.

In the second sample, we can take Pokemons with strengths $\{2, 4, 6\}$, and there is no larger group with $gcd \neq 1$.

C. Felicity is Coming!

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

It's that time of the year, Felicity is around the corner and you can see people celebrating all around the Himalayan region. The Himalayan region has n gyms. The i-th gym has g_i Pokemon in it. There are m distinct Pokemon types in the Himalayan region numbered from 1 to m. There is a special evolution camp set up in the fest which claims to evolve any Pokemon. The type of a Pokemon could change after evolving, subject to the constraint that if two Pokemon have the same type before evolving, they will have the same type after evolving. Also, if two Pokemon have different types before evolving, they will have different types after evolving. It is also possible that a Pokemon has the same type before and after evolving.

Formally, an *evolution plan* is a permutation f of $\{1, 2, ..., m\}$, such that $f(x) = y$ means that a Pokemon of type x evolves into a Pokemon of type y.

The gym leaders are intrigued by the special evolution camp and all of them plan to evolve their Pokemons. The protocol of the mountain states that in each gym, for every type of Pokemon, the number of Pokemon of that type before evolving any Pokemon should be equal the number of Pokemon of that type after evolving all the Pokemons according to the evolution plan. They now want to find out how many distinct *evolution plans* exist which satisfy the protocol.

Two evolution plans f_1 and f_2 are distinct, if they have at least one Pokemon type evolving into a different Pokemon type in the two plans, i. e. there exists an i such that $f_1(i) \neq f_2(i)$.

Your task is to find how many distinct *evolution plans* are possible such that if all Pokemon in all the gyms are evolved, the number of Pokemon of each type in each of the gyms remains the same. As the answer can be large, output it modulo $10^9 + 7$.

Input

The first line contains two integers n and m ($1 \leq n \leq 10^5$, $1 \leq m \leq 10^6$) — the number of gyms and the number of Pokemon types.

The next n lines contain the description of Pokemons in the gyms. The i-th of these lines begins with the integer g_i ($1 \leq g_i \leq 10^5$) — the number of Pokemon in the i-th gym. After that g_i integers follow, denoting types of the Pokemons in the i-th gym. Each of these integers is between 1 and m.

The total number of Pokemons (the sum of all g_i) does not exceed $5 \cdot 10^5$.

Output

Output the number of valid evolution plans modulo $10^9 + 7$.

Examples

input

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

output

```
1
```

input

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

output

```
6
```

input

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

output

```
2
```

input

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

output

```
1
```

input

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

output

```
24
```

Note

In the first case, the only possible evolution plan is:

In the second case, any permutation of $(1,\ 2,\ 3)$ is valid.

In the third case, there are two possible plans:

In the fourth case, the only possible evolution plan is:

D. Felicity's Big Secret Revealed

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

The gym leaders were fascinated by the evolutions which took place at Felicity camp. So, they were curious to know about the secret behind evolving Pokemon.

The organizers of the camp gave the gym leaders a PokeBlock, a sequence of n ingredients. Each ingredient can be of type 0 or 1. Now the organizers told the gym leaders that to evolve a Pokemon of type k ($k \geq 2$), they need to make a valid set of k cuts on the PokeBlock to get smaller blocks.

Suppose the given PokeBlock sequence is $b_0 b_1 b_2 \dots b_{n-1}$. You have a choice of making cuts at $n+1$ places, i.e., Before b_0, between b_0 and b_1, between b_1 and b_2, ..., between b_{n-2} and b_{n-1}, and after b_{n-1}.

The $n+1$ choices of making cuts are as follows (where a | denotes a possible cut):

$$| b_0 | b_1 | b_2 | \dots | b_{n-2} | b_{n-1} |$$

Consider a sequence of k cuts. Now each pair of consecutive cuts will contain a binary string between them, formed from the ingredient types. The ingredients before the first cut and after the last cut are wasted, which is to say they are not considered. So there will be exactly $k-1$ such binary substrings. Every substring can be read as a binary number. Let m be the maximum number out of the obtained numbers. If all the obtained numbers are positive and the set of the obtained numbers contains all integers from 1 to m, then this set of cuts is said to be a valid set of cuts.

For example, suppose the given PokeBlock sequence is 101101001110 and we made 5 cuts in the following way:

$$10 | 11 | 010 | 01 | 1 | 10$$

So the 4 binary substrings obtained are: 11, 010, 01 and 1, which correspond to the numbers 3, 2, 1 and 1 respectively. Here $m = 3$, as it is the maximum value among the obtained numbers. And all the obtained numbers are positive and we have obtained all integers from 1 to m. Hence this set of cuts is a valid set of 5 cuts.

A Pokemon of type k will evolve only if the PokeBlock is cut using a valid set of k cuts. There can be many valid sets of the same size. Two valid sets of k cuts are considered different if there is a cut in one set which is not there in the other set.

Let $f(k)$ denote the number of valid sets of k cuts. Find the value of . Since the value of s can be very large, output s modulo 10^9+7.

Input

The input consists of two lines. The first line consists an integer n ($1 \leq n \leq 75$) — the length of the PokeBlock. The next line contains the PokeBlock, a binary string of length n.

Output

Output a single integer, containing the answer to the problem, i.e., the value of s modulo 10^9+7.

Examples

input

```
4
1011
```

output

```
10
```

input

```
2
10
```

output

```
1
```

Note

In the first sample, the sets of valid cuts are:

Size 2: |1|011, 1|01|1, 10|1|1, 101|1|.

Size 3: |1|01|1, |10|1|1, 10|1|1|, 1|01|1|.

Size 4: |10|1|1|, |1|01|1|.

Hence, $f(2)=4$, $f(3)=4$ and $f(4)=2$. So, the value of $s=10$.

In the second sample, the set of valid cuts is:

Size 2: |1|0.

Hence, $f(2)=1$ and $f(3)=0$. So, the value of $s=1$.

E. Bash Plays with Functions

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

Bash got tired on his journey to become the greatest Pokemon master. So he decides to take a break and play with functions.

Bash defines a function $f_0(n)$, which denotes the number of ways of factoring n into two factors p and q such that $gcd(p, q) = 1$. In other words, $f_0(n)$ is the number of ordered pairs of positive integers (p, q) such that $p \cdot q = n$ and $gcd(p, q) = 1$.

But Bash felt that it was too easy to calculate this function. So he defined a series of functions, where f_{r+1} is defined as:

Where (u, v) is any ordered pair of positive integers, they need not to be co-prime.

Now Bash wants to know the value of $f_r(n)$ for different r and n. Since the value could be huge, he would like to know the value modulo $10^9 + 7$. Help him!

Input

The first line contains an integer q $(1 \leq q \leq 10^6)$ — the number of values Bash wants to know.

Each of the next q lines contain two integers r and n $(0 \leq r \leq 10^6,\ 1 \leq n \leq 10^6)$, which denote Bash wants to know the value $f_r(n)$.

Output

Print q integers. For each pair of r and n given, print $f_r(n)$ modulo $10^9 + 7$ on a separate line.

Example

input

```
5
0 30
1 25
3 65
2 5
4 48
```

output

```
8
5
25
4
630
```

F. Team Rocket Rises Again

time limit per test: 2.5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

It's the turn of the year, so Bash wants to send presents to his friends. There are n cities in the Himalayan region and they are connected by m bidirectional roads. Bash is living in city s. Bash has exactly one friend in each of the other cities. Since Bash wants to surprise his friends, he decides to send a Pikachu to each of them. **Since there may be some cities which are not reachable from Bash's city, he only sends a Pikachu to those friends who live in a city reachable from his own city**. He also wants to send it to them as soon as possible.

He finds out the minimum time for each of his Pikachus to reach its destination city. Since he is a perfectionist, he informs all his friends with the time their gift will reach them. A Pikachu travels at a speed of 1 meters per second. His friends were excited to hear this and would be unhappy if their presents got delayed. Unfortunately Team Rocket is on the loose and they came to know of Bash's plan. They want to maximize the number of friends who are unhappy with Bash.

They do this by destroying exactly one of the other $n - 1$ cities. This implies that **the friend residing in that city dies, so he is unhappy as well**.

Note that **if a city is destroyed, all the roads directly connected to the city are also destroyed and the Pikachu may be forced to take a longer alternate route**.

Please also note that only friends that are waiting for a gift count as unhappy, even if they die.

Since Bash is already a legend, can you help Team Rocket this time and find out the maximum number of Bash's friends who can be made unhappy by destroying exactly one city.

Input

The first line contains three space separated integers n, m and s ($2 \le n \le 2 \cdot 10^5$, , $1 \le s \le n$) — the number of cities and the number of roads in the Himalayan region and the city Bash lives in.

Each of the next m lines contain three space-separated integers u, v and w ($1 \le u, v \le n$, $u \ne v$, $1 \le w \le 10^9$) denoting that there exists a road between city u and city v of length w meters.

It is guaranteed that no road connects a city to itself and there are no two roads that connect the same pair of cities.

Output

Print a single integer, the answer to the problem.

Examples

input

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

output

```
2
```

input

```
7 11 2
1 2 5
1 3 5
2 4 2
2 5 2
3 6 3
3 7 3
4 6 2
3 4 2
6 7 3
4 5 7
4 7 7
```

output

```
4
```

Note

In the first sample, on destroying the city 2, the length of shortest distance between pairs of cities $(3, 2)$ and $(3, 4)$ will change. Hence the answer is 2.

G. Can Bash Save the Day?

time limit per test: 5 seconds
memory limit per test: 768 megabytes
input: standard input
output: standard output

Whoa! You did a great job helping Team Rocket who managed to capture all the Pokemons sent by Bash. Meowth, part of Team Rocket, having already mastered the human language, now wants to become a master in programming as well. He agrees to free the Pokemons if Bash can answer his questions.

Initially, Meowth gives Bash a weighted tree containing n nodes and a sequence $a_1, a_2..., a_n$ which is a permutation of $1, 2, ..., n$. Now, Mewoth makes q queries of one of the following forms:

- `1 l r v`: meaning Bash should report , where $dist(a, b)$ is the length of the shortest path from node a to node b in the given tree.
- `2 x`: meaning Bash should swap a_x and a_{x+1} in the given sequence. This new sequence is used for later queries.

Help Bash to answer the questions!

Input

The first line contains two integers n and q ($1 \le n \le 2 \cdot 10^5$, $1 \le q \le 2 \cdot 10^5$) — the number of nodes in the tree and the number of queries, respectively.

The next line contains n space-separated integers — the sequence $a_1, a_2, ..., a_n$ which is a permutation of $1, 2, ..., n$.

Each of the next $n - 1$ lines contain three space-separated integers u, v, and w denoting that there exists an undirected edge between node u and node v of weight w, ($1 \le u, v \le n$, $u \ne v$, $1 \le w \le 10^6$). It is guaranteed that the given graph is a tree.

Each query consists of two lines. First line contains single integer t, indicating the type of the query. Next line contains the description of the query:

- **t = 1**: Second line contains three integers a, b and c ($1 \le a, b, c < 2^{30}$) using which l, r and v can be generated using the formula given below:
  - ,
  - ,
  - .
- **t = 2**: Second line contains single integer a ($1 \le a < 2^{30}$) using which x can be generated using the formula given below:
  - .

The ans_i is the answer for the i-th query, assume that $ans_0 = 0$. If the i-th query is of type 2 then $ans_i = ans_{i-1}$. It is guaranteed that:

- **for each query of type 1**: $1 \le l \le r \le n$, $1 \le v \le n$,
- **for each query of type 2**: $1 \le x \le n - 1$.

The operation means bitwise exclusive OR.

Output

For each query of type 1, output a single integer in a separate line, denoting the answer to the query.

Example

input

```
5 5
4 5 1 3 2
4 2 4
1 3 9
4 1 4
4 5 2
1
1 5 4
1
22 20 20
2
38
2
39
1
36 38 38
```

output

```
23
37
28
```

Note

In the sample, the actual queries are the following:

- `1 1 5 4`

- 1 1 3 3
- 2 3
- 2 2
- 1 1 3 3