

## Problem A. Add More Zero

Input file:           standard input  
Output file:         standard output  
Time limit:          1 second  
Memory limit:       128 megabytes

There is a youngster known for amateur propositions concerning several mathematical hard problems. Today he is going to prepare a thought-provoking problem on a specific type of supercomputer which has the ability to support calculating operations for integers between 0 and  $(2^m - 1)$  (inclusive). As a young man born with ten fingers, he loves the powers of 10 so much, which results in his eccentricity that he always ranges integers he would like to use from 1 to  $10^k$  (inclusive). For ease of processing, all integers he would probably use in this interesting problem ought to be as computable as this supercomputer could. Given the positive integer  $m$ , your task is to determine maximum possible integer  $k$  that is suitable for the specific supercomputer.

### Input

The input contains multiple (about  $10^5$ ) test cases.  
Each test case in only one line contains an integer  $m$  ( $1 \leq m \leq 10^5$ ).

### Output

For each test case, output “Case # $x$ :  $y$ ” in one line (without quotes), where  $x$  indicates the case number starting from 1, and  $y$  denotes the answer to the corresponding case.

### Example

| standard input | standard output |
|----------------|-----------------|
| 1              | Case #1: 0      |
| 64             | Case #2: 19     |

## Problem B. Balala Power!

Input file:           standard input  
Output file:         standard output  
Time limit:          2 seconds  
Memory limit:       128 megabytes



Talented Mr. Tang has  $n$  strings consisting of only lowercase letters. He wants to charge them with Balala Power (he could change each letter ranged from ‘a’ to ‘z’ into any number ranged from 0 to 25, but every two different letters must not be changed into the same number) so that he could casually calculate the sum of these strings as integers in the base of 26.

Mr. Tang wants you to maximize the sum. Note that no string in this problem can have any leading zeros when regarded as integers, except for the string “0”. Therefore, he guarantees that at least one type of letter would not appear at the beginning of any given string.

The sum may be quite large, so you should output it modulo  $(10^9 + 7)$  instead.

### Input

The input contains multiple (about 20) test cases.

For each test case, the first line contains an integer  $n$  ( $1 \leq n \leq 10^5$ ), indicating the number of strings.

The  $i$ -th of the next  $n$  lines contains a string  $s_i$  ( $1 \leq |s_i| \leq 10^5$ ) consisting of only lowercase letters. It is guaranteed that  $\sum_{i=1}^n |s_i| \leq 10^6$  for each test case.

### Output

For each test case, output “Case #x: y” in one line (without quotes), where  $x$  indicates the case number starting from 1, and  $y$  denotes the answer to the corresponding case.

### Example

| standard input | standard output |
|----------------|-----------------|
| 1              | Case #1: 25     |
| a              | Case #2: 1323   |
| 2              | Case #3: 18221  |
| aa             |                 |
| bb             |                 |
| 3              |                 |
| a              |                 |
| ba             |                 |
| abc            |                 |

## Problem C. Colorful Tree

Input file:           standard input  
Output file:         standard output  
Time limit:          2 seconds  
Memory limit:       128 megabytes

There is a tree having  $n$  nodes, the  $i$ -th node of which has a type of color, denoted by an integer  $c_i$ .

The path between every two nodes is unique, of which we define the value is the number of distinct types of colors appearing on it.

Calculate the sum of values of all possible paths,  $\frac{n(n-1)}{2}$  in total, between two different nodes on the tree.

### Input

The input contains multiple (about 50) test cases.

For each test case, the first line contains an integer  $n$  ( $2 \leq n \leq 2 \times 10^5$ ), indicating the number of nodes.

The next line contains  $n$  integers, the  $i$ -th number of which is  $c_i$  ( $1 \leq c_i \leq n$ ), denoting the color of the  $i$ -th node.

Each of the next  $(n - 1)$  lines contains two integers  $x, y$  ( $1 \leq x, y \leq n, x \neq y$ ), representing an edge between the  $x$ -th node and the  $y$ -th one. It is guaranteed that given edges form a tree.

### Output

For each test case, output “Case #x: y” in one line (without quotes), where  $x$  indicates the case number starting from 1, and  $y$  denotes the answer to the corresponding case.

### Example

| standard input                                                                  | standard output           |
|---------------------------------------------------------------------------------|---------------------------|
| 3<br>1 2 1<br>1 2<br>2 3<br>6<br>1 2 1 3 2 1<br>1 2<br>1 3<br>2 4<br>2 5<br>3 6 | Case #1: 6<br>Case #2: 29 |

## Problem D. Division Game

Input file:            **standard input**  
Output file:         **standard output**  
Time limit:          6 seconds  
Memory limit:       128 megabytes

There are  $k$  piles of stones in a circle, labeled from 0 to  $(k - 1)$ , where the number of the stones in each pile is  $n$  initially.

You can do some operations in rounds. The operation of the  $i$ -th round is to change the pile labeled  $((i - 1) \bmod k)$ , which allows you to remove some (at least one) stones from this pile, such that the old number of stones in this pile is a multiple of the new number. It also implies that you need to keep at least one stone in the pile after this operation.

The game will end if at least one pile contains only one stone. Given the positive integers  $n$  and  $k$ , your task is to calculate for each pile, the number of different ways to do operations which can cause the game to end so that this pile is the last changed one.

The integer  $n$  can be gigantic, so  $n$  would be given using its prime factorization, in other words, assuming that  $n = \prod_{i=1}^m p_i^{e_i}$  for distinct prime numbers  $p_1, p_2, \dots, p_m$ ,  $m$  and  $(p_i, e_i)$  would be given instead of  $n$ .

The answer may be fairly large, so you should output the answer modulo 985661441 instead.

### Input

The input contains multiple (about 200) test cases.

For each test case, the first line contains two integers  $m, k$  ( $1 \leq m, k \leq 10$ ).

The  $i$ -th line of the next  $m$  lines contains two integers  $p_i, e_i$  ( $2 \leq p_i \leq 10^9, e_i \geq 1$ ). It is guaranteed that  $p_1, p_2, \dots, p_m$  are distinct prime numbers and  $\sum_{i=1}^m e_i \leq 10^5$  for each test case.

It is also guaranteed that no more than 5 test cases satisfy that  $\sum_{i=1}^m e_i \geq 10^4$ .

### Output

For each test case, output “**Case #x:**  $y_0 \ y_1 \ \dots \ y_{k-1}$ ” in one line (without quotes), where  $x$  indicates the case number starting from 1, and  $y_i$  denotes the number of different ways modulo 985661441 for the pile labeled  $i$  in the corresponding case.

### Example

| standard input | standard output          |
|----------------|--------------------------|
| 1 1            | Case #1: 2               |
| 2 2            | Case #2: 3               |
| 2 1            | Case #3: 6 4             |
| 3 1            | Case #4: 1499980 1281085 |
| 5 1            |                          |
| 1 2            |                          |
| 2 3            |                          |
| 2 2            |                          |
| 2 4            |                          |
| 5 4            |                          |

## Problem E. Expectation of Division

Input file:           standard input  
Output file:         standard output  
Time limit:          3 seconds  
Memory limit:       128 megabytes

*To be frank with you, this problem is an extension of a classic problem, of which the colossal data range might increase its difficulty.*

Let's define a type of operation with respect to a positive integer  $n$  as replacing it with one of its positive factor  $d$ .

Given the integer  $n$  ( $n > 1$ ), you are asked to determine the expected times of doing this operation repeatedly until  $n$  is changed into 1 for the first time, assuming that  $d$  is selected with equal probability among all the possible factors at any time.

For ease of calculation,  $n$  and all its distinct prime factors  $p_1, p_2, \dots, p_m$  will be given.

To avoid the precision issue, let's express the expected value as a reduced fraction  $\frac{a}{b}$ , and you should output the minimum non-negative integer  $c$  such that  $bc \equiv a \pmod{10^9 + 7}$ .

### Input

The input contains multiple (about  $2 \times 10^5$ ) test cases.

For each test case, the first line contains two integers  $n, m$  ( $2 \leq n \leq 10^{24}, m \geq 1$ ), where  $m$  indicates the number of distinct prime factors of  $n$ .

The second lines contains  $m$  distinct prime numbers  $p_1, p_2, \dots, p_m$  ( $2 \leq p_1, p_2, \dots, p_m \leq 10^6$ ).

### Output

For each test case, output "Case #x: y" in one line (without quotes), where  $x$  indicates the case number starting from 1, and  $y$  denotes the answer to the corresponding case.

It is guaranteed that the answer exists for each test case.

### Example

| standard input | standard output    |
|----------------|--------------------|
| 2 1            | Case #1: 2         |
| 2              | Case #2: 500000006 |
| 4 1            | Case #3: 666666674 |
| 2              | Case #4: 833333342 |
| 6 2            | Case #5: 666666674 |
| 2 3            | Case #6: 233333338 |
| 8 1            |                    |
| 2              |                    |
| 10 2           |                    |
| 2 5            |                    |
| 12 2           |                    |
| 2 3            |                    |

## Problem F. Function

Input file:            **standard input**  
Output file:        **standard output**  
Time limit:         1 second  
Memory limit:      128 megabytes

You are given a permutation  $a$  obtained from 0 to  $(n-1)$ , and a permutation  $b$  obtained from 0 to  $(m-1)$ . Let's define a function  $f$ , which maps each integer between 0 and  $(n-1)$ , to an integer between 0 and  $(m-1)$ .

Please calculate the number of different functions  $f$  satisfying that  $f(i) = b_{f(a_i)}$  for each  $i$ .

Two functions are considered different if and only if there exists at least one integer mapped to different integers in these functions.

The answer can be a bit large, so you should output it modulo  $(10^9 + 7)$  instead.

### Input

The input contains multiple test cases.

For each test case, the first line contains two integers  $n, m$  ( $1 \leq n, m \leq 10^5$ ).

The second line contains  $n$  integers, the  $i$ -th number of which is  $a_{i-1}$  ( $0 \leq a_{i-1} \leq n-1$ ).

The third line contains  $m$  integers, the  $i$ -th number of which is  $b_{i-1}$  ( $0 \leq b_{i-1} \leq m-1$ ).

It is guaranteed that the sum of  $n$  and the sum of  $m$  in all test cases are no more than  $10^6$  respectively.

### Output

For each test case, output "**Case #x: y**" in one line (without quotes), where  $x$  indicates the case number starting from 1, and  $y$  denotes the answer to the corresponding case.

### Example

| standard input                                 | standard output          |
|------------------------------------------------|--------------------------|
| 3 2<br>1 0 2<br>0 1<br>3 4<br>2 0 1<br>0 2 3 1 | Case #1: 4<br>Case #2: 4 |

## Problem G. Gear Up

Input file:            **standard input**  
Output file:          **standard output**  
Time limit:           6 seconds  
Memory limit:        128 megabytes

**constroy** has some gears, whose radii may vary. Two gears may be adjacent due to one of the following conditions:

- they mesh together, which causes them to have equal linear velocity on edge; or
- they are fixed on the same shaft, which causes them to have equal angular velocity.

He guarantees that there are no two gears that satisfy both of the above conditions, and there is at most one transmission path between every two gears, where a transmission path is a sequence of different gears, in which two consecutive gears are adjacent.

Now, **constroy** assigns an angular velocity to one of these gears, and then he wants to know the largest angular velocity among them.

But **sd0061** thinks it is too easy for you, so he decides to replace some gears and then ask you the question.

### Input

The input contains multiple (about 30) test cases.

For each test case, the first line contains three integers  $n, m, q$  ( $0 \leq n, m, q \leq 10^5$ ,  $m < n$ ), indicating the number of gears, the number of adjacent pairs and the number of operations respectively.

The second line contains  $n$  integers, the  $i$ -th number of which is  $r_i$  ( $r_i \in \{2^0, 2^1, \dots, 2^{30}\}$ ), denoting the radius of the  $i$ -th gear.

Each of the next  $m$  lines contains three integers  $a, x, y$  ( $a \in \{1, 2\}$ ,  $1 \leq x, y \leq n$ ,  $x \neq y$ ), representing that the  $x$ -th gear and the  $y$ -th one are adjacent due to the  $a$ -th condition.

The next  $q$  lines describe the operations in chronological order. Each line contains three integers  $a, x, y$  ( $a \in \{1, 2\}$ ,  $1 \leq x \leq n$ ,  $y \in \{2^0, 2^1, \dots, 2^{30}\}$ ), representing an operation where:

- if  $a = 1$ , the operation is to replace the  $x$ -th gear with another one of radius  $y$ ;
- if  $a = 2$ , the operation is to assign an angular velocity of  $y$  to the  $x$ -th gear and then ask you to determine the maximum angular velocity among all the gears.

### Output

For each test case, firstly, output “**Case #x:**” in one line (without quotes), where  $x$  indicates the case number starting from 1.

Then, for each operation with  $a = 2$ , output a real number in one line, denoting the natural logarithm of the maximum angular velocity, with the precision for exactly three digits after the decimal point.

## Example

| standard input | standard output |
|----------------|-----------------|
| 4 3 4          | Case #1:        |
| 1 4 16 2       | 1.386           |
| 1 2 4          | Case #2:        |
| 1 2 3          | 2.773           |
| 2 1 4          | 3.466           |
| 1 1 16         | 2.773           |
| 1 2 4          |                 |
| 2 4 4          |                 |
| 1 4 16         |                 |
| 4 3 5          |                 |
| 2 16 4 8       |                 |
| 2 1 2          |                 |
| 1 2 3          |                 |
| 1 1 4          |                 |
| 2 1 4          |                 |
| 1 3 8          |                 |
| 2 1 16         |                 |
| 1 4 1          |                 |
| 2 1 8          |                 |

## Problem H. Hints of sd0061

Input file:           standard input  
Output file:         standard output  
Time limit:          3 seconds  
Memory limit:       128 megabytes

sd0061, a legend from Beihang University ACM-ICPC Team, retired last year leaving a group of noobs with no idea how to cope with  $m$  upcoming contests. What sd0061 left for them is only a set of hints.

There are  $n$  noobs in the team, the  $i$ -th of which has a rating,  $a_i$ . sd0061 prepared one hint for each contest. The hint for the  $j$ -th contest is a number,  $b_j$ , which means that the noob with the  $(b_j + 1)$ -th lowest rating is ordained by sd0061 to compete in the  $j$ -th contest.

The coach is going to ask constroy to list these contestants. Before that, constroy peeked these hints and found that  $b_i + b_j \leq b_k$  if  $b_i \neq b_j$ ,  $b_i < b_k$ ,  $b_j < b_k$ .

Now, you are in charge of making the list for constroy.

### Input

The input contains multiple (about 15) test cases.

For each test case, the first line contains five integers  $n, m, A, B, C$  ( $1 \leq n \leq 10^7$ ,  $1 \leq m \leq 100$ ,  $0 \leq A, B, C < 2^{32}$ ).

The second line contains  $m$  integers, the  $i$ -th number of which is  $b_i$  ( $0 \leq b_i < n$ ).

Ratings of these  $n$  noobs are obtained by calling the following C/C++ function  $n$  times, the  $i$ -th result of which is  $a_i$ .

```
unsigned x = A, y = B, z = C;
unsigned rng61() {
    unsigned t;
    x = x ^ (x << 16);
    x = x ^ (x >> 5);
    x = x ^ (x << 1);
    t = x;
    x = y;
    y = z;
    z = (t ^ x) ^ y;
    return z;
}
```

In this function, **unsigned** means the unsigned 32-bit integer type, and  $\wedge$  means the bitwise XOR operation. Note any integer overflow may occur in the function should be ignored, and  $x, y, z$  are reset to their initial values  $A, B, C$  respectively at the beginning of each test case.

### Output

For each test case, output “**Case #x:**  $y_1 \ y_2 \ \dots \ y_m$ ” in one line (without quotes), where  $x$  indicates the case number starting from 1, and  $y_i$  denotes the rating of the noob ordained for the  $i$ -th contest in the corresponding case.

### Example

| standard input                         | standard output                               |
|----------------------------------------|-----------------------------------------------|
| 3 3 1 1 1<br>0 1 2<br>2 2 2 2 2<br>1 1 | Case #1: 1 1 202755<br>Case #2: 405510 405510 |

## Problem I. I Curse Myself

Input file:            **standard input**  
Output file:          **standard output**  
Time limit:           4 seconds  
Memory limit:        128 megabytes

You are given a connected undirected graph having weighted edges, where it is guaranteed that each edge belongs to at most one simple cycle.

Assuming that the weight of a weighted spanning tree is the sum of weights on its edges, we define  $V(k)$  as the weight of the  $k$ -th smallest weighted spanning tree obtained from this graph, or in case that there are only less than  $k$  different weighted spanning trees, we define  $V(k)$  as zero.

Please calculate  $\left( \sum_{k=1}^K k \cdot V(k) \right) \bmod 2^{32}$ .

### Input

The input contains multiple test cases.

For each test case, the first line contains two integers  $n, m$  ( $2 \leq n \leq 1000, n-1 \leq m \leq 2n-3$ ), indicating the number of nodes and the number of edges in this graph respectively.

Each of the next  $m$  lines contains three integers  $x, y, z$  ( $1 \leq x, y \leq n, 1 \leq z \leq 10^6$ ), representing an edge of weight  $z$  between the  $x$ -th node and the  $y$ -th one. It is guaranteed that no multi-edges or self-loops are permitted in this graph.

The last line contains an integer  $K$  ( $1 \leq K \leq 10^5$ ).

It is guaranteed that the sum of  $K$  in all test cases is no more than  $10^6$ .

### Output

For each test case, output “**Case #x: y**” in one line (without quotes), where  $x$  indicates the case number starting from 1, and  $y$  denotes the answer to the corresponding case.

### Example

| standard input                                                                                                                                        | standard output                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| 4 3<br>1 2 1<br>1 3 2<br>1 4 3<br>1<br>3 3<br>1 2 1<br>2 3 2<br>3 1 3<br>4<br>6 7<br>1 2 4<br>1 3 2<br>3 5 7<br>1 5 3<br>2 4 1<br>2 6 2<br>6 4 5<br>7 | Case #1: 6<br>Case #2: 26<br>Case #3: 493 |

## Problem J. Journey with Knapsack

Input file:           standard input  
Output file:         standard output  
Time limit:          3 seconds  
Memory limit:       128 megabytes

KazaQ has a knapsack of volume  $2n$  and  $n$  kinds of food, where the volume of one piece of the  $i$ -th kind of food is  $i$ , and the number of pieces of that kind he would like to put into the knapsack is no more than  $a_i$ . If he put too much food in it, he would feel uncomfortable. Besides, since he has an odd taste for food,  $a_1, a_2, \dots, a_n$  are distinct.

KazaQ plans to go on a journey. Before that, he needs to take some food and one type of equipment he owns. He has  $m$  types of equipment, where the  $i$ -th one's volume is  $b_i$ , and some types of equipment may have the same volume.

KazaQ intends to know in how many ways he can pack up his belongings into his knapsack to make it full, where two ways are considered different if and only if the types of equipment he carries vary, or in case they are the same, the numbers of at least one kind of food in these two ways are different. Can you help him?

The answer may be extremely large, so you should output the answer modulo  $(10^9 + 7)$  instead.

### Input

The input contains multiple (about 100) test cases.

For each test case, the first line contains two integers  $n, m$  ( $1 \leq n \leq 5 \times 10^4, 1 \leq m \leq 2n$ ).

The second line contains  $n$  distinct integers  $a_1, a_2, \dots, a_n$  ( $0 \leq a_1 < a_2 < \dots < a_n \leq 2n$ ).

The third line contains  $m$  integers  $b_1, b_2, \dots, b_m$  ( $1 \leq b_1, b_2, \dots, b_m \leq 2n$ ).

It is guaranteed that no more than 5 test cases satisfy that  $n \geq 10^3$ .

### Output

For each test case, output “Case #x: y” in one line (without quotes), where  $x$  indicates the case number starting from 1, and  $y$  denotes the answer to the corresponding case.

### Example

| standard input | standard output |
|----------------|-----------------|
| 1 1            | Case #1: 1      |
| 1              | Case #2: 2      |
| 1              | Case #3: 6      |
| 2 2            |                 |
| 1 2            |                 |
| 3 4            |                 |
| 3 3            |                 |
| 1 2 3          |                 |
| 2 3 3          |                 |

## Problem K. KazaQ's Socks

Input file:           standard input  
Output file:         standard output  
Time limit:          1 second  
Memory limit:       128 megabytes

KazaQ wears socks every day.

Before the first day, he has  $n$  pairs of socks in his closet, labeled from 1 to  $n$ .

Every morning, he would put on a pair of socks with the smallest label in the closet.

Every evening, he would take off socks and put them into a basket. After that, if there are  $(n - 1)$  pairs of old socks in the basket, lazy KazaQ will have to wash them. These socks can be dried out on the next day and then will be put back to the closet in the evening.

KazaQ would like to know which pair of socks he would wear on the  $k$ -th day.

### Input

The input contains multiple (about 2000) test cases.

Each test case in only one line contains two integers  $n, k$  ( $2 \leq n \leq 10^9, 1 \leq k \leq 10^{18}$ ).

### Output

For each test case, output “**Case #x: y**” in one line (without quotes), where  $x$  indicates the case number starting from 1, and  $y$  denotes the answer to the corresponding case.

### Example

| standard input | standard output |
|----------------|-----------------|
| 3 7            | Case #1: 3      |
| 3 6            | Case #2: 1      |
| 4 9            | Case #3: 2      |

## Problem L. Limited Permutation

Input file:           standard input  
Output file:         standard output  
Time limit:          3 seconds  
Memory limit:       128 megabytes

For a permutation  $p_1, p_2, \dots, p_n$  obtained from 1 to  $n$ , it is uncomplicated for each  $i$  to calculate  $(l_i, r_i)$  meeting the condition that  $\min(p_L, p_{L+1}, \dots, p_R) = p_i$  if and only if  $l_i \leq L \leq i \leq R \leq r_i$ .

Given the positive integers  $n, (l_i, r_i)$ , you are asked to calculate the number of possible permutations  $p_1, p_2, \dots, p_n$  obtained from 1 to  $n$ , meeting the above condition.

The answer may be exceedingly large, so you should output the answer modulo  $(10^9 + 7)$  instead.

### Input

The input contains multiple test cases.

For each test case, the first line contains an integer  $n$  ( $1 \leq n \leq 10^6$ ).

The second line contains  $n$  integers  $l_1, l_2, \dots, l_n$  ( $1 \leq l_i \leq i$  for each  $i$ ).

The third line contains  $n$  integers  $r_1, r_2, \dots, r_n$  ( $i \leq r_i \leq n$  for each  $i$ ).

It is guaranteed that the sum of  $n$  in all test cases is no more than  $3 \times 10^6$ .

### Output

For each test case, output “**Case #x:**  $y$ ” in one line (without quotes), where  $x$  indicates the case number starting from 1, and  $y$  denotes the answer to the corresponding case.

### Example

| standard input | standard output |
|----------------|-----------------|
| 3              | Case #1: 2      |
| 1 1 3          | Case #2: 3      |
| 1 3 3          |                 |
| 5              |                 |
| 1 2 2 4 5      |                 |
| 5 2 5 5 5      |                 |