



**Problemset  
of  
Mirror of Independence Day  
Programming Contest 2023**

DEPARTMENT OF CSE  
**MIST**



**15 April 2023**

You get 13 Pages, 11 Problems and 300 Minutes.

## Problem A. Rain Rain Go Away, Come Again Another Day!

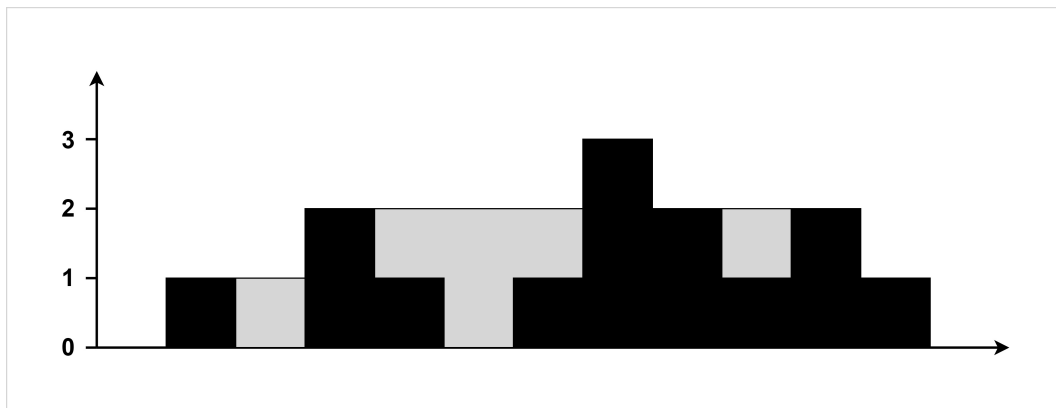
Input file:            `standard input`  
Output file:          `standard output`  
Time limit:           1 second  
Memory limit:        512 megabytes

One day on a rainy morning, Tahura was enjoying a cup of tea on her balcony when she noticed that some bricks were laying on the waterway trapping the rainwater that the waterway was supposed to drain. All the bricks were square-shaped and some were stacked on top of another. Now she wanted to see how much water it can trap after it had rained for several hours.

She made a few assumptions —

- The drain has infinite depth.
- The width of each brick is 1.
- The water beyond the ends on both sides of the brick structure is discarded. More specifically she only measured the water trapped between the first and the last brick.

Then she represented the brick structure with an array of integers. Each element of the array represents a stack of bricks and equals the number of bricks in that stack.



If The elevation map of the brick structure looked like this, the corresponding array would look like  $[0, 1, 0, 2, 1, 0, 1, 3, 2, 1, 2, 1]$ .

Given the array representing the elevation map, can you find the maximum amount of water that could be trapped in the brick structure for her?

### Input

The first line contains a single integer  $t$  ( $1 \leq t \leq 10^4$ ) — the number of test cases. The description of the test cases follows.

The first line of each test case contains a single integer  $n$  ( $1 \leq n \leq 2 \cdot 10^4$ ) — the length of the array.

The second line of each test case contains  $n$  integers  $h_i$  ( $0 \leq h_i \leq 10^5$ ) — the elevation array.

It is guaranteed that the sum of  $n$  over all test cases doesn't exceed  $2 \cdot 10^4$ .

### Output

For each test case, output the maximum amount of water that could be trapped by the elevated brick structure.

## Example

| standard input          | standard output |
|-------------------------|-----------------|
| 2                       | 6               |
| 12                      | 9               |
| 0 1 0 2 1 0 1 3 2 1 2 1 |                 |
| 6                       |                 |
| 4 2 0 3 2 5             |                 |

## Problem B. Signature Nightmare

Input file:            **standard input**  
Output file:         **standard output**  
Time limit:          1 second  
Memory limit:       512 megabytes

There are  $n$  offices and you need to get some signatures from each of them to complete a single form. You're given an array  $a_1, a_2, \dots, a_n$  representing the number of signatures needed from the  $i$ -th place to complete the form. You're also given an array  $b_1, b_2, \dots, b_n$  representing the maximum number of signatures in total you can get from the  $i$ -th place.

You can also go to the Boss and get at most  $k$  special signatures from him. Each of these signatures has the power to act as a substitute for any other required signature. 1 special signature can be used in place of exactly 1 other required signature.

Find out the maximum number of forms you can complete.

### Input

The first line contains a single integer  $t$  ( $1 \leq t \leq 10^4$ ) — the number of test cases. The description of the test cases follows.

The first line of each test case contains two integers  $n$  and  $k$  ( $1 \leq n \leq 10^5$ ;  $1 \leq k \leq 10^9$ ) — the number of offices and the number of special signatures granted to you.

The second line of each test case contains  $n$  integers  $a_i$  ( $1 \leq a_i \leq 10^9$ ) — the number of signatures needed from the  $i$ -th place to complete one form.

The third line of each test case contains  $n$  integers  $b_i$  ( $1 \leq b_i \leq 10^9$ ) — the maximum number of signatures in total you can get from the  $i$ -th place.

It is guaranteed that the sum of  $n$  over all test cases doesn't exceed  $10^5$ .

### Output

For each test case, output the maximum number of forms you can complete.

### Example

| standard input               | standard output |
|------------------------------|-----------------|
| 1<br>3 1<br>2 1 4<br>11 3 16 | 4               |

## Problem C. Optimal Pairing

Input file:            `standard input`  
Output file:          `standard output`  
Time limit:           1 second  
Memory limit:        512 megabytes

You're given an array  $a_1, a_2, \dots, a_n$ . You need to pair the elements of the array with each other into  $\frac{n}{2}$  pairs. The length of the array  $n$  is an even number.

The cost of a pair  $(a_i, a_j)$  is  $\max(a_i, a_j)$ . You need to find the minimum possible sum of costs of the  $\frac{n}{2}$  pairs, over all the possible ways to pair the elements with each other.

### Input

The first line contains a single integer  $t$  ( $1 \leq t \leq 10^4$ ) — the number of test cases. The description of the test cases follows.

The first line of each test case contains an integer  $n$  ( $1 \leq n \leq 10^5$ ) — the length of the array.

The second line of each test case contains  $n$  integers  $a_i$  ( $1 \leq a_i \leq 10^9$ ) — the array.

It is guaranteed that the sum of  $n$  over all test cases doesn't exceed  $10^5$ .

### Output

For each test case, output the minimum possible sum of costs of the  $\frac{n}{2}$  pairs in an optimal pairing.

### Example

| standard input            | standard output |
|---------------------------|-----------------|
| 1<br>8<br>7 4 3 2 1 5 8 6 | 20              |

## Problem D. Unwanted Divisors

Input file:            `standard input`  
Output file:         `standard output`  
Time limit:          2 seconds  
Memory limit:       512 megabytes

You're given an array of integers  $a_1, a_2, \dots, a_n$ . You need to process  $q$  queries. In each query, you're given an integer  $b$ . you need to find the number of divisors of  $b$  that aren't present in the array  $a$ .

### Input

The first line contains a single integer  $t$  ( $1 \leq t \leq 10^4$ ) — the number of test cases. The description of the test cases follows.

The first line of each test case contains two integers  $n$  and  $q$  ( $1 \leq n \leq 10^5$ ;  $1 \leq q \leq 10^5$ ) — the length of the array and the number of queries.

The second line of each test case contains  $n$  integers  $a_i$  ( $1 \leq a_i \leq 10^5$ ) — the array.

The third line of each test case contains  $q$  integers  $b_i$  ( $1 \leq b_i \leq 10^5$ ) — the queries.

It is guaranteed that the sum of  $n$  over all test cases doesn't exceed  $10^5$ , the sum of  $q$  doesn't exceed  $10^5$ .

### Output

For each test case, output the answers to the queries on separate lines.

### Example

| standard input | standard output |
|----------------|-----------------|
| 1              | 1               |
| 5 2            | 2               |
| 1 3 6 9 10     |                 |
| 6 18           |                 |

## Problem E. Unwanted Divisors Again

Input file:            **standard input**  
Output file:         **standard output**  
Time limit:          5 seconds  
Memory limit:       512 megabytes

You're given an array of integers  $a_1, a_2, \dots, a_n$  and an integer  $m$ . Find the number of divisors of  $m$  that don't have any multiples in  $a$ .

In other words, count the number of integers  $d$  satisfying -

- $d$  divides  $m$ .
- $d$  doesn't divide  $a_i$  for  $1 \leq i \leq n$ .

### Input

The first line contains a single integer  $t$  ( $1 \leq t \leq 100$ ) — the number of test cases. The description of the test cases follows.

The first line of each test case contains two integers  $n$  and  $m$  ( $1 \leq n \leq 10^6$ ;  $1 \leq m \leq 10^9$ ) — the length of the array and the given integer.

The second line of each test case contains  $n$  integers  $a_i$  ( $1 \leq a_i \leq 10^9$ ) — the array.

### Output

For each test case, output the number of divisors of  $m$  that don't have any multiples in  $a$ .

### Example

| standard input | standard output |
|----------------|-----------------|
| 3              | 1               |
| 3 6            | 0               |
| 1 3 10         | 1               |
| 1 1            |                 |
| 1              |                 |
| 4 12           |                 |
| 1 4 6 10       |                 |

## Problem F. Xored Pairs

Input file:           standard input  
Output file:         standard output  
Time limit:          1 second  
Memory limit:       512 megabytes

You need to construct an array  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i < 2^{30}$ ) that satisfies  $m$  given constraints. Each constraint is one of two types —

1.  $a_i \neq x$ , for given  $i$  and  $x$ .
2.  $a_i \oplus a_j = x$ , for given  $i, j$  and  $x$ .

### Input

The first line contains a single integer  $t$  ( $1 \leq t \leq 10^4$ ) — the number of test cases. The description of the test cases follows.

The first line of each test case contains two integers  $n$  and  $m$  ( $1 \leq n \leq 10^5$ ;  $1 \leq m \leq 10^5$ ) — the length of the required array and the number of constraints.

The following  $m$  lines each contain the description of a constraint —

- 1 i x — First type of constraint. It means  $a_i \neq x$ , ( $1 \leq i \leq n$ ;  $0 \leq x < 2^{30}$ ).
- 2 i j x — Second type of constraint. It means  $a_i \oplus a_j = x$ , ( $1 \leq i, j \leq n$ ;  $i \neq j$ ;  $0 \leq x < 2^{30}$ ).

It is guaranteed that the sum of  $n$  over all test cases doesn't exceed  $10^5$ , and the sum of  $m$  doesn't exceed  $10^5$ .

### Output

For each test case, if it is impossible to construct such an array, print "No".

Otherwise, print "Yes" in the first line. In the second line print  $n$  integers corresponding to the constructed array  $a$  ( $0 \leq a_i < 2^{30}$ ).

You can output the answer in any case (upper or lower). For example, the strings "yEs", "yes", "Yes", and "YES" will be recognized as positive responses.

If there are multiple solutions, print any.

### Example

| standard input | standard output                  |
|----------------|----------------------------------|
| 2              | Yes                              |
| 3 4            | 1013245966 1013245965 1013245964 |
| 1 1 0          | No                               |
| 2 1 2 3        |                                  |
| 2 2 3 1        |                                  |
| 2 3 1 2        |                                  |
| 3 4            |                                  |
| 1 1 0          |                                  |
| 2 1 2 4        |                                  |
| 2 2 3 1        |                                  |
| 2 3 1 2        |                                  |

## Problem G. Keyboard Warrior Roshid

Input file:            `standard input`  
Output file:        `standard output`  
Time limit:         1 second  
Memory limit:      512 megabytes

Roshid is a university student who works hard to maintain his grades and excel in his classes. He spends long hours studying and taking notes, both in the library and at home on his laptop. Currently, Roshid is in the thick of midterm exam season. He has been working hard to prepare for his impending examinations for weeks, reading through textbooks and taking copious notes. Roshid has been drafting up study materials and practice problems on his laptop for countless hours, but he's now run into trouble. His keyboard's fifth row, which includes the keys for the letters `z` to `m`, has abruptly stopped functioning.

Roshid panics for a moment, thinking about all the time he's spent preparing for his exams. How will he be able to complete his assignment if he can't type the characters in the fifth row? How will he conquer this exam war? Roshid realizes that he'll need to fix the issue before he can proceed with his studies. However, before he can do that, he wonders what his work will look like if he types the rest of the writing without the keys functioning properly.

### Input

The first line contains a single integer  $t$  ( $1 \leq t \leq 10^4$ ) — the number of test cases. The description of the test cases follows.

Each test case consists of a single line containing a non-empty string  $s$  ( $1 \leq |s| \leq 10^6$ ) only consisting of lowercase English letters — the text that Roshid wants to type.

It is guaranteed that the sum of  $|s|$  over all test cases doesn't exceed  $10^6$ .

### Output

Output a single line for each test case containing the resulting text that would be displayed on the screen if Roshid types the rest of the writing.

### Example

| standard input | standard output |
|----------------|-----------------|
| 3              | idp             |
| idpc           |                 |
| mcc            | ist             |
| mist           |                 |

## Problem H. Wonder Island

Input file:            `standard input`  
Output file:         `standard output`  
Time limit:          2 seconds  
Memory limit:       512 megabytes

Saimon is a British Billionaire and he wanted to spend his last Christmas vacation in Northern Canada. So, he took a couple of his friends and sailed his yacht on a sunny day in December. But after four days, while passing through the North Atlantic Ocean his ship hit an iceberg and everyone but him died. Somehow, he managed to survive and reached a new island country. The island was so beautiful and full of magical events.

Since Saimon is a successful businessman, among all the magical events, a business model attracted him the most. While observing the model for a couple of weeks, he found some interesting facts about this model. A special type of currency named Emm coin is used for all the transactions in this small island country. By investing in a pair of Emm coins, from the fourth day and on, it is possible to make a profit of another pair of Emm coins. Saimon somehow managed  $k$  pairs of Emm coins. Based on the information about the magical business model, how many Emm coins would he have after  $n$  days?

He tried to find the answer, but failed. In fact, he is great in business but poor in Math. So, he contacted your team and asked you to solve this problem for him.

### Input

The first line contains a single integer  $t$  ( $1 \leq t \leq 1000$ ) — the number of test cases. The description of the test cases follows.

The first line of each test case contains two integers  $k$  and  $n$  ( $1 \leq k \leq 1000$ ;  $1 \leq n \leq 10^5$ ) — the number of Emm coin pairs Saimon initially had and the total number of days.

### Output

For each test case, output the number of Emm coins Saimon would have after  $N$  days. Your answer might be arbitrarily large, so output the answer modulo  $10^9 + 7$ .

### Example

| standard input | standard output |
|----------------|-----------------|
| 3              | Case 1: 4       |
| 1 4            | Case 2: 18      |
| 1 8            | Case 3: 38      |
| 1 10           |                 |

## Problem I. Colorful Queries

Input file:            `standard input`  
Output file:         `standard output`  
Time limit:          1 second  
Memory limit:       512 megabytes

You have  $n$  items in a vertical stack, the color of the  $i$ -th item from the top is  $c_i$ .

You need to process  $q$  queries on this stack. In each query, you'll be given a color  $d$  and you need to —

- Find the top-most item of color  $d$  in the current stack and print its position from the top of the stack.
- And then, move this item to the top of the stack.

### Input

The first line contains a single integer  $t$  ( $1 \leq t \leq 10^4$ ) — the number of test cases. The description of the test cases follows.

The first line of each test case contains two integers  $n$  and  $q$  ( $1 \leq n \leq 10^5$ ;  $1 \leq q \leq 10^5$ ) — the number of items in the stack and the number of queries.

The second line of each test case contains  $n$  integers  $c_1, c_2, \dots, c_n$  ( $1 \leq c_i \leq n$ ) — colors of the items.

The third line of each test case contains  $q$  integers  $d_1, d_2, \dots, d_q$  ( $1 \leq d_i \leq n$ ) — colors in the queries.

It is guaranteed that the sum of  $n$  over all test cases doesn't exceed  $10^5$  and the sum of  $q$  doesn't exceed  $10^5$ . Also, for every  $d_i$  in the queries, there is at least one item in the stack with color  $d_i$ .

### Output

For each test case, output  $q$  lines containing the answers to the corresponding queries.

### Example

| standard input | standard output |
|----------------|-----------------|
| 1              | 5               |
| 7 5            | 2               |
| 2 1 1 4 3 3 1  | 3               |
| 3 2 1 1 4      | 1               |
|                | 5               |

## Problem J. Traveling Alien Masud

Input file:            `standard input`  
Output file:         `standard output`  
Time limit:          3 seconds  
Memory limit:       1024 megabytes

Masud found a map of the earth and is planning on visiting it. There are a total of  $n$  cities. Each city belongs to only one country. Two cities are connected via one-way roads. Two cities are considered to be in the same country if it is possible to reach the second city (directly or indirectly) starting from the first one and also reach the first city starting from the second one. Otherwise, the cities are of different countries. Now Masud is very busy and can only prepare the necessary documents to travel to at most two different countries. But he wants to visit as many cities as possible. Can you help him find the maximum number of cities he can visit by traveling to at most two countries?

### Input

The first line contains a single integer  $t$  ( $1 \leq t \leq 25$ ) — the number of test cases. The description of the test cases follows.

The first line of each test case contains two integers  $n$  and  $m$  ( $1 \leq n \leq 10^5$ ;  $1 \leq m \leq 10^5$ ) — the number of cities and the number of one-way roads connecting them.

The following  $m$  lines of the test case contain two integers  $u$  and  $v$  ( $1 \leq u, v \leq n$ ;  $u \neq v$ ) denoting a directional road from city  $u$  to city  $v$ .

### Output

For each test case, output the maximum number of cities Masud can visit during his visit to earth on a separate line.

### Example

| standard input                                                   | standard output |
|------------------------------------------------------------------|-----------------|
| 1<br>7 8<br>1 2<br>2 3<br>3 4<br>4 1<br>5 6<br>6 7<br>7 5<br>1 5 | 7               |

## Problem K. An Incantation Long Remembered

Input file:            `standard input`  
Output file:          `standard output`  
Time limit:           `3 seconds`  
Memory limit:        `512 megabytes`

You're given a list of abilities as an array of strings  $s_1, s_2, \dots, s_n$ . **All characters of each ability will be unique to each other.** You want to build a spell using these abilities. To make a spell these abilities can be inserted, appended, or prepended to the spell in multiple iterations. For example, let's consider the abilities are `abc`, `def`, and `pqrt`.

1. `""` (empty string)
2. `"abc"` (`abc` is appended/prepended in the empty string)
3. `"aefbc"` (`def` is inserted at index 1 of the previous string)
4. `"aefbcpqrt"` (`pqrt` is appended in the previous string)
5. `"pqrtadefbcpqrt"` (`pqrt` is prepended in the previous string)
6. `"pqrtdefabcbcpqrt"` (`abc` is inserted at index 7 of the previous string)
7. ...

And so on. The spells can be generated in many iterations. Each spell is valid except for the empty string at iteration 1.

Now, given a spell, can you verify whether the spell is valid or not?

### Input

The first line contains a single integer  $t$  ( $1 \leq t \leq 10^4$ ) — the number of test cases. The description of the test cases follows.

The first line of each test case contains an integer  $n$  ( $1 \leq n \leq 13$ ) — the number of abilities.

The following  $n$  lines of the test case will contain the ability strings  $s_i$  ( $1 \leq |s_i| \leq 26$ ).

The last line of the test case will contain the spell  $S$  consisting of lowercase English letter ( $1 \leq |S| \leq 10^5$ ).

It is guaranteed that the sum of  $|S|$  over all test cases doesn't exceed  $10^5$ .

Please note that, in each test case, all the characters of each ability will be unique to each other. In other words, no character appears twice among the abilities.

### Output

For each test case, if it is impossible to construct such a spell, print `No`.

Otherwise, print `Yes` in the first line. In the second line print the number of iterations required to create the spell using the abilities.

You can output the answer in any case (upper or lower). For example, the strings `yEs`, `yes`, `Yes`, and `YES` will be recognized as positive responses.

## Example

| standard input | standard output |
|----------------|-----------------|
| 2              | Yes             |
| 3              | 5               |
| abc            | No              |
| def            |                 |
| pqrt           |                 |
| pqrtadefbcpqrt |                 |
| 3              |                 |
| abc            |                 |
| def            |                 |
| pqrt           |                 |
| pqrtadefbcpqrt |                 |