

## A. Officer Anany Collecting String Subsequences

1 second, 64 megabytes

Anany is now serving in the army. Although he is an officer, in the first a few days, he is enlisted as trooper under training. The officer in charge of his training is the officer Mohamed Yasser. Mohamed Yasser gives Anany an exercise that he must do every morning. Anany has to walk on a string of characters. He must collect one 'A', then one 'B', then one 'C', ..., and finally one 'Z' in **alphabetical order**.

As you know, Anany is lazy, which is why fate brought him to the military service. In order to minimize his walking distance, he wants to move forward only in the smallest possible substring. That is, he wants to find the smallest substring that contains all the characters in the alphabet  $[A - Z]$  as a subsequence. Find only the length of that substring; you should let Anany depend on himself to find it.

### Input

The first line of the input contains an integer  $T$  ( $1 \leq T \leq 44$ ), the number of testcases.

The first line of each testcase contains an integer  $n$  ( $26 \leq n \leq 777$ ), the length of the string.

The second line contains a string  $s$  consisting of uppercase English letters ( $|s| = n$ ).

### Output

For each test case, output one integer denoting the length of the shortest substring that contains all the characters in the alphabet  $[A - Z]$  as a subsequence. It's guaranteed that there exist at least one such substring.

input
3 35 FORCESABCDEFDIVGHIJKLMNOPQRSTUVWXYZ 34 ABCDEFDIVGHIJKLMNICPCOPQRSTUVWXYZO 39 AINBSHAMSCPCDEFGHIJKLMANANYOPQRSTUVWXYZ
output
29 33 39

In the first test case, the solution is the substring  $s[13 \dots 41]$ :

"ABCDEFDIVGHIJKLMNOPQRSTUVWXYZ".

## B. Anany in the Army

2 seconds, 64 megabytes

Anany was happily doing his military service, but someday he got bored. He was daydreaming about the moment he gets back to problem-solving when he suddenly noticed three sticks on the ground. He challenged his soul (pun intended) to solve the following problem:

You can choose **one** stick and increase its length by any amount (integer or non-integer) less than or equal to  $k$  units. You will then form a triangle using the sticks. What is the maximum area of a triangle that you can get?

### Input

The first line of input contains an integer  $T$ , the number of test cases.

Each of the next  $T$  lines contains 4 space-separated integers each — the lengths of the 3 sticks, followed by  $k$ .

All the numbers are positive integers and do not exceed  $10^4$ .

It's guaranteed you can always form a non-degenerate triangle.

### Output

For each test case, print the maximum area on a single line. The answer will be considered correct if its absolute or relative error does not exceed  $10^{-4}$ .

input
2 1 1 1 2 3 4 4 1
output
0.500000000 6.928203230

## C. Sort?

1 second, 256 megabytes

Hamza has created the following sorting algorithm:

```
function sort(int u){
    if(vis[u])
        return
    vis[u] = 1
    ans.append(a[u])
    for i from 1 to n:
        if(a[u] and a[i])
            sort(i)
    for i from 1 to n:
        if((a[u] and not a[i]) == a[u])
            sort(i)
}
```

To sort an array, he calls `sort(1)`.

He doesn't know how the array ends up after running the algorithm (sorted?), so he's asking for your help.

### Input

The first line contains a single integer  $n$  ( $1 \leq n \leq 10^5$ ) — the length of the array.

The second line contains  $n$  space-separated integers denoting the array  $a$  ( $0 \leq a_i \leq 10^9$ ).

### Output

Print the final array after running the algorithm on it.

input
3 1 2 3
output
1 3 2

input
5 3 3 4 2 3
output
3 3 2 3 4

For simplicity, in this portion of the problem, we'll consider the least significant 3 bits.

The *and* between two integers  $x$  and  $y$  is the bitwise anding between the two numbers:

$1 \text{ and } 0 = 0 \text{ and } 1 = 0 \text{ and } 0 = 0$

$1 \text{ and } 1 = 1$

For example:

$7_{10}$  and  $5_{10} = 5_{10}$   
 $7_{10} = 111_2, 5_{10} = 101_2$   
 $111_2$  and  $101_2 = 101_2 = 5_{10}$

The not of an integer  $x$  is taking the complement of the integer (flipping it bits):

$not\ 0 = 1$

$not\ 1 = 0$

For example:

$not\ 5_{10} = 2_{10}$

Explanation:

$5_{10} = 101_2$

$not\ 101_2 = 010_2 = 2_{10}$

D. YSYS

2 seconds, 256 megabytes

Yassora the terrorist was on a wild run in Pillowoslovakia. Pillowoslovakia is a country that consists of some cities connected by bi-directional roads. Yassora's group has its headquarters in some unknown (yet) city. Enter Pillow Holmes to save the day.

Pillow Holmes knows the following about Yassora's movements:

- On day 0, he's in the headquarters.
- If on the  $i^{th}$  day he's in city  $u$ . Then, on the  $(i + 1)^{th}$  day, he'll be in city  $u$  or a city  $v$  connected to it by a road. In other words: everyday, he either stays put or moves along one road.
- He plants bombs as he moves, and he detonates them later.

Pillow Holmes gave you a log of the bombings that Yassora orchestrated. Each event in that log consists of a day and a city. An event  $bomb(d, c)$  means that a bomb that Yassora had planted a bomb in city  $c$  on day  $d$  or an earlier day, and it exploded on day  $d$ .

Pillow Holmes has also exploited the following technical difficulty in YSYS:

- The bombs explode in the same order they were planted by Yassora.** In other words: if he plants a bomb  $a$  and then another bomb  $b$ , he must detonate bomb  $a$  before bomb  $b$ .

Now, Pillow Holmes gave you the log of the bombings sorted in chronological order (sorted by day). He wants to know the number of possible cities the headquarters could exist. Remember that Yassora is in the headquarters on day 0.

Input

The first line contains 3 integers  $n, m$ , and  $q$  ( $2 \leq n \leq 2000, 1 \leq m \leq 10^4, 1 \leq q \leq 10^6$ ) — the number of cities, the number of roads, and the number of entries in the log respectively.

The following  $m$  lines contains the description of a road each, in the format  $u\ v$  ( $1 \leq u, v \leq n$ ) — the two cities connected by this road.

The following  $q$  lines contains two integers each  $d$  and  $c$  ( $1 \leq d \leq 10^7, 1 \leq c \leq n$ ) — which means that a bomb exploded on day  $d$  in city  $c$ .

It's guaranteed that Pillowoslovakia has no self-roads or multiple roads between the same pair of cities.

Output

Print the number of cities where the headquarters could be.

input
5 4 4 1 2 1 3 2 4 2 5 2 3 4 1 5 3 7 4
output
3

input
5 5 3 1 2 2 3 3 4 4 5 1 5 7 1 77 2 777 3
output
5

In the first sample, the cities where the headquarters could have been are 1, 2, and 3.

E. Baby Ehab's X(OR)

1 second, 256 megabytes

Baby Ehab has finished studying bitwise operations. He especially loved the two operations bitwise  $OR$  and bitwise  $XOR$  (of course).

He thought he'd mess with Baby Badawy and play with.. Well.. His cubes. Baby Badawy had an array  $a$  of cubes, lined up from left to right. The  $i_{th}$  cube had a number  $a_i$  written on it. On each of the following  $q$  operations, he would do one of the following 2 updates:

- $1\ l\ r$ : for every cube between positions  $l$  and  $r$ , Baby Ehab will replace the number on the  $i_{th}$  cube with  $a_i \mid (a_i - 1)$ , where  $\mid$  is the bitwise  $OR$  operator.
- $2\ l\ r$ : for every cube between positions  $l$  and  $r$ , Baby Ehab will replace the number on the  $i_{th}$  cube with  $a_i \oplus (a_i - 1)$ , where  $\oplus$  is the bitwise  $XOR$  operator.

Baby Badawy decided to make a stand against Ehab's bullying and all the babysitters were behind our golden boy. He told Ehab that he can tell him the sum of all the numbers in the array after each operation.

You, a stranger, decided to help our golden boy. Can you?

Input

The first line contains 2 integers  $n$  and  $q$  ( $1 \leq n, q \leq 3 \times 10^5$ ), the number of elements in the array  $a$  and the number of operations.

The second line contains  $n$  space-separated integers denoting the array  $a$  ( $1 \leq a_i \leq 3 \times 10^5$ ).

The following  $q$  lines will each contain an update as described in the statement ( $1 \leq l_i \leq r_i \leq n$ ).

Output

After each update, print the sum of the array on a single line.

input
3 3 1 2 3 1 1 3 2 2 2 2 1 3

output
7
5
3

input
5 5
5 5 5 5 5
1 5 5
2 1 1
2 2 3
1 3 4
2 1 5

output
25
21
13
13
5

F. Geometry?

1 second, 64 megabytes

Baby Ehab likes *xor*, but KEE likes geometry. That's why this next problem is a combination of both!

Given two lines having an integer power-of-2 slope each  $s_1$  and  $s_2$ , and passing through the point  $(0, 0)$ ; along with an interval on the  $x - axis$ , find the value of the following summation:

$$(\sum (y_{1,i} \oplus y_{2,i})) \bmod 1000000007$$

Where  $y_{1,i}$ ,  $y_{2,i}$  are the  $y - coordinates$  of the first and second line respectively for the  $i_{th}$  integer  $x - coordinate$  in the interval provided.

Input

The first line contains an integer  $T$  ( $1 \leq T \leq 1000$ ), the number of testcases.

Each testcase consists of one line containing 4 integers  $s_1, s_2, x_l, x_r$  ( $1 \leq s_1, s_2 \leq 2^{20}, 1 \leq x_l, x_r \leq 10^{12}$ ) where  $s_1, s_2$  are the slopes of the lines and  $x_l, x_r$  are the boundaries of the interval on the x-axis. It's guaranteed that  $s_1, s_2$  are an integer powers-of-2 (i.e.  $s = 2^k$  for some non-negative integer  $k$ ).

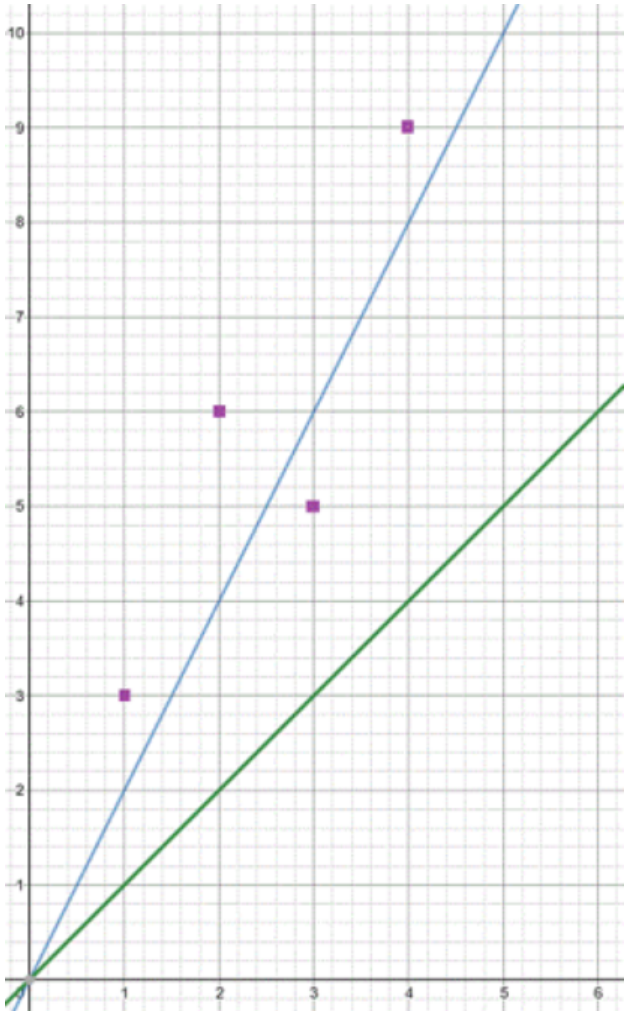
Output

For each testcase, output one line containing the sum of the *xor* of the y-coordinates on both lines in the given interval. To avoid large numbers, output the sum modulo 1000000007. The sum is calculated under the modulo; however, the xor at each individual sum is not.

input
3
1 2 1 4
4 8 3 7
128 1024 100000 125000

output
26
204
832054810

The first test:



The squares denotes the xor at each point.  
 $(1 \oplus 2) + (2 \oplus 4) + (3 \oplus 6) + (4 \oplus 8) = 3 + 6 + 5 + 12 = 26.$

It's recommended to use 64-bit integer types to avoid overflow.

G. Baby Ehab and a GCD Problem, Of Course

1 second, 256 megabytes

Ehab the baby was playing with special cubes that had numbers written on them.

Every while, Baby Badawy, who envied Ehab for being younger than him, would throw all the cubes with numbers between a pair of numbers ( $l$  and  $r$ ) he chose on Baby Ehab.

Every time Badawy threw cubes on Ehab, Ehab would just add them to his cubes, calculate the *GCD* of the numbers written on **all** of the cubes he has so far, and write the result on a piece of paper.

Note that the *GCD* of a sequence of numbers is the greatest number that divides all of them.

Can you guess the numbers Ehab wrote knowing which cubes Badawy threw each time?

Input

The first line contains a single integer  $q$  ( $1 \leq q \leq 10^5$ ) — the number of times Badawy threw cubes on Ehab.

Each of the following  $q$  lines will contain two integers  $l_i$  and  $r_i$  ( $1 \leq l_i \leq r_i \leq 10^{18}$ ) — meaning that for each number between  $l_i$  and  $r_i$  (inclusive), Badawy will throw a cube on Ehab with that number written on it.

Output

Output  $q$  integers each on a single line — the numbers that Ehab wrote down.

input
4 2250 2250 126 126 1 6 6 8
output
2250 18 1 1

## H. Shortest Array

1 second, 256 megabytes

Pillow is well known for being unexpected, so while all the students were amazed by how you can get the shortest path to all nodes in a graph in polynomial time, Pillow was asking himself: "If I know the shortest path from a node to all the other nodes, can I get that node?".

Formally, you are given a connected graph  $g$  with edges of positive weights, and an array  $s$ . You are asked to find a node  $x$  such that  $dist(x, y)$  equals  $s_y$  — where  $dist(x, y)$  is the the length of the shortest path between vertices  $x$  and  $y$  in  $g$ .

Yassora, a role model student, asked Pillow: "Wouldn't that be the node  $y$  such that  $s_y = 0$ ?".

Pillow, surprised from the naivety of that question, told Yassora that life wasn't that easy, so he decided that a shortest path needs to consist of at least one move, making Yassora utter the words: "Outstanding Move".

### Input

The first line contains two integers  $n$  and  $m$  ( $1 \leq n \leq 2 \times 10^5$ ,  $0 \leq m \leq 2 \times 10^5$ ) — the number of nodes and the number of edges in the graph.

The second line contains  $n$  space-separated integers denoting the array  $s$  ( $1 \leq s_i \leq 10^{15}$ ).

The following  $m$  lines contain information about the edges. Each line contains three integers  $u_i$ ,  $v_i$ , and  $w_i$  ( $1 \leq w_i \leq 10^9$ ,  $u_i \neq v_i$ ).

Note that our graph is a good graph. Good graphs do not contain loops or multiple edges.

### Output

If there's no node that generates this shortest path array, print "-1". Otherwise, print the **smallest indexed** node that generates this shortest path array.

input
4 4 6 3 6 6 1 3 6 1 2 3 3 2 3 4 2 3
output
1

input
4 4 6 3 4 8 1 3 4 1 2 3 2 4 5 3 4 4
output
1

input
5 6 8 4 4 8 6 1 2 4 1 3 4 2 4 4 3 4 4 1 5 3 5 3 2
output
4

## I. Ehab The Baby Learned Graphs

1 second, 64 megabytes

After learning to walk, Ehab the baby started learning graph theory. He never found the definition of the *xor* of 2 graphs, so he decided that the *xor* of 2 undirected graphs having the same number of vertices is defined as follows: the resulting graph has the same number of vertices and an edge exists in it if and only if it exists in exactly one of the two input graphs.

Now, Ehab made a new graph out of.. Well.. His cubes. The graph is undirected and connected, and it has  $n$  vertices and  $m$  edges. Ehab asked his "babysitter" Laggy to find at most  $n + m$  trees consisting of  $n$  nodes such that:

- His graph is the *xor* of them. The *xor* of one tree is itself. The *xor* of  $t > 1$  trees is (the *xor* of the first  $t - 1$  trees) *xor* (the last tree).
- If you take the diameter of each tree and take the maximum of them, the result should be as small as possible. The diameter of a tree is the length of its longest path.

Laggy, of course, didn't solve it, so it's your turn. Can you solve it?

### Input

The first line contains a single integer integer  $n$  ( $2 \leq n \leq 100$ ) — the number of vertices in the graph.

The next  $n$  lines contain  $n$  integers each. If the  $j_{th}$  number in the  $i_{th}$  line is 1, then vertices  $i$  and  $j$  are connected by an edge. If it's 0, then they're not.

It's guaranteed that the graph is connected and doesn't have self-loops.

### Output

If it's impossible to solve the problem for this case, print one line containing "-1". Otherwise:

On the first line, print the number of trees  $t$  you wish to use. It should be at most  $n + m$ .

On the next  $t$  output blocks, you should print the trees. Each block consists of  $n - 1$  lines containing 2 integers  $u$  and  $v$  ( $1 \leq u, v \leq n$ ) which mean that there's an edge between nodes  $u$  and  $v$  in this tree. Each block must describe a tree.

If there are multiple solutions, print any of them. You can print the edges of a tree in any order.

input
4 0 1 1 0 1 0 1 0 1 1 0 1 0 0 1 0

output
4
3 1
3 2
1 4
3 2
3 4
4 1
2 3
2 4
3 1
2 1
2 4
1 3

input
3
0 1 1
1 0 1
1 1 0
output
-1

input
3
0 1 1
1 0 0
1 0 0
output
1
1 2
1 3

### J. ABC

1 second, 256 megabytes

You're given an string  $s$  of length  $n$ . Each character in it is either 'a', 'b', or 'c'. In addition, the string contains at most 1 'b' character.

You can swap the values of any two indices. Find the minimum number of swaps needed so that for every index  $i$  ( $1 \leq i \leq n - 1$ ), either  $s_i = s_{i+1}$  or  $s_i = \text{'b'}$  or  $s_{i+1} = \text{'b'}$ .

#### Input

The first line contains a single integer  $n$  ( $1 \leq n \leq 10^5$ ) — the length of the string.

The second line contains the string  $s$ .

#### Output

If there's no way to satisfy the condition in the statement, print "-1". Othwerwise, print the minimum number of swaps to achieve the condition.

input
3
acb
output
1

### K. Plants Watering

1 second, 256 megabytes

Pillow has a garden of plants of various heights, forming a beautiful line from left to right. Each plant grows with its own rate. Pillow had just taught The Captain about sorted sequences. The Captain wanted to know the first integer moment of time when the sequence of plants will become sorted in a non-decreasing order according to height.

Formally, you are given two arrays  $h$  and  $g$ , where plant  $i$  has a height of  $h_i$  units initially, and each moment its height increases by  $g_i$  units.

Find the first integer moment of time when the sequence of plants will become sorted in non-decreasing order according to height.

#### Input

The first line contains a single integer  $n$  ( $1 \leq n \leq 10^5$ ) — the number of plants.

The second line contains  $n$  space-separated integers denoting the array  $h$  ( $1 \leq h_i \leq 10^9$ ).

The third line contains  $n$  space-separated integers denoting the array  $g$  ( $1 \leq g_i \leq 10^9$ ).

#### Output

If there's no such moment, print "-1". Otherwise, print the first one.

input
5
1 2 3 4 5
5 4 3 2 1
output
0

input
4
2 3 1 4
1 2 4 3
output
1

input
4
3 2 4 1
3 2 5 20
output
-1

### L. The Expected Square

1 second, 64 megabytes

Ehab the baby likes playing games with *XOR* (of course), so he will play the following game. He is given some number  $n$ . Then, he defines a number  $x$  which is initially 0. In each move, he thinks of a number  $r$ , where  $r$  is a non-negative number strictly less than  $2^n$  chosen **uniformly at random**, and changes  $x$  to  $x \oplus r$ , where  $\oplus$  denotes the bitwise *XOR* operation. After making the first move, he stops when  $x$  becomes 0 again.

Baby Ehab was thinking of the expected value of **the square** of the number of moves that he will play. More formally, let  $m$  be the number of moves he plays in a game, then Baby Ehab wants to know  $E(m^2)$ , Can you know it?

Let the answer be represented as  $p/q$ , then you are required to find the value of  $p * q^{-1} \mod 10^9 + 7$ .

#### Input

The first line of input contains  $t$ , the number of test cases.

Each of the following  $t$  lines contains one integer,  $n$  ( $1 \leq n \leq 10^9$ ).

#### Output

For each test case, output the required answer on a single line.

input
3
2
3
10
output
28
120
2096128

### M. Baby Ehab's Whining Chance

1 second, 256 megabytes

Baby Ehab had just been stung by increasing subsequences in IOI, so he decided to make himself feel better about his skills. He chose an integer  $n$ , wrote down all integers from 1 to  $n$ , and calculated the longest increasing subsequence like a true genius! Enter Baby Badawy again as envious as ever.

Baby Badawy decided to play a trick on Baby Ehab. He replaced every integer in the sequence with the sum of its digits. Of course, Baby Ehab started crying. It was IOI all over again!

So, his "babysitter" Laggy hired you to find the longest increasing subsequence of this sequence so that Baby Ehab would stop crying.

Hurry up! You only have 300 minutes before Baby Ehab dies of dehydration.

#### Input

The only line contains a single integer  $n$  ( $1 \leq n \leq 10^{100000}$ ).

#### Output

Print the longest increasing subsequence required in the statement.

input
10
output
9

input
99

output
18

input
777
output
24

## N. Baby !Ehab

1 second, 256 megabytes

Baby !Ehab found this problem really hard: given 3 numbers, check if **at least** 2 of them are equal.

#### Input

The first line contains an integer  $T$ , the number of testcases.

The next  $T$  lines, each contains 3 positive numbers. The numbers are between 1 and 20.

#### Output

For each testcase, print "YES" if **at least** 2 of the 3 numbers are equal, "NO" otherwise.

input
2 1 2 2 2 10 20
output
YES NO