

Problem A. Blank

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

There are N blanks arranged in a row. The blanks are numbered $1, 2, \dots, N$ from left to right.

Tom is filling each blank with one number in $\{0, 1, 2, 3\}$. According to his thought, the following M conditions must all be satisfied. The i_{th} condition is:

There are exactly x_i different numbers among blanks $\in [l_i, r_i]$.

In how many ways can the blanks be filled to satisfy all the conditions? Find the answer modulo 998244353.

Input

The first line of the input contains an integer $T(1 \leq T \leq 15)$, denoting the number of test cases.

In each test case, there are two integers $n(1 \leq n \leq 100)$ and $m(0 \leq m \leq 100)$ in the first line, denoting the number of blanks and the number of conditions.

For the following m lines, each line contains three integers l, r and x , denoting a condition ($1 \leq l \leq r \leq n$, $1 \leq x \leq 4$).

Output

For each testcase, output a single line containing an integer, denoting the number of ways to paint the blanks satisfying all the conditions modulo 998244353.

Example

standard input	standard output
2	4
1 0	96
4 1	
1 3 3	

Problem B. Operation

Input file: **standard input**
Output file: **standard output**
Time limit: 4 seconds
Memory limit: 256 megabytes

There is an integer sequence a of length n and there are two kinds of operations:

- **0 l r**: select some numbers from $a_l \dots a_r$ so that their **xor** sum is maximum, and print the maximum value.
- **1 x**: append x to the end of the sequence and let $n = n + 1$.

Input

There are multiple test cases. The first line of input contains an integer $T (T \leq 10)$, indicating the number of test cases.

For each test case:

The first line contains two integers $n, m (1 \leq n \leq 5 \times 10^5, 1 \leq m \leq 5 \times 10^5)$, the number of integers initially in the sequence and the number of operations.

The second line contains n integers $a_1, a_2, \dots, a_n (0 \leq a_i < 2^{30})$, denoting the initial sequence.

Each of the next m lines contains one of the operations given above.

It's guaranteed that $\sum n \leq 10^6, \sum m \leq 10^6, 0 \leq x < 2^{30}$.

And operations will be encrypted. You need to decode the operations as follows, where **lastans** denotes the answer to the last type 0 operation and is initially zero:

For every type 0 operation, let $l = (l \text{ xor lastans}) \bmod n + 1$, $r = (r \text{ xor lastans}) \bmod n + 1$, and then swap l and r if $l > r$.

For every type 1 operation, let $x = x \text{ xor lastans}$.

Output

For each type 0 operation, please output the maximum **xor** sum in a single line.

Example

standard input	standard output
1	1
3 3	3
0 1 2	
0 1 1	
1 3	
0 3 4	

Problem C. Milk

Input file: **standard input**
Output file: **standard output**
Time limit: 3 seconds
Memory limit: 256 megabytes

As we all know, Tom loves milk very much, but he never manages to finish a bottle of milk by himself, cause Jerry always swallows at least half of it!

But now, he is alone in the Wonderland of Milk! The Wonderland can be seen as a grid with n rows and m columns. Tom numbers the rows with integers from 1 to n from top to bottom and the columns with integers from 1 to m from left to right. He starts from the upper-left cell $(1, 1)$.

Tom can move down if and only if he is in the middle column of the grid (the $\frac{m+1}{2}$ th column). Otherwise, he can only go left or right. Each movement takes him 1 second.

There are k bottles of milk on the grid. The i th bottle is located in the r_i th row and c_i th column, and Tom needs t_i seconds to finish it.

In order to enjoy the milk alone, he wants to do it as fast as possible. So he wants to know the minimum time he needs to finish i ($i = 1, 2, \dots, k$) bottles of milk.

Input

The first line contains one integer T ($1 \leq T \leq 2500$), the number of test cases.

For each test case, the first line contains 3 integers n, m, k ($3 \leq n, m \leq 10^9, 1 \leq k \leq 10^4, \sum k \leq 6 \times 10^4$), where m is guaranteed to be an odd number.

Each of the next k lines contains 3 integers r_i, c_i, t_i ($1 \leq r_i \leq n, 1 \leq c_i \leq m, c_i \neq \frac{m+1}{2}, 1 \leq t_i \leq 10^9$), the position of i th bottle of milk and the time it takes Tom to drink.

Output

For each test case, output one line containing k integers separated by space, the i th of which indicates the minimum time Tom needs to finish exactly i bottles of milk.

Example

standard input	standard output
1 7 7 2 3 7 1 4 5 1	8 15

Problem D. Vacation

Input file: **standard input**
Output file: **standard output**
Time limit: 5 seconds
Memory limit: 256 megabytes

Tom and Jerry are going on a vacation. They are now driving on a one-way road and several cars are in front of them. To be more specific, there are n cars in front of them. The i th car has a length of l_i , the head of it is s_i from the stop-line, and its maximum velocity is v_i . The car Tom and Jerry are driving is l_0 in length, and s_0 from the stop-line, with a maximum velocity of v_0 .

The traffic light has a very long cycle. You can assume that it is always green light. However, since the road is too narrow, no car can get ahead of other cars. Even if your speed can be greater than the car in front of you, you still can only drive at the same speed as the anterior car. But when not affected by the car ahead, the driver will drive at the maximum speed. You can assume that every driver here is very good at driving, so that the distance of adjacent cars can be kept to be 0.

Though Tom and Jerry know that they can pass the stop-line during green light, they still want to know the minimum time they need to pass the stop-line. We say a car passes the stop-line once the head of the car passes it.

Please notice that even after a car passes the stop-line, it still runs on the road, and cannot be overtaken.

Input

This problem contains multiple test cases.

For each test case, the first line contains an integer n ($1 \leq n \leq 10^5$, $\sum n \leq 2 \times 10^6$), the number of cars.

The next three lines each contains $n + 1$ integers, l_i, s_i, v_i ($1 \leq s_i, v_i, l_i \leq 10^9$). It's guaranteed that $s_i \geq s_{i+1} + l_{i+1}, \forall i \in [0, n - 1]$

Output

For each test case, output one line containing the answer. Your answer will be accepted if its absolute or relative error does not exceed 10^{-6} .

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

The answer is guaranteed to exist.

Example

standard input	standard output
1	3.5000000000
2 2	5.0000000000
7 1	
2 1	
2	
1 2 2	
10 7 1	
6 2 1	

Problem E. Path

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Years later, Jerry fell in love with a girl, and he often walks for a long time to pay visits to her. But, because he spends too much time with his girlfriend, Tom feels neglected and wants to prevent him from visiting her.

After doing some research on the neighbourhood, Tom found that the neighbourhood consists of exactly n houses, and some of them are connected with directed road. To visit his girlfriend, Jerry needs to start from his house indexed 1 and go along the shortest path to hers, indexed n .

Now Tom wants to block some of the roads so that Jerry has to walk longer to reach his girl's home, and he found that the cost of blocking a road equals to its length. Now he wants to know the minimum total cost to make Jerry walk longer.

Note, if Jerry can't reach his girl's house in the very beginning, the answer is obviously zero. And you don't need to guarantee that there still exists a way from Jerry's house to his girl's after blocking some edges.

Input

The input begins with a line containing one integer $T(1 \leq T \leq 10)$, the number of test cases.

Each test case starts with a line containing two numbers $n, m(1 \leq n, m \leq 10000)$, the number of houses and the number of one-way roads in the neighbourhood.

m lines follow, each of which consists of three integers $x, y, c(1 \leq x, y \leq n, 1 \leq c \leq 10^9)$, denoting that there exists a one-way road from the house indexed x to y of length c .

Output

Print T lines, each line containing a integer, the answer.

Example

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

Problem F. Typewriter

Input file: standard input
Output file: standard output
Time limit: 1.5 seconds
Memory limit: 256 megabytes

One day, Jerry found a strange typewriter. This typewriter has 2 input modes: pay p coins to append an arbitrary single letter to the back, or q coins to copy a string that has already been outputted and paste it in the back.

Jerry now wants to write a letter to Tom. The letter is a string S which contains only lowercase Latin letters. But as Jerry is not very wealthy, he wants to know the minimum number of coins he needs to write this letter.

Input

This problem contains multiple test cases. Process until the end of file.

For each test case, the first line contains string S ($|S| \leq 2 \times 10^5$, $\sum |S| \leq 5 \times 10^6$), consisting of only lowercase Latin letters. And the second line contains 2 integers p and q ($1 \leq p, q < 2^{31}$).

Output

For each test case, output one line containing the minimum number of coins Jerry needs to pay.

Example

standard input	standard output
abc	3
1 2 aabaab	6
2 1	

Problem G. Meteor

Input file: standard input
Output file: standard output
Time limit: 7.5 seconds
Memory limit: 256 megabytes

Though time passes, there is always someone we will never forget.

"The probability of being hit by a meteor is one in a billion, but it is much more miraculous, to meet you in my life." said Tom to Jerry with affection.

"One in a billion? I may not agree with you." answered Jerry proudly, *"Let's do the math."*

...

Thinking of the days they have happily spent together, Tom can't help bursting into tears. Though Jerry has been gone for a long time, Tom still misses him every day. He remembers it was a sunny afternoon when Jerry and him were lying in the yard, working on the probability of a man being hit by a meteor.

Unlike Jerry, he was always slow. Jerry got the answer soon, but Tom was stuck as usual. In the end, Tom lost patience and asked Jerry to tell him the answer.

"I can't be so straightforward," snickered Jerry, *"the only thing I will tell you is that the answer is $\frac{p}{q}$, where $p, q \leq n$, $\gcd(p, q) = 1$."*

"Is it $\frac{1}{n}$?"

"Is it $\frac{1}{n-1}$?"

...

If answered *"No"*, he would try the minimum larger number that satisfies the requirement.

Tom only remembered n given by Jerry, and k , the times that he tried, but forgot what matters the most: Jerry's answer. Now, he wants you to help him find it.

Input

The first line contains an integer $T(T \leq 10^2)$, the number of test cases.

The next T lines, each line contains two numbers, $n, k(2 \leq n \leq 10^6)$, indicating a query.

The answer is guaranteed to be in $(0, 1]$.

Output

T lines, each line contains a fraction in term of p/q , where $\gcd(p, q) = 1$.

Example

standard input	standard output
5	1/1
4 6	1/5
5 1	1/3
9 9	1/1
3 4	3/5
7 11	

Note

g++ compiler on HDOJ supports `__int128`.

Problem H. Desert

Input file: **standard input**
 Output file: **standard output**
 Time limit: 12 seconds
 Memory limit: 256 megabytes

Tom and Jerry got lost in a desert. But they are not bored or afraid at all, because they found a truly interesting thing to do: researching the varied kinds of kakti in the desert!

After careful observations, they found that a kaktus can be seen as a graph consisting of several nodes. To be precise, an n -node kaktus is an undirected connected graph that has only 1 node as the root, in which every edge belongs to exactly one simple cycle and all nodes are unlabelled. You can refer to sample case for a better understanding.

Finding something new is always exciting, but it brings them new doubts.

"How many different kinds of n -node kakti are there?" wondered Tom.

"Your problem is simpler than mine. I am wondering how many different kinds of 1, 2, 3, ..., n -node kakti there are." said Jerry.

Input

One line containing a number n ($1 \leq n \leq 10^5$).

Output

n lines, each line containing a number, the number of n -node kakti.

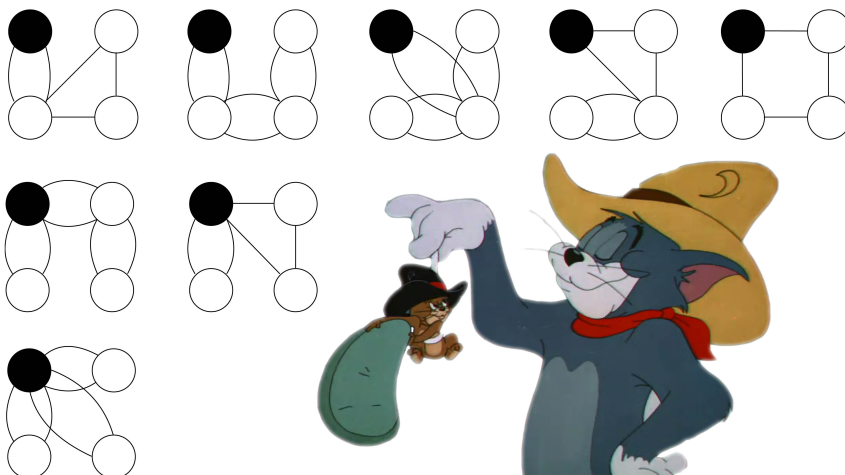
As these numbers can be very large, output them modulo 998244353.

Example

standard input	standard output
4	1 1 3 8

Note

Here are the 8 4-node kakti.



Problem I. String

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Tom has a string containing only lowercase letters. He wants to choose a subsequence of the string whose length is k and lexicographical order is the smallest. It's simple and he solved it with ease.

But Jerry, who likes to play with Tom, tells him that if he is able to find a lexicographically smallest subsequence satisfying following 26 constraints, he will not cause Tom trouble any more.

The constraints are: the number of occurrences of the i th letter from **a** to **z**(indexed from 1 to 26) must in $[L_i, R_i]$.

Tom gets dizzy, so he asks you for help.

Input

The input contains multiple test cases. Process until the end of file.

Each test case starts with a single line containing a string S ($|S| \leq 10^5$) and an integer k ($1 \leq k \leq |S|$).

Then 26 lines follow, each line two numbers L_i, R_i ($0 \leq L_i \leq R_i \leq |S|$).

It's guaranteed that S consists of only lowercase letters, and $\sum |S| \leq 3 \times 10^5$.

Output

Output the answer string.

If it doesn't exist, output -1 .

Problem J. Kingdom

Input file: **standard input**
Output file: **standard output**
Time limit: 3 seconds
Memory limit: 256 megabytes

Tom and Jerry were sent to a kingdom for some intelligence purpose. They were ordered to draw a map of the kingdom.

It's known that the kingdom consists of n cities, numbered from 1 to n , along with $n - 1$ bidirectional roads connecting them. The cities form a binary tree, whose root is the kingdom's capital. The nodes on the tree either have no child, or a single left child, or a single right child, or both.

To avoid being caught spying, the commander ordered Tom and Jerry not to directly draw the map, but to record some specific information separately, which combined can deduce what the kingdom looks like.

When on a tree node, Tom was ordered to:

1. Record the label of current node.
2. Go to left child node(if any).
3. Go to right child node(if any).

Jerry was ordered to:

1. Go to left child node(if any).
2. Record the label of current node.
3. Go to right child node(if any).

But things often go south in Tom and Jerry's lives. When returned homeland with their information, they found that some of the numbers they wrote down are spotted. In despair, they decide to calculate how many kinds of form can the kingdom be according to the remaining information. As the number is large, you are only required to print it mod 998244353.

Input

The first line contains an integer $T(1 \leq T \leq 10)$, the number of test cases. Then T cases follow.

For each test case:

In the first line, a integer $n(1 \leq n \leq 100)$, the number of cities in the kingdom.

In the second line, n integers $a_1, a_2, a_3, \dots, a_n$ separated by space, Tom's sequence.

In the third line, n integers $b_1, b_2, b_3, \dots, b_n$ separated by space, Jerry's sequence($0 \leq a_i, b_i \leq n$).

The spotted numbers are indicated by 0.

Output

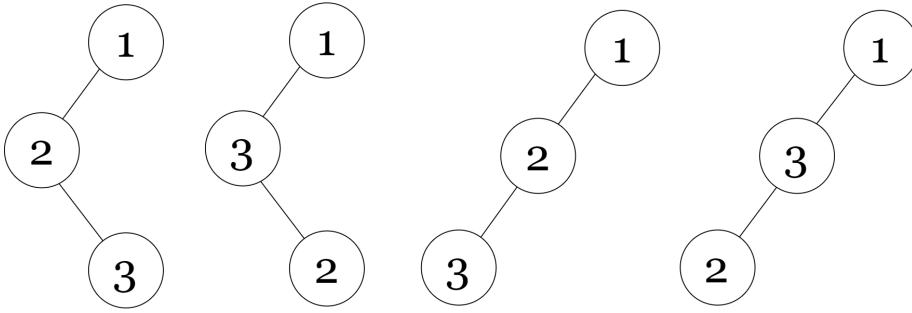
For each test case, output one line containing one integer, the number of corresponding kingdom's form mod 998244353.

Example

standard input	standard output
1 3 1 0 0 0 0 1	4

Note

Here are the 4 possible forms.



Problem K. Function

Input file: standard input
Output file: standard output
Time limit: 5 seconds
Memory limit: 256 megabytes

Jerry is fond of functions. He thinks the mystery of the universe is hidden behind the notations, variables and numbers.

Of all functions, he thinks gcd and $\lfloor x \rfloor$ are the most fascinating, and that something combines gcd with truncation should be even more marvelous.

Therefore, he comes up with a problem: calculate

$$\sum_{i=1}^n \gcd(\lfloor \sqrt[3]{i} \rfloor, i) \pmod{998244353}$$

Input

The input contains several test cases.

The first line contains an integer $T(1 \leq T \leq 11)$, the number of test cases.

Each of the next T lines contains a integer $n(1 \leq n \leq 10^{21})$, indicating a query.

Output

For each test case, output one line containing an integer indicating the corresponding answer.

Example

standard input	standard output
10	103
64	327
180	1069
526	522
267	1693
775	1379
649	1631
749	1998
908	606
300	492
255	

Note

g++ compiler on HDOJ supports `__int128`.

Problem L. Sequence

Input file: **standard input**
Output file: **standard output**
Time limit: 3 seconds
Memory limit: 256 megabytes

Tom gets an integer sequence a indexed from 1 with length n from Jerry and he wants to apply k kinds of operations to this sequence.

For a type k operation, he calculates $b_i = \sum_{j=i-k \cdot x} a_j (0 \leq x, 1 \leq j \leq i)$ for every i ranged from 1 to n and then replaces each a_i with $b_i \bmod 998244353$.

He wonders what the final sequence looks like after m operations.

Input

The first line contains an integer $T (T \leq 10)$, denoting the number of testcases.

For each test case, the first line contains two integers n and $m (1 \leq n \leq 10^5, 1 \leq m \leq 10^6)$, representing the length of sequence a and the number of operations.

The following line contains n integers denoting the sequence $a (1 \leq a_i \leq 10^9)$.

The last line contains m integers representing the sequence of operations. Let c_i be the i th number in this sequence, it means the type of i th operation is $c_i (1 \leq c_i \leq 3)$.

It is guaranteed that $\sum n \leq 2.1 \times 10^5, \sum m \leq 2.1 \times 10^6$.

Output

For each test case, output one line containing an integer ans .

$ans = (1 \cdot a_1) \oplus (2 \cdot a_2) \oplus \dots \oplus (n \cdot a_n)$, a_i is the i th element of sequence a after m operations, \oplus means bitwise exclusive OR operation.

Example

standard input	standard output
2	233
5 2	121
2 4 2 1 1	
1 1	
5 2	
3 2 2 4 1	
2 2	

Problem M. Code

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

After returning with honour from ICPC(International Cat Programming Contest) World Finals, Tom decides to say goodbye to ICPC and start a new period of life. He quickly gets interested in AI.

In the subject of Machine Learning, there is a classical classification model called perceptron, defined as follows:

Assuming we get a set of training samples: $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$, with their inputs $\mathbf{x} \in \mathbb{R}^d$, and outputs $y \in \{-1, 1\}$. We will try to find a function $f(\mathbf{x}) = \text{sign}(\sum_{i=1}^d w_i \cdot x_i + b) = \text{sign}(\mathbf{w}^T \cdot \mathbf{x} + b)$ so that $f(\mathbf{x}_i) = y_i, i = 1, 2, \dots, N$.

\mathbf{w}, \mathbf{x} mentioned above are all d -dimensional vectors, i.e. $\mathbf{w} = (w_1, w_2, \dots, w_d), \mathbf{x} = (x_1, x_2, \dots, x_d)$. To simplify the question, let $w_0 = b, x_0 = 1$, then $f(\mathbf{x}) = \text{sign}(\sum_{i=0}^d w_i \cdot x_i) = \text{sign}(\mathbf{w}^T \cdot \mathbf{x})$. Therefore, finding a satisfying function $f(\mathbf{x})$ is equivalent to finding a proper \mathbf{w} .

To solve the problem, we have a algorithm, *PLA(Popcorn Label Algorithm)*.

According to *PLA*, we will randomly generate \mathbf{w} .

If $f(\mathbf{x}) = \text{sign}(\mathbf{w}^T \cdot \mathbf{x})$ fails to give any element $(\mathbf{x}_i, y_i) \in D$ the right classification, i.e. $f(\mathbf{x}_i) \neq y_i$, then we will replace \mathbf{w} with another random vector. We will do this repeatedly until all the samples $\in D$ are correctly classified.

As a former-JBer, Tom excels in programming and quickly wrote the pseudocode of *PLA*.

```
w := a random vector
while true do
  flag:=true
  for i:=1 to N do
    if f(x[i]) != y[i] then
      flag:=false
      break
  if flag then
    break
  else
    w := a random vector
return w
```

But Tom found that, in some occasions, *PLA* will end up into an infinite loop, which confuses him a lot. You are required to help Tom determine, when performed on a given sample set D , if *PLA* will end up into an infinite loop. Print **Infinite loop!** if so, or **Successful!** otherwise.

We only consider cases when $d = 2$ for simplification.

Input

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

Each test case begins with a line containing a single integer $n(1 \leq n \leq 100)$, size of the set of training samples D .

Then n lines follow, the i th of which contains three integers $x_{i,1}, x_{i,2}, y_i$ ($-10^5 \leq x_{i,1}, x_{i,2} \leq 10^5, y_i \in \{-1, 1\}$), indicating the i th sample (\mathbf{x}_i, y_i) in D , where $\mathbf{x}_i = (x_{i,1}, x_{i,2})$.

Output

For each test case, output a single line containing the answer: **Infinite loop!** or **Successful!**.

Example

standard input	standard output
3	Successful!
2	Successful!
1 1 1	Infinite loop!
2 0 -1	
4	
0 0 1	
2 0 -1	
1 1 1	
1 -1 -1	
6	
0 0 1	
2 0 -1	
1 1 1	
1 -1 -1	
1 0 1	
0 1 -1	

Note

$$\text{sign}(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}$$