Problem A. Another Chess Problem

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

You are given an infinite set $S = \{(x, y) | x, y \in Z, \neg\exists i, j \in Z \ s.t. \ x = 2i + j, y = i - 2j\}$ and two pairs $(x_1, y_1), (x_2, y_2) \in S$.

You have a pair (x, y) in your right pocket which equals to (x_1, y_1) in the beginning. You can pay Y_UME a coin at a time and then change (x, y) to one element in S whose distance to (x, y) is 1. The distance between (a_1, b_1) and (a_2, b_2) is defined as $|a_1 - a_2| + |b_1 - b_2|$.

You expect to get (x_2, y_2) and minimize the number of coins given to Y_UME. Output the minimum number of coins and output the number of different ways to achieve the goal modulo 998244353.

Note that two ways differ if and only if there exsits some i such that i-th steps in the two ways are different.

Input

The first line contains one positive integer $T (T \leq 2 \times 10^5)$ denoting the number of test cases.

Each case starts with a line containing four integers $x_1, y_1, x_2, y_2 (-10^5 \leq x_1, y_1, x_2, y_2 \leq 10^5)$.

Output

For each test case, output one line containing two integers denoting the minimum number of coins and the number of different ways modulo 998244353.

Example

standard input	standard output
3 1 1 5 2 1 1 2 5 1 1 1 1	7 5 5 1 0 1

Problem B. Beauty Of Unimodal Sequence

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

You are given an array of n integers $a_1, a_2, ..., a_n$. We define that a sequence $p_1, p_2, ..., p_k (k \in [1, n])$ is beautiful if and only if these conditions are met:

- $1 \leq p_1 < p_2 < ... < p_k \leq n$.
- There exists $t (t \in [1, k])$ satisfying $a_{p_1} < a_{p_2} < ... < a_{p_t}$ and $a_{p_t} > a_{p_{t+1}} > ... > a_{p_k}$.

You need to find all the longest beautiful sequences, and output the lexicographically smallest one and the lexicographically largest one in them. Check the examples below for better understanding.

Input

There are multiple test cases.

Each case starts with a line containing a positive integer $n (n \leq 3 \times 10^5)$.

The second line contains n integers $a_1, a_2, ..., a_n (1 \leq a_i \leq 10^9)$.

It is guaranteed that the sum of Ns in all test cases is no larger than 10^6.

Output

For each test case, output two lines, the first of which depicts the lexicographically smallest one in longest beautiful sequences and the second of which depicts the lexicographically largest one in longest beautiful sequences.

Examples

standard input	standard output
7 1 4 2 5 7 4 6	1 2 4 5 6 1 3 4 5 7
3 1 2 3	1 2 3 1 2 3
3 3 2 1	1 2 3 1 2 3

Problem C. Coefficient

Input file:	standard input
Output file:	standard output
Time limit:	14 seconds
Memory limit:	256 megabytes

Given a function $f(x) = \frac{b}{c+e^{ax+d}}$, where $a \not\equiv 0 \pmod{998244353}$.

Denote x_0 as the smallest real solution of the equation: $ax + d = 0$, and note that the solution always exists.

Output the coefficient of the item $(x - x_0)^n$ in the Taylor series of $f(x)$ at $x = x_0$. The answer may be very large, so you just need to output the answer modulo 998244353.

Note that for the given n, your task is to answer q queries which share the same parameter n.

Note that it is not guaranteed that the answer could be represented as rational fraction $\frac{p}{q}$ where $gcd(p, q) = 1$, or q has no multiplicative inverse element modulo 998244353. If it can, print it as pq^{-1} modulo 998244353 which is not negative. Otherwise just print -1.

If you knew little about *gcd* in mathematic, please refer to https://en.wikipedia.org/wiki/Greatest_common_divisor.

If you knew little about *Taylor_series* in mathematic, please refer to https://en.wikipedia.org/wiki/Taylor_series.

Input

There are multiple test cases.

Each case starts with a line containing two integers n and q seperated by a space.

Next q lines in every test case will include four integers a, b, c, d per line, seperated by 3 spaces.

It is guaranteed that $\forall t \in \{a, b, c, d\}, |t| \leq 10^9$ and $n, q \in [0, 5 \times 10^4]$.

It is guaranteed that the sum of n and the sum of q in all test cases are both no larger than 3×10^5.

Output

For each query in each test case, output the only line containing just one integer denoting the answer if there would be, or -1 otherwise.

Example

standard input	standard output
0 1 1 1 1 1	499122177

Problem D. Double Tree

Input file:	standard input
Output file:	standard output
Time limit:	15 seconds
Memory limit:	1024 megabytes

You are given two trees, both with N vertices, numbered from 1 to N.

At the beginning, the weights of edges of both trees and the values of vertices of the first tree are given.

There are Q operations. In each operation, the value of a vertice of the first tree will be modified. Please output $max_{1\le u<v\le n}\{T_1.dis(u,v)+T_2.dis(u,v)+val(u)+val(v)\}$ after every operation.

Note that the symbol $T_{id}.dis(u,v)$ denotes the distance between vertice u and vertice v in the tree id.

Note that the distance between two vertices in some tree is defined as the sum of the weights in the simple path between the two vertices.

Note that the symbol $val(u)$ denotes the value of the vertice u in the first tree.

Input

There are multiple test cases.

Each case starts with a line containing two positive integers $N, Q(2 \le N \le 10^5, 1 \le Q \le 10^5)$.

The second line contains N integers $a_1, a_2, ..., a_N(1 \le a_i \le 10^9)$ denoting the values of vertices of the first tree.

Then follow $N-1$ lines. Each line contains three integers $u, v, w(1 \le u, v \le n, 1 \le w \le 10^9)$ describing the first tree. Namely, u, v, w means that there is an edge weighted w between vertice u and vertice v.

Then follow $N-1$ lines. Each line contains three integers $u, v, w(1 \le u, v \le n, 1 \le w \le 10^9)$ describing the second tree.

Then follow Q lines. Each line contains two integers $u, w(1 \le u \le n, 1 \le w \le 10^9)$, denoting the value of vertice u of the first tree will be modified to w.

It is guaranteed that the sum of Ns and the sum of Qs in all test cases are both no larger than 2×10^5.

There is only one test case which contains N and Q that are both larger than 2×10^4.

It is guaranteed that N and Q in every other test case are all no larger than 2×10^4.

Output

For each test case, output Q lines. Each line contains an integer denoting the answer.

Example

standard input	standard output
5 3 1 2 3 4 5 1 2 1 1 3 2 1 4 3 1 5 4 1 2 1 1 3 2 1 4 3 1 5 4 1 100 1 1 2 100	113 23 115

Problem E. Everything Is Generated In Equal Probability

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

One day, Y_UME got an integer N and an interesting program which is shown below:

1 an interesting program

```
1: function Subsequence(Array)
2:     result ← randomly select a subsequence of Array which could be empty in equal probability
3:     return result
4: end function
5: function CntInversionPairs(Array)
6:     return the number of inversion pairs of Array
7: end function
8: function Calculate(Array)
9:     cnt ← 0
10:    if Length(Array) > 0 then
11:        cnt ← CntInversionPairs(Array)
12:        Temp ← Subsequence(Array)
13:        cnt ← cnt + Calculate(Temp)
14:    end if
15:    return cnt
16: end function
```

Y_UME wants to play with this program. Firstly, he randomly generates an integer $n \in [1, N]$ in equal probability. And then he randomly generates a permutation of length n in equal probability. Afterwards, he runs the interesting program(function calculate()) with this permutation as a parameter and then gets a returning value. Please output the expectation of this value modulo 998244353.

A permutation of length n is an array of length n consisting of integers only $\in [1, n]$ which are pairwise different.

An inversion pair in a permutation p is a pair of indices (i, j) such that $i > j$ and $p_i < p_j$. For example, a permutation $[4, 1, 3, 2]$ contains 4 inversions: $(2, 1), (3, 1), (4, 1), (4, 3)$.

In mathematics, a subsequence is a sequence that can be derived from another sequence by deleting some or no elements without changing the order of the remaining elements. Note that empty subsequence is also a subsequence of original sequence.

Refer to `https://en.wikipedia.org/wiki/Subsequence` for better understanding.

Input

There are multiple test cases.

Each case starts with a line containing one integer $N(1 \leq N \leq 3000)$.

It is guaranteed that the sum of Ns in all test cases is no larger than 5×10^4.

Output

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

Examples

standard input	standard output
1	0
2	332748118
3	554580197

Problem F. Fantastic Magic Cube

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

You are given a positive integer N and a set of six-tuples. We define the value of a six-tuple $(l_x, r_x, l_y, r_y, l_z, r_z)$ is $\sum_{l_x \le x \le r_x, l_y \le y \le r_y, l_z \le z \le r_z,} x \oplus y \oplus z$. In the beginning, the set has only an element $(0, N-1, 0, N-1, 0, N-1)$. You can do the following steps repeatedly until the size of S equals to N^3:

- Pick a six-tuple $(l_x, r_x, l_y, r_y, l_z, r_z)(l_x < r_x \text{ or } l_y < r_y \text{ or } l_z < r_z)$ from the set.
- You can choose one element of $\{x, y, z\}$.
  * Assuming you chose x, it must be satisfied that $l_x < r_x$. Then you should pick an integer $t \in [l_x, r_x)$, erase $(l_x, r_x, l_y, r_y, l_z, r_z)$ from the set, add $(l_x, t, l_y, r_y, l_z, r_z)$ and $(t+1, r_x, l_y, r_y, l_z, r_z)$ into the set, and you will get the product of values of these two new six-tuples.
  * Assuming you chose y, it must be satisfied that $l_y < r_y$. Then you should pick an integer $t \in [l_y, r_y)$, erase $(l_x, r_x, l_y, r_y, l_z, r_z)$ from the set, add $(l_x, r_x, l_y, t, l_z, r_z)$ and $(l_x, r_x, t+1, r_y, l_z, r_z)$ into the set, and you will get the product of values of these two new six-tuples.
  * Assuming you chose z, it must be satisfied that $l_z < r_z$. Then you should pick an integer $t \in [l_z, r_z)$, erase $(l_x, r_x, l_y, r_y, l_z, r_z)$ from the set, add $(l_x, r_x, l_y, r_y, l_z, t)$ and $(l_x, r_x, l_y, r_y, t+1, r_z)$ into the set, and you will get the product of values of these two new six-tuples.

Maximize the sum of values you got and output it modulo 998244353.

Note that $\oplus$ means exclusive or, for more details refer to `https://en.wikipedia.org/wiki/Exclusive_or`.

Input

There are multiple test cases.

Each case starts with a line containing one positive integer $N(N \le 10^6)$.

We guarantee that the sum of Ns in all test cases is no larger than 3×10^6.

Output

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

Example

standard input	standard output
2	6

Problem G. Game

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

Alice and Bob are playing a game again. The game can be described as follows:

A chess board sized 3×3 is used for the game. But some cells in it have been banned before the game, which implies that no one could place a chess piece onto it. And some chess pieces have been placed on some cells, while some other cells remain empty. In a word, a cell can be in 4 states:

- "." means that it is empty.
- "O" means that it contains a white chess piece.
- "X" means that it contains a black chess piece.
- "#" means that it is banned.

There is no doubt that no two chess pieces can be placed onto the same cell whenever. Alice can operate white chess pieces and Bob can operate black chess pieces.

If it is Alice's turn, she should choose some cell (x, y) containing a white chess piece, and firstly ban the cell (x, y), and then she must choose one of the following three additional operations:

1) ban the two cells next to the cell (x, y) on the left and right.
2) ban the two cells next to the cell (x, y) which lie above and below (x, y).
3) operate both of the aforementioned two.

Note that if Alice chooses the first one, the two cells $(x, y-1)$ and $(x, y+1)$ could be not in the actually chessboard, she would just ban the cell(s) in the chessboard. But it is not meant that she cannot choose the first option. Namely, the actual effect of the three options could be the same.

If it is Bob's turn, he should choose some cell (x, y) containing a black chess piece, and firstly ban the cell (x, y), and then eat some melon seeds. Yes, actually he can do nothing additionally.

Whoever cannot operate on his turn loses, i.e. the other one is the winner of the game. As we all know, Alice and Bob always try their best to win and we can assume that they are smart enough.

The number of different states of the chessboard may be large. And for each one, we can make out who would be the winner and give some analysis before the game actually starts.

Today, Alice and Bob are playing many games simultaneously of the kind described above. Every someone's turn is to choose any game he can operate on to play. And no one can skip any his turn and do nothing.

You will be given a positive integer n, and n states of games denoting the total state of the whole game. Make out who would be the winner.

If Alice always wins, no matter who is the first player, output "*Alice*" without quotes.

If Bob always wins, no matter who is the first player, output "*Bob*" without quotes.

If the first player always wins, no matter who is the first player, output "*First*" without quotes.

If the second player always wins, no matter who is the first player, output "*Second*" without quotes.

If none of the situations would output, then output "*Others*" without quotes please.

Input

The first line contains the only integer T denoting the number of the test cases.

Each case begins with an empty line and another line containing the only integer n denoting the number of the states of the games. The following are the n states.

For each state, you should read an empty line firstly, and then a 3×3 chessboard. To be beautiful and striking, the chessboard is actually a 3×5 char-array. For each line, the chess pieces are separated by an additional character | as you can see in the following examples.

It is guaranteed that the sum of n in all test cases is no larger than 1100000.

See examples for better understanding.

Output

Output one line per testcase containing one word from the set { “Alice” , “Bob” , “First” , “Second” , “Others” }. The words’ exact meannings have been explained as above.

Example

standard input	standard output
1 2 #|X|O O|#|X #|.|X X|#|. X|#|O O|#|.	Alice

Problem H. Harmonious Army

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

Now, Bob is playing an interesting game in which he is a general of a harmonious army. There are n soldiers in this army. Each soldier should be in one of the two occupations, Mage or Warrior. There are m pairs of soldiers having combination ability. There are three kinds of combination ability. If the two soldiers in a pair are both Warriors, the army power would be increased by a. If the two soldiers in a pair are both Mages, the army power would be increased by c. Otherwise the army power would be increased by b, and $b = a/4 + c/3$, guaranteed that $4|a$ and $3|c$. Your task is to output the maximum power Bob can increase by arranging the soldiers' occupations.

Note that the symbol $a|b$ means that a divides b, e.g. , $3|12$ and $8|24$.

Input

There are multiple test cases.

Each case starts with a line containing two positive integers $n(n \leq 500)$ and $m(m \leq 10^4)$.

In the following m lines, each line contains five positive integers u, v, a, b, c $(1 \leq u, v \leq n, u \neq v, 1 \leq a, c \leq 4 \times 10^6, b = a/4 + c/3)$, denoting soldiers u and v have combination ability, guaranteed that the pair (u, v) would not appear more than once.

It is guaranteed that the sum of n in all test cases is no larger than 5×10^3, and the sum of m in all test cases is no larger than 5×10^4.

Output

For each test case, output one line containing the maximum power Bob can increase by arranging the soldiers' occupations.

Example

standard input	standard output
3 2 1 2 8 3 3 2 3 4 3 6	12

Problem I. I Love Palindrome String

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

You are given a string $S = s_1 s_2 .. s_{|S|}$ containing only lowercase English letters. For each integer $i \in [1, |S|]$, please output how many substrings $s_l s_{l+1} ... s_r$ satisfy the following conditions:

- $r - l + 1$ equals to i.
- The substring $s_l s_{l+1} ... s_r$ is a palindrome string.
- $s_l s_{l+1} ... s_{\lfloor (l+r)/2 \rfloor}$ is a palindrome string too.

$|S|$ denotes the length of string S.

A palindrome string is a sequence of characters which reads the same backward as forward, such as *madam* or *racecar* or *abba*.

Input

There are multiple test cases.

Each case starts with a line containing a string $S (1 \leq |S| \leq 3 \times 10^5)$ containing only lowercase English letters.

It is guaranteed that the sum of $|S|$s in all test cases is no larger than 4×10^6.

Output

For each test case, output one line containing $|S|$ integers. Any two adjacent integers are separated by a space.

Example

standard input	standard output
abababa	7 0 0 0 3 0 0

Problem J. Just Skip The Problem

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

Y_UME has just found a number x in his right pocket. The number is a non-negative integer ranging from 0 to $2^n - 1$ inclusively. You want to know the exact value of this number. Y_UME has super power, and he can answer several questions at the same time. You can ask him as many questions as you want. But you must ask all questions simultaneously. In the i-th question, you give him an integer y_i ranging from 0 to $2^n - 1$ inclusively, and he will answer you if $x \& y_i$ equals to y_i or not. Note that each question you ask has a index number. Namely, the questions are ordered in certain aspect. Note that Y_UME answer all questions at the same time, which implies that you could not make any decision on the remaining questions you could ask according to some results of some of the questions.

You want to get the exact value of x and then minimize the number of questions you will ask. How many different methods may you use with only minimum number of questions to get the exact value of x? You should output the number of methods modulo $10^6 + 3$.

Two methods differ if and only if they have different number of questions or there exsits some i satisfying that the i-th question of the first method is not equal to the i-th of the second one.

Input

There are multiple test cases.

Each case starts with a line containing one positive integer $n (n \leq 10^9)$.

Output

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

Example

standard input	standard output
2	2

Problem K. Keen On Everything But Triangle

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

N sticks are arranged in a row, and their lengths are $a_1, a_2, ..., a_N$.

There are Q querys. For i-th of them, you can only use sticks between l_i-th to r_i-th. Please output the maximum circumference of all the triangles that you can make with these sticks, or print -1 denoting no triangles you can make.

Input

There are multiple test cases.

Each case starts with a line containing two positive integers $N, Q(N, Q \leq 10^5)$.

The second line contains N integers, the i-th integer $a_i(1 \leq a_i \leq 10^9)$ of them showing the length of the i-th stick.

Then follow Q lines. i-th of them contains two integers $l_i, r_i(1 \leq l_i \leq r_i \leq N)$, meaning that you can only use sticks between l_i-th to r_i-th.

It is guaranteed that the sum of Ns and the sum of Qs in all test cases are both no larger than 4×10^5.

Output

For each test case, output Q lines, each containing an integer denoting the maximum circumference.

Example

standard input	standard output
5 3 2 5 6 5 2 1 3 2 4 2 5	13 16 16

Problem L. Longest Subarray

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

You are given two integers C, K and an array of N integers $a_1, a_2, ..., a_N$. It is guaranteed that the value of a_i is between 1 to C.

We define that a continuous subsequence $a_l, a_{l+1}, ..., a_r (l \le r)$ of array a is a good subarray if and only if the following condition is met:

$$\forall x \in [1, C],\ \sum_{i=l}^{r} [a_i = x] = 0 \ or \ \sum_{i=l}^{r} [a_i = x] \ge K$$

It implies that if a number appears in the subarray, it will appear no less than K times.

You should find the longest good subarray and output its length. Or you should print 0 if you cannot find any.

Input

There are multiple test cases.

Each case starts with a line containing three positive integers $N, C, K (N, C, K \le 10^5)$.

The second line contains N integer $a_1, a_2, ..., a_N (1 \le a_i \le C)$.

We guarantee that the sum of Ns, the sum of Cs and the sum of Ks in all test cases are all no larger than 5×10^5.

Output

For each test case, output one line containing an integer denoting the length of the longest good subarray.

Example

standard input	standard output
7 4 2 2 1 4 1 4 3 2	4