Nowcoder Multi-university Contest 2024 #4

Problem List

A	LCT
B	Pull the Box
C	Sort4
D	Plants vs. Zombies (Sunflower Edition)
E	String God's String
F	Good Tree
G	Horse Drinks Water
H	Yet Another Origami Problem
I	Friends
J	Zero
K	Calculate the Expression
L	Deceleration

Prepared by: Black Ice Tea

2024/7/25

Problem A. LCT

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

Note the **unusual memory limit** for this problem.

Given a rooted tree with n nodes and $n-1$ edges. The i-th edge can be described by (a_i, b_i), which represents an edge connecting node a_i and node b_i, where node a_i is the parent of node b_i.

There are $n-1$ queries. The i-th query contains an integer c_i, asking for the length of the longest simple path starting from node c_i in the graph (forest) formed by the first i edges. **It is guaranteed that there does not exist j such that $1 \le j \le i$ and $b_j = c_i$, i.e., c_i is the root of a tree in the forest.**

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^5$). The description of the test cases follows.

The first line of each test case contains an integer n ($2 \le n \le 10^6$), representing the number of nodes in the tree.

The next $n-1$ lines each contain three integers a_i, b_i, c_i ($1 \le a_i, b_i, c_i \le n, a_i \ne b_i$), representing that the parent of node b_i is a_i, and the i-th query is c_i.

It is guaranteed that:

- For each test case, the $n-1$ edges form a rooted tree.
- For each test case, for any $1 \le i \le n-1$, there does not exist j such that $1 \le j \le i$ and $b_j = c_i$.
- The sum of n over all test cases does not exceed 10^6.

Output

For each test case, output $n-1$ integers in one line. The i-th integer represents your answer to the i-th query.

Examples

standard input	standard output

6
4
3 4 1
2 3 1
1 2 1
4
3 4 3
2 1 2
3 2 3
4
1 2 1
3 4 3
2 3 1
4
2 3 1
2 4 2
1 2 1
2
1 2 1
2
2 1 2
	0 0 3
	1 1 2
	1 1 3
	0 1 2
	1
	1

2
5
1 2 3
1 3 4
3 4 1
1 5 1
15
10 14 10
5 8 5
1 3 1
4 7 10
2 5 2
1 2 1
3 4 1
9 10 9
11 13 11
5 6 1
10 12 9
8 9 1
11 15 11
7 11 1
	0 0 2 2
	1 1 1 1 2 3 3 2 1 3 2 6 1 6

Problem B. Pull the Box

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

Fallleaves is struggling with a game called "Pull the Box", and he wants to know if it is possible to win the game.

The game is played on an undirected graph with n nodes and m edges, without multiple edges or self-loops. Nodes are numbered from 1 to n. The player starts at node x and can move along the edges but cannot overlap with the box. The goal is to "pull" a box, initially located at node 1, to node n. For two **different** edges (u, v) and (v, w), where u, v, and w are **distinct**, if the box is at node u and the player is at node v, the player can move the box to node v and move himself to node w. Note that u and w cannot be the same, i.e., the player cannot move through the box.

Fallleaves wants you to write a program to determine whether the player can achieve the game's goal and, if possible, the minimum number of edges the player should move through before achieving the game's goal.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^5$). The description of the test cases follows.

The first line of each test case contains three integers n, m, x ($3 \le n \le 10^6$, $0 \le m \le 10^6$, $2 \le x \le n-1$), representing the number of nodes, the number of edges, and the starting node of the player, respectively.

The next m lines each contain two integers u, v ($1 \le u, v \le n, u \ne v$), representing an undirected edge connecting nodes u and v. It is guaranteed that there is no multiple edges or self-loops.

It is guaranteed that the sum of n over all test cases does not exceed 10^6, and the sum of m over all test cases does not exceed 10^6.

Output

For each test case, if the player can achieve the game's goal, output "Vegetable fallleaves" on the first line, and on the second line, output the minimum number of moves required to achieve the goal. Otherwise, output "Boring Game" in a single line.

Example

standard input	standard output

3
3 3 2
1 2
2 3
3 1
8 10 4
1 2
2 3
3 4
4 5
5 6
5 3
6 1
1 7
7 8
8 1
4 3 2
1 2
1 3
3 4
	Vegetable fallleaves
	2
	Vegetable fallleaves
	8
	Boring Game

Note

In sample case 2:

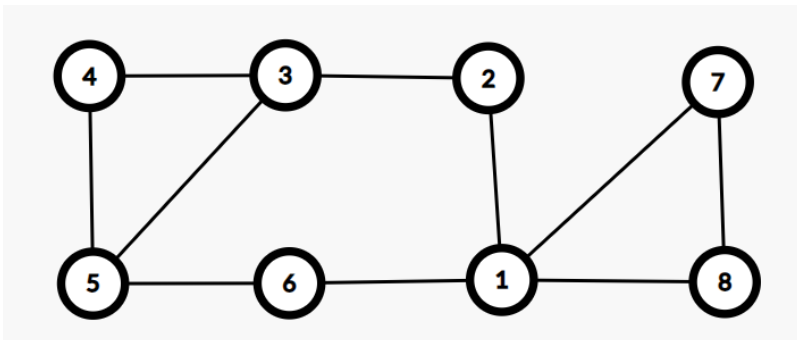

Fig. 1. Graph of the 2nd sample case.

Initially, the player is at point 4, and the box is at point 1.

The player moves along $4 \to 5 \to 6$. (2 moves in this step. 2 moves in total.)

The player pulls the box to point 6. The player goes to point 5. (1 move in this step. 3 moves in total.)

The player moves along $5 \to 3 \to 2 \to 1$. (3 moves in this step. 6 moves in total.)

The player pulls the box to point 1. The player goes to point 8. (1 move in this step. 7 moves in total.)

The player pulls the box to point 8. The player goes to point 7. (1 move in this step. 8 moves in total.)

The player moved through 8 edges in total and achieved the goal. So you output "Vegetable fallleaves" in the first line. As 8 can be shown to be the minimum moves required to achieve the goal, you output "8" in the second line.

Problem C. Sort4

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

Given a permutation[†] of length n. In one operation, you can choose **at most** four elements from the permutation and swap their positions arbitrarily. What is the minimum number of operations required to sort the permutation in ascending order?

[†] A permutation of length n is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, $[2, 3, 1, 5, 4]$ is a permutation, but $[1, 2, 2]$ is not a permutation (2 appears twice in the array), and $[1, 3, 4]$ is also not a permutation ($n = 3$ but there is 4 in the array).

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^5$). The description of the test cases follows.

The first line of each test case contains an integer n ($1 \leq n \leq 10^6$), indicating the length of the permutation.

The second line contains a permutation $a_1, a_2, \ldots, a_n$.

It is guaranteed that the sum of n across all test cases does not exceed 10^6.

Output

For each test case, output a single integer, indicating the minimum number of operations required to sort the permutation.

Example

standard input	standard output

4
4
4 2 1 3
6
4 2 5 1 3 6
8
5 4 6 3 1 8 2 7
10
1 2 9 10 3 6 8 4 5 7
	1
	1
	3
	2

Problem D. Plants vs. Zombies (Sunflower Edition)

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

Alice has recently become fascinated with Plants vs. Zombies (Sunflower Edition).

There are **two** types of sunflowers: the i-th type requires p_i sun to plant and generates q_i sun per round.

Initially, you have s sun.

At the beginning of each round, you can buy and plant **each type** of sunflower **at most once**. At the end of each round, you will get all the sun generated by your sunflowers planted in this and previous rounds.

Assuming you can plant an unlimited number of sunflowers, calculate the maximum amount of sun you can have after n rounds.

Input

The first line contains two integers s and n ($1 \le s, n \le 2 \times 10^7$), representing the initial amount of sun you have and the number of rounds the game will be played.

The second line contains two integers p_1 and q_1 ($1 \le p_1, q_1 \le 1023$), describing the attributes of the first type of sunflower.

The third line contains two integers p_2 and q_2 ($1 \le p_2, q_2 \le 1023$), describing the attributes of the second type of sunflower.

Output

Output a single integer, which is the maximum amount of sun you can have after n rounds.

Examples

standard input	standard output

75 3
50 25
25 15
	125

50 4
25 15
50 25
	125

Note

In the first sample test case, Alice starts with 75 sun and has 3 rounds to plant sunflowers. Here is the optimal strategy:

- In the first round, spend $50 + 25 = 75$ sun to plant both types of sunflowers. At the end of the round, $25 + 15 = 40$ sun would be generated.
- In the second round, spend 25 sun to plant the second type of sunflower and keep $40 - 25 = 15$ sun. At the end of the round, $25 + 15 + 15 = 55$ sun would be generated, and your sun would increase to $15 + 55 = 70$.
- In the third round, do not buy new sunflowers. After collecting 55 sun for this round, you would have $70 + 55 = 125$ sun.

Problem E. String God's String

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

One day, the String God showed Akura a very beautiful string s that contains only the characters a and b. Akura was mesmerized and wanted to replicate the same string.

Akura starts with an empty string, and he can add either an a or a b to the end of the string each time until he gets s. However, the String God's string has very powerful antimemetics, and once Akura looks away from s, he forgets all the information about it. Thus, Akura doesn't know which character to add each step and can only rely on intuition. After adding a character, he can look at s to check if he made a mistake. If he adds the wrong character, Akura will correct the mistake in the best possible way.

Specifically, let $f(x)$ be the suffix of Akura's string of length x, $g(x)$ be the prefix of s of length x, and c be the largest c such that $f(c) = g(c)$:

- If the character Akura adds increases c by 1, he added the correct character.
- Otherwise, he added the wrong character. He will then add the fewest number of characters to match $g(c)$ again, and then add the correct character to increase the match length to $c + 1$.

Once c equals the length of s, Akura will have a suffix equal to s, thus creating a copy of s.

For example, if the String God's string is $abba$ and Akura currently has ab. If Akura adds an a, which is incorrect, he will add a b to match the prefix of length 2 again, making the string $abab$, and then add a b to match the prefix of length 3.

Akura wants to know the number of possible ways to add n characters to complete this task. He wants the answers for all $n \in [0, t]$. Since the answer can be very large, you only need to provide the answer modulo $998,244,353$.

Input

The first line contains an integer t ($1 \le t \le 10^5$), representing the upper limit of n.

The second line contains a string s (the length of s does not exceed 10^5 and it contains only the characters a and b).

Output

Output $t+1$ integers in one line, where the i-th integer represents the number of ways to add $n = i-1$ characters to obtain s, modulo $998,244,353$.

Example

standard input	standard output
5 abba	0 0 0 0 1 2

Problem F. Good Tree

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

Given a tree where all edge weights are 1, define $f(u) = \sum_v dis(u, v)$, where v represents all nodes in the tree, and $dis(u, v)$ is the length of the simple path between node u and node v.

A tree is called "good" if there exist two nodes u and v such that $f(u) - f(v) = x$. Given integer x, determine the minimum number of nodes required for the tree to be "good".

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^5$). The description of the test cases follows.

Each test case contains an integer $x (1 \le x \le 10^{18})$, representing the value for which you need to determine the minimum number of nodes required for the tree to be "good".

Output

For each test case, output a single integer, representing the minimum number of nodes required for the tree to be "good".

It can be shown that the answer always exists.

Example

standard input	standard output
3 2 3 114514	4 5 678

Problem G. Horse Drinks Water

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

This problem is a variant of the General's Horse Drinking Water problem.

Given that the general's horse is at point (x_G, y_G), the tent is at point (x_T, y_T), and the river is located along the positive half of the x-axis and the positive half of the y-axis, find the shortest distance the horse needs to travel to drink water from the river and return to the tent.

Input

Each test contains multiple test cases. The first line contains the number of test cases t $(1 \le t \le 10^5)$. The description of the test cases follows.

Each test case consists of a single line containing four integers x_G, y_G, x_T, y_T $(0 \le x_G,\ y_G,\ x_T,\ y_T \le 10^9)$, describing the positions of the general's horse and the tent.

Output

For each test case, output a single integer representing the shortest distance the horse needs to travel to drink water from the river and return to the tent.

Your answer is considered correct if its absolute or relative error does not exceed 10^{-9}.

Formally, let your answer be a, and the jury's answer be b. Your answer is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-9}$.

Example

standard input	standard output

5
1 3 3 3
1 1 1 1
1 5 4 2
11 4 5 14
19 1 9 810
	4
	2
	5.8309518948
	18.8679622641
	809.4844038028

Note

In test case 3, the optimal path is shown in Fig. 2:

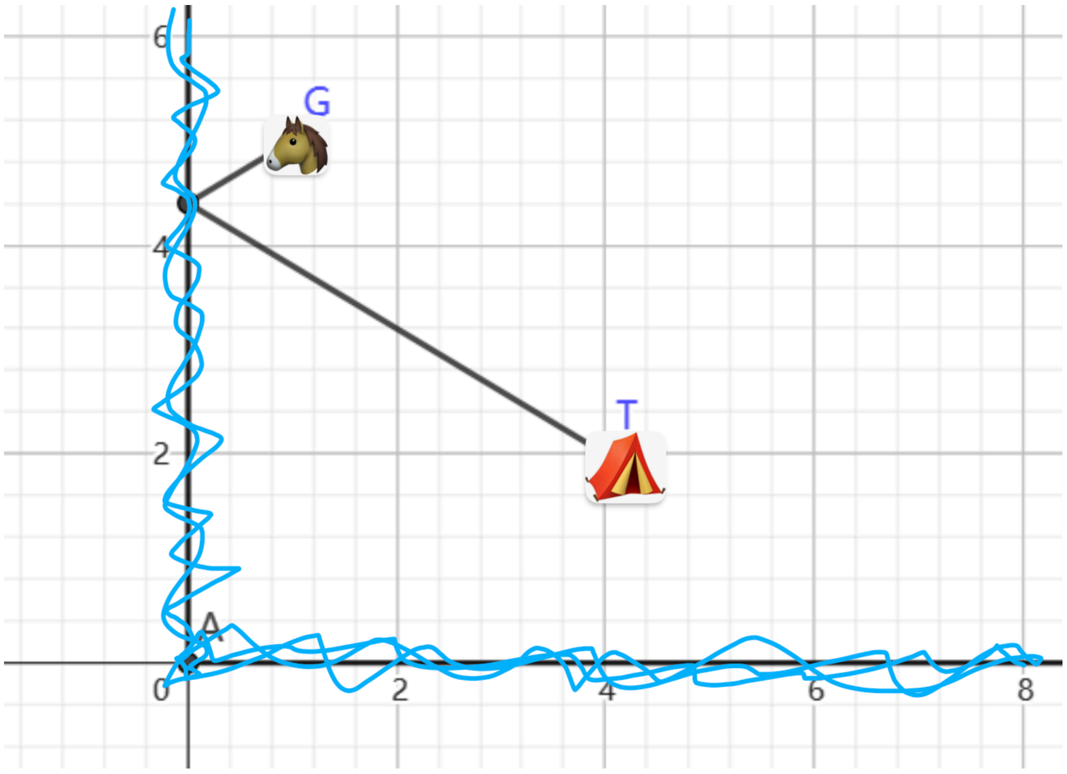

Fig. 2. The optimal path in sample case 3.

Problem H. Yet Another Origami Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Fried-chicken hates origami problems! He always fails at solving origami problems that others can solve easily (for example, CF1966E and HDU6822). However, today Fried-chicken is the problem setter! It's his turn to give contestants a very difficult origami problem!

Given an array a of length n, you can perform the following operations any number of times (possibly zero):

Choose an index p, and then perform one of the following two operations:

1. For all i such that $a_i \le a_p$, let $a_i \leftarrow a_i + 2(a_p - a_i)$.
2. For all i such that $a_i \ge a_p$, let $a_i \leftarrow a_i - 2(a_i - a_p)$.

For example, if the original array is $[2, 4, 5, 3]$ and you choose $p = 2$ and perform operation 1, the array will become $[6, 4, 5, 5]$.

Now, you want to minimize the range of the array through these operations. Recall that the range of an array is the difference between the maximum and minimum elements of the array.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 5 \times 10^5$). The description of the test cases follows.

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

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^{16}$), describing the elements of the array a in the initial state.

It is guaranteed that the sum of n over all test cases does not exceed 5×10^5.

Output

For each test case, output one integer on a single line, representing the minimum range of the array after any number of operations.

Example

standard input	standard output

3
4
2 4 5 3
3
1 2 100
1
10000
	1
	1
	0

Problem I. Friends

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 second
Memory limit:	256 megabytes

There are n people standing in a line, numbered from 1 to n from left to right. Among these people, there are m pairs of friends.

Define an interval $[l, r]$ $(1 \leq l \leq r \leq n)$ as friendly if and only if every pair of people within the interval are friends.

How many friendly intervals are there?

Input

The first line contains two integers n and m $(1 \leq n, m \leq 10^6)$, representing the number of people and the number of pairs of friends, respectively.

The next m lines each contain two integers u and v $(1 \leq u < v \leq n)$, representing a pair of friends.

It is guaranteed that there are no duplicate friend pairs in the input.

Output

Output a single integer representing the number of friendly intervals.

Example

standard input	standard output

5 9
2 3
1 3
4 5
3 5
1 2
2 4
1 5
1 4
2 5
	9

Problem J. Zero

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

Given a string s of length n, which only contains the characters 0, 1, and ?.

For any interval $[l, r]$ $(1 \le l \le r \le n)$, if the substring from the l-th to the r-th character is entirely composed of 1s, its value is $(r - l + 1)^k$; otherwise, its value is 0. The value of the string is defined as the sum of the values of all intervals.

Assuming that every ? in s is independently replaced with 0 or 1 with equal probability, what is the expected value of the string? Output the result modulo $998,244,353$.

Input

The first line contains two integers n and k $(1 \le n \le 10^5, 1 \le k \le 30)$, representing the length of the string and the parameter for calculating the value, respectively.

The second line contains a string s of length n, which consists only of the characters 0, 1, and ?.

Output

Output a single integer, the expected value of the string modulo $998,244,353$.

Formally, let $M = 998\,244\,353$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \bmod M$. In other words, output such an integer x that $0 \le x < M$ and $x \cdot q \equiv p \pmod{M}$.

Example

standard input	standard output
10 3 01??110???	873463930

Problem K. Calculate the Expression

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

As everyone knows, Mr. Chicken is a master of artificial intelligence. Unfortunately, the AI he trained, Little Chicken, is not very skilled. Especially in mathematics, Little Chicken cannot even correctly solve elementary arithmetic problems.

To improve Little Chicken's abilities, Mr. Chicken plans to provide it with a large number of arithmetic expressions for training. Due to Little Chicken's poor skills, Mr. Chicken will only give it expressions containing only addition and multiplication to solve.

However, Mr. Chicken doesn't want to manually generate a large number of expressions. Instead, he uses a lazy method. He generates a string-stored expression and then extracts a substring from it where both ends are numbers (such a substring is guaranteed to be a valid expression). He chooses this substring as the training data for Little Chicken.

Of course, always extracting substrings from a fixed expression doesn't seem very effective for training, so Mr. Chicken will occasionally modify the characters in the expression. Mr. Chicken ensures that the modified expression remains valid.

Now, Mr. Chicken needs your help to calculate the value of the expression to verify Little Chicken's answers.

Formally, Mr. Chicken provides you with a string s of length n, containing only digits, plus signs, and multiplication sign, forming a valid expression†. Mr. Chicken will also provide a sequence of operations of the following types:

- $1\ l\ r$: Queries the value of the substring from the l-th to the r-th character. The substring is guaranteed to be a valid expression.
- $2\ x\ c$: Modifies the x-th character of the original expression to c. It is guaranteed that the modified expression s remains valid.

For each query operation, output the result modulo $998,244,353$.

† A valid expression should satisfy the following conditions:

- It contains only the digits 0-9 and the operators + and *.
- No two operators are adjacent.
- Operators do not appear at the beginning or end of the expression.

Please note that under this definition, numbers **may have leading zeros**.

Input

The first line contains an integer q ($1 \le q \le 5 \times 10^5$), representing the number of operations.

The second line contains a string s ($|s| \le 5 \times 10^5$, $s_i \in \{0 \sim 9, +, *\}$).

The next q lines each contain an operation. The first number of each line is o, representing the type of operation:

- If $o = 1$, it is followed by two integers l and r $(1 \le l \le r \le n)$, representing a query.
- If $o = 2$, it is followed by an integer x and a character c $(1 \le x \le n,\ c \in \{0 \sim 9, +, *\})$, representing a modification.

It is guaranteed that all query strings and the string s at any time are valid expressions.

Output

For each query operation, output a single integer representing the result modulo $998,244,353$.

Example

standard input	standard output
5 1+2*3 1 1 5 1 3 5 2 2 4 1 1 5 1 2 5	7 6 426 126

Problem L. Deceleration

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

There is a car driving on a number line.

You need to handle the following two types of operations a total of q times:

1. At the x-th second, add a deceleration plan. After the car has traveled for x seconds, its speed will be halved. Note that there can be multiple deceleration plans at the same second. If there are n deceleration plans in one second, its speed will be reduced to $\frac{1}{2^n}$ of its original speed.
2. Assuming the car starts from position 0 at time 0 with a speed of 1 and moves in the positive direction, calculate the time it takes to reach position x.

For all operations of the second type, output the result modulo $1,000,000,007$.

Input

The first line contains a positive integer q ($1 \le q \le 5 \times 10^5$), representing the number of operations.

The next q lines each contain two integers o and x ($o \in \{1, 2\}$, $1 \le x \le 10^9$), representing the type of operation and its parameter.

Output

For all operations of the second type, output a single integer on a new line, representing the result modulo $1,000,000,007$.

Example

standard input	standard output

11
2 1
2 5
2 114514
1 5
2 4
2 6
1 3
2 4
2 5
1 114514
2 1919810
	1
	5
	114514
	4
	7
	5
	9
	15243944