Problem A. Girls Band Party

Input file:	`standard input`
Output file:	`standard output`

You are currently playing a game called "Garupa". In an event of the game, you are trying to get more event points. You have n cards, each with its own name, color, and power. When you play the game, you can put five cards of different names into your deck. The base point of this event is the sum of the power of the cards in your deck. On top of that, the event publishes a color and five names as bonus attributes, which means that each time you have a card with a bonus color in the deck, you end up with a 20% increase in event point. And each time you have a card with a bonus name in the deck, you will eventually get 10% of the event point (bonus values are calculated by addition, and we round down when calculating the final event point). Please find the maximum event point you will eventually get.

Input

The first line is an integer T ($1 \le T \le 50$), which is the number of test cases.

For each set of input data, input a positive integer n ($5 \le n \le 100000$) in the first line to indicate the number of cards you have.

The next n lines, the i-th line input two strings $name_i$, $color_i$ and a positive integer $power_i$ ($1 \le power_i \le 50000$) separated by spaces, indicating the name, color, and power of the i-th card. The input data ensures that there are at least five cards with different names.

The next line input five strings representing the five bonus names. The input data guarantees that the bonus names are different.

The last line input a string representing a bonus color.

The input data ensures that all strings consist of only uppercase and lowercase letters and the max length of them is 10, and the sum of n in all sets of input data does not exceed 1500000.

Output

For each set of data, output only one line of a positive integer, indicating the maximum number of bonus points that you will eventually get.

Example

standard input	standard output
1 6 Saaya Power 45000 Kokoro Happy 45000 Kasumi Cool 45000 Rimi Power 45000 Aya Pure 45000 Aya Power 2000 Saaya Tae Kasumi Rimi Arisa Power	382500

Problem B. So Easy

Input file: `standard input`
Output file: `standard output`

Mr. G invents a new game whose rules are given as follows.

Firstly, he has a $n \times n$ matrix, all elements of which are 0 initially. Then, he follows up with some operations: in each time he chooses a row or a column, and adds an arbitrary positive integer to all the elements in the selected row or column. When all operations have been finished, he hides an element in the matrix and the element is modified to -1.

Now given the final matrix, you are asked to find out what the hidden element should be before the very last hiding operation.

Input

The first line contains a single integer n $(2 \le n \le 1000)$.

Next n lines represent the matrix after the operations. Each element in the matrix satisfies $-1 \le a_{i,j} \le 1000000$, and exactly one element is -1.

Output

Output a single integer, the hidden element.

Example

standard input	standard output
3 1 2 1 0 -1 0 0 1 0	1

Problem C. Image Processing

Input file: `standard input`
Output file: `standard output`

Brabo has n images and an image processing APP. The i-th image, for any $1 \le i \le n$, has a contrast value v_i. To make the images better, the APP receives a batch of images together (which contains at least k images) and the contrast between these images should be as close as possible.

Brabo has already known the contrast values v_i of all these images, and now he has to determine a partition splitting images into groups so that each group has at least k images, and each image should belong to a certain group. Moreover, the maximal difference of contrast values for images in the same group should be as small as possible. Note that Brabo cannot rearrange the order of these images. That is, each group must contain several images with continuous indexes.

Let's denote c_i as the smallest maximal difference of contrast values for splitting the first i images into groups. Your task is to compute these values: $c_1, c_2, \cdots, c_n$. Note that when it is impossible to partition the first i images, c_i is regarded as 0.

Input

The first line contains two integers n ($1 \le n \le 1000000$) and k ($1 \le k \le n$) — the number of images, and each group of images should contain no less than k images.

The next line contains n integers $x_1, x_2, \cdots, x_n$ ($0 \le x_i \le 2 \times 10^9$) — the encrypted contrast v_i of these images. The actual v_i is $x_i \oplus c_{i-1}$, where $\oplus$ denotes bitwise exclusive-or. Note that $c_0 = 0$. It is guaranteed that $1 \le v_i \le 10^9$ after decryption.

Output

Output n lines, where the i-th ($1 \le i \le n$) line contains a single integer, the smallest contrast differences c_i.

Example

standard input	standard output
5 2 50 110 190 120 34	0 60 80 90 80

Note

In the sample test, $v = [50, 110, 130, 40, 120]$.

Problem D. Easy Problem

Input file: `standard input`
Output file: `standard output`

A sequence $(a_1, a_2, \cdots, a_n)$ is (n, m, d)-good if $1 \le a_i \le m$ $(1 \le i \le n)$ and $\gcd(a_1, a_2, \cdots, a_n) = d$.

Given four integers n, m, d and k, you are asked to calculate the sum of $f(q, k)$ for each (n, m, d)-good sequence q, where $f((a_1, a_2, ...a_n), k) = (a_1 a_2 \cdots a_n)^k$ for the sequence $q = (a_1, a_2, ...a_n)$.

Since the answer could be very large, you only need to output the answer modulo 59964251.

Input

The first line is an integer T $(1 \le T \le 20)$, which is the number of test cases.

For each test case, the first line contains four integers n $(1 \le n \le 10^{100000})$, m $(1 \le m \le 100000)$, d $(1 \le d \le 100000)$, k $(1 \le k \le 10^9)$, which are described in the problem description.

Output

For each test case, output a line containing a single integer.

Example

standard input	standard output
1 3 3 3 1	27

Problem E. XOR Tree

Input file: `standard input`
Output file: `standard output`

You are given a tree with n nodes labelled from 1 to n, the root of which is the node 1, and the node i has a given value a_i for each $i = 1, 2, \ldots, n$.

We define $d(x, y)$ as the number of edges in the shortest path from the node x to the node y, and define a multiset $p(x, k)$ as $\{a_y \mid y$ is in the subtree of x and $d(x, y) \le k\}$. Note that here $a_x \in p(x, k)$.

We define the score of any arbitrary set as the sum of squares of XORs of any two numbers. For example, the score of the set $\{1, 1, 2, 3\}$ should be

$$(1 \oplus 1)^2 + (1 \oplus 2)^2 + (1 \oplus 3)^2 + (1 \oplus 2)^2 + (1 \oplus 3)^2 + (2 \oplus 3)^2 = 27$$

where $\oplus$ denotes the bitwise exclusive-or.

Now you are given the parameter k. For each node x you need to compute the score of $p(x, k)$.

Input

The first line of input contains two integers n, k ($1 \le k \le n \le 100000$), the number of nodes of the tree and the parameter described above.

The second line of input contains n integers, the i-th number a_i ($1 \le a_i \le 10^9$) is the value of the i-th node.

The third line of input contains $n - 1$ integers, the i-th number f_{i+1} ($1 \le f_{i+1} \le i$) is the parent of the $(i + 1)$-th node.

Output

Output n lines, the i-th line contains a single integer, the score of $p(i, k)$. Note that the answer can be extremely large, please output it modulo 2^{64} instead.

Example

standard input	standard output
6 1 4 3 2 4 3 1 1 1 2 2 5	86 98 0 0 4 0

Problem F. Function!

Input file: `standard input`
Output file: `standard output`

Define the function

$$f_a(x) = a^x \ (a > 0 \wedge a \neq 1)$$

for all $x \in (-\infty, +\infty)$.

You are asked to calculate the value of

$$\sum_{a=2}^{n} \left(a \sum_{b=a}^{n} \lfloor f_a^{-1}(b) \rfloor \lceil f_b^{-1}(a) \rceil \right)$$

where $f_a^{-1}(x)$ is the inverse function of $f_a(x)$, $\lfloor x \rfloor$ is the largest integer that is less than or equal to x, and $\lceil x \rceil$ is the smallest integer that is greater than or equal to x.

Since the value could be very large, please output the value modulo 998244353.

Input

An integer n $(2 \le n \le 10^{12})$ described above.

Output

An integer denotes the value you have calculated modulo 998244353.

Examples

standard input	standard output
2	2
10	236

Problem G. Pot!!

Input file: `standard input`
Output file: `standard output`

Little Q is very sleepy, and he really needs some ~~impenetrable hard problems~~ coffee to make him awake. At this time, Little L brings a pot to Little Q, and he states the pot as follows.

For a prime number p, if $p^m | n$ and $p^{m+1} \nmid n$, we say $\text{pot}_p(n) = m$.

The pot is very special that it can make everyone awake immediately.

Now Little L provides n ($1 \le n \le 10^5$) integers $a_1, a_2, \cdots, a_n$ to Little Q, each of which is 1 initially. After that, Little L shows 2 types of queries:

- `MULTIPLY l r x` : For every $i \in [l, r]$ ($1 \le l \le r \le n$), multiply a_i by x ($2 \le x \le 10$).
- `MAX l r` : Calculate the value of

$$\max_{l \le i \le r} \left\{ \max_{p | a_i} \{\text{pot}_p(a_i)\} \right\} \ (1 \le l \le r \le n),$$

  where p is prime.

Now you need to perform q ($1 \le q \le 10^5$) queries of these two types of queries described above.

If you perform a "`MULTIPLY`" query, you don't need to output anything.

If you perform a "`MAX`" query, you need to output a line like "`ANSWER y`", where y the value you've calculated.

Input

The first line contains two integers n ($1 \le n \le 10^5$) and q ($1 \le q \le 10^5$), the number of integers and the number of queries.

Each of the next q lines contains one type of query described above.

Output

For each "`MAX`" query, output one line in the format of "`ANSWER y`", where y the value you have calculated.

Example

standard input	standard output
5 6 MULTIPLY 3 5 2 MULTIPLY 2 5 3 MAX 1 5 MULTIPLY 1 4 2 MULTIPLY 2 5 5 MAX 3 5	ANSWER 1 ANSWER 2

Note

If m and n are non-zero integers, or more generally, non-zero elements of an integral domain, it is said that m divides n if there exists an integer k, or an element k of the integral domain, such that $m \times k = n$, and this is written as $m \mid n$.

Problem H. Delivery Route

Input file: `standard input`
Output file: `standard output`

Pony is the boss of a courier company. The company needs to deliver packages to n offices numbered from 1 to n. Especially, the s-th office is the transfer station of the courier company.

There are x ordinary two-way roads and y one-way roads between these offices. The delivery vans will consume c_i power if they pass through the i-th road. In general, the power consumption on one road must be non-negative. However, thanks to the experimental charging rail, the consumption may be negative on some one-way roads.

Besides, Pony got the following public information. The relevant department promised that if there is a one-way street from a_i to b_i, it is impossible to return from b_i to a_i.

To avoid the delivery vans anchoring on the road, Xiaodao wants to find these lowest power consumptions from the transfer station to these offices.

Input

The first line contains four integers n $(1 \le n \le 25000), x, y$ $(1 \le x, y \le 50000)$, and s $(1 \le s \le n)$. This is followed by $x + y$ lines, each line of which contains three integer a_i, b_i $(1 \le a_i, b_i \le n, a_i \ne b_i)$ and c_i $(-10000 \le c_i \le 10000)$ describing the roads. The first x given roads are ordinary two-way roads, and the last y given roads are one-way roads.

Output

The output should contain n lines, the i-th line represents the minimum energy consumption from s-th to the i-th office if possible, or output "`NO PATH`" if it is impossible to reach the i-th office.

Example

standard input	standard output
6 3 3 4	NO PATH
1 2 5	NO PATH
3 4 5	5
5 6 10	0
3 5 -100	-95
4 6 -100	-100
1 3 -10	

Problem I. Base62

Input file: `standard input`
Output file: `standard output`

As we already know, base64 is a common binary-to-text encoding scheme. Here we define a special series of positional systems that represent numbers using a base (a.k.a. radix) of 2 to 62. The symbols '`0`' – '`9`' represent zero to nine, and '`A`' – '`Z`' represent ten to thirty-five, and '`a`' – '`z`' represent thirty-six to sixty-one. Now you need to convert some integer z in base x into base y.

Input

The input contains three integers x, y $(2 \leq x, y \leq 62)$ and z $(0 \leq z < x^{120})$, where the integer z is given in base x.

Output

Output the integer z in base y.

Example

standard input	standard output
16 2 FB	11111011

Problem J. Toad's Travel

Input file: `standard input`
Output file: `standard output`

A toad is travelling in Byteland, which consists of some cities and some roads, each of which connects a pair of cities. More specifically, the map of Byteland is an undirected connected edge-weighted graph in which every edge lies on at most one simple cycle. The toad is in the city numbered by 1 at first and wants to go through all the roads at least once.

TIME IS MONEY!

The toad must minimize the total length of the path in his travelling.

Input

The first line contains two integers n, m $(2 \le n \le 10^5, n-1 \le m \le 2n-2)$, indicates the number of cities and roads in Byteland.

Each of the next m lines contains three integers u_i, v_i, w_i $(1 \le u_i, v_i \le n, u_i \ne v_i, 0 \le w_i \le 10^5)$, representing a road with a length of w_i connects u_i and v_i. It's guaranteed that each pair of cities will be connected with at most one road.

Output

Output a single integer, indicating the minimum possible sum.

Example

standard input	standard output
6 7 1 2 1 1 3 1 2 3 1 3 4 1 3 5 1 4 5 1 2 6 1	8

Note

In the sample test, one of the best paths is

$$1 \to 2 \to 6 \to 2 \to 3 \to 4 \to 5 \to 3 \to 1$$

and the total length of the path is 8.

Problem K. Largest Common Submatrix

Input file: `standard input`
Output file: `standard output`

You are given two $n \times m$ matrices, and the elements of each matrix are ranged from 1 to $n \times m$ and pairwise distinct. You need to find the common submatrix with the largest size between these two matrices.

Example:

Matrix A:

$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 8 & 7 & 9 \end{matrix}$$

Matrix B:

$$\begin{matrix} 5 & 6 & 1 \\ 7 & 9 & 3 \\ 2 & 4 & 8 \end{matrix}$$

Largest common submatrix:

$$\begin{matrix} 5 & 6 \\ 7 & 9 \end{matrix}$$

Input

The first line of input contains two integers n ($1 \le n \le 1000$) and m ($1 \le m \le 1000$), denoting the number of rows and columns of each matrix.

Each of the next n lines contain m integers per line, denoting the first matrix $A = (a_{i,j})_{n\times m}$. And again, each of the next n lines contains m integers per line, denoting the second matrix $B = (b_{i,j})_{n\times m}$.

It is guaranteed that $1 \le a_{i,j}, b_{i,j} \le n \times m$, and $a_{i_1,j_1} \ne a_{i_2,j_2} \wedge b_{i_1,j_1} \ne b_{i_2,j_2}$ always holds for each pair of (i_1, j_1) and (i_2, j_2), where $i_1 \ne i_2 \vee j_1 \ne j_2$.

Output

Output an integer representing the size of the largest common submatrix.

Example

standard input	standard output
3 4 5 6 7 8 1 2 3 4 9 10 11 12 5 6 8 7 1 2 4 3 12 11 10 9	4

Note

Largest common submatrix in the sample test:

$$\begin{matrix} 5 & 6 \\ 1 & 2 \end{matrix}$$

Problem L. Xian Xiang

Input file: `standard input`
Output file: `standard output`

In recent days, Raven is addicted to a game called Xian Xiang.

At the beginning of this game, there are some objects in an $n \times m$ rectangle. Every object has k types of attributes. The player could delete any two objects by linking these two objects with a folding line, each part of which is either horizontal or vertical. However, the line can only change direction at most once, and there cannot be any object on its path (the rule is similar to that of Lian Lian Kan). Whenever the objects are deleted, the cells of the object will be left empty, and a score will be given according to the number of matching attributes of the two deleted objects. If there is 0 common attribute between two objects, the score will be s_0; if there is 1 common attribute, the score will be s_1, ...; and if there are p common attributes, the score will be s_p.

There is a scoreboard and Raven is eager to go to the top of the scoreboard. So he must try his best to get the highest score in the game.

Input

The first line is an integer T ($1 \leq T \leq 20$), which indicates the number of test cases.

Each case starts with three positive integers n, m, k ($1 \leq n, m \leq 7, 1 \leq k \leq 5$), representing the length, width, and the number of the attributes of the object respectively.

Next, there are $n \times m$ strings with a length of k, showing the corresponding objects. Initially, if the cell is blank, the string is assigned with k "-"; if there is an object in the cell, the string is assigned with k small-case letters. Each small-case letter represents a kind of attribute. If the letters at the same position of the string are the same for the linked two objects, it indicates that the two objects have the same attribute of that position.

Then there are $k+1$ positive integers $s_0, s_1, \cdots, s_k$ ($1 \leq s_i \leq 10000$) in the last line for every group of data.

It is guaranteed that the number of objects in each group of data is an even number and no more than 18.

Output

For each case, output an integer in a line, showing the highest possible score for the game.

Example

standard input	standard output
3 2 2 3 aaa aaa bbb bbb 1 10 100 1000 2 3 3 aaa --- bbb bbb --- aaa 1 10 100 1000 1 4 3 aaa bba abb aaa 1 10 100 1000	2000 2 1010

Note

In the first sample, delete two objects in the first row and the second row and you will score 2000 points.

In the second sample, as the same objects cannot be linked, you can only score 2 points.

In the third sample, delete two objects in the middle, and then delete two objects at the far left and far right.

Problem M. Crazy Cake

Input file: `standard input`
Output file: `standard output`

Mr. Li buys a circular strawberry cake on his birthday, and the pastry chef puts n identical strawberries around the cake evenly. Now Mr. Li wants to divide the cake following the rules below:

- Each cutting track must be a segment connecting two strawberries.
- Any two cutting tracks cannot intersect strictly inside the cake, but they may share a common strawberry.

Note that cutting tracks are allowed to connect two adjacent strawberries, and Mr. Li can also do nothing.

You are asked to help Mr. Li calculate the number of different ways in cutting the cake. Two ways are regarded as the same if they coincide after rotating the cake. Since the answer could be very large, you only need to tell him the answer modulo 1000000007.

Input

The first line contains an integer T $(1 \le T \le 10^5)$, the number of test cases.

Each of the test cases contains an integer n $(2 \le n \le 10^6)$, the number of strawberries.

Output

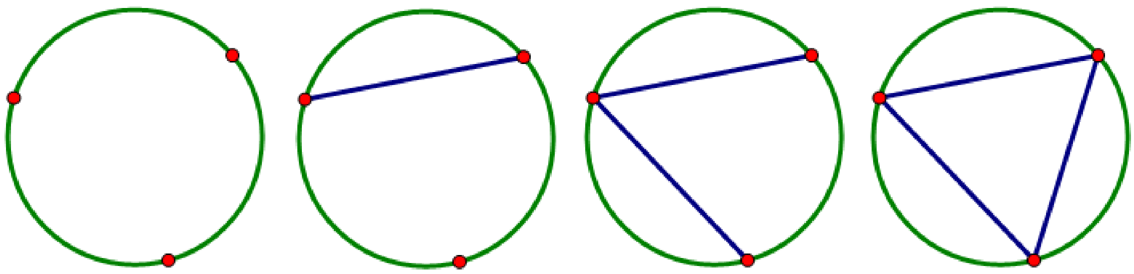
For each test case, output an integer denotes the number of different ways of cutting the cake modulo 1000000007.

Example

standard input	standard output
2 2 3	2 4

Note

The following figure shows the 4 ways of cutting the cake with 3 strawberries.



Problem N. Fibonacci Sequence

Input file: `standard input`
Output file: `standard output`

The Fibonacci sequence is a sequence of natural numbers, and is defined as follows:

- $F_1 = 1$;
- $F_2 = 1$; and
- $F_n = F_{n-1} + F_{n-2}$ for $n > 2$.

Write a program to output the first 5 numbers in the Fibonacci sequence.

Input

There is no input for this problem.

Output

Output 5 integers indicating the first 5 numbers in the Fibonacci sequence. Any two adjacent numbers in the output are separated by exactly one space and there is no extra space or symbol at the end of the line.

Example

standard input	standard output
`(no input)`	`1 1 2 3 5`