2022 Nowcoder Multi-University Training Contest Sponsored by NIO, Round 11: EZEC and QCJJ fan club

China

August, 17, 2022

牛客竞赛

AC.NOWCODER.COM

Problem A. Alternating 2.0

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

Let's call a binary string (a string where each character is either 0 or 1) x of length k an alternating string if and only if $x_i \neq x_{i+1}$ for all $i \in [1, k-1]$.

You can perform the following operation on a binary string x of length k:

- Choose a substring $x_l x_{l+1} \ldots x_r$ and invert all characters of this substring, that is, replace 0 by 1 and vice versa.

Let's call beauty of a binary string x the minimum number of operations that need to be performed on x to make it an alternating string.

Now you are given a binary string s of length n.

You need to support q queries. Each query is represented by two indices l and r, denoting that you have to calculate the sum of beauty of all $2^{r-l+1} - 1$ non-empty subsequences of $s_l s_{l+1} \ldots s_r$ modulo $10^9 + 7$.

The queries are given in a compressed format. Let l_i and r_i be the bounds for the i-th query. For $i \in [1, q]$, let $f = ((l_{i-1} \cdot a_l + b_l) \bmod n) + 1$ and $g = ((r_{i-1} \cdot a_r + b_r) \bmod n) + 1$. If $f > g$, they are swapped. Then assign f to l_i and g to r_i. The value of $a_l, b_l, a_r, b_r, l_0, r_0$ will be given in the input.

Let ans_i be the answer modulo $10^9 + 7$ to the i-th query. You need to output $\bigoplus_{i=1}^{q} (ans_i + 23 \cdot i) = (ans_1 + 23 \cdot 1) \oplus (ans_2 + 23 \cdot 2) \oplus \ldots \oplus (ans_q + 23 \cdot q)$.

Here, $\oplus$ denotes the bitwise XOR operation.

String x is a subsequence of string y if and only if x can be obtained from y by deletion of several (possibly, zero) characters.

Input

The first line contains two integers n, q ($1 \leq n, q \leq 5 \times 10^6$) — the length of the binary string s and the number of queries.

The second line contains a binary string s of length n.

The third line contains six integers $a_l, b_l, a_r, b_r, l_0, r_0$ ($0 \leq a_l, b_l, a_r, b_r, l_0, r_0 \leq 10^9$) — the parameters to generate the queries.

Output

Print a single integer — the value of $\bigoplus_{i=1}^{q} (ans_i + 23 \cdot i)$.

Example

standard input	standard output
3 2 001 0 0 1 0 1 1	40

Note

In the first query, $l_1 = 1, r_1 = 2, s_1 s_2 = 00$, all non-empty subsequences of 00 are $0, 0, 00$, beauty of which are $0, 0, 1$ respectively, with the sum of $0 + 0 + 1 = 1$.

In the second query, $l_2 = 1, r_2 = 3, s_1s_2s_3 = 001$, all non-empty subsequences of 001 are $0, 0, 1, 00, 01, 01, 001$, beauty of which are $0, 0, 0, 1, 0, 0, 1$ respectively, with the sum of $0 + 0 + 0 + 1 + 0 + 0 + 1 = 2$.

Thus, $\bigoplus_{i=1}^{q}(ans_i + 23 \cdot i) = (1 + 23 \cdot 1) \oplus (2 + 23 \cdot 2) = 40$.

Problem B. Bustling City

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Zhou Kangyang created human civilization.

In April 11451 A.D. human built n cities on Mars. For each i, there is a merchant in the city i and a **unidirectional** road from the city i to the city a_i.

At the beginning of each year, the merchant in the city i will go along the road to the city a_i.

Let's call a city start to be **bustling** in the x-th year iff the x-th year is the earliest year in which there are at least k merchants in the city.

Please tell me, in which year each city start to be bustling.

Input

The first line contains two integers n, k ($2 \le k < n \le 10^6$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$).

Output

Print n integers, the i-th of them is the number of the year in which city i start to be bustling. If a city will never be bustling, output '-1'.

Examples

standard input	standard output
3 2 1 3 2	-1 -1 -1
5 2 2 3 1 5 5	-1 -1 -1 -1 1
10 2 1 6 8 5 9 9 3 8 10 5	-1 -1 -1 -1 1 -1 -1 1 1 2

Problem C. Cmostp

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	512 megabytes

There is a string S and q questions.

For each question $[l, r]$, please print the number of distinct substrings who at least once end in the range $[l, r]$.

We called a substring's end positions are the positions it ends in the string.

Input

The first line contains two integers n, q ($1 \le n \le 5 \times 10^5, 1 \le q \le 5 \times 10^5$), indicating the length of the string S and the number of questions.

The second line contains a string S with length n containing only lowercase English letters.

In each of the next q lines, there are two integers $l, r (1 \le l \le r \le n)$ describing a question.

Output

q lines in total, each line should contain only one integer - the answer to the corresponding question.

Examples

standard input	standard output
6 6 aaaaaa 1 2 1 3 1 4 1 5 1 6 4 6	2 3 4 5 6 6
14 3 znloveloveqcjj 1 14 3 11 2 10	94 53 44

Note

The index of a string in this problem starts from 1.

A substring of S can be obtained by removing zero or more(but not all) characters from the beginning and the end of S.

Problem D. Directions

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

In this problem, we assume that the Earth is a perfect sphere.

For two positions x, y on the equator, the **west distance** $w_{x,y}$ is defined as the distance from x to y if you keep going west, the **east distance** $e_{x,y}$ is defined as the distance from x to y if you keep going east.

We called x is to the west of y if $w_{y,x} \leq e_{y,x}$, x is to the east of y if $w_{y,x} > e_{y,x}$.

A **direction matrix** D is defined as a $n \times n$ matrix satisfying these conditions:

- For each integer i such that $1 \leq i \leq n$, $D_{i,i} = \texttt{*}$.
- For each integer pair (i, j) such that $1 \leq i, j \leq n$ and $i \neq j$, $D_{i,j} \in \{\texttt{W}, \texttt{E}\}$.

A **direction matrix** D is **real** if there are n distinct positions $a_1, a_2, \cdots, a_n$ on the equator satisfying these conditions:

- All points meet the anticlockwise order when we look down at the equator from above the north pole.
- For each integer pair (i, j) such that $1 \leq i, j \leq n$ and $i \neq j$, if $D_{i,j} = \texttt{W}$, a_i is to the west of a_j, otherwise a_i is to the east of a_j.

Now you are given a **direction matrix** D', but some `W` and `E` are replaced by `?`. If you can replace each `?` by `W` or `E`, how many **real direction matrix** you can get? Print the number modulo 998244353.

Input

Each test contains multiple test cases. The first line contains the number of test cases T ($1 \leq T \leq 250$). Description of the test cases follows.

The first line contains a single integer n ($2 \leq n \leq 500$).

Then n lines follow. The i-th of these lines contains a string D'_i of length n consisting of `W`, `E`, `?`, `*`, describing the direction matrix D'.

It is guaranteed that the sum of n over all test cases does not exceed 500.

Output

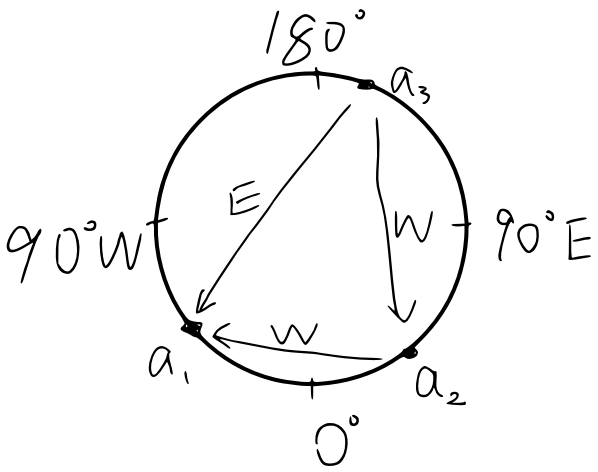
For each test case, output a single line contains a single integer, indicating the answer modulo 998244353.

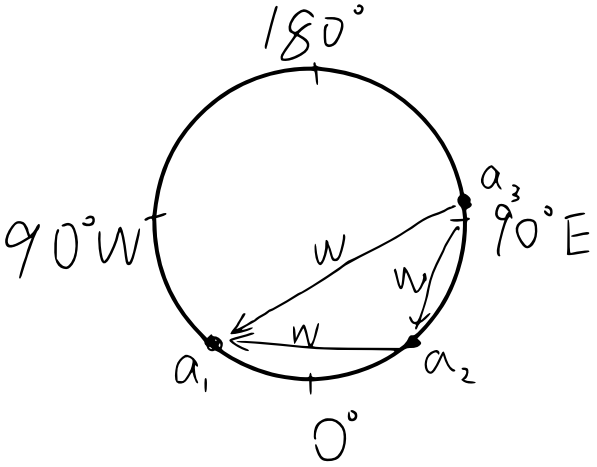
Example

standard input	standard output
3 3 *W? ?*? ?E* 4 *??? ?*?? ??*? ???* 5 *W??? ?*W?? ??*W? ???*? ????*	2 8 13

Note

Here is the explanation of the first test case when we look down at the equator from above the north pole.





Problem E. Everyone is bot

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

Have you ever used the chat application QQ? Well, in a chat group of QQ, administrators have permission to muzzle a user for some days.

There is a famous activity called "Fudu", which means repeating the sentence from the last guy sent.

As we all known, the penultimate person who do "Fudu" will be muzzled.

But Administrator Mieputrygub Bot thinks it's too boring, so he decided to muzzle **the last** p-th person who do "Fudu".

Now, there are n persons in a chat group who want to do "Fudu". If few people participate in "Fudu", it will be boring, so the number of participants n will be greater than or equal to p. If everyone does "Fudu", it is certain that existing a man who is **the last** p-th.

However, the way they do "Fudu" is unusual, there will be several rounds. In each round, n persons perform the following actions **in sequence**.

- If he did "Fudu" in the any previous rounds, he can't do "Fudu" in this round. It means each person can do "Fudu" **only once**.

- Otherwise, he can choose whether to do "Fudu".

If there is a round nobody do "Fudu", the process of doing "Fudu" will end.

For the person i, if he is the j-th person to do "Fudu", he can get $a_{i,j}$ bottles of Ice Black Tea.

However, if he was the last p-th person to do "Fudu", he will be muzzled, **not get any Ice Black Tea**. He also need to give Mieputrygub Bot 154 bottles of Ice Black Tea. It means he get -154 bottles of Ice Black Tea.

If the last p-th person does not exist, nobody will be muzzled.

Everyone wants to maximize the amount of Ice Black Tea they can get, you need to find out how much Ice Black Rea that each person can get finally.

Input

The first line contains two postive integers n, p ($1 \le p \le n \le 10^3$).

Next n lines, i-th line contains n postive integers $a_{i,1}, a_{i,2}, \ldots, a_{i,n}$ ($1 \le a_{i,j} \le 10^9$).

For $j \neq k$, $a_{i,j} \neq a_{i,k}$.

Output

Output one line with n integers, i-th integers means the number of Ice Black Tea i-th person will get.

Examples

standard input	standard output
3 2 1 14 5 141 9 1 9 8 10	1 0 0
5 3 1000 4 3 2 1 4 1000 3 2 1 4 3 1000 2 1 4 3 2 1000 1 4 3 2 1 1000	1000 1000 0 0 0

Note

For the first example:

The first person in the first round will do "Fudu" and get 1 bottle of Ice Black Tea. And the remaining two are holding each other back.(If someone first do "Fudu" in this round , then another one can do "Fudu" in the next round to let first one be the last 2-th person who do "Fudu")

Problem F. Flame blast magician master qcjj

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

Qcjj has been playing a role-playing game "Flame blast magician master" recently, in which the player will play as a hero who uses skills to attack monsters.

The world of the game can be regarded as a plane constituted by a two-dimensional coordinate of size $P \times Q$. The lower left corner is the coordinate origin $(0, 0)$.

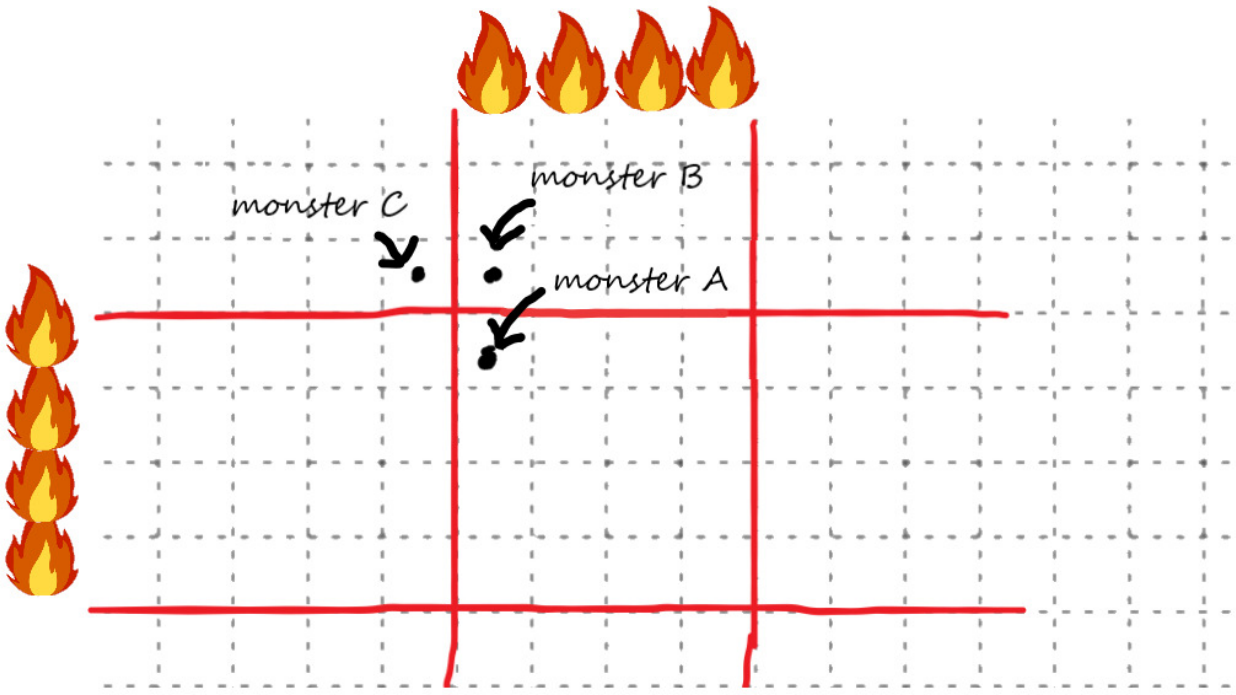
At the beginning of the game, there are N monsters. Among these monsters, the i-th monster is located in $p_i(x_i, y_i)$ with the initial HP(health point) H_i.

There may be multiple monsters in the same position on this two-dimensional axis. When the player hits the monster with a skill attack, the monster's HP will be reduced by 1

When a monster's HP $H_i \leq 0$, the monster will die immediately, and the player will receive the score V_i.

The killed monster will quit the game permanently.

Players can use the unique hero skill "Flame Blast" , generating a burning wall, perpendicular or parallel to the axis, attacking certain consecutive rows or columns.



As shown in the figure, for example, there are three monsters A, B, C in the two-dimensional coordinate axis. Assume that the initial HP of those three monsters is all equal to 2 and the killing score is all equal to 1. Qcjj uses the "Flame Blast" skill twice.

Suppose the wall created by the first use of the skill is horizontal to the x axis, Attacking monsters A, B downwards from the top infinite distance. The A, B monster HP is reduced to 1. Since none of the monsters have been killed yet, the player's *score* at this moment is 0 points.Then the second skill creating the wall perpendicular to the x axis, attack monster A from the far left infinity to the right. The skill successfully hits A, causing the hp of A to be reduced to 0, and the player gets a total of 1 points for killing the monster A. At this time, the player's *score* is 1.

The entire process of the game can be abstracted as M events. Events are divided into three types. The first two events allow the player to use the 'Flame Blast' skill to create a horizontal or vertical wall with the x axis as described earlier, which is used to attack monsters. The third event is the appearance of a monster at a specific position on the axis plane.

Now you need to write an instant game plugin that outputs the player's current score after each event.

In order to ensure the "immediacy" of your game plug-in, a part of the input data will be encrypted, forcing online decryption, the decryption method is described in "input description".

Input

The first line of input is four positive integers N, M, P, Q ($1 \le N, M, P, Q \le 10^5$), Respectively represent the initial number of monsters N, a total of M events occurred in the entire game process, and the size of the game map is $P \times Q$. The lower left corner of the game map is the coordinate origin $(0, 0)$.

Next input N lines of data.

each line of four non-negative integers x_i, y_i, H_i, V_i ($0 \le x_i < P, 0 \le y_i < Q, 1 \le H_i \le 10^5, 1 \le V_i \le 10^4$).

Respectively represent the horizontal and vertical coordinates x_i, y_i of the monster, the initial HP H_i of the monster, and the reward score V_i after killing the monster.

Next M lines, each line begins with a parameter indicating the type of event $Type_i$.

When $Type_i = 1$, it means that the player releases the wall parallel to the x coordinate axis, and attacks the monster downward from the uppermost infinite distance. At this time, it is necessary to continue to input two parameters l'_i, r'_i ($0 \le l'_i, r'_i < P$), where l'_i, r'_i represents the encrypted attack range.

The decryption formula is:

$$\begin{cases} l = \min((l' + score) \bmod P, (r' + score) \bmod P) \\ r = \max((l' + score) \bmod P, (r' + score) \bmod P) \end{cases}$$

where *score* represents the player's score before this attack.

Then all $x_i \in [l, r]$ monsters are hit by the player's skills and suffer 1 points of damage. If a monster's HP is $H_i \le 0$ at this time, the monster will be killed, the player gets a kill score of V_i.

When $Type_i = 2$, it means that the player releases the wall perpendicular to the x coordinate axis, and attacks the monster from the far left infinity to the right. At this time, it is necessary to continue to input two parameters l'_i, r'_i ($0 \le l'_i, r'_i < Q$), where l'_i, r'_i represents the encrypted attack range.

The decryption formula is:

$$\begin{cases} l = \min((l' + score) \bmod Q, (r' + score) \bmod Q) \\ r = \max((l' + score) \bmod Q, (r' + score) \bmod Q) \end{cases}$$

where *score* represents the player's score before this attack.

Then all $y_i \in [l, r]$ monsters are hit by the player's skills and suffer 1 points of damage. If a monster's HP at this time is $H_i \le 0$, then the monster is killed, the player gets a kill score of V_i.

When $Type_i = 3$, it means that a new monster has occurred. At this time, you need to continue to input four non-negative integers x_i, y_i, H_i, V_i ($0 \le x_i < P, 0 \le y_i < Q, 1 \le H_i \le 10^5, 1 \le V_i \le 10^4$).

Respectively represent the horizontal and vertical coordinates of the monster x_i, y_i, the initial HP of the monster H_i, and the reward score V_i after killing the monster.

The player's initial *score* is 0.

Output

Output M lines, each representing the player's game score *score* after each event.

Examples

standard input	standard output
3 4 15 8 5 4 2 1 5 5 2 1 4 5 2 1 1 5 8 2 1 4 3 4 5 1 1 2 4 1	0 1 1 3
12 20 10 15 3 14 2 9 9 12 1 5 8 7 1 10 7 6 4 9 0 6 4 10 1 13 5 9 5 13 3 1 0 9 2 1 5 11 3 10 1 8 2 8 3 13 3 9 9 4 1 3 1 6 3 3 0 8 2 1 1 0 4 2 5 0 1 0 9 3 0 7 3 5 1 9 7 1 1 6 1 5 6 1 1 5 2 13 1 3 0 0 2 2 2 14 11 3 0 14 1 5 2 12 6 2 9 1 1 6 6 2 7 5 1 9 0 1 0 9	0 0 9 24 24 24 32 53 56 56 56 56 57 57 57 80 80 80 95 95

Note

For the first example:

The player's initial *score* is 0.

The first event releases the wall that is parallel to the x axis, and attacks the monster downwards from the top infinite distance. Deal 1 damage.

The decrypted l and r are respectively 5 and 8 to attack the first two monsters, and the score at this time

is $score = 0$.

The second event releases the wall perpendicular to the x axis, and attacks the monster from the far left infinite distance to the right. Deal 1 damage. The decrypted l and r are respectively 1 and 4 to attack the first monster, and the score at this time is $score = 1$.

The third event adds a new monster. Note that the added new monster can be located at the same coordinates as the existing monster. At this time, the score has not changed, and it is still $score = 1$.

The fourth event releases a wall perpendicular to the x axis, dealing 1 point of damage, and attacks the monster from the far left infinite distance to the right. The decrypted l and r are respectively 2 and 5. Since the first monster has died, all the remaining monsters and the newly added monsters are attacked, and the score at this time is $score = 3$.

Note that the coordinates of the monsters may overlap.

Problem G. Good red-string

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

We define a string consisting of several disjoint *red* sequences as a "good red-string". For example, *rerded* is a good red-string , because it can be performed as **re***r***d***ed*, which can consist of 2 disjoint *red* sequences. Inversely, *redrde* is not a good red-string.

Now you are given a string with characters 'r','e','d' and '?'. You can replace '?' with 'r','e' or 'd' arbitrarily. Please determine if the string can be defined as a good red-string.

Input

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

Each test case has a string t, consisting of 'r','e','d' and '?'.

It's guaranteed that for each case $|s|$ is divisible by 3, and the sum of $|s|$ over all test cases does not exceed 3×10^5.

Output

For each test case, print a single line containing "Yes" if the given string can be defined as a good string and "No" otherwise.

You can print letters in any case (upper or lower).

Example

standard input	standard output
6 ????dd ??? ??d redder ?e? r??e?d	Yes Yes Yes No Yes Yes

Note

"r??e?d" can be a good red-string like "rreedd" or "reredd".

Problem H. Here is an Easy Problem of Zero-chan

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

Zero-chan has a rooted tree with n nodes. The root of given tree is node 1. She defines $f(x) = \prod_{i=1}^{n} lca(x, i)$.

$lca(u, v)$ meas the Least Common Ancestor of node u and node v.

Zero-chan gives you some integers x and asks you to calculate: the number of suffix zeros of $f(x)$

Input

First line contains 2 integers n, q $(1 \le q \le n \le 10^5)$ - the size of given tree and the number of queries.

Each of the next $n-1$ lines contains two integers u, v $(1 \le u, v \le n, u \ne v)$ indicating an undirected edge between node u and node v. It is guaranteed that the given edges form a tree.

The following line containing q integers describes the queries. Each of query has a integer x $(1 \le x \le n)$.

Output

For each query, print a integer - the answer of the query.

Example

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

Problem I. Innocent longing

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

In juvenile years, I used to innocently long for his company. But finally, it is only confusion left by him in my mind.

For a given permutation π of integers in $[1, n]$, we call a subsegment $\pi_l, \pi_{l+1}, \ldots, \pi_r$ an **interval** $[l, r]$ iff the difference between the maximum value and the mininum value in the subsegment equals to $r - l$.

For a given interval $[l, r]$, we call it a **non-trivial interval** iff $2 \leq r - l + 1 < n$.

A permutation π **could be in my mind**, iff there is at least one non-trivial interval whose length is greater than k.

I will give you n, k, please tell me how many permutations could be in my mind, modulo 998244353.

Input

The first line contains two integers n, k ($1 \leq k < n \leq 10^5$).

Output

Output a single line contains a single integer, indicating the answer modulo 998244353.

Examples

standard input	standard output
3 2	0
4 2	20
300 20	940881556
100000 1	123738250

Problem J. Jellyfish and its dream

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

Jellyfish can only long drift throughout their lives as if they have no thoughts of their own. But you know, how we know that jellyfish really have no dreams, right?

— S4M, *Jellyfish*

You are given a 0-indexed array a of length n where $a_i \in \{0, 1, 2\}$.

You can perform the operation below:

- Choose an index i such that $(a_i + 1) \bmod 3 = a_{(i+1) \bmod n}$, then let $a_i \leftarrow (a_i + 1) \bmod 3$.

Jellyfish's dream is to **make all elements equal**. Check if its dream could be achieved after performing some (possibly zero) operations.

Input

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

The first line contains a single integer n ($2 \le n \le 10^6$).

The second line contains n integers $a_0, a_1, \ldots, a_{n-1}$ ($0 \le a_i < 3$).

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

Output

For each test case, output one line. If all element can be equal after some (possibly zero) operations, print "Yes", otherwise print "No".

You can print letters in any case (upper or lower).

Example

standard input	standard output
5 2 1 0 3 0 2 1 4 0 1 2 2 5 0 2 1 0 1 6 2 0 2 1 0 1	Yes No Yes No Yes

Problem K. Killer Sajin's Matrix

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

Sajin gives you 3 integers n, m, k, and gives you a task:

Find any 01-matrix a with n rows and m columns where sum of each row and each column is odd. Among all of $n \times m$ elements, there are exactly k elements have value 1 and $n \times m - k$ elements have value 0.

Input

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

Each test case has 3 integers n, m, k ($1 \le n, m, k \le 10^5$) - the height and width of the 01-matrix, the number of elements with value 1, respectively.

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

Output

If the matrix is not existed, print a single line containing "No".

Otherwise print a line containing "Yes" and the next each k line print 2 integers i, j, which means $a_{i,j} = 1$.

If there are multiple answers, print any.

You can print letters in any case (upper or lower).

Example

standard input	standard output
4 2 2 1 2 2 2 3 3 3 4 4 10	No Yes 1 2 2 1 Yes 1 3 2 2 3 1 Yes 1 3 1 2 1 1 2 3 2 2 2 1 3 4 3 3 3 2 4 1

Problem L. Lndjy and the mex

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

lndjy has his own unique interest in numbers. Today he came up with a multiset S consists of n non-negative integers which are not greater than n. More precisely, he has given a sequence of non-integers $a_0, a_1, \ldots, a_n$, representing that for each integer $0 \le i \le n$, the number of elements in S equal to i is exactly a_i. It is guaranteed that $\sum_{i=0}^{n} a_i = n$.

lndjy also showed his great interest in Combinatorics. So he wants to make sequences with S. Let the set of all different sequences which has exactly the same elements as S be A. It's easy to prove that $|A| = \frac{n!}{a_0!a_1!\cdots a_n!}$.

What's more, lndjy loves subsegments and the mex. He defined a function $f_b(l, r)$ on all subsegments of any given sequence $b \in A$ as

$$f_b(l, r) = \operatorname{mex}\{b_l, b_{l+1}, \ldots, b_r\}$$

where $\operatorname{mex} X$ is the smallest non-negative integer which is not present in the multiset X. For example, $\operatorname{mex}\{0, 1, 1, 5, 4\} = 2$, $\operatorname{mex}\{1, 2, 3\} = 0$ and $\operatorname{mex} \varnothing = 0$.

Finally, he wants you to calculate the value of

$$\sum_{b \in A} \sum_{1 \le l \le r \le n} f_b(l, r) \bmod 998244353$$

Input

The first line contains one integer n ($1 \le n \le 10^5$).

The second line contains $n + 1$ integers $a_0, a_1, \ldots, a_n$ ($0 \le a_i \le n, \sum_{i=0}^{n} a_i = n$).

Output

Print a single integer, equals to $\sum_{b \in A} \sum_{1 \le l \le r \le n} f_b(l, r) \bmod 998244353$.

Examples

standard input	standard output
1 0 1	0
2 2 0 0	3
5 2 2 1 0 0 0	639
10 4 1 2 1 2 0 0 0 0 0 0	3495060

Problem M. Maimai DX 2077

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

dXqwq likes playing *maimai DX UNiVERSE PLUS*, since he can't fly to Japan and SEGA completely ignores the feelings of Chinese players, he can only play *maimai DX 2077*.

In *maimai DX 2077*, you need to press the button or touch the screen when NOTE appears. There are 4 types of NOTE: TAP, HOLD, SLIDE, and BREAK. There are also 5 judgments for each NOTE: CRITICAL PERFECT, PERFECT, GREAT, GOOD, and MISS.

For each type of NOTE, you can get some **standard points** according to the judgment. For BREAK NOTEs, you can get some **extra points** according to the judgment.

Here is the table of standard points:

NOTE Type	C. PERFECT	PERFECT	GREAT	GOOD	MISS
TAP	1		0.8	0.5	0
HOLD	2		1.6	1.0	0
SLIDE	3		2.4	1.5	0
BREAK	5		2.5	2	0

Here is the table of extra points:

NOTE Type	C. PERFECT	PERFECT	GREAT	GOOD	MISS
BREAK	1	0.5	0.4	0.3	0

Let A, B be the standard points and extra points you will have if you get CRITICAL PERFECT on every NOTE, and A_0, B_0 be the standard points and extra points you have, the **achievement score** for this track is defined as $\frac{A_0}{A} \cdot 100\% + \frac{B_0}{B} \cdot 1\%$.

You are given the number of NOTEs for each type and each judgment, and calculate the achievement score.

Input

There are 4 lines in a test, each line denotes a type of NOTE: TAP, HOLD, SLIDE and BREAK.

The i-th line contains 5 integers $c_{i,0}, c_{i,1}, \cdots, c_{i,4}$ ($0 \le c_{i,j} \le 10^3$), each integer denotes a type of judgment: CRITICAL PERFECT, PERFECT, GREAT, GOOD, and MISS.

It's guaranteed that there is at least one BREAK note.

Output

Print the **percentage** of the achievement score in a line. Your answer will be considered correct if the absolute or relative error between yours and the standard answer is no more than 10^{-6}. Formally, let your answer be a and the jury's answer be b, and then your answer is considered correct if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-6}$.

Examples

standard input	standard output
311 131 24 1 2 48 20 4 0 0 36 0 0 1 0 35 15 1 0 0	99.523505378
224 133 15 0 0 45 14 0 0 0 57 0 2 1 0 15 16 0 0 0	100.051026393
324 210 26 2 2 13 14 1 0 0 102 0 3 3 0 9 4 0 0 0	99.369444233

Note

Noticed that the rules of *maimai DX 2077* are different from *maimai DX 2022*.