The 3rd Universal Cup

Stage 39: Tokyo

June 7-8, 2025

This problem set should contain 17 problems (A to L) on 34 numbered pages.

Problem A. Array Similarity

Time limit: 2 seconds
Memory limit: 1024 megabytes

Let $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n)$ be two sequences of equal length. We say a and b are *similar* if and only if for every $i = 1, 2, \ldots, n$,

$$a_i = \max(a_1, a_2, \ldots, a_i) \quad \text{holds exactly when} \quad b_i = \max(b_1, b_2, \ldots, b_i).$$

You are given a sequence $(A_1, A_2, \ldots, A_N)$. Answer Q queries. In the i-th query you are given integers $L_{i,1}, R_{i,1}, L_{i,2}, R_{i,2}$. Determine whether the two subsequences

$$(A_{L_{i,1}}, A_{L_{i,1}+1}, \ldots, A_{R_{i,1}}) \quad \text{and} \quad (A_{L_{i,2}}, A_{L_{i,2}+1}, \ldots, A_{R_{i,2}})$$

are similar.

Input

The input is given in the following format:

$$\begin{array}{l} N \;\; Q \\ A_1 \;\; A_2 \;\; \ldots \;\; A_N \\ L_{1,1} \;\; R_{1,1} \;\; L_{1,2} \;\; R_{1,2} \\ L_{2,1} \;\; R_{2,1} \;\; L_{2,2} \;\; R_{2,2} \\ \vdots \\ L_{Q,1} \;\; R_{Q,1} \;\; L_{Q,2} \;\; R_{Q,2} \end{array}$$

- All input values are integers.
- $1 \le N \le 2 \times 10^5$.
- $1 \le Q \le 2 \times 10^5$.
- $1 \le A_i \le 10^9$.
- $1 \le L_{i,1} \le R_{i,1} \le N$.
- $1 \le L_{i,2} \le R_{i,2} \le N$.
- $R_{i,1} - L_{i,1} = R_{i,2} - L_{i,2}$.

Output

Print Q lines. On the i-th line, print "`Yes`" if the subsequences

$$(A_{L_{i,1}}, A_{L_{i,1}+1}, \ldots, A_{R_{i,1}}) \quad \text{and} \quad (A_{L_{i,2}}, A_{L_{i,2}+1}, \ldots, A_{R_{i,2}})$$

are similar; otherwise, print "`No`".

Example

standard input	standard output
10 6 3 1 4 1 5 9 2 6 5 3 1 3 3 5 1 5 6 10 1 1 9 9 1 9 1 9 1 3 6 8 5 8 7 10	Yes No Yes Yes No Yes

Note

In the first query, $(3, 1, 4)$ and $(4, 1, 5)$ are similar, so the output is "`Yes`".

In the second query, $(3, 1, 4, 1, 5)$ and $(9, 2, 6, 5, 3)$ are not similar, so the output is "`No`".

In the third query, note that it is possible to have $L_{i,1} = R_{i,1}$ and $L_{i,2} = R_{i,2}$.

In the fourth query, note that it is possible to have $L_{i,1} = L_{i,2}$ and $R_{i,1} = R_{i,2}$.

Problem B. Bracket Character Frequency

Time limit: 2 seconds
Memory limit: 1024 megabytes

A string S consisting of the characters only `(` and `)` is called **correct parenthesis sequence** if and only if it satisfies any of the following conditions.

- S is an empty string.
- S is formed by concatenating `(`, A, `)` in this order where A is a correct parenthesis sequence.
- S is formed by concatenating A and B in this order where A and B are correct parenthesis sequences both of which are not empty strings.

You are given integers N, K and an integer sequence of length $2K$, $A = (A_1, A_2, \cdots, A_{2K})$.

Determine if there is a tuple of N correct parenthesis sequences which satisfies the following conditions.

- The lengths of the N correct parenthesis sequences are all $2K$.
- For $i = 1, 2, \cdots, 2K$, among the N correct parenthesis sequences there are exactly A_i which the i-th character is `(`.

You are given T test cases. Answer each test case separately.

Input

The input is given in the following format:

T
case_1
case_2
$\vdots$
case_T

Each test case is given in the following format:

N K
A_1 A_2 $\cdots$ A_{2K}

- All input values are integers.
- $1 \le T \le 10^5$.
- $1 \le N \le 10^{12}$.
- $1 \le K \le 2 \times 10^5$.
- $0 \le A_i \le N$.
- Over all test cases in a single input, the sum of K is at most 5×10^5.

Output

Print T lines. The i-th line should contain the answer for the i-th test case. In detail, print `Yes` if there is a tuple of N correct parenthesis sequences which satisfies the conditions; otherwise, print `No`.

Example

standard input	standard output
2 3 3 3 2 2 0 2 0 3 3 3 0 2 3 1 0	Yes No

Note

In the first test case, a tuple of 3 correct parenthesis sequences `()()()`, `((()))`, `(())()` satisfies the conditions. In the second test case, there is no tuple of 3 correct parenthesis sequences which satisfies the conditions.

Problem C. Card Deck

Time limit: 2 seconds
Memory limit: 1024 megabytes

There are 10^{100} cards numbered from 1 to 10^{100}, stacked so that card i is the i-th card from the top. There is one empty bag. You will perform the following operation exactly M times:

> Look at the top K cards, choose any number of them (possibly zero) and put those chosen cards into the bag. Return the unchosen cards to the top of the deck in their original relative order.

After all M operations, consider every possible set of cards that could be in the bag. Compute the sum of the sizes of those sets, then output that sum modulo 998244353.

You are given T test cases. For each test case, output the required value.

Input

The input is given in the following format:

```
T
case_1
⋮
case_T
```

Each test case is given as:

```
K M
```

- All input values are integers.
- $1 \leq T \leq 10^5$.
- $1 \leq K < 998244353$.
- $1 \leq M < 998244353$.

Output

Output T lines. On the i-th line, print the answer for the i-th test case.

Example

standard input	standard output
3 2 1 3 2 20250308 410338673	4 81 509595821

Note

In the first example, the possible sets of cards in the bag are $\{\}, \{1\}, \{2\}, \{1, 2\}$, and the sum of their sizes is 4.

Problem D. Digits of Prefix Product

Time limit: 2 seconds
Memory limit: 1024 megabytes

This is an **output-only** problem. No input is provided.

Output a sequence of 10 positive integers $(a_1, a_2, \ldots, a_{10})$ that satisfies **all** of the following conditions. All integers should be represented in standard decimal notation without leading zeros.

- Each of $a_1, a_2, \ldots, a_{10}$ contains no digit equal to 0
- Each of $a_1, a_2, \ldots, a_{10}$ has at least 100 digits
- The total number of digits in $a_1, a_2, \ldots, a_{10}$ is at most 10^5
- For each i $(1 \leq i \leq 10)$, let $b_i = a_1 \times a_2 \times \cdots \times a_i$. Then, in each b_i, every pair of adjacent digits must be different (no two adjacent digits are the same)

Input

No input is provided.

Output

Output the sequence $a_1, a_2, \ldots, a_{10}$ in this order, one number per line, using decimal notation without leading zeros.

Note

Defining $(a_1, a_2, \ldots, a_{10})$ as $(a_1, a_2, \ldots, a_{10}) = (28, 19, 2, 19, 15, 3, 14, 14, 29, 27)$ results in:

- $b_1 = 28$
- $b_2 = 532$
- $b_3 = 1064$
- $b_4 = 20216$
- $b_5 = 303240$
- $b_6 = 909720$
- $b_7 = 12736080$
- $b_8 = 178305120$
- $b_9 = 5170848480$
- $b_{10} = 139612908960$

This output satisfies conditions 1, 3, and 4. However, it does **not** satisfy condition 2, so it is judged as incorrect.

Problem E. Edge Coloring Problem

Time limit: 3 seconds
Memory limit: 1024 megabytes

You are given a simple undirected graph with N vertices and $\frac{N(N-3)}{2}$ edges. The graph is represented by N binary strings $S_1, S_2, \ldots, S_N$, where the j-th character of S_i is `1` if there is an edge between vertex i and vertex j, and `0` otherwise. Notably, the i-th character of S_i is always `0`.

The degree of each vertex in the graph is exactly $N-3$.

Now, you need to assign a positive integer to each edge of the graph. This assignment is called **edge coloring** if any two edges sharing a common vertex are assigned distinct integers. The smallest possible maximum integer used in any valid edge coloring is called the **edge chromatic number** of the graph.

Your task is to determine the edge chromatic number of the graph and also find one valid edge coloring that achieves this number.

Input

The input is given in the following format:

$$\begin{array}{l} N \\ S_1 \\ S_2 \\ \vdots \\ S_N \end{array}$$

- $4 \le N \le 300$.
- $S_1, S_2, \ldots, S_N$ are binary strings of length N containing only `0` and `1`.
- The given graph is a simple undirected graph where each vertex has a degree of $N-3$.

Output

Print the edge chromatic number C, followed by a $N \times N$ grid where the cell (i, j) contains the integer $c_{i,j}$ assigned to the edge between vertex i and vertex j. The format should be as follows:

$$\begin{array}{l} C \\ c_{1,1}\ c_{1,2}\ \cdots\ c_{1,N} \\ c_{2,1}\ c_{2,2}\ \cdots\ c_{2,N} \\ \vdots \\ c_{N,1}\ c_{N,2}\ \cdots\ c_{N,N} \end{array}$$

If there is no edge between vertex i and vertex j, output -1 for $c_{i,j}$. In particular, $c_{i,i}$ should always be -1.

If multiple valid outputs exist, any of them are considered correct.

Examples

standard input	standard output
6 011100 101010 110001 100011 010101 001110	3 -1 2 3 1 -1 -1 2 -1 1 -1 3 -1 3 1 -1 -1 -1 2 1 -1 -1 -1 2 3 -1 3 -1 2 -1 1 -1 -1 2 3 1 -1
5 01001 10100 01010 00101 10010	3 -1 2 -1 -1 1 2 -1 3 -1 -1 -1 3 -1 1 -1 -1 -1 1 -1 3 1 -1 -1 3 -1

Note

In the first example, vertex 1 is connected to vertices $2, 3$, and 4. These edges must be assigned distinct integers, so the edge chromatic number is at least 3.

In the example output, the edges connecting vertex 1 to vertices $2, 3$, and 4 are assigned integers $2, 3$, and 1, respectively. All edges sharing a common vertex have distinct integers. The same property holds for all other vertices, satisfying the edge coloring condition, with an edge chromatic number of 3.

Problem F. Fourier Coefficients

Time limit: 8 seconds
Memory limit: 1024 megabytes

This is an **interactive problem**. Your program will interact with the judge via standard input and output. The judge's execution may take up to 1.3 seconds.

You are given an integer N. The judge secretly chooses a function

$$f(x) := \sum_{k=0}^{N-1} A_k \cos(kx),$$

where each $A_0, A_1, \dots, A_{N-1}$ is an integer with $0 \le A_k < 998244353$.

You must determine the values of $A_0, A_1, \dots, A_{N-1}$ by interacting as described below:

You will output N pairs of integers $(X_1, Y_1), \dots, (X_N, Y_N)$. Each pair must satisfy

$$0 \le X_i \le Y_i < 998244353, \quad Y_i \ne 0.$$

The judge will then respond with N integers $Z_1, \dots, Z_N$, where

$$Z_i \;=\; f\big(\arccos(X_i/Y_i)\big) \bmod 998244353.$$

Detailed definition of Z_i. Under the constraints on X_i, Y_i, the value $f(\arccos(X_i/Y_i))$ is a rational number. Write it in lowest terms as P_i/Q_i; one can show $Q_i \not\equiv 0 \pmod{998244353}$. Then Z_i is defined to be the unique integer $0 \le Z_i < 998244353$ satisfying

$$Z_i\, Q_i \;\equiv\; P_i \pmod{998244353}.$$

Such a Z_i always exists and is unique.

Input

- All inputs are integers.
- $1 \le N \le 5 \times 10^5$.

Interaction Protocol

This is an interactive problem. Your program will interact with the judge via standard input and output.

First, read the integer N from standard input:

```
N
```

Then output N query pairs (X_i, Y_i) in the following format, satisfying the constraints above:

$$\begin{array}{l} X_1\ Y_1 \\ X_2\ Y_2 \\ \vdots \\ X_N\ Y_N \end{array}$$

If your output is valid, the judge will reply with N lines:

$$\begin{array}{l} Z_1 \\ Z_2 \\ \vdots \\ Z_N \end{array}$$

If your output is invalid, you will receive:

```
-1
```

If you receive "-1", your program must terminate immediately.

Finally, after receiving the Z_i, output the hidden coefficients $A_0, A_1, \ldots, A_{N-1}$ in order:

$$
\begin{array}{l}
A_0 \\
A_1 \\
\vdots \\
A_{N-1}
\end{array}
$$

Note

- **After every output operation, print a newline and flush standard output.** If you fail to flush, you may receive a TLE verdict.
- If you produce an invalid output at any point or your program terminates unexpectedly, the verdict is undefined.
- Immediately terminate your program after printing the answer (or after reading "-1"). Otherwise, the verdict is undefined.
- Extra newlines or any deviation from the specified format will be judged as invalid.
- The judge is non-adaptive: the values $A_0, \ldots, A_{N-1}$ are fixed at the start and do not change during the interaction.

Example

Suppose $N = 2$ and $(A_0, A_1) = (3, 2)$. A possible interaction is shown below.

Input	Your Output	Explanation
2		You read N.
	0 1 1 1	You query two valid pairs (X_i, Y_i).
3 5		The judge returns $Z_1 = 3$, $Z_2 = 5$.
	3 2	You output the recovered (A_0, A_1).

Problem G. Guarding Plan

Time limit: 2 seconds
Memory limit: 1024 megabytes

There are N security guards standing on a 2D coordinate plane. The i-th guard is standing at the point (x_i, y_i).

You may perform the following operation any number of times (including zero):

> Choose two points where security guards are currently standing, and select any point on the line segment connecting those two points. If no guard is already standing at the selected point, place a new guard there.

A guard standing at point (a, b) monitors all guards located in the region where both the x-coordinate is less than or equal to a and the y-coordinate is less than or equal to b .

A guard who is not being monitored by any other guard is called a **necessary guard**.

Determine the minimum possible number of necessary guards in the final configuration, and also the minimum number of operations required to achieve that configuration.

Input

The input is given in the following format:

$$\begin{array}{l} N \\ x_1\ y_1 \\ x_2\ y_2 \\ \vdots \\ x_N\ y_N \end{array}$$

- All input values are integers.
- $1 \leq N \leq 2 \times 10^5$.
- $0 \leq x_i, y_i \leq 10^9$.
- $(x_i, y_i) \neq (x_j, y_j) (i \neq j)$.

Output

In the first line, print a single integer — the minimum possible number of necessary guards. In the second line, print a single integer — the minimum number of operations required to achieve that configuration.

Examples

standard input	standard output
5 1 6 2 4 3 3 4 2 6 1	4 1
3 0 0 1 2 2 1	2 0
7 10 49 9 27 59 8 19 22 0 50 25 23 33 13	4 1

Note

In the first example, select the point (3,4), which lies between the guards at points (1,6) and (6,1), and place a new guard there. After this operation, the necessary guards will be the four stationed at (1,6), (3,4), (4,2), and (6,1).

Problem H. Hidden Sequence Rotation

Time limit: 2 seconds
Memory limit: 1024 megabytes

This is an **interactive problem** (a problem where your program interacts with the judge via standard input/output).

You are given an integer N. The judge holds a hidden sequence $A = (A_0, \ldots, A_{N-1})$ of length N, where each element is an integer between 1 and 10^5. Note that throughout the problem, indices are 0-based.

For integers $s = 0, \ldots, N-1$ and $l = 1, \ldots, N$, define the sequence $A(s, l)$ as follows:

- A sequence of length l whose i-th element is $A_{(s+i) \bmod N}$ for $i = 0, \ldots, l-1$.

You can make up to **20 queries** to the judge in the following format:

- You output a list of integer pairs $((s_0, l_0), \ldots, (s_{k-1}, l_{k-1}))$ that satisfies the following constraints:
  - $1 \le k \le N$
  - $0 \le s_i \le N-1$
  - $1 \le l_i \le N$
  - $\sum_{i=0}^{k-1} l_i \le N$
- In response, the judge returns all indices $i = 0, \ldots, k-1$ such that $A(s_i, l_i)$ is lexicographically minimal among the sequences. In other words, the judge returns the set $\{i \mid 0 \le i < k,\ A(s_i, l_i) = \min_{0 \le i' < k} A(s_{i'}, l_{i'})\}$.

Using these queries, determine all values of $s = 0, \ldots, N-1$ such that $A(s, N)$ is lexicographically minimal. In other words, identify the set $\{s \mid 0 \le s < N,\ A(s, N) = \min_{0 \le s' < N} A(s', N)\}$.

Note that the judge is **not adaptive**, meaning the sequence A is fixed before the interaction for each test case.

Input

The input is given in the following format:

N

- N is an integer in the range $1 \le N \le 10^5$.

Output

Once you have determined the answer, output it in the following format:

! n
s_0
s_1
$\vdots$
s_{n-1}

Here, n is an integer, each s_i is a distinct integer in the range $0 \le s_i < N$, and it must be satisfied that

$$\{s_0, \ldots, s_{n-1}\} = \{s \mid 0 \le s < N,\ A(s, N) = \min_{0 \le s' < N} A(s', N)\}.$$

Interaction Protocol

You may issue queries by outputting them to standard output in the following format:

$$\begin{array}{l} ?\ k \\ s_0\ l_0 \\ s_1\ l_1 \\ \vdots \\ s_{k-1}\ l_{k-1} \end{array}$$

Ensure that your queries satisfy the conditions described above.

If the query is valid, the judge will respond with:

$$\begin{array}{l} k' \\ i_0 \\ i_1 \\ \vdots \\ i_{k'-1} \end{array}$$

Here, it holds that $k', i_0, i_1, \ldots, i_{k'-1}$ are integers, $0 \le i_0 < i_1 < \cdots < i_{k'-1} < k$, and

$$\{i_0, i_1, \ldots, i_{k'-1}\} = \{i \mid 0 \le i < k,\ A(s_i, l_i) = \min_{0 \le i' < k} A(s_{i'}, l_{i'})\}.$$

If your query is invalid (for example, if it violates the constraints or exceeds the allowed number of queries), the judge will respond with:

```
-1
```

If `-1` is received, terminate your program immediately.

Notes for Interaction

- Be sure to flush the output after every print. Failure to do so may result in a Time Limit Exceeded (TLE) verdict.
- If your program produces invalid output or exits improperly during interaction, the judge's behavior is undefined.
- After printing the final answer or receiving `-1`, your program must terminate immediately. Failure to do so may result in undefined behavior.
- Avoid unnecessary newlines or spaces in your output, as these may be treated as formatting errors.

Note

The following is an example interaction for $N = 6$, with the hidden sequence $A = (1, 2, 3, 1, 2, 4)$:

Input	**Output**	**Description**
`6`		N is given.

	`? 3`
	`0 1`
	`1 1`
	`3 1`
		Querying sequences (1), (2), and (1).

`2`
`0`
`2`
		Sequences at index 0 and 2 are lexicographically smallest.

	`? 2`
	`0 3`
	`3 3`
		Querying sequences $(1, 2, 3)$ and $(1, 2, 4)$.

`1` `0`		Sequence at index 0 is lexicographically smallest.
	`! 1` `0`	Only $s = 0$ lexicographically minimizes the sequence $A(s, N)$, so that is printed as output.

Problem I. Insert AB or BA

Time limit: 2 seconds
Memory limit: 1024 megabytes

You are given two strings S and T, both consisting of the characters `A` and `B`.

You can perform the following two types of operations any number of times (including zero), in any order:

- Insert the string `AB` at any position in S. This operation costs X.
- Insert the string `BA` at any position in S. This operation costs Y.

Note that insertions can be made at the beginning or end of the string as well.

Determine whether it is possible to transform S into T using these operations. If it is possible, output the minimum total cost required to do so.

Input

The input is given in the following format:

```
S T
X Y
```

- X and Y are integers.
- S and T consist only of the characters `A` and `B`.
- $1 \le |S| \le |T| \le 8000$.
- $1 \le X \le 10^9$.
- $1 \le Y \le 10^9$.

Output

If it is possible to transform S into T, output the minimum total cost required on a single line. If it is not possible, output `-1`.

Examples

standard input	standard output
AB ABAABB 5 3	8
AAAAAA AAAAAA 2 3	0
AAAAA BBBBBBB 9982 44353	-1
AAABBABABBBBBBABBABBA AAABABBABABBABBBBBABBBBBAAAAABBABABBAABBA 1 100000	300007

Note

In the first example, $S =$ `AB`. You can transform S into $T =$ `ABAABB` by performing the following operations:

- Insert `BA` between the 1st and 2nd characters of `AB`, resulting in `ABAB`.

- Insert `AB` between the 3rd and 4th characters of `ABAB`, resulting in `ABAABB`.

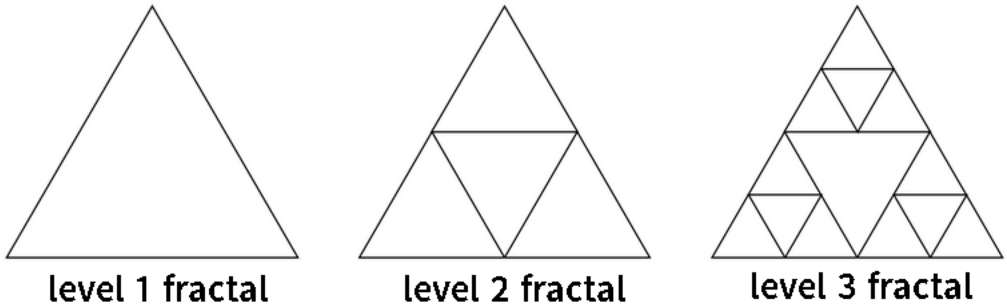
In this case, the total cost is $3 + 5 = 8$, which is the minimum possible total cost to achieve the transformation.

Problem J. Journey through the Fractal

Time limit: 2 seconds
Memory limit: 1024 megabytes

A **level i triangle** is a regular triangle with side length 2^{i-1}. A **level i fractal** is a figure generated by these rules:

- A level 1 fractal is a level 1 triangle.
- For $i \geq 1$, a level $i+1$ fractal is formed by placing three level i fractals so they perfectly touch each side of a level i triangle (see image below).



You are given a level L fractal.

Alice starts by picking any triangle in the fractal. Then, she can move to any unvisited triangle that shares an edge with her current triangle.

Alice can make up to K moves. The **score** is the sum of the levels of all triangles Alice visits (including her starting triangle).

Find the maximum possible score modulo 998244353. Note that you are asked for the maximum possible score, not the maximum remainder.

Input

The input is given in the following format:

L K

- All input values are integers.
- $1 \leq L \leq 10^9$.
- $1 \leq K \leq 10^{18}$.

Output

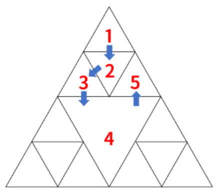
Print the answer in a single line.

Examples

standard input	standard output
3 4	6
998244353 1000000000000000007	756221200

Note

In the first testcase, Alice can visit four level 1 triangles and one level 2 triangle as below:


1
2
3
5
4

Problem K. K-rep Array

Time limit: 2 seconds
Memory limit: 1024 megabytes

For a positive integer K, a sequence V consisting of positive integers is said to be K-rep if it satisfies the following condition:

- There exists a sequence B of length K consisting of positive integers, such that the sequence B' obtained by repeating B for 10^{100} times contains V as a contiguous subsequence.

You are given a sequence $A = (A_1, A_2, \ldots, A_N)$ of length N, where each element is either a positive integer or -1. For each $K = 1, 2, \ldots, N$, solve the following problem:

- Determine whether there exists a replacement of each -1 in A with a positive integer such that the resulting sequence is K-rep.

Input

The input is given in the following format:

```
N
A_1 A_2 ... A_N
```

- All inputs are integers.
- $1 \le N \le 2 \times 10^5$.
- $1 \le A_i \le N$ or $A_i = -1$ for each i.

Output

Output a string of length N. The i-th character should be `1` if there exists a replacement satisfying the condition for the case $K = i$, and `0` otherwise.

Example

standard input	standard output
5 1 2 -1 2 1	01011

Note

In the example, one possible replacement of the elements $A_i = -1$ is the sequence $(1, 2, 3, 2, 1)$. For $K = 4$, let $B = (2, 3, 2, 1)$. Since the sequence B' obtained by repeating B contains $(1, 2, 3, 2, 1)$ as a contiguous subsequence, $(1, 2, 3, 2, 1)$ is K-rep.

Problem L. LIS Triangle

Time limit: 2 seconds
Memory limit: 1024 megabytes

You are given three positive integers N, K, and L. Your task is to determine whether there exists an integer sequence P of length N that satisfies all of the following conditions. If such a sequence exists, output one such sequence.

- P is a permutation of the sequence $(K, K+1, \ldots, K+N-1)$.
- The length of the longest increasing subsequence[1] of P is exactly L.
- For every integer i such that $1 \leq i \leq N-2$, there exists a non-degenerate triangle[2] with side lengths P_i, P_{i+1}, and P_{i+2}.

You are given T test cases. Answer each test case separately.

Input

The input is given in the following format:

```
T
case_1
case_2
⋮
case_T
```

Each test case is given in the following format:

```
N K L
```

- All input values are integers.
- $1 \leq T \leq 50000$.
- $3 \leq N \leq 2 \times 10^5$.
- $1 \leq K \leq 2 \times 10^5$.
- $1 \leq L \leq N$.
- Over all test cases in a single input, the sum of N is at most 2×10^5.

Output

If there is no sequence P that satisfies all the conditions, output:

```
No
```

Otherwise, output:

[1] A **subsequence** of a sequence P is a sequence formed by removing zero or more elements from P without changing the order of the remaining elements. A **longest increasing subsequence** of P is a strictly increasing subsequence of P that has the maximum possible length.

[2] A non-degenerate triangle is a triangle whose three vertices do not lie on the same straight line.

Yes
P_1 P_2 $\ldots$ P_N

Any valid sequence P that satisfies all conditions is accepted.

Example

standard input	standard output
3 6 3 4 5 5 5 7 1 2	Yes 3 6 4 7 5 8 Yes 5 6 7 8 9 No

Note

In the first example, one valid sequence is $P = (3, 6, 4, 7, 5, 8)$. Other valid sequences may also exist.

In the second example, the only valid sequence is $P = (5, 6, 7, 8, 9)$.

In the third example, no sequence P satisfies all the conditions.

Problem M. Minimum Distance Tree

Time limit: 2 seconds
Memory limit: 1024 megabytes

You are given a connected, undirected, weighted simple graph G with N vertices numbered from 1 to N and M edges. The i-th edge connects vertices u_i and v_i with weight w_i.

Determine whether there exists a weighted tree T with N vertices also numbered from 1 to N such that, for every pair of vertices u and v, the shortest path length between u and v in G is equal to the shortest path length between u and v in T.

Input

The input is given in the following format:

$$
\begin{array}{l}
N\ M \\
u_1\ v_1\ w_1 \\
u_2\ v_2\ w_2 \\
\vdots \\
u_M\ v_M\ w_M
\end{array}
$$

- All input values are integers.
- $2 \le N \le 5 \times 10^5$.
- $N - 1 \le M \le 5 \times 10^5$.
- $1 \le u_i, v_i \le N$.
- $1 \le w_i \le 10^9$.
- The given graph is simple and connected.

Output

If such a tree T exists, output:

```
Yes
```

Otherwise, output:

```
No
```

Examples

standard input	standard output
3 3 1 2 3 2 3 4 3 1 100	Yes
3 3 1 2 3 2 3 4 3 1 2	No

Note

In the first example, a tree T with 3 vertices, in which vertex 1 is connected to vertex 2 with weight 3, and vertex 2 is connected to vertex 3 with weight 4, satisfies the condition.

In the second example, no such tree T exists. For example, a tree where vertex 1 is connected to vertex 2 with weight 2, and vertex 1 is connected to vertex 3 with weight 2 does not satisfy the condition, because the shortest path between 1 and 2 in G is 3, while in this tree it is 2, which is not equal.

Problem N. Nice Bouquets

Time limit: 2 seconds
Memory limit: 1024 megabytes

The Universal Tropical Plant Center (UTPC) owns a plantation where N trees are planted in a straight line from west to east, numbered from 1 to N starting from the west.

The plantation is about to enter a K-day harvest period. During this period, each tree will bloom exactly one flower per day, in one of three colors: red, green, or blue. The blooming schedule of the i-th tree over the K days is represented by a string S_i, where the j-th character $S_{i,j}$ indicates the color of the flower that blooms on the j-th day: 'R' for red, 'G' for green, and 'B' for blue.

Each day, all flowers that have bloomed are harvested and grouped into bouquets of three flowers each. A valid bouquet must consist of either all flowers of the same color or all of different colors. Ideally, the harvested flowers should be grouped into such bouquets without any leftovers. However, it may not always be possible to do so.

To address this, UTPC allows the following operations to be performed on the trees **before** the harvest period begins. Each operation can be performed any number of times, but no tree can be operated on more than once:

- Choose an integer i ($1 \le i \le N$) and cut down the i-th tree. No flowers will be harvested from this tree.
- Choose an integer i ($1 \le i \le N$) and apply a growth accelerator to the i-th tree. The tree will bloom two flowers each day for the period, both of the color indicated by $S_{i,j}$ on the j-th day.

Operations cannot be performed after the harvest period begins.

Since UTPC's office is located on the western edge of the plantation, it prefers not to operate on trees that are further east. Therefore, the **cost** of a set of operations is defined as the **index of the easternmost tree that is operated on**. If no trees are operated on, the cost is 0.

Determine the minimum possible cost needed to ensure that, on every day of the harvest period, all flowers can be grouped into valid bouquets without any leftovers. Note that if all trees are cut down, it is considered that there are no leftover flowers.

Input

The input is given in the following format:

```
N K
S_1
S_2
⋮
S_N
```

- N and K are integers.
- $1 \le N, K$.
- $NK \le 10^5$.
- $|S_i| = K$.
- Each character in S_i is one of 'R', 'G', or 'B'.

Output

Print a single integer — the minimum cost to ensure no flowers are left ungrouped on any day of the harvest period.

Examples

standard input	standard output
4 5 RGBGR BGGBR RBGBR RRRRR	2
3 3 RGB BGG GGR	0
3 4 GGGG BGGG GGGR	3
6 4 BGGB BGGB RGBG RRRR GGGG BBBB	3

Note

In the first example, by cutting down the second tree, the flowers that bloom are:

- Day 1: RRR
- Day 2: GBR
- Day 3: BGR
- Day 4: GBR
- Day 5: RRR

Bouquets can be formed without leftovers on each day. The cost is 2, and it is not possible to achieve the goal with cost 1.

In the second example, no operations are necessary, so the cost is 0.

In the third example, all trees must be cut down to satisfy the condition. The cost is 3.

In the fourth example, by applying a growth accelerator to tree 1 and cutting down tree 3, the daily blooms become:

- Day 1: BBBRGB
- Day 2: GGGRGB
- Day 3: GGGRGB

- Day 4: BBBRGB

These can all be grouped into valid bouquets. The cost is 3.

Problem O. One Different Inequality

Time limit: 3 seconds
Memory limit: 1024 megabytes

You are given an integer N and a string S of length $N-1$ consisting of the characters < and >.

Let $P = (P_1, P_2, \cdots, P_N)$ be a permutation of $(1, 2, \cdots, N)$.

A permutation P is called a **Good Permutation** if it satisfies the following condition.

- For every i $(1 \le i \le N-1)$, if i-th character of S is <, then $P_i < P_{i+1}$; if it is >, then $P_i > P_{i+1}$.

A permutation P is called a **Wonderful Permutation** if it satisfies the following condition.

- P is a Good Permutation.
- The number of indices i $(1 \le i \le N-1)$ such that $|P_i - P_{i+1}| = 1$ is maximum among all Good Permutations.

Your task is to count the number of Wonderful Permutations modulo 998244353.

Input

The input is given in the following format:

```
N
S
```

- N is an integer.
- $2 \le N \le 2 \times 10^5$
- S is a string of length $N-1$ consisting of < or >.

Output

Output answer on single line.

Examples

standard input	standard output
5 <<>>	2
40 <<>><>><>>>><><<><><><<>><<<<>><><<<>><	535474657

Note

In the first test case, $(1,2,5,4,3)$ and $(2,3,5,4,1)$ are Good Permutations. The number of i which satisfies $|P_i - P_{i+1}| = 1$ is $3, 2$.

We can prove that the maximum number of i which satisfies $|P_i - P_{i+1}| = 1$ among Good Permutations is 3, and Wonderful Permutations are $(1,2,5,4,3)$ and $(3,4,5,2,1)$.

Problem P. Perfect Suika Game on a Tree

Time limit: 5 seconds
Memory limit: 1024 megabytes

You are given a tree T with N vertices, labeled from 1 to N. The i-th edge connects the vertices u_i and v_i.

Each vertex is assigned a positive integer called its **level**. Initially, the level of vertex $v = 1, 2, \ldots, N$ is A_v.

We consider the following problem on tree T:

> Determine whether it is possible to transform the tree T into a tree consisting of only a single vertex by performing the following operation exactly $N - 1$ times:
>
> - Select an edge whose endpoints have the same level and contract it. Let l be the common level of the two endpoints; then, the new vertex resulting from the contraction will have level $l + 1$.

You are given Q queries to process. In the i-th query, you are given the edge number e_i. After swapping the levels of the vertices u_{e_i} and v_{e_i} in the tree T (this swap also affects all subsequent queries), output the answer to the problem described above.

Input

The input is given in the following format:

$$
\begin{array}{l}
N \\
u_1\ v_1 \\
u_2\ v_2 \\
\vdots \\
u_{N-1}\ v_{N-1} \\
A_1\ A_2\ \ldots\ A_N \\
Q \\
e_1 \\
e_2 \\
\vdots \\
e_Q
\end{array}
$$

- All input values are integers.
- $2 \le N \le 2 \times 10^5$.
- $1 \le u_i, v_i \le N$.
- $1 \le A_i \le N$.
- $1 \le Q \le 2 \times 10^5$.
- $1 \le e_i \le N - 1$.
- The given graph is a tree.

Output

Output Q lines. For the i-th query, after swapping the levels of vertices u_{e_i} and v_{e_i}, output `Yes` if it is possible to transform T into a single-vertex tree using the operations described above; otherwise, output `No`.

Examples

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td>4
1 2
1 3
1 4
1 1 2 3
4
1
2
3
1</td><td>Yes
No
No
Yes</td></tr>
<tr><td>20
1 2
1 3
2 4
1 5
2 6
5 7
4 8
3 9
6 10
7 11
11 12
12 13
13 14
14 15
15 16
16 17
17 18
18 19
19 20
4 4 7 3 8 2 8 6 4 2 3 3 4 5 6 5 4 3 3 6
10
8
19
5
9
19
10
19
19
10
19</td><td>No
No
No
No
Yes
No
No
No
Yes
No</td></tr>
</table>

Note

In the first query in the first example, after swapping the levels of vertices $u_1 = 1$ and $v_1 = 2$, the levels of vertices $1, 2, 3, 4$ become $1, 1, 2, 3$ respectively. In this case, it is possible to perform the operations (selecting suitable edges) such that the tree becomes a single vertex with level 4. Therefore, the output is `Yes`. You may also find the following figure helpful.

In the second query, after swapping the levels of vertices $u_2 = 1$ and $v_2 = 3$, the levels of vertices $1, 2, 3, 4$

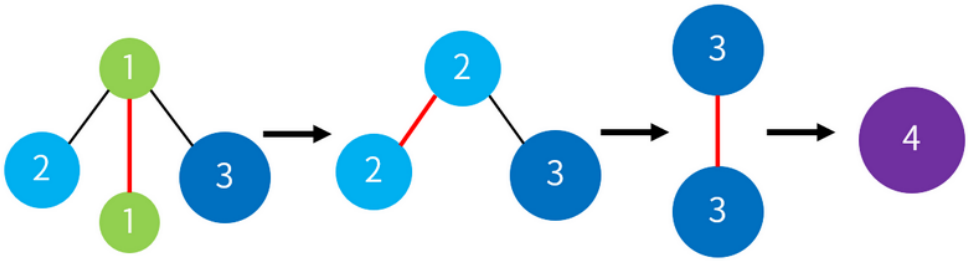


Рис. 1: Illustration of the first query in the first testcase

become $2, 1, 1, 3$ respectively. In this case, no operation can be performed at all, and it is impossible to transform the tree into a single vertex. Therefore, the output is `No`.

Problem Q. Quadratic Pieces

Time limit: 2 seconds
Memory limit: 1024 megabytes

You are given an integer sequence $A = (A_1, A_2, \ldots, A_N)$ of length N.

A contiguous subsequence $(A_L, A_{L+1}, \ldots, A_R)$, defined by integers L and R such that $1 \le L \le R \le N$, is said to be **quadratic** if the following condition is satisfied:

- There exists real numbers a, b, c such that for every integer i satisfying $L \le i \le R$, the equation $A_i = ai^2 + bi + c$ holds.

Your task is to partition the sequence A into several contiguous **quadratic** subsequences. Among all possible ways to do this, output the minimum number of such subsequences.

You are given T test cases. Output the answer for each test case.

Input

The input is given in the following format:

```
T
case_1
case_2
⋮
case_T
```

Each test case is given in the following format:

```
N
A_1 A_2 ... A_N
```

- All inputs are integers.
- $1 \le T \le 10^5$.
- $1 \le N \le 2 \times 10^5$.
- $-10^{18} \le A_i \le 10^{18}$.
- Over all test cases in a single input, the sum of N is at most 2×10^5.

Output

Output T lines.
For the i-th line, print the answer to the i-th test case.

Example

standard input	standard output
4 12 -16 -9 -4 -1 0 0 0 0 1 4 9 16 8 2 0 2 5 0 3 0 8 1 0 5 1000000000000000000 250000000000000000 0 250000000000000000 1000000000000000000	3 3 1 1

Note

In the first example, the given sequence can be partitioned into three quadratic subsequences: $(-16, -9, -4, -1)$, $(0, 0, 0)$, and $(0, 1, 4, 9, 16)$. For each of these subsequences, $(a, b, c) = (-1, 10, -25)$, $(0, 0, 0)$, and $(1, -16, 64)$ satisfies the condition. It is not possible to divide the sequence into fewer than 3 quadratic subsequences, so the answer is 3.

In the fourth example, note that the input values may exceed the 32-bit integer range.