Problem A. A

Time limit 2000 ms
Mem limit 1048576 kB

Problem Statement

Given is a three-digit integer N. Does N contain the digit 7?

If so, print `Yes`; otherwise, print `No`.

Constraints

- $100 \leq N \leq 999$

Input

Input is given from Standard Input in the following format:

```
N
```

Output

If N contains the digit 7, print `Yes`; otherwise, print `No`.

Sample 1

Input	Output
117	Yes

117 contains 7 as its last digit.

Sample 2

Input	Output
123	No

123 does not contain the digit 7.

Sample 3

Input	Output
777	Yes

the Median of the Median of the Median

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Today is YQH's birthday, and she received a positive integer sequence of length n, denoted as $\{a_i\}_{i=1}^n$, as a gift.

YQH is very interested in medians, so she wants to find the median of the median of the median for this sequence. Specifically, let $b_{l,r}$ be the median of the multiset $\{a_i\}_{l \le i \le r}$, and let $c_{l,r}$ be the median of the multiset $\{b_{i,j}\}_{l \le i \le j \le r}$. Then, what YQH wants to find is the median of the multiset $\{c_{l,r}\}_{1 \le l \le r \le n}$. However, she finds this task too difficult, so she asked you for help.

Note: If you are not familiar with the concept of a median, the median of a multiset of size m is the $\lceil m/2 \rceil$-th smallest element in it.

Input

A single positive integer on the first line represents n.

The second line contains n positive integers representing $a_1, \ldots, a_n$.

It is guaranteed that $1 \le n \le 2000$ and $1 \le a_i \le 10^9$.

Output

Output a single line with one integer representing the answer.

Examples

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

Problem C. C

Time limit 1500 ms
Mem limit 524288 kB
OS Windows

There is a string of length n, $S[l..r]$ represents the string concatenated from the lth character to the rth character, and S_{len} is the length of the string($S[1..S_{len}]$ represents the whole S string).

We define F_G as the number of positive integers x that satisfy the following conditions:

1. $1 \le x \le G_{len}$

2. $G[1,x] = G[G_{len} - x + 1, G_{len}]$

3. The length of the common part of the intervals $[1,x]$ and $[G_{len} - x + 1, G_{len}]$ is greater than 0 and is divisible by k.

Now ask for the value of $\prod_{i=1}^{n}(F_{S[1..i]} + 1)$ modulo 998244353.

Input

The first line of input is a positive integer $T(T \le 10)$ representing the number of data cases.

For each cases:

first line input a string S of lowercase letters, no longer than 10^6.

second line input a positive integer $k(1 \le k \le S_{len})$.

Output

For each cases, output a line with a positive integer representing the answer.

Sample

Input	Output
1 abababac 2	24

Hint

Note that the stack space of the judge system is a bit small, please pay attention to the reasonable allocation of memory.

Max Minus Min

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

You are given array a of n integers. You can perform the following operation at most once:

- Choose any integers $1 \le l \le r \le n$, and any integer x. Then, for each $l \le i \le r$, replace a_i with $a_i + x$.

Find the smallest possible value of $\max(a_1, a_2, \ldots, a_n) - \min(a_1, a_2, \ldots, a_n)$ you can get after performing this operation at most once.

Input

The first line contains a single integer t $(1 \le t \le 10^5)$ — the number of test cases. The description of test cases follows.

The first line of each test case contains a single integer n $(1 \le n \le 2 \cdot 10^5)$ — the length of the array.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ $(0 \le a_i \le 10^9)$ — elements of the array.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, in the first line output a single integer — the smallest possible value of $\max(a_1, a_2, \ldots, a_n) - \min(a_1, a_2, \ldots, a_n)$ you can get after performing this operation at most once.

Example

standard input	standard output
4 3 42 42 42 4 1 2 2 1 5 1 100 1 100 1 6 1 2 3 4 5 6	0 0 99 2

Note

In the first test case, you don't need to make any operations, since $max - min = 0$ already.

In the second test case, you can choose $x = -1$, and segment $a[2:3]$. The array will become $[1, 1, 1, 1]$, with $max - min = 0$.

In the third test case, $max - min$ initially is 99. Unfortunately, it's not possible to decrease this value with a single operation.

In the fourth test case, $max - min$ initially is 5, but we can choose $x = 3$ and segment $a[1:3]$. The array will become $[4, 5, 6, 4, 5, 6]$, with $max - min = 2$.

Problem J
’S No Problem
Time limit: 3 seconds

The Yllihc Engineering and Technological Institute (YETI), located in northern Snowblovia, has two problems: snow and money. Specifically, they have too much of the former and not enough of the latter. Every winter (and fall and spring, for that matter) the campus is covered with blankets of snow, and the sidewalks connecting campus buildings become impassable. To continue functioning, YETI needs to clear the snow from the sidewalks connecting the buildings of the campus. Since budget is an issue, these sidewalks are a minimal set that allow a path to exist between any two buildings.

With the money saved by not building more sidewalks, YETI bought two snow blowers. To clear the snow with them, two staff members take the two snow blowers out of the buildings (or building) they are stored in, and push them along the sidewalks, clearing the snow. Each sidewalk must be traversed at least once. Each snow blower, after it has finished, is stored in the building it is currently next to (and, during the next snowfall, the snow blowers will be pushed in the reverse direction—and so on, throughout the eleven months of the snow season).

The YETI maintenance crew want to choose the storage buildings of the snow blowers and design the routes along which they will be pushed, so that they minimize the total distance the two machines will travel out in the cold (to protect both the precious equipment and the staff members from freezing). Note that the routes might involve pushing along already-cleared sidewalks, as seen in Figure J.1, which shows an optimal solution for the sidewalk layout of Sample Input 1.

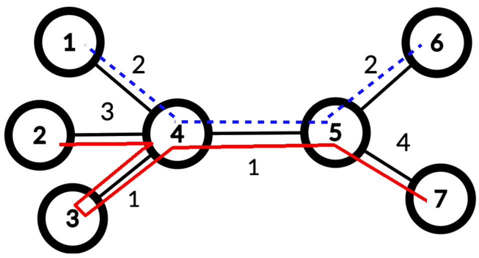


Figure J.1: Illustration of Sample Input 1 showing one possible pair of optimal routes.

YETI would ask their Computer Science Department to figure this out, but it was wiped out in the Great Blizzard of ’06, so they have come to you for help.

Input

The first line of input contains an integer n ($4 \le n \le 100\,000$), the number of buildings on the YETI campus. Buildings are numbered from 1 to n. Each of the remaining $n-1$ lines contains three integers a, b, and d indicating that a sidewalk exists between buildings a and b ($1 \le a, b \le n$; $a \ne b$) of length d ($1 \le d \le 500$).

Output

Output the minimum total distance the snow blowers must travel to remove snow from all sidewalks.

Sample Input 1	Sample Output 1
7 1 4 2 2 4 3 3 4 1 4 5 1 5 6 2 5 7 4	15

Sample Input 2	Sample Output 2
4 1 2 1 2 3 2 3 4 3	6

I Flipped The Calendar...

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

While flipping through the calendar, Nikolai wondered: how many rows are in the calendar for a specific year?

The calendar consists of 12 sheets, each corresponding to a month from January to December. Each sheet lists all the days of the respective month. The days on each sheet are arranged in rows by week: the days of one week are in one row, the days of different weeks are in different rows. In this calendar, the week starts on Monday.

For example, if a month has 31 days and the first day of the month is Sunday (as in January 2023), then there will be six rows on the calendar sheet for that month:

						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Remember that in a leap year, February has 29 days, and in a non-leap year, it has 28 days. A year is considered a leap year if its number is divisible by 400 or divisible by 4 but not by 100. For example, 2000, 2004, and 2040 are leap years, while 1900, 1982, and 2039 are not.

Input

The first line contains the year number y ($1970 \le y \le 2037$).

Output

Output the number of rows in the calendar for the given year.

Example

standard input	*standard output*
2023	63

Alice and Bob

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	1024 megabytes

Alice and Bob are playing a game.

They are given a permutation p, and take turns to perform the following operation, with Alice going first:

- Operation: Rearrange $p_{1 \cdots p_1}$ in any desired order.

If someone do two operations with the same p_1, he or she loses.

Alice and Bob are both strategically adept and will always choose the optimal operation to secure a win.

Given all permutations of size n, determine how many of them Bob will win, modulo 998244353.

Input

The first line of the input contains a single integer n ($1 \le n \le 10^7$).

Output

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

Examples

standard input	standard output
1	1
2	1
10	997920
100	188898954

Jackpot

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

You are given an integer array a of $2n$ integers. In one operation, you can do the following:

- Choose any two adjacent elements in the array, let's say, a_i and a_{i+1}. Then, delete them, and add $|a_i - a_{i+1}|$ to your score.

  Note that the indices are recalculated after the operation.

You are going to perform this operation n times, deleting all the elements in the end. What is the largest score you can get?

Input

The first line contains a single integer t $(1 \le t \le 10^5)$ — the number of test cases. The description of test cases follows.

The first line of each test case contains a single integer n $(1 \le n \le 2 \cdot 10^5)$.

The second line of each test case contains $2n$ integers $a_1, a_2, \ldots, a_{2n}$ $(0 \le a_i \le 10^9)$ — elements of the array.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output a single integer — the largest possible score you can get after performing the operation n times.

Example

standard input	standard output
3 2 42 42 42 42 1 42 69 3 1 2 3 4 5 6	0 27 9

Note

In the first test case, we can choose the first two elements, and delete them, getting score of 0, and then choose the remaining two elements, delete them, getting score of 0 again.

In the second test case, the only possible operation is to choose both elements and to get score of $|42 - 69| = 27$.

In the third test case, we can do the following sequence of operations:

1. Choose elements $3, 4$. Delete them, and get a score of 1. The array will become $[1, 2, 5, 6]$.
2. Choose elements $2, 5$. Delete them, and get a score of 3. The array will become $[1, 6]$.
3. Choose elements $1, 6$. Delete them, and get score of 5. The total score is $1 + 3 + 5 = 9$.

Rooted Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	512 megabytes

You are given a rooted tree that initially contains only a single vertex (which is the root vertex). A vertex is considered a *leaf* if it does not have any children. You have the ability to perform a specific operation exactly K times on this tree, according to the following steps:

- Choose a leaf vertex u randomly with uniform probability.
- Add M new leaf vertices as children of vertex u.

Define the depth of the vertex u (denoted by $d(u)$) as follows:

- For the root vertex, $d(u) = 0$.
- For any other vertex, $d(u) = d(v) + 1$, where v is the parent of vertex u.

Your task is to determine the expected value of the sum of the depths of all vertices in the tree after performing the specified operation K times, modulo $(10^9 + 9)$.

Input

The first line of the input contains two integers M and K ($1 \le M \le 100$, $1 \le K \le 10^7$).

Output

Output a single line contains a single integer, indicating the answer modulo $(10^9 + 9)$.

Formally, assuming the answer is simplified to the form p/q (i.e., p and q are coprime), please output x such that $qx \equiv p \pmod{(10^9 + 9)}$, and $0 \le x < 10^9 + 9$. It can be proven that x exists and is unique.

Examples

standard input	standard output
6 2	18
2 6	600000038
83 613210	424200026

Matrix Distances

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

Mr. Ham has a matrix of size $n \times m$, where each cell is filled with a color value $c_{i,j}$. Mr. Ham is interested in the relationships between cells of the same color and wants to calculate the sum of Manhattan distances between all pairs of cells with the same color.

The Manhattan distance between two cells (x_1, y_1) and (x_2, y_2) is given by $d_M = |x_1 - x_2| + |y_1 - y_2|$.

Formally, please compute:

$$\text{Total Sum} = \sum_{C} \sum_{(x_i,y_i)\in S_C} \sum_{(x_j,y_j)\in S_C} |x_i - x_j| + |y_i - y_j|$$

Here, C denotes the set of all distinct colors. S_C denotes the set of coordinates (x, y) with the color value equal to the specified color C.

Input

The first line contains two integers n and m ($1 \leq n, m \leq 1000$), representing the number of rows and columns in the matrix.

The following contains n lines. Each line i contains m space-separated integers $c_{i,j}$ ($1 \leq c_{i,j} \leq 10^9$), representing the color values in the matrix.

Output

Output a single integer, the sum of Manhattan distances for all pairs of cells with the same color.

Examples

standard input	standard output

`2 2`
`1 1`
`2 2`
	`4`

`4 4`
`1 3 2 4`
`2 1 2 3`
`1 3 3 2`
`3 2 1 4`
	`152`

Note

In the first example, the distinct color values are 1 and 2.

1. For color 1:
   - The coordinates with color 1 are $(1, 1)$ and $(1, 2)$.
   - The Manhattan distance between $(1, 1)$ and $(1, 1)$ is $|1 - 1| + |1 - 1| = 0$.
   - The Manhattan distance between $(1, 1)$ and $(1, 2)$ is $|1 - 1| + |1 - 2| = 1$.
   - The Manhattan distance between $(1, 2)$ and $(1, 1)$ is $|1 - 1| + |2 - 1| = 1$.
   - The Manhattan distance between $(1, 2)$ and $(1, 2)$ is $|1 - 1| + |2 - 2| = 0$.
2. For color 2:

- The coordinates with color 1 are $(2, 1)$ and $(2, 2)$.
- The Manhattan distance between $(2, 1)$ and $(2, 1)$ is $|2 - 2| + |1 - 1| = 0$.
- The Manhattan distance between $(2, 1)$ and $(2, 2)$ is $|2 - 2| + |1 - 2| = 1$.
- The Manhattan distance between $(2, 2)$ and $(2, 1)$ is $|2 - 2| + |2 - 1| = 1$.
- The Manhattan distance between $(2, 2)$ and $(2, 2)$ is $|2 - 2| + |2 - 2| = 0$.

So, the total sum of Manhattan distances for all pairs of cells with the same color is $1 + 1 + 1 + 1 = 4$.

Make Max

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

You are given a sequence of n positive integers $[a_1, \ldots, a_n]$. You can apply the following operation to the sequence:

- Select a subarray $a[l \ldots r]$ ($1 \le l < r \le n$) where not all elements are identical (i.e., there exist two integers $l \le i < j \le r$ such that $a_i \ne a_j$), and then change every element in this subarray to $\max_{l \le i \le r} a_i$.

Determine the maximum number of such operations that can be performed.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 1000$). Description of the test cases follows.

The first line of each test case contains an integer n ($1 \le n \le 2 \times 10^5$).

The second line of each test case contains n integers $a_1, \ldots, a_n$ ($1 \le a_i \le 10^9$).

The sum of n over all test cases does not exceed 4×10^5.

Output

For each test case, print the maximum number of operations that can be performed.

Example

standard input	standard output
4 2 1 2 2 2 2 7 1 1 1 2 2 2 2 3 1 2 3	1 0 3 3

Note

In the first test case, an optimal sequence of operations is:

1. Select $a[1 \ldots 2]$ and apply the operation, so that a becomes $[2, 2]$.

In the second test case, no operation can be performed.

In the fourth test case, an optimal sequence of operations is:

1. Select $a[1 \ldots 2]$ and apply the operation, so that a becomes $[2, 2, 3]$.
2. Select $a[2 \ldots 3]$ and apply the operation, so that a becomes $[2, 3, 3]$.
3. Select $a[1 \ldots 3]$ and apply the operation, so that a becomes $[3, 3, 3]$.

Knapsack

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Little Cyan Fish, an inexperienced businessman, recently launched a store named *Queen's Organic Jewelry*. This jewelry store houses n gemstones, where the i-th gemstone is priced at w_i dollar and has a beauty of v_i. Prior to visiting the store, you have W dollars in hand, which you plan to use to purchase gemstones of the greatest possible total beauty.

Interestingly, Little Cyan Fish's store is running a promotion today. Any visitor to the store can select any k gemstones and take them home absolutely free of charge! With this opportunity at hand, you're keen to know the maximum total beauty of gemstones you could obtain with your W dollars, assuming you adopt the optimal strategy.

Please bear in mind that the store stocks only one unit of each gemstone, so you cannot obtain the same gemstone more than once. Also note that you don't have to spend all the money.

Input

There is only one test case in each test file.

The first line of the input contains three integers n, W and k ($1 \le n \le 5 \times 10^3$, $1 \le W \le 10^4$, $0 \le k \le n$), indicating the total number of gemstones in the store, the amount of money you have and the number of gemstones you can take for free.

For the following n lines, the i-th line contains two integers w_i and v_i ($1 \le w_i \le W$, $1 \le v_i \le 10^9$), indicating the price and the beauty of the i-th gemstone.

Output

Output one line containing one integer indicating the answer.

Examples

standard input	standard output
4 10 1 9 10 10 1 3 5 5 20	35
5 13 2 5 16 5 28 7 44 8 15 8 41	129

Note

In the first example, Little Cyan Fish's shop holds 4 gemstones and you are permitted to take 1 gemstone for free. One optimal strategy involves taking the first gemstone for free, and purchasing the third and fourth gemstones.

Gemstone	**Price** w_i	**Beauty** v_i	**Action**
1	9	10	Take for free
2	10	1	/
3	3	5	Purchase
4	5	20	Purchase

So the answer is $10 + 5 + 20 = 35$.

Indeterminate Equation

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

Given positive integers n, k, find the number of positive integer solutions to the indeterminate equation $a^k - b^k = n$.

Input

The first line of input is an integer T $(1 \le T \le 20)$ indicating the number of queries. The following T lines, each contain two integers n, k indicating a single query. It is guaranteed that $1 \le n \le 10^{18}$, $3 \le k \le 64$.

Output

For each query, output a single line contains a single integer, indicating the answer.

Example

standard input	standard output
3 7 3 15 4 31 5	1 1 1

Problem N. N

Time limit 4000 ms
Mem limit 262144 kB
OS Windows

When we display a string on a LED screen, there may be different energy consumptions for displaying different characters. Given a string S with length n consisting of lowercase English letters and the energy consumption of each letter. Assume that the energy consumption of a string is the sum of the energy consumptions of all its characters. Please find the k-th smallest energy consumption of all the **different** substrings of S.

Note that a substring of a string S is a contiguous subsequence of S. We say two strings S and T are different if the lengths of them are different or there exists at least one position i satisfying $S[i] \neq T[i]$.

Input

The first line of the input contains an integer T $(1 \leq T \leq 2 \times 10^3)$ --- the number of test cases.

The first line of each test case contains two integers n $(1 \leq n \leq 10^5)$ and k $(1 \leq k \leq \frac{n(n+1)}{2})$.

The second line of each test case contains a string S with length n consisting of lowercase English letters.

The third line of each test case contains 26 integers $c_a, c_b, \ldots, c_z$ $(1 \leq c_\alpha \leq 100)$ --- the energy consumption of each letter.

It is guaranteed that the sum of n among all test cases does not exceed 8×10^5.

Output

For each test case, output a single integer in a separate line --- the k-th smallest energy consumption, or -1 if there is no answer.

Sample

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td><pre>2
5 5
ababc
3 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1
5 15
ababc
3 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1</pre></td><td><pre>4
-1</pre></td></tr>
</table>

Problem O. O

Time limit 5000 ms
Mem limit 1048576 kB
OS Windows

There is a convex polygon with n vertices. Vertices are numbered from 1 to n (both inclusive) in counter-clockwise order, and vertex i has a value of $f(i)$.

We say a subset of the vertices is palindromic, if their values constitute a palindrome in counter-clockwise order. More formally, let's say the subset contains k vertices $v_0, v_1, \cdots, v_{k-1}$ in counter-clockwise order. There should exist an integer d such that $0 \le d < k$, and for all $0 \le i < k$ we have $f(v_{(d+i) \bmod k}) = f(v_{(d-1-i) \bmod k})$.

Among all palindromic subsets, find the one whose convex hull has the largest size.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains an integer n ($3 \le n \le 500$) indicating the number of vertices of the convex polygon.

The second line contains n integers $f(1), f(2), \cdots, f(n)$ ($1 \le f(i) \le 10^9$) where $f(i)$ is the value of the i-th vertex.

For the following n lines, the i-th line contains two integers x_i and y_i ($-10^9 \le x_i, y_i \le 10^9$) indicating the coordinates of the i-th vertex. The vertices are listed in counter-clockwise order. The convex polygon is guaranteed to have positive size and no two vertices coincide. However there might be three vertices lying on the same line.

It's guaranteed that the sum of n of all test cases does not exceed 10^3.

Output

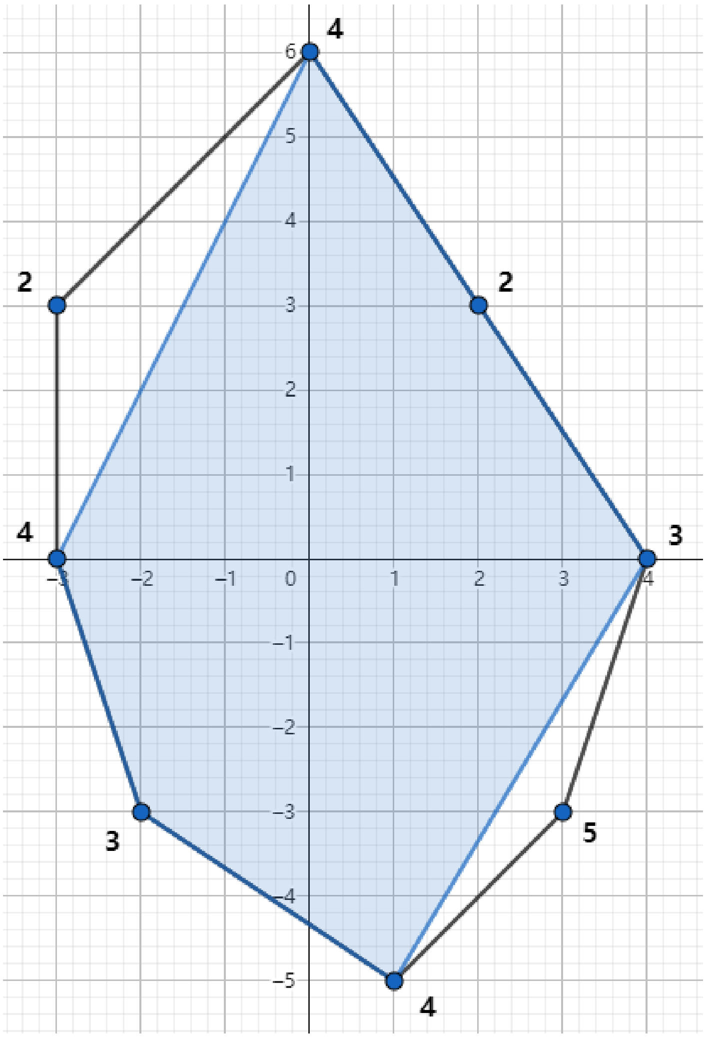
For each test case output one line containing one integer, indicating the size of the largest convex hull of a palindromic subset, multiplied by 2. It can be proven that this value is always an integer.

Examples

Input	Output
3 8 2 4 2 4 3 4 5 3 2 3 0 6 -3 3 -3 0 -2 -3 1 -5 3 -3 4 0 3 1 2 3 0 0 1 0 0 1 3 1 1 1 0 0 1 0 0 1	84 0 1

Note

The first sample test case is illustrated below. Choose vertices $2, 4, 5, 6, 8$ and consider $d = 1$, then the value sequence $\{4, 3, 4, 3, 4\}$ is a palindrome.


4
2
2
4
3
3
5
4

Not Another Path Query Problem

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

What age is it that you are still solving traditional path query problems?

After reading the paper *Distributed Exact Shortest Paths in Sublinear Time*, you have learned how to solve the distributed single-source shortest paths problem in $\mathcal{O}(D^{1/3} \cdot (n \log n)^{2/3})$. To give your knowledge good practice, Little Cyan Fish prepared the following practice task for you.

Little Cyan Fish has a graph consisting of n vertices and m bidirectional edges. The vertices are numbered from 1 to n. The i-th edge connects vertex u_i to vertex v_i and is assigned a weight w_i.

For any path in the graph between two vertices u and v, let's define the value of the path as the bitwise AND of the weights of all the edges in the path.

As a fan of high-value paths, Little Cyan Fish has set a constant threshold V. Little Cyan Fish loves a path if and only if its value is at least V.

Little Cyan Fish will now ask you q queries, where the i-th query can be represented as a pair of integers (u_i, v_i). For each query, your task is to determine if there exists a path from vertex u_i to vertex v_i that Little Cyan Fish would love it.

Input

There is only one test case in each test file.

The first line contains four integers n, m, q and V ($1 \le n \le 10^5$, $0 \le m \le 5 \times 10^5$, $1 \le q \le 5 \times 10^5$, $0 \le V < 2^{60}$) indicating the number of vertices, the number of edges, the number of queries and the constant threshold.

For the following m lines, the i-th line contains three integers u_i, v_i and w_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$, $0 \le w_i < 2^{60}$), indicating a bidirectional edge between vertex u_i and vertex v_i with the weight w_i. There might be multiple edges connecting the same pair of vertices.

For the following q lines, the i-th line contains two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$), indicating a query.

Output

For each query output one line. If there exists a path whose value is at least V between vertex u_i and v_i output `Yes`, otherwise output `No`.

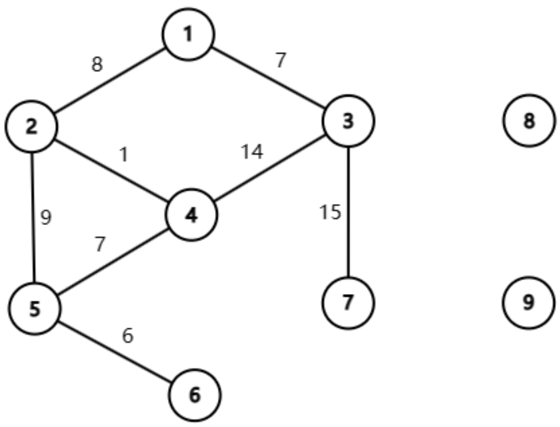
Examples

standard input	standard output
9 8 4 5 1 2 8 1 3 7 2 4 1 3 4 14 2 5 9 4 5 7 5 6 6 3 7 15 1 6 2 7 7 6 1 8	Yes No Yes No
3 4 1 4 1 2 3 1 2 5 2 3 2 2 3 6 1 3	Yes

Note

We now use & to represent the bitwise AND operation.

The first sample test case is shown as follows.



- For the first query, a valid path is $1 \to 3 \to 4 \to 5 \to 6$, whose value is $7 \,\&\, 14 \,\&\, 7 \,\&\, 6 = 6 \geq 5$.
- For the third query, a valid path is $7 \to 3 \to 4 \to 5 \to 6$, whose value is $15 \,\&\, 14 \,\&\, 7 \,\&\, 6 = 6 \geq 5$.
- For the fourth query, as there is no path between vertex 1 and 8, the answer is `No`.

For the only query of the second sample test case, we can consider the path consisting of the 2-nd and the 4-th edge. Its value is $5 \,\&\, 6 = 4 \geq 4$.