The 3rd Universal Cup

Stage 16: Nanjing

November 9-10, 2024

This problem set should contain 13 problems on 20 numbered pages.

Based on

International Collegiate Programming Contest (ICPC)

Hosted by

Prepared by

Problem A. Hey, Have You Seen My Kangaroo?

Time limit: 3 seconds
Memory limit: 128 megabytes

Please note the UNUSUAL MEMORY LIMIT of this problem.

After the great success in 2018, 2019, 2020, 2021, 2022, and 2023, the Nanjing University of Aeronautics and Astronautics (NUAA) will host the *International Collegiate Programming Contest* (ICPC) Nanjing regional for the seventh time in a row.

Team ***Power of Two*** and team ***Three Hold Two*** won the champion title for Tsinghua University in 2018 and 2019. In 2020, 2021, and 2022, team ***Inverted Cross*** from Peking University won the three-peat champion titles. In 2023, another team ***Reborn as a Vegetable Dog*** from Peking University won the title. They also won the 46th ICPC World Champion, reclaiming the trophy for the EC region after 13 years!

This year, around 335 teams are participating in the contest. At most 33 gold medals, 66 silver medals, and 99 bronze medals will be awarded (note that these numbers are for reference only). We are looking forward to seeing the participants' outstanding performance! We also want to express our gratitude for the hard work done by all staff and volunteers for this contest. Thank you all for your great contribution to this contest!

Photo taken in the 2023 ICPC Asia Nanjing Regional Contest

In the 2018 contest, problem K, ***Kangaroo Puzzle***, requires the contestants to construct an operation sequence for the game:

> The puzzle is a grid with n rows and m columns ($1 \le n, m \le 20$), and there are some (at least 2) kangaroos standing in the puzzle. The player's goal is to control them to get together. There are some walls in some cells, and the kangaroos cannot enter the cells with walls. The other cells are empty. The kangaroos can move from an empty cell to an adjacent empty cell in four directions: up, down, left, and right.
>
> There is exactly one kangaroo in every empty cell in the beginning, and the player can control the kangaroos by pressing the buttons U, D, L, R on the keyboard. The kangaroos will move simultaneously according to the button you press.
>
> The contestant needs to construct an operating sequence of at most 5×10^4 steps consisting of U, D, L, R only to achieve the goal.

In the 2020 contest, problem A, ***Ah, It's Yesterday Once More***, requires the contestants to construct an input map to hack the following code of the problem described before:

```
#include <bits/stdc++.h>
using namespace std;
string s = "UDLR";
int main()
{
  srand(time(NULL));
  for (int i = 1; i <= 50000; i++) putchar(s[rand() % 4]);
  return 0;
}
```

Furthermore, in the 2021 contest (Problem A, ***Oops, It's Yesterday Twice More***), the 2022 contest (Problem A, ***Stop, Yesterday Please No More***), and the 2023 contest (Problem A, ***Cool, It's Yesterday Four Times More***), every year we have a problem related to the kangaroos! We would like to introduce all these problems to you, but if we do so every year, we may have a 500-page statement for one single problem in the 3024 contest. Therefore, we omit them this time. Besides, you may already have seen them in the practice contest.

Now, in the 2024 contest, as everyone expects, the kangaroo problem is back again! We don't know why problem setters are so obsessed with kangaroos, but the problem is as follows:

You are given a grid with n rows and m columns. There are some walls in some cells, and the kangaroos cannot enter the cells with walls. The other cells are empty and each contains a kangaroo. The kangaroos can move from an empty cell to an adjacent empty cell in four directions: up, down, left, and right.

You can control the kangaroos by pressing the buttons U, D, L, R on the keyboard. The kangaroos will move simultaneously according to the button you press. Specifically, for any kangaroo located in the cell on the i-th row and the j-th column, indicated by (i, j):

1. Button U: it will move to $(i-1, j)$ if $i > 1$ and $(i-1, j)$ is not a wall. Otherwise, it will stay in the same cell.

2. Button D: it will move to $(i+1, j)$ if $i < n$ and $(i+1, j)$ is not a wall. Otherwise, it will stay in the same cell.

3. Button L: it will move to $(i, j-1)$ if $j > 1$ and $(i, j-1)$ is not a wall. Otherwise, it will stay in the same cell.

4. Button R: it will move to $(i, j+1)$ if $j < m$ and $(i, j+1)$ is not a wall. Otherwise, it will stay in the same cell.

You are given an operating sequence $s_1 s_2 \ldots s_k$ consisting only of characters 'U', 'D', 'L', and 'R'. The operations are performed infinitely according to the sequence. Specifically, if $1 \le t \le k$, the t-th operation is s_t; Otherwise if $t > k$, the t-th operation is the same as the $(t-k)$-th operation. For each $1 \le i \le n \times m$, find the smallest integer v_i such that after performing v_i operations, at most i cells will contain kangaroos.

Input

There is only one test case in each test file.

The first line contains three integers n, m, and k ($1 \le n, m \le 2 \times 10^5$, $1 \le n \times m \le 2 \times 10^5$, $1 \le k \le 200$), indicating the number of rows and columns of the grid, and the length of the operating sequence.

The second line contains a string $s_1 s_2 \cdots s_k$ ($s_i \in \{\text{'U'}, \text{'D'}, \text{'L'}, \text{'R'}\}$), indicating the operating sequence.

For the following n lines, the i-th line contains a binary string $a_{i,1} a_{i,2} \cdots a_{i,m}$ ($a_{i,j} \in \{\text{'0'}, \text{'1'}\}$). If $a_{i,j} = \text{'1'}$ then cell (i, j) is empty; Otherwise if $a_{i,j} = \text{'0'}$ then cell (i, j) is blocked and cannot be entered. It is guaranteed that there is at least one empty cell in the grid.

Output

Output $n \times m$ lines, where the i-th line contains an integer v_i, indicating the minimum number of operations needed so that at most i cells will contain kangaroos. If this is impossible, just output `-1` on this line.

Examples

standard input	standard output
3 3 6 ULDDRR 010 111 010	-1 4 2 1 0 0 0 0 0
3 3 6 ULDDRR 010 111 011	7 4 2 1 1 0 0 0 0
1 5 1 R 11111	4 3 2 1 0

Problem B. Birthday Gift

Time limit:	1 second
Memory limit:	1024 megabytes

Grammy's birthday is approaching, and she gets a sequence A from her friends as a gift. The sequence consists of only 0, 1, and 2. Grammy thinks that the sequence is too long, so she decides to modify A to make it shorter.

Formally, Grammy can perform an arbitrary number of operations. Each time she can choose one of the following three operations to perform:

- Change any 2 into 0 or 1.
- Choose two adjacent 0s, erase them, and concatenate the rest of the parts.
- Choose two adjacent 1s, erase them, and concatenate the rest of the parts.

Calculate the minimum sequence length Grammy can get.

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 and only line contains a string of length n ($1 \le n \le 2 \times 10^5$) consisting of digits 0, 1, and 2, indicating the initial sequence A.

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

Output

For each test case, output one line containing one integer indicating the minimum sequence length Grammy can get.

Example

standard input	standard output
5 0110101 01020102 0000021111 1012121010 0100202010	3 4 0 6 0

Problem C. Topology

Time limit:	2 seconds
Memory limit:	1024 megabytes

You are given a tree consisting of n vertices, rooted at vertex 1. It is guaranteed that every vertex has a smaller index than all of its children. A topological order of this tree is a permutation $p_1, p_2, \ldots, p_n$ of n that satisfies the following constraint: For all $1 \le i < j \le n$, vertex p_j is not the parent of vertex p_i.

For each $1 \le i \le n$, calculate the number of topological orders of the given tree satisfying $p_i = i$, modulo $998\,244\,353$.

Input

There is only one test case in each test file.

The first line contains an integer n ($2 \le n \le 5\,000$), denoting the number of vertices of the tree.

The second line contains $(n-1)$ integers $f_2, f_3, \ldots, f_n$ ($1 \le f_i < i$), where f_i is the parent of vertex i.

Output

Output one line containing n integers $a_1, a_2, \cdots, a_n$ separated by a space, where a_i is the number of topological orders satisfying $p_i = i$, modulo $998\,244\,353$.

Examples

standard input	standard output
4 1 1 2	3 2 1 2
9 1 1 2 2 3 3 4 5	672 420 180 160 152 108 120 170 210

Note

For the first sample test case, all topological orders of the tree are $\{1,2,3,4\}$, $\{1,3,2,4\}$ and $\{1,2,4,3\}$. There are 3 of them satisfying $p_1 = 1$, 2 of them satisfying $p_2 = 2$, 1 of them satisfying $p_3 = 3$, and 2 of them satisfying $p_4 = 4$.

Problem D. Toe-Tac-Tics

Time limit:	1 second
Memory limit:	1024 megabytes

Alice and Bob are playing Toe-Tac-Tics on n boards with 3 rows and 3 columns. Some cells on the boards are initially empty, while the others already contain some marks. Alice moves first, and they take turns to select a board and put their marks into an empty cell on that board. Alice's mark is 'x' and Bob's mark is 'o'.

Each player must make sure that no three same marks are in any row, column, or diagonal on any board after his/her move. The player who cannot make a valid move on their turn loses, and the other player wins.

Given the initial state of the n boards, you need to determine who wins, assuming both players play optimally for victory.

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 ($1 \le n \le 10^5$), indicating the number of boards in the game.

Then n boards of size 3×3 follow. For each board:

- There will first be an empty line if it is not the first board.
- For the following three lines, the i-th line contains a string $s_{i,1}s_{i,2}, s_{i,3}$ of length 3 consisting of characters 'x', 'o', and '.', describing a board of size 3×3. Let (i, j) be the cell on the i-th row and the j-th column. If $s_{i,j} =$ 'x' then cell (i, j) contains a mark 'x'; If $s_{i,j} =$ 'o' then cell (i, j) contains a mark 'o'; If $s_{i,j} =$ '.' then cell (i, j) is empty.

It is guaranteed that no three same marks are in any row, column, or diagonal on any board. It is also guaranteed that the sum of n for all test cases does not exceed 10^5.

Output

For each test case, output `Alice` if Alice wins the game, or `Bob` if Bob wins the game.

Example

standard input	standard output
4 1 ... ... 1 ... oo. oo. 2 ... oo. oo. ... xx. xx. 2 ..x xo. ... xo. o.. .x.	Alice Alice Bob Bob

Problem E. Left Shifting 3

Time limit:	1 second
Memory limit:	1024 megabytes

Given a string $S = s_0 s_1 \cdots s_{n-1}$ of length n, you can shift S to the left for at most k times (including zero times). Calculate the maximum number of "nanjing" substrings contained in the string after the operations.

More formally, let $f(S, d)$ be the string obtained by shifting S to the left d times. That is, $f(S, d) = s_{(d+0) \bmod n} s_{(d+1) \bmod n} \cdots s_{(d+n-1) \bmod n}$. Let $g(f(S, d), l, r) = s_{(d+l) \bmod n} s_{(d+l+1) \bmod n} \cdots s_{(d+r) \bmod n}$. Let $h(d)$ be the number of integer pairs (l, r) such that $0 \le l \le r < n$ and $g(f(S, d), l, r) = \texttt{nanjing}$. Find an integer d such that $0 \le d \le k$ to maximize $h(d)$ and output this maximized value.

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 two integers n and k ($1 \le n \le 2 \times 10^5$, $0 \le k \le 10^9$) indicating the length of the string and the maximum number of left shifts you can perform.

The second line contains a string $s_0 s_1 \cdots s_{n-1}$ of length n. The string consists of lower-cased English letters.

It's guaranteed that the sum of n of all test cases will not exceed 5×10^5.

Output

For each test case, output one line containing one integer, indicating the maximum number of "nanjing" substrings contained in the string.

Example

standard input	standard output
4	2
21 10	1
jingicpcnanjingsuanan	3
21 0	0
jingicpcnanjingsuanan	
21 3	
nanjingnanjingnanjing	
4 100	
icpc	

Note

For the first sample test case, we can shift the string to the left 6 times and get the string "pcnanjingsuananjingic". There are two "nanjing" substrings.

For the second sample test case, because $k = 0$, we cannot perform any left shifting. There is one "nanjing" substring in the original string.

Problem F. Subway

Time limit:	2.5 seconds
Memory limit:	1024 megabytes

In Pigeland, the subway system is quite advanced. It consists of n sites, numbered from 1 to n, and k directed subway lines, numbered from 1 to k. Subway line i travels through sites $x_{i,1}, x_{i,2}, \cdots, x_{i,p_i}$ in order, where $x_{i,j}$ is the j-th site visited by line i. It takes $w_{i,j}$ units of time to travel from site $x_{i,j}$ to site $x_{i,j+1}$ on line i.

When multiple lines meet at the same site, passengers can transfer between lines. If a passenger is at a site on line x, while line y also passes through this site, he/she can spend $a_y \times b_x$ units of time to transfer from line x to line y, where a_y and b_x are given coefficients for lines y and x. After transferring, the passenger is still at the same site, but on line y.

You start at site 1. Find the minimum time needed to reach site s for all $2 \leq s \leq n$. In particular, you can start by choosing any line at site 1 with no transfer time cost. It is guaranteed that all sites are reachable from site 1.

Input

There is only one test case in each test file.

The first line contains two integers n and k ($2 \leq n \leq 2 \times 10^5$, $1 \leq k \leq 2 \times 10^5$), indicating the number of sites and the number of subway lines.

The second line contains k integers $a_1, a_2, \cdots, a_k$ ($1 \leq a_i \leq 10^6$).

The third line contains k integers $b_1, b_2, \cdots, b_k$ ($1 \leq b_i \leq 10^6$).

For the following k lines, the i-th line first contains an integer p_i ($2 \leq p_i \leq n$), indicating the number of sites line i travels through. Then $(2p_i - 1)$ integers $x_{i,1}, w_{i,1}, x_{i,2}, \ldots, x_{i,p_i-1}, w_{i,p_i-1}, x_{i,p_i}$ follow ($1 \leq x_{i,j} \leq n$, $1 \leq w_{i,j} \leq 10^9$), where $x_{i,j}$ is the j-th site visited by line i, and $w_{i,j}$ is the travel time from site $x_{i,j}$ to site $x_{i,j+1}$ on line i. The sites traveled through by a subway line are distinct.

It is guaranteed that $\sum\limits_{i=1}^{k}(p_i - 1) \leq 2 \times 10^5$.

Output

Output one line containing $(n - 1)$ integers $d_2, d_3, \cdots, d_n$ separated by a space, where d_i is the minimum time cost from site 1 to site i.

Examples

standard input	standard output

6 3
1 5 1
5 5 1
3 1 2 2 3 3
3 5 1 2 1 4
3 3 4 5 4 6
	2 5 21 14 18

6 3
1 5 1
5 5 1
5 1 2 2 100 3 100 6 1 4
5 1 100 2 4 3 100 5 1 4
2 3 1 5
	2 31 43 37 136

Problem G. Binary Tree

Time limit:	1 second
Memory limit:	1024 megabytes

This is an interactive problem.

Given a binary tree with n vertices, your task is to find a special vertex s in the tree with at most $p = \lfloor \log_2 n \rfloor$ queries. That is to say, p is the largest integer such that $2^p \le n$.

Each query consists of two different vertices u and v. The interactor will output an integer t $(0 \le t \le 2)$ as the answer. Let $d(a, b)$ be the number of edges on the simple path from vertex a to vertex b.

- If $t = 0$, then vertex u is nearer to the special vertex. That is, $d(u, s) < d(v, s)$.
- If $t = 1$, then the distances from u and v to the special vertex are the same. That is, $d(u, s) = d(v, s)$.
- If $t = 2$, then vertex v is nearer to the special vertex. That is, $d(u, s) > d(v, s)$.

Note that the interactor is adaptive, meaning that the answer for each test case is not pre-determined. The interactor can determine the special vertex according to your queries, as long as its answer does not conflict with the previous queries and answers.

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 $(2 \le n \le 10^5)$ indicating the number of vertices in the binary tree.

For the following n lines, the i-th line contains two integers x_i and y_i $(0 \le x_i, y_i \le n)$, indicating the left and right child of the i-th vertex. If $x_i = 0$, then the i-th vertex has no left child; if $y_i = 0$, then the i-th vertex has no right child.

It is guaranteed that the sum of n for all test cases will not exceed 2×10^5.

Interaction Protocol

To ask a query, output one line. First output `?` followed by a space, then print two different integers u and v $(1 \le u, v \le n)$ separated by a space. After flushing your output, your program should read a single integer t indicating the answer to your query.

If you want to guess the special vertex, output one line. First output `!` followed by a space, then print an integer s $(1 \le s \le n)$ indicating the special vertex. After flushing your output, your program should continue processing the next test case, or exit immediately if there are no more test cases. Note that your guess does not count as a query.

To flush your output, you can use:

- `fflush(stdout)` (if you use `printf`) or `cout.flush()` (if you use `cout`) in C and C++.
- `System.out.flush()` in Java.
- `stdout.flush()` in Python.

Example

standard input	standard output
2	
5	
0 0	
1 5	
2 4	
0 0	
0 0	
	? 5 1
1	
	? 1 4
0	
	! 2
2	
0 2	
0 0	
	? 2 1
2	
	! 1

Problem H. Border Jump 2

Time limit:	2 seconds
Memory limit:	1024 megabytes

You are given a string S consisting of lower-cased English letters. You need to perform some operations on S until it becomes empty. Each time you can perform one of the following three operations:

1. Delete the first character of S.
2. Delete the last character of S.
3. Choose a good substring S' of S and replace S with S'.

A non-empty string S' is called a good substring of string S if and only if $S' \neq S$, S' is a prefix of S, and the reverse of S' is a suffix of S. The reverse of a string $p_1 p_2 \cdots p_k$ of length k is another string $p_k p_{k-1} \cdots p_1$ of length k.

What's the maximum number of type 3 operations can you perform?

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 and only line contains a string S ($1 \le |S| \le 10^5$) consisting of lower-cased English letters.

It is guaranteed that the sum of $|S|$ over all test cases does not exceed 2×10^5.

Output

For each test case, output one line containing one integer indicating the maximum number of type 3 operations you can perform.

Example

standard input	standard output
3 aaaa abbaabba xy	3 4 0

Note

For the first sample test case: `aaaa` $\xrightarrow{\text{op. 3}}$ `aaa` $\xrightarrow{\text{op. 3}}$ `aa` $\xrightarrow{\text{op. 3}}$ `a` $\xrightarrow{\text{op. 2}}$ $\varnothing$.

For the second sample test case: `abbaabba` $\xrightarrow{\text{op. 3}}$ `abbaabb` $\xrightarrow{\text{op. 1}}$ `bbaabb` $\xrightarrow{\text{op. 3}}$ `bbaab` $\xrightarrow{\text{op. 1}}$ `baab` $\xrightarrow{\text{op. 3}}$ `baa` $\xrightarrow{\text{op. 1}}$ `aa` $\xrightarrow{\text{op. 3}}$ `a` $\xrightarrow{\text{op. 1}}$ $\varnothing$.

Problem I. Bingo

Time limit:	1 second
Memory limit:	1024 megabytes

Given two integers n, m and an integer sequence $a_1, a_2, \cdots, a_{nm}$ of length $n \times m$, we're going to fill a grid of n rows and m columns with the integers from the sequence. More specifically, let (i, j) be the cell on the i-th row and the j-th column, we'll put the $((i-1) \times m + j)$-th element of the sequence (that is, $a_{(i-1)\times m+j}$) into that cell.

We say an integer k is a "bingo integer" of the sequence, if after filling all the cells, at least one of the two following conditions is satisfied.

- There is at least one row, where all integers in the cells of that row are less than or equal to k.
- There is at least one column, where all integers in the cells of that column are less than or equal to k.

It is easy to see that a sequence may have multiple bingo integers, however in this problem, we're only interested in the smallest bingo integer.

Calculate the sum of the smallest bingo integers for all $(nm)!$ permutations of the given sequence. As the answer may be large, output the answer modulo 998 244 353.

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 two integers n and m ($1 \le n, m \le 2 \times 10^5$, $1 \le n \times m \le 2 \times 10^5$), indicating the number of rows and columns of the grid.

The second line contains $n \times m$ integers $a_1, a_2, \cdots, a_{nm}$ ($0 \le a_i < 998\,244\,353$) indicating the given sequence.

It's guaranteed that the sum of $n \times m$ of all test cases will not exceed 4×10^5.

Output

For each test case, output one line containing one integer indicating the answer.

Example

standard input	standard output

4
2 2
1 3 2 4
3 1
10 10 10
1 3
20 10 30
3 4
1 1 4 5 1 4 1 9 1 9 8 10
	56
	60
	60
	855346687

Note

For the first sample test case, if 1 and 2 are not on the same row or column, then the smallest bingo integer will be 3, otherwise the smallest bingo integer will be 2. There are 8 permutations where 1 and 2 are not on the same row or column, so the answer is $8 \times 3 + (4! - 8) \times 2 = 56$.

For the second sample test case, the smallest bingo integer is always 10, so the answer is $3! \times 10 = 60$.

Problem J. Social Media

Time limit:	1 second
Memory limit:	1024 megabytes

On a social media platform, users can leave comments under others' posts to express their thoughts. However, these comments are not visible to everyone. Specifically, for user C to see user A's comments under user B's post, he/she has to be friends with both A and B at the same time. If a user leaves a comment under his/her own post, all his/her friends can see this comment.

As an active user on this platform, you would like to see as many comments as possible. There are k users (not counting you) on the platform, numbered from 1 to k. There are also m comments on the platform, but you might not be able to see them all because you only have n friends. As you need to participate in the 2024 ICPC Asia Nanjing Regional Contest, you don't have time to make too many new friends. What's the maximum number of comments you can see if you make at most two new friends on the platform?

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 three integers n, m, and k ($1 \le n \le k \le 2 \times 10^5$, $1 \le m \le 2 \times 10^5$) indicating the number of your friends, the number of comments, and the number of users (not counting you) on the platform.

The second line contains n distinct integers $f_1, f_2, \cdots, f_n$ ($1 \le f_i \le k$) indicating your friends on the platform.

For the following m lines, the i-th line contains two integers a_i and b_i ($1 \le a_i, b_i \le k$) indicating a comment written by user a_i under user b_i's post.

It's guaranteed that neither the sum of k nor the sum of m of all test cases will exceed 2×10^5.

Output

For each test case, output one line containing one integer, indicating the maximum number of comments you can see if you make at most two new friends on the platform.

Example

standard input	standard output

5
4 12 7
5 7 3 6
3 6
2 2
1 4
2 4
1 3
7 6
4 1
5 4
1 1
1 1
2 1
3 7
2 7 6
2 4
1 2
3 2
2 5
5 4
2 6
4 6
2 6
1 1 2
1
1 2
2 1 2
1 2
1 2
2 1 100
24 11
11 24
	9
	5
	1
	1
	1

Note

For the first sample test case, you can make friends with user 1 and 4.

For the second sample test case, you can make friends with user 5 and 6.

For the third sample test case, you can make friends with user 2.

For the fourth and fifth sample test cases, you don't need to make new friends because you can already see all comments.

Problem K. Strips

Time limit: 1 second
Memory limit: 1024 megabytes

There are w cells arranged in a row, numbered from 1 to w from left to right. Among the cells, n of them are red, m of them are black, and the remaining $(w - n - m)$ cells are white.

You need to cover all the red cells with some strips. Each strip must cover k continuous cells. Find a way to cover all red cells while satisfying all the following constraints:

- Each red cell is covered by a strip.
- No black cell is covered by a strip.
- No two strips cover the same cell. That is, each cell is covered by at most one strip.
- The number of strips used is as small as possible.

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 four integers n, m, k and w ($1 \le n, m \le 10^5$, $1 \le k \le w \le 10^9$, $n + m \le w$), indicating the number of red cells, the number of black cells, the length of each strip and the total number of cells.

The second line contains n integers $a_1, a_2, \cdots, a_n$ ($1 \le a_i \le w$), indicating that cell a_i is red.

The third line contains m integers $b_1, b_2, \cdots, b_m$ ($1 \le b_i \le w$), indicating that cell b_i is black.

It's guaranteed that the given $(n + m)$ cells are distinct. It's also guaranteed that neither the sum of n nor the sum of m of all test cases will exceed 2×10^5.

Output

For each test case:

If it is possible to cover all the red cells while satisfying all constraints, first output one line containing one integer c indicating the smallest number of strips used. Then output another line containing c integers $l_1, l_2, \cdots, l_c$ ($1 \le l_i \le w - k + 1$) separated by a space, where l_i is the left-most cell covered by the i-th strip. If there are multiple valid answers, you can output any of them.

If it is not possible to do so, just output `-1` in one line.

Example

standard input	standard output

4
5 2 3 16
7 11 2 9 14
13 5
3 2 4 11
6 10 2
1 11
2 1 2 6
1 5
3
2 1 2 6
1 5
2
	4
	6 2 14 9
	-1
	2
	1 4
	-1

Problem L. $P \oplus Q = R$

Time limit:	4 seconds
Memory limit:	1024 megabytes

Alice wants to train herself to solve constructive problems. So her friend Kei, a super artificial intelligence, generates the following problem for Alice.

Given an integer n, construct two permutations $P = p_1, p_2, \cdots, p_n$ and $Q = q_1, q_2, \cdots, q_n$ of $0, 1, \ldots, (n-1)$, such that the sequence $R = r_1, r_2, \cdots, r_n$ is still a permutation of $0, 1, \ldots, (n-1)$, where $r_i = p_i \oplus q_i$. Here $x \oplus y$ means the bitwise exclusive-or of x and y.

Alice solves this problem with her powerful calculating ability and she decides to share this problem with you. Can you solve it?

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 and only line contains one integer n ($1 \le n \le 2 \times 10^5$) indicating the length of the permutation.

It is guaranteed that the sum of n of all test cases will not exceed 2×10^6.

Output

For each test case:

If there exist two permutations satisfying the constraint, first output `Yes` in one line. Then output a second line containing n integers $p_1, p_2, \ldots, p_n$ separated by a space. Finally output a third line containing n integers $q_1, q_2, \ldots, q_n$ separated by a space. If there are multiple valid answers, you can output any of them.

If there do not exist two permutations satisfying the constraint, just output `No` in one line.

Example

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

Note

For the second test case, $R = \{3, 0, 1, 2\}$ is still a permutation of $0, 1, 2, 3$.

Problem M. Ordainer of Inexorable Judgment

Time limit:	1 second
Memory limit:	1024 megabytes

Neuvillette is the Chief Justice of Fontaine, known as the Iudex, and he is renowned for his unassailable impartiality. As a playable character in the world-famous game *Genshin Impact*, he is known for his powerful charged attack that can hit enemies within a specific range.

Since he is very powerful, many players use him while challenging almost every quest. However, not everybody in Teyvat is happy about this, especially other ADC (Attack Damage Carry) characters, including Kamisato Ayaka, Keqing, etc. Together, they decide to persuade Mihoyo to nerf Neuvillette in the game. To do so, they must submit a report about Neuvillette's total damage in several scenarios.

Created from Genshin Impact official material

Each battle scenario happens on a two-dimensional plane. Neuvillette stands on $(0, 0)$ facing (x_0, y_0) initially, making a charged attack which lasts for t units of time, and rotates 1 rad counter-clockwise per unit of time. That is to say, Neuvillette turns a circle counter-clockwise in 2π units of time.

Consider a ray from $(0, 0)$ towards the direction Neuvillette faces. The attack range is the set of points whose distance to the ray is at most d. If the target, whose shape is a convex polygon, has common points with the attack range, it will suffer 1 continued damage per unit of time.

As an experienced programmer, you are summoned by Ayaka. This time, your task is to calculate the damage the target incurs in the first t units of time.

Input

There is only one test case in each test file.

The first line contains five integers n, x_0, y_0, d, and t ($3 \le n \le 100$, $-10^4 \le x_0, y_0 \le 10^4$, $x_0^2 + y_0^2 > 0$, $1 \le d, t \le 10^4$).

For the following n lines, the i-th line contains two integers x_i and y_i ($-10^4 \le x_i, y_i \le 10^4$), indicating the coordinates of the i-th vertex of the convex polygon.

All n vertices are given in counter-clockwise order, and any three of them are not collinear. It is also guaranteed that the shape has no common points with the circle centered at $(0, 0)$ with radius d. That is to say, there does not exist a point inside or on the boundary of the convex polygon, while at the same time inside or on the boundary of the circle.

Output

Output one line containing one real number, indicating the damage the target incurs in the first t units of time.

Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}. Formally speaking, suppose that your output is a and the jury's answer is b, your output is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-6}$.

Examples

standard input	standard output

3 1 0 1 1
1 2
2 1
2 2
	1.000000000000

3 1 0 1 2
1 2
2 1
2 2
	1.570796326795

3 1 0 1 10000
1 2
2 1
2 2
	2500.707752257475

Note

The figure below simultaneously shows the initial state of the sample test cases.