The 2019 ICPC Caribbean National Contests

Real Contest Problem Set

Document composed by 12 pages (not including this cover)

Problem set developers

Marcelo Fornet (NEAR, Cuba)

Carlos Joa (INTEC, República Dominicana)

Carlos Toribio (Google, República Dominicana)

Elio Govea (UPR, Cuba)

José Carlos Gutiérrez (Universität Bremen, Cuba)

Rafael Fernández (Cuba)

Ariel Cruz (UH, Cuba)

Norge Vizcay (Cuba)

Daniel Otero (Universität Bremen, Cuba)

Leandro Castillo (Dropbox, USA)

Roberto Abreu (República Dominicana)

Aurora Gil (Cuba)

Reynaldo Gil (Datys, Cuba)

Dovier Ripoll (UCI, Cuba)

October 5th, 2019

Problem A. Guess the number

Randy is playing a simple game called "Guess the Number": there is a hidden integer V whose value is between 1 and N. The goal of the game is to guess this number V. In each round, Randy picks an integer between 1 and N as his guess. If the guess matches V, the game ends and he wins a prize. Otherwise, the game continues and asks Randy for his next guess, and so on.

In each round, Randy will use the following strategy to make his guess:

1. He randomly selects an integer x between 1 and N.
2. If x was not tried in any previous round, x would be his guess in the current round. This guess costs him $1.
3. Otherwise, as the guess in the current round, he would pick the smallest not tried integer that is larger than x (but not larger than N). This guess also costs him $1.
4. If no such number exists, he discards x and goes to step 1. Discarding number x has no cost for Randy.

Question: what is the average amount of money Randy has to pay before he wins the prize?

Input

The input consists of a single line containing two integers N ($1 \le N \le 20$) and V ($1 \le V \le N$).

Output

Output a single line with the answer to the question described above. Your answer is considered correct if its (absolute or relative) difference from the expected answer is at most 10^{-6}.

Example

standard input	standard output
2 2	1.5
3 2	1.944444444
10 6	5.305939329

Problem B. Which permutation

Juan likes permutations. He loves them so much that, whenever he is given a sequence of numbers, he generates a set with all possible permutations of that sequence. Now he wants to use the Bucket Sort algorithm to sort this huge set of permutations. To do so, he needs to know, for a given permutation P, how many permutations are lexicographically smaller than P.
Once he learns how to do this computation efficiently, he will use parallel processing and will be able to sort the huge set of permutations in no time. So he trusted you with this task: given a sequence P of integers (not necessarily distinct), determine how many permutations of this sequence exist that are lexicographically smaller than P.

A permutation of a sequence S is any rearrangement of the elements of S.

Let P and Q be two permutations of the same sequence of numbers. We say P is lexicographically smaller than Q if $P_i < Q_i$ where i is the first index in which permutations P and Q differ. For example, permutation $(3, 2, 9, 2, 10)$ is lexicographically smaller than $(3, 2, 10, 2, 9)$.

In the first sample test case, the six permutations that are lexicographically smaller than $(3, 1, 2, 3)$ are: $(1, 2, 3, 3)$, $(1, 3, 2, 3)$, $(1, 3, 3, 2)$, $(2, 1, 3, 3)$, $(2, 3, 1, 3)$, and $(2, 3, 3, 1)$.

Input

The first line of the input contains an integer N $(1 \leq N \leq 200,000)$. The next line has N positive integers P_i $(1 \leq P_i \leq 10^9)$ representing the sequence P.

Output

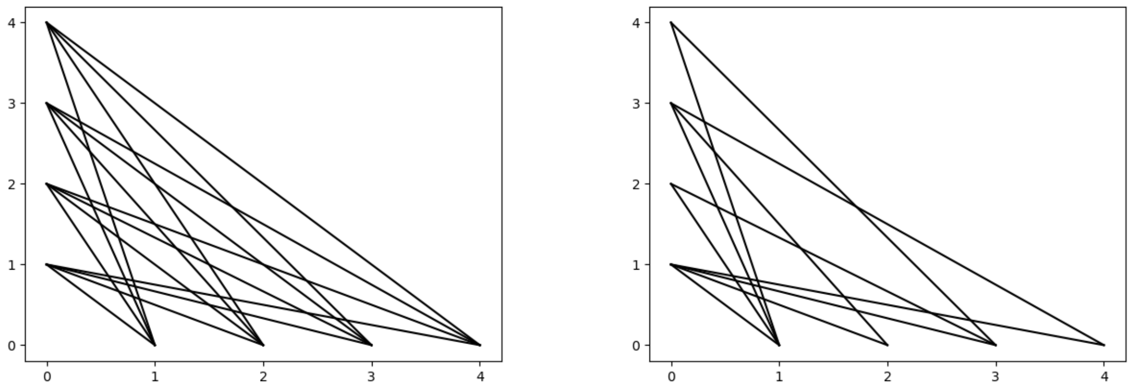
Print the number of permutations of sequence P that are lexicographically smaller than P, modulo prime number 998244353.

Example

standard input	standard output
4 3 1 2 3	6
13 13 1 2 3 4 5 6 7 8 9 10 11 12	756797435

Problem C. Straight Lines

You have every point with positive integer coordinates not greater than N over the x and y axis available and you want to draw as many straight lines as you can, with the condition that every line must pass through one of those points on each axis and there are no parallel lines. How many lines can you draw? How many ways can you draw these lines to achieve this maximum?



These two images correspond to sample case where $N = 4$:

1. The image on the left shows all the lines you could draw if there were no restriction over parallel lines.
2. The image on the right shows one way to draw 11 lines where no parallel lines are allowed.

Input

A line with an integer N ($1 \leq N \leq 10^9$).

Output

A line with two integers c and w, representing the maximun number of lines you can draw and the number of ways to do it respectively, modulo the prime 998244353.

Example

standard input	standard output
1	1 1
4	11 16

Problem D. Bridge

Fito has a new box of colored cubes. Since he is so eager to play with them, he wishes to build a bridge by placing all cubes in a row. However, to build an elegant bridge, there must not exist two consecutive cubes of the same color. As if this were not enough, Fito decided that the first and last cubes on the bridge have specific colors. Your task is to help Fito determine a distribution of the cubes to build the bridge.

Input

First, there is a line with an integer t denoting the number of test cases. Every test case consists of two lines. The first line has four integers n, k, p and q ($1 \leq n, k \leq 10^5$, $1 \leq p, q \leq k$), the number of cubes, the number of colors, the color of the first and the last cube in the row, respectively. The second line contains n integers a_i ($1 \leq a_i \leq k$), the color of each cube. It is guaranteed that the total number of cubes among all testcases is at most 10^5.

Output

For every test case, print `"No"` (without quotes) if there is no way to distribute the cubes to build the bridge that satisfies Fito constraints, or `"Yes"` (without quotes) otherwise. Additionally, if there is a solution, print n integers (the colors) in the order that Fito must place each block to build the bridge. Any valid answer will be accepted. **Note that ALL cubes must be used.**

Example

standard input	standard output
3 1 1 1 1 1 8 3 3 1 2 2 2 2 3 3 1 1 8 3 3 1 3 3 3 2 2 2 1 1	Yes 1 No Yes 3 2 3 2 3 1 2 1

Problem E. Xela

Xela is a new programming language designed to compute algebraic operations with nested loops extremely fast. Your task is to build the interpreter of such language. Language specifications:

1. There are 26 variables in *Xela*, one for each lower case English character from a to z. Initially, all variables start at 0.
2. All algebraic operations are taken modulus prime 998244353.
3. All scalars in the code are in the range $[0, 998244353)$.
4. *Assignment*: Instruction that assigns to a variable an scalar or the value of another variable. Example: $a = 42$ (Assigns value 42 to a) or $z = x$ (Assigns current value of x to z).
5. *Addition*: Instruction that adds to any variable an scalar or the value of another variable. Example $a \mathrel{+}= 31415$ (Adds value 31415 to a) or $y \mathrel{+}= a$ (Adds value from a to y).
6. *Subtraction*: Instruction that substracts from any variable an scalar or the value of another variable. Example $a \mathrel{-}= 1$ (Subtracts value 1 from a) or $v \mathrel{-}= u$ (Subtracts value in u from v).
7. *Multiplication*: Instruction that multiplies a variable by a scalar *(multiply to another variable is not allowed)*. Example $a \mathrel{*}= 2$ (Multiplies a times 2 and stores value in a).
8. *Loops*: Blocks of code that contain a repeat parameter which is always a scalar in the range $[1, 998244353)$ *(variables are not allowed as loops parameters)*. All the code inside the loop is executed *repeat* number of times. Indentation inside loops is using 4 spaces (see examples for a better understanding of the syntax). Inside every block, there is at least one instruction. Multiple instructions in the same block will have the same indentation.
9. All lines in the program should be valid instructions or loop blocks.

Given a code written in *Xela* calculate the last value of every variable.

Input

A valid *Xela* program with at most 100 lines.

Output

For every variable whose value is different from 0, print its value in a separate line using the format "*variable* = *value*" without quotes. Variables must be listed in alphabetical order. If all values are 0 at the end of the program print `"INITIAL STATE"` without quotes.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>a = 1
b = 1
loop 10
    c = a
    a += b
    b = c
    c = 0</pre></td><td><pre>a = 144
b = 89</pre></td></tr>
<tr><td><pre>a = 0
loop 100000000
    c = 1
    d = a
    a -= c
    loop 100000000
        c *= 2
        d += c
    d *= 3
    d -= 1
    a += d</pre></td><td><pre>a = 916538251
c = 113003797
d = 108226118</pre></td></tr>
<tr><td><pre>x = 17
loop 23
    x *= 2
x *= 7
x += 1</pre></td><td><pre>INITIAL STATE</pre></td></tr>
</table>

Problem F. Dates

Rosita wrote in a notebook the dates (day and month) of k consecutive days and stored this notebook in a box. After a few years Rosita found her notebook, but sadly the months she wrote were unreadable. Rosita wants to know all possible months of the last date she wrote in the notebook.

Input

A line with an integer k ($1 \le k \le 300$) denoting how many dates were written. Next line contains k integer numbers: the days of every date in the same order they were written. These days are guaranteed to be from a valid sequence of consecutive dates.

Output

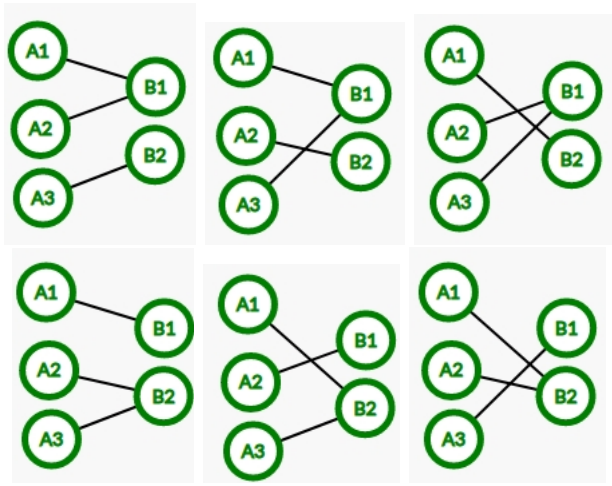
Print the number of possible months of the last date Rosita wrote in her notebook. In the next lines print the possible months, one per line, in calendar order.

Example

standard input	standard output
10 1 2 3 4 5 6 7 8 9 10	12 January February March April May June July August September October November December
2 31 1	7 January February April June August September November
11 20 21 22 23 24 25 26 27 28 29 1	1 March

Problem G. How many

Given two sets A and B that have n and m elements respectively with $n \geq m$, determine how many ways it is possible to match all the elements so that each element of A is paired with exactly one element of B and each element of B is paired with at least one element of A. Below are all the possible assignments for the first sample.



Input

Input consists of a single line with n ($1 \leq n \leq 10^{18}$) and m ($1 \leq m \leq 10^5$) separated by a space between them.

Output

Print the number of ways in which the elements of the sets can be matched as requested in the problem. Since this number can be quite large, print the remainder of the division of that number with prime 998244353

Example

standard input	standard output
3 2	6
10 3	55980
314159265358979323 84626	683366796

Problem H. Colors I

The difference between this problem and "Colors II" is in bold.

There are n points on the plane. Each of these points has been assigned a certain color, but not that all points are of the same color. Determine the largest possible (Euclidean) distance between any pair of different-colored points. **All these points lie on a straight line.**

Input

The first line of the input contains a single integer n ($2 \leq n \leq 250000$). Each of the following n lines describes a single point. A point's description consists of three integers x_i, y_i and c_i ($-10^9 \leq x_i, y_i \leq 10^9, 1 \leq c_i \leq n$) separated by single spaces; the numbers x_i and y_i are the i-th point's coordinates, and c_i is the id of the point's color.

There are at least two different colors and no two points have the same coordinates.

It is guaranteed that all n points lie on a straight line.

Output

The first and only line of the output should contain the square of the maximum distance between two points of different colors.

Example

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

Problem I. Colors II

There are n points on the plane. Each of these points has been assigned a certain color, but not that all points are of the same color. Determine the largest possible (Euclidean) distance between any pair of different-colored points.

Input

The first line of the input contains a single integer n ($2 \leq n \leq 250000$). Each of the following n lines describes a single point. A point's description consists of three integers x_i, y_i and c_i ($-10^9 \leq x_i, y_i \leq 10^9, 1 \leq c_i \leq n$) separated by single spaces; the numbers x_i and y_i are the i-th point's coordinates, and c_i is the id of the point's color.

There are at least two different colors and no two points have the same coordinates.

Output

The first and only line of the output should contain the square of the maximum distance between two points of different colors.

Example

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

Problem J. Anagram

Two strings are anagrams of each other if the letters of one string can be rearranged to form the other string. Given a string s, find the number of pairs of substrings of s which are anagrams of each other.

For example $s = wow$ the list of all anagrammatic pairs is $[w, w], [wo, ow]$ at positions $([0..0], [2..2])$ and $([0..1], [1..2])$ respectively.

Input

A single line containing a string s to analyze. String s only contains lowercase English characters $[a..z]$ and $|s| \leq 100$.

Output

Print the number of anagrammatic pairs in the given string.

Example

standard input	standard output
abba	4
abcd	0

Problem K. White and black

You are given n sets with k different numbers each. Every pair of sets are different (i.e., there is at least one number that is in one set which is not in the other set). Every set must be colored either white or black in such a way that it is possible to solve the following puzzle:

One person secretly chooses one of the sets and reveals its color and $k - 1$ arbitrary numbers from this set. Another person must use this information to guess which is the missing number. Notice that the second person knows the color and the content of every set in advance.

The problem is to determine if there exists a way to color the sets such that the puzzle has a solution independently of the actions from the first person (i.e., the second person can always determine the missing number).

Input

The first line has two integers: n and k $(1 \leq n \leq 100, 1 \leq k \leq 10)$, the number of sets and total numbers per set.

Each of the following n lines has k numbers v_i $(1 \leq v_i \leq 100)$, the elements of each set.

Output

On the first line, if there exists a solution, print the word `"Yes"` (without quotes); otherwise print the word `"No"` (without quotes).

If there is a solution, print a string on the second line with n characters where the i-th character is 'W' if the set i should be colored in white and 'B' if it should be colored in black. Any valid solution will be accepted.

Example

standard input	standard output
2 3 1 2 3 2 3 4	Yes WB
2 3 1 2 3 3 4 5	Yes WW
3 3 1 2 3 2 3 4 3 4 1	No