Problem A. Almost Prefix Concatenation

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

A string $A = a_1a_2\cdots a_n$ of length n is a concatenation of n characters $a_1, a_2, \ldots, a_n$, and its length is denoted by $|A|$. Similarly, the concatenation of two strings $A = a_1a_2\cdots a_n$ and $B = b_1b_2\cdots b_m$ is $a_1a_2\cdots a_nb_1b_2\cdots b_m$, denoted by $A + B$.

The edit distance between two strings $A = a_1a_2\cdots a_n$ and $B = b_1b_2\cdots b_n$ of the same length n is the number of indices i such that $a_i \neq b_i$.

We call the string A formed by the first k characters of another string B ($k \leq |B|$) as the k-th prefix of B, and a string P as an almost prefix of another string Q if $|P| \leq |Q|$ and the edit distance between P and the $|P|$-th prefix of Q is at most 1.

Given two strings S and T consisting of lowercase English letters, you are asked to find all ways to split S into many parts such that each part is a non-empty almost prefix of string T, and then report the sum of the squared number of parts of all ways in modulo 998244353. More formally, let $S = P_1 + P_2 + \ldots + P_n$ be a possible way, you are asked to calculate

$$\left(\sum_{\substack{S=P_1+P_2+\ldots+P_n \\ \forall_{i=1,2,\ldots,n} P_i \text{ is an almost prefix of } T}} n^2 \right) \bmod 998\,244\,353.$$

Input

The first line of the input contains a string S ($1 \leq |S| \leq 1\,000\,000$), consisting of only lowercase English letters.

The next line contains a string T ($1 \leq |T| \leq 1\,000\,000$), consisting of only lowercase English letters.

Output

Print a single line containing a single integer: the sum of the squared number of parts of all ways in modulo $998\,244\,353$.

Examples

standard input	standard output
`ababaab` `aba`	`473`
`ac` `ccpc`	`5`

Note

In the first sample case ($S =$ `ababaab`, $T =$ `aba`), there are 19 ways to split:

- 1 way of 3 parts, which is `ab` + `aba` + `ab`;
- 6 ways of 4 parts, such as `a` + `b` + `aba` + `ab`;
- 7 ways of 5 parts, such as `a` + `b` + `ab` + `a` + `ab`;
- 4 ways of 6 parts, such as `a` + `b` + `a` + `b` + `a` + `ab`;
- 1 way of 7 parts, which is `a` + `b` + `a` + `b` + `a` + `a` + `b`.

Therefore, the result for the first sample case is $(3^2 + 6 \times 4^2 + 7 \times 5^2 + 4 \times 6^2 + 7^2) \bmod 998244353 = 473$.

Problem B. Palindromic Beads

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2.5 seconds
Memory limit:	512 megabytes

You are going to explore a cave full of beautiful beads. The cave consists of n rooms labeled by $1, 2, \ldots, n$, connected by $(n-1)$ bidirectional tunnels like a tree. In the i-th room, there is exactly one bead colored c_i.

The cave is not owned by you, so your actions are restricted. Before your entrance to the cave, you need to select two rooms x and y (maybe $x = y$), and submit your application to the owner. Then you can enter the cave at the x-th room, and walk along the shortest path from the x-th room to the y-th room. Every time you visit a new room, including the first room x and the last room y, you can choose to either pick the bead in that room up or not. Finally, you need to concatenate all the beads that you picked up into a string according to the order you picked them up. You must ensure the colors of each bead in the string is palindromic, otherwise you are not allowed to take these beads back home. Note that a string is palindromic if and only if it reads the same from left to right and vice versa.

You want to make the palindromic string as long as possible. What's the maximum possible number of beads that you can take away?

Input

The first line of the input contains a single integer n ($1 \leq n \leq 200\,000$), denoting the number of rooms.

The second line contains n integers $c_1, c_2, \ldots, c_n$ ($1 \leq c_i \leq n$), denoting the color of the bead in each room. **It is guaranteed that each color will appear at most twice.**

Each of the next $(n-1)$ lines contains two integers u_i and v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$), denoting a bidirectional tunnel between the u_i-th room and the v_i-th room. It is guaranteed that the roads form a tree.

Output

Print a single line containing an integer, denoting the number of beads that you can take away.

Examples

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

Problem C. Clique Challenge

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

A clique of a graph G is a set X of vertices of G with the property that every pair of distinct vertices in X are adjacent in G. You are given an undirected graph G with n vertices and m edges, please find the number of distinct non-empty cliques of graph G.

Input

The first line of the input contains two integers n and m ($1 \le n, m \le 1\,000$), denoting the number of vertices and the number of edges.

Each of the following m lines contains two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$), describing an undirected edge between the u_i-th vertex and the v_i-th vertex.

It is guaranteed that there will be at most one edge between each pair of different vertices.

Output

Print a single line containing an integer, denoting the number of cliques. Note that the answer may be extremely large, so please print it modulo $(10^9 + 7)$ instead.

Examples

standard input	standard output
3 2 1 2 2 3	5
3 3 1 2 1 3 2 3	7

Note

In the first example, cliques are $\{1\}$, $\{2\}$, $\{3\}$, $\{1, 2\}$ and $\{2, 3\}$.

In the second example, cliques are $\{1\}$, $\{2\}$, $\{3\}$, $\{1, 2\}$, $\{1, 3\}$, $\{2, 3\}$ and $\{1, 2, 3\}$.

Problem D. Discrete Fourier Transform

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

Given a sequence of integer $f_0, f_1, \ldots, f_{n-1}$, the discrete Fourier transform gives a sequence of complex numbers $F_0, F_1, \ldots, F_{n-1}$ that

$$F_t = \sum_{s=0}^{n-1} f_s \; e^{-2\pi i s t/n}$$

for each $t = 0, 1, \ldots, n-1$, where $e^{i\theta} = \cos\theta + i\sin\theta$, and i is the imaginary unit that $i^2 = -1$.

You may reset f_k to any integer value to minimize the maximum value among $|F_0|, |F_1|, \ldots, |F_{n-1}|$, where $|z| = |p + qi| = \sqrt{p^2 + q^2}$ ($p, q \in \mathbb{R}$) is the modulus of the complex number z.

Input

The first line contains two integers n ($1 \le n \le 2\,000$) and k ($0 \le k < n$).

The second line contains n integers $f_0, f_1, \ldots, f_{n-1}$ ($-2\,000 \le f_i \le 2\,000$).

Output

Output a line containing a single real number, indicating the minimum of the maximum value among $|F_0|, |F_1|, \ldots, |F_{n-1}|$ after resetting f_k to any integer value.

Your answer is acceptable if its absolute or relative error does not exceed 10^{-9}. Formally speaking, suppose that your output is a and the jury's answer is b, your output is accepted if and only if $\frac{|a-b|}{\max\{1,|b|\}} \le 10^{-9}$.

Example

standard input	standard output
`3 2` `1 1 0`	`2.0`

Problem E. Robot Experiment

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

You are developing a type of lunar exploring robot. Now the robot is under experiment on a large field. The field can be regarded as an infinite 2D plane. Initially, the robot is at $(0, 0)$. Then it will execute n commands sequentially. Each command can be represented as a single upper-case English letter:

- "`L`": Move from (x, y) to $(x - 1, y)$.
- "`R`": Move from (x, y) to $(x + 1, y)$.
- "`D`": Move from (x, y) to $(x, y - 1)$.
- "`U`": Move from (x, y) to $(x, y + 1)$.

Note that there are some obstacles on the field except the origin point $(0, 0)$. When the robot is trying to move to the next position, it will check whether there is an obstacle at the target coordinate. If there is an obstacle, the robot will stay at the current position, mark the current command as executed, and continue executing the following commands.

You have sent n commands to the robot, but unfortunately, due to some bugs, the map of the field and the position of the robot can not be transmitted to you. Assume the robot has finished executing all the n commands. Before going to the field, you are wondering which places may you find the robot. Please write a program to find all the possible positions the robot will finally locate at.

Input

The first line of the input contains a single integer n ($1 \le n \le 20$), denoting the number of commands.

The second line contains a string s of length n ($s_i \in \{$"`L`", "`R`", "`D`", "`U`"$\}$), the i-th letter denoting the i-th command sent to the robot.

Output

First output a single line containing an integer k, denoting the number of possible positions. Then output k lines, each line contains two integers x and y, denoting the robot can finally locate at (x, y). When $k \ge 2$, you should print the answers in ascending order of x, and then in ascending order of y in case of a tie.

Examples

standard input	standard output
2 RU	4 0 0 0 1 1 0 1 1
4 LRUD	4 0 -1 0 0 1 -1 1 0

Problem F. Flying Ship Story

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	4 megabytes

The memory limit of this problem (4 megabytes) is unusual!

There are many islands separated by the Byte Ocean. Clever people invented the flying ship such that they can travel to other lands. Today, millions of salesmen gather at Byteland to celebrate the annual international trading festival. You are maintaining the trading database here. Initially, the database is empty, then q events will happen, each is one of the following two types:

- "`1 x y w`" ($1 \le x, y, w \le 10^9$): A new good is on sale. It is from the x-th island, its type is numbered as y, and its price is w. Note that two goods can share the same type but from different islands.
- "`2 x y`" ($1 \le x, y \le 10^9$): It is a query event. A salesman from the x-th island with the y-th type of good is searching for other types of goods from other islands. You need to report the price of the most expensive good, which is not from the x-th island, and the type of which is not y. When there is no finding, please report "`0`" instead. This is only for searching, the good that you find will not be deleted from the database.

Due to the low level of technology in Byteland, the space of the database is very limited, but you still need to maintain the database efficiently.

Input

The first line of the input contains a single integer q ($1 \le q \le 1\,000\,000$), denoting the number of events.

Each of the next q lines describes an event in formats described in the statement above, except that some parameters are encrypted in order to enforce online processing.

Let *last* be the previous price that you answered. Note that *last* should be reset to 0 at the beginning of the input. For each operation, x, y and w are encrypted: their actual values are $x \oplus last$, $y \oplus last$ and $w \oplus last$. In the expressions above, the symbol "$\oplus$" denotes the bitwise exclusive-or operation. Also, note that the constraints described in the statement above apply to the corresponding parameters only after decryption, the encrypted values are not subject to those constraints.

Output

For each query, print a single line containing a single integer, denoting the price of the good that you find, or zero if there is no finding.

Example

standard input	standard output
5 1 2 3 1 1 4 5 2 2 2 2 2 3 7 2 3 4	2 1 0

Problem G. GCD of Pattern Matching

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

For any positive integer x, its m-based representation is a string of digits $d_{n-1}d_{n-2}\cdots d_1d_0$ where $x = \sum_{i=0}^{n-1} d_i m^i$, $0 < d_{n-1} < m$, and $\forall_{i=0,1,\dots,n-2} 0 \le d_i < m$.

Let Σ be the set of all possible characters. We call that a string $S = s_1s_2\cdots s_n$ matches with a pattern $P = p_1p_2\cdots p_n$ if and only if there exists a mapping function $f : \Sigma \to \Sigma$ such that $\forall_{i=1,2,\dots,n} f(s_i) = p_i$ and $\forall_{a,b\in\Sigma,a\ne b} f(a) \ne f(b)$.

Given an integer m and a pattern P consisting of lowercase English letters, find all positive integers in m-based representation that match the pattern, and report their greatest common divisor (GCD) in 10-based representation.

It is guaranteed for each test case that there always exists at least one integer whose m-based representation matches the pattern.

Input

The first line of the input contains a single integer T ($1 \le T \le 500\,000$), denoting the number of test cases.

Each of the following T lines describes a test case and contains an integer m and a string P ($2 \le m \le 16$, $1 \le |P| \le 16$), separated by a single space.

Output

For each of the T test cases, print a single line containing a single integer: the GCD of all matched positive integers (in 10-based representation).

Example

standard input	standard output
5 10 ccpcccpc 10 cpcpcp 10 cpc 4 cpccpc 4 dhcp	10001 10101 1 65 3

Note

For the last sample case, all integers of length 4 with no duplicate digits in 4-based representation can match `dhcp`, whose digits have a constant sum $0+1+2+3=6$ (e.g. 1023, 1302, 3210). Together with $\sum_{i=0}^{n-1} d_i 4^i \equiv \sum_{i=0}^{n-1} d_i \pmod 3$ and $\gcd(1023, 3210) = 3$, we can conclude the answer is 3.

Problem H. Hurricane

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

Byteland has n cities numbered from 1 to n, and $\frac{n(n-1)}{2}$ two-way roads connecting all pairs of cities directly. However, m roads become impassable due to the recent severe hurricane.

To assess the disaster damage, you are asked to calculate the number of pairs of cities that the minimum number of roads on the paths containing only passable roads between the two cities is exactly k, for each $k = 1, 2, \ldots, n-1$. Note that the pairs of cities that are not connected with passable roads should not be counted.

Input

The first line contains two integers n and m ($2 \le n \le 100\,000$, $0 \le m \le \min\left(\frac{n(n-1)}{2}, 200\,000\right)$) indicating the number of cities and the number of impassable roads in Byteland.

Each of the following m lines contains two integers u and v ($1 \le u < v \le n$) indicating that a road connecting cities u_i and v_i becomes impassable. It is guaranteed that each road appears at most once.

Output

Output a single line containing $(n-1)$ integers, the k-th of which indicates the number of pairs of cities that the minimum number of roads on the paths between the two cities is exactly k.

Examples

standard input	standard output
4 2 1 2 3 4	4 2 0
4 6 1 2 1 3 1 4 2 3 2 4 3 4	0 0 0

Problem I. Monster Generator

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

You are playing a computer game fighting against monsters. To become the master in the game, you decide to train yourself today and in the following m days using the monster generator.

The generator will generate n monsters every day for training. Assume it is the k-th ($0 \le k \le m$) day now, you need to assign s_k health points (HP) before fighting against monsters. Then you need to beat all the n monsters generated by the generator. You can fight against these monsters in an arbitrary order that you like. During the fight with the i-th monster, the player will lose $a_i + \Delta a_i \times k$ HP. And when the player finally beats the monster, the player will be awarded $b_i + \Delta b_i \times k$ HP. Note that when HP becomes negative (< 0), the training will fail, so never let this happen. Note that no matter how many HP you have, you still need to reassign your initial HP in the beginning of the next day's training.

The less initial HP you need, the more skilled you are. What's the minimum possible amount of total initial HP $s_0 + s_1 + s_2 + \cdots + s_{m-1} + s_m$ that can be achieved?

Input

The first line of the input contains two integers n and m ($1 \le n \le 100$, $0 \le m \le 10^{18}$).

Each of the next n lines contains four integers $a_i, \Delta a_i, b_i$ and Δb_i ($1 \le a_i, \Delta a_i, b_i, \Delta b_i \le 10^{15}$), denoting a monster.

Output

Print a single line containing an integer, denoting the minimum amount of total initial HP. Note that the answer may be extremely large, so please print it modulo 2^{64} instead.

Example

standard input	standard output
3 5 3 1 5 2 4 2 1 3 1 9 100 1	113

Problem J. Find the Gap

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

You are given n points in the 3D space. Please find two parallel planes such that all the n points are inside the gap of the two parallel planes, and the length of the gap is minimized.

Input

The first line of the input contains a single integer n ($1 \le n \le 50$), denoting the number of points.

Each of the following n lines contains three integers x_i, y_i and z_i ($1 \le x_i, y_i, z_i \le 10\,000$), describing a point (x_i, y_i, z_i). It is guaranteed that all the n points are pairwise distinct.

Output

Print a single line containing a single real number: the minimum possible length of the gap with an absolute or relative error of at most 10^{-9}.

Precisely speaking, assume that your answer is a and the jury's answer is b. Your answer will be considered correct if and only if $\frac{|a-b|}{\max\{1,|b|\}} \le 10^{-9}$.

Examples

standard input	standard output
8 1 1 1 1 1 2 1 2 1 1 2 2 2 1 1 2 1 2 2 2 1 2 2 2	1.000000000000000
5 1 1 1 1 2 1 1 1 2 1 2 2 2 1 1	0.707106781186548

Problem K. Sequence Shift

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2.5 seconds
Memory limit:	512 megabytes

You are given two sequences of length n: $[a_1, a_2, \ldots, a_n]$ and $[b_1, b_2, \ldots, b_n]$. The value of $f(a, b)$ is defined as $f(a, b) = \max\{a_i + b_i\}$, where $1 \le i \le n$.

The sequence b can be shifted. You will then be given q operations, each operation can be divided into the following two steps:

- First, shift the sequence b to the left by one position, and drop the first element, so the sequence b' will be $[b'_1 = b_2,\ b'_2 = b_3,\ \ldots,\ b'_{n-1} = b_n]$.
- Then, append v to the rightmost place of b, so the sequence b' will be $[b'_1 = b_2,\ b'_2 = b_3,\ \ldots,\ b'_{n-1} = b_n,\ b'_n = v]$.

In this problem, your task is to figure out the value of $f(a, b)$ before/after each operation.

Input

The first line of the input contains two integers n and q ($1 \le n \le 1\,000\,000$, $0 \le q \le 1\,000\,000$), denoting the length of the sequences and the number of operations.

The second line contains n integers $a_1, a_2, \ldots, a_n$, denoting the sequence a.

The third line contains n integers $b_1, b_2, \ldots, b_n$, denoting the initial sequence b.

Each of the next q lines contains a single integer v, denoting the value that will be appended in each operation. The value of v will be encrypted in order to enforce online processing.

It is guaranteed that all the values of a_i, b_i and v are chosen uniformly at random from integers in the range $[1, 10^9]$. The randomness condition does not apply to the sample test(s), but your solution must pass the sample test(s) as well.

Let *last* be the previous value of $f(a, b)$ that you answered. For each operation, the actual value of v is $v \oplus last$. In the expressions above, the symbol "$\oplus$" denotes the bitwise exclusive-or operation. Also, note that the constraints described in the statement above apply to the corresponding parameters only after decryption, the encrypted values are not subject to those constraints.

Output

Print $q + 1$ lines.

Output a single integer in the first line, denoting the initial value of $f(a, b)$.

In the k-th line ($2 \le k \le q + 1$), output a single integer denoting the current value of $f(a, b)$ after the $(k - 1)$-th operation.

Example

standard input	standard output
5 3 1 4 3 2 5 7 5 8 3 2 3 6 4	11 13 16 25

Problem L. Partially Free Meal

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

A new restaurant is opened in Byteland. To attract more customers, the meal is partially free. Specifically, there are n types of dishes on sale, labeled by $1, 2, \ldots, n$. Each dish can not be ordered more than once. For the i-th dish, its basic price is a_i dollars, and its event price is b_i dollars. Assume you have ordered k dishes $p_1, p_2, \ldots, p_k$ ($1 \leq p_i \leq n, p_i < p_{i+1}$), the total amount of dollars that you need to pay for is:

$$\sum_{i=1}^{k} a_{p_i} + \max_{i=1}^{k} \{b_{p_i}\}$$

You are a customer at this restaurant, you decide to order exactly k dishes, what's the minimum possible amount of dollars that you need to pay for?

Input

The first line of the input contains a single integer n ($1 \leq n \leq 200\,000$), denoting the number of dishes.

Each of the following n lines contains two integers a_i and b_i ($1 \leq a_i, b_i \leq 10^9$), denoting the basic price and the event price of each dish.

Output

Print n lines, the k-th ($1 \leq k \leq n$) of which containing an integer, denoting the minimum possible amount of dollars that you need to pay for when you order exactly k dishes.

Example

standard input	standard output
3 2 5 4 3 3 7	7 11 16