The 46th International Collegiate Programming Contest
Asia Shenyang Regional Contest

November 21

Problems

A A Bite of Teyvat
B Bitwise Exclusive-OR Sequence
C Cards of Magic
D Cross the Maze
E Edward Gaming, the Champion
F Encoded Strings I
G Encoded Strings II
H Line Graph Matching
I Linear Fractional Transformation
J Luggage Lock
K Matrix Operations
L Perfect Matchings
M String Problem

Do not open before the contest has started.

Problem A. A Bite of Teyvat

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

Xiangling, one of the greatest chef in Teyvat, is preparing for the Moonchase banquet. Xiangling has bought n round plates and her friend and companion Guoba will help place these n plates on the table in a line. The i-th plate placed has radius r_i and the center of this plate locates at $(x_i, 0)$ on the table.

However, Paimon the emergency food has been tired of waiting for the banquet a long time and begins finding the total area covered by the plates on the table after each placement.

Pixiv ID: 93526437

Input

The first line contains an integer n $(1 \le n \le 10^5)$, indicating the number of plates Xiangling has bought.

Then follow n lines, the i-th of which contains two integers x_i $(-10^5 \le x_i \le 10^5)$ and r_i $(1 \le r_i \le 10^6)$, indicating that the i-th plate placed by Guoba has radius r_i and the center of this plate locates at $(x_i, 0)$ on the table.

Output

Output n lines, the i-th of which contains a real number, indicating the total area covered by the plates on the table after Guoba places the first i-th plates.

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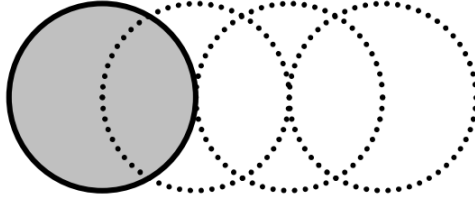
Example

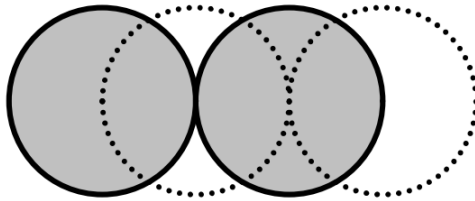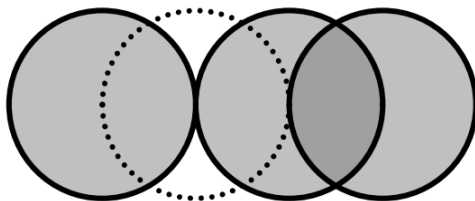| standard input	standard output
4	3.141592653589793
0 1	6.283185307179586
2 1	8.196408262160623
3 1	8.881261518532902
1 1	

Note

In the sample case:

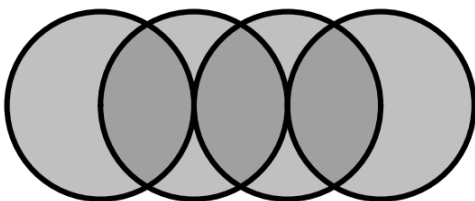1. The total area covered by the first plate is π;

2. The total area covered by the first two plates is 2π;

3. The total area covered by the first three plates is $\frac{14\pi+3\sqrt{3}}{6}$;

4. The total area covered by all the four plates is $\frac{4\pi+3\sqrt{3}}{2}$.

Problem B. Bitwise Exclusive-OR Sequence

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

AAA has a sequence $a_1, a_2, \ldots, a_n$ with m constraints, each of which is described as $a_u \oplus a_v = w$, where $\oplus$ denotes the bitwise exclusive-OR operation.

More precisely, the bitwise exclusive-OR operation is a binary operation which is equivalent to applying logical exclusive-OR to every pair of bits located on the same positions in binary notation of operands. In other words, a binary digit of the result is equal to 1 if and only if bits on the respective positions in the operands are different. For example, if $X = 109_{10} = 1101101_2$ and $Y = 41_{10} = 101001_2$, then $X \oplus Y = 1000100_2 = 68_{10}$.

Now AAA wants to find out the minimum sum of all the elements in the sequence, or determine that the sequence meets all the constraints does not exist.

Input

The first line contains two integers n ($1 \le n \le 10^5$) and m ($0 \le m \le 2 \times 10^5$), denoting the length of sequence and the number of conditions.

The follow m lines, each of which contains three integers u, v ($1 \le u, v \le n$) and w ($0 \le w < 2^{30}$), indicating a constraint that $a_u \oplus a_v = w$.

Output

Output a line containing a single integer, indicating the minimum sum of all the elements in the sequence or -1 if the sequence meets all the constraints does not exist.

Examples

standard input	standard output
3 2 1 2 1 2 3 1	1
3 3 1 2 1 2 3 1 1 3 1	-1

Note

In the first sample case, the sequence $[a_1, a_2, a_3] = [0, 1, 0]$ meets all the constraints and has the minimum sum of all the elements.

Problem C. Cards of Magic

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

Raiden Shogun, the ruler of Inazuma, is playing a new card game invented by her friend Yae Miko.

Pixiv ID: 93716380

The player needs to defeat a monster of health point (HP) n by using magic cards properly.

Specifically, there are three kinds of magic cards:

1. Waterman: Summon the Waterman on the field and the Waterman will stay on the field forever. Each time when you cast a magic card and the Waterman has been on the field, it will reduce the monster's HP by 1. Note that if you cast another Waterman card when the Waterman has already been on the field, nothing will happen except the monster's HP decresing by 1;

2. Fireball: Reduce the monster's HP by 2;

3. Copy: Copy any card casted before this card and receive a copy of the copied card.

At the beginning of each turn, the player receives a new card from one of the three kind in equal possibility of $\frac{1}{3}$. Then the player can choose to cast some magic cards in his hand one by one and the casted cards will take effect respectively. Since there is no limitation for the number of cards in hand, the player can choose to keep cards and do nothing in the turn.

The monster will be defeated if its HP becomes nonpositive. Now Raiden Shogun hopes you can calculate the expected number of turns to defeat the monster if she plays optimally to minimize the expected number of turns.

Input

The only line contains a single integer n ($1 \le n \le 2 \times 10^5$), indicating the monster's HP.

Output

Output a line containing an integer, indicating the expected number of turns to defeat the monster modulo $998\,244\,353$ if playing optimally.

More precisely, if the reduced fraction of the answer is $\frac{p}{q}$, what you should provide is the minimum nonnegative integer r such that $qr \equiv p \pmod{998\,244\,353}$. You may safely assume that such r always exists.

Examples

standard input	standard output
1	831870296
5	835978299

Note

In the first sample case:

- If receiving a Waterman card in the first turn, just cast it and the monster will be defeated no matter receiving which kind of magic card in the next turn;

- If receiving a Fireball card in the first turn, just cast it and defeat the monster immediately;

- If receiving a Copy card in the first turn, do nothing in the following turns until receiving a Waterman card or a Fireball card. Once receiving a Waterman card, the monster will be defeated by casting the Waterman card and then casting any card in hand. Once receiving a Fireball card, the monster will be defeated by casting the Fireball card immediately.

In all, the expected number of turns to defeat the monster of HP 1 is $\frac{11}{6}$ if playing optimally.

Problem D. Cross the Maze

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

"Ad astra abyssoque."

The adventurers from Teyvat are trapped in a rectangular maze of $a \times b$ square cells, which is surrounded by walls from the outside. The floorboards of the maze are so fragile that no two adventurers are allowed moving to or staying in the same cell of the maze **anytime**.

When they are trying to cross the maze, they find n escape ropes in the maze and realize that the number of the escape ropes in the maze is exactly the same as the number of the adventurers trapped in the maze. However, the escape ropes are of low quality that a rope breaks immediately when an adventurer escape from the maze with it.

In every second, each of the adventurers can independently and simultaneously move from the cell (x, y) to left cell $(x, y - 1)$, the right cell $(x, y + 1)$, the upper cell $(x - 1, y)$, the lower cell $(x + 1, y)$ in the maze, or just stay in place. When reaching a cell contains an intact escape rope after moving, an adventurer can decide to get out of the maze with the help of the escape rope immediately and then the rope breaks, or stay in the maze and leave the escape rope for others. Note that an adventurer can also leave the maze before moving if the adventurer has already located at a cell contains an escape rope.

Lumine the traveller wants to guide all the trapped adventurers cross the maze as soon as possible that she can continue searching for her family.

Input

The first line contains three positive integers n ($1 \le n \le 100$), a and b ($1 \le a, b \le 100, 1 \le a \times b \le 100$), denoting the number of the adventurers trapped in the maze as well as the number of the escape ropes in the maze and the number of rows and columns in the maze.

Then follow n lines, each of which contains two integers s_x ($1 \le s_x \le a$) and s_y ($1 \le s_y \le b$), denoting an adventurer located at the cell (s_x, s_y) in the maze initially. All the adventurers are located at different cells initially.

Then again follow n lines, each of which contains two integers e_x ($1 \le e_x \le a$) and e_y ($1 \le e_y \le b$), denoting an escape rope located at the cell (e_x, e_y) in the maze. All the escape ropes are located at different cells.

It is guaranteed that at least one adventurer located at the cell without escape ropes initially.

Output

The first line contains an integer t, denoting the minimum time in seconds needed for all the trapped adventurers crossing the maze.

Then output n lines, each of which contains four integers s_x, s_y, e_x and e_y, followed by a string of length t. It means that an adventurer located at the cell (s_x, s_y) initially will get out of the maze with the escape rope located at the cell (e_x, e_y) by taking actions according to the string. The i-th character in the string points out the action taken in the i-th second, where L means going to the left cell, R means going to the right cell, U means going to the upper cell, D means going to the lower cell, S means staying in place, and P means the adventurer has left the maze and never goes back.

If there are multiple solutions, you may output any.

Examples

standard input	standard output
3 4 4	2
1 1	1 1 1 3 RR
1 4	1 4 2 3 DL
4 4	4 4 2 4 UU
1 3	
2 3	
2 4	
3 2 2	1
1 1	1 1 1 1 P
1 2	1 2 2 2 D
2 2	2 2 2 1 L
1 1	
2 1	
2 2	
2 3 3	2
1 1	1 1 2 2 DR
1 3	1 3 1 2 LP
1 2	
2 2	

Problem E. Edward Gaming, the Champion

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

On November 6, 2021, the Chinese team Edward Gaming (EDG) defeated the South Korea team DWG KIA (DK) to win the 2021 *League of Legends* World championship in Reykjavík, Iceland, lifting the Summoner's Cup for the first time in their history.

While both teams had looked dominant throughout the competition, DK arguably had the advantage. The team hadn't lost a single game until they reached the semi-finals and was the only team to make it out of the Group Stage without a single defeat. They were clearly the team to beat.

EDG had given them a hit at the very first game of the final. The game started with a well-executed gank in the bot lane by EDG for the first blood. Later, EDG took every single Drake and the Baron, and ultimately destroyed the DK's Nexus after 35 minutes.

But DK wouldn't leave it unanswered. They maintained an advantage throughout the second game. Not even the incredible Baron steal by EDG's legendary jungler, *Jiejie*, could help the team.

The third game turned out to be a difficult one. EDG seems to have control over more resources during the first 30 minutes. However, DK constantly killed every single dragon, and they finally took down the Nexus with the Hand of Baron.

In the fourth game, EDG had rethought their approach and took higher precedence in the control over dragons. The strategy had immediately taken effect, and they won the game after 33 minutes.

All things came down to the last game of the finals. Initially, DK took up the first dragon without much resistance from EDG. Shortly after, EDG picked first blood as DK took the Herald. Everything was fairly even at that moment. The balance finally started to tip in EDG's favor during a team fight in the mid-lane, with EDG killing DK's midlaner *Showmaker* before they had a chance to respond. The fight finally ended up with four kills and one death for EDG. They snowballed their advantage and finally secured the trophy.

The triumph of the Worlds 2021 made EDG the first team from LPL to win both the Mid-Season Invitational and the World Championship. You have just written a long string to celebrate this great victory. How many occurrences of `edgnb` as a continuous substring are there in the long string? Please write a program to count the number.

Input

The only line contains a nonempty string, which consists of no more than 200 000 lowercase Latin letters, `a` to `z`.

Output

Output a line containing a single integer, indicating the number of occurrences of `edgnb` in the given string.

Example

standard input	standard output
edgnb	1

Problem F. Encoded Strings I

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

Tommy has just invented an interesting string encoding algorithm, which is described below.

- For every string S, we may define a character-mapping function F_S, which maps every character occurring in S to a lowercase letter, as below

$$F_S(c) = \text{chr}(G(c, S))$$

where $\text{chr}(i)$ is the $(i+1)$-th lowercase Latin letter, and $G(c, S)$ is the number of different characters after the last occurrence of c in S.

- To encode a string S by Tommy's algorithm, replace every character c in S by $F_S(c)$ simultaneously.

For example, the encoded string of `abc` is `cba`, and the encoded string of `cac` is `aba`.

You are given a string of length n and then encode all the n **nonempty prefixes**. Your task is to find the encoded string that has the greatest lexicographical order among all the encoded strings.

Input

The first line contains an integer n ($1 \le n \le 1\,000$).

The second line contains a string of length n, which consists of only the first 20 lowercase letters, `a` to `t`.

Output

Output the encoded string that has the greatest lexicographical order among all the encoded strings.

Examples

standard input	standard output
4 aacc	bbaa
3 aca	ba

Note

In the first sample case, the four nonempty prefixes `a`, `aa`, `aac` and `aacc` will be encoded to `a`, `aa`, `bba` and `bbaa` respectively, where `bbaa` has the greatest lexicographical order.

In the second sample case, the three nonempty prefixes `a`, `ac` and `aca` will be encoded to `a`, `ba` and `aba` respectively, where `ba` has the greatest lexicographical order.

Problem G. Encoded Strings II

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

Tommy has just invented an interesting string encoding algorithm, which is described below.

- For every string S, we may define a character-mapping function F_S, which maps every character occurring in S to a lowercase letter, as below

$$F_S(c) = \mathrm{chr}(G(c, S))$$

where $\mathrm{chr}(i)$ is the $(i+1)$-th lowercase Latin letter, and $G(c, S)$ is the number of different characters after the last occurrence of c in S.

- To encode a string S by Tommy's algorithm, replace every character c in S by $F_S(c)$ simultaneously.

For example, the encoded string of `abc` is `cba`, and the encoded string of `cac` is `aba`.

You are given a string of length n and then encode all the $2^n - 1$ **nonempty subsequences**. Your task is to find the encoded string that has the greatest lexicographical order among all the encoded strings.

Input

The first line contains an integer n $(1 \le n \le 1\,000)$.

The second line contains a string of length n, which consists of only the first 20 lowercase letters, `a` to `t`.

Output

Output the encoded string that has the greatest lexicographical order among all the encoded strings.

Examples

| standard input	standard output
4 aacc	bbaa
4 acac	bba

Problem H. Line Graph Matching

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

In the mathematical discipline of graph theory, the line graph of a simple undirected weighted graph G is another simple undirected weighted graph $L(G)$ that represents the adjacency between every two edges in G.

Precisely speaking, for an undirected weighted graph G without loops or multiple edges, its line graph $L(G)$ is an undirected weighted graph such that:

- Each vertex of $L(G)$ represents an edge of G;

- Two vertices of $L(G)$ are adjacent if and only if their corresponding edges share a common endpoint in G, and the weight of such edge between this two vertices is the sum of the weights of their corresponding edges.
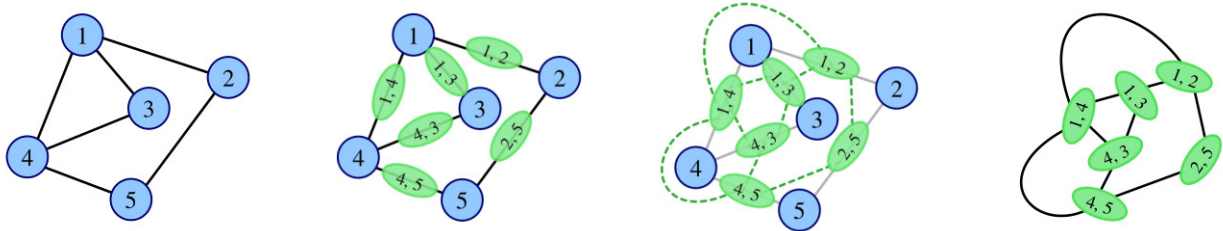
A maximum weighted matching in a simple undirected weighted graph is defined as a set of edges where no two edges share a common vertex and the sum of the weights of the edges in the set is maximized.

Given a simple undirected weighted connected graph G, your task is to find the sum of the weights of the edges in the maximum weighted matching of $L(G)$.

Input

The first line contains two integers n ($3 \leq n \leq 10^5$) and m ($n-1 \leq m \leq \min(\frac{n(n-1)}{2}, 2 \times 10^5)$), indicating the number of vertices and edges in the given graph G.

Then follow m lines, the i-th of which contains three integers u, v ($1 \leq u, v \leq n$) and w ($1 \leq w \leq 10^9$), indicating that the i-th edge in the graph G has a weight of w and connects the u-th and the v-th vertices. It is guaranteed that the graph G is connected and contains no loops and no multiple edges.

Output

Output a line containing a single integer, indicating the sum of the weights of the edges in the maximum weighted matching of $L(G)$.

Examples

standard input	standard output
5 6 1 2 1 1 3 2 1 4 3 4 3 4 4 5 5 2 5 6	21
6 5 1 2 4 2 3 1 3 4 3 4 5 2 5 6 5	12
5 5 1 2 1 2 3 2 3 4 3 4 5 4 5 1 5	14

Problem I. Linear Fractional Transformation

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

The linear fractional transformations are the functions $f(z) = \frac{az+b}{cz+d}$ ($a, b, c, d \in \mathbb{C}, ad - bc \neq 0$) mapping the extended complex plane $\mathbb{C} \cup \{\infty\}$ onto itself.

Given $f(z_1) = w_1$, $f(z_2) = w_2$ and $f(z_3) = w_3$, where z_1, z_2 and z_3 are pairwise distinct complex numbers and w_1, w_2 and w_3 are also pairwise distinct complex numbers, your task is to calculate $f(z_0)$ for a certain complex number z_0. It can be shown that the answer is always unique to the given contraints.

Input

The input contains several test cases, and the first line contains an integer T ($1 \leq T \leq 10^5$), indicating the number of test cases.

To clarify the input format, we denote $z_0 = p_0 + q_0 i$, $z_1 = p_1 + q_1 i$, $z_2 = p_2 + q_2 i$, $z_3 = p_3 + q_3 i$, $w_1 = r_1 + s_1 i$, $w_2 = r_2 + s_2 i$ and $w_3 = r_3 + s_3 i$, where i is the imaginary unit that $i^2 = -1$.

Then for each test case, the first line contains four integers p_1, q_1, r_1 and s_1, the second contains four integers p_2, q_2, r_2 and s_2, the third line contains four integers p_3, q_3, r_3 and s_3, and the fourth line contains only two integers p_0 and q_0. It is guaranteed that all these integers are in the range $[-100, 100]$ and the answer $f(z_0)$ satisfies $|f(z_0)| < 10^6$, where $|z| = |p + qi| = \sqrt{p^2 + q^2}$ ($p, q \in \mathbb{R}$) is the modulus of the complex number z.

Output

For each test case, output a line containing two real numbers c_0 and d_0, indicating the real part and the imaginary part of $f(z_0)$.

Your answer is acceptable if the absolute or relative errors of both the real part and the imaginary part do not exceed 10^{-6}. Formally speaking, suppose that your output is x and the jury's answer is y, your output is accepted if and only if $\frac{|x-y|}{\max(1,|y|)} \leq 10^{-6}$.

Example

standard input	standard output
2	1.000000000000000 0.000000000000000
-1 0 0 -1	0.000000000000000 1.000000000000000
0 1 -1 0	
1 0 0 1	
0 -1	
-1 0 -1 0	
0 1 0 -1	
1 0 1 0	
0 -1	

Note

In the first sample case we have $f(z) = iz$, and in the second sample case we have $f(z) = 1/z$.

Problem J. Luggage Lock

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

Eileen has a big luggage and she would pick a lot of things in the luggage every time when A-SOUL goes out for a show. However, if there are too many things in the luggage, the 4-digit password lock on the luggage will be hard to rotate.

The state of lock is the four digits on the lock. In one step, she can choose consecutive digits to rotate either up or down. For example, she can rotate 0000 to 0111 or 0900 in one step because the rotated digits are consecutive, but she can't rotate 0000 to 0101 in one step. Since she has little strength, she wants to rotate the lock as few times as possible.

Now the lock is at state $a_0a_1a_2a_3$ and the password is $b_0b_1b_2b_3$. As a fan of A-SOUL, you are asked to help Eileen find out the optimal plan to unlock but you only need to tell Eileen how many times she has to rotate.

Input

The first line contains one integer T ($1 \le T \le 10^5$), denoting the numer of test cases.

Each of the test cases contains a line containing the initial state $a_0a_1a_2a_3$ and the target state $b_0b_1b_2b_3$.

Output

For each test case, output a line containing a single integer, denoting the minimum steps needed to unlock.

Example

standard input	standard output
6	1
1234 2345	1
1234 0123	4
1234 2267	5
1234 3401	1
1234 1344	4
1234 2468	

Problem K. Matrix Operations

Input file:	standard input
Output file:	standard output
Time limit:	15 seconds
Memory limit:	512 megabytes

Given an $n \times n$ matrix where each element $a_{i,j}$ ($1 \leq i, j \leq n$) in the matrix is 0 initailly, you are asked to successively perform n groups of operations on the matrix.

For each group of operations, you are given six parameters x, y, z_1, z_2, z_3 and z_4 and need to do the following operations in order:

1. Find the maximum element among $a_{i,j}$ ($1 \leq i < x, 1 \leq j < y$), denoted by w_1;

2. Find the maximum element among $a_{i,j}$ ($1 \leq i < x, y \leq j \leq n$), denoted by w_2;

3. Find the maximum element among $a_{i,j}$ ($x \leq i \leq n, 1 \leq j < y$), denoted by w_3;

4. Find the maximum element among $a_{i,j}$ ($x \leq i \leq n, y \leq j \leq n$), denoted by w_4;

5. Increase each element $a_{i,j}$ ($1 \leq i < x, 1 \leq j < y$) by z_1;

6. Increase each element $a_{i,j}$ ($1 \leq i < x, y \leq j \leq n$) by z_2;

7. Increase each element $a_{i,j}$ ($x \leq i \leq n, 1 \leq j < y$) by z_3;

8. Increase each element $a_{i,j}$ ($x \leq i \leq n, y \leq j \leq n$) by z_4.

After performing each group of operations, you need to output the values of w_1, w_2, w_3 and w_4.

Input

The first line contains an integer n ($2 \leq n \leq 10^5$), indicating the number of rows as well as columns in the matrix as well as the number of groups of operations.

Then follow n lines, each of which contains contains six integers x, y ($1 < x, y \leq m$), z_1, z_2, z_3 and z_4 ($1 \leq z_1, z_2, z_3, z_4 \leq 10^9$), indicating the parameters of a group of operations as described above.

Output

For each operation, output a line containing four integers, indicating the values of w_1, w_2, w_3 and w_4.

Example

standard input	standard output
3 3 3 1 2 3 4 2 3 1 2 3 4 3 2 1 2 3 4	0 0 0 0 1 2 3 4 4 6 6 8

Problem L. Perfect Matchings

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

AAA gets a complete graph of $2n$ vertices, where every pair of distinct vertices is connected by a unique edge, as a birthday present. However, AAA thinks the complete graph is not that beautiful and he decides to delete $2n - 1$ edges that form a tree.

Now he wonders the number of different perfect matchings in the remaining graph. Note that a perfect matching is a set of n edges swhere no two edges share a common vertex. Since the answer may be very large, you only need to output the answer modulo $998\,244\,353$.

Input

The first line contains a single integer n $(2 \le n \le 2\,000)$.

Each of the next $2n - 1$ lines contains two integers u and v $(1 \le u, v \le 2n)$, representing an edge deleted from the complete graph. It is guaranteed that the given edges form a tree of $2n$ vertices.

Output

Output a line containing a single integer, representing the answer modulo $998\,244\,353$.

Examples

standard input	standard output
2 1 2 1 3 3 4	1
3 1 2 2 3 3 4 4 5 5 6	5

Problem M. String Problem

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

JB hates solving string problems. Therefore, when his friend Potato gives him a string problem to solve, he immediately gives it to you and continues playing Genshin Impact, the greatest game in the world.

You are given a string and then for each nonempty prefix, you need to find the largest substring in lexicographical order and point out the leftmost occurrence of the largest substring.

Input

The only line contains a string S $(1 \le |S| \le 10^6)$, which consists of lowercase Latin letters, a to z.

Output

Output $|S|$ lines, the i-th of which contains two integers l_i and r_i, indicating the leftmost occurrence of the largest substring in the prefix of length i.

Examples

standard input	standard output
potato	1 1 1 2 3 3 3 4 3 5 5 6
pbpbppb	1 1 1 2 1 3 1 4 1 5 5 6 5 7