Problem A. Amount payable

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

Alexa is the founder of a company that specializes in selling video cards. Previously, she used to set prices for her products in an arbitrary manner, but she has now realized that it would be more profitable if all prices end in 99.

You have been given a positive integer N, which represents the current price of a card. Your task is to find the nearest positive integer to N that ends in 99. If there are two such numbers that are equally close to N, you should choose the larger one.

Input

The input contains one integer N ($1 \le N \le 10^4$), the price of a video card. It is guaranteed that the number N does not end in 99.

Output

Print one integer, the closest positive integer that ends in 99. In case of a tie, print the bigger one.

Examples

standard input	standard output
10	99
249	299
1000	9999

Problem B. Badkey

Input file:	standard input
Output file:	standard output
Time limit:	3 second
Memory limit:	512 mebibytes

Mary is the founder and CEO of Badkey, the notebook vendor that specializes in selling the DIY sets of notebooks.

With N notebooks available, Mary wants to assign each one a unique, easy-to-remember name. However, manually coming up with names for each notebook would be a time-consuming task.

Your assignment is to generate N distinct names that are easy to read and remember. To meet this requirement, the names must adhere to the following criteria:

1. Each name has length between 3 and 20, and only consists of lowercase English letters.
2. Three consecutive letters of a name cannot all be vowels or consonants. Here we consider `a`, `e`, `i`, `o`, `u` vowels, while the remaining 21 letters are consonants.

For example, `badkey`, `ibm`, and `sdio` are all valid names, whereas `thinkpad`, `it` and `aeons` are invalid.

Input

The input consists of one integer N ($1 \le N \le 30000$), the number of names to generate.

Output

Print N lines, each of them containing a name. It can be proven that it is possible to generate N different names.

Example

standard input	standard output
3	latitude precision elitebook

Problem C. Ramen

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

Grammy likes to eat noodles. She divided a very long strip of noodle into N parts of unit length. Each part i has deliciousness a_i. She would like to fold the noodle into one piece of unit length before eating by repeating the following operation several (possibly, zero) times.

Let n be the current length of the noodle. In each operation, Grammy can choose a length ℓ such that $2\ell \le n$ and $a_i > 0$ for all $i \le \ell$, and fold the noodle $a_1, a_2, \ldots, a_\ell, a_{\ell+1}, \ldots, a_{2\ell}, a_{2\ell+1}, \ldots, a_n$ into $a_{\ell+1} + a_\ell, a_{\ell+2} + a_{\ell-1}, \ldots, a_{2\ell} + a_1, a_{2\ell+1}, \ldots, a_n$, where n is the length of the noodle before the operation. After the operation, the length will become $n - \ell$.

Grammy wants to know whether she can fold the noodle to length 1, can you tell her?

Input

The first line of input contains a single integer N ($1 \le N \le 100\,000$).

The second line contains N integers a_i ($-20\,000 \le a_i \le 20\,000$), representing the deliciousness of each part of the noodle.

Output

If Grammy can fold the noodle to length 1, output a single line with the word "`YES`". Otherwise, output a single line with the word "`NO`".

Examples

standard input	*standard output*
3 1 2 -5	YES
5 2 -5 2 3 1	NO

Problem D. Rotate Sum 2

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

Grammy loves geometry. Today, she takes out her precious convex polygon and plays with it on a piece of paper. The polygon has n vertices numbered from 1 to n in counterclockwise order. For vertex i, the next vertex in this order is $i^+ = i \bmod n + 1$, and the previous one is $i^- = (i + n - 2) \bmod n + 1$.

Firstly, Grammy draws a horizontal line on the paper. Secondly, she chooses two vertices i and j of the polygon independently and equiprobably. Thirdly, she places the edge between vertex i and vertex i^- on the line, landing all other vertices above the line, and draws a vertical line through vertex j. Next, she rotates the polygon clockwise, taking vertex i as the rotation center, until vertex i^+ hits the line. When vertex i^+ hits the line, she changes the rotation center to vertex i^+ and rotates again until vertex i^{++} (the next after i^+) hits the line. She repeats this operation until vertex i hits the line again. Finally, she draws another vertical line through the vertex j and calculates the area between the trajectory of vertex j and the three lines.

Since you do not know which points Grammy will choose, you want to calculate the expected value of the area.

Input

The first line contains a single integer n ($3 \le n \le 100\,000$), denoting the number of vertices in the polygon.

Each of the following n lines contains two integers x_i and y_i ($-10^9 \le x_i, y_i \le 10^9$), denoting the coordinates of a vertex of the polygon. The vertices are given in counterclockwise order. It is guaranteed that the polygon is strictly convex.

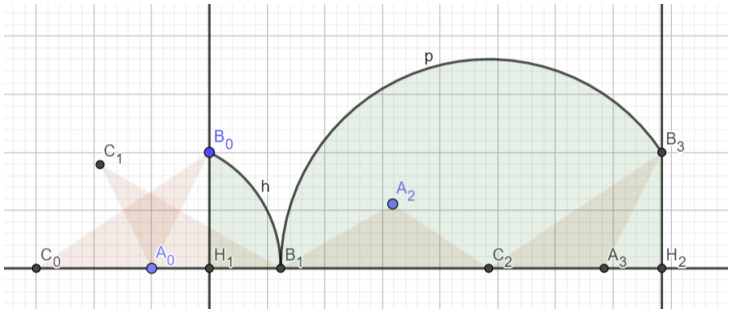
Output

Output a single real number denoting the expected area. The answer is considered correct if its absolute of relative error does not exceed 10^{-4}.

Example

standard input	*standard output*
3 1 -1 1 1 -1 2	18.763234503173919

Note



For the first example, if the i-th vertex is marked as A_0, and the j-th vertex is marked as B_0, then the polygon will be $A_3B_3C_2$ after 3 rotations, and the trajectory of vertex j is arc h and arc p. The area of the green part is the answer in this case.

Problem E. Easter Egg

Input file:	standard input
Output file:	standard output
Time limit:	3 second
Memory limit:	512 mebibytes

Egor's grandmother organizes an annual event called *The Great Egg Hunt* at their family mansion during Easter. The event starts with filling a giant Easter Egg with candy, followed by hiding the egg in a randomly selected room of the mansion. The person who finds the egg first gets to keep all the candy inside. This tradition has been ongoing in Egor's family even before Egor was born.

The family mansion has N rooms connected by $N-1$ doorways. It is guaranteed that the mansion is connected, meaning it is possible to walk from any room to any other room through the doorways.

Egor has a searching technique called *Egg First Search*.

1. If Egor is in an unexplored room, he searches that room. If the egg is in that room, he will find it.
2. Otherwise, he moves to an adjacent room that is in the direction of any of the closest unexplored rooms, picking one at random if there are multiple options.

Egor takes 1 unit of time to search a room and also 1 unit of time to move to an adjacent room.

Egor wants to know which starting room(s) will minimize the expected time to find the egg, given a map of the mansion.

Input

The first line consists of one integer N ($1 \le N \le 2 \cdot 10^5$), the number of rooms in the mansion. The next $N-1$ lines consist of two space separated integers u and v ($1 \le u, v \le N$, $u \ne v$), a pair of rooms that are connected by a doorway. It is guaranteed that the mansion is connected, meaning it is always possible to walk from any room to any other room using the doorways.

Output

On the first line print an integer M, the number of optimal starting rooms. On the second line, print M space separated integers $S_1, \ldots, S_M$ ($1 \le S_1 < S_2 < \ldots < S_M \le N$), the list of optimal starting rooms.

Examples

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

Problem F. Fine Ukraina

Input file:	standard input
Output file:	standard output
Time limit:	14 second
Memory limit:	512 mebibytes

In the Ukraine, everything is optimized to the maximum extent possible. This includes the road network, which has been designed such that there is only one route between any two cities, ensuring that GPS route planners always provide optimal directions.

Ukraine also has a network of bus lines, with each line traveling back and forth between two cities and stopping at every city in between. However, there has been a recent scandal in which it was discovered that the current bus line network is non-optimal and uses too many lines.

You have been hired by Ukraine to create a new, optimal network of bus lines that meets two requirements:

- Every pair of cities must be accessible by bus.
- The number of bus lines should be minimized.

To accomplish this, you will model Ukraine as an undirected graph with cities as nodes and roads as edges. Each bus line will be represented by two cities, the starting and ending points of the line.

Input

The first line consists of one integer N ($3 \le N \le 2 \cdot 10^5$), the number of cities in Ukraine. The next $N-1$ lines consists of two integers u and v ($1 \le u, v \le N$, $u \ne v$), a pair of cities that are connected by a road. It is guaranteed that Ukraine is connected, meaning it is possible to travel between any two cities using the road network.

Output

On the first line output the optimal number of bus lines M. Then output M lines, where the i-th line should contain two space separated integers a_i, b_i ($1 \le a_i, b_i \le N$), the two endpoints of the i-th bus line.

Examples

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

Problem G. Geometry

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	1024 mebibytes

Grammy has a special two-dimensional coordinate system: the angle between the positive half-axis of the X-axis and the positive half-axis of the Y-axis is 60 degrees.

Consider the following graph. The vertices are all integer coordinates (x, y) such that at least one of x, y is odd and $-2a + 1 \le x \le 2a - 1$, $-2b + 1 \le y \le 2b - 1$, $-2c + 1 \le x + y \le 2c - 1$. The edges from (x, y) go to $(x, y + 1)$, $(x, y - 1)$, $(x + 1, y)$, $(x - 1, y)$, $(x + 1, y - 1)$, and $(x - 1, y + 1)$.

Find the size of the maximum independent set of vertices in this graph. Additionally, find the number of such sets modulo 998 244 353.

Input

The first line contains an integer T ($1 \le T \le 10$), denoting the number of test cases.

Each of the following T lines contains three integers a, b, c ($1 \le a, b, c \le 10^6$).

Output

Output T lines. Each line must contain two integers: the size of the maximum independent set and the number of such sets. **Please note that the size should not be taken modulo** 998 244 353.

Example

standard input	*standard output*
6 2 1 2 1 1 137 3 94 95 3 1998 1996 998244 353999 999999 50 120 150	7 4 4 1 1124 31585548 23951 33873190 1289433675488 748596399 23600 480090154

Note

The following picture shows the situation for the first and second test case of the sample.

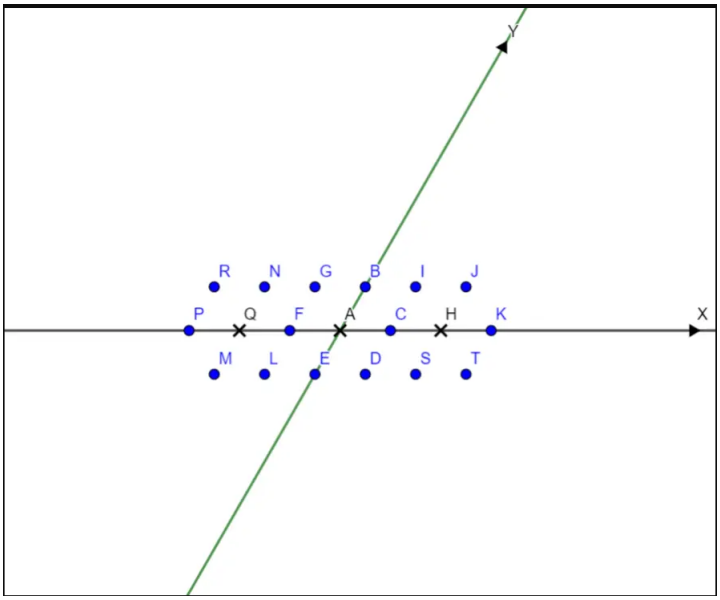
Point J has coordinates $(2, 1)$, point F has coordinates $(-1, 0)$, and point H has coordinates $(2, 0)$. Among these three points, only H has even X-coordinate and even Y-coordinate. The neighbours of point A are $BCDEFG$.

In the first test case, the points that satisfy the conditions are $NGBIJPFCKMLEDST$.

The size of the maximum independent set is 7, and there are 4 ways: $PNLBDJT$, $RMFBDJT$, $RMGECJT$, $RMGEISK$.

In the second test case, the points that satisfy the conditions are $GBIFCLED$.

The size of the maximum independent set is 4, and there is one way: $LGID$.



Picture for test case 1 and 2.

Problem H. Grid system

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

As part of your machine learning coursework at university, you have been assigned the task of developing a program that can distinguish between vertical and horizontal bars in images. To generate training data, you have utilized the following method: First, create an $N \times N$ grid and set all its entries to zero. Next, select a row or a column and set all its entries to one. Finally, flip at most N randomly chosen cells by changing a zero to a one or a one to a zero.

Although generating data using this method is straightforward, determining the correct output for each generated matrix is a time-consuming task. It would take several hours to manually determine the output for all the training data. If only there was a program that could automatically determine all the outputs.

You have been given an $N \times N$ matrix generated using the aforementioned method. Your task is to develop a program that can determine whether it was a row or a column that was originally filled with ones or if it is impossible to determine (due to both possibilities or error in the method implenentation that caused generation of the random martix).

Input

The first line of input consists of an integer N ($2 \leq N \leq 1000$), the size of the grid.

The following N lines each contain a string of length N consisting of zeros and ones. These are the rows of the grid.

It is guaranteed that the input was generated by taking a grid of zeros, putting ones on a row or a column, and then flipping at most N cells.

Input

If the bar was vertical (a column), print "`I`". If it was horizontal (a row), print "`-`". If it is impossible to determine (because it could be both), print "`B`". If it cannot be generated by taking a grid of zeros, putting ones on a row or a column, and then flipping at most N cells, "`?`".

Examples

standard input	standard output
5 01100 01000 01001 00000 01000	I
3 111 000 111	-
3 010 101 010	B

Problem I. Infectious Disease

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	1024 mebibytes

In the year of 2202, a strange disease begins to spread in a city of n people.

To prevent the disease from spreading, experts invented a strong vaccine called Mysterious Oscar. On day 0, one citizen is infected by the disease, and another citizen is vaccinated. If a person becomes vaccinated, he/she will be cured immediately and will not catch or spread the disease anymore.

On each subsequent day d ($d > 0$), the infected citizens will infect others one by one. Each of the citizens who were infected strictly before day d will choose one uninfected and unvaccinated citizen to infect equiprobably. If at some point, one infected citizen has no unvaccinated and uninfected citizens to choose from, then he/she will do nothing.

After infection, the vaccinated citizens will persuade others to take the vaccine one by one. Each of the citizens who were vaccinated strictly before day d will choose 2 different unvaccinated citizens equiprobably, and persuade them so that they become vaccinated. If at some point, one vaccinated citizen has less than 2 unvaccinated citizens to choose, then he/she will persuade all the remaining unvaccinated citizens to take the vaccine.

Grammy wants to know how many days will pass before the disease will be fully extinguished. Please tell her the expected number of days before all patients become cured.

It can be shown that the answer can be expressed as an irreducible fraction $\frac{x}{y}$, where x and y are integers and $y \not\equiv 0 \pmod{10^9+7}$. Output the integer equal to $x \cdot y^{-1} \pmod{10^9+7}$. In other words, output such an integer a that $0 \le a < 10^9+7$ and $a \cdot y \equiv x \pmod{10^9+7}$.

Input

The only line contains an integer n ($2 \le n \le 1.4 \cdot 10^7$), denoting the population of the city.

Output

Output a single integer, denoting the expected number of days before all patients become cured modulo 10^9+7.

Examples

standard input	*standard output*
2	1
114	505208013

Note

In the first sample, one citizen took the vaccine on day 0, and he/she persuaded the other citizen, the only patient, to take the vaccine on day 1, so the disease must be completely cured on day 1.

Problem J. Positive String

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

Grammy had a unique insight about strings. She thinks that a string is *positive* if and only if it is lexicographically larger than its reversal.

Now you are given a string, please find out how many contiguous substrings of it are positive according to Grammy's insight.

Input

The single line contains a string S ($1 \le |S| \le 200\,000$), consisting of lowercase English letters only.

Output

Output a single integer denoting the number of positive substrings of S.

Examples

standard input	*standard output*
jjikkollp	4
pbpbppb	7