Problem A. Recruiting

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Hurry! Two well-known software companies are recruiting programmers. Initially the total number of unemployed programmers is n. The companies are hiring them one by one, alternately. In one turn a company hires one of the programmers who has not been hired yet. The first company starts hiring.

Furthermore, there are some pairs of friends among the programmers. Of course, a programmer may have several friends. Friendship is a symmetrical relationship, so if a is a friend of b then b is a friend of a.

All such pairs are known to the companies, and the companies follow the rule: a new programmer hired by a company must have at least one friend among the programmers already working in this company. The only exception is a programmer that a company starts with, he can be chosen arbitrarily. It may happen that after a number of turns a company can not longer hire anyone else according to the rule. In this case it stops hiring, while the other company can continue.

As usual, not all the programmers are created equal. There are three geniuses among them, and each company wants to hire as many geniuses as possible. Note that each company always can guarantee one genius to itself starting with a genius. So the question is, which company will hire two geniuses, if they both use optimal strategies.

Note that both companies have the full information during the hiring: friendship relations, who are geniuses and which programmers were hired by each company in each turn.

Input

The first line of the input contains two integers n and m ($3 \le n \le 10^5$, $2 \le m \le 2 \cdot 10^5$) — the number of programmers and the number of pairs of friends among them, respectively. The programmers are numbered with integers from 1 to n. The geniuses have numbers 1, 2 and 3. The next m lines decribe pairs of friends. Each line contains a pair of integers a_i, b_i ($1 \le a_i, b_i \le n, a_i \ne b_i$), where a_i and b_i are friends in the i-th pair. No pair occurs more than once, even in reverse order.

Output

Output the number of the company (1 or 2) which will recruit two geniuses. It is guaranteed that one company will hire two geniuses.

Examples

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

Note

In the second example, programmers form a symmetric cycle of length 6. If the first company starts with a genius (say, number 1), the second one takes the programmer number 5. Then if the first company recruits the 4-th programmer, the second one recruits the 2-nd genius, and now it is closer to the 3-rd genius. Otherwise, if the first company starts with a usual programmer (say, number 4), the second company takes the 1-st genius and afterwards also wins. The remaining cases are considered symmetrically. In all the cases the second company has a strategy of recruiting two geniuses.

Problem B. Bonnie and Clyde

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Bonnie and Clyde are into robbing banks. This time their target is a town called Castle Rock. There are n banks located along Castle Rock's main street; each bank is described by two positive integers x_i, w_i, where x_i represents the distance between the i-th bank and the beginning of the street and w_i represents how much money the i-th bank has. The street can be represented as a straight line segment, that's why values of x_i can be regarded as the banks' coordinates on some imaginary coordinate axis.

This time Bonnie and Clyde decided to split, they decided to rob two different banks at a time. As robberies aren't exactly rare in Castle Rock, Bonnie and Clyde hope that the police won't see the connection between the two robberies. To decrease the chance of their plan being discovered by the investigation, they decided that the distance between the two robbed banks should be no less than d.

Help Bonnie and Clyde find two such banks, the distance between which is no less than d and the sum of money in which is maximum.

Input

The first input line contains a pair of integers n, d ($1 \le n \le 2 \cdot 10^5, 1 \le d \le 10^8$), where n is the number of banks and d is the minimum acceptable distance between the robberies. Then n lines contain descriptions of banks, one per line. Each line contains two integers x_i, w_i ($1 \le x_i, w_i \le 10^8$), x_i shows how far the i-th bank is from the beginning of the street and w_i shows the number of money in the bank. Positions of no two banks coincide. The banks are given in the increasing order of x_i.

Output

Print two integer numbers — indicies of the required banks. The banks are numbered starting from 1 in the order in which they follow in the input data. You may print indicies in any order. If there are many solutions, print any of them. If no such pair of banks exists, print "`-1 -1`" (without quotes).

Examples

standard input	standard output
6 3 1 1 3 5 4 8 6 4 10 3 11 2	5 3

Problem C. Building Foundation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

A new office building is to appear in Berland soon. Its construction has just been started, and the first problem builders are facing is to lay the foundation.

The ground at construction site area has already been hardened along n segments. Each segment is given by integer coordinates of its endpoints in the site area coordinate system. Every segment has a positive length and is parallel to either Ox axis or Oy axis. It's important to note that the ground hardening was done in such a way that only perpendicular segments could possibly have common points.

The decision has been made for the foundation to have a rectangular form. The rectangle must have the following properties:

- it should have a positive area,
- its sides should be parallel to one of the coordinate axes,
- its sides should be situated on the hardened ground, i.e. each point of its perimeter should belong to at least one segment out of the n hardened ones.

You are to help estimating the difficulty of choosing such a rectangle. Write a program that finds the number of rectangles that can possibly be used as a foundation.

Input

The first line contains integer n ($1 \le n \le 600$) — the number of hardened segments. Each of the following n lines contains four space-separated integers x_1, y_1, x_2, y_2 ($-10^9 \le x_1, y_1, x_2, y_2 \le 10^9$) — coordinates of the segments' endpoints. Each segment has positive length and is parallel to either Ox axis or Oy axis. No two horizontal segments have a common point. No two vertical segments have a common point.

Output

Print a single integer — the number of rectangles that can possibly be used as a foundation.

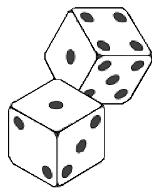
Examples

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

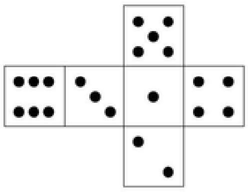
Problem D. Dice Tower

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Polycarp loves not only to play games, but to invent ones as well. He has recently been presented with a board game which also had lots of dice. Polycarp quickly noticed an interesting phenomenon: the sum of dots on any two opposite sides equals 7.

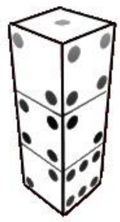


The dice



An unfolded die

Polycarp invented the following game. He asks somebody to tell a positive integer n and then he constructs a dice tower putting the dice one on another one. A tower is constructed like that: Polycarp puts a die on the table and then (if he wants) he adds more dice, each time stacking a new die on the top of the tower. The dice in the tower are aligned by their edges so that they form a perfect rectangular parallelepiped. The parallelepiped's height equals the number of dice in the tower and two other dimensions equal 1 (if we accept that a die's side is equal to 1).



An example of a tower whose height equals 3

Polycarp's aim is to build a tower of minimum height given that the sum of points on all its outer surface should equal the given number n (outer surface: the side surface, the top and bottom faces).

Write a program that would determine the minimum number of dice in the required tower by the given number n. Polycarp can construct any towers whose height equals 1 or more.

Input

The only input line contains integer n ($1 \le n \le 10^6$).

Output

Print the only integer — the number of dice in the required tower. If no such tower exists, print `-1`.

Examples

standard input	standard output
50	3
7	-1
32	2

Problem E. Computer Network

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Computer network of Berland's best physical and mathematical school has a "tree" topology. That is, the network has no cycles; it connects n computers by $n-1$ cables. Each cable connects exactly two different computers.

The school has really old equipment and the network is slow. The network administrator defined for each cable a value t_i, which represents the average time to transmit a packet from computer a_i to computer b_i (or vice versa), where a_i, b_i are computers connected by the i-th cable.

For two computers a and b the average time to transmit a packet from one computer to another is the sum of all t_i's for the cables on the path from a to b. Of course, an important characteristic of the network is the maximum possible average time of transmitting a packet between two computers. This characteristic is called μ.

In light of national innovations and modernizations it was decided to improve the school computer network and reduce the value of μ.

It was decided to replace some cables with new ones. New cables have such a high speed of data transmission that any packet passes through the cable in negligibly little time. In practice, the time of a packet transmission for new cables equals 0. For each cable its price p_i was determined — the cost of replacing the i-th cable by the new one. Of course, after replacing the i-th cable, the new one will still connect computers a_i and b_i.

Help the network administrator to find such set of cables, that after they are replaced, the value of μ becomes less and the total cost of replacing these cables is minimal. Note that in this problem you do not have to minimize μ, but you should make it less than its initial value.

Input

The first line contains a single integer n ($2 \le n \le 10^5$), n is the number of computers in the network. Then $n-1$ lines contain descriptions of all cables in the form of four integers a_i, b_i, t_i, p_i ($1 \le a_i, b_i \le n$; $1 \le t_i, p_i \le 10^4$), where a_i, b_i are numbers of computers connected by the i-th cable, t_i is the average transmission time of a packet through the cable and p_i is the cost of replacing the i-th cable with a new one.

Output

In the first line print the required minimum possible cost of replacements. In the second line print the number of cables in the required set in arbitrary order. In the third line print the numbers of these cables. Consider the cables numbered from 1 in the order they appear in the input. If there are multiple solutions, print any of them.

Examples

standard input	standard output
4 1 2 3 3 1 3 8 33 1 4 3 7	10 2 1 3
4 1 2 3 5 2 3 5 2 3 4 5 4	2 1 2

Problem F. Dirty Dishes

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

They say there are three things that one cannot have enough of looking at: they are the fire burning, the water flowing and others working. Jack and Jill have been married for several years but Jack never gets tired of watching Jill cleaning the kitchen in her swift and neat way!

As Jill is cleaning the kitchen, she piles up the dirty dishes by the sink. Each newly found dirty dish is put on the top of the pile and when Jill wants to wash the dishes, she takes a dish from the top of the pile.

Jack has been watching his wife attentively and each time she added a new dirty c-colored dish to the pile, Jack made a note in his notebook that a c-colored dish had been added to the pile. Similarly, when Jill took a c-colored dish from the top of the pile, Jack noted that a c-colored dish had been taken from the pile. From time to time Jack would leave to get more popcorn and then he would probably miss some of Jill's actions. In this case he wrote an asterisk "`*`" in his notebook.

Next day, as Jack was scanning through the notes, he got interested: what is the least number of dirty dishes that could be on the kitchen before the cleaning? Note that Jill doesn't arrange the dishes in more than one pile. Jill only puts the dishes to the top of the pile and only takes them from the top of the pile. Before the cleaning and after it the pile is empty.

Input

The only input line contains the notes from Jack's notebook. Using Latin alphabet, he wrote a lowercase letter if Jill added a dish of the corresponding color to the pile. Also, Jack wrote an uppercase letter if Jill took a dish of the corresponding color from the pile. For example, the string "`afFaAA`" represents the following actions:

- Jill adds an a-colored dish to the pile,
- Jill adds an f-colored dish to the pile,
- Jill takes an f-colored dish from the pile,
- Jill adds an a-colored dish to the pile,
- Jill takes an a-colored dish from the pile,
- Jill takes an a-colored dish from the pile.

An "`*`" represents the periods of time when Jack went to get more popcorn and could have missed one or more Jill's actions. It is guaranteed that Jack went to get more popcorn no more than five times. So the number of characters "`*`" doesn't exceed 5.

The given string consists of lowercase and uppercase Latin letters and asterisks "`*`". It is not empty and contains no more than 2500 characters. The input contains at least one letter.

Output

On the first line of the input file print the single number — the least possible number of dirty dishes on the kitchen before the cleaning. Print `-1` if Jack's notes surely have a mistake and there's no solution.

Examples

standard input	standard output
ab*bB	3
afFaAA	3
**bbB*Da*	4
a**b	-1

Note

An example of a sequence of actions for the first sample is abBAbB.

In the second sample Jack never left the room.

In the third sample he left the room four times, two of them one after another. An example of Jill's action succession is like that: "`bbBdDaAB`".

The fourth sample illustrates no solution.

Problem G. Berland Chess

Input file: `standard input`
Output file: `standard output`
Time limit: 4 seconds
Memory limit: 256 megabytes

Berland Chess is a single-player game played on a $n \times m$ rectangular chessboard. The chessboard has n rows, m columns, and is divided into $n \times m$ squares.

There are two colors of pieces that can be present on the chessboard — white and black. You play as White. According to the rules, the chessboard always has the only white piece — the white king, the only piece you have in your possession. All black pieces standing on the chessboard belong to your opponent, a computer-controlled chess robot running "Chess Bermaster" software. There are four kinds of chess pieces:

- "`*`" — white king,
- "`K`" — black knight,
- "`B`" — black bishop,
- "`R`" — black rook.

No other pieces are allowed. Each chess piece has its own method of movement. Moves are made to vacant squares except when capturing an opponent's piece. Here are the rules of a single move, described for each chess piece:

- white king moves exactly one square horizontally, vertically, or diagonally;
- black knight moves two squares horizontally then one square vertically, or one square horizontally then two squares vertically;
- black bishop moves any number of vacant squares in any diagonal direction;
- black rook moves any number of vacant squares vertically or horizontally.

With the exception of any movement of the knight, pieces cannot jump over each other. Moves of a knight are not blocked by other pieces as it just jumps to the new location.

The goal of the game is to capture all opponent's black pieces by the white king. Fortunately for you, "Chess Bermaster" is stuck in the infinite loop today due to a bug in software, so black pieces will not be moving during the game at all.

You may never move white king into a position where he could be captured by one of the black pieces. When you capture a black piece yourself, the corresponding black piece gets removed from the chessboard and the white king replaces it on its square.

Find the minimum number of moves it will take the white king to capture all the black pieces.

Input

The first line of input contains two integer numbers n, m ($1 \le n, m \le 15$) — the number of rows and columns on the chessboard, correspondingly. The next n lines contain m characters each — configuration of the chessboard. Each character will be one of the following:

- "`.`" — an empty space,
- "`*`" — white king,
- "`K`" — black knight,

- "B" — black bishop,
- "R" — black rook.

There will be exactly one white king on the chessboard, and it will not be under attack in its initial position. The total number of pieces on the chessboard will never exceed 15. There are no restrictions on number of black pieces by specific kinds. For example, it is allowed that the chessboard contains three or more black knights.

Output

Print a single integer — the minimum number of moves it will take the white king to capture all black pieces. If there are no black pieces on the chessboard, output 0. If it's impossible to capture all black pieces, output the only integer -1.

Examples

standard input	standard output
7 9 ....R.K.R.. .... *........	9

Problem H. Divisibility

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Inspired by Stephen Graham, the King of Berland started to study algorithms on strings. He was working days and nights, having a feeling that the full potential in this area is still to be unlocked. And he was right!

One day, all the sudden, he made a huge breakthrough by discovering the fact that strings can be magically transformed into integer numbers. It was so simple! You just have to map different letters to different digits and be careful enough not to introduce any leading zeroes.

Here is what he wrote in his textbook about the string 'lalala':

- it can be transformed to an 282828 by mapping 'l' to 2, and 'a' to 8
- it can also be transformed to 909090 by mapping 'l' to 9, and 'a' to 0
- a couple of examples of invalid transformations are 050505 (the resulting number has a leading zero), 333333 (different letters are mapped to the same digit), 123456 (no mapping to the original letters at all)

But then things started to become more interesting. Obviously, it was known from very beginning that a single string can potentially be mapped to a variety of different integer numbers. But the King couldn't even imagine that all numbers produced by the same string pattern might have common properties!

For example, every single number that can be produced from string 'lalala' is always divisible by 259, irrespective of the letter-to-digit mapping you choose. Fascinating!

So the King ended up with the following problem. For any given string, he wanted to come up with an algorithm to calculate the set of its divisors. A number is called a divisor of the given string if all positive integers, that could possibly be produced from the given string, are divisible by it.

As usual, the King desperately wants you to help him, so stop thinking and start acting!

Input

Input consists of multiple test cases. The first line of input contains an integer number n ($1 \le n \le 100$) — the number of test cases.

Each of the next n lines contains a string pattern to be processed. Each pattern consists of lowercase Latin letters. Its length will always be between 1 and 14 characters, and the number of different characters in the pattern will never exceed 10 to ensure correct mapping to digits 0-9.

Output

For every test case print a line with the corresponding test case number and the calculated list of divisors. Output positive divisors in an increasing order, separating them with a single space. Format your output according to the example given in the sample test case.

Examples

standard input	standard output
5 cat bbb ololo lala icpcicpc	Case 1: 1 Case 2: 1 3 37 111 Case 3: 1 Case 4: 1 101 Case 5: 1 73 137 10001

Problem I. Emoticons

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

A berland national nanochat Bertalk should always stay up-to-date. That's why emoticons highlighting was decided to be introduced. As making emoticons to be highlighted is not exactly the kind of task one performs everyday but this task had to be done as soon as possible, the following simple rule was decided to be introduced: a round opening or closing bracket *should not* be considered part of an emoticon if:

- this is an opening bracket and there exists the nearest bracket following to the right. The nearest round bracket to the right should be a closing bracket and there shouldn't be anything between the brackets but spaces and Latin letters,
- or else it can be a closing bracket and there exists the nearest round bracket following to the left. The nearest round bracket to the left should be an opening bracket. Besides, there shouldn't be anything between the brackets but spaces and Latin letters.

If a bracket doesn't satisfy the conditions, it is considered a part of an emoticon. For example, let's consider the string "`Hi:) (it is me) I have bad news:-((`". In the string only the brackets that outline "`it is me`" aren't emoticons. Note that an opening bracket immediatelly followed by a closing bracket, i.e. "`()`", are not parts of emoticons by definition.

Your task is to print the number of brackets that are parts of emoticons in the given string.

Input

The input data consist of a single non-empty string. The length of the string does not exceed 10^5 characters. The string consists of lowercase and uppercase Latin letters, spaces, round brackets and punctuation marks: "`-`", "`:`", "`,`", "`;`". The string does not begin with and does not end with a space.

Output

Print a single number — the required number of brackets that are part of emoticons.

Examples

standard input
`Hi:) (it is me) I have bad news:-((`
standard output
`3`

standard input
`((two plus two equals four))`
standard output
`2`

Problem J. Multiswap Sorting

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You are employed to implement a new fast sorting algorithm called multiswap sorting. The basic idea is simultaneous execution of multiple parallel swaps. You are given an array containing n integer elements $a_1, a_2, \ldots, a_n$. At each step of the algorithm you must select one or more nonintersecting pairs of elements and swap the elements in each of the selected pairs.

For example, you are given the array [5, 4, 3, 2, 1]. At one step you can select two pairs $(5, 1)$ and $(4, 2)$, swap elements in them and get the array 1, 2, 3, 4, 5. Pairs $(1, 2)$ and $(2, 3)$ cannot be selected at one step, because they have the common element 2. So it is possible to sort the array [5, 4, 3, 2, 1] in one step.

Sort the given array in the minimum possible number of steps carrying out selection of pairs at each step optimally. Note that you are not required to minimize the total number of single swaps but the number of steps.

Input

The first line contains an integer n ($1 \le n \le 1000$) — the number of elements in the array. In the second line the elements a_i are given. The numbers a_i are integers not exceeding 10^9 by absolute value.

Output

In the first line output the minimum number of steps k. The next k lines should describe multiswaps in the form «$p\ i_1\ j_1\ i_2\ j_2\ \ldots i_p\ j_p$», where $p > 0$ is a number of pairs selected at the current step, $i_s\ j_s$ are the indices of elements in the s-th pair ($i_s \ne j_s$, indices of elements in distinct pairs must be distinct). The elements are indexed by integers from 1 to n according to their positions in the array *at the current step*. The order of pairs and the order of elements in pairs are unimportant. If there are multiple solutions with the minimum number of steps, output any.

Examples

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

Note

In the last example, after the first step the array takes the form 1, 3, 2, 2. At the second step 3 is swapped with the last 2. Note that the swap of the 3-rd and the 4-th elements at the first step does not change the array (these elements are equal). However, the answer with this pointless swap, as well as without it, is optimal, because your goal is to minimize the number of steps but not the number of swaps.

Problem K. Traffic Lights

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

The main street of Berland's capital is a segment with length s. The main street has traffic lights installed along it. The traffic lights have been functioning since time immemorial, cyclically changing colors from red to green and vice versa. Each traffic light can be described by four numbers x_i, r_i, g_i, d_i, which stand for:

- x_i — the distance from the beginning of the street to the traffic light ($1 \le x_i \le s-1$),
- r_i — the duration of the red light illumination ($10 \le r_i \le 20$),
- g_i — the duration of the green light illumination ($10 \le g_i \le 20$),
- d_i — the minimum non-negative moment of time when the traffic light changes the light from green to red ($0 \le d_i < r_i + g_i$).

Each traffic light retains its light cycle from the ancient past.

The King of Berland asked the transport minister to customize the traffic lights according to a "green wave" principle. That means that if you start driving from the beginning of the street at the recommended speed, you can drive through the entire street without stopping, that is, you pass each traffic light when it has the green light on.

Now it is time to show the "green wave" to the King, but the work is not even started yet. You may assume that the King starts driving at the moment of time 0 from the beginning of the street. So the minister decided to choose some speed v_0 and tell the king that the "green wave" works specifically for this speed. Moreover, they can switch any number of traffic lights to the "always green" mode. The minister' aim is to ensure that if the King drives through the street at the recommended speed v_0 he encounters no red traffic light. Driving exactly at the moment when the colors are changed is not considered driving through the red light.

Any transport's maximum speed is limited in Berland: it should not exceed v_{max}. On the other hand, the King will be angry if the recommended speed is less than v_{min}. Thus, the minister should choose such value of v_0, which satisfies the inequation $v_{min} \le v_0 \le v_{max}$.

Help the minister to find such v_0 value, that the number of traffic lights to switch to the "always green" mode is minimum. If v_0 is not uniquely defined, choose the maximum possible value of v_0.

Input

The first line of the input data contains four integers n, s, v_{min} and v_{max} ($1 \le n < 20000, 1 \le s \le 20000, 10 \le v_{min} \le v_{max} \le 50$), where n is the number of traffic lights on the street, s is the length of the street in meters, v_{min} and v_{max} are speed limits in meters per second.

Then n lines contain descriptions of the traffic lights, one per line. Each description consists of four integer numbers x_i, r_i, g_i, d_i ($1 \le x_i \le s-1, 10 \le r_i, g_i \le 20, 0 \le d_i < r_i + g_i$), where x_i, r_i, g_i, d_i are explained above. Values r_i, g_i, d_i are given in seconds and x_i — in meters. No two traffic lights are located at one point.

Output

On the first line, print the sought value v_0 containing no less than 10 digits after the decimal point. On the second line, print the number of traffic lights that need to be switched to the "always green" mode for the found v_0. The third line should contain the numbers of those traffic lights. It doesn't matter you print empty third line or print only two lines in case of no traffic lights to switch. The traffic lights are numbered starting from 1 in the order in which they appear in the input data. Print the numbers of the traffic lights in any order.

Examples

standard input	standard output
3 1000 10 30 500 10 10 10 501 10 10 0 600 10 10 0	16.7000000000 0
2 1000 10 30 500 10 10 10 600 10 20 2	25.0000000000 0
4 1000 10 30 800 10 15 20 500 20 10 15 501 20 10 5 600 10 20 15	20.0400000000 1 2

Problem L. BR Privatization

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Berland Railways (BR) is about to finish their existence as public property. Privatization is approaching.

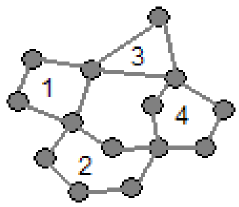
Currently, BR consists of n stations, some pairs are connected by bidirectional roads. Between each pair of stations there is no more than one road and any road connects exactly two different stations.

Since each Berland railway network was developed independently, the topology of BR has an interesting form. BR consists of areas. Each area is a set of stations connected in a cycle in some order. Each area borders with exactly two different other areas and shares a single station with each of them. So exactly two stations in each area are shared by neighbouring areas. Thus, each station

- either belongs to one area and has two roads going from it
- or belongs to two neighbouring areas at the same time and has four roads going from it.

Since the BR form a connected network, one can say that the areas also form a single cycle of areas.

The figure below illustrates BR's possible structure.



In this case BR has 14 stations, 18 roads and consists of four areas numbered from 1 to 4:

- the first one borders with the second and the third and has four stations;
- the second one borders with the first and the fourth and has six stations;
- the third one borders with the first and the fourth and has three stations;
- the fourth one borders with the second and the third and has five stations.

Two major companies are planning to buy most of the BR stations. However the antitrust committee has its own requirements. No two stations connected by a road should belong to the same company. It frustrates the owners of the first and the second company, they may not even be able to purchase all the stations without violating this condition.

Both companies have decided to buy stations in such a way that the total number of stations purchased by both companies is maximal. If there are multiple ways for such distribution, they agree on any of them.

Help them to carry out their plans, write a program that finds the desired buying plan. Note that the information about BR is given to you as a list of stations and roads, but the information about areas is not explicitly defined.

Input

The first line contains a pair of integers n, m ($6 \le n \le 10^5; 9 \le m \le 10^5$), where n is the number of stations, and m is the number of roads. All stations are arbitrarily numbered from 1 to n. Then m lines follow, each containing a pair of integers a_i, b_i ($1 \le a_i, b_i \le n; a_i \ne b_i$) and indicating the road that connects the stations with numbers a_i and b_i. The roads are given in arbitrary order.

BR consists of at least three areas, each contains at least three stations.

Output

In the first line print non-negative integer number n_1 — the number of stations to be purchased by the first company. Then n_1 numbers follow — the numbers of stations to be purchased by the first company.

In the second line print non-negative integer number n_2 — the number of stations to be purchased by the second company. Then n_2 numbers follow — the numbers of stations to be purchased by the second company.

If there are multiple solutions, print any of them.

Examples

standard input	standard output
14 18 1 2 1 4 1 7 1 8 2 4 3 4 3 14 4 5 5 6 6 14 7 11 7 10 7 9 8 9 10 12 11 14 12 13 13 14	6 2 5 7 8 12 14 7 1 3 6 9 10 11 13
6 9 1 2 1 6 2 6 2 4 2 3 3 4 4 6 4 5 5 6	2 2 5 2 1 3

Note

The first example corresponds to the illustration from the problem statement.