Problem A. Automated Telephone Exchange

Input file:	`ate.in`
Output file:	`ate.out`
Time limit:	3 seconds
Memory limit:	256 megabytes

In St Petersburg phone numbers are formatted as "XXX–XX–XX", where the first three digits represent index of the Automated Telephone Exchange (ATE). Each ATE has exactly 10 000 unique phone numbers.

Peter has just bought a new flat and now he wants to install a telephone line. He thinks that a phone number is lucky if the arithmetic expression represented by that phone number is equal to zero. For example, the phone number 102–40–62 is lucky ($102 - 40 - 62 = 0$), and the number 157–10–47 is not lucky ($157 - 10 - 47 = 100 \neq 0$).

Peter knows the index of the ATE that serves his house. To get an idea of his chances to get a lucky number he wants to know how many lucky numbers his ATE has.

Input

The input file contains a single integer number n — the index of Peter's ATE ($100 \leq n \leq 999$).

Output

Output a single integer number — the number of lucky phone numbers Peter's ATE has.

Examples

ate.in	ate.out
196	3
239	0

Problem B. Black Square

Input file:	`black.in`
Output file:	`black.out`
Time limit:	3 seconds
Memory limit:	256 megabytes

Inspired by Kazimir Malevich's masterpiece "Black Square", Peter Palevich is planning to create his own version. He prepared a rectangular grid containing $m \times n$ white cells arranged in m rows of n cells each.

Peter painted some of the cells black, so that the black cells formed a square of size $s \times s$ cells. But later that day Peter became disappointed with his painting and destroyed it, cutting it to horizontal stripes of size $1 \times n$ and burning them in the fireplace.

Next morning Peter changed his mind and decided to restore his painting. He tried to find its remains in the fireplace, and fortunately one of the stripes, namely the k-th from the top, survived the fire.

Now Peter wonders whether it is possible to restore the painting based on this stripe. Help him to do it.

Input

The first line of the input file contains four integer numbers: m, n, s and k ($1 \le m, n \le 5000$; $1 \le s \le \min(m, n)$; $1 \le k \le m$).

The second line contains n characters and describes the k-th line of the painting, '`.`' stands for a white cell, '`*`' stands for a black cell.

Output

If the initial painting can be uniquely restored, output "`Unique`".

If there are several paintings that could have been painted by Peter, output "`Ambiguous`".

If there are no possible paintings, output "`Impossible`".

Examples

black.in	black.out
4 4 1 2 ..*.	Unique
4 4 2 2 ..**	Ambiguous
4 4 3 2 .*.*	Impossible

Problem C. Cube Coloring

Input file:	`cube.in`
Output file:	`cube.out`
Time limit:	3 seconds
Memory limit:	256 megabytes

In a distant galaxy called Ẍöş, there lives a robot named A. C. Magnifizer. Mr. Magnifizer — or "A. C. M.," as he prefers to be called — is extremely happy when he is at work coloring beautiful shapes.

Yesterday, an old artist came to Mr. Magnifizer and offered him a job. The artist needs a cube to be colored, which is a significant part of his next installation. But he has only the *vision* of what the cube should look like.

A *face vision* is what the cube should look like when observing a certain face of it. It consists of the color of the face which the observer is looking at (the "main" face), followed by the list of colors of the faces which have common edge with the main face. The order of adjacent faces is not fixed, because the face vision defines the impression, not the particular details.

The vision of the cube consists of the face visions of all the cube's faces, again, in no particular order.

Although Mr. Magnifizer is experienced in coloring cubes, as well as other shapes, this problem seems very difficult to him. Help him! Given the vision of the cube, find a way to color a cube to satisfy it.

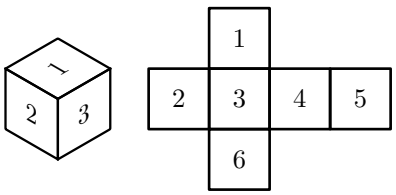
Input

The single line of the input file contains six words. Each word describes the face vision of a certain face of the cube. The description consists of five English uppercase letters — the color of the main face (the first letter) and the colors of its adjacent faces. Equal letters define equal colors and different letters define different colors.

Output

If it is impossible to color the cube because some of the visions contradict each other, output "`Impossible`".

If there is a single way to color the cube (the ways that can be turned one into another by rotating the cube are considered equal), output the colors of the faces in a single line, from the first face to the sixth one, as shown on the picture below. Any rotation of the cube is acceptable.



If there are multiple ways to color the cube, output any two of them in the format described above, each in its own line.

Examples

cube.in	cube.out
ABCDE FGHIJ KLMNO PQRST UVWXY ZABCD	Impossible
OZZZZ ZOZZZ ZZOZZ ZZZOZ ZZZZO ZZZZZ	ZOZZZZ
ABCOO BCAOO CABOO OOOAB OOOBC OOOCA	ABCOOO ACBOOO

Problem D. Dice

Input file:	dice.in
Output file:	dice.out
Time limit:	3 seconds
Memory limit:	256 megabytes

Along with other things, Feadagor is fond of playing tabletop role playing games. He has just discovered a new game and he'd like to play it with his friends. Unluckily, he cannot call his friends right now, as the game requires quite an unusual set of dice. The description of the game says that there must be n dice, and i-th die must have a_i faces. Each die must be shaped so its faces fall equiprobably.

According to the game manual, the numbers from 1 to m, where $m = \sum_{i=1}^{n} a_i$, must be written on the faces, and each number from the interval must be used exactly once. The numbers arrangement must be chosen so that if you throw all the dice simultaneously, then the mathematical expectation E of the total value in such experiment will be maximal.

The manual says that only Maiar have enough wisdom to arrange the numbers properly (and therefore your only choice is to buy the dice just for 133 dollars, telepathy is quite expensive now). But maybe there is a simpler way to make the proper arrangement?

Input

The first line of the input file contains a single integer number n ($1 \le n \le 1000$). The following line contains n integer numbers a_1, a_2 ... a_n ($1 \le a_i \le 100$).

Output

The first line of the output file must contain maximal possible expectation E — a floating-point number precise up to 5 digits after the decimal point.

The following n lines must contain the numbers arrangement: i-th line must contain a_i integer numbers — the numbers to be written on the faces of i-th die.

Example

dice.in	dice.out
2 1 4	7.50000 5 1 2 3 4

Problem E. Electrification

Input file:	`electro.in`
Output file:	`electro.out`
Time limit:	3 seconds
Memory limit:	256 megabytes

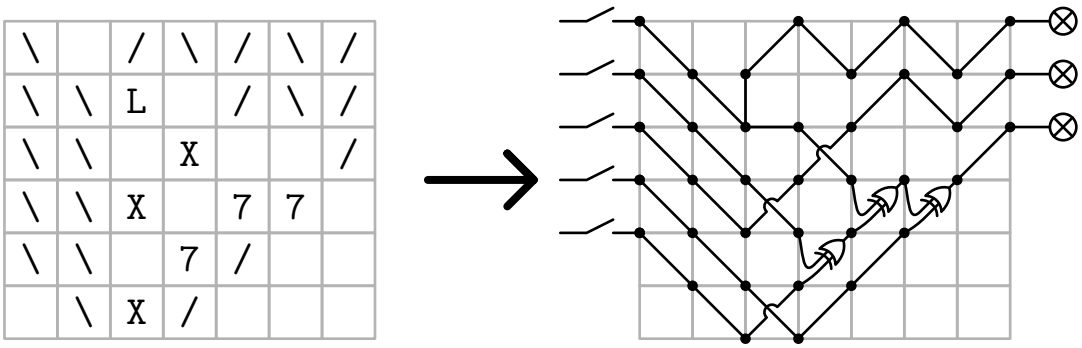
Ellen is a technologist at Advanced Circuit Mastership, and she's currently designing an electrical scheme for a light management system in a luxury hotel room.

A scheme will be embedded in the ceiling of the room. It will be a rectangular grid with each cell containing one element. Below we provide a list of all possible elements with the description of their functions, preceded by their one-character representations.

- '`.`' – empty cell; doesn't have any input or output signals;
- '`/`' – wire; takes input signal from the lower left corner and transmits it to the upper right one;
- '`\`' – wire; takes input signal from the upper left corner and transmits it to the lower right one;
- '`X`' – intersection; takes input signals from the upper left and the lower left corners and transmits it to the lower right and the upper right corner, correspondingly;
- '`L`' – wire splitter; takes input signal from the lower left corner and transmits it both to the upper left and to the lower right corners;
- '`7`' – exclusive or; takes input signals from the upper left and the lower left corners and transmits the XOR of these two signals to the upper right corner.

The hotel room has n switches and m lamps. The switches will be connected to the n topmost corners of the left side of the scheme, and the lamps — to the m topmost corners of the right side. Ellen's task is to create the scheme connecting the switches and lamps, satisfying the following scheme design guidelines.

The room designers assigned each lamp with a set of switches. A lamp must change its state (turn on if it was off, and turn off if it was on) each time when a switch that is assigned to this lamp is flipped. A lamp mustn't change its state when a switch that isn't assigned to this lamp is flipped.



The following wire connection rules must be satisfied:

- the lower left corner of the '`L`' element must have exactly one input wire. Other two corners of this element must have exactly one output wire each;
- the upper right corner of the '`7`' element must have exactly one output wire. Other two corners of this element must have exactly one input wire each;
- a corner that is connected to a switch must have exactly one output wire;
- a corner that is connected to a lamp must have exactly one input wire;
- a corner that is on the left border and isn't connected to a switch, as well as a corner that is on the right border and isn't connected to a lamp, must have no input and no output wires;
- each other corner must have either no input and no output wires, or exactly one input and exactly one output wire.

Note that the set of possible elements is designed so that cyclic dependencies are impossible.

Ellen is given the switches assignment. Help her design an appropriate scheme.

Input

The first line of the input file contains two integer numbers n and m — the number of switches and lamps located in the room being designed ($1 \le n, m \le 10$).

The following n lines contains m characters each. The j-th character in the i-th line shows whether the i-th switch is assigned to the j-th lamp. Digit 1 denotes "assigned", digit 0 denotes "not assigned".

Each lamp is assigned with at least one switch, and each switch is assigned to at least one lamp.

Output

The first line of the output file should contain two integer numbers h and w, the dimensions of the scheme ($\max(m, n) - 1 \le h \le 1000$; $1 \le w \le 1000$).

The following h lines should contain w characters each. The only allowed characters are '`.`', '`/`', '`\`', '`X`', '`L`', and '`7`'.

The scheme must be valid (in terms of wire connection rules), and satisfy the assignments given in the input file.

If there are multiple satisfying schemes, output any one of them.

Examples

electro.in	electro.out
5 3 101 001 010 001 001	6 7 \./\/\/ \\L./\/ \\.X../ \\X.77. \\.7/.. .\X/...
2 2 01 10	1 1 X

Problem F. Flat

Input file:	`flat.in`
Output file:	`flat.out`
Time limit:	3 seconds
Memory limit:	256 megabytes

You are one of the developers of software for a real estate agency. One of the functions you are to implement is calculating different kinds of statistics for flats the agency is selling. Each flat consists of different types of rooms: bedroom, bathroom, kitchen, balcony and others.

The cost of the flat is equal to the product of reduced total area and the cost of one square metre. Reduced total area is the total area of all rooms except for balconies plus one half of balconies total area.

You will be given some information about the area of each room in the flat and the cost of one square metre. You are to calculate the following values for the flat:

- the total area of all rooms;
- the total area of all bedrooms;
- the cost of the flat.

Input

The first line of the input file contains two integer numbers n ($1 \le n \le 10$) and c ($1 \le c \le 100\,000$) — number of rooms in the flat and the cost of one square metre, respectively.

Each of the following n lines contains an integer number a_i ($1 \le a_i \le 100$) and a word t_i — the area of i-th room and its type, respectively. Word t_i is one of the following: "`bedroom`", "`bathroom`", "`kitchen`", "`balcony`", "`other`".

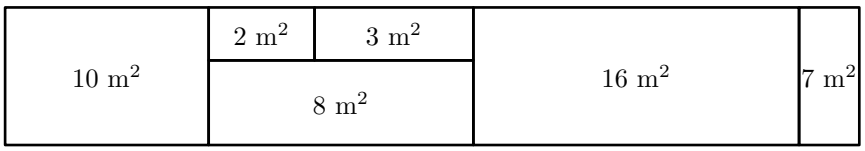
Output

The first line of the output file should contain one integer number — the total area of all rooms of the flat. The second line of the output file should contain one integer number — the total area of bedrooms of the flat. The third line of the output file should contain one real number — the cost of the flat with precision not worse than 10^{-6}.

Examples

flat.in	flat.out
6 75000 8 other 3 bathroom 2 bathroom 10 kitchen 16 bedroom 7 balcony	46 16 3187500
2 75123 10 kitchen 15 balcony	25 0 1314652.5

The figure shows the flat from the first example.



Problem G. Galaxy Interconnection

Input file:	`galaxy.in`
Output file:	`galaxy.out`
Time limit:	3 seconds
Memory limit:	256 megabytes

> And then, eight hundred and eight years ago, there occurred an event of such great importance that it marked a new era in the history of mankind — the Era of the Great Circle.
>
> Andromeda: A Space-Age Tale
> Ivan Yefremov

More than 1000 years passed from the first ACM ICPC. Several civilizations of our Galaxy, including the Earth, are united in the *Great Circle* whose members exchange and relay scientific and cultural information. Due to the distribution of gravity and dark energy flows in galaxy, there are bidirectional *communication channels* between some planets. Via these channels, civilizations can distribute knowledge across all the Great Circle.

It's now the time to distribute researches between civilizations: each planet has to choose one of k main research areas (such as *Repagular Calculus* or *Given String Theory*).

The choice of k was not accidental. The Great Circle started from *Initial Circle* — a set of k planets that were connected with channels forming a cycle. Later new channels were investigated and built and new planets were connected to the Great Circle. However due to high energy cost of communication channels, each planet of the Great Circle has strictly less than k channels.

Two planets connected with direct channel shouldn't choose the same research area: they'd better choose different ones and share research results.

There is one more limitation. Periodically civilizations send space expeditions in order to explore new planets and visit neighbors from the Great Circle. Expeditions are planned in such a way that spaceships can fly from one planet to another if these planets are connected with communication channel. This helps to prepare the expedition: to build refuel stations, optimize ship communication systems, etc.

Some expeditions, so-called *Research Audits*, are planned in such a way that a space ship starts from some planet, then makes $k - 1$ jumps, visiting total of k planets (including origin). These k planets together should provide all k research areas: the expedition will check the progress of researching.

Your task is to choose one research area for each planet in such a way that:

- there are no two planets connected by communication channel with the same research area;
- it is possible to send Research Audit expedition from every planet of the Great Circle.

Input

First line of the input file contains three integer numbers: n, k and m — the number of planets in the Great Circle, the number of research areas and the number of communication channels correspondingly ($3 \le n \le 5000$; $3 \le k \le \min(n, 10)$; $1 \le m \le 10000$).

The following m lines describe channels, one per line. Communication channel is described by two integer numbers — identifiers of the planets it connects. Planets are identified by integer numbers from 1 to n in such a way that planets from 1 to k form the Initial Cycle.

Output

Output exactly n integers in one line, i-th integer should identify a research area for planet i (an integer number from 1 to k).

Examples

galaxy.in	galaxy.out
5 4 5 1 2 2 3 3 4 4 1 5 3	1 2 3 4 2
6 4 9 1 2 2 3 3 4 4 1 5 2 5 3 6 5 6 4 6 1	1 2 3 4 4 2

Problem H. High security

Input file:	high.in
Output file:	high.out
Time limit:	3 seconds
Memory limit:	256 megabytes

Vasya is a system administrator of a new social network, named *Five Contacts* or just *V Contacts*.

High security is very important in social networks. One of the problems is spammers, who can steal users' passwords and send spam messages or do other bad things.

According to security policy of V Contacts each password consists of five lowercase or uppercase English letters or digits. Vasya wants to analyze a database of all users' passwords in order to know its security level. The security level depends on how many pairs of passwords are the same, differ in one position, two positions, etc.

Input

The first line of the input file contains one integer number n — the total number of users of V Contacts ($1 \le n \le 50\,000$).

Each of the following n lines contains a single user password.

Output

The output file should contain six numbers a_i ($0 \le i \le 5$) — the number of pairs of passwords which are different in exactly i positions.

Examples

high.in	high.out
3 abcde ABCDE 12345	0 0 0 0 0 3
7 aaaaa aaaaa aaaa1 aaaA2 aaBCD a6543 XxXxX	1 2 3 4 5 6

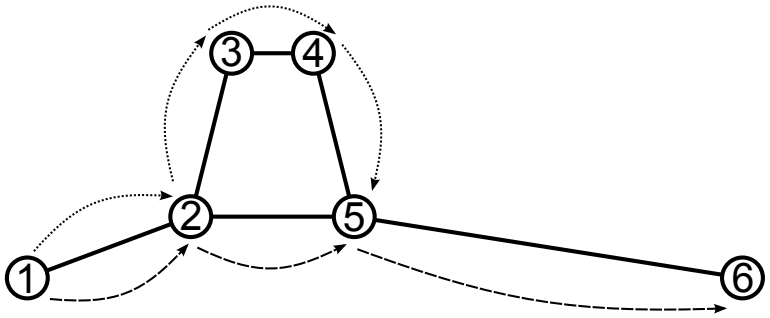
Problem I. Immediate Delivery

Input file:	immediate.in
Output file:	immediate.out
Time limit:	3 seconds
Memory limit:	256 megabytes

Mike and John are delivery boys of *Immediate Delivery*. One day they were asked to deliver a lot of packages all over the city.

The transport system of the city they work in consists of junctions and roads connecting these junctions. All roads are bidirectional and all junctions are accessible from each other (directly or indirectly).

Mike and John should visit all the junctions to deliver all packages. They want to split this task into two parts in a way that minimizes the time of the last delivery.



Input

The first line contains two integer numbers n and m, the number of junctions and roads ($1 \le n \le 18$).

The following m lines describe roads. Each line contains three integer numbers: x_i and y_i ($1 \le x_i, y_i \le n$), the numbers of junctions connected with this road and t_i ($1 \le t_i \le 1000$), the time required to drive it. There is at most one road connecting any two junctions. No road connects a junction to itself.

The office of the Immediate Delivery is placed at the junction number 1.

Output

The first line of the output file must contain the minimal possible time when the last package can be delivered.

The second line must contain the route for Mike: the number of roads p, traveled by Mike and $p + 1$ numbers of junctions he visited in order they were visited, starting with junction number 1.

The third line must contain the route for John in the same format.

Examples

immediate.in	immediate.out
6 6 1 2 10 2 3 10 3 4 5 4 5 10 5 6 20 2 5 10	40 3 1 2 5 6 4 1 2 3 4 5

Problem J. John's Inversions

Input file:	john.in
Output file:	john.out
Time limit:	3 seconds
Memory limit:	256 megabytes

John has recently come across the definition of an inversion.

An *inversion* in a sequence of numbers s_k is any pair s_i, s_j such that $i < j$ and $s_i > s_j$.

John believes that inversions are a perfect tool for estimation of how well a sequence of numbers is sorted. The smaller the number of inversions in the sequence, the better it is sorted. For example, the sequences sorted in the ascending order have zero inversions in them.

Peter gave John a stack of n cards. Each card has two numbers written on it — one in red and one in blue color. John is anxious to apply his knowledge of inversions to that stack.

He places the cards in front of him in arbitrary order and calculates the total number of *nice inversions* in front of him. John considers an inversion to be nice if it consists of the numbers of the same color. In our case nice inversion can be formed by either two blue or two red numbers. If the number of nice inversions is too big by John's standards, he rearranges the cards and repeats the process.

Your task is to help John find out the minimal possible number of nice inversions he can get while following his algorithm.

Input

The first line of the input file contains one integer number n — the number of cards in the deck ($1 \le n \le 100\,000$). The following n lines describe one card each. The i-th line contains two integer numbers r_i and b_i ($1 \le r_i, b_i \le 10^9$) — the numbers written on i-th card in red and blue colors respectively.

Output

Output should contain exactly one integer number — the minimal possible number of nice inversions.

Examples

john.in	john.out
3 10 3 20 2 30 1	3
4 2 2 5 25 2 1 10 9	1

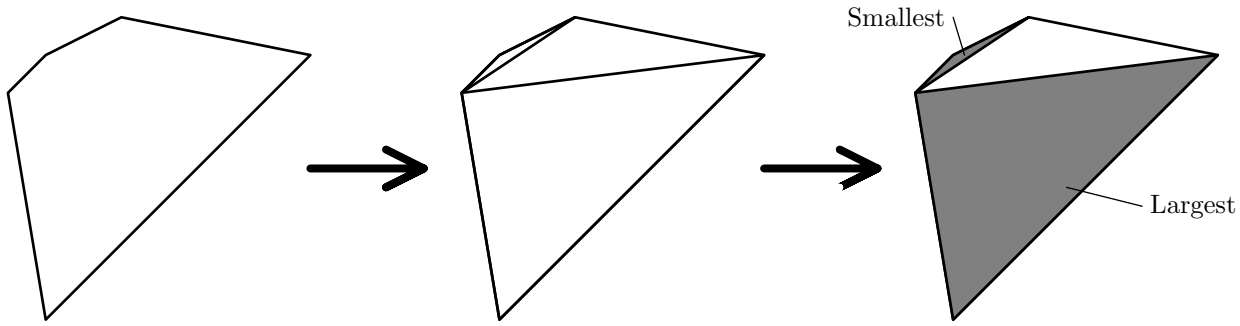
Problem K. Kids Like Cakes

Input file:	`kids.in`
Output file:	`kids.out`
Time limit:	5 seconds
Memory limit:	256 megabytes

Kids like cakes. It's obvious. In this problem, we consider only cakes having the form of a convex polygon, as viewed from above.

Every cake needs to be split into pieces. In this problem, every piece should have a form of a non-degenerate triangle with vertices at original cake's vertices. The pieces may not intersect, and their union should form the original cake.

Some kids also like fairness. Let's call the *unfairness number* of a cake the maximal possible difference between the areas of the largest and the smallest pieces of it.



Your task is to find the unfairness number of a given cake.

Input

The first line of the input file contains a single integer number n — the number of vertices of the cake ($4 \le n \le 5000$). The following n lines contain two integers x_i, y_i each — the coordinates of the vertices ($-10^8 \le x_i, y_i \le 10^8$).

Output

The first line of the output file must contain a single number with exactly one digit after the decimal point — the unfairness number of the cake.

The following two lines must describe the way to split the cake to get such unfairness number. The first of them should contain three indices of vertices of the largest piece in it. The second one should contain three indices of vertices of the smallest piece. Vertices are numbered from 1 to n.

Example

kids.in	kids.out
5 0 0 -1 6 0 7 2 8 7 7	24.0 2 5 1 2 3 4