Problem A. Valya and Letter

Time limit:	1 second
Memory limit:	512 megabytes

Valya got bored with social media, so she decided to write a good old handwritten letter to her friend. Valya wrote the letter on a rectangular sheet of paper. The sheet's dimensions are equal to n and m cm. Then she found a rectangular envelope with dimensions equal to h and w cm.

Unfortunately the letter could be too big to be placed into the envelope. So she might need to fold it several times. In one move Valya can fold the letter in half vertically or horizontally.

After folding the letter if necessary, Valya would put it into the envelope. Valya is a perfectionist, so she would always put the letter into the envelope in such way that its sides are parallel to the envelope's sides. The letter can be placed into the envelope if its corresponding dimensions don't exceed the envelope's dimensions. Before placing the letter Valya can rotate it by 90 degrees. For example, if the letter's dimensions are equal to 10 and 20 cm, and the envelope's dimensions are equal to 20 and 10 cm, Valya doesn't need to fold the letter, she can rotate it by 90 degrees to fit it into the envelope.

Valya doesn't want the letter to be crumpled so she wants to fold it the minimal possible number of times. Help her find this number.

Input

The first line of the input contains four integers n, m, h and w denoting the dimensions of the letter and dimensions of the envelope, respectively $(1 \le n, m, h, w \le 10^{18})$.

Output

Output one number — the minimal number of folds required, so that Valya could put the letter into the envelope.

Examples

standard input	standard output
10 20 20 10	0
3 3 2 2	2

Problem B. Sightseeing Tour

Time limit:	2 seconds
Memory limit:	512 megabytes

A group of n friends has decided to take a tour. They can visit some of m cities during the tour.

The tour guide asked each person to tell her his wishes about visiting cities. Each person can claim for some cities that he wants to visit them, and for some other cities that he wants to avoid visiting them.

The group always travels together. If the group visits some city, all people who claimed that they want to avoid visiting that city get upset. If the group doesn't visit some city, all people who claimed that they want to visit that city get upset.

The guide understands that sometimes it is not possible to satisfy all wishes. She wants to make a plan which cities to visit, so that each person gets upset at most once.

Help the guide to choose which cities to visits to satisfy all wishes, except at most one for each person, or find out that it is impossible.

Input

The first line of input contains three integers: n, m and k — the number of friends, the number of cities and the total number of wishes ($1 \le n, m, k \le 100\,000$).

Each of the following k lines contains two integers a and b and describes a wish ($1 \le a \le n, 1 \le |b| \le m$). If $b > 0$, the person a wants to visit the city b. If $b < 0$, the person a wants to avoid visiting the city $-b$. No wish is listed more than once, no person simultaneously wants to visit some city and to avoid visiting it.

Output

If there is no solution, output -1.

In the other case the first line of output must contain a single integer k — the number of cities to visit by the group.

The second line must contain k integers — the numbers of the cities to visit. They can be listed in any order.

If there are several possible correct answers, any of them can be printed. Note that you need not neither maximize nor minimize k, you can output any correct answer.

Examples

standard input	standard output
3 5 6 1 2 1 3 1 -4 2 3 2 4 2 5	3 2 3 5
3 3 6 1 -1 1 2 2 -2 2 3 3 -3 3 1	0

Problem C. How to Fail at Programming Contest

Time limit: 1 second
Memory limit: 512 megabytes

Gennady is the best in competitive programming. He can solve any problem, so he has never lost a contest. But today he has decided to lose a contest because winning all contests is not interesting.

But Gennady can't abandon solving problems because it is unsportsmanlike behavior. So he has decided to just choose a bad strategy that minimizes his total points for the contest.

There are n problems in the contest numbered from 1 to n. If a contestant solves problem i he gets p_i points. Gennady has read all problems and came up with a solution for each one. He knows that for the problem i he needs exactly t_i minutes to write a solution. The final thing to do is to choose the order to write the solutions for the problems. Gennady noticed that he had T minutes remaining until the end of the contest.

Gennady wants to use the following strategy. He chooses a problem that he hasn't solved yet and writes a solution for it. He never chooses the problem that he can't finish in time. When the solution is ready, Gennady submits it and gets p_i points for this problem. Submitting and testing doesn't take any time. Then he moves to another problem. When Gennady understands that he can't solve any of the remaining problems in time he stops coding.

Now Gennady wants to choose the order of solving problems that minimizes his score for the contest. Help him to find out the smallest number of points that he can get following the rules above.

Input

The first line of input contains two integers n and T denoting the number of problems and the time until the end of the contest ($1 \le n, T \le 2000$).

The following n lines describe the problems: the i-th line contains two integers t_i, p_i denoting the time needed for Gennady to solve this problem and the number of points this problem costs ($1 \le t_i \le 2000$, $1 \le p_i \le 10^6$).

Output

Output one number — the minimal number of points that Gennady can get.

Examples

standard input	standard output
4 9 4 2 4 5 3 4 2 10	7
1 1 2 1	0

Problem D. Berland Railroads

Time limit: 1 second
Memory limit: 512 megabytes

Berland government has decided to build a system of high speed railroads in the country. There are n cities in Berland, numbered from 1 to n. Some pairs of cities will be connected by a high speed railroad. Travelers will be able to move by train in both directions along each road.

High speed trains are the future of transport, so it is required to create such plan of railroad construction, that it is possible to get from any city to any other using railroads. To save money it was decided to build the minimal possible number of railroads to satisfy this criteria.

The government of the i-th city wants to support exactly d_i trains. So the number of high speed railroads connecting the i-th city to others must be exactly d_i. Fortunately it turned out that $d_1 + d_2 + \ldots + d_n = 2n - 2$.

The government wants people to travel between cities using as few changes between trains as possible. You are asked to create such plan of railroads that the maximal number of changes needed to travel from a city to another was minimal.

Input

The first line of input contains an integer n — the number of cities in Berland ($2 \le n \le 200\,000$). The next line contains n integers: $d_1, d_2, \ldots, d_n$ ($1 \le d_i < n$). The number of cities connected to the i-th city must be d_i for each i. It is guaranteed that $d_1 + d_2 + \ldots + d_n = 2n - 2$.

Output

Output $n - 1$ lines — the descriptions of the railroads in the optimal plan. Each line must contain two integers s_i and f_i — cities to be connected by a railroad, $1 \le s_i, f_i \le n$, $s_i \ne f_i$.

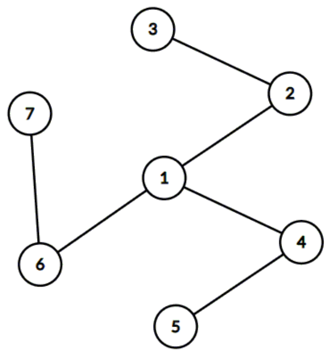
The number of cities connected to the i-th city must be equal to d_i for all i. The maximum number of train changes needed to travel from a city to another must be minimal possible. If there are several optimal plans, output any of them. It is guaranteed that there is at least one plan that satisfies all conditions.

Examples

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

Note

The optimal answer for the first sample test is the following:



It is possible to get from any city to any other using railroads. Each city has the required number of connected cities, for example the city 1 is connected to three cities: 2, 4 and 6 ($d_1 = 3$).

The maximum number of train changes needed to get from a city to another is for example for cities 3 and 5, 3 transfers are needed. First you must travel from city 3 to city 2, then from city 2 to city 1, then from city 1 to city 4, and finally from city 4 to city 5. There is no plan that requires fewer changes for all pairs of cities.

Problem E. Difficult Problems

Time limit: 1 second
Memory limit: 512 megabytes

Yesterday, for the first time, Tosha took part in a programming contest. But the problems were so difficult that sometimes he wanted to scream. Tosha knew that noise during the contest is prohibited, so he had to scream on paper: he sometimes wrote a few letters "A" on a piece of paper, when he felt that the task was too difficult. The more difficult the problem was, the more letters "A" Tosha wrote down in the process of solving it.

The next day Tosha wanted to boast to his classmates that he had participated in the contest and solved a lot of problems. But he forgot the number of problems and even didn't have the statements to check. Fortunately, Tosha saved his notes, so now he can roughly estimate the number of problems.

He remembers that all the problems had different non-zero difficulty, which means that solving each task he wrote a distinct positive number of letters "A". And these screaming letters conveniently stands out, because there are no other uppercase letters, all other notes he made in lowercase. Note that he could write down some lowercase notes between "A"-s he wrote for the same problem.

Help Tosha to find out what is the the maximum number of problems that could have been there in the contest.

Input

Input contains a nonempty string s consisting of lowercase English letters and characters "A". The length of s does not exceed 10^6, it contains at least one "A".

Output

Print one integer — the maximum number of problems in the contest.

Example

standard input	standard output
dfsAAfftaAbcdAAtoshaAtoAApA	3

Problem F. Race of robots

Time limit: 1 second
Memory limit: 512 megabytes

The annual race of robots takes place today in Berland. Ivan is one of it's participants. Before the race starts, he decided to find out how does the racing field look like.

Racing field is a rectangle of size $n \times m$ divided into nm unit cells. Rows of the rectangle are numbered from 1 to n, columns are numbered from 1 to m. We denote as (i, j) the cell located at the row i and the column j. There can be a barrier between each pair of adjacent cells.

There are n robots participating in the race. Before the race starts, they are located in cells of the first column: $(1, 1), (2, 1), \ldots, (n, 1)$. Finish is located in the cell (n, m). When the race begins, each robot starts moving towards the finish. From the cell (i, j) robot can move either to the cell $(i + 1, j)$ or to the cell $(i, j + 1)$. If the robot is moving between cells not separated by a barrier, it moves between these cells immediately. Otherwise, it takes one second to move between them.

Each robot knows locations of all barriers in advance and is moving to the finish cell using the path that needs minimum time to follow. Note that robots are very small, so any number of them may stay in one cell during the race. We denote the field *fair*, if the time needed to reach the finish is equal for all robots.

Ivan managed to take a look on the racing field and remembered some information about it. Namely, for some pairs of adjacent cells he definitely knows whether there is a barrier between them or not. Also he is sure that the racing field is fair. Ivan is wondering to know, how many fields satisfying all these requirements do exist. Unfortunately, now he is busy preparing to the competition, so he asked you to figure it out. As the answer might be very huge, Ivan only wants to know it modulo 998244353.

Input

The first line of input contains two integers n and m — the size of the field ($1 \le n \le 5$, $1 \le m \le 15$).

The second line contains a single integer k — the number of pairs of cells Ivan knows the information about ($0 \le k \le 2nm - n - m$).

Each of the next k lines contains five integers r_1, c_1, r_2, c_2 and w ($1 \le r_1 \le r_2 \le n$, $1 \le c_1 \le c_2 \le m$, $0 \le w \le 1$). If $w = 0$, then there is no barrier between cells (r_1, c_1) and (r_2, c_2). If $w = 1$, then there is a barrier between these cells. It's guaranteed, that cells (r_1, c_1) and (r_2, c_2) are adjacent. It's guaranteed, that no pair of cells will be mentioned in the input more than once.

Output

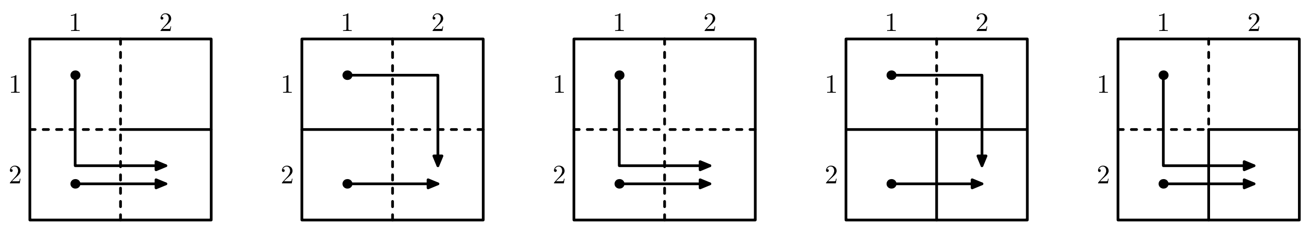
Output single integer — the number of fair fields, satisfying all Ivan's conditions, modulo 998244353.

Example

standard input	standard output
2 2 1 1 1 1 2 0	5

Note

Illustration for the sample test. Solid lines denote a barrier, dashed lines denote its absence.



Problem G. Flexible Segments

Time limit: 1 second
Memory limit: 512 megabytes

A great mathematician Vladimir Germanovich has noticed an interesting property of some segments of positive integers while exploring them for new patterns.

Vladimir calls the segment of positive integers $l, l+1, \ldots, r$ *flexible* if he is able to change every number of this segment by exactly one in such way, that the product of numbers on this segment doesn't change. That is, there exists a sequence $a_l, a_{l+1}, \ldots, a_r$ with the following properties:

- $a_k = k \pm 1$
- $l \cdot (l+1) \cdot \ldots \cdot r = a_l \cdot a_{l+1} \cdot \ldots \cdot a_r$

Now Vladimir Germanovich wants to know if he is able to build flexible segment of any length. Given positive integer n find any flexible segment that consists of n consecutive positive integers or tell that there is no such segment.

Input

The only line contains an integer n ($1 \le n \le 10\,000$) — the length of required segment.

Output

The first line of output must contain "`YES`" if there exists a flexible segment of n positive integers. Otherwise it must contain "`NO`".

If such segment exists the second and the third lines must contain the description of this segment.

The second line should contain the only integer l ($1 \le l \le 1\,000\,000$) — the first element of this segment. It is guaranteed that if a flexible segment of length n exists then there exists a flexible segment $[l; r]$ of length n such that $1 \le l \le 1\,000\,000$.

The third line should contain a string of length n without spaces. It must consist of "`+`" and "`-`" characters, the $(k - l + 1)$-th character of this string should be "`-`" if $a_k = k - 1$, or "`+`" if $a_k = k + 1$.

Examples

standard input	standard output
1	NO
4	YES 2 -+++

Note

In the second example $n = 4$, $l = 2$, $r = l + n - 1 = 5$. The answer is the following: $a_2 = 2 - 1 = 1$, $a_3 = 3 + 1 = 4$, $a_4 = 4 + 1 = 5$, $a_5 = 5 + 1 = 6$. The product of integers from l to r is $2 \cdot 3 \cdot 4 \cdot 5 = 120$. The product of a_k is $a_2 \cdot a_3 \cdot a_4 \cdot a_5 = 1 \cdot 4 \cdot 5 \cdot 6 = 120$. Thus, the segment $[2; 5]$ is flexible.

Problem H. Secret Code

Time limit: 1 second
Memory limit: 512 megabytes

Bogdan is a fan of riddles and puzzles. He asked his friend Anton to come up with a secret code that Bogdan would then decode.

Anton has decided to use a non-negative integer without extra leading zeroes as a secret code. The code must satisfy the following condition. If you consider any three consecutive digits of it as a three-digit integer, it is divisible by three.

Anton has given all digits of his secret code, as well as possibly some other digits, to Bodgan. He claims that the maximal number satisfying the above condition that can be created out of these digits is the secret code.

Help Bogdan to find out what is the secret code.

Input

Input contains 10 integers: $c_0, \ldots, c_9$, here c_i is the number of digits i that Anton has given to Bogdan ($0 \le c_i \le 100\,000$). The sum of c_i is strictly positive and doesn't exceed $100\,000$.

Output

Output the maximal integer that can be created out of these digits. It must satisfy the condition that an integer formed by any three consecutive digits is divisible by three. It is not required to use all of the given digits. Note that any one-digit or two-digit number automatically satisfies the above condition, because it doesn't have three consecutive digits. The answer must not contain extra leading zeroes: the first digit can be 0 only if the number is zero, in this case it must be the only digit.

Examples

standard input	standard output
1 2 3 0 0 0 0 0 0 0	21021
1 1 1 1 1 1 1 1 1 1	9876543210

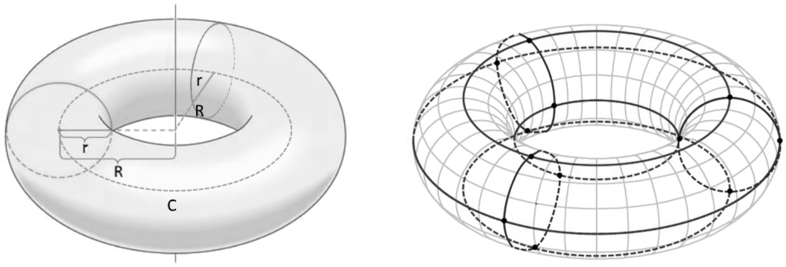
Problem I. Torus Travel

Time limit: 1 second
Memory limit: 512 megabytes

Torus is a surface that composed of points at a distance of r from a circle C with radius R in 3D space. C is then called a *central circle* of a torus; line perpendicular to the plane of C that also contais the center of C is called torus' *axis*; R and r are *major* and *minor* torus radiuses, accordingly. A circle on a torus is called *bigger* if it's center is on torus' axis and *lesser* if it's radius is r and it's plane contains torus' axis.

Young traveler Senya lives on a torus-shaped planet with major radius R and minor radius r. There is a regular road map on this planet: n lesser roads at equal distance from each other (located on lesser circles), and 4 bigger roads (on bigger circles): outer — the most distant from the axis, inner — the least distant one, and also nothern and southern equal to the central circle on the opposite sides of the planet.

Each lesser road is a property of one of n countries. Each country has only 4 cities on the intersections of it's lesser road with all 4 bigger roads.



Left picture illustrates the definition and shows minor and major radiuses. The right one shows 4 greater roads and $n = 3$ lesser roads with cities on their intersections.

Senya wants to become The Great Traveler, which means that he wants to visit every country on the planet. He considers a country visited if he has traveled along the road between two different cities of this country. Of course, movement on his planet is allowed only by the roads.

Please, help Senya to find the distance he need to cover to become The Great Traveler, if he starts his journey in a city on the inner road.

Input

The first line of input contains three integers: r, R — minor and major torus radiuses and n — number of countries ($1 \le r < R \le 10^9$; $1 \le n \le 10^9$).

Output

Output should contain one real number — minimal distance Senya has to cover. Your answer should have an absolute or relative error not greater than 10^{-9}.

Examples

standard input	standard output
1 2 1	1.570796327
1 3 4	18.849555922

Note

In the second example let i_k, s_k, n_k, o_k represent the cities of k-th country on the inner, southern, nothern and outer roads, accordingly. One of the minimal paths looks like: $i_1 \to s_1 \to s_2 \to i_2 \to i_3 \to n_3 \to i_3 \to i_4 \to s_4$. Covered distance equals to $\frac{5}{4}$ of country length, two segments of inner road and one of southern. So the answer is $\frac{5}{4} \cdot 2\pi \cdot 1 + 2 \cdot \frac{2\pi \cdot 2}{4} + \frac{2\pi \cdot 3}{4} = 6\pi \approx 18.849555922$.

Problem J. Amalthea's new walk

Time limit: 1 second
Memory limit: 512 megabytes

When she was a princess in a small chess castle, Amalthea used to walk each day in the golden ceremonial hall. The floor of the hall was divided into unit square cells. Amalthea's route was the same each day: she started on a certain cell, made certain moves from a cell to its side-wise neighbour cell, and returned to the initiall cell. Since there were no prohibitions in her childhood, she could visit the same cell several times.

After many years of such walks, the gold on the visited cells faded. She memorized the faded pattern for life.

Now, after becoming a countess in a large chess palace, she has decided to restore that pattern using silver tiles in the new golden ceremonial hall, to remind her of her childhood.

The royal sculptor laid out this pattern in silver tiles, but, in order to please the mistress, he made the pattern twice as big. Now each faded cell from the childhood pattern corresponds to 2×2 square of silver tiles. For each cell (x, y) from the pattern, he placed silver tiles into cells $(2x, 2y)$, $(2x, 2y+1)$, $(2x+1, 2y)$, and $(2x+1, 2y+1)$.

The sculptor has already been executed, but now Amalthea wants to learn how to traverse the new set of silver cells. As before, she will move only to a side-wise neighbour cell. As before, she has to return to the initial cell in the end. But now, since the palace etiquette is quite strict, she can not visit the same cell twice or more. The exception is moving to the initial cell at the last move.

To avoid the execution of the royal mathematician, find a valid route for Amalthea.

Input

The first line contains one integer n — the number of cells in the pattern from the childhood ($1 \le n \le 30\,000$).

Each of the next n lines contains two integers x_i, y_i — coordinates of the faded cell ($0 \le x_i, y_i \le 1000$).

It's guaranteed that no cell is given twice, and that the area is side-wise connected.

Output

Output $4n$ lines, in each of them output the coordinates of the respective cell of Amalthea's route in the palace. Every two consecutive cells, as well as the first and the last cell, must share a side. The route must contain all silver-tiled cells.

Example

standard input	standard output
3 0 0 0 1 1 0	0 0 0 1 0 2 0 3 1 3 1 2 1 1 2 1 3 1 3 0 2 0 1 0

Problem K. Formula 42

Time limit: 1 second
Memory limit: 256 megabytes

A new racing track is being constructed for Flatland Grand Prix Formula 42 race. Well known track architect Herman Kilkin was invited, and he has prepared the track plan. However, when the construction began it turned out that there is a problem with the project.

Formula 42 racing track is an area delimited by two borders. Each border is a closed polyline that is a boundary of a convex polygon. The inner border is completely inside the outer border and has no common points with it. The racing car in Formula 42 is a circle. There are no limits for its radius, so the larger is the circle, the more spectacular the race is.

The problem with the track project is that Herman has only designed the outer border and the inner border separately. Their relative position is not specified. Now the company that is constructing the track can choose any relative position of the borders. They want to place them in such way that the race was most spectacular. However, they can only make parallel translations of the borders, but cannot rotate, flip or modify them in any other way.

It is required to find such relative position of the borders, that the resulting track can be used for the race with the maximal possible size of the racing car. The track can be used by a racing car of radius r, if the car of such radius can drive a complete lap around the inner border while staying at track. It is allowed to touch the borders. Formally, for a point P inside the inner border there must be such continuous closed polyline, such that P is inside this polyline, and for any point on the polyline a circle of radius r with its center in this point is inside the outer border and outside of the inner border, touching is allowed.

Help the construction company to deal with Herman's plan, and find out for what maximal radius of the racing car they can construct a racing track with the given design of its inner and outer borders.

Input

Input contains descriptions of two convex polygons, their boundaries give the shape of the outer and the inner borders, respectively.

The first line contains n — the number of vertices of the first polygon ($3 \le n \le 100$). The following n lines contain two integers x_i and y_i each — coordinates of the vertices of the first polygon ($0 \le x_i, y_i \le 1\,000$).

The next line contains an integer m — the number of vertices of the second polygon ($3 \le m \le 100$). The following m lines contain two integers x'_i and y'_i each — coordinates of the vertices of the second polygon ($0 \le x'_i, y'_i \le 1\,000$).

Vertices of each polygon are given in counterclockwise order. Polygons are convex, no three points of one polygon are on the same line. It is guaranteed that the polygons can be translated in such way that the second one is completely inside the first one, and their boundaries have no common points.

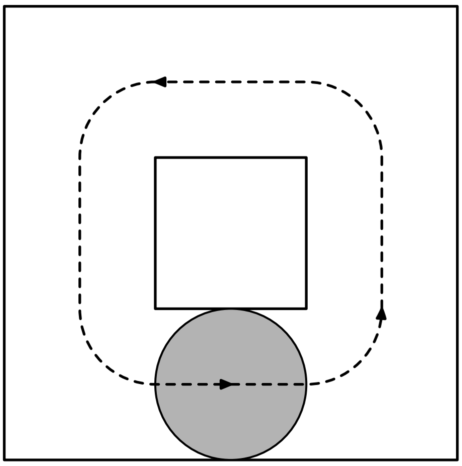
Output

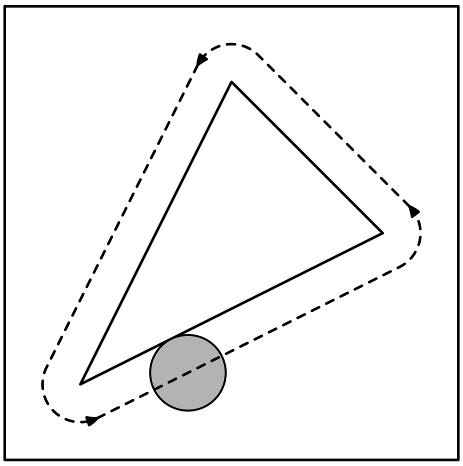
Output one real number — the maximal possible radius of the racing car. The answer must be correct with absolute or relative error not exceeding 10^{-6}.

Examples

standard input	standard output
4 0 0 3 0 3 3 0 3 4 0 0 1 0 1 1 0 1	0.5
4 0 0 3 0 3 3 0 3 3 0 0 2 1 1 2	0.25

Note





Problem L. How Many Tests

Time limit: 1 second
Memory limit: 512 megabytes

When judges prepare tests for the programming contest problem, they number tests from 1 to n.

It is convenient if the files with tests are shown in their correct order, from 1 to n. But file managers sort files by their names as strings, so if the name of the test file is equal to its number, the file order is not correct, for example «10» goes before «2».

To avoid such problem, file names are prepended with leading zeroes. Judges use minimal possible number of leading zeroes to make names of all files have the same length. For example, if the problem has 10 tests, the names of the files with tests are «01», «02», «03», «04», «05», «06», «07», «08», «09» and «10».

Andrew is an experienced judge, so he always uses the described way to name the files with tests. Recently he has found some files with tests of some ancient problem at his old hard drive. Unfortunately, the drive is damaged, so some tests are missing. Help Andrew to find out how many tests could be there in the problem. He wants to know the minimal and the maximal possible number of tests.

Input

The first line of input contains an integer k — the number of files ($1 \le k \le 1000$). The following k lines contain file names. All these lines are non-empty, have equal length that doesn't exceed 9. File names are distinct, they only contain digits. No file name contains only zeroes.

Output

Output two integers: the minimal and the maximal number of tests that the problem could have.

Example

standard input	standard output
3 05 10 08	10 99