Problem A. Jara's Legacy

Time Limit	4548 ms
Mem Limit	1572864 kB
Code Length Limit	50000 B
OS	Linux

Victor Jara was a Chilean teacher, theater director and political activist. He is widely recognized because of his talent as poet and song writer. His most recognized work is probably the song "A Desalambrar" that can be translated from Spanish as "Unwire". In this song Jara assures that people are the rightful owner of the lands, and so wire fences that delimit private properties should be cut down to allow access to everybody. Although Jara's proposal is far from being fulfilled, some of his convinced listeners keep trying to make it happen. Since they must face several enemies, they try to make their job efficient by only cutting down the necessary number of fences and not more.

Each fence can be modeled as a segment (straight line) connecting two points in the XY plane. These endpoints are considered to be part of the fence. A cut in a fence removes any contiguous part of the fence except the endpoints. An area is said to be free if and only if, for any pair of points not lying over a fence, there is a (not necessarily straight) line that connect these points without crossing any fence. Given the location of the fences, your job is to calculate the minimum number of fences that need to be cut down to make the area free, according to the above definition.

Input

Each test case is described using several lines. The first line contains an integer N indicating the number of fences in the area ($1 \le N \le 10^5$).

Each of the next N lines describes a different fence using four integers $X0$, $Y0$, $X1$ and $Y1$ ($-10^4 \le X0, Y0, X1, Y1 \le 10^4$). These values represent that there is a fence whose endpoints in the XY plane are $(X0, Y0)$ and $(X1, Y1)$. You may assume that for each fence its two endpoints are distinct. Besides, within each test case, the intersection of any pair of fences is either empty or it is an endpoint of both fences. The end of input is indicated with a line containing a single -1.

Output

For each test case, output a single line containing a single integer representing the minimum number of fences that need to be cut down to make the area free.

Example

Input	Output
9 -50 0 0 0 0 0 50 0 -50 0 0 50 0 50 50 0 -50 0 0 -50 0 -50 50 0 0 0 0 -50 0 -50 50 -50 50 -50 50 0 2 0 1 2 3 0 0 2 2 -1	4 0

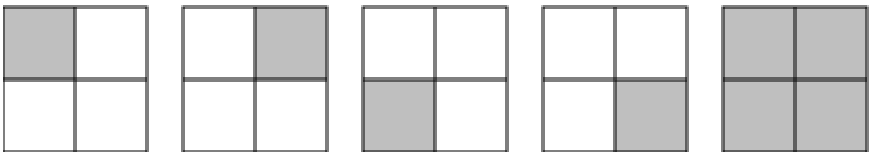
Problem B. Feynman

Time Limit	1000 ms
Mem Limit	1572864 kB
Code Length Limit	50000 B
OS	Linux

Richard Phillips Feynman was a well known American physicist and a recipient of the Nobel Prize in Physics. He worked in theoretical physics and also pioneered the field of quantum computing. He visited South America for ten months, giving lectures and enjoying life in the tropics. He is also known for his books "Surely You're Joking, Mr. Feynman!" and "What Do You Care What Other People Think?", which include some of his adventures below the equator.

His life-long addiction was solving and making puzzles, locks, and cyphers. Recently, an old farmer in South America, who was a host to the young physicist in 1949, found some papers and notes that is believed to have belonged to Feynman. Among notes about mesons and electromagnetism, there was a napkin where he wrote a simple puzzle: "how many different squares are there in a grid of $N \times N$ squares?".

In the same napkin there was a drawing which is reproduced below, showing that, for N=2, the answer is 5.



Input

The input contains several test cases. Each test case is composed of a single line, containing only one integer N, representing the number of squares in each side of the grid ($1 \le N \le 100$).

The end of input is indicated by a line containing only one zero.

Output

For each test case in the input, your program must print a single line, containing the number of different squares for the corresponding input.

Example

Input	Output
2 1 8 0	5 1 204

Problem C. Adding Reversed Numbers

Time Limit	5000 ms
Mem Limit	1572864 kB
Code Length Limit	50000 B
OS	Linux

The Antique Comedians of Malidinesia prefer comedies to tragedies. Unfortunately, most of the ancient plays are tragedies. Therefore the dramatic advisor of ACM has decided to transfigure some tragedies into comedies. Obviously, this work is very hard because the basic sense of the play must be kept intact, although all the things change to their opposites. For example the numbers: if any number appears in the tragedy, it must be converted to its reversed form before being accepted into the comedy play.

Reversed number is a number written in Arabic numerals but the order of digits is reversed. The first digit becomes last and vice versa. For example, if the main hero had 1245 strawberries in the tragedy, he has 5421 of them now. Note that all the leading zeros are omitted. That means if the number ends with a zero, the zero is lost by reversing (e.g. 1200 gives 21). Also note that the reversed number never has any trailing zeros.

ACM needs to calculate with reversed numbers. Your task is to add two reversed numbers and output their reversed sum. Of course, the result is not unique because any particular number is a reversed form of several numbers (e.g. 21 could be 12, 120 or 1200 before reversing). Thus we must assume that no zeros were lost by reversing (e.g. assume that the original number was 12).

Input

The input consists of N cases (equal to about 10000). The first line of the input contains only positive integer N. Then follow the cases. Each case consists of exactly one line with two positive integers separated by space. These are the reversed numbers you are to add.

Output

For each case, print exactly one line containing only one integer - the reversed sum of two reversed numbers. Omit any leading zeros in the output.

Example

```
Sample input:
3
24 1
```

```
4358 754
305 794

Sample output:
34
1998
1
```

Problem D. The Explosion

Time Limit	100 ms
Mem Limit	1572864 kB
Code Length Limit	50000 B
OS	Linux

The day of 6.XII.2003 in Megabyteland began calm and quietly as any other day. Some people went to work, some - to school, some - to store to buy food. Drivers were traditionally stuck in traffic jams, drinking coffee and reading morning newspaper. Suddenly the regularity of this day was disturbed by huge explosion. "They blew up the embassy of Bajtocja!!!" - somebody cried. Everybody began to run away in panic.

Police works pretty good in Megabyteland and first radiocars appeared near the embassy only few seconds after the explosion. All the people near the embassy were detained. Some of these people are the organizers of the explosion, but the others could by just occasional witnesses. During the testification each person named exactly one perpetrator. It is known, that if a man is not a perpetrator, than he always says the truth (he haven't a reason to lie, have he?). However, perpetrators want to make the work of the police more difficult, so a perpetrator can name any person during the testification (even himself).

The policemen are in the very hard situation. They should arrest some group of potential perpetrators, but it is difficult to determine who is guilty and who is not from the data they have. There exists many groups of potential perpetrators, that don't contradict to any of the testimonies. The policemen want to arrest as small innocent people as possible. So they would like to choose the group with minimal number of people.

Write a program that, given the number of detained people and their testimonies, will determine the number of people in the smallest group of potential perpetrators, that don't contradict to the testimonies.

Input

The first line of the input contains a single integer T, the number of testcases ($1 \le T \le 10$).

First line of each testcase contains integer number N ($2 \le N \le 100000$), equal to the number of detained people (the people are numbered from 1 to N). The i-th of the following N lines contain one integer number Pi ($1 \le Pi \le N$). Here Pi is the man whom i-th man testified to be guilty.

Output

The output should consist of T lines, containing one integer number for each testcase - the number of people in the smallest group of potential perpetrators, that don't contradict to the testimonies.

Example

Input	Output
1 3 2 3 1	2

Problem E. Closest Triplet

Time Limit	37470 ms
Mem Limit	1572864 kB
Code Length Limit	50000 B
OS	Linux

Closest pair is an old problem that asks to find, given a set of N points in the plane, the pair that minimizes the distance between them. This problem can easily be solved using roughly N^2 operations by testing all possible pairs of points and keeping at each step the optimal pair. With a more clever approach, the problem has been solved using $\sim N \log N$ operations. Closest triplet is an analogous problem which also takes a set of N points as input, and asks for the triplet (group of three points) that minimizes the sum of the three distances between each pair of them. In this case there is also a trivial solution that tests all possible triplets using roughly N^3 operations. However, since you are a clever programmer, we are confident that you are able to find a better algorithm.

Input

The input contains several test cases, each one described in several lines. The first line contains an integer N indicating the number of points in the set ($3 \le N \le 3000$). Each of the next N lines describes a different point of the set using two integers X and Y separated by a single space ($1 \le X, Y \le 10^6$); these values represent the coordinates of the point in the XY plane. You may assume that within each test case no two points have the same location. The last line of the input contains a single -1 and should not be processed as a test case.

Output

For each test case output a single line with a real number representing the sum of the distances between each pair of points of any closest triplet of the set of points. Round the result to the closest rational number with three decimal places. In case of ties, round up. Always use exactly three digits after the decimal point, even if it means finishing with a zero.

Example

Input	Output
4 1 1 4 1 1 5 1000 1000 9 100000 200000 200000 200000 150000 286603 60000 140000 240000 140000 150000 340000 1 340000 300000 340000 150000 87087 -1	12.000 300000.796

Problem F. To and Fro

Time Limit	1000 ms
Mem Limit	1572864 kB
Code Length Limit	50000 B
OS	Linux

Mo and Larry have devised a way of encrypting messages. They first decide secretly on the number of columns and write the message (letters only) down the columns, padding with extra random letters so as to make a rectangular array of letters. For example, if the message is “There’s no place like home on a snowy night” and there are five columns, Mo would write down

```
t o i o y
h p k n n
e l e a i
r a h s g
e c o n h
s e m o t
n l e w x
```

Note that Mo includes only letters and writes them all in lower case. In this example, Mo used the character ‘x’ to pad the message out to make a rectangle, although he could have used any letter. Mo then sends the message to Larry by writing the letters in each row, alternating left-to-right and right-to-left. So, the above would be encrypted as

```
toioynnkpheleaigshareconhtomesnlewx
```

Your job is to recover for Larry the original message (along with any extra padding letters) from the encrypted one.

Input

There will be multiple input sets. Input for each set will consist of two lines. The first line will contain an integer in the range 2...20 indicating the number of columns used. The next line is a string of up to 200 lower case letters. The last input set is followed by a line containing a single 0, indicating end of input.

Output

Each input set should generate one line of output, giving the original plaintext message, with no spaces.

Example

Input	Output
5 toioynnkpheleaigshareconhtomesnlewx 3 ttyohhieneesiaabss 0	theresnoplacelikehomeonasnowynightx thisistheeasyoneab

Problem G. Escaping from escaping

Time Limit	7000 ms
Mem Limit	1572864 kB
Code Length Limit	50000 B
OS	Linux

A communications protocol is a set of rules designed to transfer information in a communications system. Elisa's job is to write programs that implement parts of said protocols. It is often necessary for this to transfer sequences of fields, and in order to know where a field ends and the next one begins it is customary to insert a separator between each pair of consecutive fields. Using a simple separator such as a space, a comma or a semicolon is inconvenient because sometimes the fields to be transferred contain these same characters. The standard solution in these cases is to insert an "*escaping*" character just before every appearance of a separator inside a field, so that it can be thus distinguished from a real separator. Elisa thinks this solution will increase a lot the length of the data to be transmitted, so she has decided to use a separator that is complex enough for it to never appear inside the data. In this way she hopes to escape the inefficient alternative of escaping separators.

To choose the ideal separator, Elisa has compiled a *log*, which is nothing else than a long string of characters that is representative of the data that her protocol needs to manage. After thinking about the problem for a while, Elisa reached the conclusion that any non-empty string of characters that does not appear inside the log would be an acceptable separator for use within her protocol. But because she is interested in minimizing the length of the data to be transmitted, she would like to know the minimal length that an acceptable separator can have. She immediately wrote a program to calculate this length, and is now testing it for the particular case in which both the log and the acceptable separators only contain binary digits (`'0'` and `'1'`). Can you anticipate the results?

Input

The first line contains an integer number $\mathbf{T}$, the number of test cases ($\mathbf{1 \le T \le 200}$). Each of the following $\mathbf{T}$ lines contains a log, which is a non-empty string of at most $\mathbf{10^5}$ binary digits.

Output

For each test case, print a single line containing an integer number representing the minimal length of an acceptable separator for the given log.

Example

Input	Output
3 011101001 100010110011101 11111	3 4 1

Problem H. Equilibrium

Time Limit	1046 ms
Mem Limit	1572864 kB
Code Length Limit	50000 B
OS	Linux

The mean and the median usually confuse the students because of their similar spelling, but they are quite different concepts. In this problem we are going to work with the mean and the median of a set consisting of **N** pairwise distinct integers, where **N** is odd. The mean of such set is defined, as usual, as the sum of the numbers divided by **N**, while the median is the unique element in the set that is greater than (**N**-1)/2 of its elements, and less than the other (**N**-1)/2 elements in the set. For instance, if the set is {0, 2, 6, 4, 13}, then the mean is 5 while the median is 4.

We aim to make student's lives easier by generating "balanced" sets, that is, sets consisting of an odd number of pairwise distinct integers where the mean and the median coincide. For example, the set {0, 2, 6, 4, -2} is balanced, since it has **N**=5 different integers, and the mean and the median are both equal to 2.

The following procedure has been suggested in order to obtain balanced sets. A set with an even number of distinct integers is chosen, and an extra integer distinct from every element in the set is added to it, in such a way that the resulting set is balanced. We want you to check if the given procedure works. Therefore your task is, given **N**-1 distinct integers, with odd **N**, count the number of balanced sets that can be formed by following the described procedure.

Input

The input contains several test cases. Each test case is described with two lines. The first line contains a single odd positive integer **N** that indicates the number of elements the balanced set must have (3 <= **N** <= 499). The second line contains **N**-1 distinct integers **Z_i** that represent the given elements of the set (-10^{14} <= **Z_i** <= 10^{14} for 1 <= **i** <= **N**-1). The last line of the input contains the number -1, and should not be processed as a test case.

Output

For each test case, output a single line with an integer representing the total number of different balanced sets that can be obtained by adding an integer to the given set, as explained in the problem description.

Example

Input:

```
5
0 2 6 4
7
1 2 3 4 5 8
3
-100000000000000 100000000000000
-1
```

Output:

```
3
1
3
```

Problem I. Constellation of the parallelogram

Time Limit	1000 ms
Mem Limit	1572864 kB
Code Length Limit	50000 B
OS	Linux

One morning, when Cyrus Samsa woke from troubled dreams, he found himself lying on his bed, but not transformed into a monstrous insect. However, when he looked up at the still starry sky, he found that the visual setting of the cosmos showed four stars positioned as the vertices of a perfect parallelogram. Being a big geometry enthusiast, he called those four stars "constellation of the parallelogram".

A parallelogram is a four-sided polygon with its pairs of opposite sides being of identical length. Equivalently, a parallelogram is a convex quadrilateral whose diagonals intersect at their midpoints. The following figure illustrates both definitions, for a parallelogram with sides of lengths **L** and **l**, and diagonals of lengths **D** and **d**.

Figure 1

Unfortunately, astronomers still do not agree on which were the stars that Cyrus was looking at. The problem is that on each image obtained of the sky's configuration, there are usually many sets comprising four stars that are the vertices of a parallelogram.

A starry sky is represented on an image as a set of points in the Cartesian plane, each of them corresponding to a star. The following figure shows three copies of the same image, each illustrating one of the three sets of four stars which are the vertices of a parallelogram, which are marked in the figure with solid lines.

Figure 2

Your task is to safeguard the good name of astronomers by calculating, given an image, the number of parallelograms contained in it.

Input

The first line contains an integer **N** indicating the number of stars on the image to be analyzed (**4 ≤ N ≤ 1000**).

The following **N** lines describe each of the stars on the image with two integers X_i and Y_i, indicating respectively the **X** and **Y** coordinates of the star on the Cartesian plane ($-10^8 \le X_i, Y_i \le 10^8$ for $i = 1, 2, \ldots, N$). You may assume that no two stars on the image are on the same position.

Output

Print a single line containing a single integer representing the number of sets of four points on the input image that are the vertices of a parallelogram.

Example 1

Input	Output
7 0 1 1 3 2 4 3 1 4 2 4 3 5 0	3

Example 2

Input	Output
11 0 0 0 1 0 -1 1 0 1 1 1 -1 -1 0 -1 1 -1 -1 0 2 1 2	44

Example 3

Input	Output
4 0 0 0 1 0 2 0 3	0

Example 4

Input	Output
6 -100000000 100000000 100000000 -100000000 -100000000 -100000000 100000000 100000000 1 1 -1 -1	2

Problem J. E - Publish of Perish

Time Limit	1000 ms
Mem Limit	1572864 kB
Code Length Limit	50000 B
OS	Linux

“Publish or perish” is the academic life’s fundamental motto. It refers to the fact that publishing your work frequently is the only way to guarantee access to research funds, bright students and career advances. But publishing is not enough. It is necessary that your work is *referenced* (or *cited*). That is, your papers must be mentioned as source of information in other people’s publications, to attest the quality and relevance of your research. The more citations a paper receives from other authors, the more it is considered influential.

In 2005 Jorge E. Hirsch, a physicist at the University of California at San Diego, proposed a way to evaluate the scientific impact of a researcher, based on the citations his or her papers have received. The *h-index*, as Hirsch’s proposal became known, is a number based on the set of a researcher’s most cited papers. It is defined in Hirsch’s own words as: A scientist has index h if h of his N_p papers have at least h citations each, and the other (N_p - h) papers have at most h citations each.

Albert Einstein, for example, published 319 papers in scientific journals and has an h-index equal to 46. It means 46 of his papers have received 46 or more citations each, and all of his remaining 273 papers have 46 citations or less each. Given the information of how many citations each paper from a given researcher has received, write a program to calculate that researcher’s h-index.

Input

The input contains several test cases. The first line of a test case contains one integer N indicating the number of papers a researcher has published ($1 \le N \le 10^3$). The second line contains a list of N integers M_i, separated by one space, representing the number of citations each of the N papers from that author has received ($0 \le M_i \le 10^3$, for $1 \le i \le N$). The end of input is indicated by a line containing only one zero.

Output

For each test case in the input, your program must print a single line, containing one single integer, the h-index for the given list of citations.

Example

Input	Output
4 1003 1 200 2 10 1 1 1 0 1 1 0 1 1 1 7 6 5 4 3 2 1 0 0	2 1 3