Problem A. Accurate Movement

|	
Time limit:	2 seconds
Memory limit:	512 megabytes

Amelia is studying modeling. She is interested in models with moving parts.

As her first task, she made a rectangular box of size $2 \times n$, which contains two parallel rails and a rectangular bar on each of them. The short bar has size $1 \times a$, and the long one has size $1 \times b$. The long bar has a stopper at each end, and the short one is always between these two stoppers.

Example position   Initial position   Target position
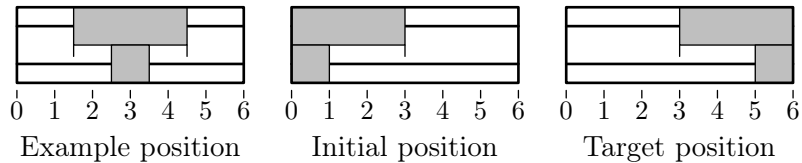
The bars can be moved along the rails, one bar at a time, as long as the short bar is between the stoppers. So, on each *move* Amelia selects one of the bars and moves it, while the other one stays in place.

Initially, both bars are aligned to one side of the box, and Amelia wants them to be aligned to the other side in as few moves as possible. What is the minimal number of moves she should do to achieve her goal?

Input

The only input line contains three integers a, b, and n ($1 \le a < b \le n \le 10^7$).
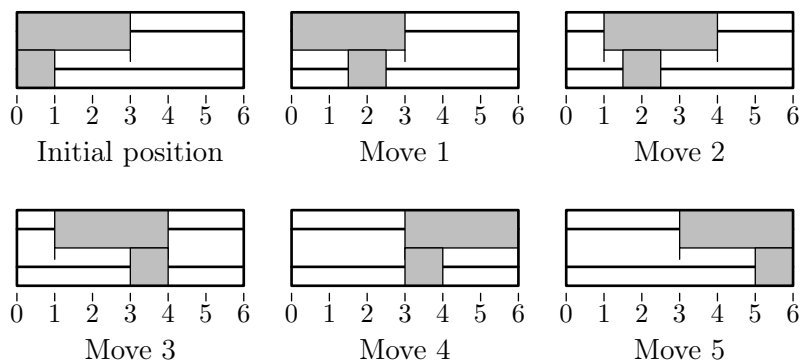
Output

The only output line should contain a single integer — the minimal number of moves Amelia needs to do.

Examples

standard input	standard output
1 3 6	5
2 4 9	7

Note

A possible solution for the first example is shown below.

Initial position   Move 1   Move 2

Move 3   Move 4   Move 5

Problem B. Bad Treap

|	
Time limit:	2 seconds
Memory limit:	512 megabytes

A *treap*, also known as *Cartesian tree*, is a data structure that stores a set of keys in a binary search tree.

Each node of this tree is characterized by a pair (x, y).

The x values of the nodes are the stored keys. They obey "the binary search tree rule": all x values in the left subtree of the node are smaller than its x value, and all x values in the right subtree of the node are larger than its x value.
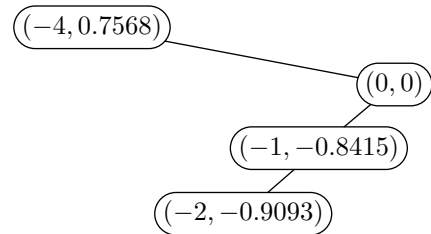
The y values of the nodes obey "the heap rule": y value of a node is less than or equal to the y value of its parent.

The y value for each created node is usually selected randomly according to some distribution. This results in nice average time complexity of many operations.

Since this data structure unites some properties of a binary search tree and a heap, its name is a "portmanteau" term made of two words: TRee + hEAP = TREAP.
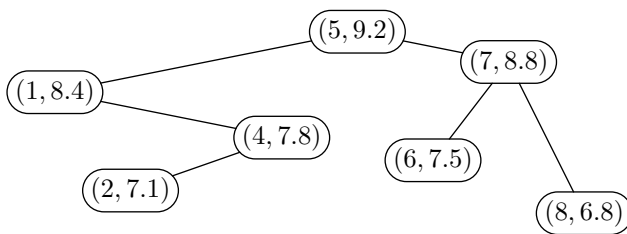
A treap with 7 nodes; its height is 4   A treap for the sample test below

Benjamin decided that nondeterminism in y value selection procedure is bad, as it makes execution time differ from one run to another. He decided to calculate y for each node deterministically based on its x. He selected the rule $y = \sin(x)$. He is especially glad that distinct integer x's will always have distinct y's.

Barbara understands that while the nondeterministic treap shows its worst performance when provided "bad" random sequence, the deterministic treap shows its worst performance when provided "bad" set of keys. She wants to explain it to Benjamin by showing him n integer keys, which, being stored in his data structure, will form a treap of height n — the "most unbalanced" possible situation.

Help Barbara by providing n such keys. All these keys should fit into the 32-bit signed integer type.

Input

The input consists of a single integer n ($1 \leq n \leq 50\,000$).

Output

Output n lines containing distinct integer keys. All keys must be between -2^{31} and $2^{31} - 1$, inclusive.

A treap built on these keys with the rule $y = \sin(x)$ must have height n.

Example

standard input	standard output
4	-2
	0
	-1
	-4

Problem C. Cross-Stitch

Time limit: 2 seconds

Memory limit: 512 megabytes

Cross-stitch is a form of needlework known since prehistoric times. A cross-stitch pattern consists of several crosses on the face of the fabric, that are connected on the backside. Traditionally, the whole pattern should be embroidered by one thread.

Carol is going to mass-produce cross-stitch patterns. Each pattern will be accompanied by the rectangular patch of fabric and the thread that is required to embroider this pattern. Carol wants to minimize the length of the thread that is required for the pattern.
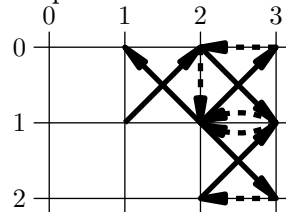
You are given the face of the pattern. You should design the backside, so that the total length of the thread is minimal. The crosses on the pattern's face are 8-connected, i.e. it is possible to reach each cross from any other by a sequence of chess king's moves.

Input

The first line of the input contains two integers w and h — the width and the height of the patch of fabric ($1 \le w, h \le 100$).

The following h lines contain the face of the pattern. Each of them contains w characters, where 'X' represents a cross, and '.' — an empty space. The face of the pattern contains at least one cross and crosses are 8-connected.

Output

The first line of output should contain the single integer n — the number of stitches required to embroider the pattern.

The following $n + 1$ lines should contain the coordinates of the points where a needle comes from the backside to the face or back: x_i, y_i for $i = 0..n$. The upper-left corner of the patch has coordinates $(0, 0)$, and the bottom-right one — (w, h). The first stitch goes to the face of the pattern, the second one — to the backside, the third one — to the face, and so on. It is possible to duplicate stitches at the backside, but not at the face. None of the stitches should have zero length.

Example

standard input	standard output
3 2	11
.XX	1 1
..X	2 0
	2 1
	3 0
	2 0
	3 1
	2 1
	3 2
	2 2
	3 1
	2 1
	1 0

Problem D. Double Palindrome

Time limit:	2 seconds
Memory limit:	512 megabytes

A *palindrome* is a string that reads the same backward as forward. For example, `rotator`, `lil` and `abba` are palindromes, but `shalash` is not.

A *double palindrome* is a string that is either a palindrome or a concatenation of two (not necessarily distinct) palindromes. For example, `susanna`, `potato` and `abba` are double palindromes, but `zzyzx` and `abaabb` are not.

Dalila has just realized that her name is a double palindrome! Now she wonders how many non-empty strings of length at most n composed of the first k English letters have the same property. Help her find this number and output it modulo $998\,244\,353$.

Input

The only line contains two integers n and k — the maximum length of the string and the size of the alphabet ($1 \le n \le 10^5$; $1 \le k \le 26$).

Output

Output the number of non-empty double palindromes of length at most n composed of the first k English letters, modulo $998\,244\,353$.

Examples

standard input	standard output
3 3	33
6 2	114
42 7	83419789

Note

In the first example the strings to be counted are: a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, aac, aba, abb, aca, acc, baa, bab, bba, bbb, bbc, bcb, bcc, caa, cac, cbb, cbc, cca, ccb, ccc.

Problem E. Equidistant

Time limit:	2 seconds
Memory limit:	512 megabytes

In 2019 ICPC subregions structure was changed a little. Now for each subregion, we need to choose the best place for the subregion finals. To make things fair we want to choose a city in such a way that all teams will spend the same amount of time to get there.

To make things simpler we say that all teams will use trains to get to the finals. The railroad system can be represented as a tree where each city is a vertex and some pairs of cities are connected by a railroad. It takes exactly one hour to get from one city to another if they are directly connected.

You are given a description of the railroad system and the cities where teams are located. You need to choose any city that has an equal distance to all teams' cities or detect that no such city exists.

Input

The first line of the input contains two integers n and m — the number of cities and the number of teams ($1 \leq m \leq n \leq 2 \cdot 10^5$). Each of the following $n - 1$ lines contains two integers v_i and u_i — the indices of the cities connected by the i-th railroad ($1 \leq v_i, u_i \leq n$). It is guaranteed that for each pair of cities there is exactly one simple path connecting them.

The next line contains m integers $c_1, c_2, \ldots, c_m$ — the cities of the teams ($1 \leq c_i \leq n$). All teams are located in different cities.

Output

If it is impossible to choose a city fairly, output a single word "NO". Otherwise, output a the word "YES" at the first line. The second line should contain a single integer — the city where subregion finals should be held. If there is more than one solution, output any of them.

Examples

standard input	standard output
6 3 1 2 2 3 3 4 4 5 4 6 1 5 6	YES 3
2 2 1 2 1 2	NO

Problem F. Foreach

Time limit: 2 seconds
Memory limit: 512 megabytes

Do you like really good programming languages? Let's, for example, discuss PHP.

PHP, like many other languages, supports arrays. For purposes of this problem, we only consider an array with consecutive integer indexes starting from 0. It would be denoted as $[a_0, a_1, \ldots, a_{n-1}]$.

PHP, like many other languages, supports variables. All of them must start with $ symbol. We will have two in this problem: array $a, and its element $x.

Also, as many other languages, PHP supports loops over all elements of array. We consider only two of four possible forms of `foreach` loop:

non-reference form: `foreach ($a as  $x) // code`
reference form: `foreach ($a as &$x) // code`

Formally, non-reference form goes through elements of the array one by one, and sets the value of $x to the value of the corresponding element and then executes specified code. Reference form goes though elements of the array one by one and makes $x a reference to the corresponding element and then executes code. So, in the first case, if one changes the value of $x, the value of the array element wouldn't be changed, while in the second one it would.

PHP supports a lot of other control-flow constructions. Two of them we are interested in are `if` and `break`. As in other languages, `break` stops the loop immediately, and `if` allows to execute code only if some condition holds.

Unfortunately, unlike many other languages, PHP doesn't have variable scopes. So, even after a reference form `foreach` loop, variable $x still would be a reference to an element of the array it was set to last time. So, changing its value after the loop would still change the element of the array. On the other side, making $x a reference to another element doesn't change the value of the previous element. Let's consider an example.

One has an array $[1, 2, 3]$ and executes the following code

```
foreach ($a as &$x) if ($x == 2) break;
foreach ($a as  $x) if ($x == 2) break;
```

After the first loop $x is a reference to the middle element of the array. On the first step of the second loop, the array becomes $[1, 1, 3]$, because the value of $x is changed to 1, and it is the reference to the middle element. On the second step, the value of $x would be changed to a new value of the next element of the array, which turned out to be 1 again. It's still not equal to 2, so on the next step, the value of $x changes to 3, and the resulting state of the array is $[1, 3, 3]$.

As you see, code can change arrays even having no explicit sets of variables.

Can you create a program that changes one array to another using only lines of the form shown in the example above?

Input

The first line of the input contains one integer n — the length of the array $a ($1 \le n \le 50$). The second line contains n integers s_i — the initial state of the array ($1 \le s_i \le 100$). The third line contains n integers t_i — the target state of the array ($1 \le t_i \le 100$).

Output

If it is not possible to change the state of array $a from s to t, output a single line containing a single integer -1.

Otherwise, the first line of the output must contain an integer k — the number of lines in the program ($0 \le k \le 10\,000$). You don't have to minimize this value.

Each of the following k lines should be either in the form

```
foreach ($a as &$x) if ($x == <some integer value>) break;
```

or

```
foreach ($a as  $x) if ($x == <some integer value>) break;
```

All integers should be positive and should not exceed 100. No other variables or language constructions are allowed. You should follow the format as close as possible, including whitespaces (e. g. there should be two spaces between `as` and `$x` in the non-reference form). Note that since there is no "`Presentation Error`" outcome in the contest rules, if you fail to follow these requirements, you will get "`Wrong answer`" outcome.

Your code will be executed by formal rules described in the statement. It must convert array `$a` from the initial state to the target state.

Examples

standard input	standard output

3
1 2 3
1 3 3
	2
	foreach ($a as &$x) if ($x == 2) break;
	foreach ($a as $x) if ($x == 2) break;

2 1 2 1 3	-1

Problem G. Golf Time

Time limit: 2 seconds
Memory limit: 512 megabytes

Galina is an aspiring golfer and spends all her free time at a minigolf park. Her favorite golf course in this park is a rectangle, with one of its sides parallel to the north-south direction, and the other one parallel to the east-west direction. The ball is really small and can be considered a point.

Galina wants to practice a specific shot called "put". In this shot, a ball is placed somewhere inside the golf course and hit north-east, obtaining the speed of $\sqrt{2}$ inches per second. This means that the ball starts moving by exactly 1 inch to the north and 1 inch to the east per second.

The golf park uses an innovative technology that eliminates all friction, and the ball will continue to move at the initial speed regardless of the traveled distance. Collisions with golf course borders will be perfectly elastic. Fancy!

Inside the course there is a pond that is represented as a simple polygon with edges parallel to the sides of the golf course. If the ball touches the pond, it sinks momentarily.

You are given a list of starting coordinates for Galina's next put training session. For each starting position, assuming a put is shot from this place, calculate how long the ball will travel before falling into the pond and find exactly where it happens, or determine that the ball will move indefinitely.

Input

The first line contains two integers w and h, the width and the height of the golf course in inches ($4 \leq w, h \leq 5 \cdot 10^8$). A coordinate system is introduced in such a way that the corners of the golf course have coordinates $(0,0)$, $(w,0)$, (w,h), $(0,h)$, and the point $(1,1)$ is to the north-east of the point $(0,0)$.

The second line contains a single integer n, denoting the number of vertices in the polygon ($4 \leq n \leq 1000$).

Each of the following n lines contains two integers x_i and y_i, denoting the coordinates of the i-th vertex ($1 \leq x_i \leq w - 1$; $1 \leq y_i \leq h - 1$). The vertices are listed in traversal order.

All polygon vertices are distinct and none of them lie at the polygon's edge. All polygon edges are either vertical ($x_i = x_{i+1}$; $x_n = x_1$) or horizontal ($y_i = y_{i+1}$; $y_n = y_1$), and none of them intersects another.

The next line contains a single integer t, denoting the number of starting points for a shot ($1 \leq t \leq 100$).

Each of the following t lines contains two integers $\tilde{x}_i$ and $\tilde{y}_i$, denoting the coordinates of the i-th starting point ($1 \leq \tilde{x}_i \leq w - 1$; $1 \leq \tilde{y}_i \leq h - 1$). No starting point lies inside or on the border of the pond.

Note that all starting points are independent of each other. In particular, you can assume that there is exactly one ball inside the golf course at any moment.

Output

For each starting point in order of input, output either a single integer -1 if the ball will never sink, or three integers τ x_s y_s, denoting the time before the ball sinks in seconds, and coordinates of the point where it happens ($\tau > 0$; $1 \leq x_s \leq w - 1$; $1 \leq y_s \leq h - 1$). Point (x_s, y_s) must belong to the border of the pond.
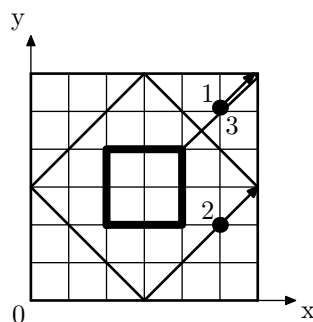
Examples

standard input	standard output
10 8 6 1 4 1 5 3 5 3 6 4 6 4 4 4 2 2 1 7 4 7 7 5	2 4 4 3 4 6 13 3 4 15 2 4
6 6 4 2 2 4 2 4 4 2 4 3 5 5 5 2 5 5	3 4 4 -1 3 4 4

Note

In the first example, the ball trajectories look as shown:
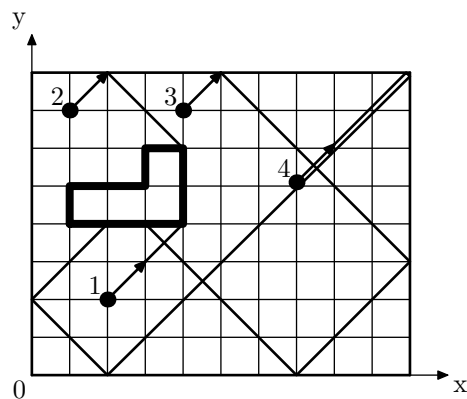
In the second example, starting positions 1 and 3 coincide, and for the position 2 the ball will never stop:

Problem H. High Load Database

Time limit: 2 seconds
Memory limit: 512 megabytes

Henry profiles a high load database migration script. The script is the list of n transactions. The i-th transaction consists of a_i queries. Henry wants to split the script to the minimum possible number of batches, where each batch contains either one transaction or a sequence of consecutive transactions, and the total number of queries in each batch does not exceed t.

Unfortunately, Henry does not know the exact value of t for the production database, so he is going to estimate the minimum number of batches for q possible values of t: $t_1, t_2, \ldots, t_q$. Help Henry to calculate the number of transactions for each of them.

Input

The first line contains a single integer n — the number of transactions in the migration script $(1 \le n \le 200\,000)$.

The second line consists of n integers $a_1, a_2, \ldots, a_n$ — the number of queries in each transaction $(1 \le a_i;$ $\sum a_i \le 10^6)$.

The third line contains an integer q — the number of queries $(1 \le q \le 100\,000)$.

The fourth line contains q integers $t_1, t_2, \ldots, t_q$ $(1 \le t_i \le \sum a_i)$.

Output

Output q lines. The i-th line should contain the minimum possible number of batches, having at most t_i queries each. If it is not possible to split the script into the batches for some t_i, output "Impossible" instead.

Remember that you may not rearrange transactions, only group consecutive transactions in a batch.

Example

standard input	standard output
6	2
4 2 3 1 3 4	Impossible
8	4
10 2 5 4 6 7 8 8	5
	4
	3
	3
	3

Problem I. Ideal Pyramid

Time limit:	2 seconds
Memory limit:	512 megabytes

Every pharaoh cares about his legacy. Reigning pharaoh Inaros the Great wants to be remembered for a long time. He is going to build the largest pyramid the humankind has ever seen.

Of course, the proper pyramid should have four sides at the bottom, oriented to the cardinal directions — two sides of the pyramid should go exactly from north to south, and two sides should go from east to west. The perfectly balanced pyramid should have the slope angle of the side equal to $45°$, no more, no less. To simplify construction, the pyramid should have integer height and integer coordinates of the center.

There are n obelisks at the construction site. The i-th obelisk is a pillar with coordinates (x_i, y_i) and height h_i. Inaros wants to build a pyramid so that each existing obelisk would be inside the pyramid. The obelisk is inside the pyramid if the height of the pyramid at the obelisk position is greater than or equal to the height of the obelisk.

Since the pharaoh wants to finish the construction of the pyramid during his life, he wants to find the smallest possible pyramid that contains all of the obelisks.

Input

The first line contains a single integer n — the number of obelisks ($1 \le n \le 1000$).

Each of the following n lines contains three integers x_i, y_i, h_i — the coordinates of the i-th obelisk and its height ($-10^8 \le x_i, y_i \le 10^8$; $1 \le h_i \le 10^8$).

Output

Output three integers x, y, h — the coordinates of the center (x, y) of the optimal pyramid and its height.

Examples

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

Problem J. Just the Last Digit

Time limit:	2 seconds
Memory limit:	512 megabytes

Jessie has recently started jogging and tracking the progress with a fitness band. There are n spectacular spots on a nearby hill. As running uphill is hard for an amateur jogger, Jessie is only going to run downhill. The spots have numbers from 1 to n, the higher the number is, the lower the spot is. Some pairs of spots are connected by trails, and for our purpose, we will only consider trails $i \to j$ going from a higher spot to a lower spot ($i < j$).
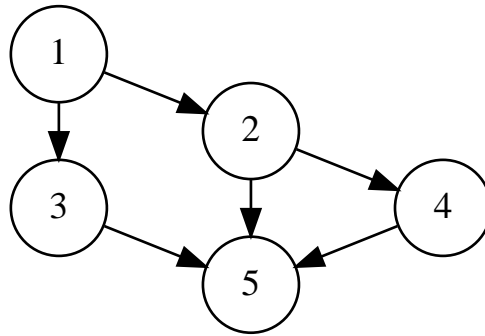
Jessie successfully finished some number of jogs, running through each possible sequence of spots, for which a trail between any two consecutive spots exists, exactly once. Now Jessie is planning to restore the map of all trails with the help of data collected by the fitness band. Unfortunately, the display on the band is small, and can only show the last digit of the number of jogs Jessie did between each pair of spots i and j where $1 \le i < j \le n$. Can you help Jessie restore the map of the hill given this data?

Input

The first line of the input contains an integer n — the number of spots on the hill ($2 \le n \le 500$). Next, n lines follow: the i-th line contains n characters $a_{i,1}, a_{i,2}, \ldots, a_{i,n}$. Character $a_{i,j}$ is the last digit of the number of different jogs made by Jessie starting at the i-th spot and ending at the j-th spot. For every $i \ge j$, $a_{i,j} = 0$.

It is guaranteed that a solution always exists for the given input data.

Output

Print n lines, describing the map of the hill in the similar format: the i-th line should contain n characters, where j-th character is 1 if there is a trail from the i-th spot to the j-th spot, and 0 otherwise. For every $i \ge j$, the j-th character in the i-th row must be 0.

Example

standard input	standard output
5	01100
01113	00011
00012	00001
00001	00001
00001	00000
00000	

Problem K. King's Children

Time limit:	2 seconds
Memory limit:	512 megabytes

The King of Byteland has decided to divide the country into provinces, one for each of his children.

The map of Byteland is a rectangular matrix consisting of n rows and m columns. Each king's child has a castle, located in one of the cells of the matrix (each cell contains at most one castle). King wants to split the Byteland in the following way:

- Each province must be a rectangle.

- Each cell of the Byteland territory must belong to exactly one province.

- Each province must contain the castle of exactly one king's child, the owner of this province.

The king loves all his children, but one of them is his favorite. The king wants to split the kingdom in such a way that the area of the province that belongs to his favorite child is as large as possible. Can you help him to solve this tricky task?

Input

The first line of the input contains two integers n and m ($1 \le n, m \le 1000$). The following n lines contain m characters each. Each character represents one cell of the matrix. Character '.' represents an empty cell, uppercase letters from 'A' to 'Z' represent the castles. Letter 'A' corresponds to the castle of the king's favorite child.

All letters are different. There is exactly one letter 'A'.

Output

Output the same matrix, replacing each character '.' with the lowercase letter, corresponding to the owner of the province containing this cell.

Example

standard input	standard output
6 8 ......X. .F...... ...A.... ....P.. ..L.....	xxxxxxXx fFaaaaaa ffaAaaaa ffaaaaaa lllllPpp llLllppp

Problem L. Lengths and Periods

Time limit:	2 seconds
Memory limit:	512 megabytes

In mathematics and computer science, the critical exponent of a string describes the largest number of times its contiguous substring is repeated in a row. The trick is that it can be a fraction. For example, the critical exponent of "Mississippi" is $7/3$, as it contains the substring "ississi", which is of length 7 and period 3.

The formal definition is as follows. Let w and x be non-empty strings. x is said to occur in w with exponent α, for positive rational α, if there is a substring y in w such as $y = x^n x_0$ where x^n is x repeated n times, x_0 is a prefix of x, n is the integer part of α, and the length $|y|$ is equal to $\alpha|x|$. The critical exponent of w is the maximum α over all x^α that occur in w.

Given a string w, find its critical exponent.

Input

The only line contains a string w — a sequence of lowercase English letters ($1 \le |w| \le 200\,000$).

Output

Output the critical exponent of w as an irreducible fraction p/q where p and q are integers without leading zeroes.

Examples

standard input	standard output
mississippi	7/3
abab	2/1

Problem M. Managing Difficulties

Time limit:	2 seconds
Memory limit:	512 megabytes

Every day a new programming problem is published on Codehorses. Thus, n problems will be published in the following n days: the difficulty of the i-th problem is a_i.

Polycarp wants to choose exactly three days i, j and k ($i < j < k$) so that the difference of difficulties on the day j and the day i is equal to the difference of difficulties on the day k and day j. In other words, Polycarp wants equality $a_j - a_i = a_k - a_j$ to be true.

Determine the number of possible ways for Polycarp to choose the three days in the desired way.

Input

The first line contains an integer t — the number of test cases in the input ($1 \le t \le 10$). Then t test case descriptions follow.

The first line of a test case contains an integer n — the number of the days ($3 \le n \le 2000$).

The second line of the test case contains n integers $a_1, a_2, \ldots, a_n$, where a_i is the difficulty of the problem on the i-th day ($1 \le a_i \le 10^9$).

Output

Output t integers — the answers for each of the test cases in the input, in the order they are given. The answer to a test case is the number of triples of indices i, j, and k such that $1 \le i < j < k \le n$ and $a_k - a_j = a_j - a_i$.

Example

standard input	standard output
4	1
5	1
1 2 1 2 1	4
3	5
30 20 10	
5	
1 2 2 3 4	
9	
3 1 4 1 5 9 2 6 5	