Problem A. King's Inspection

Time limit:	1 second
Memory limit:	512 megabytes

The Far Far Away Kingdom treasury is in trouble — the King is coming with an inspection. It is known that the King likes order. So the chief treasurer needs to put the three chests that store the Kingdom gold coins in order. He looked into the chests and found out that they contain a, b and c coins, respectively. But the King would definitely like the chests to contain the same number of coins each.

To make the number of coins equal, the chief treasurer started to put additional coins to them with both hands. Each second he chooses two distinct chests and adds one more coin to each one of them.

Help the chief treasurer find out how many seconds he needs to make the number of coins in the chests equal.

Input

Input contains three lines.

The first line contains an integer a. The second line contains an integer b. The third line contains an integer c ($1 \le a, b, c \le 5 \cdot 10^8$).

Output

Output one integer — the minimum number of seconds that the chief treasurer needs to make the number of coins in the chests equal.

Examples

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

Note

One optimal strategy for the first test is the following: first, add a coin to chests 1 and 3, then add a coin to chests 1 and 2, and again to chests 1 and 2.

All chests in the second test already contain equal number of coins.

Problem B. Cash Gap

Time limit:	1 second
Memory limit:	512 megabytes

A recently founded "NERC" (New Era Russian Coders) company, of course, has an accounting department. And they are very concerned about the budget of the company. In particular, they don't want the company to experience a *cash gap* — a condition where they need to pay more money than they have in their account now.

At the moment the company has s euros on its account. The chief accountant has prepared a plan of transactions for m following days. During this period n transactions are planned.

For each transaction the change it makes to the company's account is known, but the exact date of the operation is unknown. Money transferred by clients may not be credited to the account immediately, and on the other hand, a claim to pay a bill can suddenly be received. For each transaction, only the range of days the transaction can happen is known in advance. If several transactions would happen on the same day, they can be performed in any order.

Help the accounting department to check if there exists an order of transactions such that the company will experience a cash gap.

Input

The first line contains integers n, m and s — the number of payments, the number of days in the plan and the initial amount of money the company has ($1 \leqslant n, m \leqslant 1000$; $1 \leqslant s \leqslant 10^6$).

The i-th of the following n lines contains a description of the i-th payment in the following format: "`count from to`", which means that on any day from `from` to `to`, the amount of money will be changed by `count` euros ($-10^6 \leqslant$ `count` $\leqslant 10^6$; $1 \leqslant$ `from` $\leqslant$ `to` $\leqslant m$).

Output

Print "`YES`" if a cash gap can occur, or "`NO`" if such situation is impossible.

Examples

standard input	standard output

`4 3 100`
`100 1 2`
`-100 1 2`
`1 2 3`
`0 3 3`
	`NO`

`4 3 100`
`100 1 2`
`-100 1 2`
`1 2 3`
`-1 2 2`
	`YES`

Note

In the first example there is a single transfer from the account, and thus before it there will be at least 100 euros available in the account, so the cash gap is impossible.

In the second example, it could happen that both transfers from the account (for 100 euros and for 1 euro) will be requested on the second day, before all other transactions. In this case by the time the second transaction occurs there will be no money left on the account — it's a cash gap.

Problem C. Jet Trains

Time limit:	2 seconds
Memory limit:	512 megabytes

The year is 3019. By now the humanity is living in cities floating in the atmosphere on air balloons and jet engines. There are n cities on the planet connected by the jet trains network. If the cities a and b are connected by the train route, people can travel between them in any direction. The a-th city can be reached from the b-th city if and only if there is a path between a and b, where each edge of the path represent a train route.

People of 3019 love to be friends, and it happens so people from the same city have many friends in common. Let us say that the friendship connects not pairs of people, but pairs of cities. The friendship of two cities is mutual.

Sometimes people from a city may decide to organize a celebration and invite all the people from friendly cities. If the celebration is announced to be in city a, all people from the friendly cities of a try to get to a using jet trains network. People from all friendly cities of a that can reach the city a join the celebration.

It has been decided to create a special system called "Celebration 3019" to evaluate the number of people on a certain celebration. There is information about cities which are friends, and the current state of the jet trains network. The system must handle the following requests: "If the celebration is announced in the city v, people from how many cities will attend it?". Besides, there should be an option to add an information about new pair of friendly cities, and about new jet train route. Fortunately, the train routes never cancel and once friended, the cities remain friends forever.

Please help the mankind to develop such a system.

Input

The first line contains three integers n, m, k — number of cities, number of pairs of friendly cities and number of jet trains routes ($1 \le n \le 10^5$, $0 \le m, k \le 10^5$).

The following m lines contain two integers a and b each ($1 \le a, b \le n$, $a \ne b$) — pairs of friendly cities. It is guaranteed that each pair is listed at most once.

The following k lines contain two integers a, b ($1 \le a, b \le n$, $a \ne b$) — pairs of cities connected by the jet train routes. There is at most one route between each pair of cities.

The next line contains the integer q ($0 \le q \le 10^5$) — the number of requests to handle, and the following q lines represent requests.

- Request "`T` a b" means that there is now a jet train route between a and b ($1 \le a, b \le n$, $a \ne b$). It is guaranteed that there was no direct route between a and b before the request.
- Request "`F` a b" means that a and b are friends now ($1 \le a, b \le n$, $a \ne b$). It is guaranteed that a and b were not friends before the request.
- Request "`?` v" means that a request is made to the system: "If the celebraion is announced in the city v, people from how many cities will attend it?" ($1 \le v \le n$)

Output

For each request "`?` v" print the answer to it on a new line.

Example

standard input	standard output
4 2 2 1 2 1 3 1 2 1 4 5 ? 1 F 4 1 ? 1 T 4 3 ? 1	1 2 3

Note

The answer to the first request equals one, because only one of the cities $2, 3$ has a train route to the city 1. The answer to the second request equals two, because before the request cities 1 and 4 became friends, and they had a train route connecting them. The answer to the third request equals three, because there is now possible to travel from the city 3 to the city 1 using trains network: first, people can travel to city 4, and then to city 1.

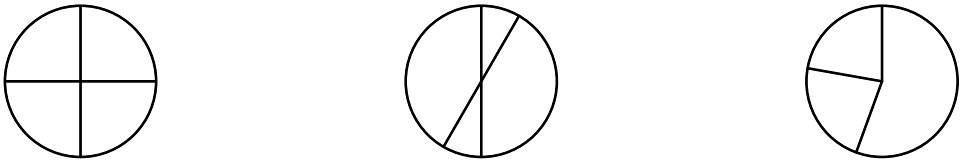
Problem D. Cutting Pizza

Time limit:	1 second
Memory limit:	512 megabytes

Technical committee members were working late and ordered a pizza. When they were delivered a round, tasty and satisfying pizza, they faced a problem: they wanted to cut this pizza as quickly as possible.

Each technical committee member announced the number α_i, which describes how much pizza she wants to eat. For convenience, they decided to use the angular measure of the desired piece in degrees: the number α_i means that this person wants to get exactly one piece of pizza in the form of a sector with an angular measure of α_i degrees.

They use a special pizza knife to cut pizza. It can make two types of cuts: straight cuts from the center of the pizza to the edge (the knife passes the radius of the circle) and straight cuts from one edge to the other edge through the center (the knife passes the diameter of the circle). After cutting, the obtained sectors can be distributed in any order, one to each eater, and the remaining pieces remain in the box.



What is the minimum number of cuts they need to make, so that each member of the technical committee gets exactly the requested piece of pizza?

Input

The first line contains an integer n — the size of the technical committee ($1 \le n \le 16$). The next line contains n integers α_i — angular measures in degrees of the desired pizza sectors ($1 \le \alpha_i < 360$).

It is guaranteed that the sum of all α_i does not exceed 360.

Output

Output the integer m in the first line — the minimum number of cuts they need. Output descriptions of the cuts in the next m lines.

Each cut is represented by two integers β_i ($0 \le \beta_i < 360$) and $type_i$ — the rotation angle and the type of the cut. The rotation angle is measured from a fixed axis in a fixed direction (for example, from a north-directed axis, clockwise). The type of the cut $type_i$ is 0 if the knife passes the radius, and is 1 if the knife passes the diameter.

Examples

standard input	standard output

4
90 90 90 90
	2
	0 1
	90 1

2
30 30
	2
	0 1
	30 1

3
200 80 80
	3
	0 0
	200 0
	280 0

Problem E. Unique Solution

Time limit:	1 second
Memory limit:	512 megabytes

Professor is preparing a task for higher math students.

The task is the following. The students are given n integers $x_1, x_2, \ldots, x_n$, and an integer m ($1 \le m < 2^n$).

The student must choose n integers $a_1, a_2, \ldots, a_n$, each either -1, 0, or 1, at least one non-zero value be chosen. The chosen integers must satisfy the condition that $a_1x_1 + a_2x_2 + \ldots + a_nx_n$ is divisible by m.

The professor has decided that the answer to the task should be some given array of integers $a_1, a_2, \ldots, a_n$ ($-1 \le a_i \le 1$, at least one of them is not equal to 0). To make his job of checking students' solutions easier, he wants to choose such integers $x_1, x_2, \ldots, x_n$ and an integer m, that his array $a_1, a_2, \ldots, a_n$ is the only possible solution. Unfortunately it is not possible, because the array $-a_1, -a_2, \ldots, -a_n$ is always a solution too.

So the professor relaxes his requirements, and wants the only two solutions be $a_1, a_2, \ldots, a_n$ and $-a_1, -a_2, \ldots, -a_n$.

Help him choose integers $x_1, x_2, \ldots, x_n$ and an integer m.

Input

The first line of input contains an integer n ($1 \le n \le 30$).

The next line contains n integers $a_1, a_2, \ldots, a_n$ ($-1 \le a_i \le 1$). At least one of a_i is not equal to 0.

Output

The first line of output must contain and integer m ($1 \le m < 2^n$).

The next line must contain n integers $x_1, x_2, \ldots, x_n$ ($-2^{30} < x_i < 2^{30}$).

If there are several possible answers, output any of them.

It is known that the answer always exists.

Example

standard input	standard output
2 1 -1	3 1 4

Note

In the given example the students must choose a_1 and a_2 so that $a_1 + 4a_2$ is divisible by 3. There are two possible solutions:

- $a_1 = 1$, $a_1 = -1$ ($a_1 + 4a_2 = 1 - 4 = -3$, divisible by 3) and
- $a_1 = -1$, $a_2 = 1$ ($a_1 + 4a_2 = -1 + 4 = 3$, divisible by 3).

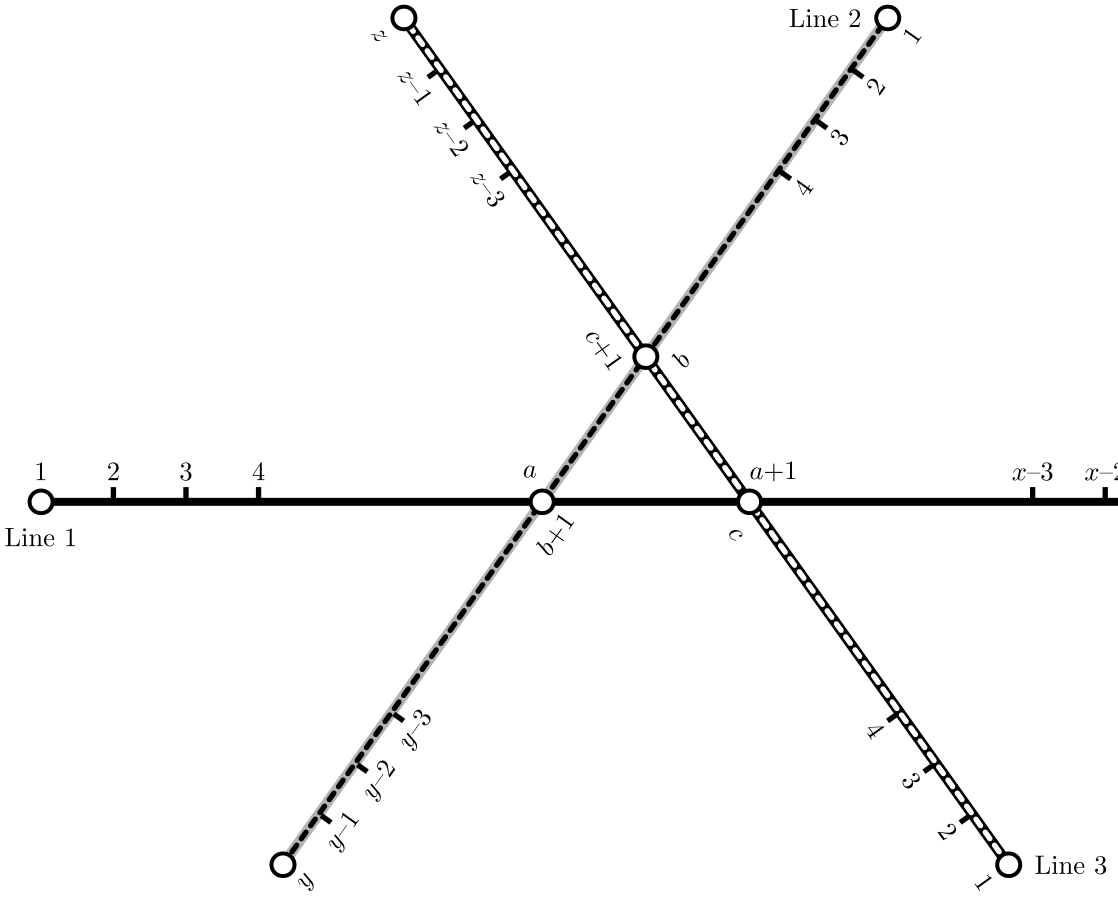
Professor's requirements are met.

Problem F. Metro 2345

Time limit:	1 second
Memory limit:	512 megabytes

Kekopolis has three metro lines, they intersect near the city center, forming a triangle of three tunnels between adjacent stations. The first line has x stations, the second line has y stations, the third line has z stations. Stations at each line are numbered consequently starting from 1.

Stations where the lines intersect allow passengers to change between lines. It is possible to change between station a of the first line and station $b+1$ of the second line, station $a+1$ of the first line and station c of the third line, and station b of the second line and station $c+1$ of the third line.



The lines have different speed of trains. The train at the first line travels between adjacent stations in t_1 seconds, the train at the second line does it in t_2 seconds, the train at the third line does it in t_3 seconds. It takes d seconds to change between lines. Let us ignore the time to enter or leave the station or the train, and the time the train stops at the station.

Dima needs to get to Kekopolis Programming Contest. He starts his way at home that is located near station i of the k-th line, and needs to get to Kekopolis University that is located near station j of the l-th line. Dima overslept as usually, so he is getting late. Dima wants to get to his destination as soon as possible.

Find the minimum time Dima needs to get from his home the the university. Dima doesn't enter nor exists metro at exchange station.

Input

The first line of input contains three integers x, y and z ($2 \le x, y, z \le 10^9$).

The second line contains three integers a, b and c ($1 \le a < x$, $1 \le b < y$, $1 \le c < z$).

The third line contains four integers t_1, t_2, t_3 and d ($1 \le t_1, t_2, t_3, d \le 10^9$).

The fourth line contains four integers k, i, l and j ($1 \le k, l \le 3$, it is guaranteed that the k-th line has station i, the l-th line has station j, the starting station is different from the destination station, in particular they are not the two stations at the same interchange).

Output

Output one integer — the minimum time needed to travel from the starting station to the destination station.

Examples

standard input	standard output

4 4 4
2 2 2
1 1 1 1
1 1 2 1
	4

4 4 4
2 2 2
1 10 1 1
1 1 3 4
	5

4 4 4
2 2 2
1 1 1 1
1 1 1 4
	3

Problem G. Weight Overflow

Time limit:	1 second
Memory limit:	512 megabytes

John has recently bought an electronic balance scale from a well-known website. The scale has two plates that you can put items to. After you put some items to the scale it shows which of the plates outweighs another one, or tells that the weights on the plates are equal, and the scale is balanced.

John has started experimenting with the scale and found out that weighing results are quite strange sometimes. It turns out that the scale calculates the total weight of the items at each plate in grams and then compares the resulting values. However, since the software was written by former sport programmers, the scale calculates the weight of the items at the plate modulo some integer m.

John has n special weights, the i-th weight has the mass of a_i grams. He wants the scale to tell him that it is balanced by putting some of these weights to the scale plates.

Help John decide which weights he needs to put to the scale plates. He should put at least one weight to the scale.

Input

The first line of input contains two integers n and m — the number of weights John has, and the modulo for weight calculation ($1 \le n \le 25$; $1 \le m < 4 \cdot 10^7$).

The next line contains n integers $a_1, \ldots, a_n$ — masses of the weights John has ($1 \le a_i \le 10^9$).

Output

If there exists an answer, print k — the number of weights that John should put to the first plate, at the first line. The second line must contain the numbers of these weights.

Then print m — the number of weights that John should put to the second plate. Print the numbers of these weights at the next line.

Weights are numbered from 1 to n in the order they are given in input.

If there is no answer, print a single number "-1".

Examples

standard input	standard output

4 14
1 3 7 10
	1
	4
	2
	2 3

3 7
1 2 4
	3
	1 2 3
	0

Problem H. Checking Answers to Test

Time limit:	1 second
Memory limit:	512 megabytes

Today students had a test. The test consists of n multiple-choice questions. Each question has four answer options "A", "B", "C" and "D", but only one of them is correct.

The teacher knows that the students often cheat. So he wants to find all pairs of students who have similar answers. Two students have similar answers if for each of them more than half of their correct answers match with the other student's answers, and more than half of their incorrect answers match with the other student's answers.

Help the teacher find all pairs of students who have similar answers to the test.

Input

The first line of input contains one integer n denoting the number of questions in the test ($1 \le n \le 100$).

The second line contains correct answers, they are represented by a string of length n. The string consists of letters "A", "B", "C" and "D", the i-th letter is the correct answer for the i-th question.

The third line contains one integer m denoting the number of students, who had the test today ($1 \le m \le 100$).

Next m lines contain students' answers in the same format as the correct answers. The i-th line contains the answers given by the i-th student.

Output

The first line of output must contain the number of pairs of students who have similar answers. The following lines must contain all such pairs of students, in any order. You can output the numbers in one pair in any order.

Examples

standard input	standard output
3 AAA 4 ABA ABA CBA CAA	1 1 2

6
ABCDAB
3
ABCCCC
BBCDCC
ACCDCC
	3
	1 2
	1 3
	2 3

Problem I. Magic Trick

Time limit:	1 second
Memory limit:	512 megabytes

Artem came to a circus, and now he is going to take part in the magic trick.

Artem secretly chooses permutation — an array of n distinct integers from 1 to n. Let the chosen permutation be $[a_1, a_2, \ldots, a_n]$. For every i from 1 to n Artem takes the set $\{a_i, a_{i+1}, a_{i+2}\}$ (we assume that $a_{n+1} = a_1$, $a_{n+2} = a_2$).

He shuffles the elements of each set, and then shuffles the sets themselves. After that he reports the resulting sets to the Magician.

You are that Magician. You need to figure out the permutation Artem has chosen.

Input

The first line contains an integer n ($3 \le n \le 200\,000$), the number of elements in the permutation.

Each of the following n lines contains three distinct integers $a_{i,1}, a_{i,2}, a_{i,3}$ ($1 \le a_{i,j} \le n$), the sets that Artem has reported to the Magician.

It's guaranteed, that the Artem's sets correspond to at least one valid permutation.

Output

Print the permutation of n elements Artem has secretly chosen.

If there are several possible permutations that can lead to the given sets of triples, print any suitable permutation.

Examples

standard input	standard output

6
3 4 1
5 1 6
5 4 2
2 4 3
2 5 6
6 1 3
	1 3 4 2 5 6

3
1 2 3
2 3 1
1 2 3
	1 2 3

Note

In the second example, you can print any permutation of 1, 2 and 3.

Problem J. Superpermutations

Time limit:	1 second
Memory limit:	512 megabytes

Once, during his informatics class, Dima had solved all the problems about permutations. Then he came up with a problem about a *superpermutation*.

A *superpermutation* of order n is a sequence of integers from 1 to n, such that every permutation of numbers from 1 to n occurs as a continuous subsegment in this sequence.

Dima quickly came up with an algorithm for generating superpermutations:

- $s_1 = [1]$.
- Initially, set s_{n+1} equal to s_n.
- Consider all subsegments of s_n of length n from left to right, in order of increasing of their beginning.
- If a current subsegment $s_n[l, l+1, \ldots, l+n-1]$ is a permutation of numbers from 1 to n (that means that every number from 1 to n occurs exactly once), then insert numbers $n+1, s_n[l], s_n[l+1], \ldots, s_n[l+n-1]$ into s_{n+1} right after the last element $s_n[l+n-1]$ of the corresponding subsegment.

Let's take a look at how to construct superpermutations of order 1, 2, 3 and 4:

By definition, $s_1 = [1]$.

Set $s_2 = [1]$. Consider the only subsegment of length 1 in s_1: $[1]$ is a permutation, so we insert $[2, 1]$ into s_2 after it. The result is $s_2 = [1, \mathbf{2}, \mathbf{1}]$.

Set $s_3 = [1, 2, 1]$. Consider all subsegments of length 2 in s_2. $[1, 2]$ is a permutation, after inserting $[3, 1, 2]$, we get $s_3 = [1, 2, \mathbf{3}, \mathbf{1}, \mathbf{2}, 1]$. $[2, 1]$ is also a permutation, so we insert $[3, 2, 1]$ into s_3, we get $s_3 = [1, 2, 3, 1, 2, 1, \mathbf{3}, \mathbf{2}, \mathbf{1}]$.

Initially, set $s_4 = [1, 2, 3, 1, 2, 1, 3, 2, 1]$. Consider all subsegments of length 3 in s_3:

- $[1, 2, 3]$ is a permutation, so we insert $[4, 1, 2, 3]$ after it. Now $s_4 = [1, 2, 3, \mathbf{4}, \mathbf{1}, \mathbf{2}, \mathbf{3}, 1, 2, 1, 3, 2, 1]$.
- $[2, 3, 1]$ is a permutation, so we insert $[4, 2, 3, 1]$ after it. Now $s_4 = [1, 2, 3, 4, 1, 2, 3, 1, \mathbf{4}, \mathbf{2}, \mathbf{3}, \mathbf{1}, 2, 1, 3, 2, 1]$.
- $[3, 1, 2]$ is a permutation, so we insert $[4, 3, 1, 2]$ after it. Now $s_4 = [1, 2, 3, 4, 1, 2, 3, 1, 4, 2, 3, 1, 2, \mathbf{4}, \mathbf{3}, \mathbf{1}, \mathbf{2}, 1, 3, 2, 1]$.
- $[1, 2, 1]$ is not a permutation, so nothing happens here, we continue with the next subsegment.
- $[2, 1, 3]$ is a permutation, so we insert $[4, 2, 1, 3]$ after it. Now $s_4 = [1, 2, 3, 4, 1, 2, 3, 1, 4, 2, 3, 1, 2, 4, 3, 1, 2, 1, 3, \mathbf{4}, \mathbf{2}, \mathbf{1}, \mathbf{3}, 2, 1]$.
- We do the same with two remaining permutations of length 3: $[1, 3, 2]$ and $[3, 2, 1]$.

Dima noticed that he came up with a pretty efficient way of constructing a superpermutation, because every permutation occurs exactly once. To make sure he did not make a mistake, Dima wants to find a position where a given permutation $a_1, \ldots, a_n$ occurs in his superpermutation s_n. Positions are numbered starting with 1.

Since the length of s_n is huge, you need to find the index of the first element of s_n from which the occurrence of the given permutation starts, modulo $10^9 + 7$.

Input

The first line contains a single integer n — the length of the permutation ($1 \le n \le 300\,000$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ — the given permutation ($1 \le a_i \le n$, all a_i are different).

Output

Output a single integer — the position of the occurrence of the permutation $a_1, a_2, \ldots, a_n$ in the superpermutation of order n, modulo $10^9 + 7$.

Examples

standard input	standard output
3 2 3 1	2
4 2 3 1 4	6
4 4 3 1 2	14

Problem K. Preparing Tests

Time limit:	2 seconds
Memory limit:	512 megabytes

Nick is preparing a problem for a programming contest. The contest is very soon, but he hasn't prepared tests yet. Nick has decided to use multitest concept to simplify the process of test generation: each input contains one or more tests, one immediately after another.

Each test starts with an integer m, and then contains m pairs of integers a_i, b_i. These pairs represent edges of undirected graph, the vertices of the graph are numbered from 0 to infinity. The graph in the test must not contain loops nor multi-edges, and it must be a forest, no cycles are allowed.

Let us consider some examples of correct and incorrect input data:

Input data	**Description**
`3 1 2 2 0 2 3`	One test, it contains a tree of 4 vertices 0, 1, 2 and 3, edges $(1, 2)$, $(2, 0)$, $(2, 3)$. Other vertices are isolated.
`1 2 1 2 2 0 2 3`	Two tests, the first one contains a tree with two vertices 1 and 2, the second one contains a tree with three vertices: 0, 2 and 3.
`2 0 1 2 3`	One test, two trees: one with vertices 0 and 1, another with vertices 2 and 3.
`0`	One test, a graph with no edges.
`0 0 0`	Three tests, each contains a graph with no edges.
~~`1 1 1`~~	Incorrect test, a loop at vertex 1.
~~`2 1 2 1 2`~~	Incorrect test, it contains a multi-edge.
~~`2 1 2 2 1`~~	Incorrect test, it contains a multi-edge.
~~`3 1 2 1 3 2 3`~~	Incorrect test, the graph contains a cycle.
~~`2 1 2 1 3 2 3`~~	Incorrect input, the second test starts with 2, so it must contain two edges descriptions, but only one integer follows.

The task of generating tests got easier with multitest, but still the tests must be generated. Nick has decided to use an array of non-negative integers that he had previously used in another problem. He will choose subarrays of the array as tests for the current problem.

Now Nick wants to know the answer to the question: how many different subarrays of the given arrays are correct inputs for the given problem. Two subarrays are different if their first or their last indices are different.

Input

The first line of input contains n — the number of elements in the array ($1 \le n \le 300\,000$).

The second line contains n integers a_i — the array itself ($0 \le a_i \le 10^9$).

Output

Output one integer — the number of different subarrays that are correct inputs for the given problem.

Examples

standard input	standard output
5 2 1 3 4 1	2
8 1 3 1 2 2 0 2 3	5

Note

The following subarrays are correct inputs in the first test (shown by brackets):

- `[2 1 3 4 1]`
- `2 [1 3 4] 1`

The following subarrays are correct inputs in the second test:

- `[1 3 1] 2 2 0 2 3`
- `1 [3 1 2 2 0 2 3]`
- `1 3 1 [2 2 0 2 3]`
- `1 3 1 2 2 [0] 2 3`
- `[1 3 1 2 2 0 2 3]`