

**Codeforces Round #143 (Div. 2)****A. Team**

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

One day three best friends Petya, Vasya and Tonya decided to form a team and take part in programming contests. Participants are usually offered several problems during programming contests. Long before the start the friends decided that they will implement a problem if at least two of them are sure about the solution. Otherwise, the friends won't write the problem's solution.

This contest offers  $n$  problems to the participants. For each problem we know, which friend is sure about the solution. Help the friends find the number of problems for which they will write a solution.

**Input**

The first input line contains a single integer  $n$  ( $1 \leq n \leq 1000$ ) — the number of problems in the contest. Then  $n$  lines contain three integers each, each integer is either 0 or 1. If the first number in the line equals 1, then Petya is sure about the problem's solution, otherwise he isn't sure. The second number shows Vasya's view on the solution, the third number shows Tonya's view. The numbers on the lines are separated by spaces.

**Output**

Print a single integer — the number of problems the friends will implement on the contest.

**Sample test(s)**

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

input
2 1 0 0 0 1 1
output
1

**Note**

In the first sample Petya and Vasya are sure that they know how to solve the first problem and all three of them know how to solve the second problem. That means that they will write solutions for these problems. Only Petya is sure about the solution for the third problem, but that isn't enough, so the friends won't take it.

In the second sample the friends will only implement the second problem, as Vasya and Tonya are sure about the solution.

## B. Magic, Wizardry and Wonders

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Vasya the Great Magician and Conjuror loves all kinds of miracles and wizardry. In one wave of a magic wand he can turn an object into something else. But, as you all know, there is no better magic in the Universe than the magic of numbers. That's why Vasya adores math and spends a lot of time turning some numbers into some other ones.

This morning he has  $n$  cards with integers lined up in front of him. Each integer is not less than 1, but not greater than  $l$ . When Vasya waves his magic wand, two rightmost cards vanish from the line and a new card magically appears in their place. It contains the difference between the left and the right numbers on the two vanished cards. Vasya was very interested to know what would happen next, and so he waved with his magic wand on and on, until the table had a single card left.

Suppose that Vasya originally had the following cards: 4, 1, 1, 3 (listed from left to right). Then after the first wave the line would be: 4, 1, -2, and after the second one: 4, 3, and after the third one the table would have a single card with number 1.

Please note that in spite of the fact that initially all the numbers on the cards were not less than 1 and not greater than  $l$ , the numbers on the appearing cards can be anything, no restrictions are imposed on them.

It is now evening. Vasya is very tired and wants to return everything back, but does not remember which cards he had in the morning. He only remembers that there were  $n$  cards, they contained integers from 1 to  $l$ , and after all magical actions he was left with a single card containing number  $d$ .

Help Vasya to recover the initial set of cards with numbers.

### Input

The single line contains three space-separated integers:  $n$  ( $2 \leq n \leq 100$ ) — the initial number of cards on the table,  $d$  ( $|d| \leq 10^4$ ) — the number on the card that was left on the table after all the magical actions, and  $l$  ( $1 \leq l \leq 100$ ) — the limits for the initial integers.

### Output

If Vasya is mistaken, that is, if there doesn't exist a set that meets the requirements given in the statement, then print a single number -1, otherwise print the sought set containing  $n$  integers from 1 to  $l$ . Separate the integers by spaces. Print the integers in the order, in which they were written on the cards from left to right. If there are several suitable sets of numbers, you can print any of them.

### Sample test(s)

input
3 3 2
output
2 1 2
input
5 -4 3
output
-1
input
5 -4 4
output
2 4 1 4 1

## C. To Add or Not to Add

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

A piece of paper contains an array of  $n$  integers  $a_1, a_2, \dots, a_n$ . Your task is to find a number that occurs the maximum number of times in this array.

However, before looking for such number, you are allowed to perform not more than  $k$  following operations — choose an arbitrary element from the array and add 1 to it. In other words, you are allowed to increase some array element by 1 no more than  $k$  times (you are allowed to increase the same element of the array multiple times).

Your task is to find the maximum number of occurrences of some number in the array after performing no more than  $k$  allowed operations. If there are several such numbers, your task is to find the minimum one.

### Input

The first line contains two integers  $n$  and  $k$  ( $1 \leq n \leq 10^5$ ;  $0 \leq k \leq 10^9$ ) — the number of elements in the array and the number of operations you are allowed to perform, correspondingly.

The third line contains a sequence of  $n$  integers  $a_1, a_2, \dots, a_n$  ( $|a_i| \leq 10^9$ ) — the initial array. The numbers in the lines are separated by single spaces.

### Output

In a single line print two numbers — the maximum number of occurrences of some number in the array after at most  $k$  allowed operations are performed, and the minimum number that reaches the given maximum. Separate the printed numbers by whitespaces.

### Sample test(s)

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

### Note

In the first sample your task is to increase the second element of the array once and increase the fifth element of the array twice. Thus, we get sequence 6, 4, 4, 0, 4, where number 4 occurs 3 times.

In the second sample you don't need to perform a single operation or increase each element by one. If we do nothing, we get array 5, 5, 5, if we increase each by one, we get 6, 6, 6. In both cases the maximum number of occurrences equals 3. So we should do nothing, as number 5 is less than number 6.

In the third sample we should increase the second array element once and the fifth element once. Thus, we get sequence 3, 2, 2, 2, 2, where number 2 occurs 4 times.

## D. Magic Box

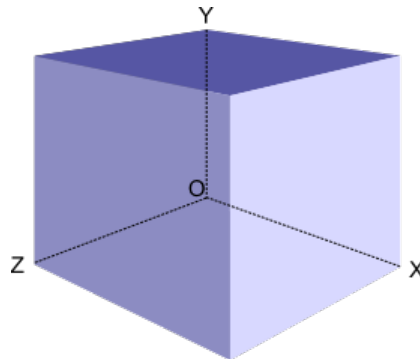
time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

One day Vasya was going home when he saw a box lying on the road. The box can be represented as a rectangular parallelepiped. Vasya needed no time to realize that the box is special, as all its edges are parallel to the coordinate axes, one of its vertices is at point  $(0, 0, 0)$ , and the opposite one is at point  $(x_1, y_1, z_1)$ . The six faces of the box contain some numbers  $a_1, a_2, \dots, a_6$ , exactly one number right in the center of each face.



The numbers are located on the box like that:

- number  $a_1$  is written on the face that lies on the ZOX plane;
- $a_2$  is written on the face, parallel to the plane from the previous point;
- $a_3$  is written on the face that lies on the XOY plane;
- $a_4$  is written on the face, parallel to the plane from the previous point;
- $a_5$  is written on the face that lies on the YOZ plane;
- $a_6$  is written on the face, parallel to the plane from the previous point.

At the moment Vasya is looking at the box from point  $(x, y, z)$ . Find the sum of numbers that Vasya sees. Note that all faces of the box are not transparent and Vasya can't see the numbers through the box. The picture contains transparent faces just to make it easier to perceive. You can consider that if Vasya is looking from point, lying on the plane of some face, then he can not see the number that is written on this face. It is enough to see the center of a face to see the corresponding number for Vasya. Also note that Vasya always reads correctly the  $a_i$  numbers that he sees, independently of their rotation, angle and other factors (that is, for example, if Vasya sees some  $a_i = 6$ , then he can't mistake this number for 9 and so on).

### Input

The first input line contains three space-separated integers  $x, y$  and  $z$  ( $|x|, |y|, |z| \leq 10^6$ ) — the coordinates of Vasya's position in space. The second line contains three space-separated integers  $x_1, y_1, z_1$  ( $1 \leq x_1, y_1, z_1 \leq 10^6$ ) — the coordinates of the box's vertex that is opposite to the vertex at point  $(0, 0, 0)$ . The third line contains six space-separated integers  $a_1, a_2, \dots, a_6$  ( $1 \leq a_i \leq 10^6$ ) — the numbers that are written on the box faces.

It is guaranteed that point  $(x, y, z)$  is located strictly outside the box.

### Output

Print a single integer — the sum of all numbers on the box faces that Vasya sees.

### Sample test(s)

input
2 2 2 1 1 1 1 2 3 4 5 6
output
12
input
0 0 10 3 2 3 1 2 3 4 5 6
output
4

### Note

The first sample corresponds to perspective, depicted on the picture. Vasya sees numbers  $a_2$  (on the top face that is the darkest),  $a_6$  (on the right face that is the lightest) and  $a_4$  (on the left visible face).

In the second sample Vasya can only see number  $a_4$ .

## E. Cactus

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

A connected undirected graph is called a *vertex cactus*, if each vertex of this graph belongs to at most one simple cycle.

A *simple cycle* in a undirected graph is a sequence of distinct vertices  $v_1, v_2, \dots, v_t$  ( $t > 2$ ), such that for any  $i$  ( $1 \leq i < t$ ) exists an edge between vertices  $v_i$  and  $v_{i+1}$ , and also exists an edge between vertices  $v_1$  and  $v_t$ .

A *simple path* in a undirected graph is a sequence of not necessarily distinct vertices  $v_1, v_2, \dots, v_t$  ( $t > 0$ ), such that for any  $i$  ( $1 \leq i < t$ ) exists an edge between vertices  $v_i$  and  $v_{i+1}$  and furthermore each **edge occurs no more than once**. We'll say that a simple path  $v_1, v_2, \dots, v_t$  starts at vertex  $v_1$  and ends at vertex  $v_t$ .

You've got a graph consisting of  $n$  vertices and  $m$  edges, that is a vertex cactus. Also, you've got a list of  $k$  pairs of interesting vertices  $x_i, y_i$ , for which you want to know the following information — the number of distinct simple paths that start at vertex  $x_i$  and end at vertex  $y_i$ . We will consider two simple paths distinct if the sets of edges of the paths are distinct.

For each pair of interesting vertices count the number of distinct simple paths between them. As this number can be rather large, you should calculate it modulo 1000000007 ( $10^9 + 7$ ).

### Input

The first line contains two space-separated integers  $n, m$  ( $2 \leq n \leq 10^5$ ;  $1 \leq m \leq 10^5$ ) — the number of vertices and edges in the graph, correspondingly. Next  $m$  lines contain the description of the edges: the  $i$ -th line contains two space-separated integers  $a_i, b_i$  ( $1 \leq a_i, b_i \leq n$ ) — the indexes of the vertices connected by the  $i$ -th edge.

The next line contains a single integer  $k$  ( $1 \leq k \leq 10^5$ ) — the number of pairs of interesting vertices. Next  $k$  lines contain the list of pairs of interesting vertices: the  $i$ -th line contains two space-separated numbers  $x_i, y_i$  ( $1 \leq x_i, y_i \leq n$ ;  $x_i \neq y_i$ ) — the indexes of interesting vertices in the  $i$ -th pair.

It is guaranteed that the given graph is a vertex cactus. It is guaranteed that the graph contains no loops or multiple edges. Consider the graph vertices are numbered from 1 to  $n$ .

### Output

Print  $k$  lines: in the  $i$ -th line print a single integer — the number of distinct simple ways, starting at  $x_i$  and ending at  $y_i$ , modulo 1000000007 ( $10^9 + 7$ ).

### Sample test(s)

input
10 11 1 2 2 3 3 4 1 4 3 5 5 6 8 6 8 7 7 6 7 9 9 10 6 1 2 3 5 6 9 9 2 9 3 9 10
output
2 2 2 4 4 1