A. Distance

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Let's denote the Manhattan distance between two points p_1 (with coordinates (x_1, y_1)) and p_2 (with coordinates (x_2, y_2)) as $d(p_1, p_2) = |x_1 - x_2| + |y_1 - y_2|$. For example, the distance between two points with coordinates $(1, 3)$ and $(4, 2)$ is $|1 - 4| + |3 - 2| = 4$.

You are given two points, A and B. The point A has coordinates $(0, 0)$, the point B has coordinates (x, y).

Your goal is to find a point C such that:

- both coordinates of C are non-negative integers;
- $d(A, C) = \dfrac{d(A, B)}{2}$ (without any rounding);
- $d(B, C) = \dfrac{d(A, B)}{2}$ (without any rounding).

Find any point C that meets these constraints, or report that no such point exists.

Input

The first line contains one integer t ($1 \le t \le 3000$) — the number of test cases.

Each test case consists of one line containing two integers x and y ($0 \le x, y \le 50$) — the coordinates of the point B.

Output

For each test case, print the answer on a separate line as follows:

- if it is impossible to find a point C meeting the constraints, print "-1 -1" (without quotes);
- otherwise, print two non-negative integers not exceeding 10^6 — the coordinates of point C meeting the constraints. If there are multiple answers, print any of them. It can be shown that if any such point exists, it's possible to find a point with coordinates not exceeding 10^6 that meets the constraints.

Example

input

```
10
49 3
2 50
13 0
0 41
42 0
0 36
13 37
42 16
42 13
0 0
```

output

```
23 3
1 25
-1 -1
-1 -1
21 0
0 18
13 12
25 4
-1 -1
0 0
```

Note

Explanations for some of the test cases from the example:

- In the first test case, the point B has coordinates $(49, 3)$. If the point C has coordinates $(23, 3)$, then the distance from A to B is $|49 - 0| + |3 - 0| = 52$, the distance from A to C is $|23 - 0| + |3 - 0| = 26$, and the distance from B to C is $|23 - 49| + |3 - 3| = 26$.
- In the second test case, the point B has coordinates $(2, 50)$. If the point C has coordinates $(1, 25)$, then the distance from A to B is $|2 - 0| + |50 - 0| = 52$, the distance from A to C is $|1 - 0| + |25 - 0| = 26$, and the distance from B to C is $|1 - 2| + |25 - 50| = 26$.

- In the third and the fourth test cases, it can be shown that no point with integer coordinates meets the constraints.
- In the fifth test case, the point B has coordinates $(42, 0)$. If the point C has coordinates $(21, 0)$, then the distance from A to B is $|42 - 0| + |0 - 0| = 42$, the distance from A to C is $|21 - 0| + |0 - 0| = 21$, and the distance from B to C is $|21 - 42| + |0 - 0| = 21$.

B. Special Permutation

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

A permutation of length n is an array $p = [p_1, p_2, \dots, p_n]$ which contains every integer from 1 to n (inclusive) exactly once. For example, $p = [4, 2, 6, 5, 3, 1]$ is a permutation of length 6.

You are given three integers n, a and b, where n is an even number. Print any permutation of length n that the minimum among **all its elements of the left half** equals a and the maximum among **all its elements of the right half** equals b. Print -1 if no such permutation exists.

Input

The first line of the input contains one integer t ($1 \le t \le 1000$), the number of test cases in the test. The following t lines contain test case descriptions.

Each test case description contains three integers n, a, b ($2 \le n \le 100$; $1 \le a, b \le n$; $a \ne b$), where n is an even number (i.e. $n \bmod 2 = 0$).

Output

For each test case, print a single line containing any suitable permutation. Print -1 no such permutation exists. If there are multiple answers, print any of them.

Example

input

```
7
6 2 5
6 1 3
6 4 3
4 2 4
10 5 3
2 1 2
2 2 1
```

output

```
4 2 6 5 3 1
-1
6 4 5 1 3 2
3 2 4 1
-1
1 2
2 1
```

C. Chat Ban

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

You are a usual chat user on the most famous streaming platform. Of course, there are some moments when you just want to chill and spam something.

More precisely, you want to spam the emote triangle of size k. It consists of $2k - 1$ messages. The first message consists of one emote, the second one — of two emotes, ..., the k-th one — of k emotes, the $k+1$-th one — of $k - 1$ emotes, ..., and the last one — of one emote.

For example, the emote triangle for $k = 3$ consists of 5 messages:

Of course, most of the channels have auto moderation. Auto moderator of the current chat will ban you right after you spam at least

x emotes in succession (you can assume you are the only user in the chat). Now you are interested — how many messages will you write before getting banned? Or maybe you will not get banned at all (i.e. will write all $2k-1$ messages and complete your emote triangle successfully)? Note that if you get banned as a result of writing a message, this message is also counted.

You have to answer t independent test cases.

Input

The first line of the input contains one integer t ($1 \le t \le 10^4$) — the number of test cases. The next t lines describe test cases.

The only line of the test case contains integers k and x ($1 \le k \le 10^9; 1 \le x \le 10^{18}$).

Output

For each test case, print the number of messages you will write before getting banned for the corresponding values k and x.

Example

input

```
7
4 6
4 7
1 2
3 7
2 5
100 1
1000000000 923456789987654321
```

output

```
3
4
1
4
3
1
1608737403
```

Note

Let's analyze the test cases of the example.

1. In the first test case, you write three messages containing 1, 2 and 3 emotes respectively, and since $1+2+3 \ge 6$, you get banned after that.
2. In the second test case, you write four messages containing 1, 2, 3 and 4 emotes respectively, and since $1+2+3+4 \ge 7$, you get banned after that.
3. In the third test case, you write one message containing exactly 1 emote. It doesn't get you banned, since $1 < 2$, but you have already finished posting your emote triangle. So you wrote one message successfully.
4. In the fourth test case, you write four messages containing 1, 2, 3 and 2 emotes respectively, and since $1+2+3+2 \ge 7$, you get banned after that.
5. In the fifth test case, you write three messages containing 1, 2 and 1 emote respectively. It doesn't get you banned, since $1+2+1 < 5$, but you have already finished posting your emote triangle. So you wrote three messages successfully.
6. In the sixth test case, since $x = 1$, you get banned as soon as you send your first message.
7. The seventh test case is too large to analyze, so we'll skip it.

D. X-Magic Pair

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

You are given a pair of integers (a, b) and an integer x.

You can change the pair in two different ways:

- set (assign) $a := |a-b|$;
- set (assign) $b := |a-b|$,

where $|a-b|$ is the absolute difference between a and b.
The pair (a, b) is called x-magic if x is obtainable either as a or as b using only the given operations (i.e. the pair (a, b) is x-magic if $a = x$ or $b = x$ after some number of operations applied). You can apply the operations any number of times (even zero).

Your task is to find out if the pair (a, b) is x-magic or not.

You have to answer t independent test cases.

Input

The first line of the input contains one integer t ($1 \le t \le 10^4$) — the number of test cases. The next t lines describe test cases.

The only line of the test case contains three integers a, b and x ($1 \le a, b, x \le 10^{18}$).

Output

For the i-th test case, print YES if the corresponding pair (a, b) is x-magic and NO otherwise.

Example

input

```
8
6 9 3
15 38 7
18 8 8
30 30 30
40 50 90
24 28 20
365 216 52
537037812705867558 338887693834423551 3199921013340
```

output

```
YES
YES
YES
YES
NO
YES
YES
YES
```

E. Messages

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Monocarp is a tutor of a group of n students. He communicates with them using a conference in a popular messenger.

Today was a busy day for Monocarp — he was asked to forward a lot of posts and announcements to his group, that's why he had to write a very large number of messages in the conference. Monocarp knows the students in the group he is tutoring quite well, so he understands which message should each student read: Monocarp wants the student i to read the message m_i.

Of course, no one's going to read all the messages in the conference. That's why Monocarp decided to pin some of them. Monocarp can pin any number of messages, and if he wants anyone to read some message, he should pin it — otherwise **it will definitely be skipped by everyone**.

Unfortunately, even if a message is pinned, some students may skip it anyway. For each student i, Monocarp knows that they will read at most k_i messages. Suppose Monocarp pins t messages; if $t \le k_i$, then the i-th student will read all the pinned messages; but if $t > k_i$, the i-th student will choose exactly k_i random pinned messages (all possible subsets of pinned messages of size k_i are equiprobable) and read only the chosen messages.

Monocarp wants to maximize the expected number of students that read their respective messages (i.e. the number of such indices i that student i reads the message m_i). Help him to choose how many (and which) messages should he pin!

Input

The first line contains one integer n ($1 \le n \le 2 \cdot 10^5$) — the number of students in the conference.

Then n lines follow. The i-th line contains two integers m_i and k_i ($1 \le m_i \le 2 \cdot 10^5$; $1 \le k_i \le 20$) — the index of the message which Monocarp wants the i-th student to read and the maximum number of messages the i-th student will read, respectively.

Output

In the first line, print one integer t ($1 \le t \le 2 \cdot 10^5$) — the number of messages Monocarp should pin. In the second line, print t **distinct** integers c_1, c_2, ..., c_t ($1 \le c_i \le 2 \cdot 10^5$) — the indices of the messages Monocarp should pin. The messages can be listed in any order.

If there are multiple answers, print any of them.

Examples

input

```
3
10 1
10 2
5 2
```

output

```
2
5 10
```

input

```
3
10 1
5 2
10 1
```

output

```
1
10
```

input

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

output

```
3
2 3 4
```

input

```
3
13 2
42 2
37 2
```

output

```
3
42 13 37
```

Note

Let's consider the examples from the statement.

1. In the first example, Monocarp pins the messages 5 and 10.
   - if the first student reads the message 5, the second student reads the messages 5 and 10, and the third student reads the messages 5 and 10, the number of students which have read their respective messages will be 2;
   - if the first student reads the message 10, the second student reads the messages 5 and 10, and the third student reads the messages 5 and 10, the number of students which have read their respective messages will be 3.

   So, the expected number of students which will read their respective messages is $\frac{5}{2}$.
2. In the second example, Monocarp pins the message 10.
   - if the first student reads the message 10, the second student reads the message 10, and the third student reads the message 10, the number of students which have read their respective messages will be 2.

   So, the expected number of students which will read their respective messages is 2. If Monocarp had pinned both messages 5 and 10, the expected number of students which read their respective messages would have been 2 as well.
3. In the third example, the expected number of students which will read their respective messages is $\frac{8}{3}$.
4. In the fourth example, the expected number of students which will read their respective messages is 2.

F. Armor and Weapons

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Monocarp plays a computer game. There are n different sets of armor and m different weapons in this game. If a character equips the i-th set of armor and wields the j-th weapon, their power is usually equal to $i + j$; but some combinations of armor and weapons synergize well. Formally, there is a list of q ordered pairs, and if the pair (i, j) belongs to this list, the power of the character equipped with the i-th set of armor and wielding the j-th weapon is not $i + j$, but $i + j + 1$.

Initially, Monocarp's character has got only the 1-st armor set and the 1-st weapon. Monocarp can obtain a new weapon or a new set of armor in one hour. If he wants to obtain the k-th armor set or the k-th weapon, he must possess a combination of an armor set and a weapon that gets his power to k **or greater**. Of course, after Monocarp obtains a weapon or an armor set, he can use it to obtain new armor sets or weapons, but he can go with any of the older armor sets and/or weapons as well.

Monocarp wants to obtain the n-th armor set **and** the m-th weapon. What is the minimum number of hours he has to spend on it?

Input

The first line contains two integers n and m ($2 \le n, m \le 2 \cdot 10^5$) — the number of armor sets and the number of weapons, respectively.

The second line contains one integer q ($0 \le q \le \min(2 \cdot 10^5, nm)$) — the number of combinations that synergize well.

Then q lines follow, the i-th line contains two integers a_i and b_i ($1 \le a_i \le n$; $1 \le b_i \le m$) meaning that the a_i-th armor set synergizes well with the b_i-th weapon. All pairs (a_i, b_i) are distinct.

Output

Print one integer — the minimum number of hours Monocarp has to spend to obtain **both** the n-th armor set and the m-th weapon.

Examples

input

```
3 4
0
```

output

```
3
```

input

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

output

```
2
```

Note

In the first example, Monocarp can obtain the strongest armor set and the strongest weapon as follows:

1. Obtain the 2-nd weapon using the 1-st armor set and the 1-st weapon;
2. Obtain the 3-rd armor set using the 1-st armor set and the 2-nd weapon;
3. Obtain the 4-th weapon using the 3-rd armor set and the 2-nd weapon.

In the second example, Monocarp can obtain the strongest armor set and the strongest weapon as follows:

1. Obtain the 3-rd armor set using the 1-st armor set and the 1-st weapon **(they synergize well, so Monocarp's power is not 2 but 3)**;
2. Obtain the 4-th weapon using the 3-rd armor set and the 1-st weapon.

G. Max Sum Array

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given an array $c = [c_1, c_2, \dots, c_m]$. An array $a = [a_1, a_2, \dots, a_n]$ is constructed in such a way that it consists of integers $1, 2, \dots, m$, and for each $i \in [1, m]$, there are exactly c_i occurrences of integer i in a. So, the number of elements in a is exactly $\sum\limits_{i=1}^{m} c_i$.

Let's define for such array a the value $f(a)$ as

$$f(a) = \sum_{\substack{1 \le i < j \le n \\ a_i = a_j}} j - i.$$

In other words, $f(a)$ is the total sum of distances between all pairs of equal elements.

Your task is to calculate the maximum possible value of $f(a)$ and the number of arrays yielding the maximum possible value of $f(a)$. Two arrays are considered different, if elements at some position differ.

Input

The first line contains a single integer m ($1 \le m \le 5 \cdot 10^5$) — the size of the array c.

The second line contains m integers $c_1, c_2, \dots, c_m$ ($1 \le c_i \le 10^6$) — the array c.

Output

Print two integers — the maximum possible value of $f(a)$ and the number of arrays a with such value. Since both answers may be too large, print them modulo $10^9 + 7$.

Examples

input

```
6
1 1 1 1 1 1
```

output

```
0 720
```

input

```
1
```

```
1000000
```

output

```
499833345 1
```

input

```
7
123 451 234 512 345 123 451
```

output

```
339854850 882811119
```

Note

In the first example, all possible arrays a are permutations of $[1, 2, 3, 4, 5, 6]$. Since each array a will have $f(a) = 0$, so maximum value is $f(a) = 0$ and there are $6! = 720$ such arrays.

In the second example, the only possible array consists of 10^6 ones and its $f(a) = \sum_{1 \le i < j \le 10^6} j - i = 166\,666\,666\,666\,500\,000$ and $166\,666\,666\,666\,500\,000 \bmod 10^9 + 7 = 499\,833\,345$.