Sponsored by Telegram

Codeforces Round #751 (Div. 1)

A. Array Elimination

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

You are given array $_1, \ _2, \ldots, \ $, consisting of non-negative integers.

Let's define operation of "elimination" with integer parameter $(1 \le \ \ \le \ )$ as follows:

- Choose distinct array indices $1 \le \ _1 < \ _2 < \ldots < \ \ \le \ $.
- Calculate $= \ _1 \ \& \ \ _2 \ \& \ \ldots \ \&$, where $\&$ denotes the bitwise AND operation (notes section contains formal definition).
- Subtract from each of $_1, \ \ _2, \ldots,$; all other elements remain untouched.

Find all possible values of , such that it's possible to make all elements of array equal to 0 using a finite number of elimination operations with parameter . It can be proven that exists at least one possible for any array .

Note that you **firstly choose and only after that perform elimination operations with value you've chosen initially**.

Input

Each test contains multiple test cases. The first line contains the number of test cases $(1 \le \ \ \le 10^4)$. Description of the test cases follows.

The first line of each test case contains one integer $(1 \le \ \ \le 200\,000)$ — the length of array .

The second line of each test case contains integers $_1, \ \ _2, \ldots, \ \ (0 \le \ \ < 2^{30})$ — array itself.

It's guaranteed that the sum of over all test cases doesn't exceed $200\,000$.

Output

For each test case, print all values , such that it's possible to make all elements of equal to 0 in a finite number of elimination operations with the given parameter .

Print them in increasing order.

Example

input

```
5
4
4 4 4 4
4
13 7 25 19
6
3 5 3 1 7 1
1
1
5
0 0 0 0 0
```

output

```
1 2 4
1 2
1
1
1 2 3 4 5
```

Note

In the first test case:

- If $=1$, we can make four elimination operations with sets of indices $\{1\}, \{2\}, \{3\}, \{4\}$. Since $\&$ of one element is equal to the element itself, then for each operation $=$, so $- \ = \ - \ = 0$.
- If $=2$, we can make two elimination operations with, for example, sets of indices $\{1, 3\}$ and $\{2, 4\}$: $= \ _1 \ \& \ \ _3 = \ _2 \ \& \ \ _4 = 4 \ \& \ 4 = 4$. For both operations $=4$, so after the first operation $_1 - \ = 0$ and $_3 - \ = 0$, and after the second operation — $_2 - \ = 0$ and $_4 - \ = 0$.
- If $=3$, it's impossible to make all equal to 0. After performing the first operation, we'll get three elements equal to 0 and one equal to 4. After that, all elimination operations won't change anything, since at least one chosen element will always be equal to 0.
- If $=4$, we can make one operation with set $\{1, 2, 3, 4\}$, because $= \ _1 \ \& \ \ _2 \ \& \ \ _3 \ \& \ \ _4 = 4$.

In the second test case, if $=2$ then we can make the following elimination operations:

- Operation with indices $\{1,3\}$: $= {}_1 \& {}_3 = 13 \,\&\, 25 = 9$. ${}_1 - = 13 - 9 = 4$ and ${}_3 - = 25 - 9 = 16$. Array will become equal to $[4,7,16,19]$.
- Operation with indices $\{3,4\}$: $= {}_3 \& {}_4 = 16 \,\&\, 19 = 16$. ${}_3 - = 16 - 16 = 0$ and ${}_4 - = 19 - 16 = 3$. Array will become equal to $[4,7,0,3]$.
- Operation with indices $\{2,4\}$: $= {}_2 \& {}_4 = 7 \,\&\, 3 = 3$. ${}_2 - = 7 - 3 = 4$ and ${}_4 - = 3 - 3 = 0$. Array will become equal to $[4,4,0,0]$.
- Operation with indices $\{1,2\}$: $= {}_1 \& {}_2 = 4 \,\&\, 4 = 4$. ${}_1 - = 4 - 4 = 0$ and ${}_2 - = 4 - 4 = 0$. Array will become equal to $[0,0,0,0]$.

Formal definition of bitwise AND:

Let's define bitwise AND ($\&$) as follows. Suppose we have two non-negative integers and , let's look at their binary representations (possibly, with leading zeroes): $\ldots {}_2 {}_1 {}_0$ and $\ldots {}_2 {}_1 {}_0$. Here, is the -th bit of number , and is the -th bit of number . Let $=$ $\&$ is a result of operation $\&$ on number and . Then binary representation of will be $\ldots {}_2 {}_1 {}_0$, where:

$$= \begin{cases} 1, \text{ if } = 1 \text{ and } = 1 \\ 0, \text{ if } = 0 \text{ or } = 0 \end{cases}$$

B. Frog Traveler

time limit per test: 2 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

Frog Gorf is traveling through Swamp kingdom. Unfortunately, after a poor jump, he fell into a well of meters depth. Now Gorf is on the bottom of the well and has a long way up.

The surface of the well's walls vary in quality: somewhere they are slippery, but somewhere have convenient ledges. In other words, if Gorf is on meters below ground level, then in one jump he can go up on any integer distance from 0 to meters inclusive. (Note that Gorf can't jump down, only up).

Unfortunately, Gorf has to take a break after each jump (including jump on 0 meters). And after jumping up to position meters below ground level, he'll slip exactly meters down while resting.

Calculate the minimum number of jumps Gorf needs to reach ground level.

Input

The first line contains a single integer $(1 \leq \leq 300\,000)$ — the depth of the well.

The second line contains integers ${}_1, {}_2, \ldots,$ $(0 \leq \leq)$, where is the maximum height Gorf can jump from meters below ground level.

The third line contains integers ${}_1, {}_2, \ldots,$ $(0 \leq \leq -)$, where is the distance Gorf will slip down if he takes a break on meters below ground level.

Output

If Gorf can't reach ground level, print -1. Otherwise, firstly print integer — the minimum possible number of jumps.

Then print the sequence ${}_1, {}_2, \ldots,$ where is the depth Gorf'll reach after the -th jump, but before he'll slip down during the break. Ground level is equal to 0.

If there are multiple answers, print any of them.

Examples

input

```
3
0 2 2
1 1 0
```

output

```
2
1 0
```

input

```
2
1 1
1 0
```

output

```
-1
```

input

```
10
0 1 2 3 5 5 6 7 8 5
```

```
9 8 7 1 5 4 3 2 0 0
```

output

```
3
9 4 0
```

Note

In the first example, Gorf is on the bottom of the well and jump to the height 1 meter below ground level. After that he slip down by meter and stays on height 2 meters below ground level. Now, from here, he can reach ground level in one jump.

In the second example, Gorf can jump to one meter below ground level, but will slip down back to the bottom of the well. That's why he can't reach ground level.

In the third example, Gorf can reach ground level only from the height 5 meters below the ground level. And Gorf can reach this height using a series of jumps $10 \Rightarrow 9$ $9 \Rightarrow 4$ 5 where $\Rightarrow$ is the jump and is slipping during breaks.

C. Optimal Insertion

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

You are given two arrays of integers $_1,$ $_2, \ldots,$ and $_1,$ $_2, \ldots,$.

You need to insert all elements of into in an arbitrary way. As a result you will get an array $_1,$ $_2, \ldots,$ $_+$ of size $+$.

Note that you are not allowed to change the order of elements in , while you can insert elements of at arbitrary positions. They can be inserted at the beginning, between any elements of , or at the end. Moreover, elements of can appear in the resulting array in any order.

What is the minimum possible number of inversions in the resulting array ? Recall that an inversion is a pair of indices $(,)$ such that $<$ and $>$.

Input

Each test contains multiple test cases. The first line contains the number of test cases $(1 \leq \leq 10^4)$. Description of the test cases follows.

The first line of each test case contains two integers and $(1 \leq , \leq 10^6)$.

The second line of each test case contains integers $_1,$ $_2, \ldots,$ $(1 \leq \leq 10^9)$.

The third line of each test case contains integers $_1,$ $_2, \ldots,$ $(1 \leq \leq 10^9)$.

It is guaranteed that the sum of for all tests cases in one input doesn't exceed 10^6. The sum of for all tests cases doesn't exceed 10^6 as well.

Output

For each test case, print one integer — the minimum possible number of inversions in the resulting array .

Example

input

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

output

```
0
4
6
```

Note

Below is given the solution to get the optimal answer for each of the example test cases (elements of are underscored).

- In the first test case, $= [\underline{1}, 1, \underline{2}, 2, \underline{3}, 3, 4]$.
- In the second test case, $= [1, 2, \underline{3}, \underline{2}, \underline{1}, 3]$.
- In the third test case, $= [\underline{1}, 1, 3, \underline{3}, \underline{5}, \underline{3}, \underline{1}, 4, 6]$.

D. Difficult Mountain

time limit per test: 2 seconds

memory limit per test: 512 megabytes
input: standard input
output: standard output

A group of alpinists has just reached the foot of the mountain. The initial difficulty of climbing this mountain can be described as an integer .

Each alpinist can be described by two integers and , where is his skill of climbing mountains and is his neatness.

An alpinist of skill level is able to climb a mountain of difficulty only if $\leq$. As an alpinist climbs a mountain, they affect the path and thus may change mountain difficulty. Specifically, if an alpinist of neatness climbs a mountain of difficulty the difficulty of this mountain becomes $\max(\ , \ )$.

Alpinists will climb the mountain one by one. And before the start, they wonder, what is the maximum number of alpinists who will be able to climb the mountain if they choose the right order. As you are the only person in the group who does programming, you are to answer the question.

Note that after the order is chosen, each alpinist who can climb the mountain, must climb the mountain at that time.

Input

The first line contains two integers and $(1 \leq \ \leq 500\,000; 0 \leq \ \leq 10^9)$ — the number of alpinists and the initial difficulty of the mountain.

Each of the next lines contains two integers and $(0 \leq \ , \ \leq 10^9)$ that define the skill of climbing and the neatness of the -th alpinist.

Output

Print one integer equal to the maximum number of alpinists who can climb the mountain if they choose the right order to do so.

Examples

input

```
3 2
2 6
3 5
5 7
```

output

```
2
```

input

```
3 3
2 4
6 4
4 6
```

output

```
2
```

input

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

output

```
3
```

Note

In the first example, alpinists 2 and 3 can climb the mountain if they go in this order. There is no other way to achieve the answer of 2.

In the second example, alpinist 1 is not able to climb because of the initial difficulty of the mountain, while alpinists 2 and 3 can go up in any order.

In the third example, the mountain can be climbed by alpinists 5, 3 and 4 in this particular order. There is no other way to achieve optimal answer.

E. Phys Ed Online

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

Students of one unknown college don't have PE courses. That's why of them decided to visit a gym nearby by themselves. The gym is open for days and has a ticket system. At the -th day, the cost of one ticket is equal to . You are free to buy more than one ticket per day.

You can activate a ticket purchased at day either at day or any day later. Each activated ticket is valid only for days. In other words, if you activate ticket at day , it will be valid only at days $,\ +1, \dots,\ +\ -1$.

You know that the -th student wants to visit the gym at each day from to inclusive. Each student will use the following strategy of visiting the gym at any day ($\le\ \le$):

1. person comes to a desk selling tickets placed near the entrance and buy several tickets with cost apiece (possibly, zero tickets);
2. if the person has at least one activated and still valid ticket, they just go in. Otherwise, they activate one of tickets purchased today or earlier and go in.

Note that each student will visit gym only starting , so each student has to buy at least one ticket at day .

Help students to calculate the minimum amount of money they have to spend in order to go to the gym.

Input

The first line contains three integers , and $(1 \le\ ,\ \le 300\,000;\ 1 \le\ \le\ )$ — the number of days, the number of students and the number of days each ticket is still valid.

The second line contains integers $_1,\ _2, \dots,$ $(1 \le\ \le 10^9)$ — the cost of one ticket at the corresponding day.

Each of the next lines contains two integers and $(1 \le\ \le\ \le\ )$ — the segment of days the corresponding student want to visit the gym.

Output

For each student, print the minimum possible amount of money they have to spend in order to go to the gym at desired days.

Example

input

```
7 5 2
2 15 6 3 7 5 6
1 2
3 7
5 5
7 7
3 5
```

output

```
2
12
7
6
9
```

Note

Let's see how each student have to spend their money:

- The first student should buy one ticket at day 1.
- The second student should buy one ticket at day 3 and two tickets at day 4. Note that student can keep purchased tickets for the next days.
- The third student should buy one ticket at day 5.
- The fourth student should buy one ticket at day 7.
- The fifth student should buy one ticket at day 3 and one at day 4.

F. Two Sorts

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Integers from 1 to (inclusive) were sorted lexicographically (considering integers as strings). As a result, array $_1,\ _2, \dots,$ was obtained.

Calculate value of $(\ _{=1}((\ -\ )\mod 998244353))\mod 10^9+7$.

$\mod$ here means the remainder after division by . This remainder is always non-negative and doesn't exceed -1. For example, $5\mod 3 = 2$, $(-1)\mod 6 = 5$.

Input

The first line contains the single integer $(1 \le\ \le 10^{12})$.

Output

Print one integer — the required sum.

Examples

input
```
3
```
output
```
0
```

input
```
12
```
output
```
994733045
```

input
```
21
```
output
```
978932159
```

input
```
1000000000000
```
output
```
289817887
```

Note

A string is lexicographically smaller than a string if and only if one of the following holds:

- is a prefix of , but $\neq$;
- in the first position where and differ, the string has a letter that appears earlier in the alphabet than the corresponding letter in .

For example, 42 is lexicographically smaller than 6, because they differ in the first digit, and $4 < 6$; $42 < 420$, because 42 is a prefix of 420.

Let's denote 998244353 as .

In the first example, array is equal to $[1, 2, 3]$.

- $(1-1) \mod \quad = 0 \mod \quad = 0$
- $(2-2) \mod \quad = 0 \mod \quad = 0$
- $(3-3) \mod \quad = 0 \mod \quad = 0$

As a result, $(0+0+0) \mod 10^9+7 = 0$

In the second example, array is equal to $[1, 10, 11, 12, 2, 3, 4, 5, 6, 7, 8, 9]$.

- $(1-1) \mod \quad = 0 \mod \quad = 0$
- $(2-10) \mod \quad = (-8) \mod \quad = 998244345$
- $(3-11) \mod \quad = (-8) \mod \quad = 998244345$
- $(4-12) \mod \quad = (-8) \mod \quad = 998244345$
- $(5-2) \mod \quad = 3 \mod \quad = 3$
- $(6-3) \mod \quad = 3 \mod \quad = 3$
- $(7-4) \mod \quad = 3 \mod \quad = 3$
- $(8-5) \mod \quad = 3 \mod \quad = 3$
- $(9-6) \mod \quad = 3 \mod \quad = 3$
- $(10-7) \mod \quad = 3 \mod \quad = 3$
- $(11-8) \mod \quad = 3 \mod \quad = 3$
- $(12-9) \mod \quad = 3 \mod \quad = 3$

As a result, $(0+998244345+998244345+998244345+3+3+3+3+3+3+3+3) \mod 10^9+7 = 2994733059 \mod 10^9+7 = 994733045$