A. Setting up Camp

```
Input file:     standard input
Output file:    standard output
Time limit:     1 second
Memory limit:   256 megabytes
```

The organizing committee plans to take the participants of the Olympiad on a hike after the tour. Currently, the number of tents needed to be taken is being calculated. It is known that each tent can accommodate up to 3 people.

Among the participants, there are a introverts, b extroverts, and c universals:

- Each introvert wants to live in a tent alone. Thus, a tent with an introvert must contain exactly one person — only the introvert himself.
- Each extrovert wants to live in a tent with two others. Thus, the tent with an extrovert must contain exactly three people.
- Each universal is fine with any option (living alone, with one other person, or with two others).

The organizing committee respects the wishes of each participant very much, so they want to fulfill all of them.

Tell us the minimum number of tents needed to be taken so that all participants can be accommodated according to their preferences. If it is impossible to accommodate the participants in a way that fulfills all the wishes, output -1.

Input

Each test consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. This is followed by the descriptions of the test cases.

Each test case is described by a single line containing three integers a, b, c ($0 \le a, b, c \le 10^9$) — the number of introverts, extroverts, and universals, respectively.

Output

For each test case, output a single integer — the minimum number of tents, or -1 if it is impossible to accommodate the participants.

Standard Input	Standard Output
10 1 2 3 1 4 1 1 4 2 1 1 1 1 3 2 19 7 18 0 0 0 7 0 0 0 24 0 1000000000 1000000000 1000000000	3 -1 3 -1 3 28 0 7 8 1666666667

Note

In the first test case, 1 tent will be given to the introverts, 1 tent will be shared by two extroverts and one universal, and the last tent will be shared by two universals. In total, 3 tents are needed.

In the second test case, three extroverts will take 1 tent, and 1 tent will be taken by an introvert. Then, one extrovert and one universal will be left. This extrovert will not be able to live with two others.

,

B. Fireworks

```
Input file:     standard input
Output file:    standard output
Time limit:     1 second
Memory limit:   256 megabytes
```

One of the days of the hike coincided with a holiday, so in the evening at the camp, it was decided to arrange a festive fireworks display. For this purpose, the organizers of the hike bought two installations for launching fireworks and a huge number of shells for launching.

Both installations are turned on simultaneously. The first installation launches fireworks every a minutes (i.e., after $a, 2 \cdot a, 3 \cdot a, \ldots$ minutes after launch). The second installation launches fireworks every b minutes (i.e., after $b, 2 \cdot b, 3 \cdot b, \ldots$ minutes after launch).

Each firework is visible in the sky for $m + 1$ minutes after launch, i.e., if a firework was launched after x minutes after the installations were turned on, it will be visible every minute from x to $x + m$, inclusive. If one firework was launched m minutes after another, both fireworks will be visible for one minute.

What is the maximum number of fireworks that could be seen in the sky at the same time?

Input

Each test consists of several test cases. The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. Then follow the descriptions of the test cases.

The first and only line of each test case contains integers a, b, m ($1 \le a, b, m \le 10^{18}$) — the frequency of launching for the first installation, the second installation, and the time the firework is visible in the sky.

Output

For each set of input data, output a single number — the maximum number of fireworks that can be seen simultaneously.

Standard Input	Standard Output
6 6 7 4 3 4 10 7 8 56 5 6 78123459896 1 1 1 1 1 1000000000000000000	2 7 17 28645268630 4 2000000000000000002

Note

In the first set of input data, the fireworks are visible in the sky for 5 minutes. Since the first installation launches fireworks every 6 minutes, and the second one every 7 minutes, two fireworks launched from the same installation will not be visible in the sky at the same time. At the same time, after 7 minutes from the start of the holiday, one firework from the first and one from the second camp will be visible. Thus, it is possible to see no more than 2 fireworks simultaneously.

In the third set of input data, 17 fireworks will be visible after 112 minutes:

- 9 fireworks launched from the first installation at times [$56, 63, 70, 77, 84, 91, 98, 105, 112$];

- 8 fireworks launched from the second installation at times $[56, 64, 72, 80, 88, 96, 104, 112]$.

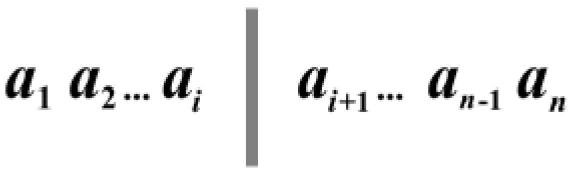
C. Left and Right Houses

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

In the village of Letovo, there are n houses. The villagers decided to build a big road that will divide the village into left and right sides. Each resident wants to live on either the right or the left side of the street, which is described as a sequence $a_1, a_2, \ldots, a_n$, where $a_j = 0$ if the resident of the j-th house wants to live on the left side of the street; otherwise, $a_j = 1$.

The road will pass between two houses. The houses to the left of it will be declared the left-side, and the houses to the right will be declared the right-side. More formally, let the road pass between houses i and $i+1$. Then the houses at positions between 1 and i will be on the **left** side of the street, and at positions between $i+1$ and n will be on the **right** side. The road also **may** pass before the first and after the last house; in this case, the entire village is declared to be either the right or left side, respectively.

To make the design fair, it was decided to lay the road so that at least half of the residents on each side of the village are satisfied with the choice. That is, among x residents on one side, at least $\lceil \frac{x}{2} \rceil$ should want to live on that side, where $\lceil x \rceil$ denotes rounding up a real number x.



To the left of the road, there will be i houses, among the corresponding a_j there must be at least $\lceil \frac{i}{2} \rceil$ zeros. To the right of the road, there will be $n-i$ houses, among the corresponding a_j there must be at least $\lceil \frac{n-i}{2} \rceil$ ones.

Determine after which house i the road should be laid in order to satisfy the described condition and be as close to the middle of the village as possible. Formally, among all suitable positions i, minimize $\left|\frac{n}{2} - i\right|$.

If there are multiple suitable positions i with the minimum $\left|\frac{n}{2} - i\right|$, output the smaller one.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 2 \cdot 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n ($3 \le n \le 3 \cdot 10^5$). The next line of each test case contains a string a of length n, consisting only of 0 and 1.

It is guaranteed that the sum of n over all test cases does not exceed $3 \cdot 10^5$.

Output

For each test case, output a single number i — the position of the house after which the road should be laid (if it should be laid before the first house, output 0). We can show that the answer always exists.

Standard Input	Standard Output
7 3 101	2 3 2

6 010111 6 011001 3 000 3 110 3 001 4 1100	3 0 1 0

Note

Let's consider the first example of input data.

If we lay the road after the first house, there will be one house $a_1 = 1$ on the left side of the street, the resident of which would like to live on the right side of the street. Then 0 out of 1 residents on the even side will be satisfied with the choice, which means that the road cannot be laid after house 1.

If we lay the road after the second house, 1 out of 2 residents on the left side (with preferences $a_1 = 1$, $a_2 = 0$) and 1 out of 1 resident on the right side (with preference $a_3 = 1$) will be satisfied with the choice. More than half of the residents on each side are satisfied with the choice, which means that the road can be laid after house 2. We can show that this is the optimal answer.

,

D. Seraphim the Owl

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

The guys lined up in a queue of n people, starting with person number $i = 1$, to ask Serafim the Owl about the meaning of life. Unfortunately, Kirill was very busy writing the legend for this problem, so he arrived a little later and stood at the end of the line after the n-th person. Kirill is completely dissatisfied with this situation, so he decided to bribe some people ahead of him.

For the i-th person in the queue, Kirill knows two values: a_i and b_i. If at the moment Kirill is standing at position i, then he can choose any position j such that $j < i$ and exchange places with the person at position j. In this case, Kirill will have to pay him a_j coins. And for each k such that $j < k < i$, Kirill will have to pay b_k coins to the person at position k. Kirill can perform this action any number of times.

Kirill is thrifty, so he wants to spend as few coins as possible, but he doesn't want to wait too long, so Kirill believes he should be among the first m people in line.

Help Kirill determine the minimum number of coins he will have to spend in order to not wait too long.

Input

Each test consists of several sets of input data. The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. Then follows the description of the test case.

The first line of each test case contains two integers n and m ($1 \le m \le n \le 200\,000$) — the number of people in the queue besides Kirill and the maximum allowable final position of Kirill, respectively.

The second line contains n integers $a_1, a_2, \dots, a_n$ separated by spaces ($1 \le a_i \le 10^9$).

The third line contains n integers $b_1, b_2, \dots, b_n$ separated by spaces ($1 \le b_i \le 10^9$).

It is guaranteed that the sum of the values of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output a single integer — the minimum number of coins Kirill needs to spend.

Standard Input	Standard Output

4 4 2 7 3 6 9 4 3 8 5 6 2 6 9 7 1 8 3 5 8 8 1 4 1 7 7 7 2 9 2 6 5 9 9 1 10 7 1 4 9 2 1 2 3 1 1	14 22 9 3

,

E. Binary Search

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

Anton got bored during the hike and wanted to solve something. He asked Kirill if he had any new problems, and of course, Kirill had one.

You are given a permutation p of size n, and a number x that needs to be found. A permutation of length n is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, $[2, 3, 1, 5, 4]$ is a permutation, but $[1, 2, 2]$ is not a permutation (2 appears twice in the array), and $[1, 3, 4]$ is also not a permutation ($n = 3$ but there is 4 in the array).

You decided that you are a cool programmer, so you will use an advanced algorithm for the search — binary search. However, you forgot that for binary search, the array must be sorted.

You did not give up and decided to apply this algorithm anyway, and in order to get the correct answer, you can perform the following operation **no more than** 2 times before running the algorithm: choose the indices i, j ($1 \le i, j \le n$) and swap the elements at positions i and j.

After that, the binary search is performed. At the beginning of the algorithm, two variables $l = 1$ and $r = n + 1$ are declared. Then the following loop is executed:

1. If $r - l = 1$, end the loop
2. $m = \lfloor \frac{r+l}{2} \rfloor$
3. If $p_m \le x$, assign $l = m$, otherwise $r = m$.

The goal is to rearrange the numbers in the permutation before the algorithm so that after the algorithm is executed, p_l is equal to x. It can be shown that 2 operations are always sufficient.

Input

Each test consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 2 \cdot 10^4$) — the number of test cases. Then follow the descriptions of the test cases.

The first line of each test case contains two integers n and x ($1 \le x \le n \le 2 \cdot 10^5$) — the length of the permutation and the number to be found.

The second line contains the permutation p separated by spaces ($1 \le p_i \le n$).

It is guaranteed that the sum of the values of n for all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output an integer k ($0 \le k \le 2$) on the first line — the number of operations performed by you. In the next k lines, output 2 integers i, j ($1 \le i, j \le n$) separated by a space, indicating that you are swapping the elements at positions i and j.

Note that you do not need to minimize the number of operations.

Standard Input	Standard Output
5 6 3	0 1

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

,

F. Kirill and Mushrooms

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

As soon as everyone in the camp fell asleep, Kirill sneaked out of the tent and went to the Wise Oak to gather mushrooms.

It is known that there are n mushrooms growing under the Oak, each of which has magic power v_i. Kirill really wants to make a magical elixir of maximum strength from the mushrooms.

The strength of the elixir is equal to the product of the **number** of mushrooms in it and the **minimum** magic power among these mushrooms. To prepare the elixir, Kirill will sequentially pick one mushroom growing under the Oak. Kirill can gather mushrooms in any order.

However, it's not that simple. The Wise Oak informed Kirill of a permutation of numbers p from 1 to n. If Kirill picks only k mushrooms, then the magic power of all mushrooms with indices $p_1, p_2, \dots, p_{k-1}$ will become 0. Kirill will not use mushrooms with zero magic power to prepare the elixir.

Your task is to help Kirill gather mushrooms in such a way that he can brew the elixir of maximum possible strength. However, Kirill is a little scared to stay near the oak for too long, so out of all the suitable options for gathering mushrooms, he asks you to find the one with the minimum number of mushrooms.

A permutation of length n is an array consisting of n different integers from 1 to n in any order. For example, $[2, 3, 1, 5, 4]$ is a permutation, but $[1, 2, 2]$ is not a permutation (2 appears in the array twice) and $[1, 3, 4]$ is also not a permutation ($n = 3$, but 4 appears in the array).

Input

Each test consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 200\,000$) — the number of mushrooms.

The second line contains an array v of size n ($1 \le v_i \le 10^9$) — the magic powers of the mushrooms.

The third line contains a permutation p of numbers from 1 to n.

It is guaranteed that the sum of the values of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output two integers separated by a space — the maximum strength of the elixir that can be brewed and the minimum number of mushrooms that Kirill needs to use for this.

Standard Input	Standard Output
6 3 9 8 14 3 2 1 5 1 2 3 4 5	16 2 9 3 8 2 20 2 5 1 20 2

1 2 3 4 5 6 1 2 3 4 5 6 6 5 4 3 2 1 5 1 4 6 10 10 2 1 4 5 3 4 2 2 5 5 4 2 3 1 5 1 2 9 10 10 1 4 2 3 5	

Note

In the first example, you need to take the mushrooms with indices 1 and 2, so the strength of the elixir is equal to $2 \cdot \min(a_1, a_2) = 2 \cdot \min(9, 8) = 2 \cdot 8 = 16$. Note that the magic power of the mushroom with index 3 after picking two mushrooms will become 0.

G. Cook and Porridge

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

Finally, lunchtime!

n schoolchildren have lined up in a long queue at the cook's tent for porridge. The cook will be serving porridge for D minutes. The schoolchild standing in the i-th position in the queue has a priority of k_i and eats one portion of porridge in s_i minutes.

At the beginning of each minute of the break, the cook serves the first schoolchild in the queue one portion of porridge, after which the schoolchild goes to eat their portion. If the i-th schoolchild is served a portion at the beginning of the x-th minute, then they will return to the queue **at the end** of the $(x + s_i)$-th minute.

When the i-th schoolchild returns to the queue, the schoolchildren at the end of the queue whose priority is **strictly lower** than that of the i-th schoolchild must let them pass. Thus, they will stand in the queue behind the last schoolchild whose priority is **not lower** than their own. That is, behind the last schoolchild j with $k_j \geq k_i$. If there is no such schoolchild in the queue, the i-th schoolchild will stand at the front of the queue.

If several schoolchildren return at the same time, they will return to the queue in ascending order of their s_i.

For example, if $n = 3$, $D = 3$, $k = [2, 3, 2]$, and $s = [2, 1, 3]$, the serving will occur as follows:

- At the beginning of minute 1, the students in the queue are $[1, 2, 3]$, and student 1 is served porridge;
- at the beginning of minute 2, the students in the queue are $[2, 3]$, and student 2 is served porridge;
- at the beginning of minute 3, the student in the queue is $[3]$, and student 3 is served porridge;
- at the end of minute 3, student 2 returns to the queue, and the queue becomes $[2]$;
- at the end of minute 3, student 1 returns to the queue, and the queue becomes $[2, 1]$, as his priority is lower.

Determine the minimum number of minutes after the start of the break that each schoolchild will receive porridge at least once, or report that this will not happen within D minutes.

Input

Each test consists of several test cases. The first line contains a single integer t ($1 \leq t \leq 1000$) — the number of test cases. This is followed by a description of the test cases.

The first line of each test case contains two integers n and D ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq D \leq 3 \cdot 10^5$) — the number of schoolchildren in the queue and the break time, respectively.

The next n lines contain two integers k_i and s_i ($1 \leq k_i, s_i, \leq 10^9$) — the priority and the time to eat one portion of porridge for the respective schoolchild. The schoolchildren are given in the order they stand in the queue (from the front to the end).

It is guaranteed that the sum of the values of n for all input data sets does not exceed $2 \cdot 10^5$. Similarly, it is guaranteed that the sum of the values of D for all input data sets does not exceed $3 \cdot 10^5$.

Output

For each test case, output the minimum number of minutes after which each schoolchild will receive porridge at least once. If this does not happen within the break time, output -1.

Standard Input	Standard Output
7 3 3 2 2 3 1 2 3 5 10 10 3 7 1 11 3 5 1 6 1 5 20 4 2 7 2 8 5 1 5 3 1 5 17 1 3 8 2 8 3 2 2 1 1 5 14 8 2 4 2 1 3 8 3 6 4 1 11 4 5 5 14 8 2 4 2 1 3 8 3 6 4	3 -1 12 6 6 1 6

H. GCD is Greater

```
Input file:     standard input
Output file:    standard output
Time limit:     3 seconds
Memory limit:   512 megabytes
```

In the evenings during the hike, Kirill and Anton decided to take out an array of integers a of length n from their backpack and play a game with it. The rules are as follows:

1. Kirill chooses from 2 to $(n-2)$ numbers and encircles them in red.
2. Anton encircles all the remaining numbers in blue.
3. Kirill calculates the **greatest common divisor** (GCD) of all the red numbers.
4. Anton calculates the **bitwise AND** of all the blue numbers and adds the number x to the result.
5. If the GCD of all the red numbers is strictly greater than the sum of the bitwise AND of all the blue numbers and the number x, then Kirill wins; otherwise, Anton wins.

Help Kirill to beat Anton or tell if it's impossible.

Input

Each test consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 20\,000$) — the number of test cases. Then follows the description of the test cases.

The first line of each test case contains two integers n and x ($4 \le n \le 4 \cdot 10^5$, $0 \le x \le 4 \cdot 10^5$) — the number of integers and the number x respectively.

The second line contains an array a of length n ($1 \le a_i \le 4 \cdot 10^5$).

It is guaranteed that the sum of n for all test cases does not exceed $4 \cdot 10^5$. It is also guaranteed that the sum of the maximum values of a_i for each test case does not exceed $4 \cdot 10^5$.

Output

For each test case, output "YES" on the first line if the condition can be met, on the second line, output the number of chosen numbers by Kirill and the numbers themselves in any order separated by a space, and on the third line, output the size of the second set and the numbers in it.

Otherwise, output "NO".

You can output each letter in any case (lowercase or uppercase). For example, the strings "yEs", "yes", "Yes", and "YES" will be accepted as a positive answer.

Standard Input	Standard Output
8 4 1 4 3 1 8 4 1 4 5 8 4 5 0 1 1 1 1 1 5 2 31 63 127 63 31 4 1	YES 2 4 8 2 3 1 YES 2 4 4 2 5 8 NO YES 2 63 63 3 31 127 31

1 3 3 3 8 3 4 3 4 1 2 2 5 3 4 2 1 4 3 6 8 48 31 61 37 15 53 26 61 12	YES 2 3 3 2 1 3 YES 2 4 4 6 3 1 2 2 5 3 YES 2 3 6 2 1 4 YES 2 61 61 6 31 37 15 53 26 12