A. Entertainment in MAC

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

Congratulations, you have been accepted to the Master's Assistance Center! However, you were extremely bored in class and got tired of doing nothing, so you came up with a game for yourself.

You are given a string s and an **even** integer n. There are two types of operations that you can apply to it:

1. Add the reversed string s to the end of the string s (for example, if $s = \texttt{cpm}$, then after applying the operation $s = \texttt{cpmmpc}$).
2. Reverse the current string s (for example, if $s = \texttt{cpm}$, then after applying the operation $s = \texttt{mpc}$).

It is required to determine the lexicographically smallest[†] string that can be obtained after applying **exactly** n operations. Note that you can apply operations of different types in any order, but you must apply exactly n operations in total.

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

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

Input

Each test consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 500$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains a single **even** integer n ($2 \le n \le 10^9$) — the number of operations applied to the string s.

The second line of each test case contains a single string s ($1 \le |s| \le 100$), consisting of lowercase English letters, — the string to which the operations are applied.

Output

For each test case, output a single line — the lexicographically smallest string that can be obtained after applying exactly n operations.

Standard Input	Standard Output

5 4 cpm 2 grib 10 kupitimilablodarbuz 1000000000 capybara 6 abacaba	cpm birggrib kupitimilablodarbuz arabypaccapybara abacaba

Note

In the first test case, you can apply the operation of the second type (i.e., reverse the string s) 4 times. Then the string s will remain equal to `cpm`.

In the second test case, you can do the following:

- Apply the operation of the second type, after which s will become equal to `birg`.
- Apply operation of the first type (i.e., add the reversed string s to the end of the string s), after which s will become equal to `birggrib`.

B. Informatics in MAC

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

In the Master's Assistance Center, Nyam-Nyam was given a homework assignment in informatics.

There is an array a of length n, and you want to divide it into $k > 1$ subsegments† in such a way that the $\mathrm{MEX}^\ddagger$ on each subsegment is equal to the same integer.

Help Nyam-Nyam find any suitable division, or determine that it does not exist.

†A division of an array into k subsegments is defined as k pairs of integers $(l_1, r_1), (l_2, r_2), \ldots, (l_k, r_k)$ such that $l_i \le r_i$ and for each $1 \le j \le k-1$, $l_{j+1} = r_j + 1$, and also $l_1 = 1$ and $r_k = n$. These pairs represent the subsegments themselves.

‡ MEX of an array is the smallest non-negative integer that does not belong to the array.

For example:

- MEX of the array $[2, 2, 1]$ is 0, because 0 does not belong to the array.
- MEX of the array $[3, 1, 0, 1]$ is 2, because 0 and 1 belong to the array, but 2 does not.
- MEX of the array $[0, 3, 1, 2]$ is 4, because 0, 1, 2, and 3 belong to the array, but 4 does not.

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 ($2 \le n \le 10^5$) — the length of the array a.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ $(0 \le a_i < n)$ — the elements of the array a.

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

Output

For each test case, output a single integer -1 if a suitable division does not exist.

Otherwise, on the first line, output an integer k ($2 \le k \le n$) — the number of subsegments in the division.

Then output k lines — the division into subsegments. The i-th line should contain two integers l_i and r_i ($1 \le l_i \le r_i \le n$) — the boundaries of the i-th subsegment.

The following conditions must be satisfied:

- For all $1 \le j \le k-1$, $l_{j+1} = r_j + 1$;
- $l_1 = 1$, $r_k = n$.

If there are multiple possible solutions, output any of them.

Standard Input	Standard Output

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

Note

In the first test case, the array a can be divided into 2 subsegments with boundaries $[1, 1]$ and $[2, 2]$:

- MEX of the first subsegment $[0]$ is 1, as 0 belongs to the subsegment, but 1 does not.
- MEX of the second subsegment $[0]$ is 1, as 0 belongs to the subsegment, but 1 does not.

In the second test case, it can be proven that the required division does not exist.

In the third test case, the array a can be divided into 3 subsegments with boundaries $[1, 3]$, $[4, 5]$, $[6, 8]$:

- MEX of the first subsegment $[0, 1, 7]$ is 2, as 0 and 1 belong to the subsegment, but 2 does not.
- MEX of the second subsegment $[1, 0]$ is 2, as 0 and 1 belong to the subsegment, but 2 does not.
- MEX of the third subsegment $[1, 0, 3]$ is 2, as 0 and 1 belong to the subsegment, but 2 does not.

C. Messenger in MAC

```
Input file:     standard input
Output file:    standard output
Time limit:     3 seconds
Memory limit:   256 megabytes
```

In the new messenger for the students of the Master's Assistance Center, Keftemerum, an update is planned, in which developers want to optimize the set of messages shown to the user. There are a total of n messages. Each message is characterized by two integers a_i and b_i. The time spent reading the set of messages with numbers $p_1, p_2, \ldots, p_k$ ($1 \le p_i \le n$, all p_i are **distinct**) is calculated by the formula:

$$\sum_{i=1}^{k} a_{p_i} + \sum_{i=1}^{k-1} |b_{p_i} - b_{p_{i+1}}|$$

Note that the time to read a set of messages consisting of **one** message with number p_1 is equal to a_{p_1}. Also, the time to read an empty set of messages is considered to be 0.

The user can determine the time l that he is willing to spend in the messenger. The messenger must inform the user of the maximum possible size of the set of messages, the reading time of which does not exceed l. Note that the maximum size of the set of messages can be equal to 0.

The developers of the popular messenger failed to implement this function, so they asked you to solve this problem.

Input

Each test consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 5 \cdot 10^4$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains two integers n and l ($1 \le n \le 2000$, $1 \le l \le 10^9$) — the number of messages and the time the user is willing to spend in the messenger.

The i-th of the next n lines contains two integers a_i and b_i ($1 \le a_i, b_i \le 10^9$) — characteristics of the i-th message.

It is guaranteed that the sum of n^2 over all test cases does not exceed $4 \cdot 10^6$.

Output

For each test case, output a single integer — the maximum possible size of a set of messages, the reading time of which does not exceed l.

Standard Input	Standard Output
5 5 8 4 3 1 5 2 4 4 3 2 3 1 6	3 1 2 1 0

4 10 3 12 4 8 2 1 2 12 5 26 24 7 8 28 30 22 3 8 17 17 5 14 15 3 1000000000 998244353 179 239 228 1337 993 1007	

Note

In the first test case, you can take a set of three messages with numbers $p_1 = 3$, $p_2 = 2$, and $p_3 = 5$. The time spent reading this set is equal to
$a_3 + a_2 + a_5 + |b_3 - b_2| + |b_2 - b_5| = 2 + 1 + 2 + |4 - 5| + |5 - 3| = 8$.

In the second test case, you can take a set of one message with number $p_1 = 1$. The time spent reading this set is equal to $a_1 = 4$.

In the fifth test case, it can be shown that there is no such non-empty set of messages, the reading time of which does not exceed l.

D. Exam in MAC

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

The Master's Assistance Center has announced an entrance exam, which consists of the following.

The candidate is given a set s of size n and some strange integer c. For this set, it is needed to calculate the number of pairs of integers (x, y) such that $0 \leq x \leq y \leq c$, $x + y$ **is not** contained in the set s, and also $y - x$ **is not** contained in the set s.

Your friend wants to enter the Center. Help him pass the exam!

Input

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

The first line of each test case contains two integers n and c ($1 \leq n \leq 3 \cdot 10^5$, $1 \leq c \leq 10^9$) — the size of the set and the strange integer.

The second line of each test case contains n integers $s_1, s_2, \ldots, s_n$ ($0 \leq s_1 < s_2 < \ldots < s_n \leq c$) — the elements of the set s.

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 integer — the number of suitable pairs of integers.

Standard Input	Standard Output
8 3 3 1 2 3 1 179 57 4 6 0 3 5 6 1 1 1 5 10 0 2 4 8 10 5 10 1 3 5 7 9 4 10 2 4 6 7 3 1000000000 228 1337 998244353	3 16139 10 2 33 36 35 499999998999122959

Note

In the first test case, the following pairs are suitable: $(0, 0)$, $(2, 2)$, $(3, 3)$.

In the third test case, the following pairs are suitable: $(0, 1)$, $(0, 2)$, $(0, 4)$, $(1, 3)$, $(2, 6)$, $(3, 4)$, $(3, 5)$, $(4, 5)$, $(4, 6)$, $(5, 6)$.

E. Distance Learning Courses in MAC

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

The New Year has arrived in the Master's Assistance Center, which means it's time to introduce a new feature!

Now students are given distance learning courses, with a total of n courses available. For the i-th distance learning course, a student can receive a grade ranging from x_i to y_i.

However, not all courses may be available to each student. Specifically, the j-th student is only given courses with numbers from l_j to r_j, meaning the distance learning courses with numbers $l_j, l_j + 1, \ldots, r_j$.

The creators of the distance learning courses have decided to determine the final grade in a special way. Let the j-th student receive grades $c_{l_j}, c_{l_j+1}, \ldots, c_{r_j}$ for their distance learning courses. Then their final grade will be equal to $c_{l_j} \mid c_{l_j+1} \mid \ldots \mid c_{r_j}$, where $\mid$ denotes the bitwise OR operation.

Since the chatbot for solving distance learning courses is broken, the students have asked for your help. For each of the q students, tell them the maximum final grade they can achieve.

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. The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the number of distance learning courses.

Each of the following n lines contains two integers x_i and y_i ($0 \le x_i \le y_i < 2^{30}$) — the minimum and maximum grade that can be received for the i-th course.

The next line contains a single integer q ($1 \le q \le 2 \cdot 10^5$) — the number of students.

Each of the following q lines contains two integers l_j and r_j ($1 \le l_j \le r_j \le n$) — the minimum and maximum course numbers accessible to the j-th student.

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

Output

For each test case, output q integers, where the j-th integer is the maximum final grade that the j-th student can achieve.

Standard Input	Standard Output
3 2 0 1 3 4 3 1 1 1 2	1 5 4 15 11 15 15 7 1 3 3 3

2 2 4 1 7 1 7 3 10 2 2 5 1 3 3 4 2 3 1 4 1 2 6 1 2 2 2 0 1 1 1 3 3 0 0 4 3 4 5 5 2 5 1 2	

Note

In the first test case:

1. The maximum grade for the first student is 1:

   - On the first distance learning course, he will receive a grade of 1.

   Therefore, the final grade is 1.
2. The maximum grade for the second student is 5:

   - On the first distance learning course, he will receive a grade of 1.
   - On the second distance learning course, he will receive a grade of 4.

   Therefore, the final grade is $1 \mid 4 = 5$.
3. The maximum grade for the third student is 4:

   - On the second distance learning course, he will receive a grade of 4.

   Therefore, the final grade is 4.

In the second test case:

1. The maximum grade for the first student is 15:

   - On the first distance learning course, he will receive a grade of 7.
   - On the second distance learning course, he will receive a grade of 4.
   - On the third distance learning course, he will receive a grade of 8.

Therefore, the final grade is $7 \mid 4 \mid 8 = 15$.

2. The maximum grade for the second student is 11:

    - On the third distance learning course, he will receive a grade of 9.
    - On the fourth distance learning course, he will receive a grade of 2.

    Therefore, the final grade is $9 \mid 2 = 11$.

F. Andrey's Tree

```
Input file:     standard input
Output file:    standard output
Time limit:     4 seconds
Memory limit:   256 megabytes
```

Master Andrey loves trees† very much, so he has a tree consisting of n vertices.

But it's not that simple. Master Timofey decided to steal one vertex from the tree. If Timofey stole vertex v from the tree, then vertex v and all edges with one end at vertex v are removed from the tree, while the numbers of other vertices remain unchanged. To prevent Andrey from getting upset, Timofey decided to make the resulting graph a tree again. To do this, he can add edges between any vertices a and b, but when adding such an edge, he must pay $|a - b|$ coins to the Master's Assistance Center.

Note that the resulting tree **does not contain** vertex v.

Timofey has not yet decided which vertex v he will remove from the tree, so he wants to know for each vertex $1 \le v \le n$, the minimum number of coins needed to be spent to make the graph a tree again after removing vertex v, as well as which edges need to be added.

† A tree is an undirected connected graph without cycles.

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 ($5 \le n \le 2 \cdot 10^5$) — the number of vertices in Andrey's tree.

The next $n - 1$ lines contain a description of the tree's edges. The i-th of these lines contains two integers u_i and v_i ($1 \le u_i, v_i \le n$) — the numbers of vertices connected by the i-th edge.

It is guaranteed that the given edges form a tree.

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

Output

For each test case, output the answer in the following format:

For each vertex v (in the order from 1 to n), in the first line output two integers w and m — the minimum number of coins that need to be spent to make the graph a tree again after removing vertex v, and the number of added edges.

Then output m lines, each containing two integers a and b ($1 \le a, b \le n, a \ne v, b \ne v$, $a \ne b$) — the ends of the added edge.

If there are multiple ways to add edges, you can output any solution with the minimum cost.

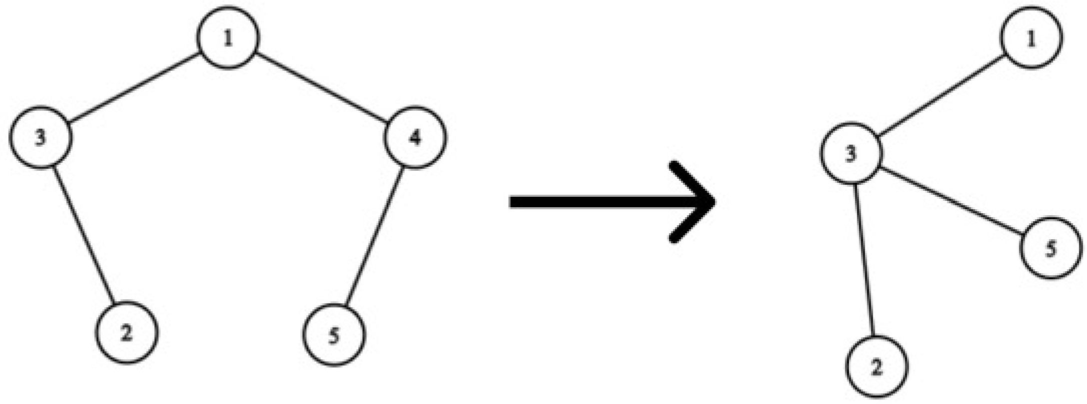
Standard Input	Standard Output
3 5 1 3	1 1 3 4

1 4 4 5 3 2 5 4 2 4 3 3 5 5 1 5 2 1 1 5 1 4 1 3	0 0 1 1 1 2 2 1 3 5 0 0 0 0 0 0 1 1 1 2 1 1 1 2 1 1 1 2 3 3 2 3 4 5 3 4 0 0 0 0 0 0 0 0

Note

In the first test case:

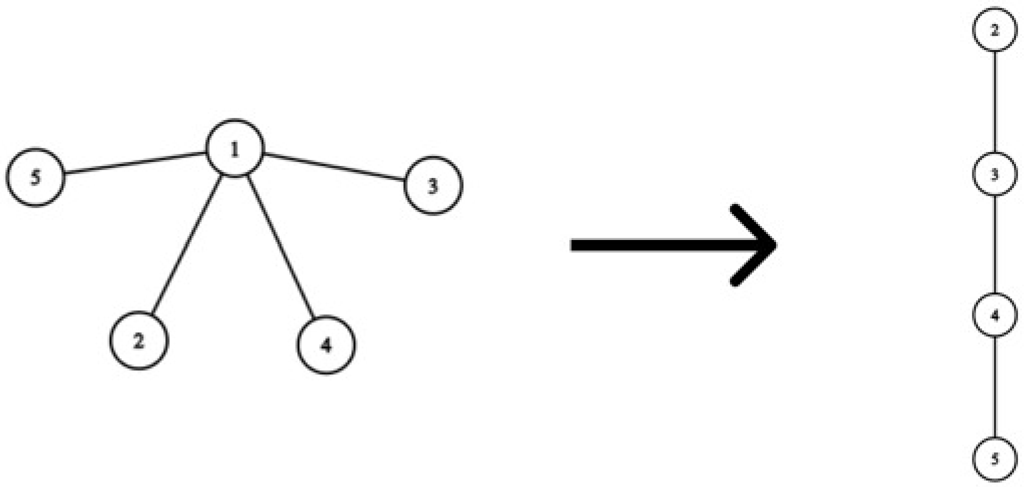
Consider the removal of vertex 4:



The optimal solution would be to add an edge from vertex 5 to vertex 3. Then we will spend $|5-3|=2$ coins.

In the third test case:

Consider the removal of vertex 1:

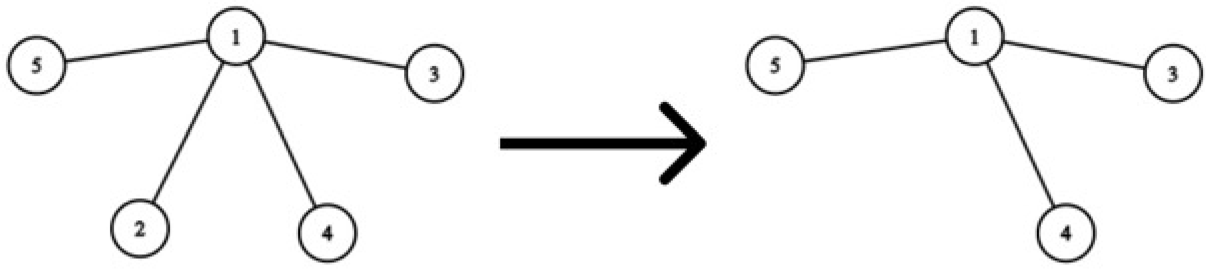


The optimal solution would be:

- Add an edge from vertex 2 to vertex 3, spending $|2-3|=1$ coin.
- Add an edge from vertex 3 to vertex 4, spending $|3-4|=1$ coin.
- Add an edge from vertex 4 to vertex 5, spending $|4-5|=1$ coin.

Then we will spend a total of $1+1+1=3$ coins.

Consider the removal of vertex 2:



No edges need to be added, as the graph will remain a tree after removing the vertex.