A. Turtle and Piggy Are Playing a Game

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

Turtle and Piggy are playing a number game.

First, Turtle will choose an integer x, such that $l \le x \le r$, where l, r are given. It's also guaranteed that $2l \le r$.

Then, Piggy will keep doing the following operation until x becomes 1:

- Choose an integer p such that $p \ge 2$ and $p \mid x$ (i.e. x is a multiple of p).
- Set x to $\frac{x}{p}$, and the score will increase by 1.

The score is initially 0. Both Turtle and Piggy want to maximize the score. Please help them to calculate the maximum score.

Input

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

The first line of each test case contains two integers l, r ($1 \le l \le r \le 10^9, 2l \le r$) — The range where Turtle can choose the integer from.

Output

For each test case, output a single integer — the maximum score.

Standard Input	Standard Output
5 2 4 3 6 2 15 6 22 114514 1919810	2 2 3 4 20

Note

In the first test case, Turtle can choose an integer x, such that $2 \le x \le 4$. He can choose $x = 4$. Then Piggy can choose $p = 2$ for 2 times. After that, x will become 1, and the score will be 2, which is maximized.

In the second test case, Turtle can choose an integer $3 \le x \le 6$. He can choose $x = 6$. Then Piggy can choose $p = 2$, then choose $p = 3$. After that, x will become 1, and the score will be 2, which is maximum.

In the third test case, Turtle can choose $x = 12$.

In the fourth test case, Turtle can choose $x = 16$.

,

B. Turtle and an Infinite Sequence

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

There is a sequence $a_0, a_1, a_2, \ldots$ of infinite length. Initially $a_i = i$ for every non-negative integer i.

After every second, each element of the sequence will **simultaneously** change. a_i will change to $a_{i-1} \mid a_i \mid a_{i+1}$ for every positive integer i. a_0 will change to $a_0 \mid a_1$. Here, $\mid$ denotes bitwise OR.

Turtle is asked to find the value of a_n after m seconds. In particular, if $m = 0$, then he needs to find the initial value of a_n. He is tired of calculating so many values, so please help him!

Input

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

The first line of each test case contains two integers n, m ($0 \le n, m \le 10^9$).

Output

For each test case, output a single integer — the value of a_n after m seconds.

Standard Input	Standard Output
9 0 0 0 1 0 2 1 0 5 2 10 1 20 3 1145 14 19198 10	0 1 3 1 7 11 23 1279 19455

Note

After 1 second, $[a_0, a_1, a_2, a_3, a_4, a_5]$ will become $[1, 3, 3, 7, 7, 7]$.

After 2 seconds, $[a_0, a_1, a_2, a_3, a_4, a_5]$ will become $[3, 3, 7, 7, 7, 7]$.

C. Turtle and an Incomplete Sequence

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

Turtle was playing with a sequence $a_1, a_2, \ldots, a_n$ consisting of positive integers. Unfortunately, some of the integers went missing while playing.

Now the sequence becomes incomplete. There may exist an arbitrary number of indices i such that a_i becomes -1. Let the new sequence be a'.

Turtle is sad. But Turtle remembers that for every integer i from 1 to $n-1$, either $a_i = \left\lfloor \frac{a_{i+1}}{2} \right\rfloor$ or $a_{i+1} = \left\lfloor \frac{a_i}{2} \right\rfloor$ holds for the original sequence a.

Turtle wants you to help him complete the sequence. But sometimes Turtle makes mistakes, so you need to tell him if you can't complete the sequence.

Formally, you need to find another sequence $b_1, b_2, \ldots, b_n$ consisting of positive integers such that:

- For every integer i from 1 to n, if $a'_i \neq -1$, then $b_i = a'_i$.
- For every integer i from 1 to $n-1$, either $b_i = \left\lfloor \frac{b_{i+1}}{2} \right\rfloor$ or $b_{i+1} = \left\lfloor \frac{b_i}{2} \right\rfloor$ holds.
- For every integer i from 1 to n, $1 \leq b_i \leq 10^9$.

If there is no sequence $b_1, b_2, \ldots, b_n$ that satisfies all of the conditions above, you need to report -1.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^5$). The description of the test cases follows.

The first line of each test case contains a single integer n ($2 \leq n \leq 2 \cdot 10^5$) — the length of the sequence.

The second line of each test case contains n integers $a'_1, a'_2, \ldots, a'_n$ ($a'_i = -1$ or $1 \leq a'_i \leq 10^8$) — the elements of the sequence a'.

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

Output

For each test case, if there is no sequence $b_1, b_2, \ldots, b_n$ that satisfies all of the conditions, output a single integer -1.

Otherwise, output n integers $b_1, b_2, \ldots, b_n$ — the elements of the sequence $b_1, b_2, \ldots, b_n$ you find. The sequence should satisfy that $1 \leq b_i \leq 10^9$ for every integer i from 1 to n. If there are multiple answers, print any of them.

Standard Input	Standard Output
9 8 -1 -1 -1 2 -1 -1 1 -1 4 -1 -1 -1 -1	4 9 4 2 4 2 1 2 7 3 6 13 3 1 2 4 9 18 -1 -1

6 3 -1 -1 -1 9 -1 4 -1 5 -1 6 4 2 -1 -1 3 4 1 2 3 4 2 4 2 5 -1 3 -1 3 6 13 -1 -1 3 -1 -1 -1 -1 7 -1 -1 3 -1 -1	-1 4 2 6 3 1 3 6 3 1 3 1 3 7 3 7 3 1 3 1 3

Note

In the first test case, $[4, 2, 1, 2, 1, 2, 1, 3]$ can also be the answer, while $[4, 2, 5, 10, 5, 2, 1, 3]$ and $[4, 2, 1, 2, 1, 2, 1, 4]$ cannot.

In the second test case, $[1, 2, 5, 2]$ can also be the answer.

From the fourth to the sixth test cases, it can be shown that there is no answer, so you should output -1.

,

D. Turtle and Multiplication

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

Turtle just learned how to multiply two integers in his math class, and he was very excited.

Then Piggy gave him an integer n, and asked him to construct a sequence $a_1, a_2, \ldots, a_n$ consisting of integers which satisfied the following conditions:

- For all $1 \leq i \leq n$, $1 \leq a_i \leq 3 \cdot 10^5$.
- For all $1 \leq i < j \leq n-1$, $a_i \cdot a_{i+1} \neq a_j \cdot a_{j+1}$.

Of all such sequences, Piggy asked Turtle to find the one with the **minimum** number of **distinct** elements.

Turtle definitely could not solve the problem, so please help him!

Input

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

The first line of each test case contains a single integer n ($2 \leq n \leq 10^6$) — the length of the sequence a.

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

Output

For each test case, output n integers $a_1, a_2, \ldots, a_n$ — the elements of the sequence a.

If there are multiple answers, print any of them.

Standard Input	Standard Output
3 2 3 4	114514 114514 1 2 2 3 3 4 4

Note

In the third test case, $a = [3, 4, 2, 6]$ violates the second condition since $a_1 \cdot a_2 = a_3 \cdot a_4$. $a = [2, 3, 4, 4]$ satisfy the conditions but its number of distinct elements isn't minimum.

,

E. Turtle and Intersected Segments

```
Input file:     standard input
Output file:    standard output
Time limit:     5 seconds
Memory limit:   512 megabytes
```

Turtle just received n segments and a sequence $a_1, a_2, \ldots, a_n$. The i-th segment is $[l_i, r_i]$.

Turtle will create an undirected graph G. If segment i and segment j intersect, then Turtle will add an undirected edge between i and j with a weight of $|a_i - a_j|$, for every $i \neq j$.

Turtle wants you to calculate the sum of the weights of the edges of the minimum spanning tree of the graph G, or report that the graph G has no spanning tree.

We say two segments $[l_1, r_1]$ and $[l_2, r_2]$ intersect if and only if $\max(l_1, l_2) \le \min(r_1, r_2)$.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^5$). The description of the test cases follows.

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

The i-th of the following n lines contains three integers l_i, r_i, a_i ($1 \le l_i \le r_i \le 10^9, 1 \le a_i \le 10^9$) — the i-th segment and the i-th element of the sequence.

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

Output

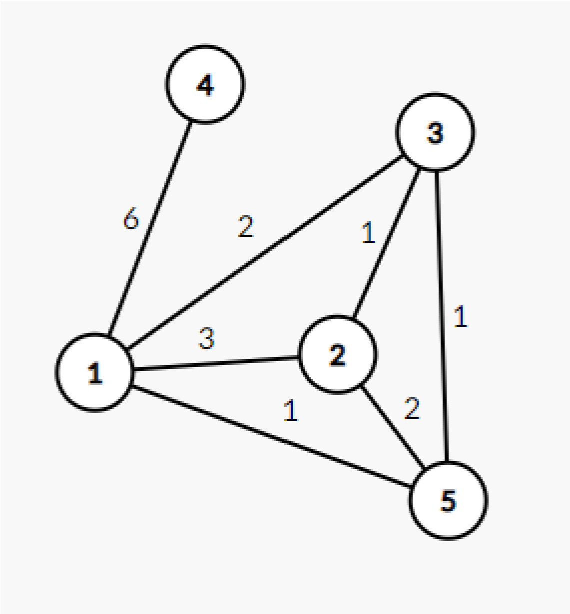
For each test case, output a single integer — the sum of the weights of the edges of the minimum spanning tree of the graph G. If the graph G has no spanning tree, output -1.

Standard Input	Standard Output
4 5 1 7 3 2 4 6 3 5 5 6 7 9 3 4 4 5 2 7 3 1 3 6 4 5 5 6 7 9 1 1 4 4 1 4 3 1 2 1 3 4 5 1 4 4	9 13 4 -1

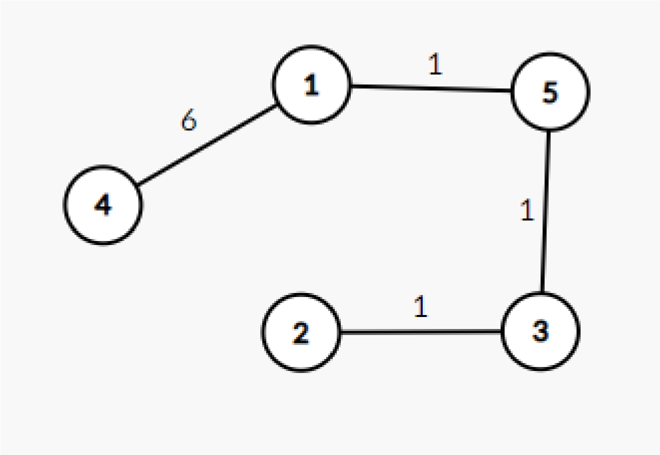
3 1 3 1 2 3 3 4 5 8	

Note

In the first test case, the graph G is as follows:

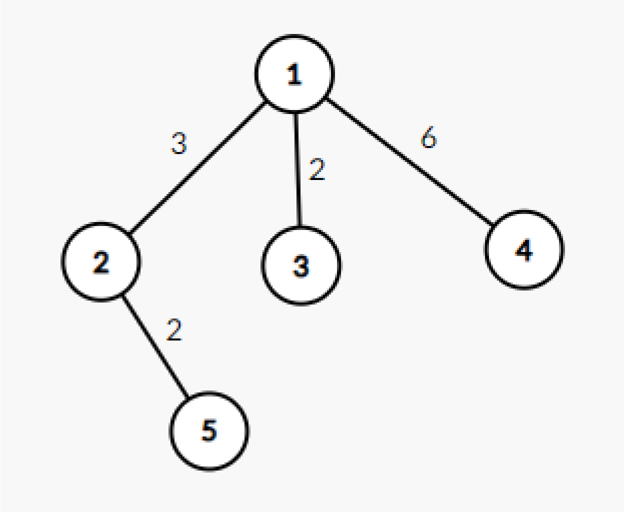


One of the minimum spanning trees of G is as follows:



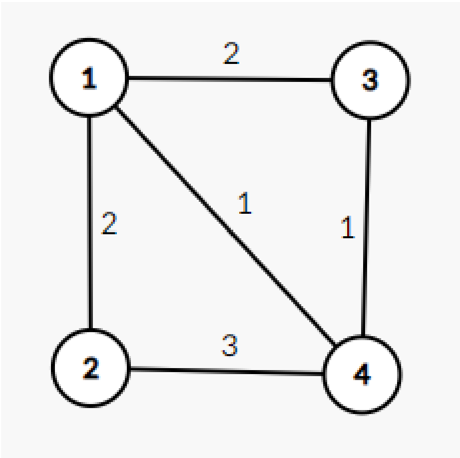
The sum of the weights of the edges of the minimum spanning tree is 9.

In the second test case, the graph G is as follows:

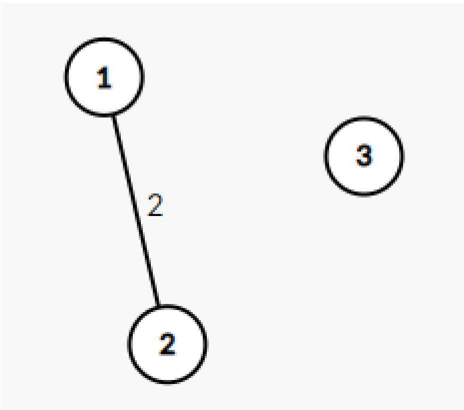


G is already a tree, and the sum of the weights of the tree is 13.

In the third test case, the graph G is as follows:



In the fourth test case, the graph G is as follows:



It's easy to see that G is not connected, so G has no spanning tree.

F. Turtle and Paths on a Tree

```
Input file:     standard input
Output file:    standard output
Time limit:     4 seconds
Memory limit:   1024 megabytes
```

Note the unusual definition of MEX in this problem.

Piggy gave Turtle a **binary tree**† with n vertices and a sequence $a_1, a_2, \ldots, a_n$ on his birthday. The binary tree is rooted at vertex 1.

If a set of paths $P = \{(x_i, y_i)\}$ in the tree covers each edge **exactly once**, then Turtle will think that the set of paths is *good*. Note that a good set of paths can cover a vertex twice or more.

Turtle defines the *value* of a set of paths as $\sum\limits_{(x,y)\in P} f(x, y)$, where $f(x, y)$ denotes the $\operatorname{MEX}^\ddagger$ of all a_u such that vertex u is on the simple path from x to y in the tree (including the starting vertex x and the ending vertex y).

Turtle wonders the **minimum** value over all good sets of paths. Please help him calculate the answer!

†A binary tree is a tree where every non-leaf vertex has at most 2 sons.

$^\ddagger\operatorname{MEX}$ of a collection of integers $c_1, c_2, \ldots, c_k$ is defined as the smallest **positive** integer x which does not occur in the collection c. For example, MEX of $[3, 3, 1, 4]$ is 2, MEX of $[2, 3]$ is 1.

Input

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

The first line of each test case contains a single integer n ($2 \le n \le 2.5 \cdot 10^4$) — the number of vertices in the tree.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the elements of the sequence a.

The third line of each test case contains $n - 1$ integers $p_2, p_3, \ldots, p_n$ ($1 \le p_i < i$) — the parent of each vertex in the tree.

Additional constraint on the input: the given tree is a binary tree, that is, every non-leaf vertex has at most 2 sons.

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

Output

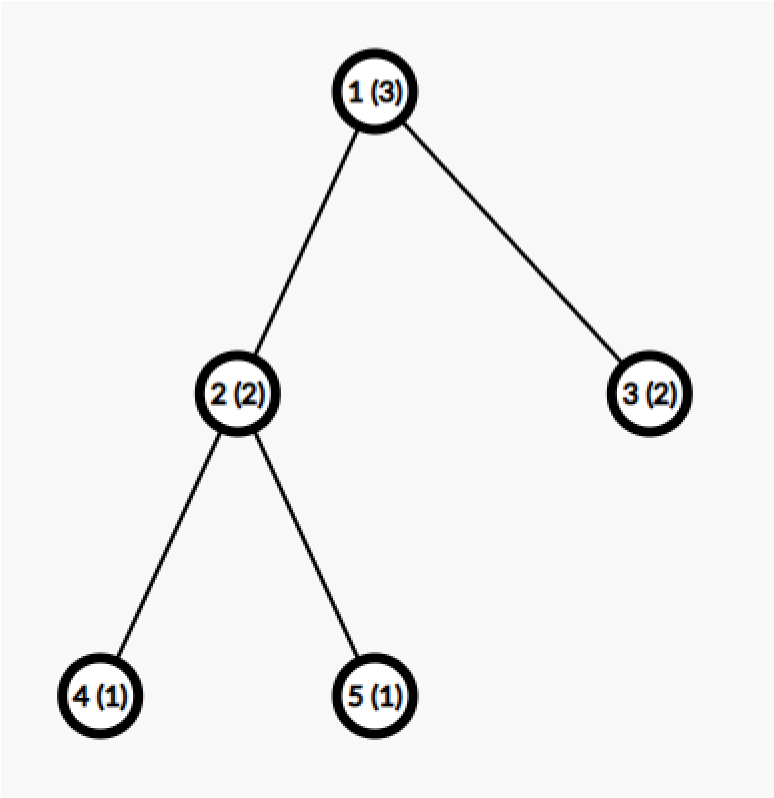
For each test case, output a single integer — the minimum value over all good sets of paths.

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

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

Note

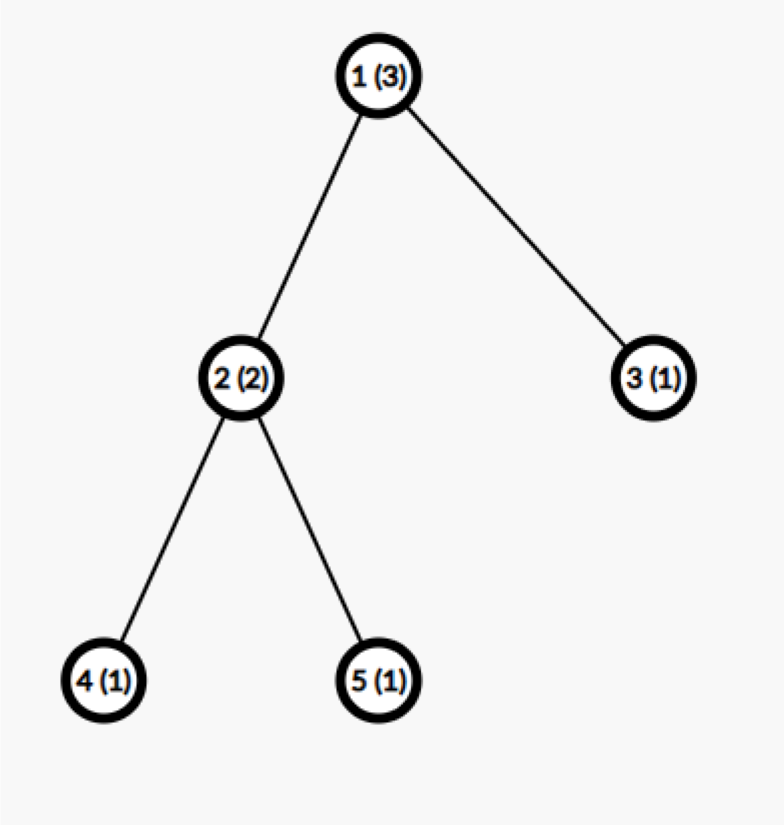
In the first test case, the tree is as follows. The number in brackets denotes the weight of the vertex:



The good set of paths with the minimum value is $\{(2,3),(4,5)\}$.

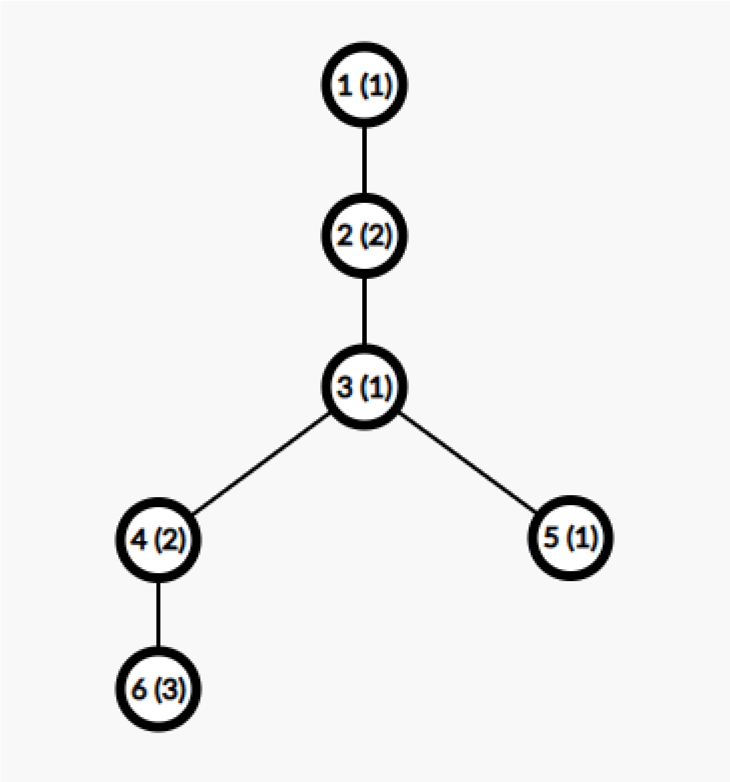
Note that in this test case $\{(4,5)\}$ and $\{(3,4),(4,5)\}$ are not good sets of paths because each edge should be covered exactly once.

In the second test case, the tree is as follows:



The set of good paths with the minimum value is $\{(1,2),(1,3),(4,5)\}$.

In the third test case, the tree is as follows:



The set of good paths with the minimum value is $\{(1,6),(3,5)\}$.