CODEFORCES β
Sponsored by Telegram

Educational Codeforces Round 77 (Rated for Div. 2)

A. Heating

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Several days ago you bought a new house and now you are planning to start a renovation. Since winters in your region can be very cold you need to decide how to heat rooms in your house.

Your house has n rooms. In the i-th room you can install at most c_i heating radiators. Each radiator can have several sections, but the cost of the radiator with k sections is equal to k^2 burles.

Since rooms can have different sizes, you calculated that you need at least sum_i sections in total in the i-th room.

For each room calculate the minimum cost to install at most c_i radiators with total number of sections not less than sum_i.

Input

The first line contains single integer n ($1 \le n \le 1000$) — the number of rooms.

Each of the next n lines contains the description of some room. The i-th line contains two integers c_i and sum_i ($1 \le c_i, sum_i \le 10^4$) — the maximum number of radiators and the minimum total number of sections in the i-th room, respectively.

Output

For each room print one integer — the minimum possible cost to install at most c_i radiators with total number of sections not less than sum_i.

Example

input

```
4
1 10000
10000 1
2 6
4 6
```

output

```
100000000
1
18
10
```

Note

In the first room, you can install only one radiator, so it's optimal to use the radiator with sum_1 sections. The cost of the radiator is equal to $(10^4)^2 = 10^8$.

In the second room, you can install up to 10^4 radiators, but since you need only one section in total, it's optimal to buy one radiator with one section.

In the third room, there 7 variants to install radiators: $[6, 0]$, $[5, 1]$, $[4, 2]$, $[3, 3]$, $[2, 4]$, $[1, 5]$, $[0, 6]$. The optimal variant is $[3, 3]$ and it costs $3^2 + 3^2 = 18$.

B. Obtain Two Zeroes

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given two integers a and b. You may perform any number of operations on them (possibly zero).

During each operation you should choose any positive integer x and set $a := a - x$, $b := b - 2x$ or $a := a - 2x$, $b := b - x$. Note that you may choose different values of x in different operations.

Is it possible to make a and b equal to 0 simultaneously?

Your program should answer t independent test cases.

Input

The first line contains one integer t ($1 \le t \le 100$) — the number of test cases.

Then the test cases follow, each test case is represented by one line containing two integers a and b for this test case ($0 \le a, b \le 10^9$).

Output

For each test case print the answer to it — YES if it is possible to make a and b equal to 0 simultaneously, and NO otherwise.

You may print every letter in any case you want (so, for example, the strings yEs, yes, Yes and YES will all be recognized as positive answer).

Example

input

```
3
6 9
1 1
1 2
```

output

```
YES
NO
YES
```

Note

In the first test case of the example two operations can be used to make both a and b equal to zero:

1. choose $x = 4$ and set $a := a - x$, $b := b - 2x$. Then $a = 6 - 4 = 2$, $b = 9 - 8 = 1$;
2. choose $x = 1$ and set $a := a - 2x$, $b := b - x$. Then $a = 2 - 2 = 0$, $b = 1 - 1 = 0$.

C. Infinite Fence

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

You are a rebel leader and you are planning to start a revolution in your country. But the evil Government found out about your plans and set your punishment in the form of correctional labor.

You must paint a fence which consists of 10^{100} planks in two colors in the following way (suppose planks are numbered from left to right from 0):

- if the index of the plank is divisible by r (such planks have indices 0, r, $2r$ and so on) then you must paint it red;
- if the index of the plank is divisible by b (such planks have indices 0, b, $2b$ and so on) then you must paint it blue;
- if the index is divisible both by r and b **you can choose the color** to paint the plank;
- otherwise, you don't need to paint the plank at all (and it is forbidden to spent paint on it).

Furthermore, the Government added one additional restriction to make your punishment worse. Let's list all **painted** planks of the fence in ascending order: if there are k consecutive planks with the same color in this list, then the Government will state that you failed the labor and execute you immediately. If you don't paint the fence according to the four aforementioned conditions, you will also be executed.

The question is: will you be able to accomplish the labor (the time is not important) or the execution is unavoidable and you need to escape at all costs.

Input

The first line contains single integer T $(1 \le T \le 1000)$ — the number of test cases.

The next T lines contain descriptions of test cases — one per line. Each test case contains three integers r, b, k $(1 \le r, b \le 10^9$, $2 \le k \le 10^9)$ — the corresponding coefficients.

Output

Print T words — one per line. For each test case print REBEL (case insensitive) if the execution is unavoidable or OBEY (case insensitive) otherwise.

Example

input

```
4
1 1 2
2 10 4
5 2 3
3 2 2
```

output

```
OBEY
REBEL
OBEY
OBEY
```

D. A Game with Traps

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are playing a computer game, where you lead a party of m soldiers. Each soldier is characterised by his agility a_i.

The level you are trying to get through can be represented as a straight line segment from point 0 (where you and your squad is initially located) to point $n+1$ (where the boss is located).

The level is filled with k traps. Each trap is represented by three numbers l_i, r_i and d_i. l_i is the location of the trap, and d_i is the danger level of the trap: whenever a soldier with agility lower than d_i steps on a trap (that is, moves to the point l_i), he gets instantly killed. Fortunately, you can disarm traps: if you move to the point r_i, you disarm this trap, and it no longer poses any danger to your soldiers. Traps don't affect you, only your soldiers.

You have t seconds to complete the level — that is, to bring some soldiers from your squad to the boss. Before the level starts, you choose which soldiers will be coming with you, and which soldiers won't be. After that, you have to bring **all of the chosen soldiers** to the boss. To do so, you may perform the following actions:

- if your location is x, you may move to $x+1$ or $x-1$. This action consumes one second;
- if your location is x and the location of your squad is x, you may move to $x+1$ or to $x-1$ with your squad in one second. You may not perform this action if it puts some soldier in danger (i. e. the point your squad is moving into contains a non-disarmed trap with d_i greater than agility of some soldier from the squad). This action consumes one second;
- if your location is x and there is a trap i with $r_i = x$, you may disarm this trap. This action is done instantly (it consumes no time).

Note that after each action both your coordinate and the coordinate of your squad should be integers.

You have to choose the maximum number of soldiers such that they all can be brought from the point 0 to the point $n+1$ (where the boss waits) in no more than t seconds.

Input

The first line contains four integers m, n, k and t ($1 \le m, n, k, t \le 2 \cdot 10^5$, $n < t$) — the number of soldiers, the number of integer points between the squad and the boss, the number of traps and the maximum number of seconds you may spend to bring the squad to the boss, respectively.

The second line contains m integers a_1, a_2, ..., a_m ($1 \le a_i \le 2 \cdot 10^5$), where a_i is the agility of the i-th soldier.

Then k lines follow, containing the descriptions of traps. Each line contains three numbers l_i, r_i and d_i ($1 \le l_i \le r_i \le n$, $1 \le d_i \le 2 \cdot 10^5$) — the location of the trap, the location where the trap can be disarmed, and its danger level, respectively.

Output

Print one integer — the maximum number of soldiers you may choose so that you may bring them all to the boss in no more than t seconds.

Example

input

```
5 6 4 14
1 2 3 4 5
1 5 2
1 2 5
2 3 5
3 5 3
```

output

```
3
```

Note

In the first example you may take soldiers with agility 3, 4 and 5 with you. The course of action is as follows:

- go to 2 without your squad;
- disarm the trap 2;
- go to 3 without your squad;
- disartm the trap 3;
- go to 0 without your squad;
- go to 7 with your squad.

The whole plan can be executed in 13 seconds.

E. Tournament

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

output: standard output

You are organizing a boxing tournament, where n boxers will participate (n is a power of 2), and your friend is one of them. All boxers have different strength from 1 to n, and boxer i wins in the match against boxer j if and only if i is stronger than j.

The tournament will be organized as follows: n boxers will be divided into pairs; the loser in each pair leaves the tournament, and $\frac{n}{2}$ winners advance to the next stage, where they are divided into pairs again, and the winners in all pairs advance to the next stage, and so on, until only one boxer remains (who is declared the winner).

Your friend really wants to win the tournament, but he may be not the strongest boxer. To help your friend win the tournament, you may bribe his opponents: if your friend is fighting with a boxer you have bribed, your friend wins even if his strength is lower.

Furthermore, during each stage you distribute the boxers into pairs as you wish.

The boxer with strength i can be bribed if you pay him a_i dollars. What is the minimum number of dollars you have to spend to make your friend win the tournament, provided that you arrange the boxers into pairs during each stage as you wish?

Input

The first line contains one integer n ($2 \le n \le 2^{18}$) — the number of boxers. n is a power of 2.

The second line contains n integers a_1, a_2, ..., a_n, where a_i is the number of dollars you have to pay if you want to bribe the boxer with strength i. Exactly one of a_i is equal to -1 — it means that the boxer with strength i is your friend. All other values are in the range $[1, 10^9]$.

Output

Print one integer — the minimum number of dollars you have to pay so your friend wins.

Examples

input

```
4
3 9 1 -1
```

output

```
0
```

input

```
8
11 -1 13 19 24 7 17 5
```

output

```
12
```

Note

In the first test case no matter how you will distribute boxers into pairs, your friend is the strongest boxer and anyway wins the tournament.

In the second test case you can distribute boxers as follows (your friend is number 2):

$1 : 2, 8 : 5, 7 : 3, 6 : 4$ (boxers $2, 8, 7$ and 6 advance to the next stage);

$2 : 6, 8 : 7$ (boxers 2 and 8 advance to the next stage, you have to bribe the boxer with strength 6);

$2 : 8$ (you have to bribe the boxer with strength 8);

F. Colored Tree

time limit per test: 2.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You're given a tree with n vertices. The color of the i-th vertex is h_i.

The value of the tree is defined as $\sum\limits_{h_i = h_j, 1 \le i < j \le n} dis(i,j)$, where $dis(i,j)$ is the number of edges on the shortest path between i and j.

The color of each vertex is lost, you only remember that h_i can be any integer from $[l_i, r_i]$(inclusive). You want to calculate the sum of values of all trees meeting these conditions modulo $10^9 + 7$ (the set of edges is fixed, but each color is unknown, so there are $\prod\limits_{i=1}^{n} (r_i - l_i + 1)$ different trees).

Input

The first line contains one integer n ($2 \le n \le 10^5$) — the number of vertices.

Then n lines follow, each line contains two integers l_i and r_i ($1 \le l_i \le r_i \le 10^5$) denoting the range of possible colors of vertex i.

Then $n - 1$ lines follow, each containing two integers u and v ($1 \le u, v \le n$, $u \ne v$) denoting an edge of the tree. It is guaranteed that these edges form a tree.

Output

Print one integer — the sum of values of all possible trees, taken modulo $10^9 + 7$.

Example

input

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

output

```
22
```

Note

In the first example there are four different ways to color the tree (so, there are four different trees):

- a tree with vertices colored as follows: $\{1, 1, 1, 1\}$. The value of this tree is $dis(1, 2) + dis(1, 3) + dis(1, 4) + dis(2, 3) + dis(2, 4) + dis(3, 4) = 10$;
- a tree with vertices colored as follows: $\{1, 2, 1, 1\}$. The value of this tree is $dis(1, 3) + dis(1, 4) + dis(3, 4) = 4$;
- a tree with vertices colored as follows: $\{1, 1, 1, 2\}$. The value of this tree is $dis(1, 2) + dis(1, 3) + dis(2, 3) = 4$;
- a tree with vertices colored as follows: $\{1, 2, 1, 2\}$. The value of this tree is $dis(1, 3) + dis(2, 4) = 4$.

Overall the sum of all values is $10 + 4 + 4 + 4 = 22$.