A. Robin Helps

1 second, 256 megabytes

There is a little bit of the outlaw in everyone, and a little bit of the hero too.

The heroic outlaw Robin Hood is famous for taking from the rich and giving to the poor.

Robin encounters n people starting from the 1-st and ending with the n-th. The i-th person has a_i gold. If $a_i \ge k$, Robin will take all a_i gold, and if $a_i = 0$, Robin will give 1 gold if he has any. Robin starts with 0 gold.

Find out how many people Robin gives gold to.

Input

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

The first line of each test case contains two integers n, k ($1 \le n \le 50, 1 \le k \le 100$) — the number of people and the threshold at which Robin Hood takes the gold.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 100$) — the gold of each person.

Output

For each test case, output a single integer, the number of people that will get gold from Robin Hood.

input

```
4
2 2
2 0
3 2
3 0 0
6 2
0 3 0 0 0 0
2 5
5 4
```

output

```
1
2
3
0
```

In the first test case, Robin takes 2 gold from the first person and gives a gold to the second person.

In the second test case, Robin takes 3 gold and gives 1 gold to each of the next 2 people.

In the third test case, Robin takes 3 gold and so only gives gold to 3 other people.

B. Robin Hood and the Major Oak

1 second, 256 megabytes

In Sherwood, the trees are our shelter, and we are all children of the forest.

The Major Oak in Sherwood is known for its majestic foliage, which provided shelter to Robin Hood and his band of merry men and women.

The Major Oak grows i^i new leaves in the i-th year. It starts with 1 leaf in year 1.

Leaves last for k years on the tree. In other words, leaves grown in year i last between years i and $i + k - 1$ inclusive.

Robin considers even numbers lucky. Help Robin determine whether the Major Oak will have an even number of leaves in year n.

Input

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

Each test case consists of two integers n, k ($1 \le n \le 10^9$, $1 \le k \le n$) — the requested year and the number of years during which the leaves remain.

Output

For each test case, output one line, "YES" if in year n the Major Oak will have an even number of leaves and "NO" otherwise.

You can output the answer in any case (upper or lower). For example, the strings "yEs", "yes", "Yes", and "YES" will be recognized as positive responses.

input

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

output

```
NO
YES
NO
NO
YES
```

In the first test case, there is only 1 leaf.

In the second test case, $k = 1$, so in the 2-nd year there will be $2^2 = 4$ leaves.

In the third test case, $k = 2$, so in the 2-nd year there will be $1 + 2^2 = 5$ leaves.

In the fourth test case, $k = 2$, so in the 3-rd year there will be $2^2 + 3^3 = 4 + 27 = 31$ leaves.

C. Robin Hood in Town

2 seconds, 256 megabytes

In Sherwood, we judge a man not by his wealth, but by his merit.

Look around, the rich are getting richer, and the poor are getting poorer. We need to take from the rich and give to the poor. We need Robin Hood!

There are n people living in the town. Just now, the wealth of the i-th person was a_i gold. But guess what? The richest person has found an extra pot of gold!

More formally, find an $a_j = max(a_1, a_2, \ldots, a_n)$, change a_j to $a_j + x$, where x is a non-negative integer number of gold found in the pot. If there are multiple maxima, it can be any one of them.

A person is unhappy if their wealth is **strictly less than half** of the average wealth*.

If **strictly more than half** of the total population n are unhappy, Robin Hood will appear by popular demand.

Determine the minimum value of x for Robin Hood to appear, or output -1 if it is impossible.

*The average wealth is defined as the total wealth divided by the total population n, that is, $\frac{\sum a_i}{n}$, the result is a real number.

Input

The first line of input contains one integer t ($1 \le t \le 10^4$) — the number of test cases.

The first line of each test case contains an integer n ($1 \le n \le 2 \cdot 10^5$) — the total population.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^6$) — the wealth of each person.

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

Output

For each test case, output one integer — the minimum number of gold that the richest person must find for Robin Hood to appear. If it is impossible, output -1 instead.

input

```
6
1
2
2
2 19
3
1 3 20
4
1 2 3 4
5
1 2 3 4 5
6
1 2 1 1 1 25
```

output

```
-1
-1
0
15
16
0
```

In the first test case, it is impossible for a single person to be unhappy.

In the second test case, there is always 1 happy person (the richest).

In the third test case, no additional gold are required, so the answer is 0.

In the fourth test case, after adding 15 gold, the average wealth becomes $\frac{25}{4}$, and half of this average is $\frac{25}{8}$, resulting in 3 people being unhappy.

In the fifth test case, after adding 16 gold, the average wealth becomes $\frac{31}{5}$, resulting in 3 people being unhappy.

D. Robert Hood and Mrs Hood

2 seconds, 256 megabytes

Impress thy brother, yet fret not thy mother.

Robin's brother and mother are visiting, and Robin gets to choose the start day for each visitor.

All days are numbered from 1 to n. Visitors stay for d continuous days, all of those d days must be between day 1 and n inclusive.

Robin has a total of k risky 'jobs' planned. The i-th job takes place between days l_i and r_i inclusive, for $1 \le i \le k$. If a job takes place on any of the d days, the visit overlaps with this job (the length of overlap is unimportant).

Robin wants his brother's visit to overlap with the maximum number of distinct jobs, and his mother's the minimum.

Find suitable start days for the visits of Robin's brother and mother. If there are multiple suitable days, choose the earliest one.

Input

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

The first line of each test case consists of three integers n, d, k ($1 \le n \le 10^5, 1 \le d, k \le n$) — the number of total days, duration of the visits, and the number of jobs.

Then follow k lines of each test case, each with two integers l_i and r_i ($1 \le l_i \le r_i \le n$) — the start and end day of each job.

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 two integers, the best starting days of Robin's brother and mother respectively. Both visits must fit between day 1 and n inclusive.

input

```
6
2 1 1
1 2
4 1 2
1 2
2 4
7 2 3
1 2
1 3
6 7
5 1 2
1 2
3 5
9 2 1
2 8
9 2 4
7 9
4 8
1 3
2 3
```

output

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

In the first test case, the only job fills all 2 days, both should visit on day 1.

In the second test case, day 2 overlaps with 2 jobs and day 1 overlaps with only 1.

In the third test case, Robert visits for days $[1, 2]$, Mrs. Hood visits for days $[4, 5]$.

E. Rendez-vous de Marian et Robin

5 seconds, 256 megabytes

In the humble act of meeting, joy doth unfold like a flower in bloom.

Absence makes the heart grow fonder. Marian sold her last ware at the Market at the same time Robin finished training at the Major Oak. They couldn't wait to meet, so they both start without delay.

The travel network is represented as n vertices numbered from 1 to n and m edges. The i-th edge connects vertices u_i and v_i, and takes w_i seconds to travel (all w_i are even). Marian starts at vertex 1 (Market) and Robin starts at vertex n (Major Oak).

In addition, h of the n vertices each has a single horse available. Both Marian and Robin are capable riders, and could mount horses in no time (i.e. in 0 seconds). Travel times are halved when riding. Once mounted, a horse lasts the remainder of the travel. Meeting must take place on a vertex (i.e. not on an edge). Either could choose to wait on any vertex.

Output the earliest time Robin and Marian can meet. If vertices 1 and n are disconnected, output -1 as the meeting is cancelled.

Input

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

The first line of each test case consists of three integers n, m, h ($2 \le n \le 2 \cdot 10^5,\ 1 \le m \le 2 \cdot 10^5,\ 1 \le h \le n$) — the number of vertices, the number of edges and the number of vertices that have a single horse.

The second line of each test case contains h distinct integers $a_1, a_2, \ldots, a_h$ ($1 \le a_i \le n$) — the vertices that have a single horse available.

Then follow m lines of each test case, each with three integers u_i, v_i, w_i ($1 \le u_i, v_i \le n,\ 2 \le w_i \le 10^6$) — meaning that there is an edge between vertices u_i and v_i with travel cost w_i seconds without a horse.

There are no self loops or multiple edges. By good fortune, all w_i are even integers.

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

Output

For each test case, output a single integer, the earliest time Robin and Marian can meet. If it is impossible for them to meet, output -1.

input

```
6
2 1 1
1
1 2 10
3 1 2
2 3
1 2 10
3 3 1
2
1 2 4
1 3 10
2 3 6
4 3 2
2 3
1 2 10
2 3 18
3 4 16
3 2 1
2
1 2 4
1 3 16
7 7 1
3
1 5 2
2 6 12
1 2 12
6 4 8
7 3 4
6 3 4
7 6 4
```

output

```
5
-1
6
19
14
12
```

In the first test case, Marian rides from vertex 1 to vertex 2, Robin waits.

In the second test case, vertices 1 and 3 are not connected.

In the third test case, both Marian and Robin travel to vertex 2 to meet.

In the fourth test case, Marian travels to vertex 2 without a horse, mounts the horse at vertex 2 and rides to vertex 3 to meet Robin.

In the fifth test case, Marian travels to vertex 2 without a horse, mounts the horse at vertex 2 and rides back to vertex 1 and then to vertex 3. Robin waits.

F. Sheriff's Defense

2 seconds, 256 megabytes

"Why, master," quoth Little John, taking the bags and weighing them in his hand, "here is the chink of gold."

The folk hero Robin Hood has been troubling Sheriff of Nottingham greatly. Sheriff knows that Robin Hood is about to attack his camps and he wants to be prepared.

Sheriff of Nottingham built the camps with strategy in mind and thus there are exactly n camps numbered from 1 to n and $n-1$ trails, each connecting two camps. Any camp can be reached from any other camp. Each camp i has initially a_i gold.

As it is now, all camps would be destroyed by Robin. Sheriff can strengthen a camp by subtracting exactly c gold from **each of its neighboring camps** and use it to build better defenses for that camp. Strengthening a camp **doesn't change** its gold, only its neighbors' gold. A camp can have negative gold.

After Robin Hood's attack, all camps that have been strengthened survive the attack, all others are destroyed.

What's the maximum gold Sheriff can keep in his surviving camps after Robin Hood's attack if he strengthens his camps optimally?

Camp a is neighboring camp b if and only if there exists a trail connecting a and b. Only strengthened camps count towards the answer, as others are destroyed.

Input

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

Each test case begins with two integers n, c ($1 \le n \le 2 \cdot 10^5, 1 \le c \le 10^9$) — the number of camps and the gold taken from each neighboring camp for strengthening.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$) — the initial gold of each camp.

Then follow $n-1$ lines, each with integers u, v ($1 \le u, v \le n$, $u \ne v$) — meaning that there is a trail between u and v.

The sum of n over all test cases doesn't exceed $2 \cdot 10^5$.

It is guaranteed that any camp is reachable from any other camp.

Output

Output a single integer, the maximum gold Sheriff of Nottingham can keep in his surviving camps after Robin Hood's attack.

```
input
```

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

```
output
```

```
3
8
0
17
26
```

In the first test case, it is optimal to strengthen the second base. The final gold at each base is $[1, 3, 0]$.

In the second test case, it is optimal to strengthen all bases. The final gold at each base is $[2, 4, 2]$.

In the third test case, it is optimal to not strengthen any base.

G. Milky Days

2 seconds, 256 megabytes

What is done is done,
and the spoilt milk
cannot be helped.

Little John is as little as night is day — he was known to be a giant, at possibly 2.1 *metres tall. It has everything to do with his love for milk.*

His dairy diary has n entries, showing that he acquired a_i pints of fresh milk on day d_i. Milk declines in freshness with time and stays drinkable for a maximum of k days. In other words, fresh milk acquired on day d_i will be drinkable between days d_i and d_i+k-1 inclusive.

Every day, Little John drinks drinkable milk, up to a maximum of m pints. In other words, if there are less than m pints of milk, he will drink them all and not be satisfied; if there are at least m pints of milk, he will drink exactly m pints and be satisfied, and it's a *milk satisfaction day*.

Little John always drinks **the freshest** drinkable milk first.

Determine the number of *milk satisfaction days* for Little John.

Input

The first line of the input contains a single integer t ($1 \le t \le 10^4$), the number of test cases.

The first line of each test case consists of three integers n, m, k ($1 \le n$, m, $k \le 10^5$), the number of diary entries, the maximum pints needed for a milk satisfaction day, and the duration of milk's freshness.

Then follow n lines of each test case, each with two integers d_i and a_i ($1 \le d_i$, $a_i \le 10^6$), the day on which the milk was acquired and the number of pints acquired. They are sorted in increasing values of d_i, and all values of d_i are distinct.

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 a single integer, the number of *milk satisfaction days*.

```
input
```

```
6
1 1 3
1 5
2 3 3
1 5
2 7
4 5 2
1 9
2 6
4 9
5 6
5 2 4
4 7
5 3
7 1
11 2
12 1
4 1 3
5 10
9 4
14 8
15 3
5 5 5
8 9
10 7
16 10
21 5
28 9
```

```
output
```

```
3
3
4
5
10
6
```

In the first test case, 5 pints of milk are good for 3 days before spoiling.

In the second test case, the following will happen:

- On day 1, he will receive 5 pints of milk and drink 3 of them (leaving 2 pints from day 1);
- On day 2, he will receive 7 pints of milk and drink 3 of them (leaving 2 pints from day 1 and 4 pints from day 2);
- On day 3, he will drink 3 pints from day 2 (leaving 2 pints from day 1 and 1 pint from day 2);
- On day 4, the milk acquired on day 1 will spoil, and he will drink 1 pint from day 2 (no more milk is left).

H. Robin Hood Archery

3 seconds, 256 megabytes

At such times archery
was always the main
sport of the day, for the
Nottinghamshire
yeomen were the best
hand at the longbow in
all merry England, but
this year the Sheriff
hesitated...

Sheriff of Nottingham has organized a tournament in archery. It's the final round and Robin Hood is playing against Sheriff!

There are n targets in a row numbered from 1 to n. When a player shoots target i, their score increases by a_i and the target i is destroyed. The game consists of turns and players alternate between whose turn it is. Robin Hood always starts the game, then Sheriff and so on. The game continues until all targets are destroyed. Both players start with score 0.

At the end of the game, the player with most score wins and the other player loses. If both players have the same score, it's a tie and no one wins or loses. In each turn, the player can shoot any target that wasn't shot before. Both play optimally to get the most score possible.

Sheriff of Nottingham has a suspicion that he might lose the game! This cannot happen, you must help Sheriff. Sheriff will pose q queries, each specifying l and r. This means that the game would be played only with targets $l, l+1, \dots, r$, as others would be removed by Sheriff before the game starts.

For each query l, r, determine whether the Sheriff can **not lose** the game when only considering the targets $l, l+1, \dots, r$.

Input

The first line of input contains one integer t ($1 \le t \le 10^4$) — the number of test cases.

The first line of each test case contains two integers n, q ($1 \le n, q \le 2 \cdot 10^5$) — the number of targets and the queries Sheriff will pose.

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 10^6$) — the points for hitting each target.

Then follow q lines, each with two integers l and r ($1 \le l \le r \le n$) — the range of the targets that is considered for each query.

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

Output

For each query, output "YES", if the Sheriff **does not lose the game** when only considering the targets $l, l+1, \dots, r$, and "NO" otherwise.

You can output the answer in any case (upper or lower). For example, the strings "yEs", "yes", "Yes", and "YES" will be recognized as positive responses.

input

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

output

```
NO
NO
YES
NO
NO
YES
```