Problem A. String Game

Time Limit 1000 ms
Mem Limit 262144 kB

Topicwise Problem Link

Dummy Problem Link

Problem B. Children Holiday

Time Limit 2000 ms

Mem Limit 262144 kB

Topicwise Problem Link

Dummy Problem Link

Problem C. Very Easy Task

Time Limit 1000 ms
Mem Limit 262144 kB

Topicwise Problem Link

Dummy Problem Link

Problem D. Hamburgers

Time Limit 2000 ms
Mem Limit 262144 kB

Topicwise Problem Link

Dummy Problem Link

Problem E. Cows in Stalls

Time Limit 1000 ms
Mem Limit 262144 kB

Topicwise Problem Link

Dummy Problem Link

Problem F. Factory Machines

Time Limit 1000 ms
Mem Limit 524288 kB

Topicwise Problem Link

A factory has n machines which can be used to make products. Your goal is to make a total of t products.

For each machine, you know the number of seconds it needs to make a single product. The machines can work simultaneously, and you can freely decide their schedule.

What is the shortest time needed to make t products?

Input

The first input line has two integers n and t: the number of machines and products.

The next line has n integers $k_1, k_2, \ldots, k_n$: the time needed to make a product using each machine.

Output

Print one integer: the minimum time needed to make t products.

Constraints

- $1 \leq n \leq 2 \cdot 10^5$
- $1 \leq t \leq 10^9$
- $1 \leq k_i \leq 10^9$

Example

Input	Output
3 7 3 2 5	8

Explanation: Machine 1 makes two products, machine 2 makes four products and machine 3 makes one product.

Problem G. Maximum Median

Time Limit 2000 ms
Mem Limit 262144 kB

Topicwise Problem Link

You are given an array a of n integers, where n is odd. You can make the following operation with it:

- Choose one of the elements of the array (for example a_i) and increase it by 1 (that is, replace it with $a_i + 1$).

You want to make the median of the array the largest possible using at most k operations.

The median of the odd-sized array is the middle element after the array is sorted in non-decreasing order. For example, the median of the array $[1, 5, 2, 3, 5]$ is 3.

Input

The first line contains two integers n and k ($1 \le n \le 2 \cdot 10^5$, n is odd, $1 \le k \le 10^9$) — the number of elements in the array and the largest number of operations you can make.

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

Output

Print a single integer — the maximum possible median after the operations.

Examples

Input	Output
3 2 1 3 5	5

Input	Output
5 5 1 2 1 1 1	3

Input	Output
7 7 4 1 2 4 3 4 4	5

Note

In the first example, you can increase the second element twice. Than array will be $[1, 5, 5]$ and it's median is 5.

In the second example, it is optimal to increase the second number and than increase third and fifth. This way the answer is 3.

In the third example, you can make four operations: increase first, fourth, sixth, seventh element. This way the array will be $[5, 1, 2, 5, 3, 5, 5]$ and the median will be 5.

Problem H. K-th Not Divisible by n

Time Limit 1000 ms
Mem Limit 262144 kB

Topicwise Problem Link

You are given two positive integers n and k. Print the k-th positive integer that is not divisible by n.

For example, if $n = 3$, and $k = 7$, then all numbers that are not divisible by 3 are: $1, 2, 4, 5, 7, 8, 10, 11, 13 \ldots$. The 7-th number among them is 10.

Input

The first line contains an integer t ($1 \le t \le 1000$) — the number of test cases in the input. Next, t test cases are given, one per line.

Each test case is two positive integers n ($2 \le n \le 10^9$) and k ($1 \le k \le 10^9$).

Output

For each test case print the k-th positive integer that is not divisible by n.

Examples

Input	Output
6 3 7 4 12 2 1000000000 7 97 1000000000 1000000000 2 1	10 15 1999999999 113 1000000001 1

Problem I. Wooden Toy Festival

Time Limit 3000 ms
Mem Limit 262144 kB

Topicwise Problem Link

In a small town, there is a workshop specializing in woodwork. Since the town is small, only **three** carvers work there.

Soon, a wooden toy festival is planned in the town. The workshop employees want to prepare for it.

They know that n people will come to the workshop with a request to make a wooden toy. People are different and may want different toys. For simplicity, let's denote the pattern of the toy that the i-th person wants as a_i ($1 \le a_i \le 10^9$).

Each of the carvers can choose an integer pattern x ($1 \le x \le 10^9$) in advance, **different carvers can choose different patterns**. x is the integer. During the preparation for the festival, the carvers will perfectly work out the technique of making the toy of the chosen pattern, which will allow them to cut it out of wood instantly. To make a toy of pattern y for a carver who has chosen pattern x, it will take $|x - y|$ time, because the more the toy resembles the one he can make instantly, the faster the carver will cope with the work.

On the day of the festival, when the next person comes to the workshop with a request to make a wooden toy, the carvers can choose who will take on the job. At the same time, the carvers are very skilled people and can work on orders for different people **simultaneously**.

Since people don't like to wait, the carvers want to choose patterns for preparation in such a way that the **maximum** waiting time over all people is as **small** as possible.

Output the *best* maximum waiting time that the carvers can achieve.

Input

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

Then follow the descriptions of the test cases.

The first line of a test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the number of people who will come to the workshop.

The second line of a test case contains n integers $a_1, a_2, a_3, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the patterns of toys.

The sum of all n values over all test cases does not exceed $2 \cdot 10^5$.

Output

Output t numbers, each of which is the answer to the corresponding test case — the *best* maximum waiting time that the carvers can achieve.

Examples

Input	Output
5 6 1 7 7 9 9 9 6 5 4 2 1 30 60 9 14 19 37 59 1 4 4 98 73 1 2 6 3 10 1 17 15 11	0 2 13 0 1

Note

In the first example, the carvers can choose patterns 1, 7, 9 for preparation.

In the second example, the carvers can choose patterns 3, 30, 60 for preparation.

In the third example, the carvers can choose patterns 14, 50, 85 for preparation.

Problem J. Building an Aquarium

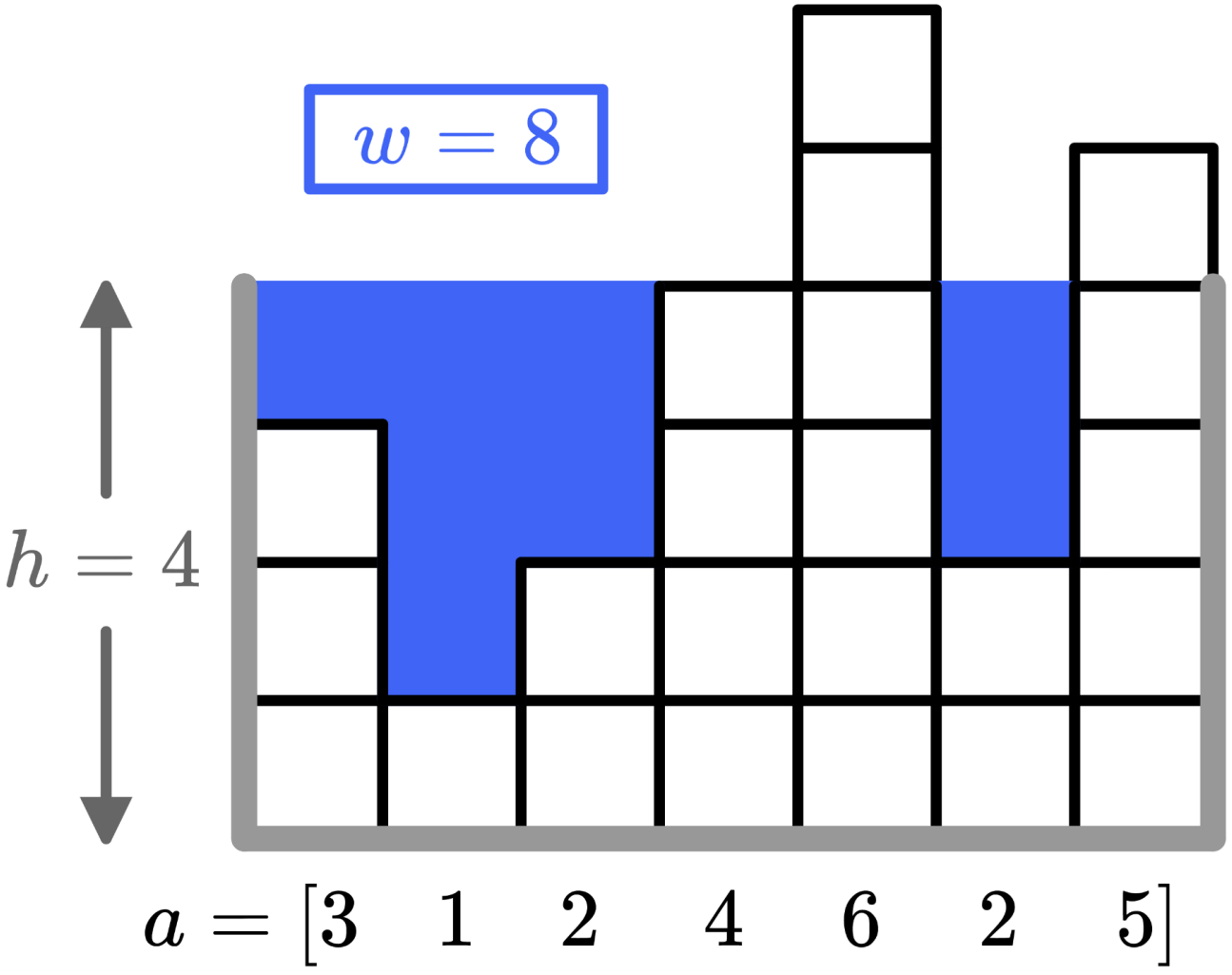
Time Limit 2000 ms
Mem Limit 262144 kB

Topicwise Problem Link

You love fish, that's why you have decided to build an aquarium. You have a piece of coral made of n columns, the i-th of which is a_i units tall. Afterwards, you will build a tank around the coral as follows:

- Pick an integer $h \geq 1$ — the *height* of the tank. Build walls of height h on either side of the tank.
- Then, fill the tank up with water so that the height of each column is h, unless the coral is taller than h; then no water should be added to this column.

For example, with $a = [3, 1, 2, 4, 6, 2, 5]$ and a height of $h = 4$, you will end up using a total of $w = 8$ units of water, as shown.



You can use at most x units of water to fill up the tank, but you want to build the biggest tank possible. What is the largest value of h you can select?

Input

The first line 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 positive integers n and x ($1 \le n \le 2 \cdot 10^5$; $1 \le x \le 10^9$) — the number of columns of the coral and the maximum amount of water you can use.

The second line of each test case contains n space-separated integers a_i ($1 \le a_i \le 10^9$) — the heights of the coral.

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

Output

For each test case, output a single positive integer h ($h \ge 1$) — the maximum height the tank can have, so you need at most x units of water to fill up the tank.

We have a proof that under these constraints, such a value of h always exists.

Examples

Input	Output
5 7 9 3 1 2 4 6 2 5 3 10 1 1 1 4 1 1 4 3 4 6 1984 2 6 5 9 1 8 1 1000000000 1	4 4 2 335 1000000001

Note

The first test case is pictured in the statement. With $h = 4$ we need 8 units of water, but if h is increased to 5 we need 13 units of water, which is more than $x = 9$. So $h = 4$ is optimal.

In the second test case, we can pick $h = 4$ and add 3 units to each column, using a total of 9 units of water. It can be shown that this is optimal.

In the third test case, we can pick $h = 2$ and use all of our water, so it is optimal.

Problem K. Save More Mice

Time Limit 4000 ms
Mem Limit 262144 kB

Topicwise Problem Link

There are one cat, k mice, and one hole on a coordinate line. The cat is located at the point 0, the hole is located at the point n. All mice are located between the cat and the hole: the i-th mouse is located at the point x_i ($0 < x_i < n$). At each point, many mice can be located.

In one second, the following happens. First, **exactly one** mouse moves to the right by 1. If the mouse reaches the hole, it hides (i.e. the mouse will not any more move to any point and will not be eaten by the cat). Then (**after that** the mouse has finished its move) the cat moves to the right by 1. If at the new cat's position, some mice are located, the cat eats them (they will not be able to move after that). The actions are performed until any mouse hasn't been hidden or isn't eaten.

In other words, the first move is made by a mouse. If the mouse has reached the hole, it's saved. Then the cat makes a move. The cat eats the mice located at the pointed the cat has reached (if the cat has reached the hole, it eats nobody).

Each second, you can select a mouse that will make a move. What is the maximum number of mice that can reach the hole without being eaten?

Input

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

Each test case consists of two lines. The first line contains two integers n and k ($2 \le n \le 10^9$, $1 \le k \le 4 \cdot 10^5$). The second line contains k integers $x_1, x_2, \ldots x_k$ ($1 \le x_i < n$) — the initial coordinates of the mice.

It is guaranteed that the sum of all k given in the input doesn't exceed $4 \cdot 10^5$.

Output

For each test case output on a separate line an integer m ($m \geq 0$) — the maximum number of mice that can reach the hole without being eaten.

Examples

Input	Output
3 10 6 8 7 5 4 9 4 2 8 1 1 1 1 1 1 1 1 12 11 1 2 3 4 5 6 7 8 9 10 11	3 1 4

Problem L. Final Boss

Time Limit 2000 ms
Mem Limit 262144 kB

Topicwise Problem Link

You are facing the final boss in your favorite video game. The boss enemy has h health. Your character has n attacks. The i'th attack deals a_i damage to the boss but has a cooldown of c_i turns, meaning the next time you can use this attack is turn $x + c_i$ if your current turn is x. Each turn, you can use all attacks that are not currently on cooldown, **all at once**. If all attacks are on cooldown, you do nothing for the turn and skip to the next turn.

Initially, all attacks are not on cooldown. How many turns will you take to beat the boss? The boss is beaten when its health is 0 or less.

Input

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

The first line of each test case contains two integers h and n ($1 \le h, n \le 2 \cdot 10^5$) – the health of the boss and the number of attacks you have.

The following line of each test case contains n integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 2 \cdot 10^5$) – the damage of your attacks.

The following line of each test case contains n integers $c_1, c_2, ..., c_n$ ($1 \le c_i \le 2 \cdot 10^5$) – the cooldown of your attacks.

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

Output

For each test case, output an integer, the minimum number of turns required to beat the boss.

Examples

Input	Output
8 3 2 2 1 2 1 5 2 2 1 2 1 50 3 5 6 7 5 6 7 50 3 2 2 2 3 3 3 90000 2 200000 200000 1 1 100000 1 1 200000 6 7 3 2 3 2 3 1 2 6 5 9 5 10 7 7 21 6 1 1 1 1 1 1 5 5 8 10 7 6	1 3 15 25 1 19999800001 1 21

Note

For the first test case, you can use attacks 1 and 2 on the first turn, dealing 3 damage in total, and slaying the boss.

For the second case, you can beat the boss in 3 turns by using the following attacks:

Turn 1: Use attacks 1 and 2, dealing 3 damage to the boss. The boss now has 2 health left.

Turn 2: Use attack 2, dealing 1 damage to the boss. The boss now has 1 health left.

Turn 3: Use attack 1, dealing 2 damage to the boss. The boss now has -1 health left. Since its health is less than or equal to 0, you beat the boss.

For the sixth test case: remember to use 64-bit integers as the answer can get large.

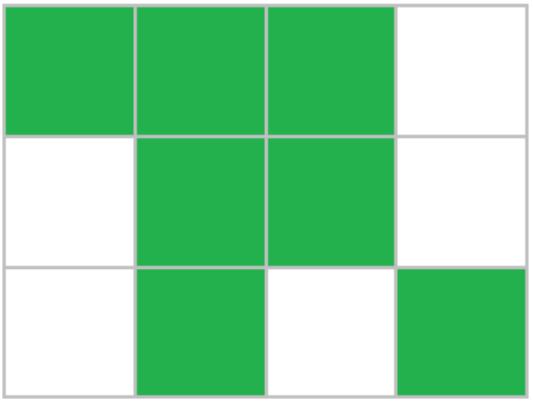
Problem M. Place of the Olympiad

Time Limit 1000 ms
Mem Limit 262144 kB

Topicwise Problem Link

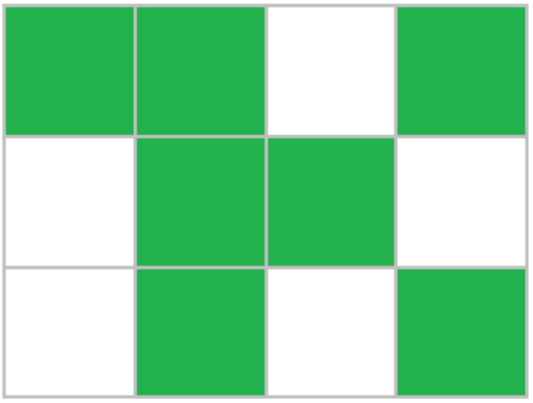
For the final of the first Olympiad by IT Campus "NEIMARK", a rectangular venue was prepared. You may assume that the venue is divided into n rows, each containing m spots for participants' desks. A total of k participants have registered for the final, and each participant will sit at an individual desk. Now, the organizing committee must choose the locations for the desks in the venue.

Each desk occupies one of the m spots in a row. Moreover, if several desks occupy consecutive spots in the same row, we call such a group of desks a *bench*, and the number of desks in the group is the bench's length. For example, seating 7 participants on a 3×4 venue (with $n = 3, m = 4$) can be arranged as follows:



In the figure above, the first row has one bench of length 3, the second row has one bench of length 2, and the third row has two benches of length 1.

The organizing committee wants to choose the locations so that the length of the longest bench is as small as possible. In particular, the same 7 desks can be arranged in a more optimal way, so that the lengths of all benches do not exceed 2:



Given the integers n, m, and k, determine the minimum possible length of the longest bench.

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.

A single line of each test case contains three positive integers — n, m, k ($1 \leq n, m, k \leq 10^9$, $k \leq n \cdot m$).

Output

For each test case, output a single number — the minimum possible length of the longest bench.

Examples

Input	Output
5 3 4 7 5 5 5 1 13 2 2 4 7 1 5 4	2 1 1 4 2

Problem N. Min Max MEX

Time Limit 2000 ms
Mem Limit 262144 kB

Topicwise Problem Link

You are given an array a of length n and a number k.

A subarray is defined as a sequence of one or more consecutive elements of the array. You need to split the array a into k non-overlapping subarrays $b_1, b_2, \ldots, b_k$ such that the union of these subarrays equals the entire array. Additionally, you need to maximize the value of x, which is equal to the minimum $\text{MEX}(b_i)$, for $i \in [1..k]$.

$\text{MEX}(v)$ denotes the smallest non-negative integer that is not present in the array v.

Input

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

The first line of each test case contains two integers n, k $(1 \leq k \leq n \leq 2 \cdot 10^5)$ — the length of the array and the number of segments to split the array into.

The second line of each test case contains n integers a_i $(0 \leq a_i \leq 10^9)$ — the elements of the array.

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

Output

For each query, output a single number — the maximum value x such that there exists a partition of the array a into k subarrays where the minimum MEX equals x.

Examples

Input	Output
7 1 1 0 5 1 0 1 3 2 4 6 2 2 1 0 0 1 2 5 5 0 0 0 0 0 5 2 2 3 4 5 6 6 2 0 0 1 1 2 2 4 4 1 0 0 0	1 5 3 1 0 1 0

Problem O. Quests

Time Limit 3000 ms
Mem Limit 262144 kB

Topicwise Problem Link

There are n quests. If you complete the i-th quest, you will gain a_i coins. You can only complete at most one quest per day. However, once you complete a quest, you cannot do the same quest again for k days. (For example, if $k = 2$ and you do quest 1 on day 1, then you cannot do it on day 2 or 3, but you can do it again on day 4.)

You are given two integers c and d. Find the maximum value of k such that you can gain at least c coins over d days. If no such k exists, output `Impossible`. If k can be arbitrarily large, output `Infinity`.

Input

The input consists of multiple test cases. The first line contains an 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 three integers n, c, d ($2 \le n \le 2 \cdot 10^5$; $1 \le c \le 10^{16}$; $1 \le d \le 2 \cdot 10^5$) — the number of quests, the number of coins you need, and the number of days.

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 rewards for the quests.

The sum of n over all test cases does not exceed $2 \cdot 10^5$, and the sum of d over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output one of the following.

- If no such k exists, output `Impossible`.
- If k can be arbitrarily large, output `Infinity`.
- Otherwise, output a single integer — the maximum value of k such that you can gain at least c coins over d days.

Please note, the checker is **case-sensitive**, and you should output strings exactly as they are given.

Examples

Input	Output
6 2 5 4 1 2 2 20 10 100 10 3 100 3 7 2 6 4 20 3 4 5 6 7 4 100000000000 2022 8217734 927368 26389746 627896974 2 20 4 5 1	2 Infinity Impossible 1 12 0

Note

In the first test case, one way to earn 5 coins over 4 days with $k = 2$ is as follows:

- Day 1: do quest 2, and earn 2 coins.
- Day 2: do quest 1, and earn 1 coin.
- Day 3: do nothing.
- Day 4: do quest 2, and earn 2 coins.

In total, we earned $2 + 1 + 2 = 5$ coins.

In the second test case, we can make over 20 coins on the first day itself by doing the first quest to earn 100 coins, so the value of k can be arbitrarily large, since we never need to do another quest.

In the third test case, no matter what we do, we can't earn 100 coins over 3 days.

Problem P. Ice Cream Balls

Time Limit 1000 ms
Mem Limit 262144 kB

Topicwise Problem Link

Tema decided to improve his ice cream making skills. He has already learned how to make ice cream in a cone **using exactly two** balls.

Before his ice cream obsession, Tema was interested in mathematics. Therefore, he is curious about the **minimum** number of balls he needs to have in order to make **exactly** n **different** types of ice cream.

There are plenty possible ice cream flavours: $1, 2, 3, \ldots$. Tema can make two-balls ice cream with any flavours (probably the same).

Two ice creams are considered different if their sets of ball flavours are different. For example, $\{1, 2\} = \{2, 1\}$, but $\{1, 1\} \neq \{1, 2\}$.

For example, having the following ice cream balls: $\{1, 1, 2\}$, Tema can make only two types of ice cream: $\{1, 1\}$ and $\{1, 2\}$.

Note, that Tema do not need to make all the ice cream cones at the same time. This means that he making ice cream cones independently. Also in order to make a following cone $\{x, x\}$ for some x, Tema needs at least 2 balls of type x.

Help Tema answer this question. It can be shown that answer always exist.

Input

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

The first line of each test case contains a single integer n ($1 \leq n \leq 10^{18}$) — the number of ice cream types that Tema wants to make.

Output

For each test case, output a single integer — the minimum number of balls Tema needs to buy.

Examples

Input	Output
5 1 3 6 179 1000000000000000000	2 3 4 27 2648956421

Note

In the first sample, it is enough to have following balls types: $\{1, 1\}$. **Note, that set** $\{1\}$ **if not enough since we need at least** 2 **balls of a type** 1 **in order to make such cone** $\{1, 1\}$.

In the second sample, it is not possible to make it with 2 balls, but it can be done with these balls: $\{1, 2, 3\}$.

In the third sample, $\{1, 2, 3, 4\}$ is optimal answer, so we can get following ice cream cones: $\{1, 2\}$, $\{1, 3\}$, $\{1, 4\}$, $\{2, 3\}$, $\{2, 4\}$, $\{3, 4\}$.

Problem Q. Klee's SUPER DUPER LARGE Array!!!

Time Limit 2000 ms
Mem Limit 262144 kB

Topicwise Problem Link

Klee has an array a of length n containing integers $[k, k+1, ..., k+n-1]$ in that order. Klee wants to choose an index i ($1 \leq i \leq n$) such that $x = |a_1 + a_2 + \cdots + a_i - a_{i+1} - \cdots - a_n|$ is minimized. Note that for an arbitrary integer z, $|z|$ represents the absolute value of z.

Output the minimum possible value of x.

Input

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

Each test case contains two integers n and k ($2 \leq n, k \leq 10^9$) — the length of the array and the starting element of the array.

Output

For each test case, output the minimum value of x on a new line.

Examples

Input	Output
4 2 2 7 2 5 3 1000000000 1000000000	1 5 1 347369930

Note

In the first sample, $a = [2, 3]$. When $i = 1$ is chosen, $x = |2 - 3| = 1$. It can be shown this is the minimum possible value of x.

In the third sample, $a = [3, 4, 5, 6, 7]$. When $i = 3$ is chosen, $x = |3 + 4 + 5 - 6 - 7| = 1$. It can be shown this is the minimum possible value of x.