

Problem A. Prime Magic

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

Walk Alone has a sequence $a_1, a_2, \dots, a_n$, and he can use a magic on it:

- Choose an odd prime number p and an interval $[l, r] \subseteq [1, n]$ satisfying $r - l + 1 = p$, and then add 1 or -1 to every a_i in this interval. That is, choose $x \in \{1, -1\}$, then $\forall i \in [l, r], a_i \leftarrow a_i + x$.

Remind that odd prime is numbers that are odd greater than 1 and can only be divided by 1 and itself.

Walk Alone likes growing things, so he wants to use this magic to make this sequence non-decreasing. Besides, Walk Alone doesn't like negative numbers, so during the whole operations any number cannot be negative.

He wants to accomplish this idea with minimal number of magic usage. Although Walk Alone has this magic, he doesn't know in which order to use it, so he asks you for help.

Input

There are T test cases you need to answer.

First line of the whole input contains a single integer T ($1 \leq T \leq 100$) denoting the number of test cases.

For each test case, the first line contains one integer n ($10 \leq n \leq 2 \times 10^3$), denoting the length of the sequence.

The following line contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^5$), denoting the sequence.

It's guaranteed that in all test cases, $\sum n \leq 2 \times 10^4$.

Output

For each test case, output a single integer in a line denoting the minimum number of magic usage to accomplish the task.

Example

standard input	standard output
4	12
10	96
1 10 8 2 3 4 1 2 1 8	0
12	4
29 31 2 1 34 29 31 24 18 2 9 10	
10	
1 2 3 4 5 6 7 8 9 10	
10	
2 2 1 1 2 2 1 1 2 2	

Problem B. Mode

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 512 megabytes

Walk Alone likes to go with the flow, so he designs a function on digits about mode.

Let $f(x)$ be the maximal occurrence among the digits in decimal expression of the number x . For example, $f(133) = 2$ since digit 3 occurs twice, while $f(213) = f(0) = 1$ since every digit appears exactly once in both numbers. Walk Alone gives you a task to calculate the range sum of function f , i.e. $\sum_{i=l}^r f(i)$.

Input

The first line contains one integer T ($1 \leq T \leq 10^3$) — the number of test cases. Next T test cases follow.

The first and only line of each test cases contains two integers l, r ($0 \leq l \leq r \leq 10^{18}$) — the range of the calculation.

Output

For each test case, print a single integer in a line denoting the sum.

Example

standard input	standard output
3	21
1 20	233
5 199	2553052375
0 1000000000	

Problem C. Darkness I

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

Princess Celestia, is an Alicorn pony, the former co-ruler of Equestria alongside her younger sister Princess Luna. Princess Celestia raises the sun and governs Equestria, while Princess Luna raises the moon and oversees the realm of dreams.

In early days, Luna felt overshadowed and ignored by the ponies of Equestria, who slept through her beautiful night sky. These negative emotions festered and eventually led her to be consumed by the darkness and power of the mythical creature known as Nightmare Moon.

The darkness was spreading, and could be transformed into the following model:

In an $n \times m$ white 2D grid, there are several black grid points.

Every second, all white points that are 4-neighbors of at least two black points will turn black.

What is the minimum number of black points needed to turn the entire grid black in the end?



Input

The first line contains two integers, n and m ($1 \leq n, m \leq 10^5$), representing the size of the grid.

Output

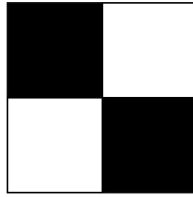
Output a single integer representing the minimum number of black points needed.

Example

standard input	standard output
2 2	2

Note

One of the minimal placement schemes of the sample is shown below.



Problem D. Darkness II

Input file: standard input
Output file: standard output
Time limit: 6 seconds
Memory limit: 1024 megabytes

Princess Celestia and Princess Luna argued, and the spreading darkness could be transformed into the following model:

In an infinitely large white 2D grid, there are initially n black grid points.

Every second, all white points that are 4-neighbors of at least two black points will turn black.

What is the total number of black points in the end?



Time to put our differences behind us. We were meant to rule together, little sister. Will you accept my friendship?

— Celestia offering Luna redemption

Input

The first line contains an integer n ($1 \leq n \leq 10^5$), representing the total number of grid points.

The next n lines each contain two integers a and b ($-10^9 \leq a, b \leq 10^9$), representing the positions of black grid points. The black points are guaranteed to be distinct from each other.

Output

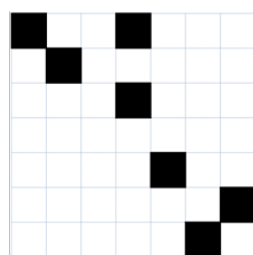
Output a single integer representing the total number of black points in the end.

Example

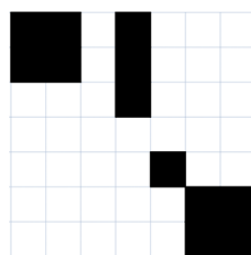
standard input	standard output
7 1 1 1 4 2 2 3 4 5 5 6 7 7 6	21

Note

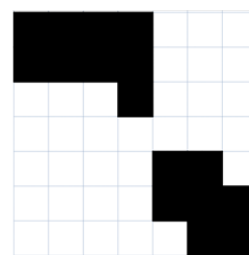
The change of the grid over time is shown below. After Time 6 no more grid will turn black, so the answer is 21.



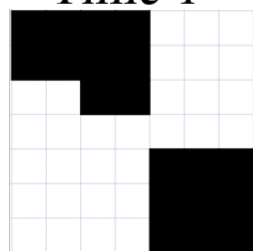
Time 1



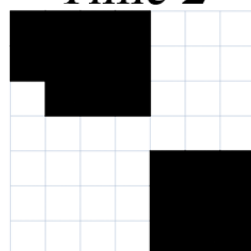
Time 2



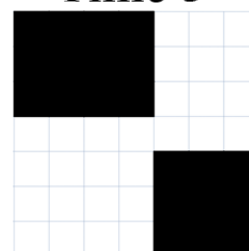
Time 3



Time 4



Time 5



Time 6

Problem E. Inverse Counting Path

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

Walk Alone is an expert in dynamic programming, but he gets bored with traditional dynamic programming problem like counting paths on a 2-dimension grid, so he wants to do it in reverse. The problem he raised is as follows:

On a 2-dimension grid of size $n \times n$, originally you are on grid $(1, 1)$. The grid consists of 0 and 1, and you can only walk on the grid with number 1 in it. You can only go down or right, i.e. you can only increase your x or y by one. Also you cannot walk outside the grid.

Given the number x of ways to walk from $(1, 1)$ to (n, n) , you need to construct a grid of $n \times n$ so that the ways to walk is exactly x . However, since Walk Alone's brain is too small to memorize such a big grid, **you need to guarantee that the size of the grid n is equal to or smaller than 30.**

Input

The only line of the input contains one integer x ($1 \leq x \leq 10^9$), denoting the ways to walk.

Output

The first line of the output contains the size of the grid n . Remind that you need to guarantee $1 \leq n \leq 30$.

The following n lines each contains n integers $a_{i,j} \in \{0, 1\}$ denoting the grid, where 0 denotes you cannot walk on the grid while 1 is on the contrast.

Examples

standard input	standard output
3	3 1 1 0 1 1 0 1 1 1
10	4 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1

Problem F. Inverse Manacher

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 512 megabytes

Walk Alone is an expert in strings, but he gets bored with traditional algorithms like Manacher, so he wants to do it in reverse. The problem he raised is as follows:

There's a hidden string S only consisting character 'a' and 'b'. You are given n —the length of the string. Then the string S is transformed as follows:

- First, add two character $\&|$ to the beginning of the string.
- Second, for every character in the original string, add a character $|$ to the right of it.

For example, string $abaaa$ is transformed into $\&|a|b|a|b|a|$, and string $aaaaa$ is transformed into $\&|a|a|a|a|a|$. After this operation, the length of the string S has become $2n + 2$. Let the transformed string be T .

He then will give you a sequence $\{a_n\}$ of length $2n + 2$, the i -th of which denotes the maximal length of palindrome centered at the i -th character. Formally, $a_i = j$ where $j = \max\{x | x \in \mathbb{N}^*, \forall y \leq x, T_{i-y+1} = T_{i+y-1}\}$. For example, the original string $abaaa$ can generate the final sequence $\{1, 1, 2, 1, 4, 1, 2, 3, 4, 3, 2, 1\}$. Note that every character in the string is compared through ASCII code, so character $\&$ cannot be in pair with any other character.

Your work is to restore the string S using sequence $\{a_n\}$.

Input

The first line of the input contains a single integer n ($1 \leq n \leq 10^6$) denoting the length of the string.

The following line contains $2n + 2$ integers, the i -th of which denotes a_i .

Output

The only line of the output contains a string of length n only consisting of character 'a' and 'b', denoting your answer. Note that you don't need to output the added character $\&$ and $|$.

It's guaranteed that there will be at least one string that can generate sequence $\{a_n\}$. If there's more than one string, output any of them.

Example

standard input	standard output
5 1 1 2 1 4 1 2 3 4 3 2 1	abaaa

Problem G. Guess the Polynomial

Input file: `standard input`
Output file: `standard output`
Time limit: 5 seconds
Memory limit: 256 megabytes

This is an interactive problem.

There is a hidden polynomial $f(x) = \sum_{i=1}^n a_i \cdot x^{p_i}$ where $0 \leq n \leq 10^3$, $1 \leq a_i \leq 10^5$, $0 \leq p_i \leq 8 \cdot 10^6$, and $p_1, p_2, \dots, p_n$ are pairwise distinct.

You can query in the format “? x ”, and the interactor will respond you $f(x) \bmod 998\,244\,353$.

You need to find the function by no more than $2 \cdot 10^4$ queries.

Interaction Protocol

You can make no more than $2 \cdot 10^4$ queries to guess the polynomial. To make a query, output “? x ” ($0 \leq x < 998\,244\,353$) on a separate line, then you should read the response from standard input.

To give your answer, firstly print “! n ” on a separate line, where n indicates the number of monomials. For each monomial $a_i \cdot x^{p_i}$, print “ $p_i \ a_i$ ” on a separate line. You can print the n monomials in an arbitrary order. The output of the answer is **not** counted towards the limit of $2 \cdot 10^4$ queries.

After that, your program should terminate.

After printing a query, do not forget to output end of line and flush the output. To do this, use `fflush(stdout)` or `cout.flush()` in C++, `System.out.flush()` in Java, `flush(output)` in Pascal, or `stdout.flush()` in Python.

It is guaranteed that the polynomial is fixed in advance.

Example

standard input	standard output
	? 0
3	
	? 1
4	
	? 2
7	
	! 2
	0 3
	2 1

Note

The polynomial in the example is $f(x) = x^2 + 3$.

Problem H. Binary Crazyiness

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 512 megabytes

Walk Alone has a graph of n nodes and m bi-directional edges. He defines a function of $f(u, v) = (\deg_u \oplus \deg_v)(\deg_u | \deg_v)(\deg_u \& \deg_v)$, where $\deg_u$ denotes the number of edges that associated with node u , and $\oplus, |, \&$ denotes bitwise xor, bitwise or and bitwise and. He wants you to calculate $\left(\sum_{i=1}^n \sum_{j=i}^n f(i, j)\right) \bmod 998\,244\,353$.

Input

The first line of the input contains two integers n, m ($1 \leq n \leq 10^6, 0 \leq m \leq 10^6$) denoting the number of nodes and edges.

Each of the following m lines contains two integers u, v ($1 \leq u, v \leq n$) representing a bi-directional edge (u, v) .

The graph might have duplicated edges and self-loops. Remind that a self-loop on node x will contribute 2 to $\deg_x$.

Output

Output a single integer denoting the answer.

Examples

standard input	standard output
6 6 1 3 2 3 1 4 2 5 3 6 4 6	30
6 4 1 2 3 5 2 4 3 6	0

Problem I. Step

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 512 megabytes

You are given n rings, and the length of the i -th ring is p_i ($1 \leq i \leq n$). For the i -th ring, the next position of p_i is 1. At the very beginning, there is a pony at position 1 of each ring, and the ponies move faster and faster every day.

Specifically, a pony moves 1 step on the first day, and 2 steps on the second day, and so on. Formally, the pony will move k steps on the k -th day for $k \in \mathbb{N}^*$.

It is clear that on a certain day, all the ponies will arrive at position 1. Now Miku wants to know the earliest day m (except day 0) when all the n ponies will arrive at position 1.

Input

The first line of the input contains a single positive integer n ($1 \leq n \leq 10^5$) – the number of rings.

The second line of the input contains n positive integers p_i ($1 \leq p_i \leq 10^7$) – the length of each rings.

It is guaranteed that $\text{lcm}(p_1, p_2, \dots, p_n) \leq 10^{18}$. Remind that $\text{lcm}(p_1, p_2, \dots, p_n)$ stands for the least common multiple (LCM) of $\{p_1, p_2, \dots, p_n\}$.

Output

Output a single positive integer denoting the earliest day that all the n ponies will arrive at position 1.

Examples

standard input	standard output
3 3 6 9	8
4 114 514 1919 810	26138699

Problem J. Expansion

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

Applejack is a female Earth pony. She lives and works at Sweet Apple Acres with her grandmother Granny Smith, her older brother Big McIntosh, her younger sister Apple Bloom, and her dog Winona. She represents the element of honesty.

Now Applejack is going to cultivate a row of wasteland, and it could be transformed into the following model:

There are n cells in a row, each containing an integer a_i . Applejack start by cultivating cell number 1 and have 0 resources **just before the end of time unit 1**.

Later, assuming Applejack have cultivated cells $[1, x]$, at the beginning of each unit time, she can cultivate cell $x + 1$, and at the end of each unit time, she obtain resources equal to the sum of integers on all cells she have cultivated.

Applejack need to ensure that she always have non-negative resources(**even after cultivating all cells**). What is the minimum time needed to occupy all cells, or is it impossible?



Input

The first line contains an integer n ($2 \leq n \leq 10^5$) which represents the length of the row of cells.

The second line contains n integers a_i ($-10^8 \leq a_i \leq 10^8$), representing the coefficients on the cells.

Output

Output a single integer representing the minimum amount of time needed to occupy all cells, if a solution exists.

Otherwise, output -1 .

Examples

standard input	standard output
3 1 -3 4	4
3 1 -4 2	-1
6 1 -2 1 -4 5 -1	10

Note

In the first example, Applejack can stay cell 1 until time 3, then she can cultivate cell 2 and cell 3 at time 3 and time 4.

And at the end of time 1, time 2, time 3, time 4, she have 1, 2, 0, 2 resource correspondingly.

In the second example, after cultivating all cells, the resources will eventually be negative.

Problem K. Dice Game

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 512 megabytes

There are $n + 1$ people playing a game with dice. Every person has a dice of m faces, each of which is equal-probability. Don't spend time wondering how this kind of dice can exist—You can image a generator which can generate numbers among $[1, m]$ with equal-probability.

At the beginning, each person throws his own dice, and the person with the smallest number of points loses. If more than one person throws the smallest number of dice at the same time, those who have thrown the smallest number of dice continue to throw dice until finally one person loses.

Walk Alone doesn't like randomness. He uses magic to fix the points the first person throws every time. That is, the first person will always throw x points when the first person throws his dice, where x is pre-chosen by Walk Alone, while other n people still throw their dices randomly.

Now Walk Alone wants to know for every $x \in [1, m]$, the probability the first person will lose.

To be more precise, you need to output the probability modulo 998 244 353. It can be proven that the probability can be presented as a fraction $\frac{p}{q}$. You need to output $p \cdot q^{998\,244\,351} \bmod 998\,244\,353$.

Input

The only line of input contains two integers n, m ($1 \leq n \leq 10^5$, $2 \leq m \leq 10^5$) denoting the number of people and the number of faces a dice is.

Output

Output m integers in a line, the i -th number denoting the probability modulo 998 244 353 when $x = i$.

Example

standard input	standard output
3 5	1 577110017 873463809 982646785 0

Problem L. Game

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

Sweetie Drops, or Bon Bon, is a female Earth pony with a cutie mark of three candies. Lyra Heartstrings, is a female unicorn pony with a cutie mark of a lyre.

They always appear together in TV show as background characters. In The Last Problem, Lyra is shown in newspaper as having gotten married to Sweetie Drops.

The fans use LyraBon as the femslash ship between Lyra Heartstrings and Sweetie Drops.

Today they are playing a black-white game, the game progresses turn by turn, Bon Bon moves **first**.

Initially, there are $n + 1$ points connected by either white or black edge in a row. In one move a pony must choose **at least two points** and then do some removal.

More specific, Bon Bon can choose some **consecutive** points **without black edge** between any two of them, and then remove these points and all edge(s) connected to them. Be attention that black edge may be removed in the process.

Similarly, Lyra can choose consecutive points without **white** edge and then do removal in same way.

It is required that after either Bon Bon or Lyra's removal, the points **must not** be split into two separate blocks.

The pony who cannot make a move loses. If both ponys play optimally, could Bon Bon ultimately win the game?



Input

The first line contains an integer T ($1 \leq T \leq 10^6$), representing the number of test cases.

Each test case contains a binary string of length n ($1 \leq \sum n \leq 2 \times 10^6$), representing the colors of the edges, 0 represents a white edge, and 1 represents a black edge.

Output

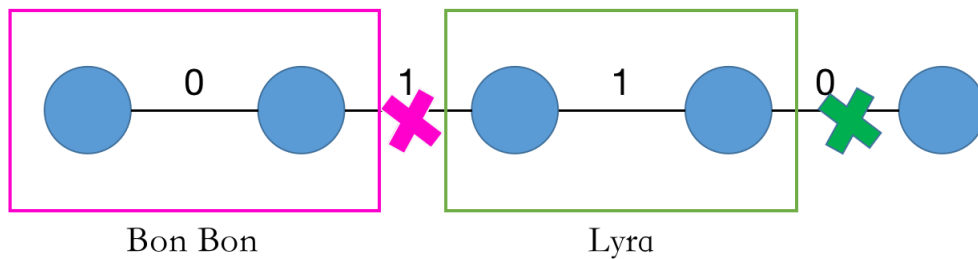
For each test case, output one line. If Bon Bon wins, output “YES”, otherwise output “NO” (without the quotes).

Example

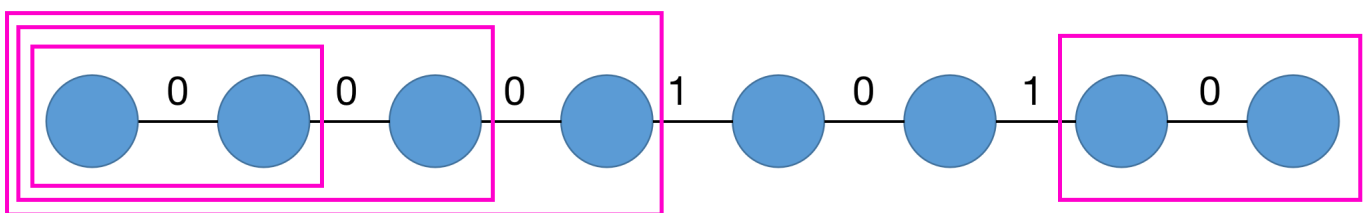
standard input	standard output
3 0110 0001010 1001100	NO YES YES

Note

In the first test case, Bon Bon could first choose the leftmost two points, then remove them, the edge between them and the edge marked with a purple cross in the figure below. In Lyra’s turn, she could remove similarly and leave only one point(the rightmost one). Thus Bon Bon could not move and lose.



In the second test case, Bon Bon has four choices in the first move. Note that the edge connect to the chosen points will be removed, too.



Problem M. Different Billing

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 512 megabytes

Once upon a time, there was an onsite programming contest, which had attracted many teams to participate in. However, different teams had a different billing standard. In general, there were three types of teams:

- Teams of Type A could participate in the contest for free.
- Teams of Type B had to pay 1000 dollars for the expenses for board and lodging.
- Teams of Type C had to pay 1000 dollars for the expenses for board and lodging and 1500 dollars for entry fee.

Finally the contest was held successfully with x teams in total participating in, and the host had earned y dollars. However, the details about the number of teams of each type had vanished. Now Walk Alone wants to restore the details using x and y .

Input

The only line of input contains two integers x and y ($0 \leq x \leq 10^6$, $0 \leq y \leq 10^9$), denoting the number of teams in total and the total amount of money the host had earned.

Output

If there's a way to construct the number three types of teams satisfying the condition above, output three integers A, B, C to denote the number of teams of type A, B and C. You need to guarantee that $0 \leq A, B, C$. If there are multiple ways, output any of them.

If there's no answer satisfying the condition above, output -1 instead.

Examples

standard input	standard output
800 1500000	200 0 600
0 0	0 0 0
500 100	-1

Note

For the first sample test case you can also output 50 500 250, which can also satisfy the condition above.

For the second sample test case, there's nobody who needs to bill participating in the contest, so no money can be earned by the host.

For the third sample test case, the host cannot earn such little money. So output -1 .