

Problem A. Square Sum

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 256 megabytes

You're given an integer m and a sequence of integers $z_1, \dots, z_n$.

For each z_i , calculate the number of integers x, y ($0 \leq x, y < m$) such that

$$x^2 + y^2 \equiv z_i \pmod{m}.$$

Input

First line of input contains two integers m ($1 \leq m \leq 10^9$) and n ($1 \leq n \leq 10^5$).

The second line contains n integers z_i ($0 \leq z_i < m$).

Output

For each z_i , output the number of pairs x, y ($0 \leq x, y < m$) such that $x^2 + y^2 \equiv z_i \pmod{m}$.

Examples

standard input	standard output
3 3 0 1 2	1 4 4
4 4 0 1 2 3	4 8 4 0
5 1 3	4

Problem B. Super Meat Bros

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

Super Meat Bros is a manga about two brothers Meatio and Meatigi who ultimately love meat.

Signature feature of this manga is that both brothers have their own story arcs that progress independently. Each brother will have a story of zero or more arcs, each arc containing at most n manga issues. For an arc that would last k issues, the mangaka knows a_k ways to make a story about Meatio and b_k ways to make a story about Meatigi.

The mangaka will make two stories, one about Meatio and the other about Meatigi. The story about Meatio is created in the following way: Until the mangaka is bored, they choose a number $k \leq n$ and append a new story arc of k issues to the story in one of a_k distinct ways. Correspondingly, for Meatigi story, the mangaka chooses k and appends a new story arc of k issues in one of b_k distinct ways.

After full stories of several arcs are prepared for both Meatio and Meatigi, they will be merged together in a way that preserves internal order of stories. That is, if issues x and y are related to the same brother and in his story x comes before y , it will also come before y in the merged story. Other than that merging can be arbitrary, in particular story arcs do **not** have to form a contiguous subsequence.

You're given $a_1, \dots, a_n$ and $b_1, \dots, b_n$, calculate the number of ways to create a full volume of m issues.

Input

First line of input contains two integers n ($1 \leq n \leq 300$) and m ($1 \leq m \leq 10^9$).

Second line contains n integers $a_1, \dots, a_n$ ($1 \leq a_i \leq 10^9$).

Third line contains n integers $b_1, \dots, b_n$ ($1 \leq b_i \leq 10^9$).

Output

Output a single integer which is the number of ways to create a full volume of m issues.

Since the answer might be very large, output it modulo $10^9 + 9$.

Examples

standard input	standard output
2 3 1 1 1 1	18
3 4 1 2 3 1 3 2	180

Note

Let's denote issues about Meatio with lowercase English letters and issues about Meatigi with uppercase English letters. We will also use same letters to denote issues belonging to the same story arc and assign letters to arcs in alphabetic order. In this notion, following combinations are possible in the first example:

abc, abb, aab, ABC, ABB, AAB, abA, aaA, ABa, AAa, aAB, AAA, Aab, Aaa, aAb, aAa, AaB, AaA.

Problem C. Testing Subjects Usually Die

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

Hello, and again, welcome to the Aperture
Science Enrichment Center.

You wake up in an unfamiliar room and a robotic voice greets you. You're in trouble.

The voice explains that you're being a subject of an ethically questionable series of experiments. The test you're undergoing right now goes as follows. An AI picked an integer number from 1 to n in such a way that the number k had a probability $\frac{p_k}{p_1 + \dots + p_n}$ to be picked. You need to guess the number.

If you guess it incorrectly, you'll be put to sleep, your memory will be erased and you will undergo the same test again. With a probability of c percent, the number chosen by the AI will be re-picked by the same procedure and with a probability of $100 - c$ percent, it will stay the same.

You have no idea how many times you already participated in this test and what numbers you have picked before, but you clearly want to spend as little time on it as possible. Therefore, you will choose a probability distribution $q_1, \dots, q_n$ and will say the number k with the probability q_k .

What is the minimum expected number of guesses you need to make before completing the test?

Input

First line of input contains two integers n and c ($2 \leq n \leq 10^5$, $0 \leq c \leq 100$).

Second line of input contains n integer numbers $p_1, \dots, p_n$ ($1 \leq p_i \leq 10^3$).

Output

Output a single floating-point number, which is the minimum possible expected number of guesses.

Your answer will be considered correct if its absolute or relative error doesn't exceed 10^{-6} .

Examples

standard input	standard output
4 100 25 25 25 25	4
2 0 1 4	1.8000000000

Note

For the purposes of this task, a probability distribution is a sequence of real numbers $q_1, \dots, q_n$ such that $0 \leq q_i$ and $q_1 + \dots + q_n = 1$.

Problem D. Triterminant

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

Let $b_1, b_2, \dots, b_n$ be a sequence of integers. A sequence of polynomials $A_1, A_2, \dots, A_n$ is defined as

$$A_k(x) = \det \begin{bmatrix} x & b_1 & 0 & \dots & 0 \\ 1 & x & b_2 & \dots & 0 \\ 0 & 1 & x & \cdot & \vdots \\ \vdots & \vdots & \cdot & \ddots & b_k \\ 0 & 0 & \dots & 1 & x \end{bmatrix}$$

We call $b_1, b_2, \dots, b_n$ good if for all k , all coefficients of A_k do not exceed 1 by the absolute value.

You're given a sequence $c_1, c_2, \dots, c_n$ such that $c_k \in \{-1, 1\}$. You can change any number c_k to $-c_k$.

What is the minimum numbers of the sequence elements you should change to get a good sequence?

Input

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

The first line of each test case contains a single integer n ($1 \leq n \leq 10^5$).

The second line contains n integers $c_1, c_2, \dots, c_n$ (c_k is either -1 or 1).

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

Output

For each test case, output the minimum number of $c_1, c_2, \dots, c_n$ elements that must be changed to obtain a good sequence.

If there is no valid way to obtain a good sequence from $c_1, c_2, \dots, c_n$, output a single integer -1 .

Example

standard input	standard output
3	2
4	0
1 1 1 1	2
2	
1 -1	
5	
-1 1 1 1 -1	

Note

$c = (1, -1, 1, -1)$ is a good sequence and can be obtained from $(1, 1, 1, 1)$ in 2 changes.

Problem E. Garbage Disposal

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

There are 10^9 types of garbage and 10^9 types of garbage bins in your country. You are only allowed to dispose garbage of type x into a garbage bin of type y if $\gcd(x, y) = 1$, where $\gcd(x, y)$ denotes the greatest common divisor (GCD) of integers x and y .

In your neighborhood, only garbage of type $L \leq x \leq R$ ever occurs, and there are only garbage bins of types $L \leq y \leq R$ available. To avoid overflowing the bins, you want to throw each piece into distinct bin. Given L and R , find a valid distribution or report that it does not exist.

Input

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

The first line of each test case contains two integers L and R ($1 \leq L \leq R \leq 10^9$).

It is guaranteed that the sum of $R - L + 1$ over all test cases does not exceed 10^5 .

Output

For each test case, if there is no valid distribution print -1 .

Otherwise, output $R - L + 1$ distinct integers $y_L, y_{L+1}, \dots, y_R$ ($L \leq y_i \leq R$), such that $\gcd(y_i, i) = 1$ for every i from L to R .

If there are multiple solutions, print any.

Example

standard input	standard output
3	2 1 4 5 3
1 5	11 10 13 12
10 13	-1
100 100	

Note

In the first test case, $\gcd(1, 1) = \gcd(2, 3) = \gcd(3, 4) = \gcd(4, 5) = \gcd(5, 2) = 1$.

In the second test case, $\gcd(10, 13) = \gcd(11, 10) = \gcd(12, 11) = \gcd(13, 12) = 1$.

In the third test case, the only possible assignment is $y_{100} = 100$, but $\gcd(100, 100) = 100 \neq 1$.

Problem F. Palindromic Polynomial

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

Author: Dmytro Fedoriaka

A *palindromic polynomial* is a non-zero polynomial whose coefficients read the same in both directions.

Alan had a palindromic polynomial A of a degree $d \leq 10^4$. He wrote down its values modulo $10^9 + 9$ in n distinct integer points. Then he lost the polynomial. Now he wants to restore it from the points. Help Alan find any palindromic polynomial of degree at most 10^4 which passes through all given points.

Formally, you are given a list of pairs $(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$, $0 \leq x_i, y_i < 10^9 + 9$. Your task is to find any polynomial $A(x) = a_d x^d + a_{d-1} x^{d-1} + \dots + a_1 x + a_0$, such that:

- $0 \leq d \leq 10^4$ and $a_d \neq 0$;
- $0 \leq a_i < 10^9 + 9$ for $i = 0 \dots d$;
- $P(x_i) \equiv y_i \pmod{10^9 + 9}$ for $i = 1 \dots n$;
- $a_i = a_{d-i}$ for $i = 0 \dots d$.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 100$). The description of the test cases follows.

The first line of each test case contains an integer n ($1 \leq n \leq 10^3$) — the number of points.

The second line of each test case contains n distinct integers $x_1, x_2, \dots, x_n$ ($0 \leq x_i < 10^9 + 9$).

The third line of each test case contains n integers $y_1, y_2, \dots, y_n$ ($0 \leq y_i < 10^9 + 9$).

It is guaranteed that the sum of n over all test cases does not exceed 10^3 .

Output

For each test case, print -1 if there is no polynomial that satisfies all the conditions. Otherwise, on the first line print d — the degree of the found polynomial ($0 \leq d \leq 10^4$), and on the next line print $d + 1$ integers $a_0, a_1, \dots, a_d$ ($0 \leq a_i < 10^9 + 9$, $a_d \neq 0$).

If there are multiple solutions, print any of them.

Example

standard input	standard output
8	1
2	2 2
0 1	3
2 4	2 3 3 2
3	2
0 1 2	1 2 1
2 10 36	8
4	1 2 3 4 5 4 3 2 1
0 1 2 3	3
1 4 9 16	1 666666672 666666672 1
5	3
0 1 2 3 4	1 666666672 666666672 1
1 25 961 14641 116281	-1
2	-1
2 500000005	
5 375000004	
2	
2 500000005	
5 375000004	
2	
2 500000005	
1 2	
3	
2 500000005 3	
5 375000004 10	

Note

The polynomial of degree d has exactly $d + 1$ coefficients, even though some of them may be zeros. The leading coefficient of a polynomial cannot be zero unless the polynomial is constant zero.

Hence, the following polynomials are palindromic:

- $A(x) = 2x^3 + 3x^2 + 3x + 2$ — coefficients are $[2, 3, 3, 2]$.
- $A(x) = 5x^4 + 10x^2 + 5$ — coefficients are $[5, 0, 10, 0, 5]$.
- $A(x) = x^4 + 1$ — coefficients are $[1, 0, 0, 0, 1]$.
- $A(x) = 1$ — coefficients are $[1]$.

The following polynomials are **not** palindromic:

- $A(x) = 2x^3 + 3x^2 + 3x + 1$ — coefficients are $[2, 3, 3, 1]$.
- $A(x) = 2x^4 + 3x^3 + 3x^2 + 2x$ — coefficients are $[2, 3, 3, 2, 0]$.
- $A(x) = x^5 + x$ — coefficients are $[1, 0, 0, 0, 1, 0]$.

As a special case, the polynomial $A(x) = 0$ does not satisfy condition $a_d \neq 0$ and will not be accepted as an answer.

Also note that you **do not** need to minimize the degree of the polynomial.

Problem G. Palindromic Differences

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

Author: Dmytro Fedoriaka

For an array $a = [a_1, a_2, \dots, a_n]$, $n \geq 2$, its *difference array* is defined as $[a_2 - a_1, a_3 - a_2, \dots, a_n - a_{n-1}]$.

The array $a = [a_1, a_2, \dots, a_n]$ is a *palindrome* if it doesn't change after being reversed.

A permutation of array a is an array which has the same elements as a , but possibly in a different order.

You are given an array a of length n . Find the number of distinct permutations of a whose difference array is a palindrome. Two arrays a and b of same length are distinct if and only if for some i , $a_i \neq b_i$.

As this number can be very large, print it modulo $10^9 + 9$.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 100$). The description of the test cases follows.

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

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($-10^9 \leq a_i \leq 10^9$).

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

Output

For each test case, print a single number on a separate line — the answer to the test case modulo $10^9 + 9$.

Example

standard input	standard output
5	2
3	1
2 3 1	0
4	24
1 1 1 1	645120
3	
1 2 4	
7	
0 200 0 200 50 100 150	
14	
-1 0 1 2 3 4 5 6 7 8 9 10 11 12	

Note

In the first test case, the array $[2, 3, 1]$ has six permutations: $[1, 2, 3]$, $[1, 3, 2]$, $[2, 1, 3]$, $[2, 3, 1]$, $[3, 1, 2]$, $[3, 2, 1]$. Their difference arrays are $[1, 1]$, $[2, -1]$, $[-1, 2]$, $[1, -2]$, $[-2, 1]$, $[-1, -1]$. Of them only two are palindromes: $[1, 1]$, $[-1, -1]$. So, the only two permutations with palindromic difference arrays are $[1, 2, 3]$ and $[3, 2, 1]$.

In the second test case, there is only one permutation $[1, 1, 1, 1]$. Its difference array $[0, 0, 0]$ is a palindrome.

In the third test case, none of permutations has a palindromic difference array.

Problem H. Graph Isomorphism

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

Two undirected graphs with n vertices G_1 and G_2 are called isomorphic if there is a permutation $p_1, p_2, \dots, p_n$, such that

$$(u, v) \text{ is an edge of } G_1 \iff (p_u, p_v) \text{ is an edge of } G_2$$

Given an undirected graph G , you should determine whether it is true that there are no more than n distinct graphs that are isomorphic to G .

Two undirected graphs with the same number of vertices are considered distinct if their sets of edges are distinct.

Input

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

The first line of each test case contains two positive integers n and m ($1 \leq n, m \leq 10^5$) — the number of vertices and the number of edges in the graph.

Following m lines contain a pair of integers u and v each ($1 \leq u, v \leq n$), meaning that there is an edge between u and v .

The graph does not contain loops or multiple edges. It is guaranteed that the sums of n and m over all test cases do not exceed 10^5 each.

Output

For each test case, output YES if there are at most n distinct graphs isomorphic to the given graph. Otherwise, output NO.

Example

standard input	standard output
3	YES
3 3	YES
1 2	NO
2 3	
3 1	
3 2	
1 2	
2 3	
5 5	
1 2	
2 3	
3 4	
4 5	
5 1	

Problem I. DAG Generation

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

To generate a directed acyclic graph (DAG), we start with an empty set A of the DAG vertices and the set $B = \{1, 2, \dots, n\}$ of candidate vertices.

Then, we add vertices to the DAG one by one in the following manner:

1. We pick a set $X \subseteq A$ and a vertex $u \in B$ uniformly at random;
2. We draw arcs from all vertices of X into u ;
3. We add u to A , and remove u from B .

In the end, we get a DAG on n vertices. We used this procedure twice and generated two DAGs on n vertices. What is the probability that they are distinct?

Two DAGs are considered distinct if their sets of directed edges are distinct.

Input

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

The first line of each test case contains a single integer n ($1 \leq n \leq 10^5$) — the number of vertices in the DAG.

Output

For each testcase, print the probability that the DAGs are distinct modulo $10^9 + 9$.

Formally, let $M = 10^9 + 9$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \pmod{M}$. In other words, output such an integer x that $0 \leq x < M$ and $x \cdot q \equiv p \pmod{M}$.

Example

standard input	standard output
4	0
1	375000004
2	117187502
3	778748905
100	

Note

For $n = 2$, the answer is $\frac{5}{8}$.

For $n = 3$, the answer is $\frac{121}{128}$.

Problem J. Persian Casino

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

Prince of Persia walks into a casino holding the dagger of time...

Initially he has a single gold coin. He goes to a roulette and starts betting. On each bet he chooses between red and black and must bet a positive integer amount of coins on it. If he wins, he gets double his bet back. Otherwise he loses the bet. Roulette outcomes are distributed uniformly (i.e. both red and black have probability of $\frac{1}{2}$) and independently from each other.

After the roulette outcome is known, Prince may rollback to the point in time immediately before he made the bet and redo it in any way he wants (possibly betting on a different color or betting a different amount of coins). The roulette outcome will not change after doing rollback. Prince wants to make a total of n bets and he may use the rollback at most m times throughout them. Rolling back and redoing a bet does not count as making a new bet.

Prince wants to make sure that before each of n bets he has at least 1 coin in his possession to make a valid bet, while otherwise maximizing the expected amount of coins he will leave with. Given n and m , determine the expected amount of coins Prince would leave with. If it's not possible to guarantee that Prince makes a sequence of n valid bets, print **bankrupt** instead.

Input

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

The first line of each test case contains two integers n and m ($1 \leq n, m \leq 10^5$) — the number of betting rounds Prince will go through and the number of times he is allowed to use the rollback ability of the dagger of time.

The sum of m over all test cases does not exceed 10^6 .

Output

For each test case, if it is not possible to guarantee that Prince makes a sequence of N valid bets, print **bankrupt**, otherwise print the expected amount of coins Prince is going to win modulo $10^9 + 9$.

Formally, let $M = 10^9 + 9$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \pmod{M}$. In other words, output such an integer x that $0 \leq x < M$ and $x \cdot q \equiv p \pmod{M}$.

Example

standard input	standard output
4	3
2 1	bankrupt
4 1	7
3 2	869788168
57639 34614	

Note

Consider the first test case. According to the rules of the game, he must bet exactly one coin.

- With probability $\frac{1}{2}$, he loses the bet. Since he now has no money but must make another bet, he must rollback and bet the same coin on the opposite color. After this, he has 2 coins and no rollbacks. Suppose he bets 1 coin on his second bet.

- With probability $\frac{1}{2}$, he loses this bet and has no opportunity to rollback. In this scenario, he ends up with 1 coin.
- With probability $\frac{1}{2}$, he wins this bet and again has no opportunity to rollback (e.g. to bet more). He ends up with 3 coins.
- With probability $\frac{1}{2}$, he wins this bet. There is nothing to gain from rolling back. He now has 2 coins and 1 rollback. Suppose he bets 1 coin on his second bet. Whichever way his second bet goes, he now knows the outcome of the roulette roll. Since he has no further bets to make, there is nothing to lose from rolling back. Therefore, he should roll back and bet all his money on the winning color. After that, he ends up with 4 coins.

In total, the expected number of coins Prince of Persia has is $\frac{1}{4} \cdot 1 + \frac{1}{4} \cdot 3 + \frac{1}{2} \cdot 4 = 3$. It can be shown that this strategy is optimal for the test case.

For the second test case, suppose that Prince of Persia loses the first bet he makes. In that case, he has run out of money, but still has 3 more bets to make. Thus, he is forced to use his only rollback and bet his only coin on the opposite color. Now he has 2 coins. For the remaining three bets, suppose he loses every time. Even if he only bets one coin every time, he will run out of money: he has 2 coins before the second bet, 1 coin before the third bet and no money to make the fourth bet.

Problem K. Determinant, or...?

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

You're given an array $a_0, a_1, \dots, a_{2^n-1}$.

Consider a $2^n \times 2^n$ matrix A such that $A_{ij} = a_{i|j}$, where $i|j$ is the bitwise OR of the numbers i and j .

Find the determinant of A .

Input

The first line of input contains a single integer n ($1 \leq n \leq 20$).

The second line of input contains 2^n integers $a_0, a_1, \dots, a_{2^n-1}$ ($0 \leq a_i < 10^9 + 9$).

Output

Print a single integer, the determinant of A modulo $10^9 + 9$.

Examples

standard input	standard output
1 5 2	6
2 3 1 5 4	999999997
3 53 37 42 42 84 37 66 8	47229676

Note

In the first example, the determinant is

$$\begin{vmatrix} a_0 & a_1 \\ a_1 & a_1 \end{vmatrix} = \begin{vmatrix} 5 & 2 \\ 2 & 2 \end{vmatrix} = 10 - 4 = 6.$$

In the second example, the determinant is

$$\begin{vmatrix} 3 & 1 & 5 & 4 \\ 1 & 1 & 4 & 4 \\ 5 & 4 & 5 & 4 \\ 4 & 4 & 4 & 4 \end{vmatrix} = -12 \equiv 999999997 \pmod{10^9 + 9}.$$

Problem L. Directed Vertex Cacti

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

You are given integers n and m .

Count the number of directed graphs G without loops and multiple edges that satisfy all of the following:

- G contains exactly n vertices, labeled $1, \dots, n$.
- Every vertex lies on at most one simple cycle.
- There are exactly m edges that do not belong to any cycle.

Two graphs are considered different if there exist vertices with labels u and v such that the edge $u \rightarrow v$ exists in one graph, but not the other.

A simple cycle is a directed cycle that visits each vertex at most once.

Input

The only line of the input contains two integers n and m ($1 \leq n, m \leq 10^6$).

Output

Print the answer to the problem modulo $10^9 + 9$.

Examples

standard input	standard output
3 1	18
4 4	360
39847 348708	983575456

Note

The phrase “without multiple edges” means that there can’t be two different edges of the form $u \rightarrow v$. However, it is allowed to have an edge $u \rightarrow v$ and an edge $v \rightarrow u$.

Problem M. Siteswap

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

Siteswap (<https://en.wikipedia.org/wiki/Siteswap>) is a juggling notation that allows to represent juggling patterns.

In siteswap, it is assumed that throws happen by two alternating hands on beats that are equally spaced in time. A siteswap pattern is a sequence of numbers in which throws are represented by non-negative integers that specify the number of beats in the future after which the object is caught and thrown again.

In other words, a siteswap pattern is a sequence $a_1, a_2, \dots, a_n$, in which a_k describes the throw made on the k -th beat and means that the object is caught and thrown again after a_k seconds, that is, on the $(k + a_k)$ -th second.

For example, the number 1 means that the object is passed to the other hand to be thrown immediately on the next turn, and the number 2 means that the next throw with the same hand happens with the same object. The special number 0 in the pattern is used to denote that the hand is empty on the corresponding beat. See notes for more examples.

A siteswap pattern is valid if it can be repeated indefinitely in such way that each beat the current hand catches and then immediately throws at most one object.

For each valid pattern, it is possible to uniquely identify the number of objects needed to perform it. We may further classify these objects into those that eventually change the hand they're thrown with and those that do not. Given a valid siteswap pattern, find the number of objects that do not change the hand for each hand, and the number of objects that do change hand.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 100$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \leq n \leq 10^5$) — the length of the siteswap pattern.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^9$) — a valid siteswap pattern.

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

Output

For each test case, output three numbers.

1. The number of objects that are only thrown by the first hand (i. e. only on odd beats).
2. The number of objects that are only thrown by the second hand (i. e. only on even beats).
3. The number of objects that are thrown by both hands.

