The 3rd Universal Cup Semifinals

Contest Session

August 24, 2025

This problem set contains 13 problems and spans 21 pages.

ID	English Title
A	Iridescent Universe 2
B	Queue Editor
C	Adjacent Add
D	Circular Matching
E	Circular Convolution 2
F	Even Circuit
G	Master of Cards
H	Shortcut on Tree
I	Ethanol
J	International Olympiad in Fishing
K	Degree Sequence 3
L	Bot Friends 2
M	Traveling in Cells 3

Sponsored by

HUAWEI

ZhenFund
真格基金

Problem A. Iridescent Universe 2

Time limit: 2.5 seconds
Memory limit: 1024 megabytes

The Universe of Cup is where the Little Cyan Fish lives. The biggest planet in the universe, called the Rainow-Earth, is a sphere centered at $(0, 0, 0)$ with a radius r in the 3D Euclidean space.

Little Cirrhinus Molitorella (小鲮鱼) is running the airlines in the Rainow Earth. There are $2n$ cities as points on the Rainow Earth. There is a flight along the shortest route between the $(2i - 1)$-th city and the $2i$-th city on the surface of the Rainow Earth for each $i = 1, 2, \ldots, n$.

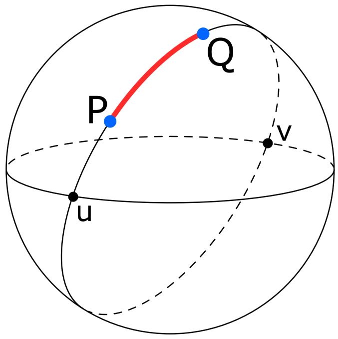


Figure 1: The shortest route between two points on a sphere.
CC BY-SA 4.0 by CheCheDaWaff on Wikimedia Commons

Little Cyan Fish is going to build a university — Pretty Kind University. However, the busy air traffic disturbs Little Cyan Fish a lot, and he wants to build the university at a place that is as far away from the noise as possible. Given his tolerance level k, Little Cyan Fish wants to find the maximum value of such d that at most k of the n flights have the minimum distance from the university **strictly** less than d.

Note that the distance between two points is calculated by measuring the shortest path on the surface of the Rainow Earth, which is NOT the Euclidean distance in 3D Euclidean space.

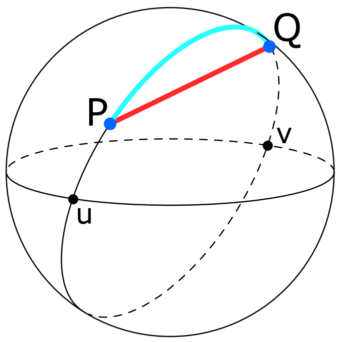


Figure 2: The distance between P and Q is the length of the light blue route, NOT the red segment.
CC BY-SA 4.0 by PlatypeanArchcow on Wikimedia Commons

Input

There are multiple test cases in a single test file. The first line of the input contains an integer T ($T \geq 1$), indicating the number of test cases. For each test case:

The first line contains three integers n ($1 \leq n \leq 100$), k ($0 \leq k < n$), and r ($1 \leq r \leq 100$), indicating the number of flights, Little Cyan Fish's tolerance level, and the radius of the Rainow Earth.

The next $2n$ lines describe all the cities. The i-th line of these lines contains three integers x, y, and z ($-100 \leq x, y, z \leq 100$, $x^2 + y^2 + z^2 > 0$), indicating that the i-th city has coordinates

$\left(\frac{rx}{\sqrt{x^2+y^2+z^2}}, \frac{ry}{\sqrt{x^2+y^2+z^2}}, \frac{rz}{\sqrt{x^2+y^2+z^2}}\right)$.

It is guaranteed that the $(2i-1)$-th city and the $2i$-th city cannot coincide with each other and cannot be directly opposite each other on the Rainow Earth for each $i = 1, 2, \ldots, n$. Therefore, the shortest path on the surface of the Rainow Earth for each flight is uniquely determined.

It is guaranteed that the sum of n over all test cases does not exceed 100.

Output

For each test case, output a line containing a single real number, indicating the maximum value of d.

Your answer is acceptable if its absolute or relative error does not exceed 10^{-6}. Formally speaking, suppose that your output is x and the jury's answer is y, and your output is accepted if and only if $\frac{|x-y|}{\max(1,|y|)} \le 10^{-6}$.

Example

standard input	standard output
3 1 0 100 0 0 1 0 1 0 2 0 100 1 1 0 1 -1 0 -1 0 1 -1 0 -1 2 1 100 1 1 0 1 -1 0 -1 0 1 -1 0 -1	235.619449019234 117.809724509617 235.619449019234

Note

For the first test case, the exact answer is 75π.

For the second test case, the exact answer is $\frac{75}{2}\pi$.

For the third test case, the exact answer is 75π.

Problem B. Queue Editor

Time limit: 3 seconds
Memory limit: 1024 megabytes

Little Cyan Fish has two queues Q_1 and Q_2, with sizes m and $m+1$, respectively. Initially, both of the queues are empty.

He needs to sequentially add several elements to both queues simultaneously. Formally, he needs to find a sequence $c_1, c_2, \ldots, c_k$, and add $c_1, c_2, \ldots, c_k$ to both queues in order.

When an element is added, if it is already in the queue, nothing happens; otherwise, the element is added to the end of the queue. If the number of elements in the queue exceeds its size, the element at the front of the queue is popped out.

Now, Little Cyan Fish gives you the final state of the two queues. The elements in these two queues, from front to back, are $a_1, a_2, \ldots, a_m$ and $b_1, b_2, \ldots, b_{m+1}$, respectively.

Little Cyan Fish wants to know if it is possible to construct a sequence c that satisfies this condition, with the length of the sequence not exceeding $50 \cdot m$. He knows that under the constraints of the problem, if a solution exists, its length will not exceed $50 \cdot m$.

Input

There are multiple test cases in a single test file. The first line of the input contains an integer T ($T \geq 1$) indicating the number of test cases. For each test case:

The first line of the input contains an integer m ($m \geq 4$), indicating the sizes of the two queues.

The next line of the input contains m integers $a_1, a_2, \ldots, a_m$ ($1 \leq a_i \leq 10^9$). It is guaranteed that the elements in a are all distinct.

The following line contains $m+1$ integers $b_1, b_2, \ldots, b_{m+1}$ ($1 \leq b_i \leq 10^9$). It is guaranteed that the elements in b are all distinct.

It is guaranteed that the sum of m over all test cases will not exceed 2×10^4.

Output

For each test case, if it is impossible to choose the subsequence, output a single line "`No`".

Otherwise, the first line of the output should contain the word "`Yes`". Then, the next line of the output should contain a single integer k ($1 \leq k \leq 50 \cdot m$), indicating the length of the sequence.

Then output a line with k integers $c_1, c_2, \ldots, c_k$ ($1 \leq c_i \leq 10^9$), indicating the constructed sequence.

If there are multiple possible solutions, you may output any of them.

Example

standard input	standard output
3 4 2 3 4 5 1 2 3 4 5 5 1 3 5 7 9 2 4 6 8 10 11 4 100 200 300 400 100 200 300 400 500	Yes 5 1 2 3 4 5 No Yes 9 100 200 300 400 500 100 200 300 400

Problem C. Adjacent Add

Time limit: 1 second
Memory limit: 1024 megabytes

Little Cyan Fish has a sequence of n integers, denoted by $a_1, a_2, \ldots, a_n$.

For a given integer $k \geq 2$, Little Cyan Fish can perform the following series of operations any number of times (including zero):

- First, choose an integer i such that $1 \leq i \leq n-1$, and choose an integer x (x can be negative).
- Then, add x to a_i, and add $k \cdot x$ to a_{i+1}.

Little Cyan Fish wants to know if he can make all the elements of a equal after performing any number of operations.

Input

There are multiple test cases in a single test file. The first line of the input contains an integer T ($T \geq 1$) indicating the number of test cases. For each test case:

The first line of the input contains two integers n and k ($n \geq 2$, $2 \leq k \leq 10^9$).

The next line of the input contains n integers $a_1, a_2, \cdots, a_n$ ($0 \leq a_i \leq 10^9$).

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

Output

For each test case, output a single line with a single word "Yes" if it is possible to make all elements of A equal, or "No" otherwise.

Example

standard input	standard output
3 3 2 9 4 2 2 4 4 7 5 3 40 63 64 96 1	Yes Yes No

Note

For the first test case, you can make all elements of a equal using the following operations:

- Choose $i = 2$, $x = 4$. Add 4 to a_2 and 8 to a_3. The array becomes $a = (9, 8, 10)$.
- Choose $i = 1$, $x = 1$. Add 1 to a_1 and 2 to a_2. The array becomes $a = (10, 10, 10)$.

For the second test case, you can make all elements of a equal using the following operations:

- Choose $i = 1$, $x = -1$. Add -1 to a_1 and -4 to a_2. The array becomes $a = (3, 3)$.

Problem D. Circular Matching

Time limit: 3 seconds
Memory limit: 1024 megabytes

Let U be a string of length $2m$ that contains exactly m 0s and m 1s. We define $f(U)$ as the answer to the following subproblem:

> Consider $2m$ points placed at equal intervals around a circle, numbered from 1 to $2m$ in a clockwise direction. Initially, each point has one ball placed on it. If $U_i = \texttt{0}$, the ball at point i is red. If $U_i = \texttt{1}$, the ball is blue. You may perform the following operation any number of times (including zero):
>
> - Choose one ball. Suppose it is currently on point i. You may move it to point $i+1$ or $i-1$.
>
> Here, point $2m+1$ refers to point 1, and point 0 refers to point $2M$.
>
> Your goal is to reach a state where, for every point i, the number of red balls and the number of blue balls on that point are equal. Find the minimum number of operations required to achieve this goal.

You are given a string S of length n consisting of 0s and 1s. You are also given q queries. Each query consists of two integers l and r. Let T be the substring of S from the l-th to the r-th character (inclusive). It is guaranteed that T contains an equal number of 0s and 1s. Compute $f(T)$.

Input

The first line of the input contains two integers n and q ($1 \le n, q \le 2 \times 10^5$).

The next line of the input contains a string S of length n consisting of 0s and 1s.

The next q lines describe all the queries. The i-th line of these lines contains two integers l_i and r_i ($1 \le l_i \le r_i \le n$), indicating a query. It is guaranteed that the substring of S from l to r contains the same number of 0s and 1s.

Output

For each query, output a single line containing a single integer, indicating the answer.

Examples

standard input	standard output
10 3 1101000110 2 5 6 9 1 10	2 2 7
29 10 11000001110001010001100100001 16 21 24 25 6 11 7 12 1 10 14 21 10 11 1 4 14 17 8 21	5 1 5 5 13 6 1 2 2 15

Note

Let's explain the first query of the first test case. $T = \texttt{1010}$. Let's consider the subproblem with $U = T$. To achieve the goal, the following sequence of operations is optimal:

- Initially, red balls are placed at points 2 and 4, and blue balls are placed at points 1 and 3.
- Move the blue ball from point 3 to point 2.
- Move the red ball from point 4 to point 1.

Thus, $f(T) = 2$.

Problem E. Circular Convolution 2

Time limit: 3 seconds
Memory limit: 1024 megabytes

After winning the *Best Paper Award* at *the 5202 annual IEEE Symposium on Foundations of Computer Science (FOCS)* by solving the $(\min, +)$-convolution problem in $O(n^{1.999})$, Little Cyan Fish wants you to solve the following problem.

Little Cyan Fish defines the $(\min, +)$ circular convolution of two sequences $a_0, a_1, \cdots, a_{n-1}$ and $b_0, b_1, \cdots, b_{n-1}$ of length n as another sequence $a \times b$ such that:

$$(a \times b)_k = \min_{(i+j) \equiv k \pmod n} (a_i + b_j)$$

For a positive integer t, Little Cyan Fish defines the t-th power of a sequence $a_0, a_1, \cdots, a_{n-1}$ as follows:

$$a^t = \begin{cases} a & t = 1 \\ a^{t-1} \times a & t > 1 \end{cases}$$

Now, Little Cyan Fish gives you a **randomly generated** sequence $a_0, a_1, \cdots, a_{n-1}$ of length n. He wants you to calculate the sequence a^n, i.e. $\underbrace{a \times a \times \cdots \times a}_{n \text{ times}}$.

Input

The first line of the input contains a single integer n ($1 \le n \le 5 \times 10^5$).

The next line of the input contains n integers $a_0, a_1, \cdots, a_{n-1}$ ($1 \le a_i \le 10^9$), indicating the sequence.

It is guaranteed that each element of $\{a_n\}$ is generated by choosing an integer from 1 to 10^9 **independently and uniformly at random**. There are no more than 100 test cases (including the examples) in this problem.

Output

Output a single line containing n integers $c_0, c_1, \cdots, c_{n-1}$, indicating the answer.

Examples

standard input	standard output
2 82688973 409689707	165377946 492378680
3 965805101 983238551 643391778	1930175334 2252588657 2270022107

Problem F. Even Circuit

Time limit: 2 seconds
Memory limit: 1024 megabytes

Little Cyan Fish has a sequence of n positive integers $a_1, a_2, \ldots, a_n$. He wants to choose a *shortest non-empty subsequence* of the sequence a with *an even length*, such that the XOR-sum of the subsequence is zero.

Formally, Little Cyan Fish wants to find an array of indices $1 \le i_1 < i_2 < \ldots < i_k \le n$, such that:

- $k > 0$
- $k \equiv 0 \pmod 2$
- $a_{i_1} \oplus a_{i_2} \oplus \ldots \oplus a_{i_k} = 0$

Here, $\oplus$ denotes the bitwise exclusive OR operation (XOR). For example, $2 \oplus 3 = 1$, $5 \oplus 1 = 4$, $3 \oplus 3 = 0$.

Little Cyan Fish wants you to determine whether it is possible to choose such a subsequence, and if yes, what is the shortest length of that subsequence.

Input

The first line of the input contains a single integer n ($2 \le n \le 2 \times 10^5$).

The next line of the input contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i < 2^{22}$).

Output

If it is impossible to choose the subsequence, output a single line "No".

Otherwise, the first line of the output should contain the word "Yes". Then, the next line of the output should contain a single integer, indicating the smallest possible k.

Examples

standard input	standard output
3 1 2 1	Yes 2
5 7 4 3 1 2	Yes 4
6 40 63 64 9 6 1	No

Note

In the first test case, when $k = 2$ and $i_1 = 1, i_2 = 3$, the conditions in the problem are satisfied. Therefore, the answer is 2.

Problem G. Master of Cards

Time limit: 1 second
Memory limit: 1024 megabytes

Little Cyan Fish is a master of cards. Today, he received $3n$ cards. There are a total of n types of cards, and each type has exactly 3 identical cards. Each card of type i has a triple of integers written on it: (a_i, b_i, c_i).

He can perform the following operation any number of times:

- First, choose 2 cards. These cards must satisfy the following condition:
  - Suppose the two cards are of type i and type j. Then at least one of the following must be true: $a_i = a_j$, $b_i = b_j$, or $c_i = c_j$.
- Then, discard both selected cards. (Discarded cards cannot be used again.)

The goal of Little Cyan Fish is to perform as many operations as he can. Find a possible plan for him!

Input

The first line of the input contains a single integer n ($1 \leq n \leq 2 \times 10^5$).

The next n lines of the input describe all the cards. The i-th line of these lines contains three integers a_i, b_i, and c_i ($1 \leq a_i, b_i, c_i \leq n$).

Output

The first line of the output should contain a single integer k, indicating the maximum number of operations Little Cyan Fish can perform.

The next k lines describe all the operations. The i-th line of these lines contains two integers u_i and v_i, indicating an operation.

Examples

standard input	standard output
2 1 2 2 2 1 2	3 2 2 2 1 1 1
3 1 2 3 2 2 1 3 3 1	4 1 1 2 2 3 3 2 3

Note

In the first sample test case, you can perform 3 operations in total, which is the maximum. The operations are as follows:

- Discard two cards of type 2.
- Discard one card of type 2 and one card of type 1.
- Discard two cards of type 1.

Problem H. Shortcut on Tree

Time limit: 2 seconds
Memory limit: 1024 megabytes

You are given a directed tree with n vertices, where each vertex is numbered from 1 to n. The tree is rooted at vertex 1, and it is guaranteed that all vertices are reachable from the root. For each $2 \le i \le n$, the tree has a directed edge from vertex p_i to vertex i.

Little Cyan Fish wants to add up to n additional directed edges to this graph to make the following condition satisfied:

- For any pair of different integers (u, v) such that $1 \le u \le n$ and $1 \le v \le n$, it is possible to go from vertex u to vertex v using at most 4 edges.

Help Little Cyan Fish to find a possible way to add the edges.

Input

There are multiple test cases in a single test file. The first line of the input contains an integer T ($T \ge 1$) indicating the number of test cases. For each test case:

The first line of the input contains a single integer n ($n \ge 2$).

The next line of the input contains $n-1$ integers $p_2, p_3, \ldots, p_n$ ($1 \le p_i < i$), indicating the parent of each vertex $2 \le i \le n$.

It is guaranteed that the sum of n over all test cases does not exceed 4 000.

Output

For each test case, if it is impossible to add at most n edges to satisfy Little Cyan Fish's requirement, output a single line containing a single word "`No`".

Otherwise, the first line of the output contains a single word "`Yes`".

The next line of the output contains the number of added edges m ($0 \le m \le n$). The next m lines each describe one added edge as two integers u_i and v_i ($1 \le u_i, v_i \le n$) — the start and end of the i-th added edge.

Example

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

Note

In the first test case, you can satisfy the condition in the problem by adding an edge from vertex 3 to vertex 1.

Problem I. Ethanol

Time limit: 2 seconds
Memory limit: 1024 megabytes

Little Cyan Fish is playing with some bottles. On his table, there are $n+2$ sufficiently large containers numbered $0, 1, \ldots, n+1$.

Initially, container 0 contains water of mass $\frac{X}{1000}$, and container $n+1$ is empty. For each $i = 1, \ldots, n$, container i contains a uniform mixture of total mass 1, containing $\frac{E_i}{1000}$ ethanol and $1 - \frac{E_i}{1000}$ water.

Little Cyan Fish may repeat the following operation any (finite) number of times (possibly zero):

1. Choose an index i ($0 \le i \le n$) and a real number $x > 0$ such that, if $i \ge 1$, container i currently contains a mixture of mass at least $1 + x$ (this condition is not required if $i = 0$).
2. Transfer a mixture of mass x from container i to container $i+1$, then stir the contents of container $i+1$ so that the resulting mixture becomes uniform.

Yeah... I know you are sick of seeing yet another constructive problem. Therefore, Little Cyan Fish proposes this gift for you. He asks you to find the supremum (the least upper bound not necessarily attainable) of the mass of ethanol that may end up in container $n+1$ after performing a finite sequence of operations. Does this sound good to you?

Input

There are multiple test cases in a single test file. The first line of the input contains an integer T ($1 \le T \le 1\,000$), indicating the number of test cases. For each test case:

The first line of the input contains two integers n and X ($1 \le n \le 20$, $1 \le X \le 1\,000$).

The next line of the input contains n integers $E_1, E_2, \ldots, E_n$ ($0 \le E_i \le 1\,000$).

Output

For each test case, print a single line containing a single real number, indicating the supremum of the ethanol mass in container $n+1$ after operations.

Your answer is acceptable if its absolute or relative error does not exceed 10^{-9}. Formally speaking, suppose that your output is x and the jury's answer is y, and your output is accepted if and only if $\frac{|x-y|}{\max(1,|y|)} \le 10^{-9}$.

Examples

standard input	standard output
2 1 900 1000 2 345 678 910	0.59343034025940088812 0.29768625581348055032
1 3 210 406 364 961	0.18956125995075295122

Note

For example, for the first test case, you can make the mass of ethanol that ends up in container 2 equal to $\frac{23}{45}$ by performing the following operations.

1. Transfer a mixture of mass 0.5 from container 0 to container 1.
2. Transfer a mixture of mass 0.4 from container 1 to container 2.
3. Transfer a mixture of mass 0.4 from container 0 to container 1.
4. Transfer a mixture of mass 0.5 from container 1 to container 2.

In fact, you can achieve approximately 0.593... by performing operations properly.

Problem J. International Olympiad in Fishing

Time limit: 2 seconds
Memory limit: 1024 megabytes

The 1st International Olympiad in Fishing ("IOF") is planned to be hosted in Atlantis this year. The best fish all over the world will compete for the title of the *World Fish Champion*.

Little Cyan Fish is going to participate in the championship. The chairman of the contest, called Little Gelly Fish, has told him the rules of this year's contest: a card game called "Balatro" will be used to determine the winner.

In the game, there are n cards, indexed with integers from 1 to n. Each card has two additional attributes: the rank and the suit. Both the rank and the suit are independently and uniformly selected random integers within the range $[1, n]$, and two different cards may have the same rank or suit.

The index of each card is written on the back of the card, while the rank and suit are on the front. Currently, all the cards are placed on the table in ascending order of their indices, with their backs facing up. Therefore, the player knows the indices of all the cards but cannot see the rank or suit of each card.

There are also two buttons labeled "Rank" and "Suit" on the table. Little Gelly Fish told Little Cyan Fish that he can operate these two buttons, with the following rules:

- If Little Cyan Fish presses the button labeled "Rank", all the cards will be sorted in ascending order based on their ranks.
- If Little Cyan Fish presses the button labeled "Suit", all the cards will be sorted in ascending order based on their suits.

Little Gelly Fish noted that the sorting process will be *stable*. This means, after pressing one of the buttons, for any two cards, if their corresponding attributes (either rank or suit, depending on the button) are different, the card with the smaller value will appear earlier in the sorted order. If their corresponding attributes are the same, their relative order will remain unchanged.

"You can press the button multiple times," Little Gelly Fish said, "and... you must tell me the exact rank and suit of each card."

Little Cyan Fish wants to know that, under the optimal strategy, what is the probability of guessing the ranks and suits correctly. Since Little Gelly Fish lives in the world of $\mathbb{F}_{998\,244\,353}$, you only need to give the answer modulo 998 244 353.

Input

The first line of the input contains one integer n ($1 \le n \le 5 \times 10^5$).

Output

Output a single line containing a single integer, indicating the answer, modulo 998 244 353.

Examples

standard input	standard output
2	686292993
5	301495273
52	126716306

Note

There are two cards in the first example, and under the constraints of this example, Little Cyan Fish can adopt the following strategy:

First, Little Cyan Fish will press the "Rank" button once to check if the two cards are swapped, and then press the "Suit" button once to check if the two cards are swapped.

Let's consider all possible outcomes:

- If both the "Rank" button and the "Suit" button cause the cards to swap, Little Cyan Fish will guess as follows:
  - The first card has a rank of 2 and a suit of 1.
  - The second card has a rank of 1 and a suit of 2.
- If the "Rank" button causes a swap but the "Suit" button does not, Little Cyan Fish will guess as follows:
  - The first card has a rank of 2 and a suit of 2.
  - The second card has a rank of 1 and a suit of 1.
- If neither the "Rank" button nor the "Suit" button causes a swap, Little Cyan Fish will guess as follows:
  - The first card has a rank of 1 and a suit of 1.
  - The second card has a rank of 2 and a suit of 2.
- If the "Rank" button does not cause a swap but the "Suit" button causes a swap, Little Cyan Fish will press the "Rank" button again:
  - If a swap occurs after pressing the "Rank" button for the second time, Little Cyan Fish will guess as follows:
    - The first card has a rank of 1 and a suit of 2.
    - The second card has a rank of 2 and a suit of 1.
  - If no swap occurs after pressing the "Rank" button for the second time, Little Cyan Fish will guess as follows:
    - The first card has a rank of 1 and a suit of 2.
    - The second card has a rank of 1 and a suit of 1.

It can be proven that with the strategy described above, Little Cyan Fish has a $\frac{5}{16} \equiv 686\,292\,993 \pmod{998\,244\,353}$ probability of guessing all ranks and suits correctly, and this is optimal among all strategies.

Problem K. Degree Sequence 3

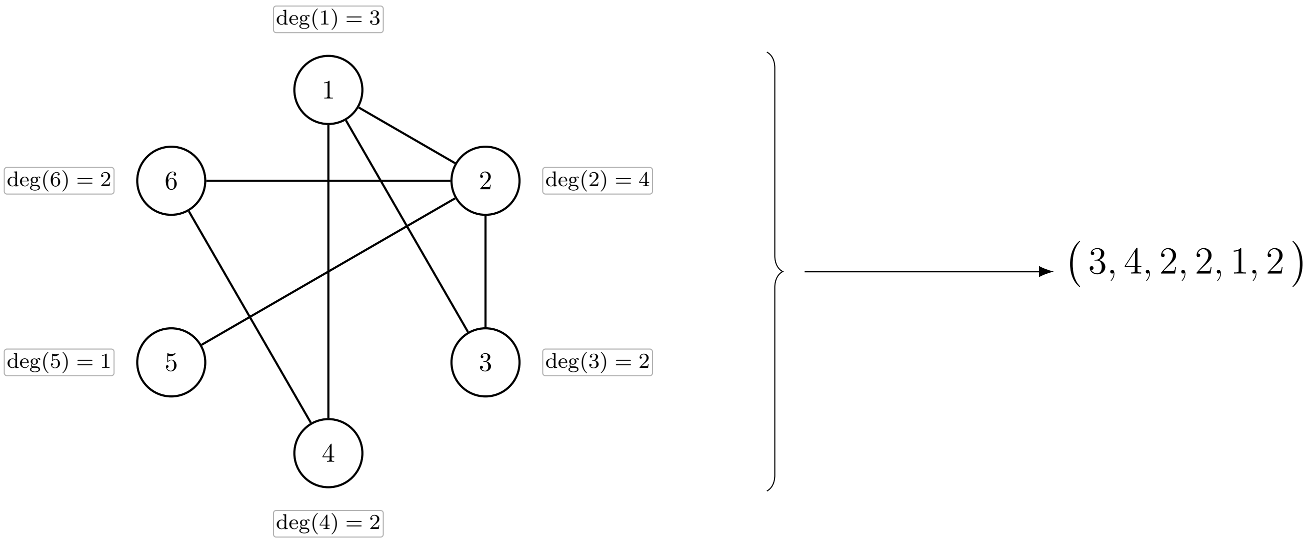
Time limit: 3 seconds
Memory limit: 1024 megabytes

In *The 2nd Universal Cup Semifinals*, you discover some records of the Mayan civilization's study of graph theory at Chichén Itzá. Among the ruins, you found some *degree sequences* of some trees recorded by the ancient Maya.

Later, at the *2024 ICPC Training Camp* in Gui'an, you may (or may not) have solved the task *Degree Sequence 2*. Little Cyan Fish is eager to share that problem with you, but since the contest might appear as a regular stage in The 4th Universal Cup, he cannot reveal it just yet.

But don't be disappointed! Who said you must solve Degree Sequence 2 before attempting Degree Sequence 3? Here it is!

Recall the definition of the degree sequence: for an **undirected**, **simple** (i.e. no multiple edges or self loops) graph of n vertices, its degree sequence is an integer sequence of length n, denoted by $d_1, d_2, \ldots, d_n$, such that d_i is equal to the degree (i.e. the number of edges that are incident to the vertex) of the vertex i.



A *sequence a* is said to be *graphic*, or a *valid degree sequence*, if there exists a simple undirected graph, such that the degree sequence of the graph is exactly a. For example, $(3, 4, 2, 2, 1, 2)$ is a valid degree sequence, as the graph described above will give it as a degree sequence.

Now, Little Cyan Fish gives you a sequence $a_1, a_2, \ldots, a_n$. Little Cyan Fish wants you to convert the sequence to a valid *degree sequence* of a simple undirected graph. To do that, Little Cyan Fish can perform the following operations as many times as he wants:

- Choose an index $1 \le i \le n$ and update $a_i \leftarrow a_i - 1$. The cost of such operation is b_i dollars.
- Choose an index $1 \le i \le n$ and update $a_i \leftarrow a_i + 1$. The cost of such operation is also b_i dollars.

Given the sequences a and b, your task is to find the minimum total cost to convert the sequence a into a valid degree sequence.

Input

There are multiple test cases in a single test file. The first line of the input contains an integer T ($T \ge 1$), indicating the number of test cases. For each test case:

The first line contains a single integer n ($1 \le n \le 10^5$).

The next line of the input contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le n$), indicating the initial sequence.

The next line of the input contains n integers $b_1, b_2, \ldots, b_n$ ($1 \le b_i \le 10^9$), indicating the cost to make a change.

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

Output

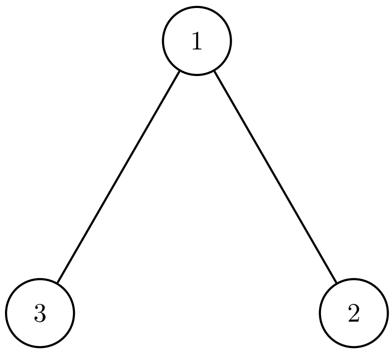
For each test case, output a single line containing a single integer, indicating the answer.

Example

standard input	standard output
3 3 2 1 1 100 1000 10000 3 2 1 0 100 10 1 5 1 2 3 4 5 1 10 100 1000 10000	0 1 10002

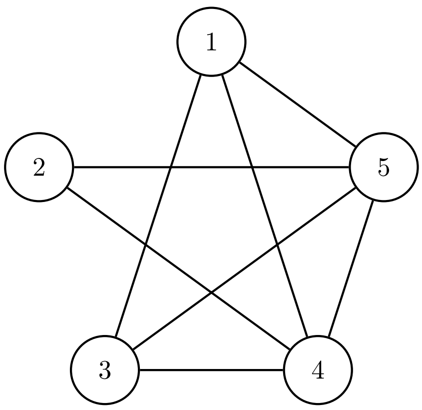
Note

For the first test case, we do not need to perform any operations, as $(2, 1, 1)$ is already a valid degree sequence.



For the second test case, the optimal plan is to update $a_3 \leftarrow a_3 + 1$, so that the sequence a becomes $(2, 1, 1)$. The total cost is $b_3 = 1$.

For the third test case, the optimal plan is to update $a_5 \leftarrow a_5 - 1$, and then update $a_1 \leftarrow a_1 + 1$ twice, so that the sequence a becomes $(3, 2, 3, 4, 4)$. The total cost is $2 \cdot b_1 + b_5 = 10002$.



Problem L. Bot Friends 2

Time limit: 3 seconds
Memory limit: 1024 megabytes

Little Cyan Fish is commanding your favorite bot friends on a map. The map is represented as a connected undirected graph with n vertices and m edges. The vertices are labeled from 1 to n, and the i-th vertex has weight a_i. The edges are labeled from 1 to m, and the i-th edge has weight w_i.

Initially, there are n bots, one on each vertex: the i-th bot is placed on vertex i. On each day, Little Cyan Fish may perform any number of the following operations:

- Choose a bot x currently on vertex u, and an incident edge (u, v) with weight w. Move the bot from u to v. This operation costs w dollars.
- Choose two bots x and y currently on the same vertex u. Merge them into a single bot. This operation costs a_u dollars.

Little Cyan Fish really wants to make you happy, but... Well, you only love one bot. Therefore, Little Cyan Fish must merge all bots into a single bot. Help him find the minimum total cost of operations required to achieve this!

Input

There are multiple test cases in a single test file. The first line of the input contains an integer T ($T \geq 1$) indicating the number of test cases. For each test case:

The first line of the input contains two integers n and m ($n \geq 1$), indicating the number of vertices and the number of edges.

The next line of the input contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^{12}$), indicating the weight of each vertex.

The next m lines of the input describe all the edges. The i-th line of these lines contains three integers u_i, v_i, and w_i ($1 \leq u_i, v_i \leq n$, $1 \leq w_i \leq 10^9$, $u_i \neq v_i$), indicating an edge connecting the vertex u_i and v_i. It is guaranteed that the graph is connected, but there might be multiple edges connecting the same pair of vertices.

It is guaranteed that the sum of n over all test cases will not exceed 5×10^5, and the sum of m over all test cases will not exceed 10^6.

Output

For each test case, output a single line containing a single integer, indicating the minimum total cost required to merge all the bots into a single bot you love.

It can be proven that there will always be a valid plan under the constraints of the problem.

Example

standard input	standard output
3 4 4 2 3 7 1 1 2 3 1 3 1 2 3 2 3 4 2 5 4 100000 100000 100000 100000 1 1 2 10 2 3 100 3 4 1000 4 5 10000 1 0 1000000000	12 43214 0

Problem M. Traveling in Cells 3

Time limit: 4 seconds
Memory limit: 1024 megabytes

Have you ever traveled to the Kingdom of Little Cyan Fish? It is a beautiful — maybe even magical — land, imagined as a straight line of $\left(2 \times 10^{100} + 1\right)$ cells, numbered consecutively from -10^{100} to 10^{100}. The Motto of the Kingdom "*In Code We Trust*", reflects the importance of competitive programming in this lovely place. Its pride and glory is the Universal Cup.

Figure 3: Hosting The 2nd Universal Cup Finals.

The president of the kingdom — none other than the Little Cyan Fish — is about to choose the host city for The 3rd Universal Cup Finals, scheduled for 2026. There are n qualified teams, with the i-th team currently residing in cell x_i.

Before the finals begin, all teams must travel to the chosen host cell. In the kingdom, there are two possible modes of transportation:

- Railway: The kingdom has a main rail line, where each cell i is directly connected to $i-1$ and $i+1$. Moving between two adjacent cells costs 1 dollar.
- Air flights: Each cell also has an airport. From cell i, flights are available to $i-a$ or $i+a$, each at a fixed cost of b dollars.

Little Cyan Fish does not care how long the journey takes, but the total travel cost is a serious concern, as the budget is limited. Your task is to determine the minimum total amount of money Little Cyan Fish must prepare if the host cell is chosen optimally.

Input

There are multiple test cases in a single test file. The first line of the input contains an integer T ($T \geq 1$) indicating the number of test cases. For each test case:

The first line of the input contains three integers n, a and b ($n \geq 1$, $1 \leq a, b \leq 10^{12}$), indicating the number of the qualified teams, and the parameters of the airlines.

The next line of the input contains n integers $x_1, x_2, \cdots, x_n$ ($-10^{12} \leq x_i \leq 10^{12}$), indicating the initial position of each team.

It is guaranteed that the sum of n over all test cases does not exceed 3×10^5.

Output

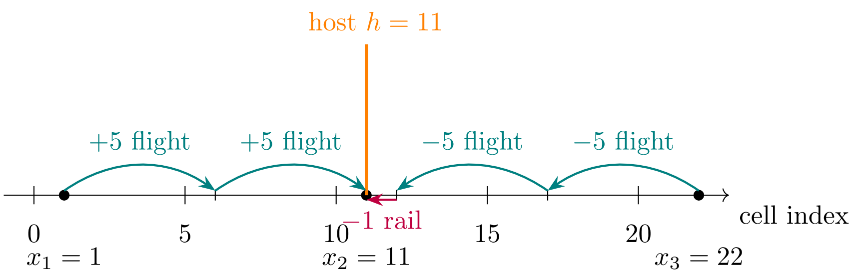
For each test case, output a single line containing a single integer, indicating the answer.

Example

standard input	standard output
3 3 5 1 1 11 22 4 5 3 1 3 5 8 7 6 3 2 9 15 24 33 40 53	5 7 55

Note

For the first test case, one of the optimal plans is to host The 3rd Universal Cup Finals at the 11-th cell.



$a = 5,\ b = 1$
Paths: $1 \to 6 \to 11$: \$2; 11 : \$0;
$22 \to 17 \to 12 \to 11$: \$3
Total: \$5