Problem A. Aloha

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

Chiaki lives in a special universe where the the law of conservation of energy is quite different. Specifically, an equation $v^2 + 2gy = 0$ can be found to describe the law, where v is it's speed at point (x, y) and g is the gravitational acceleration which equals to 10.

At first, Chiaki is at origin $(0, 0)$ without any kinetic energy and she would like to go to some point (x, y) by some pipes.

Chiaki has three pipes whose length are l_1, l_2 and l_3. She must use the pipes to build a tunnel, which is the route to the destination.

- At least one pipe should be used, each pipe can only be used once and the pipe cannot be bent or cut off.
- The tunnel must start from $(0, 0)$ and end at the destination (x, y).
- The pipes must be connected end to end.

Chiaki would like to know whether it is possible to reach the destination and the minimum time to go to the destination if it is possible.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 500$), indicating the number of test cases. For each test case:

The first line contains five integers x, y, l_1, l_2 and l_3 ($-1000 \le x, y \le 1000$, $1 \le l_1, l_2, l_3 \le 1000$) – the coordinators of the destination and the length of each pipe.

Output

For each test case, output an real number denoting the minimum time, or a string "Impossible!" (without the quotes) if Chiaki cannot reach the destination.

Your answer will be considered correct if and only if the absolute error or relative error of your answer is less than 10^{-8}.

Example

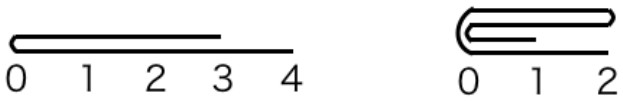
standard input	standard output
4 0 1 1 1 1 3 -4 2 3 3 -1000 0 499 499 3 0 -8 10 2 1000	Impossible! 1.11803398874989490253 90.00544834369912905458 1.56351606067883830420

Problem B. Origami

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

Chiaki has a very big sheet of paper. This sheet has a form of rectangle with dimensions $1 \times n$ and numbers from 1 to n was written on each small 1×1 grid. Chiaki would like to fold the paper using the following operations:

1. Fold the sheet of paper at position p_i to the right. After this query the leftmost part of the paper with dimensions $1 \times p_i$ must be above the rightmost part of the paper with dimensions $1 \times ([\textit{current width of sheet}] - p_i)$.
2. Fold the sheet of paper at position p_i to the left. After this query the rightmost part of the paper with dimensions $1 \times ([\textit{current width of sheet}] - p_i)$ must be above the leftmost part of the paper with dimensions $1 \times p_i$.



After performing the above operations several times, the sheet of paper has dimensions 1×1. If we write down the number on each grid from top to bottom, we will have a permutation of n.

Now given a permutation of n, Chiaki would like to know whether it is possible to obtain the permutation using the above operations.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \leq n \leq 10^6$), indicating the length of the paper.

The second line contains n integers $a_1, a_2, \ldots, a_n$, which is a permutation of n, indicating the integers marked in the grids of the resulting sheet of paper from top to bottom.

It's guaranteed that the sum of n in all test cases will not exceed 10^6.

Output

For each test case output one line. If it's possible to obtain the permutation, output "Yes" (without the quotes), otherwise output "No" (without the quotes).

Example

standard input	standard output
3 4 2 1 4 3 7 2 5 4 3 6 1 7 4 1 3 2 4	Yes Yes No

Problem C. Continued Story

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

Chiaki has a rooted tree with n nodes numbered 1 to n. The root is node 1. Each edge has a positive integer weight on it.

Now, two players are playing a game on the tree. They play the game in turn. In each turn, the player should choose an edge and decrease its weight by 1.

If an edge's weight becomes 0, it will be removed. After an edge is removed, the tree will be split into two parts. The part without the root node should be discarded (i.e. permanently removed from the tree).

If in a turn, a player has no edge left to choose, he loses the game.

Chiaki, as the player with the first turn, would like to know which edges she can choose at the first turn, to make sure she will win?

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \leq n \leq 10^6$) – the number of nodes in the tree.

Each of the next $n-1$ lines contains two integers p_i and w_i ($2 \leq i \leq n$, $1 \leq p_i \leq n$, $1 \leq w_i \leq 10^9$) where p_i denotes the parent of the i-th node and w_i is the weight of the edge between i and p_i.

It's guaranteed that the sum of n in all test cases will not exceed 10^6.

Output

For each test case, output two lines.

The first line is a single number m, the number of edges Chiaki can choose at the first move to make sure Chiaki will win.

The second line contains m increasing numbers separated with one space. Output the number u means you can choose the edge between node u and its parent.

Example

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

Problem D. The Easiest One

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

Given an nonnegative integer x, Chiaki can perform the following two operations:

1. obtain $x - 1$ from x.
2. obtain $x - 2^i$ from x, if x AND 2^i is not 0.

Let $f(x, y)$ be the minimum operations needed to change x to y using the above operations, Chiaki would like to know the value of

$$\sum_{0 \le y \le x \le n} f(x, y)$$

where n is a given number.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains an integer n in binary representation ($0 \le n < 2^{500}$) without leading zeros.

It's guaranteed that the sum of lengths of n over all test cases will not exceed 500.

Output

For each test case, output the answer modulo $(10^9 + 7)$.

Example

standard input	standard output
10 0 1 10 11 100 101 110 111 1000 1001	0 1 3 7 13 22 31 43 60 83

Problem E. Set

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

Chiaki is going to create n sets $S_0, S_1, \ldots, S_{n-1}$ under the following restrictions:

- $|S_i| = m$.
- $|S_i \setminus S_{(i-1+n) \bmod n}| \geq l_i$.
- $|S_0 \cup S_1 \cup \cdots \cup S_{n-1}|$ should be minimized.

Note that $|S|$ means the size of set S and $A \setminus B$ means set difference which is defined as $\{x : x \text{ in } A \text{ and } x \text{ not in } B\}$

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains two integers n and m ($2 \leq n \leq 10^6$, $0 \leq m \leq 10^{18}$) – the number of sets and the size of each set.

The second line contains n integers $l_0, l_1, \ldots, l_{n-1}$ ($0 \leq l_i \leq 10^{18}$).

It's guaranteed that the sum of n in all test cases will not exceed 10^6.

Output

For each test case, output an integer denoting the minimum value of $|S_0 \cup S_1 \cup \cdots \cup S_{n-1}|$. Or -1 if the restrictions cannot be satisfied.

Example

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

Problem F. Old Problem

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

Chiaki has an $n \times m$ grid map and there are $(n+1) \times (m+1)$ grid points on the map. She would like to know the number of grid right triangles whose area is $\frac{s}{2}$.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \leq T \leq 10000$), indicating the number of test cases. For each test case:

The first line contains three integers n, m and s ($1 \leq n, m, s \leq 10^8$).

Output

For each test case, output the answer modulo $(10^9 + 7)$.

Example

standard input	standard output
2 1 1 1 2 2 2	4 24

Problem G. Periodic Palindrome

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

Chiaki has a string $\ldots ww^r ww^r \ldots$ with infinite length, where $w = w_1 w_2 \ldots w_m$ and $w^r = w_m w_{m-1} \ldots w_1$.

Chiaki cuts out a substring $s = s_1 s_2 \ldots s_n$ $(m < n)$ from the infinite string. And it's guaranteed s contains w or w^r as a substring. She would like to know the number of pairs (i, j) where $1 \leq i \leq j \leq n$ and $s_{i..j}$ is a possible value of w or w^r.

Input

There are multiple test cases. The first line of input contains an integer T indicating the number of test cases. For each test case:

The first line contains a string s $(2 \leq |s| \leq 10^6)$ consisting only of lowercase English letters.

It is guaranteed that the sum of $|s|$ in all cases does not exceed 10^6.

Output

For each test case, output an integer denoting the answer.

Example

standard input	standard output
1 aaa	5

Problem H. Accel World

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

Chiaki was trapped in a strange place which can be regarded as a connected undirected graph with n vertices (numbered from 1 to n) and m weighted edges (numbered from 1 to m). In the beginning, Chiaki is at vertex 1 with speed v equals to 1 unit per second and would like to go to the vertex n.

There are some special vertices in the graph, called *acceleration vertex*. Once Chiaki reached an acceleration vertex, her speed will be doubled: from v to $2v$. The same acceleration vertex can be used multiple times to achieve multiple acceleration, but it only works under this limitation: the last acceleration vertex Chiaki visited should not equal to the current acceleration vertex Chiaki reached.

For example, vertex 2 and 3 are acceleration vertices while 1, 4 and 5 are not. If Chiaki chooses the path $1 \to 2 \to 3 \to 2 \to 5$, Chiaki's speed would be accelerated for three times (8 unit per second in the end). But if the path is $1 \to 2 \to 4 \to 2 \to 4 \to 2 \to 5$, Chiaki would have only one acceleration.

Chiaki would like to know the minimum time needed to reach the vertex n.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains three integers n, m and k ($2 \le n \le 100$, $1 \le m \le 8000$, $0 \le k \le n$) – the number of vertices, the number of edges and the number of special vertices.

Each of the following m lines contains three integers u_i, v_i and w_i ($1 \le u_i, v_i \le n$, $1 \le w_i \le 1000$) denoting an edge with w_i unit length connecting u_i and v_i.

The next line contains k integers $p_1, p_2, \ldots, p_k$ ($2 \le p_i \le n - 1$) denoting the index of each *acceleration vertex*.

It's guaranteed that the sum of n over all test cases will not exceed 1000 and the sum of m over all test cases will not exceed 80000.

Output

For each test case, output a real number t denoting the minimum time to reach the vertex n and an integer s denoting the maximum times of acceleration Chiaki can achieve among all optimal solutions.

Your answer for t will be considered correct if and only if the absolute error or relative error of your answer is less than 10^{-8}. And by the way, if s is greater than 32767, output "Burst!" (without the quotes) instead.

Example

standard input	standard output
2 1 2 1 2 1 1 2 5 4 2 1 2 1 2 3 1 3 4 1 4 5 1 2 3 6 7 2 1 2 2 2 4 2 4 6 2 1 3 2 3 4 2 4 5 4 5 6 4 3 4	2 2.0000000000 Burst! 3.5000000000 2

Problem I. Hello, Hello and Hello

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

A ternary string is a sequence of digits, where each digit is either 0, 1, or 2.

Chiaki has a nonempty ternary string s. Initially, the characters are sorted in non-decreasing order (i.e. all 0s appear before all 1s and all 1s appear before all 2s). Chiaki would like to shuffle the characters such that no two consecutive characters have the same value using the following operation: choose two integers l and r ($l \leq r$), take characters from position l to position r inclusively, and move them to the end of the string.

Chiaki would like to know the minimum number of operations needed.

Input

There are multiple test cases. The first line of input is an integer T indicates the number of test cases. For each test case:

The first line contains a ternary string s ($1 \leq |s| \leq 10^6$).

It is guaranteed that the sum of all $|s|$ does not exceed 10^6.

Output

For each test case, output "-1" (without the quotes) if Chiaki can not shuffle the string using the operations described above. Otherwise, output an integer k in the first line – the minimum number of operations needed. Each of the next k lines output two integers l and r – denoting an operation.

Example

standard input	standard output
2 001122 000022	2 4 5 1 1 -1

Problem J. Sortable Path on Tree

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

Chiaki has a tree with n nodes numbered 1 to n. Each nodes has a positive integer weight w_i on it.

Chiaki would like to know the number of unordered pairs (u, v) such that:

> Let $t_1 \to t_2 \to \cdots \to t_k$ $(t_1 = u, t_k = v)$ be the shortest path from u to v. Then the sequence $w_{t_1}, w_{t_2}, \ldots, w_{t_k}$ or the sequence $w_{t_k}, w_{t_{k-1}}, \ldots, w_{t_1}$ can be sorted into nondecreasing order using several circular shift operations.

Note that a circular shift is the operation of rearranging the entries in a sequence, either by moving the final entry to the first position, while shifting all other entries to the next position, or by performing the inverse operation.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains an integer n $(1 \le n \le 10^5)$ – the number of nodes in the tree.

The second line contains n integers $w_1, w_2, \ldots, w_n$ $(1 \le w_i \le 10^5)$.

Each of the next $n-1$ lines contains two integers u and v $(1 \le u, v \le n, u \ne v)$ denoting an edge on tree.

It's guaranteed that the sum of n in all test cases will not exceed 10^5.

Output

For each test case, output an integer denoting the answer.

Example

standard input	standard output
1 4 1 2 3 4 1 2 2 3 3 4	10