Problem A
Pac-Man

Time limit: 1 second

Memory limit: 1024 megabytes

Problem Description

Pac-Man is a classic video game developed in 1980s. The player controls the character "Pac-Man" to eat dots in a maze while avoiding enemy characters "ghosts." When Pac-Man eats all dots in the maze, the player wins. When any ghost catches Pac-Man, the player loses.

Adam is learning how to create games with modern programming tools. He would like to make an imitation of the Pac-Man game with some changes. He changes the playable character from "Pac-Man" to a "ghost," and he also alter the winning condition. When Pac-Man eats all dots in the maze, the player loses. When the playable ghost catches Pac-Man, the player wins.

To prepare the movement of Pac-Man, Adam uses crowd sourcing.

Input Format

The input has exactly nine lines. Each of them is a string which is either `Tiger` or `Lion`, and it represents a casted vote from an anonymous voter.

Output Format

Output one line containing the next king's name.

Technical Specification

- Every string in the input is either `Tiger` or `Lion`.
- The next king must be qualified by at least five votes.
- There are multiple input files.
- There is only one test case in each input file.

Sample Input 1

```
Lion
Lion
Tiger
Tiger
Lion
Lion
Tiger
Tiger
Tiger
```

Sample Output 1

```
Tiger
```

Problem B
Folding

Time limit: 1 second
Memory limit: 1024 megabytes

Problem Description

Your friend Bob is a garbage collector working in a factory. The factory manufactures various kinds of chemical substances, and the toxic waste they may produce must be collected with caution. Every piece of toxic waste has an identifier that represents its chemical composition. It is very dangerous to put two pieces of toxic waste with different identifiers into a garbage bag, since they might produce some chemical reactions that lead to explosion, fire, toxic smoke or other chemical hazards.

Today, Bob has to collect n very small pieces of toxic waste, and their identifiers are $a_1, a_2, \ldots, a_n$, respectively. These pieces are small enough to fit in one garbage bag, but Bob might as well use more bags to avoid any potential hazard. Please write a program to help Bob to calculate the minimum number of garbage bags required to safely collect all n pieces of toxic waste.

Input Format

The first line of the input contains one integer n where n is the number of pieces of toxic wastes. The second line of the input contains n positive integers $a_1, \ldots, a_n$ which are the identifiers associated with the corresponding pieces.

Output Format

Output the answer in a line.

Technical Specification

- $2 \leq n \leq 10^5$, and $a_1, \ldots, a_n$ are positive integers at most 10^9.
- There are multiple input files.
- There is only one test case in each input file.

Sample Input 1

```
3
1 1 3
```

Sample Output 1

```
2
```

Sample Input 2

```
5
5 5 6 8 8
```

Sample Output 2

```
3
```

Sample Input 3

```
7
4 1 2 8 8 8 8
```

Sample Output 3

```
4
```

Almost blank page

Problem C
Circles

Time limit: 2 seconds

Memory limit: 1024 megabytes

Problem Description

The change-making problem is a classical competitive programming problem. The problem is about a currency system. In this system, there are n available denominations of coins $\$c_1, \$c_2, \ldots, \$c_n$ where $c_1 = 1$ and $c_2, \ldots, c_n$ are all integers. Assume you can have unlimited supply of each denomination of coins. The change-making problem is: given a target x, make a change of $\$x$ with minimum number of coins.

The change-making problem has appeared so many times in programming contest. Many contestants know how to solve this problem. However, there are also many contestants doing it wrongly. For example, many of them use greedy algorithm which does not work. This greedy algorithm repeatedly takes the coin with greatest value which does not exceed x. This algorithm actually works for some currency system but not for this case: \$1, \$3, \$4 and the target $x = 6$. We call such case a counterexample to this greedy algorithm, and $x = 6$ is a witness of the currency system \$1, \$3, \$4.

Please write a program to find the minimum witness for a given currency system. If there does not exist a witness or the minimum witness is greater than 10^5, your program should output -1.

Input Format

The first line contains an integer n to indicate the number of denominations in the currency system. The second line contains n integers $c_1, \ldots, c_n$ where the currency system has $\$c_1, \ldots, \$c_n$.

Output Format

Output the minimum witness x if $x \leq 10^5$. Otherwise output -1.

Technical Specification

- $n \leq 50$
- $c_1 = 1 < c_2 < \cdots < c_n \leq 10^5$
- There are multiple input files.
- There is only one test case in each input file.

Sample Input 1

```
3
1 3 4
```

Sample Output 1

```
6
```

Sample Input 2

```
5
1 5 10 20 50
```

Sample Output 2

```
-1
```

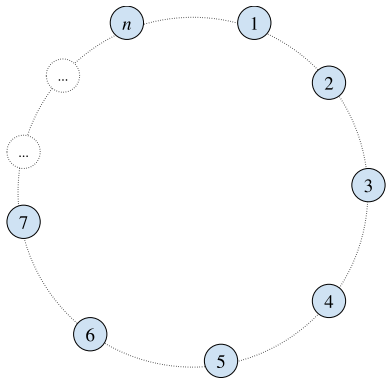
Problem D
Last Will

Time limit: 1 second

Memory limit: 1024 megabytes

Problem Description

You are playing a tactical game and facing a brutal combat. Your rival's commander uses a circular formation to protect their headquarter, and you have to disrupt their defense to win the battle. The enemy's circular formation consists of n soldiers numbered from 1 to n. In the beginning, soldier i and soldier j are adjacent if $|i - j| \in \{1, n - 1\}$.



You only have a small crew of warriors. Your force is too weak to fight more than two soldiers or any two soldiers who are not adjacent. Moreover, if you try to attack a single soldier, both of their adjacent soldiers will come to the rescue. In such situation, it is equivalent to fighting three soldiers. Thus, you may only launch attacks to aim at the gap between two adjacent soldiers. By doing so, you have a chance to take down these two soldiers. Your enemy will fill the gap after your attack. For example, soldiers 3 and n will become adjacent if you take down soldiers 1 and 2. You can repeatedly take down the soldiers until no one can defend their headquarter.

Unfortunately, you are still unable to defeat them in some situations. Each of the soldiers has their own value, and there are at most k different kinds of values in total. You should have heard "United we stand, divided we fall" before. The soldiers with the same value can unite, and the soldiers with different values cannot. When you attack two soldiers with different values, you always take them down. But when you attack two soldiers with the same value, they will not fall.

Please write a program to find out an attacking strategy to win the battle by disrupting the enemy's defense. That is, take down all n soldiers of the circular formation.

Input Format

The first line contains two integers n and k. n is the number of enemy soldiers. k is the number of kinds of different values. Their values are numbered from 1 to k. The second line contains

n integers $v_1, \ldots, v_n$ where soldier i's value is v_i for $i \in \{1, \ldots, n\}$.

Output Format

If there is no way to disrupt the enemy's defense, output -1. Otherwise, output $\frac{n}{2}$ lines. The i-th line describes your i-th attack with two integers p_i and q_i separated by a blank. Your i-th attack is to take down soldiers p_i and q_i. At that time, they must be adjacent and have different values.

Technical Specification

- $2 \leq k \leq n \leq 1000$
- n is even.
- If there are multiple solutions, you may output any one of them.
- There are multiple input files.
- There is only one test case in each input file.

Sample Input 1

```
10 2
1 1 1 1 1 1 2 2 2 2
```

Sample Output 1

```
-1
```

Sample Input 2

```
10 4
1 2 1 3 4 4 1 2 1 3
```

Sample Output 2

```
1 2
3 4
7 8
6 9
10 5
```

Sample Input 3

```
10 4
1 2 1 3 1 4 2 4 2 3
```

Sample Output 3

```
1 2
3 4
7 8
5 6
9 10
```

Problem E
Exact Length

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

In the Konosuba World, Megumin is a famous archwizard for her trademark Explosion magic. She must cast Explosion once a day or die. Her Explosion magic is extremely powerful and consumes a lot of energy. Megumin can only cast Explosion once a day, and she always becomes exhausted after casting Explosion. That is why her teammate Kazuma always carries her home and worries about her.

Today, Megumin practices her Explosion on a plain area. There are n trees on the plain, and the i-th tree is located at (x_i, y_i) for $1 \leq i \leq n$. You may assume that $(x_i, y_i) \neq (x_j, y_j)$ for $i \neq j$. Megumin wants to find a tree to stand on its top, then cast her Explosion with the longest spell in the Konosuba World. Her Explosion can destroy everything within a circular area of radius r. That is, if the area is centered at (c_x, c_y) and a tree is located at (x, y) where $(x - c_x)^2 + (y - c_y)^2 \leq r^2$, then the tree will disappear after Megumin's Explosion.

Kazuma is trying to find an ideal way for Megumin to cast Explosion. Assume that Megumin stands on the top of the tree located at (x, y) and then casts Explosion on the circular area centered at (c_x, c_y). This way is ideal if it satisfies the following conditions.

- Megumin won't hurt herself while casting Explosion. That is, $(x - c_x)^2 + (y - c_y)^2 > r^2$.
- All the other trees must disappear after Megumin's Explosion. That is, for any i such that $(x_i, y_i) \neq (x, y)$, we have $(x_i - c_x)^2 + (y_i - c_y)^2 \leq r^2$.

Please help Kazuma to find out an ideal way. If there is no such way for Megumin to cast, output -1. Otherwise, please tell Kazuma where the tree for Megumin to stand and the center of Megumin's Explosion are.

Input Format

The first line contains two integers n and r. n is the number of the trees, and r is the radius of Explosion. The following n lines describe the positions of the trees. Each of them contains two integers x and y, and the corresponding tree is located at (x, y).

Output Format

If there is no ideal way, output -1 on a line. Otherwise, output two lines. The first line contains two integers x and y separated by a blank. Megumin can safely cast Explosion on the top of the tree at (x, y). The second line contains two real numbers c_x and c_y separated by a blank. Megumin should cast Explosion certered at (c_x, c_y) to destroy all the other trees. Please note that c_x and c_y may be fractional numbers.

Technical Specification

- $1 < n \leq 50000$
- $1 \leq r \leq 10^7$
- $x, y \in [-10^7, 10^7]$
- Your output may not contain any string token longer than 100 characters.
- You may need to use `__int128` and `__float128` if you use C/C++.
- You may need to use `BigInteger` and `BigDecimal` if you use Java/Kotlin.
- Your output will be checked by a Kotlin program using `BigDecimal` with a 256-digit precision.
- There are multiple input files.
- There is only one test case in each input file.
- There always exists a solution which can be represented in the given format.

Sample Input 1

```
4 1
1 0
0 1
2 2
3 3
```

Sample Output 1

```
-1
```

Sample Input 2

```
4 1
0 0
1 1
0 2
1 0
```

Sample Output 2

```
1 0
0 1
```

Sample Input 3

```
5 1
0 0
1 1
1 0
0 1
2 2
```

Sample Output 3

```
2 2
0.5 0.5
```

Problem F
Homework

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

Integer factorization serves an important role in many cryptography systems. It is about finding two positive integers p and q for a given positive composite number n such that $n = pq$ and $1 < p \leq q < n$. However, it is a well-known NP-intermediate candidate. We still don't have any algorithm to solve it in polynomial time.

Taylor, a number theorist, created another factorization problem as follows.

Given a prime number p and two integers $a_0, a_1 \in \{0, 1, \ldots p-1\}$. Find two integers $b_0, b_1 \in \{0, 1, \ldots p-1\}$ such that $a_0 \equiv b_0 \cdot b_1 \pmod{p}$ and $a_1 \equiv b_0 + b_1 \pmod{p}$.

"This factoring is way much cooler, in the sense that it can be computed efficiently," said Taylor. Now, he invites you to enjoy this new variant of factorization.

Input Format

The first line contains an integer $1 \leq T \leq 100$ indicating the number of test cases. For each test case, there is a line containing three non-negative integers p, a_0, a_1 separated by a single blank.

Output Format

For each test case, output a line containing b_0 and b_1 in ascending order separated by a single blank if b_0 and b_1 satisfy the equations. If there are multiple solutions, you may output any of them. If there is no solution, output -1.

Technical Specification

- $1 \leq T \leq 100$
- $1 < p < 2^{31}$ and p is a prime number.
- $a_0, a_1 \in \{0, 1, \ldots, p-1\}$.

Sample Input 1

```
2
2 1 0
2 1 1
```

Sample Output 1

```
1 1
-1
```

Almost blank page

Problem G
rareone

Time limit: 10 seconds
Memory limit: 1024 megabytes

Problem Description

Artwork Credit: バフコ twitter: `@bafuko_seiso`
Youtube Channel: `https://www.youtube.com/channel/UC94NiQXdPedmfOE_qlIBd7Q`

Bafuko is a gifted student. Despite being a teenager, she is studying advanced algorithms now. She just learned about graphs and trees. The instructor of the class gave a programming assignment today. This task is about recovering the adjacency list of a tree graph $G_T = (V, E)$ from a given undirected graph $G_U = (V, F)$. The properties of G_T and G_U are listed as follows.

- G_T is a tree graph.
- G_T and G_U have the same vertex set $V = \{1, \ldots, n\}$.
- For every vertex $v \in V$, the degree of v in G_T is at most 3.
- The edge $\{u, v\}$ belongs to F if and only if the distance between u and v in G_T is either 1 or 2.

You, as a professional programmer, will like to finish the assignment in 20 minutes and brag about how easy it is while playing Bafuko's favorite game — Super Smash Sisters. Let's see if you can do it!

Input Format

First line contains a number T indicating the number of test cases. For each test case, the first line contains a number n indicating the number of vertices in the tree. Then n lines follow. The i-th line descibes the neighbors of i in G_U. It is started by c_i, the numbers of neighbors of i, then c_i distinct numbers $v_{i,1}, \ldots, v_{i,c_i}$ follow, where $\{i, v_1\}, \ldots, \{i, v_{c_i}\} \in F$ are edges in the given graph G_U.

Output Format

For each test case, please output n lines to describe the adjacency list of the tree graph G_T. For the i-th line, please ouput the number of neighbors of i and then the neighbors of i in ascending order. Separate adjacent numbers by a space.

Technical Specification

- $7 \leq n \leq 10^5$
- $1 \leq T \leq 4.5 \times 10^5$
- The sum of n's in all test cases in a single test file is at most 3.6×10^6.
- For each test case, you can always find a unique tree graph G_T.

Sample Input 1

```
2
7
2 2 3
4 1 3 4 6
6 1 2 4 5 6 7
4 2 3 5 6
2 3 4
4 2 3 4 7
2 3 6
7
2 2 3
3 1 3 4
4 1 2 4 5
4 2 3 5 6
4 3 4 6 7
3 4 5 7
2 5 6
```

Sample Output 1

```
1 2
2 1 3
3 2 4 6
2 3 5
1 4
2 3 7
1 6
1 2
2 1 3
2 2 4
2 3 5
2 4 6
2 5 7
1 6
```

Problem H
Table Tennis

Time limit: 4 seconds
Memory limit: 1024 megabytes

Problem Description

Mr. Oshiro wants to build a new hotel in Celeste Mountain Range. He needs you to find a good place him. First, Oshiro wants to build the hotel on a continuous segment without any valley in between, and the hotel should be as wide as possible. However, some places are too dangerous for the construction, so he will ask you this question several times: what is the maximum width of a constructible hotel in a certain segment of Celeste Mountain Range?

To clarify the problem statement, we define the following terms.

- Celeste Mountain Range is n kilometers long.
- The topography (shape of the land surface) is described by an integer sequence $h_0, \ldots, h_{n-1}$ indicating the average heights in meters. h_i is the average height in the area from the i-th kilometer to the $(i+1)$-th kilometer of Celeste Mountain Range.
- Every query will be a continuous segment in Celeste Mountain Range that has corresponding sequence of heights.
- A valley is a place which is lower than its neighbors. In this problem, we call an area from i-th kilometer to the $(i+1)$-th kilometer a valley if $h_i < \min(h_{i-1}, h_{i+1})$ and $0 < i < n-1$. The area before the first kilometer and the area after the $(n-2)$-th are not considered as valleys, because h_{-1} and h_n are not defined.
- A continuous segment is a unimodal interval if it has no valley in it except perhaps at either of its ends. For example, if the query segment has heights $(1, 2, 3, 4, 3, 4, 5)$, then segments with heights $(1, 2, 3, 4, 3)$ and $(3, 4, 5)$ are unimodal intervals, but that with height $(3, 4, 3, 4)$ is not.
- Mr. Oshiro only builds his hotel on a unimodal interval. Thus, the answer to the aforementioned query with heights $(1, 2, 3, 4, 3, 4, 5)$ is 5.

Mysterious phenomena often occur in Celeste Mountain Range. Sometimes the height of a certain area will change. But don't worry, Mr. Oshiro will tell you the information just in time before he comes to you with more and more questions!

Input Format

The first line contains one integer T, which is the number of test cases. In each test case, the first line contains one integer n, the length of Celeste Mountain Range in kilometers. The second line contains n integers denoting the initial heights. The third line contains one integer q which is the total number of queries and updates. Then q lines follow. Each of them is either

in form "1 p d" or "2 ℓ r". "1 p d" represents a mysterious phenomenon where p is the position and d is the amount of the height changed. That is, h_p is updated with a new value $h_p + d$. Note that d might be negative to indicate that a decrease in height. "2 ℓ r" represents the query from Oshiro. He wants to know the maximum width of the constructible hotel in the segment (ℓ, r) with height $(h_\ell, \ldots, h_r)$.

Output Format

For each query, output one line containing the maximum width of a constructible hotel.

Technical Specification

- $1 \leq T \leq 10$
- $1 \leq n \leq 10^5$
- $1 \leq q \leq 10^5$
- $p, \ell, r \in \{0, 1, \ldots, n-1\}$
- The heights will always be in the range $[1, 10^9]$.
- It is guaranteed that after any change, $h_{i-1} \neq h_i$ for all $0 < i < n$.

Sample Input 1

```
1
10
1 2 3 2 1 2 3 2 1 2
6
2 1 3
2 3 5
1 4 4
1 3 2
1 5 2
2 2 9
```

Sample Output 1

```
3
2
7
```