# Problem A
# Pac-Man

Time limit: 1 second

Memory limit: 1024 megabytes

## Problem Description

Pac-Man is a maze-chase video game developed in 1980s. The player controls the character "Pac-Man" to eat dots in a maze while avoiding enemy characters "ghosts." All characters may move in four directions: up, down, left, right. The game ends in two conditions:

- Pac-Man eats all dots in the maze. In this case, the player wins.
- Any ghost catches Pac-Man. In this case, the player loses.

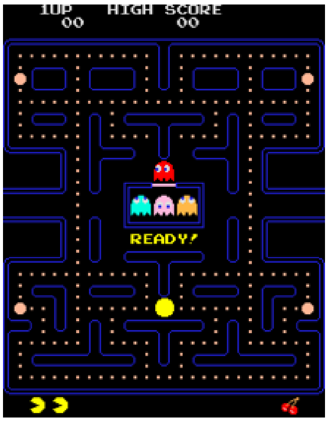



Figure 1: Pac-Man gameplay (image from Wikipedia)

Adam is learning how to create games with modern programming tools. To practice the skills, he tries to make an imitation of the Pac-Man game with some modification. In Adam's game, the playable character is a "ghost," and the enemy character is "Pac-Man." Since he changes the roles of the ghost and Pac-Man, he also changes the ending conditions of the game.

- Pac-Man eats all dots in the maze. In this case, the player loses.
- The ghost controlled by the player catches Pac-Man. In this case, the player wins.

Adam has almost developed the first full functioning version of his game. He wants to test his game and creates a simple stage for testing. The maze of the stage is based on 10-by-10 grid. We label the cell lying at the intersection of row $r$ and column $c$ with $(r, c)$. In this problem, rows and columns are numbered from 0 to 9. Each grid cell contains exact one dot. The exterior boundary of the grid are walls. No characters may move to the area outside of the grid. Inside the grid, there are no walls or obstacles. All characters may move freely from a cell to any cell adjacent to it. Note that two grid cells $(r_1, c_1)$ and $(r_2, c_2)$ are adjacent to each other if and only if $|r_1 - r_2| + |c_1 + c_2| = 1$.

Adam has to prepare the movements of Pac-Man for the testing. He needs a set of trajectory with diversity, but any of the trajectories must satisfy the following requirements.

- Pac-Man can eat all dots in the maze if it follows the trajectory.
- Pac-Man moves at most 10000 steps.

Adam needs your help to generate a trajectory starting at cell $(x, y)$. Please write a program to generate a trajectory of Pac-Man satisfying all requirements above and starting at cell $(x, y)$.

## Input Format

The input has exactly one line which consists of two integers $x$ and $y$ separated by a blank. You are asked to generate a trajectory starting at cell $(x, y)$.

## Output Format

You must output a requested trajectory in the following format. The trajectory is represented by $m+1$ lines where $m$ is the number of steps of the trajectory. The $i$-th line contains two integers $r_i$ and $c_j$. Pac-Man will be in cell $(r_i, c_i)$ after moving $i$ steps along the trajectory.

## Technical Specification

- $m \leq 10000$
- $x, y, r_i, c_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ for $i \in \{1, 2, \ldots, m+1\}$.
- Cells $(r_i, c_i)$ and $(r_{i+1}, c_{i+1})$ are adjacent to each other for $i \in \{1, 2, \ldots, m\}$.
- $\{(r_1, c_1)\} \cup \{(r_2, c_2)\} \cup \cdots \cup \{(r_{m+1}, c_{m+1})\}$

## Sample Input 1

```
0 0
```

## Sample Output 1

```
0 0
0 1
0 2
0 3
0 4
0 5
...
...
...
9 3
9 2
9 1
9 0
```

## Note

The sample output section does not contain the correct output, since it ignores a large part of the answer. Please download the correct sample test cases from the judge system.

# Problem B
# Folding

Time limit: 1 second
Memory limit: 1024 megabytes

## Problem Description

There is a transparent tape. Its length is exact one meter ($10^9$ nanometers). In this problem, all numbers are integers, and we use a number to denote a position on the tape. The number $p$ denote the position of the point has a distance $p$ nanometers from the head of tape.

Bob is a master dyer, so he can color the tape precisely in nanometer scale. He colors two sectors $[p_1, q_1]$ and $[p_2, q_2]$ into red. The color of the tape at the position in the range from $p_1$ to $q_1$ is red. The color of the tape at the position in the range from $p_2$ to $q_2$ is also red. And the rest part remains transparent.

To verify Bob's skill, we ask Ben, the tape folding master, to help us. Ben can fold the tape perfectly at any position. If Ben fold the tape at $x$, then the new position of a certain point $p$ will be one of the following cases.

- If $p = x$, then it becomes the new head of tape, i.e, it becomes 0.
- If $p > x$, then it becomes $p - x$.
- If $p < x$, then it becomes $x - p$.

After Ben folds the tape, we measure the total length of the red part of the new tape. If the red part has the expected length, then we will believe Bob and Ben are both masters in their skills. Obviously, the color of some position of the new tape is determined by the colors of the corresponding positions of the old tape. A position of the new tape is colored in red if one of the corresponding positions in the old tape is colored in red.

Bob has already colored the tape, and Ben has proposed the positions to be folded. Please write a program to compute the expected lengths colored in red.

## Input Format

The first line contains four integers $p_1, q_1, p_2, q_2$ separated by blanks. Bob has colored the sectors $[p_1, q_1]$ and $[p_2, q_2]$. The second line contains an integer $q$ indicating the number of positions to be folded by Ben. Each of the remaining $q$ lines contains an integer $x$ indicating the positions to be folded by Ben.

## Output Format

For each position, output the expected total length of the new tape where are colored in red.

## Technical Specification

- $0 \le p_1 < q_1 < p_2 < q_2 \le 10^9$
- $0 \le x \le 10^9$

- $q \leq 10^6$

## Sample Input 1

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

## Sample Output 1

```
3
2
3
3
2
3
3
3
3
3
```

# Problem C
# Circles

Time limit: 2 seconds
Memory limit: 1024 megabytes

## Problem Description

There are $n$ magical circles on a plane. They are centered at $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, respectively. In the beginning, the radius of each circle is 0. The radius of all magical circles will grow at the same rate until they touch another magical circle. Write a program calculate the total area of the sum of all magical circles at the end of growing.

## Input Format

The first line contains an integer $n$ to indicate the number of magical circles. The $i$-th of the following $n$ lines contains two integers $x_i$ and $y_i$ indicating that the $i$-th magical circle is centered at $(x_i, y_i)$.

## Output Format

Output the total area of the circles. A relative error of $10^{-6}$ is acceptable.

## Technical Specification

- $n \leq 2000$
- $x_i, y_i \in [-10^5, 10^5]$ for $i \in \{1, 2, \ldots, n\}$.

## Sample Input 1

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

## Sample Output 1

```
3.14159265359
```

Almost blank page

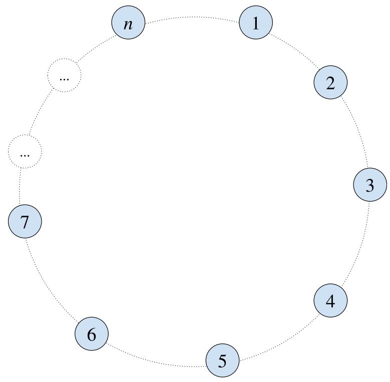
# Problem D
# Last Will

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.

# 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.

# Problem F
# Homework

Time limit: 3 seconds

Memory limit: 1024 megabytes

## Problem Description

There are $n$ children (numbered from 1 to $n$) who are learning the arithmetic operations, which include *addition* "+", *subtraction* "−", *multiplication* "×", and *division* "÷", on rational numbers. Each child has a paper sheet with only one zero on it. Their teacher, Frank, will give out $q$ operations. The $i$-th operation consists of an operator $c_i$ and an integer $x_i$. However, Frank only wants some children to perform the operation. Only children $\ell_i, \ell_{i+1}, \ldots, r_i$ are asked to append the operator $c_i$ and the number $x_i$ to their paper sheet. After Frank's assignment, every child has an expression to evaluate.

For example, let $n = 3$, $q = 2$, $c_1$ be "+", $x_1 = 1$, $\ell_1 = 1$, $r_1 = 2$, $c_2$ be "−", $x_2 = 2$, $\ell_2 = 2$, $r_2 = 3$. The expressions of children 1, 2 and 3 are $0 + 1$, $0 + 1 - 2$ and $0 - 2$, respectively.

However, Frank is really lazy and wants to verify the answers quickly. So he asks you to calculate the sums of the values of all children's expressions. If the value of the expression assigned to child $i$ is $\frac{a_i}{b_i}$, then you have to use $a \times b^{-1} \mod 10^9 + 7$ instead. $b^{-1}$ is any number satisfying $b \times b^{-1} \equiv 1 \mod 10^9 + 7$. If the sum is greater than $10^9 + 6$, then return the sum modulo $10^9 + 7$ to Frank.

Note: The arithmetic operations has PEMDAS rule, that is, Multiplication/Division before Addition/Subtraction.

## Input Format

The first line contains two integers $n$ and $q$ separated by a blank. The $i$-th of following $q$ lines contains $\ell_i, r_i, c_i, x_i$ separated by blanks.

## Output Format

Output the number that you should return to Frank.

## 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

```
3 2
1 2 + 1
2 3 - 2
```

## Sample Output 1

```
1000000005
```

Almost blank page

# Problem G
# Cactus

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/UC94NiQXdPedmf0E_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 $3.6 \times 10^6$.
- For each test case, you can always find a unique tree graph $G_T$.

# Problem H
# rareone

Time limit: 10 seconds

Memory limit: 1024 megabytes

## Problem Description

There's no such thing as public opinion.

*Jordan Ellenberg, American Mathematician*

In K City lives $n$ residents, they want to build a connection network with each other. However, some residents want the network wire coloured black while the others want the wire coloured white. The opinion of resident $i$ can be quantified as a number $a_i$. If we build a network wire between residents $i$ and $j$, the cost of this wire will be $a_i \times a_j$.

The mayor of K City wants to network built such that:

1. There is exactly $n - 1$ wire used.
2. For any two different residents $i$ and $j$, there exists a sequence $p_1, \cdots, p_k$ such that $p_1 = i$, $p_k = j$ and residents $p_\ell$ and $p_{\ell+1}$ share a wire for $1 \le \ell < k$.

In other words, the network should be a tree.

You, the renowned mathematician of K City, want to know not only the *minimum* cost to build the network. In the name of confusion, you also want to know the *maximum* cost!

## Input Format

The first line begins with a number $n$ indicating the number of residents. The second line contains $n$ numbers $a_1, a_2, \ldots, a_n$. The opinion of resident $i$ is the quantified as $a_i$.

## Output Format

Output two numbers separated by a blank in a line. The numbers are the *minimum* cost and the *maximum* cost to build the network, respectively. Since the numbers may be extremely large, you have to modulo the answer with $10^9 + 7$.

## Technical Specification

- $1 \le n \le 10^6$
- $|a_i| \le 10^6$

## Sample Input 1

```
10
-5 -10 -7 -7 -3 -1 -7 -5 -8 -6
```

## Sample Output 1

```
58 490
```

## Sample Input 2

```
10
-5 1 2 -2 -1 1 -5 5 -10 6
```

## Sample Output 2

```
999999779 183
```

## Sample Input 3

```
10
0 0 0 0 0 0 0 0 0 0
```

## Sample Output 3

```
0 0
```

## Sample Input 4

```
10
10 8 9 3 8 8 0 5 3 10
```

## Sample Output 4

```
0 540
```

# Problem I
# Table Tennis

Time limit: 4 seconds

Memory limit: 1024 megabytes

## Problem Description

## Input Format

## Output Format

## Technical Specification