ICPC ASIA DHAKA REGIONAL CONTEST 2023

Hosted by

BUBT

Bangladesh University of Business and Technology

Supported by

DIGITAL BANGLADESH
Skilled. Equipped. DigitalReady.

ICT DIVISION
FUTURE IS HERE

BANGLADESH COMPUTER COUNCIL
COMPUTER FOR EVERYTHING

04 November, 2023

You get 16 Pages, 11 Problems & 300 Minutes

Bangladesh Association of Problem Setters

Sponsored By

Gold Sponsors

Therap (BD)

DSi Dynamic Solution Innovators Ltd.

SIBL Social Islami Bank Limited

PRAN

Other Sponsors

AG AROMA GROUP

AFREEN ENTERPRISE — COMMITTED TO EXCELLENCE —

JoulesLabs

Problem A

Gona Guni

You are given a tree of **N** nodes. You can pick a non-empty set of nodes **S** from the tree. Create a minimal size connected subgraph **G** from the tree such that it covers all nodes in the set.
cost(S) = size of minimum vertex cover of G.

Note that, minimum vertex cover of a graph is a set of vertices that includes at least one end point of every edge of the graph and size of the set is as minimum as possible.
You need to find the sum of **$cost(S)^M$** for all possible sets **S** modulo **998244353.**
Notes: Minimum vertex cover of a single node graph is 0.

Input

First line will contain a single integer **$T (1 \le T \le 3000)$** representing the test cases. For each test case, there will be a single line containing two integers **N ($1 \le N \le 3 \times 10^5$)** and **M ($0 \le M \le 200$)** separated by space. After that **N - 1** lines follow. In each line there will be two space separated integers **U** and **V** denoting an edge between node **U** and **V** (**$1 \le U, V \le N$**). **The sum of N over all test cases $\le 3 \times 10^5$**

Output

For each test case print a single line representing the answer modulo **998244353**.

Sample Input	Output for Sample Input
2 3 1 1 2 1 3 20 200 1 2 1 3 2 4 1 5 5 6 1 7 6 8 6 9 3 10 4 11 6 12 11 13 4 14 13 15 15 16 6 17 13 18 15 19 13 20	4 286430678

Problem B

ASCII Table

ASCII (American Standard Code for Information Interchange) Characters within the range 33-126 are the most frequently used ASCII Characters that are visible to the naked eye. These characters can be printed in tabular form in column major order in more than one way. Some of which are shown below:

```
! ( / 6 = D K R Y ` g n u |
" ) 0 7 > E L S Z a h o v }
# * 1 8 ? F M T [ b i p w ~
$ + 2 9 @ G N U \ c j q x
% , 3 : A H O V ] d k r y
& - 4 ; B I P W ^ e l s z
' . 5 < C J Q X _ f m t {
```

33-126 valued ASCII Characters printed in a 7*14 Grid

```
! * 3 < E N W ` i r {
" + 4 = F O X a j s |
# , 5 > G P Y b k t }
$ - 6 ? H Q Z c l u ~
% . 7 @ I R [ d m v
& / 8 A J S \ e n w
' 0 9 B K T ] f o x
( 1 : C L U ^ g p y
) 2 ; D M V _ h q z
```

33-126 valued ASCII Characters printed in a 9*11 Grid

```
! ) 1 9 A I Q Y a i q y
" * 2 : B J R Z b j r z
# + 3 ; C K S [ c k s {
$ , 4 < D L T \ d l t |
% - 5 = E M U ] e m u }
& . 6 > F N V ^ f n v ~
' / 7 ? G O W _ g o w
( 0 8 @ H P X ` h p x
```

33-126 valued ASCII Characters printed in a 8*12 Grid

```
! + 5 ? I S ] g q {
" , 6 @ J T ^ h r |
# - 7 A K U _ i s }
$ . 8 B L V ` j t ~
% / 9 C M W a k u
& 0 : D N X b l v
' 1 ; E O Y c m w
( 2 < F P Z d n x
) 3 = G Q [ e o y
* 4 > H R \ f p z
```

33-126 valued ASCII Characters printed in a 10*10 Grid

So no columns except the last one can have empty cells at the bottom but each column must contain at least one character (Filled from the top).

Now the problem is that somehow you have lost the dimension of the printed grid but you remember two characters that were printed in the same row. Based on this information you will have to find the possible dimension of the original printed grid.

Input

First line of the input file contains a positive integer **T (T ≤ 10000)** which denotes the number of test cases to follow. Each test case consists of two space separated characters c_1 and c_2 ($33 \le c_1, c_2 \le 126$ and $c_1 \neq c_2$) as inputs in a single line. Look at the sample input for details.

Output

For each test case, produce one or more lines of output, each of which should contain two integers that denote the dimension of a possible (Row and Column) grid. These two integers should be separated by a single space. If there is more than one solution then they should be printed in the increasing order of the value of row.

Print a blank line after the output of each test case. Look at the output for sample input for details.

Sample Input	Output for Sample Input
2 A U 1 8	1 94 2 47 4 24 5 19 10 10 20 5 1 94 7 14

Problem C

Bears

Bears were learning about rectangles! They have found that a set of **M** rectangles is called nestable, if

- All of them have integer height and width.
- There exists an ordering of them so that **i**-th rectangle's height is strictly less than **(i+1)**-th rectangle's height and **i**-th rectangle's width is strictly less than **(i+1)**-th rectangle's width, for **i=1, 2,.., M-1**.

Do you remember the array **A** of **N** integers you have found last night? For each **i = 1, 2, ..., N**, you are allowed to make at most one rectangle with area exactly A_i.

You are given the array **A.** What is the maximum number of rectangles you can make so that the Bears call them nestable?

Input

The first line will contain a single integer **T** ($1 \le T \le 10^5$). Each test case will have a single integer **N** ($1 \le N \le 10^5$), denoting the size of the block. The following line will have **N** integers separated by spaces, denoting array **A** ($1 \le A_i \le N$).
Sum of **N** over all cases does not exceed 6×10^5.

Output

For each test case, print a line consisting of the answer.

Sample Input	Output for Sample Input
3 1 1 4 1 2 3 4 9 1 2 3 4 5 6 7 8 9	1 2 3

Problem D

Pyramid

You have to make a pyramid shaped shopping mall with a square base in the desert. The surface area involves a lot of cost (Larger base needs more land processing and the larger sides creates a challenge for air conditioning in the heat of desert). So the total surface area (The square base + four triangular sides) will be fixed. Given this total surface area you will have to predict the volume of the largest possible pyramid.

Input

The input file contains at most **10000** lines of input. Each line contains a positive floating-point number not exceeding **1000000** which denotes the surface area of the Pyramid. Input is terminated by a line containing a negative value.

Output

For each line of input produce one line of output. This line should contain a floating-point number (rounded to four digits after the decimal point and there must be at least one digit on the left side of the decimal point) that denotes the maximum possible volume of the pyramid. You can assume that a very small precision error (for judge solution, output ±**0.0000001** produces the same output) will not cause a difference in output.

Sample Input	Output for Sample Input
0.01 0.02 -1	0.0001 0.0002

Image take from: https://commons.wikimedia.org/wiki/File:Kheops-Pyramid.jpg

Problem E

Crazy General

On a planet far far away from Earth, there is a crazy general in the army of Wonderland. Recently she has derived a new strategy of deploying her soldiers. She has learned that each soldier needs to watch another soldier's back. She took the phrase literally and discovered that this can be only achieved by forming cycles, e.g. soldier A watches soldier B and B watches C and C watches A. She needs to deploy her **n** soldiers in two days. On the first day, she randomly selects **k** soldiers and deploys them by forming **c1** cycles. On the next day, she deploys the remaining soldiers by forming **c2** cycles. Each of these cycles must have at least **r** soldiers and she has to deploy at least **r** soldiers on each day.

The general needs to find the most efficient strategy to win the war, but first she wants to compute the number of ways to deploy her soldiers for all possible values of **k**, when the total number of cycles in two days' deployment cannot be more than **C**, i.e. **c1 + c2 ≤ C**.

Input

The first line of input contains a single integer, **T** (**$T \le 10^5$**). Then **T** lines follow. Each of these **T** lines will contain **3** integers: **n (1 ≤ n ≤ 1000), C (1 ≤ C ≤ 1000) and r (1 ≤ r ≤ 10).**

Output

For each of the cases output "**Case <x>: <y>**" in a separate line, where **x** is the case number and **y** is the number of different possible deployments. As **y** can be quite large, output **y modulo 1,000,000,007**.

Sample Input	Output for Sample Input
3 4 2 2 3 2 2 2 2 1	Case 1: 6 Case 2: 0 Case 3: 2

Explanation of Sample I/O

1st case: As **r** = 2 and **n** = 4 and she needs to deploy at least **r** soldiers on both days, the only possible value of **k** is 2. Due to the same constraints, the only possible values of **c1** and **c2** are 1. Now, she can choose 2 soldiers for the 1st day in 6 ways and then there is only 1 way to make a cycle. For the 2nd day, we have to deploy the remaining soldiers and again we can make a cycle in only 1 way. Therefore, the total number of possible deployments is 6.

2nd Case: There are not enough soldiers to satisfy the constraints. Therefore, the total number of possible deployments is 0.

3rd Case: There are only 2 ways to deploy them. It is possible to make a cycle consisting of only 1 soldier.

Problem F

Uncle Bob and XOR Sum

Uncle Bob, the Mayor of Bytelands is not only a great leader but also an enthusiastic problem solver. He is so obsessed with problem solving that he hired a team to create new problems for him everyday. But no matter how hard the problem is, Uncle Bob solves it within a few minutes. As the days pass by, he feels bored with solving such easy problems. So, he threatens the team to come up with an exciting problem or they have to look for a new job. After hours of brainstorming the team came up with the following problem.

Given two arrays of integers **A** and **B**, determine the number of **non-empty** subsets of **A** where the **XOR Sum** of the subset does not contain a submask that is present in the array **B.** More formally, if the XOR SUM of the chosen subset is **S** and the submasks of **S** are $\{m_1, m_2, \ldots, m_n\}$ then $m_i \notin B$ $(1 \le i \le n)$. As the number of valid **non-empty** subsets can be huge, you need to provide the remainder of the answer when divided by **1000,000,007.**

A set **A** is a subset of a set **S** if all the elements in **A** are also elements of **S**. For example, **{1, 2}** is a subset of **{1, 2, 3}**. Because the set **{1, 2, 3}** contains all the elements of set **{1, 2}**. An **empty set {}** is also a subset of a set. Here is an example of the all possible subsets of a set **{1, 2, 3} = [{}, {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}]**. If we exclude the empty subset from all possible subsets, then it becomes all possible **non-empty** subsets. Note that, **{1, 4}** is not a subset of **{1, 2, 3}** as the set **{1, 2, 3}** does not contain **4**.

XOR SUM of a set $\{a_1, a_2, \ldots, a_k\} = a_1 \oplus a_2 \oplus \ldots \oplus a_k$ (the character $\oplus$ denotes bitwise XOR)

A submask of a number **n** represents a **bitmask** (binary representation of a number) in which a bit can only be set if the bit in the bitmask of **n** is set at the same position. Here are some of the examples of submasks:
Submasks of **1 (1) = {0, 1}**
Submasks of **3 (11) = {00, 01, 10, 11}**
Submasks of **5 (101) = {000, 001, 100, 101}**

Looking at the problem, Uncle Bob was stunned. He never thought in a million years that he would be given a problem that requires any bitwise operations let alone XOR SUM. Now he regrets threatening the team. He is aware that not being able to solve this problem will harm his reputation as a Mayor. So he hired you to solve it for him. Would you be able to help Uncle Bob and save him from his misery? Also, make sure not to tell anybody that Uncle Bob has hired you!

Input

Input will have multiple test cases. The first line of the input contains an integer, **T ($1 \le T \le 100$)**, representing the number of test cases. For each test case:

- The first line contains two positive integers **N ($1 \le N \le 10^5$)** and **K ($1 \le K \le 10$)** representing the length of the array **A** and **B**, respectively.
- The second line contains **N** space separated integers $a_1, a_2, \dots, a_n$ ($0 \le a_i \le 2^{31} - 1$).
- The third line contains **K** space separated integers $b_1, b_2, \dots, b_k$ ($0 \le b_i \le 2^{31} - 1$).

You can safely assume that the sum of N over all test cases will not exceed 10^5.

Output

For each case, print the case number followed by the answer modulo **1,000,000,007 (the remainder when divided by 1,000,000,007)**. Please, see the sample for details.

Sample Input	Output for Sample Input
3 1 1 1 1 2 1 1 2 4 2 1 1 3 1	Case 1: 0 Case 2: 3 Case 3: 1

Problem G

Pess Chuzzle

Random unrelated information: In chess, it is illegal to move 2 pawns simultaneously in the first move. If you are doing this, please do me a favor and stop.

You have an $\mathbf{n} \times \mathbf{8}$ chess board. Rows are numbered 1 to 8 from bottom to top. Columns are numbered 1 to n from left to right. Initially there are **n** rooks on the bottom row. You also have a permutation $\mathbf{p_1, p_2, \ldots, p_n}$. A permutation is a sequence of n integers from 1 to n, in which all the numbers occur exactly once.

You have to move the rooks such that the rook which was initially at column **i**, ends up at column $\mathbf{p_i}$ of the bottom row. You also have to minimize the number of moves.

In a move a rook can move horizontally or vertically, but not over another rook. Two rooks can never be on the same cell at the same time.

Input

The first line contains the number of test cases **T**. First line of each test case has an integer **n**. The second line has **n** space separated integers $\mathbf{p_1, p_2, \ldots, p_n}$.

Constraints

- **$1 \le T \le 100$**
- **$1 \le n \le 10^5$**
- **The sum of n over all test cases doesn't exceed 10^5**

Output

For each case, output the minimum number of moves **k**. Each of the next **k** lines contain 4 integers **c1, r1, c2, r2** meaning a rook is moved from column **c1** row **r1** to column **c2** row **r2**. If there are multiple possible outputs, print any. Do **not** print trailing whitespaces in any line.

Sample Input	Output for Sample Input
1 5 2 3 5 4 1	8 3 1 3 2 2 1 3 1 5 1 5 5 3 2 5 2 1 1 2 1 5 5 1 5 1 5 1 1 5 2 5 1

Problem H

Island Invasion

Big Island and Tiny Island are two neighboring islands in the Little Sea. The king of Big Island wants to conquer Tiny island to appease his fragile ego. However, the wind blows strong in Little Sea and the ships are slow and light. Therefore, reaching Tiny island may be difficult and time consuming; in some cases, even impossible. Furthermore, the weather is volatile so the wind direction is hard to predict. Thus the king would like to make plans in advance and find the minimum time to reach Tiny Island from Big Island given various wind velocities.

Big Island and Tiny Island can be described as two **non-intersecting convex** polygons of **n** and **m** vertices respectively. The king will ask you **q** queries. In each query, you will be given a vector $\vec{w} = (w_x, w_y)$ representing the wind velocity and an integer **s,** the maximum speed of Big Island ships. The ships can choose to depart from any point of Big Island and choose to move in any direction they like. If the ships' own velocity is described by $\vec{v}$, then $|\vec{v}|$ must not be greater than **s** and the ships' resulting velocity will be $\vec{v} + \vec{w}$.

Your task is to determine for each query, if it is possible to reach any point of Tiny Island. If so, you also need to find the minimum time needed.

Input

The first line contains the number of test cases **T**.

First line of each test case has two integers **n** and **m**. Each of the next **n** lines contain two integers **x** and **y**, the coordinates of a vertex of Big Island. Each of the next **m** lines contain two integers **x** and **y**, the coordinates of a vertex of Tiny Island. For each polygon, vertices are given in counter-clockwise order.

The next line contains a single integer **q**. Each of the next **q** lines contain three integers, $\mathbf{w_x}$, $\mathbf{w_y}$ and **s.**

Constraints

- **$1 \le T \le 1111$**
- **$3 \le n, m \le 10^5$**
- **The sum of n + m over all test cases doesn't exceed 2×10^5**
- **$-10^9 \le x, y \le 10^9$**
- **$-10^9 \le w_x, w_y \le 10^9$**
- **$0 \le s \le 10^9$**

Output

For each query output a single line.

- If it is not possible to reach any point of Tiny Island, print -1.
- Otherwise print a single real number, the minimum time needed to reach any point of Tiny Island.

Your answer will be considered correct if its absolute or relative error doesn't exceed 10^{-6}. Namely, if your answer is ***a***, and the jury's answer is ***b***, then your answer is considered correct, if $\frac{|a-b|}{max(1, |b|)} \leq 10^{-6}$

Sample Input	Output for Sample Input
1 4 4 0 0 0 1 1 1 1 0 2 2 2 3 3 3 3 2 2 1 1 1 -1 -1 1	0.5857864376 -1

Problem I

Qwiksort

You are given an array containing the integers from **1** to **2n**, each number appearing exactly once. Your goal is to sort the array in increasing order using a special operation called "Qwiksort". In one execution of Qwiksort, you can choose a contiguous range of size **n** and sort them in increasing order in-place. For example, if **n = 3** and the array is **[3, 2, 4, 1, 6, 5]**, and you choose to apply Qwiksort to a contiguous range from second to fourth position, the array becomes **[3, 1, 2, 4, 6, 5]**.

You can apply the Qwiksort operation zero or more times to sort the array. But due to limitations of our server, you can only do it at most **10** times on a particular array. Please print the operations required to sort the array in increasing order. **You do not need to minimize the number of operations. It is guaranteed that it's possible to sort the arrays with at most 10 Qwiksort operations.**

Input

First line of input will contain a single integer **T** denoting the number of independent test cases. Then the descriptions of **T** test cases follow. Each test case will start with a line containing an integer **n**. The next line will contain **2n** space separated integers, the elements of the array to sort.

Constraints

- **$1 \le T \le 40000$**
- **$2 \le n \le 1000$**
- **Sum of n over all test cases does not exceed 2×10^5**

Output

For each test case, first print the number of Qwiksort operations **k (k ≤ 10)** required to sort the array in a line. Then print **k** lines, each line containing two space separated integers **l** and **r** denoting 1-based index specifying the beginning and end of the range that you applied Qwiksort to.

Note that **$1 \le l \le r \le 2n$** and **$r - l + 1 = n$** must be ensured.

Sample Input	Output for Sample Input
2 5 1 2 3 4 5 10 9 8 7 6 2 1 2 3 4	2 6 10 2 6 0

Problem J

Point Table

Once upon a time, in the La Leuga football league, there were only three football teams: Team A, Team B, and Team C. Each team played against each other exactly once. So each team had to play two matches. For a win a team gets 3 points, for a draw each team gets 1 point and for a loss they don't get any points. The point table at the end of the season was as follows.

Table: Point table of La Liga (1927)

Team Name	Points
Team A	6
Team B	4
Team C	1

One day, a football fan was looking at the point table (the table is sorted based on team name) and noticed something strange. Team A has 6 points, which means they must have won both of their matches. However, Team B has 4 points, which means team B must not have lost any matches. This is a contradiction since Team A must have won their match against Team B and Team C. Therefore, the point table is invalid.

So, the La Liga authority decided to investigate further and asked for your help. You are given the points of team A, team B and team C. You have to figure out whether the given table is invalid or not.

Input

The first line will contain a single integer **T ($1 \le T \le 350$)**, denoting the number of test cases. Each test case will have three integers **P_A, P_B**, and **P_C ($0 \le P_A, P_B, P_C \le 6$)**, denoting their points in a single line.

Output

For each of the cases output "**Case <x>: <y>**" in a separate line, where **x** is the case number and **y** is the desired answer. If the point table is invalid, then print "**invalidum**", otherwise print "**perfectus**" (without quotes).

Sample Input

```
3
6 1 1
3 3 3
6 6 6
```

Output for Sample Input

```
Case 1: perfectus
Case 2: perfectus
Case 3: invalidum
```

Problem K

Strategies in Sequential Games

A k player turn based game (also known as sequential game) can be represented as a rooted tree T where each non-leaf node is colored with one of k colors. Each node in the tree represents a game state. Suppose that the current state of the game corresponds to a node u with color i. Then the player i gets to make a move. Each edge connecting node u to one of its children corresponds to a valid move at that game state. If player i gives the move corresponding to the edge (u,v) then the game state will change from u to v. The leaf nodes indicate the terminal states of the game (the game is over and no player can give a move) and thus are colorless.

A strategy for player i is just an assignment of moves for all possible game states that may arise in the game. Note that not all states can arise in a game because by assigning a move in one state we can guarantee that certain states may never arise. Similar to how a game can be represented with a rooted tree T, a strategy for player i can be represented with a (rooted) subtree of that tree T.

More formally, a strategy for a player i given a rooted tree T (where all the non-leaf nodes are colored with integers from 1 to k) is a rooted subtree T' that satisfies the following constraints:

1. T and T' share the same root.
2. Every non-leaf node with color i in T' has exactly one child.
3. For every non leaf node in T' with color not equal to i, all of its children in T are also in T'.

In this problem, you will be given a rooted tree T for such a k player game. For each player, we want to find out how many strategies the player can have.

Input

The first line of the input will contain an integer **t** representing the number of test cases. Each test case is described by 3 lines. On the first line, you will be given a pair of integers **n** and **k**, the number of nodes in the tree and the number of players in the game T. On the second line, you will be given **n** space separated integers $p_1, p_2, p_3, \dots, p_n$ $(0 \le p_i \le n)$ where p_i denotes the parent of the i^{th} node in the tree for all non-root nodes. $p_i = 0$ if **i** is the root. On the third line, you will be given **n** space separated integers $c_1, c_2, \dots, c_n$ $(0 \le c_i \le k)$ where c_i denotes the color of node **i** for all non-leaf nodes. $c_i = 0$ if i is a leaf.

Constraints

- $1 \le k \le n \le 10^6$ and the sum of **n** over all test cases will not exceed 10^6.

Output

For each test case you will print the case number and **k** integers $s_1, s_2, \dots, s_k$ where s_i is the number of strategies for player **i** in the game T modulo 10^9+7

Sample Input	Output for Sample Input
1 17 3 0 1 2 2 3 3 4 6 6 7 7 8 8 10 10 11 11 1 2 3 1 0 2 3 3 0 2 2 0 0 0 0 0 0	Case 1: 1 6 6

Explanation for this problem is available in the PDF version of the problem statement.