# Avian Darts

Time Limit: 2000/1000 MS (Java/Others)

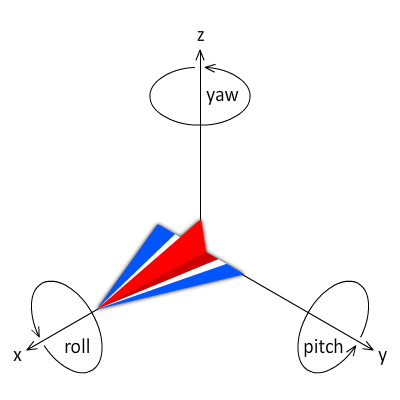
Memory Limit: 524288/524288 K (Java/Others)

Special Judge

## Problem Description

International Combat Aerobatics Competition “Avian Darts” is opening soon and Andrei was selected as a participant.

He and his wingmen planned several awesome aerobatic flights that will certainly delight the judges and award them gold medals. The only problem is to form a formation at the end of competition and that’s why they want to know their exact coordinates and attitudes after performing aerobatics.



He navigates his aircraft by using the joystick and pedals to roll, pitch and yaw. Aerobatic flights consist of courses whose rotational speeds about three axes of the aircraft are given. As you know, direction of velocity always aims the frontwards of the aircraft.

Their aircrafts are 5th generation aircrafts: “Су-57 (Sukhoi-57)”. Therefore, they are not affected by weather and never enter a stall which means that they’re very safe during flight.

You can ignore thrust vectoring, angle of attack, size of aircraft, lift, air resistance, inertia and gravity in this problem.

## Input

Input consists of multiple test cases.

First line of each test case contains a non-negative integer (), the number of courses during the flight.

The next n lines contain 5 integers , , , and .

, , are rotational speeds of rolling, pitching, yawing in degrees per second and then, and denote speed of aircraft in ms-1 and duration in seconds.

Aircraft rolls to the left if while it rolls to the right if . It climbs (dives) if (). It also slides right (left) if ().

, , , ,

It is guaranteed that there will be no more than test cases.

Input is terminated with end of file.

Initially, Andrei’s aircraft is located at and unit vectors towards its front, left and up are , and as shown in the figure.

## Output

Answer for each test case has four lines each of which contains 3 real numbers.

Output x, y and z coordinates on the first line.

On the second line, output x, y and z elements of a unit vector towards the front. Output unit vectors towards the left and up on the third and fourth lines respectively.

Lengths of unit vectors must be equal to 1. Your answer will be acceptable if its absolute or relative error does not exceed .

Output a blank line after each test case.

## Sample Input

|  |
| --- |
| 1  60 0 0 340 3  2  90 0 0 500 1  0 -30 0 200 3  1  0 0 30 100 6 |

## Sample Output

|  |
| --- |
| 1020.000 0.000 0.000  1.000 0.000 0.000  0.000 -1.000 0.000  0.000 0.000 -1.000  881.97186342 -381.97186342 0.000  0.000 -1.000 0.000  0.000 0.000 1.000  -1.000 0.000 0.000  0.000 381.97186342 0.000  -1.000 0.000 0.000  0.000 -1.000 0.000  0.000 0.000 1.000 |

## Hint

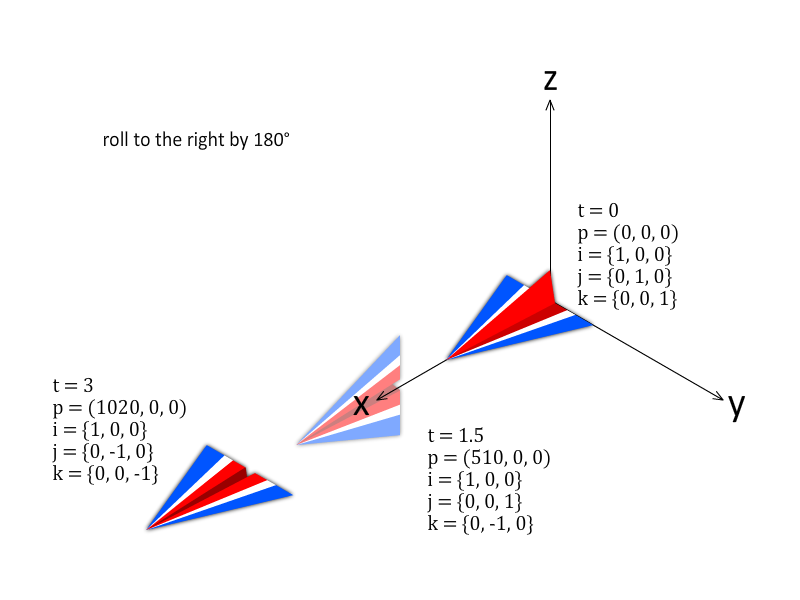


Figure 1. Rolling in the 1st test case

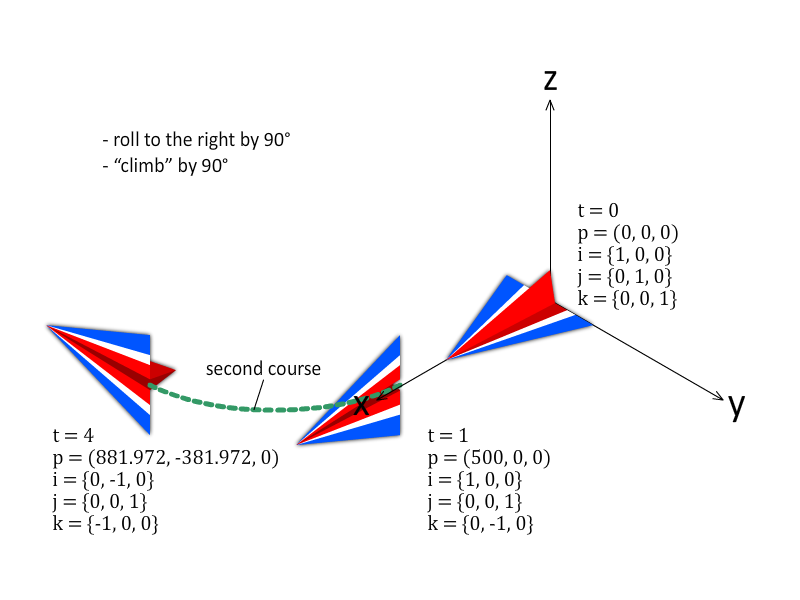


Figure 2. Rolling and pitching in the 2nd test case

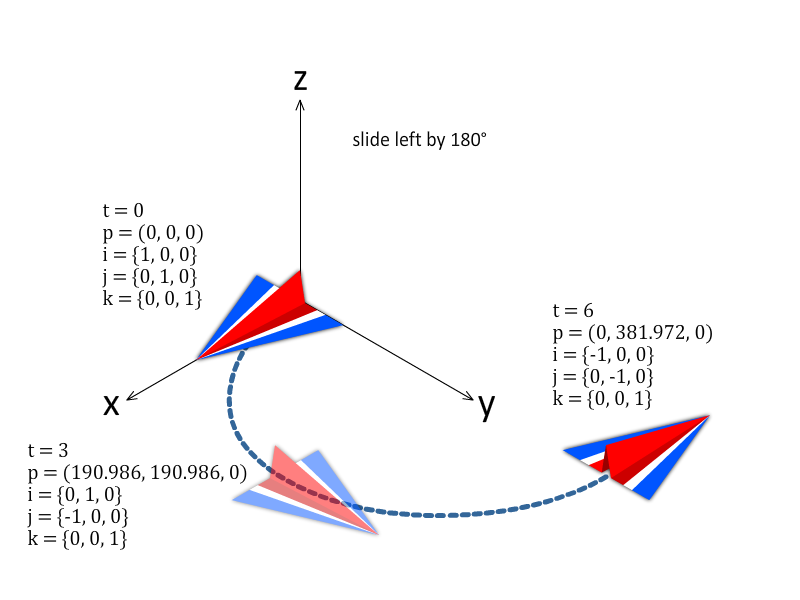


Figure 3. Sliding in the 3rd test case

shows the coordinates of Andrei. , and are unit vectors towards front, left and up of the aircraft.

# Boring Task

Time Limit: 8000/4000 MS (Java/Others)

Memory Limit: 524288/524288 K (Java/Others)

## Problem Description

You are given tasks numbered from 1 to and a single machine able to process the tasks. At any time, the machine can only process a single task.

For every integer between 1 and , you are given 2 integers - , denoting when the th task is created, that means you can only process the th task not earlier than , and , denoting the period of time the machine takes to completely process the th task.

All the tasks are individual, so you can assign tasks in any order. Also, if the machine starts to process a task, then it never stops working until the task is finished. The machine can immediately start the other task (obviously, this task should be available) after finishing a task.

The , for the th task, is defined as the delay time of the task that is equal to the difference between the time when the machine starts to process the th task and . You want to minimize the maximum waiting time of the given tasks. Find the minimum value of maximum waiting time.

## Input

The first line of the input contains a single integer (), the number of test cases. Each test case consists of lines.

The first line of each test case contains a single integer (), the number of tasks. The next line contains integers (), the earliest time the th task can be processed. The 3rd line integers (), the period of time the machine takes to finish the th task. It’s guaranteed that the sum of over all test cases doesn’t exceed

## Output

For each test case, you should print the minimum possible value of the maximum waiting time.

## Sample Input

|  |
| --- |
| 2  3  1 3 2  1 1 1  3  1 3 2  3 2 1 |

## Sample Output

|  |
| --- |
| 0  2 |

# Cookies

Time Limit: 9000/4500 MS (Java/Others)

Memory Limit: 524288/524288 K (Java/Others)

## Problem Description

Elsa’s elder brother Eric has got cookies in his icebox and each cookie has a special number written on it. Let’s denote the number written on the th cookie by . is defined as follows:

Here, states the median value of divisors of . Let has divisors, then is ⌈k/2⌉th smallest divisor. For example, , , .

One day, Eric opened the icebox and recognized that some of his cookies are missing. His sister Elsa had eaten some of them.

On the first day, she ate cookies that have multiples of as indexes and then, re-indexed them starting from according to their original order. In the same way, she ate cookies with multiples of as indexes on the th day and re-indexed them. If there were less than cookies left, she ate none of them. Elsa continued to do so for days.

As Elsa is keen on math and Eric didn’t want to blame his sister, Eric asked her to find out the number written on the th of remaining cookies.

## Input

The first line of the input contains an integer , denoting the number of test cases. The first line of each test case contains 3 positive integers , and . The next line contains integers .

## Output

For each test case, output one line containing the answer. If the th cookie doesn’t exist, then output -1 in a single line.

## Sample Input

|  |
| --- |
| 2  25 5 9  3 5 8 12 10  25 5 2  3 5 8 12 10 |

## Sample Output

|  |
| --- |
| 31  2 |

# Distinct Sub-palindromes

Time Limit: 5000/2500 MS (Java/Others)

Memory Limit: 524288/524288 K (Java/Others)

## Problem Description

is a string of length . consists of lowercase English alphabets.

Your task is to count the number of different with the minimum number of distinct sub-palindromes. Sub-palindrome is a palindromic substring.

Two sub-palindromes and are distinct if their lengths are different or for some (), . For example, string “aaaa” contains only 4 distinct sub-palindromes which are “a”, “aa”, “aaa” and “aaaa”.

## Input

­­The first line contains an integer (), denoting the number of test cases.

The only line of each test case contains an integer ().

## Output

For each test case, output a single line containing the number of different strings with minimum number of distinct sub-palindromes.

Since the answer can be huge, output it modulo .

## Sample Input

|  |
| --- |
| 2  1  2 |

## Sample Output

|  |
| --- |
| 26  676 |

# Fibonacci Sum

Time Limit: 4000/2000 MS (Java/Others)

Memory Limit: 524288/524288 K (Java/Others)

## Problem Description

The Fibonacci numbers are defined as below:

()

Given three integers , and , calculate the following summation:

Since the answer can be huge, output it modulo .

## Input

­­The first line contains an integer (), denoting the number of test cases. Each test case contains three space separated integers in the order: , , ().

## Output

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

## Sample Input

|  |
| --- |
| 2  5 1 1  2 1 2 |

## Sample Output

|  |
| --- |
| 12  2 |

# Finding a MEX

Time Limit: 6000/3000 MS (Java/Others)

Memory Limit: 524288/524288 K (Java/Others)

## Problem Description

Given an undirected graph . All vertices are numbered from to . And every vertex has a value of . Let . Also, equals MEX(minimum excludant) value of . A MEX value of a set is the smallest non-negative integer which doesn’t exist in the set.

There are two types of queries.

Type 1: **1** **u** **x** – Change to ()**.**

Type 2: **2** **u** – Calculate the value of .

For each query of type 2, you should answer the query.

## Input

The first line of input contains a single integer () denoting the number of test cases. Each test case begins with a single line containing two integers (), () denoting the number of vertices and number of edges in the given graph.

The next line contains integers and th of them is a value of ().

The next lines contain edges of the graph. Every line contains two integers , meaning there exist an edge between vertex and .

The next line contains a single integer () denoting the number of queries.

The next lines contain queries described in the description.

## Output

For each query of type 2, output the value of in a single line.

## Sample Input

|  |
| --- |
| 1  5 4  0 1 2 0 1  1 2  1 3  2 4  2 5  5  2 2  1 2 2  2 2  1 3 1  2 1 |

## Sample Output

|  |
| --- |
| 2  2  0 |

# Hunting Monsters

Time Limit: 6000/3000 MS (Java/Others)

Memory Limit: 524288/524288 K (Java/Others)

## Problem Description

A group of monsters has invaded into the town. You, as a guardian of the town, should protect the town from the attack of those monsters. As you are a smart warrior, you are going to eliminate of them so that all the remaining monsters are scared and let them quit the town themselves.

In order to defeat a monster, you should consume unit of energy while you restore unit of energy after defeating it. If your energy before attack is less than , you will be exterminated and the town will be ruined. These values and depends on the monster, so they may be distinct for different monsters. You may arbitrarily choose monsters to be killed. Note that you can defeat monsters in any order. You want to know the minimum amount of initial energy needed to kill at least monsters. You are required to calculate the answer for all .

## Input

The first line of the input contains a single integer (), denoting the number of test cases. Each test case consists of lines.

The first line of each test case contains a single integer (), the number of monsters. The next line contains integers (), the amount of energy needed to defeat the th monster. The third line contains integers (), the amount of energy you will gain after killing the th monster. It is guaranteed that the sum of over all test cases does not exceed

## Output

Let be the least initial energy needed to defeat monsters.

You should calculate all for . In order to avoid huge output, you are only required to get the following.

For each test case, print a single integer in a single line.

## Sample Input

|  |
| --- |
| 2  3  2 3 1  1 5 4  4  1 2 3 4  1 1 2 3 |

## Sample Output

|  |
| --- |
| 6  30 |

## Hint

For the first test case, .

For the second test case, . For example, if you have 4 units of initial energy, you can kill the 4th, 3rd, 2nd and then 1st monster in order. First you kill 4th monster, your energy will be . Next, you attack 3rd monster, your energy will be. After killing them, you kill 2nd one, then energy will become . Finally, you can kill 1st one since your energy is not less than .

# Integral Calculus

Time Limit: 7000/3500 MS (Java/Others)

Memory Limit: 524288/524288 K (Java/Others)

## Problem Description

Ely has just faced a complicated math problem. So she now needs assistance of you – a talented programmer. The problem is as below.

Given a positive integer , calculate the following value:

We can prove that it is a rational number. Let’s denote it by , where and Since these numbers can be very huge for large , you are only to print the value . Here is multiplicative inverse of modulo .

## Input

The first line of the input contains a single integer , the number of test cases. Each of the next lines contains an integer , which is described above.

## Output

For each test case, you should print the answer in a single line.

## Sample Input

|  |
| --- |
| 1  1 |

## Sample Output

|  |
| --- |
| 400000006 |

# Leading Robots

Time Limit: 2000/1000 MS (Java/Others)

Memory Limit: 524288/524288 K (Java/Others)

## Problem Description

Sandy likes to play with robots. He is going to organize a running competition between his robots. And he is going to give some presents to the winners. Robots are arranged in a line. They have their initial position (distance from the start line) and acceleration speed. These values might be different. When the competition starts, all robots move to the right with speed:

Here is acceleration speed and is time from starting moment.

Now, the problem is that, how many robots can be the leader from the starting moment?

Here leader means the unique rightmost robot from the start line at some moment. That is, at some specific time, if a robot is rightmost and unique then it is the leading robot at that time. There can be robots with same initial position and same acceleration speed.

The runway is so long that you can assume there's no finish line.

## Input

The input consists of several test cases. The first line of input consists of an integer , indicating the number of test cases. The first line of each test case consists of an integer , indicating the number of robots. Each of the following  lines consist of two integers:  indicating a robot's position and its acceleration speed.

## Output

For each test case, output the number of possible leading robots on a separate line.

## Sample Input

|  |
| --- |
| 1  3  1 1  2 3  3 2 |

## Sample Output

|  |
| --- |
| 2 |

# Math is Simple

Time Limit: 7000/3500 MS (Java/Others)

Memory Limit: 524288/524288 K (Java/Others)

## Problem Description

Here is a simple math. You are given an integer . Your task is to calculate the following:

## Input

­­The first line contains an integer (), denoting the number of test cases. The only line of each test case contains an integer ().

## Output

In fact, the answer can be represented as a fraction , where and Since these numbers can be very huge for large , you are only required to print .

For each test case, print a single line with the calculated value.

## Sample Input

|  |
| --- |
| 3  2  3  4 |

## Sample Output

|  |
| --- |
| 499122177  1  831870295 |

## Hint

Real answer for sample test cases are respectively.

# Minimum Index

Time Limit: 3000/1500 MS (Java/Others)

Memory Limit: 524288KB / 524288KB (Java/Others)

## Problem Description

Let be a string of length . For any integer between to , the -th suffix of is defined as . For example, the 4th suffix of “contest” is “test” and the 1st suffix of “suffix” is “suffix” itself.

One can consider lexicographically smallest suffix of . Let the -th suffix is lexicographically minimum among all suffixes of . Conveniently, let’s call such *minimum index of* . Given a string , your task is to calculate the sum of (minimum index of ) over all . This value may be too large, you are only required to print it modulo .

## Input

The first line of the input contains a single integer (), the number of test cases. Each of the next lines contains a non-empty string with at most lowercase English letters. The total length of the input string over all test cases doesn’t exceed .

## Output

For each test case, you should print the answer in a single line.

## Sample Input

|  |
| --- |
| 1  aab |

## Sample Output

|  |
| --- |
| 1238769 |

## Hint

The minimum index of “a”, “aa”, “aab” is 1, 2, 1 respectively. So the answer is

(1+2·1112+11122)mod(109+7)=1238769.

# Mow

Time Limit: 2000/1000 MS (Java/Others)

Memory Limit: 524288/524288 K (Java/Others)

Special Judge

## Problem Description

Nick is in charge of managing a lawn which can be represented as a convex polygon.

Since he didn’t manage the lawn for a long time, the grass has grown up too long and he doesn’t like it. So, he decided to mow the grass in the lawn.

He can either mow the grass by hand or hire a mowing machine.

Mowing by hand costs euro(s) per unit area and hiring a mowing machine costs euro(s) per unit area. Unfortunately, the circle-shaped mowing machine must not get out of his lawn while mowing, that is, any point of the machine must be strictly inside the lawn or on the border. The machine cuts all the grass in its circle.

Any grass is considered to be cut only once even though the machine passed over it several times.

Find out the minimal amount of money he needs for mowing his lawn.

## Input

­­The first line contains an integer , denoting the number of test cases.

The first line of each test case contains two integers () and (), which is radius of the machine.

The next line contains two integers and ().

Following n lines contain two integers and (), coordinates of points representing his lawn in order of traversal.

It is guaranteed that r is not equal to the radius of inscribed circle.

## Output

Output a single line containing the minimal amount of money you need.

Your answer will be considered correct if its absolute or relative error doesn’t exceed .

## Sample Input

|  |
| --- |
| 1  4 1  1 0  0 0  4 0  4 4  0 4 |

## Sample Output

|  |
| --- |
| 0.858407346410 |