ACM-ICPC Live Archive

# 7827 WSI Extreme

Internet has made our life very easy. When you would like to visit some place for vacation, you can book your flights at `canoe.com` and apartment at `waterbnb.com`. Are you done? Well, if you have Mr Mosquito or Mr Macho-the-Cynophobic (Cynophobia = fear of dogs) then — no! You need to be very careful while selecting apartments. You need to make sure there are enough washrooms! Probably if they did not have the need of food, they would have loved to spend their whole life in washroom. Waterbnb has an excellent idea to solve this problem. They are planning to provide an interface where one can input the approximate times all the guests spend in the washroom, it will compute a washroom-satisfaction-index ($WSI$) for each of the apartments and thus helping one to choose apartments. Your task is to help computing $WSI$.

$WSI$ for a person is the time he needs to wait plus the time he spends in the washroom. So if 2 persons spend 2 minutes and 5 minutes in a washroom respectively, and a person spends 3 minutes after the previous two persons, then $WSI$ for the third person would be $2 + 5 + 3 = 10$ minutes.

Now how to compute $WSI$ of an apartment, given the approximate time-spends of the guests? First you need to assign the guests to the washrooms — who goes to which washroom and after whom. For that assignment, you can compute $WSI$ for all of the guests. $WSI$ of that assignment is the sum of $WSI$ of all the guests. Minimum $WSI$ among all the assignments is the $WSI$ of the apartment. Let's give an example.

Suppose there are 3 guests (A, B and C). Approximate time spends of the guests in the washroom are: A: 3 minutes, B: 5 minutes, C: 8 minutes.

Let us compute $WSI$ for 2 apartments:

- Apartment 1: Three washrooms. All of them goes to separate washroom. $WSI = 3+5+8 = 16$.
- Apartment 2: Two washrooms. Let us show two possible assignments:
- Assignment 1: Washroom1 (First A, then B), Washroom2 (C). $WSI = (3 + (3 + 5)) + 8 = 19$.
- Assignment 2: Washroom1 (First B, then C), Washroom2 (A). $WSI = (5 + (5 + 8)) + 3 = 21$.

There may be some more assignments. But out of all, assignment 1 is the best. So $WSI$ of the two washrooms apartment would be 19.

However, each person may require different time in washroom for each day depending on the circumstance. Thus, your program must be able to update the washroom usage time for each guest. When the time of any guest is updated, you must compute $WSI$ again.

## Input

First line contains number of test cases $T$ ($T \leq 16$)

In the first line of each test case, there are two positive integers: Number of guests $G$ ($G \leq 50000$) and number of washrooms $W$ ($W \leq 50000$).

Second line of the test contains $G$ positive integers, each not more than 100 000. The $i$-th of these $G$ numbers denote the approximate time taken by the $i$-th guest in a washroom.

In the next line there will be a positive integer $Q$ ($Q \leq 10000$) denoting number of updates. Hence $Q$ updates will follow. Each update consists of 2 positive integers in a line. First number will denotes a guest number (1-indexed and at most $G$) and the second number (at most 100 000) will denote the updated approximate time by the guest.

## Output

For each test case, print the case number in a line. Then in following $Q$ lines, print $WSI$ of the apartment after each update. For details please consult sample input output.

## Sample Input

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

## Sample Output

```
Case 1:
15
8
Case 2:
52
31
```

ACM-ICPC Live Archive

# 7828 Average

Suppose 4 teachers are grading the presentation of a student and all their given marks are integers. The actual score is computed by taking arithmetic average of these 4 marks. Even though the marks are integers, the average can still be a fraction. But in some cases the average can also be an integer and in some rare cases the one of given marks can be equal to the average. Such rare incidents are called *matching events*. Considering all possible marks given by all teachers, you will have to count the total number of *matching events*. If the average mark matches with marks given by more than 1 teacher then for each match a *matching event* should be counted.

## Input

There are at most 1001 test cases. The description of each test case is given below:

The input for each test case consists of two integers $N$ $(2 \leq N \leq 60)$ and $fullmarks$ $(1 \leq fullmarks \leq 200)$. Here $N$ denotes the number of teachers who are grading the student and $fullmarks$ denotes the maximum possible mark that any of the teachers can give. The minimum mark that a teacher can give is always zero.

Input is terminated by a line containing two zeroes.

## Output

For each test case produce one line of output. This line contains the value $M\%1000000007$ where $M$ is the total number of matching events for the given input.

### Illustration of 2nd Sample Input

| Marks given by three teachers | Average | Why it is a matching event? |
|---|---|---|
| $\underline{0}$ 0 0 | 0 | Average matches with mark given by 1st teacher |
| 0 $\underline{0}$ 0 | 0 | Average matches with mark given by 2nd teacher |
| 0 0 $\underline{0}$ | 0 | Average matches with mark given by 3rd teacher |
| $\underline{1}$ 1 1 | 1 | Average matches with mark given by 1st teacher |
| 1 $\underline{1}$ 1 | 1 | Average matches with mark given by 2nd teacher |
| 1 1 $\underline{1}$ | 1 | Average matches with mark given by 3rd teacher |

So these are the 6 matching events

## Sample Input

```
4 100
3 1
0 0
```

## Sample Output

```
1373732
6
```

ACM-ICPC Live Archive

# 7829 Big Bang

A Big Bang has happened at coordinate (0,0,0) at time $T = 0$. Specifically, a "*Fixed Velocity Lattice Big Bang*" had happened. Just before the Big Bang, every particle was at (0,0,0) with zero volume and infinite density. After the Big Bang the particles propagate in a Cartesian 3D space. The movements of particles follow the following rules:

1. The direction of particles are always along a straight line (in 3D space), starting at (0,0,0).
2. All particles start their journey at the time $T = 0$. Two particles can have same direction but in that case they cannot have same speed. In other words no two particles occupy the same place at the same time.
3. The speed of each particle is uniform.
4. The number of particles can be considered unlimited.
5. The velocity of the particles are such that the $x$, $y$ and $z$ components are always on the strictly positive side of respective axis.
6. All particles are so small that they can be considered as points.
7. All the particles is not visible. A particle is visible if it reaches a lattice point (all three coordinates are integers) at an integer time (Measured in micro-seconds) after the bigbang. This is because particle sensors are only at lattice points and they are activated once in every microsecond.

Now if sensors are placed in all lattice points that have strictly positive $x$, $y$ and $z$ values not exceeding a positive integer $N$, and measures are taken up to $N$-th microseconds after bigbang, maximum how many distinct particles can be detected? For example the particle that is detected at (1,1,1) after 1 microsecond of Big-bang and the at (2,2,2) after 2 microsecond after Big-bang are basically the same particle as both have same direction and speed.

## Input

The input contains 5 positive integers as input. All these are possible values of $N$: First four integers do not exceed 20 000. The last line contains the integer '80000'.

## Output

For each integer in the input produce one line of output which denotes the maximum number of distinct particles that can be detected.

**Remarks:** The output of 5-th case is not shown for obvious reasons. '?' denotes a single digit

## Sample Input

```
1
1
10000
10000
80000
```

## Sample Output

```
1
1
9239427676877311
9239427676877311
????????????????????
```

ACM-ICPC Live Archive

# 7830 Find C

Two of your friends Alpha and Beta are at $A$ and $B$, two integer coordinates in 2D plane. You want to take their picture from another integer coordinate $C$ fulfilling following conditions:

1. $C$ has to be different from $A$ and $B$.
2. There is no integer coordinate point on the line segment $AC$ other than its end points.
3. There is no integer coordinate point on the line segment $BC$ other than its end points.
4. Triangle $ABC$ must have positive area, that is, $C$ must not lie on the line going through $A$ and $B$.
5. There is no integer coordinate point strictly inside triangle $ABC$.

Don't forget this is the era of digital photography where people don't take only one snap. So you also need to find out $K$ such $C$ for each $A$ and $B$ from where you will take pictures.

A point $P$ having coordinate $(x, y)$ is called integer coordinate if both $x$ and $y$ are integers.

## Input

First line of the input contains number of test cases $T$ $(1 \leq T \leq 1000)$.

In following $T$ lines, there are 5 integers: $A_x, A_y, B_x, B_y, K$. Coordinates of $A$ and $B$ are $(A_x, A_y)$ and $(B_x, B_y)$ respectively. $A$ and $B$ are distinct points. $(-10^9 \leq A_x, A_y, B_x, B_y \leq 10^9$, $0 \leq$ sum of all $K \leq 20000)$

## Output

For each case print $K$ lines each containing coordinate of $C$ in: '$C_x$ $C_y$' format. You may assume that there are at least $K$ such $C$ points. You may output the $C$ points in any order, but these $K$ points has to be distinct. Also all the $C_x$ and $C_y$ have to be between $-10^{14}$ and $10^{14}$. For details of output format please consult the sample input output.

## Sample Input

```
2
1 1 10 1 2
0 0 -10 -10 2
```

## Sample Output

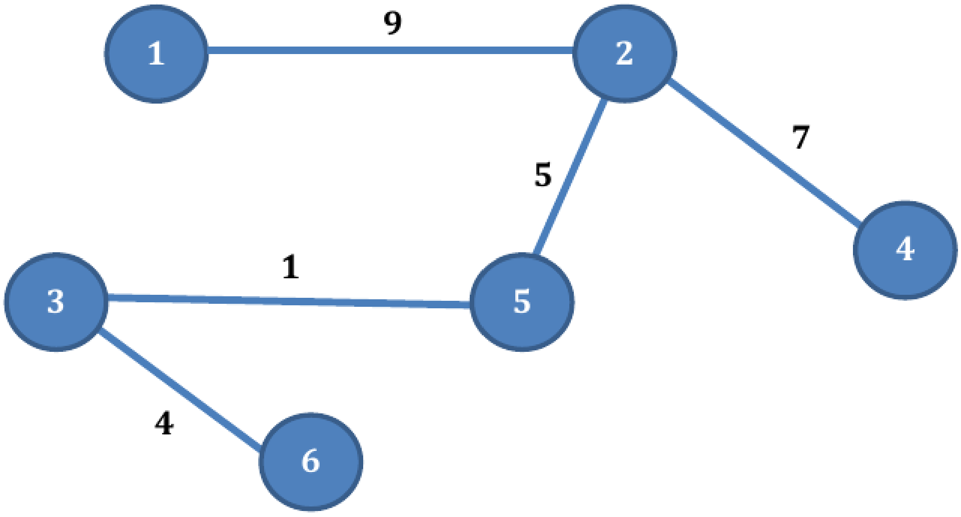
```
2 2
3 2
-1 0
-2 -1
```

ACM-ICPC Live Archive

# 7831 ACM Tax

Mr. Peter lives in the magical lands of ACM City. The city consists of $N$ districts and $N-1$ two-way roads. Each road connects a pair of districts, and for each road we know the length of the road. All districts are connected to each other, meaning Mr. Peter can always travel from any district $A$ to any district $B$.

The ACM City has a very special rule for travelling. If anyone wants to travel from district $A$ to district $B$, they must pay the tax equal to the **median of the length of all roads from district $A$ to district $B$**. For example, this ACM City of 6 districts.



All circles represent districts and lines represent roads connecting two districts. Numbers inside the circles are the number of each district and numbers above the lines are the length of each road. If Mr. Peter travels from district 1 to district 3, the length of all roads would be 1, 5, and 9, therefore he must pay the tax of 5 dollars. However, if Mr. Peter travels from district 6 to district 4, he must pay 4.5 dollars, for the length of the roads are 1, 4, 5, and 7.

Mr. Peter is a very curious person. He wants to know the tax he needs to pay to travel from any two districts. However, Mr. Peter is too lazy to calculate the tax himself, so he asks you, the best programmer in the city, to calculate the tax Mr. Peter needs to pay from district $A$ to district $B$.

*Note :* The 'Median' is the middle value of a sorted list of numbers. For example, the median of the list 1, 5, 7, 7, 9, 10, 16 is 7. If the amount of numbers in the list is even, the median is the sum of two middle numbers divided by 2. For example, the median of the list 1, 5, 7, 9, 10, 17, 28, 30 is (9+10)/2, which is 9.5.

Given the ACM City of $N$ districts, for each question, find the median of the length of all path from district $A$ to district $B$.

## Input

The first line contains number of test case $T$ ($T \leq 15$). Then for each test case:

The first line contains one integer $N$ ($2 \leq N \leq 50000$) — the number of districts.

The next $N-1$ lines contain three integers $U_i$, $V_i$, $W_i$ ($1 \leq U_i, V_i \leq N$, $U_i \neq V_i$, $1 \leq W_i \leq 100000$) — the roads connecting district $U_i$, and $V_i$ with the length $W_i$.

The next line contains one integer $Q$ ($1 \leq Q \leq 100000$) — the number of questions Mr. Peter wants to ask.

The next $Q$ lines contain two integers $A_i, B_i$ ($1 \leq A_i, B_i \leq N$, $A_i \neq B_i$) — the questions Mr. Peter wants to ask how much he needs to pay to travel from district $A_i$ to district $B_i$.

## Output

Print $Q$ lines. In each line contains one real number with one digit after the decimal point — the tax Mr. Peter needs to pay to travel from district $A_i$ to district $B_i$.

## Sample Input

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

## Sample Output

```
5.0
4.5
4.0
```

ACM-ICPC Live Archive

# 7832 Dictionary Game

We all know the famous game scrabble, where we want to make words using tiles. We will play a game with words. But it is reverse of scrabble. Here instead of making words we want to destroy them. It is a two player game.

The game starts with a dictionary of words. At each move, the current player chooses a word of his choice (not necessarily from the dictionary). The constraints for choosing the word is, it must match the prefix of any (one or more) of the current dictionary words.

When a player chooses a word as its move, all the dictionary words that has a prefix equal to the chosen word, are split into two parts:

1. The matched part without the last character of the chosen word.
2. The unmatched part starting from the last character of the chosen word.

We remove all the unmatched part from the dictionary and keep only the matched part in the dictionary. The words that don't have matched prefix with the chosen word are kept intact in the dictionary. In this situation the next player makes its move in the same fashion. If the chosen word is of length 1, then all the dictionary words starting with that character will be removed.

For example, if we have a dictionary consisting the following words:

```
bangladesh
bangalore
band
bandana
```

Now if the first player chooses the word "bang" then after the move the dictionary will become:

```
ban
ban
band
bandana
```

A player loses, when it cannot make a valid move. Given the dictionary of words, who will win if both players played optimally? To make it more interesting, can you find out who will win if you update the dictionary by adding new words?

Formally, you will be given few operations where you need to add a new word to the dictionary. After adding you need to calculate the winner between player 1 and 2 for the current dictionary. These added words are permanent. Means each of these added words will remain in the dictionary for that particular test case.

## Input

First line consists of an integer $T$, which is the number of test cases ($T \leq 10$).

Each case will start with an integer $N$ ($0 < N \leq 50000$), the number of words in the dictionary.

Next $N$ lines will contain dictionary words DW. Length of each DW will be less than or equal to 40.

Next line will contain an integer number $Q$, the number of operations ($0 < Q \leq 50000$).

Next $Q$ lines will contain query words QW. Each QW is the word that you now want to add to the current dictionary. Let's call these "Query words". Length of each $QW$ will be less than or equal to 40. All the dictionary words and query words will only consist of lower case letters.

## Output

For each query word QW, you have to find out who will win, if you add QW to the current dictionary. First print the case number, then for each query print '1' or '2' based on whether player 1 or 2 will win.

## Sample Input

```
2
4
bangladesh
bangalore
band
bandana
2
egg
apple
1
orange
2
cat
tiger
```

## Sample Output

```
Case 1:
2
1
Case 2:
1
2
```

ACM-ICPC Live Archive

# 7833 Binary Strings

A young boy got really curious about binary strings. This string contains only 1s and 0s hence the name binary. His particular interest was about those strings for which no two ones are side by side. Specifically he wanted to know the number of strings of a certain length that consisted of only ones and zeroes and there are no two consecutive ones.

After solving this problem, the young boy got even more curious. Now he wants to know the number of binary strings which satisfies the following properties:

- The length of the string is between $L$ and $R$, inclusive ($1 \leq L \leq R \leq 10^{18}$)
- The length of string is divisible by an integer $K$ ($3 \leq K \leq 10^9$)
- It is a binary string with no two consecutive ones.

Now can you help him to find out the number of strings that satisfies the above conditions? Since the number can be huge, you need to print it modulo 1 000 000 007.

## Input

The first line is an integer $T$ ($1 \leq T \leq 10000$), the number of tests. In the next $T$ lines there are three integers $L$, $R$ and $K$.

## Output

Print $T$ lines, in each line print the case id and the result modulo 1 000 000 007. See the samples for more details.

**Explanation:**

For the first case some example strings are “101”, “000”, “010” “101001”, “000010000”, etc.

## Sample Input

```
2
1 10 3
1 10 5
```

## Sample Output

```
Case 1: 115
Case 2: 157
```

ACM-ICPC Live Archive

# 7834 Witcher Potion

In a mythical world, there is a fighting contest where each contestant fights a monster one by one and needs to defeat monsters as many as possible until the contestant is knocked out. Among fighter tribes, a Witcher tribe has an edge over others since they can take advantage from energy-boosting, but poisonous, potions. These potions instantly increase his or her energy and as long as a Witcher does not run out of energy, he will not be knocked out. Unfortunately, these potions contain toxicity and he can drink at most one potion bottle once he defeats each monster.

Geralt, a legendary Witcher, enters this tournament with $N$ potion bottles. Each bottle has its own strength and toxicity level. If he drinks a potion with strength $E$, his energy increases by $E$ units, but it will not exceed 100. Potion with toxicity level $P$ increases toxicity level in his blood by $P$ units. If the toxicity level in blood reaches 100, Geralt will be knocked out. Therefore, he will never let that happens.

However, his body can quickly release blood toxicity. In $T$ minutes, his blood toxicity level reduces by $T$ units too. This allows Geralt to safely drink another potion bottle when he defeats a monster.

Since Geralt knows exactly how much energy and time he needs to spend to defeat a monster, he knows that he can plan how to drink potions that will let him defeat many monsters and become the winner of this contest. Nevertheless, calculation seems not to be his strength.

Your task is to analyze potion and monster data. Then, tell Geralt the maximum number of monsters he can defeat before he will be knocked out. Assume that all monsters are the same (need the same amount of energy and time to be defeated). Furthermore, if the needed energy is $K$ units and Geralt has $K$ or less energy units, Geralt will not be able to defeat a monster.

## Input

The first line contains a positive integer $C \leq 20$, representing the number of test cases. For each test case, the input is as follows:

**Line 1** contains positive integers $K$ and $M$ where $K$ and $M$ are the amount of energy and time that Geralt needs to spend for each monster. Also, $10 \leq K \leq 80$ and $M \leq 100$.

**Line 2** contains $N$, the number of potions Geralt has where $1 \leq N \leq 8$.

**Line 3** contains positive integers $E_1, E_2, E_3, \ldots, E_N$, the potion strength of bottles 1 to $N$.

**Line 4** contains positive integers $P_1, P_2, P_3, \ldots, P_N$, the potion toxicity level of bottles 1 to $N$.

$(1 \leq E_i, P_i \leq 100)$

## Output

For each test case, your program will print the maximum number of monsters that Geralt can defeat, one line for one case.

**Note:** at the beginning of the contest, Geralt has 100 energy units and 0 blood toxicity level. Also, Geralt may choose not to drink any potion when he defeats a monster, even though he still has some left in his inventory. Once he has no potion, he needs to rely solely on the rest of his energy

## Sample Input

```
4
30 20
6
30 30 50 100 40 50
10 30 80 90 50 60
20 30
6
10 30 80 90 50 60
30 30 50 100 40 50
50 30
8
10 60 70 20 70 50 60 40
15 40 50 30 60 70 40 75
30 50
8
15 40 50 30 60 70 40 75
10 60 70 20 70 50 60 40
```

## Sample Output

```
10
16
7
15
```

ACM-ICPC Live Archive

# 7835 Sky Tax

New Bangkok is a newly built province of Thailand that is floating in the sky. In order for the province to be able to float, each city is supported by a spaceship. Because all spaceships are moving in same direction and velocity, the structure of the province stays still.

Cities are connected by sky ways. A sky way is a floating road connects two cities of New Bangkok together, so a citizen can commute from a city to another city by sky ways. With well urban planning, it is guaranteed that one can commute from any city to all other cities. Also, from a city to another city, there is only one simple route, i.e., no skyway that is used twice.

Because it is new, the province changes its capital city rapidly. This province also has a strange rule, this is, a city $A$ must handle tax from a city $B$ if a route from $B$ to the capital city must pass through $A$. So it could be that a city have to handle taxes of many cities.

In this problem, we provide that structure of New Bangkok, an initial capital city, and number of queries. For each query, we ask you to either (1) move the capital city of the province to city $R$. or (2) given a city $X$, tell us how many cities that $X$ has to handle taxes.

## Input

The first line contains an integer $T \leq 10$, number of test cases. Then for each test case:

The first line of the test case contains three integers $1 \leq N \leq 100000$, $1 \leq Q \leq 50000$, $1 \leq R \leq N$, denote number of cities, number of queries, and an initial capital city in respective order.

Each of next $N - 1$ lines gives an information of a sky way. It contains two integers $1 \leq A \leq N$ and $1 \leq B \leq N$, $A \neq B$, there is a sky way connects $A$ and $B$.

Each of next $Q$ lines gives an information about query. It contains two integers $0 \leq S \leq 1$ and $1 \leq U \leq N$.

If $S$ is 0, then it asks you to move the capital city to city $U$. Otherwise, it asks you to compute number of cities that $U$ needs to handle taxes.

## Output

You must print answer of test cases and queries in the order given in the input. For test case $I$, you must start with a line containing ‘Case #$I$:’ (without double quotes).

Then, for each query that needs an answer, print the answer in a line.

## Sample Input

```
2
5 5 1
1 5
3 4
3 5
2 1
1 1
1 2
0 5
1 5
1 3
1 5 1
```

```
1 1
1 1
0 1
1 1
1 1
```

## Sample Output

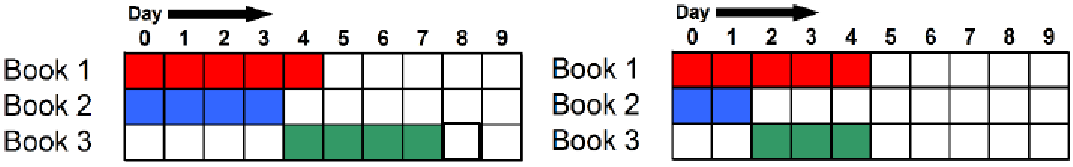
```
Case #1:
5
1
5
2
Case #2:
1
1
1
1
```

ACM-ICPC Live Archive

# 7836 Printing Press

A book publisher needs to print $N$ books in $X$ days. Though normally books can be published in parallel, but for some reason the publisher imposes restriction on some books i.e. you cannot start publishing on *Harry Potter 2* before publishing *Harry Potter 1*. There are $M$ restrictions in $(u, v)$ format, which means the printing process of the $v$-th book can start after the $u$-th book is published. It takes $A_i$ days to publish the $i$-th book and the cost of this publishing is $C_i$. But if the publisher provides the printing presses additional resources like printing machine then the publication time of the $i$-th book can be reduced by $R_i$ days, but the total time must not be less than $B_i$ days ($A_i - R_i \geq B_i$), that means the book will take $A_i - R_i$ days to publish instead of Ai days. But to reduce a single day the publisher has to pay extra $D_i$ amount of money.

For the 3rd sample test case: In normal scenario, it will take 8 days to publish all the three books. But if we reduce 2 days for *Book2* and 1 day for *Book3* then, all of the three books can be published within 5 days.



*Normal scenario*

*Better scenario*

You have to calculate the minimum amount of money the publisher has to spend to print all the $N$ books within $X$ days.

## Input

First line of the input contains an integer $T$ ($T \leq 300$), denoting the number of test cases to follow. Each test case will maintain the following format:

$N$ $X$
$A1$ $A2$ $\dots$ $AN$
$B1$ $B2$ $\dots$ $BN$
$C1$ $C2$ $\dots$ $CN$
$D1$ $D2$ $\dots$ $DN$
$M$
$u1$ $v1$
$\dots$
$uM$ $vM$

Where $A_i$ ($1 \leq A_i \leq 1000000$), $B_i$ ($1 \leq B_i \leq A_i$), $C_i$ ($1 \leq C_i \leq 1000000$), $D_i$ ($0 \leq D_i \leq 100$) has been described above. $M$ ($0 \leq M \leq N * (N-1)/2$) is the number of restrictions and integers $u_i$ and $v_i$ ($1 \leq u_i, v_i \leq N$) denote that the publishing of $v_i$-th book cannot start before $u_i$-th book has been published. There will be no cyclic restriction in the input.

Constraint on $N$:

- 85% test cases: $N \leq 30$
- 3 test cases: $N = 200$
- Rest of the test cases: $30 \leq N \leq 100$

## Output

First line of each test case will contain the test case number and '`Impossible`', if it is not possible to print all the $N$ books within $X$ days. Otherwise output will contain $2 * N + 1$ integers, first of them is the minimum cost to publish all the $N$ books within $X$ days and next $N$ pairs of integers are $S_i$ and $R_i$ $(0 \leq S_i, R_i \leq 10000000)$, where $S_i$ is the starting day of publishing of $i$-th book and $R_i$ is the number of days reduced to print the $i$-th book. If there are more than one possible answers print any one of them.

### Explanation

In the first test case, without any extra resources, it will take 5 + 3 = 8 days to complete both the books. But the publisher needs to print the books within 6 days. With additional resources, first book will start on 0-th day and continue till $0 + 4 - 1 = 3$rd day and the second book will start on 4th day and continue till $4 + 2 - 1 = 5$th day and it will cost $1 * 1 + 1 * 1 = 2$ unit extra and the total cost is initial cost + extra cost = 3 + 2 = 5.

## Sample Input

```
3
2 6
5 3
3 2
1 2
1 1
1
1 2
2 5
5 3
4 2
1 2
1 1
1
1 2
3 5
5 4 4
5 2 2
1 1 1
0 1 2
1
2 3
```

## Sample Output

```
Case 1: 5 0 1 4 1
Case 2: Impossible
Case 3: 7 0 0 0 2 2 1
```

ACM-ICPC Live Archive

# 7837 Expected Number of Connected Components

$N$ nodes are given, each node is identified with a distinct integer (**between 1 to $N$, inclusive**). You randomly pick a subset of these nodes. The probability of having each individual node in your subset is given. In your subset, two nodes have edge between them if the GCD of the numbers identifying these two nodes is **greater than 1**. Count the expected number of connected components in your subset (subgraph).

## Input

First line of the input $T$ ($T \leq 100$) denotes the number of test cases. Then $T$ cases follows. Each case consists of 2 lines. The first line has a number $N$ ($1 \leq N \leq 100$) denoting the number of nodes. The next line consists of $N$ real numbers. The $i$-th ($1 \leq i \leq N$) real number $P_i$ ($0 \leq P_i \leq 1$) denotes the probability of the node $i$-th to be in the subset. Each number may contain up to 2 digits after the decimal point.

## Output

For each case you have to find the expected value as mentioned above. Say the expected value is $E$. You need to output $E \times (100^N)$ *modulo* 1 000 000 007. It is guaranteed that this number is an integer.

**Explanation of first case**

| Connected Components (in Brackets) | Expected Value |
|---|---|
| (1) | .0625 |
| (1) (2) | .0625 × 2 |
| (1) (2) (3) | .0625 × 3 |
| (1)(2, 4)(3) | .0625 × 3 |
| (1)(2,4) | .0625 × 2 |
| (1)(3) | .0625 × 2 |
| (1)(3)(4) | .0625 × 3 |
| (1)(4) | .0625 × 2 |
| (2) | .0625 |
| (2) (3) | .0625 × 2 |
| (2,4)(3) | .0625 × 2 |
| (2,4) | .0625 |
| (3) | .0625 |
| (3)(4) | .0625 × 2 |
| (4) | .0625 |
| Total | 1.75 |

So instead of printing 1.75 you need to print $1.75 \times 100 \times 100 \times 100 \times 100 = 175000000$.

## Sample Input

```
2
4
0.5 0.5 0.5 0.5
6
0.5 0.5 0.5 0.5 0.5 0.5
```

## Sample Output

```
175000000
999985132
```

ACM-ICPC Live Archive

# 7838 Coordinates

In this problem, you are an advisor to a galactic empire that tries to locate all the rebel bases on a planet Taboo. The map of the planet Taboo is a grid of size $10^8 \times 10^8$ and a position of a rebel based can be described by coordinate $(x, y)$ where $0 \leq x, y < 10^8$.

The empire captures a number of the rebels that surrender and interrogate them about the locations of the bases. Unfortunately, each surrendered rebel can only tell how far along $X$ axis and $Y$ axis a pair of bases are. Your job in this problem is to help reconstructing possible coordinates of all the bases that is consistent with all the interrogations.

## Input

First line consists of two integers, $N$ and $M$ where $1 \leq N \leq 100000$ is the number of bases and $N \leq M \leq 1000000$ is the number of captured rebels.

The next $M$ lines consist of 4 integers, $a_i$, $b_i$, $dx_i$, and $dy_i$ where $1 \leq a_i, b_i \leq N$, $-10^8 \leq dx_i, dy_i \leq 10^8$ indicating that the $x$-coordinate of base $b_i$ subtracted with the $x$-coordinate of base $a_i$ is $dx_i$ and the $y$-coordinate of base $b_i$ subtracted with the $y$-coordinate of base $a_i$ is $dy_i$

It is guaranteed that we can always deduce the positions of the bases from the input.

## Output

The output consists of $N$ lines each line consists of 2 integers $x_j$ and $y_j$ indicating possible position of the $j$-th base. You can answer any solution that is consistent with all the interrogations. Translation of map is possible to outside of $[0, 10^8]$, however, the output coordinate of each base must be between $-10^9$ to $10^9$

### Explanation

This is one of the possible coordinates, assuming that the first base is at (0,0)

## Sample Input

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

## Sample Output

```
0 0
3 0
3 3
```