Problem A. oval-and-rectangle

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	32 megabytes

Patrick Star found an oval!

The half of longer axes is on the x-axis with length a.

The half of shorter axes is on the y-axis with length b.

Patrick Star is planning to choose a real number c uniformly randomly from $[0, b]$, Patrick Star will get a rectangle after that:

1. Its four vertexes are on the outline of the oval.
2. Its two sides parallel to coordinate axis.
3. One of its sides is $y = c$.

Patrick Star wants to know the expectations of the rectangle's perimeter.

Input

The first line contain a integer T (no morn than 10), the following is T test case, for each test case :

Each line contains contains two integer a, b ($0 < b < a < 10^5$). Separated by an white space.

Output

For each test case output one line denotes the expectations of the rectangle's perimeter .

You should keep exactly 6 decimal digits and ignore the remain decimal digits.

It is guaranteed that the 7-th decimal digit of answer will not be 0 or 9.

Example

standard input	standard output
1 2 1	8.283185

Problem B. bookshelf

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	64 megabytes

Patrick Star bought a bookshelf, he named it ZYG !!

Patrick Star has N book .

The ZYG has K layers (count from 1 to K) and there is no limit on the capacity of each layer !

Now Patrick want to put all N books on ZYG :

1. Assume that the i-th layer has $cnt_i (0 \le cnt_i \le N)$ books finally.
2. Assume that $f[i]$ is the i-th fibonacci number ($f[0] = 0, f[1] = 1, f[2] = 1, f[i] = f[i-2] + f[i-1]$).
3. Define the stable value of i-th layers $stable_i = f[cnt_i]$.
4. Define the beauty value of i-th layers $beauty_i = 2^{stable_i} - 1$.
5. Define the whole beauty value of ZYG $score = gcd(beauty_1, beauty_2, ..., beauty_k)$(Note: $gcd(0, x) = x$).

Patrick Star wants to know the expected value of $score$ if Patrick choose a distribute method randomly !

Input

The first line contain a integer T (no morn than 10), the following is T test case, for each test case :

Each line contains contains three integer $n, k (0 < n, k \le 10^6)$.

Output

For each test case, output the answer as a value of a rational number modulo $10^9 + 7$.

Formally, it is guaranteed that under given constraints the probability is always a rational number $\frac{p}{q}$ (p and q are integer and coprime, q is positive), such that q is not divisible by $10^9 + 7$. Output such integer a between 0 and $10^9 + 6$ that $p - aq$ is divisible by $10^9 + 7$.

Example

standard input	standard output
1 6 8	797202805

Problem C. Ringland

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	128 megabytes

Patrick Star found the Ringland! It's marriage season in Ringland!

Ringland has a form of a circle's boundary of length L.

There are N bridegrooms and N brides, and bridegrooms decided to marry brides.

Of course, each bridegroom should choose exactly one bride, and each bride should be chosen by exactly one bridegroom.

All objects in Ringland are located on the boundary of the circle, including the capital, bridegrooms' castles and brides' palaces. The castle of the i-th bridegroom is located at the distance a_i from the capital in clockwise direction, and the palace of the i-th bride is located at the distance b_i from the capital in clockwise direction.

Let's define the inconvenience of a marriage the summary distance of every bride should walk along the circle from her palace to her bridegroom's castle in the shortest direction (in clockwise or counter-clockwise direction).

Help Patrick Star to calculate the smallest inconvenience among all marriages !

Input

The first line contain a integer T (no morn than 10), the following is T test case, for each test case :

The first line contains two integers N and L ($1 \le N \le 5 * 10^5, 1 \le L \le 10^9$) — the number of bridegrooms and brides and the length of Ringland.

The next line contains N integers, $a_1, a_2, ..., a_n$ ($0 \le a_1 \le a_2 \le ... \le a_n < L$) — the distances from the capital to the castles of bridegrooms in clockwise direction.

The next line contains N integers, $b_1, b_2, ..., b_n$ ($0 \le b_1 \le b_2 \le ... \le b_n < L$) — the distances from the capital to the palaces of brides in clockwise direction.

Attention: you may need fast read function.

Output

For each test case print one line,

This line contain only one integer, the smallest possible inconvenience of the wedding.

Example

standard input	standard output
1 2 5 3 4 0 2	2

Problem D. Shoot Game

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

You are playing a shooting game. The rules of the game are like this: You are in a two-dimensional plane and stand at $(0, 0)$. There are some obstacles above the x-axis. The location of each obstacle can be expressed as a tuple (H, L, R), It means there is an obstacle at the height of H, interval L to R. The i_{th} obstacle with W_i defense power.

You can shoot out "Energy beam of life". Each time you can consume X vitality then shoot out an energy beam with X attack power. The energy beam is a ray, When an energy beam hit an obstacle. If it's attack power not less than defense power of obstacle, it will destroy and pass through this obstacle. Otherwise it will disappear in smoke.

Now you want to find an optimal strategy to destroy all obstacles and consume minimum vitality.

Input

The first line contain a integer T (no morn than 10), the following is T test case, for each test case :

The first line of each test case contains a integers n ($1 \leq$ n ≤ 300), number of obstacle.

Each of the next n lines contains 4 integers H_i, L_i, R_i, W_i, ($1 \leq H_i \leq 1,000,000,000$, $-1,000,000,000 \leq L_i \leq R_i \leq 1,000,000,000$, $0 \leq W_i \leq 1,000,000,000$) means information of obstacles.

Output

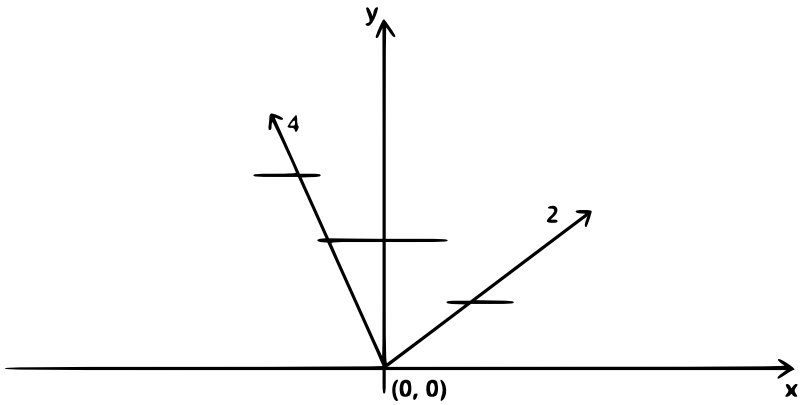
For each test case output the answer as described previously.

Example

standard input	standard output
2 3 1 1 2 2 2 -1 1 4 3 -2 -1 3 3 1 -1 1 2 2 -1 1 3 3 0 2 0	6 3

Note

The first testcase as shown in the picture:


y
4
2
(0, 0)
x

Problem E. black-and-white

Input file:	`standard input`
Output file:	`standard output`
Time limit:	12 seconds
Memory limit:	256 megabytes

Patrick Star found a world consist of N * N girds. Every gird is either black or write.

If one black gird a is not "reachable" from another one black gird b, we said that the pair of (a, b) is "special pair" ! What's more, Patrick Star regards the score of this pair as the Manhattan Distance between the two girds.

For every two girds, if there exist an path begin from one and end at another one, also this path consists only black girds and turns at most once, then the two girds are "reachable" from each other.

For better understanding of possible paths, every pair of gird (a, b) has only two possible paths while ignore the other limits, marked by #, and both paths include "a" and "b".

```
========
=a#####=
=#====#=
=#####b=
```

And see the text-picture below to get better understanding about 'reachable', we use "1" to stand for black gird and "0" to the white.

```
11111111
10110000
10011110
11111111
```

pair of [(2, 4), (4, 8)] : reachable. by path (2, 4) -¿ (4, 4) -¿ (4, 8).

pair of [(1, 5), (4, 8)] : not reachable, Manhattan Distance is —1 - 4— + —5 - 8— = 6.

pair of [(1, 5), (4, 4)] : reachable. by path (1, 5) -¿ (1, 4) -¿ (4, 4).

pair of [(1, 5), (4, 3)] : not reachable, Manhattan Distance is —1 - 4— + —5 - 3— = 5. ...

Now Patrick Star wants to know the max score among all special pairs.

If there is not any special pair , just output "Opps!".

Moreover, assume that the maximum score described above is D (if exist), Patrick Star wants know how many special pair has score of D. (unordered, that is to say $pair(a, b)$ is the same as $pair(b, a)$).

It is guaranteed that the two answers no more than 2147483647.

Input

The first line contain a integer T(no more than 10), the following is T test case, for each test case:

The first line contains one integers $N(1 \leq N \leq 2000)$.

The following N lines show an N * N girds.

each line contain a string of length N consist of "0" or "1" which mean "white" or "black"

Output

For each test case , output one line :

1. If no exist special pair, just output "Opps!".

2. Else output two integers separated by white space, the first one is the maximum score, and the second one is the number of special pairs reach the maximun score.

Example

standard input	standard output
1 4 1011 1001 1001 1111	5 3

Problem F. foam-transformation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	8 seconds
Memory limit:	256 megabytes

Operation "T" on array "a" with length at m has following description:

T i k, which holds $1 \leq i < m$, i,k are both integers,denotes that $a_i += 1 * (-1)^k, a_{i+1} += 4 * (-1)^k, .., a_{i+j} += (j+1)^2 * (-1)^k, ..., a_m += (m-i+1)^2 * (-1)^k$.

A array a with length n is "acid" if and only if :

1. If we add 5 zero elements at the end of a (length is $n+5$ now).

2. After 1, we can use finite operation "T" on a to make a become all zero array.

The "acidity" of a array "a" is the number of nonempty subintervals which is "acid".

(more formally, the number of pairs (l,r) satisfying $a_l, a_{l+1}...a_r$ is "acid",$1 \leq l \leq r \leq length(a)$)

Now,you are given an integer n and an array "a" consisting n integers.You should maintain the "acidity" of the array dynamicly.

Given an integer m, following m operations like that:

U i x, denoting $a_i += x, a_{i+1} += x, 1 \leq i < n$.

You should output an answer before all operations,

Then print one more answer after each operation denoting the dynamic "acidity".

Yes, as the writer is too lazy, we didn't encrypt the input data in any way, help yourself if you can solve this problem by some off line way :)

Input

The first line contain a integer T (no morn than 10), the following is T test case, for each test case :

The first line,two integers n and m, satisfying $2 \leq n \leq 100000, 1 < m \leq 100000$.

The second line consists n integers whose absolute values are not greater than 100000.

Next m lines,each would be shown like U i x,$1 \leq i < n, |x| \leq 100000$,which denotes a modifying operation.

Output

Print m+1 answers one perline denoting the dynamic "acidity" of the given array.

Example

standard input	standard output
1 20 10 -63541 0 -59055 -41170 0 0 0 -21343 25072 0 -76818 -59156 0 4435 59829 0 0 -56094 0 0 U 5 -62470 U 13 -52045 U 18 95988 U 13 -76265 U 7 35037 U 6 -41898 U 8 -71979 U 18 48427 U 16 -4208 U 15 34206	15 12 11 9 9 7 6 6 6 4 3

Note

you may need fast io.

Problem G. Variance-MST

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 second
Memory limit:	256 megabytes

Given a edge-weighted graph, your task is to compute the spanning tree with the smallest variance.

Formally, if w_e denotes the weight of edge e then the variance of the tree with n vertices is

$\frac{\sum_e (w_e - A)^2}{n-1}$, where $A = \sum_e \frac{w_e}{n-1}$

Input

The first line contain a integer T (no morn than 10), the following is T test case, for each test case :

First line contains two positive integer n and m denoting the number of vertices and edges of the graph.

Each of the following m lines contains three positive integers u_i , v_i , w_i,denoting the i_{th} edge connects the vertices u_i and v_i with the weight w_i.

It is guaranteed the graph is connected.

$2 \leq n \leq 100000$

$1 \leq m \leq 200000$

$1 \leq u_i, v_i \leq n$

$u_i \neq v_i$

$0 \leq w_i \leq 100000$

It is guaranteed that sum of n less than 400000, m less than 600000.

Output

Let P/Q be the number of correct answers, represented as an irreducible fraction. Print PQ^{-1} modulo 998244353.

each test case one line.

Example

standard input	standard output
1 4 6 1 2 2 1 3 4 2 3 6 4 1 7 4 2 5 4 3 3	665496236

Problem H. Rectangle Outline

Input file:	standard input
Output file:	standard output
Time limit:	10 seconds
Memory limit:	256 megabytes

Given a series of rectangular obstacles, one per line. Rectangular obstacles can overlap each other. Each rectangular obstacle is specified as two points on 2D plane that specify the opposite corners of a rectangle. Your job is to calculate the outline of the coverage area of all rectangular.

If the outline is made up of more than one closed polyline, print "Oops!". Otherwise, output the answer as follows: The first output point has the smallest X coordinate and then smallest Y coordinate. The second point has the same X coordinate with the first point. Then output other points in outline order. You can see more details in Sample.

It's guarantee that every rectangle's sides are parallel to OX and OY. Any point on the 2D plane will not be the vertices of two rectangles at the same time.

Input

The first line contain a integer T(no more than 10), the following is T test case, for each test case:

The first line of each test case contains a integers n ($1 \leq n \leq 100{,}000$), number of rectangular obstacles.

Each of the next n lines contains 4 integers x1, y1, x2, y2 ($1 \leq x1 ¡ x2 \leq 1{,}000{,}000{,}000$, $1 \leq y1 ¡ y2 \leq 1{,}000{,}000{,}000$) specify a rectangular obstacles.

It is guaranteed that the sum of all n does not exceed 400,000.

Output

Output the answer as described previously.

Examples

standard input	standard output
3 2 1 1 3 3 2 2 4 4	8 1 1 1 3 2 3 2 4 4 4 4 2 3 2 3 1
4 1 2 2 5 3 2 4 5 2 1 3 3 2 4 3 6	Oops!
2 1 1 2 2 3 3 4 4	Oops!

Problem I. Werewolf

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

"The Werewolves" is a popular card game among young people.In the basic game, there are 2 different groups: the werewolves and the villagers.

Each player will debate a player they think is a werewolf or not.

Their words are like "Player x is a werewolf." or "Player x is a villager.".

What we know is :

1. Villager won't lie.

2. Werewolf may lie.

Of cause we only consider those situations which obey the two rules above.

It is guaranteed that input data exist at least one situation which obey the two rules above.

Now we can judge every player into 3 types :

1. A player which can only be villager among all situations,

2. A player which can only be werewolf among all situations.

3. A player which can be villager among some situations, while can be werewolf in others situations.

You just need to print out the number of type-1 players and the number of type-2 players.

No player will talk about himself.

Input

The first line of the input gives the number of test cases T.Then T test cases follow.

The first line of each test case contains an integer N,indicating the number of players.

Then follows N lines,i-th line contains an integer x and a string S,indicating the i-th players tell you,"Player x is a S."

limits:

$1 \leq T \leq 10$

$1 \leq N \leq 100,000$

$1 \leq x \leq N$

$S \in$ "villager"."werewolf"

Output

For each test case,print the number of type-1 players and the number of type-2 players in one line, separated by white space.

Example

standard input	standard output
1 2 2 werewolf 1 werewolf	0 0

Problem J. Chopping hands

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 seconds
Memory limit:	32 megabytes

Patrick Star wants to collect some cards.The number of cards' type is M,numbered from 1 to M.

There are N card packs in the card shop:

1.The cost of i_{th} card pack is C_i, note that C_i may be negative.

2.The number of cards in i_{th} card pack is D_i.

3.The cards in all card packs are known,some types of card can appear in a pack more than once.

4.All of card packs can be bought only once.

Once Patrick Star gets an i_{th} type of card,he will regard $Value_{i,j}$ as this card's value if the card is j_{th} got as the i_{th} type.

(If you buy two packs [type 1,type 1,type 2] and [type 1,type 2,type 3],you will get value $[Value_{1,1}, Value_{1,2}, Value_{2,1}, Value_{1,3}, Value_{2,2}, Value_{3,1}]$)

After Patrick Star choose some card packs bought,cost S (may be negative too):

Firstly,sort all cards by their value. We will get the median of the value as mid.

(When the number of variables is an odd number, the value of the variable in the middle position is the median.when the number of variables is an even number, the median is the average of the 2 variable values in the middle position.)

Then Patrick Star considers his satisfaction V as $V = 2 * mid$

Patrick Star wants to know the maximum value of $V - S$.

Note that Patrick Star will buy at least one pack.

Input

The first line contain a integer T (no morn than 10), the following is T test case, for each test case :

the first line contain two integers N and M.

The second line contain N integers separated by spaces,C_1, C_2, C_n

Then next N lines,the first integer of each line is D_i,and the D_i integers after that represents the type of each card in i-th pack.

Then next M lines,the first integer t_i of each line represents the number of times the i-th card appear in all packs,and the t integers after that,j-th integer represents $Value_{i,j}$.

Limits:

$1 \le N \le 21$

$1 \le M \le 200$

$-1,000,000 \le C_i \le 1,000,000$

$1 \leq D_i \leq 100, \sum_{i=1}^{N} D_i \leq 100$

$0 \leq Value_{i,j} \leq 100,000,000$

Output

For each test case, please output an integer in a line represented the answer.It's clear that the answer is an integer.

Example

standard input	standard output
1 4 1 1 2 3 4 1 1 1 1 1 1 1 1 4 4 3 2 1	7

Problem K. sacul

Input file:	standard input
Output file:	standard output
Time limit:	1 seconds
Memory limit:	32 megabytes

Patrick Star found a kind of magical matrixs, he named them $HMBB$!!!

We use p to express the c-th prime number.

The size of $HMBB_i$ is $p^i * p^i$!

What is more surprising is that :

1. For any n, the element on the i-th row j-th column $HMBB_n[i][j] = (C(i,j)\ mod\ p) > 0)\ ?\ 1 : 0$.

2. Where $C(i,j)$ is count of method to choose j balls (unordered) from i balls which are pairwise distinct.

3. Note that rows and columns both count from zero.

Patrick Star defined $F[n][k]$ the sum of all elements of $(HMBB_n)^k$, $F[n][k] = \sum_{i=0}^{p^n-1} \sum_{j=0}^{p^n-1} HMBB_n^k[i][j]$

Now Patrick wants to know $(\sum_{i=1}^{n} \sum_{j=1}^{k} F[i][j])\ mod\ (10^9+7)$!

Input

The first line contain a integer T (no morn than 10), the following is T test case, for each test case :

Each line contains contains three integer c, n, k ($0 < n \leq 10^9$, $0 < c, k \leq 10^5$). Separated by an white space.

Output

For each test case output one line denotes the answer that Patrick Star wants to know.

Example

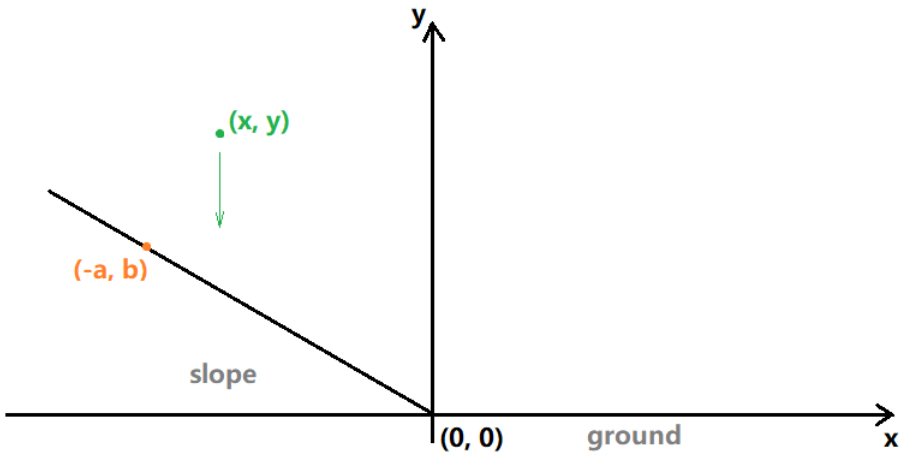
standard input	standard output
1 1 1 1	3

Problem L. Pinball

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

There is a slope on the 2D plane. The lowest point of the slope is at the origin. There is a small ball falling down above the slope. Your task is to find how many times the ball has been bounced on the slope.

It's guarantee that the ball will not reach the slope or ground or Y-axis with a distance of less than 1 from the origin. And the ball is elastic collision without energy loss. Gravity acceleration $g = 9.8m/s^2$.



Input

There are multiple test cases. The first line of input contains an integer T ($1 \leq T \leq 100$), indicating the number of test cases.

The first line of each test case contains four integers a, b, x, y ($1 \leq a, b, -x, y \leq 100$), indicate that the slope will pass through the point(-a, b), the initial position of the ball is (x, y).

Output

Output the answer.

It's guarantee that the answer will not exceed 50.

Example

standard input	standard output
1 5 1 -5 3	2