

Problem A. Awesome Shawarma

Input file: **awesome.in**
Output file: **standard output**
Balloon Color: **Silver**

Fouad has a raw awesome shawarma, and he is in a city which is represented as an undirected tree. He heard that there is a magical oven that will cook the Shawarma, to make it so delicious. However, In order to acquire the magical oven, there need to be two conditions that should be satisfied in this city:

- One extra edge that must be added to join two different nodes in the tree (It is allowed to join two nodes, which were previously connected by a direct edge).
- The number of bridges after adding the new edge should be between $[L, R]$ inclusively.

Please help Fouad to acquire the magical oven, to cook the awesome shawarma by counting in how many ways he can add an edge that satisfies the conditions above.

Input

The first line of the input file contains a single integer T , the number of test cases.

Each test case begins with a line containing three integers N , L , and R ($2 \leq N \leq 10^5$, $0 \leq L \leq R \leq N-1$), in which N is the number of nodes, and L and R are the minimum and maximum numbers of allowed bridges, respectively.

Then $N - 1$ lines follow, each line contains two integers X_i and Y_i ($1 \leq X_i, Y_i \leq N$), giving an edge between nodes X_i and Y_i .

Output

For each test case, print a single line containing the number of ways to add one new edge such that the number of bridges in the new graph is within the range $[L, R]$ inclusively.

Example

| awesome.in | standard output |
|------------|-----------------|
| 2 | 6 |
| 5 1 2 | 10 |
| 1 2 | |
| 2 3 | |
| 4 5 | |
| 2 4 | |
| 5 1 3 | |
| 1 2 | |
| 2 3 | |
| 4 5 | |
| 2 4 | |

Note

A bridge in a graph is an edge, such that if it is removed, the graph will become disconnected.

In the second sample, the result is acquired by connecting an edge between any two different nodes.

Problem B. Baklava Tray

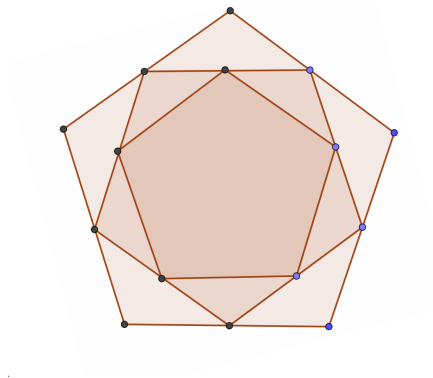
Input file: **baklava.in**
Output file: **standard output**
Balloon Color: **Cyan**

In the ACPC closing, the contestants decided to celebrate their hard work during the whole season by getting a very large tray of Baklava (Baklava).

During the order process, they started to watch the baker making trays for other customers. They noticed that he first draws an N -sided regular polygon with area 1 and put crushed hazelnut on the whole polygon, then he proceeds to draw the second polygon inside it by joining the midpoints of the sides of the first one and then put cashews on this polygon. Then, he proceeds with different types of nuts to draw an infinite sequence of N -sided polygons inside each other and each of them is formed by joining the midpoints of the sides of the latest drawn polygon. Consequently, the outer most polygon contains one type of nuts (hazelnuts), the second polygon contains two types (hazelnuts and cashews) and so on; this way, each polygon contains all the nuts of the polygons preceding it as well.

After the baker is done with the contestants' order, 10^4 persons with their forks will hit the tray at random places only once (yes people from other hotels all over Sharm were excited about this tray). The deliciousness factor is then computed by summing the number of nut types hit by the fork of each person and this is computed independently for each person. Please note that, if a person hits the i^{th} inner-most polygon, then he will hit exactly i nut types.

Help the Regional Contest Director determine the expected deliciousness factor for a tray with N sides.



The first three polygons for $N = 5$.

Input

The first line of the input contains a single integer T specifying the number of test cases.

Each test case consists of a single line containing a single integer N ($3 \leq N \leq 800$) representing the number of sides of the tray.

Output

For each test case, print a single line containing a single decimal number (rounded to exactly 5 decimal places) representing the expected deliciousness factor for the corresponding test case.

The output will be checked with a relative error, therefore a small difference in the output integer values might be tolerated.

Example

| baklava.in | standard output |
|------------|-----------------|
| 5 | 13333.33333 |
| 3 | 20000.00000 |
| 4 | 28944.27191 |
| 5 | 104721.35955 |
| 10 | 2536365.55794 |
| 50 | |

Note

A regular polygon of N sides is a polygon where all the sides are of the same length, and all the inner angles are the same.

Problem C. Coffee

Input file: `coffee.in`
Output file: `standard output`
Balloon Color: `Green`

Given the prices of different types of coffee (latte, cappuccino, ...) for each size (small, medium, and large), and a list of orders of a group of persons. Your task is to find how much will each person pay eventually. The cost that a person needs to pay is the cost of coffee he/she ordered in addition to the delivery fees. The delivery fees for each person are 100\$ divided by the number of persons (rounded down).

Eventually, the final cost should ignore 1\$ greater or less than what should be paid to the nearest multiple of 5 (i.e 44\$ and 46\$ will be rounded to 45\$. However, 47\$ and 48\$ will not be changed).

Input

The first line of the input contains a single integer T specifying the number of test cases.

Each test case begins with a line containing two integers C and P ($1 \leq C, P \leq 100$), in which C is the number of different types of coffees, and P is the number of unique persons.

Then C lines follow, giving the coffee types. Each line contains a string N and three integers S , M , and L ($1 \leq S, M, L \leq 100$), in which N is a coffee name, S , M , and L are the prices for small, medium, and large sizes, respectively. Each coffee type will appear exactly once per test case.

Then P lines follow, giving the list of orders. Each line contains three strings X , Y , and Z , in which X is a person name, Y is the coffee size ($Y \in \{small, medium, large\}$), and Z is a coffee name. It is guaranteed that all the given P names are distinct, and each person will order coffee type that exists.

Both the coffee and person names are non-empty strings consisting of lowercase and uppercase English letters with a length of no more than 15 letter.

Output

For each test case, print P lines in which each line contains two space-separated values, the name of the person and the total cost he/she will pay for his/her order. Print the persons in the same order as given in the input.

Example

| coffee.in | standard output |
|----------------------------|-----------------|
| 1 | Mohammed 67 |
| 3 3 | Mostafa 70 |
| cappuccino 28 34 41 | Ahmad 60 |
| latte 25 31 38 | |
| flatwhite 26 33 47 | |
| Mohammed medium cappuccino | |
| Mostafa large latte | |
| Ahmad small flatwhite | |

Note

The delivery fees is 100\$ and divided by 3 persons, so each of them will pay $\lfloor \frac{100}{3} \rfloor = 33\$$.

- The cost for “Mohammed” is 34\$ for drinks, 33\$ for delivery fees, this will be 67\$, which is the final cost.
- The cost for “Mostafa” is 38\$ for drinks, 33\$ for delivery fees, which will be 71\$, then 1\$ is ignored to the nearest multiple of 5, so the final cost is 70\$.

- The cost for “Ahmad” is 26\$ for drinks, 33\$ for delivery fees, which will be 59\$, then 1\$ is ignored to the nearest multiple of 5, so the final cost is 60\$.

Problem D. Dull Chocolates

Input file: `dull.in`
Output file: `standard output`
Balloon Color: `Yellow`

Fouad wants to eat a chocolate bar, so he bought a rectangular chocolate bar that has N rows and M columns of chocolate.

Fouad found that most of the cells in that chocolate bar are dark chocolate except for K cells which are white chocolate. Now, he wants to eat a prefix sub-grid of the bar. A prefix sub-grid is a rectangular sub-grid that starts at the first cell $(1, 1)$ and ends at any cell (i, j) . Fouad is also wondering how many perfect prefix sub-grids of the bar. A perfect prefix sub-grid is a one that contains an odd number of white chocolate cells.

Help Fouad find the number of perfect prefix sub-grids and non-perfect prefix sub-grids.

Input

The first line of the input containing a single integer T specifying the number of test cases.

Each test case begins with a line containing three integers N , M , and K ($1 \leq N, M \leq 10^9$, $0 \leq K \leq 10^3$), in which N and M are the number of rows and columns in the chocolate bar, respectively, and K is the number of white chocolate cells.

Then K lines follow, each line containing two integers X_i and Y_i ($1 \leq X_i \leq N$, $1 \leq Y_i \leq M$), giving the positions of the white chocolate cells. It is guaranteed that all the given positions are unique.

Output

For each test case, print a single line containing two space-separated integers representing the number of perfect prefix sub-grids and non-perfect prefix sub-grids, respectively.

Example

| <code>dull.in</code> | <code>standard output</code> |
|----------------------|------------------------------|
| 3 | 0 9 |
| 3 3 0 | 4 5 |
| 3 3 1 | 14 11 |
| 2 2 | |
| 5 5 5 | |
| 1 5 | |
| 2 1 | |
| 3 3 | |
| 4 2 | |
| 5 4 | |

Note

In the second test case, the chocolate bar can be represented as follows:

```
ddd
dwd
ddd
```

in which 'd' represents dark chocolate cells while 'w' represents white chocolate cells. There are four perfect prefix sub-grids that end at cells: $(2, 2)$, $(2, 3)$, $(3, 2)$, and $(3, 3)$.

Problem E. Exciting Menus

Input file: exciting.in
Output file: standard output
Balloon Color: White

In a restaurant, given N menus, where the menus are represented by the strings S^1, \dots, S^N . Each string S^i is associated with an array A^i (of the same length) of joy levels. In this problem, you should choose a submenu defined by three integers (i, j, k) , where i is the index of one of the menus, and j, k specify a substring of this chosen menu.

We need to find the submenu of the highest quality, where the quality Q is defined by the function:

$$Q(i, j, k) = \text{popularity}(S_{j,k}^i) \cdot A_k^i \cdot |S_{j,k}^i|$$

- $S_{j,k}^i$ = The substring $[j, k]$ (where j, k are one-based, inclusive) of the string S^i .
- $\text{popularity}(S_{j,k}^i)$ = number of menus (from S^1, \dots, S^N) that contain the submenu-string $S_{j,k}^i$ as a prefix.
- $|S_{j,k}^i|$ is the size of the submenu, which is the length of the substring representing the submenu.
- A_k^i is the joy level of the submenu (k^{th} entry of the i^{th} array), please note that the index j is not used here.

Can you find the submenu of the highest quality? Please output the corresponding highest quality of the function Q .

Input

The first line of the input contains a single integer T specifying the number of test cases.

Each test case begins with a line containing an integer N ($1 \leq N \leq 10^5$) specifying the number of menus in the restaurant.

Then N lines follow, the i^{th} line contains a string S^i representing the i^{th} menu. All the menus are non-empty strings consisting of lowercase English letters, and the sum of their lengths will not exceed 10^5 per test case.

Then N lines follow, the i^{th} line contains array A^i , in which A^i has the same length as string S^i and ($0 \leq A_k^i \leq 10^9$). The values of each array are space-separated.

Output

For each test case, print a single line containing an integer Q representing the maximum quality corresponding to the best submenu (i, j, k) .

Example

| exciting.in | standard output |
|-------------|-----------------|
| 2 | 500 |
| 2 | 600 |
| aabaa | |
| aa | |
| 0 0 0 0 100 | |
| 0 1 | |
| 4 | |
| bbbaa | |
| aa | |
| aab | |
| aax | |
| 0 0 0 0 100 | |
| 0 0 | |
| 0 0 0 | |
| 0 0 0 | |

Note

The triples corresponding to the answers of the two given cases are (1, 1, 5) and (1, 4, 5).

Problem F. Flipping El-fetiera

Input file: `fetiera.in`
Output file: `standard output`
Balloon Color: `Orange`

Fouad was craving for Fetiera, so he went to a Fetier restaurant and ordered one. The chef told him that he sells a square Fetiera in the form of a $N \times N$ matrix.

On the surface of the Fetiera, there is one Semsema on each 1×1 cell of the Fetiera. Each Semsema can be either on the top side or the bottom side of the Fetira, but he cannot have two Semsemas in one cell on both sides.

Fouad loves to watch the chef flipping Fetiera, and this one was way more interesting than usual. The chef chooses a random (uniformly at random) rectangular submatrix and flips it in place. Whenever he flips a submatrix of the Fetira, the Semsemas which were on top will be on the bottom and vice versa.

Given the initial state of the Fetiera, and knowing that the chef did the flipping K times, Fouad was wondering how many Semsemas will be on the top side of it. Therefore, he is asking you to help him find the expected number of Semsemas on the top side.

Input

The first line of input contains a single integer T specifying the number of test cases.

Each test case begins with a line containing two space-separated integers N and K ($1 \leq N \leq 300$, $0 \leq K \leq 300$), in which N is the size of the Fetiera matrix, and K is the number of flipping operations.

Then N lines follow, each line i contains N space-separated values F_{i1}, \dots, F_{iN} ($F_{ij} \in \{0, 1\}$), in which F_{ij} representing the top side of the j^{th} cell in the i^{th} row of the Fetiera (1 means the Semsema is on the top side and 0 means the Semsema is on the bottom side in the initial configuration).

Output

For each test case, print a single line containing a single decimal number (rounded to exactly 5 decimal places) representing the expected number of Semsemas on the Fetiera after making the flipping operation for exactly K times. The output will be checked with a relative error.

Example

| <code>fetiera.in</code> | <code>standard output</code> |
|-------------------------|------------------------------|
| 2 | 7.57280 |
| 4 2 | 4.58728 |
| 1 0 1 1 | |
| 0 0 0 1 | |
| 1 0 0 0 | |
| 0 0 0 0 | |
| 3 3 | |
| 1 1 1 | |
| 0 0 0 | |
| 1 0 0 | |

Problem G. Greatest Chicken Dish

Input file: `gcdrng.in`
Output file: `standard output`
Balloon Color: `Gold`

Fouad, the ACPC judges, and organizers were hungry, so they went to a restaurant. This restaurant makes chicken cubes in seekh (skewers), such that each seekh has N chicken cubes in which the i^{th} cube has an integer A_i that represents the mixture of spices used to cook that cube. A set of successive cubes are considered delicious from Fouad's perspective when the greatest common divisor (GCD) of their mixture of spices is exactly equal to D .

Since Fouad wants to try many sub-parts of the seekh, he keeps asking the judges some queries such that each query consists of three integers L , R , and D , and he wants to know what is the number of consecutive sub-parts of the seekh in the range A_L, A_{L+1}, \dots, A_R that the GCD of their mixtures of spices is equal to D ; that is the count of all pairs (i, j) such that $\text{gcd}(A_i, A_{i+1}, \dots, A_j) = D$ and $L \leq i \leq j \leq R$.

Since the judges want to relax and not solve problems and just eat, they asked you to solve this problem for them. Can you solve it?

Input

The first line of the input contains a single integer T specifying the number of test cases.

Each test case begins with a line containing two integers N and Q ($1 \leq N \leq 10^5$, $1 \leq Q \leq 5 \cdot 10^4$), in which N is the number of chicken cubes in a seekh, and Q is the number of queries Fouad will ask.

Then a line follows containing N integers A_1, \dots, A_N ($1 \leq A_i \leq 10^6$), in which A_i represents the mixture of spices used to cook the i^{th} chicken cube.

Then Q lines follow, each line contains three space-separated integers L , R , and D ($1 \leq L \leq R \leq N$, $1 \leq D \leq 10^6$), giving the queries.

Output

For each test case, print a single line per query containing the number of consecutive sub-parts of the seekh in the given range $[L, R]$ that the GCD of their mixtures of spices is equal to D .

Example

| <code>gcdrng.in</code> | <code>standard output</code> |
|------------------------|------------------------------|
| 1 | 6 |
| 8 4 | 14 |
| 1 12 24 8 4 16 2 3 | 0 |
| 3 7 4 | 9 |
| 1 8 1 | |
| 1 4 3 | |
| 2 6 4 | |

Note

Greatest Common Divisors of multiple arguments is computed recursively according to the equation $\text{gcd}(x_1, x_2, \dots, x_n) = \text{gcd}(\text{gcd}(x_1, \dots, x_{n-1}), x_n)$.

Problem H. Hawawshi Decryption

Input file: hawawshi.in
Output file: standard output
Balloon Color: Pink

Hawawshi is a traditional dish in Egypt, it's really appreciated that there are people who are storing it in highly secure safes. The safes are encrypted by a pseudo-random number generator that generates a sequence of random numbers according to the equation: $R_{n+1} = (a \cdot R_n + b) \bmod p$, where a , b , and p are integers, p is a prime number, and R_0 is the first generated number which is also known as the seed of the sequence.

Some secret information has been delivered to you that the seed of the random generator was an integer selected at random from the range $[A, B]$ inclusively, furthermore, that the key number of the Hawawshi-safe is a number that appeared as one of the first N generated random numbers (namely, R_0, R_1, \dots, R_{N-1}). You are going to try some different key numbers X , but first, you need to know what is the probability that X has appeared in the first N generated random numbers?

Input

The first line of the input contains a single integer T specifying the number of test cases.

Each test case consists of a single line containing seven integers N , X , A , B , a , b , and p .

($1 \leq N, X, A, B, a, b \leq p - 1$, $1 < p < 10^8$, $1 \leq A \leq B \leq \min(100, p - 1)$), in which p is a prime number.

Output

For each test case, print a single line containing a reduced fraction q/r (q, r are integers) representing the probability that the number X appears in the first N generated random numbers (by the pseudo-random number generator given). It's guaranteed that it is possible to represent the probability as a reduced fraction of integers as requested.

Example

| hawawshi.in | standard output |
|----------------|-----------------|
| 5 | 1/2 |
| 2 2 4 5 2 1 7 | 0/1 |
| 6 3 4 5 2 1 7 | 2/5 |
| 2 2 1 5 2 1 7 | 1/1 |
| 4 5 5 5 4 7 11 | 0/1 |
| 4 6 5 5 4 7 11 | |

Note

As a demonstration of the pseudo-random number generator, if $a = 2, b = 1, p = 7, R_0 = 2$, then the generated numbers are 2, 5, 4, 2, 5, 4, 2, 5, 4, \dots .

In the last two test cases, $a = 4, b = 7, p = 11, R_0 = 5$, and therefore the generated sequence is 5, 5, 5, \dots .

Problem I. Ice-cream Knapsack

Input file: icecream.in
Output file: standard output
Balloon Color: Blue

There is a wonderful ice-cream shop that contains N ice-creams, such that each ice-cream is represented by two numbers C_i and H_i denoting the number of calories and the happiness value, respectively.

You want to buy exactly K ice-creams such that the calories of the densest ice-cream (the one with most calories) are as minimal as possible. If there is more than one way to do that, you want to maximize the total happiness of the ice-creams you will buy, that is the sum of the happiness values of the chosen ice-creams.

Input

The first line of the input contains a single integer T specifying the number of test cases.

Each test case begins with a line containing two integers N and K ($1 \leq K \leq N \leq 10^5$), in which N is the number of ice-creams in the shop, and K is the number of ice-creams you want to buy.

Then a line follows containing N integers C_1, \dots, C_N ($0 \leq C_i \leq 10^9$), in which C_i is the number of calories in the i^{th} ice-cream. Then a line follows containing N integers H_1, \dots, H_N ($0 \leq H_i \leq 10^9$), in which H_i is the happiness value of the i^{th} ice-cream.

Output

For each test case, print a single line containing two space-separated integers representing the calories of the densest ice-cream you will buy and the total happiness of the ice-creams you will buy, respectively.

Remember that your goal is to buy K ice-creams such that the calories of the densest ice-cream (the one with most calories) are as minimal as possible. If there is more than one way to do that, you want to maximize the total happiness of the ice-creams you will buy.

Example

| icecream.in | standard output |
|-------------|-----------------|
| 1 | 3 12 |
| 5 3 | |
| 1 2 3 4 5 | |
| 5 4 3 2 1 | |

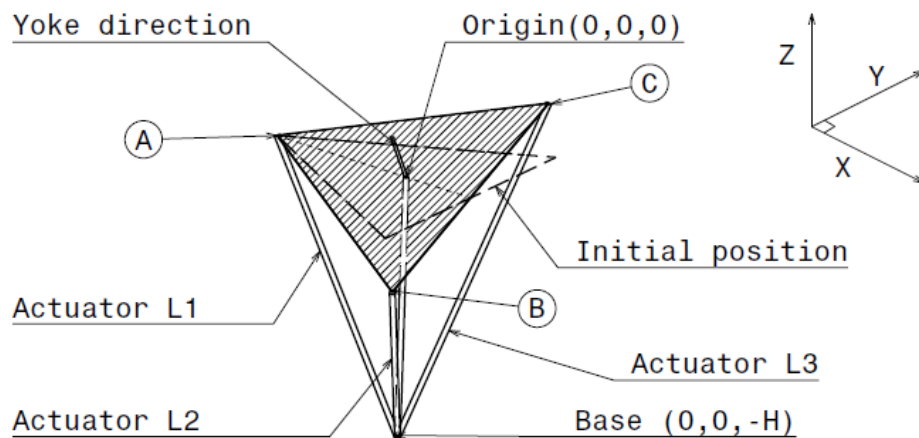
Problem J. Journey to Jupiter

Input file: jupiter.in
Output file: standard output
Balloon Color: Fucia

It's the year 10,007 AD, and all the delicious food in the Arab region has gone extinct. Luckily for you, you made once a time travel in the past and you could taste how delicious this food was. An alien rumor has come to you saying that there's still remaining Arab food in Jupiter. Since the Alien rumors, unlike humans', are usually true, you decided to build your own spacecraft to go to Jupiter. In order to test the control system you embedded in your spacecraft, you want to build a flight simulator, which is a device that artificially re-creates spacecraft flight and the environment in which it flies, for astronaut training and other purposes.

In this problem, a flight simulator consists of a 2-dimensional equilateral triangle of side length L , with three actuators (mechanical devices, that can be treated as line segments of modifiable lengths) attached to the vertices A, B, C of the triangle, respectively. The other ends of the actuators are all attached to a static base point located at $(0, 0, -H)$. The midpoint of the triangle (located at $(0, 0, 0)$) is attached to the same base using a spherical connection, which means that the triangle can rotate in any direction but its midpoint cannot move.

The yoke (a device used to control the rotations of the triangle) has the direction vector along $\mathbf{N} = (N_x, N_y, N_z)$ which is always normal on the plane of the triangle. By changing the direction of the yoke, the length of each actuator changes and the triangle rotates (the yoke will remain normal on the triangle), while the midpoint remains at $(0, 0, 0)$. Initially, all the actuators have the same length, the triangle lies horizontally in the XY plane (i.e. $z = 0$) and the yoke is pointing along the positive Z -axis (i.e. initially $\mathbf{N} = (0, 0, 1)$). The point A is initially located on the negative X -axis ($z_A = y_A = 0, x_A < 0$) and the vertices A, B, C are in an anticlockwise order relative to the yoke direction.



We want you to write a program that makes the simulator rotate given a new yoke direction. Formally, given both of a yoke direction and the location of the point A after rotation, we need you to calculate the corresponding lengths of the actuators L_1, L_2, L_3 , which are attached to the vertices A, B, C respectively, so that the yoke direction is still normal on the triangle.

Input

The first line of the input contains a single integer T specifying the number of test cases.

Each test case consists of a single line containing eight integers $N_x, N_y, N_z, A_x, A_y, A_z, L$, and H ($-500 \leq A_x, A_y, A_z, N_x, N_y \leq 500, 1 \leq N_z, H, L \leq 500$), in which N_x, N_y , and N_z is the yoke direction

after rotation, A_x , A_y , and A_z is the position of the point A after rotation, L is the side length of the triangle, and H is the point in the base where actuators are attached

It is guaranteed that the following constraints hold for all test cases:

- $A_x, A_y, A_z \neq 0$
- $A_x N_x + A_y N_y + A_z N_z = 0$
- $A_x^2 + A_y^2 + A_z^2 = \frac{L^2}{3}$.

Output

For each test case, print a single line containing three space-separated decimal numbers (rounded to exactly 6 decimal places), the lengths L_1, L_2, L_3 of the actuators connected to the vertices A, B, C corresponding to the given query.

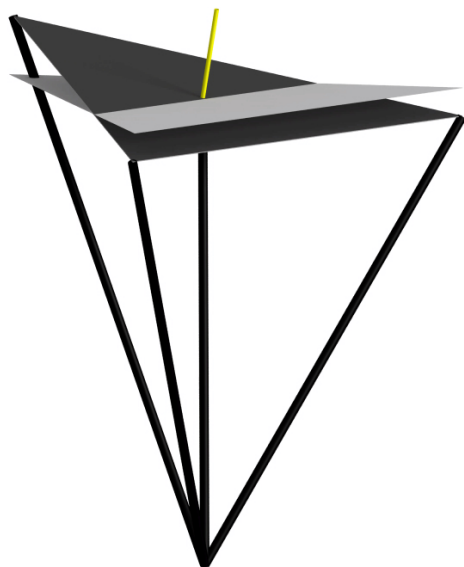
The given input will guarantee that there is exactly one solution. The numbers will be checked with a relative error.

Example

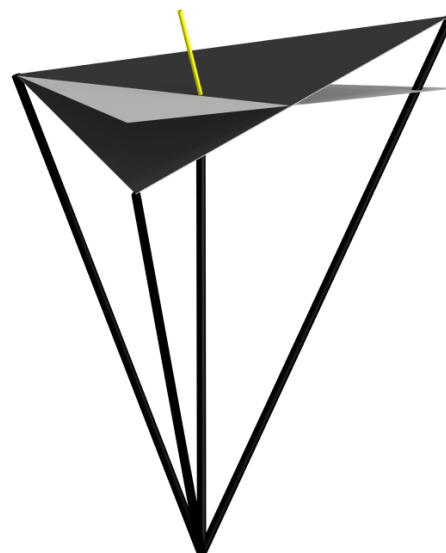
| jupiter.in | standard output |
|------------------------|------------------------------|
| 2 | 13.964240 20.238885 7.237923 |
| 13 -18 1 7 5 -1 15 12 | 8.831761 11.473743 6.507936 |
| -12 -17 1 7 -5 -1 15 3 | |

Note

The figures below show different positions of the flight simulator for different yoke directions :



Rotation in XZ
plane



Rotation in YZ
plane

Problem K. Khoshaf

Input file: `khoshaf.in`
Output file: `standard output`
Balloon Color: `Black`

The judges were sitting and they wanted to try Khoshaf (Mix of dried fruits soaked in Apricot juice). They went to a restaurant to order it, however, they found that there is only one dish remaining, so they decided to make a problem, and the first to solve it will have this remaining dish.

The problem is as follows: Given four integers N , K , L , and R , count the number of arrays of length N that contains integer values within the range $[L, R]$ inclusively and has exactly K continuous sub-intervals with sum divisible by 3. The output answer should be taken modulo $10^9 + 7$. Please note that the sub-intervals may **overlap**. Can you help the chief judge to have the final dish?

Input

The first line contains a single integer T specifying the number of test cases.

Each test case consists of a single line containing four integers N , K , L , and R ($1 \leq N, K \leq 10^4$, $1 \leq L \leq R \leq 10^9$), as described in the problem statement.

Output

For each test case, print a single line containing the number of arrays of length N that contains integer values within the range $[L, R]$ inclusively and has exactly K continuous sub-intervals with sum divisible by 3. The output answer should be taken modulo $10^9 + 7$.

Example

| <code>khoshaf.in</code> | <code>standard output</code> |
|-------------------------|------------------------------|
| 3 | 6 |
| 3 2 1 3 | 20 |
| 4 3 1 3 | 1808 |
| 5 4 4 8 | |

Note

In the first test case, any of the following arrays will contain exactly two sub-intervals whose sums are divisible by 3:

$[1,2,1] \rightarrow$ The sub-intervals are $[1,2]$ and $[2,1]$ with indices $[0,1]$ and $[1,2]$ respectively.

$[1,3,2] \rightarrow [3], [1,3,2]$

$[2,1,2] \rightarrow [2,1], [1,2]$

$[2,3,1] \rightarrow [3], [2,3,1]$

$[3,1,3] \rightarrow [3], [3]$

$[3,2,3] \rightarrow [3], [3]$

Problem L. Looking for Taste

Input file: looking.in
Output file: standard output
Balloon Color: Red

Fouad went to a restaurant and while reading the menu, he found out that it includes an interesting piece of information about each meal. This piece of information was the tastes each meal contains.

The menu contains N meals such that each meal is represented by an integer A_i where the j^{th} bit in the binary representation of A_i represents the availability of taste j in this meal (1 means you will feel the j^{th} taste while eating it and 0 means you will not).

You will feel taste j if you eat some meals, such that any of them has the bit j equals to 1. In other words, the tastes included by eating meals A_i and A_j are the tastes included by the number $A_i \odot A_j$, where \odot is the bitwise OR operation.

Fouad wants to eat many meals so that he can enjoy the food as maximum as possible, but he cannot eat more than K meals (because of his strict diet). Since not all the tastes are the same, he is asking for your help to select a subset of K meals, such that the number representing the overall food is as maximum as possible (the bitwise OR of the chosen meals is as maximum as possible).

Input

The first line of the input consists of a single integer T , the number of test cases.

Each test case starts with a line containing two space-separated integers N, K (where $20 \leq K \leq N \leq 10^5$).

The second line of the test case contains N space-separated integers A_1, \dots, A_N (where for all i we have $0 \leq A_i \leq 10^6$).

Output

For each test case, print a single line containing a single integer X representing the maximum value of tastes Fouad can feel by eating a subset of at most K meals. If the subset of meals Fouad will eat will contain the i^{th} taste, then the i^{th} bit of the number X must be 1, otherwise, the i^{th} bit must be 0.

Example

| looking.in | standard output |
|---|-----------------|
| 1 | 63 |
| 25 25 | |
| 1 32 45 53 12 45 32 47 53 61 51 50 2 43 | |
| 47 1 2 3 4 5 6 7 8 9 10 | |

Note

In the sample, he can eat all the meals, so the result is bitwise OR of all the given numbers.