中国大学生程序设计竞赛

The 2017 China Collegiate Programming Contest

Qinhuangdao Site

Contest Section

October 29, 2017

Sponsored by

Neuedu 东软睿道 G-bits 吉比特 雷霆游戏 Leiting Games

Problem List

A	Balloon Robot
B	Expected Waiting Time
C	Crusaders Quest
D	Graph Generator
E	String of CCPC
F	Getting Lost
G	Numbers
H	Prime Set
I	Triangulation
J	Tree Equation
K	Diversity and Variance
L	One-Dimensional Maze
M	Safest Buildings

This problem set should contain 13 (thirteen) problems on 20 (twenty) numbered pages. Please inform a runner immediately if something is missing from your problem set.

Problem A. Balloon Robot

Description

The 2017 China Collegiate Programming Contest Qinhuangdao Site is coming! There will be n teams participating in the contest, and the contest will be held on a huge round table with m seats numbered from 1 to m in clockwise order around it. The i-th team will be seated on the s_i-th seat.

BaoBao, an enthusiast for competitive programming, has made p predictions of the contest result before the contest. Each prediction is in the form of (a_i, b_i), which means the a_i-th team solves a problem during the b_i-th time unit.

As we know, when a team solves a problem, a balloon will be rewarded to that team. The participants will be unhappy if the balloons take almost centuries to come. If a team solves a problem during the t_a-th time unit, and the balloon is sent to them during the t_b-th time unit, then the unhappiness of the team will increase by $t_b - t_a$. In order to give out balloons timely, the organizers of the contest have bought a balloon robot.

At the beginning of the contest (that is to say, at the beginning of the 1st time unit), the robot will be put on the k-th seat and begin to move around the table. If the robot moves past a team which has won themselves some balloons after the robot's last visit, it will give all the balloons they deserve to the team. During each unit of time, the following events will happen ***in order***:

1. The robot moves to the next seat. That is to say, if the robot is currently on the i-th ($1 \le i < m$) seat, it will move to the $(i + 1)$-th seat; If the robot is currently on the m-th seat, it will move to the 1st seat.
2. The participants solve some problems according to BaoBao's prediction.
3. The robot gives out balloons to the team seated on its current position if needed.

BaoBao is interested in minimizing the total unhappiness of all the teams. Your task is to select the starting position k of the robot and calculate the minimum total unhappiness of all the teams according to BaoBao's predictions.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains three integers n, m and p ($1 \le n \le 10^5$, $n \le m \le 10^9$, $1 \le p \le 10^5$), indicating the number of participating teams, the number of seats and the number of predictions.

The second line contains n integers $s_1, s_2, \ldots, s_n$ ($1 \le s_i \le m$, and $s_i \ne s_j$ for all $i \ne j$), indicating the seat number of each team.

The following p lines each contains two integers a_i and b_i ($1 \le a_i \le n$, $1 \le b_i \le 10^9$), indicating that the a_i-th team solves a problem at time b_i according to BaoBao's predictions.

It is guaranteed that neither the sum of n nor the sum of p over all test cases will exceed 5×10^5.

Output

For each test case output one integer, indicating the minimum total unhappiness of all the teams according to BaoBao's predictions.

Example

standard input	standard output
4 2 3 3 1 2 1 1 2 1 1 4 2 3 5 1 2 1 1 2 1 1 2 1 3 1 4 3 7 5 3 5 7 1 5 2 1 3 3 1 5 2 5 2 100 2 1 51 1 500 2 1000	1 4 5 50

Note

For the first sample test case, if we choose the starting position to be the 1st seat, the total unhappiness will be (3-1) + (1-1) + (6-4) = 4. If we choose the 2nd seat, the total unhappiness will be (2-1) + (3-1) + (5-4) = 4. If we choose the 3rd seat, the total unhappiness will be (1-1) + (2-1) + (4-4) = 1. So the answer is 1.

For the second sample test case, if we choose the starting position to be the 1st seat, the total unhappiness will be (3-1) + (1-1) + (3-2) + (3-3) + (6-4) = 5. If we choose the 2nd seat, the total unhappiness will be (2-1) + (3-1) + (2-2) + (5-3) + (5-4) = 6. If we choose the 3rd seat, the total unhappiness will be (1-1) + (2-1) + (4-2) + (4-3) + (4-4) = 4. So the answer is 4.

Problem B. Expected Waiting Time

Description

DreamGrid is a famous coach of competitive programming. He is so kind that many competitors are willing to ask him for advice.

DreamGrid meets exactly n competitors every day in his office, and thus n "arriving" events and n "selecting" events will happen. An arriving event at time t indicates that a competitor arrives at the waiting room of the office at time t, and a selecting event at time t indicates that DreamGrid randomly selects (with equal probability) a competitor from the waiting room to talk with. Of course, if the selecting event happens, the waiting room must not be empty. After the talk, the competitor leaves the office and never comes back.

After several days, DreamGrid starts to be curious about the average of the total expected waiting time of every competitor in all valid cases. The waiting time of a competitor is the time he is selected by DreamGrid minus the time he arrives at the waiting room. A case is a sequence of length $2n$ consisting of n arriving events and n selecting events, where the i-th event will happen at time a_i. For a valid case, it must be satisfied that when a selecting event happens, the waiting room must not be empty.

For example, let's denote an arriving event as 'A', and a selecting event as 'S'. If $n = 2$, $a_1 = 1$, $a_2 = 2$, $a_3 = 3$ and $a_4 = 4$, then the sequence "AASS" is valid, but the sequence "ASSA" is not valid, as the "selecting" event happening at time $a_3 = 3$ is not valid.

As the answer may not be an integer, you are supposed to calculate $ab^{-1} \bmod p$, where $\frac{a}{b}$ (a and b are coprime) is the answer, $p > 2n$ and p is prime, and b^{-1} is the modular multiplicative inverse of b with respect to the modulus p. It's easy to prove that the prime factors of b will never be larger than $2n$.

Input

There are multiple test cases. The first line of input contains an integer T ($1 \le T \le 10^4$), indicating the number of test cases. For each test case:

The first line contains five integers n, p, b_0, A, B ($1 \le n \le 10^6, 2n < p \le 2 \times 10^9$, $0 \le b_0, A, B < p$), where p is a prime number. The meanings of n and p are described above. The rest of them is a generator of a, where $a_0 = 0$, $a_i = a_{i-1} + b_i + 1$ and $b_i = (A \cdot b_{i-1} + B) \mod p$ for all $1 \le i \le 2n$.

It is guaranteed that the sum of n in all test cases does not exceed 10^7.

Output

For each test case, output one integer denoting the answer in a single line.

Example

standard input	standard output
5 1 1000000007 0 1 0 2 1000000007 0 1 1 2 7 5 2 3 3 31 15 6 24 20 1000000007 0 1 0	1 12 1 21 879705565

Note

Let's denote the arriving event as 'A', and the selecting event as 'S'.

In the first test case, we have $a_1 = 1$ and $a_2 = 2$, and there is only one valid sequence "AS". The waiting time of the only competitor is 2-1 = 1, so the answer is 1, and we need to output 1 mod 1000000007 = 1.

In the second test case, we have $a_1 = 2$, $a_2 = 5$, $a_3 = 9$ and $a_4 = 14$. There are two valid sequences "ASAS" and "AASS".

For the first sequence, the expected waiting time of the first arriving competitor is 5-2 = 3, and the expected waiting time of the second arriving competitor is 14-9 = 5.

For the second sequence, the expected waiting time of the first arriving competitor is ((9-2)+(14-2))/2 = 9.5, and the expected waiting time of the second arriving competitor is ((9-5)+(14-5))/2 = 6.5. So the answer is ((3+5)+(9.5+6.5))/2 = 12, and we need to output 12 mod 1000000007 = 12.

In the third test case, we have $a_1 = 7$, $a_2 = 9$, $a_3 = 15$ and $a_4 = 22$. Just like the analysis for the 2nd sample test case, the total expected waiting time for the sequence "ASAS" is (9-7)+(22-15) = 9, and the total expected waiting time for the sequence "AASS" is ((15-7)+(22-7))/2+((15-9)+(22-9))/2 = 21. So the answer is (9+21)/2 = 15, and we need to output 15 mod 7 = 1.

Problem C. Crusaders Quest

Description

Crusaders Quest is an interesting mobile game. A mysterious witch has brought great darkness to the game world, and the only hope for your kingdom is to save the Goddesses so that they can unleash their power to fight against the witch.

In order to save the game world, you need to choose three heroes to fight for victory and use their skills wisely. Nine skill blocks of three different types (three blocks per type) will be presented at the bottom of the screen. If k ($k \geq 1$) consecutive blocks are of the same type, you can tap on them and eliminate them, thus triggering the powerful skill they represent. After the elimination, the blocks to their left will be connected with the blocks to their right. Moreover, if $k = 3$ consecutive blocks of the same type are eliminated, the powerful skill they unleash will be upgraded to a super skill, which is the most powerful skill of all.

DreamGrid is a newbie in this game, and he wants to trigger the super skill as many times as he can. Given nine skill blocks satisfying the description above, please help DreamGrid calculate the maximum number of times he can trigger the super skill.

Input

There are multiple test cases. The first line of input contains an integer T (about 50), indicating the number of test cases. For each test case:

The first line contains a string s ($|s| = 9$) consisting of three 'g's, three 'a's and three 'o's, representing the nine skill blocks of three different types. Each type of character represents one type of skill block.

Output

For each test case, output an integer denoting the maximum number of times DreamGrid can trigger the super skill.

Example

standard input	standard output
7 gggaaaooo aaoogggoa googgaaao agogaooag goooggaaa gogogoaaa gaogaogao	3 3 2 1 3 2 1

Note

For the first sample test case, DreamGrid can first eliminate "aaa" (one super skill triggered), thus changing the skill blocks to "gggooo". He can then eliminate "ggg" (another super skill triggered) and finally eliminate "ooo" (a third super skill triggered). So the answer is 3.

For the second sample test case, DreamGrid can first eliminate "ggg" (one super skill triggered), thus changing the skill blocks to "aaoooa". He can then eliminate "ooo" (another super skill triggered) and finally eliminate "aaa" (a third super skill triggered). So the answer is also 3.

For the third sample test case, DreamGrid can first eliminate "aaa" (one super skill triggered), thus changing the skill blocks to "googgo". He can then eliminate "oo" to obtain "gggo", and eliminate "ggg" (another super skill triggered) to obtain "o". So the answer is 2. It is easy to prove that he cannot trigger the super skill three times under this arrangement of skill blocks.

Problem D. Graph Generator

Description

DreamGrid has made a graph generator. The generator will take in a single integer n and generate an undirected graph with n vertices.

More specifically, the generator will generate an empty graph G and a permutation p of $\{1, 2, \ldots, n\}$. After that, the generator will do the following operations n times and during the q-th operation:

1. find the all connected components $C_1, C_2, \ldots, C_m$ in G.
2. choose a subset S_q of $\bigcup_{i=1}^{m} C_i$ such that no two vertices in S_q belong to the same connected component.
3. create a new vertex with index p_q in G and add an edge between p_q and every vertex v in $\bigcup_{i \in S_q} C_{bel_i}$, where bel_i is the index of connected component which i belongs to.

Given the final generated graph, DreamGrid would like to know all the p_q and S_q.

Input

There are multiple test cases. The first line of input contains an integer T, indicating the number of test cases. For each test case:

The first line contains two integers n and m ($1 \le n \le 10^5, 0 \le m \le \min(10^5, \frac{n(n-1)}{2})$) – the number of vertices and the number of edges.

Each of the next m lines contains two integers u_i and v_i ($1 \le u_i, v_i \le n$). There will be no self loops or multiple edges.

It is guaranteed that neither the sum of all n nor the sum of all m exceeds 2×10^6.

Output

For each test case, if the the graph cannot be generated by the generator, output "No" in the first line. Otherwise, output "Yes" in the first line. Then in the i-th of the following line, output two integers q_i and s_i ($1 \le q_i \le n$) followed with s_i integers: $a_1, a_2, \ldots, a_{s_i}$ ($1 \le a_j \le n$) – the index of the newly created vertex and the subset during the i-th operation. If there are multiple solutions, print any of them.

Example

standard input	standard output
3 3 0 4 4 1 2 2 3 3 4 2 4 5 5 1 2 2 3 3 4 4 5 2 4	Yes 1 0 2 0 3 0 Yes 1 0 4 0 3 1 4 2 2 1 3 No

Problem E. String of CCPC

Description

BaoBao has just found a string s of length n consisting of 'C' and 'P' in his pocket. As a big fan of the China Collegiate Programming Contest, BaoBao thinks a substring $s_i s_{i+1} s_{i+2} s_{i+3}$ of s is "good", if and only if $s_i = s_{i+1} = s_{i+3} =$ 'C', and $s_{i+2} =$ 'P', where s_i denotes the i-th character in string s. The value of s is the number of different "good" substrings in s. Two "good" substrings $s_i s_{i+1} s_{i+2} s_{i+3}$ and $s_j s_{j+1} s_{j+2} s_{j+3}$ are different, if and only if $i \neq j$.

To make this string more valuable, BaoBao decides to buy some characters from a character store. Each time he can buy one 'C' or one 'P' from the store, and insert the character into any position in s. But everything comes with a cost. If it's the i-th time for BaoBao to buy a character, he will have to spend $i - 1$ units of value.

The final value BaoBao obtains is the final value of s minus the total cost of all the characters bought from the store. Please help BaoBao maximize the final value.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \leq n \leq 2 \times 10^5$), indicating the length of string s.

The second line contains the string s ($|s| = n$) consisting of 'C' and 'P'.

It's guaranteed that the sum of n over all test cases will not exceed 10^6.

Output

For each test case output one line containing one integer, indicating the maximum final value BaoBao can obtain.

Example

standard input	standard output
3 3 CCC 5 CCCCP 4 CPCP	1 1 1

Note

For the first sample test case, BaoBao can buy one 'P' (cost 0 value) and change s to "CCPC". So the final value is 1 - 0 = 1.

For the second sample test case, BaoBao can buy one 'C' and one 'P' (cost 0 + 1 = 1 value) and change s to "CCPCCPC". So the final value is 2 - 1 = 1.

For the third sample test case, BaoBao can buy one 'C' (cost 0 value) and change s to "CCPCP". So the final value is 1 - 0 = 1.

It's easy to prove that no strategies of buying and inserting characters can achieve a better result for the sample test cases.

Problem F. Getting Lost

Description

BaoBao is lost on an infinite two-dimensional plane! Starting from (a_x, a_y), BaoBao has to find his way to his house located at (b_x, b_y). But this is not an easy task for him, as there are n round obstacles scattering on the plane. The center of the i-th obstacle, whose radius is r_i, is located at (x_i, y_i). BaoBao is free to move on the plane, but his route can not intersect with (but can be tangent to) any obstacle.

We say BaoBao has successfully arrived at his house, if and only if the distance between the final position of BaoBao and (b_x, b_y) is not greater than R, and the segment connecting the final position of BaoBao and (b_x, b_y) does not intersect with (but can be tangent to) any obstacle.

BaoBao is eager to go home, so your task is to help BaoBao find the shortest route which starts from (a_x, a_y) and leads to his successful arrival to his house.

Input

There are multiple test cases. The first line of the input contains an integer T (about 60), indicating the number of test cases. For each test case:

The first line contains one integer n ($0 \le n \le 2$), indicating the number of obstacles on the plane.

The second line contains two integers a_x and a_y ($-10^4 \le a_x, a_y \le 10^4$), indicating the starting point of BaoBao.

The third line contains three integers b_x, b_y ($-10^4 \le b_x, b_y \le 10^4$) and R ($0 \le R \le 10^4$), indicating the position of BaoBao's house and the distance mentioned in the description.

The following n lines each contains two integers x_i, y_i ($-10^4 \le x_i, y_i \le 10^4$) and r_i ($0 < r_i \le 10^4$), indicating the center and the radius of an obstacle.

It's guaranteed that both (a_x, a_y) and (b_x, b_y) are outside of or on the boundaries of the obstacles, and the obstacles do not contain, intersect with or be tangent to each other.

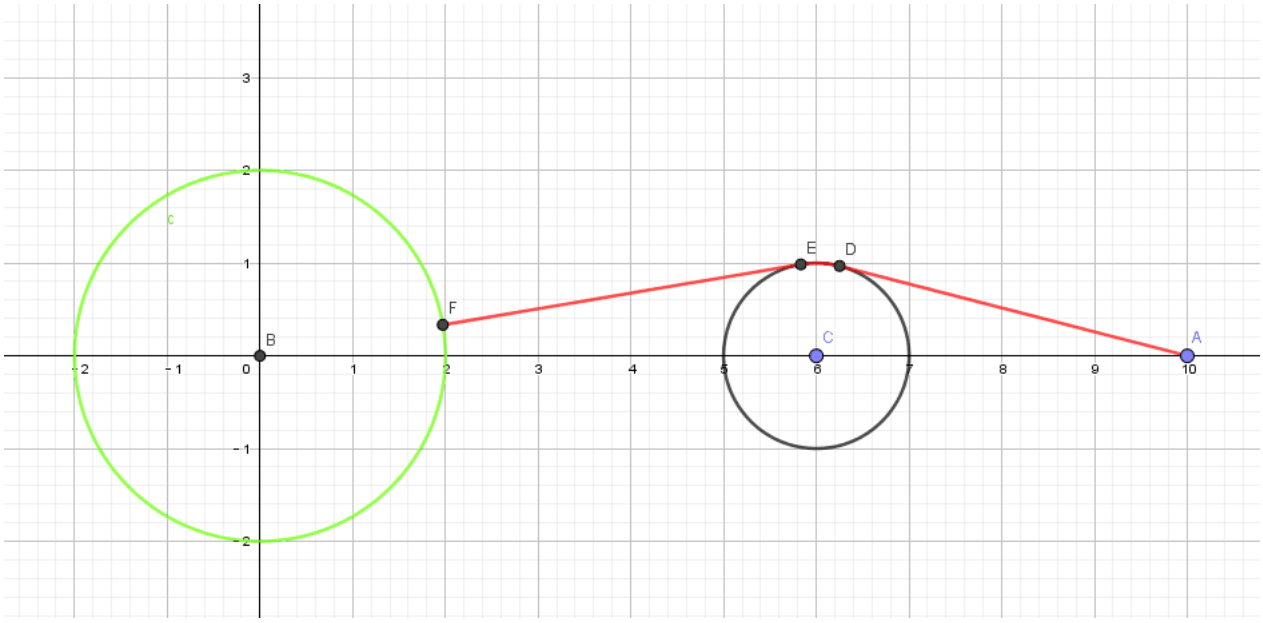
Output

For each test case output one line, indicating the shortest distance for BaoBao to arrive home. Your answer will be considered correct if and only if the absolute error or relative error of your answer is less than 10^{-6}.

Example

standard input	standard output
2 1 10 0 0 0 2 6 0 1 0 0 10 0 0 5	8.209191463668802 5.000000000000000

Note



The image above illustrates the first sample test case, where

$$|AD| = 3.8729833462 \qquad |\overset{\frown}{DE}| = 0.4201283344 \qquad |EF| = 3.9160797831$$

so the answer is $|AD| + |\overset{\frown}{DE}| + |EF| = 8.2091914637$.

Problem G. Numbers

Description

DreamGrid has a nonnegative integer n. He would like to divide n into m nonnegative integers $a_1, a_2, \ldots, a_m$ and minimizes their bitwise or (i.e. $n = a_1 + a_2 + \cdots + a_m$ and a_1 OR a_2 OR ... OR a_m should be as small as possible).

Input

There are multiple test cases. The first line of input contains an integer T, indicating the number of test cases. For each test case:

The first line contains two integers n and m ($0 \leq n < 10^{1000}, 1 \leq m < 10^{100}$).

It is guaranteed that the sum of the length of n does not exceed 20000.

Output

For each test case, output an integer denoting the minimum value of their bitwise or.

Example

standard input	standard output
5 3 1 3 2 3 3 10000 5 1244 10	3 3 1 2000 125

Problem H. Prime Set

Description

Given an array of n integers $a_1, a_2, \ldots, a_n$, we say a set $\{i, j\}$ is a prime set of the given array, if $i \neq j$ and $a_i + a_j$ is prime.

BaoBao has just found an array of n integers $a_1, a_2, \ldots, a_n$ in his pocket. He would like to select at most k prime set of that array to maximize the size of the union of the selected sets. That is to say, to maximize $|\bigcup_{i=1}^{m} p_i|$ by carefully selecting m and $p_1, p_2, \ldots, p_m$, where $m \leq k$ and p_i is a prime set of the given array. Please help BaoBao calculate the maximum size of the union set.

Input

There are multiple test cases. The first line of the input is an integer T, indicating the number of test cases. For each test case:

The first line contains two integers n and k ($1 \leq n \leq 3 \times 10^3$, $0 \leq k \leq \frac{n(n-1)}{2}$), their meanings are described above.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^6$), indicating the given array.

It's guaranteed that the sum of n over all test cases will not exceed 10^4.

Output

For each test case output one line containing one integer, indicating the maximum size of the union of at most k prime set of the given array.

Example

standard input	standard output
4 4 2 2 3 4 5 5 3 3 4 12 3 6 6 3 1 3 6 8 1 1 1 0 1	4 3 6 0

Note

For the first sample test case, there are 3 prime sets: {1, 2}, {1, 4} and {2, 3}. As $k = 2$, we can select {1, 4} and {2, 3} to get the largest union set {1, 2, 3, 4} with a size of 4.

For the second sample test case, there are only 2 prime sets: {1, 2} and {2, 4}. As $k = 3$, we can select both of them to get the largest union set {1, 2, 4} with a size of 3.

For the third sample test case, there are 7 prime sets: {1, 3}, {1, 5}, {1, 6}, {2, 4}, {3, 5}, {3, 6} and {5, 6}. As $k = 3$, we can select {1, 3}, {2, 4} and {5, 6} to get the largest union set {1, 2, 3, 4, 5, 6} with a size of 6.

Problem I. Triangulation

Description

DreamGrid has a point set P of n points. The points are labeled from 1 to n.

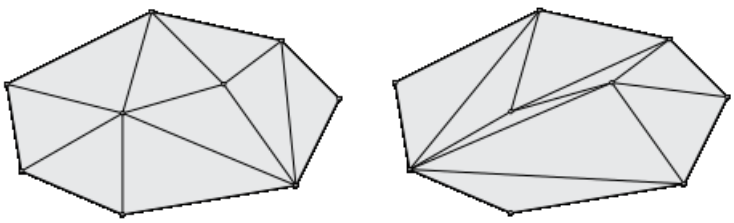
He would like to draw some segments between some pairs of points such that the final result forms a triangulation. The cost for drawing segment between points u and v is $w_{u,v}$.

DreamGrid would like to know the minimum total cost and the number of triangulations which can achieve the minimum total cost.

A triangulation of a point set P is a collection $\mathcal{T}$ of triangles, such that

1. $\text{conv}(P) = \bigcup_{T \in \mathcal{T}} T$, where $\text{conv}(P)$ is the convex hull of P.
2. $P = \bigcup_{T \in \mathcal{T}} V(T)$, where $V(T)$ is the set of three vertices of triangle T.
3. For every distinct pair $T, U \in \mathcal{T}$, $T \cap U$ is either a common vertex, or a common edge, or empty.

For example, the following are two different triangulations of the same set of 9 points.



From Wikipedia. https://en.wikipedia.org/wiki/Point_set_triangulation

Input

There are multiple test cases. The first line of input contains an integer T (about 70), indicating the number of test cases. For each test case:

The first line contains an integer n ($3 \le n \le 18$) – the number of points.

Each of the next n lines contains two integers x_i and y_i ($0 \le x_i, y_i \le 10^6$), denoting the coordinates of the i-th point. No three points lie on the same line.

The i-th of the next n lines contains n integers $w_{i,1}, w_{i,2}, \ldots, w_{i,n}$ ($0 \le w_{i,j} \le 10^6, w_{i,i} = 0, w_{i,j} = w_{j,i}$), indicating the cost for drawing segments.

Output

For each test case, output two integers denoting the minimum cost and the number of triangulations.

Example

standard input	standard output
2 4 0 0 1 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 4 0 0 3 0 1 3 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0	5 2 6 1

Problem J. Tree Equation

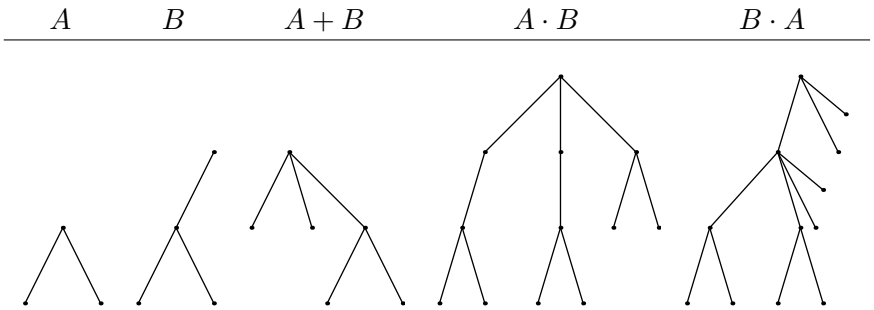
Description

Tired of solving mathematical equations, DreamGrid starts to solve equations related to rooted trees.

Let A and B be two two arbitrary rooted trees and $r(T)$ denotes the root of T. DreamGrid has defined two basic operations:

- **Addition.** $T = A + B$ is built by merging the two roots $r(A)$, $r(B)$ into a new root $r(T)$. That is the subtrees of A and B (if any) become the subtrees of $r(T)$.
- **Multiplication.** $T = A \cdot B$ is built by merging $r(B)$ with each vertex $x \in A$ so that all the subtrees of $r(B)$ become new subtrees of x.

The following picture may help you understand the operations.



Given three rooted trees A, B and C, DreamGrid would like to find two rooted trees X and Y such that $A \cdot X + B \cdot Y = C$.

Input

There are multiple test cases. The first line of input contains an integer T, indicating the number of test cases. For each test case:

The first line contains three integers n_a, n_b and n_c ($2 \le n_a, n_b \le n_c \le 10^5$) – the number of vertices in rooted tree A, B and C, respectively.

The second line contains n_a integers $a_1, a_2, \ldots, a_{n_a}$ ($0 \le a_i < i$) – where a_i is the parent of the i-th vertex in tree A.

The third line contains n_b integers $b_1, b_2, \ldots, b_{n_b}$ ($0 \le b_i < i$) – where b_i is the parent of the i-th vertex in tree B.

The fourth line contains n_c integers $c_1, c_2, \ldots, c_{n_c}$ ($0 \le c_i < i$) – where c_i is the parent of the i-th vertex in tree C.

Note that if $a_i = 0$ ($b_i = 0$ or $c_i = 0$), then the i-th vertex is the root of the tree A (B or C).

It is guaranteed that the sum of all n_c does not exceed 2×10^6.

Output

For each test case, if you can not find a solution, output "Impossible" (without quotes) in the first line.

Otherwise, output two integers n_x and n_y ($1 \le n_x, n_y \le 10^5$) denoting the number of vertices in rooted tree X and Y in the first line.

Then in the second line, output n_x integers $x_1, x_2, \ldots, x_{n_x}$ ($0 \le x_i < i$) – where x_i is the parent of the i-th vertex in tree X.

Then in the third line, output n_y integers $y_1, y_2, \ldots, y_{n_y}$ $(0 \leq y_i < i)$ – where y_i is the parent of the i-th vertex in tree Y.

If there are multiple solutions, print any of them.

Example

standard input	standard output
2 2 3 10 0 1 0 1 2 0 1 1 3 4 3 6 3 1 9 4 3 10 0 1 2 2 0 1 2 0 1 1 3 4 3 6 3 1 9	Impossible 2 1 0 1 0

Problem K. Diversity and Variance

Description

Helianthuswolf Co., Ltd. is a multinational "Intestnet" company. Its revenue from global markets grew year by year. Helianthuswolf's culture is the most important key to its success.

What is Helianthuswolf's culture? Helianthuswolf values innovation, passion, quality, collaboration, and diversity. As Helianthuswolf has grown into a giant successful multinational company and has a huge growing number of global offices, now it values diversity the most.

Anyway, the giant ship is not always smooth sailing. Helianthuswolf has recently suffered a financial crisis. It decides to lay off to survive the crisis. Each department has to lay off a certain number of programmers. To combat age discrimination and to respect the diversity culture, Helianthuswolf decides to keep age diversity after the layoff.

But the recruiters in Helianthuswolf are exaggerated and stereotyped slightly. They need a quantifiable formula to measure diversity. After reading the math book, they think variance is the best measure of diversity. So they want to maximize the variance of the ages of the remaining programmers after the layoff.

In probability theory and statistics, variance is the expectation of the squared deviation of a random variable from its mean. The variance of $\{x_1, x_2, ..., x_k\}$ is $\frac{1}{k}\sum_{i=1}^{k}(x_i - \frac{1}{k}\sum_{j=1}^{k}x_j)^2$.

Please help the recruiters to write a program to decide which programmers will be laid off.

Input

There are multiple test cases. The first line of input contains an integer T, indicating the number of test cases. For each test case:

The first line contains two integers n and m ($1 \le n \le 10^5$, $0 \le m < n$), indicating the number of programmers in that team and the number of programmers the company will lay off.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^4$), indicating the age of each programmer.

It is guaranteed that the sum of n over all the cases is less than 2×10^6.

Output

For each test case, output $n-m$ integers in one line in increasing order. The $n-m$ integers are the indices of the remaining programmers. If there are multiple solutions maximizing the variance, please choose the lexicographically smallest one. DO NOT output extra spaces at the end of each line.

Sequence $\{x_1, x_2, \ldots, x_p\}$ is lexicographically smaller than sequence $\{y_1, y_2, \ldots, y_q\}$, if either $p < q$ and $x_i = y_i$ for all $1 \le i \le p$, or there exists an integer r ($r < p$, $r < q$) such that $x_i = y_i$ for all $1 \le i \le r$ and $x_{r+1} < y_{r+1}$.

Example

standard input	standard output
9 3 2 34 35 36 4 2 20 35 41 74 5 2 40 30 50 20 10 5 2 50 40 30 20 10 5 2 10 20 30 40 50 5 3 30 40 20 10 50 5 4 10 20 30 40 50 5 1 30 50 40 20 10 5 0 30 50 40 20 10	1 1 4 1 3 5 1 2 5 1 2 5 4 5 1 2 3 4 5 1 2 3 4 5

Note

In the first case, laying any two programmers off does not change variance, so we have three candidate answers "1", "2" and "3". Among the three answers, "1" has the smallest lexicographical order. So we output "1" (without quotes).

In the second case, laying a 35-year-old (the 2nd one) and a 41-year-old (the 3rd one) programmer off maximises the variance. As "1 4" is lexicographically smaller than "4 1", we output "1 4" (without quotes).

In the third case, we have 12 candidate answers "1 3 5", "1 5 3", "3 1 5", "3 5 1", "5 1 3", "5 3 1", "3 4 5", "3 5 4", "4 3 5", "4 5 3", "5 3 4" and "5 4 3". As "1 3 5" is the lexicographically smallest one, we output "1 3 5" (without quotes).

Problem L. One-Dimensional Maze

Description

BaoBao is trapped in a one-dimensional maze consisting of n grids arranged in a row! The grids are numbered from 1 to n from left to right, and the i-th grid is marked with a character s_i, where s_i is either 'L' or 'R'.

Starting from the m-th grid, BaoBao will repeatedly take the following steps until he escapes the maze:

- If BaoBao is in the 1st grid or the n-th grid, then BaoBao is considered to arrive at the exit and thus can escape successfully.
- Otherwise, let BaoBao be in the t-th grid. If $s_t =$ 'L', BaoBao will move to the $(t-1)$-th grid; If $s_t =$ 'R', Baobao will move to the $(t+1)$-th grid.

Before taking the above steps, BaoBao can change the characters in some grids to help himself escape. Concretely speaking, for the i-th grid, BaoBao can change s_i from 'L' to 'R', or from 'R' to 'L'.

But changing characters in grids is a tiring job. Your task is to help BaoBao calculate the minimum number of grids he has to change to escape the maze.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains two integers n and m ($3 \le n \le 10^5$, $1 < m < n$), indicating the number of grids in the maze, and the index of the starting grid.

The second line contains a string s ($|s| = n$) consisting of characters 'L' and 'R'. The i-th character of s indicates the character in the i-th grid.

It is guaranteed that the sum of n over all test cases will not exceed 10^6.

Output

For each test case output one line containing one integer, indicating the minimum number of grids BaoBao has to change to escape the maze.

Example

standard input	standard output
3 3 2 LRL 10 4 RRRRRRRLLR 7 4 RLLRLLR	0 2 1

Note

For the first sample test case, BaoBao doesn't have to change any character and can escape from the 3rd grid. So the answer is 0.

For the second sample test case, BaoBao can change s_8 to 'R' and s_9 to 'R' and escape from the 10th grid. So the answer is 2.

For the third sample test case, BaoBao can change s_4 to 'L' and escape from the 1st grid. So the answer is 1.

Problem M. Safest Buildings

Description

PUBG is a multiplayer online battle royale video game. In the game, up to one hundred players parachute onto an island and scavenge for weapons and equipment to kill others while avoiding getting killed themselves. BaoBao is a big fan of the game, but this time he is having some trouble selecting the safest building.

There are n buildings scattering on the island in the game, and we consider these buildings as points on a two-dimensional plane. At the beginning of each round, a circular safe area whose center is located at (0, 0) with radius R will be spawned on the island. After some time, the safe area will shrink down towards a random circle with radius r ($r \leq R$). The whole new safe area is entirely contained in the original safe area (may be tangent to the original safe area), and the center of the new safe area is uniformly chosen within the original safe area.

The buildings covered by the new safe area is called the safe buildings. Given the radius of the safe areas and the positions of the buildings, BaoBao wants to find all the buildings with the largest probability to become safe buildings.

Input

There are multiple test cases. The first line of input contains an integer T, indicating the number of test cases. For each test case:

The first line contains three integers n ($1 \leq n \leq 100$), R and r ($1 \leq r \leq R \leq 10^4$), indicating the number of buildings and the radius of two safe circles.

The following n lines each contains 2 integers x_i and y_i ($-10^4 \leq x_i, y_i \leq 10^4$), indicating the coordinate of the buildings. Here we assume that the center of the original safe circle is located at $(0, 0)$, and all the buildings are inside the original circle.

It's guaranteed that the sum of n over all test cases will not exceed 5000.

Output

For each test case output two lines.

The first line contains an integer m, indicating the number of buildings with the highest probability to become safe buildings.

The second line contains m integers separated by a space in ascending order, indicating the indices of safest buildings.

Please, DO NOT output extra spaces at the end of each line.

Example

standard input	standard output
2 3 10 5 3 4 3 5 3 6 3 10 4 -7 -6 4 5 5 4	1 1 2 2 3