Problem A. Algorithmia's Sky

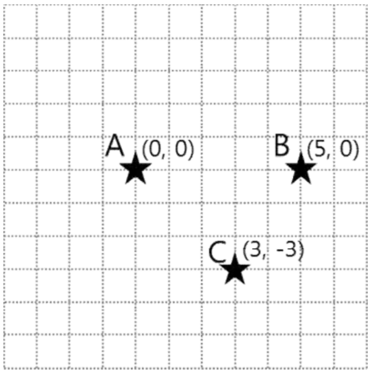
Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 second
Memory limit:	512 mebibytes

There are many stars and planets that brightly shine in the night sky on Algorithmia planet.

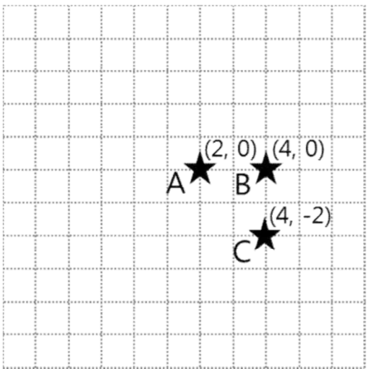
Consider astronomers making a photo of the sky every night at midnight with a fixed camera; then stars and planets can be represented as points with integer coordinates in a two-dimensional plane.

Comparing photos from two subsequent days, its easy to see that stars are stable (and have the same coordinates all the time), and planets are moving with constant speed (d_x, d_y) where d_x is the x-axis coordinate change during the day, d_y is the y-axis coordinate change during the day. The speed of each object is independent of other object, and objects can overlap each other at any time. The speed of the star is considered to be equal to $(0, 0)$.

For example, in the picture below the coordinates of planet A are (0, 0), the coordinates of planet B are (5, 0) and the coordinates of planet C are (3, -3).



The next picture was taken one day later. The coordinates of the three planets were changed to (2, 0), (4, 0), and (4, -2), respectively.



Thus, the speed of planet A is (2, 0), the speed of planet B is (-1, 0), and the speed of planet C is (1, 1). Therefore, the coordinates of three planets two days later will be (4,0), (3,0), (5,1), respectively.

For each day, the maximum D of squared distance between two objects at the picture taken at this day is recorded.

You will given initial coordinates and speed of all objects. Calculate minimum value of D among the first t days.

Input

The first line of the input contains two integers n ($2 \le n \le 3 \cdot 10^4$) — the number of objects on the sky and t — the day when last picture was taken ($1 \le t \le 10^7$). Each of the next n lines contains four integers x, y, dx, dy ($-10^7 \le x, y \le 10^7$, $-100 \le dx, dy \le 100$) — initial coordinates of the next object and its speed, respectively. Note that two objects may coincide.

Output

In the first line print one integer — the earliest date when minimum D was recorded. The second line must contain the value of D at this day.

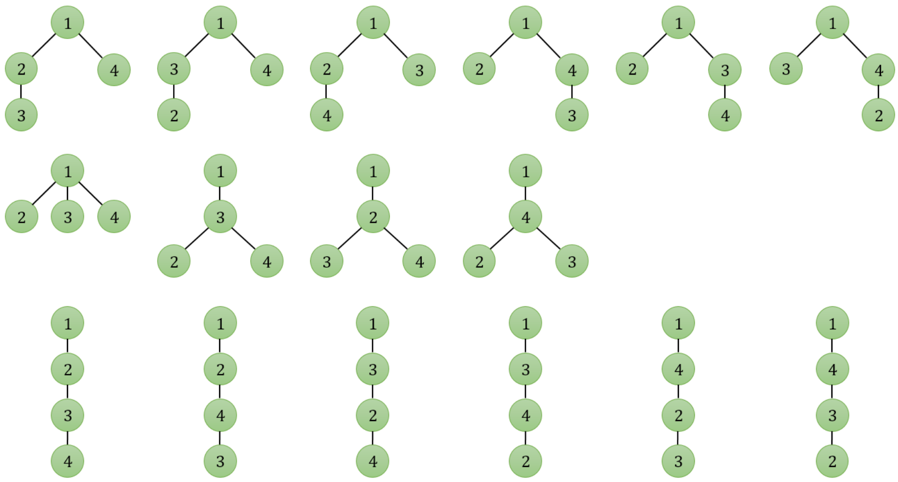
Example

standard input	standard output
3 3 0 0 2 0 5 0 -1 0 3 -3 1 1	2 5
2 1 -4 -2 2 1 4 2 -2 -1	1 20

Problem B. Build The Trees

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

There are n vertices labeled by sequential integers between 1 and n. Consider all trees connecting all of these vertices. We will say that two trees differ if there exists at least one pair of vertices, which is connected directly in one of the trees and is not connected in another.



On the picture above all distinct trees for $n = 4$ are listed.

There is also a set of m edges. Calculate the number of distinct trees on n labeled vertices, containing all those edges modulo $10^9 + 7$.

Input

The first line of the input contains two integers n and m ($1 \le n \le 10^8$, $0 \le m \le 10^5$) — the number of labeled vertices in a tree and the number of pre-determined edges. Then m lines follow, each containing two distinct integers between 1 and n, inclusively — the numbers of vertices connected by the respective edge.

It is not guaranteed that a tree with all given edges exists; if such a tree does not exist, output zero as the answer.

Output

Print one integer — the number of distinct trees with n labeled vertices, containing all the given edges, modulo $10^9 + 7$.

Example

standard input	standard output
4 0	16
4 1 1 2	8

Problem C. Contest On TV

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

"Pop Contest" is a Byteotian TV program. In this program, n contestants compete with each other to get selected to the ByteVision national team, which consists of k contestants.

In the contest, there are two selection tests. In the first test, contestants dance to pop music. In the second test, contestants sing pop music.

All tests are done now, and for each i contestant i scored a_i points in first test, and b_i points in second test. Their final score is $a_i + b_i$.

Until now only the first test was aired on TV. So people only know the sequence a. However, Bytica leaked a sequence c to an internet bulletin, which is a shuffled version the sequence b (second test scores).

Fans want their ByteVision national team to be great dancers, so they hope the final scores to not differ much. Formally, fans "like" the scores, if there is no top-k contestant by the results of the first test, who got lower final score than any of the non-top-k contestants after the first test. As the fans don't know the array b, they will try all n-number permutations p to find b.

Formally, they will let $b_i = c_{p_i}$, and check if they like that permutation or not.

You should calculate the number of permutations liked by fans, modulo $10^9 + 7$.

Input

In the first line, two integers n and k are given ($1 \le k \le n \le 100$). In the second line, the array $a_1, \ldots, a_n$ is given ($1 \le a_i \le 1000$, all a_i are pairwise distinct). In the third line, the array $c_1, \ldots c_n$ is given ($1 \le c_i \le 1000$, all c_i are pairwise distinct).

Output

Print the answer to the problem modulo $10^9 + 7$.

Examples

standard input	standard output
4 2 1 3 2 4 7 3 5 1	7
5 4 200 1000 400 800 600 1 5 4 2 3	120
5 4 1 5 4 2 3 200 1000 400 800 600	24
6 3 84 95 70 75 100 68 91 87 89 61 55 50	68

Problem D. Damage

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

In a popular roleplaying game a hero has n battle skills, indexed 1 through n. If a hero invokes skill i, it connects with probability p_i and deals d_i damage in this case, otherwise it fails to do anything.

When a hero is fighting a monster, the skills in one round of battle are invoked automatically following an algorithm:

1. If there are skills that were not yet invoked during this round, one of those skills is chosen equiprobably for invocation, otherwise the hero ends this round of battle without doing any damage;

2. If the invoked skill j connects successfully (i.e. with probability p_j), then the hero does d_j damage to the monster and ends this round of battle; otherwise step 1 repeats and so on.

Calculate the expected amount of damage done to the monster after a single round of battle.

Input

The first line of the input contains integer n ($1 \le n \le 5000$), which represents the number of skills.

The next n lines contain information about the skills; i-th of those lines contains two integers p'_i, d_i ($1 \le p'_i, d_i \le 10^9$). p'_i is equal to $p_i \cdot 10^9$, where p_i is the probability that the skill i connects upon invocation; d_i is the amount of damage inflicted upon the opponent in case the skill connects.

Output

Output expected damage done to monster at one round of battle. If the answer is in the form of an irreducible fraction p/q, (that is, p and q are integers, $q > 0$, and $(p, q) = 1$), then print $(p \cdot q^{-1}) \bmod (10^9 + 7)$. It is guaranteed that $q^{-1} \bmod (10^9 + 7)$ is well-defined.

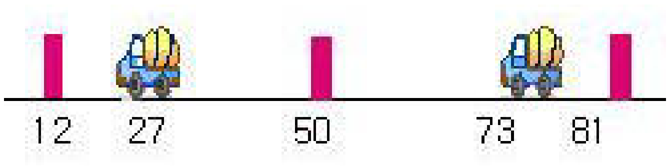
Example

standard input	standard output
1 500000000 10	5
9 1 9 2 8 3 7 4 6 5 5 6 4 7 3 8 2 9 1	93531355

Problem E. Engines And Pumps

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

There are several fire pumps on the straight line. Several fire engines stopped at the same line to fill the water. The number of pumps is equal to or greater than the number of fire engines.



The figure above shows that two fire engines (positions 27 and 73) were stopped between the three pumps (positions 12, 50 and 81).

To fill the tank of fire engine, you need connect the pump and the fire truck by the hosepipe. To save time, all fire engines are trying to fill the tanks at the same time. Only one fire engine can be connected to each pump. The length of the hosepipe used for that is the distance between the pump and the fire engine.

In the figure, the first fire engine is connected to the first pump (hosepipe length is 15), the second fire truck is connected to the third pump (hosepipe length is 8), so the sum of hosepipe lengths used is $23 = 15 + 8$, which minimizes the sum of the hosepipe lengths.

Given the location of the pumps and the location of the fire engines, write a program which calculates minimum total length of hosepipe needed to connect the pumps to the fire engines.

Input

First line of the input contains two integers: number of pumps p and number of fire engines f, respectively ($1 \le p \le 10^5$, $1 \le f \le 10^5$, $f \le p$). The second line consists of p distinct integers given in the ascending order — the coordinates of the pumps. The third line consists of f distinct integers given in ascending order — the coordinates of the fire engines. A pump and a fire engine may share the location. Coordinates are positive integers less than 10^6.

Output

Print minimum sum of the lengths of the hosepipes needed to fill all the tanks. You may assume that it does not exceed $2^{31} - 1$.

Example

standard input	standard output
3 2 12 50 81 27 73	23

Problem F. Funny Lottery

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Bob has r red marbles, g green marbles, and b blue marbles. He wants to play following game alone:

First, he puts all $r + g + b$ marbles in a lottery wheel. If he pushes a button on this lottery wheel, it will pick one of the marbles in it uniformly random, and spit that marble to him.

He will push the button until the lottery wheel spits a blue marble k times. Every time he gets a marble, he will do the following:

1. If he gets a red marble, he does nothing.
2. If he gets a green marble or a blue marble, he puts that marble in the lottery wheel again.

Find the expected value of times he pushes the button until the lottery wheel spits a blue marble K times.

Input

The first line contains one integer t ($1 \le t \le 10^3$) — the number of test cases.

Each test case consists of four integers r, g, b, and k ($1 \le r, g, b, k \le 10^9$) — the number of red marbles, the number of green marbles, the number of blue marbles, and the number of blue marbles he should obtain in order to terminate the process.

Output

For each test case, print the expected value per line. If the answer is an irreducible fraction a/b, print $(a \cdot b^{-1}) \bmod 10^9 + 7$. It is guaranteed that b^{-1} always exists.

Examples

standard input	standard output
4 1 1 1 1 1 2 3 4 1000 1000 1 1000 50000 50000 50000 10000000	500000006 569010428 490804548 595034885
4 9 9 9 1 9 9 9 2 9 9 9 4 9 9 9 1000	300000005 470000009 700700016 814176661

Problem G. Giant Aquarium

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

On Figure 1 the giant aquarium seen from the front is shown. This aquarium is full of water. If you drill a hole in the bottom of the aquarium (the horizontal line), the water in the aquarium will be released through the hole.

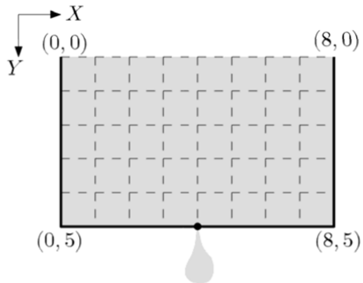


그림 1. 수족관과 구멍.

As shown in Figure 1, the X-axis increases from left to right, and the Y-axis increases from top to bottom. If there is a hole in the horizontal line representing the floor, all the water in a position that is equal to or smaller than the y-coordinate of the horizontal line and is able to flow into the hole according to gravity is poured out through the hole. Therefore, the water in the aquarium of Figure 1 is completely exhausted.

The amount of water in the aquarium corresponds to the area occupied by the water. The unit of the amount of water is L (liters). Thus, in figure 1, when the aquarium is filled with water, the amount of water is 40L, which is equal to the area occupied by water.

The bottom of the aquarium, as shown in Figure 2, not usually is straight line, but may look complicated.

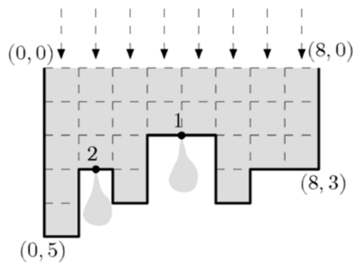


그림 2. 수족관의 처음 상태. 물의 양은
26L이고 구멍은 2개.

The bottom of the aquarium is alternating between the horizontal line and the vertical line. In addition, when the aquarium floor is viewed vertically from above the aquarium, as shown in Figure 2, the bottom of the aquarium is visible (i.e., all horizontal lines are visible).

The hole can be located only in the middle of a horizontal segment. There can be at most one hole per one horizontal segment. Now, let's place two holes on two horizontal segments as shown on the Figure 3. Then the amount of water to be poured out should be maximized. If two holes are placed as shown in Figure 3, only 7L of the 26L, which is the original amount of water, is left in the aquarium and 19L is out.

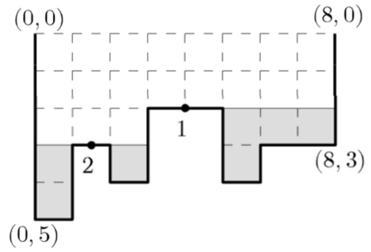


그림 3. 19L가 빠져나가고 7L만 남음.

However, if you drill two holes, as shown in Figure 4, only 1L remains and 25L is out. As a result, when two holes are created, there is a hole arrangement capable of extracting a maximum of 25 L of water.

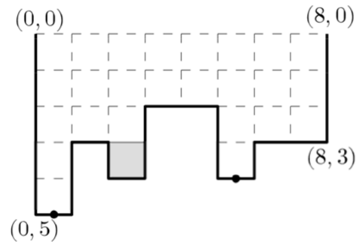


그림 4. 25L가 빠져나가고 1L만 남음.

Given the aquarium, write a program to output the maximum amount of water to be poured out when k holes are arranged so that the maximum amount of water is discharged.

Input

The first line of the input consists of one integer n — the number of vertices of the contour of the aquarium ($4 \le n \le 3 \cdot 10^5$, n is even).

The sides forming the boundary of the aquarium are starting from the vertex (0, 0), starting from the vertical line, alternating between the horizontal line and the vertical line, and ending in a vertical line. Therefore, there is always one more vertical line than the horizontal line. Each of next N lines contain two integers x_i and y_i ($0 \le x_i, y_i \le 10^9$) — coordinates of the vertices of contour listed in counterclockwise direction.

You may assume that the sides forming the boundary of the aquarium are starting from the vertex (0, 0), starting from the vertical line, alternating between the horizontal line and the vertical line, and ending in a vertical line, additionally, $y_n = 0$. Therefore, there is always one more vertical line than the horizontal line.

Last line contains one integer k — the number of holes to be drilled ($1 \le k \le n/2$).

Output

Print one integer — maximum amount of water to be poured out after drilling k holes.

Examples

standard input	standard output
14 0 0 0 5 1 5 1 3 2 3 2 4 3 4 3 2 5 2 5 4 6 4 6 3 8 3 8 0 2	25

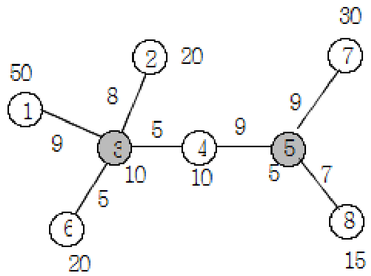
Problem H. Hospital

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

There are n villages numbered from 1 to n and a road network connecting these villages. In this road network, there is only one route from each village to any of the other villages. The number of people living in each village and the time it takes to pass each road are given. There are only two hospitals in two of these villages.

The time taken from village A to village B is the sum of the travel times of the roads on the route from A to B.

For example, consider the following road network:



The circles represent the villages, the number in the circle is the number of the village, and the number outside is the number of people in the village. The line segment represents the road, and the number on the line segment represents the travel time of the road. In addition, the two villages where the hospitals are built are painted in gray.

To improve the medical service in this area it was decided to reduce travel time for some roads. Cost of reducing travel time by C is C, but travel time can't be improved to be less than L regardless of amount of money paid (but some road may already have the travel time less than L). The total budget for the improvement of medical services cannot be greater than B.

There are two principially different ways to improve the medical service. First way is to minimize sum of times t_i for each person, when t_i is time for a person to travel to the nearest hospital. Second way is to minimize maximum time needed for someone to travel to nearest hospital. To compare those ways, the government asked you to calculate both values: the sum in cases when budget was spend in first way, and the longest travel time to hospital in case, when second way will be selected.

Input

First line of the input contains two positive integers B and L — maximum budget for the road improvement and minimum possible travel time for any road, respectively ($1 \le B \le 4 \cdot 10^6$, $1 \le L \le 1000$). Second line contains one integer n — the number of villages ($2 \le n \le 4000$). Third line contains N integers between 1 and 500 inclusively; i-th of those integers denote number of people living in i-th village. Each of next $N - 1$ lines contains three positive integers indicating the information of the road, that is, the numbers a_i and b_i of villages connected by this road and initial travel time t_i ($1 \le a_i, b_i \le N$, $1 \le t_i \le 1000$).

Last line contains two different integers between 1 and N, inclusively — numbers of two villages where a hospital is built.

Output

Print two integers: the result of first way of budget spending in first line, and the result of second way at the second line.

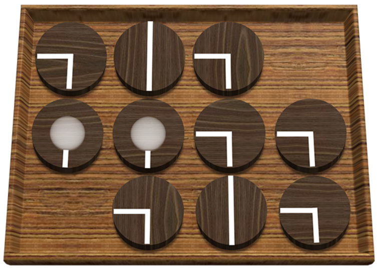
Example

standard input	standard output
7 6 8 50 20 10 10 5 20 30 15 1 3 9 3 2 8 3 4 5 4 5 9 7 5 9 8 5 7 3 6 5 3 5	875 7

Problem I. ICPC and Light Bulbs

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

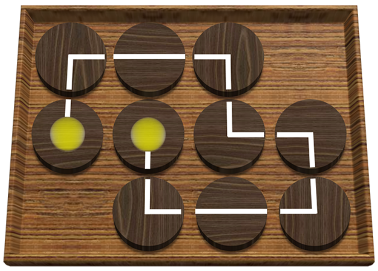
Bill is a member of International Committee of Puzzle Creation. Recently, he has made a puzzle to light all the bulbs on the board. The puzzle looks like the following:



The board consists of n rows and m columns of cells. Each cell contains either a bulb, a wire or nothing. There are exactly two bulbs, and each bulb can be connected with an adjacent cell. There are two types of wires in this puzzle: q-type and l-type. A q-type wire connects two cells that share a vertex, and an l-type wire connects two cells that share an edge.

In order to light all the bulbs, they should be connected by wires (i.e. there must be a path of wires connecting the two bulbs) It can be done by rotating each cell by 90-degree as many times as you want. Bill thought this was too easy, so he added one more constraint: all the wires must be used to connect the bulbs (i.e. the path of wires connecting two bulbs must consist of all the wires in the board).

The figure below shows an example of a valid solution.



Bill provided this idea to ICPC and got the test version of this puzzle. He tried to solve the puzzle, but he couldn't solve it because the size of puzzle was quite large for him. Why don't you help Bill to solve the puzzle?

Input

The first line of input contains an integer t ($1 \leq t \leq 10$), the number of test cases.

Each test case starts with a line containing two space-separated integers n and m ($1 \leq n, m \leq 500$), where n is the number of rows of the board and m is the number of columns of the board.

Each of the next n lines contains a string of length m, describing the board. The j-th character of the i-th string ($1 \leq i \leq n$, $1 \leq j \leq m$) describes the cell in the i-th row from the top and the j-th column from the left. There are four kinds of characters '`O`' (ASCII code 79), '`q`' (ASCII code 113), '`l`' (ASCII code 108), and '`*`' (ASCII code 42) which describes a cell: '`O`' means a bulb, '`q`' means a q-type wire, '`l`' means an l-type wire, and '`*`' means empty cell.

Output

For each test case:

If there exists a valid solution, first print "YES" (without quotes) in a single line, and then print how a valid solution looks in the next n lines. The i-th ($1 \le i \le n$) line of them should contain exactly m characters. The j-th character of the i-th line ($1 \le i \le n$, $1 \le j \le m$) should be determined as follows:

- If the corresponding cell was a bulb, the character should be one of '^v<>' (ASCII Code 94, 118, 60 and 62, respectively), according to the direction (up, down, left, and right, respectively) to which the bulb is connected.
- If the corresponding cell was a q-type wire, the character should be one of "qdbp" (ASCII Code 113, 100, 98 and 112, respectively), according to the position of two ends (left and down, up and left, right and up, down and right correspondingly) of the wire.
- If the corresponding cell was a l-type wire, the character should be one of "l-" (ASCII Code 108 and 45, respectively), according to the position of two ends (up and down, left and right, respectively) of the wire.
- If the corresponding cell contained nothing, the character should be '*' (ASCII code 42).

If there exists several solutions, it is allowed to print any of them.

If there is no solution, just print "NO" (without quotes) in a single line.

Examples

standard input	standard output
4	YES
3 4	p-q*
qlq*	^vbq
00qq	*b-d
*qlq	NO
2 4	YES
0qqq	*pq*pq**
0qqq	pd^pdbq*
5 8	bq*bqpd*
*qq*qq**	*bqpdbqv
qq0qqqq*	**bd**bd
qq*qqqq*	YES
*qqqqqq0	p-qp--q
qqqq	bqbdp-d
6 7	*b<pdpq
qlqqllq	pqpd*^l
qqqqqlq	lbdp-ql
*q0qqqq	b--d*bd
qqqq*0l	
lqqqlql	
qllq*qq	

Problem J. Jokes from Loki

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Odin made a simple game to teach binary search for students. The game proceeds in following way:

1. First, the computer selects any integer $x \in [1, n]$ randomly, and tells the player the value of n.
2. Next, the player selects some integer q, and asks computer "Is $x \le q$?". The computer correctly responds to player's question as "Yes" or "No". The goal of player is to deduce the value x by making as few questions as possible.
3. If the player deduced the value x, he/she can ask computer "Is $x = q$" for exactly one time. If the value x is q, the game is considered successful, otherwise it is not. After that, the game is immediately terminated.

Odin was proud of his game, and told Loki about it. However, as Loki wanted to troll Odin, he secretly told the optimal strategy of Odin's game to his students. Odin was very angry about it, so he gave an extra problem for Loki.

Odin modified the game: if the computer's value is x, the player should always play a successful game, and also the number of questions should be lower or equal to a_x. Loki should find a strategy for that game regardless of the computer's value x. The sequence a_i is non-decreasing (in other words, $a_1 \le a_2 \le \ldots \le a_n$)

Loki tried to solve his assignment, but failed to find any strategy for it. Loki didn't even figured out how to start the first move — so he is secretly asking for your help!

You should first find out if there is any strategy for successfully playing Odin's modified game. If there is any, you should find all the values of q which Loki can ask for first question and successfully play the game.

Input

First line of the input contains one integer n ($1 \le n \le 10^6$).

Then n integers $a_1, a_2, \ldots, a_n$ follow ($1 \le a_1 \le a_2 \le \ldots \le a_n \le 10^9$).

Output

In line 1, print the number of possible q for first question. If there is infinite number of possible q, you should print "`inf`" instead.

In line 2, print the possible values of q in increasing order. If there is infinite or no possible q, you should print nothing.

Examples

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

Problem K. Kitchen-based Economy

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3.5 seconds
Memory limit:	512 mebibytes

In Byteland, there is a road network with n cities and m roads. Each roads connect two different cities, and there is at most one direct roads connecting two different cities. Every road have a length of 1.

Interestingly, the road network is a form of Cactus graph. Cactus graph is a connected graph, such that no two distinct simple cycles share an edge. In each city, there are a_i citizens, and b_i restaurants.

People in Byteland like restaurants. So every day, each person selects one of the restaurants and visits it, and after that returns to their home. All restaurants in Byteland have equal probabilities to be selected. All choices for different people are independent.

When visiting some restaurants in Byteland, they only use shortest path to visit them. If there is more than one shortest paths, they select one of them with equal probabilities.

The king's advisor Byteazar decided to build toll gates in one of m roads. He wants to build it on a road with maximum expected number of people using that road.

You should find the roads which have maximum expected number of people using it. If there is more than one such roads, you should print all of them.

Input

First line of the input consists of two integers n and m — number of cities and number of roads, respectively ($2 \le n \le 2 \cdot 10^5$, $2 \le m \le 3 \cdot 10^5$).

Each of next n lines contains two integers a_i and b_i ($1 \le a_i, b_i \le 10^4$) — number of citizens and the restaurants in the i-th city, respectively.

Each of next m lines contains two integers s_i and e_i ($1 \le s_i < e_i \le n$) — numbers of cities, connected by a road.

Output

In first line print one integer k — number of roads with maximum expected number of people. Each of the next k lines must contain two integers — cities, connected by the respective optimal road. Print roads in lexicographical order (i.e. sort pairs by increasing of s_i, then by increasing of e_i).

Examples

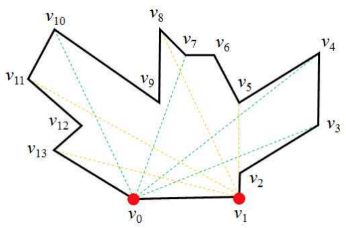
standard input	standard output
5 4 1 1 1 1 1 1 1 1 1 1 1 2 2 3 3 4 4 5	2 2 3 3 4
5 5 1 1 1 1 2 1 2 1 5 1 1 2 1 3 2 4 3 4 4 5	1 4 5
5 6 1 1 1 2 1 1 1 2 1 1 1 2 1 3 1 4 1 5 2 3 4 5	2 1 2 1 4

Problem L. Light Sources

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Given a polygon with n vertices. Denote coordinates of vertice v_i as (x_i, y_i). Then:

- $y_0 = y_1$, $x_0 < x_1$;
- for any $2 \le k \le n-1$ $y_k > y_0$;
- if you place two light sources in v_0 and v_1, whole polygon will be illumniated by at least one of those light sources (a point is illuminated by a light source, placed at some other point, if the segment connecting those points does not intersect the border of the polygon, but may touch it).



It is easy to prove that the shortest path inside the polygon between two vertices is the polyline, connecting some vertices of the polygon.

For given two vertices v_a and v_b build the shortest path inside the polygon between those vertices.

Input

First line of the input contains one integer n $(3 \le n \le 10^5)$— the number of vertices in the polygon. Then n lines follow; i-th of those lines contains two integers x_{i-1} and y_{i-1} — coordinates of v_{i-1} $(-10^9 \le x_i, y_i \le 10^9$, $y_0 = y_1$; $x_0 < x_1$, $y_i > y_1$ if $i > 1)$. The last line consists of two integers a and b $(0 \le a, b \le N-1$, $a \ne b)$ — indices of given vertices.

Output

In first line print one integer l — the number of vertices in the shortest path. In second line print l integers — indices of vertices of shortest path in order they follow in the path; first index must be equal to a, and last — to b. If several answers are possible, choose one with minimum number of vertices, then lexicorgraphically minimal.

Example

standard input	standard output
14 5 2 9 2 9 3 12 5 12 8 9 6 8 8 7 8 6 9 6 6 2 9 1 7 3 5 2 4 4 11	4 4 5 9 11
9 4 2 10 2 12 5 8 4 9 7 6 7 4 5 3 6 2 6 5 2	3 5 3 2